MINISTÈRE DE LÉNSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

BADJI MOKHTAR ANNABA UNIVERSITY

Faculty of Technology

Department of Computer science

Submitted in partial fulfillment of the requirements for Doctorate degree in Computer science

Option: Artificial Intelligence and Image Processing

Presented by **sahib khouloud**

Theme

ROBUST DEEP LEARNING MODELS FOR COMPUTER-AIDED DETECTION APPLIED ON SKIN CANCER

Jury:

President	Souici-Meslati Labiba	Professor	University of Badji Mokhtar Annaba
Director	Melouah Ahlem	Professor	University of Badji Mokhtar Annaba
Co-Director	Touré Fadel	Professor	University of Trois-Rivières Quebec
Examiner	Azizi Nabiha	Professor	University of Badji Mokhtar Annaba
Examiner	FAROU Brahim	Professor.	University of 8 Mai 1945 Guelma

2023-2024

Acknowledgements

Thanks be to the Almighty God for providing me with the faith, bravery, and patience to complete this task work.

I would like to express my heartfelt thanks to my supervisor, Pr. Melouah Ahlem of Badji Mokhtar Annaba University, for the trust she has placed in me, via his constant presence, direction, modesty, counsel, and constructive remarks for the successful completion of this job.

I would like to thank Dr. Touré Fadel of the University of Quebec in Trois-Rivières for co-supervising me, as well as for his orientation, availability, listening, and patience over the duration of this work.

I extend my appreciation to all those who assisted in enhancing my work and provided valuable feedback that contributed to refining this thesis.

I want to express profound thanks to my parents and two brothers for their encouragement and prayers, which played a pivotal role in enabling me to accomplish this humble task. I am truly grateful for the confidence they have shown in me.

The last two persons I want to thank are my husband and my son "Yanis". I would like to thank my husband for his constant presence, his encouragement, and his valuable advice. I need to thank my little boy for being such a bundle of joy and laughter.

Finally, I want to thank everyone who helped this effort along in whatever manner.

ملخص

الميلانوما الخبيثة هي أشد أنواع سرطان الجلد فتكاًا، وتزداد انتشاراً، وخاصة بين الأشخاص ذوي البشرة الفاتحة المعرضة لأشعة الشمس. ويمكن علاج الميلانوما بشكل فعال في مراحلها المبكرة، لذلك فإن الكشف المبكر أمر بالغ الأهمية في زيادة فرص النجاة. يلعب الأخصائيون في أمراض الجلد دوراً هاماًا في الكشف عن الميلانوما من خلال فحص الخصائص السريرية الجلدية المجهرية مثل حدود اللويحات الجلدية والشبكات الصبغية، والتي تشير إلى المرض بشكل أساسي. ومع ذلك، يمكن أن يكون هذا النهج صعباًا ومستهلكاًا للوقت بسبب الاختلافات في حجم ولون اللويحات الجلدية والتباين المنخفض ووجود مكونات مثل الشعر والمستحضرات التجميلية وفقاعات الهواء التي يمكن أن تؤثر على التشخيص الدقيق، حتى بالنسبة للخبراء.
لمعالجة على هذه التحديات، تقوم هذه الأطروحة بتطوير أساليب موثوقة وفعالة لتحليل صور الدرموسكوبي بشكل آلي يمكن أن تساعد في الكشف المبكر عن الميلانوما الخبيثة. وعلى وجه الخصوص، تقدم هذه الأطروحة إطار عمل مقترح لتقسيم وتصنيف الآفات الجلدية تلقائياًا من صورة الإدخال. يهدف الإطار المقترح إلى تحسين دقة وفعالية تحديد وتصنيف الآفات الجلدية، وبخاصة الميلانوما، باستخدام أساليب التعلم العميق.
يتضمن الإطار العمل العديد من المراحل: المعالجة الأولية، وتقسيم آفات الجلد، وتصنيف الآفات. في مرحلة المعالجة الأولية، يتم تطبيق فيلتر جاوسي لتحسين جودة الصورة عن طريق تقليل الأثار مثل الضباب والشعر والعلامات والبقع. كما يتم استخراج منطقة الاهتمام (آفة الجلد) وإزالة الأجزاء غير الضرورية من الصورة من خلال القص التلقائي. لتقسيم آفات الجلد، يتم اقتراح اثنين من أساليب التعلم العميق، وهما ط-نت وψ-نت.
ط-نت هو هيكل مزدوج المشفر-الفك التشفير يستخدم مشفر-فك شبكة ضست، ومشفر-فك شبكة رنت، وشبكة هرم الميزات لإنشاء هيكل شبكة عصبية تحويلية قوي لتقسيم الصور. من ناحية أخرى، يعد ψ-نت بمثابة هيكل تعلم عميق يدمج العديد من التغييرات على شبكة ونت الأساسية لزيادة أدائها في تقسيم الصور. لنسبة لتصنيف الأورام، يستخدم الإطار المقترح شبكة ينتَن-ضسنت، وهي نوع خاص من شبكات العصب العصبي التحويلي التي تقوم بتحَليل الميزات بشكل مستقل. تتكون هذه الشبكة من طبقات التحويل العصبي والتجميع، بالإضافة إلى كتل ينتَن-ضست. المكون الأساسي المسؤول عن التصنيف هو كتلة ينتَن-ضست التي تَجمع بين وحدة ينتَن وشبكة عصبية متبقية، مما يمنح الشبكة اَلمرونة في تعلم الميزات بمقاييسَ مختلفة.
تم تقييم النماذج المقترحة على أربعة مجموعات بيانات عامة: يضير ٦١٠٢ و يضير ٧١٠٢ و يضير ٨١٠٢ و صس٢. حققت نماذج ط-نت و ψ-نت مستويات

دقة عالية (أكثر من ٧٩٪) على جميع قواعد البيانات، متفوقة على الأساليب الحالية وتقدم حساسية وخصوصية ومؤشر جاكارد ومعامل التشابه دايس أفضل. على النقيض، يختلف أداء شبكة يندتن ضست للتصنيف تبعاًا لمجموعة البيانات ومعايير التقييم. ثم تمت مقارنَة النتائج مع الأساليب الحالية في الأدبيات، مما يدل على أن الاستراتيجية المقترحة دقيقة وفعالة في كل من تقسيم وتصنيف حدود الأورام.

كلمات مفتاحية: سرطان الجلد ، الورم الجلدي ، التعلم العميق ، جهاز فك التشفير ، التجزئة ، الشبكة العصبية التلافيفية ، التصنيف.

Abstract

Malignant melanoma, an extremely fatal form of skin cancer, is becoming more common, particularly among individuals with fair skin who are exposed to sunlight. Melanoma may be efficiently treated in its early stages, therefore early detection is crucial for increasing survival chances. Dermatologists play an important role in melanoma detection by examining dermoscopic clinical characteristics such as lesion boundaries and pigment networks, which are essential indications of the illness. However, this approach can be difficult and time-consuming due to differences in lesion size and color, low contrast, and the presence of components such as hair, lubricants, and air bubbles that can interfere with an accurate diagnosis, even for experts.

To address these challenges, this thesis develops reliable and efficient automated dermoscopy image analysis methods that can aid in the early detection of malignant melanoma. In particular, this thesis provides a proposed framework for automatically segmenting and classifying skin lesions from an input image. The proposed framework aims to improve the accuracy and efficiency of identifying and classifying skin lesions, particularly melanoma, using deep learning methods. The framework involves several stages: preprocessing, skin lesion segmentation, and lesion classification. In the preprocessing stage, a Gaussian filter is applied to enhance the image quality by reducing artifacts such as fog, hairs, marks, and stains. Also, the region of interest is extracted, and unnecessary portions of the image are removed through automatic cropping. For skin lesion segmentation, two deep learning methods, W-net, and ψ-net are proposed. W-net is a dual encoder-decoder architecture that uses a ResNet encoder-decoder, a ConvNet encoder-decoder, and a Feature Pyramid network to create a robust CNN architecture for image segmentation. On the other hand, ψ-net is a deep learning architecture that incorporates many changes to the baseline Unet network to increase its segmentation performance. For lesion classification, the proposed framework uses the Inception-Resnet network, a particular form of convolutional neural network that performs feature extraction on its own. This network consists of convolutional and pooling layers, as well as Inception-ResNet blocks. The key component responsible for classification is the Inception-ResNet block, which combines the Inception module with a residual neural network, giving the network the flexibility to learn features at various scales. The suggested models are evaluated on four publicly available datasets: ISIC 2016, ISIC 2017, ISIC 2018, and PH2. W-net and ψ-net models achieved high accuracy levels (over 97%) on all databases, outperforming existing methods and offering better sensitivity, specificity, Jaccard index, and dice similarity. In contrast, the Inception ResNet network's performance for classification

varies depending on the dataset and evaluation metrics. The results were then compared to current approaches in the literature, demonstrating that our suggested strategy is accurate, and efficient in both the segmentation and classification of skin lesions. .
Key words: *Skin cancer, melanoma, Deep learning, encoder-decoder, segmentation, convolutional neural network, classification.*

Résumé

Le mélanome malin, une forme extrêmement mortelle de cancer de la peau, devient de plus en plus courant, en particulier chez les personnes à la peau claire exposées au soleil. Le mélanome peut être efficacement traité à ces stades précoces, donc une détection précoce est cruciale pour augmenter les chances de survie. Les dermatologues jouent un rôle important dans la détection du mélanome en examinant les caractéristiques cliniques dermoscopiques telles que les limites des lésions et les la nature de pigmentations, qui sont des indicateurs essentiels de la maladie. Cependant, cette approche peut être difficile et chronophage en raison des différences de taille et de couleur de la lésion, d'un faible contraste et de la présence de composants tels que des cheveux, des lubrifiants et des bulles d'air qui peuvent interférer avec un diagnostic précis, même pour un expert.

Pour relever ces défis, cette thèse propose des méthodes fiables et efficaces pour l'analyse d'images de dermoscopie automatisée afin d'aider à la détection précoce du mélanome malin. En particulier, cette thèse propose un systéme pour segmenter et classifier automatiquement les lésions cutanées à partir d'une image d'entrée. Le cadre proposé vise à améliorer l'exactitude et l'efficacité de l'identification et de la classification des lésions cutanées, en particulier le mélanome, en utilisant des méthodes d'apprentissage profond.

Le systéme comprend plusieurs étapes : le prétraitement, la segmentation des lésions et la classification des lésions cutanées. Dans l'étape de prétraitement, un filtre gaussien est appliqué pour améliorer la qualité de l'image en réduisant les artefacts tels que le brouillard, les cheveux, les marques et les taches. De plus, la région d'intérêt est extraite et les parties inutiles de l'image sont supprimées grâce à un recadrage automatique. Pour la segmentation des lésions cutanées, deux méthodes d'apprentissage , W-net et ψ-net, sont proposées. W-net est une architecture d'encodeur-décodeur double qui utilise un encodeur-décodeur ResNet, un encodeur-décodeur ConvNet et un réseau Feature Pyramid afin de créer une architecture CNN robuste pour la segmentation d'image. D'autre part, ψ-net est une architecture d'apprentissage en profondeur qui intègre de nombreux changements au réseau de base Unet pour augmenter la performance de segmentation. Pour la classification des lésions, le cadre proposé utilise le réseau Inception-Resnet, une forme particulière de réseau de neurones convolutifs qui effectue l'extraction de caractéristiques de manière autonome. Ce réseau se compose de couches de convolution et de regroupements, ainsi que de blocs Inception-ResNet. Le composant clé responsable de la classification est le bloc Inception-ResNet, qui combine le module Inception avec un réseau de neurones résiduels, donnant au réseau la flexibilité d'apprentissage des

caractéristiques à différentes échelles.

Les modèles proposés sont évalués sur quatre ensembles de données public : ISIC 2016, ISIC 2017, ISIC 2018 et PH2. Les modèles W-net et ψ-net ont atteint des niveaux de précision élevés (plus de 97%) sur toutes les bases de données, surpassant les méthodes existantes et offrant une meilleure sensibilité, spécificité, indice de Jaccard et similarité de Dice. En revanche, la performance du réseau Inception ResNet pour la classification varie en fonction de l'ensemble de données et des métriques d'évaluation. Les résultats ont été comparés aux approches récentes dans la littérature. La comparaison de nos résultats avec ceux d'approche récente dans la littérature a démontré que les méthodes proposées sont plus précises, efficaces, soit pour la segmentation, soit pour la classification des lésions cutanées.
Mots clés : *Cancer de la peau, mélanome, Deep learning, encodeur-décodeur, segmentation, réseau de neurones convolutifs, classification.*

Contents

List of Figures

List of Tables

List of Abbreviations

ACC:	Accuracy
BN:	Batch normalization
Bce dice loss:	Binary Cross Entropy and Dice Loss
CAD:	Computer-aided diagnosis
Conv:	Convolutional Layer
CNN :	Convolutional neural network
DenseNet:	Densely connected convolutional network
EFP:	Efficient feature pyramid
FP:	Feature pyramid
FP:	False positive
FN:	False negatif
FCN:	Fully Convolutional Networks
FR:	Fire-ResNet
ISIC-2016:	International Skin Imaging Collaboration 2016
ISIC-2017:	International Skin Imaging Collaboration 2017
ISIC-2018:	International Skin Imaging Collaboration 2018
MSC:	Multi-scale Convolutional
MLP:	Multi-layer Perceptron
NN:	Neural network
ROI:	Region of interest
Relu:	Rectified linear activation function
ResNet:	Deep Residual Networks
SAG:	Spatial gate attention
SPEC:	Specificity
SENS:	Sensitivity
TP:	True positive
TN:	True negatif

Chapter 1

INTRODUCTION

1.1 Motivation

Malignant melanoma is a type of skin cancer that originates from melanocytes, which are responsible for producing pigments in the skin. Compared to other types of skin cancers, melanoma is more likely to spread and metastasize to other organs, making it the deadliest form of skin cancer. In recent years, there has been an increase in the incidence and mortality rates of malignant melanoma in countries such as Algeria, Canada, the UK, and the USA. Early diagnosis is crucial, and computer-aided diagnostic methods could play a vital role in reducing mortality rates and associated medical costs. Early detection of skin cancer, particularly malignant melanoma, is possible for two reasons (Saurat, 2004): First, skin cancer is readily visible on the skin. Second, the prognosis of melanoma is closely related to its thickness, and it can be successfully treated if detected early when the thickness is less than 1 mm. However, diagnosis is still subjective and dependent on the visual perception and clinical experience of dermatologists (Braun et al., 2005) (Maglogiannis and Doukas, 2009) (Hoshyar, 2015).

Considerable efforts have been made to develop methods for detecting skin cancers and various imaging techniques such as dermoscopy are used for this proposes(Argenziano et al., 2003). These methods have been shown to improve the diagnosis rate of melanoma compared to visual inspection alone by increasing detection accuracy to 75-85% (Argenziano et al., 2003). Thus, we still need an automatic detection system for melanoma that can reduce false detections and provide higher accuracy. This thesis aims to advance the field of skin cancer detection by developing novel and effective deep learning models for the segmentation and classification of skin lesions.

Such research could have practical applications in accurately diagnosing and treating skin cancer, ultimately leading to improved patient outcomes.

1.2 Research Objectives

In recent years, there has been a growing interest in utilizing computer-based methods for the detection of skin cancer. These systems aim to provide a second opinion on diagnoses with higher accuracy and reliability than what a human expert could achieve. Various studies have been conducted on automated detection of melanoma, highlighting the potential benefits that can be derived from these systems. However, there are still several challenges involved, and new contributions in this field are highly valued. It is widely acknowledged that more dependable and trustworthy detection systems require higher accuracy. Typically, an automated skin cancer diagnosis involves various components, such as preprocessing, segmentation, feature extraction and selection, and classification.

This thesis aims to develop and evaluate a comprehensive skin lesion detection system with a particular focus on improving the accuracy of automated skin cancer segmentation and classification. The research will involve investigating and experimenting with different techniques and algorithms for each step of the detection process.A new segmentation algorithms will be proposed to differentiate skin lesions from background tissue and other lesion types. An algorithm for lesion classification will be developed to determine whether a detected lesion is benign or malignant.

The overall the objectif of this thesis is to contribute to the development of a reliable and automated skin cancer detection system that can improve the accuracy of diagnoses and assist clinicians in making treatment decisions more quickly and effectively. This is a crucial area of research since skin cancer is a common and potentially life-threatening disease, and early detection is essential for successful treatment.

1.3 Research Contributions

The contribution of this thesis is focused on the development of an automatic diagnosis system that utilizes machine learning algorithms to automatically classify skin cancer tumors as either benign or malignant with greater accuracy and efficiency than human experts. To achieve this objective, the following steps will be taken:

- implementation, and evaluation of multiple solutions for skin lesion segmentation using encoder-decoder models: W-net and ψ-net to develop effective and accurate methods for delineating the borders of skin lesions in images.
- Analysis and comparison of the proposed segmentation architectures with other architectures mentioned in this thesis using different evaluation metrics.
- Investigation of the effects of segmentation on skin lesion classification
- Implementation of an approach to skin lesion classification using a novel CNN model involves developing a new CNN architecture specifically designed for skin lesion classification called the Inception Resnet model. Then, training the proposed CNN model using skin lesion image datasets and evaluated its accuracy and generalization performance on a separate validation dataset.
- Analysis and comparison of proposed classification architecture with other methods mentioned in this thesis using different evaluation metrics.

1.4 Thesis Outline

This section describes the structure of this thesis, which is divided into seven chapters, including the introduction and conclusion. The organization of the main chapters is depicted in Figure 1.1. The body of the thesis is structured as follows:

Chapter 2 provides a general overview of skin cancer and its basic concepts, without delving too deeply into the subject. It serves as a foundation to better understand the topic for applying technical knowledge in addressing skin cancer-related challenges

Chapter 3 gives a technical overview of deep learning concepts, particularly convolutional neural networks, by describing the building elements that comprise a CNN and different CNN architectures applied for segmentation and classification tasks.

Chapter 4 provides a comprehensive overview of the state-of-the-art techniques and methodologies for skin image segmentation and classification. It includes a detailed analysis of recent research studies and their performance evaluations.

Chapter5 discusses the implementation details of the improved proposed approach for medical image segmentation. It includes information on the architecture choices made for the segmentation task, such as the proposed deep learning network used, the training

data, and the hyperparameters tuned during training. The chapter also presents the results of the segmentation task and a detailed analysis of the performance of the proposed approach.

Chapter 6 introduces two innovative image segmentation algorithms: W-net and ψ-net. These algorithms are designed to address challenges in image segmentation tasks and offer novel solutions to enhance segmentation performance. W-net and ψ-net likely employ unique techniques and architectures to improve the accuracy and robustness of object segmentation in images.

Chapter 7 presents the Inception-ResNet model for image identification, presenting a detailed explanation of its architecture that combines Inception modules with residual connections. The algorithm's techniques are elaborated upon, underscoring their relevance and potential advancements within the domain of deep learning and computer vision.

Chapter 8 introduces the results of the proposed methodology for skin image segmentation and classification. It compares the performance of the proposed approach with existing methods and algorithms.

Chapter 9 provides a conclusion to the thesis work by summarizing the research methodology, results, and findings and highlighting prospective research directions that are presently being investigated.

1.5 Publications

_ KHOULOUD Sahib, AHLEM Melouah, FADEL Touré, SLIM Amel,. W-net and inception residual network for skin lesion segmentation and classification. Applied Intelligence, 2022, p. 1-19.

_ KHOULOUD Sahib, AHLEM Melouah, SLIM Amel. Unet model for COVID19 infected area segmecancer-relatedimages. In: 4th International Conference on Artificial Intelligence in Renewable Energetic. IEEE, 2020.

_ KHOULOUD, Sahib, AHLEM, Melouah, FADEL, Toure, SLIM Amel. Improved Residual U-Net for COVID-19 Lung Infection Multi-Class Segmentation in CT Image. In : 2022 International Conference of Advanced Technology in Electronic and Electrical Engineering (ICATEEE). IEEE, 2022. p. 1-6.

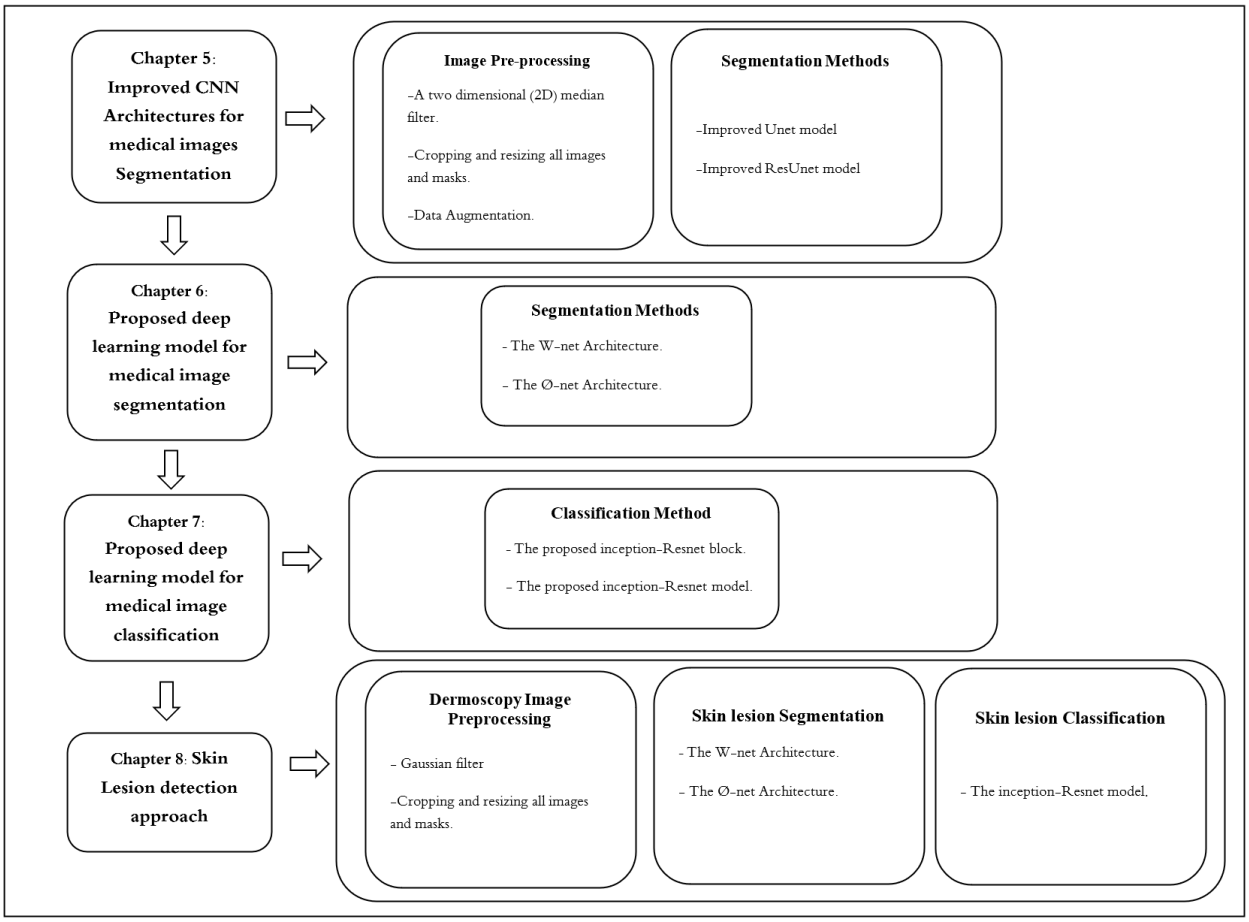


Figure 1.1: An illustration of the chapter structure.

Part I

DEEP LEARNING APPROACHES FOR SKIN CANCER DETECTION: BACKGROUND AND OVERVIEW

Skin cancer is a significant global health concern, and its early and accurate detection is crucial for improving patient outcomes. Recent advancements in artificial intelligence (AI) and deep learning have shown great promise in medical image analysis, particularly in skin cancer detection. Part One of this thesis serves as a fundamental introduction to the application of deep learning in this context, equipping readers with essential background knowledge and technical insights. The chapter provides a comprehensive overview of skin cancer, including valuable information on human skin, skin cancer types, and diagnostic techniques. Additionally, it introduces deep neural networks, with a specific focus on convolutional neural networks (CNNs), known for their proficiency in image segmentation and classification tasks. Part One also includes a comprehensive review of state-of-the-art techniques for skin image segmentation and classification, underscoring their significance in improving the diagnostic process and patient care. By laying this groundwork, Part One aims to pave the way for further exploration of deep learning-based skin cancer detection in subsequent sections of the thesis, ultimately contributing to enhanced skin cancer diagnosis and better patient outcomes through advanced AI technologies.

Chapter 2

HUMAN SKIN CANCER

2.1 Introduction

Skin cancer is one of the main causes of mortality and a global hazard to humans. This cancer can be cured if it is diagnosed at an early stage. In recent years, non-invasive imaging technologies like dermoscopy have evolved and developed These technologies help in screening and improving diagnostic accuracy. Dermoscopy provides a greater magnification for the inspection of skin lesions, allowing for a more detailed analysis of morphological features. This technique can improve the vision of various clinical aspects of a pigmented skin lesion that are otherwise undetectable to the naked eye.

Several medical diagnosis approaches are used to help dermatologists identify melanoma skin cancer, including the 3-point checklist (Soyer et al., 2004a), the ABCDE rule (Nachbar et al., 1994), the 7-point checklist (Argenziano et al., 1998), and pattern analysis (Masood, 2016). In general, the collected clinical characteristics must be evaluated by specialist doctors utilizing a variety of dermoscopic criteria, such as asymmetry, border, colors, and differential structures, to provide the final clinical diagnosis.

The chapter is divided into three sections: the first section describes the structure of human skin and different types of skin cancer; the second part focuses on skin cancer imaging techniques; and the third section discusses diagnosing techniques in detail.

2.2 Human Skin

The skin is the largest organ of the human body and serves as a vital protective barrier. It has several important functions, including absorbing sensory inputs from the external environment and regulating body temperature. In addition, the skin stores water,fat, and plays a role in vitamin D synthesis. Human skin is made up of multiple layers including the epidermis, dermis, and hypodermis (Melissa Conrad Stöppler, 2021). Figure 2.1 presents the layers and cells of Human skin.

- **The epidermis**: is primarily made up of squamous, flat, scale-like cells. Basal cells are round cells that sit underneath the squamous cells. Melanocytes generate melanin, which gives the skin its color. They are also located in the deepest layer of the epidermis.(Encyclopaedia Britannica, 2021)
- **The dermis (inner layer of skin)**: is positioned beneath the epidermis, serving as the underlying layer of the skin. It serves several important functions in the body's overall protection and support system. Composed of connective tissue, the dermis contains a variety of complex structures that contribute to its diverse functions.(Encyclopaedia Britannica, 2015)
- **The subcutis**: works as the interface of the skin with the rest of the body. Even if it is part of the skin, the subcutis (hypodermis) is not usually regarded as a skin layer. It is located underneath the dermis and links the skin to the bones and muscles. It serves as a conduit for large blood and lymphatic vessels. Binding and fat (adipose) tissue are present inside the hypodermis.(Brandi Jones, 2022)

2.3 Skin Cancer

The human body consists of living cells that undergo development, division, and natural cell death. Cancer occurs when cell division becomes aberrant and uncontrolled, leading to the growth of abnormal cells (*What Is Cancer?* 2014). Skin cancer begins in the skin as the most prevalent cancer in humans. According to the World Health Organisation skin cancer is one of the most prevalent and greatest causes of human death(*Cancer Facts & Figures* 2022). In Algeria, skin cancer accounted for 0.38% of total deaths, with 711 recorded cases in the latest WHO report from 2020 (*ALGERIA: SKIN CANCERS* 2020). Although not always fatal, early diagnosis is crucial for effective treatment.

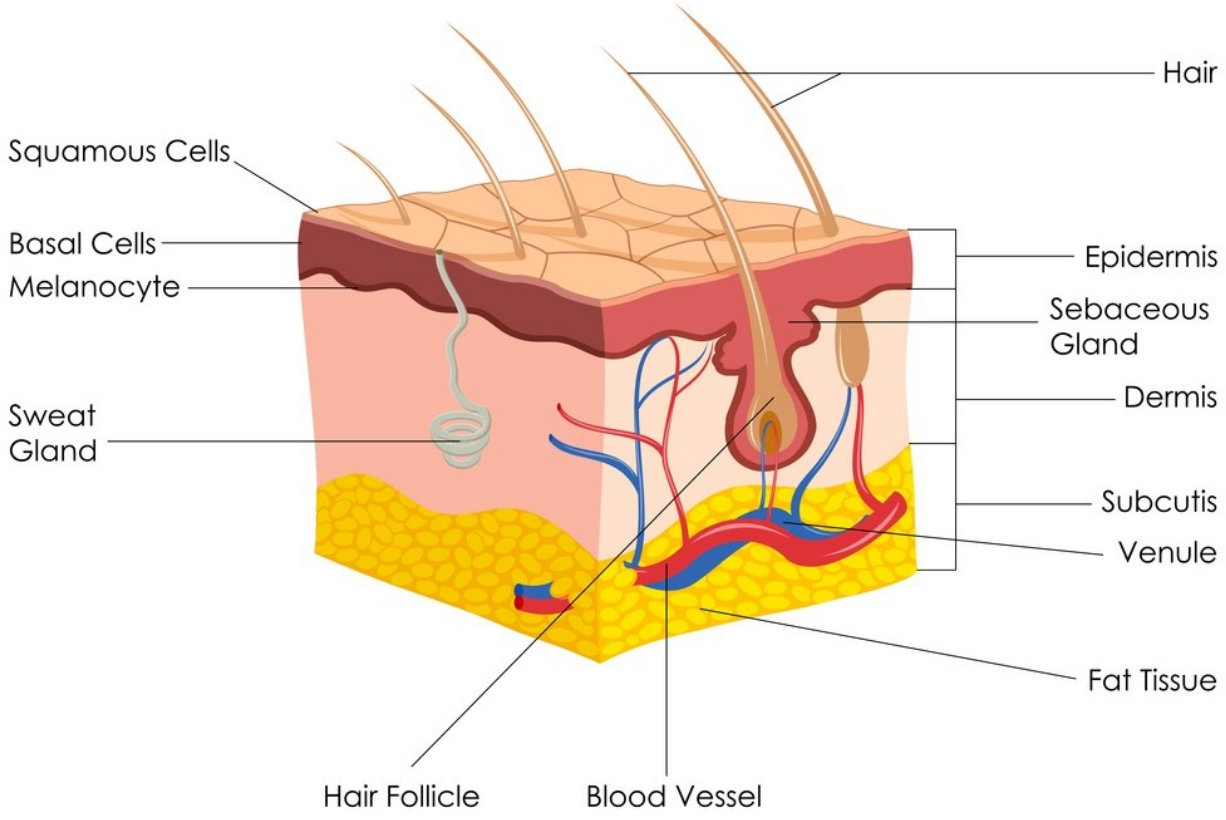


Figure 2.1: The epidermis and dermis layers in human skin with squamous cells, basal cells and melanocyte.(Joy McElroy, 2022)

Therefore, studying human skin and different types of skin cells plays a vital role in the early detection and diagnosis of skin cancer. This section will be divided into two subsections: the first will explore the causes of skin cancer, and the second will discuss the various forms of skin malignancies.

2.3.1 Causes of Skin Cancer

The primary cause of skin cancer is the uncontrolled growth of cells within the body. In the case of skin cancer, this uncontrolled growth occurs when one of the three cell types that make up the skin undergoes abnormal reproduction. These cancerous cells have the

potential to metastasize, meaning they can spread through the lymph system to other parts of the body. As a result, cancer can develop in various regions of the body and subsequently spread to other areas (*American cancer society* 2013).

Another significant cause of skin cancer is the unregulated development of abnormal skin cells. This typically happens when the DNA in skin cells becomes irreparably damaged, often due to exposure to ultraviolet (UV) light. Ultraviolet rays from the sun or tanning beds can cause DNA damage when the skin is not adequately protected (*Skin Cancer Fundation* 2022). When DNA is altered, it disrupts the normal control mechanisms of skin cell growth. This leads to changes in genes and an accelerated multiplication of skin cells, ultimately resulting in the development of malignant tumors.

2.3.2 Type of Skin Cancer

Skin cancers are generally divided into two types: malignant melanoma and non-melanoma skin cancer (NMSC). Non-melanoma skin cancer comprises basal cell carcinoma (BCC) and squamous cell carcinoma (SCC) as its two main types.

2.3.2.1 Melanoma

Melanoma is one of the world's deadliest and fastest-growing cancer forms. It is a malignant type of melanocytic lesion. Melanocytes are skin cells that are situated in the skin's outer epidermis and can be seen by the human eye. This lesion develops quicker and has a greater ability to invade tissues and spread to other organs. Melanomas are typically black or brown, although they can also be skin-colored. Melanoma can be treated by excision surgery if detected early. The primary challenge for early treatments is the high prevalence of false-negative malignant melanoma (Walter et al., 2013). In addition, melanoma is nearly usually curable if treated early, but if not, cancer can progress and spread to other regions of the body, where it becomes difficult to cure and can be deadly. Thus, early detection and treatment are important for avoiding damage. Figure 2.2 shows the sample image of malignant melanoma. The many kinds of malignant melanomas are as follows (Owen and Telfer, 2005):

- **Nodular melanoma**: Nodular melanomas occur in all areas of the body, notably the male trunk. It accounts for around 15% of all melanomas, and shortly after, it appears invasive. It usually appears similar to the development of the blood vessel. The lesions are frequently asymmetrical, having irregular borders.

- **Superficial spreading melanoma**: is the most prevalent form of melanoma, constituting around 70% of all cases. Superficially streami melanomas are brown lesions that are flat or slightly raised and have irregular asymmetric margins. It has irregular asymmetric boundaries and is generally larger than 6 mm in diameter..

- **Lentigo maligna melanoma**: accounts for around 5% of all melanomas. The precursor lesion is usually larger than 3 cm in diameter and acquires a dark brown-to-black hue or a raised blue-black nodule when it becomes invasive..

- **Amelanotic melanoma**: is uncommon, frequently difficult to identify, and usually has a false prognosis. It is not very common, comprising only about 8% of all melanoma cases. Amelanotic melanoma can be confused with other skin disorders.

- **Desmoplastic melanoma**: is uncommon and represents around 1.7% of the total. Locally aggressive and hard to identify. Most of these cancers arise in old and 1/2 a melanocytic people's heads and necks.

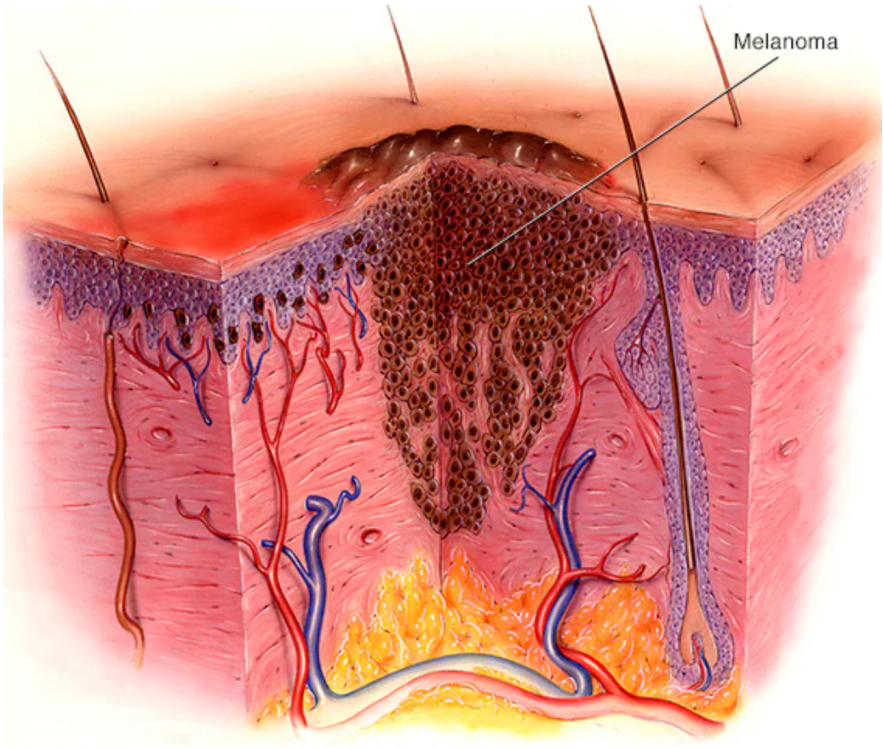


Figure 2.2: Malignant melanoma.(*Melanoma* 2022)

2.3.2.2 Basal-Cell Carcinoma (BCC)

Basal cell carcinom is the most commonly occurring type of skin cancer and is also the most frequently diagnosed form of cancer in general. It starts in the basal cells, which are a kind of skin cell that produces new skin cells when the old ones die. Basal cell carcinoma is a slow-growing skin cancer that might take years to appear after prolonged or strong sun exposure. If you're exposed to a lot of sun or use tanning beds, you can acquire it at an early age (*Basal cell carcinoma* 2022). Almost all basal cell carcinomas may be effectively removed without problems if detected and treated early. Figure 2.3 is a sample of basal cell carcinoma. Basal cell carcinoma has four major clinical forms (*Basal Cell Carcinoma* 2022):

- **Nodular Basal Cell Carcinoma**: Nodular basal cell carcinoma is characterized by its clinical appearance as a translucent nodule with prominent telangiectatic vessels. It typically presents as a small, pearly nodule that is translucent and often exhibits surface telangiectasia. Nodular basal cell carcinomas are commonly found on the face, particularly along embryonal fusion planes. However, they can also develop anywhere on the body that has experienced intermittent intense sun exposure.

- **Superficial Spreading Basal Cell Carcinoma**: Superficial spreading basal cell carcinoma is most commonly found on the upper back. It manifests as shallow plaques that are pink to almost skin-colored and have a slow growth rate. These lesions tend to be fragile and can easily bleed with minimal pressure, such as running a fingernail over the affected area, resulting in multiple tiny bleeding spots.

- **Sclerosing (morphoeic) basal cell carcinoma**: It is a scar-like plaque with a poorly defined edge. It is a whitish lesion that is steadily growing. Basal cell carcinoma nodules may not be visible until the late stages, but the area of sclerotic scarring can expand significantly before becoming clinically detectable as skin cancer. It is most frequent on the face and, due to its size at the time of diagnosis, can cause substantial morbidity.

- **Pigmented Basal Cell Carcinoma**: In dark-skinned individuals, especially Asians, pigmented basal cell carcinoma occurs. The distinguishing aspect is that pigment flecks around the base of the nodule are not present in melanoma when fully pigmented.

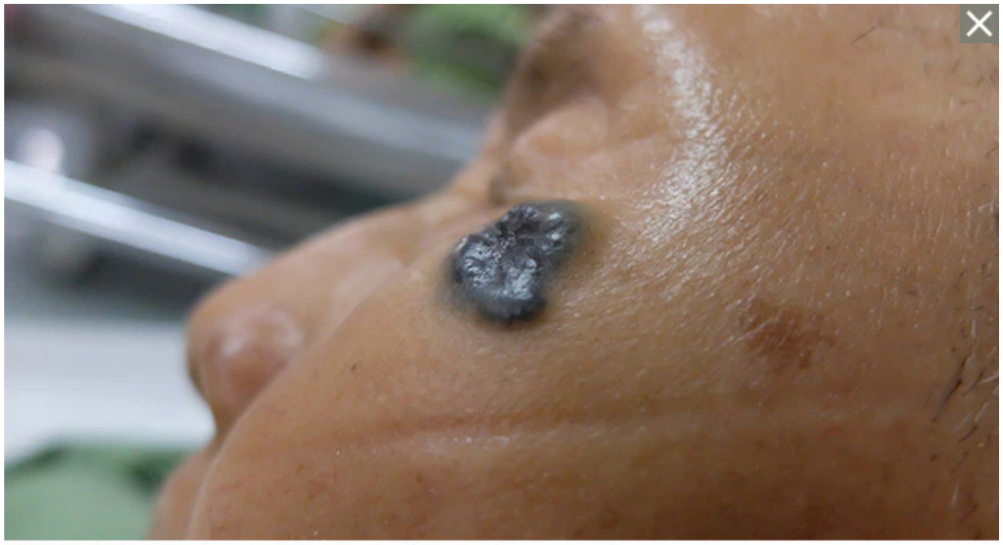


Figure 2.3: Basal cell carcinoma .(*Basal cell carcinoma* 2022)

2.3.2.3 Squamous-Cell Carcinoma (SCC)

The second-most prevalent form of skin cancer is squamous cell carcinoma. Scaly red patches, open sores, raised growths with a central depression, or warts are common symptoms. Along with UV exposure, immunosuppression is a major risk factor for SCC. Squamous cell carcinoma usually begins as a small lump, expands over time, and turns into an ulcer. Compared to BCCs, squamous-cell carcinomas have more irregular shapes with a crusted surface, and they are more likely to spread to other organs. (Cakir, Adamson, and Cingi, 2012). Figure 2.4 is a sample of Squamous Cell Carcinoma.

Figure 2.4: Squamous Cell Carcinoma .(Cakir, Adamson, and Cingi, 2012)

2.4 Skin Cancer Imaging Techniques

In the early stages, skin cancer has a 90% cure rate, whereas in the late stages, the cure rate drops to 50% (DiChiara, 2010). To enhance the accuracy of diagnosing skin malignancies or lesions, non-invasive and high-resolution imaging methods have been developed (Hibler, Qi, and Rossi, 2016). Improved diagnostic specificity through high-resolution imaging techniques has the potential to reduce unnecessary excisions, associated costs, and improve treatment outcomes, especially for melanoma, which often poses challenges in diagnosis. Currently, the most commonly used imaging techniques for skin cancer diagnosis include reflectance confocal microscopy, optical coherence tomography,Dermoscopy, and ultrasound (Dhawan et al., 2009)(Stockfleth, Rosen, and Schumaak, 2010):

- **Dermoscopy** : Dermoscopy, also known as epiluminescence microscopy (ELM) or skin surface microscopy, is a non-invasive imaging method used for identifying skin malignancies. It involves illuminating the skin using surface reflectance-dominating lighting techniques. By reducing surface reflections through the use of cross-polarized light and an immersion solution like immersion oil or alcohol, dermoscopy enables the visualization of color and subsurface features such as blood vessels or pigment, aiding in the early detection of skin cancer. Research has shown that incorporating dermoscopy into diagnostic practices can enhance accuracy by 5% to 30%, depending on the type of skin lesion being evaluated (Binder et al., 1995)(Kittler et al., 2002).

- **Ultrasound**: Ultrasound is a common non-invasive technique since it is flexible, painless, and has minimal risk. The ultrasonic waves that rebound from the tissue can be used to visualize the skin morphology during this operation. Although this method can penetrate the skin deeply to measure tumor thickness and evaluate lymph nodes, the practical use is limited due to the low resolution, which makes it difficult to distinguish between skin lesions histomorphological.

- **Optical Coherence Tomography (OCT)**:The OCT approach is based on interferometry and is non-invasive. It creates two-dimensional and cross-sectional images. OCT can scan microscopic structures (a few microns) in real-time and identify healthy from malignant tissue. Although OCT scans may distinguish micromorphology, blood vessels, and adnexial structures. Furthermore, the OCT cannot properly identify the diagnosis of melanoma without histological confirmation. As a result, the OCT is not a useful tool for melanoma diagnosis and is unable to consistently identify tumor invasion at an early stage.

- **Reflectance confocal microscopy (RCM)**: RCM is the most expensive of the skin imaging methods. This approach is non-invasive and painless, and it can observe the cellular intricacies of the skin in real-time using a laser focused on a single location on the skin. Water, hemoglobin, and melanin are among the skin chromophores that RCM can detect using refraction indices. RCM can readily distinguish reflected light from the skin because biological structures (cells, melanin, hemoglobin, etc.) have distinct refraction indices. A point light source irradiates a tiny area of tissue, which is then reflected and sent to the detector.

2.5 Diagnosis Methods of Skin Cancer

Skin lesions cannot be accurately distinguished as malignant or benign melanoma solely through imaging techniques. To address this issue and improve diagnostic accuracy, researchers are working on developing automated evaluation methods that can surpass human capabilities in terms of precision. Inexperienced dermatologists are more likely to make incorrect diagnoses, leading to defects, increased costs, and potential patient morbidity. To combat these challenges, there are four established and consistent diagnostic approaches for skin cancer, with a primary focus on melanoma: the ABCD-E rule (Nachbar et al., 1994), the 3-point checklist (Soyer et al., 2004b), the 7-point checklist (Argenziano et al., 1998) , and the Menzies' method (Menzies, 2012), which involves pattern analysis. These methods aim to enhance understanding and aid in differentiating between various types of skin cancer.

- **ABCD-E Rule**: (Nachbar et al., 1994) introduced this approach in 1994 . ABCDE is an acronym that stands for asymmetry, border, color, diameter, and changing over time, and it refers to five dermoscopic criteria for the semi-quantitative assessment of skin lesions. Figure 2.5 illustrates the five rules of ABCD/ABCDE rule. Each case's meaning is described and illustrated below (*Skin Cancer Fundation* 2022)(Adjed, 2017):
  - **Asymmetry**: A pigment is called asymmetric if drawing one vertical or horizontal line does not result in the two parts of the mole being identical.
  - **Border**: It can be considered that a case is malignant if the pigment boundary is not smooth.

- **Color**: The majority of benign pigments are only one color (often brown color). Melanoma comes in a range of hues, including black, red, blue, and/or white.
- **Diameter**: Benign pigments are often smaller in diameter than malignant pigments. Melanoma is often more than 6 mm in diameter; however, it can be smaller.
- **Evolving**: Benign pigment takes on the same appearance throughout time. Malignant melanoma, on the other hand, develops or changes in size, form, color, and/or elevation.

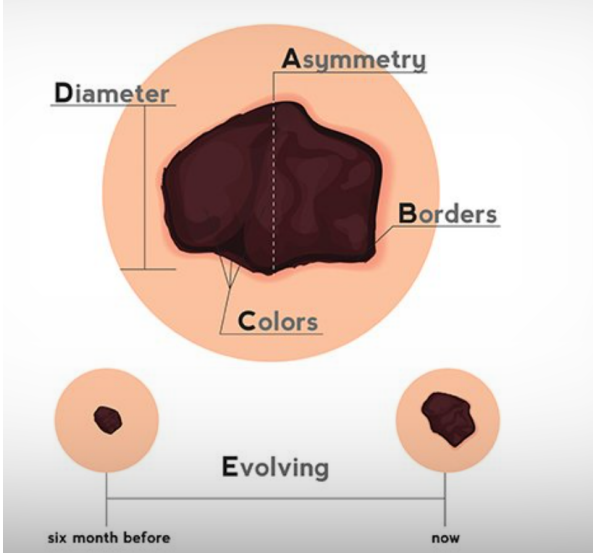


Figure 2.5: ABCDE of Skin Cancer.(*ABCDE of Skin Cancer* 2019)

- **The 3-Point Checklist**: involves the examination of three specific criteria: asymmetry, atypical pigment networks, and blue-white structures. Any of these signs or symptoms point to the potential existence of melanoma (Soyer et al., 2004a).
- **The 7-Point Checklist**: is a score diagnostic analysis that takes into account just seven conventional ELM criteria, which include (Argenziano et al., 1998) (Maglogiannis and Doukas, 2009)(Adjed, 2017):
  - **Atypical pigment network**: thicker and irregular line segments in the lesion that are black, brown, or gray.
  - **A blue-whitish veil**: irregular, confluent, gray-blue to whitish-blue diffuse pigmentation that can be linked to changes in the pigment network.

- **An atypical vascular pattern**: in the regression area, there are no linear-irregular and/or dotted red vessels.
- **Irregular streaks**: pseudopods or radial streaming abnormalities near the lesion's perimeter.
- **Irregular dots/globules**: areas of uneven form and/or distribution in black, brown, or gray.
- **Irregular pigmentation**: regions without features that are black, brown, or gray and have an uneven form or distribution.
- **Regression structures**: regions that seem like white scars and/or blue peppers (gray-blue areas and/or several blue-gray dots).

- **The Menzies method** : is an effective approach used to distinguish between benign tumors and melanoma in melanocytic lesions. It employs a set of two negative features, namely symmetry and single color, as well as a positive set consisting of nine features. The positive features include numerous brown dots, pseudopods, radial streaming, scar-like depigmentation, peripheral black dots, multiple hues, multiple blue/gray dots, and a large pigment network. In order to classify a lesion as melanoma, it must exhibit at least one of the positive features while simultaneously lacking both symmetry and a single color. This method assists dermatologists in making accurate diagnoses based on these distinctive features(Menzies, 2012)(*Menzies Method* 2022).

- **Pattern Analysis** : is a useful technique for identifying distinct patterns in skin lesions, which can aid in the diagnosis of melanocytic lesions and differentiate between benign and malignant lesions. Global and local patterns can be analyzed, including the reticular, globular, cobblestone, homogenous, starburst, parallel, multicomponent, and nonspecific patterns, which are commonly associated with benign melanocytic lesions. Additionally, other patterns, such as the reticular pattern characterized by a network-like appearance, and the globular pattern, with rounded shapes, are also observed in different types of melanocytic lesions (Masood, 2016).

2.6 Conclusion

Skin cancer is a growing concern globally, and early detection is crucial for improving patient survival rates. This chapter has highlighted the significance of early detection as

a promising strategy to reduce the mortality rate associated with skin cancer. By investigating the attributes of human skin and employing diagnostic methodologies, a greater comprehension of skin cancer can be achieved, leading to the development of efficient detection techniques.The chapter has provided valuable information on human skin, skin cancer, and diagnostic techniques. With continued advancements in knowledge and technology, early detection of skin cancer can be improved, leading to better outcomes for patients.

Chapter 3

DEEP LEARNING CONCEPTS AND ARCHITECTURES

3.1 Introduction

Deep learning and neural networks are buzzwords in today's technological landscape. These approaches have already demonstrated their efficacy in solving complex issues that were previously considered unachievable by computers. Deep learning, in particular, is a machine learning approach that has high adaptability and generalizability. It has become one of the primary techniques for image processing, and convolutional neural networks (CNNs) are the go-to deep learning models for this task. Unlike traditional machine learning techniques, CNNs do not require a clear explanation of the task-solving process. Instead, it is sufficient to specify the expected behavior of the solver. However, processing images in challenging modalities is still an active research area, and there is much to be learned in this regard.

This chapter provides a technical overview of deep learning concepts, particularly convolutional neural networks, that are essential for understanding the rest of the thesis. We will outline the key building blocks of a CNN, including its different layers and concepts related to training. Furthermore, we will briefly explain the CNN architecture used for segmentation and classification tasks. By the end of this chapter, readers will have a solid understanding of the fundamental concepts and techniques used in deep learning models for image processing.

3.2 Artificial Neural Network

3.2.1 Neural Network Architecture

Neural networks are computational systems that draw inspiration from the human brain and replicate the signaling process between biological neurons. They are composed of interconnected neurons organized in layers (Figure 3.1). Specifically, feed-forward neural networks are employed, where the output of one layer serves as the input to the next layer. These networks exhibit a unidirectional flow of information, with no loops, ensuring that input is continuously forwarded. Consequently, neurons within a layer do not have direct connections to each other. Hidden layers refer to the layers situated between the input and output layers. While the initial layer makes simple decisions based on the input, subsequent layers make complex decisions based on the decisions made in preceding layers (Snuverink, 2017).Optimizing neural networks for diverse tasks involves configuring the weights, which can be a challenging task due to the potential interdependence of millions of factors. To evaluate the performance of the network, a loss function is utilized to compare the layer representations from the network output with the predicted output. The resulting score from this assessment acts as a feedback signal for the optimizer. Using a technique called backpropagation, the optimizer adjusts the weights of the network layers based on the feedback signal. Backpropagation starts with the final loss value and propagates the contribution of each parameter down the layers. This technique enables the calculation of modifications necessary to minimize the loss and optimize the model (Chollet, 2021) (Goodfellow, Bengio, and Courville, 2016) (Yi, 2020).

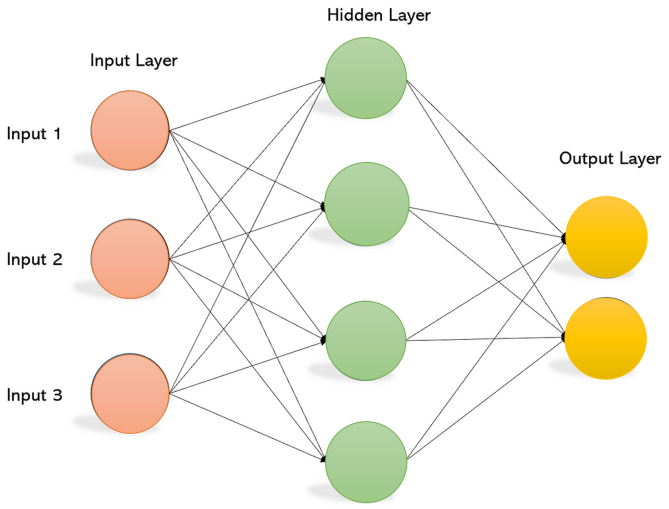


Figure 3.1: Neural Netrwork structure.

3.2.2 Activation Function

The activation function plays a crucial role in deciding whether a neuron should be activated or not. It accomplishes this by computing a weighted sum and subsequently adding a bias to it. The primary purpose of the activation function is to introduce non-linearity to the output of a neuron. The activation function is sometimes referred to as a "transfer function".If the activation function's output range is restricted, it is referred to as a "squashing function." Many activation functions are nonlinear, which is referred to as "nonlinearity" in layer or network design. The choice of activation function has a substantial influence on the neural network's capabilities and performance, and various activation functions may be utilized in different regions of the model. Although networks are designed to employ the same activation function for all nodes in a layer, the activation function is applied within or after the internal processing of each node in the network. Activation functions are generally crafted to be differentiable, which permits the calculation of the first derivative for a specified input value.This characteristic is essential as neural networks typically rely on the backpropagation of error algorithm for training, necessitating the existence of the prediction error derivative to adjust the model's weights. Various activation functions are utilized in neural networks, although only a select few are expected to be applied in practical scenarios for the hidden and output layers.(Brownlee, 2021).

Sigmoid Function

The sigmoid function (Figure 3.2) is the most frequent form of activation function which is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

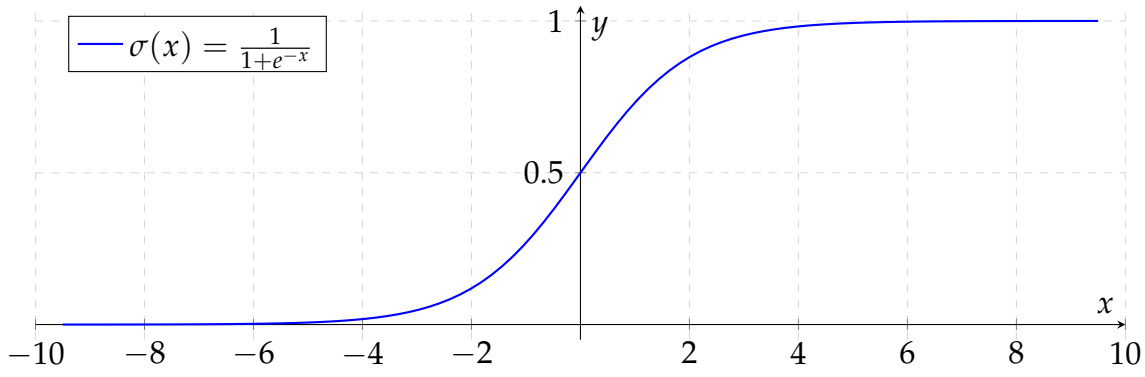


Figure 3.2: Sigmoid Activation Function.

Hyperbolic Tangent

Despite the sigmoid's popularity, practically and theoretically, a sigmoidal activation function in the form of a hyperbolic tangent is occasionally chosen for neural network. The logistic sigmoid is also linearly transformed onto the interval [-1,1] using the hyperbolic tangent tanh(x) (Figure 3.3). As the input approaches a big positive or negative integer, both activation functions saturate (Farhadi, 2017). This signal saturation is responsible for the well-known vanishing gradient problem when training deep neural networks, as we will discover later (Charmchi, 2018). The former function of the rescaling logistic sigmoid is:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{3.2}$$

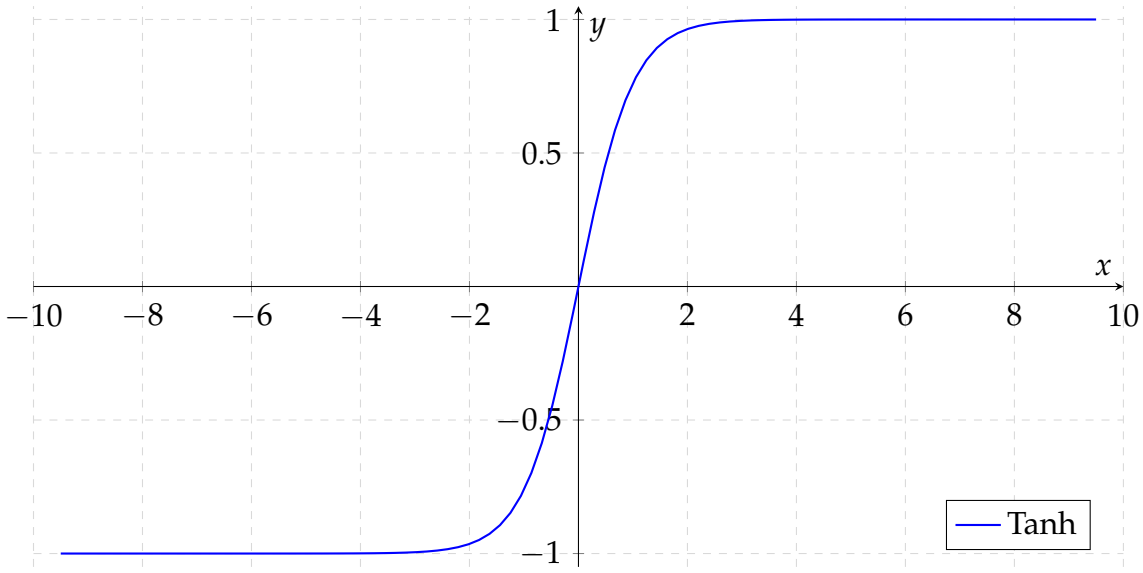


Figure 3.3: Tanh Activation Function.

Rectified Linear Unit(ReLu)

The rectified linear activation function (Figure 3.4) has been popular in recent years since it provides both the requisite non-linearity and a rapid computation when compared to sigmoid or hyperbolic tangent which is defined as (Krizhevsky, Sutskever, and Hinton, 2017):

$$Relu(x) = max(0, x) \tag{3.3}$$

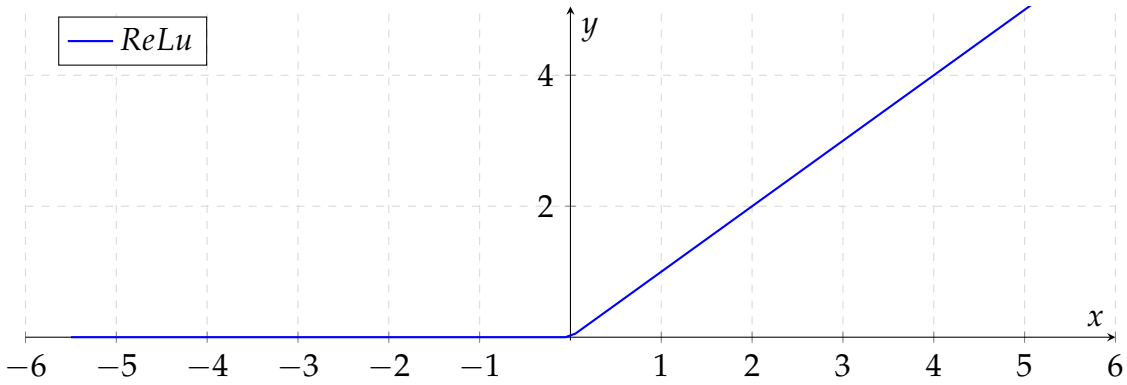


Figure 3.4: ReLu Activation Function.

3.2.3 Backpropagation

The core of neural network training is backpropagation. Backpropagation, which is an abbreviation for "backward propagation of errors," is a widely used technique for training artificial neural networks. It's a technique for fine-tuning the weights of a neural network depending on the previous epoch's error rate (i.e., iteration). Fine-tuning the weights can lead to a reduction in error rates and an improvement in the model's generalization, thereby increasing its reliability. Backpropagation is helpful in calculating the gradient of a loss function with respect to all of the network's weights.(Al-Masri, 2019). Target classes are required for error computation in supervised learning. After then, the fault is transmitted to all nodes in previous tiers. This error er (Equation 3.5) is calculated as a gradient of the loss function L with respect to the weights W_{mj} of each layer, given the node x and activation function σ (Uličnỳ, 2015).

$$p_j = \sigma(\sum_{m=1}^{n} W_{mj} x_m) \tag{3.4}$$

$$er = \frac{\partial L}{\partial W_{mj}} = \frac{\partial L}{\partial p_j} \frac{\partial p_j}{\partial W_{mj}} \tag{3.5}$$

In order to compute the partial derivative of the loss function L with respect to a given weight W_mj, the chain rule must be applied. An optimization technique, such as gradient descent, updates weights based on the error (Uličnỳ, 2015).

3.3 Deep learning paradigm

Deep learning, a subfield of artificial intelligence, strives to accurately emulate the intricate workings of the human brain (Casey, 2020). With the rapid advancement of computer power and the vast availability of data, deep learning has emerged as the predominant approach in machine learning (LeCun, Bengio, Hinton, et al., 2015). It possesses the capability to discern significantly more intricate patterns than conventional machine learning methods. Consequently, deep learning has become a powerful tool in fostering the development of artificial intelligence (AI) in recent years. Convolutional neural networks (CNNs) are a key component in all deep learning methodologies and have played a pivotal role in improving state-of-the-art image processing techniques. CNNs have played a vital role in significantly advancing the accuracy and efficiency of diagnosis in computer-aided diagnosis (CAD) systems. Deep learning techniques, facilitated by CNNs, have revolutionized CAD systems by enabling more precise and efficient diagnoses. This has paved the way for improved healthcare outcomes and enhanced decision-making capabilities in clinical settings. (Singh et al., 2020).

3.4 Basics of convolutional neural networks

In recent years, there has been significant progress in the development of convolutional neural networks (CNNs), a type of deep neural networks. CNNs have gained prominence due to their ability to exploit the inherent characteristics of images, where nearby pixels hold more correlation than distant ones. This property enables CNNs to efficiently process image data compared to fully connected feedforward networks (LeCun et al., 1998). The key components of the CNNs algorithm are: convolution, pooling, and fully-connected layers. These operations have the advantage of being able to be applied to either the input images or the output of the hidden layers.

3.4.1 Convolutional Layer

The convolution layer is a fundamental operation in convolutional neural networks (CNNs) where input neurons and model weights are combined linearly. In CNNs, this layer is commonly used on 3-dimensional input blobs to produce 2-dimensional output blobs when applied to images (Simonyan and Zisserman, 2014) (Krizhevsky, Sutskever, and Hinton, 2017). Convolutional layers are essential in generating feature maps from input

images or lower-level feature maps. They consist of kernels or filters with smaller dimensions than the input image. These kernels slide over the image, performing element-wise multiplication with the input to generate a feature map (Snuverink, 2017). The convolution operation involves multiplying the filter elements with the input elements element-wise. Starting from the leftmost component, the filter slides over the input image with a specified stride. The results are then summed, along with a bias term, to produce an output unit. This process continues until the filter has traversed all input locations and computed all output units. Each filter is assigned different weights to specialize in detecting specific features in the input. Adjusting the depth, stride, and zero-padding parameters allows for control over the size and number of feature maps. The number of filters used affects the depth of the output, while the stride determines the displacement of the kernel across the input image. Zero-padding involves adding zeros around the matrix's border to manage the output feature map's size (Naceur, 2020). The process of convolution is visualized in Figure 3.5.

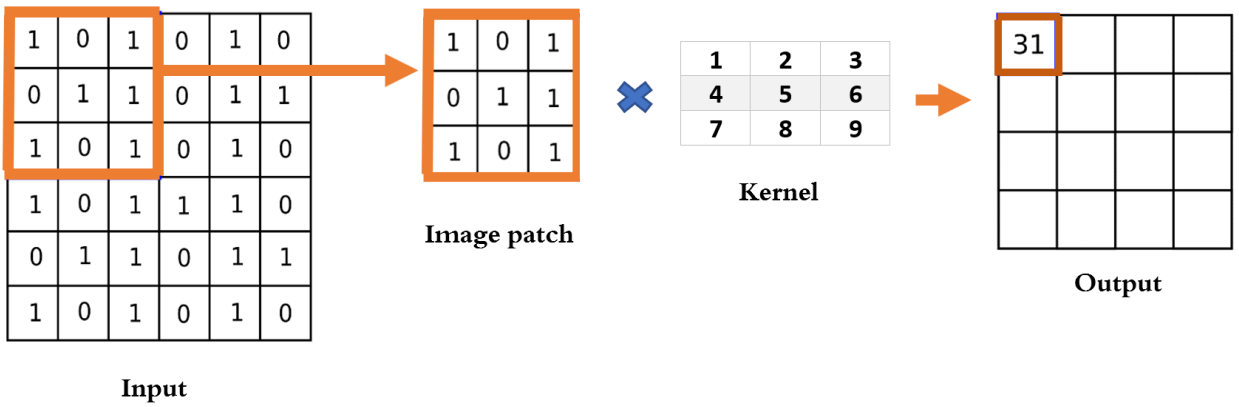


Figure 3.5: Convolution process.

3.4.2 Pooling Layer

Pooling is applied to a limited number of input neurons, and a maximum, total, or average is often computed over that region. Pooling introduces translation invariance, allowing for the reduction of feature space dimensions, and lowers computational costs.Downsampling layers are another name for pooling layers. Maxpooling is a common pooling approach. The down sampled output is created by taking the greatest input value within the kernel, resulting in a reduced output. A pooling layer has a kernel and a stride of comparable

length. Pooling layers have two key advantages: they reduce the amount of weights and the likelihood of overfitting the training data (Snuverink, 2017).

3.4.3 Fully connected layer

The output of the pooling layer undergoes a crucial step before entering the Fully Connected (FC) layers. It is flattened to form a vector of features, which serves as the input layer for the FC network. The FC layers, resembling a Multi-Layer Perceptron (MLP), consist of three distinct layer types: input, hidden, and output. The primary function of the hidden layers within the FC network is to transform non-linear space representations into a more concise and discriminative feature space. Lower-level features like lines and edges are extracted by the initial hidden layers, while higher-level representative features are progressively extracted by subsequent hidden layers. At the output layer, a classifier is employed to facilitate classification tasks. This classifier categorizes the extracted input features into predefined classes. Examples of common classifiers include Softmax, Sigmoid, or other standard machine learning classifiers."(Naceur, 2020).

3.5 Deep learning Architectures for Image Classification

In the past decade, there have been significant advancements in CNN architectures (Shrestha and Mahmood, 2019)(Khan et al., 2020a). The design of the model architecture plays a crucial role in improving the performance of various applications. Over the years, CNN architectures have undergone several modifications, including structural reformulation, regularization techniques, parameter optimizations, and more. These changes have been introduced from 1989 to the present, contributing to the evolution of CNN architecture. Notably, the remarkable improvement in CNN performance can be attributed to two key factors: the restructuring of processing units and the development of novel building blocks (Alzubaidi et al., 2021). These advancements have had a significant impact on the capabilities and efficiency of CNN models. To explore different approaches to classification, let's consider some notable CNN architectures: Inception, ResNet, SqueezeNet, MobileNet, DenseNet, and EfficientNet. Each of these architectures has its unique characteristics and has demonstrated impressive performance in various computer vision tasks.

3.5.1 Inception

(Szegedy et al., 2015) presented the Inception Module with the GoogLeNet architecture in their work "Going deeper with convolutions". While simply adding layers and using large filter numbers to enhance the depth of a network is a fairly straightforward approach to attempting to improve network performance, the disadvantages are a risk of overfitting and a higher computing cost. (Szegedy et al., 2015) developed a method to overcome this problem with the introduction of the Inception Module. The main concept is to combine the advantages of sparse and dense computing. Figure 3.6 shows a graphical illustration of the Inception Module.

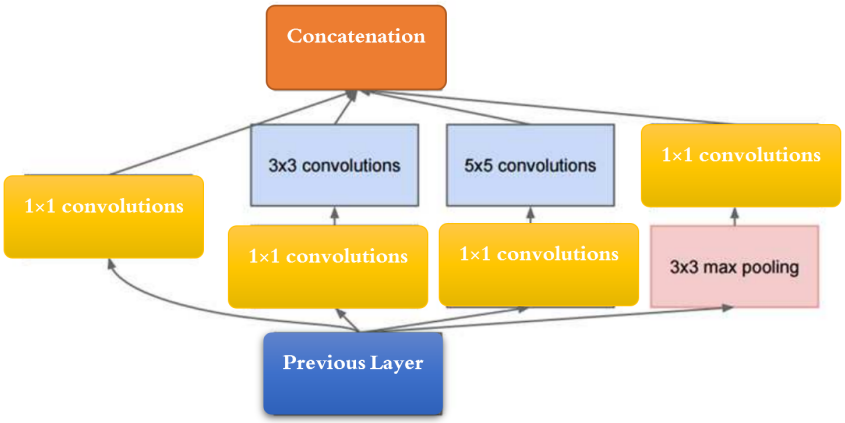


Figure 3.6: Inception Module.

Convolutional neural networks (CNNs) using Inception Modules are used to cut down on computational costs. Neural networks must be constructed effectively since they deal with a large number of images with a wide range of highlighted image content, also known as salient parts. Convolution is performed on input with not one, but three distinct sizes of filters in the most basic version of an inception module (1x1, 3x3, 5x5). In addition, maximum pooling is used. Rather of having one convolutional layer or one pooling layer, convolutional and pooling layers function in parallel. The authors reduced the dimensionality of the feature maps by inserting 1x1 convolutional layers before the 3x3 and 5x5 layers because this would result in too many outputs. Besides, A ReLU unit is placed after these 1x1 layers. convolutions 3x3 and 5x5 can cover larger spread-out clusters, but 1x1 convolutions can only cover extremely small clusters. This is why smaller convolutions are employed in the early layers of the CNN, but 3x3 and 5x5 are more prevalent in the

later layers, when high-level reasoning takes place. The outputs are then concatenated and forwarded to the next layer. The network develops progressively broader, not deeper, by configuring the CNN to complete its convolutions on the same level.

3.5.2 Deep Residual Networks (ResNet)

The ResNet, introduced by Microsoft Research Asia, is a deep neural network that addresses the challenge of degradation in performance when stacking more convolutional layers (He et al., 2016). While increasing network depth initially improves accuracy, there comes a point where the performance starts to degrade. To overcome this issue, the authors proposed a residual learning framework. The residual learning framework enables the training of deep neural networks by incorporating shortcut connections between layers, allowing for improved network depth without increasing parameters or complexity (Chen et al., 2020). Residual networks, often referred to as ResNets, have gained widespread popularity in the literature and have demonstrated significant success in various deep learning applications (Liu et al., 2019). In the literature, different variants of ResNet are distinguished based on the number of convolutional layers they possess. For instance, ResNet-101 consists of 101 convolutional layers, while ResNet-34 has 34 convolutional layers. A building block of ResNet is illustrated in Figure 3.7.

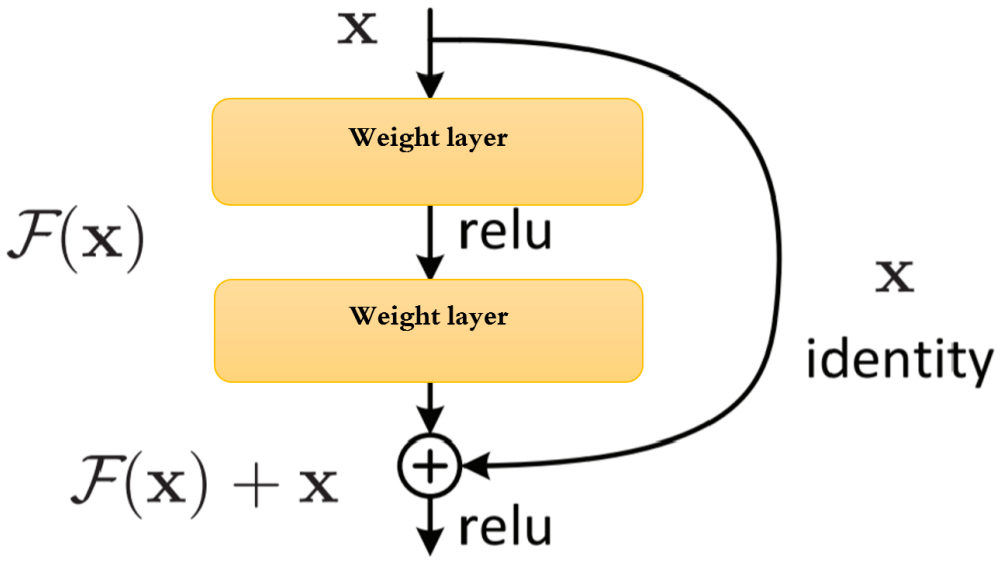


Figure 3.7: A building block for residual learning.

As is typical, the input x is passed into a stacking of convolutional layers. A shortcut is also implemented in addition to this. Shortcut connections are connections between layers that skip one or more layers. This shortcut connection is an identity function in residual networks. As a result, the output of this shortcut equals its input (Kölbl, 2017). The identity function's output (x) is then added to F(x), which is the output of the weight layers:

$$F(x) + x. \tag{3.6}$$

3.5.3 SqueezeNet

SqueezeNet was developed by (Iandola et al., 2016) with the goal of reducing the number of parameters required to adjust the network while maintaining accuracy comparable to other well-known models. The essential features of this network are the reduction in the number of input channels for each convolutional layer, as well as the downsampling late in the network, resulting in a large activation map for the convolutional layers (Abreu Dias, 2019). In the SqueezeNet paper, the author outlines three main strategies: First, they employed 1×1 filters instead of 3×3 filters to minimize network parameters. Because 1×1 kernels have 9 fewer parameters than 3×3 kernels, it seems to be more efficient. Second, they Reduced the number of input channels to 3×3 kernels. To reduce the depth of the model and the computation for the following 3 ×3 kernels, they employed the 1×1 kernels as a bottleneck layer called a squeeze layer. Finally, downsampling later in the network to preserve the feature maps while keeping the huge activation maps. The theory is that the larger activation map is preserved because the network's downsampling is delayed, resulting in improved classification accuracy (Gaikwad, 2018). To implement the above strategy (Iandola et al., 2016) employed the fire module in the SqueezeNet architecture, which has two types of layers: a squeeze layer and an expanded layer. A SqueezeNet is a network that consists of a number of fire modules and a few pooling layers. The feature map size is kept the same by the squeeze and expand layers, but the former reduces the depth while the latter increases it. In neural networks, squeezing (bottleneck layer) and expansion behavior is common. To obtain a high-level abstract, another common pattern is to increase depth while lowering feature map size. Figure 3.8 shows the Fire layer of SqueezeNet Architecture.

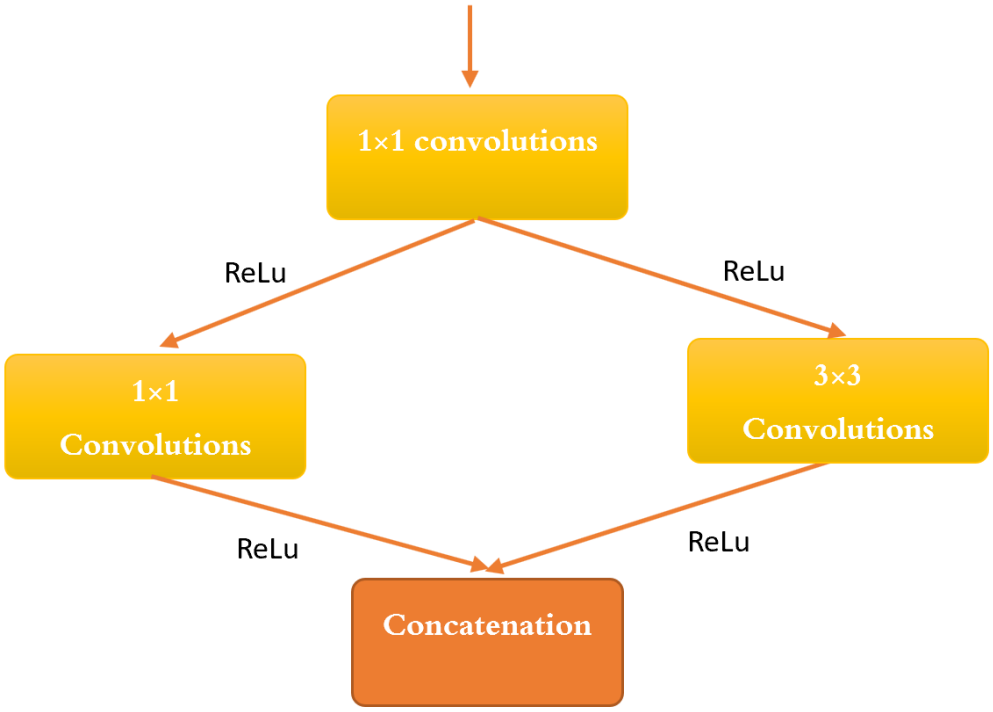


Figure 3.8: Fire layer of SqueezeNet Architecture.

3.5.4 MobileNet

Researchers worked to develop CNN architectures that were specifically designed to work on mobile and embedded devices. In this aspect, MobileNets are currently the most common network family. In comparison to other CNN architectures like ResNets or VGGNets, they provide a favorable trade-off between computational costs and network accuracy. So far, three versions of MobileNets have been released, each of which builds on the preceding version while also introducing new concepts.

In 2017, a Google team presented MobileNetV1 (Howard et al., 2017). It is made up of VGG-style building blocks that significantly decrease the model's parameters and MACs by replacing traditional convolutional layers with depthwise separable convolutions and pointwise convolutions. After every convolutional layer, a batch normalization layer and a ReLU layer are applied consecutively. By replacing standard convolutions with depthwise-separable convolutions, the number of parameters is drastically decreased. As a result, it appears that the model will lose its descriptiveness. The concrete structure of a MobileNetV1 is defined by two hyperparameters: width and resolution multiplier.

MobileNetV2 (Sandler et al., 2018) was created with mobile platforms in consideration, with low computational complexity and great performance. It splits convolutions into two layers using depth-wise separable convolutions instead of the traditional convolutional layer. A depth-wise separable convolution is applied before a point-wise convolution in depth-wise separable convolutions. In contrast, to complete convolutional operators, which apply convolutions to all channels, depth-wise convolutions apply convolutions to a single input channel at a time. Following the depth-wise convolutions, the point wise convolution computes linear combinations of the input channels to generate new features. Depth-wise separable convolutions are made up of these layers, which may be used to substitute regular convolutional layers at a reduced computational cost. The depth-wise convolutions in the MobileNetV2 architecture 35 leverage linear bottleneck shortcut connections to produce inverted residual blocks. To ensure that less information is lost, MobileNet v2 35 employs linear bottlenecks to extend channel dimensions in low dimensions. Inverted residual blocks are formed when linear bottlenecks and depth-wise separable convolution are combined. Levels with a minimal number of channels are connected, and no non-linear activation function is utilized in these layers. The inverted residual blocks minimize computing complexity while maximizing feature extraction efficiency.

MobileNet V3 (Howard et al., 2019) is the most recent version of a series of MobileNets that are meant to increase accuracy while keeping resource limits in mind. A network-aware platform architecture search was combined with a net adapt method to create it. In terms of accuracy and latency, it has outperformed previous MobileNets. The authors added again some concepts, including squeeze-and-excitation models and more modern activation functions. Squeeze-and-excitation (SE) modules are generic additions to any CNN architecture(Howard et al., 2019). The idea behind them is to give the model the ability to weigh certain channels or regions of a feature map more for a certain input. A global pooling operation at the First layer of the module makes the module cheap to compute since all the following layers act on a spatial resolution of 1x1. Besides the usual ReLU, MobileNetV3 investigates the effect of applying different activation functions. Thus, inverted residual bottlenecks, squeeze and excite blocks made up MobileNet V3. Figure 3.9 shows a building block of MobileNet V3.

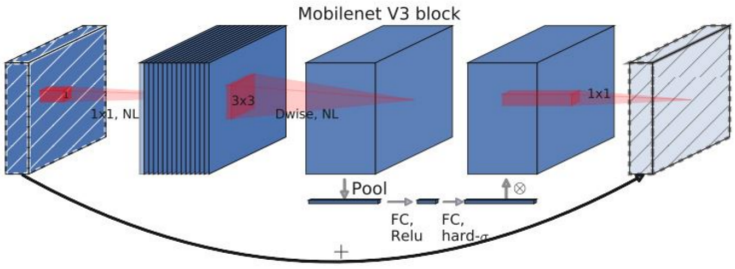


Figure 3.9: Building block of MobileNet V3 .(Howard et al., 2019)

3.5.5 DenseNet

A densely connected convolutional network (DenseNet) is a network presented by(Huang et al., 2017) in 2017. It is made up of densely connected CNN layers. The outputs of each layer in a dense block are connected. When CNNs get deeper and deeper, information about the input or gradient that passes through several layers tends to vanish by the time it reaches the network's end (or beginning). DenseNet tackles this problem with a simple connectivity pattern called dense connectivity. It provides full information flow across levels in the network by directly connecting all layers using feature maps of the same size. Each layer takes data from all previous levels as input and then sends its own feature maps to all subsequent layers (Gu, 2017). The authors argue in (Huang et al., 2017) that the network required fewer parameters than standard convolutional networks since it did not need to relearn redundant feature maps. DenseNet architecture differs from ResNet 28 in that it uses a concatenation of the previous input with the output of the current layer rather than adding several levels. DenseNet uses a composite function incorporating batch normalization and ReLU After a 3 ×3 convolution layer. the composite function's result is concatenated with the input before being passed to the next composite function. A bottleneck 1×1 convolution layer with 128 filters is used to start the composite function. This is done to lower the size of the input feature maps and increase the efficiency of the larger convolution. Following that, batch normalization and ReLU activation are performed with the 3×3 convolution layers, which have just 32 filters. Each composite function only needs to perform a little piece because of the filter concatenation throughout the network. The network executes a set number of composite functions in a dense block of 40 before compressing it using a transition layer. This compression begins with a 1×1 convolution to reduce filter dimensionality, followed by a 2×2 average pool with a stride of two to half the output size (Berhold, 2019). The DenseNet architecture is

a powerful approach to improve the efficiency of each layer in a CNN and to apply it to the following convolutional layers and the dense classification layer. Figure 3.10 shows a graphical architecture of DenseNet.

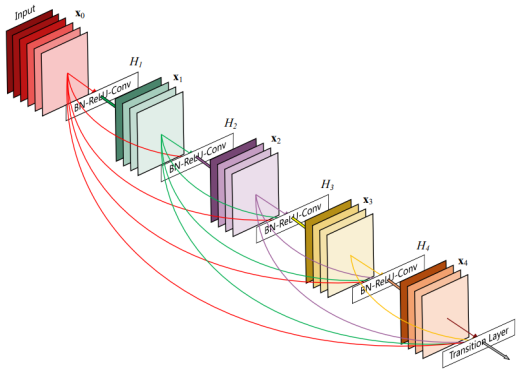


Figure 3.10: The Structure of DenseNet Architecture (Huang et al., 2017).

3.5.6 EfficientNet

(Tan and Le, 2019) was the first to present EfficientNet in their article "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" in 2019. The purpose of this article was to address the issue of scaling neural network architectures to improve accuracy. The availability of processing resources generally restricts model complexity in many convolutional neural network applications. EfficientNet allows for effective scaling of model complexity, allowing models to be better adaptable to computing resource constraints. EfficientNet also allows for efficient model scaling in cases when more resources are available. Scaling in three dimensions allows EfficientNet to be more flexible. Depth scaling, width scaling, and resolution scaling are the three scaling dimensions. Deeper networks may capture richer and more complicated information, but broader networks are quicker to train and can capture finer-grained features (Tan and Le, 2019). The term "resolution scaling" refers to the process of changing the resolution of input images. More fine-grained patterns are caught when higher-dimensional input images are used (Tan and Le, 2019). The various scaling dimensions are interdependent, and a "compound coefficient" is developed in (Tan and Le, 2019)to successfully scale networks at many dimensions. Eight possible designs were proposed with varying complexities, ranging from EfficientNet-B0 to EfficientNet-B7, with B0 having the lowest complexity and B7 having the most. Figure 3.11 shows the network scaling dimensions in EfficientNet20.

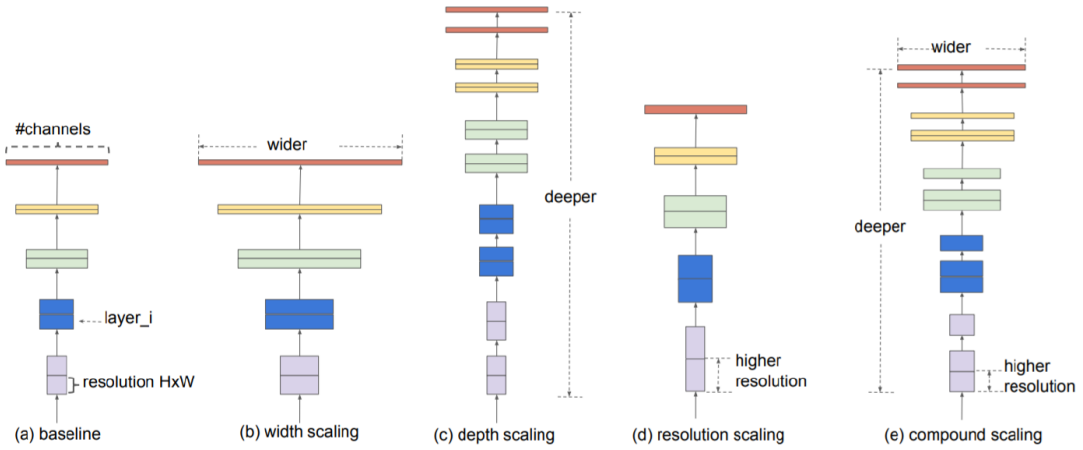


Figure 3.11: EfficientNet dimension scaling.(Tan and Le, 2019)

3.6 Deep learning Architectures for Image Segmentation

Image classification is a common application of CNNs, but directly using them for image segmentation is not feasible. To make CNNs suitable for segmentation tasks, several techniques have been developed. Successful approaches include the sliding window method, fully convolutional networks (FCNs), U-Net, SegNet, and DeepLab v3. These techniques involve modifying the network architecture to enable accurate and effective image segmentation alongside classification.

3.6.1 Sliding Window

The sliding window technique was among the methods employed to perform image segmentation using CNNs. It involves analyzing a small cropped region surrounding each pixel to determine its classification (Torralba, Murphy, and Freeman, 2004). However, accurately determining the optimal size of the cropped region poses a challenge, as it requires striking a balance between computational demands and the amount of contextual information considered. If the cropped area is too large, the computational cost increases significantly, while a too-small region may lack sufficient information for accurate segmentation. Compared to more contemporary approaches, the sliding window method is computationally intensive and relatively inefficient (Lux, 2018).

3.6.2 Fully Convolutional Networks

One of the significant advancements in image segmentation is attributed to the Fully Convolutional Network (FCN), a variant of the Convolutional Neural Network (CNN) introduced by (Long, Shelhamer, and Darrell, 2015) in 2015. Unlike traditional CNNs, the FCN eliminates fully connected layers and replaces them with convolution layers, enabling analysis of images of varying sizes. By employing convolution layers, the FCN computes nonlinear filters for each layer's output vectors, resulting in a network capable of processing inputs of any size and generating outputs with corresponding spatial dimensions. The FCN also retains spatial information from the input, which is crucial for semantic segmentation, a task involving both localization and classification. Figure 3.12 provides an example of this transformation. Although the FCN accommodates inputs of arbitrary sizes, the output resolution is reduced through convolutions without padding. This design choice minimizes the size of filters and computational requirements, resulting in a coarse output that is downscaled by a factor equal to the pixel stride of the output units' receptive fields (Long, Shelhamer, and Darrell, 2015)(Yi, 2020).

In the Fully Convolutional Network article by(Long, Shelhamer, and Darrell, 2015), it is noted that the final fully connected layer can be viewed as a 1x1 convolution that covers the entire region. When dense layers are employed, the input size is fixed, necessitating scaling when different input sizes are desired. However, by replacing dense layers with convolutions, this constraint is eliminated. When a larger input image size is provided, the output not only includes class predictions, as in the case of typical input-sized images, but also a feature map. Moreover, the observed behavior of the final feature map corresponds to the heatmap of the desired class, making it valuable information for segmentation tasks. Since the feature map obtained at the output layer is downsampled due to the performed convolutions, it is beneficial to upsample it using an interpolation approach. In the network architecture, the encoder handles downsampling, while the decoder handles upsampling—a common pattern in various architectures that involves reducing size with the encoder and then increasing the sample rate with the decoder (Posada Moyano, 2021).

3.6.3 SegNet

SegNet is an end-to-end architecture introduced by Badrinarayanan et al. (2017) (Badrinarayanan, Kendall, and Cipolla, 2017) as a more computationally efficient and generalizable alternative to Fully Convolutional Networks (FCNs). One of its key features is the

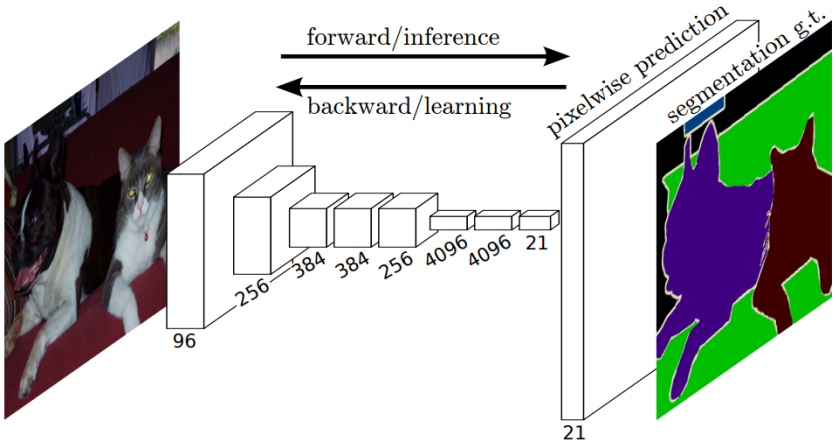


Figure 3.12: Fully convolutional networks may effectively train to produce dense predictions for per-pixel tasks as semantic segmentation.(Long, Shelhamer, and Darrell, 2015)

utilization of an up-sampling layer, which allows for the expansion of feature maps without introducing learnable parameters. The overall structure of SegNet resembles that of an autoencoder, comprising an encoder and a decoder. In SegNet, the encoder network consists of 13 convolutional layers, mirroring the architecture of the VGG16 network. These convolutional layers are responsible for generating a set of feature maps through convolution operations with a filter bank. Batch normalization is applied to the feature maps, followed by the rectified linear unit (ReLU) activation function. Subsequently, the feature maps undergo sub-sampling using max-pooling with a window size of 2x2 and a stride of 2, providing translation invariance over small spatial shifts in the input image. The decoder network in SegNet has a symmetric structure to the encoder, with 13 layers corresponding to each layer in the encoder. To create sparse feature maps, the decoder uses the memorized max-pooling indices from the corresponding encoder feature map to upsample its input. On the other hand, dense feature maps are generated by convolving the feature maps with a trainable decoder filter bank. Batch normalization is applied to each of these feature maps, and the resulting high-dimensional feature representation is passed through a trainable softmax classifier, which performs pixel-wise classification. At each pixel, the predicted segmentation corresponds to the class with the highest probability.

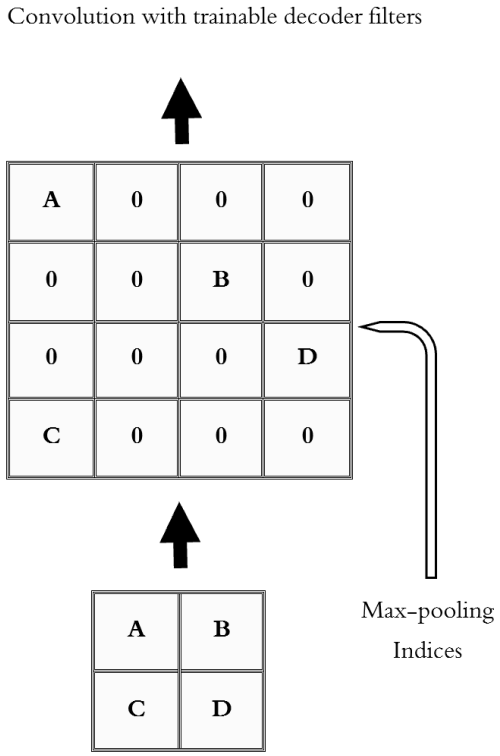


Figure 3.13: An example of upsampling applying max-pooling indices in SegNet.

3.6.4 Unet

The U-Net architecture is a variant of Fully Convolutional Networks (FCNs) that is widely used for semantic segmentation tasks. Its name "U-Net" originates from its U-shaped network structure. Although there are similarities between U-Net and SegNet (Badrinarayanan, Kendall, and Cipolla, 2017) architectures, they were developed for different segmentation purposes. In 2015, Ronneberger et al. (Ronneberger, Fischer, and Brox, 2015) introduced the U-Net architecture specifically for biomedical image segmentation and localization. The network architecture of U-Net is depicted in Figure 3.14. One of the notable advantages of U-Net over traditional CNNs is its ability to provide both classification and localization information in its output. Localization refers to the labeling of individual pixels in an image with class-specific information. U-Net surpasses FCNs by achieving more accurate segmentations even with limited training images. This is achieved by employing upsampling layers with a higher number of feature channels, allowing the propagation of contextual information to higher resolution layers.

The U-Net architecture is composed of two main paths: the contracting path (left) and

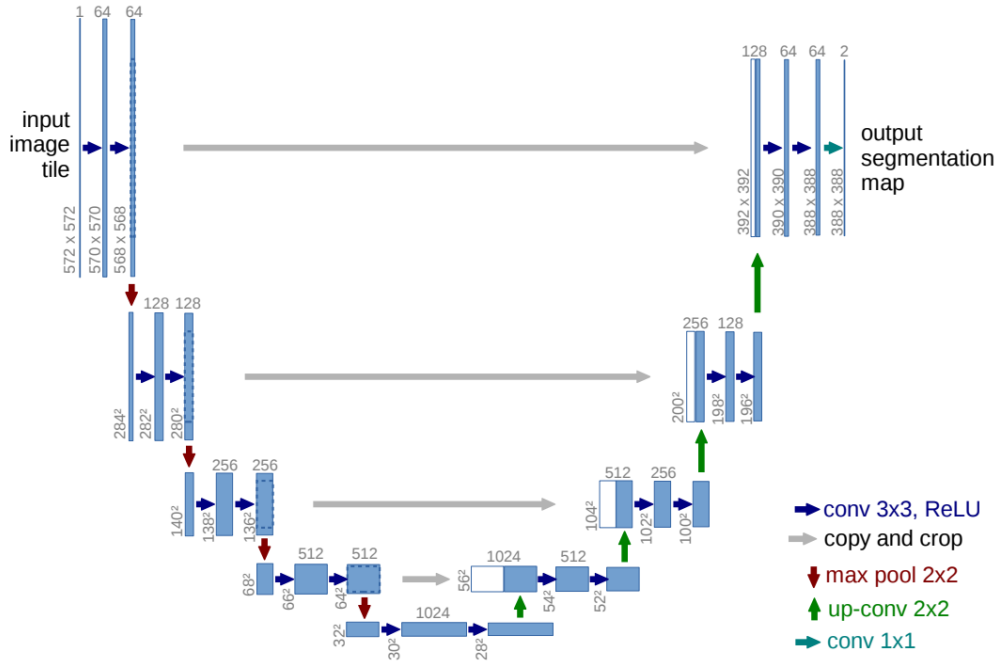


Figure 3.14: A schematic of the U-Net architecture.(Ronneberger, Fischer, and Brox, 2015)

the expanding path (right), connected by skip connections. This configuration follows the common encoder-decoder architecture notation. The contracting path functions as a convolutional network, where successive 3x3 convolutions are applied, followed by ReLU activation and 2x2 max pooling with a stride of 2. The number of channels is doubled at each downsampling stage.The expanding path initiates by performing upsampling on the feature map, which is followed by a 2x2 convolution operation (referred to as up-convolution) that reduces the number of channels by half. The upsampled feature map is then concatenated with the corresponding feature map from the contracting path. Two 3x3 convolutions with ReLU activation are subsequently applied (Ronneberger, Fischer, and Brox, 2015). The skip connections enable the transfer of pattern information from the encoder to the decoder, facilitating the retrieval of fine-grained features during predictions (Minaee et al., 2021).Compared to the original Fully Convolutional Network (FCN), U-Net is designed to capture and exploit contextual information more effectively. While FCN may not efficiently consider global contextual information, U-Net addresses this limitation by leveraging contextual information more efficiently (Minaee et al., 2021). The U-Net architecture was initially employed for biomedical image segmentation tasks with limited training data. To augment the training data, data augmentation techniques were

utilized. The results demonstrated that U-Net achieved remarkable segmentation performance in the field of medical image analysis (Ronneberger, Fischer, and Brox, 2015)(Borgersen, 2021).

3.6.5 Deep Lab

DeepLab is a segmentation model created and open-sourced by Google. DeepLab's architecture has evolved over the years, from DeepLabV1 in 2014 to DeepLabV3+ in 2019. Each of the improved versions introduces new concepts and dramatically increases performance. DeepLab, unlike the previous architectures, employs a method called Atrous Convolution to downsample the feature maps instead of max-pooling. Atrous convolution efficiently increases the field of vision of filters without increasing the number of parameters or processing.

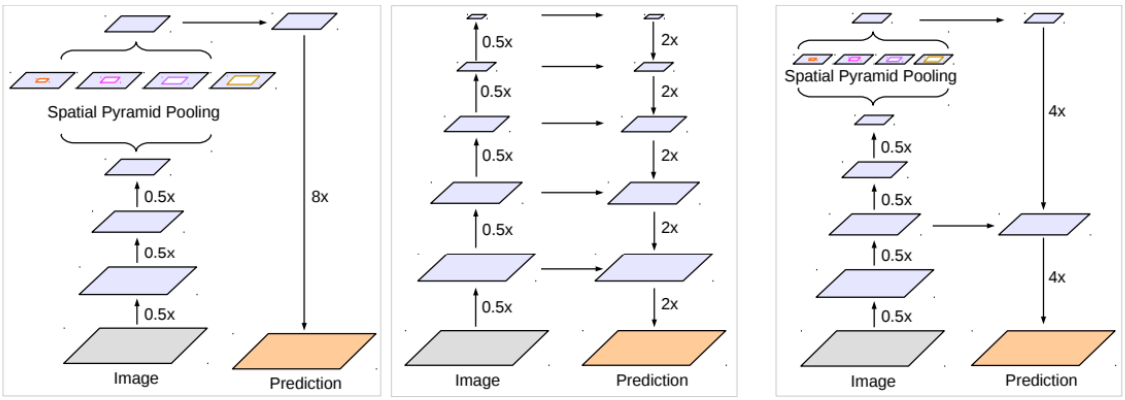
DeepLabV1's (Chen et al., 2014) success in semantic segmentation is owing to various improvements. Due to DCNN invariance, the advances address diminishing feature resolution and reducing localization accuracy. The spatial resolution is significantly reduced as a result of successive pooling operations. DeepLabV1 removes the down-sampling operator from the final few layers of DCNN and instead up-samples the filters in succeeding convolutional layers, resulting in higher sampling rate feature maps. DeepLabV1 also employs a fully connected Conditional Random Field to capture small details (CRF).

DeepLabV2 (Chen et al., 2017a) addresses the issue of object existence at multiple scales to boost the effectiveness of DeepLabV1 architecture. Atrous Spatial Pyramid Pooling (ASPP) is the recommended solution .The objective is to fuse numerous atrous convolutions with various rates to the input feature map. Since objects of the same class might have varying scales in the image, ASPP considers the different scales, which improves accuracy.

The Deep Lab v3 (Chen et al., 2017b) semantic segmentation system departs from the Encoder-Decoder design by restricting the amount of downsampling operations utilized in the encoding path. Using atrous convolutions with increasing dilation factors, it achieves a field of view comparable to models with a higher number of max-pooling layers (Hernandez, 2019). DeepLabV3 (Chen et al., 2017b) was designed to capture sharper object boundaries by gradually recovering spatial information. DeepLabV3's design is

based on a unique encoder-decoder architecture with atrous separable convolution. Many computer vision tasks, including object detection, human position estimation, and semantic segmentation, have effectively used the basic encoder-decoder structure. DeepLabV3 employs depthwise separable convolution in addition to the encoder-decoder network to improve computational efficiency. The basic convolution is factored into a depthwise convolution, which is then followed by a pointwise convolution. The depthwise convolution, in particular, executes a spatial convolution independently for each input channel. The pointwise convolution is used to combine the depthwise convolution output and to modify the number of channels.

DeepLabV3+ (Chen et al., 2018) is the improved version of DeepLabV3 by including a basic yet effective decoder module to recover the object boundaries, as illustrated in Figure 3.15 The encoder part employs Modified Aligned Xception(Dai et al., 2017) as its primary feature extractor. Depthwise separable convolution with striding replaces all max-pooling procedures. The encoder works by decreasing the resolution of the input by a factor of 16. Instead of simply upsampling the features by 16, the decoder first upsamples them by a factor of 4 and concatenates them with the matching low-level features. Following concatenation, 3×3 convolutions are applied, and the feature map is upsampled by a factor of 4 to do the prediction.



(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Conv

Figure 3.15: The proposed model DeepLabv3+ which employs the spatial pyramid pooling module (a), with the encoder-decoder structure (b).(Chen et al., 2017b)

3.7 Conclusion

In this chapter we have provided an introduction to deep neural networks, with a focus on convolutional neural networks (CNNs). We have explored the essential building blocks of CNNs, including convolution, non-linear activation functions, pooling, upsampling, and fully-connected layers. These operations play a crucial role in the CNN algorithm, allowing for effective feature extraction and representation. Additionally, we have discussed the application of deep convolutional architectures for image segmentation and classification tasks. CNNs have demonstrated remarkable success in these domains, achieving state-of-the-art results on various datasets. By leveraging the hierarchical nature of CNNs and their ability to learn meaningful features, we can tackle complex computer vision tasks with improved accuracy and efficiency. The knowledge gained from this chapter provides a solid foundation for understanding and implementing CNNs in practical applications.

Chapter 4

SKIN CANCER PROCESS AND LITERATURE REVIEW

4.1 Introduction

Skin cancer is a prevalent and potentially fatal disease that affects a significant number of people worldwide (Ashraf et al., 2020). It is divided into two main types: melanoma and non-melanoma (Elgamal, 2013). Melanoma is a dangerous and uncommon form of skin cancer that arises when melanocytes, the cells responsible for producing skin pigment, undergo uncontrolled growth. Early detection is crucial for effective treatment, as advanced melanoma can metastasize and lead to severe consequences (Dildar et al., 2021). In contrast, non-melanoma skin cancers are generally easier to treat. Therefore, timely detection plays a vital role in successful skin cancer therapy (*What's the difference between melanoma and non-melanoma skin cancer?* 2016).

The conventional method for diagnosing skin cancer is through a biopsy, in which a sample of a suspicious skin lesion is taken and examined to determine malignancy. However, this approach is often is difficult and expensive (Dildar et al., 2021). Consequently, there is a growing need for alternative approaches that can rapidly and cost-effectively diagnose skin cancer using computer technology. This has prompted researchers to explore literature review approaches to investigate skin cancer symptoms and distinguish between melanoma and non-melanoma cases.

Deep learning, a field inspired by the human nervous system, has made significant strides in various scientific and industrial domains. Its rapid advancement in processing biomedical data, including medical images and records, has enabled researchers to extract more

precise and reliable information. Deep learning has demonstrated impressive results in a wide range of medical image processing tasks, primarily due to the increasing availability of biomedical data (Weber, Mandl, and Kohane, 2014) (Filimon and Albu, 2014)(Serener and Serte, 2020)(Zhang et al., 2021).

In recent years, a variety of deep learning algorithms have been employed in computer-based skin cancer detection. This section aims to present a comprehensive literature review that examines both classical techniques and current approaches utilizing deep learning methodologies for the analysis of skin lesion images. By harnessing the power of deep learning and capitalizing on the abundance of biomedical data, researchers strive to develop more accurate and efficient systems for the early detection of skin cancer.

This section is subdivided into four main sections. Section 4.2 describes steps for skin cancer detection. The skin image pre-processing step is presented in Section 4.4. Section 4.5, and 4.6 summarize different related works for skin image segmentation and classification.

4.2 Skin cancer detection steps

Automated detection of skin lesions is a critical step in computerized melanoma diagnosis systems (Celebi et al., 2009)(Ganster et al., 2001)(Celebi et al., 2015). Accurate segmentation of skin lesions from surrounding tissues is necessary to identify distinct features for melanoma classification (Ganster et al., 2001) (Schaefer et al., 2014). However, segmentation is challenging due to variations in color, texture, shape, size, and lesion location within the image frame. Additionally, lesion imaging poses complexities such as variations in skin tone and the presence of artifacts like hair, blood vessels, ruler marks, air bubbles, ebony frames, and color illumination. These intricate visual features of skin lesions are illustrated in Figure 4.1 and are described below (Al-Masni et al., 2018)(Adegun and Viriri, 2021): .

- **Size and shape variations**: the accurate identification of skin lesions is hindered by several factors. First, size and shape variations among skin lesions contribute to the complexity of analyzing these images. Lesions can vary greatly in their location, size, and overall form. As a result, image analysis approaches often involve image pre-processing as a necessary step to ensure accurate analysis.
- **Artifacts presence**: the presence of artifacts further affects the ability to identify skin lesions in images. These artifacts, such as hair, bubbles, and blood vessels, introduce

compromising signals that can interfere with manual interpretation and computer-aided segmentation procedures.

- **Low contrast**: the poor contrast between the lesion region and the surrounding skin presents challenges in precisely segmenting the lesions.
- **Irregular fuzzy boundaries**: Irregular and fuzzy boundaries in skin lesion images create difficulties for algorithms attempting to refine contours and locate lesion boundaries. Obtaining precise images of lesion boundaries during the preprocessing stage, particularly for simple asymmetry prediction, can be challenging.
- **Color illumination** : The illumination of the dermoscopic images can be affected by the color, texture, light rays, and reflections of the skin lesion image, producing the images multiresolution.

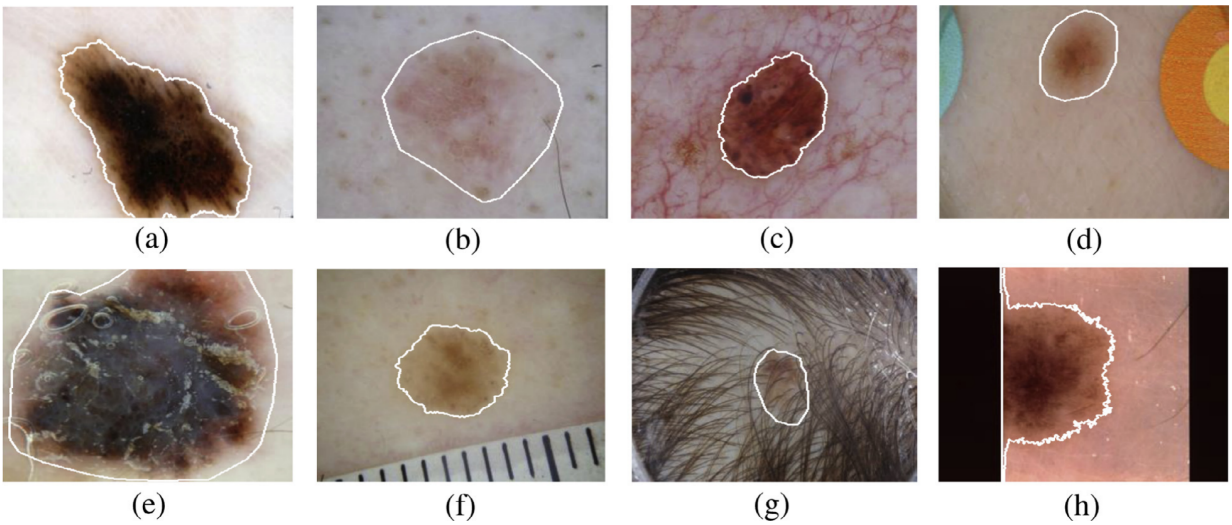


Figure 4.1: Examples of challenging skin lesions cases (a) irregular fuzzy boundaries, (b) low contrast, (c) blood vessels, (d) color illumination, (e) bubbles, (f) ruler mark artifact, (g) hair artifact, and (h) frame artifact.(Al-Masni et al., 2018)

Broadly, the entire pipeline for detecting and diagnosing melanoma cancer has been divided into key processing stages, encompassing tasks such as image pre-processing, image segmentation, feature extraction, and the classification of lesion images (Koundal and Sharma, 2019) (Ünver and Ayan, 2019), illustrated in Figure 4.2.

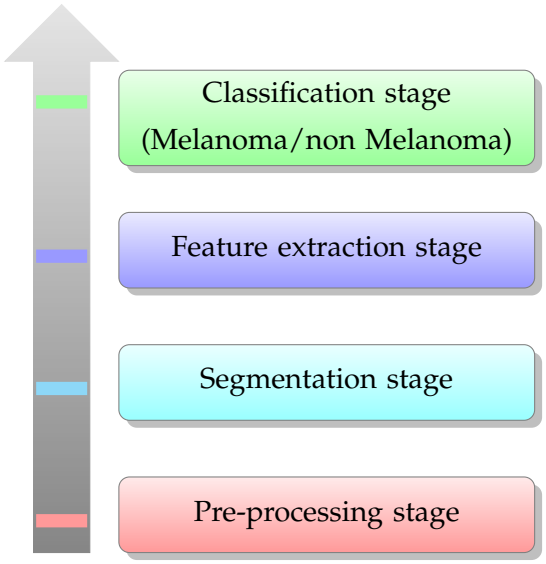


Figure 4.2: Pipeline process of melanoma cancer diagnosis.

4.3 Database discription

In this study, we utilized four publicly available datasets of skin lesions to train and evaluate the proposed skin cancer detection framework. The datasets used are as follows:
The PH2 dataset (Mendonça et al., 2015) was obtained from Hospital Pedro Hispano in Matosinhos, Portugal, and the University of Porto. It comprises 200 dermoscopic images, consisting of 80 common nevi, 80 atypical nevi, and 40 melanomas. The images in the PH2 dataset vary in size, ranging from 553 x 763 to 577 x 769 pixels.
The ISIC 2016 skin lesion challenge database (Gutman et al., 2016) provides skin lesion images categorized as malignant or benign. It consists of 900 training images, with 173 images classified as melanoma and 727 as non-melanoma. Additionally, there are 379 testing images, including 75 melanoma and 304 non-melanoma images. The images in the ISIC 2016 dataset have varying sizes, ranging from 566 x 679 to 2848 x 4228 pixels.
The ISIC 2017 (Codella et al., 2018) skin lesion challenge contains a total of 2750 images, with 2000 RGB images in the training set, 600 images in the testing set, and 150 images in the validation set. The images' sizes in the ISIC 2017 dataset range from 453 x 679 to 4499 x 6748 pixels.
The ISIC 2018 (Codella et al., 2019)includes 1000 test images from the HAM10000 dataset Tschandl, Rosendahl, and Kittler, 2018 and 2594 RGB dermoscopic training images, and 2594 RGB dermoscopic training images. The images in the ISIC 2018 dataset have spatial resolutions ranging from 540 x 722 to 4499 x 6748 pixels.

4.4 Skin image preprocessing

Preprocessing is primarily used to prepare images for more efficient processing and accurate feature recognition. Preprocessing an image can minimize irrelevant information, increase the intensity of relevant information, simplify the data, and enhance dependability. This procedure included data augmentation, image enhancement, image resizing, and normalization. Image preprocessing procedures are described below (Zhang et al., 2021):
Normalization: is a common technique used in machine learning and image processing to rescale data to a common range, typically [0, 1] or [-1, 1]. Its purpose is to ensure that different features or variables have similar scales, preventing any one feature from dominating the learning process due to its larger magnitude. By rescaling the data, normalization can help improve model performance, convergence, and optimization efficiency. It is not a guaranteed solution to data alteration or failure, but it can prevent biases caused by varying scales and facilitate better comparison and interpretation of feature importance.
Resize: Skin lesions are often small compared to the overall size of skin images (Gomez et al., 2007) (Kaymak, Esmaili, and Serener, 2018). To process these images efficiently for deep learning tasks, preprocessing is necessary due to the high resolution of the original lesion images, which can result in a high computation cost (Harangi, 2018). To maintain the integrity of the lesion contour, a common approach is to first crop the images to the center and then resize them as needed. Scaling and clipping (Mahbod et al., 2020) techniques are commonly employed to adjust the images to the appropriate size, considering both computational requirements and information density.
Data augmentation: is a technique used to address the limited data problem in skin disease identification. It involves applying transformations like rotation, random cropping, and noise addition to the existing dataset. These transformations create new samples that expand the dataset and improve the model's ability to generalize. By simulating different perspectives and variations, data augmentation enhances the model's learning capacity and reduces the risk of overfitting.

Figure 4.3 depicts different image processing approaches that may be used to expand the image database (Howard, 2013).

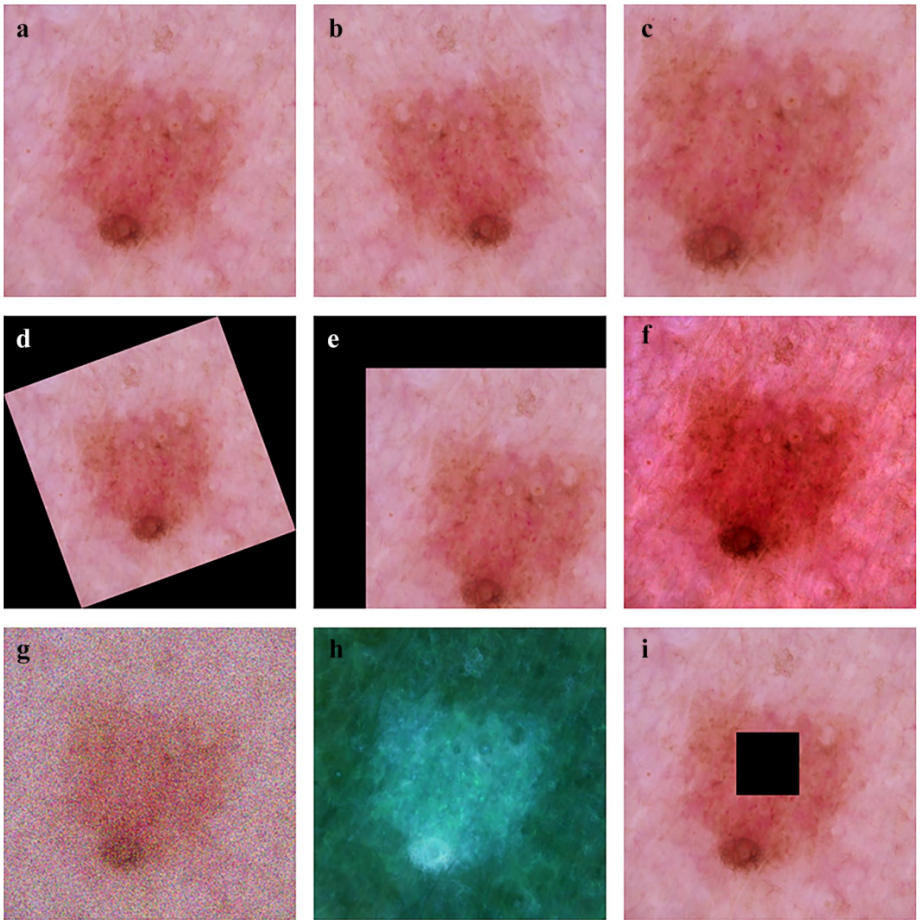


Figure 4.3: Examples of data argumentation (a) Original image; (b) Flip; (c) Random crop; (d) Rotation; (e) Shift; (f) Color jittering; (g) Noise; (h) Standardization; and (i) Paste .(Zhang et al., 2021)

4.5 Related work for skin lesion Segmentation

Skin lesion segmentation methods have been extensively researched, employing traditional computer vision, machine learning, and deep learning techniques. While traditional computer vision approaches have been used, they often yield inadequate results on low-contrast skin images. In contrast, deep learning techniques have emerged as a promising solution for biomedical image segmentation, including skin lesion segmentation. These methods leverage neural networks, such as convolutional neural networks (CNNs), and specialized architectures like fully convolutional networks (FCNs) and U-Net. Deep learning approaches have shown superior performance in automatically segmenting skin lesions, surpassing the capabilities of traditional methods. This section presents traditional and current approaches for skin lesion segmentation.

4.5.1 Performanace Evaluation

Six widely used metrics Recall (Sensitivity (SENS)), Accuracy (ACC), Precision, F1_Score, Jaccard index, and Specificity (SPEC)—were utilized to evaluate the performance of the proposed system. These are the metrics' definitions:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4.1}$$

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{4.2}$$

$$Specificity = \frac{TN}{(TN + FP)} \tag{4.3}$$

$$DiceCoefficient = \frac{(TP \times 2)}{(2TP + FP + FN)} \tag{4.4}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{4.5}$$

$$F1_{Score} = \frac{2 * (Recall * Precision)}{Recall + Precision} \tag{4.6}$$

$$Jaccardindex = \frac{TP}{(TP + FP + FN)} \tag{4.7}$$

4.5.2 Previous Related Works to Lesion Segmentation

The segmentation and identification of lesions in dermoscopic images are widely recognized as highly challenging tasks due to the presence of numerous abnormalities. Consequently, extensive research has been conducted to explore various algorithms and approaches for effectively extracting the affected lesion regions from skin cancer imaging. (Emre Celebi et al., 2008) described a rapid and unsupervised method for recognizing boundaries in dermoscopy images of pigmented skin lesions. They introduced the statistical region merging technique as the basis of their approach. This technique involves merging neighboring regions iteratively based on statistical similarity measures. By applying this technique to dermoscopy images, they aimed to automatically identify and

delineate the boundaries of pigmented skin lesions without the need for manual intervention or supervision. The proposed method offered a time-efficient and automated solution for boundary recognition in dermoscopy images, which has important implications for dermatology and the analysis of skin lesions. (Yueksel and Borlu, 2009) introduced a method for precise segmentation of pigmented skin lesion boundaries from healthy skin. They employed the type-2 fuzzy logic technique to determine an automated threshold value. This technique is advantageous as it allows for uncertainty modeling, making it more robust in handling variations in the boundaries of skin lesions. By utilizing the type-2 fuzzy logic technique, they aimed to develop an effective and automated thresholding approach that could accurately differentiate between pigmented lesions and healthy skin. Their method provided a significant contribution to the field of dermatology by improving the accuracy of boundary segmentation in the analysis of pigmented skin lesions. (Celebi et al., 2010) introduced a fusion-based approach for accurately segmenting lesion borders in dermoscopic images. They utilized a combination of thresholding methods, including Otsu's algorithm, Kapur's algorithm, Huang's algorithm, and Kittler's algorithm. By combining the outputs of these thresholding methods, the authors aimed to enhance the accuracy and robustness of lesion border segmentation. This fusion-based approach was designed to overcome the challenges associated with the variability in dermoscopic images. The fusion of multiple thresholding methods allowed for improved adaptation to different image characteristics, resulting in more precise segmentation results. The proposed approach provided a comprehensive and effective solution for accurately delineating lesion borders in dermoscopic images, contributing significantly to the field of dermatology and the analysis of skin lesions. (Garnavi et al., 2010) introduced an automated segmentation technique for skin lesions. They analyzed different color channels using color space analysis and clustering with histogram thresholding. The optimal color channel was chosen for lesion segmentation, and Otsu's thresholding method was used to accurately separate the lesions from the background. This approach enabled automated and reliable skin lesion segmentation. (Zhou et al., 2011) combined the mean shift algorithm with gradient vector flow (GVF) for image segmentation. The mean shift operation is integrated into the GVF cost function to guide the segmentation process in the right direction. This integration allows for improved accuracy and effectiveness of image segmentation by leveraging the local structure information provided by mean shift and the global contour evolution capability of GVF. (Abbas, Fondón, and Rashid, 2011) presented an unsupervised method for segmenting multiple lesions using modified Region-based Active Con-

tours (RACs). Their approach utilized iterative thresholding to automatically establish the level set and imposed maximum smoothing limits on the Courant–Friedreichs–Lewy (CFL) function to ensure curve stability during the segmentation process. The goal was to improve segmentation accuracy while reducing artifacts in the results. (Nowak, Ogorzalek, and Pawlowski, 2012) created an adaptive filter influenced by Swarm Intelligence (SI) optimization techniques for recognizing pigment network architectures. Their approach involved applying filters (agents) randomly to different areas of the image and allowing each filter to adapt its output based on information from neighboring filters. This collaborative behavior among the filters enhanced the accuracy of recognizing pigment network architectures by collectively analyzing and adapting their outputs. (Castillejos et al., 2012) developed a technique for segmenting and detecting the boundary of a skin lesion using clustering techniques based on the wavelet transform. The approach involved grouping homogeneous pixels into clusters using algorithms such as K-Means, Fuzzy C-Means, and Cluster Preselection Fuzzy C-Means. By leveraging the wavelet transform and clustering algorithms, the technique facilitated accurate segmentation and boundary detection of skin lesions. (Lee and Chen, 2014) suggested a technique for distinguishing malignant borders from skin imaging using the fuzzy c-mean approach (FCM) and the type-2 fuzzy set algorithm. Their method aimed to find an optimal threshold value to effectively differentiate malignant borders. By leveraging the FCM algorithm and the type-2 fuzzy set algorithm, their technique accurately identified malignant borders while managing uncertainty and imprecision. (Jaworek-Korjakowska and Tadeusiewicz, 2014) presented a technique for detecting and classifying border irregularity, which is an important parameter in the widely used dermoscopy diagnostic algorithm called the ABCD rule. Their approach utilized image processing and pattern recognition techniques to automatically identify and classify irregular borders in dermoscopy images, contributing to the assessment of lesion malignancy. (Glaister, Wong, and Clausi, 2014) presented a texture-based skin lesion segmentation method for accurately segmenting skin lesions based on their unique texture characteristics. Their approach involved utilizing an illumination-corrected image to learn a collection of typical texture distributions and calculating a texture distinctiveness measure for each distribution. By analyzing the distinctiveness of the texture patterns, their method aimed to achieve precise segmentation of skin lesions. (Bi et al., 2016) introduced an automated melanoma identification approach for dermoscopy images. Their method utilized multi-scale lesion-biased representation (MLR) and combined reverse classification (JRC). The MLR representation allowed for the description of

skin lesions using closely related histograms generated from various rotations and scales. By incorporating information from multiple scales and orientations, the MLR representation provided a comprehensive understanding of the lesion's characteristics. The integration of MLR with JRC further improved the accuracy of melanoma identification in dermoscopy images. (Pennisi et al., 2016) proposed an approach for segmenting lesions. Their technique began by employing the closing technique to eliminate unwanted elements such as hair and outlier pixels. Subsequently, two segmentation algorithms were simultaneously applied, producing two separate images. One image represented the skin region, while the other was generated through edge detection using Delaunay Triangulation. The researchers then combined these two images, leveraging their complementary information, to obtain the final segmentation of the lesion. This method aimed to enhance the precision of lesion segmentation by integrating multiple segmentation algorithms and edge detection techniques.

4.5.3 Current Technologies Related to Skin Lesion Segmentation.

In the realm of skin lesion segmentation, various research studies have explored the utilization of different deep learning architectures, including Fully Convolutional Networks (FCN), U-Net, SegNet, and DeepLab. Several studies have explored and developed effective models using these architectures. Here are some discussions on these notable architectures:

- *Fully Convolutional Networks (FCN)* : (Bi et al., 2017) developed a multi-stage fully convolutional network (FCN) approach for skin lesion segmentation. They addressed the limitations of existing methods, which often struggled with fuzzy boundaries, low contrast, inhomogeneous textures, and artifacts in skin lesions. The proposed method utilized FCNs to combine low-level appearance information with high-level semantic information. Multiple FCNs were trained at different stages to capture complementary visual characteristics of skin lesions. They introduced a parallel integration method to combine the outputs of these stages, resulting in improved segmentation accuracy with well-defined lesion boundaries. The performance of the approach was evaluated on the ISBI 2016 Skin Lesion Challenge dataset and the PH2 dataset, where it outperformed other state-of-the-art methods. (Yuan, 2017) presented fully convolutional network that leveraged 19-layer deep convolutional neural networks (CNNs) trained end-to-end, without relying on prior knowledge of the data. They proposed strategies to address the challenges of limited training

data and introduced a novel loss function based on Jaccard distance to handle the pixel imbalance issue. The effectiveness, efficiency, and generalization capabilities of the proposed framework were evaluated on two publicly available databases: the ISBI 2016 dataset and the PH2 database. Experimental results demonstrated that the method outperformed state-of-the-art algorithms on these databases. (Al-Masni et al., 2018) created a segmentation method called Full Resolution Convolutional Networks (FrCN) for skin lesion segmentation. FrCN directly learned the full resolution features of each pixel in the input data, eliminating the need for preprocessing or post-processing operations. The method was evaluated on the ISBI 2017 Challenge dataset and the PH2 dataset, comparing it with other deep learning segmentation approaches. FrCN achieved high segmentation performance. It outperformed other methods, including FCN, U-Net, and SegNet, in terms of the Jaccard index and segmentation accuracy. The authors concluded that utilizing the full spatial resolutions of the input image improved the segmentation performance by capturing specific and prominent features . (Huang, Zhao, and Yang, 2019) have developed a segmentation technique called Object Scale-Oriented Fully Convolutional Networks (OSO-FCNs) for skin lesion segmentation, with a specific focus on melanoma detection. Their approach involves adapting the VGG-16 network into FCNs and introducing a training strategy called OSO training, which takes into account the scale of lesions. The training dataset is divided into subsets based on the object occupation ratio, and three scale-oriented FCNs are trained. The proposed method is evaluated on the ISIC2016 dataset, and the experimental results demonstrate that the OSO-FCNs achieve high segmentation accuracy, outperforming or performing closely to other state-of-the-art algorithms. The study highlights the potential of the OSO-FCN technique for accurate and robust skin lesion segmentation, particularly in the context of melanoma detection. (Shan et al., 2020) suggested a segmentation approach called FC-DPN for automatic skin lesion segmentation in dermoscopy images. The method combines the advantages of fully convolutional networks (FCN) and dual path networks (DPN) to tackle challenges such as diverse lesion characteristics, low contrast, and image artifacts. FC-DPN incorporates sub-DPN blocks to enhance feature reusage and re-exploitation, resulting in improved feature representation and discrimination. Experimental results on the modified ISBI 2017 Skin Lesion Challenge test dataset and the PH2 dataset showcased the superior performance of FC-DPN compared to FC-DenseNets and other existing segmentation algorithms. (Kaymak, Kay-

mak, and Ucar, 2020) developed an automatic skin lesion segmentation technique to aid in the diagnosis of melanoma. They employed modified Convolutional Neural Network (CNN) architectures to create Fully Convolutional Network (FCN) architectures for this task. Four different FCN architectures, including FCN-AlexNet, FCN-8s, FCN-16s, and FCN-32s, were utilized for semantic segmentation of skin lesions. The study was conducted on the ISIC 2017 dataset, and the performance of these architectures was evaluated for the first time on this dataset. The FCNs were trained separately after preprocessing the dataset images, and their accuracies and Dice coefficients were calculated on a validation dataset. The experimental results showed that the proposed FCN architectures were effective in accurately segmenting skin lesions. (Öztürk and Özkaya, 2020) introduced an enhanced FCN (iFCN) architecture for automatic skin lesion segmentation. The proposed iFCN addresses challenges related to undesirable residues, unclear boundaries, and color variations in skin lesion images. By incorporating spatial information and a residual structure, the iFCN achieves precise segmentation without the need for preprocessing or post-processing steps. The performance of the iFCN was evaluated on the ISBI 2017 Challenge and PH2 datasets, demonstrating its effectiveness in accurately segmenting skin lesions.

- *U-net Architecture* : (Vesal, Ravikumar, and Maier, 2018) introduced SkinNet, a modified version of the U-Net architecture, to address the challenges of skin lesion segmentation. The goal was to develop an accurate and automated approach for the early detection and segmentation of skin lesions, which is crucial for improving patient survival rates in cases of skin cancer. SkinNet incorporated dilated and densely block convolutions, enabling the integration of multi-scale and global context information during the training process.The performance of SkinNet was evaluated using the ISBI 2017 challenge dataset, and comparisons were made with other state-of-the-art techniques. SkinNet demonstrated superior performance in terms of metrics such as the Dice coefficient, Jaccard index, and sensitivity. The results highlight the effectiveness of SkinNet in accurately segmenting skin lesions and emphasize its potential for advancing the early diagnosis and treatment of skin cancer. (Nguyen et al., 2020) proposed an enhanced version of the U-Net architecture for skin lesion segmentation. Their approach incorporates EfficientNetB4 in the encoder and utilizes residual blocks from the ResNet architecture in the decoder. The proposed method aims to address challenges such as artifacts, blurred boundaries, and irregular edges

in dermoscopic images. Evaluation on the ISIC 2017 and 2018 datasets demonstrates improved performance, measured by metrics such as the Dice coefficient and Jaccard index. This work contributes to the advancement of automated melanoma diagnosis by improving the accuracy and efficiency of skin lesion segmentation. (Zafar et al., 2020) proposed an approach called Res-Unet for automating the segmentation of skin lesion boundaries. The primary objective of their work was to improve the accuracy and efficiency of skin lesion analysis in the context of diagnosing and treating skin cancer. The Res-Unet model was developed by combining the U-Net and ResNet architectures, leveraging their respective strengths in image segmentation and feature extraction. The researchers trained the model using the ISIC 2017 dataset and evaluated its performance on both the ISIC 2017 test set and the PH2 dataset. Notably, they also incorporated image inpainting techniques to remove unwanted hair artifacts, further enhancing the segmentation results. The findings of their study demonstrated promising outcomes, showcasing the potential of the Res-Unet approach in accurately delineating skin lesion boundaries. (Phan et al., 2021) developed a deep learning method based on the U-Net architecture to address the challenges of accurately delineating skin lesions. Their approach incorporates lesion segmentation, boundary distance map regression, and contour detection tasks. By including these auxiliary tasks, the model improves object localization and pixel-wise classification in the transition region between lesion and healthy tissues. Additionally, the selective kernel modules help handle variations in lesion sizes. The proposed method was evaluated on ISBI2016, ISBI 2017, and PH2 datasets, demonstrating its effectiveness in skin lesion segmentation. (Vesal, Ravikumar, and Maier, 2018) introduced SkinNet, a modified version of the U-Net architecture, to address the challenges of skin lesion segmentation. The goal was to develop an accurate and automated approach for the early detection and segmentation of skin lesions, which is crucial for improving patient survival rates in cases of skin cancer. SkinNet incorporated dilated and densely block convolutions, enabling the integration of multi-scale and global context information during the training process.The performance of SkinNet was evaluated using the ISBI 2017 challenge dataset, and comparisons were made with other state-of-the-art techniques. SkinNet demonstrated superior performance in terms of metrics such as the Dice coefficient, Jaccard index, and sensitivity. The results highlight the effectiveness of SkinNet in accurately segmenting skin lesions and emphasize its potential for advancing the early diagnosis and treatment of

skin cancer. (Nguyen et al., 2020) proposed an enhanced version of the U-Net architecture for skin lesion segmentation. Their approach incorporates EfficientNetB4 in the encoder and utilizes residual blocks from the ResNet architecture in the decoder. The proposed method aims to address challenges such as artifacts, blurred boundaries, and irregular edges in dermoscopic images. Evaluation on the ISIC 2017 and 2018 datasets demonstrates improved performance, measured by metrics such as the Dice coefficient and Jaccard index. This work contributes to the advancement of automated melanoma diagnosis by improving the accuracy and efficiency of skin lesion segmentation. (Zafar et al., 2020) proposed an approach called Res-Unet for automating the segmentation of skin lesion boundaries. The primary objective of their work was to improve the accuracy and efficiency of skin lesion analysis in the context of diagnosing and treating skin cancer. The Res-Unet model was developed by combining the U-Net and ResNet architectures, leveraging their respective strengths in image segmentation and feature extraction. The researchers trained the model using the ISIC 2017 dataset and evaluated its performance on both the ISIC 2017 test set and the PH2 dataset. Notably, they also incorporated image inpainting techniques to remove unwanted hair artifacts, further enhancing the segmentation results. The findings of their study demonstrated promising outcomes, showcasing the potential of the Res-Unet approach in accurately delineating skin lesion boundaries. (Phan et al., 2021) developed a deep learning method based on the U-Net architecture to address the challenges of accurately delineating skin lesions. Their approach incorporates lesion segmentation, boundary distance map regression, and contour detection tasks. By including these auxiliary tasks, the model improves object localization and pixel-wise classification in the transition region between lesion and healthy tissues. Additionally, the selective kernel modules help handle variations in lesion sizes. The proposed method was evaluated on ISBI2016, ISBI 2017, and PH2 datasets, demonstrating its effectiveness in skin lesion segmentation.

- *SegNet Architecture*: (Ninh et al., 2019) developed a modified version of the SegNet architecture for skin lesion segmentation in dermoscopic images. Their approach aimed to enhance computational efficiency by reducing the downsampling and upsampling layers, while still achieving accurate segmentation results. The effectiveness of the proposed method was evaluated on the ISIC 2017 dataset, showing promising outcomes without specifying specific numerical values.

- *DeepLab Architecture*: (Azad et al., 2020) introduced Attention Deeplabv3+, an enhanced version of the Deeplabv3+ model for skin lesion segmentation. The proposed approach incorporated attention mechanisms in two stages to improve the segmentation performance. By assigning weights to channels and integrating different layers of the atrous convolution, the model focused on relevant information and achieved state-of-the-art performance on the ISIC 2017, ISIC 2018, and PH2 datasets. Table 4.1 summarizes current skin lesion segmentation methods.

These current technologies demonstrate the ongoing advancements in skin lesion segmentation using deep learning approaches, contributing to improved accuracy and performance in this field.

Table 4.1: Summary of current skin lesion segmentation methods.

Study	Year	Dataset	Preprocessing	Data augmentation	Input Size	Architecture	Measures		
							Acc	SE	Jacc
(Yuan, 2017)	2017	ISBI2017	✓	X	192 × 256	FCN			78.4%
(Bi et al., 2017)	2017	ISBI2016	X	X	768×560	FCN	94.70%	94.34%	85.84%
		PH2					94.24%	94.89 %	83.99 %
(Al-Masni et al., 2018)	2018	ISBI 2017	✓	✓	192 × 256	FCN	96.69%	85.40 %	-
(Huang, Zhao, and Yang, 2019)	2019	ISIC2016	X	X	height of 256 pixel	FCN	95.8%	92.5%	-
(Shan et al., 2020)	2020	ISBI2017	✓	✓	128 × 192		95.14%	84.56%	80.02%
		PH2					93.63%	94.77%	83.51%
(Kaymak, Kaymak, and Ucar, 2020)	2020	ISBI2017	X	X	250 ×250	FCN	93.7%		70.3%
(Öztürk and Özkaya, 2020)	2020	ISBI2017	X	X	-	FCN	95.30%	85.44%	78.34%
		PH2					96.92%	96.88%	87.1%
(Bissoto et al., 2018)	2018	ISIC2018	✓	✓	256×256	U-net			
(Vesal, Ravikumar, and Maier, 2018)	2018	ISBI 2017	X	X	512 × 512		93.2%	93%	76.67%
(Zafar et al., 2020)	2020	ISIC 2017	✓	X	256×256	U-net	-	-	77.2%
		PH2					-	-	85.4%
(Phan et al., 2021)	2021	ISIC2016	X	X	192 × 288	U-net	96.48%		87.08%
		ISIC 2017					94.5%	-	79.95 %
		PH2					96.01%	-	89.22 %
(Ninh et al., 2019)	2019	ISIC 2017	✓	X	144×192	SegNet	-	-	77.1%
	2020	ISIC 2017	X	X		Deeplab	96.98%	88.51%	96.98%

4.6 Related work for Skin image classification

4.6.1 Performanace Evaluation

After training a model and saving the best parameters, the test set was evaluated using the previously stored parameters. The ground truth targets given by the dataset were then used to compare the resulting predicted labels. The output of classifiers is assessed using a set of performance criteria in order to determine how closely the prediction resembles the truth. As a consequence, four sets are directly obtained after outcomes attribution computation:

- True Positives: The True Positives set comprises all of the malignancies identified as malignant by the classifier applied, and they are marked by TP.
- False positives: The set of false positives comprises all of the benign that were falsely identified by the classifier and marked as FP.
- True Negatives: The True Negatives collection comprises all of the benign l that were categorized as benign by the classifier employed and were noticed by TN.
- False negatives: The set of false negatives comprises all of the malignant that were mistakenly identified by the classifier employed, and they are labeled FN.

The goal is to get the FP and FN to zero. False positives need additional biopsies for doctors and analysts. False negatives, on the other hand, are more harmful due to the lack of medical care for these false negatives (unhealthy persons).
Since various papers use different databases, it is ideal to establish a percentage of detection or accuracy for comparison. Many performance equations are proposed and used in classification [. The following are the most often used performance measures in image detection:

Recall or Sensitivity (SENS): the percentage of true positives successfully detected This is the percentage of cancerous areas that are accurately detected.

$$Recall(sensitivity) = \frac{True\_positive}{True\_positive + False\_negatif} \tag{4.8}$$

Accuracy (ACC): the percentage of correct predictions to total predictions. The number of regions accurately detected based on their ground truth is divided by the total number of predicted test regions.

$$Accuracy = \frac{True\_positive + True\_positive}{allcases} \tag{4.9}$$

Specificity(SPEC): The ratio of correctly detected negatives, also known as the True Negative Rate, is the percentage of negatives that are correctly identified.

$$Specificity = \frac{True\_negatif}{True\_negatif + False\_positive} \tag{4.10}$$

Precision: widely used in information retrieval, expresses the proportion of positives (malignant regions) that are, in fact, positive (malignant regions).

$$Precision = \frac{True\_positif}{True\_positif + False\_positive} \tag{4.11}$$

4.6.2 Previous Related Works to Lesion Classification

Skin lesion classification is a crucial task in dermatology, particularly when distinguishing between melanoma and non-melanoma lesions. Over the years, numerous studies have been conducted to develop effective techniques for accurately identifying and differentiating these lesions. In this section, we will explore some of the previous related works that have contributed to the field of skin lesion classification.
(Stanley, Stoecker, and Moss, 2007) investigated the use of relative color histogram analysis to distinguish pigmented skin lesions. The research focused on analyzing color features in different regions of skin lesions in dermoscopy images. The goal was to improve the discrimination capability between malignant melanoma and benign lesions. The study utilized features such as percent melanoma color and color clustering ratio to quantify the presence of melanoma-colored pixels within the lesions. The researchers observed promising results, demonstrating the potential of the relative color histogram analysis technique for effective skin lesion discrimination. (Situ et al., 2008) utilized a Bag-of-Features technique for detecting malignant melanoma. They represented skin lesions using histograms of codewords or clusters derived from a training dataset. The study employed Naive Bayes and Support Vector Machines as classifiers. The authors focused on optimizing the model selection scheme to minimize false negative errors, which are more critical in melanoma screening. (Iyatomi et al., 2008) proposed an Internet-based melanoma screening system. The system aimed to differentiate between different types of skin lesions by computing and utilizing 428 features related to objective characteristics. These features were used as input to a neural network classifier, which employed the ABCD rule and differential architectures of the lesions for classification purposes. The

system achieved promising results in terms of sensitivity and specificity, although specific values were not provided. The authors highlighted the accessibility and accuracy of their system in providing melanoma diagnosis remotely via an online platform. (Di Leo et al., 2008) conducted a study centered around the detection of the atypical pigmented network, an important dermatological structure. Their proposal focused on determining the malignancy of skin lesions by utilizing decision-tree classification techniques in combination with image processing algorithms. The researchers aimed to estimate chromatic and structural characteristics associated with the atypical pigmented network, thereby enhancing the accuracy of lesion malignancy determination. (Sadeghi et al., 2011) presented a method for the detection and visualization of pigment network patterns in dermoscopic images. Their approach involved multiple steps. Initially, a Laplacian of Gaussian (LoG) filter was utilized to locate cyclic structures or meshes in the images. The filtered images were then transformed into a series of graphs using an 8-connected components analysis. A search was conducted to identify loops or cyclic graphs. The measurement of the distance between the holes facilitated the construction of a new graph, which allowed for the detection of the pigment network. (Sheha, Mabrouk, Sharawy, et al., 2012) proposed an automated technique for melanoma diagnosis. They utilized the gray-level Co-occurrence matrix (GLCM) to retrieve various variables that captured important textural information from dermoscopic images. These variables were then used as input for a multilayer perceptron classifier (MLP) to differentiate between Melanocytic Nevi and Malignant Melanoma. By leveraging the MLP, the authors were able to effectively classify the skin lesions and provide an automated approach for melanoma diagnosis. (Marques, Barata, and Mendonça, 2012) focused on the identification of melanoma lesions in dermoscopy images. While previous research has explored the use of various texture, color, and shape characteristics in melanoma detection, the specific roles and discriminative capabilities of these features remain unclear. The objective of their study was to provide a deeper understanding of the function of texture and color characteristics in melanoma identification. By analyzing a dataset of dermoscopy images, they aimed to shed light on the distinct contributions of texture and color features and their potential for improving the accuracy of melanoma detection. (Riaz et al., 2014) proposed an approach for classifying dermoscopy images by combining texture and color features. They utilized a texture feature known as a local binary pattern (LBP), which captures scale-adaptable patterns at each pixel based on the strength of LBPs. The extracted texture features were used to construct a histogram representation. In addition, they employed HSV (Hue, Saturation,

Value) histograms to extract color features from the images. These texture and color characteristics were then concatenated to create a feature vector for each image. Support vector machines (SVM) were chosen as the classification algorithm to categorize the dermoscopy images. (Celebi and Zornberg, 2014) proposed a machine-learning technique for quantifying clinically important colors in dermoscopy images. They employed the K-means clustering method to group the images, with the number of clusters (K value) determined individually for each image using five commonly used cluster validity criteria. This approach aimed to analyze and quantify the various color components present in the dermoscopy images. By automatically clustering the images based on color, the authors provided a method for objectively quantifying and assessing the clinically relevant colors in skin lesions. (Alfed et al., 2015) introduced a novel technique for melanoma diagnosis. They devised a method to separate pigment networks from dermoscopy images using a bank of direction filters. By employing these filters, they were able to isolate and extract the specific features and patterns associated with pigment networks. From the segmented image, a set of attributes was extracted, capturing key characteristics relevant to melanoma diagnosis. These attributes were then utilized as input for an Artificial Neural Network (ANN) classifier. The ANN effectively distinguished between melanocytic nevi and malignant melanoma based on the extracted attributes. (Rastgoo et al., 2015) presented an automated method for melanoma diagnosis focusing on dysplastic nevi. The proposed framework not only considered color and shape characteristics but also emphasized the analysis of texture features based on the "ABCD" clinical rule. By leveraging a combination and comparison of various texture features, the aim was to capture discriminative information for differentiating between benign and malignant lesions. The performance of the proposed method was evaluated using different classification approaches, including support vector machines (SVM), gradient boosting, and random forest algorithms. These techniques were utilized to classify the lesions based on the extracted features, enabling automated and accurate melanoma diagnosis from dysplastic nevi. (Oliveira et al., 2018) presented a computational technique for analyzing skin lesions and determining their characteristics. The proposed method involved the extraction of information related to asymmetry, border properties, color distribution, and texture analysis from skin lesion images. To improve the quality of the images and reduce noise, the authors applied an anisotropic diffusion filter. They then employed an active contour model without edges to segment the lesion boundaries. For the classification step, a support vector machine (SVM) was used to classify the lesions based on the extracted features. The goal of this ap-

proach was to enhance the accuracy of lesion analysis and classification by incorporating various image analysis techniques and machine learning algorithms.

4.6.3 Current technologies Related to Skin lesion classification.

In recent years, the field of skin cancer classification has witnessed significant advancements, particularly with the advent of deep learning techniques. Researchers have extensively investigated the potential of deep learning algorithms for accurately and automatically classifying skin lesions. In this section, we will discuss several notable deep learning-based approaches for skin lesion classification. (Kalouche, Ng, and Duchi, 2016) proposed a deep learning-based approach for the detection and classification of skin lesions, specifically focusing on melanoma. The goal was to develop an accessible and cost-effective computer-vision tool that could accurately track and classify suspicious skin lesions as either benign or malignant using images captured by a standard camera. The researchers trained a fine-tuned VGG-16 Convolutional Neural Network (CNN) on a dataset consisting of healthy skin images and skin lesion images. The proposed system showed promising results in accurately categorizing the input images into normal skin and lesion classes. (Nasr-Esfahani et al., 2016) focused on the detection of melanoma lesions using a pre-trained convolutional neural network (CNN). The experimental results revealed that their CNN-based method exhibited superior diagnostic accuracy compared to other existing techniques. This highlights the potential of their deep-learning system to aid dermatologists in the early detection of melanoma, potentially enhancing the accuracy of skin cancer diagnosis. (Mandache et al., 2018) explored the application of full field optical coherence tomography (FFOCT) for optical biopsy in skin excisions resulting from Mohs surgery. They aimed to develop a computer-aided diagnosis (CAD) tool using deep learning techniques, particularly convolutional neural networks (CNNs). The researchers trained a CNN-based feature extractor to identify characteristic properties in FFOCT data and a classifier to generalize the data distribution. The results showed promising performance, demonstrating the potential of CNNs in accurately detecting tumoral areas in FFOCT data for skin cancer diagnosis. (Sarkar, Chatterjee, and Hazra, 2019) present a study on melanoma classification in dermoscopic skin lesion images. They propose a deep depthwise separable residual convolutional algorithm and employ preprocessing techniques for noise removal and image enhancement. The model uses multi-channel image matrices and incorporates visualizations for lesion detection and classification evaluation. The results indicate high accuracy on various datasets, demonstrating the potential of the

proposed algorithm for accurate melanoma classification. (Walker et al., 2019) utilized a convolutional neural network (CNN) architecture based on the Inception V2 network to classify dermoscopic pictures as either malignant or benign. The CNN achieved a high sensitivity in accurately identifying malignant cases and demonstrated a strong discriminatory performance as measured by the AUC score. Additionally, the study explored the generation of a feature representation for potential use in sonification applications. (Mahbod et al., 2019) developed an automatic computerized method for classifying skin lesions using optimized deep features from popular CNNs such as AlexNet, VGG16, and ResNet-18. Support vector machine classifiers were trained on these features, and the outputs were combined to obtain the final classification. The proposed method showed promising results for melanoma and seborrheic keratosis classification on the ISIC 2017 dataset. (Alqudah, Alquraan, and Qasmieh, 2019)presented a novel approach for classifying skin lesions using transfer learning and optimization techniques with GoogleNet and AlexNet models. Their study focused on accurately categorizing images from the ISIC database into three main classes: benign, melanoma, and seborrheic keratosis. The proposed method demonstrated promising results in achieving high classification accuracy for both segmented and non-segmented datasets. (Khan et al., 2020b) introduced a novel approach for localizing and recognizing skin cancer. The researchers combined deep learning with an iteration-controlled Newton-Raphson (IcNR) feature selection technique. The framework involved localizing lesions using a faster region-based convolutional neural network (RCNN), extracting deep features using the pre-trained DenseNet201 model, and selecting the most discriminant features using IcNR. The selected features were then utilized for classification using multilayered feed-forward neural networks. The proposed approach demonstrated superior performance compared to existing methods, showcasing improved accuracy and efficiency. (Jayapriya and Jacob, 2020) developed a hybrid approach for lesion classification. Their method utilized a combination of a deep residual network and a hand-crafted feature to extract features from segmented lesions. These features were then fed into a support vector machine (SVM) for classification. The performance analysis of their framework yielded promising results in terms of specificity for the ISBI 2016 and ISIC 2017 datasets. (Ratul et al., 2020) proposed a computer-aided skin lesion classification system using transfer learning and dilated convolution. The study aimed to improve the accuracy of skin lesion prognosis by addressing challenges such as low contrast, visual resemblance, and blurred lesion borders. Four popular pre-trained architectures (VGG16, VGG19, MobileNet, and InceptionV3) were employed with dilated

convolution on the HAM10000 dataset. The results showed promising performance, with the highest achieved accuracy obtained using dilated InceptionV3. The study demonstrates the potential of dilated convolution in improving classification accuracy. (Akram et al., 2020) focused on the problem of discriminant feature selection for skin lesion classification. They employed three deep learning models, namely DenseNet 201, Inception-ResNet-v2, and Inception-V3, to extract and classify features from the dermoscopic images. Their proposed system's effectiveness was assessed by evaluating its performance on four distinct dermoscopic datasets, namely PH2, ISIC MSK, ISIC UDA, and ISBI-2017. The results showed promising accuracy rates in classifying skin lesions on these datasets. (Rahman et al., 2021) conducted research on the categorization of different skin lesions using ensemble learning methods. They employed a grid search strategy and trained five baseline CNN models, including ResNeXt, SeResNeXt, ResNet, Xception, and DenseNet. The performance of these models was evaluated using macro-average recall scores. Two ensemble learning algorithms, namely the average ensemble and the weighted ensemble, were used to determine the best model. The results indicated that DenseNet achieved the highest macro-average recall score, followed by ResNeXt, SeResNeXt, Xception, and ResNet.

Table 4.2 summarizes current skin lesion classification methods. These studies highlight the effectiveness of employing deep learning techniques for the classification of skin lesions. By leveraging advanced algorithms and models like Convolutional Neural Networks (CNNs), researchers have achieved promising outcomes in accurately categorizing skin lesions as either benign or malignant, with a particular emphasis on melanoma detection. The use of deep learning has improved the accuracy and efficiency of skin lesion classification, providing potential benefits in the early detection and diagnosis of skin cancer. This technological advancement holds promise for enhancing patient outcomes by enabling timely interventions and treatments.

Table 4.2: Summary of current skin lesion classification methods.

Study	Year	Dataset	Classifier and Training Algorithm		Measures	
				Accuracy	Sensitivity	Precision
(Kalouche, Ng, and Duchi, 2016)	2016	ISIC data	VGG-16 and CNN	78%	-	-
(Nasr-Esfahani et al., 2016)	2016	MED-NODE Database	CNN	81%	_	-
(Mandache et al., 2018)	2018	40 FF-OCT images	CNN	95.93 %	95.2%	-
(Mahbod et al., 2019)	2019	ISIC 2017	SVM classification with features extracted and VGG16	-	-	-
			deep models AlexNet, ResNet-18, and VGG16	-	-	-
(Sarkar, Chatterjee, and Hazra, 2019)	2019	ISIC	Deep depthwise separable residual convolutional	99.5%	-	-
		PH2		96.7%	-	-
		DermIS		94.44%	-	-
(Alqudah, Alquraan, and Qasmieh, 2019)	2019	ISIC data	GoogleNet and AlexNet	92.2%	87.7%	89.6%
(Walker et al., 2019)	2019	ISIC-2017	Inception-v2	-	86%	-
(Khan et al., 2020b)	2020	ISIC 2016	DenseNet201	94.2%	94.2%	-
(Khan et al., 2020b)	2020	ISIC 2017		93.4%	93%	-
(Jayapriya and Jacob, 2020)	2020	ISIC 2016	A deep residual network , a hand-crafted feature and SVM	88.92 %	69.33%	-
		ISIC 2017		85.3%	81.28%	-
(Akram et al., 2020)	2020	PH2	DenseNet 201, Inception-ResNet-v2 and Inception-V3	98.8%	-	-
		ISIC MSK		99.2%	-	-
(Rahman et al., 2021)	2021	HAM10000ata	ResNeXt, SeResNeXt, ResNet,	.88%	94%	87%
			Xception, and DenseNet			-

4.7 Conclusion

This chapter provides an extensive review of the current state-of-the-art methods that encompass both traditional approaches and recent advancements in the field of skin lesion analysis, specifically focusing on the segmentation and classification aspects. Regarding segmentation, the chapter covers established techniques such as thresholding, and region growing which have been widely used for segmenting skin lesions. Additionally, it discusses recent advancements in segmentation methods, particularly deep learning-based approaches like U-Net and FCN. These methods leverage the power of convolutional neural networks to accurately segment skin lesions from surrounding tissue. In terms of classification, the chapter explores traditional machine learning algorithms such as support vector machines and K-means that have been employed for skin lesion classification. It also delves into the advancements made with deep learning models, specifically convolutional neural networks (CNNs) such as Inception, ResNet, and DenseNet. These CNN-based architectures have demonstrated exceptional performance in accurately classifying different types of skin lesions. By covering both traditional and recent approaches in segmentation and classification, the chapter offers a comprehensive understanding of the progress in these areas of skin lesion analysis. It serves as a valuable resource for researchers and practitioners, providing insights into the effectiveness of different techniques and paving the way for further research and development to improve the segmentation and classification accuracy of skin lesion analysis methods.

Part II

CONTRIBUTION

In recent years, deep learning has emerged as a transformative and highly promising approach in the domain of medical image analysis, specifically in tasks related to segmentation and classification. This revolutionary paradigm, driven by Convolutional Neural Networks (CNNs) and advanced architectures like U-Net and DeepLab, has significantly impacted computer vision, revolutionizing the way medical images are processed and analyzed. Image segmentation, a critical task in medical diagnostics, involves accurately delineating specific structures or diseased tissues within the images. Deep learning models, with their ability to automatically learn intricate features and hierarchies from data, have demonstrated impressive results in achieving precise segmentation, providing valuable insights for enhanced diagnosis, treatment planning, and disease monitoring. Furthermore, deep learning's supremacy in image classification tasks has greatly improved the accuracy of medical image categorization, empowering healthcare professionals to make well-informed decisions based on the automatic recognition of various pathologies. As the field of medical image analysis continues to evolve, the versatility and potential of deep learning remain at the forefront of cutting-edge research, promising to reshape healthcare and pave the way for exciting advancements in medical research.
In this part of the thesis, our primary focus is on advancing skin lesion detection. We introduce two encoder-decoder models, W-net and ψ-net, designed to enhance skin lesion segmentation and improve border delineation in medical images. Additionally, we present the Inception Resnet model for the classification of skin lesions, with the goal of providing a comprehensive approach to dermatology image detection. This research holds the promise of substantial benefits for dermatology, potentially leading to enhanced diagnostic accuracy and improved patient care.

Chapter 5

IMPROVED CNN ARCHITECTURES FOR MEDICAL IMAGES SEGMENTATION

5.1 Introduction

In the domain of medical image analysis, image segmentation remains a challenging and vital research area. Medical images often exhibit complex structures and high variability, making accurate segmentation crucial for identifying and characterizing diseased tissues. Automated segmentation of medical images plays a crucial role in improving diagnosis, treatment planning, and disease monitoring, ultimately leading to better patient care and outcomes.

The motivation behind this research is to develop an advanced Computer-Aided Detection (CAD) system based on cutting-edge deep learning approaches for automatically detecting and segmenting medical images with high precision, sensitivity, and specificity rates. By addressing the need for precise and efficient segmentation in medical imaging, this study aims to contribute to the field of image segmentation and advance the overall progress of medical image analysis.

In this chapter, we present the contributions of the thesis in the field of image segmentation, with a specific focus on medical image analysis. We delve into the proposed methodology, highlighting the specific techniques and approaches employed. Furthermore, we showcase the experimental results, demonstrating the advancements and contributions made in the field of medical image segmentation. The outcomes of this research are ex-

pected to have significant implications for the development of automated CAD systems and enhance medical image analysis practices

5.2 Improved Unet Model

5.2.1 Segmentation pipeline

The process of creating image region segmentation maps involves a well-defined pipeline, as depicted in Figure 5.1. This pipeline consists of multiple steps aimed at achieving accurate segmentation results.

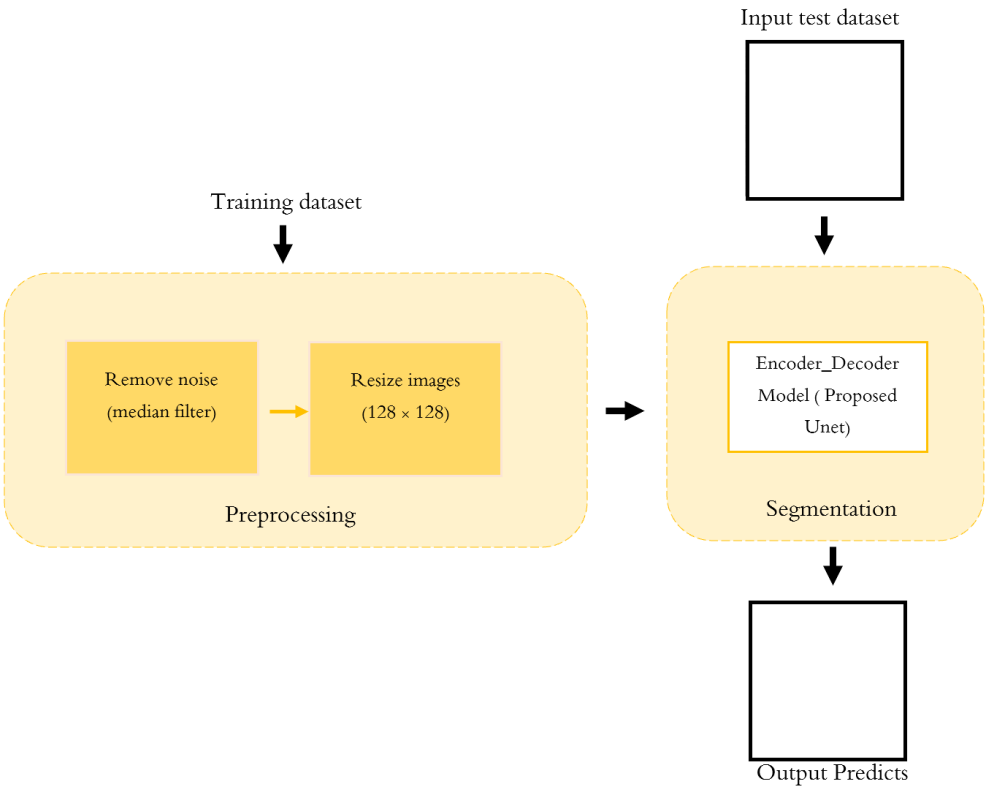


Figure 5.1: Framework of medical image segmentation system.

The initial stage of the pipeline involves preprocessing the medical images to enhance their quality and clarity, which in turn improves the performance of the segmentation process. One crucial aspect of this preprocessing step is the removal of digitization noise, which can adversely affect the images. To address this, a two-dimensional median filter is applied. The median filter functions by substituting each pixel in the image with the median value derived from the neighboring pixel. By doing so, it effectively reduces the

noise while preserving important features and edges within the image. This step helps to create cleaner and more reliable images, which are crucial for accurate segmentation. Additionally, to ensure consistency across different neural networks, the size of both the images and masks is reduced to 180 × 180 pixels. This resizing process has multiple benefits. Firstly, it facilitates a uniform input size for the deep learning models, enabling easier integration and comparison. Secondly, it can help reduce computational complexity and memory requirements, making it more feasible to train and deploy the models. an example showcasing the preprocessing step is typically provided in Figure 5.2. The preprocessed images appear cleaner, with reduced digitization noise, and have been resized to a consistent size.

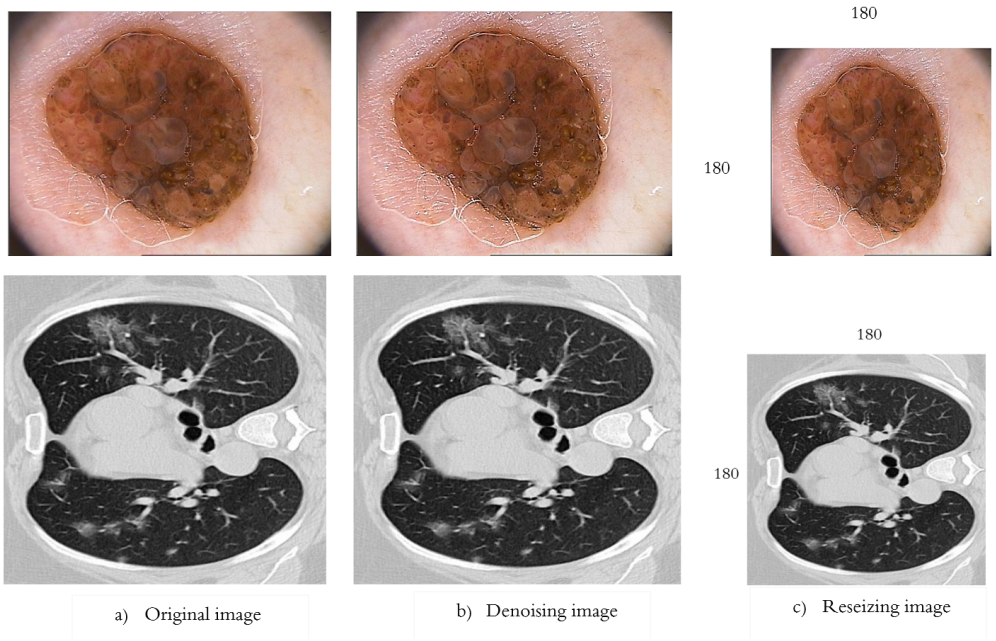


Figure 5.2: Image preprocessing stage.

5.2.2 Architecture of the improved Unet.

The enhanced U-Net architecture presented in this study is a modification of the original U-Net architecture, which has proven effective for medical image segmentation (Ronneberger, Fischer, and Brox, 2015). The U-Net architecture consists of two main pathways: an analysis path and a synthesis path. The analysis path follows a convolutional neural network (CNN) structure, while the synthesis path, also known as the expansion phase, includes an up-sampling layer and a deconvolution layer.

In the proposed improved U-Net architecture (Khouloud, Ahlem, and Amel, 2020), several changes have been introduced in both the encoder (left side) and decoder (right side) pathways. In the encoder, the training data is downsampled, and feature maps are extracted using a CNN architecture. Each down-sampling module in the encoder includes two convolutional layers (3×3, no padding) and a maximum pooling layer (2×2, stride=2) with the Exponential Linear Unit (ELU) activation function. The number of convolution kernels ranges from 12 to 192 across these modules.

In the decoder, each up-sampling module comprises a deconvolutional layer (2×2) followed by two convolutional layers (3×3, no padding) with decreasing numbers of convolution kernels (96, 48, 24, and 12). Batch normalization and dropout layers are inserted after each convolutional and deconvolutional layer. The final output is obtained through a 1×1 convolutional layer with Rectified Linear Unit (ReLU) activation. Notably, the inclusion of batch normalization and dropout layers between the convolutions and ReLU activation functions is a distinctive aspect of the proposed U-Net architecture, aiming to facilitate faster convergence during training. For a visual representation of the enhanced U-Net architecture during the training phase,refer to Figure 5.3. Additionally, a comprehensive description of the proposed UNet architecture can be found in Figure 5.4 (Khouloud, Ahlem, and Amel, 2020).

Layer (type)	Output Shape	Param #	Connected to
input_22 (InputLayer)	[(None, 128, 128, 1)	0	
conv2d_289 (Conv2D)	(None, 128, 128, 12)	120	input_22[0][0]
conv2d_290 (Conv2D)	(None, 128, 128, 12)	1308	conv2d_289[0][0]
max_pooling2d_60 (MaxPooling2D)	(None, 64, 64, 12)	0	conv2d_290[0][0]
conv2d_291 (Conv2D)	(None, 64, 64, 24)	2616	max_pooling2d_60[0][0]
conv2d_292 (Conv2D)	(None, 64, 64, 24)	5208	conv2d_291[0][0]
max_pooling2d_61 (MaxPooling2D)	(None, 32, 32, 24)	0	conv2d_292[0][0]
conv2d_293 (Conv2D)	(None, 32, 32, 48)	10416	max_pooling2d_61[0][0]
conv2d_294 (Conv2D)	(None, 32, 32, 48)	20784	conv2d_293[0][0]
max_pooling2d_62 (MaxPooling2D)	(None, 16, 16, 48)	0	conv2d_294[0][0]
conv2d_295 (Conv2D)	(None, 16, 16, 96)	41568	max_pooling2d_62[0][0]
conv2d_296 (Conv2D)	(None, 16, 16, 96)	83040	conv2d_295[0][0]
max_pooling2d_63 (MaxPooling2D)	(None, 8, 8, 96)	0	conv2d_296[0][0]
conv2d_297 (Conv2D)	(None, 8, 8, 192)	166080	max_pooling2d_63[0][0]
conv2d_298 (Conv2D)	(None, 8, 8, 192)	331968	conv2d_297[0][0]
conv2d_transpose_60 (Conv2DTran	(None, 16, 16, 96)	73824	conv2d_298[0][0]
concatenate_60 (Concatenate)	(None, 16, 16, 192)	0	conv2d_transpose_60[0][0] conv2d_296[0][0]
conv2d_299 (Conv2D)	(None, 16, 16, 96)	165984	concatenate_60[0][0]
conv2d_300 (Conv2D)	(None, 16, 16, 96)	83040	conv2d_299[0][0]
conv2d_transpose_61 (Conv2DTran	(None, 32, 32, 48)	18480	conv2d_300[0][0]
concatenate_61 (Concatenate)	(None, 32, 32, 96)	0	conv2d_transpose_61[0][0] conv2d_294[0][0]
conv2d_301 (Conv2D)	(None, 32, 32, 48)	41520	concatenate_61[0][0]
conv2d_302 (Conv2D)	(None, 32, 32, 48)	20784	conv2d_301[0][0]
conv2d_transpose_62 (Conv2DTran	(None, 64, 64, 24)	4632	conv2d_302[0][0]
concatenate_62 (Concatenate)	(None, 64, 64, 48)	0	conv2d_transpose_62[0][0] conv2d_292[0][0]
conv2d_303 (Conv2D)	(None, 64, 64, 24)	10392	concatenate_62[0][0]
conv2d_304 (Conv2D)	(None, 64, 64, 24)	5208	conv2d_303[0][0]
conv2d_transpose_63 (Conv2DTran	(None, 128, 128, 12)	1164	conv2d_304[0][0]
concatenate_63 (Concatenate)	(None, 128, 128, 24)	0	conv2d_transpose_63[0][0] conv2d_290[0][0]
conv2d_305 (Conv2D)	(None, 128, 128, 12)	2604	concatenate_63[0][0]
conv2d_306 (Conv2D)	(None, 128, 128, 12)	1308	conv2d_305[0][0]
conv2d_307 (Conv2D)	(None, 128, 128, 1)	13	conv2d_306[0][0]

Total params: 1,092,061
Trainable params: 1,092,061
Non-trainable params: 0

Figure 5.3: Summary of the trained model.

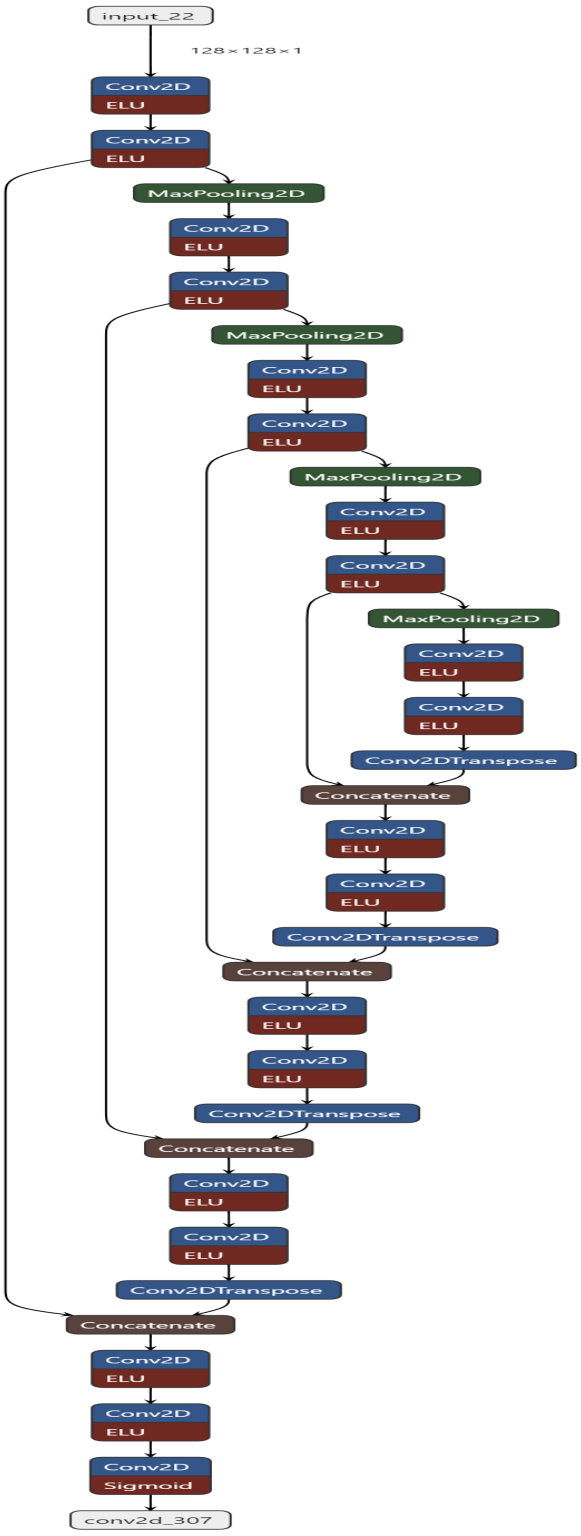


Figure 5.4: systematic view of the proposed UNet architecture.

5.2.3 Tests and Results

Hyper-parameters are critical in machine learning models, especially for deep neural networks, as they significantly impact performance. There are two primary methods for tuning hyper-parameters: manual and automatic. In this study, we focused on manual hyper-parameter tuning, involving a careful comparison of different combinations to identify the optimal configuration. The network architecture in this study utilized the BCE Dice loss function, a combination of Binary Cross Entropy (BCE) and Dice Loss. This loss function is commonly used in image segmentation tasks to optimize pixel-level accuracy and spatial coherence. During training, a batch size of 20 was used, indicating that the model processed 20 samples in each iteration. The batch size is chosen to balance computational efficiency and model convergence. The Adam optimizer was employed with an initial learning rate of 0.0001. Adam is an optimization algorithm that adapts the learning rate for each parameter based on gradient moment estimates. To address overfitting, a dropout layer with a dropout probability of 0.5 was incorporated into the model. Dropout randomly deactivates a fraction of input units during training to improve generalization. In terms of data allocation, 15% of the volumes were reserved for validation. Table 5.1 is a detailed list of the experimental training parameters.

Table 5.1: Settings for training parameters

Parameter	**value**
Optimizer	Adam optimization
Loss	bce_dice_loss
Learning rate	0.0001
Batch size	20
Number of iterations (Epoch)	120
Validation Split	15%

Table 5.2 illustrates the impact of different optimizers on the model's performance when utilizing the Binary Cross Entropy and Dice Loss function on the Skin lesion segmentation dataset (PH2). The results indicate that the Adam optimizer achieves the best overall performance across all metrics. Specifically, the Adam optimizer obtained a Dice similarity of 83%, precision of 93.4%, and mIoU of 98.2%.

Table 5.2: Overall performance: Precision,mIoU, and DIC for various Optimizer settings.

Optimizer	Loss	Measures		
		Precision	mIoU	DIC
SGD	Binary Cross Entropy and Dice Loss	0.922	0.973	0.81
RMSprop		0.924	0.975	0.82
Adam		0.934	0.982	0.83

The proposed U-net model for skin lesion segmentation on the PH2 dataset demonstrated convergence during training and testing, as illustrated in Figure 5.5. The loss curve stabilized close to zero, indicating accurate predictions of skin lesion segmentation. The accuracy curve initially started at 0.2 but eventually reached a rate above 96%, indicating successful segmentation of the data.

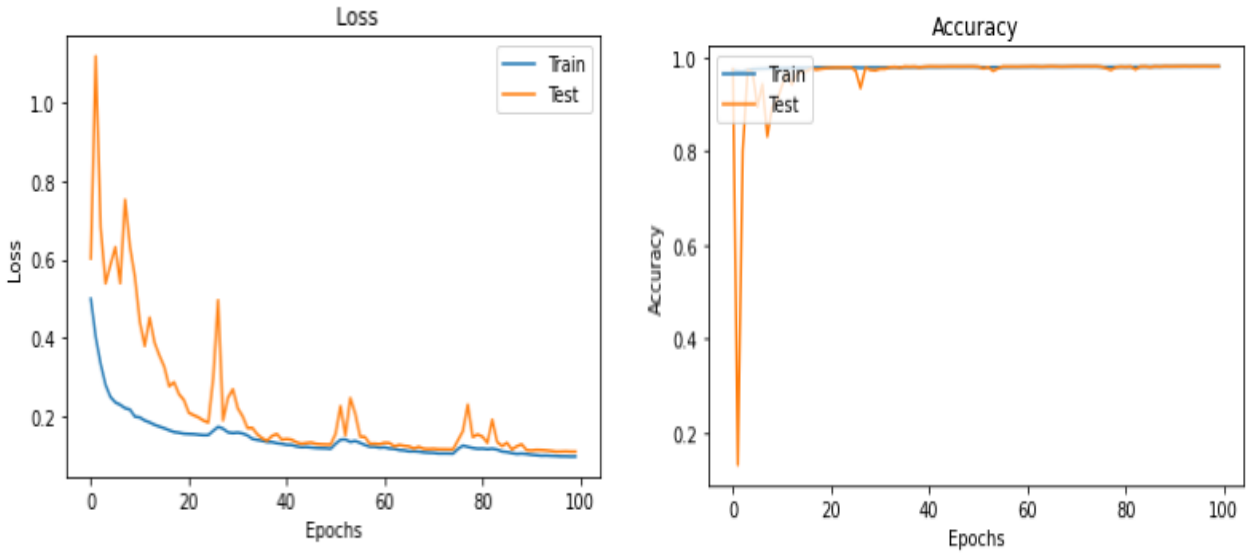


Figure 5.5: Loss and Accuracy curves of the suggested U-net model for training and testing.

In Figure 5.6, the dice coefficient loss and dice coefficient performance curves demonstrate the model's ability to accurately segment the skin lesions in the PH2 dataset. The decreasing loss values below 0.25 and the rising dice coefficient values above 0.8 indicate improved segmentation accuracy as the number of training epochs increases.

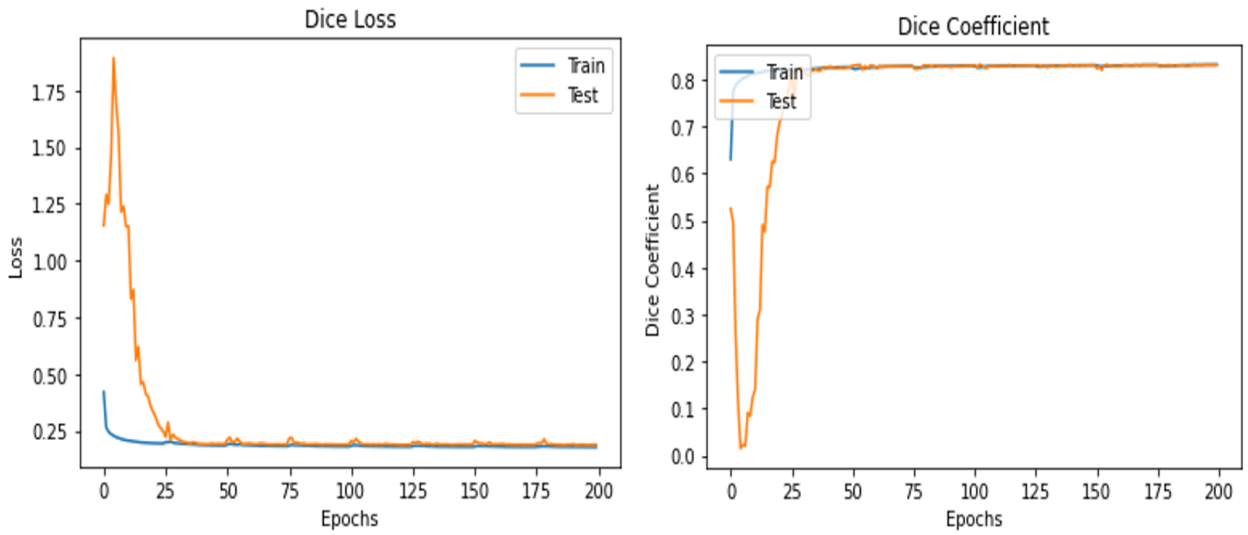


Figure 5.6: The dice loss and dice coefficient curves of the suggested U-net model for training and testing

Figure5.7 depicts the performance of the proposed U-net model in terms of recall and specificity on the PH2 dataset. The increasing recall curve, reaching over 96% for training and over 94% for testing, indicates the model's ability to identify a high percentage of true positive cases in skin lesion segmentation. The specificity curve, on the other hand, increases to over 95% for training and over 96% for testing, signifying the model's capability to accurately identify true negative cases.

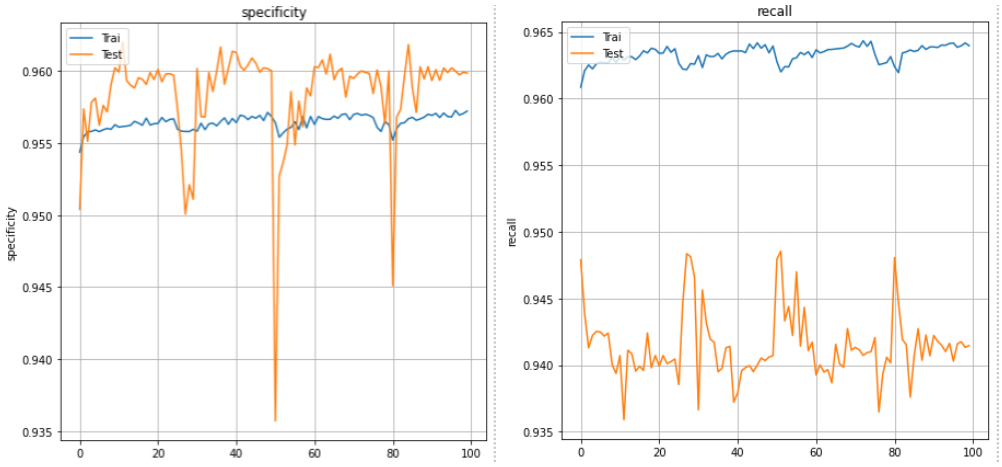


Figure 5.7: The recall and specificity curves of the suggested U-net model for training and testing.

Figure 5.8 and Figure 5.9 showcase the segmentation results obtained from our proposed model on 8 image examples, encompassing skin lesion images, COVID-19 CT images, and lung segmentation. The results of our approach demonstrate promising effectiveness as they closely resemble the ground truth. However, there are certain limitations observed during the segmentation process. In image 3 of Figure 5.8, a skin lesion image, the U-net model accurately captures the zone but struggles to precisely delineate the contour of the region. This leads to difficulties in accurately defining the exact borders of these regions. Similarly, the image 3 in Figure 5.9, a COVID-19 CT image, the U-net model successfully extracts the infected zone but encounters challenges in correctly identifying the different image labeling. These observations shed light on the limitations of our approach when it comes to accurately delineating contours and identifying specific image labeling in certain cases.

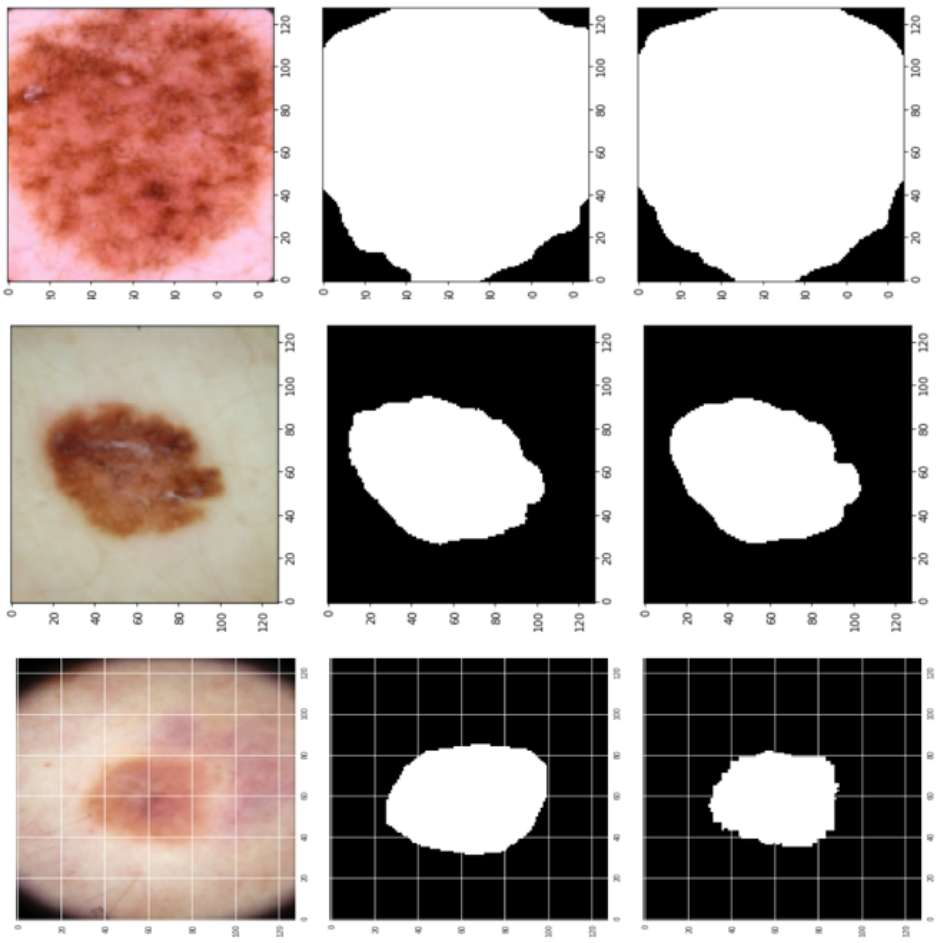


Figure 5.8: Qualitative evaluation of the proposed U-net model performance on skin lesion images.

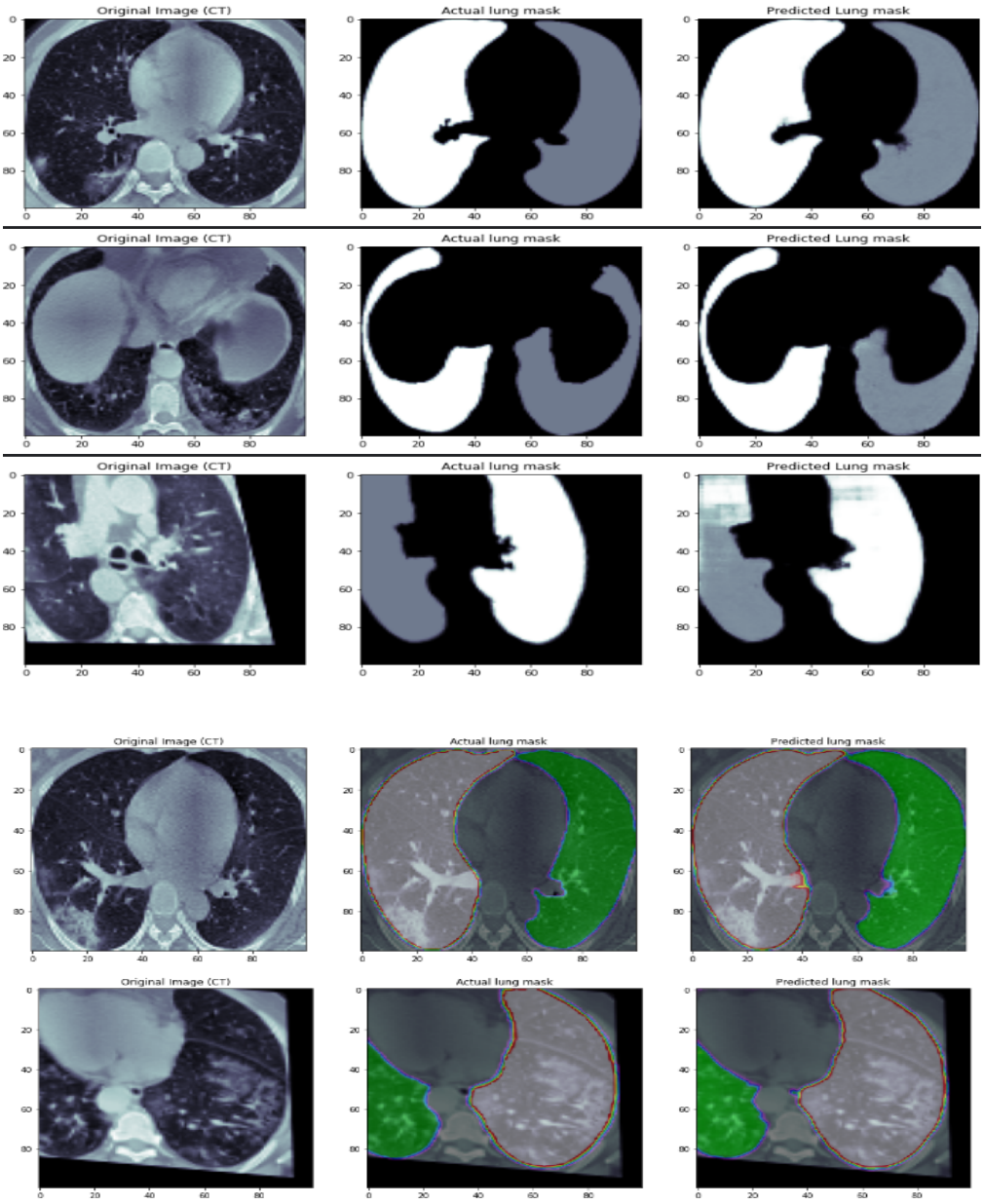


Figure 5.9: Qualitative evaluation of the proposed U-net model performance on COVID-19 CT images.

5.2.4 Comparison and Discussions

To assess the feasibility and effectiveness of our proposed enhanced approach, we conducted experiments using a selection of random samples from COVID-19 and skin lesion datasets to compare the segmentation performance of our proposed U-net model with the standard U-net model. The segmentation results obtained from our proposed U-net

model are shown in Figure 5.10.

The analysis of these results indicates that the proposed U-net model outperforms the standard U-net model in terms of segmentation accuracy. Our model demonstrates a closer alignment with the ground truth, accurately identifying and delineating lesions in both COVID-19 and skin lesion images. However, it is important to acknowledge that there may be certain instances where our model faces challenges in accurately identifying different image labels, particularly in CT COVID-19 images. Despite these challenges, the overall performance of our proposed approach highlights its effectiveness in improving segmentation performance when compared to the standard U-net model. Our U-net model consistently achieves a closer alignment with the ground truth, accurately delineating and identifying lesions in both COVID-19 and skin lesion images. These results underscore the efficacy of our approach in enhancing segmentation performance.

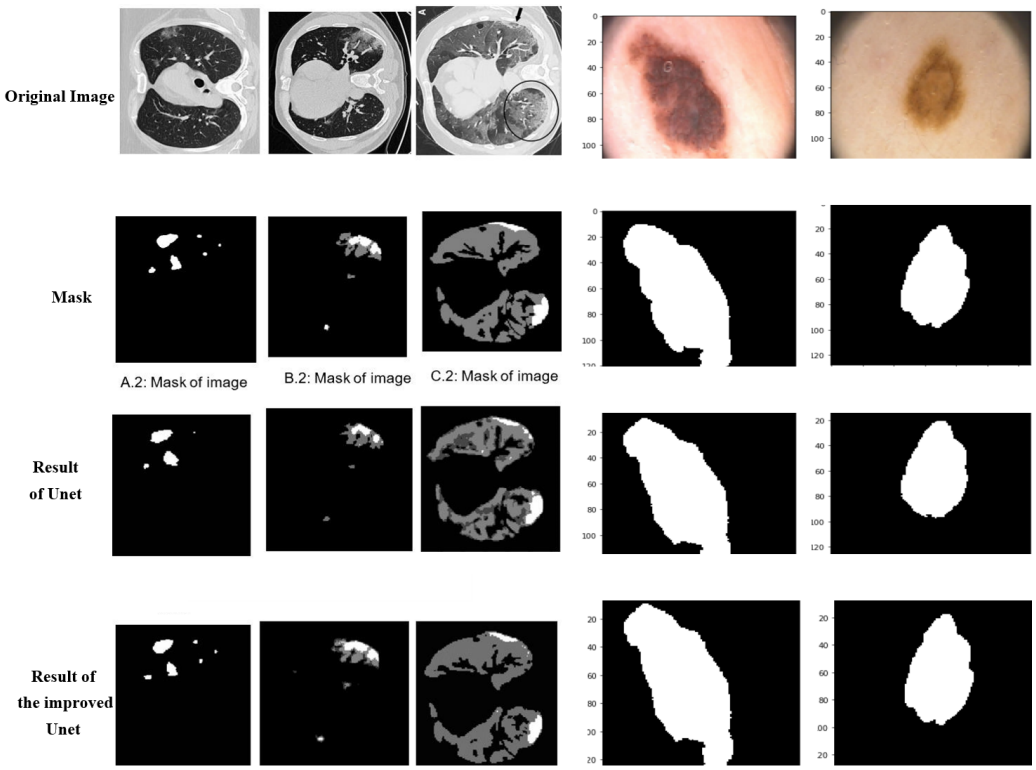


Figure 5.10: The visualization results of the Proposed model and u-net.

Table 5.3 presents a comprehensive comparison between the proposed U-net model and the U-net model for lung segmentation, lung infection segmentation, and skin lesion segmentation. The table highlights the performance metrics used to evaluate both models. Upon analyzing the metrics, it becomes evident that the proposed U-net model

consistently outperforms the U-net model in all aspects.
In terms of lung infection segmentation, the proposed U-net model achieves an accuracy of 97.47%, surpassing the U-net model's accuracy of 96.07%. The sensitivity of the proposed model is also higher at 94.07% compared to the U-net model's sensitivity of 92.47%. Additionally, the specificity of the proposed U-net model reaches 98.3%, outperforming the U-net model's specificity of 98%.
For lung segmentation, the proposed U-net model achieves an accuracy of 96.76%, which exceeds the U-net model's accuracy of 96.17%. Similarly, the sensitivity of the proposed model is higher at 93.07% compared to the U-net model's sensitivity of 91.47%. The specificity of the proposed U-net model is also improved at 97.7% compared to the U-net model's specificity of 97.1%.
Moreover, in the task of skin lesion segmentation, the proposed U-net model demonstrates an accuracy of 96.4%, slightly surpassing the U-net model's accuracy of 96.2%. The sensitivity of the proposed model is 93%, while the U-net model achieves a sensitivity of 91.3%. Additionally, the specificity of the proposed U-net model is 97.2%, higher than the U-net model's specificity of 97%.
In summary, the comparison indicates that the proposed U-net model consistently outperforms the U-net model in terms of accuracy, sensitivity, and specificity for lung segmentation, lung infection segmentation, and skin lesion segmentation. These results indicate that the proposed U-net model is more effective in accurately identifying and segmenting lung infections, lung structures, and skin lesions compared to the original U-net model.

Table 5.3: Analysis quantitative of the suggested U-net against U-net.

Metrics	Infection segmentation		Lung segmentation		Skin lesion segmentation	
	Unet	Proposed Unet	Unet	Proposed Unet	Unet	Proposed Unet
Accuracy	96.07 %	97.47%	96.17%	96.76%	96.2%	96.4%
Sensitivity	92.47 %	94.07%	91.47%	93.07 %	91.3%	93 %
Specificity	98%	98.3%	97.1%	97.7%	97%	97.2 %

Table 5.4 presents a comparison between the suggested U-net technique and existing methods in the literature for lung infection segmentation. The table reveals that the proposed technique achieves a higher sensitivity of 94% compared to the other methods. This indicates that the suggested U-net model is more successful in accurately identifying lung infections. These findings suggest that the proposed technique holds promise for effective and reliable lung infection segmentation.

Table 5.4: Comparison of the proposed U-net with current literature methods for lung infection segmentation.

Authors	Dice score	Sensitivity
(Jin et al., 2020)	-	97.4%
(Zhou, Canu, and Ruan, 2021)	83.1%	86.7%
(Wang et al., 2020)	80.7%	-
Proposed unet	82%	94.07%

Discussion

The suggested model, an enhanced version of the U-Net model, has demonstrated promising results in infection lung segmentation, lung segmentation, and skin lesion segmentation tasks. Through the incorporation of additional layers and adjustments to parameter settings, the suggested model has significantly improved its overall efficiency and performance compared to the traditional U-Net model. In the infection lung segmentation task, the suggested model has shown accurate recognition of infections and closely aligned segmentation results with the real ground truth, as depicted in Figures 5.8,5.9 and 5.10. This indicates the model's effectiveness in accurately identifying and segmenting infected areas. Furthermore, the suggested model outperforms the baseline U-Net approach in terms of dice similarity and recall metrics, as evidenced by the comparison in Table 5.4. Moreover, the suggested model has also demonstrated superior performance in lung segmentation and skin lesion segmentation tasks. It has achieved higher accuracy, recall, precision, and specificity compared to the standard U-Net model, as presented in Table 5.3. The model's ability to accurately identify lung structures and segment skin lesions showcases its robustness and efficacy in these specific tasks.

Based on these findings, it can be concluded that the suggested strategy has demonstrated significant advancements compared to the basic Unet. However, it's essential to recognize that there is still untapped potential for improvement, especially as segmentation results are far from the ground truth. This underscores the ongoing need for refinements and improvements to achieve superior results.

5.3 Improved ResUnet Model

5.3.1 Segmentation Pipeline

In this section, we present an improved method for the segmentation of medical images. Figure 5.11 illustrates the steps involved in this process. We will describe each step in detail in the following subsections.

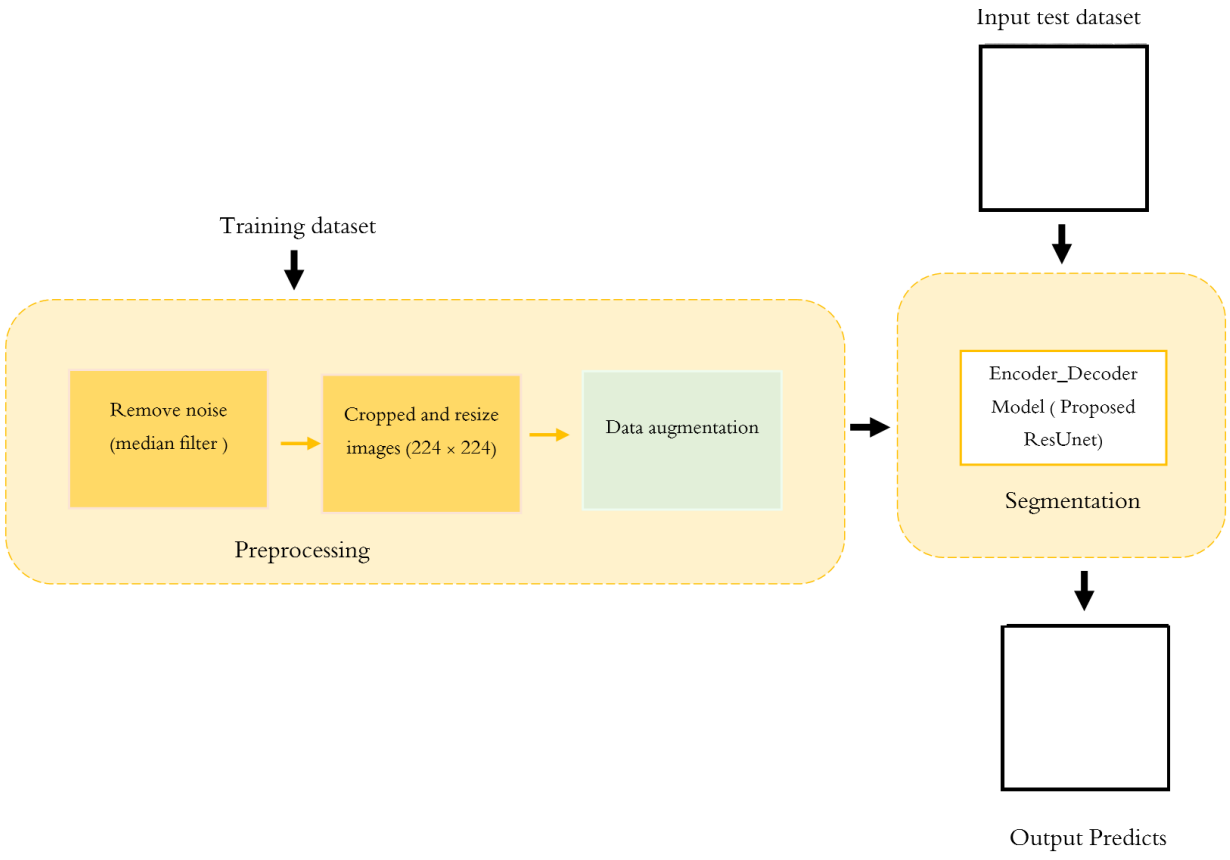


Figure 5.11: Framework of the proposed medical image segmentation System.

In the proposed method for medical image segmentation, a series of preprocessing steps are applied. Firstly, a 2D median filter is employed to eliminate digitizing noise present in the input images. To facilitate the segmentation process and optimize memory usage, the images and masks are cropped and resized to a standardized size of 224 x 224 pixels. To overcome the challenge of limited training data, image augmentation techniques are implemented. Specifically, the medical images are augmented by rotating them at angles of 90°, 180°, and 220°. These preprocessing steps enhance the quality of the images, reduce noise, focus the segmentation process, and diversify the training dataset, ultimately improving the accuracy and robustness of the segmentation algorithm. Two such examples of preprocessing are depicted in Figure 5.12.

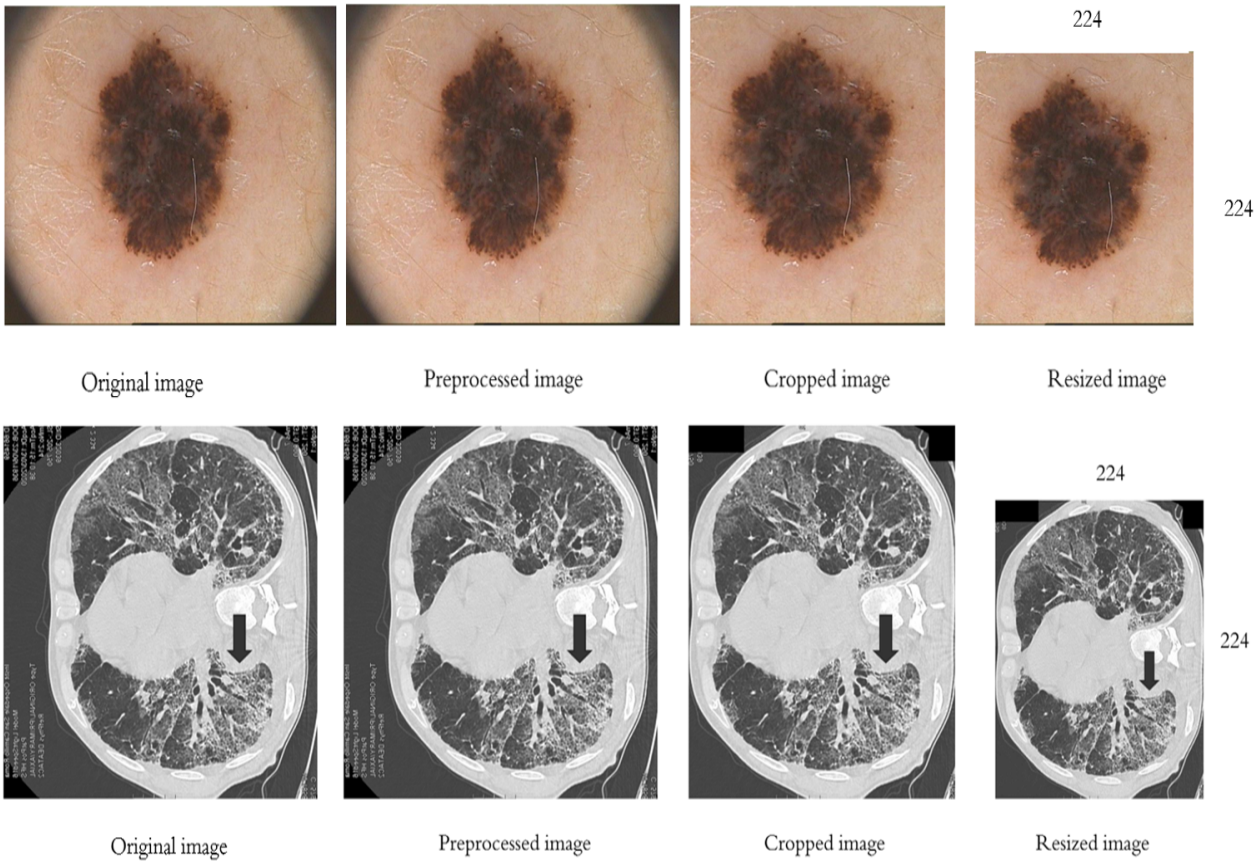


Figure 5.12: Different steps in Preprocessing of medical images.

5.3.2 Architecture of the improved Res-Unet.

The Res-Unet architecture, proposed in this study (Khouloud et al., 2022a), is an improved version of the Deep Residual U-Net (Diakogiannis et al., 2020). It combines the strengths of two powerful models: U-Net (Ronneberger, Fischer, and Brox, 2015) and deep residual learning (He et al., 2016). This combination offers several advantages in medical image segmentation.

One of the key advantages is the incorporation of residual units and skipped connections within the architecture. These components play a crucial role in network formation and facilitate the propagation of information without degradation. The residual units enable the smooth flow of information layer by layer, enabling the construction of a more complex neural network that can effectively address the degradation problem commonly encountered in deep networks. This not only improves the architecture's overall performance but also helps reduce computational costs.

The proposed Res-Unet architecture consists of four encoder blocks, a bridge, and four decoder blocks, as depicted in Figure 5.13. The residual unit, a fundamental component, comprises convolutional layers, batch normalization, ReLU activation, and an identity mapping that connects the input and output of the unit. In the encoder block, a strided convolution layer is used to downsample the feature maps by reducing the spatial dimensions. Similarly, residual units are employed in the decoding path to capture fine details and reconstruct the segmented image. The feature maps are projected onto the segmented image using a (3×3) convolutional layer followed by a sigmoid activation function. A summary of the enhanced ResUnet architecture during the training phase is shown in Figure 5.14. For a comprehensive understanding of the Res-Unet architecture, refer to Figure 5.15, which provides a detailed overview of the architecture's components and their connections. The proposed Res-Unet architecture offers an advanced approach to medical image segmentation by leveraging residual units and skipped connections to enhance network performance and information propagation. It delivers accurate segmentation results while considering computational efficiency.

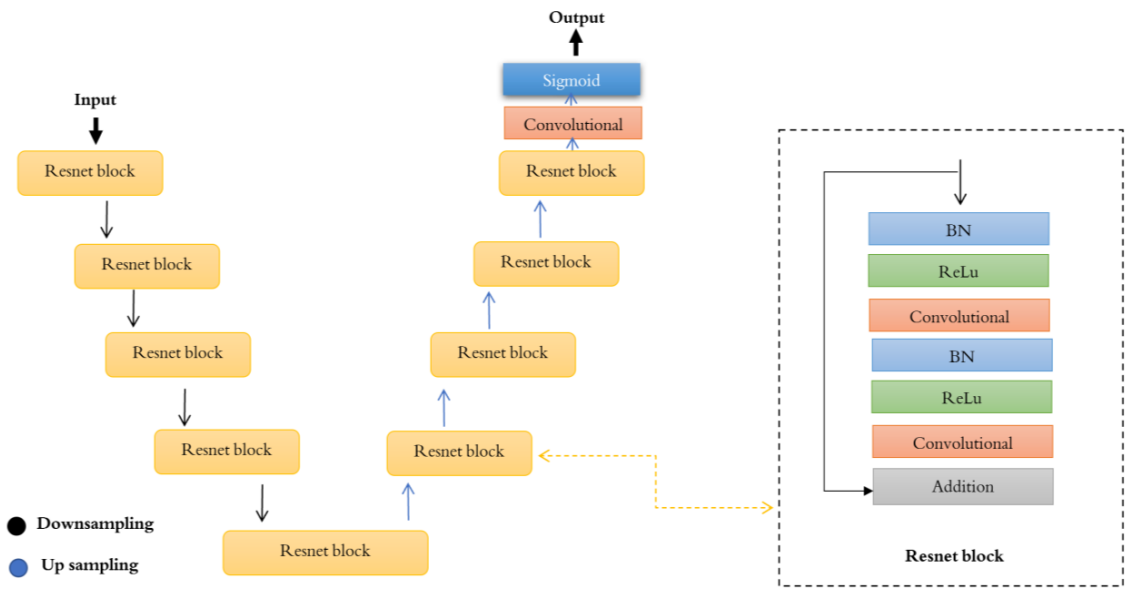
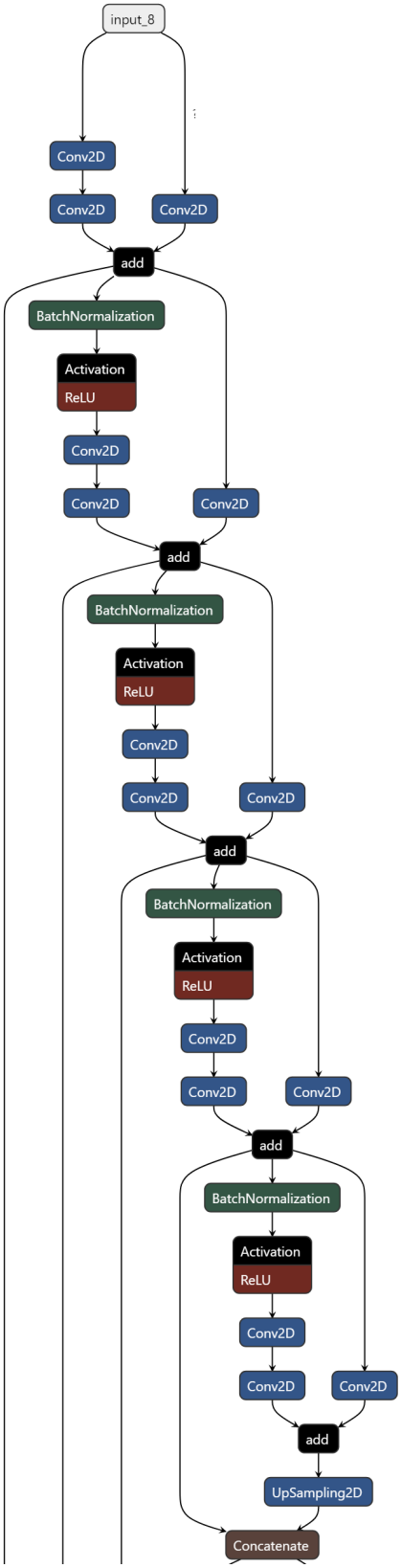


Figure 5.13: The improved Res-Unet architecture

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)	0	
conv2d (Conv2D)	(None, 224, 224, 60)	1680	input_1[0][0]
conv2d_1 (Conv2D)	(None, 224, 224, 60)	32460	conv2d[0][0]
conv2d_2 (Conv2D)	(None, 224, 224, 60)	240	input_1[0][0]
tf.__operators__.add (TFOpLambd	(None, 224, 224, 60)	0	conv2d_1[0][0] conv2d_2[0][0]
batch_normalization (BatchNorma	(None, 224, 224, 60)	240	tf.__operators__.add[0][0]
activation (Activation)	(None, 224, 224, 60)	0	batch_normalization[0][0]
conv2d_3 (Conv2D)	(None, 112, 112, 120	64920	activation[0][0]
conv2d_4 (Conv2D)	(None, 112, 112, 120	129720	conv2d_3[0][0]
conv2d_5 (Conv2D)	(None, 112, 112, 120	7320	tf.__operators__.add[0][0]
tf.__operators__.add_1 (TFOpLam	(None, 112, 112, 120	0	conv2d_4[0][0] conv2d_5[0][0]
batch_normalization_1 (BatchNor	(None, 112, 112, 120	480	tf.__operators__.add_1[0][0]
activation_1 (Activation)	(None, 112, 112, 120	0	batch_normalization_1[0][0]
conv2d_6 (Conv2D)	(None, 56, 56, 240)	259440	activation_1[0][0]
conv2d_7 (Conv2D)	(None, 56, 56, 240)	518640	conv2d_6[0][0]
conv2d_8 (Conv2D)	(None, 56, 56, 240)	29040	tf.__operators__.add_1[0][0]
tf.__operators__.add_2 (TFOpLam	(None, 56, 56, 240)	0	conv2d_7[0][0] conv2d_8[0][0]
batch_normalization_2 (BatchNor	(None, 56, 56, 240)	960	tf.__operators__.add_2[0][0]
activation_2 (Activation)	(None, 56, 56, 240)	0	batch_normalization_2[0][0]
conv2d_9 (Conv2D)	(None, 28, 28, 480)	1037280	activation_2[0][0]
conv2d_10 (Conv2D)	(None, 28, 28, 480)	2074080	conv2d_9[0][0]
conv2d_11 (Conv2D)	(None, 28, 28, 480)	115680	tf.__operators__.add_2[0][0]
tf.__operators__.add_3 (TFOpLam	(None, 28, 28, 480)	0	conv2d_10[0][0] conv2d_11[0][0]
batch_normalization_3 (BatchNor	(None, 28, 28, 480)	1920	tf.__operators__.add_3[0][0]
activation_3 (Activation)	(None, 28, 28, 480)	0	batch_normalization_3[0][0]
conv2d_12 (Conv2D)	(None, 14, 14, 480)	2074080	activation_3[0][0]
conv2d_13 (Conv2D)	(None, 14, 14, 480)	2074080	conv2d_12[0][0]
conv2d_14 (Conv2D)	(None, 14, 14, 480)	230880	tf.__operators__.add_3[0][0]
tf.__operators__.add_4 (TFOpLam	(None, 14, 14, 480)	0	conv2d_13[0][0] conv2d_14[0][0]
conv2d_16 (Conv2D)	(None, 28, 28, 480)	2074080	conv2d_15[0][0]
conv2d_17 (Conv2D)	(None, 28, 28, 480)	461280	concatenate[0][0]
tf.__operators__.add_5 (TFOpLam	(None, 28, 28, 480)	0	conv2d_16[0][0] conv2d_17[0][0]
up_sampling2d_1 (UpSampling2D)	(None, 56, 56, 480)	0	tf.__operators__.add_5[0][0]
concatenate_1 (Concatenate)	(None, 56, 56, 720)	0	up_sampling2d_1[0][0] tf.__operators__.add_2[0][0]
batch_normalization_5 (BatchNor	(None, 56, 56, 720)	2880	concatenate_1[0][0]
activation_5 (Activation)	(None, 56, 56, 720)	0	batch_normalization_5[0][0]
conv2d_18 (Conv2D)	(None, 56, 56, 240)	1555440	activation_5[0][0]
conv2d_19 (Conv2D)	(None, 56, 56, 240)	518640	conv2d_18[0][0]
conv2d_20 (Conv2D)	(None, 56, 56, 240)	173040	concatenate_1[0][0]
tf.__operators__.add_6 (TFOpLam	(None, 56, 56, 240)	0	conv2d_19[0][0] conv2d_20[0][0]
up_sampling2d_2 (UpSampling2D)	(None, 112, 112, 240	0	tf.__operators__.add_6[0][0]
concatenate_2 (Concatenate)	(None, 112, 112, 360	0	up_sampling2d_2[0][0] tf.__operators__.add_1[0][0]
batch_normalization_6 (BatchNor	(None, 112, 112, 360	1440	concatenate_2[0][0]
activation_6 (Activation)	(None, 112, 112, 360	0	batch_normalization_6[0][0]
conv2d_21 (Conv2D)	(None, 112, 112, 120	388920	activation_6[0][0]
conv2d_22 (Conv2D)	(None, 112, 112, 120	129720	conv2d_21[0][0]
conv2d_23 (Conv2D)	(None, 112, 112, 120	43320	concatenate_2[0][0]
tf.__operators__.add_7 (TFOpLam	(None, 112, 112, 120	0	conv2d_22[0][0] conv2d_23[0][0]
up_sampling2d_3 (UpSampling2D)	(None, 224, 224, 120	0	tf.__operators__.add_7[0][0]
concatenate_3 (Concatenate)	(None, 224, 224, 180	0	up_sampling2d_3[0][0] tf.__operators__.add[0][0]
batch_normalization_7 (BatchNor	(None, 224, 224, 180	720	concatenate_3[0][0]
activation_7 (Activation)	(None, 224, 224, 180	0	batch_normalization_7[0][0]
conv2d_24 (Conv2D)	(None, 224, 224, 60)	97260	activation_7[0][0]
conv2d_25 (Conv2D)	(None, 224, 224, 60)	32460	conv2d_24[0][0]
conv2d_26 (Conv2D)	(None, 224, 224, 60)	10860	concatenate_3[0][0]
tf.__operators__.add_8 (TFOpLam	(None, 224, 224, 60)	0	conv2d_25[0][0] conv2d_26[0][0]
conv2d_27 (Conv2D)	(None, 224, 224, 1)	61	tf.__operators__.add_8[0][0]

Total params: 18,294,781
Trainable params: 18,288,541
Non-trainable params: 6,240

Figure 5.14: Summary of Resunet model.


input_8
Conv2D
Conv2D
Conv2D
add
BatchNormalization
Activation
ReLU
Conv2D
Conv2D
Conv2D
add
BatchNormalization
Activation
ReLU
Conv2D
Conv2D
Conv2D
add
BatchNormalization
Activation
ReLU
Conv2D
Conv2D
Conv2D
add
BatchNormalization
Activation
ReLU
Conv2D
Conv2D
Conv2D
add
UpSampling2D
Concatenate

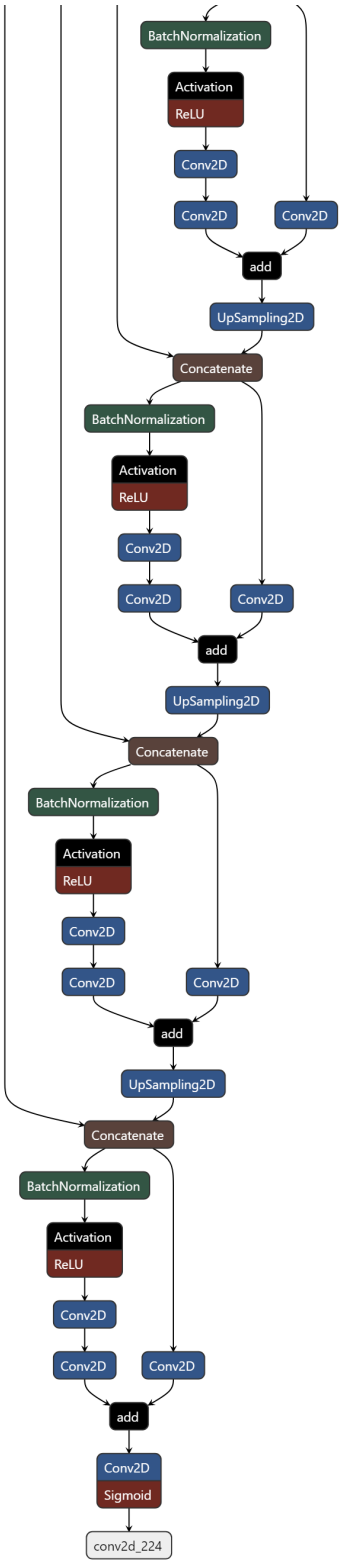


Figure 5.15: systematic view of the proposed ResUnet architecture.

5.3.3 Tests and Results

In this section, we present the findings of our study and provide an analysis of both quantitative and qualitative results obtained from the improved ResUnet model. To determine the optimal configuration, we conducted an extensive exploration of various hyperparameter combinations. The improved model incorporates the binary cross-entropy loss function and utilizes the Dice coefficient as the evaluation metric. For optimization, we employed the Adam optimizer with an initial learning rate set to 0.0001. A batch size of 25 was selected for training, and 20% of the volumes were set aside for validation purposes. The specific details of the experimental training settings can be found in Table 5.5.

Table 5.5: Settings for training parameters of ResUnet

Type	**Setting**
Optimizer	Adam
Loss	binary crossentropy
Learning rate	0.0001
Batch size	25
Number of iterations (Epoch)	100

Table 5.6 presents the results of evaluating different optimizers using the Binary Cross Entropy loss function on the skin lesion dataset. The findings indicate that the ADAM optimizer demonstrates the best overall performance across all evaluation metrics. Specifically, the ADAM optimizer achieved a Dice similarity coefficient of 83.2%, precision of 93.7%, and mean Intersection over Union (mIoU) of 98.6%. These results indicate that the ADAM optimizer effectively enhances the model's ability to accurately segment skin lesions in medical images.

Table 5.6: Overall performance: Precision,mIoU, and DIC for various Optimizer settings.

Optimizer	Loss	Measures		
		Precision	mIoU	DIC
SGD	Binary Cross Entropy	0.928	0.977	0.818
RMSprop		0.931	0.98	0.822
Adam		0.937	0.986	0.832

Figure 5.16 depicts the convergence impact of the improved ResUnet model on the PH2 dataset for skin lesion segmentation. The figure displays the training loss and accuracy curves. The training loss curve indicates that the model has achieved near-zero loss, indicating successful learning from the training data. Regarding accuracy, the model initially exhibits a low accuracy of 0.2 during testing. However, as the training progresses, the accuracy improves significantly, reaching a segmentation rate exceeding 98%. This demonstrates the enhanced convergence of the improved ResUnet model. The figure clearly demonstrates the improved model's ability to converge effectively and accurately segment skin lesions in the PH2 dataset.

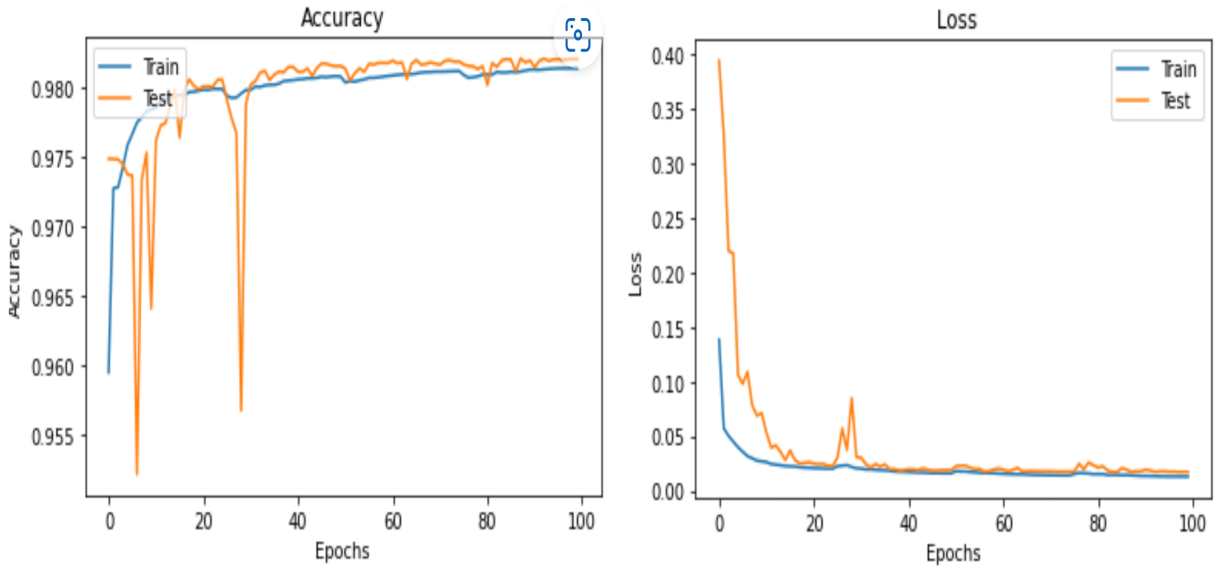


Figure 5.16: Loss and Accuracy curves of the suggested ResUnet model for training and testing.

Figure 5.17 illustrates the performance of the proposed ResUnet model on the PH2 skin lesion segmentation task. The dice coefficient loss values steadily decrease below 0.2, indicating effective segmentation. The dice coefficient values consistently rise above 0.82, indicating accurate skin lesion identification and delineation. These results demonstrate the model's effectiveness in accurately segmenting skin lesions in the PH2 dataset.

Figure 5.18 illustrates the performance curves of the ResUnet model in terms of recall and specificity for skin lesion segmentation on the PH2 dataset. As the number of iterations increases, the recall curve shows significant improvement, exceeding 96% for training and over 95% for testing.

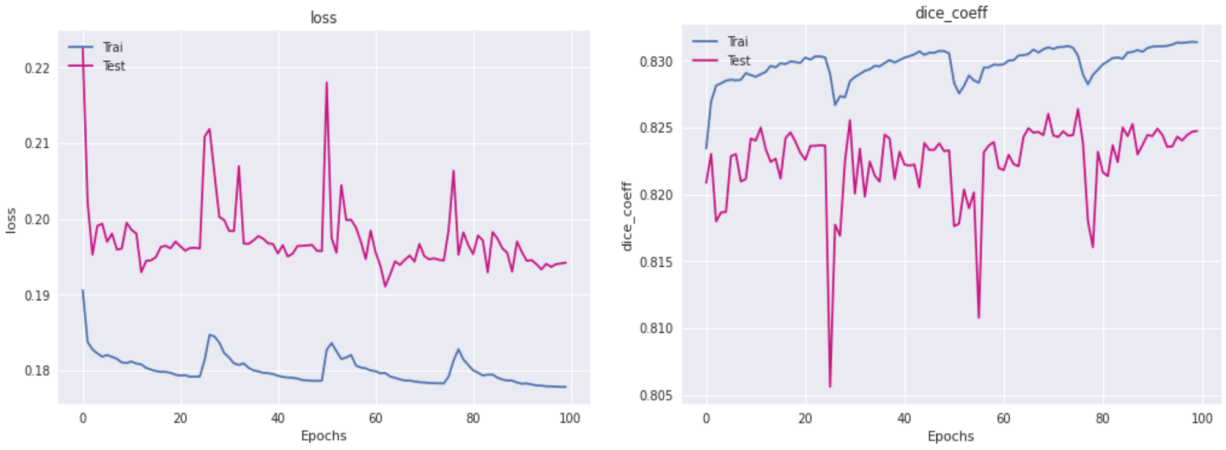


Figure 5.17: The dice loss and dice coefficient curves of the suggested ResUnet model for training and testing

Similarly, the specificity curve demonstrates an upward trend, surpassing 95% for both training and testing. These findings highlight the ResUnet model's effectiveness in correctly identifying positive cases (recall) and accurately excluding negative cases (specificity) in skin lesion segmentation on the PH2 dataset.

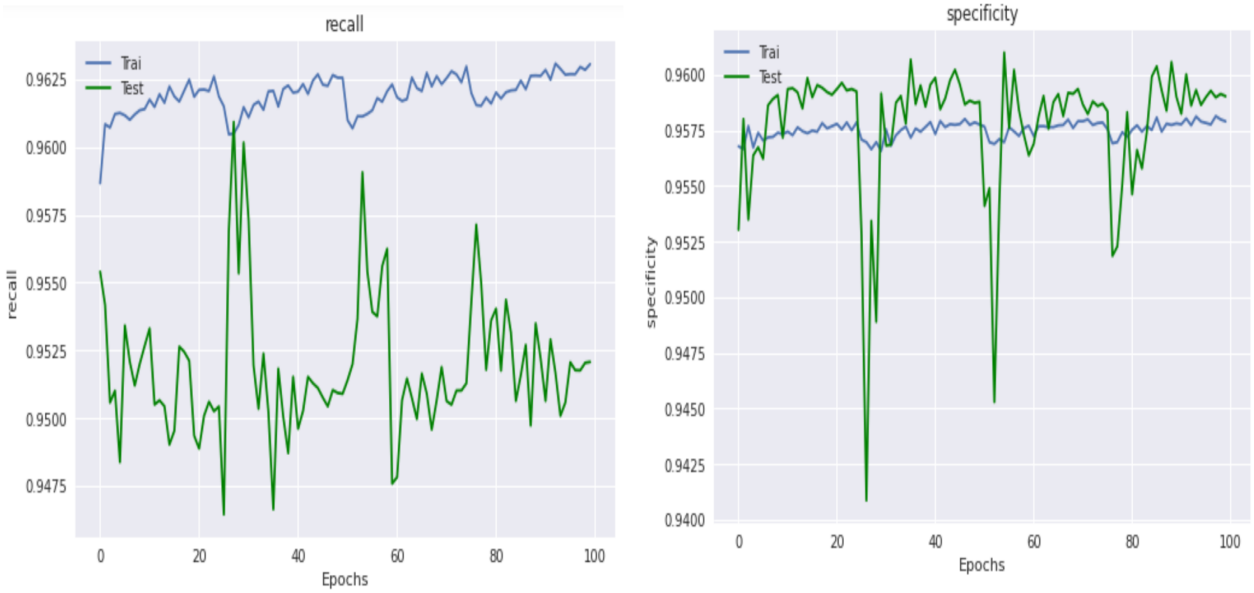


Figure 5.18: The recall and specificity curves of the suggested ResUnet model for training and testing.

Our suggested model's segmentation outcomes on five different image samples (skin lesion, COVID-19 CT, and lung segmentation) are presented in Figure 5.19. The improved results demonstrate higher accuracy and proximity to expert annotations, validating the effectiveness of our approach. However, the improved Res-Unet struggles to precisely extract the shape of the targeted zone. Further refinements or alternative techniques are required to address this limitation

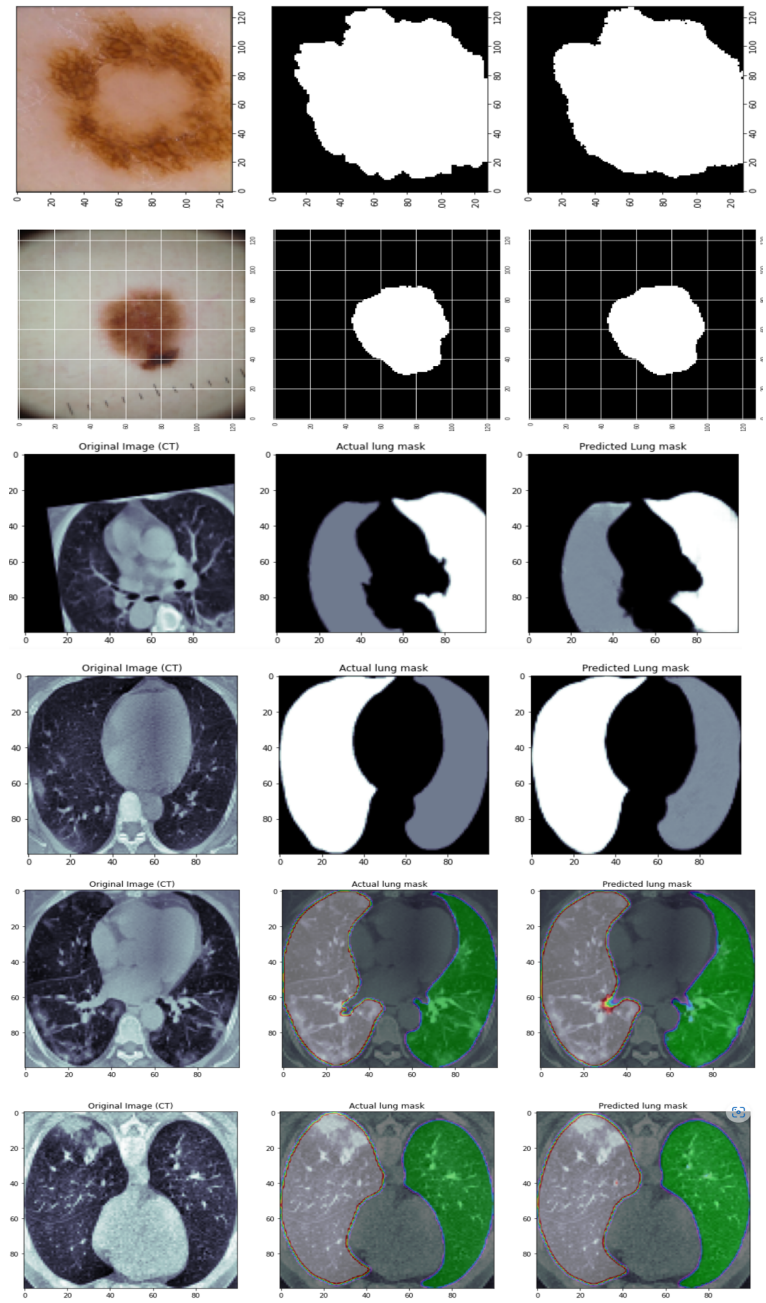


Figure 5.19: Qualitative evaluation of the proposed ResUnet model performance.

5.3.4 Comparison and Discussion

The effectiveness of the proposed ResUnet model in dataset segmentation was assessed using random samples of COVID-19 and skin lesion images. The segmentation results of the proposed model were compared to those of ResUnet (Figure 5.20). Although the enhanced model faces challenges in accurately detecting small lesions in COVID-19 images, both ResUnet and the proposed model demonstrate a similar ability to identify most skin lesion locations, to the expert's annotations. Notably, the suggested ResUnet model exhibits superior performance compared to ResUnet in segmenting COVID-19 and skin lesion images, showing improved accuracy in detecting lesion locations.

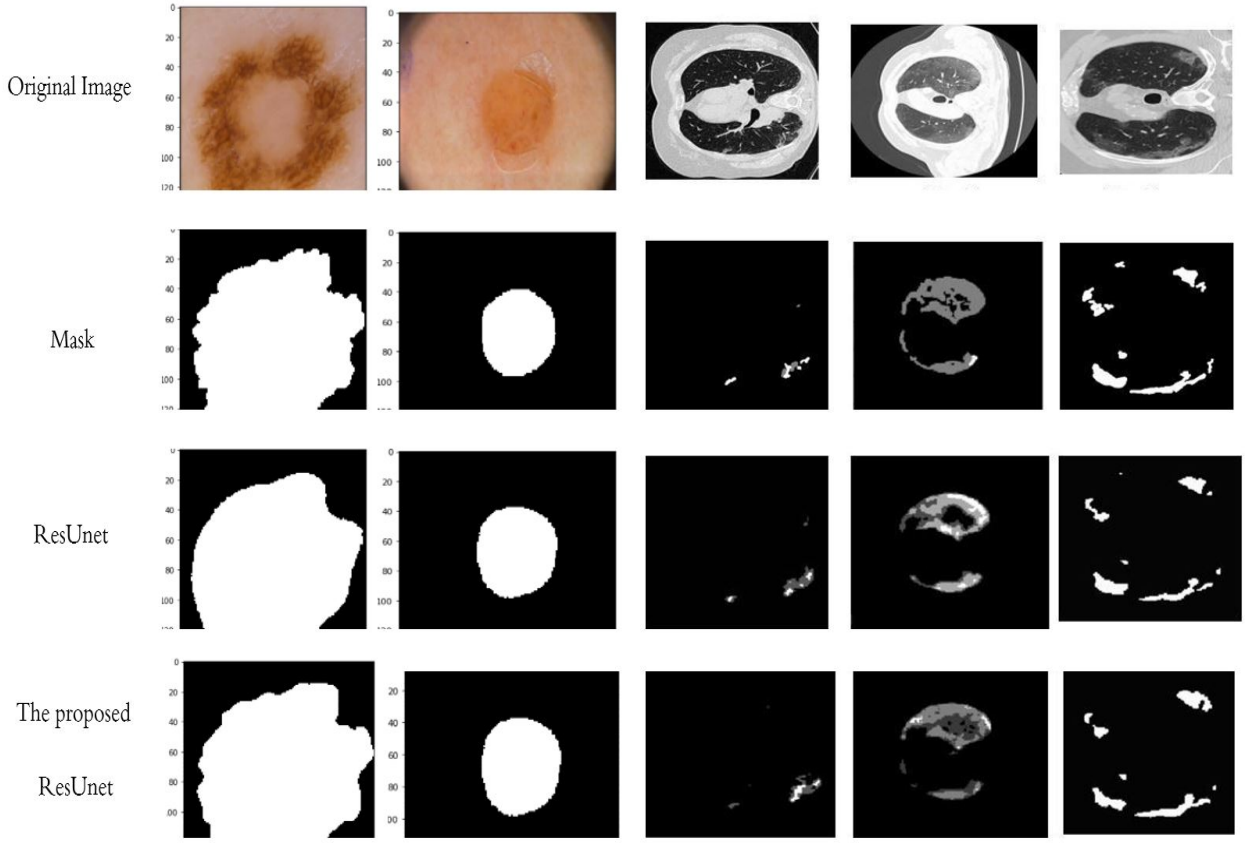


Figure 5.20: Random Samples Segmentation Results on COVID-19 and Skin Lesion Dataset.

Table 5.7 provides a comprehensive comparison between the proposed ResUnet model and ResUnet in terms of performance measures for lung segmentation, lung infection segmentation, and skin lesion segmentation. The proposed ResUnet model consistently outperforms ResUnet across all metrics.

Specifically, the proposed model achieves higher accuracy, sensitivity, and specificity in each segmentation task. These results indicate that the proposed ResUnet model is more effective in accurately segmenting lung infections, lung structures, and skin lesions compared to the original ResUnet model.

Table 5.7: Analysis quantitative of the suggested Resunet against Resunet.

Metrics	Infection segmentation		Lung segmentation		Skin lesion segmentation	
	ResUnet	Proposed ResUnet	ResUnet	Proposed ResUnet	ResUnet	Proposed ResUnet
Accuracy	97 %	97.56%	96.7%	97.2%	96.5%	96.8%
Sensitivity	94.47 %	95.01%	92.7%	93.65 %	92%	93.7 %
Specificity	98.4%	98.6%	97.9%	98.12%	97.1%	97.6 %

Table 5.8 presents a comparison of the proposed ResUnet strategy with existing methods in the COVID-19 literature. The performance of the proposed model surpasses other approaches in terms of sensitivity and dice score. It achieves a sensitivity of 95.01% and a dice score of 83.2%. In comparison, (Jin et al., 2020) achieved a sensitivity of 97.4% without reporting the dice score. (Zhou, Canu, and Ruan, 2021) achieved a dice score of 83.1% and a sensitivity of 86.7%. (Wang et al., 2020) reported a dice score of 80.7% but did not provide sensitivity information. These findings demonstrate that the proposed ResUnet strategy outperforms the average results found in the literature and and even surpasses the improved Unet, underscoring its advancements in lung infection segmentation.

Table 5.8: Comparison of the proposed ResUnet with current literature methods for lung infection segmentation.

Authors	Dice score	Sensitivity
(Jin et al., 2020)	-	97.4%
(Zhou, Canu, and Ruan, 2021)	83.1%	86.7%
(Wang et al., 2020)	80.7%	-
The improved Unet	82%	94.07%
Proposed Resunet	83.2%	95.01%

Discussion

Our study introduces a proposed ResUnet model that addresses the segmentation challenges of skin lesions and COVID-19 regions. By incorporating crucial parameters and refining the model architecture, we have achieved significant improvements in segmentation accuracy and performance measures. Table 5.7 provides a comprehensive comparison between the proposed ResUnet model and the traditional ResUnet approach. The results clearly demonstrate the superiority of our model across various metrics, including accuracy, recall, and specificity. The proposed ResUnet consistently outperforms the traditional ResUnet, yielding gains of 97.56%, 95.01%, and 98.6% in these respective measures for covid-19 segmentation task. To showcase the effectiveness of our approach, we present visual examples in Figures 5.19, and 5.20. These examples highlight the model's ability to accurately recognize infections and produce segmentation outcomes that closely align with the ground truth. The proposed ResUnet model demonstrates its capability in accurately segmenting skin lesions and COVID-19 regions, enabling efficient identification and diagnosis. Furthermore, we conducted a comparative analysis of our model against other state-of-the-art techniques, as outlined in Table 5.8, with a specific focus on dice similarity and recall. The results of this comparison clearly indicate that our approach outperforms not only the improved Unet method but also other techniques. These findings lead us to the conclusion that our proposed strategy has indeed made significant advancements when compared to the enhanced Unet. However, it's important to note that there is still room for improvement, particularly in achieving a more consistent alignment between segmentation results and ground truth data. This recognition underscores the ongoing need for continuous enhancements aimed at achieving superior results.

5.4 Conclusion

this chapter focused on the application of deep learning techniques, specifically Unet and Res-Unet, for the segmentation of medical image regions. Through various modifications and experiments, we aimed to improve the performance and accuracy of these models in the context of medical image analysis. The proposed models, including the enhanced Res-Unet and improved Unet, showed promising results in segmenting medical image regions, particularly in the case of skin lesions and COVID-19 CT scans.

The enhanced Res-Unet achieved a dice score of 83.2% and a sensitivity of 95.01%, while the improved Unet achieved a dice score of 82% and a sensitivity of 94%. These findings demonstrate the effectiveness of the proposed models in accurately identifying and delineating medical image regions. Comparative analysis with existing methods has demonstrated promising results in accurately identifying and delineating medical image regions, particularly in the case of skin lesions and COVID-19 CT scans. However, it's essential to note that the journey towards perfecting these models is far from over. The obtained results, while good, still exhibit some deviation from the ground truth. This serves as a clear indication that there remains ample room for refinement and enhancement. As we transition into the next chapter, we will delve deeper into the ongoing work and explore additional avenues for improving the accuracy, robustness, and practicality of these models in the critical domain of medical image region segmentation.

Chapter 6

PROPOSED DEEP LEARNING MODEL FOR MEDICAL IMAGE SEGMENTATION

6.1 Introduction

In this chapter, we delve into the realm of image segmentation, a vital aspect of computer vision that has witnessed remarkable advancements through deep learning techniques. The focus of our exploration lies in the introduction of two innovative segmentation algorithms: W-net and ψ-net. These architectures have been meticulously crafted to overcome the challenges inherent in image segmentation tasks, offering novel insights and solutions to enhance segmentation performance.

The architecture known as W-net represents a fusion of established methodologies, combining the strengths of a ResNet encoder-decoder, a ConvNet encoder-decoder, and a Feature Pyramid network. The primary objective of the W-net is to provide a robust and accurate solution tailored to the complexities of image segmentation challenges. By integrating dual encoder-decoder frameworks, the W-net excels in identifying and highlighting regions of interest, resulting in enhanced segmentation outcomes. The strategic integration of a Feature Pyramid network within the W-net architecture addresses the concern of memory saturation, contributing to its overall effectiveness.

The second contribution, the ψ-net architecture, aims to elevate the capabilities of the baseline U-Net network. The architecture introduces several enhancements meticulously designed to amplify its segmentation performance. A pivotal advancement involves the

incorporation of a secondary convolutional encoder, fostering collaborative feature extraction between the two encoders to capture more comprehensive and informative features. Further augmenting the architecture are the innovative Fire-ResNet and Dense-Fire-ResNet blocks, designed to capture intricate image characteristics pivotal for accurate segmentation. Attention modules have been seamlessly infused into the ψ-net architecture, enabling refined focus during the segmentation process.
The chapter embarks on an in-depth exploration of these pioneering deep-learning architectures, revealing the intricate design principles and unique components of each. The discussion covers the W-net architecture in subsection 6.3 and introduces the ψ-net architecture in subsection 6.4. Subsequent sections offer a detailed exploration of each architectural element.

6.2 The proposed W-net model

The W-net architecture is a novel dual encoder-decoder architecture designed to address segmentation problems (Khouloud et al., 2022b). It aims to predict segmentation masks while simultaneously training on input images. The architecture consists of two main paths: the analysis path and the synthesis path. Each path incorporates two types of blocks: the ResNet block and the convolutional block. The analysis path, implemented as a ResNet encoder, includes multiple downsampling blocks that progressively reduce the spatial dimensions of the input image. On the other hand, the synthesis path, known as the ResNet decoder, utilizes upsampling blocks to reconstruct the spatial dimensions and refine the segmentation predictions. To compensate for feature loss caused by downsampling in the encoder stages, the W-net architecture incorporates a convolutional encoder-decoder network. This network comprises downsampling blocks in the encoder and corresponding upsampling blocks in the decoder.

The features extracted from the ResNet decoder and the convolutional encoder are combined using concatenation. This fusion of features at different levels enhances the model's robustness and strengthens its performance. Additionally, attention mechanisms are incorporated within the ResNet encoder-decoder to improve the network's performance. These attention mechanisms selectively highlight relevant features, further enhancing the segmentation accuracy. To optimize memory usage, an efficient feature pyramid is employed in the W-net architecture. This feature pyramid efficiently handles multi-scale information, aiding in the segmentation task while reducing memory consumption.

The schematic of the proposed method is shown in Figure 6.1.

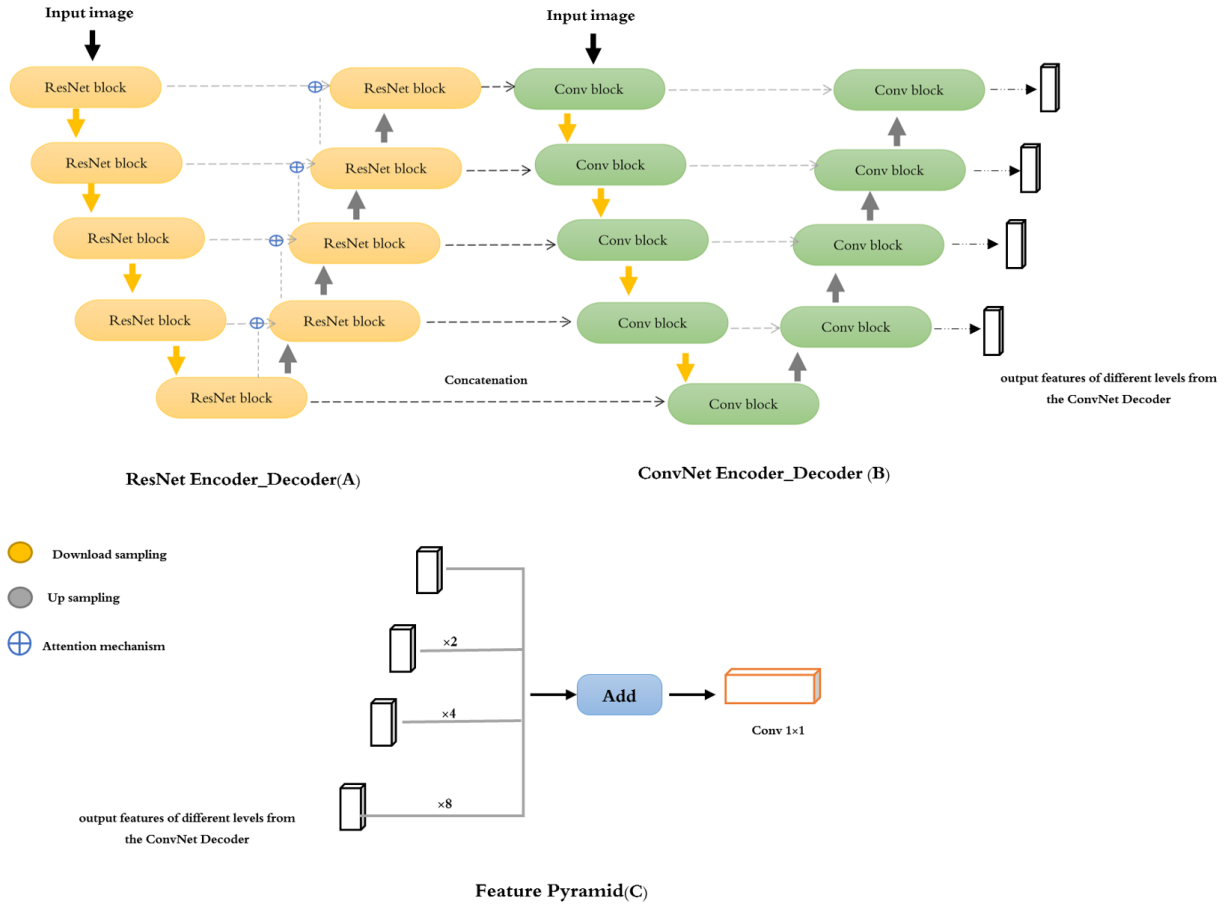


Figure 6.1: The Architecture of the W-Net: A dual encoder-decoder approach for segmentation tasks.

6.2.1 ResNet Encoder-Decoder

In the ResNet encoder component of the W-Net architecture, as described in (Khouloud et al., 2022b), the input training data is down-sampled and deep features are extracted. The ResNet encoder consists of five blocks, where the first four blocks are downsampling blocks, and the last block is a non-downsampling block. Each downsampling block consists of a residual unit and a downsampling layer.

The residual unit addresses the issue of deep gradient degradation by combining a residual map and an identity map, with the output obtained through the equation y = F(x) + x. Within each residual unit, there are two convolutional layers with a kernel size of 3 and a stride of 1. The number of convolutional kernels progressively increases from the first to the fourth downsampling block, with the values of 16, 32, 64, and 128, respectively. Batch normalization layers are incorporated after each convolutional layer to mitigate the vanishing gradient problem. Rectified Linear Unit (ReLU) activation is applied to the features following batch normalization. The last block in the ResNet encoder is the non-downsampling block, which does not include a pooling layer. Instead, it preserves the spatial dimensions of the data without further reduction. In contrast, the downsampling blocks reduce the width and height of the data by half in each block. The implementation details of the ResNet encoder in the W-Net architecture, as described above, can be visualized in Figure 6.2.

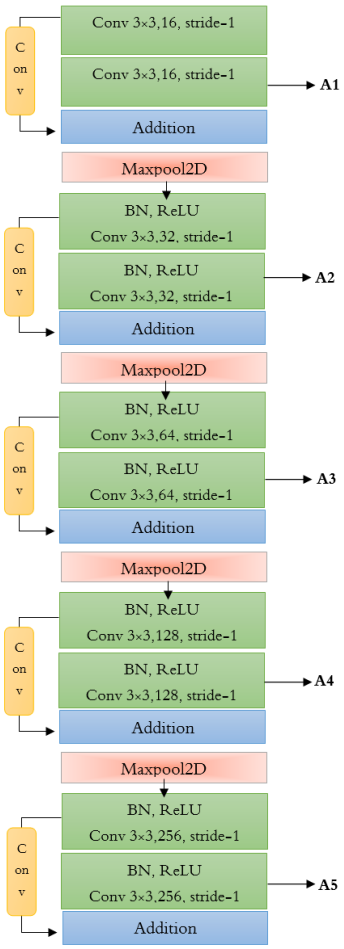


Figure 6.2: The implementation of ResNet encoder in W-net.

The ResNet decoder consists of four up-sampling blocks. Each block includes an up-sampling layer, a concatenate layer, and a ResNet block. The purpose of the up-sampling layer is to increase the spatial resolution of the features. The concatenate layer connects the features from the encoder with the up-sampled features from the decoder, preserving the information from the corresponding scales. The convolutional layers within the ResNet block have a kernel size of 3. After each convolutional layer, a batch normalization layer and a ReLU activation layer are applied. The number of convolutional kernels used in each up-sampling block is asymmetric, with the numbers being 128, 64, 32, and 16 for the first to fourth up-sampling blocks, respectively. This asymmetry helps improve the feature resolution of the decoder. To address the imprecise spatial information during up-sampling, the encoder-decoder model utilizes skip connections. These connections merge the spatial information from the down-sampling path (encoder) with the up-sampling path (decoder). However, since the earliest layers in the encoder may have poor feature representation, this can lead to duplicate low-level feature extractions.

To increase network efficiency and reduce non-task-related feature activation, an attention mechanism inspired by the work of (Oktay et al., 2018) is introduced. The attention gate module takes two inputs: vectors k and a. The vector a is obtained from the network's next-lowest layer. The vector x is passed through a strided convolution, while the vector g is processed by a 1x1 convolution. The two resulting vectors are element-wise added, promoting aligned weights and reducing unaligned weights. The combined vector goes through a ReLU activation layer and a 1x1 convolution. A sigmoid layer scales the vector between 0 and 1. The attention coefficients are up-sampled to match the original dimensions of the x vector using trilinear interpolation. These attention coefficients are then multiplied element-wise with the original x vector, scaling the vector based on its significance. The resulting vector is passed through the skip connection and further processed.The structure of the AG module is shown in Figure 6.3. The attention coefficients are computed based on Equations 6.1 and 6.2.

$$O = \sigma_2 \left\{ W_y \left[\left(\sigma_1 \left(W_k \times k + \left(W_a \times a + b_{ak} \right) \right) \right] + b_y \right\} \tag{6.1}$$

$$\sigma_2 \left(x \right) = \frac{1}{1 + e^{-x}} \tag{6.2}$$

Here, O represents the calculated attention coefficients. k and a are the feature maps given to the AG module's inputs from the decoder and encoder. W_a, W_y, W_k indicate the convolution kernels. σ_2 denote the sigmoid activation function and σ_1 is the ReLU function. The constructive features are output by the AG module by elementwise multiplying with the

relevant encoder layer output, The constructive features are generated by element-wise multiplying the relevant encoder layer output as shown in Equations 6.3 where out is the output of the AG module. The implementation of the Resnet decoder is shown in Figure 6.4.

$$a_{out} = O \times a \tag{6.3}$$

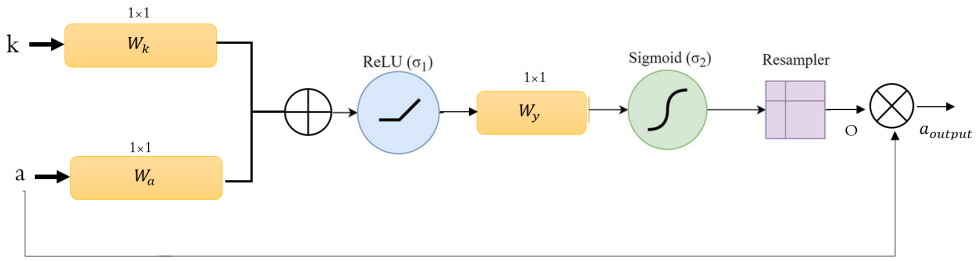


Figure 6.3: Schematic diagram of the Attention Gate (AG).

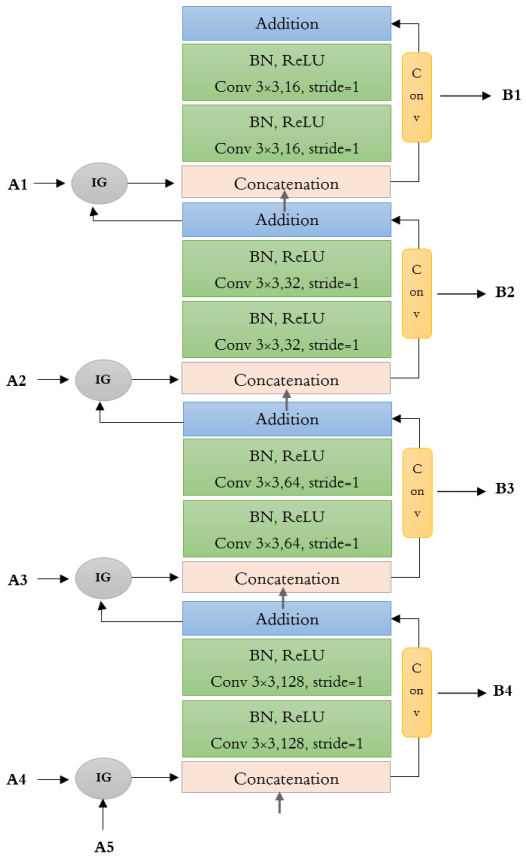


Figure 6.4: The implementation of ResNet decoder in W-net.

6.2.2 ConvNet Encoder-Decoder

The convolutional encoder component of the model consists of five downsampling blocks. Each block includes a convolutional block, a downsampling layer, and a concatenate layer. Downsampling is achieved by using convolutional layers with a stride of two and a kernel size of three. The number of convolutional kernels varies for each downsampling block, starting from 16 and doubling with each subsequent block, reaching 256 in the final block. Batch normalization and ReLU activation layers are applied after each convolutional layer. The concatenate layer plays a crucial role by connecting the features from the ResNet decoder with the corresponding features in the convolutional encoder. This accounts for the lost features during the downsampling process in the ResNet encoder due to max-pooling layers. By combining the ResNet decoder with ConvNet encoder features, the model focuses its coding process on the regions of interest, leading to improved segmentation accuracy. The concatenation operation enhances the network's feature extraction, representation abilities, and overall segmentation performance, resulting in a more robust model. During training, the data is processed through a total of four convolutional blocks. As a result, four features are obtained, each with dimensions equal to 1/1, 1/2, 1/4, and 1/8 of the original scale. The specific implementation details are shown in Figure 6.5 , and the calculation process is as follows:

$$C(x_i) = C(x_{i+1}) \oplus B(x_i) \tag{6.4}$$

where $C(x_i)$ is the output of the i_{th} downsampling block in the convolutional encoder. $C(x_{i+1})$ is the output of the (i + 1) th downsampling block in the convolutional encoder. $B(x_i)$ is the output of the ith downsampling block in the Resnet decoder.

The convolutional decoder in the model consists of four upsampling blocks, each comprising an upsampling layer, a concatenate layer, and a convolutional block. The upsampling layer utilizes interpolation techniques to increase the size of the features. The concatenate layer connects features of the same scale from the convolutional encoder with the features in the convolutional decoder after the upsampling layer. The convolutional layers within the decoder have a kernel size of three. The number of convolutional kernels decreases in an asymmetric manner from the first to the fourth upsampling block, with 128, 64, 32, and 16 kernels, respectively. This asymmetry is designed to enhance the feature resolution of the decoder. Batch normalization and ReLU activation layers are applied after each convolutional layer to improve the performance of the network. During the upsampling process, the width and height of the features are doubled, and the final upsampling block produces an output channel number of 16. The output of the convolutional decoder

maintains the same width and height as the original scale. The features extracted from each block are preserved as the outputs of the convolutional decoder, denoted as D1, D2, D3, and D4. Figure 6.6 shows this implementation.

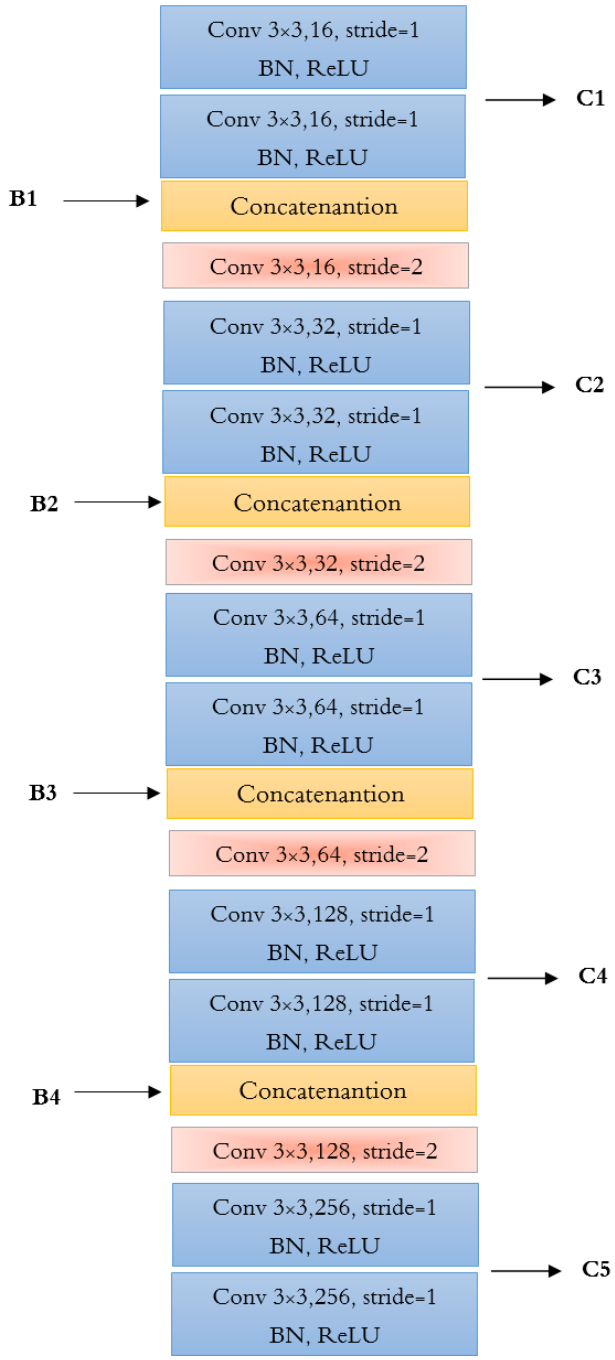


Figure 6.5: Convolutional Encoder Implementation in W-net Architecture.

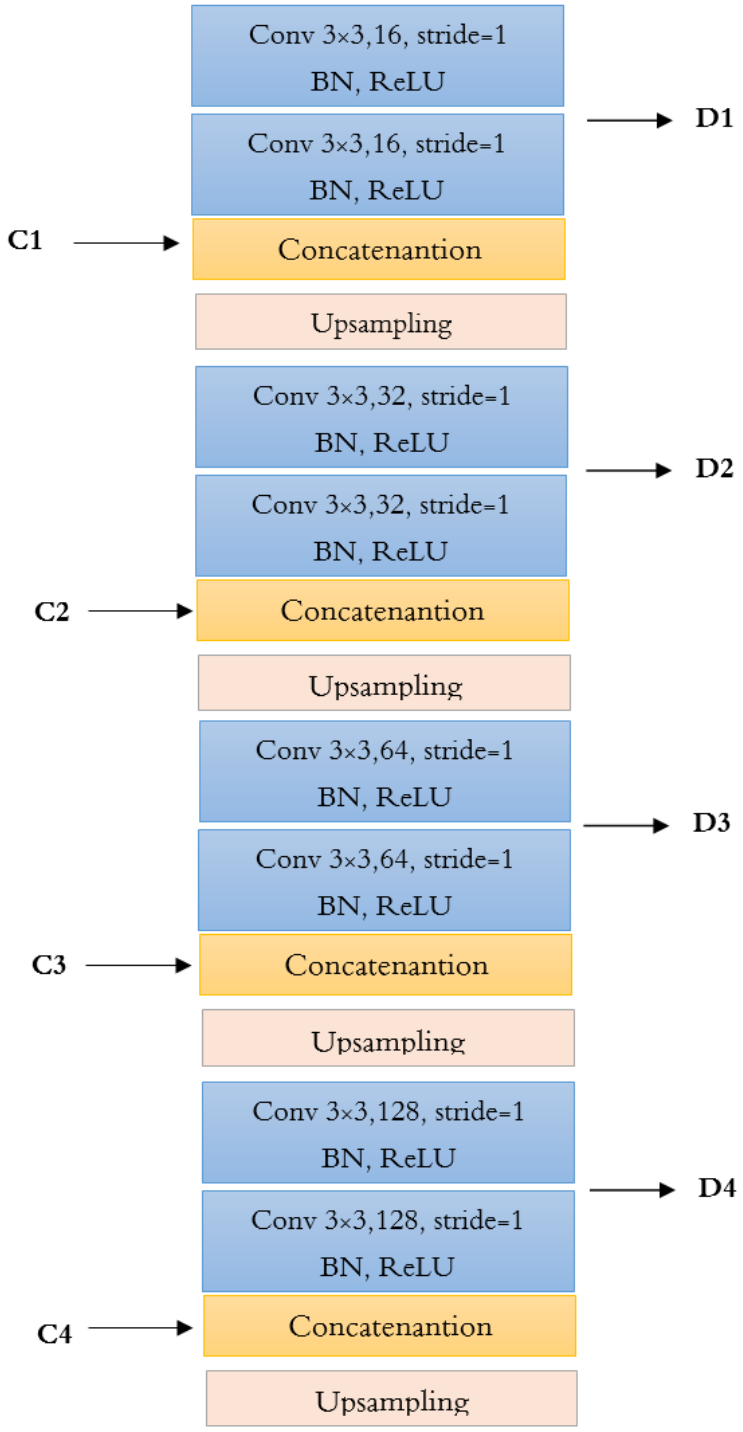


Figure 6.6: The implementation of the decoder in W-net.

6.2.3 Feature Pyramid

In our network, we implement a feature pyramid (FP) architecture inspired by the work of (Lin et al., 2021) on an efficient feature pyramid (EFP). The FP architecture allows us to create semantic feature maps at multiple scales, capturing information at different resolutions (multi-scale feature maps), which is crucial for handling objects of varying sizes effectively.

At the beginning of our implementation, we extract the output features from different layers of the convolutional decoders (D1, D2, D3, and D4). These features are then compressed using a 1x1 convolutional layer with a stride of 1, followed by batch normalization and the ReLU activation layer. The purpose of this compression is to reduce the channel number to the smallest value, which is set as 16. Subsequently, we upsample the compressed features to restore them to their original scale. However, to optimize storage space utilization, we further compress the channel number of the upsampled features. This step is necessary because feature maps with higher channel dimensions in the Feature Pyramid require more storage after upsampling. The compressed and upsampled features from different layers are combined by addition, allowing us to integrate information from multiple scales. Finally, the resulting feature map is processed through a 1x1 convolutional layer with the sigmoid activation function to obtain the final segmentation result.

The implementation is shown in Figure 6.7, and the calculation process is as follows:

$$F(x) = Sigmoid\,[E(x)] \tag{6.5}$$

$$F(x) = Sigmoid[\sum_{n}^{i=0} D(x_i)] \tag{6.6}$$

$$F(x) = Sigmoid[\sum_{n}^{i=0} [D(x_{i+1}) \oplus c(x_i)]] \tag{6.7}$$

where E(x) is the output of an efficient feature pyramid. F (x) is the output of the W-net model.

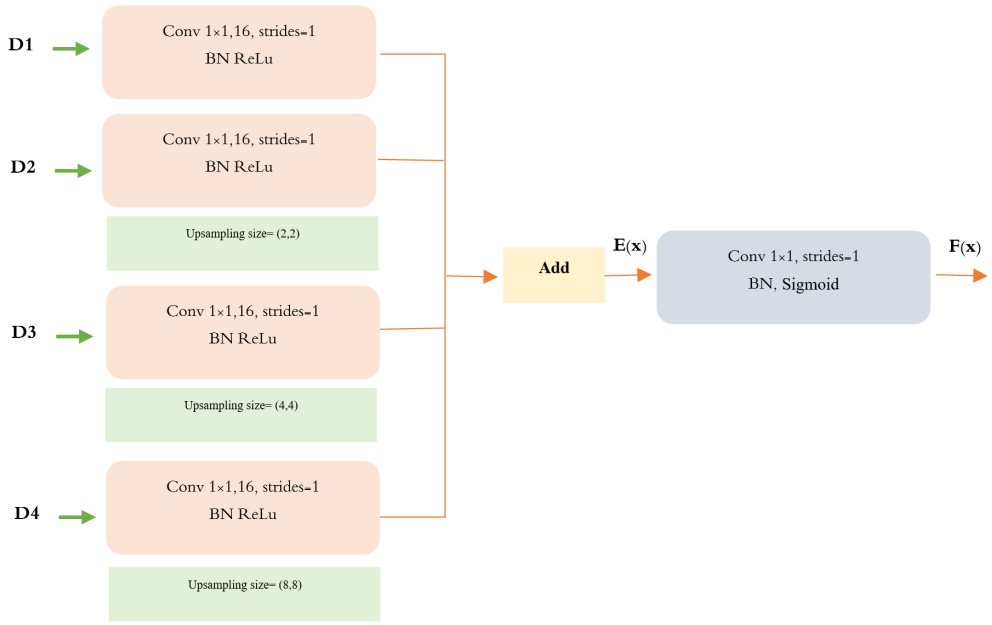


Figure 6.7: The Feature Pyramid Structure in W-net.

6.3 The proposed ψ-net model

In this thesis, an other architecture named ψ-net is presented for robust semantic segmentation. The design of ψ-net is influenced by several existing networks including SqueezeNet, DenseNet, ResNet, and U-Net. The workflow of the proposed network is depicted in Figure 6.8. It consists of three main parts: a multi-scale convolutional encoder, a DENSE-Fire U-Net encoder, and a DENSE-Fire U-Net decoder.

The DENSE-Fire U-Net architecture is introduced as a modification of the traditional U-Net, where the standard convolutional layers are replaced with Fire-ResNet modules that employ different convolutional kernel sizes. This modification aims to create a broader network without the problem of gradient disintegration. Additionally, a Dense-Fire-ResNet block is integrated into the center of the U-Net network to increase its depth. This block generates feature maps of different scales, which are then downsampled using strided convolution to extract more detailed feature representations in subsequent layers of the network. The DENSE-Fire U-Net encoder contains three Fire-ResNet downsampling blocks, while the decoder has the same number of channels as the encoder.

To further enhance the feature extraction capability of the network, a parallel encoder called the multi-scale convolutional encoder is added. This encoder consists of five downsampling blocks that use convolutional and pooling layers to extract important features from the original image.

To address the limitations of the encoder-decoder model, two types of attention modules are employed: spatial gate attention and channel attention modules. The spatial gate attention (SAG) module focuses on the spatial structure of significant regions, strengthening feature maps in the region of interest while suppressing background or unnecessary components. On the other hand, the channel attention module calibrates the concatenation of low- and high-level features to incorporate more contextual information into the low-level encoder features.

In the proposed model, different output levels of feature maps from the expansive decoder path are aggregated and concatenated to improve mask prediction and facilitate joint optimization.

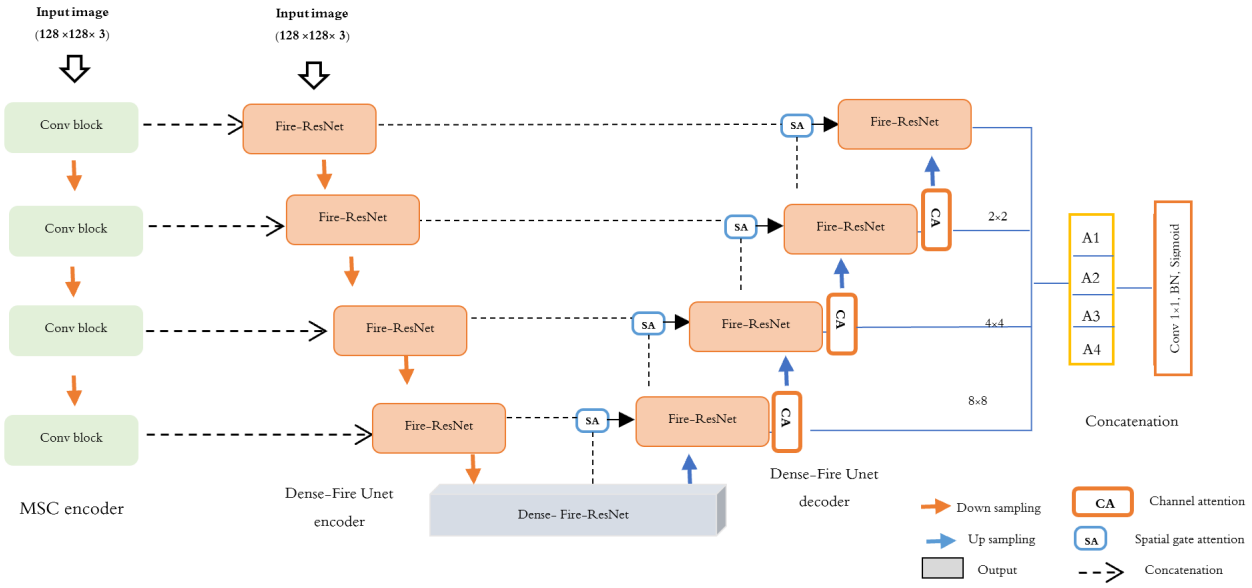


Figure 6.8: systematic view of the proposed ψ-Net architecture.

6.3.1 Multi-scale Convolutional encoder

The MSC encoding (Multi-scale Convolutional) module is composed of five convolutional blocks. Each block consists of a repeated application of two convolution layers with a fixed kernel size. The number of kernels in each block progressively increases from 16 to 32, 64, and 128, respectively. To mitigate the issue of vanishing gradients, a batch normalization layer is applied after each convolutional layer. The rectified linear unit (ReLU) activation function is then used to activate the features. Following the activation, a convolutional layer with a stride size of 2 is employed for downsampling, similar to the max-pooling

layer in the U-Net architecture.

As illustrated in Figure 6.9, the convolutional blocks in the MSC encoding module continuously extract semantic information from the input while simultaneously downsampling the feature maps into four different levels. This downsampling operation reduces the width and height of the feature maps by half compared to their original size.

To improve the network's feature extraction and representation capabilities, the acquired feature maps, denoted as A(A1), A(A2), A(A3), and A(A4), are concatenated with the feature maps from the corresponding encoder's blocks at different scales. This fusion of feature maps from different scales allows the network to capture and utilize information from various levels of detail, thereby enhancing its performance in semantic segmentation tasks.

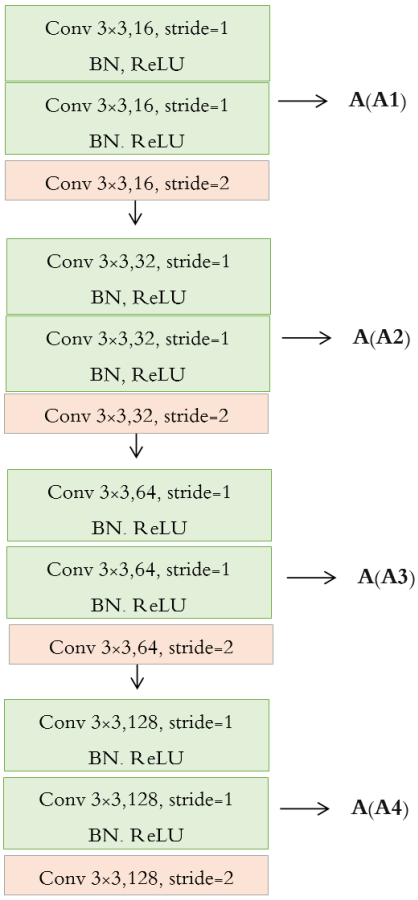


Figure 6.9: The implementation of MSC encoder in ψ-Net architecture.

6.3.2 Dense-Fire U-Net

The encoding module of the Dense-Fire U-Net consists of five downsampling blocks. The first four blocks are Fire-ResNet (FR) downsampling blocks, and the final block is a Dense-Fire-ResNet non-downsampling block. The key component of the Dense-Fire U-Net encoder is the Fire-ResNet module.

The Fire-ResNet module starts with a squeeze convolution layer, followed by parallel 1×1 filters that provide a combination of parallel 3×3 and 5×5 convolution filters in the expand layer. To reduce dimensions, an additional 1×1 convolution layer is added before each layer. Batch normalization (BN) layers are incorporated after each convolutional expand layer to normalize the activations and prevent the issue of vanishing gradients. Furthermore, an identity mapping convolution layer of size 1×1 connects the input and output of the module. The main objective of the Fire-ResNet module is to integrate feature maps from different branches with different kernel sizes. This allows the network to expand and learn a broader range of features while simultaneously being able to recognize objects of varying sizes. The inclusion of residual connections is beneficial for learning, as it enables the module to learn a function with respect to the input feature maps, facilitating optimization and improving the overall performance of the network. Figure 6.10 provides a schematic representation of the Fire-ResNet module. The Fire-ResNet unit's computation procedure is as follows:

$$M_{i+1} = K_{1\times1}(M_i + [\text{Bn}\,(\,K_{1\times1}(K_{1\times1}(M_i)) \oplus \text{Bn}(K_{3\times3}(K_{1\times1}(K_{1\times1}(M_i\,)))) \oplus \text{Bn}(K_{5\times5}(K_{1\times1}(K_{1\times1}\,(M_i))))]$$

Where $K_{n\times n}$ present the kernel size of convolutional layer and Fb denotes the BN layer. $\oplus$ present the concatenation function and M_i is the identity mapping of the i th layer.

The Fire-ResNet module is integrated into the dense connection block, which is inserted in the network's bridge. Specifically, the dense connection block consists of 12 Fire-ResNet modules. The main objective of incorporating the Fire-ResNet module into dense connections is to create a deeper and larger network while addressing the issues of gradient vanishing and redundant processing. By integrating the Fire-ResNet module into dense connections, the network benefits from increased depth and a wider capacity to learn and extract features. The dense connection block facilitates direct and efficient information flow between layers, ensuring that all layers have access to the collective knowledge of previous layers.

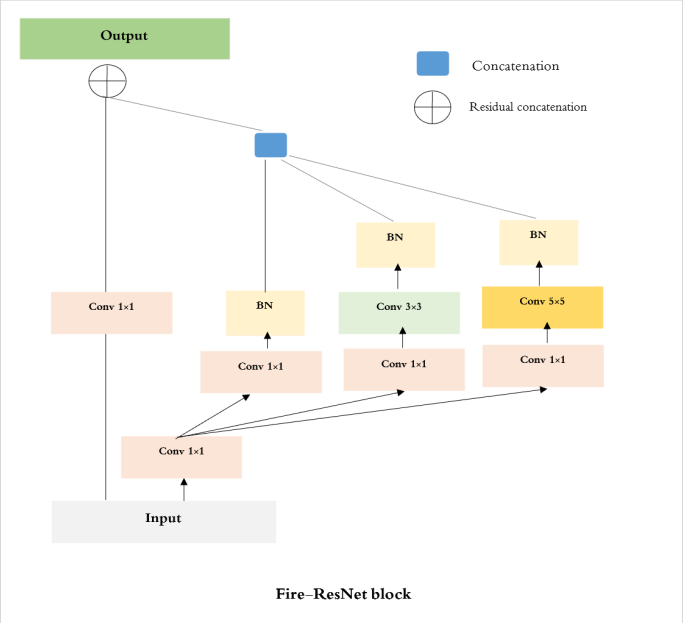


Figure 6.10: Structure of The Fire-ResNet block.

This allows for the preservation and utilization of valuable information throughout the network. The structure and functionality of the Dense-Fire-ResNet block can be visualized in Figure 6.11, providing an illustration of how the Fire-ResNet modules are organized within the dense connection block. The Dense- Fire-ResNet block's i + 1 th layer calculation procedure output is as follows:

$$M_{i+1} = f_{FR} = ([M_0, M_1, ..., M_i]) \tag{6.8}$$

f_{FR} indicates the Bottleneck layer's function, followed by the suggested Fire-ResNet module. Where $M_0, M_1, ..., M_i$ relates to the feature maps in layers 0 to ith.

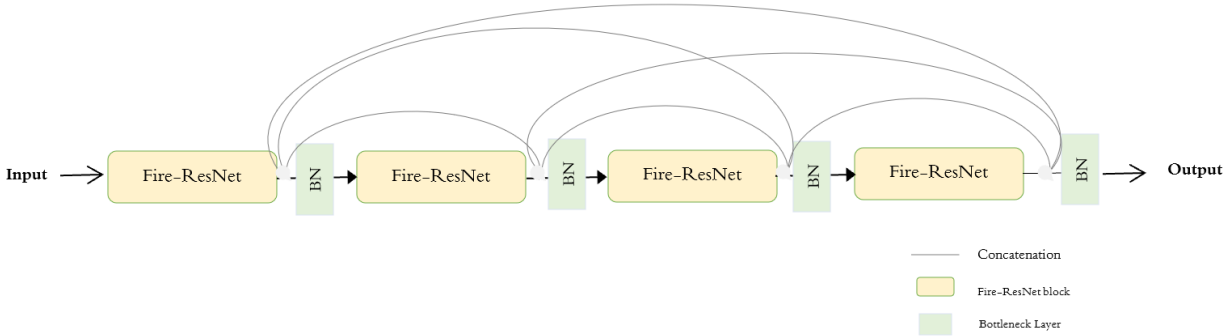


Figure 6.11: The structure of Dense Fire-ResNet block .

In the Dense Fire-ResNet encoder, each Fire-ResNet (FR) block is composed of a concatenate layer and a downsampling layer. The downsampling process is achieved using a convolutional layer with a stride of two, effectively reducing the spatial dimensions of the feature maps.The concatenate layer plays a critical role in connecting features of the same scale within the Dense-Fire U-Net encoder to the features in the MSC encoder (Multi-scale Convolutional).
This connection is established to enhance the network's feature extraction capabilities. By merging features from different scales, the network is empowered to capture and exploit features at various levels, resulting in improved segmentation performance.
Figure 6.12 illustrates the Dense-Fire-ResNet block, providing a visual representation of how the concatenate layer and downsampling layer are integrated within the block.

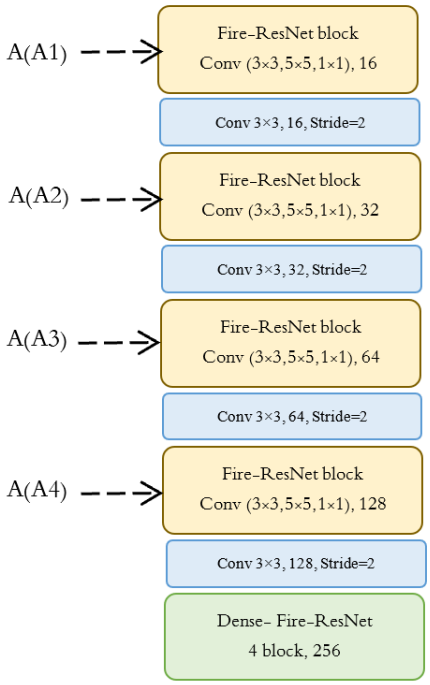


Figure 6.12: The Implementation of Dense-Fire Unet Encoder.

The Dense-Fire U-Net decoder consists of four Fire-ResNet blocks, each responsible for doubling the width and height of the features through upsampling. The number of convolution kernels used in these blocks is 128, 64, 32, and 16, respectively, as the features progress through each block. While the Dense-Fire U-Net architecture follows a basic encoder-decoder mechanism with multi-scale skip connections, it can suffer from information and computational resource wastage. The low-level characteristics of the encoder contain valuable spatial information. However, during the fusion process, where signif-

icant regions are emphasized, the network may lose important contextual information from the encoder features. As a result, the encoder-decoder model still has limitations and requires further investigation to enhance its performance. To address these limitations, the proposed architecture introduces the spatial-channel attention gate mechanism. This mechanism incorporates four spatial gate attention modules and three channel attention modules into the Dense-Fire U-Net decoder. These attention modules learn attention maps at multiple resolution levels to guide the network's focus.

The spatial gate attention module is an improvement over existing spatial attention modules, such as CBAM (Woo et al., 2018), and AG (Oktay et al., 2018), with the goal of enhancing regions of interest and suppressing irrelevant parts. The spatial gate attention map is created by combining both encoder and decoder features, incorporating spatial information. Figure 6.13 provides a visualization of this process.

The spatial attention map is generated by performing average and max pooling operations over the channel dimension of the feature maps in both the encoder and decoder. The resulting features are concatenated and further processed through several layers, including ReLU activation, 1x1 convolution, and sigmoid activation, to produce the final attention map. This attention mechanism allows the network to focus on the most relevant regions for improved segmentation results. Our spatial gate attention module produces the following output:

$$Se\,(i) = \; Conv_{7\times7} \; ([AvgP\,(i) \oplus MaxP\,(i)]) \tag{6.9}$$

$$Sd\,(i) = Conv_{7\times7} \; ([AvgP\,(i) \oplus MaxP\,(i)]) \tag{6.10}$$

$$Ss\,(i) = \sigma\,(Conv_{1\times1} \; (Relu\,(Se\,(i) \oplus SAd\,(i)))) \tag{6.11}$$

Where i is the original input feature, $AvgP\,(i)$ is an average pooling, $MaxP\,(i)$ present max pooling and $Conv_{1\times1}$ is 1× 1 convolution.$Se\,(i)$,$Sd\,(i)$ denote the generated spatial attention maps of the encoder and decoder features.$Ss\,(i)$ is The ReLU's feature map sent into another 1×1 convolution with one output channel, followed by a Sigmoid function, to produce a pixelwise attention coefficient.

The Channel attention module in the proposed architecture combines the spatial attention from the encoder with higher-level features from the decoder by assigning higher coefficients to important channels. The channel attention map is created by utilizing average and max pooling operations, along with a shared multi-layer perceptron that includes a single hidden layer. The output of the Sigmoid layer is then multiplied element-wise with the original input feature, resulting in the final output of the channel attention module. This process is visually depicted in Figure 6.14.

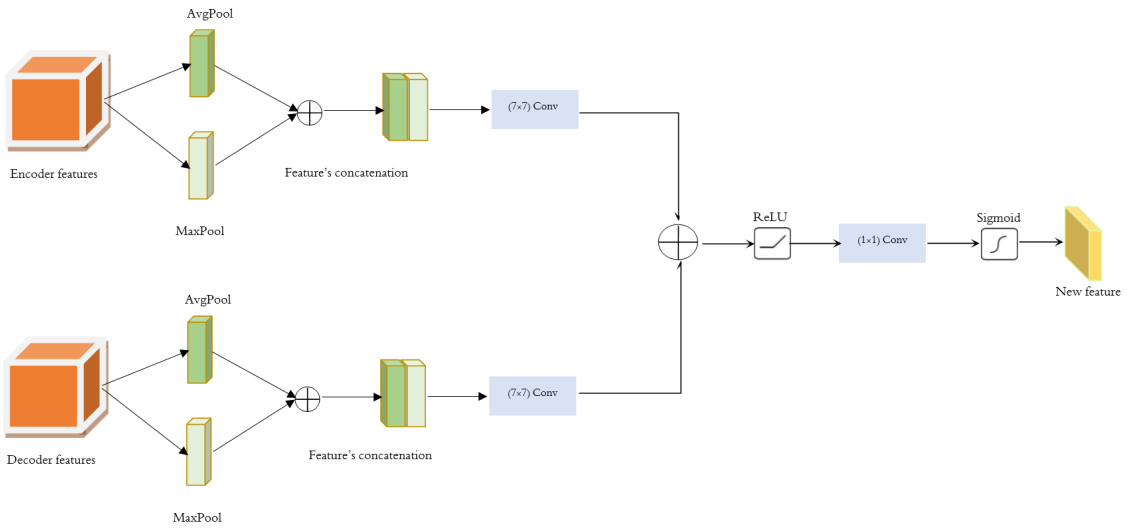


Figure 6.13: Implementation of the Spatial Gate Attention module.

The channel attention module's output is as follows:

$$Sch\,(i) = \sigma\left[(MLP\,(AvgP\,(i))) \oplus (MLP\,(MaxP\,(i)))\right] \tag{6.12}$$

$$SchA\,(i) = Sch\,(i) \otimes i \tag{6.13}$$

MLP represent multi-layer perceptron, $\otimes$ represents element-wise multiplication and, $SchA$ is the Channel Attention output

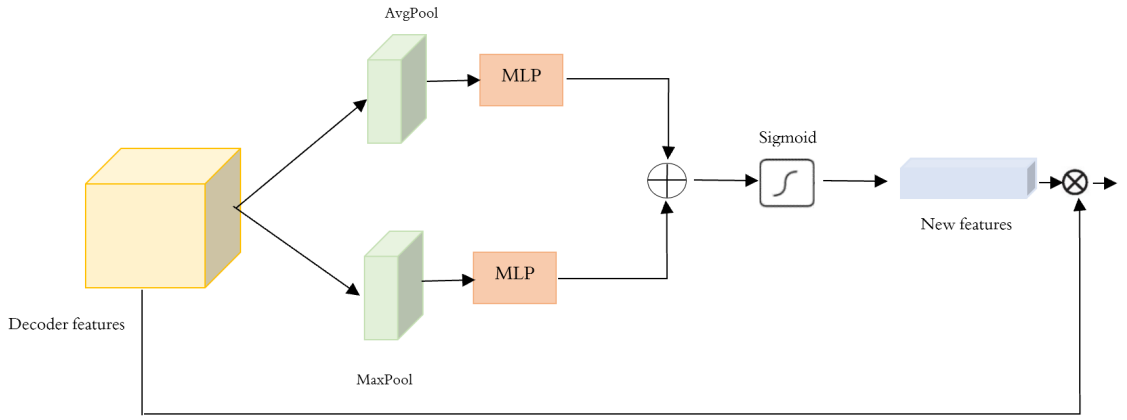


Figure 6.14: Implementation of the Channel Attention Module.

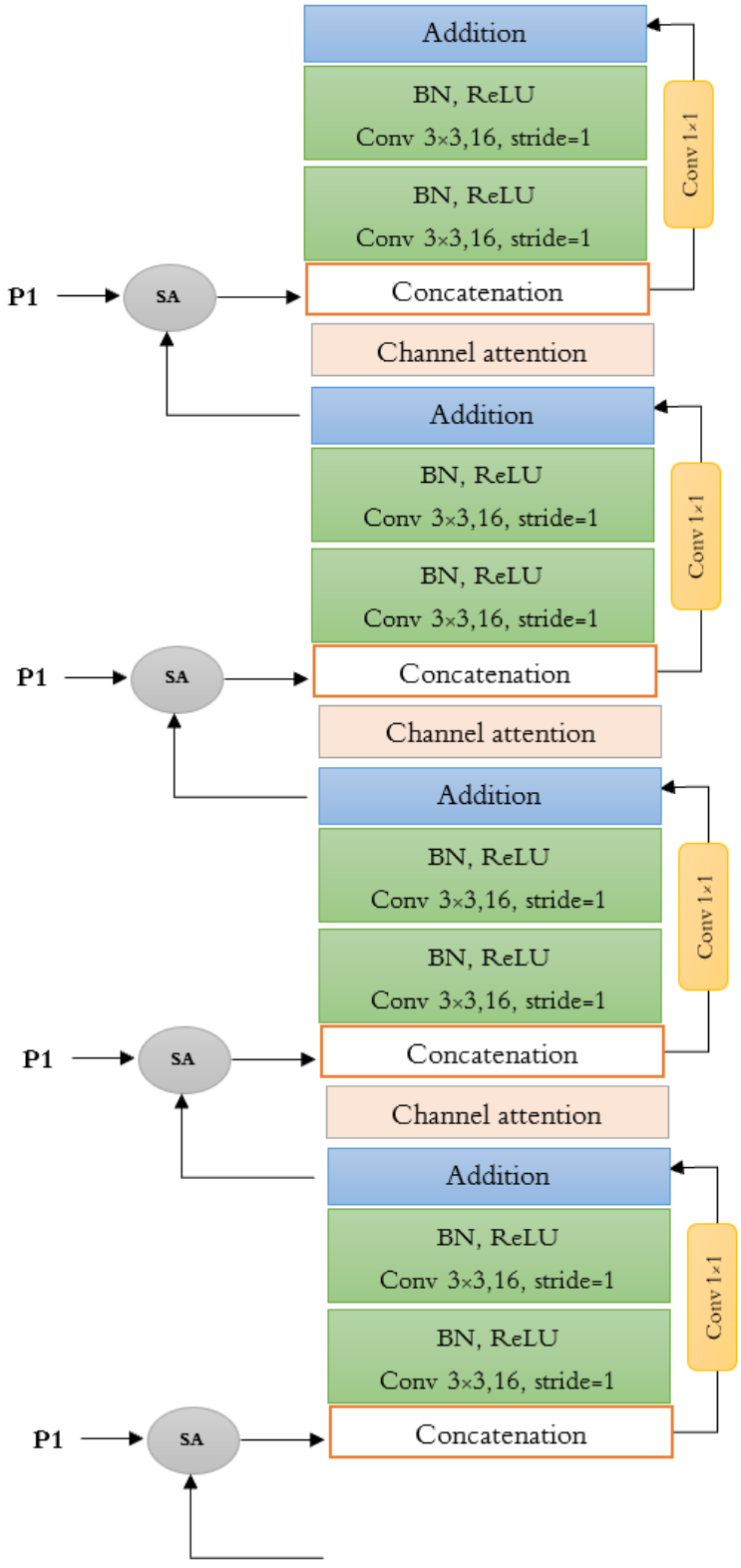


Figure 6.15: Implementation of the Dense-Fire U-Net Decoder.

6.3.3 Feature Pyramid

The feature pyramid (FP) is a crucial technique utilized in the ψ-net architecture to enhance mask prediction. It achieves this by combining the output features from the decoder at different scales. The features obtained from multiple scales are concatenated together in a pixel-wise manner and then processed through a convolutional layer. The resulting feature representation is activated using the softmax function to obtain the final segmentation output. Figure 6.16 provides a visual representation of the structure of the feature pyramid in ψ-net.

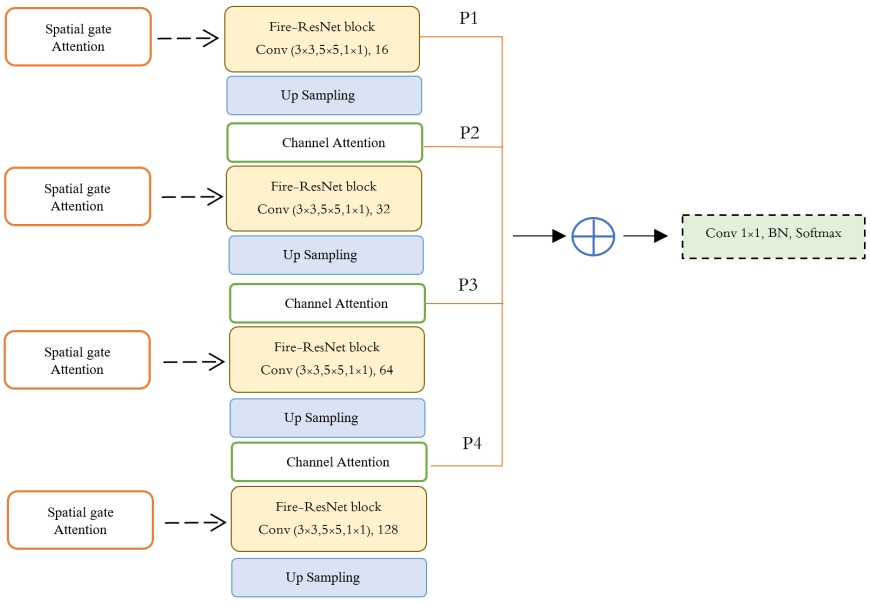


Figure 6.16: The Structure of the Feature Pyramid in ψ-net.

6.4 Conclusion

In this chapter, we have covered the essential background and theoretical foundations of the methods utilized in this thesis. The chapter is divided into two main sections.
The first part introduced the W-net model, our proposed image segmentation architecture. It utilizes a dual encoder-decoder structure, incorporating elements like a ResNet encoder-decoder, a ConvNet encoder-decoder, and a Feature Pyramid network. Additionally, we presented the ψ-net another architecture in the second section. The ψ-net draws inspiration from established networks like SqueezeNet, DenseNet, ResNet, and U-Net, aiming for robust semantic segmentation. By outlining the key components of the

ψ-net, we've provided a clear understanding of its design principles and its role in our image segmentation approach.

With this groundwork, the following chapters will explore experimental results, analysis, and discussions on the performance of both the W-net and ψ-net models. This evaluation will offer insights into their effectiveness and potential implications for image segmentation applications.

Chapter 7

PROPOSED DEEP LEARNING MODEL FOR MEDICAL IMAGE CLASSIFICATION

7.1 Introduction

The realm of pattern recognition has been significantly reshaped by the emergence of machine learning, enabling computers to autonomously uncover significant patterns within data without requiring human intervention. In this chapter, we introduce the Inception-Resnet model, a technique designed for image identification. Rooted in the convolutional neural network framework, this model comprises essential elements such as convolutional layers, pooling layers, and Inception-Resnet blocks. The key contribution of our work lies in the Inception-Resnet block, which combines the strengths of the Inception architecture and the Residual Neural Network (ResNet). Through this integration, the Inception-Resnet model achieves heightened accuracy in tasks related to image classification. Our exploration takes us into the intricate design details of the Inception-Resnet block in subsection 7.3, followed by the presentation of the specific model architecture in subsection 7.4.

7.2 The proposed Inception Resnet block

The proposed Inception Resnet block merges the Inception network with ResNet to take advantage of their respective strengths in image classification. It introduces the concept of the residual inception block, which allows for accurate predictions across multiple layers, surpassing the performance of a simple inception block. This network component is inspired by the Inception concept and serves as a crucial element in the classification network proposed in the thesis. The Inception Resnet block consists of several layers, including a (1x1) convolutional layer, two residual blocks, and a max-pooling layer followed by a (1x1) convolutional layer. The residual unit within the block comprises two convolutional blocks. The first block combines (1x1) and (3x3) convolutions, while the second block combines (1x1) and (5x5) convolutions. The parallel processing of different spatial sizes, ranging from precise details (1x1) to larger sizes (5x5), is a key aspect of this block. To address the computational complexity associated with deep neural networks, an additional (1x1) convolutional layer is introduced before the (3x3) and (5x5) convolutions. This layer helps reduce the number of input channels, thereby improving computational efficiency. Each convolutional block consists of a batch normalization (BN) layer, a ReLU activation layer, and an identity mapping that connects the input and output through a (1x1) convolutional layer.

An overview of the Inception-ResNet block is depicted in Figure 7.1, illustrating its architecture and connections. The proposed Inception Resnet block is further detailed in Algorithm 1, describing the step of the computation process. The calculation process of the inception-ResNet unit is as follows:

$$M_{i+1} = K_{1\times1}\left(M_i\right) \oplus \left[K_{1\times1}\left(M_i\right) + \left(K_{5\times5}\left(Bn\left(K_{1\times1}\left(Bn\left(M_i\right)\right)\right)\right)\right)\right] \tag{7.1}$$

$$\oplus \left[K_{1\times1}\left(M_i\right) + \left(K_{3\times3}\left(Bn\left(K_{1\times1}\left(M_i\right)\right)\right)\right)\right] \oplus K_{1\times1}\left(Maxpool\left(M_i\right)\right) \tag{7.2}$$

Where $K_{n\times n}$ present the kernel size of convolutional layer and Bn denotes the BN layer. $\oplus$ present the concatenation function and M_i is the identity mapping of the i th layer maxp present Maxpooling layer.

Algorithm 1: The proposed Inception Resnet block.

Data: input

Result: features

Stage 1: ;

```
/* S1=Maxpooling(input)                                    */
/* S1= Conv 1x1(S1)                                        */
```

Stage 2:;

```
/* S2= Conv 1x1 (input)                                    */
/* S2= Bn(S)                                               */
/* S2= ReLu(S2)                                            */
/* S2= Conv 3x3(S2)                                        */
/* S2= Concat(input,S2)                                    */
```

Stage 3:;

```
/* S3= Bn(input)                                           */
/* S3= ReLu(S3)                                            */
/* S3= Conv 1x1(S3)                                        */
/* S3= Bn(S3)                                              */
/* S3= ReLu(S3)                                            */
/* S3= Conv 5x5(S3)                                        */
/* S3= Concat(input,S3)                                    */
```

Stage 4: ;

```
/* S4= Conv 1x1(input)                                     */
```

output ;

```
/* output= Concat (S1,S2,S3,S4)                            */
```

7.3 The proposed Inception-Resnet network

The proposed Inception-Resnet network is a deep learning approach for image classification, specifically utilizing convolutional neural networks (CNNs). CNNs are well-known for their ability to directly learn features from raw image data, eliminating the need for manual feature extraction. The network architecture consists of input and output layers, with several hidden layers in between, including convolutional, max-pooling, and fully connected layers.

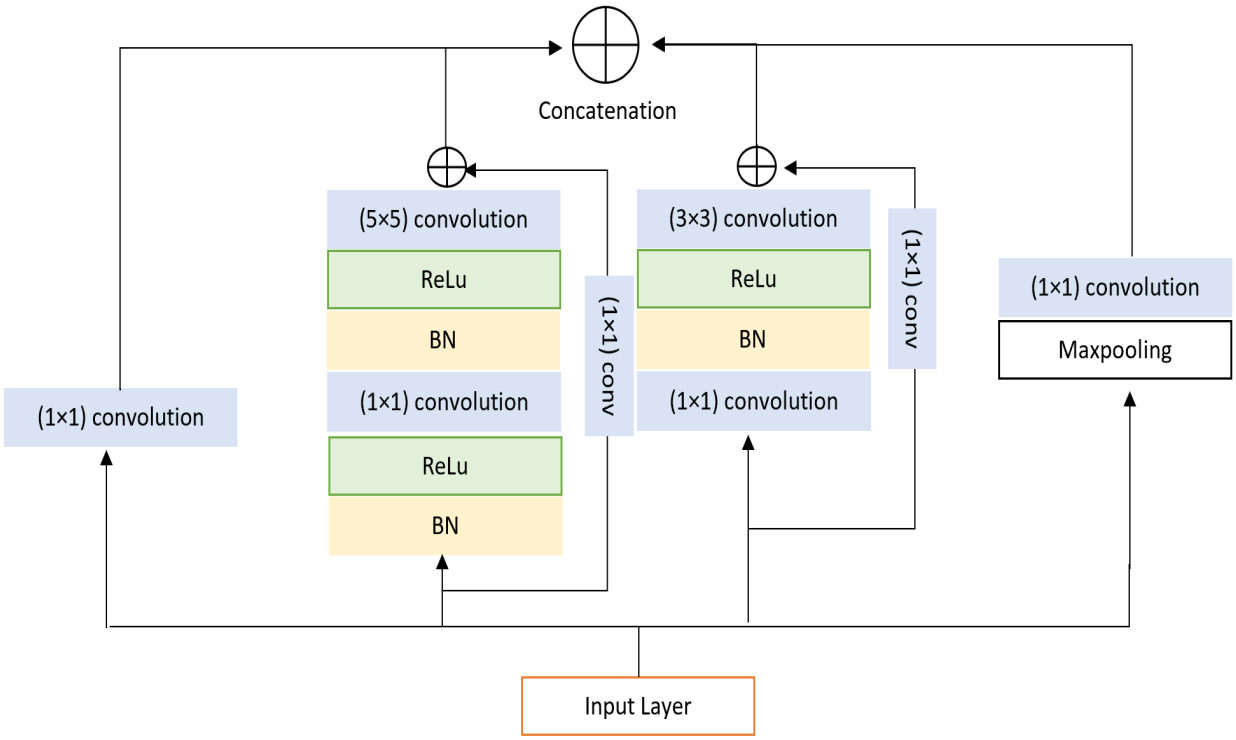


Figure 7.1: The proposed Inception Resnet block structure.

Our proposed method shares similarities with the GoogleNet network, but it introduces a unique integration of the inception ResNet module inspired by the Google LeNet inception module. The inception layer aims to capture a wide context while preserving fine details in images. Additionally, the Deep Residual component addresses the challenge of performance degradation in deeper networks by utilizing residual mappings instead of requiring layers to learn the complete underlying mappings. This helps improve model performance.

The Inception-Resnet network is designed to be a deep CNN with a smaller parameter count compared to networks such as AlexNet and GoogleNet. This architectural choice makes the model faster and more accurate. The input size to the network is 118x118x3, and the feature extraction phase involves three convolutional layers with different sizes and kernel numbers. The layers include batch normalization and dropout regularization. The network also incorporates five pooling layers, consisting of four max-pooling layers with a kernel size of 3x3 and one average pooling layer with a size of 7x7. Furthermore, the network includes four proposed Inception-Resnet blocks.

The classification phase of the network utilizes fully connected layers, followed by a softmax layer to calculate the probabilities for the best-matched class. The model architecture is depicted in Figure 7.2, and Table 7.1 provides details about the various parameters of the proposed model. The algorithm outlining the implementation of the Inception Resnet model is presented in Algorithm 2.

Algorithm 2: The proposed Inception Resnet Model for Image classification.

Data: Image im, size 224×224 RGB

Result: q Probability Value of the image classification.

Input im to the three convolutional layers, and receive the output C ;

Input C to the four Inception Resnet blocks, and receive the output I_R ;

Input I_R to the AvgPool2D layer to obtain the feature vector Avg;

Input Avg to the final FC layer, which has n neurons, and use a sigmoid activation function to generate the classification probability value q ;

return q ;

Table 7.1: The parameters of the proposed Inception-Resnet model.

Type	N Filter	Kernel size	Stride	Padding	Conv (1×1)	Conv (5×5)	Conv (3×3)	Conv (1×1)
Convolution	64	7×7	2	same	-	-	-	-
Maxpooling	-	3×3	2	same	-	-	-	-
Convolution	64	1×1	1	same	-	-	-	-
Convolution	192	1×1	1	same	-	-	-	-
Maxpooling	-	3×3	2	same	-	-	-	-
InceptionResnet block	-	-	-	same	64	96	128	16
InceptionResnet block	-	-	-	same	128	128	192	32
Maxpooling	-	3×3	2	same	-	-	-	-
InceptionResnet block	-	-	-	same	192	96	96	16
Maxpooling	-	3×3	2	same	-	-	-	-
InceptionResnet block	-	-	-	same	256	160	320	32
Average Pooling	-	7×7	1	same	-	-	-	-

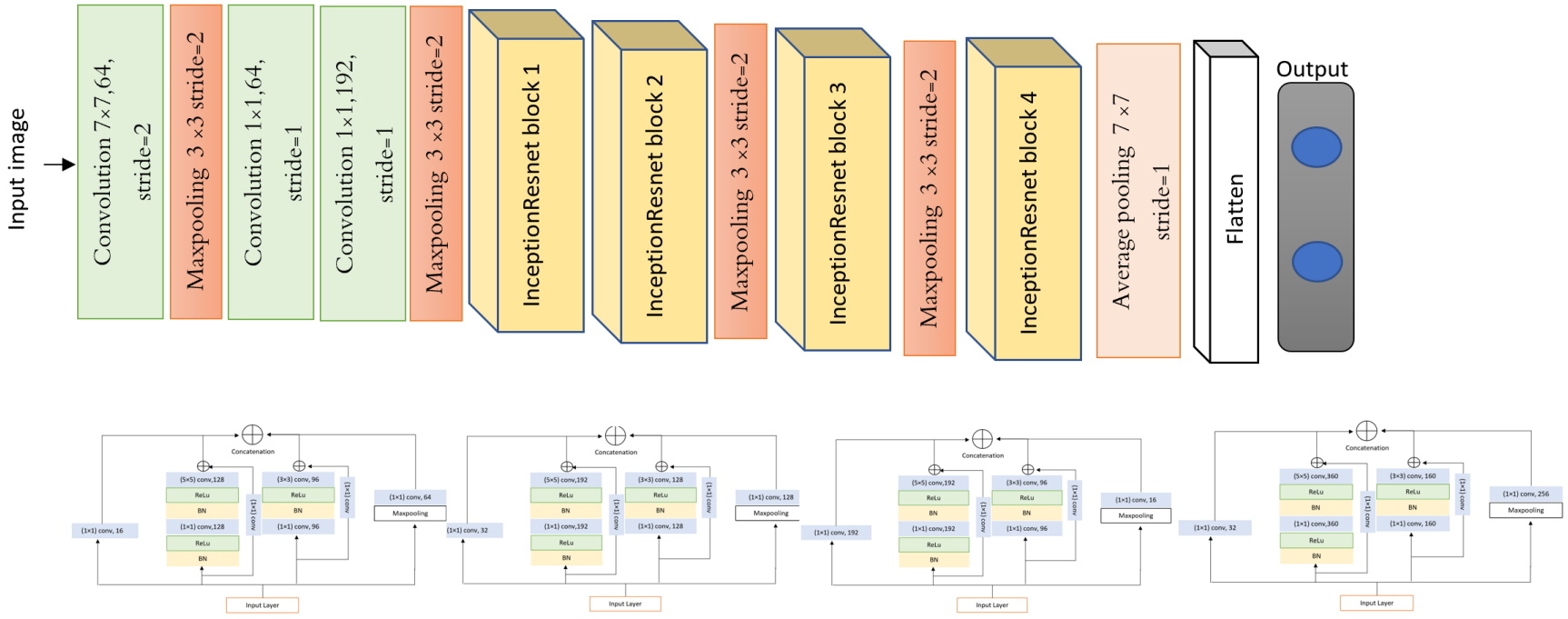


Figure 7.2: The proposed Inception Resnet model structure.

7.4 Conclusion

In this chapter, we have introduced a classification model that combines features from both the Inception and ResNet architectures by incorporating the Inception-ResNet block. The primary objective is to improve the classification model's performance by leveraging the strengths of both architectures. Through the integration of the Inception-ResNet block, the model gains the capacity to capture a broader context while preserving intricate details, resulting in enhanced accuracy in image classification tasks.

The proposed Inception-ResNet classification model presents a promising avenue for advancing image classification. By merging Inception and ResNet architectures and providing detailed insights into the implementation process, our aim is to highlight the model's potential effectiveness. In the upcoming chapter, we intend to apply the model to skin lesion datasets, exploring its capabilities in the specific domain of skin cancer. This step is essential to validate the model's performance and broaden its applicability in real-world scenarios.

Chapter 8

SKIN LESION DETECTION APPROACH

8.1 Introduction

Skin cancer is a significant global health concern, and its early detection is of utmost importance for improving patient outcomes. Dermoscopy, a non-invasive diagnostic method, has become a widely utilized technique by dermatologists for the early detection of malignant melanoma (Wighton et al., 2011). This method enables the visualization of imperceptible pigmented structures such as streaks, dots, pigment networks, and blue-white patches, which are not visible to the naked eye (Zhou et al., 2008). Despite the advantages of dermoscopy, the automated detection of melanoma from dermoscopy images remains a challenging task due to various obstacles.

The primary challenges in automated melanoma detection include the difficulty in precisely segmenting lesions due to the poor contrast between skin lesions and normal skin. Additionally, distinguishing between melanoma and non-melanoma lesions poses a significant challenge, as they often exhibit a high degree of visual resemblance (Erol, 2018). To tackle these challenges, researchers have turned to computer-aided diagnostic (CAD) methods, seeking to bridge the gap between medical expertise and engineering capabilities.

CAD systems aim to emulate dermatologists' performance in analyzing skin lesion regions, thereby assisting in the rapid and efficient differentiation between melanoma and benign tumors (Silveira et al., 2009) (Barata, Marques, and Mendonça, 2013). Dermoscopy image analysis involves enhancing images and segmenting the regions of interest (ROI)

containing the lesions. Subsequently, various characteristics are extracted and used to train a classifier to predict the lesion's label, i.e., whether it is melanoma or non-melanoma. The success of each step in this process relies on the effectiveness of the preceding ones, underscoring the significance of an integrated and effective approach (Eltayef, 2017). This chapter presents an innovative approach to automatic skin cancer detection using medical image processing. The primary emphasis is on the development and evaluation of sophisticated medical expert systems capable of accurately segmenting and classifying skin cancer tumors. The proposed algorithm integrates image quality enhancement, tumor segmentation, and melanoma classification. Throughout the chapter, we explore preprocessing techniques, segmentation methods, and the utilization of a classifier. The ultimate goal of this research is to support medical specialists in achieving prompt and precise tumor detection, leading to advancements in skin cancer diagnosis and improved patient outcomes.

8.2 Methodology

The main objective of this study is to develop an automated skin cancer diagnosis system to improve early detection of skin melanomas. The methodology involves a systematic approach to accurately classify skin cancer tumors as either benign or malignant.

The first step in the methodology is to preprocess the skin cancer images. This involves applying various image enhancement techniques such as noise reduction, contrast enhancement, and normalization to improve the overall image quality and ensure consistency in subsequent analyses.

Following image preprocessing, the next crucial step is skin lesion segmentation. This is achieved by utilizing advanced deep learning methods, specifically W-net and ψ-net, which are designed to effectively delineate the tumor regions from the surrounding healthy skin tissue. Accurate segmentation is essential for precise classification.

Once the skin lesions are successfully segmented, the final step involves skin lesion classification. For this purpose, the study employs the Inception-Resnet convolutional neural network (CNN). The Inception-Resnet CNN has demonstrated remarkable performance in various image classification tasks, making it a suitable choice for classifying the segmented skin lesions as either melanoma or non-melanoma.

The proposed framework aims to create a robust and efficient automated system capable of assisting medical professionals in diagnosing skin cancer accurately and at an early

stage. By achieving this, the study endeavors to contribute to the improvement of skin cancer diagnostics, potentially reducing diagnostic errors and enhancing patient outcomes. Figure 8.1 depicts the melanoma detection strategy.

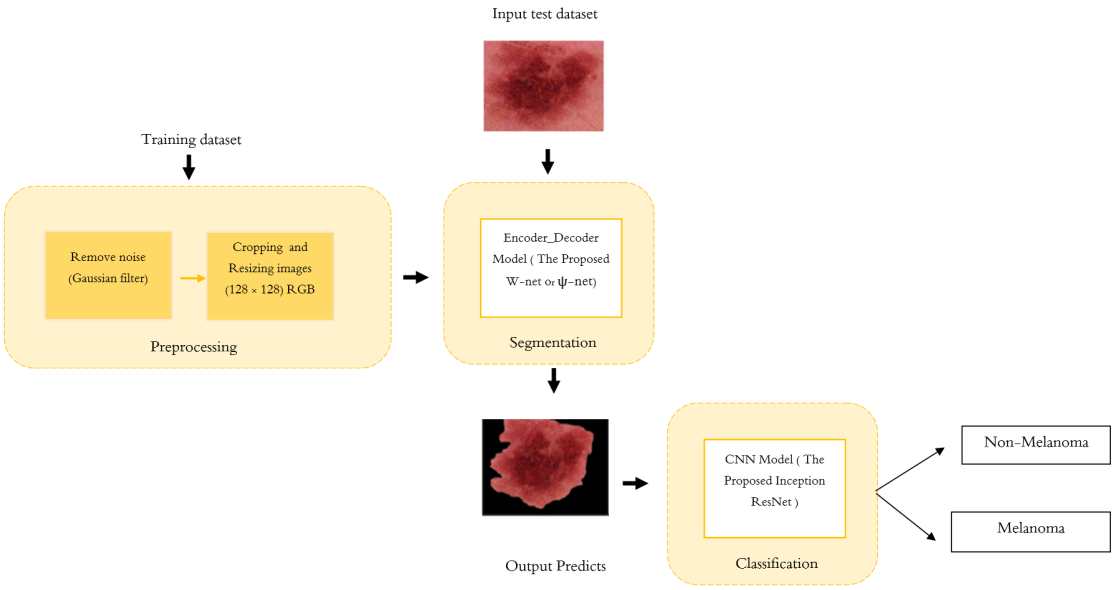


Figure 8.1: Overview of the proposed approach for skin lesions Segmentation and classification.

8.3 Implementation details and Hardware specifications.

For our project, we utilized Python as the primary programming language. The chosen deep learning framework was the Keras interface on top of TensorFlow. To support our development, we incorporated several notable libraries, including Scikit-learn for general-purpose machine learning tasks and PIL for image processing. To train and test our networks, we leveraged Kaggle Notebooks, an online service that offers cloud-based computing. During each session, Kaggle Notebooks reserved a GPU for our use, allowing us to take advantage of accelerated computations. All model run-time testing was conducted within a single session. The hardware specifications we utilized during our project are as follows:

- **CPU**: Intel(R) Xeon(R) CPU @ 2.30GHz.
- **RAM**: 12 GB.
- **GPU**: NVIDIA TESLA P100 GPUs.

  These hardware resources provided the necessary computational power for our deep learning tasks and contributed to efficient model training and evaluation.

8.4 Dermoscopy Image Preprocessing

The clarity of skin lesion images can be degraded by optical lenses used in digital cameras, making visual inspection challenging for identifying malignancies due to the complexity of digital images. Therefore, to assist clinicians in correctly diagnosing skin lesions, effective image-processing techniques are required. Image pre-processing is a crucial step in the detection process as it aims to remove noise and enhance the quality of the original image. This step is essential to minimize the impact of irregularities in the image background that could affect the final diagnosis. The main goal of this pre-processing stage is to improve the quality of melanoma images by eliminating irrelevant and surplus components from the image's background, facilitating further analysis.
Dermoscopic images often contain various types of noise, such as hairs and reflection artifacts, which can cover a significant portion of the lesion regions. The presence of these artifacts can lead to incorrect segmentation of pigmented lesions and compromise the accuracy of the diagnosis. Thus, using appropriate pre-processing techniques is crucial to improve the system's accuracy.
In this study, a Gaussian filter is employed to reduce artifacts from the images, such as fog, hairs, marks, and stains, thereby enhancing image quality. Additionally, an automatic image cropping process is used to extract the region of interest (skin lesion) from each picture and remove unnecessary or redundant portions. The image cropping process is performed in four stages. Firstly, a binary image is computed using the thresholding approach. Secondly, the Canny method is utilized to reduce data in the image and filter out unnecessary information while preserving structural elements. Subsequently, a bounding rectangle is defined using the points acquired through the Canny method, and the original image is cropped accordingly. Furthermore, the image scaling technique is applied to resize the images to a standardized width and height. This step is essential because skin cancer images may come from various sources and have different sizes. Resizing

the images serves two purposes: to improve the performance of the model and to reduce computation time by minimizing the number of pixels to be analyzed.
These pre-processing phases play a critical role in enhancing image quality, facilitating accurate skin lesion segmentation, and ultimately improving the accuracy of the skin cancer diagnosis system. Figure 8.2 shows an example of a preprocessing task.

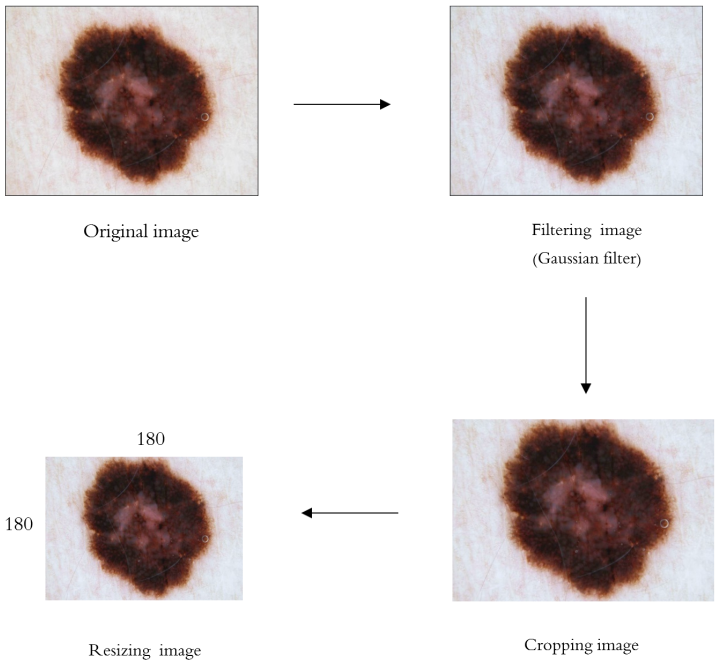


Figure 8.2: Example of a preprocessing task.

8.5 Skin Lesion Segmentation

Image segmentation is a widely used technique in various image processing applications, including object identification, border estimation, image compression, editing, and data visualization. In the context of skin lesion analysis, the main objective of image segmentation is to accurately differentiate the homogenous lesions from the surrounding healthy skin.The segmentation process is critical for effectively analyzing dermoscopy images as it significantly influences the accuracy of subsequent analysis and classification. However, achieving precise segmentation in dermoscopy images is challenging due to the considerable variations in size, shape, and color of the lesions. Additionally, there is often a lack of contrast between the lesions and the surrounding healthy skin.
To address these challenges, automatic segmentation approaches were implemented in

this study to extract the entire lesion area. The ultimate goal is to segment the enhanced skin image and separate the tumor from the background (healthy skin). Various techniques for segmentation exist, and for this research, two segmentation algorithms, namely W-net and ψ-net (as mentioned in the previous chapter), were adopted to enhance skin lesion segmentation.The proposed segmentation framework for melanoma entails using these advanced algorithms to accurately delineate the melanoma regions from the surrounding healthy skin. Figure 8.3 provides an illustration of the proposed segmentation process for melanoma lesions.

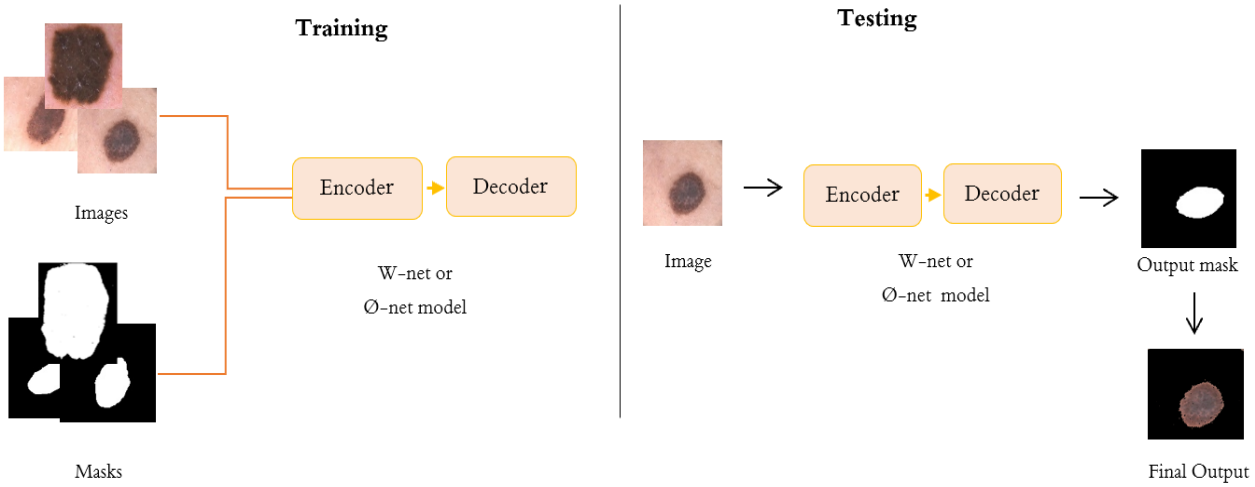


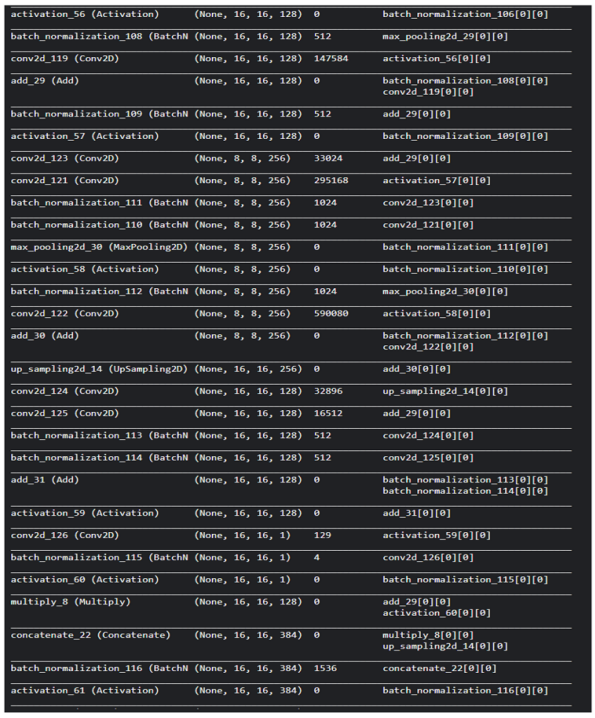
Figure 8.3: A segmentation framework of melanoma.

8.5.1 The W-net Hyper-parameters

The only hyperparameters modified from the original U-Net were batch normalization, the optimizer, and the loss function. For training of the W-Net segmentation deep learning model, the learning rate is set to 0.0001, the epoch number of 60 is adjusted accordingly to make the training steps approximately the same, the batch size is set to 8, and the Adam is used for the optimization of the network. All of the models were also trained using binary cross-entropy as a loss function to determine whether performance is truly enhanced with a loss function.Table 8.1 is a detailed list of the experimental training parameters. A summary of the W-Net architecture during the training phase is shown in Figure 8.4.

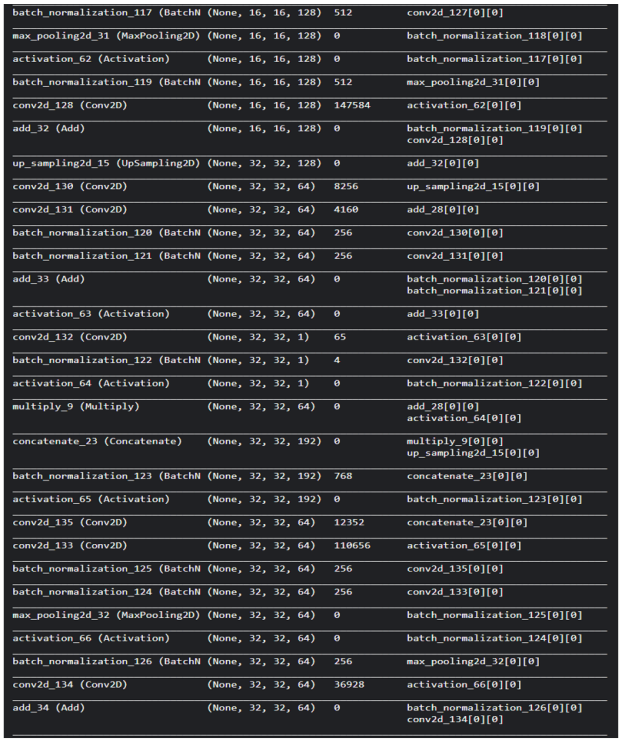
Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[(None, 128, 128, 3)	0	
conv2d_111 (Conv2D)	(None, 128, 128, 16)	64	input_3[0][0]
batch_normalization_94 (BatchNo	(None, 128, 128, 3)	12	input_3[0][0]
batch_normalization_95 (BatchNo	(None, 128, 128, 16)	64	conv2d_111[0][0]
activation_50 (Activation)	(None, 128, 128, 3)	0	batch_normalization_94[0][0]
max_pooling2d_26 (MaxPooling2D)	(None, 128, 128, 16)	0	batch_normalization_95[0][0]
conv2d_110 (Conv2D)	(None, 128, 128, 16)	448	activation_50[0][0]
batch_normalization_96 (BatchNo	(None, 128, 128, 16)	64	max_pooling2d_26[0][0]
add_26 (Add)	(None, 128, 128, 16)	0	conv2d_110[0][0] batch_normalization_96[0][0]
batch_normalization_97 (BatchNo	(None, 128, 128, 16)	64	add_26[0][0]
activation_51 (Activation)	(None, 128, 128, 16)	0	batch_normalization_97[0][0]
conv2d_114 (Conv2D)	(None, 64, 64, 32)	544	add_26[0][0]
conv2d_112 (Conv2D)	(None, 64, 64, 32)	4640	activation_51[0][0]
batch_normalization_99 (BatchNo	(None, 64, 64, 32)	128	conv2d_114[0][0]
batch_normalization_98 (BatchNo	(None, 64, 64, 32)	128	conv2d_112[0][0]
max_pooling2d_27 (MaxPooling2D)	(None, 64, 64, 32)	0	batch_normalization_99[0][0]
activation_52 (Activation)	(None, 64, 64, 32)	0	batch_normalization_98[0][0]
batch_normalization_100 (BatchN	(None, 64, 64, 32)	128	max_pooling2d_27[0][0]
conv2d_113 (Conv2D)	(None, 64, 64, 32)	9248	activation_52[0][0]
add_27 (Add)	(None, 64, 64, 32)	0	batch_normalization_100[0][0] conv2d_113[0][0]
batch_normalization_101 (BatchN	(None, 64, 64, 32)	128	add_27[0][0]
activation_53 (Activation)	(None, 64, 64, 32)	0	batch_normalization_101[0][0]
conv2d_117 (Conv2D)	(None, 32, 32, 64)	2112	add_27[0][0]
conv2d_115 (Conv2D)	(None, 32, 32, 64)	18496	activation_53[0][0]
batch_normalization_103 (BatchN	(None, 32, 32, 64)	256	conv2d_117[0][0]
batch_normalization_102 (BatchN	(None, 32, 32, 64)	256	conv2d_115[0][0]
max_pooling2d_28 (MaxPooling2D)	(None, 32, 32, 64)	0	batch_normalization_103[0][0]
activation_54 (Activation)	(None, 32, 32, 64)	0	batch_normalization_102[0][0]
batch_normalization_104 (BatchN	(None, 32, 32, 64)	256	max_pooling2d_28[0][0]

(1)



activation_56 (Activation)	(None, 16, 16, 128)	0	batch_normalization_106[0][0]
batch_normalization_108 (BatchN	(None, 16, 16, 128)	512	max_pooling2d_29[0][0]
conv2d_119 (Conv2D)	(None, 16, 16, 128)	147584	activation_56[0][0]
add_29 (Add)	(None, 16, 16, 128)	0	batch_normalization_108[0][0] conv2d_119[0][0]
batch_normalization_109 (BatchN	(None, 16, 16, 128)	512	add_29[0][0]
activation_57 (Activation)	(None, 16, 16, 128)	0	batch_normalization_109[0][0]
conv2d_123 (Conv2D)	(None, 8, 8, 256)	33024	add_29[0][0]
conv2d_121 (Conv2D)	(None, 8, 8, 256)	295168	activation_57[0][0]
batch_normalization_111 (BatchN	(None, 8, 8, 256)	1024	conv2d_123[0][0]
batch_normalization_110 (BatchN	(None, 8, 8, 256)	1024	conv2d_121[0][0]
max_pooling2d_30 (MaxPooling2D)	(None, 8, 8, 256)	0	batch_normalization_111[0][0]
activation_58 (Activation)	(None, 8, 8, 256)	0	batch_normalization_110[0][0]
batch_normalization_112 (BatchN	(None, 8, 8, 256)	1024	max_pooling2d_30[0][0]
conv2d_122 (Conv2D)	(None, 8, 8, 256)	590080	activation_58[0][0]
add_30 (Add)	(None, 8, 8, 256)	0	batch_normalization_112[0][0] conv2d_122[0][0]
up_sampling2d_14 (UpSampling2D)	(None, 16, 16, 256)	0	add_30[0][0]
conv2d_124 (Conv2D)	(None, 16, 16, 128)	32896	up_sampling2d_14[0][0]
conv2d_125 (Conv2D)	(None, 16, 16, 128)	16512	add_29[0][0]
batch_normalization_113 (BatchN	(None, 16, 16, 128)	512	conv2d_124[0][0]
batch_normalization_114 (BatchN	(None, 16, 16, 128)	512	conv2d_125[0][0]
add_31 (Add)	(None, 16, 16, 128)	0	batch_normalization_113[0][0] batch_normalization_114[0][0]
activation_59 (Activation)	(None, 16, 16, 128)	0	add_31[0][0]
conv2d_126 (Conv2D)	(None, 16, 16, 1)	129	activation_59[0][0]
batch_normalization_115 (BatchN	(None, 16, 16, 1)	4	conv2d_126[0][0]
activation_60 (Activation)	(None, 16, 16, 1)	0	batch_normalization_115[0][0]
multiply_8 (Multiply)	(None, 16, 16, 128)	0	add_29[0][0] activation_60[0][0]
concatenate_22 (Concatenate)	(None, 16, 16, 384)	0	multiply_8[0][0] up_sampling2d_14[0][0]
batch_normalization_116 (BatchN	(None, 16, 16, 384)	1536	concatenate_22[0][0]
activation_61 (Activation)	(None, 16, 16, 384)	0	batch_normalization_116[0][0]

(2)



batch_normalization_117 (BatchN	(None, 16, 16, 128)	512	conv2d_127[0][0]
max_pooling2d_31 (MaxPooling2D)	(None, 16, 16, 128)	0	batch_normalization_118[0][0]
activation_62 (Activation)	(None, 16, 16, 128)	0	batch_normalization_117[0][0]
batch_normalization_119 (BatchN	(None, 16, 16, 128)	512	max_pooling2d_31[0][0]
conv2d_128 (Conv2D)	(None, 16, 16, 128)	147584	activation_62[0][0]
add_32 (Add)	(None, 16, 16, 128)	0	batch_normalization_119[0][0] conv2d_128[0][0]
up_sampling2d_15 (UpSampling2D)	(None, 32, 32, 128)	0	add_32[0][0]
conv2d_130 (Conv2D)	(None, 32, 32, 64)	8256	up_sampling2d_15[0][0]
conv2d_131 (Conv2D)	(None, 32, 32, 64)	4160	add_28[0][0]
batch_normalization_120 (BatchN	(None, 32, 32, 64)	256	conv2d_130[0][0]
batch_normalization_121 (BatchN	(None, 32, 32, 64)	256	conv2d_131[0][0]
add_33 (Add)	(None, 32, 32, 64)	0	batch_normalization_120[0][0] batch_normalization_121[0][0]
activation_63 (Activation)	(None, 32, 32, 64)	0	add_33[0][0]
conv2d_132 (Conv2D)	(None, 32, 32, 1)	65	activation_63[0][0]
batch_normalization_122 (BatchN	(None, 32, 32, 1)	4	conv2d_132[0][0]
activation_64 (Activation)	(None, 32, 32, 1)	0	batch_normalization_122[0][0]
multiply_9 (Multiply)	(None, 32, 32, 64)	0	add_28[0][0] activation_64[0][0]
concatenate_23 (Concatenate)	(None, 32, 32, 192)	0	multiply_9[0][0] up_sampling2d_15[0][0]
batch_normalization_123 (BatchN	(None, 32, 32, 192)	768	concatenate_23[0][0]
activation_65 (Activation)	(None, 32, 32, 192)	0	batch_normalization_123[0][0]
conv2d_135 (Conv2D)	(None, 32, 32, 64)	12352	concatenate_23[0][0]
conv2d_133 (Conv2D)	(None, 32, 32, 64)	110656	activation_65[0][0]
batch_normalization_125 (BatchN	(None, 32, 32, 64)	256	conv2d_135[0][0]
batch_normalization_124 (BatchN	(None, 32, 32, 64)	256	conv2d_133[0][0]
max_pooling2d_32 (MaxPooling2D)	(None, 32, 32, 64)	0	batch_normalization_125[0][0]
activation_66 (Activation)	(None, 32, 32, 64)	0	batch_normalization_124[0][0]
batch_normalization_126 (BatchN	(None, 32, 32, 64)	256	max_pooling2d_32[0][0]
conv2d_134 (Conv2D)	(None, 32, 32, 64)	36928	activation_66[0][0]
add_34 (Add)	(None, 32, 32, 64)	0	batch_normalization_126[0][0] conv2d_134[0][0]

(3)

conv2d_150 (Conv2D)	(None, 128, 128, 32)	9248	concatenate_26[0][0]
conv2d_151 (Conv2D)	(None, 128, 128, 32)	9248	conv2d_150[0][0]
max_pooling2d_36 (MaxPooling2D)	(None, 64, 64, 32)	0	conv2d_151[0][0]
concatenate_27 (Concatenate)	(None, 64, 64, 64)	0	max_pooling2d_36[0][0] add_36[0][0]
conv2d_152 (Conv2D)	(None, 64, 64, 64)	36928	concatenate_27[0][0]
conv2d_153 (Conv2D)	(None, 64, 64, 64)	36928	conv2d_152[0][0]
max_pooling2d_37 (MaxPooling2D)	(None, 32, 32, 64)	0	conv2d_153[0][0]
concatenate_28 (Concatenate)	(None, 32, 32, 128)	0	max_pooling2d_37[0][0] add_34[0][0]
conv2d_154 (Conv2D)	(None, 32, 32, 128)	147584	concatenate_28[0][0]
conv2d_155 (Conv2D)	(None, 32, 32, 128)	147584	conv2d_154[0][0]
max_pooling2d_38 (MaxPooling2D)	(None, 16, 16, 128)	0	conv2d_155[0][0]
concatenate_29 (Concatenate)	(None, 16, 16, 256)	0	max_pooling2d_38[0][0] add_32[0][0]
conv2d_156 (Conv2D)	(None, 16, 16, 256)	590080	concatenate_29[0][0]
conv2d_157 (Conv2D)	(None, 16, 16, 256)	590080	conv2d_156[0][0]
up_sampling2d_18 (UpSampling2D)	(None, 32, 32, 256)	0	conv2d_157[0][0]
concatenate_30 (Concatenate)	(None, 32, 32, 384)	0	up_sampling2d_18[0][0] conv2d_155[0][0]
conv2d_158 (Conv2D)	(None, 32, 32, 128)	442496	concatenate_30[0][0]
conv2d_159 (Conv2D)	(None, 32, 32, 128)	147584	conv2d_158[0][0]
up_sampling2d_19 (UpSampling2D)	(None, 64, 64, 128)	0	conv2d_159[0][0]
concatenate_31 (Concatenate)	(None, 64, 64, 192)	0	up_sampling2d_19[0][0] conv2d_153[0][0]
conv2d_160 (Conv2D)	(None, 64, 64, 64)	110656	concatenate_31[0][0]
conv2d_161 (Conv2D)	(None, 64, 64, 64)	36928	conv2d_160[0][0]
up_sampling2d_20 (UpSampling2D)	(None, 128, 128, 64)	0	conv2d_161[0][0]
concatenate_32 (Concatenate)	(None, 128, 128, 96)	0	up_sampling2d_20[0][0] conv2d_151[0][0]
conv2d_162 (Conv2D)	(None, 128, 128, 32)	27680	concatenate_32[0][0]
conv2d_163 (Conv2D)	(None, 128, 128, 32)	9248	conv2d_162[0][0]
conv2d_164 (Conv2D)	(None, 128, 128, 1)	33	conv2d_163[0][0]

Total params: 4,493,165
Trainable params: 4,485,913
Non-trainable params: 7,252

(4)

Figure 8.4: The Summary of W-net.

Table 8.1: Settings for training parameters

Type	Setting
Optimizer	Adam optimization
Loss	binary cross-entropy
Learning rate	0.0001
Batch size	8
Number of iterations (Epoch)	60
Activation function	ReLU, Sigmoid

8.5.2 The ψ-net Hyper-parameters

During the training of the ψ-net segmentation deep learning model, several settings are adjusted for optimal performance. The learning rate is set to 0.0001, the batch size is configured as 10, and the number of epochs is modified to ensure balanced training stages. The Adam optimization algorithm is employed to optimize the network's parameters. To evaluate the impact of the loss function on performance, all models are trained using binary cross-entropy as the loss function. A detailed description of the experimental training settings can be found in Table 8.2. Figure 8.5 visually summarizes the architecture of the ψ-net during the training phase, illustrating the evolution and modifications made to the model throughout the training process.

Table 8.2: Settings for training parameters

Type	Setting
Optimizer	Adam optimization
Loss	binary cross-entropy
Learning rate	0.0001
Batch size	10
Number of iterations (Epoch)	120
Bias init	0
Activation function	ReLU, Sigmoid

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 128, 128, 3)	0	
batch_normalization_299 (BatchN	(None, 128, 128, 3)	12	input_2[0][0]
activation_163 (Activation)	(None, 128, 128, 3)	0	batch_normalization_299[0][0]
conv2d_221 (Conv2D)	(None, 128, 128, 16)	448	input_2[0][0]
conv2d_213 (Conv2D)	(None, 128, 128, 16)	64	activation_163[0][0]
batch_normalization_310 (BatchN	(None, 128, 128, 16)	64	conv2d_221[0][0]
batch_normalization_300 (BatchN	(None, 128, 128, 16)	64	conv2d_213[0][0]
activation_169 (Activation)	(None, 128, 128, 16)	0	batch_normalization_310[0][0]
conv2d_219 (Conv2D)	(None, 128, 128, 16)	272	batch_normalization_300[0][0]
conv2d_222 (Conv2D)	(None, 128, 128, 16)	2320	activation_169[0][0]
batch_normalization_309 (BatchN	(None, 128, 128, 16)	64	conv2d_219[0][0]
batch_normalization_311 (BatchN	(None, 128, 128, 16)	64	conv2d_222[0][0]
add_28 (Add)	(None, 128, 128, 16)	0	conv2d_219[0][0] batch_normalization_309[0][0]
activation_170 (Activation)	(None, 128, 128, 16)	0	batch_normalization_311[0][0]
conv2d_223 (Conv2D)	(None, 64, 64, 16)	272	activation_170[0][0]
conv2d_220 (Conv2D)	(None, 64, 64, 16)	272	add_28[0][0]
concatenate_54 (Concatenate)	(None, 64, 64, 32)	0	conv2d_223[0][0] conv2d_220[0][0]
batch_normalization_312 (BatchN	(None, 64, 64, 32)	128	concatenate_54[0][0]
conv2d_232 (Conv2D)	(None, 64, 64, 32)	4640	conv2d_223[0][0]
activation_171 (Activation)	(None, 64, 64, 32)	0	batch_normalization_312[0][0]
batch_normalization_323 (BatchN	(None, 64, 64, 32)	128	conv2d_232[0][0]
conv2d_224 (Conv2D)	(None, 64, 64, 32)	1056	activation_171[0][0]
activation_177 (Activation)	(None, 64, 64, 32)	0	batch_normalization_323[0][0]
batch_normalization_313 (BatchN	(None, 64, 64, 32)	128	conv2d_224[0][0]
conv2d_233 (Conv2D)	(None, 64, 64, 32)	9248	activation_177[0][0]
conv2d_230 (Conv2D)	(None, 64, 64, 32)	1056	batch_normalization_313[0][0]
batch_normalization_324 (BatchN	(None, 64, 64, 32)	128	conv2d_233[0][0]
batch_normalization_322 (BatchN	(None, 64, 64, 32)	128	conv2d_230[0][0]
activation_178 (Activation)	(None, 64, 64, 32)	0	batch_normalization_324[0][0]
add_29 (Add)	(None, 64, 64, 32)	0	conv2d_230[0][0] batch_normalization_322[0][0]
conv2d_234 (Conv2D)	(None, 32, 32, 32)	1056	activation_178[0][0]
conv2d_231 (Conv2D)	(None, 32, 32, 32)	1056	add_29[0][0]
concatenate_56 (Concatenate)	(None, 32, 32, 64)	0	conv2d_234[0][0] conv2d_231[0][0]
batch_normalization_325 (BatchN	(None, 32, 32, 64)	256	concatenate_56[0][0]
conv2d_243 (Conv2D)	(None, 32, 32, 64)	18496	conv2d_234[0][0]
activation_179 (Activation)	(None, 32, 32, 64)	0	batch_normalization_325[0][0]
batch_normalization_336 (BatchN	(None, 32, 32, 64)	256	conv2d_243[0][0]
conv2d_235 (Conv2D)	(None, 32, 32, 64)	4160	activation_179[0][0]
activation_185 (Activation)	(None, 32, 32, 64)	0	batch_normalization_336[0][0]
batch_normalization_326 (BatchN	(None, 32, 32, 64)	256	conv2d_235[0][0]
conv2d_244 (Conv2D)	(None, 32, 32, 64)	36928	activation_185[0][0]
conv2d_241 (Conv2D)	(None, 32, 32, 64)	4160	batch_normalization_326[0][0]
batch_normalization_337 (BatchN	(None, 32, 32, 64)	256	conv2d_244[0][0]
batch_normalization_335 (BatchN	(None, 32, 32, 64)	256	conv2d_241[0][0]
conv2d_242 (Conv2D)	(None, 16, 16, 64)	4160	add_30[0][0]
concatenate_58 (Concatenate)	(None, 16, 16, 128)	0	conv2d_245[0][0] conv2d_242[0][0]
batch_normalization_338 (BatchN	(None, 16, 16, 128)	512	concatenate_58[0][0]
conv2d_254 (Conv2D)	(None, 16, 16, 128)	73856	conv2d_245[0][0]
activation_187 (Activation)	(None, 16, 16, 128)	0	batch_normalization_338[0][0]
batch_normalization_349 (BatchN	(None, 16, 16, 128)	512	conv2d_254[0][0]
conv2d_246 (Conv2D)	(None, 16, 16, 128)	16512	activation_187[0][0]
activation_193 (Activation)	(None, 16, 16, 128)	0	batch_normalization_349[0][0]
batch_normalization_339 (BatchN	(None, 16, 16, 128)	512	conv2d_246[0][0]
conv2d_255 (Conv2D)	(None, 16, 16, 128)	147584	activation_193[0][0]
conv2d_252 (Conv2D)	(None, 16, 16, 128)	16512	batch_normalization_339[0][0]
batch_normalization_350 (BatchN	(None, 16, 16, 128)	512	conv2d_255[0][0]
batch_normalization_348 (BatchN	(None, 16, 16, 128)	512	conv2d_252[0][0]
activation_194 (Activation)	(None, 16, 16, 128)	0	batch_normalization_350[0][0]
add_31 (Add)	(None, 16, 16, 128)	0	conv2d_252[0][0] batch_normalization_348[0][0]
conv2d_256 (Conv2D)	(None, 8, 8, 128)	16512	activation_194[0][0]
conv2d_253 (Conv2D)	(None, 8, 8, 128)	16512	add_31[0][0]
concatenate_60 (Concatenate)	(None, 8, 8, 256)	0	conv2d_256[0][0] conv2d_253[0][0]
batch_normalization_492 (BatchN	(None, 8, 8, 256)	1024	concatenate_60[0][0]
activation_267 (Activation)	(None, 8, 8, 256)	0	batch_normalization_492[0][0]
conv2d_344 (Conv2D)	(None, 8, 8, 256)	65792	activation_267[0][0]
batch_normalization_493 (BatchN	(None, 8, 8, 256)	1024	conv2d_344[0][0]
conv2d_350 (Conv2D)	(None, 8, 8, 256)	65792	batch_normalization_493[0][0]
batch_normalization_502 (BatchN	(None, 8, 8, 256)	1024	conv2d_350[0][0]
add_44 (Add)	(None, 8, 8, 256)	0	conv2d_350[0][0] batch_normalization_502[0][0]
batch_normalization_503 (BatchN	(None, 8, 8, 256)	1024	add_44[0][0]
batch_normalization_504 (BatchN	(None, 8, 8, 256)	1024	batch_normalization_503[0][0]
activation_273 (Activation)	(None, 8, 8, 256)	0	batch_normalization_504[0][0]
conv2d_351 (Conv2D)	(None, 8, 8, 256)	65792	activation_273[0][0]
batch_normalization_505 (BatchN	(None, 8, 8, 256)	1024	conv2d_351[0][0]
conv2d_357 (Conv2D)	(None, 8, 8, 256)	65792	batch_normalization_505[0][0]
batch_normalization_514 (BatchN	(None, 8, 8, 256)	1024	conv2d_357[0][0]
add_45 (Add)	(None, 8, 8, 256)	0	conv2d_357[0][0] batch_normalization_514[0][0]
concatenate_81 (Concatenate)	(None, 8, 8, 512)	0	concatenate_60[0][0] add_45[0][0]
batch_normalization_515 (BatchN	(None, 8, 8, 512)	2048	concatenate_81[0][0]
activation_279 (Activation)	(None, 8, 8, 512)	0	batch_normalization_515[0][0]
conv2d_358 (Conv2D)	(None, 8, 8, 256)	131328	activation_279[0][0]
batch_normalization_516 (BatchN	(None, 8, 8, 256)	1024	conv2d_358[0][0]
conv2d_364 (Conv2D)	(None, 8, 8, 256)	65792	batch_normalization_516[0][0]
batch_normalization_525 (BatchN	(None, 8, 8, 256)	1024	conv2d_364[0][0]
add_46 (Add)	(None, 8, 8, 256)	0	conv2d_364[0][0] batch_normalization_525[0][0]
batch_normalization_526 (BatchN	(None, 8, 8, 256)	1024	add_46[0][0]
batch_normalization_527 (BatchN	(None, 8, 8, 256)	1024	batch_normalization_526[0][0]
activation_285 (Activation)	(None, 8, 8, 256)	0	batch_normalization_527[0][0]
conv2d_365 (Conv2D)	(None, 8, 8, 256)	65792	activation_285[0][0]
batch_normalization_528 (BatchN	(None, 8, 8, 256)	1024	conv2d_365[0][0]

activation_301 (Activation)	(None, 32, 32, 192)	0	batch_normalization_556[0][0]
conv2d_389 (Conv2D)	(None, 32, 32, 64)	12352	activation_301[0][0]
batch_normalization_557 (BatchN	(None, 32, 32, 64)	256	conv2d_389[0][0]
conv2d_395 (Conv2D)	(None, 32, 32, 64)	4160	batch_normalization_557[0][0]
batch_normalization_566 (BatchN	(None, 32, 32, 64)	256	conv2d_395[0][0]
add_51 (Add)	(None, 32, 32, 64)	0	conv2d_395[0][0] batch_normalization_566[0][0]
lambda_34 (Lambda)	(None, 32, 32, 1)	0	add_51[0][0]
lambda_35 (Lambda)	(None, 32, 32, 1)	0	add_51[0][0]
conv2d_396 (Conv2D)	(None, 32, 32, 1)	49	lambda_34[0][0]
conv2d_397 (Conv2D)	(None, 32, 32, 1)	49	lambda_35[0][0]
concatenate_94 (Concatenate)	(None, 32, 32, 2)	0	conv2d_396[0][0] conv2d_397[0][0]
conv2d_398 (Conv2D)	(None, 32, 32, 1)	98	concatenate_94[0][0]
multiply_11 (Multiply)	(None, 32, 32, 64)	0	add_51[0][0] conv2d_398[0][0]
up_sampling2d_9 (UpSampling2D)	(None, 64, 64, 64)	0	multiply_11[0][0]
lambda_36 (Lambda)	(None, 64, 64, 1)	0	up_sampling2d_9[0][0]
lambda_37 (Lambda)	(None, 64, 64, 1)	0	up_sampling2d_9[0][0]
lambda_38 (Lambda)	(None, 64, 64, 1)	0	concatenate_54[0][0]
lambda_39 (Lambda)	(None, 64, 64, 1)	0	concatenate_54[0][0]
concatenate_95 (Concatenate)	(None, 64, 64, 2)	0	lambda_36[0][0] lambda_37[0][0]
concatenate_96 (Concatenate)	(None, 64, 64, 2)	0	lambda_38[0][0] lambda_39[0][0]
conv2d_399 (Conv2D)	(None, 64, 64, 1)	98	concatenate_95[0][0]
conv2d_400 (Conv2D)	(None, 64, 64, 1)	98	concatenate_96[0][0]
batch_normalization_567 (BatchN	(None, 64, 64, 1)	4	conv2d_399[0][0]
batch_normalization_568 (BatchN	(None, 64, 64, 1)	4	conv2d_400[0][0]
add_52 (Add)	(None, 64, 64, 1)	0	batch_normalization_567[0][0] batch_normalization_568[0][0]
activation_307 (Activation)	(None, 64, 64, 1)	0	add_52[0][0]
conv2d_401 (Conv2D)	(None, 64, 64, 1)	2	activation_307[0][0]
batch_normalization_569 (BatchN	(None, 64, 64, 1)	4	conv2d_401[0][0]
activation_308 (Activation)	(None, 64, 64, 1)	0	batch_normalization_569[0][0]
multiply_12 (Multiply)	(None, 64, 64, 32)	0	concatenate_54[0][0] activation_308[0][0]
concatenate_97 (Concatenate)	(None, 64, 64, 96)	0	multiply_12[0][0] up_sampling2d_9[0][0]
batch_normalization_570 (BatchN	(None, 64, 64, 96)	384	concatenate_97[0][0]
activation_309 (Activation)	(None, 64, 64, 96)	0	batch_normalization_570[0][0]
conv2d_402 (Conv2D)	(None, 64, 64, 32)	3104	activation_309[0][0]
batch_normalization_571 (BatchN	(None, 64, 64, 32)	128	conv2d_402[0][0]
conv2d_408 (Conv2D)	(None, 64, 64, 32)	1056	batch_normalization_571[0][0]
batch_normalization_580 (BatchN	(None, 64, 64, 32)	128	conv2d_408[0][0]
add_53 (Add)	(None, 64, 64, 32)	0	conv2d_408[0][0] batch_normalization_580[0][0]
lambda_40 (Lambda)	(None, 64, 64, 1)	0	add_53[0][0]
lambda_41 (Lambda)	(None, 64, 64, 1)	0	add_53[0][0]
conv2d_409 (Conv2D)	(None, 64, 64, 1)	49	lambda_40[0][0]
conv2d_410 (Conv2D)	(None, 64, 64, 1)	49	lambda_41[0][0]
concatenate_99 (Concatenate)	(None, 64, 64, 2)	0	conv2d_409[0][0] conv2d_410[0][0]
conv2d_411 (Conv2D)	(None, 64, 64, 1)	98	concatenate_99[0][0]
multiply_13 (Multiply)	(None, 64, 64, 32)	0	add_53[0][0] conv2d_411[0][0]

up_sampling2d_10 (UpSampling2D)	(None, 128, 128, 32)	0	multiply_13[0][0]
lambda_42 (Lambda)	(None, 128, 128, 1)	0	up_sampling2d_10[0][0]
lambda_43 (Lambda)	(None, 128, 128, 1)	0	up_sampling2d_10[0][0]
lambda_44 (Lambda)	(None, 128, 128, 1)	0	add_28[0][0]
lambda_45 (Lambda)	(None, 128, 128, 1)	0	add_28[0][0]
concatenate_100 (Concatenate)	(None, 128, 128, 2)	0	lambda_42[0][0] lambda_43[0][0]
concatenate_101 (Concatenate)	(None, 128, 128, 2)	0	lambda_44[0][0] lambda_45[0][0]
conv2d_412 (Conv2D)	(None, 128, 128, 1)	98	concatenate_100[0][0]
conv2d_413 (Conv2D)	(None, 128, 128, 1)	98	concatenate_101[0][0]
batch_normalization_581 (BatchN	(None, 128, 128, 1)	4	conv2d_412[0][0]
batch_normalization_582 (BatchN	(None, 128, 128, 1)	4	conv2d_413[0][0]
add_54 (Add)	(None, 128, 128, 1)	0	batch_normalization_581[0][0] batch_normalization_582[0][0]
activation_315 (Activation)	(None, 128, 128, 1)	0	add_54[0][0]
conv2d_414 (Conv2D)	(None, 128, 128, 1)	2	activation_315[0][0]
batch_normalization_583 (BatchN	(None, 128, 128, 1)	4	conv2d_414[0][0]
activation_316 (Activation)	(None, 128, 128, 1)	0	batch_normalization_583[0][0]
multiply_14 (Multiply)	(None, 128, 128, 16)	0	add_28[0][0] activation_316[0][0]
concatenate_102 (Concatenate)	(None, 128, 128, 48)	0	multiply_14[0][0] up_sampling2d_10[0][0]
batch_normalization_584 (BatchN	(None, 128, 128, 48)	192	concatenate_102[0][0]
activation_317 (Activation)	(None, 128, 128, 48)	0	batch_normalization_584[0][0]
conv2d_415 (Conv2D)	(None, 128, 128, 16)	784	activation_317[0][0]
batch_normalization_585 (BatchN	(None, 128, 128, 16)	64	conv2d_415[0][0]
conv2d_421 (Conv2D)	(None, 128, 128, 16)	272	batch_normalization_585[0][0]
batch_normalization_594 (BatchN	(None, 128, 128, 16)	64	conv2d_421[0][0]
add_55 (Add)	(None, 128, 128, 16)	0	conv2d_421[0][0] batch_normalization_594[0][0]
lambda_46 (Lambda)	(None, 128, 128, 1)	0	add_55[0][0]
lambda_47 (Lambda)	(None, 128, 128, 1)	0	add_55[0][0]
conv2d_422 (Conv2D)	(None, 128, 128, 1)	49	lambda_46[0][0]
conv2d_423 (Conv2D)	(None, 128, 128, 1)	49	lambda_47[0][0]
concatenate_104 (Concatenate)	(None, 128, 128, 2)	0	conv2d_422[0][0] conv2d_423[0][0]
up_sampling2d_11 (UpSampling2D)	(None, 128, 128, 128	0	multiply_9[0][0]
up_sampling2d_12 (UpSampling2D)	(None, 128, 128, 64)	0	multiply_11[0][0]
up_sampling2d_13 (UpSampling2D)	(None, 128, 128, 32)	0	multiply_13[0][0]
conv2d_424 (Conv2D)	(None, 128, 128, 1)	98	concatenate_104[0][0]
batch_normalization_595 (BatchN	(None, 128, 128, 128	512	up_sampling2d_11[0][0]
batch_normalization_596 (BatchN	(None, 128, 128, 64)	256	up_sampling2d_12[0][0]
batch_normalization_597 (BatchN	(None, 128, 128, 32)	128	up_sampling2d_13[0][0]
multiply_15 (Multiply)	(None, 128, 128, 16)	0	add_55[0][0] conv2d_424[0][0]
activation_323 (Activation)	(None, 128, 128, 128	0	batch_normalization_595[0][0]
activation_324 (Activation)	(None, 128, 128, 64)	0	batch_normalization_596[0][0]
activation_325 (Activation)	(None, 128, 128, 32)	0	batch_normalization_597[0][0]
concatenate_105 (Concatenate)	(None, 128, 128, 240	0	multiply_15[0][0] activation_323[0][0] activation_324[0][0] activation_325[0][0]
conv2d_425 (Conv2D)	(None, 128, 128, 1)	241	concatenate_105[0][0]

Total params: 2,861,173
Trainable params: 2,846,711
Non-trainable params: 14,462

Figure 8.5: The summary of ψ-net.

8.5.3 Tests comparison and discussion using W-net model

8.5.3.1 Tests comparison on ISIC-2016 Dataset

The W-net model has been successfully trained and achieved a high level of accuracy in segmenting data, as demonstrated in Figure 8.6. The model effectively minimized the loss, stabilized close to zero, and demonstrated significant improvement in accuracy from 82% to over 98%. The results indicate the model's effectiveness in accurately segmenting data while avoiding overfitting, showcasing its ability to learn patterns in the data.

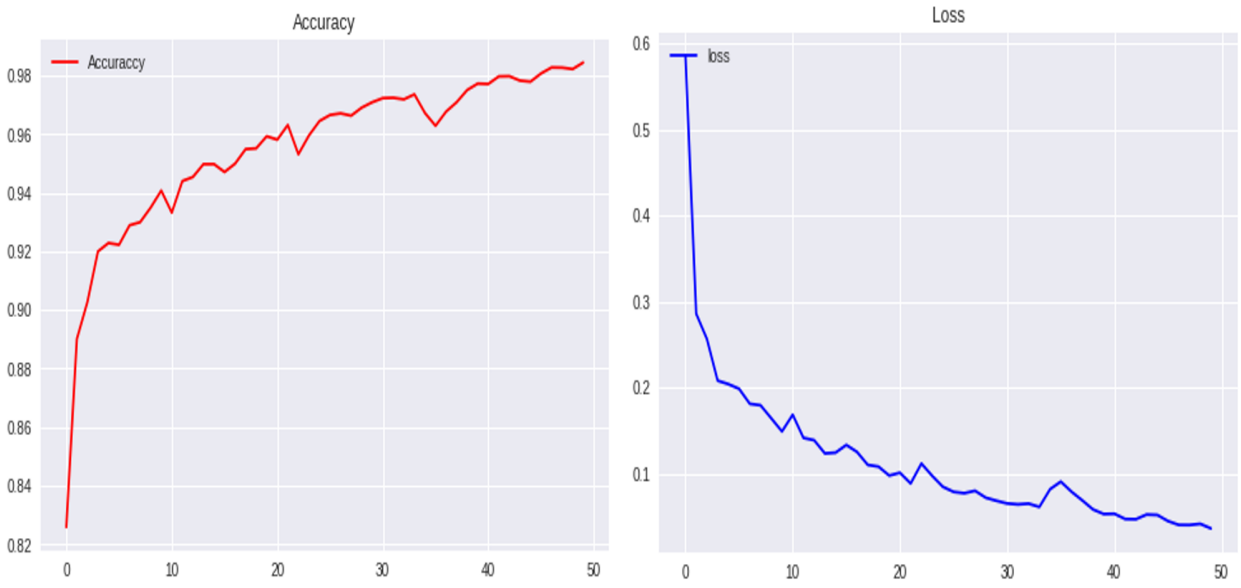


Figure 8.6: On ISIC 2016, the suggested W-net Segmentation model's accuracy and loss performance curves.

Figure 8.7 presents the results of the training process for the W-net network. The graph shows three curves representing the dice coefficient, specificity, and recall performance, which are commonly used metrics for evaluating image segmentation algorithms. As the number of training epochs increases, the values of these metrics decrease and approach more than 95%. This indicates that the network is learning and improving its performance over time, demonstrating its effectiveness in accurately segmenting the skin lesion data. The high values achieved by the metrics signify the model's proficiency in the segmentation task.

Figure 8.7: On ISIC2016, Dice coefficient, specificity and Recall performance curves of the proposed W-net Segmentation model.

The segmentation results of our W-net proposed model on four image samples are shown in Figure 8.8. The proposed strategy produces improved segmentation results that are closer to the truth, proving the effectiveness of the suggested approach.

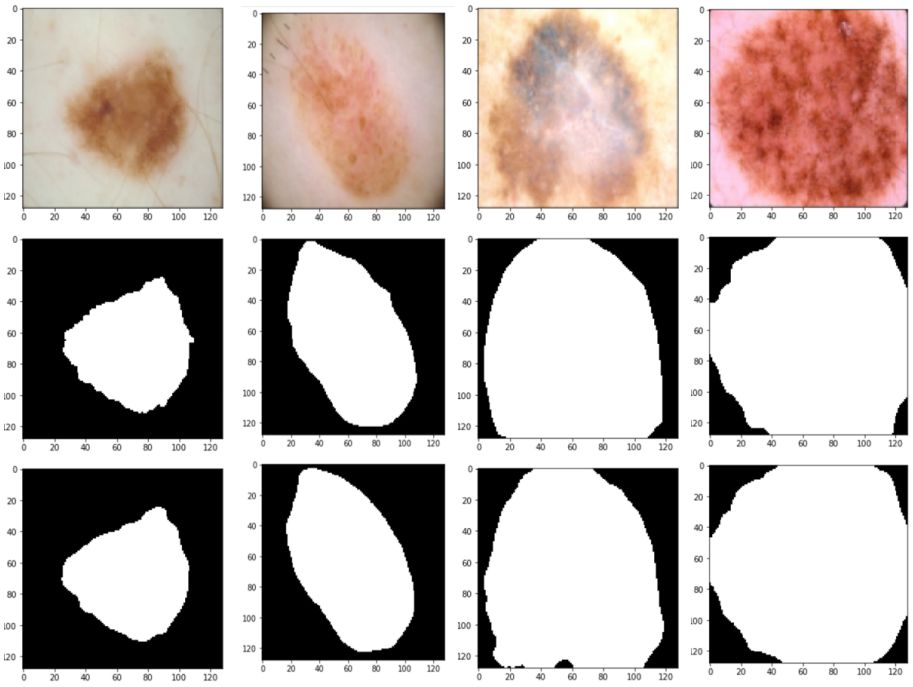


Figure 8.8: The segmentation results of W_net architecture. (1) Skin image (2) Ground truth images and (3) W-net results.

Figure 8.9 displays the segmentation results of the W-Net network. The input images are shown in the first row, the ground truth labels in the second row, and the segmented outputs of W-Net and ResUnet in the third and fourth rows, respectively. W-Net demonstrates satisfactory segmentation performance, even in challenging situations. The comparison between W-Net and ResUnet's segmentation results against the ground truth reveals that W-Net achieves more accurate segmentations. The region borders extracted by W-Net closely match the ground truth, indicating its superior performance compared to ResUnet. The dual encoder-decoder architecture of W-Net is considered more robust for segmentation compared to the single encoder-decoder model of ResUnet. The effectiveness of W-Net is highlighted by its ability to outperform ResUnet in accurately segmenting skin lesions.

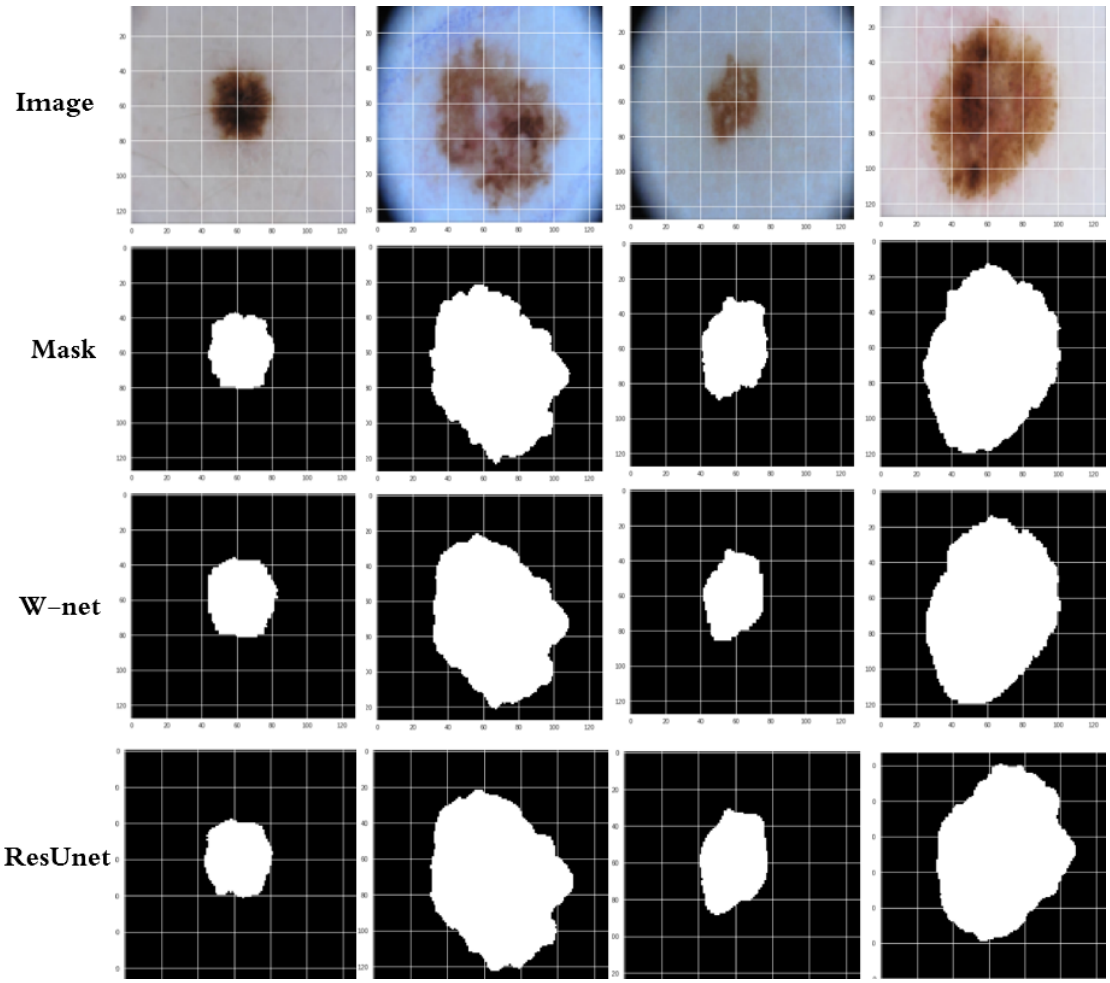


Figure 8.9: The segmentation results of W_net architecture. (1) Skin image (2) Ground truth images and (3) W-net results (4) ResUnet.

Based on the comparison provided in Table 8.3, the proposed network, W-net, outperforms the state-of-the-art techniques on the ISIC 2016 dataset. W-net achieved higher accuracy (97.5%), dice coefficient (93.7%), specificity (98.3%), and Jaccard index (88.3%) compared to other methods. However, (Huang, Zhao, and Yang, 2019) achieved a higher sensitivity than the proposed method.

Table 8.3: On the ISIC2016 dataset, a performance comparison of the suggested network and other approaches.

Method	Accuracy	Recall	Specificity	Dice similarity	Jaccard index
(Huang, Zhao, and Yang, 2019)	95.8%	96.4%	92.5 %	91.1%	84.5%
(Khan et al., 2021)	95.38%	-	-	-	-
(Phan et al., 2021)	96.48%	_	_	92.61%	87.08%
(Tong et al., 2021)	95.4%	92.7%	96.1%	-	84.5%
Proposed method	97.5%	96 %	98.30 %	93.7 %	88.3%

8.5.3.2 Tests comparison on ISIC-2017 Dataset

Figure 8.10 shows the results of training a W-net model for a segmentation task on ISIC-2017. The figure indicates that the model effectively minimized the loss, stabilizing close to zero, and achieved a high accuracy of over 99%. These results demonstrate the effectiveness of the proposed W-net model for accurate segmentation without overfitting, making it a reliable solution for the task.

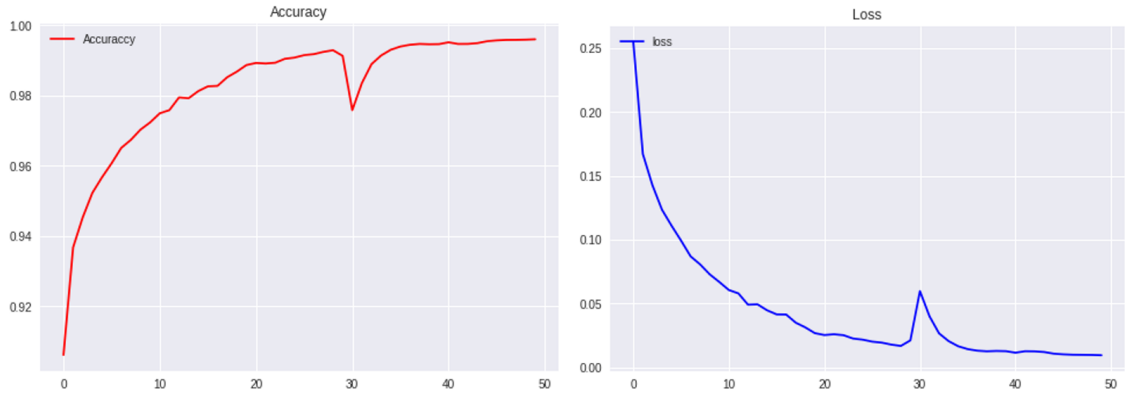


Figure 8.10: On ISIC2017, the suggested W-net Segmentation model's accuracy and loss performance curves.

Figure 8.11 displays the training results of a W-net network for an image segmentation task. The graph shows three curves representing the Dice coefficient, specificity, and recall metrics, which are commonly used for evaluating image segmentation algorithms. As the number of training epochs increases, the values of these metrics decrease and approach more than 95%. This indicates that the network is learning and improving its performance over time. The results demonstrate the effectiveness of the W-net network in accurately segmenting images, making it a promising approach for the image segmentation task.

Figure 8.11: Specificity, Dice coefficient and Recall performance curves of the suggested W-net Segmentation model on ISIC2017.

Figure 8.12 displays the results of the W-net proposed model for an image segmentation task on four image samples. The comparison between the segmentation results and the ground truth reveals that the proposed model produces improved segmentation accuracy, with the segmented regions closely resembling the ground truth. These results provide strong evidence of the effectiveness of the W-net proposed model in accurately performing the image segmentation task.

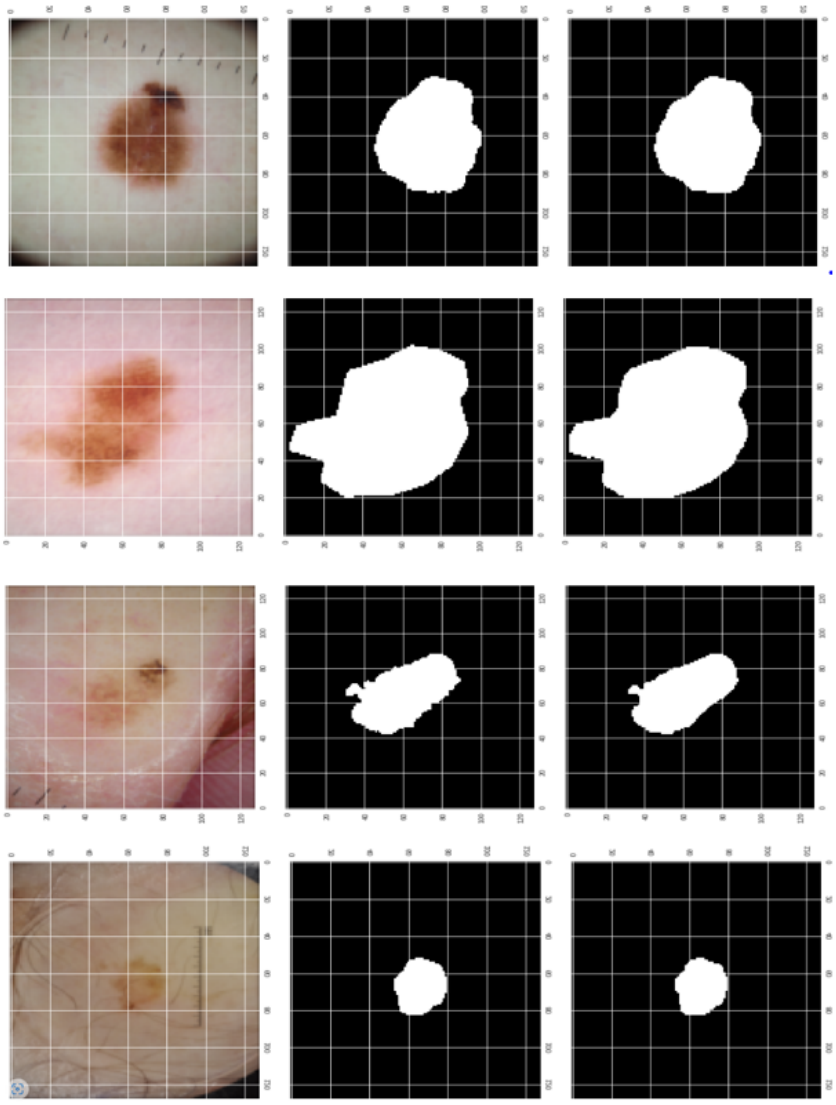


Figure 8.12: The segmentation results of W_net architecture on ISIC-2017. (1) Skin image (2) Ground truth images and (3) W-net results.

The evaluation in Figure 8.13 shows that the W-Net model performs better than the ResUnet model for lesion segmentation. It achieves more accurate segmentations that closely match the ground truth borders. The use of a dual encoder-decoder model in W-Net appears to be more robust compared to ResUnet's single encoder-decoder model, resulting in superior performance in lesion segmentation, even in challenging scenarios. Overall, the study provides strong evidence of the effectiveness of the W-Net model for accurate and reliable lesion segmentation.

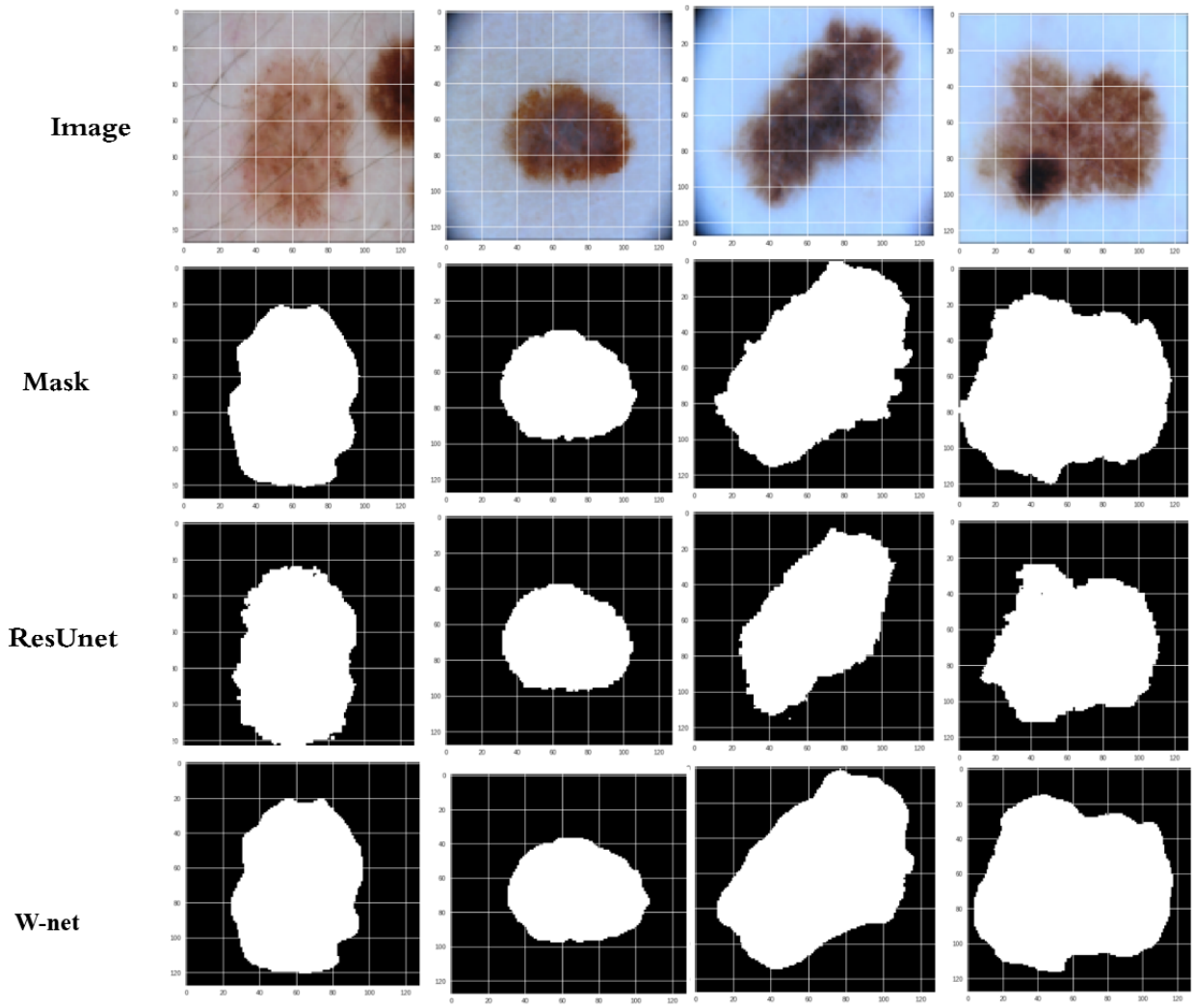


Figure 8.13: The segmentation results of W_net architecture. (1) Skin image (2) Ground truth images (3) ResUnet and (4) W-net results.

Based on the findings presented in Table 8.4, the proposed W-net architecture outperforms previous state-of-the-art strategies in several key metrics, including Accuracy, sensitivity, specificity, Dice similarity, and Jaccard index. Notably, the suggested approach achieved high values of accuracy (97.9%), sensitivity (95%), specificity (98.9%), Dice similarity coefficient (93.43%), and Jaccard index (86.86%). These results strongly indicate that the suggested method is a promising solution for medical image segmentation tasks, offering high usability and consistent performance.

Table 8.4: On the ISIC2017 dataset, a performance comparison of the proposed W-net network and other techniques.

Method	Accuracy	sensitivity	Specificity	Dice similarity	Jaccard index
(Al-Masni et al., 2018)	94.03%	85.40 %	96.69%	87.08%	77.11%
(Öztürk and Özkaya, 2020)	95.30%	85.44 %	98.08%	88.64%	78.34%
(Phan et al., 2021)	94.55%	-	-	87.61%	79.95%
(Khan et al., 2021)	95.79%	-	-	-	-
(Tong et al., 2021)	92.6%	82.5%	96.5%	-	74.2%
Proposed method	97.9 %	95%	98.9 %	93.4 %	86.86%

8.5.3.3 Tests comparison on the ISIC-2018 Dataset

Figure 8.14 shows the outcome of training a W-net model on the ISIC-2018 dataset for a segmentation challenge. The graph depicts the loss curve, which has stabilized at zero, and the accuracy curve, which has grown from a low of 75% to well above 95%.
This shows that the W-net model learnt the patterns in the data and reached a high level of accuracy in segmenting the data. The findings show that the suggested W-net model is a good match for the segmentation challenge and avoids overfitting, which is a typical problem in machine learning models. The graphic shows the W-net model's performance on the dataset and its efficacy in executing the image segmentation task.

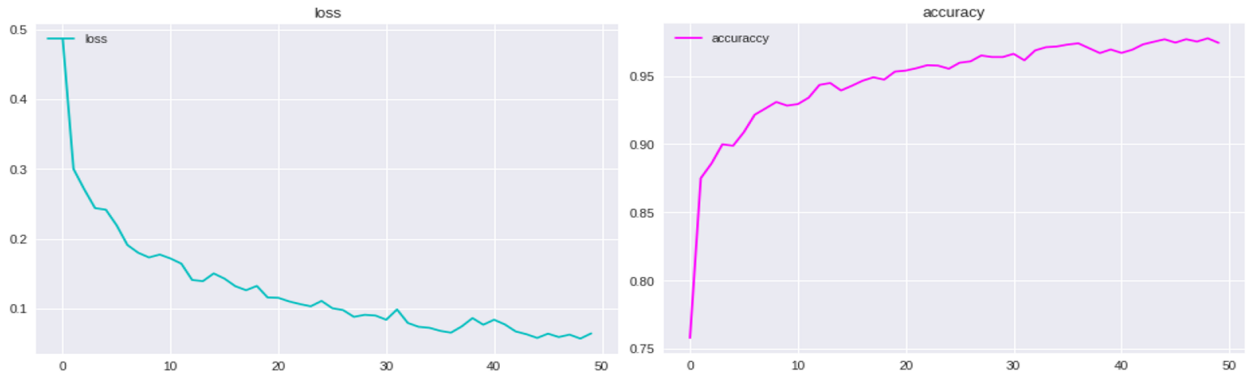


Figure 8.14: Accuracy and loss performance curves of the proposed W-net Segmentation model on ISIC-2018 Dataset.

The outcome of training a W-net network for an image segmentation task on ISIC-2018 is shown in Figure 8.15 . The graph depicts the model's performance using three regularly used metrics: Dice coefficient, Specificity, and Recall. The values of these measures rise and approach 95% as the number of training epochs grows, demonstrating that the network is learning from the training data and increasing its performance.
The graph shows that the W-net network is learning and increasing its performance from the training data, which is an important feature of a successful machine learning model. The graphic shows the W-net model's performance during the training process and its potential to improve over time.

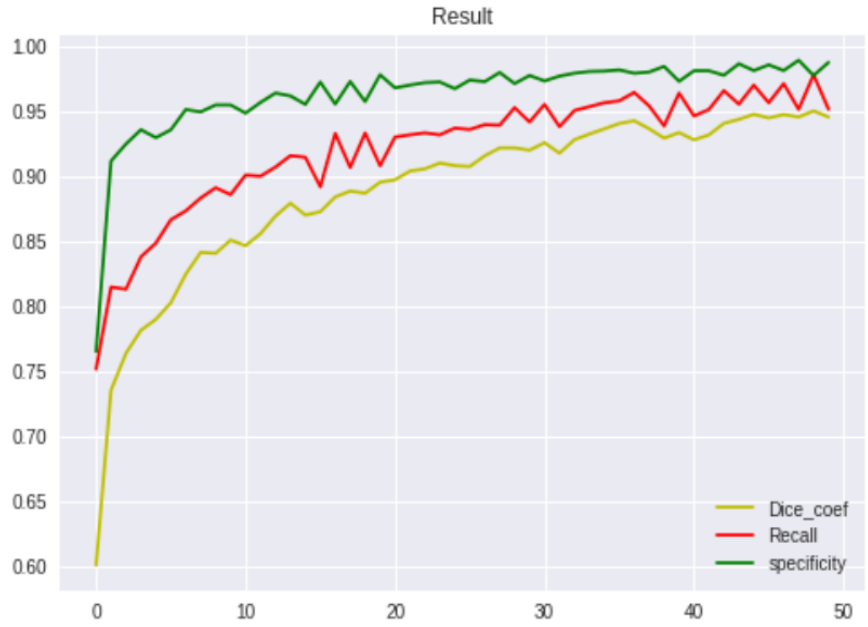


Figure 8.15: Dice coefficient, specificity and Recall performance curves of the proposed W-net Segmentation model on ISIC-2018.

Figure 8.16 shows the performance of the W-net suggested model on four image samples for an image segmentation challenge. The suggested model's findings are compared to the ground truth to demonstrate the improvement in segmentation outcomes. According to the Figure, the suggested W-net model is successful in delivering accurate segmentation results, since the results are closer to the ground truth. This demonstrates that the W-net suggested model is an effective solution for the image segmentation issue.

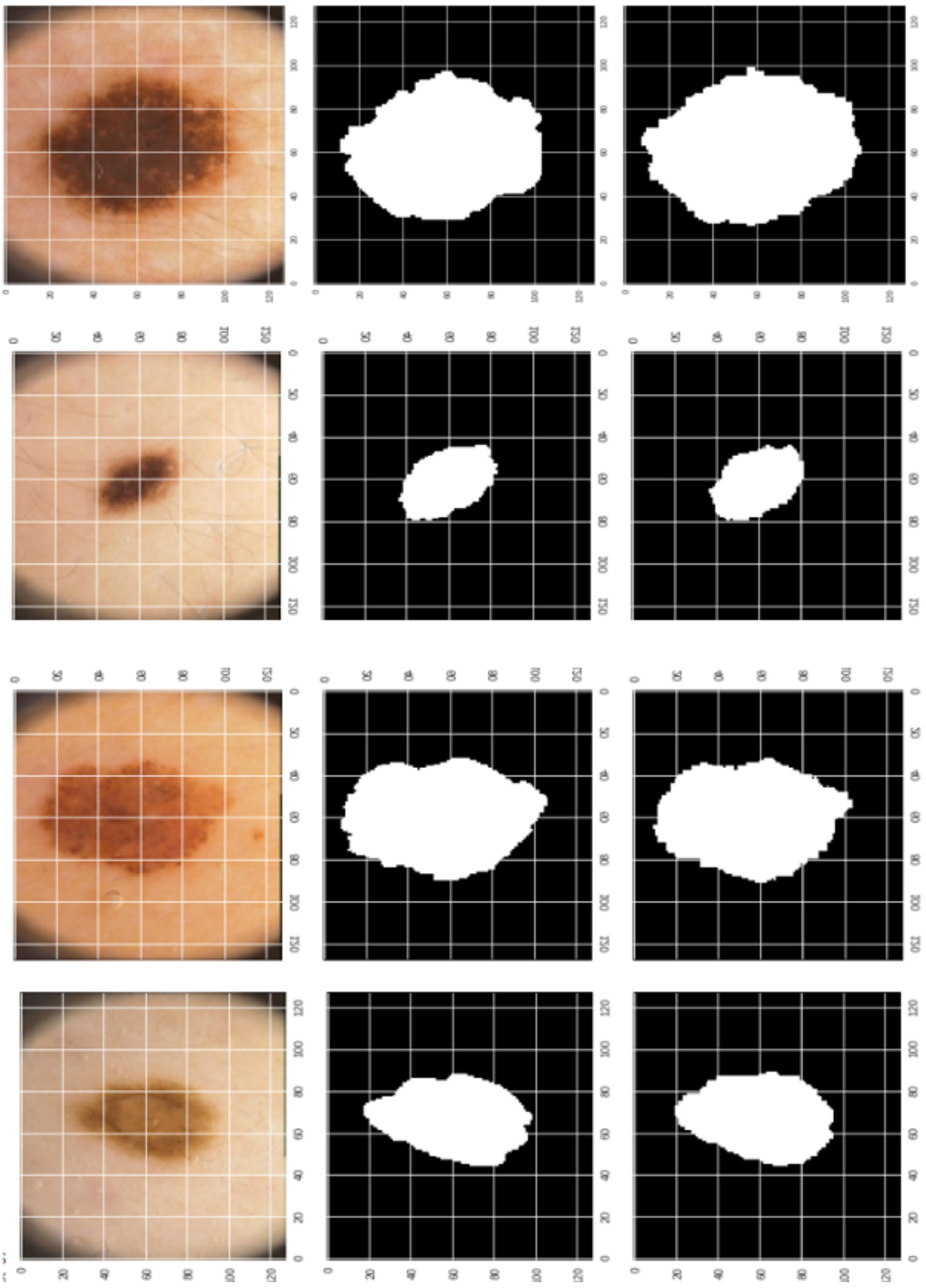


Figure 8.16: The segmentation results of W_net architecture on PH2. (1) Skin image (2) Ground truth images and (3) W-net results.

Based on the assessment and comparison provided in Figure 8.17, it is feasible to conclude that the W-Net model outperforms the ResUnet model for lesion segmentation. We demonstrate this by comparing the segmentation outputs of both models to the ground truth labels and discovering that the W-Net model produces more accurate segmentations with more closely matched region boundaries. Furthermore, we argue that employing a dual encoder-decoder model for segmentation is more robust than using a single encoder-decoder model, using W-outperformance Net's of ResUnet as evidence. W-Net segmentation results are satisfactory in all challenging settings, demonstrating the model's efficacy for lesion segmentation.

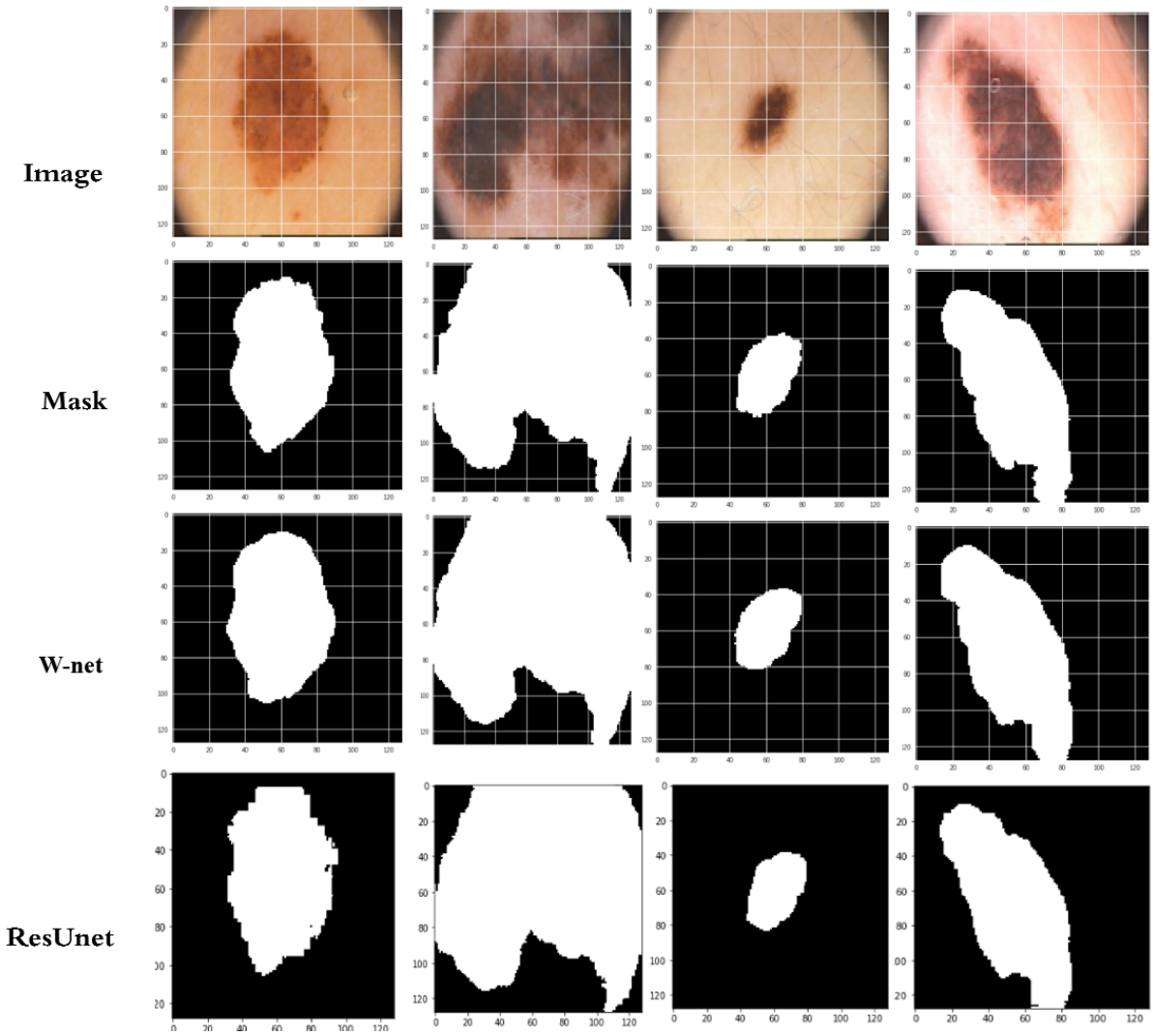


Figure 8.17: The segmentation results of W_net architecture. (1) Skin image (2) Ground truth images (3) W-net and (3) ResUnet results.

Based on the comparison provided in Table 8.5, it can be inferred that the proposed W-net model has achieved superior performance in terms of accuracy, dice coefficient, and Jaccard index, compared to the other state-of-the-art techniques. These metrics indicate the ability of the model to segment the skin lesions accurately and effectively. However, it is also worth noting that (Qamar, Ahmad, and Shen, 2021) and (Azad et al., 2020) have outperformed the proposed model in terms of sensitivity and specificity, respectively. Sensitivity refers to the ability of the model to accurately identify positive cases, while specificity refers to the ability of the model to accurately identify negative cases. Therefore, although our proposed approach has better overall segmentation performance, other techniques may have certain advantages in specific aspects of segmentation. Thus, the comparison presented in Table 8.5 suggests that the proposed W-net model is a highly competitive and effective approach for skin lesion segmentation.

Table 8.5: Performance comparison between the proposed network and other methods on the ISIC2018 dataset.

Method	Accuracy	Recall	Specificity	Dice similarity	Jaccard index
(Azad et al., 2020)	96.4%	87.5%	98.8 %	91.2%	_
(Dong et al., 2021)	96.41%	89.92 %	98.16%	91.19	83.99 %
(Khan et al., 2021)	92.69	-	-	-	-
(Qamar, Ahmad, and Shen, 2021)	96.95%	96.50%	97.00%	90.00%	83.30%
Proposed method	97.5%	95.7 %	98.5%	94.8%	89.9%

8.5.3.4 Tests comparison on the PH2 Dataset

Figure 8.18 is the result of training a W-net model for a segmentation task on the PH2 dataset. The figure shows the loss curve, which has stabilized close to zero, and the accuracy curve, which has increased from a low value of 82% to above 97%. This indicates that the W-net model has effectively learned the patterns in the data and has achieved a high level of accuracy in segmenting the data. The results demonstrate that the proposed W-net model is an effective solution for the segmentation task, and it has avoided overfitting, which is a common issue in machine learning models. The figure provides evidence of the performance of the W-net model on the PH2 dataset and its effectiveness in performing the image segmentation task.

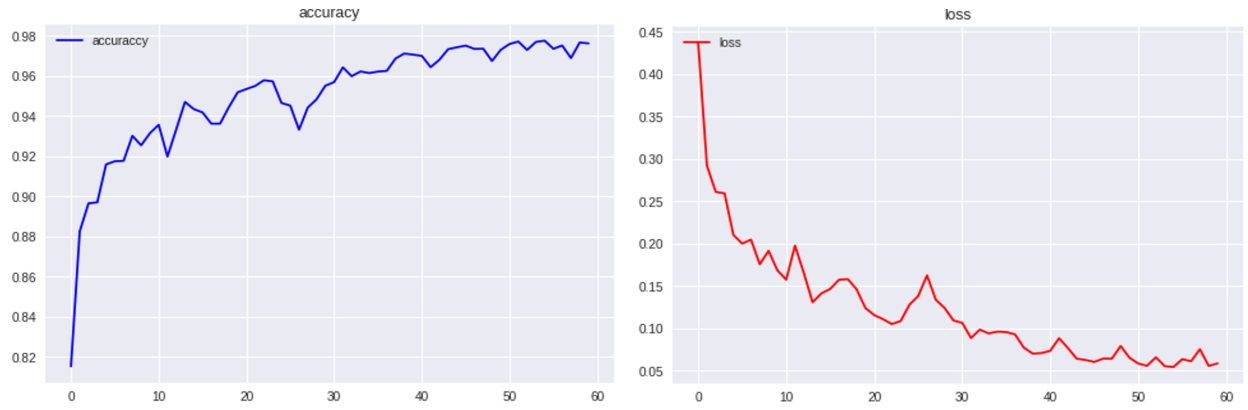


Figure 8.18: Accuracy and loss performance curves of the proposed W-net Segmentation model on PH2.

Figure 8.19 is the result of training a W-net network for an image segmentation task on PH2. The figure displays the performance of the model using three commonly used metrics: dice coefficient, specificity, and recall. As the number of training epochs increases, the values of these metrics decrease and approach more than 95%, indicating that the network is learning from the training data and improving its performance. The graph demonstrates that the W-net network is effectively learning from the training data and improving its performance, which is a crucial aspect of a successful machine learning model. The figure provides evidence of the performance of the W-net model during the training process and its ability to improve over time.

Figure 8.19: Dice coefficient, specificity and Recall performance curves of the proposed W-net Segmentation model on PH2.

Figure 8.20 demonstrates the performance of the W-net proposed model for an image segmentation task on four image samples. The results of the proposed model are compared with the ground truth to show the improvement in segmentation results. The figure indicates that the proposed W-net model is effective in producing accurate segmentation results, as the results are closer to the ground truth. This provides evidence that the W-Net proposed model is a good solution for the image segmentation task.

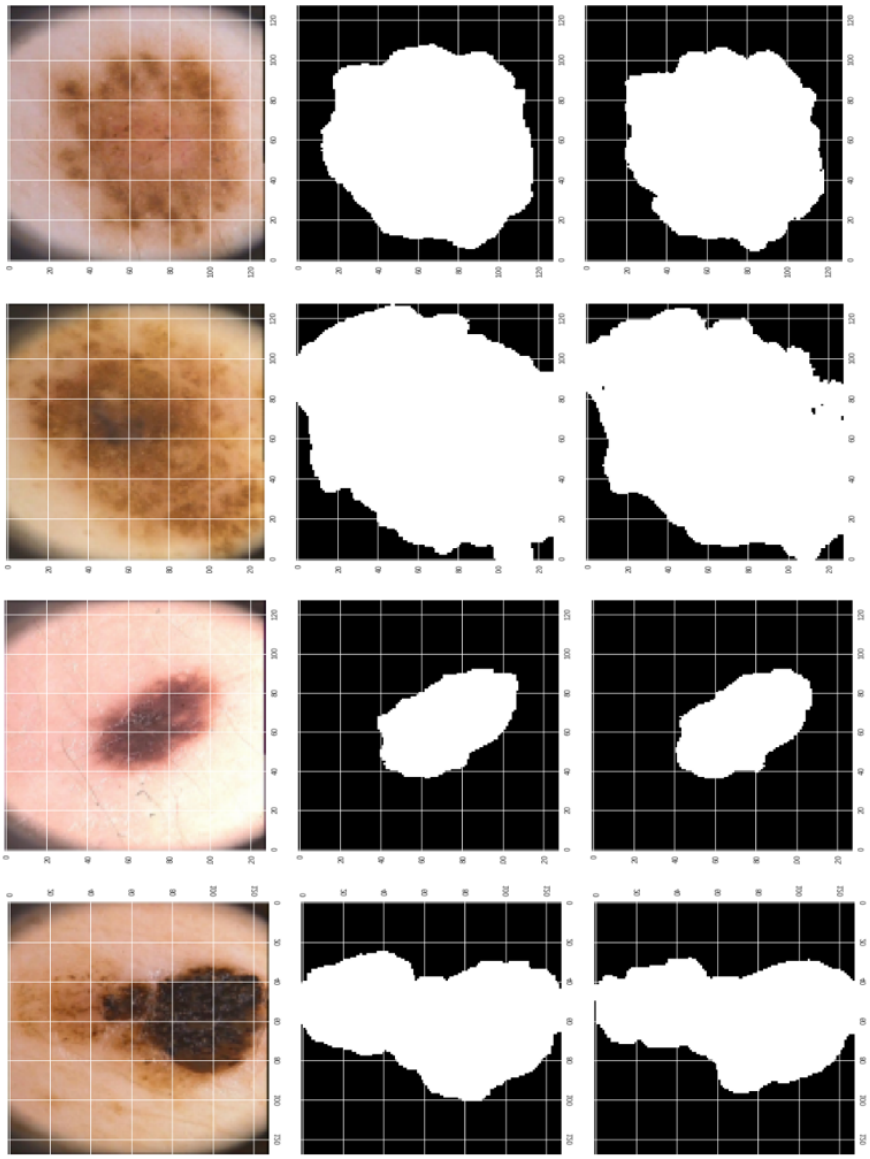


Figure 8.20: The segmentation results of W_net architecture on PH2. (1) Skin image (2) Ground truth images and (3) W-net results.

Based on the evaluation and comparison shown in the Figure 8.21, it is possible to deduce that the W-Net model is more successful for lesion segmentation than the ResUnet model. We show this by comparing the segmentation outputs of both models to the ground truth labels and finding that the W-Net model delivers more accurate segmentations with area borders that are more closely aligned with the ground truth. Additionally, we contend that utilizing a dual encoder-decoder model for segmentation is more resilient than using a single encoder-decoder model, citing W-outperformance Net's of ResUnet. W-Net segmentation results are adequate in all tough conditions, indicating the model's efficiency for lesion segmentation.

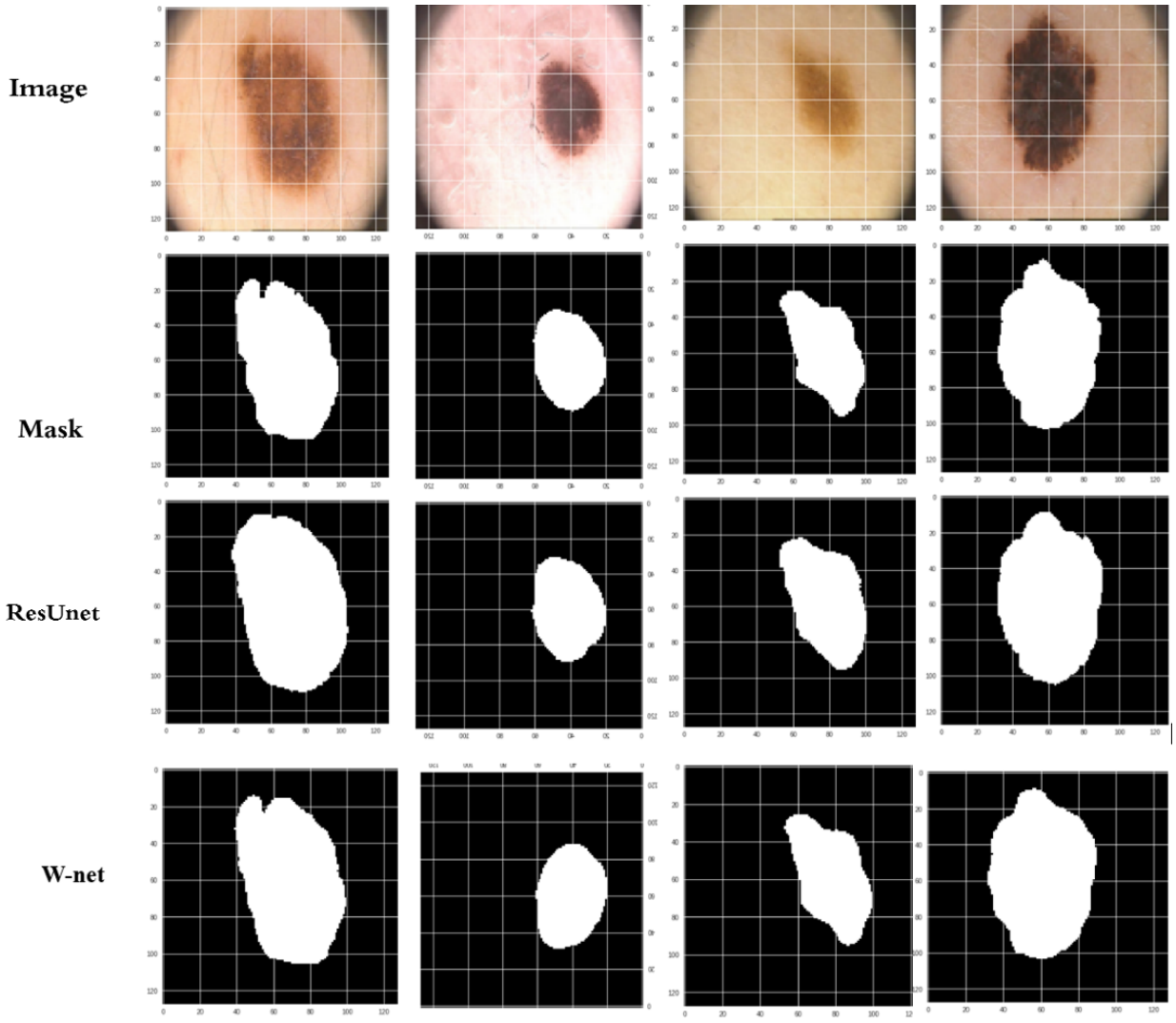


Figure 8.21: The segmentation results of W_net architecture. (1) Skin image (2) Ground truth images (3) ResUnet and (4) W-net results.

Table 8.6 presents the segmentation performance of the proposed method and compares it with state-of-the-art methods. The proposed method has been outperforming the other methods in terms of various performance metrics. The overall accuracy of the proposed method is 98.11%, which indicates that it is able to correctly classify 98.11% of the skin lesion pixels. The dice similarity coefficient, which is a measure of the overlap between the predicted and ground truth masks, is reported to be 94.84%. This indicates that the predicted masks are highly similar to the ground truth masks. The specificity of the proposed method is 98.92%, which indicates that it is able to correctly identify 98.92% of the pixels that do not belong to the skin lesion. The Jaccard index, which is a measure of the similarity between the predicted and ground truth masks, is reported to be 89.59%. This indicates that the predicted masks have a high degree of overlap with the ground truth masks. The sensitivity of the proposed method is 96.56%, which indicates that it is able to correctly identify 96.56% of the skin lesion pixels. This is a crucial performance metric, as it determines the ability of the method to accurately detect the skin lesion areas.

Table 8.6: Performance comparison between the proposed network and other methods on the PH2 dataset

Method	Accuracy	Recall	Specificity	Dice similarity	Jaccard index
(Shan et al., 2020)	93.63%	94.77 %	96.28%	90.26 %	83.51
(Xie et al., 2020)	94.2%	96.3 %	94.2%	91.9 %	85.7%
(Öztürk and Özkaya, 2020)	96.92%	96.88%	95.31%	93.02 %	87.1%
Proposed method	98.11%	96.56 %	98.92%	94.84%	89.59%

8.5.3.5 Discussion

The proposed W-net architecture for skin lesion segmentation is a significant contribution to the field of medical image analysis. The dual encoder-decoder approach of W-net is a novel idea, as it allows for more accurate lesion extraction by extracting both the global shape of the lesion and sharpening its limits. This approach is more effective than using a single encoder-decoder network, as it can provide more precise skin lesion borders, which are crucial for further treatments. The evaluation of W-net on four publicly available datasets confirms its effectiveness, with an accuracy of over 97% for all databases, outperforming all existing methods. W-net also outperforms existing approaches in terms

of sensitivity, specificity, and dice similarity.
The comparative results show that W-net is one of the best ways for skin lesion segmentation, as it effectively challenged state-of-the-art methods for skin lesion segmentation on the four datasets.
Thus, the proposed W-net architecture provides an effective solution to the problem of skin lesion segmentation and can potentially lead to improved diagnosis and treatment of skin lesions. Further research can investigate the application of W-net to other medical image analysis tasks and explore potential modifications to improve its performance even further.

8.5.4 Tests comparison and discussion ψ-net model

8.5.4.1 Tests comparison on ISIC-2016 Dataset

As shown in Figure 8.22, the ψ-net model was effectively trained and obtained a high level of accuracy in segmenting data. The figure indicates that the model effectively decreased the loss since it stabilized near zero. The accuracy curve demonstrates the model's ability since it begins with a low value of 85 percent and rises to above 98 percent. The results shown in Figure 8.22 show how successful the suggested W-net model is at segmenting data. The model obtained a high degree of accuracy while avoiding overfitting, indicating that it learned the patterns in the data successfully.

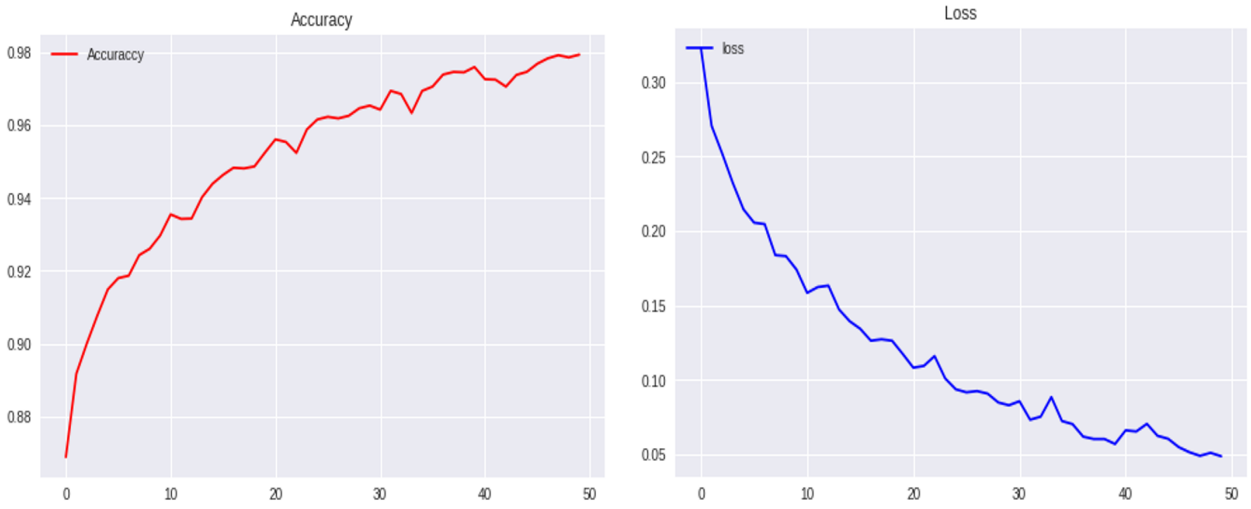


Figure 8.22: On ISIC2016, the proposed ψ-net Segmentation model's accuracy and loss performance curves.

Figure 8.23 depicts the outcomes of a ψ-net network training procedure. Three curves are depicted in the graph: one for the dice coefficient, one for specificity, and one for recall ability. The Dice coefficient, specificity, and recall are three measures typically used to assess the effectiveness of image segmentation algorithms. It is clear that as the number of training epochs rises, the metrics values increase and approach 95%. This suggests that the network is adapting and improving over time.

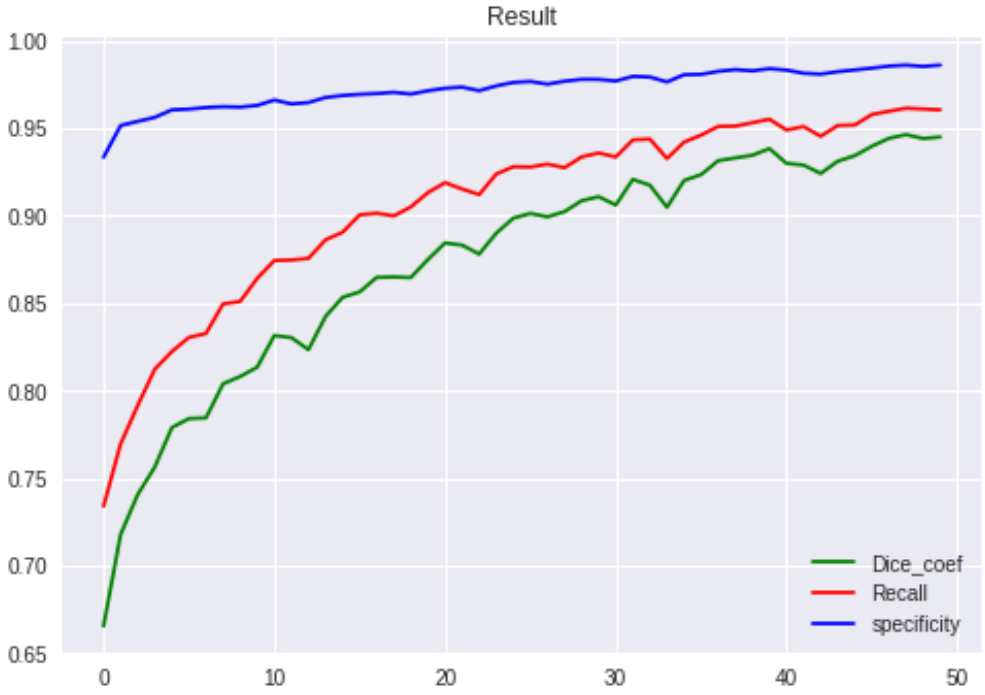


Figure 8.23: Specificity,Dice coefficient, and Recall performance curves of the proposed ψ_net Segmentation model on ISIC2016.

Figure 8.24 depicts the segmentation results of our ψ-net suggested model on four image samples. The proposed procedure yields enhanced segmentation results that are more accurate, demonstrating the effectiveness of the suggested approach.

The effectiveness of the suggested network ψ-net for lesion segmentation was evaluated using Figure 8.25. The input images are displayed in the first row, ground truth labels in the second row, and segmented outputs of ψ-net and Dense-Fire Unet are presented in the third and fourth rows, respectively.

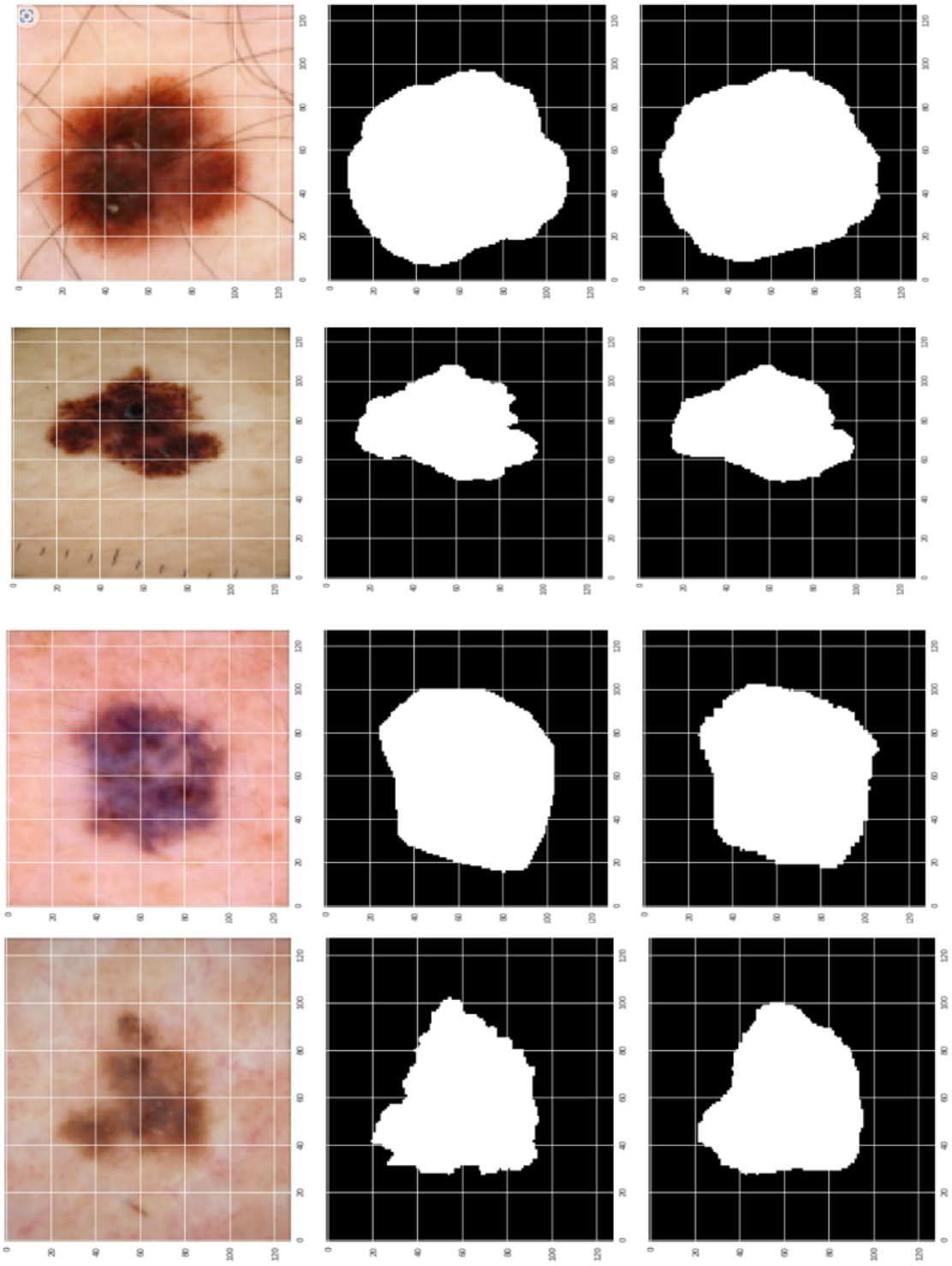


Figure 8.24: The segmentation results of ψ_net architecture. (1) Skin image (2) Ground truth images and (3) ψ-net results.

The suggested lesion network performed well in difficult situations, producing satisfactory segmentation outcomes. To confirm the effectiveness of ψ-net, segmentation results of Dense-Fire Unet and ψ-net were compared to ground truth labels. The results showed that the segmentation of ψ-net was more accurate than that of Dense-Fire Unet. The region borders extracted by ψ-net closely matched the region borders of the ground truth compared to those obtained by Dense-Fire Unet. The combination of an attention mechanism (spatial gate attention + channel attention) with a multi-scale convolution encoder improved the network performance, bringing the prediction region closer to the ground truth boundary.

This demonstrated the efficacy and accuracy of the proposed model. Overall, the findings suggest that the suggested network with attention mechanisms and a convolution encoder is an effective approach for lesion segmentation, producing more accurate results compared to Dense-Fire Unet. These findings could have important implications for improving lesion segmentation accuracy and aiding in the diagnosis and treatment of skin diseases.

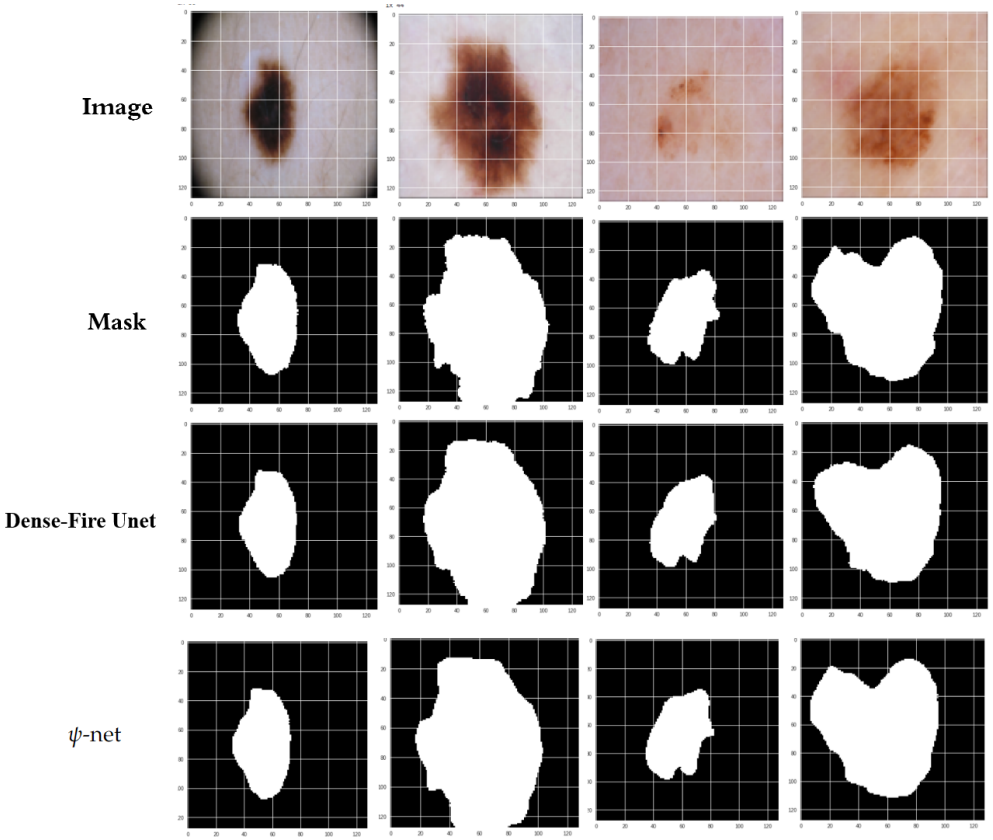


Figure 8.25: The segmentation results of ψ-net architecture. (1) Skin image (2) Ground truth images (3) Dense-Fire Unet and (4)ψ-net results .

Table 8.7 presents a comparison between the proposed network, called ψ-net, and other state-of-the-art techniques on the ISIC 2016 dataset. According to the table, the ψ-net outperforms other techniques in terms of recall, dice coefficient, and Jaccard index, with values of 96.3%, 93.81 %, and 88.5% respectively. However, it is worth noting that the proposed W-net demonstrates higher accuracy and specificity compared to the ψ-net. This suggests that while the ψ-net excels in certain areas, the W-net might have an edge in accurately identifying true positive cases.

Table 8.7: On the ISIC2016 dataset, a performance comparison of the suggested network and other approaches.

Method	Accuracy	Recall	Specificity	Dice similarity	Jacc
(Xie et al., 2020)	93.8%	87.0 %	96.4%	91.8%	85.8%
(Khan et al., 2021)	95.38%	-	-	-	-
(Phan et al., 2021)	96.48%	_	_	92.61%	87.08%
(Tong et al., 2021)	95.4%	92.7%	96.1%	-	84.5%
Proposed W-net	97.5%	96 %	98.30 %	93.7 %	88.3%
Proposed ψ-net	96.7%	96.3 %	97.69 %	93.81 %	88.5%

8.5.4.2 Results on the ISIC-2017 Dataset

The ψ-net model was effectively trained and achieved a high degree of accuracy in segmenting data, as demonstrated in Figure 8.26. The graph shows that the model effectively reduced the loss when it stabilized at zero. The accuracy curve indicates the model's capability because it starts low at 84 percent and grows to above 98 percent. The findings in Figure 8.26 demonstrate how effective the proposed model is at segmenting data. The model achieved a high degree of accuracy while avoiding overfitting, showing that it successfully understood the patterns in the data.

Figure 8.27 shows the results of a ψ-net network training operation. The graph shows three curves: one for the dice coefficient, one for specificity, and one for memory ability. The Dice coefficient, specificity, and recall are three commonly used metrics for evaluating the efficacy of picture segmentation algorithms. It is evident that as the number of training epochs increases, so do the metrics values, which reach 95%. This indicates that the network adapts and improves over time.

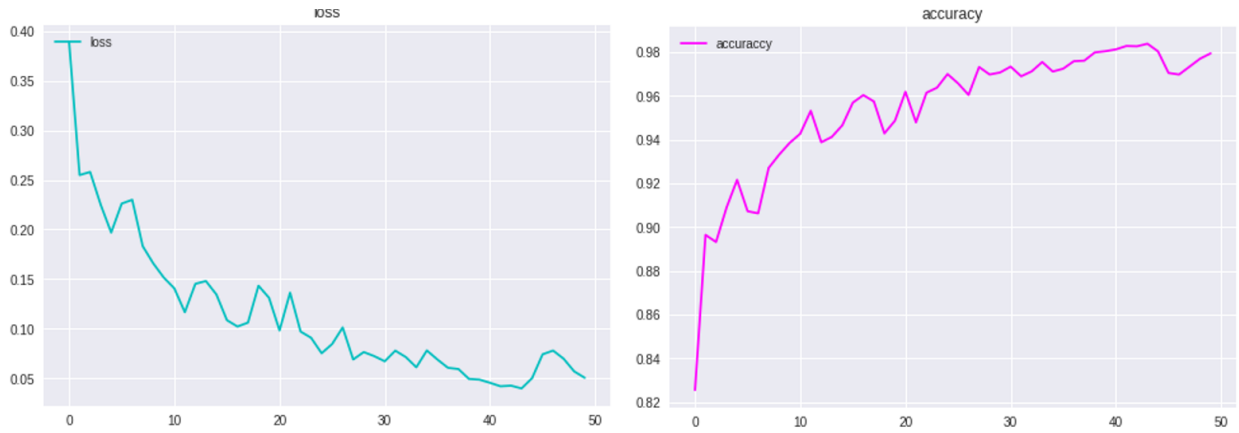


Figure 8.26: Accuracy and loss performance curves of the proposed ψ-net Segmentation model on ISIC-2017.

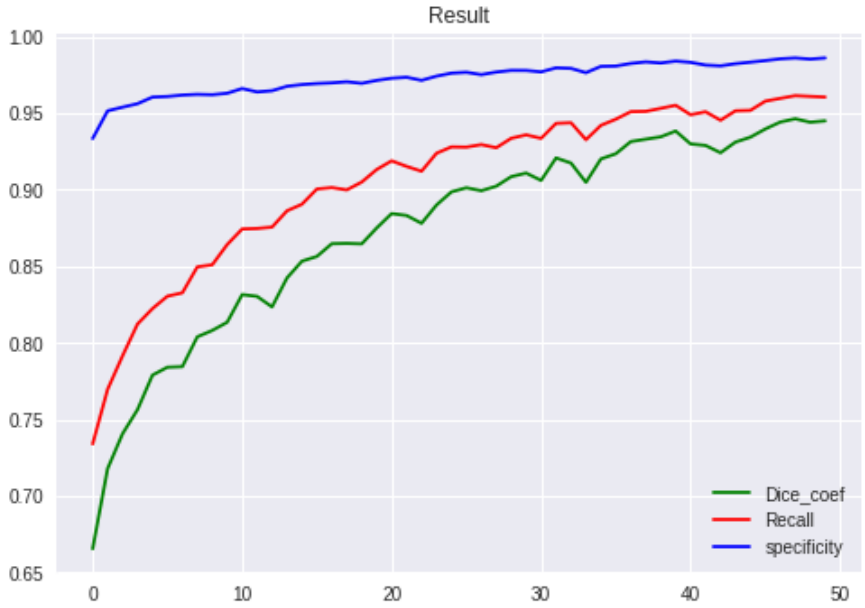


Figure 8.27: Specificity,Dice coefficient, and Recall performance curves of the proposed ψ_net Segmentation model on ISIC2017.

Figure 8.28 depicts the performance of the ψ-net suggested model on four picture samples for an image segmentation challenge. The suggested model's findings are compared to the ground truth to demonstrate the improvement in segmentation outcomes. According to Figure 8.28, the suggested ψ-net model is successful in providing accurate segmentation results since the conclusions are closer to the ground truth. This suggests that the ψ-net model is an effective solution to the image segmentation problem.

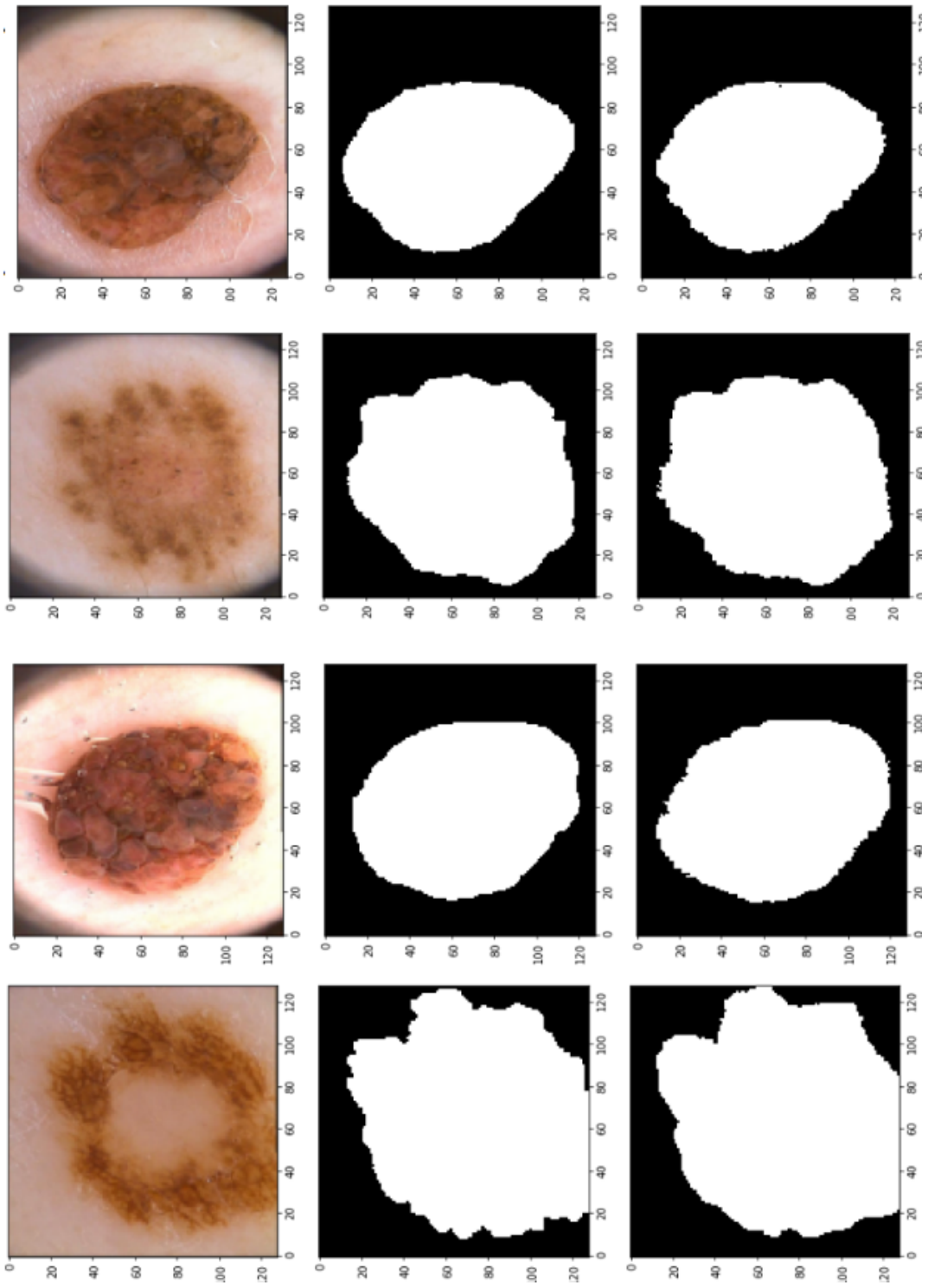


Figure 8.28: The segmentation results of ψ_net architecture. (1) Skin image (2) Ground truth images and (3) ψ-net results.

Figure 8.29 shows that the proposed skin lesion segmentation network utilizing the ISIC-2017 dataset outperformed the DENSE-Fire U-net design. The fact that the projected region is closer to the boundary of the ground truth indicates that the suggested network is more accurate in recognizing the limits of the skin lesion. The attention mechanism, which includes spatial gate attention and channel attention, as well as the convolutional encoder combination, appear to have contributed to the network's improved performance.

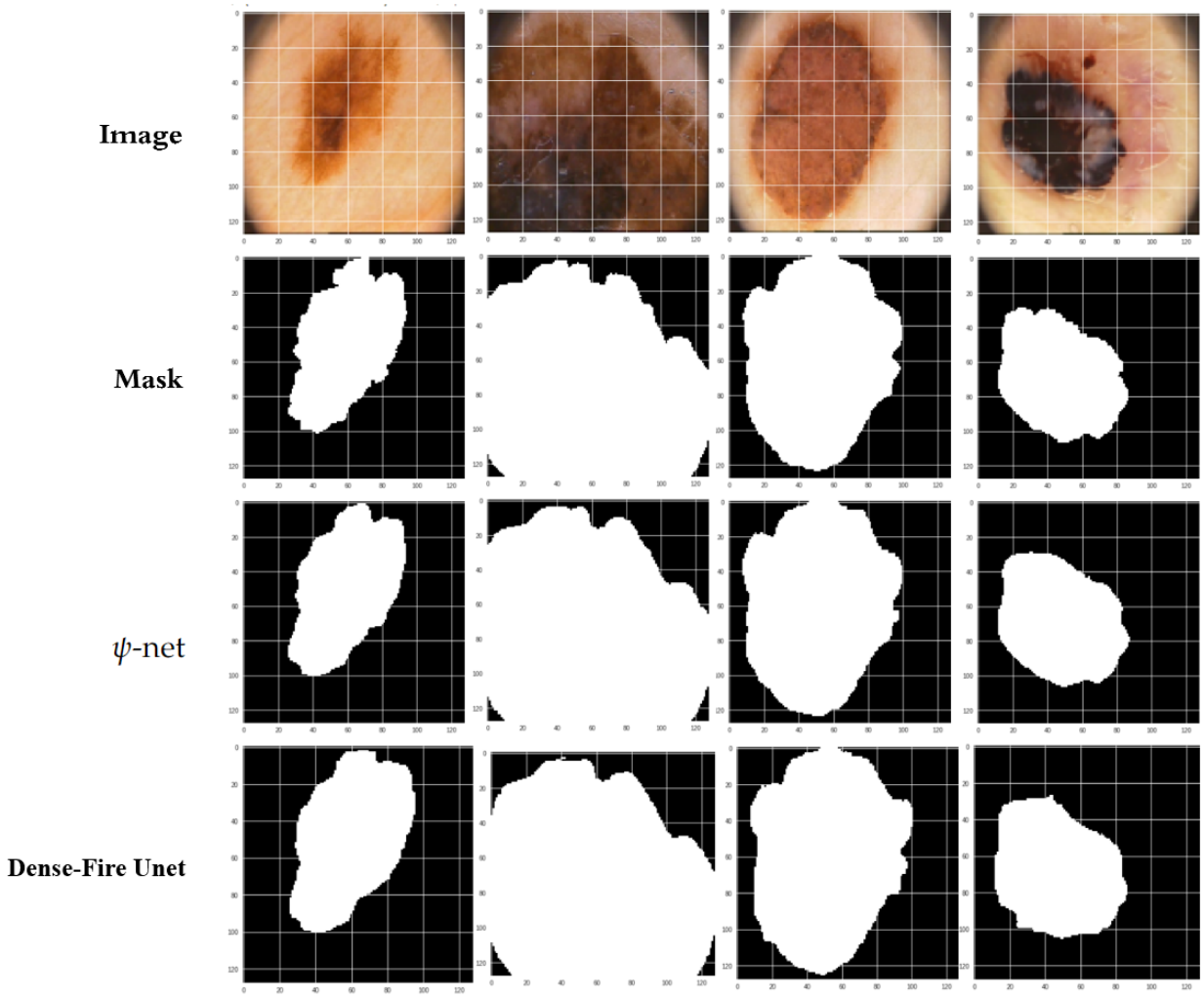


Figure 8.29: The segmentation results of ψ-net architecture. (1) Skin image (2) Ground truth images (3) ψ-net results and (4) Dense-Fire Unet.

Table 8.8 compares the proposed ψ-net segmentation method with established techniques, highlighting its superior performance in multiple metrics. A dice similarity coefficient of 94.4% indicates substantial agreement between predicted and actual masks, emphasizing accurate shape capture. The Jaccard index of 89.3% showcases strong structural similarity, affirming reliable segmentation. A sensitivity of 95.8% demonstrates the method's proficiency in detecting skin lesion pixels. However, W-net methods outperform ψ-net in accuracy and specificity, indicating better true positive identification.

Table 8.8: On the ISIC2017 dataset, a performance comparison of the proposed ψ_net network and other techniques.

Method	Accuracy	sensitivity	Specificity	Dice similarity	Jaccard index
(Al-Masni et al., 2018)	94.03%	85.40 %	96.69%	87.08%	77.11%
(Öztürk and Özkaya, 2020)	95.30%	85.44 %	98.08%	88.64%	78.34%
(Phan et al., 2021)	94.55%	-	-	87.61%	79.95%
(Khan et al., 2021)	95.79%	-	-	-	-
(Tong et al., 2021)	92.6%	82.5%	96.5%	-	74.2%
The Proposed W-net	97.9 %	95%	98.9 %	93.4 %	86.86%
Proposed method	97.3%	95.8 %	98.2%	94.4%	89.3%

8.5.4.3 Tests comparison on the PH2 Dataset

The figure 8.30 shows the results of training a ψ_net model on the PH2 dataset for a segmentation challenge. The graph depicts the loss curve, which has stabilized at zero, and the accuracy curve, which has grown from a low of 82% to well above 97%. This shows that the ψ_net model learnt the patterns in the data and reached a high level of accuracy in segmenting the data. The findings show that the suggested ψ-net model is a good match for the segmentation challenge and avoids overfitting, which is a typical problem in machine learning models. The graphic shows the ψ_net model's performance on the PH2 dataset and its efficacy in conducting image segmentation.

The outcome of training a ψ_net network for an image segmentation task on PH2 is shown in Figure 8.31. The graph depicts the model's performance using three regularly used metrics: Dice coefficient, Specificity, and Recall. The values of these measures grow and approach 95% as the number of training epochs increases, demonstrating that the network is learning from the training data and increasing its performance.

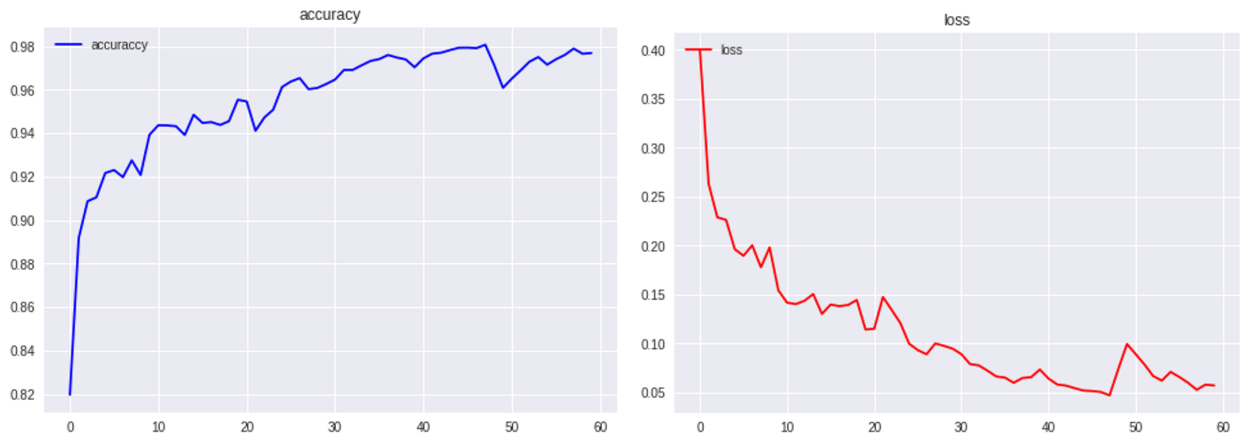


Figure 8.30: The suggested ψ_net Segmentation model's accuracy and loss performance curves on PH2.

The graph shows that the ψ_net network is learning and increasing its performance from the training data, which is an important feature of a successful machine learning model. The graphic shows the ψ_net model's performance during the training process and its potential to improve over time.

Figure 8.31: Specificity,Dice coefficient, and Recall performance curves of the proposed ψ_net Segmentation model on PH2.

Figure 8.32 shows the performance of the ψ_net suggested model on four image samples for an image segmentation challenge. The suggested model's findings are compared to the ground truth to demonstrate the improvement in segmentation outcomes. According to the Figure, the suggested ψ_net model is successful in delivering accurate segmentation results, since the results are closer to the ground truth. This demonstrates that the ψ_net suggested model is an effective solution for the image segmentation problem.

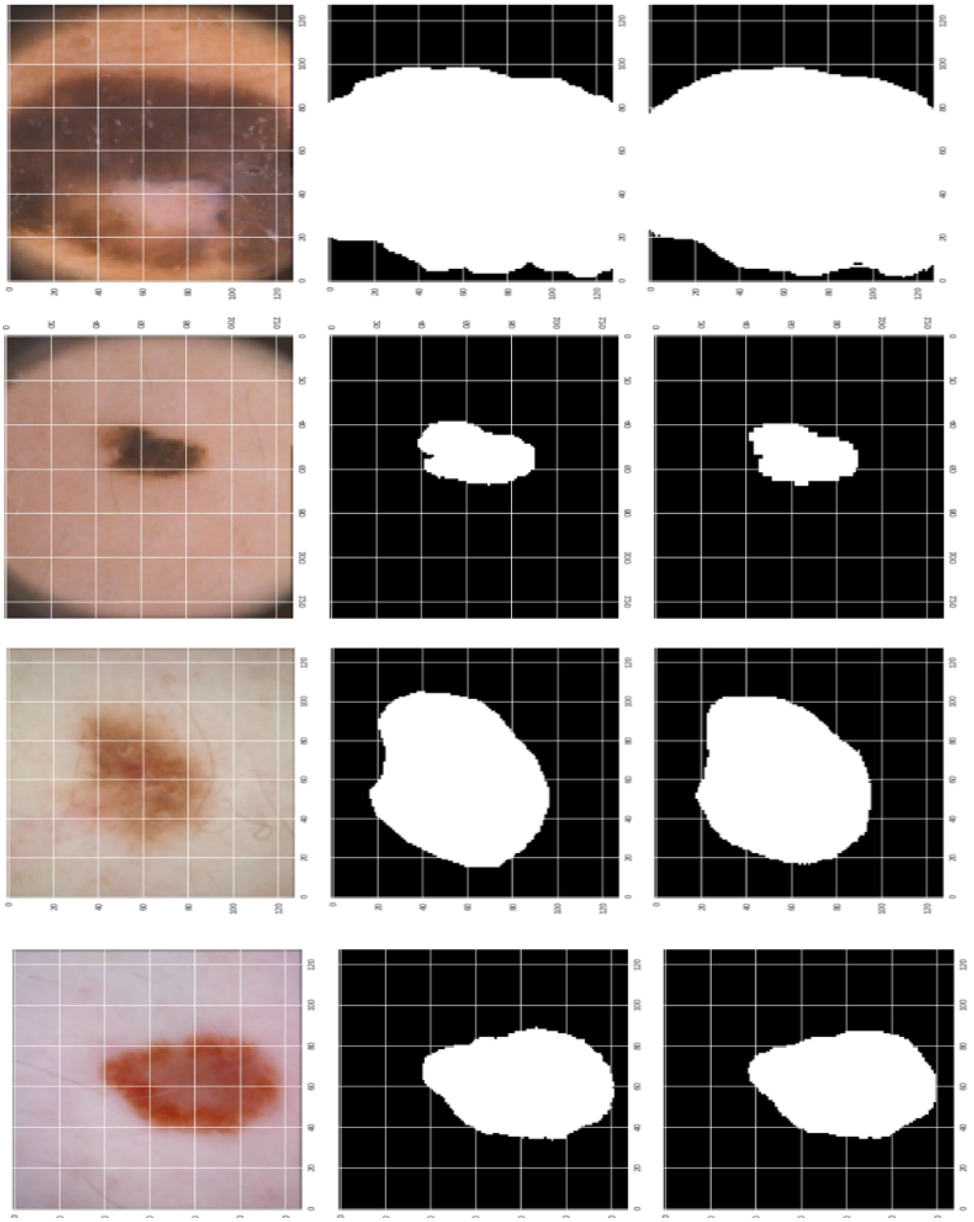


Figure 8.32: The segmentation results of ψ_net architecture. (1) Skin image (2) Ground truth images and (3) ψ_net results.

Figure 8.33 shows that the proposed skin lesion segmentation network using the PH2 dataset has achieved better results than the DENSE-Fire U-net architecture. The fact that the predicted area is closer to the ground truth's edge suggests that the proposed network has better precision in identifying the boundaries of the skin lesion. The attention mechanism, consisting of spatial gate attention and channel attention, and the convolutional encoder combination, appear to have contributed to the improved performance of the network. Overall, these results indicate that the proposed network is effective and accurate in segmenting skin lesions in the PH2 dataset.

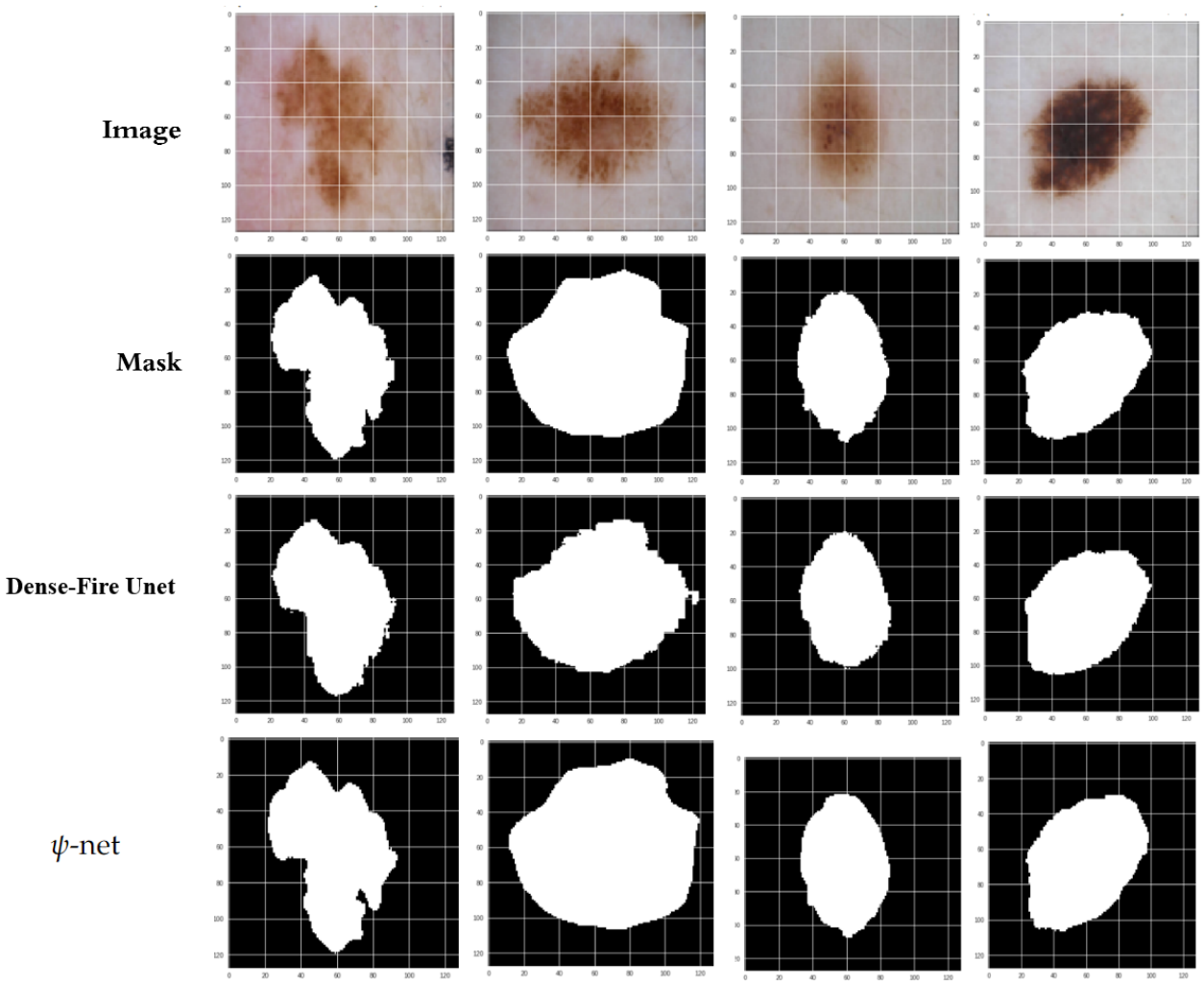


Figure 8.33: The segmentation results of ψ-net architecture. (1) Skin image (2) Ground truth images (3) ψ-net results and (4) Dense-Fire Unet.

Table 8.9 outlines the performance comparison between the proposed ψ-net segmentation approach and established techniques. Notably, the ψ-net showcases superiority across diverse benchmarks, achieving a 98.2% accuracy, an 89.6% Jaccard index similarity, and a 96.8% sensitivity for meticulous lesion identification. In essence, the ψ-net demonstrates exceptional proficiency in precisely segmenting skin lesions, surpassing prevailing methods. Nevertheless, it's noteworthy that "w-net" exhibits better results in dice similarity and specificity.

Table 8.9: Performance comparison between the proposed network and other methods on the PH2 dataset

Method	Accuracy	Recall	Specificity	Dice similarity	Jaccard index
(Tang et al., 2019)	96.69%	96.51%	95.26%	94.13%	89.40%
(Shan et al., 2020)	93.63%	94.77 %	96.28%	90.26 %	83.51
(Öztürk and Özkaya, 2020)	96.92%	96.88%	95.31%	93.02 %	87.1%
(Qiu et al., 2020)	96.20%	-	-	94.14%	89.20%
(Xie et al., 2020)	94.2%	96.3 %	94.2%	91.9 %	85.7%
The Proposed W-net	98.11%	96.7 %	98.92%	94.84%	89.59%
Proposed method	98.2%	96.8 %	98.6%	94.4%	89.6%

8.5.4.4 Discussion

the proposed network architecture called ψ-net, which is an extension of the U-Net and incorporates elements from Dense-Net, Squeeze-Net, and ResNet to improve feature extraction and segmentation performance for skin lesion segmentation in medical images. The ψ-net uses a multi-scale convolutional encoder and dual encoders to extract both the lesion's global shape and its boundaries. Additionally, we introduced a two-attention method, incorporating both spatial attention and channel attention, to enhance the model's ability to capture important information in images. The proposed framework was evaluated on four publicly available datasets: ISIC 2016, ISIC 2017, ISIC 2018, and PH2, and the results showed that the ψ-net outperformed state-of-the-art methods in terms of precision, F1-score, and Matthew Correlation Coefficient. The segmentation mask produced by the ψ-net was also visually closer to the ground truth and performed better at capturing the shape of information. However, it's crucial to note that "W-net" surpasses ψ-net in certain

metrics.

The experimental results supported the efficacy of the dual encoders, Fire-Resnet block, and attention processes for skin lesion segmentation, and we believe that the proposed architectural design has the potential to be successful. It's worth noting that the ψ-net achieved good segmentation results without pre-and post-processing, which is a significant advantage in medical imaging applications.

8.6 Skin Lesion Classification

Skin lesion image classification is a crucial task in detecting melanoma cancer and other forms of skin cancer. The process of image classification involves using machine learning algorithms to analyze the features of an image and categorize it into different classes. The accuracy of the classification is dependent on several factors, including the choice of feature descriptors, the strength of the classifier, and the size and diversity of the dataset used for training the model. In this work, an automatic Classification approach was implemented to classify the lesion into different classes. There are several techniques for classification. As part of this thesis, an algorithm (inception Resnet model as mentioned in the previous chapter) is adopted to greatly improve the accuracy and efficiency of skin lesion classification. This model leverages the segmentation results from the W-net, which has demonstrated effectiveness. Figure 8.34 draws the proposed classification framework of melanoma.

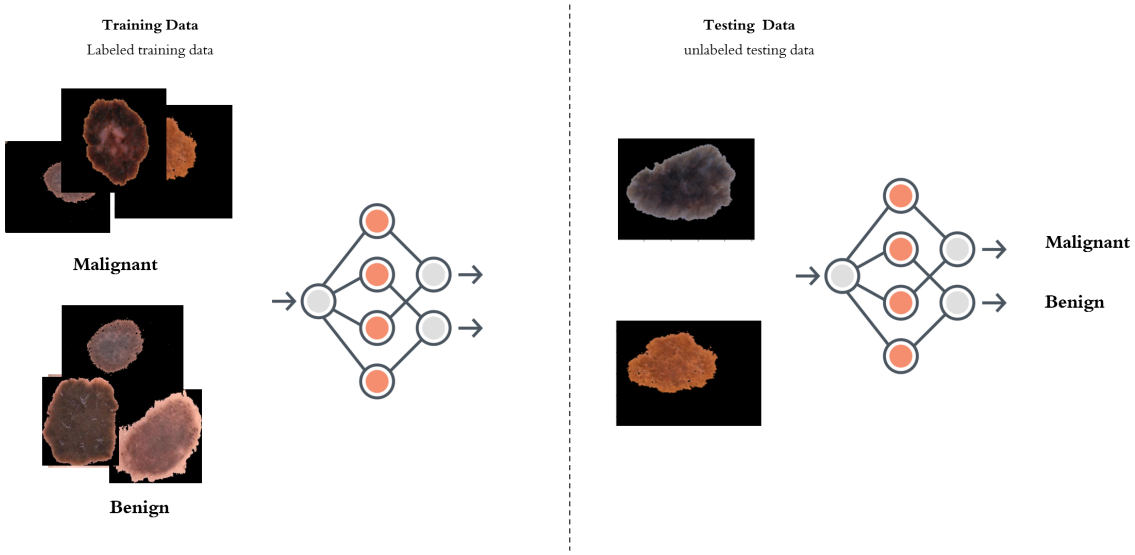


Figure 8.34: A classification framework of melanoma.

8.6.1 Hyper-parameters of inception Resnet model

Finding the optimal hyperparameters in deep neural networks poses a significant challenge, and it is crucial to automate the process of discovering these values. The ultimate goal is to improve the performance of the deep learning (DL) model by identifying the best hyperparameter configurations. In the case of training the inception ResNet DL model for classification, specific hyperparameter values are utilized. This includes setting the learning rate to 0.0001, the batch size to 8, the number of epochs to 80, the dropout rate to 0.3, and employing the Adam optimization algorithm. The model is trained using a binary cross-entropy loss function. Table 8.10 outlines the range of hyperparameters and their corresponding ideal values chosen for the inception ResNet model. It provides a comprehensive overview of the hyperparameter settings used in the training process. Furthermore, Figure 8.35 illustrates the architecture of the model during the training phase, giving a visual representation of its components and connections.

Table 8.10: Settings for training parameters

Type	**Range**	**Optimal Setting**
Optimizer	[SGD , Adam, RMSprop]	Adam optimization
Loss	-	binary cross-entropy
Learning rate	[0.1, 0.01, 0.001]	0.0001
Batch size	[8, 10, 16, 20, 25]	8
Number of iterations (Epoch)	[60, 80, 100, 120]	80
dropout rate	[0.2, 0.3, 0.5]	0.3
Activation function	-	ReLU, Sigmoid

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)	0	
conv2d (Conv2D)	(None, 112, 112, 64)	9472	input_1[0][0]
activation (Activation)	(None, 112, 112, 64)	0	conv2d[0][0]
max_pooling2d (MaxPooling2D)	(None, 56, 56, 64)	0	activation[0][0]
batch_normalization (BatchNorma	(None, 56, 56, 64)	256	max_pooling2d[0][0]
conv2d_1 (Conv2D)	(None, 56, 56, 64)	4160	batch_normalization[0][0]
activation_1 (Activation)	(None, 56, 56, 64)	0	conv2d_1[0][0]
batch_normalization_1 (BatchNor	(None, 56, 56, 64)	256	activation_1[0][0]
dropout (Dropout)	(None, 56, 56, 64)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 56, 56, 192)	110784	dropout[0][0]
activation_2 (Activation)	(None, 56, 56, 192)	0	conv2d_2[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 27, 27, 192)	0	activation_2[0][0]
batch_normalization_2 (BatchNor	(None, 27, 27, 192)	768	max_pooling2d_1[0][0]
batch_normalization_7 (BatchNor	(None, 27, 27, 192)	768	batch_normalization_2[0][0]
activation_4 (Activation)	(None, 27, 27, 192)	0	batch_normalization_7[0][0]
conv2d_6 (Conv2D)	(None, 27, 27, 128)	24704	activation_4[0][0]
batch_normalization_8 (BatchNor	(None, 27, 27, 128)	512	conv2d_6[0][0]
batch_normalization_4 (BatchNor	(None, 27, 27, 192)	768	batch_normalization_2[0][0]
conv2d_8 (Conv2D)	(None, 27, 27, 128)	24704	batch_normalization_2[0][0]
activation_5 (Activation)	(None, 27, 27, 128)	0	batch_normalization_8[0][0]
activation_3 (Activation)	(None, 27, 27, 192)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 27, 27, 96)	18528	batch_normalization_2[0][0]
batch_normalization_9 (BatchNor	(None, 27, 27, 128)	512	conv2d_8[0][0]
conv2d_7 (Conv2D)	(None, 27, 27, 128)	409728	activation_5[0][0]
conv2d_4 (Conv2D)	(None, 27, 27, 96)	165984	activation_3[0][0]
batch_normalization_5 (BatchNor	(None, 27, 27, 96)	384	conv2d_5[0][0]
add_1 (Add)	(None, 27, 27, 128)	0	batch_normalization_9[0][0] conv2d_7[0][0]
conv2d_3 (Conv2D)	(None, 27, 27, 64)	12352	batch_normalization_2[0][0]
add (Add)	(None, 27, 27, 96)	0	conv2d_4[0][0] batch_normalization_5[0][0]
batch_normalization_10 (BatchNo	(None, 27, 27, 128)	512	add_1[0][0]

(1)

max_pooling2d_2 (MaxPooling2D)	(None, 27, 27, 192)	0	batch_normalization_2[0][0]
batch_normalization_3 (BatchNor	(None, 27, 27, 64)	256	conv2d_3[0][0]
batch_normalization_6 (BatchNor	(None, 27, 27, 96)	384	add[0][0]
batch_normalization_11 (BatchNo	(None, 27, 27, 128)	512	batch_normalization_10[0][0]
conv2d_9 (Conv2D)	(None, 27, 27, 16)	3088	max_pooling2d_2[0][0]
concatenate (Concatenate)	(None, 27, 27, 304)	0	batch_normalization_3[0][0] batch_normalization_6[0][0] batch_normalization_11[0][0] conv2d_9[0][0]
batch_normalization_12 (BatchNo	(None, 27, 27, 304)	1216	concatenate[0][0]
dropout_1 (Dropout)	(None, 27, 27, 304)	0	batch_normalization_12[0][0]
batch_normalization_17 (BatchNo	(None, 27, 27, 304)	1216	dropout_1[0][0]
activation_7 (Activation)	(None, 27, 27, 304)	0	batch_normalization_17[0][0]
conv2d_13 (Conv2D)	(None, 27, 27, 192)	58560	activation_7[0][0]
batch_normalization_18 (BatchNo	(None, 27, 27, 192)	768	conv2d_13[0][0]
batch_normalization_14 (BatchNo	(None, 27, 27, 304)	1216	dropout_1[0][0]
conv2d_15 (Conv2D)	(None, 27, 27, 192)	58560	dropout_1[0][0]
activation_8 (Activation)	(None, 27, 27, 192)	0	batch_normalization_18[0][0]
activation_6 (Activation)	(None, 27, 27, 304)	0	batch_normalization_14[0][0]
conv2d_12 (Conv2D)	(None, 27, 27, 128)	39040	dropout_1[0][0]
batch_normalization_19 (BatchNo	(None, 27, 27, 192)	768	conv2d_15[0][0]
conv2d_14 (Conv2D)	(None, 27, 27, 192)	921792	activation_8[0][0]
conv2d_11 (Conv2D)	(None, 27, 27, 128)	350336	activation_6[0][0]
batch_normalization_15 (BatchNo	(None, 27, 27, 128)	512	conv2d_12[0][0]
add_3 (Add)	(None, 27, 27, 192)	0	batch_normalization_19[0][0] conv2d_14[0][0]
conv2d_10 (Conv2D)	(None, 27, 27, 128)	39040	dropout_1[0][0]
add_2 (Add)	(None, 27, 27, 128)	0	conv2d_11[0][0] batch_normalization_15[0][0]
batch_normalization_20 (BatchNo	(None, 27, 27, 192)	768	add_3[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 27, 27, 304)	0	dropout_1[0][0]
batch_normalization_13 (BatchNo	(None, 27, 27, 128)	512	conv2d_10[0][0]
batch_normalization_16 (BatchNo	(None, 27, 27, 128)	512	add_2[0][0]
batch_normalization_21 (BatchNo	(None, 27, 27, 192)	768	batch_normalization_20[0][0]

(2)

conv2d_16 (Conv2D)	(None, 27, 27, 32)	9760	max_pooling2d_3[0][0]
concatenate_1 (Concatenate)	(None, 27, 27, 480)	0	batch_normalization_13[0][0] batch_normalization_16[0][0] batch_normalization_21[0][0] conv2d_16[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 14, 14, 480)	0	concatenate_1[0][0]
batch_normalization_22 (BatchNo	(None, 14, 14, 480)	1920	max_pooling2d_4[0][0]
dropout_2 (Dropout)	(None, 14, 14, 480)	0	batch_normalization_22[0][0]
batch_normalization_27 (BatchNo	(None, 14, 14, 480)	1920	dropout_2[0][0]
activation_10 (Activation)	(None, 14, 14, 480)	0	batch_normalization_27[0][0]
conv2d_20 (Conv2D)	(None, 14, 14, 96)	46176	activation_10[0][0]
batch_normalization_28 (BatchNo	(None, 14, 14, 96)	384	conv2d_20[0][0]
batch_normalization_24 (BatchNo	(None, 14, 14, 480)	1920	dropout_2[0][0]
conv2d_22 (Conv2D)	(None, 14, 14, 96)	46176	dropout_2[0][0]
activation_11 (Activation)	(None, 14, 14, 96)	0	batch_normalization_28[0][0]
activation_9 (Activation)	(None, 14, 14, 480)	0	batch_normalization_24[0][0]
conv2d_19 (Conv2D)	(None, 14, 14, 96)	46176	dropout_2[0][0]
batch_normalization_29 (BatchNo	(None, 14, 14, 96)	384	conv2d_22[0][0]
conv2d_21 (Conv2D)	(None, 14, 14, 96)	230496	activation_11[0][0]
conv2d_18 (Conv2D)	(None, 14, 14, 96)	414816	activation_9[0][0]
batch_normalization_25 (BatchNo	(None, 14, 14, 96)	384	conv2d_19[0][0]
add_5 (Add)	(None, 14, 14, 96)	0	batch_normalization_29[0][0] conv2d_21[0][0]
conv2d_17 (Conv2D)	(None, 14, 14, 192)	92352	dropout_2[0][0]
add_4 (Add)	(None, 14, 14, 96)	0	conv2d_18[0][0] batch_normalization_25[0][0]
batch_normalization_30 (BatchNo	(None, 14, 14, 96)	384	add_5[0][0]
max_pooling2d_5 (MaxPooling2D)	(None, 14, 14, 480)	0	dropout_2[0][0]
batch_normalization_23 (BatchNo	(None, 14, 14, 192)	768	conv2d_17[0][0]
batch_normalization_26 (BatchNo	(None, 14, 14, 96)	384	add_4[0][0]
batch_normalization_31 (BatchNo	(None, 14, 14, 96)	384	batch_normalization_30[0][0]
conv2d_23 (Conv2D)	(None, 14, 14, 16)	7696	max_pooling2d_5[0][0]
concatenate_2 (Concatenate)	(None, 14, 14, 400)	0	batch_normalization_23[0][0] batch_normalization_26[0][0] batch_normalization_31[0][0] conv2d_23[0][0]

(3)

batch_normalization_38 (BatchNo	(None, 7, 7, 400)	1600	batch_normalization_33[0][0]
activation_13 (Activation)	(None, 7, 7, 400)	0	batch_normalization_38[0][0]
conv2d_27 (Conv2D)	(None, 7, 7, 320)	128320	activation_13[0][0]
batch_normalization_39 (BatchNo	(None, 7, 7, 320)	1280	conv2d_27[0][0]
batch_normalization_35 (BatchNo	(None, 7, 7, 400)	1600	batch_normalization_33[0][0]
conv2d_29 (Conv2D)	(None, 7, 7, 320)	128320	batch_normalization_33[0][0]
activation_14 (Activation)	(None, 7, 7, 320)	0	batch_normalization_39[0][0]
activation_12 (Activation)	(None, 7, 7, 400)	0	batch_normalization_35[0][0]
conv2d_26 (Conv2D)	(None, 7, 7, 160)	64160	batch_normalization_33[0][0]
batch_normalization_40 (BatchNo	(None, 7, 7, 320)	1280	conv2d_29[0][0]
conv2d_28 (Conv2D)	(None, 7, 7, 320)	2560320	activation_14[0][0]
conv2d_25 (Conv2D)	(None, 7, 7, 160)	576160	activation_12[0][0]
batch_normalization_36 (BatchNo	(None, 7, 7, 160)	640	conv2d_26[0][0]
add_7 (Add)	(None, 7, 7, 320)	0	batch_normalization_40[0][0] conv2d_28[0][0]
conv2d_24 (Conv2D)	(None, 7, 7, 256)	102656	batch_normalization_33[0][0]
add_6 (Add)	(None, 7, 7, 160)	0	conv2d_25[0][0] batch_normalization_36[0][0]
batch_normalization_41 (BatchNo	(None, 7, 7, 320)	1280	add_7[0][0]
max_pooling2d_7 (MaxPooling2D)	(None, 7, 7, 400)	0	batch_normalization_33[0][0]
batch_normalization_34 (BatchNo	(None, 7, 7, 256)	1024	conv2d_24[0][0]
batch_normalization_37 (BatchNo	(None, 7, 7, 160)	640	add_6[0][0]
batch_normalization_42 (BatchNo	(None, 7, 7, 320)	1280	batch_normalization_41[0][0]
conv2d_30 (Conv2D)	(None, 7, 7, 32)	12832	max_pooling2d_7[0][0]
concatenate_3 (Concatenate)	(None, 7, 7, 768)	0	batch_normalization_34[0][0] batch_normalization_37[0][0] batch_normalization_42[0][0] conv2d_30[0][0]
average_pooling2d (AveragePooli	(None, 1, 1, 768)	0	concatenate_3[0][0]
dropout_3 (Dropout)	(None, 1, 1, 768)	0	average_pooling2d[0][0]
flatten (Flatten)	(None, 768)	0	dropout_3[0][0]
dense (Dense)	(None, 2)	1538	flatten[0][0]

Total params: 6,755,586
Trainable params: 6,737,186
Non-trainable params: 18,400

(4)

Figure 8.35: The summary of Inception Resnet model.

8.6.2 Tests comparison of classification results of Inception Resnet model

8.6.2.1 Tests comparison on the ISIC-2016 Dataset

Figure 8.36 is displaying the Inception-Resnet training loss and accuracy curve on the ISIC2016 dataset. The the loss value decreases below 0.1 as the number of iterations increases, indicating that the model is improving in its ability to make accurate predictions. Additionally, the Inception-Resnet accuracy training curves for the ISIC2016 dataset also show performance values growing as the number of epochs increases until they exceed 0.98. This suggests that the model is becoming more accurate with increasing amounts of training.

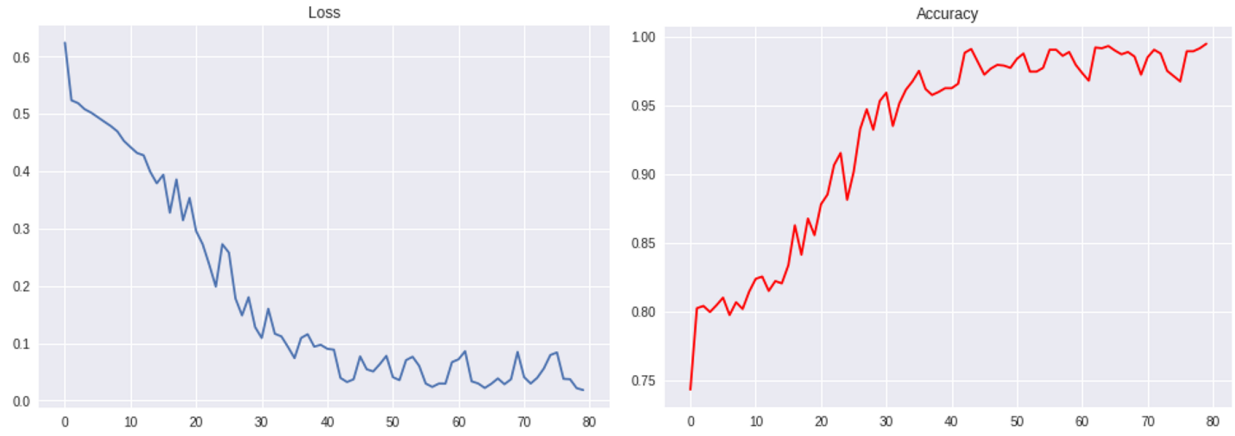


Figure 8.36: Classification accuracy and loss performance curves of the proposed inception resnet model on ISIC-2016.

Figure 8.37 displays the performance of the model during the training process over epochs. The graph depicts three commonly used metrics to evaluate the performance of image classification models: Dice coefficient, Specificity, and Recall. The increasing of these metrics over the training indicates that the model is learning from the training data and improving its performance. Specifically, the values of these metrics rise to more than 95% as the number of epochs increases, which is a good sign that the model is converging towards a successful outcome. In summary, the graph demonstrates that the Inception ResNet model is effectively learning from the training data and improving its performance on the image classification task.

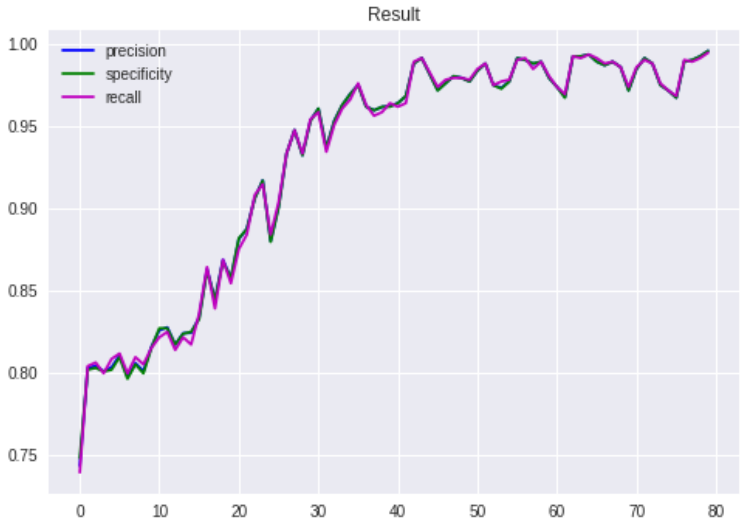


Figure 8.37: Specificity, dice coefficient and Recall performance curves of the inception resnet model Classification on ISIC2016.

Table 8.11 shows that Inception-Resnet outperforms the other methods on ISIC2016 in terms of sensitivity, specificity, accuracy, and precision, with 98% in all terms.

Table 8.11: Performance evaluation (%)of the proposed classification model on ISIC2016.

Method	Sensitivity %	Specificity %	Accuracy	Precision %
(Khan et al., 2022)	92.0 %	90.0%	92.1%	92.5%
(Khan et al., 2020b)	94,2	-	94,2	94,4
(Jayapriya and Jacob, 2020)	69.33 %	93.75%	88.92%	-
Our method	**98%**	**98%**	**98%**	**98%**

8.6.2.2 Tests comparison on the ISIC-2017 Dataset

The Inception-Resnet training loss and accuracy curve on the ISIC2017 dataset is depicted in Figure 8.38. As the number of iterations grows, the loss value decreases below 0.1. Furthermore, the Inception-Resnet accuracy training curves for the ISIC2017 dataset show performance values increasing with the number of epochs until they exceed 0.98. This shows that as the model is trained more, it becomes more accurate.

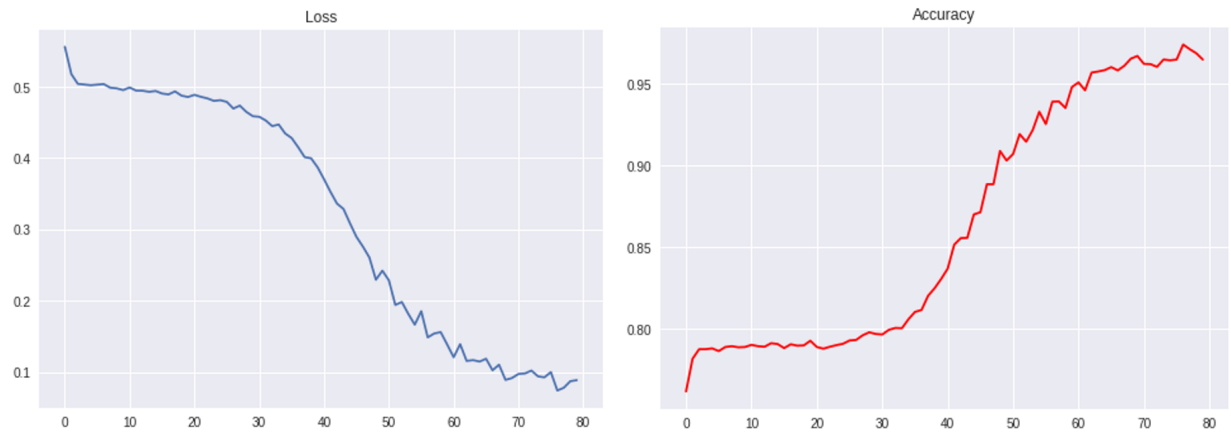


Figure 8.38: on ISIC-2017, accuracy and loss performance curves of inception resnet Classification model.

Figure 8.39 depicts the model's performance over epochs during the training process. The graph displays three generally used measures for evaluating the performance of an image classification model: dice coefficient, specificity, and recall. These metrics' growth over training suggests that the model is learning from the training data and improving its performance. Particularly, as the number of epochs increases, the values of these measures rise to more than 95%, indicating that the model is converging toward a favorable end. the graph shows that the Inception ResNet model is learning from training data and improving its performance on the image classification problem.

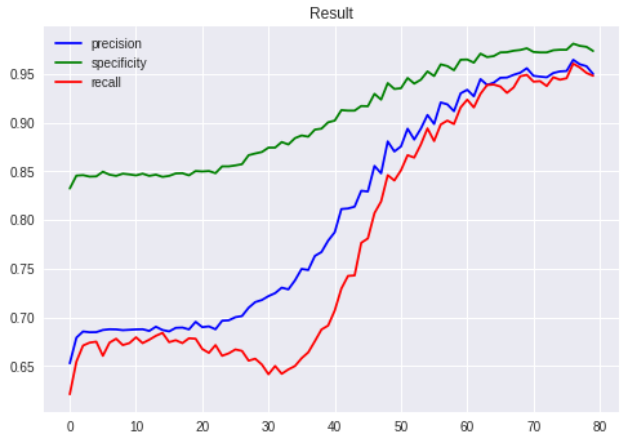


Figure 8.39: on ISIC2017, Specificity,Dice coefficient and Recall performance curves of the inception resnet model.

Inception-Resnet achieved the highest specificity (97%) and accuracy (97.1%) among the methods tested on the ISIC 2017 dataset, but at the expense of lower sensitivity and precision as shown in Table 8.12. .

Table 8.12: Performance evaluation of the proposed classification model on ISIC2017.

Method	Sensitivity %	Specificity %	Accuracy %	Precision %
(Khan et al., 2022)	**96.5%**	97.0%	96.5%	**96.5%**
(Khan et al., 2020b)	93%	-	93.4%	93.2%
(Jayapriya and Jacob, 2020)	81.28%	86.22%	85.3%	-
(Akram et al., 2020)	-	-	95.9%	-
Our method	95%	**97.1%**	**97%**	95%

8.6.2.3 Tests comparison on the PH2 Dataset

The Figure 8.40 depicts the Inception-Resnet training loss and accuracy curve on the ISIC2017 dataset. The loss value falls below 0.1 as the number of epochs increases. Furthermore, the PH2 dataset's Inception-Resnet accuracy training curves reveal that performance values increase with the number of epochs until they exceed 0.98. This demonstrates that the model grows more accurate as it is taught more.

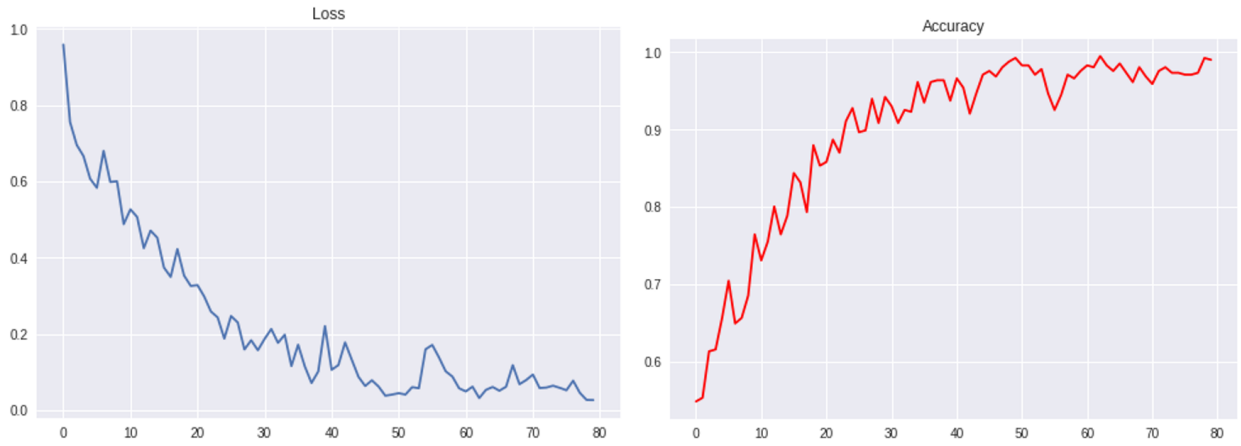


Figure 8.40: on PH2, accuracy and loss performance curves of inception resnet Classification model.

The Figure 8.41 depicts the model's performance during the training process on the ph2 dataset over epochs. The graph displays three generally used measures for evaluating the performance of an image classification model: dice coefficient, specificity, and recall. These metrics' growth over training suggests that the model is learning from the training data and improving its performance. Particularly, as the number of epochs increases, the values of these measures rise to more than 98%, showing that the Inception ResNet model is learning from training data and improving its performance on the image classification challenge.

Figure 8.41: on PH2: Specificity, Dice coefficient, and Recall performance curves of inception resnet model Classification.

For the PH2 dataset, Inception-Resnet outperforms others techniques in terms of specificity (99.1%), Sensitivity (98.3%) and precision (97.3%). However, as shown in Table 8.13, it performs lower than other methods in terms of accuracy.

Table 8.13: Performance evaluation of the proposed classification model on PH2 dataset.

Method	Sensitivity %	Specificity %	Acc %	Precision %
(Sarkar, Chatterjee, and Hazra, 2019)	-	-	96.7%	90%
(Gulati and Bhogal, 2019)	**100%**	96.87	97.5	-
(Akram et al., 2020)	-	-	98.8%	-
Our method	98.3%	**99.1%**	**98.40%**	**97.3%**

8.6.2.4 Discussion

The Inception-ResNet network represents a robust deep-learning architecture that excels in various image classification tasks. It consists of convolutional and pooling layers, as well as Inception-ResNet blocks that merge inception with residual neural networks, creating a powerful combination of techniques. Notably, the Inception-ResNet block plays a key role in enhancing feature extraction, leading to improved differentiation between different types of skin lesions. While its performance on the commonly used ISIC 2016 dataset is noteworthy, its relative success compared to existing methods can vary depending on the specific evaluation criteria employed. One inherent limitation is its reliance on the amount of training data available; expanding this dataset has the potential to enhance accuracy, which could be especially beneficial for experts engaged in the challenging task of melanoma detection and diagnosis.

8.7 Study Limitations

Even though the thesis has great findings in the area of skin cancer detection, it has some limitations that can be improved upon in future research. It is important to acknowledge these limitations as they provide a pathway for further improvement and refinement of the proposed system. Using only digital camera skin images and not including other types of images such as pathologist images is a limitation that can be addressed in future research. Incorporating a wider range of images for analysis, such as histopathological images, can enhance the accuracy and reliability of the proposed system.

Considering the depth of the melanoma is another limitation that can be improved upon in future research. Implementing 3D imaging techniques, such as optical coherence tomography or confocal microscopy, can provide a more comprehensive understanding of the melanoma's depth and improve the accuracy of detection and diagnosis.

Furthermore, implementing continuous capturing of images for improved accuracy and timeliness in diagnosis can also be a potential area for improvement. This can include using a series of images over time to track changes in the skin lesion and provide a more accurate diagnosis.

The proposed system is a valuable contribution to the field of skin cancer detection and can be modified and improved for use in other similar systems. Addressing the limitations can enhance the accuracy and reliability of the proposed system and improve its effectiveness in detecting and diagnosing skin cancer.

8.8 Conclusion

The chapter presents the experimental results of the proposed algorithms for a skin cancer detection system, which includes three main steps: pre-processing, W-net-based segmentation, ψ-net-based segmentation, and Inception Resnet model-based lesion classification. The proposed W-net model consists of a powerful CNN built using a dual encoder-decoder architecture, which addresses skin lesion segmentation problems. The suggested ψ-net architecture, which handles skin lesion segmentation issues, is inspired by various networks (U-net, SqueezeNet, ResNet, and Dense-Net). Additionally, a block-based Inception Resnet model was suggested to distinguish between different kinds of skin lesions. The proposed deep learning frameworks were tested on freely accessible databases, including the ISIC 2016, 2017, 2018, and PH2 datasets, to demonstrate the reliability of the approach. Overall, the chapter provides valuable insights into the development of deep learning-based skin cancer detection systems.

Chapter 9

CONCLUSIONS AND FUTURE WORK

The escalation in the incidence of malignant melanoma, the deadliest form of skin cancer, over recent decades underscores the imperative of early detection for effective intervention. Timely identification allows for more successful treatment, consequently mitigating mortality rates. Despite advancements in imaging technologies such as dermoscopy, the subjectivity in the initial assessment by physicians, particularly at the primary gateway of patients in the dermatology community, poses a considerable challenge. Advanced training for dermatologists becomes essential as distinguishing early-stage melanoma from benign tumors proves to be a complex task. To address this, computer-based diagnostic systems emerge as promising aids for less experienced practitioners.

In the scope of this thesis, pioneering and efficient algorithms are proposed to enhance the efficacy of computer-aided diagnostic systems dedicated to melanoma identification. This chapter furnishes a comprehensive overview of the thesis, encompassing system components, suggested algorithms, contributions, and experimental achievements. The discourse extends to potential avenues for further research in the field, culminating in a coherent conclusion.

The procedural steps of an automated melanoma detection system involve image segmentation and classification. Two algorithms, W-net, and ψ-net, are suggested for skin image segmentation. W-net, characterized by dual encoder-decoder architectures comprising a ResNet Encoder-Decoder, a ConvNet Encoder-Decoder, and a Feature Pyramid network, presents a robust CNN architecture addressing challenges in skin lesion segmentation. Conversely, ψ-net, an extension of U-Net drawing inspiration from Dense-Net, Squeeze-Net, and ResNeT, introduces innovative modifications. These include the integration of a Fire-ResNet module, placement of DenseFire-ResNet at the network's center, addition of a multi-scale convolutional encoder, and incorporation of a spatial-channel

attention gate to enhance feature extraction.

Evaluation on four publicly available datasets—ISIC 2016, ISIC 2017, ISIC 2018, and PH2—demonstrates the commendable performance of W-net and ψ-net models. Notably, these models exhibit high accuracy and specificity levels across all databases, surpassing existing methods and offering superior sensitivity, Jaccard index, and dice similarity.

Following the segmentation stage, the focus shifts to the classification stage, employing Inception-ResNet convolutional neural networks (CNNs). Inception-ResNet, amalgamating the Inception module and ResNet architecture, proves effective for categorizing segmented skin regions into different findings. Trained CNNs hold the potential to automate the classification of new segmented skin regions, thereby contributing to the diagnosis and treatment of diverse skin conditions. The proposed deep learning methods, if implemented, harbor the potential to elevate the accuracy and efficiency of skin cancer diagnosis, ultimately saving lives through enabling early detection and intervention.

9.1 Future work

This study offered fresh perspectives and methods for dealing with the essential elements of a computer-based melanoma detection system. Today, several potential routes for the development of this work have been identified. It is necessary to thoroughly test the system on a huge number of images taken under various circumstances, such as thousands of lesions from human beings, in order to establish a reliable computer-based system for the detection of skin cancer in the real world. This would require a well-plannedplanned division of the many medical institutes. This system must satisfy the demands of both expert and non-expert gainers, such as dermatologists who need a second opinion on their clinical diagnosis of a certain lesion and general practitioners who need to improve the accuracy of their diagnosis in order to feel more confident. This system may be created with medical students in mind as a training aid. The ultimate goal is to empower individuals to do basic skin analyses and seek medical guidance and therapy depending on the results obtained. This will result in time, money savings, and an earlier diagnosis.It might be advantageous to offer a large dataset with various images of the same lesion. To consider the various aspects of the lesion, these similar images can be obtained using a variety of imaging modalities, such as dermoscopy, ultrasound, and others. This may appeal to several details about the same tumor, such as the depth or surface of the lesion and other standards. As a result, the information gathered would be helpful to estimate

and predict with more accuracy. Yet, in some circumstances, taking a series of images over time might be a useful alternative for detection. Additional study on the topic of this thesis it is possible to research and enhance the newly proposed classification approach. Moreover, it may be fed with feature extraction approaches to boost accuracy.

Bibliography

Abbas, Qaisar, Irene Fondón, and Muhammad Rashid (2011). "Unsupervised skin lesions border detection via two-dimensional image analysis". In: *Computer methods and programs in biomedicine* 104.3, e1–e15.

ABCDE of Skin Cancer (2019). URL: https://www.conehealth.com/services/cancer-care/skin/abcde-of-skin-cancer/.

Abreu Dias, Domingo de (2019). "Musculoskeletal abnormality detection on x-ray using transfer learning". In.

Adegun, Adekanmi and Serestina Viriri (2021). "Deep learning techniques for skin lesion analysis and melanoma cancer detection: a survey of state-of-the-art". In: *Artificial Intelligence Review* 54.2, pp. 811–841.

Adjed, Faouzi (2017). "Skin cancer segmentation and detection using total variation and multiresolution analysis". PhD thesis. Univeristé Paris-Saclay; Université d'Evry-Val-d'Essonne; UTP Petronas.

Akram, Tallha et al. (2020). "A multilevel features selection framework for skin lesion classification". In: *Human-centric Computing and Information Sciences* 10.1, pp. 1–26.

Al-Masni, Mohammed A et al. (2018). "Skin lesion segmentation in dermoscopy images via deep full resolution convolutional networks". In: *Computer methods and programs in biomedicine* 162, pp. 221–231.

Al-Masri, Anas (2019). "How does back-propagation in artificial neural networks work". In: *Towards Data Science* 29.

Alfed, Naser et al. (2015). "Pigment network-based skin cancer detection". In: *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, pp. 7214–7217.

ALGERIA: SKIN CANCERS (2020). URL: https://www.worldlifeexpectancy.com/algeria-skin-cancers.

Alqudah, Ali Mohammad, Hiam Alquraan, and Isam Abu Qasmieh (2019). "Segmented and non-segmented skin lesions classification using transfer learning and adaptive

moment learning rate technique using pretrained convolutional neural network". In: *Journal of biomimetics, biomaterials and biomedical engineering*. Vol. 42. Trans Tech Publ, pp. 67–78.

Alzubaidi, Laith et al. (2021). "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions". In: *Journal of big Data* 8.1, pp. 1–74.

American cancer society (2013). URL: http://www.cancer.org/acs/groups/content/@research/documents/document/acspc-040951.pdf.

Argenziano, Giuseppe et al. (1998). "Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions: comparison of the ABCD rule of dermatoscopy and a new 7-point checklist based on pattern analysis". In: *Archives of dermatology* 134.12, pp. 1563–1570.

Argenziano, Giuseppe et al. (2003). "Dermoscopy of pigmented skin lesions: results of a consensus meeting via the Internet". In: *Journal of the American Academy of Dermatology* 48.5, pp. 679–693.

Ashraf, Rehan et al. (2020). "Region-of-interest based transfer learning assisted framework for skin cancer detection". In: *IEEE Access* 8, pp. 147858–147871.

Azad, Reza et al. (2020). "Attention deeplabv3+: Multi-level context attention mechanism for skin lesion segmentation". In: *European conference on computer vision*. Springer, pp. 251–266.

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla (2017). "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12, pp. 2481–2495.

Barata, Catarina, Jorge S Marques, and Teresa Mendonça (2013). "Bag-of-features classification model for the diagnose of melanoma in dermoscopy images using color and texture descriptors". In: *Image Analysis and Recognition: 10th International Conference, ICIAR 2013, Póvoa do Varzim, Portugal, June 26-28, 2013. Proceedings 10*. Springer, pp. 547–555.

Basal cell carcinoma (2022). URL: https://www.mayoclinic.org/diseases-conditions/basal-cell-carcinoma/symptoms-causes/syc-20354187#:~:text=on%20brown%20skin-,Basal%20cell%20carcinoma%20is%20a%20type%20of%20skin%20cancer%20that,a%20type%20of%20skin%20cancer..

Basal Cell Carcinoma (2022). URL: http://www.bccancer.bc.ca/books/skin-cancer-prevention-early-diagnosis-courses/course-readings/skin-cancer-early-diagnosis-readings/basal-cell-carcinoma.

Berhold, Jedediah M (2019). *Convolutional neural network architecture study for aerial visual localization*. Tech. rep. AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH.

Bi, Lei et al. (2016). "Automatic melanoma detection via multi-scale lesion-biased representation and joint reverse classification". In: *2016 IEEE 13th international symposium on biomedical imaging (ISBI)*. IEEE, pp. 1055–1058.

Bi, Lei et al. (2017). "Dermoscopic image segmentation via multistage fully convolutional networks". In: *IEEE Transactions on Biomedical Engineering* 64.9, pp. 2065–2074.

Binder, Michael et al. (1995). "Epiluminescence microscopy: a useful tool for the diagnosis of pigmented skin lesions for formally trained dermatologists". In: *Archives of dermatology* 131.3, pp. 286–291.

Bissoto, Alceu et al. (2018). "Deep-learning ensembles for skin-lesion segmentation, analysis, classification: RECOD titans at ISIC challenge 2018". In: *arXiv preprint arXiv:1808.08480*.

Borgersen, Jonas Nagell (2021). "Attention segmentation approaches for plankton images captured in-situ". MA thesis. NTNU.

Brandi Jones, MSN-ED RN-BC (2022). *The Subcutaneous Layer of Skin*. URL: https://www.verywellhealth.com/the-hypodermis-is-the-lowermost-layer-of-skin-2710144.

Braun, Ralph Peter et al. (2005). "Dermoscopy of pigmented skin lesions". In: *Journal of the American Academy of Dermatology* 52.1, pp. 109–121.

Brownlee, Jason (2021). "How to choose an activation function for deep learning". In: *Machine Learning Mastery*.

Cakir, Burak Ömür, Peter Adamson, and Cemal Cingi (2012). "Epidemiology and economic burden of nonmelanoma skin cancer". In: *Facial Plastic Surgery Clinics* 20.4, pp. 419–422.

Cancer Facts & Figures (2022). URL: https://www.cancer.org/acs/groups/content/@epidemiologysurveilance/documents/docu%20ment/acspc-036845.pdf.

Casey, Kevin (2020). "How to Explain Machine Learning in Plain English". In: *https://enterprisersproject.com/article/2019/7/machine-learning-explained-plain-english*.

Castillejos, Heydy et al. (2012). "Wavelet transform fuzzy algorithms for dermoscopic image segmentation". In: *Computational and mathematical methods in medicine* 2012.

Celebi, M Emre and Azaria Zornberg (2014). "Automated quantification of clinically significant colors in dermoscopy images and its application to skin lesion classification". In: *IEEE systems journal* 8.3, pp. 980–984.

Celebi, M Emre et al. (2009). "Lesion border detection in dermoscopy images". In: *Computerized medical imaging and graphics* 33.2, pp. 148–153.

Celebi, M Emre et al. (2010). "Robust border detection in dermoscopy images using threshold fusion". In: *2010 IEEE International Conference on Image Processing*. IEEE, pp. 2541–2544.

Celebi, M Emre et al. (2015). "A state-of-the-art survey on lesion border detection in dermoscopy images". In: *Dermoscopy image analysis* 10, pp. 97–129.

Charmchi, Sadegh (2018). "Optimized U-Net for Left Ventricle Segmentation". In.

Chen, Dongwei et al. (2020). "Deep residual learning for nonlinear regression". In: *Entropy* 22.2, p. 193.

Chen, Liang-Chieh et al. (2014). "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062*.

— (2017a). "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4, pp. 834–848.

Chen, Liang-Chieh et al. (2017b). "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587*.

Chen, Liang-Chieh et al. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818.

Chollet, Francois (2021). *Deep learning with Python*. Simon and Schuster.

Codella, Noel et al. (2019). "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)". In: *arXiv preprint arXiv:1902.03368*.

Codella, Noel CF et al. (2018). "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)". In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, pp. 168–172.

Dai, Jifeng et al. (2017). "Deformable convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 764–773.

Dhawan, Atam P et al. (2009). "Multispectral optical imaging of skin-lesions for detection of malignant melanomas". In: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, pp. 5352–5355.

Di Leo, Giuseppe et al. (2008). "An improved procedure for the automatic detection of dermoscopic structures in digital ELM images of skin lesions". In: *2008 IEEE Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*. IEEE, pp. 190–194.

Diakogiannis, Foivos I et al. (2020). "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162, pp. 94–114.

DiChiara, T (2010). *Pictures of Moles and Melanoma skin cancer–Learn o tell the difference with pictures*.

Dildar, Mehwish et al. (2021). "Skin cancer detection: a review using deep learning techniques". In: *International journal of environmental research and public health* 18.10, p. 5479.

Dong, Yuying et al. (2021). "Fac-net: Feedback attention network based on context encoder network for skin lesion segmentation". In: *Sensors* 21.15, p. 5172.

Elgamal, Mahmoud (2013). "Automatic skin cancer images classification". In: *International Journal of Advanced Computer Science and Applications* 4.3.

Eltayef, Khalid Ahmad A (2017). "Segmentation and lesion detection in dermoscopic images". PhD thesis. Brunel University London.

Emre Celebi, M et al. (2008). "Border detection in dermoscopy images using statistical region merging". In: *Skin Research and Technology* 14.3, pp. 347–353.

Encyclopaedia Britannica, The Editors of (2015). *dermisanatomy*. URL: `https://www.britannica.com/science/dermis`.

— (2021). *epidermisanatomy*. URL: `https://www.britannica.com/science/epidermis-anatomy`.

Erol, Recep (2018). *Skin Cancer Malignancy Classification with Transfer Learning*. University of Central Arkansas.

Farhadi, Farnoush (2017). *Learning activation functions in deep neural networks*. Ecole Polytechnique, Montreal (Canada).

Filimon, Delia-Maria and Adriana Albu (2014). "Skin diseases diagnosis using artificial neural networks". In: *2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, pp. 189–194.

Gaikwad, Akash S (2018). "Pruning convolution neural network (SqueezeNet) for efficient hardware deployment". PhD thesis.

Ganster, Harald et al. (2001). "Automated melanoma recognition". In: *IEEE transactions on medical imaging* 20.3, pp. 233–239.

Garnavi, Rahil et al. (2010). "Automatic segmentation of dermoscopy images using histogram thresholding on optimal color channels". In: *International Journal of Medicine and Medical Sciences* 1.2, pp. 126–134.

Glaister, Jeffrey, Alexander Wong, and David A Clausi (2014). "Segmentation of skin lesions from digital images using joint statistical texture distinctiveness". In: *IEEE transactions on biomedical engineering* 61.4, pp. 1220–1230.

Gomez, David Delgado et al. (2007). "Independent histogram pursuit for segmentation of skin lesions". In: *IEEE transactions on biomedical engineering* 55.1, pp. 157–161.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

Gu, Dongfeng (2017). "3D densely connected convolutional network for the recognition of human shopping actions". PhD thesis. Université d'Ottawa/University of Ottawa.

Gulati, Savy and Rosepreet Kaur Bhogal (2019). "Detection of malignant melanoma using deep learning". In: *Advances in Computing and Data Sciences: Third International Conference, ICACDS 2019, Ghaziabad, India, April 12–13, 2019, Revised Selected Papers, Part I 3*. Springer, pp. 312–325.

Gutman, David et al. (2016). "Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC)". In: *arXiv preprint arXiv:1605.01397*.

Harangi, Balazs (2018). "Skin lesion classification with ensembles of deep convolutional neural networks". In: *Journal of biomedical informatics* 86, pp. 25–32.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hernandez, AB (2019). "Deep Learning Models for Semantic Segmentation of Mammography Screenings."" In.

Hibler, Brian P, Qiaochu Qi, and Anthony M Rossi (2016). "Current state of imaging in dermatology". In: *Semin Cutan Med Surg* 35.1, pp. 2–8.

Hoshyar, Azadeh Noori (2015). "Automatic Skin Cancer Detection System". PhD thesis. University of Technology Sydney.

Howard, Andrew et al. (2019). "Searching for mobilenetv3". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324.

Howard, Andrew G (2013). "Some improvements on deep convolutional neural network based image classification". In: *arXiv preprint arXiv:1312.5402*.

Howard, Andrew G et al. (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861*.

Huang, Gao et al. (2017). "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Huang, Lin, Yi-gong Zhao, and Tie-jun Yang (2019). "Skin lesion segmentation using object scale-oriented fully convolutional neural networks". In: *Signal, Image and Video Processing* 13.3, pp. 431–438.

Iandola, Forrest N et al. (2016). "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360*.

Iyatomi, Hitoshi et al. (2008). "An improved internet-based melanoma screening system with dermatologist-like tumor area extraction algorithm". In: *Computerized Medical Imaging and Graphics* 32.7, pp. 566–579.

Jaworek-Korjakowska, Joanna and Ryszard Tadeusiewicz (2014). "Determination of border irregularity in dermoscopic color images of pigmented skin lesions". In: *2014 36th annual international conference of the IEEE engineering in medicine and biology society*. IEEE, pp. 6459–6462.

Jayapriya, Kalyanakumar and Israel Jeena Jacob (2020). "Hybrid fully convolutional networks-based skin lesion segmentation and melanoma detection using deep feature". In: *International Journal of Imaging Systems and Technology* 30.2, pp. 348–357.

Jin, Shuo et al. (2020). "AI-assisted CT imaging analysis for COVID-19 screening: Building and deploying a medical AI system in four weeks". In: *MedRxiv*.

Joy McElroy, Brenda Kemp (2022). *Anatomy of the Skin*. URL: `https://konadermatologist.com/anatomy-of-the-skin/`.

Kalouche, Simon, Andrew Ng, and John Duchi (2016). "Vision-based classification of skin cancer using deep learning". In: *2015, conducted on Stanfords Machine Learning course (CS 229) taught*.

Kaymak, Ruya, Cagri Kaymak, and Aysegul Ucar (2020). "Skin lesion segmentation using fully convolutional networks: A comparative experimental study". In: *Expert Systems with Applications* 161, p. 113742.

Kaymak, Sertan, Parvaneh Esmaili, and Ali Serener (2018). "Deep learning for two-step classification of malignant pigmented skin lesions". In: *2018 14th Symposium on Neural Networks and Applications (NEUREL)*. IEEE, pp. 1–6.

Khan, Asifullah et al. (2020a). "A survey of the recent architectures of deep convolutional neural networks". In: *Artificial intelligence review* 53.8, pp. 5455–5516.

Khan, Muhammad Attique et al. (2020b). “Developed Newton-Raphson based deep features selection framework for skin lesion recognition”. In: *Pattern Recognition Letters* 129, pp. 293–303.

Khan, Muhammad Attique et al. (2021). “Skin lesion segmentation and multiclass classification using deep learning features and improved moth flame optimization”. In: *Diagnostics* 11.5, p. 811.

Khan, Muhammad Attique et al. (2022). “Skin lesion segmentation and classification: A unified framework of deep neural network features fusion and selection”. In: *Expert Systems* 39.7, e12497.

Khouloud, Sahib, Melouah Ahlem, and Slim Amel (2020). “Unet model for COVID-19 infected area segmentation in CT images”. In: *4th International Conference on Artificial Intelligence in Renewable Energetic*. IEEE.

Khouloud, Sahib et al. (2022a). “Improved Residual U-Net for COVID-19 Lung Infection Multi-Class Segmentation in CT Image”. In: *2022 International Conference of Advanced Technology in Electronic and Electrical Engineering (ICATEEE)*. IEEE, pp. 1–6.

Khouloud, Sahib et al. (2022b). “W-net and inception residual network for skin lesion segmentation and classification”. In: *Applied Intelligence*, pp. 1–19.

Kittler, Harold et al. (2002). “Diagnostic accuracy of dermoscopy”. In: *The lancet oncology* 3.3, pp. 159–165.

Kölbl, Laura (2017). “Deep Convolution Neural Networks for Image Analysis”. PhD thesis.

Koundal, Deepika and Bhisham Sharma (2019). “Advanced neutrosophic set-based ultrasound image analysis”. In: *Neutrosophic set in medical image analysis*. Elsevier, pp. 51–73.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2017). “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6, pp. 84–90.

LeCun, Yann, Yoshua Bengio, Geoffrey Hinton, et al. (2015). “Deep learning. nature, 521 (7553), 436-444”. In: *Google Scholar Google Scholar Cross Ref Cross Ref*.

LeCun, Yann et al. (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Howard and Yi-Ping Phoebe Chen (2014). “Skin cancer extraction with optimum fuzzy thresholding technique”. In: *Applied intelligence* 40.3, pp. 415–426.

Lin, Fengming et al. (2021). "Path aggregation U-Net model for brain tumor segmentation". In: *Multimedia Tools and Applications* 80.15, pp. 22951–22964.

Liu, Tianyi et al. (2019). "Towards understanding the importance of shortcut connections in residual networks". In: *Advances in neural information processing systems* 32.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Lux, F (2018). *Segmentation of Dense Cell Populations Using Convolutional Neural Networks*.

Maglogiannis, Ilias and Charalampos N Doukas (2009). "Overview of advanced computer vision systems for skin lesions characterization". In: *IEEE transactions on information technology in biomedicine* 13.5, pp. 721–733.

Mahbod, Amirreza et al. (2019). "Skin lesion classification using hybrid deep neural networks". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 1229–1233.

Mahbod, Amirreza et al. (2020). "Transfer learning using a multi-scale and multi-network ensemble for skin lesion classification". In: *Computer methods and programs in biomedicine* 193, p. 105475.

Mandache, Diana et al. (2018). "Basal cell carcinoma detection in full field OCT images using convolutional neural networks". In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, pp. 784–787.

Marques, Jorge S, Catarina Barata, and Teresa Mendonça (2012). "On the role of texture and color in the classification of dermoscopy images". In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, pp. 4402–4405.

Masood, Ammara (2016). "Developing improved algorithms for detection and analysis of skin cancer". PhD thesis.

Melanoma (2022). URL: `https://www.mayoclinic.org/diseases-conditions/melanoma/symptoms-causes/syc-20374884`.

Melissa Conrad Stöppler, MD (2021). *Medical Definition of Skin*. URL: `https://www.medicinenet.com/skin/definition.htm`.

Mendonça, Teresa et al. (2015). "Ph2: A public database for the analysis of dermoscopic images". In: *Dermoscopy image analysis*.

Menzies, Scott W (2012). "d Menzies method". In: *An Atlas of Dermoscopy*. CRC Press, pp. 128–133.

Menzies Method (2022). URL: `https://dermoscopedia.org/Menzies_Method`.

Minaee, Shervin et al. (2021). “Image segmentation using deep learning: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence*.

Naceur, Mostefa Ben (2020). “Deep Neural Networks for the segmentation and classification in Medical Imaging”. PhD thesis. Université Paris-Est; Université Mohamed Khider. Faculté des Sciences de ...

Nachbar, Franz et al. (1994). “The ABCD rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions”. In: *Journal of the American Academy of Dermatology* 30.4, pp. 551–559.

Nasr-Esfahani, Ebrahim et al. (2016). “Melanoma detection by analysis of clinical images using convolutional neural network”. In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, pp. 1373–1376.

Nguyen, Duy Khang et al. (2020). “Skin lesion segmentation based on integrating efficientnet and residual block into U-Net neural network”. In: *2020 5th International Conference on Green Technology and Sustainable Development (GTSD)*. IEEE, pp. 366–371.

Ninh, Quoc Cuong et al. (2019). “Skin lesion segmentation based on modification of SegNet neural networks”. In: *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*. IEEE, pp. 575–578.

Nowak, Leszek A, Maciej J Ogorzalek, and Marcin P Pawlowski (2012). “Pigmented network structure detection using semi-smart adaptive filters”. In: *2012 IEEE 6th International Conference on Systems Biology (ISB)*. IEEE, pp. 310–314.

Oktay, Ozan et al. (2018). “Attention u-net: Learning where to look for the pancreas”. In: *arXiv preprint arXiv:1804.03999*.

Oliveira, Roberta B et al. (2018). “Computational methods for pigmented skin lesion classification in images: review and future trends”. In: *Neural Computing and Applications* 29.3, pp. 613–636.

Owen, Caroline M and Nicholas R Telfer (2005). “Skin cancer”. In: *Medicine* 33.1, pp. 64–67.

Öztürk, Şaban and Umut Özkaya (2020). “Skin lesion segmentation with improved convolutional neural network”. In: *Journal of digital imaging* 33.4, pp. 958–970.

Pennisi, Andrea et al. (2016). “Skin lesion image segmentation using Delaunay Triangulation for melanoma detection”. In: *Computerized Medical Imaging and Graphics* 52, pp. 89–103.

Phan, Tran-Dac-Thinh et al. (2021). “Skin lesion segmentation by u-net with adaptive skip connection and structural awareness”. In: *Applied sciences* 11.10, p. 4528.

Posada Moyano, Roger (2021). "Using deep learning techniques for semantic segmentation of images". B.S. thesis. Universitat Politècnica de Catalunya.

Qamar, Saqib, Parvez Ahmad, and Linlin Shen (2021). "Dense encoder-decoder–based architecture for skin lesion segmentation". In: *Cognitive Computation* 13, pp. 583–594.

Qiu, Yuming et al. (2020). "Inferring skin lesion segmentation with fully connected CRFs based on multiple deep convolutional neural networks". In: *IEEE Access* 8, pp. 144246–144258.

Rahman, Zillur et al. (2021). "An approach for multiclass skin lesion classification based on ensemble learning". In: *Informatics in Medicine Unlocked* 25, p. 100659.

Rastgoo, Mojdeh et al. (2015). "Automatic differentiation of melanoma from dysplastic nevi". In: *Computerized Medical Imaging and Graphics* 43, pp. 44–52.

Ratul, Md Aminur Rab et al. (2020). "Skin lesions classification using deep learning based on dilated convolution". In: *BioRxiv*, p. 860700.

Riaz, Farhan et al. (2014). "Detecting melanoma in dermoscopy images using scale adaptive local binary patterns". In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, pp. 6758–6761.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Sadeghi, Maryam et al. (2011). "A novel method for detection of pigment network in dermoscopic images using graphs". In: *Computerized Medical Imaging and Graphics* 35.2, pp. 137–143.

Sandler, Mark et al. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520.

Sarkar, Rahul, Chandra Churh Chatterjee, and Animesh Hazra (2019). "Diagnosis of melanoma from dermoscopic images using a deep depthwise separable residual convolutional network". In: *IET Image Processing* 13.12, pp. 2130–2142.

Saurat, JH (2004). "Dermoscopy of pigmented lesions: a valuable tool in the diagnosis of melanoma". In: *Swiss Medical Weekly* 134.0708, pp. 83–90.

Schaefer, Gerald et al. (2014). "An ensemble classification approach for melanoma diagnosis". In: *Memetic Computing* 6.4, pp. 233–240.

Serener, Ali and Sertan Serte (2020). "Geographic variation and ethnicity in diabetic retinopathy detection via deeplearning". In: *Turkish Journal of Electrical Engineering and Computer Sciences* 28.2, pp. 664–678.

Shan, Pufang et al. (2020). "Automatic skin lesion segmentation based on FC-DPN". In: *Computers in Biology and Medicine* 123, p. 103762.

Sheha, Mariam A, Mai S Mabrouk, Amr Sharawy, et al. (2012). "Automatic detection of melanoma skin cancer using texture analysis". In: *International Journal of Computer Applications* 42.20, pp. 22–26.

Shrestha, Ajay and Ausif Mahmood (2019). "Review of deep learning algorithms and architectures". In: *IEEE access* 7, pp. 53040–53065.

Silveira, Margarida et al. (2009). "Comparison of segmentation methods for melanoma diagnosis in dermoscopy images". In: *IEEE journal of selected topics in signal processing* 3.1, pp. 35–45.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Singh, Vivek Kumar et al. (2020). "Segmentation and classification of multimodal medical images based on generative adversarial learning and convolutional neural networks". PhD thesis. Universitat Rovira i Virgili.

Situ, Ning et al. (2008). "Malignant melanoma detection by bag-of-features classification". In: *2008 30th annual international conference of the IEEE engineering in medicine and biology society*. IEEE, pp. 3110–3113.

Skin Cancer Fundation (2022). URL: `https://www.skincancer.org/skin-cancer-information/r`.

Snuverink, IAF (2017). "Deep learning for pixelwise classification of hyperspectral images". PhD thesis. Thesis. Delft, Netherlands: Faculty of Mechanical, Maritime and Materials ...

Soyer, H Peter et al. (2004a). "Three-point checklist of dermoscopy". In: *Dermatology* 208.1, pp. 27–31.

— (2004b). "Three-point checklist of dermoscopy". In: *Dermatology* 208.1, pp. 27–31.

Stanley, R Joe, William V Stoecker, and Randy H Moss (2007). "A relative color approach to color discrimination for malignant melanoma detection in dermoscopy images". In: *Skin Research and Technology* 13.1, pp. 62–72.

Stockfleth, Eggert, Ted Rosen, and Steven Schumaak (2010). *Managing skin cancer*. Springer Science & Business Media.

Szegedy, Christian et al. (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tan, Mingxing and Quoc Le (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR, pp. 6105–6114.

Tang, Peng et al. (2019). "Efficient skin lesion segmentation using separable-Unet with stochastic weight averaging". In: *Computer methods and programs in biomedicine* 178, pp. 289–301.

Tong, Xiaozhong et al. (2021). "ASCU-Net: attention gate, spatial and channel attention u-net for skin lesion segmentation". In: *Diagnostics* 11.3, p. 501.

Torralba, Antonio, Kevin P Murphy, and William T Freeman (2004). "Sharing features: efficient boosting procedures for multiclass object detection". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 2. IEEE, pp. II–II.

Tschandl, Philipp, Cliff Rosendahl, and Harald Kittler (2018). "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific data* 5.1, pp. 1–9.

Uličnỳ, Matej (2015). *Methods for increasing robustness of deep convolutional neural networks*.

Ünver, Halil Murat and Enes Ayan (2019). "Skin lesion segmentation in dermoscopic images with combination of YOLO and grabcut algorithm". In: *Diagnostics* 9.3, p. 72.

Vesal, Sulaiman, Nishant Ravikumar, and Andreas Maier (2018). "SkinNet: A deep learning framework for skin lesion segmentation". In: *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*. IEEE, pp. 1–3.

Walker, BN et al. (2019). "Dermoscopy diagnosis of cancerous lesions utilizing dual deep learning algorithms via visual and audio (sonification) outputs: Laboratory and prospective observational studies". In: *EBioMedicine* 40, pp. 176–183.

Walter, Fiona M et al. (2013). "Using the 7-point checklist as a diagnostic aid for pigmented skin lesions in general practice: a diagnostic validation study". In: *British Journal of General Practice* 63.610, e345–e353.

Wang, Guotai et al. (2020). "A noise-robust framework for automatic segmentation of COVID-19 pneumonia lesions from CT images". In: *IEEE Transactions on Medical Imaging* 39.8, pp. 2653–2663.

Weber, Griffin M, Kenneth D Mandl, and Isaac S Kohane (2014). "Finding the missing link for big biomedical data". In: *Jama* 311.24, pp. 2479–2480.

What Is Cancer? (2014). URL: https://web.archive.org/web/20140625220940/http://www.cancer.gov/cancertopics/cancerlibrary/what-is-cancer.

What's the difference between melanoma and non-melanoma skin cancer? (2016). URL: https://www.premiersurgical.com/request-an-appointment/.

Wighton, Paul et al. (2011). "Generalizing common tasks in automated skin lesion diagnosis". In: *IEEE Transactions on Information Technology in Biomedicine* 15.4, pp. 622–629.

Woo, Sanghyun et al. (2018). "Cbam: Convolutional block attention module". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19.

Xie, Fengying et al. (2020). "Skin lesion segmentation using high-resolution convolutional neural network". In: *Computer methods and programs in biomedicine* 186, p. 105241.

Yi, Terence J (2020). "Semantic Segmentation of Aerial Imagery using U-Nets". In.

Yuan, Yading (2017). "Automatic skin lesion segmentation with fully convolutional-deconvolutional networks". In: *arXiv preprint arXiv:1703.05165*.

Yueksel, M Emin and Murat Borlu (2009). "Accurate segmentation of dermoscopic images by image thresholding based on type-2 fuzzy logic". In: *IEEE Transactions on Fuzzy Systems* 17.4, pp. 976–982.

Zafar, Kashan et al. (2020). "Skin lesion segmentation from dermoscopic images using convolutional neural network". In: *Sensors* 20.6, p. 1601.

Zhang, Bin et al. (2021). "Opportunities and Challenges: Classification of Skin Disease Based on Deep Learning". In: *Chinese Journal of Mechanical Engineering* 34.1, pp. 1–14.

Zhou, Howard et al. (2008). "Spatially constrained segmentation of dermoscopy images". In: *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, pp. 800–803.

Zhou, Huiyu et al. (2011). "Gradient vector flow with mean shift for skin lesion segmentation". In: *Computerized Medical Imaging and Graphics* 35.2, pp. 121–127.

Zhou, Tongxue, Stéphane Canu, and Su Ruan (2021). "Automatic COVID-19 CT segmentation using U-Net integrated spatial and channel attention mechanism". In: *International Journal of Imaging Systems and Technology* 31.1, pp. 16–27.