



PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
Mohamed Khider University BISKRA
Faculty of Exact Sciences, Natural and Life Sciences
Computer Science Department

Dissertation

Presented to obtain the academic master's degree in

Computer Science

Option: **Software Engineering and Distributed Systems**

Title

Fusing Transformers and CNNs for medical image segmentation in skin cancer detection

Presented by:

- LAIADI Merzaka

-FERHATI Selsabil

Defended on June , 2025: In front of the jury composed of:

- TELLI ABDEL MOUTIA

President

- BENTRAH AHLEM

Supervisor

- LAANANI SADEK

Examiner

Academic year : **2024/2025**

Contents

List of Figures	iii
List of Tables	iv
Acknowledgement	v
Abstract	vi
ملخص	vii
Résumé	viii
General Introduction	ix
1 Skin Cancer Diagnosis: From Clinical Imaging to Deep Learning	1
1.1 Introduction	1
1.2 Skin Anatomy	1
1.3 Skin Cancer	2
1.4 Types of skin cancer	3
1.4.1 Basal cell carcinoma	3
1.4.2 Squamous cell carcinoma	3
1.4.3 Melanoma	3
1.4.4 Merkel cell carcinoma	3
1.5 Medical imaging	4
1.6 Key Medical Imaging Modalities for Skin	4
1.6.1 Dermoscopy	4
1.6.2 Reflectance Confocal Microscopy (RCM)	5
1.6.3 Optical Coherence Tomography (OCT)	6
1.6.4 High-Frequency Ultrasound (HFUS)	6
1.7 Difficulty of detection skin cancer	7
1.8 Deep Learning for skin cancer detection	8
1.9 Conclusion	8
2 Deep learning	9
2.1 Introduction	9
2.2 Machine learning (ML)	9
2.2.1 Types of Machine learning Algorithms	11
2.2.1.1 Supervised Learning	11

2.2.1.2.	Unsupervised Learning	12
2.2.1.3	Semi-Supervised Learning.....	12
2.2.1.4	Reinforcement Learning	13
2.3	Deep learning	13
2.3.1	Transfer Learning.....	14
2.3.2	Artificial Neural Networks.....	14
2.3.3	Convolution Neural Networks (CNNs)	15
2.3.3.1	The Basics of Convolutional Neural Network	16
2.3.3.2	Activation Functions.....	17
2.3.3.3	Example of CNNs based model	18
2.3.3.4	Advantage of CNNs.....	19
2.3.3.5	Limitations of CNNs.....	20
2.3.4	Transformer.....	20
2.3.4.1	Example of Transformer-based models	22
2.3.4.2	Advantages of transformer.....	22
2.3.4.3	Limitations of Transformer	23
2.4	Advanced Training Techniques	24
2.4.1	Data Augmentation	24
2.4.2	Regularization.....	25
2.5	Evaluation Metrics	25
2.5.1	Dice Coefficient (Dice Similarity Coefficient - DSC)	25
2.5.2	Intersection over Union (IoU) / Jaccard Index	26
2.5.3	Accuracy.....	26
2.5.4	Precision	26
2.5.5	Recall	27
2.5.6	F1 Score	27
2.5.7	Hausdorff distance	27
2.5.8	Loss Function.....	28
2.6	Related Works for skin cancer detection with deep learning	28
2.6.1	Medical Image Segmentation Based on CNNs	28
2.6.2	Vision Transformer	29
2.6.3	Medical Image Segmentation Based on Transformers and CNNs.....	29

2.7	Conclusion	31
3	<i>Design, Implementation and experimental results</i>	32
3.1	Introduction	32
3.2	Design and conception	32
3.2.1.	Model description	32
3.2.2	Global design	32
3.2.3	Model architecture	33
3.2.4	Detailed architecture description.....	34
3.2.5	UML presentations	35
3.3	Implementation	38
3.3.1	Environments and developing tools	38
3.3.2	Used Image dataset details information	41
3.3.3	Implementation details	42
3.4	Model Training	49
3.4.1	Loss Function.....	49
3.4.2	Optimization Strategy	50
3.4.3	Evaluation Metrics	50
3.4.4	Training loop.....	51
3.5	Experimental Results	52
3.5.1	Performance of Our Model	52
3.5.2	Model prediction results	55
3.5.3	Model prediction results analysis.....	56
3.5.4	Comparison with State-of-the-Art Models.....	56
3.6	Discussion	57
3.7	Conclusion	57
	<i>General conclusion</i>	58
	<i>References</i>	59

List of Figures

1.1	Anatomy of skin.	2
1.2	Example of skin cancer anatomy.....	2
1.3	Types of skin cancer	4
1.4	Dermoscopy machine.	5
1.5	Reflectance Confocal Microscopy machine	5
1.6	Reflectance Confocal Microscopy machine	6
1.7	High-Frequency Ultrasound machine.....	6
2.1	Process of machine learning [25]	10
2.2	Components of learning process.....	10
2.3	Types of Machine learning	11
2.4	Supervised learning Workflow	11
2.5	Unsupervised Learning.....	12
2.6	Semi-Supervised Learning	13
2.7	Reinforcement Learning.....	13
2.8	Deep learning, a subset of a subset of AI	14
2.9	Basic Elements of Artificial Neural Network.....	15
2.10	An overview of CNN architecture and the training process.....	15
2.11	Pooling types.....	16
2.12	ReLU Activation Function Plot.....	17
2.13	Sigmoid Function.....	18
2.14	The Transformer- model architecture.....	21
2.15	Deep learning models used for skin cancer detection	27
3.1	Global design of our skin cancer segmentation system.....	32
3.2	Model detailed architecture.	32
3.3	The state transition diagram of our system.....	35
3.4	The sequence diagram of our system.	36
3.5	Examples of dermoscopic images from dataset ISIC 2017.	39
3.6	Illustration of necessary package for data preparation.	40
3.7	Dataset preprocessing parameter representation.	40
3.8	Data preprocessing illustration.	41
3.9	Storing the resulting image and mask arrays as .npy files.....	41
3.10	Loading images and masks from NumPy files.....	41
3.11	Images and masks transformations.....	42
3.12	Example of a skin lesion and its corresponding mask.....	42
3.13	Import DeiT-Small transformer.....	43
3.14	Importing of the pretrained ResNet 34.....	43
3.15	Illustration of Fusion Module (BiFusion block).....	44
3.16	Illustration of UpSampling phase.....	45

3.17 Illustration of DoubleConv phase.	45
3.18 Illustration of Conv 1X1 phase.	46
3.19 Training phase of our deep model.....	48
3.20 Loss Tracking.....	50
3.21 Accuracy Monitoring	50
3.21 Dice Score Evaluation.....	51
3.23 IoU Metric Evaluation.....	51
3.24 Qualitative Results	52

List of Tables

2.1	Deep learning models used for skin cancer detection.....	37
3.1	Illustration of tools environment	37
3.2	Illustration of packages and APIs.....	38
3.3	Illustration of used evaluation metrics.....	47
3.4	Comparison of skin lesion segmentation performance of different networks on ISIC 2017..	53

Acknowledgement

*In the name of **Allah**, the Most Gracious, the Most Merciful.*

All praise and thanks be to Allah, the Most Compassionate, the Most Merciful, for His countless blessings upon us. He granted us the strength, patience, and guidance to complete this work. Without His divine support, this achievement would not have been possible.

*We express our heartfelt gratitude to **Dr. Bentrah Ahlem** for giving us the opportunity to be among her students. Her valuable guidance, insightful remarks, and constructive criticism played a vital role in shaping this work. Her dedication inspired us to pursue excellence and originality in our research.*

*We also sincerely thank **the members of the jury** for reviewing our graduation project and enriching it with their feedback and suggestions.*

*Our deep appreciation goes to all **our teachers at the University of Mohamed Khider Biskra** for their continuous efforts in building our knowledge and supporting our academic journey.*

*We are especially grateful to **our beloved families** for their endless love, prayers, encouragement, and support. Their unwavering presence was our true foundation throughout this endeavor.*

Finally, we extend our sincere thanks to everyone who contributed to the completion of this work, near or far. May Allah reward them all abundantly.

Abstract

Medical image segmentation is one of the most challenging aspects of image analysis, as accurate segmentation of skin lesions is a key step for early and effective diagnosis of skin cancer. This field has witnessed significant progress thanks to advances in convolutional neural networks (CNNs), which have demonstrated high efficiency in extracting local features. However, their ability to capture long-term dependencies within images is limited. Transformer architectures, on the other hand, have demonstrated superiority in modeling global context, but they lack inductive biases such as autotranslation and localization, which reduces their effectiveness, especially when dealing with limited medical data. A promising solution to overcome these challenges is the combination of Transformers and CNNs within a hybrid architecture that leverages the advantages of both. In this context, this work proposes a two-branch model that combines ResNet in the CNN branch to extract local features and DeiT in the Transformer branch to capture global dependencies. The outputs of the two branches are combined through the BiFusion module, which integrates local and global information to enhance segmentation accuracy. When evaluated on the ISIC 2017 dataset, the model outperformed traditional CNN or Transformer-only based models, confirming the effectiveness of this combination in improving skin lesion segmentation.

Keywords: Medical image segmentation, skin cancer, convolutional neural networks, transformers, hybrid architecture, segmentation accuracy.

ملخص

تُعد تجزئة الصور الطبية من أصعب التحديات في تحليل الصور، حيث إن التجزئة الدقيقة لأفات الجلد تُعد خطوة أساسية للتشخيص المبكر والفعال لسرطان الجلد. وقد شهد هذا المجال تطورًا ملحوظًا بفضل التقدم في الشبكات العصبية التلافيفية (CNNs)، التي أظهرت كفاءة عالية في استخراج السمات المحلية. ومع ذلك، فإن قدرتها محدودة في التقاط التبعيات طويلة المدى داخل الصور. في المقابل، أظهرت بنى المحولات (Transformers) تفوقًا في نمذجة السياق العالمي، لكنها تفنقر إلى التحيزات الاستقرائية مثل الترجمة التلقائية والمحلية، مما يقلل من فعاليتها خاصة عند التعامل مع بيانات طبية محدودة. ومن بين الحلول الواعدة لتجاوز هذه التحديات، يبرز الدمج بين المحولات و CNNs ضمن هيكليّة هجينة تسمح بالاستفادة من مزايا كل منهما. في هذا السياق، يقترح هذا العمل نموذجًا ثنائي الفروع يجمع بين ResNet في فرع CNN لاستخلاص السمات المحلية، و DeiT في فرع Transformer لالتقاط التبعيات العالمية. ويتم دمج مخرجات الفرعين من خلال وحدة BiFusion، التي تدمج المعلومات المحلية والعالمية بشكل تكاملي لتعزيز دقة التجزئة. عند تقييم النموذج على مجموعة بيانات ISIC 2017، أظهر تفوقًا على النماذج التقليدية المعتمدة على CNN أو Transformer فقط، مما يؤكد فعالية هذا الدمج في تحسين تجزئة آفات الجلد.

الكلمات المفتاحية: تجزئة الصور الطبية، سرطان الجلد، الشبكات العصبية التلافيفية، المحولات، نموذج هجين، دقة التجزئة.

Résumé

La segmentation des images médicales est l'un des aspects les plus complexes de l'analyse d'images, car une segmentation précise des lésions cutanées est essentielle pour un diagnostic précoce et efficace du cancer de la peau. Ce domaine a connu des progrès significatifs grâce aux avancées des réseaux de neurones convolutifs (CNN), qui ont démontré une grande efficacité dans l'extraction de caractéristiques locales. Cependant, leur capacité à capturer les dépendances à long terme au sein des images est limitée. Les architectures de type Transformer, en revanche, ont démontré leur supériorité dans la modélisation du contexte global, mais elles manquent de biais inductifs tels que l'autotraduction et la localisation, ce qui réduit leur efficacité, notamment lorsqu'il s'agit de données médicales limitées. Une solution prometteuse pour surmonter ces défis est la combinaison de Transformers et de CNN au sein d'une architecture hybride exploitant les avantages des deux. Dans ce contexte, ce travail propose un modèle à deux branches combinant ResNet dans la branche CNN pour extraire les caractéristiques locales, et DeiT dans la branche Transformer pour capturer les dépendances globales. Les sorties des deux branches sont combinées via le module BiFusion, qui intègre les informations locales et globales pour améliorer la précision de la segmentation. Lorsqu'il a été évalué sur l'ensemble de données ISIC 2017, le modèle a surpassé les modèles traditionnels basés uniquement sur CNN ou Transformer, confirmant l'efficacité de cette combinaison pour améliorer la segmentation des lésions cutanées.

Mots-clés : Segmentation d'images médicales, cancer de la peau, réseaux de neurones convolutifs, transformeurs, architecture hybride, précision de la segmentation.

General Introduction

Medical image segmentation is a critical area within medical image analysis and plays a fundamental role in computer-aided diagnosis, monitoring, intervention, and treatment planning. The primary objective of medical image segmentation is to delineate regions of interest, such as organs or lesions, within medical images. Accurate segmentation enables various analytical and quantitative methods that address diverse clinical needs and assist physicians in making more precise diagnoses.

Currently, medical image segmentation techniques are widely applied in numerous domains, including cardiac segmentation, gland segmentation, skin lesion segmentation, and brain tumor detection.

With the advent of deep learning, convolutional neural networks (CNNs) have become the dominant approach in many medical image segmentation tasks. Despite their success, CNNs exhibit inherent limitations, particularly in capturing long-range dependencies due to their local receptive fields.

To overcome this, deeper networks or larger convolutional kernels are often used, which substantially increase the number of model parameters and complicate training. Furthermore, increasing network depth can cause gradient vanishing problems and slow convergence rates.

Transformers, initially developed for sequence-to-sequence modeling in natural language processing, have recently gained significant attention in computer vision. The Vision Transformer (ViT), a purely self-attention-based architecture, demonstrated competitive performance on large-scale image recognition benchmarks, provided it was pretrained on extensive datasets.

Transformers excel at modeling global contextual relationships but face challenges in capturing fine-grained details, which are crucial in medical image analysis.

To leverage the strengths of both CNNs and transformers, hybrid architectures have been proposed that combine local feature extraction capabilities of CNNs with the global dependency modeling of transformers.

In this dissertation, we propose a dual-branch deep learning model comprising a ResNet-based CNN branch for local feature extraction and a DeiT-based transformer branch for capturing global dependencies. The outputs of these branches are fused through a BiFusion module designed to integrate local and global information, thereby enhancing segmentation accuracy.

Evaluation on the ISIC 2017 dataset demonstrates that our hybrid model outperforms traditional CNN- or transformer-only approaches, confirming the effectiveness of this integration for skin lesion segmentation.

This work is organized into three main chapters:

- **The first chapter** presents the medical background, discussing skin cancer types, diagnostic challenges, and the role of artificial intelligence in modern dermatology.
- **The second chapter** introduces foundational concepts in deep learning, emphasizing techniques relevant to image segmentation.
- **The third chapter** details the practical implementation of the proposed model, including dataset preparation, model architecture, training procedures, evaluation metrics, and interpretation of results.

Through this work, we aim to contribute to the advancement of AI-assisted dermatology by exploring methods for accurate and efficient skin lesion segmentation, with the ultimate goal of supporting earlier diagnosis and improving patient outcomes.

Chapter 1

Skin Cancer Diagnosis: From Clinical Imaging to Deep Learning

1.1 Introduction

Recent developments in medical imaging have significantly improved dermatological care. These technologies offer clearer, high-resolution images of the outer and inner layers of the skin, allowing clinicians to detect subtle changes earlier and assess skin conditions more accurately. This progress supports better diagnosis of skin diseases, including the early identification of cancerous lesions and the development of personalized treatment plans.

This chapter begins by reviewing the basic structure of the skin, followed by a summary of the main types of skin cancer. The key imaging techniques, such as dermoscopy, reflectance confocal microscopy (RCM), optical coherence tomography (OCT), and high-frequency ultrasound (HFUS), are introduced, which are changing the way skin lesions are detected and examined. The chapter also addresses the current challenges in accurately diagnosing skin cancer. Finally, it discusses the growing impact of deep learning in automating skin lesion segmentation, which helps improve the speed and accuracy of skin cancer detection.

1.2 Skin Anatomy

The skin is an organ that provides the outer protective wrapping for all the body parts. It is the largest organ in the body. It is a waterproof, airtight and flexible barrier between the environment and internal organs. It keeps the internal environment of our body stable. The skin is divided into 3 layers, the epidermis, the dermis and the subcutaneous layer [1] (see figure 2.1).

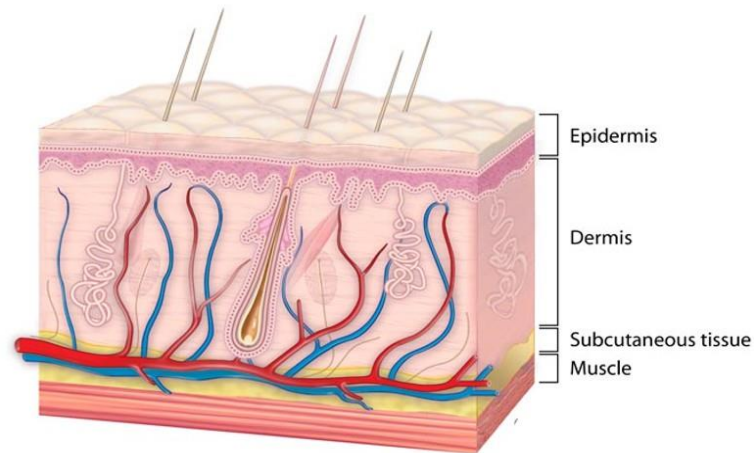


Figure 1.1: Anatomy of skin [1].

1.3 Skin Cancer

Skin cancer is the most common type of cancer worldwide, and its occurrence is constantly on the rise. It develops due to the uncontrolled growth of abnormal cells in the epidermis, the outermost skin layer, because of unrepaired DNA damage. This damage leads to mutations that create excessive cell growth, ultimately forming malignant tumors. Fortunately, if caught early enough, the great majority of skin cancers are curable and can be easily managed [2] [3] [4].

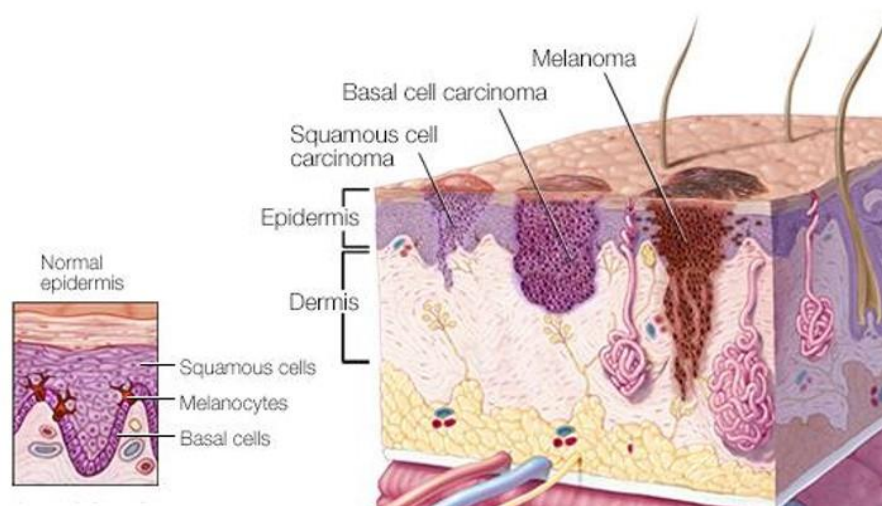


Figure 1.2: Example of skin cancer anatomy [5].

There are many types of skin cancer, each of which can look different on the skin. The following section discusses the common types of skin cancer.

1.4 Types of skin cancer

The type is defined by the location where the cancer begins. There are four main kinds: basal cell carcinoma, squamous cell carcinoma, melanoma, and Merkel cell carcinoma.

1.4.1 Basal cell carcinoma

Basal cell carcinoma begins in the basal cells, found in the skin's outermost layer. It often develops in areas exposed to the sun, such as the scalp, face, ears, neck, shoulders, and back. This cancer can appear as a waxy or pearly bump or a lesion that could be flat and flesh-colored or brown and scar-like. Alternatively, it could be a sore that bleeds or scabs and then heals, only to return again [6].

1.4.2 Squamous cell carcinoma

This type of skin cancer often appears in similar areas to basal cell carcinoma. However, unlike basal cell carcinoma, it starts in the skin's squamous cells. It can look like a firm, red nodule or a flat lesion with a scaly, crusted surface or a sore or patch that does not heal [6].

1.4.3 Melanoma

Melanoma occurs in the cells that give skin its color. It can develop in an existing mole or in normal skin. It's often found on the face or torso in men, while it frequently occurs on the lower legs in women. However, it can strike anywhere on the body in areas exposed to the sun and in places not subject to UV radiation. Symptoms include any spot or mole that is asymmetrical, has an irregular border, has changes in color, is more than one-quarter inch in diameter, or is generally changing or evolving [6].

1.4.4 Merkel cell carcinoma

Merkel cell carcinoma occurs in the Merkel cells located on the skin's surface. It is typically found in areas exposed to the sun and appears as a painless lump that could be flesh-colored, red, blue, or purple. It could be up to the size of a dime and often grows quickly. People older than 50 or immunosuppressed are at higher risk [6].

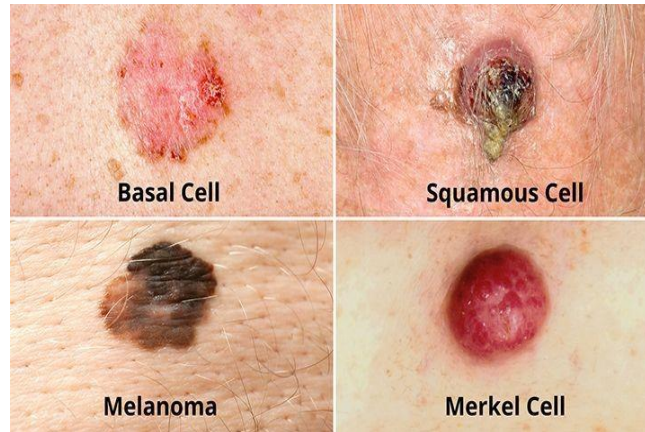


Figure 1.3: Types of skin cancer [7].

1.5 Medical imaging

Medical imaging is the process of visual representation of the structure and function of different tissues and organs of the human body for clinical purposes and medical science for detailed study of normal and abnormal anatomy and physiology of the body. Medical imaging techniques are used to show internal structures under the skin and bones, as well as to diagnose abnormalities and treat diseases. It is an important part of biological imaging [8]. It is widely used in patient care to diagnose disease, to plan treatment, and to monitor response to treatment [9]. More recently, increasing attention has been given to the application of imaging technologies for skin evaluation. A variety of techniques are currently employed to examine the skin. In the following section, we present the key medical imaging modalities used for skin assessment.

1.6 Key Medical Imaging Modalities for Skin

Medical imaging modalities for skin encompass a range of advanced techniques that allow detailed visualization and assessment of skin structure, lesions, and diseases with varying depths and resolutions. These modalities include:

1.6.1 Dermoscopy

Dermoscopy (also known as dermatoscopy or epiluminescence microscopy) is a non-invasive skin imaging technique that allows visualization of subsurface skin structures not visible to the naked eye. It enhances diagnostic accuracy for pigmented skin lesions and is especially valuable in the early detection of melanoma and other forms of skin cancer [10].



Figure 1.4: Dermoscopy machine [11].

1.6.2 Reflectance Confocal Microscopy (RCM)

Reflectance Confocal Microscopy (RCM) is a non-invasive, high-resolution imaging technique that enables real-time visualization of skin at nearly cellular resolution. It uses a low-power laser and relies on variations in the reflectance of light by different skin structures to create grayscale images of the epidermis and superficial dermis. RCM is particularly useful for diagnosing skin cancers, especially melanoma, without the need for a biopsy [12].



Figure 1.5: Reflectance Confocal Microscopy machine [13].

1.6.3 Optical Coherence Tomography (OCT)

Optical coherence tomography (OCT) is a microscopic imaging technique, which magnifies the surface of a skin lesion using near light. Used in conjunction with clinical or dermoscopic examination of suspected skin cancer, or both, OCT may offer additional diagnostic information compared to other technologies [14].



Figure 1.6: Reflectance Confocal Microscopy machine [15].

1.6.4 High-Frequency Ultrasound (HFUS)

High-Frequency Ultrasound (HFUS) is a non-invasive imaging technique that uses ultrasound waves typically above 20 MHz to produce high-resolution images of superficial tissues such as skin. It is widely used in dermatology for assessing skin tumors, inflammatory conditions, and structural skin changes due to its ability to resolve fine details in the epidermis and dermis [16].



Figure 1.7: High-Frequency Ultrasound machine [17].

1.7 Difficulty of detection skin cancer

Detecting skin cancer, particularly in its early stages, presents multiple challenges that impact timely diagnosis and treatment outcomes. Below is a comprehensive overview of the main difficulties involved:

- **Early-Stage Subtlety:** Skin cancers, especially melanoma, can be difficult to detect early because their initial appearance may be subtle and easily confused with benign lesions. Early melanomas often lack distinctive features visible to the naked eye, making clinical diagnosis challenging without specialized tools [18].
- **Dependence on Clinician Experience:** Diagnostic accuracy heavily depends on the clinician's experience and training in dermoscopy and skin cancer recognition. Studies show clinicians with more dermoscopy training have significantly higher sensitivity and specificity in melanoma diagnosis compared to those without training [19].
- **Variability in Sensitivity and Specificity:** Visual examination alone has a melanoma detection sensitivity around 60-76, which can lead to missed diagnoses or unnecessary biopsies. The specificity can also vary widely, sometimes leading to over-excision of benign lesions [19].
- **Difficulties in Certain Populations:** Skin cancer in patients with darker skin tones is often diagnosed at later stages, when it is harder to treat, due to less obvious pigmentation changes and lower awareness [20].
- **High Costs of Skin Cancer Detection:** Detecting skin cancer is costly due to multiple steps like clinical exams, imaging (e.g., dermoscopy), and biopsies. These include doctor visits, lab tests, and pathology services, with total costs ranging from hundreds to thousands of dollars. Delayed diagnosis often results in advanced stages requiring more intensive and expensive treatment, increasing the financial burden on both patients and healthcare systems [21].
- **Manual segmentation of skin lesions:** Manually interpreting dermoscopic images is laborious, requiring significant time and expertise from dermatologists. The growing number of cases further compounds this challenge, making it difficult to meet the demand for timely and accurate diagnoses³. Manual segmentation of skin lesions in dermoscopic images is also subject to inter-observer variability, leading to inconsistent diagnostic accuracy among clinicians [22].

1.8 Deep Learning for skin cancer detection

To address these challenges, automated segmentation systems have emerged as essential tools to standardize the diagnostic process, alleviate the burden on healthcare professionals, and improve diagnostic precision [22]. Skin cancer detection in particular serves as an appealing application for AI, given that diagnoses often hinge on the subjective visual interpretation of clinical and dermoscopic images. AI-assisted diagnosis promises several advantages. For instance, AI could improve access to specialist-level expertise. The scarcity of dermatologists is a serious problem in many regions, often leading to protracted waiting times for specialist appointment. In addition, there is growing optimism that AI-based systems might offer greater consistency and higher accuracy than human experts [23]. Such systems not only enhance efficiency but also deliver consistent, objective results, minimizing the subjectivity inherent in human interpretation. Automated segmentation is particularly valuable in large-scale screening programs, where rapid and accurate processing of dermoscopic images is critical. By taking advantage of deep learning techniques, these systems can learn from large datasets and continuously improve their performance, ultimately aiding in the early detection of melanomas. As a result, the development of robust automated skin lesion segmentation tools is crucial for advancing skin cancer diagnostics and empowering healthcare professionals to provide faster, more accurate care [23].

1.9 Conclusion

This chapter presented the medical foundation of our project, which focuses on the detection of skin cancer one of the most prevalent and potentially life-threatening malignancies if not diagnosed and treated at an early stage. We began by reviewing the basic anatomy of the skin, the classification of skin cancers, and the primary imaging modalities used in clinical dermatology. The diagnostic challenges associated with skin cancer, such as variability in lesion appearance and reliance on expert interpretation, were also highlighted. Subsequently, we introduced the transformative role of artificial intelligence, particularly deep learning, in addressing these challenges. We emphasized its growing use in automating and enhancing the accuracy of skin lesion analysis. Ultimately, the integration of AI into dermatological workflows holds great promise for improving diagnostic precision, reducing delays, and expanding access to expert-level evaluation. In the following chapter, we delve into the technical foundations of deep learning, focusing on convolutional neural networks, vision transformers, and their fusion strategies for effective skin lesion segmentation.

Chapter 2

Deep learning

2.1 Introduction

In recent years, Artificial Intelligence (AI) has emerged as a powerful tool in the field of medical image analysis, offering new possibilities for accurate, efficient, and automated disease diagnosis. Among various AI techniques, deep learning has shown remarkable success in understanding complex patterns in medical imaging data, particularly through convolutional neural networks (CNNs) and transformer-based architectures. This chapter provides an in-depth exploration of the fundamental concepts of AI and deep learning as they relate to skin cancer detection. We begin by introducing the key components of neural networks, followed by a discussion of prominent deep learning architectures used in image segmentation and classification. Special attention is given to hybrid models that combine CNNs and transformers, which have recently demonstrated superior performance in medical image segmentation tasks. These models represent the technological foundation upon which our proposed model is built.

2.2 Machine learning (ML)

Machine Learning is an application of Artificial Intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed [24]. The basic machine learning process can be divided into three parts:

- 1.Data input:** Past data or information is utilized as a basis for future decision-making.
- 2. Abstraction:** The input data is represented in a broader way through the underlying algorithm.
- 3.Generalization:** The abstracted representation is generalized to form a framework for making decisions.



Figure 2.1: Process of machine learning [25].

The learning process, whether by a human or a machine, can be divided into four components namely:

- data storage
- abstraction
- generalization
- evaluation

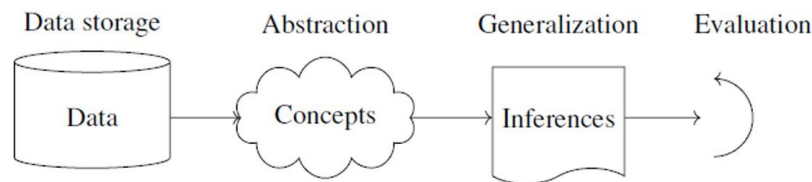


Figure 2.2: components of learning process.

- **Data storage** : Facilities for storing and retrieving huge amounts of data are an important component of the learning process. Humans and computers alike utilize data storage as a foundation for advanced reasoning. In a human being, the data is stored in the brain and data is retrieved using electrochemical signals. Computers use hard disk drives, flash memory, random access memory and similar devices to store data and use cables and other technology to retrieve data.
- **Abstraction** : is the process of extracting knowledge about stored data. This involves creating general concepts about the data as a whole. The creation of knowledge involves application of known models and creation of new models. The process of fitting a model to a dataset is known as training. When the model has been trained, the data is transformed into an abstract form that summarizes the original information.
- **Generalization** : The term generalization describes the process of turning the knowledge about stored data into a form that can be utilized for future action. These actions are to be carried out on tasks that are similar, but not identical, to those what have been seen before. In generalization, the goal is to discover those properties of the data that will be most relevant to future tasks.

- **Evaluation** : It is the process of giving feedback to the user to measure the utility of the learned knowledge. This feedback is then utilised to effect improvements in the whole learning process [26].

2.2.1 Types of Machine learning Algorithms

Machine Learning relies on different algorithms to solve data problems. The four main types of machine learning are:

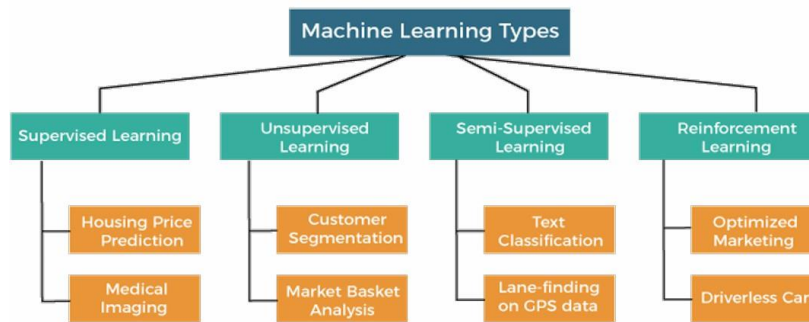


Figure 2.3: Types of Machine learning [27].

2.2.1.1 Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification.

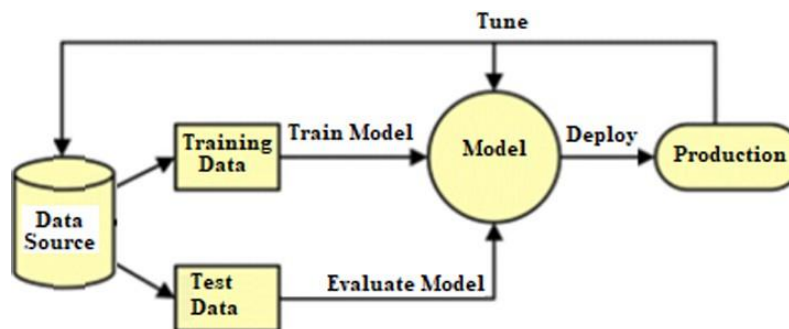


Figure2.4: Supervised learning Workflow [28].

2.2.1.2. Unsupervised Learning

These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. [28] it is seems much harder: the goal is to have the computer learn how to do something that we don't tell it how to do! [29] It uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without human intervention. [27] Algorithms are left to their own devices to discover and present the interesting structure in the data. The unsupervised learning algorithms learn few features from the data. When new data is introduced, it uses the previously learned features to recognize the class of the data. It is mainly used for clustering and feature reduction [28].

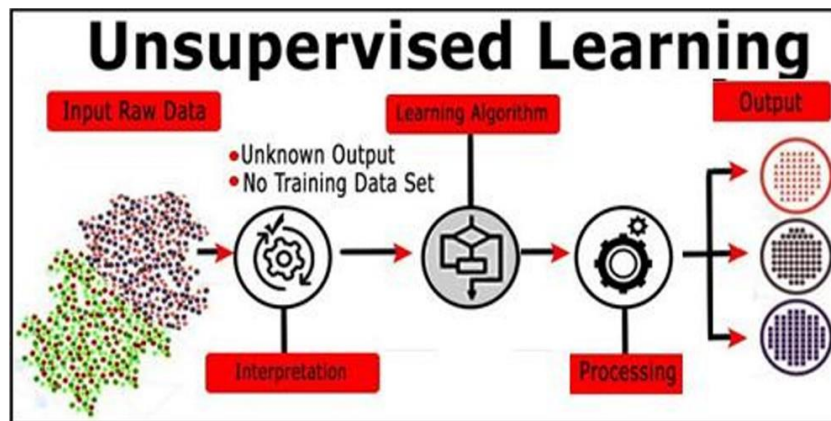


Figure 2.5: Unsupervised Learning [28].

2.2.1.3 Semi-Supervised Learning

Semi-supervised machine learning is a combination of supervised and unsupervised machine learning methods. It can be fruitful in those areas of machine learning and data mining where the unlabeled data is already present and getting the labeled data is a tedious process. With more common supervised machine learning methods, you train a machine learning algorithm on a “labeled” dataset in which each record includes the outcome information [28].

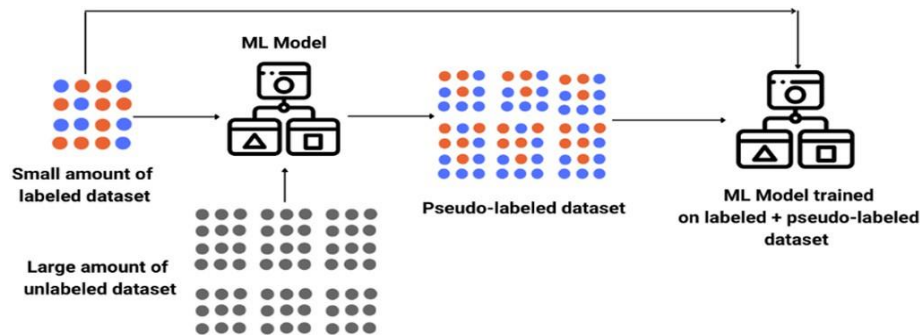


Figure 2.6: Semi-Supervised Learning [27].

2.2.1.4 Reinforcement Learning

Reinforcement learning is a “trial-and-error” learning approach to constantly interact with the environment to obtain the best strategy by maximizing the reward. “State, action, reward” are the three key elements of reinforcement learning. The model observes the decision outcome at each step, leading to the next decision to win the final goal. The game and robots are the most widely used areas of this method at present. [30]



Figure 2.7: Reinforcement Learning [27].

2.3 Deep learning

Deep learning, also known as deep neural networks, is a subfield of machine learning, which is a subset of AI, as shown in Figure 2.8. Deep learning takes inspiration from how the human brain works [31].

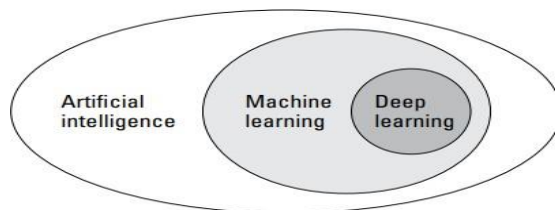


Figure 2.8: Deep learning, a subset of a subset of AI [31].

Deep learning is making major advances in solving problems that have resisted the best attempts of the artificial intelligence community for many years. It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. In addition to beating records in image recognition and speech recognition, it has beaten other machine learning techniques at predicting the activity of potential drug molecules, analyzing particle accelerator data, reconstructing brain circuits, and predicting the effects of mutations in non-coding DNA on gene expression and disease [32].

2.3.1 Transfer Learning

Transfer learning with pretrained neural network models is a frequent notion in deep learning. When training on a new goal, such as picture segmentation of medical volumes, neural networks that have been trained on another task, such as a natural image classification data set, can be utilized to initialize the network weights. The theory behind this is that the earliest layers of neural networks learn comparable notions to recognize fundamental structures like blobs and edges for various tasks or datasets. When employing pre-trained models, these concepts do not need to be re-trained [33].

2.3.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational modeling tools that have recently emerged and found extensive acceptance in many disciplines for modeling complex real-world problems. [34] An Artificial Neural Network (ANN) is a mathematical model that tries to simulate the structure and functionalities of biological neural networks. Basic building block of every artificial neural network is artificial neuron, that is, a simple mathematical model (function). Such a model has three simple sets of rules: multiplication, summation and activation. At the entrance of artificial neuron the inputs are weighted what means that every input value is multiplied with individual weight. In the middle section of artificial neuron is sum function that sums all weighted inputs and bias. At the exit of artificial neuron the sum of previously weighted inputs and bias is passing through activation function that is also called transfer function [35].

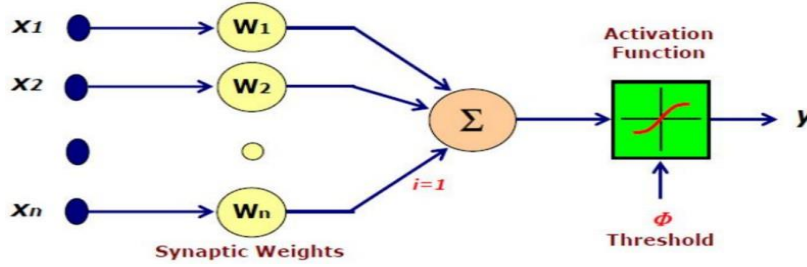


Figure 2.9: Basic Elements of Artificial Neural Network [36].

As shown in Figure 2.9, we have various inputs (X_1, X_2, \dots, X_n) and corresponding weights (W_1, W_2, \dots, W_n). We calculate the weighted sum of some of these inputs and then pass them through an activation function [29], which is nothing more than a threshold value. Threshold Value: The threshold value decides whether a neuron fires or not.

2.3.3 Convolution Neural Networks (CNNs)

One of the most impressive forms of ANN architecture is that of the Convolutional Neural Network (CNN) [37]. CNN was first introduced in the 1960s and has shown promising performance results in computer vision. CNN has become the most representative neural network in deep learning. CNN has been utilized to solve complicated visual tasks with high computation and is mainly used in image classification, segmentation, object detection, video processing, natural language processing, and speech recognition [38].

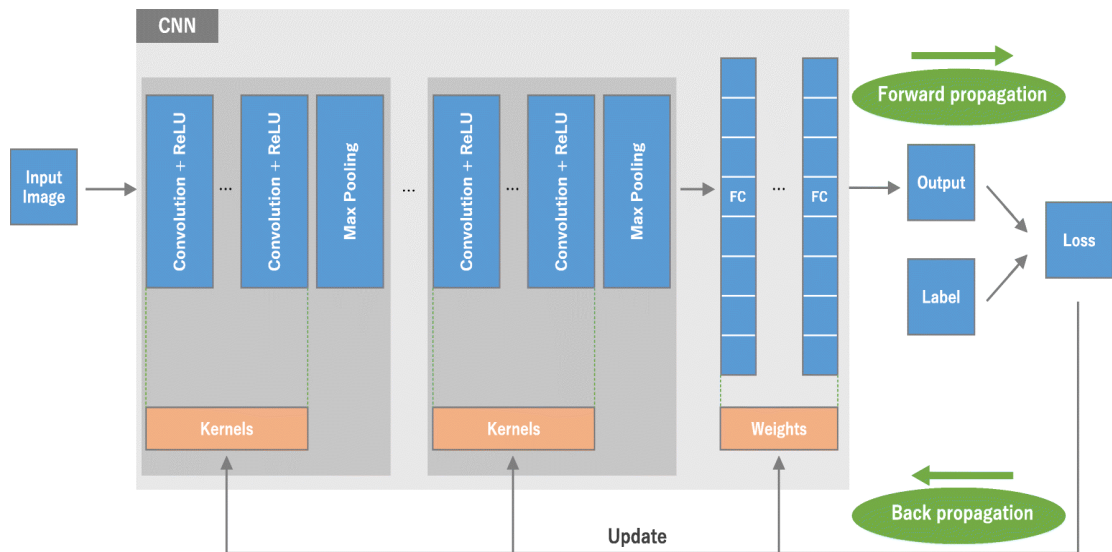


Figure 2.10: An overview of CNN architecture and the training process [39].

The CNN architecture includes several building blocks, such as convolution layers, pooling layers, and fully connected layers. A typical architecture consists of repetitions of a stack of several convolution layers and a pooling layer, followed by one or more fully connected layers. The step where input data are transformed into output through these layers is called forward propagation (Figure 2.10) [39].

2.3.3.1 The Basics of Convolutional Neural Network

A. Convolutional Layer

A convolution layer is a fundamental component of the CNN architecture that performs feature extraction, which typically consists of a combination of linear and nonlinear operations:

- Convolution is a specialized type of linear operation used for feature extraction, where a small array of numbers, called a kernel, is applied across the input, which is an array of numbers, called a tensor.
- The outputs of a linear operation such as convolution are then passed through a nonlinear activation function. Although smooth nonlinear functions, such as sigmoid or hyperbolic tangent (tanh) function, were used previously because they are mathematical representations of a biological neuron behavior, the most common nonlinear activation function used presently is the rectified linear unit (ReLU) [39].

B. Pooling layer

by executing a down-sampling procedure along the spatial dimensions, this layer hopes to reduce the amount of extracted features [40]. The pooling layer works by aggregating a set of data, where the input can be of any kind, including arrays, images, and other sorts of data [41]. There are three types of pooling layers: min-pooling, average-pooling and max-pooling.

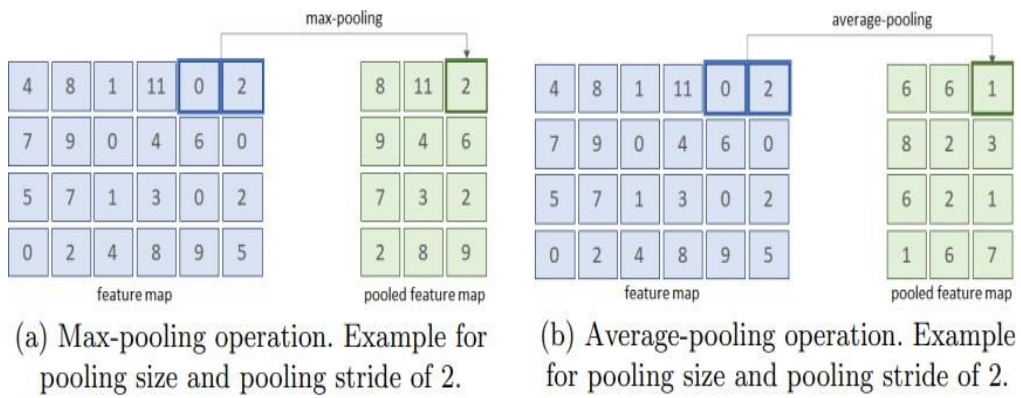


Figure 2.11: Pooling types [40].

- **Min-pooling:** it take the minimum element from a block in terms of pooling size.
- **Average-pooling:** it consists to take the average of a pooling block's pooling size (i.e., the number of elements that may be covered by a single pooling operation).
- **Max-pooling:** it represents the maximum element from a block in terms of pooling size is returned by max-pooling layers [40].

C. Fully Connected Layer

The final convolution or pooling layer's output feature maps are often flattened, converted to a one-dimensional array of integers (or vector), and linked to one or more fully connected layers. The number of output nodes in the final fully linked layer is usually equal to the number of classes. Each fully connected layer is followed by a nonlinear function, such as ReLU [39].

2.3.3.2 Activation Functions

An activation function is a mathematical function applied to the output of a neuron. Activation function decides whether a neuron should be activated by calculating the weighted sum of inputs and adding a bias term. This helps the model make complex decisions and predictions by introducing non-linearities to the output of each neuron. [42] There are many popular activation functions, such as sigmoid, Rectified Linear Unit (ReLU).

- **Rectified Linear Unit (ReLU)**

The rectified liner unit, or ReLU, is a non-linear activation function commonly employed in neural networks. The advantage of employing ReLU is that not all neurons are stimulated at the same time. This means that a neuron will only be destroyed when the linear transformation output is zero. Mathematically, it may be defined as:

$$f(x) = \max(0, x) \text{ where } x \text{ is an input value.}$$

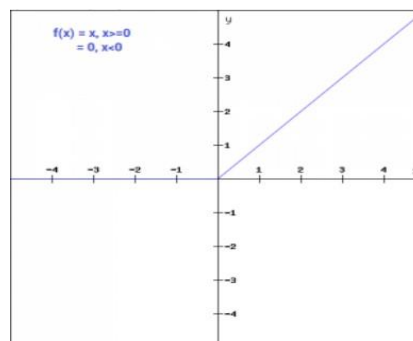


Figure 2.12: ReLU Activation Function Plot [43].

- **SIGMOID**

As a non-linear function, it is the most commonly employed activation function. The sigmoid function changes 0 to 1 values. Its definition is as follows:

$$f(x) = 1/e^{-x}$$

The sigmoid function is continuously differentiable and a smooth, S-shaped function. The derivative of the function is:

$$f(x) = 1 - \text{sigmoid}(x)$$

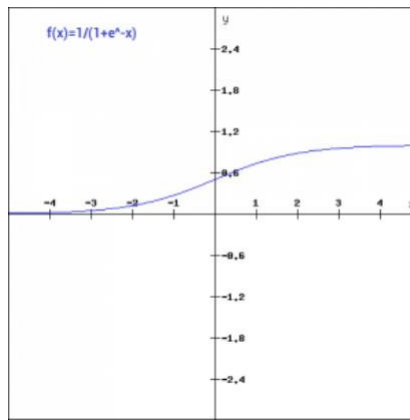


Figure 2.13: Sigmoid Function [43].

2.3.3.3 Example of CNNs based model

- **U-Net network**

The U-Net is a convolutional network architecture for fast and precise segmentation of images [44]. The U-Net model is capable of being trained on small datasets while achieving outstanding performance, a characteristic that is particularly critical in medical image segmentation tasks. Medical datasets are typically limited in size, and the requirements for annotation are relatively high, which makes the generation of large-scale labeled datasets challenging. Consequently, various U-Net-based modified network models have received widespread attention in medical image segmentation research [45].

- **Visual Geometry Group (VggNet)**

It is a typical deep Convolutional Neural Network (CNN) design with numerous layers, and the abbreviation VGG stands for Visual Geometry Group. The term deep describes the number of layers, with VGG-16 or VGG-19 having 16 or 19 convolutional layers, respectively. Innovative object identification models are

built using the VGG architecture. The VGGNet, created as a deep neural network, outperforms benchmarks on a variety of tasks and datasets outside of ImageNet. It also remains one of the most often used image recognition architectures today [46].

- **Residual Network (ResNet)**

Residual Networks ,commonly known as ResNet, represent a groundbreaking convolutional neural network (CNN) architecture developed by Kaiming He and colleagues at Microsoft Research. Introduced in their 2015 paper, "Deep Residual Learning for Image Recognition", ResNet addressed a major challenge in deep learning (DL): the degradation problem. This problem occurs when adding more layers to a very deep network leads to higher training error, contrary to the expectation that deeper models should perform better. ResNet's innovation allowed for the successful training of networks substantially deeper than previously feasible, significantly advancing the state-of-the-art in various computer vision (CV) tasks [47].

- **Densely Connected Network (DenseNet)**

A DenseNet is a type of convolutional neural network that utilises dense connections between layers, through Dense Blocks, where we connect all layers (with matching feature-map sizes) directly with each other. To preserve the feed-forward nature, each layer obtains additional inputs from all preceding layers and passes on its own feature maps to all subsequent layers [48].

2.3.3.4 Advantage of CNNs

- **Automatic Feature Extraction:** CNNs can automatically learn and extract relevant features from raw input data, eliminating the need for manual feature engineering. This capability enhances efficiency and reduces the potential for human error in feature selection [49].
- **High Accuracy in Image Recognition Tasks:** CNNs have demonstrated state-of-the-art performance in various image recognition tasks, including medical image segmentation, due to their deep architectures and ability to capture complex patterns [50].
- **Robustness to Noise and Distortions :** CNNs are robust to noise and distortions in input data, making them effective in real-world applications where medical images may be affected by various artifacts [51].
- **Efficient Processing of High-Dimensional Data :** Through local connectivity and shared weights, CNNs efficiently handle the high dimensionality of medical images,

reducing computational complexity compared to fully connected networks [52].

- **Versatility Across Medical Imaging Modalities:** CNNs have been successfully applied to various medical imaging tasks, including disease classification, localization, detection of pathological targets, organ region segmentation, and image enhancement [50] [53].

2.3.3.5 Limitations of CNNs

- **Dependence on Large Labeled Datasets:** CNNs typically require extensive labeled datasets to achieve high performance. In medical imaging, acquiring such datasets is challenging due to the need for expert annotations and patient privacy concerns [53].
- **Overfitting on Limited or Noisy Data:** When trained on limited or noisy datasets, CNNs are prone to overfitting, leading to poor generalization on unseen data. This is particularly problematic in medical imaging, where data variability is high [54].
- **Limited Interpretability:** CNNs often function as "black boxes," making it difficult to interpret their decision-making processes. In medical contexts, this lack of transparency can hinder clinical trust and acceptance [55].
- **Challenges with Capturing Global Context:** CNNs are inherently designed to capture local features due to their limited receptive fields. This makes it challenging to model global context, which is essential for understanding complex structures in medical images [56].
- **Sensitivity to Image Quality and Artifacts:** Medical images often contain noise, artifacts, or low contrast, which can adversely affect CNN performance. Without adequate preprocessing, CNNs may struggle to accurately segment such images [57].
- **High Computational Requirements:** Training and deploying CNNs, especially deep architectures, demand significant computational resources, including high-end GPUs. This can be a barrier in settings with limited infrastructure [57].

2.3.4 Transformer

Transformer Neural Networks, or simply Transformers, is a neural network architecture introduced in 2017 in the now-famous paper "Attention is all you need". The title refers to the attention mechanism, which forms the basis for data processing with Transformers. Transformer Networks have been the predominant type of Deep Learning models for NLP in recent years. They replaced Recurrent Neural Networks in all NLP tasks, and also, all Large Language Models employ the Transformer Network architecture. As well as, Transformer Networks were recently adapted for other tasks and have outperformed other Machine Learning models for image processing and video processing tasks, protein and DNA sequence prediction, time-series data processing, and

have been used for reinforcement learning tasks. Consequently, Transformers are currently the most important Neural Network architecture [58].

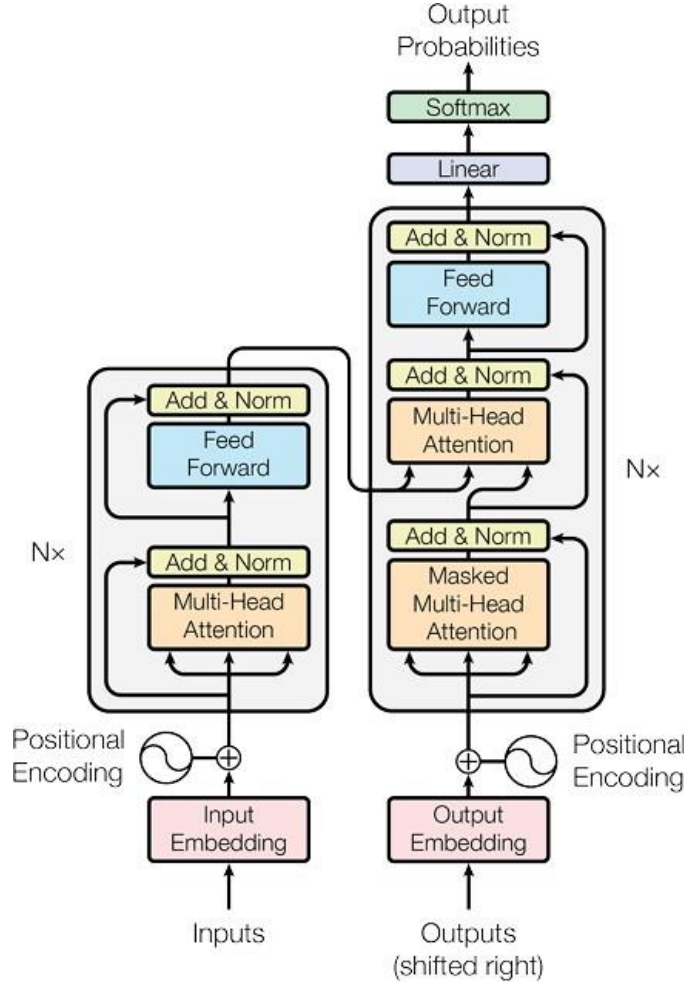


Figure 2.14: The Transformer- model architecture [59].

The Transformer architecture is based on an encoder-decoder structure composed entirely of attention mechanisms and feed-forward layers. Each encoder layer includes multi-head self-attention, followed by a position-wise feed-forward neural network, with residual connections and layer normalization applied at each step. Self-attention allows the model to compute relationships between all positions in the input simultaneously. To retain positional information, which is otherwise lost due to the lack of recurrence or convolution, the model incorporates positional encoding using sine and cosine functions. In the decoder, masked multi-head attention is employed to prevent access to future tokens during training. The final output is generated via a linear projection followed by a softmax layer. This architecture enables parallel computation, long-range dependency modeling, and significantly improves performance in various sequence modeling tasks. [58] [59].

2.3.4.1 Example of Transformer-based models

- **Vision Transformer (ViT)**

The vision transformer (ViT) introduced by Dosovitskiy et al. (2020) is an architecture directly inherited from Natural Language Processing (Vaswani et al., 2017), but applied to image classification with raw image patches as input [60]. In contrast to CNNs, which process spatial information through convolutions, ViTs divide an image into patches and process each patch as a sequence token, similar to words in a sentence. By applying self-attention, ViTs capture both local and global dependencies in images without requiring convolutional layers [61]. ViT requires a large amount of data for pre-training [62].

- **Swin transformer**

Swin Transformers mark a paradigm shift in transformer-based architectures [63]. The Swin Transformer divides the patches into non-overlapping windows and restricts the self-attention calculation in the small or shifted windows, which introduces locality inductive bias. It also utilizes patch-merging layers to generate hierarchical representations that can benefit the downstream tasks, such as object detection or semantic segmentation [64]. Swin Transformer is able to serve as a general-purpose backbone for computer vision. It has already been applied in detection, classification and segmentation and obtains strong performances [65].

- **DeiT transformer**

The DeiT (data-efficient image transformers) model was proposed in Training data-efficient image transformers and distillation through attention by Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, Hervé Jégou. DeiT are more efficiently trained transformers for image classification, requiring far less data and far less computing resources compared to the original ViT models [66]. DeiT can perform effectively with smaller datasets [62].

2.3.4.2 Advantages of transformer

- **Modeling Long-Range Dependencies:** Transformers excel at capturing long-range dependencies within data, which is crucial for medical image segmentation tasks where understanding the global context of an image is essential [67].
- **Global Feature Representation:** Through self-attention mechanisms, Transformers can capture global features in a single forward pass, enabling a comprehensive understanding of the entire image, which is beneficial for accurate segmentation[68].

- **Parallel Processing and Computational Efficiency:** Transformers allow for parallel processing of data, leading to faster computation times compared to sequential models like RNNs. This efficiency is advantageous when dealing with large medical imaging datasets [59].
- **Flexibility in Handling Varying Input Sizes:** Transformers are adaptable to inputs of varying sizes without the need for significant architectural changes, making them suitable for medical images that can differ in dimensions and resolutions [68].
- **Improved Performance on Complex Segmentation Tasks:** By effectively modeling both local and global contexts, Transformers have demonstrated superior performance in complex medical image segmentation tasks, such as delineating tumors or organs with irregular shapes [68] [69].

2.3.4.3 Limitations of Transformer

- **Data Inefficiency and Lack of Inductive Bias:** Unlike convolutional neural networks (CNNs), transformers lack inherent inductive biases like locality and translation invariance. This absence makes them less data-efficient and more prone to overfitting, particularly when training on small datasets common in medical imaging [70].
- **Limited Interpretability:** The complex architecture of transformers, characterized by numerous layers and attention heads, poses challenges for interpretability. This opacity can be problematic in critical applications like healthcare, where understanding model decisions is essential [71].
- **Challenges in Capturing Local Context:** While transformers excel at modeling global dependencies, they may struggle with capturing fine-grained local features, which are crucial in tasks like medical image segmentation. This limitation can lead to less precise segmentation results [72].
- **Dependence on Large-Scale Pretraining:** Transformers often require large-scale pre-training on extensive datasets to achieve optimal performance. In domains like medical imaging, where annotated data is scarce, this dependence can be a significant hurdle [68].

2.4 Advanced Training Techniques

There are different advanced techniques for efficient training of DL approach , such as regularization and Data Augmentation.

2.4.1 Data Augmentation

In the realm of medical imaging, the training of machine learning models necessitates a large and varied training dataset to ensure robustness and interoperability. However, acquiring such diverse and heterogeneous data can be difficult due to the need for expert labeling of each image and privacy concerns associated with medical data. To circumvent these challenges, data augmentation has emerged as a promising and cost-effective technique for increasing the size and diversity of the training dataset. Data Augmentation encompasses a suite of techniques that enhance the size and quality of training datasets such that better Deep Learning models can be built using them [73]. Data Augmentation has many techniques as follows:

- **Rotation:** rotation augmentations are done by rotating the image right or left on an axis between 1° and 359° . The safety of rotation augmentations is heavily determined by the rotation degree parameter. Slight rotations such as between 1° and 20° or -1° to -20° could be useful on digit recognition tasks, but as the rotation degree increases, the label of the data is no longer preserved post-transformation [74].
- **Flipping :** This technique, also known as mirroring, involves flipping the image horizontally or vertically, which helps the model become more robust to variations in image direction. Horizontal flipping involves flipping the image from left to right, while vertical flipping involves flipping the image from top to bottom. Both types of flipping can be used in combination to further increase the diversity of the training data [73].
- **Zooming :** Zooming is a common data augmentation technique in computational pathology. Zooming in or out of an image involves changing its size, either by enlarging or reducing its dimensions, while keeping the content of the image centered. In the context of computational pathology, zooming can help the model become more robust to variations in image scale. This is important because histopathological images can come from different sources, such as different scanners or microscopes, and can have varying sizes. By applying zoom as a data augmentation technique, the model can learn to recognize patterns at different scales and become more accurate at detecting features in images of varying sizes. Zooming can be performed in different ways, such as by cropping and resizing the image or by using a zoom function that rescales the image while preserving its aspect ratio. It is important to note that excessive zooming can lead to loss of information and can negatively impact model performance. Therefore, careful selection of the amount and type of zoom is crucial to ensure optimal model performance [73].

- **Elastic Deformation** Elastic deformation has the characteristic of affecting the intrastructural information of an image. In medical imaging, living human objects are inherently subject to naturally occurring transformations which can be extrapolated to elastic deformations for the purpose of data augmentation for training datasets. This allows the model to better generalize and accurately identify anatomical structures, even when they appear differently due to factors like patient movement or physiological changes [75].
- **Gamma Correction / Intensity Shift:** Gamma correction is a data augmentation technique that involves modifying the intensity of an image by adjusting the gamma value. Gamma is a nonlinear function that is used to encode and decode the luminance or brightness of an image. In gamma correction, the gamma value is adjusted to modify the overall brightness of the image. This technique is particularly useful when dealing with images that have low contrast or when the lighting conditions are not ideal. By adjusting the gamma value, it is possible to enhance the contrast of the image, making it easier for the model to distinguish between different structures and features. Gamma correction can be applied in a variety of ways, such as globally to the entire image or locally to specific regions of interest [73].
- **Gaussian Noise:** Gaussian noise is a type of noise that is added to the image by introducing random values drawn from a Gaussian distribution. The addition of Gaussian noise to an image can help the model become more robust to variations in image quality, as it simulates the noise that can occur during image acquisition and pre-processing. The amount of noise added to the image can be controlled by adjusting the standard deviation of the Gaussian distribution. A higher standard deviation will result in more noise being added to the image [73].

2.4.2 Regularization

Regularization is a technique used in machine learning to prevent overfitting, which occurs when a model learns the training data too well, including its noise and outliers, and performs poorly on new, unseen data. Regularization helps create models that generalize better to new data by adding a penalty to the loss function (the function the model tries to minimize during training), which keeps the model's parameters (like weights in a neural network) smaller and simpler [76].

2.5 Evaluation Metrics

2.5.1 Dice Coefficient (Dice Similarity Coefficient - DSC)

The Dice coefficient is a measure of the similarity between two sets, A and B. The coefficient ranges from 0 to 1, where 1 indicates that the two sets are identical, and 0 indicates that the two sets have no overlap [77]. It is defined as:

$$\text{DSC}(A, B) = \frac{2(A \cap B)}{A + B} \text{ where } \cap \text{ is the intersection}$$

2.5.2 Intersection over Union (IoU) / Jaccard Index

Intersection over Union (IoU) is a measure that shows how well the prediction bounding box aligns with the ground truth box. It is one of the main metrics for evaluating the accuracy of object detection algorithms and helps distinguish between "correct detection" and "incorrect detection". By measuring how well the model's prediction describes the actual region of interest, the IoU score, alongside other evaluation measures, helps researchers gauge the effectiveness and reliability of their models and make informed decisions about algorithm performance [78].

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$$

2.5.3 Accuracy

Accuracy (Acc), also known as Rand index or pixel accuracy, is one or even the most known evaluation metric in statistics. It is defined as the number of correct predictions, consisting of correct positive and negative predictions, compared to the total number of predictions [79].

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

-TP: True Positives, the number of instances where the model correctly predicted a positive class.

-TN: True Negatives, the number of instances where the model correctly predicted a negative class.

-FP: False Positives, the number of instances where the model incorrectly predicted a positive class (Type 1 error).

-FN: False Negatives, the number of instances where the model incorrectly predicted a negative class (Type 2 error) [80].

2.5.4 Precision

Precision, also known as the positive predictive value (PPV), is a classification metric that focuses on the accuracy of positive predictions. It measures the proportion of instances the model predicted as positive that were true positives. Precision ranges from 0 to 1, with higher values indicating better performance. A perfect precision of 1.0

means the model correctly identifies every positive instance without any false positives. This metric is valuable when the consequences of false positives are more severe than false negatives. It tells us how reliable the model's positive predictions are [80]. Mathematically, it is defined as:

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

2.5.5 Recall

Recall, also known as sensitivity or true positive rate (TPR), is a classification metric that measures the ability of a model to identify all relevant instances of a particular class. It is defined as the proportion of actual positive instances correctly predicted by the model. recall ranges from 0 to 1, with higher values signifying better performance. A recall of 1.0 indicates that the model perfectly identifies all positive instances without missing any [80].

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

2.5.6 F1 Score

The harmonic mean of precision and recall. It balances the two metrics into a single number, making it especially useful when precision and recall are in trade-off [81].

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.5.7 Hausdorff distance

The Hausdorff distance measures the extent to which each point of a "model" set lies near some point of an "image" set and vice versa. Thus, this distance can be used to determine the degree of resemblance between two objects that are superimposed on one another [82]. Basically, the Hausdorff metric will serve to check if a template image is present in a test image ; the lower the distance value, the best the match. That method gives interesting results, even in presence of noise or occlusion (when the target is partially hidden) [83].

2.5.8 Loss Function

A loss function is a type of objective function, which in the context of data science refers to any function whose minimization or maximization represents the objective of model training. The term loss function, which is usually synonymous with cost function or error function, refers specifically to situations where minimization is the training objective for a machine learning model. In simple terms, a loss function tracks the degree of error in an artificial intelligence (AI) model's outputs. It does so by quantifying the difference loss between a predicted value that is, the model's output for a given input and the actual value or ground truth. If a model's predictions are accurate, the loss is small. If its predictions are inaccurate, the loss is large. The fundamental goal of machine learning is to train models to output good predictions. Loss functions enable us to define and pursue that goal mathematically. During training, models learn to output better predictions by adjusting parameters in a way that reduces loss. A machine learning model has been sufficiently trained when loss has been minimized below some predetermined threshold [84].

2.6 Related Works for skin cancer detection with deep learning

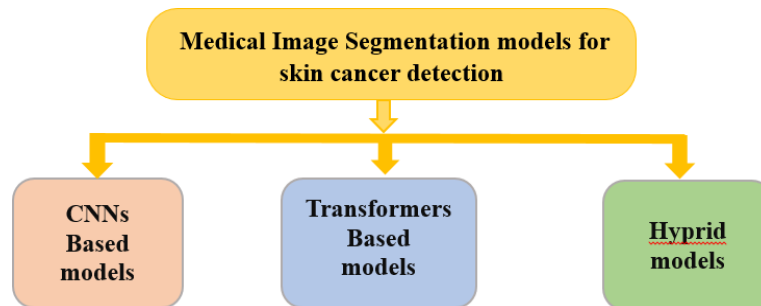


Figure 2.15: Deep learning models used for skin cancer detection.

2.6.1 Medical Image Segmentation Based on CNNs

CNNs have been widely successful in skin lesion analysis and skin cancer detection due to their strong capability to extract features from images. Key recent CNN-based models include :

- **ResNet (Residual Networks):** A deep CNN architecture that uses residual learning to ease the training of very deep networks. Widely used for skin lesion classification with high accuracy [85].
- **DenseNet (Densely Connected Convolutional Networks):** Connects each layer to every other layer to improve information flow and reduce vanishing gradients. Successfully applied in medical image classification including skin cancer [86].
- **VGGNet:** Known for its simple and deep architecture using small (3x3) convolutional

filters. Often used as a baseline CNN in medical imaging [87].

- **EfficientNet:** A scalable CNN balancing network depth, width, and resolution, achieving state-of-the-art accuracy with fewer parameters [88].
- **Inception Networks (GoogLeNet and successors):** Employ multiple parallel convolutional filters of different sizes to capture multi-scale features. Applied for skin lesion classification [89].

2.6.2 Vision Transformer

Transformers, especially Vision Transformers (ViT), are relatively new in medical image analysis and have shown promising results in skin cancer detection due to their ability to model long-range dependencies:

- **SkinDistilViT:** A lightweight ViT model using knowledge distillation to reduce model size while maintaining accuracy [90].
- **SkinSwinViT:** Employs Swin Transformer with hierarchical local and global feature extraction to improve detection accuracy [91].
- **SkinViT:** Combines Transformer, MLP, and Outlooker modules for precise and global feature extraction [92].
- **Vision Transformer:** Ensemble Uses multi-scale ViT models whose outputs are ensembled to boost classification accuracy [93].

2.6.3 Medical Image Segmentation Based on Transformers and CNNs

Hybrid models combine the local feature extraction power of CNNs with the global context modeling capability of Transformers, achieving superior diagnostic performance:

- **GS-TransUNet:** Combines Gaussian Splatting and Transformer-based UNet for precise skin lesion segmentation [94].
- **Attention Swin U-Net:** Integrates UNet with Swin Transformer and cross-attention mechanisms to improve segmentation quality [95].
- **SkinEHDLF:** A classification model that integrates CNN and Transformer techniques to enhance skin lesion recognition [96].

2.6.4 Summary table

Category	Model Name	Key Features	Datasets Used	Evaluation Metrics / Results
CNN-based Models	ResNet	Deep residual learning for easier training of deep nets	ISIC, HAM10000	Accuracy ~85-90%, AUC ~0.93-0.95
	DenseNet	Dense connections between layers for better gradient flow	ISIC, PH2	Accuracy ~88-92%, Dice ~0.85-0.89
	VGGNet	Simple deep CNN with small filters (3x3)	ISIC	Accuracy ~83-88%, AUC ~0.90
	EfficientNet	Balanced scaling of depth, width, resolution	ISIC	Accuracy ~90%, F1-score ~0.89
	Inception	Multi-scale feature extraction via parallel convolutions	ISIC	Accuracy ~87-90%, AUC ~0.92
Transformer-based Models	SkinDistilViT	Lightweight ViT using knowledge distillation	ISIC 2018	Accuracy ~91%, AUC ~0.94
	SkinSwinViT	Swin Transformer with hierarchical local/global features	HAM10000	Dice ~0.87, Accuracy ~92%
	SkinViT	Combines Transformer, MLP, and Outlooker modules	ISIC 2018 and others	Accuracy ~90-93%, F1-score ~0.90
	ViT Ensemble	Multi-scale ViT ensemble for enhanced accuracy	ISIC 2018	Accuracy up to 95%, AUC ~0.96
Hybrid CNN + Transformer	GS-TransUNet	Gaussian Splatting + Transformer-based UNet for segmentation	ISIC 2017, PH2	Dice ~0.89-0.92, Accuracy ~93%
	Attention Swin U-Net	UNet + Swin Transformer + cross-attention	ISIC, HAM10000	Dice ~0.90, Accuracy ~94%
	SkinEHDLF	CNN + Transformer for improved classification	ISIC 2020, HAM10000	Accuracy ~92%, F1-score ~0.91

Table 2.1: Deep learning models used for skin cancer detection.

2.7 Conclusion

This chapter reviewed the foundational concepts of AI and deep learning in the context of medical image analysis, with a focus on skin cancer detection. We discussed various architectures, including CNNs, U-Net, and transformer-based models, highlighting their roles in feature extraction and segmentation accuracy. The integration of CNNs with transformer modules offers a promising path forward, combining local detail sensitivity with global context awareness. These hybrid architectures have paved the way for significant advances in medical image segmentation, which are essential for reliable and precise skin lesion analysis. In the next chapter, we will present our proposed segmentation model based on a fusion of CNN and transformer techniques, and demonstrate its effectiveness on benchmark datasets.

Chapter 3**Design, Implementation and
Experimental Results**

3.1 Introduction

This chapter presents a comprehensive overview of the design, implementation, and evaluation of the skin cancer segmentation system. It begins by outlining the system's overall architecture, supported by detailed descriptions and Unified Modeling Language (UML) diagrams. Subsequently, the technical implementation aspects are discussed, including model construction and the development environment. The chapter concludes with a series of experiments conducted on the ISIC 2017 dataset, followed by an in-depth analysis and validation of the obtained results.

3.2 Design and conception**3.2.1. Model description**

The model adopts a hybrid architecture that combines Convolutional Neural Networks (CNNs) and Transformers, designed specifically for the semantic segmentation of medical images. CNNs allow the model to focus on fine-grained texture details, while Transformers provide a broader view of the image context. This combination enables accurate detection of both subtle and large-scale patterns essential for reliable medical image analysis.

3.2.2 Global design

To develop our skin cancer segmentation system, we used skin lesion images from the ISIC 2017 dataset. dataset are first loaded and processed, then used to train a deep learning model. After training, the model predicts and produces segmentation images that highlight important regions in the medical images. Figure 3.1 present the general design of our system

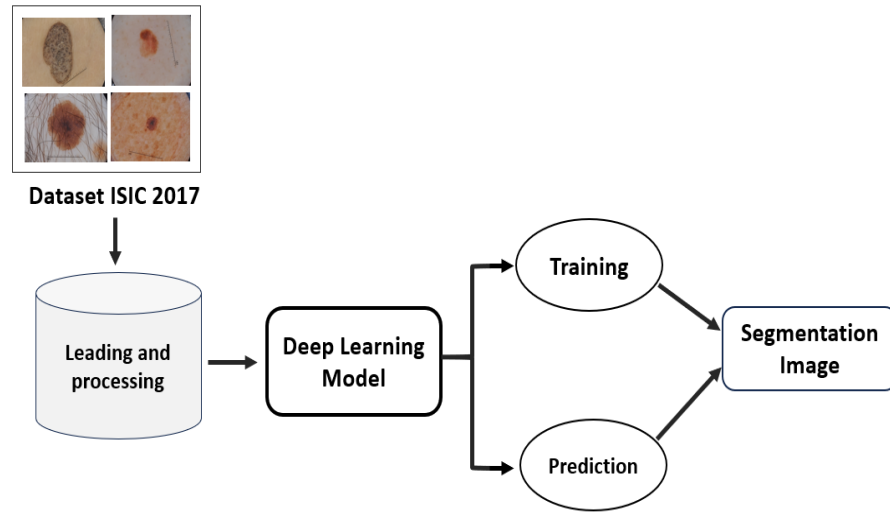


Figure 3.1: Global design of our skin cancer segmentation system.

3.2.3 Model architecture

The detailed architecture of our deep learning model is presented as follow :

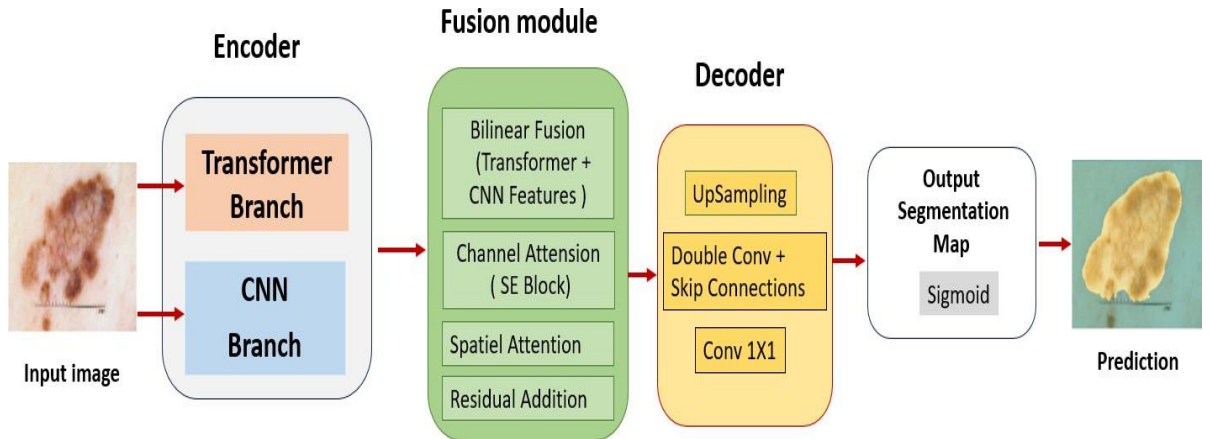


Figure 3.2: Model detailed architecture.

3.2.4 Detailed architecture description

1. **Input image:** It starts with taking an input from a skin lesion scan dermoscopic skin lesion or any other radiological scan.
2. **Encoder:** The encoder's role is to extract meaningful features from the input image. In our architecture, it consists of two parallel branches a Transformer branch and a CNN branch. Together, they convert the raw image into rich, high-level representations that capture both global context and local details, preparing the features for seamless fusion and accurate decoding.
 - A. **Transformer Branch:** captures the global relationships across the image using self-attention mechanisms. The purpose of this branch is to extract global features that reflect the interconnections between all regions of the image, enabling the model to better understand contextual dependencies and improve overall segmentation performance.
 - B. **CNN Branch :** extracts fine-grained local and spatial features from the input image using convolutional and pooling layers. These features are then transformed into rich representations that encode essential details, effectively preparing them for fusion with global features from the Transformer Branch and subsequent decoding.
3. **Fusion module** is a key component in our architecture that combines the strengths of both the Transformer and CNN branches. It fuses the global features extracted by the Trans former with the local features captured by the CNN, ensuring that the model benefits from both types of information.in This module we use several mechanisms :
 - **Bilinear Fusion (Transformer + CNN Features):** combines the features from both the Transformer and CNN branches using bilinear pooling, which captures complex interactions between the two feature sets.
 - **Channel Attention (SE Block):** Squeeze-and-Excitation (SE) blocks are used to emphasize the most important feature channels.
 - **Spatial Attention:** focuses on the most relevant spatial locations in the feature maps.
 - **Residual Addition:** adds the fused features to the original features to preserve information and facilitate gradient flow.

- 4. Decoder** The decoder transforms the abstract features obtained from the encoder and fusion module back into a detailed segmentation map with the same spatial size as the input image. It consists of: UpSampling Layers, Double Conv, Skip Connections, Conv 1x1. The goal is to reconstruct a high-resolution segmentation map, accurately delineating each region or class in the original image.

- **UpSampling Layers:** gradually increase the spatial resolution of the feature maps.
- **Double Conv + Skip Connections :** mechanism that plays a crucial role in the decoding path by refining the upsampled features through two convolutional layers. It incorporates skip connections from the encoder, allowing the model to recover fine details lost during downsampling and enhance segmentation accuracy.
- **Conv 1x1:** A 1x1 convolution to reduce the number of channels to the number of segmentation classes.

5. Output Segmentation Map

- **Sigmoid:** Applies the sigmoid function for binary segmentation to generate the final prediction map.
- **Prediction:** produces a segmentation map that highlights the regions of interest in the original image.

3.2.5 UML presentations

- **State transition diagram**

State diagrams provide a visual representation of the various states a system or an object can be in, as well as the transitions between those states. They are essential in modeling the dynamic behavior of systems, capturing how they respond to different events over time. State diagrams depict the system's life cycle, making it easier to understand, design, and optimize its behavior [97]. The state transition diagram of our system is presented as follow :

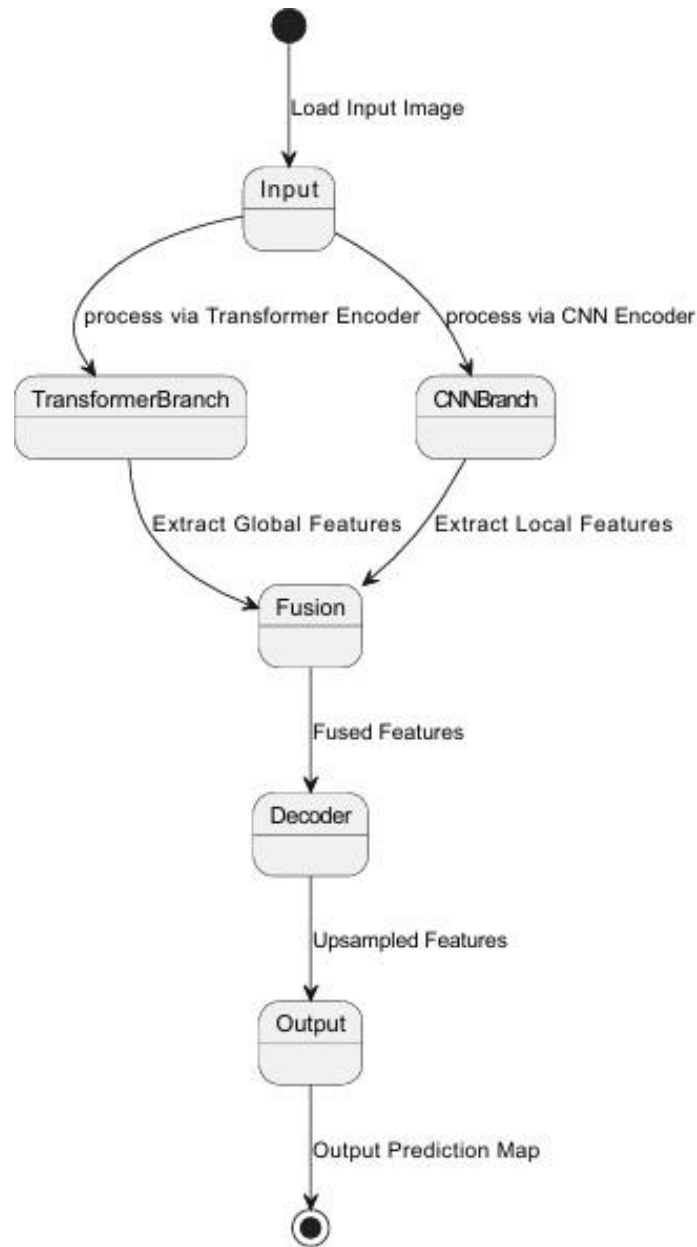


Figure 3.3: The state transition diagram of our system.

The Figure 3.3 illustrates the state transition process of our model. It begins by loading the input image, which is then processed through two parallel branches: the Transformer encoder branch and the CNN encoder branch. The Transformer branch extracts global contextual features, while the CNN branch captures local spatial features. Both sets of features are then fused together in the fusion

module to leverage the strengths of each representation. The fused features are passed through a decoder that progressively upsamples them to restore the spatial resolution. Finally, the output branch generates the segmentation prediction map.

- **Sequence diagram**

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction [98]. The sequence diagram of our system is presented as follow :

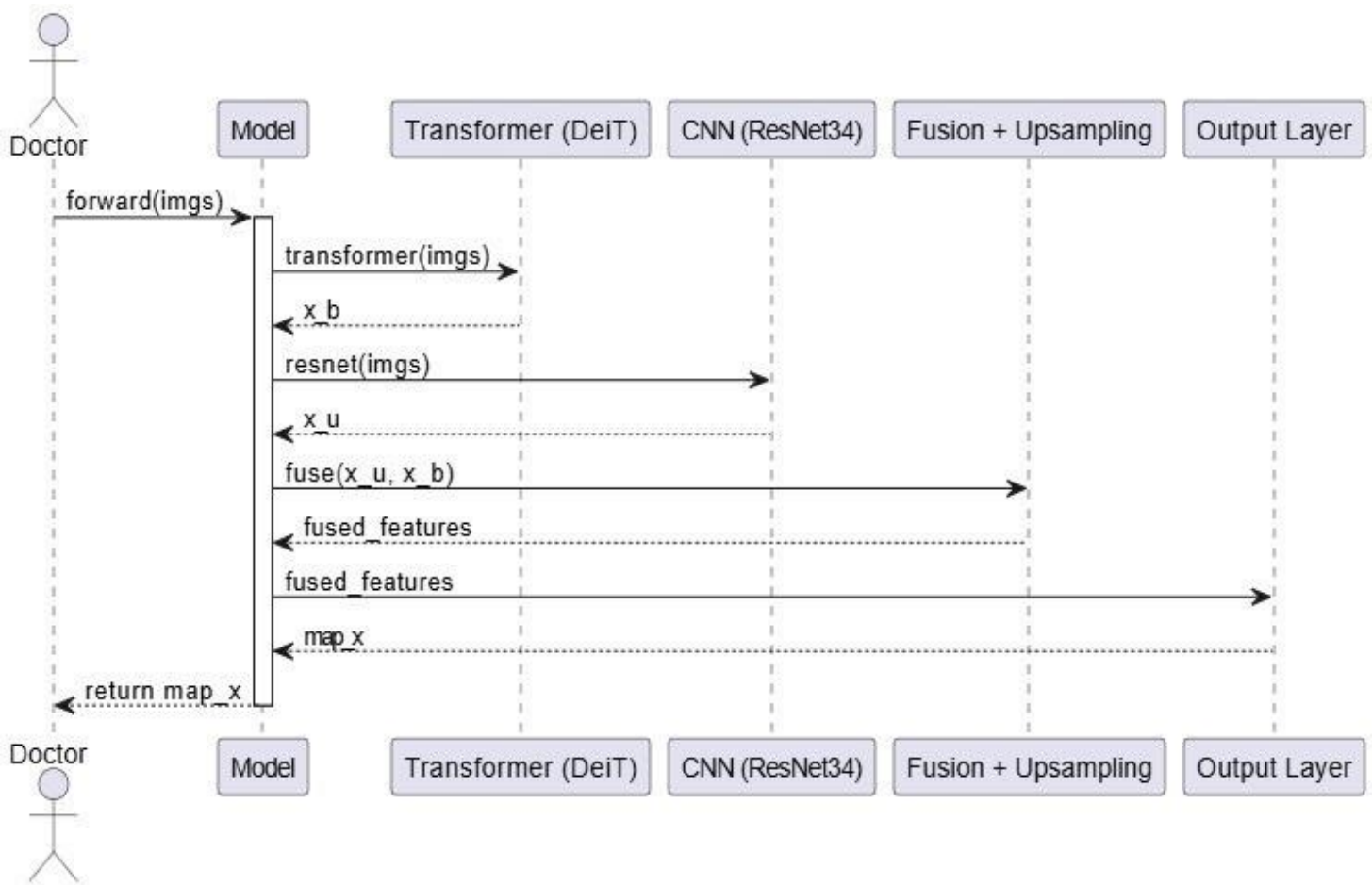


Figure 3.4: The sequence diagram of our system.

The sequence diagram in Figure 3.4 illustrates the forward pass process of the model. When an input image is provided, it is simultaneously processed by two parallel branches: a Transformer (DeiT) branch and a Convolutional Neural Network (ResNet34) branch. The Transformer branch extracts global contextual features, while the CNN branch focuses on capturing local spatial details. The outputs from both branches are then passed to a fusion and upsampling module, which integrates the multi-scale features. Finally, the fused representation is passed through the output layers to generate the segmentation map (the final prediction). This architecture enables the model to leverage both global and local information for accurate image segmentation.

3.3 Implementation

This section explains the practical steps taken to develop the skin lesion segmentation system. It includes the tools, programming environments, and hardware used during the implementation. In addition, it describes how the dataset was prepared and provides details on how the system was built and trained.

3.3.1 Environments and developing tools

We required different environments, packages, APIs, libraries and programming languages to implement our skin segmentation system.

1. Hardware configuration

Google Colab was used for training and development of the model, which is a Google-provided cloud platform that enables users to run Python notebooks with high-performance computation resource allocation. Hardware configuration in this environment is:

- **CPU:** Intel Xeon (2.3 GHz) or AMD EPYC, commonly 2 cores.
- **RAM:** 12GB maximum.
- **GPU:** NVIDIA Tesla T4 (16GB) or Tesla K80 (12GB) if available.
- **Storage:** Temporary disk with Google Drive 78GB integration.

2. Development environment

To develop our system, we used a range of tools and environments, as summarized in Table 3.1.



Tools logo	Descriptions
	Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together [99].
	Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free of charge access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education [100].

Table 3.1: Illustration of tools environment.

3. Packages and APIs

We used several programming libraries and APIs throughout the project to support the development of our deep learning model. These tools helped define the functions and components essential to the system, as shown in Table 3.2:












Package/API	Description
 PyTorch	An open-source deep learning framework for building and training neural networks [101].
 TorchVision	Provides datasets, pretrained models, and image transformations for PyTorch [102].
 OS Module	Provides functions to interact with the operating system (e.g., file and path handling) [103].
 timm	PyTorch Image Models library offering many pretrained computer vision models [104].
 argparse	A module for parsing command-line arguments passed to Python scripts [105].
 pillow	Python Imaging Library used for opening, manipulating, and saving image files [106].
 OpenCV	Powerful computer vision library used for image processing and analysis [107].
 NumPy	Fundamental package for numerical computation and array operations [108].
 matplotlib	A plotting library for creating static, animated, and interactive visualizations [109].
 Albumentations	Fast and flexible image augmentation library used in deep learning pipelines [110].
 imageio	A library for reading and writing images and videos in many formats [111].

Table 3.2: Illustration of packages and APIs.

3.3.2 Used Image dataset details information

To evaluate the performance of our skin lesion segmentation model, we used the ISIC 2017 dataset. This dataset is divided into three subsets training, validation, and testing and includes a total of 2,750 dermoscopic images. Each image comes with a binary segmentation mask that clearly outlines the lesion boundary, a critical feature for effective segmentation. The dataset is organized as follows:

- **Training set:** 2,000 images along with their ground truth segmentation masks.
- **Validation set:** 150 images along with ground truth segmentation masks.
- **Test set:** 600 images with publicly released ground truth segmentation masks, utilized for model performance evaluation.

This dataset was published as a part of the ISIC 2017 Challenge, initiated by the International Skin Imaging Collaboration, for promoting the development of computerized algorithms for the detection of skin cancer.

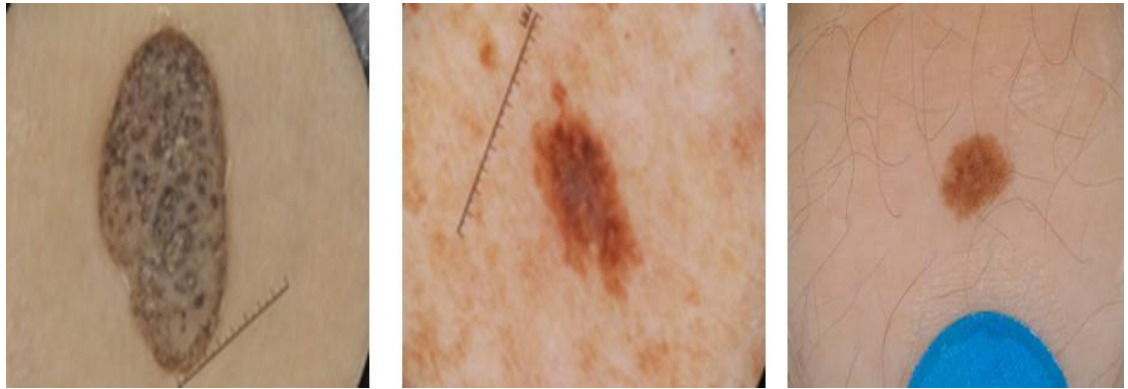


Figure 3.5: Examples of dermoscopic images from dataset ISIC 2017.

3.3.3 Implementation details

1. Data preparation and preprocessing

At this stage, we prepared the ISIC 2017 dataset to ensure it was ready for effective use in training and testing our deep learning model. The original dataset includes dermoscopic images of skin lesions along with their corresponding ground truth segmentation masks, organized into separate folders for training, validation, and testing.

To streamline further processing and improve training efficiency, we converted the raw data into structured NumPy arrays with consistent formatting.

- First, we import the necessary packages.

```
import numpy as np
import cv2
import os
```

Figure 3.6: Illustration of the necessary package for data preparation.

- Next, we define the root directory of the dataset, along with the folder names for each subset's images and masks (training, validation, and test). We also specify the number of samples in each subset and set the target size for all images and masks to 256×192 pixels.

```
root = 'data/'
data_f = ['ISIC-2017_Training_Data/', 'ISIC-2017_Validation_Data/', 'ISIC-2017_Test_v2_Data/']
mask_f = ['ISIC-2017_Training_Part1_GroundTruth/', 'ISIC-2017_Validation_Part1_GroundTruth/', 'ISIC-2017_Test_v2_Part1_GroundTruth/']
set_size = [2000, 150, 600]
save_name = ['train', 'val', 'test']

height = 192
width = 256
```

Figure 3.7: Dataset preprocessing parameter representation.

- Then, for each dataset split training, validation, and test ,we initialize empty NumPy arrays to store the preprocessed images and their corresponding masks. We then loop through all images in the current directory, read each image along with its mask, convert the image from BGR to RGB format, resize both to the specified dimensions, and store the results in the arrays.

```

for j in range(3):

    print('processing ' + data_f[j] + '.....')
    count = 0
    length = set_size[j]
    imgs = np.uint8(np.zeros([length, height, width, 3]))
    masks = np.uint8(np.zeros([length, height, width]))

    path = root + data_f[j]
    mask_p = root + mask_f[j]

    for i in os.listdir(path):
        if len(i.split('_'))==2:
            img = cv2.imread(path+i)
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            img = cv2.resize(img, (width, height))

            m_path = mask_p + i.replace('.jpg', '_segmentation.png')
            mask = cv2.imread(m_path, 0)
            mask = cv2.resize(mask, (width, height))

            imgs[count] = img
            masks[count] = mask

            count +=1
    print(count)

```

Figure 3.8: Data preprocessing illustration.

- Finally, we store the resulting image and mask arrays as .npz files (data-train.npz, mask-train.npz,.....). Data can be loaded quickly and efficiently when training the model and evaluating it.

```

np.save('{} /data_{}.npz'.format(root, save_name[j]), imgs)
np.save('{} /mask_{}.npz'.format(root, save_name[j]), masks)

```

Figure 3.9: storing the resulting image and mask arrays as .npz files.

2. Data Loading and Transformations

- **Loading images and masks from NumPy files:** The Dataset images and their corresponding masks are loaded from previously saved NumPy files. This method speeds up data access and facilitates batch processing.

```
def __init__(self, image_root, gt_root):  
    self.images = np.load(image_root)  
    self.gts = np.load(gt_root)  
    self.size = len(self.images)
```

Figure 3.10: Loading images and masks from NumPy files.

- **Defining Image and Mask Transformations (Normalization + Augmentation):** for preparing the training data, images get normalized based on ImageNet statistics and masks get converted to tensors without normalizing. Shifting, scaling, rotation, color jitter, and flipping augmentations get performed on images as well as masks for enhancing data diversity and model generalization.

```
# Image and mask transformations  
self.img_transform = transforms.Compose([  
    transforms.ToTensor(),  
    transforms.Normalize([0.485, 0.456, 0.406], # ImageNet mean  
                        [0.229, 0.224, 0.225]) # ImageNet std  
])  
self.gt_transform = transforms.Compose([  
    transforms.ToTensor()]) # No normalization for masks  
# Data augmentation using Albumentations  
self.transform = A.Compose(  
    [  
        A.ShiftScaleRotate(shift_limit=0.15, scale_limit=0.15, rotate_limit=25, p=0.5, border_mode=0),  
        A.ColorJitter(),  
        A.HorizontalFlip(),  
        A.VerticalFlip()  
    ]  
)
```

Figure 3.11: images and masks transformations.

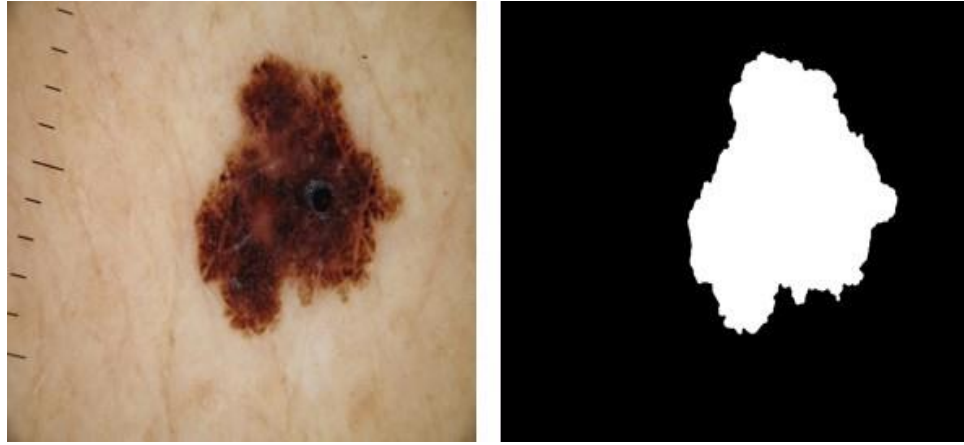


Figure 3.12: Example of a skin lesion and its corresponding mask.

3. Building our Deep Learning Model

Once the dataset is ready, we begin building our model following the architecture introduced earlier in this chapter (see Figure 3.1) . We then present the implementation details for each component of the design.

1- Encoder Block

A. Transformer Branch

We implemented the Transformer branch using the DeiT-small-244 Vision Transformer. It captures global features by dividing the input image into patches and applying self-attention across the resulting sequence of tokens. The output embeddings are then reshaped into a 2D feature map to match the dimensions of the CNN output, enabling effective multi-scale fusion.

```
from .DeiT import deit_small_patch16_224 as deit
```

Figure 3.13: Import DeiT-Small transformer.

B. CNN Branch

We implemented the CNN branch using a ResNet-34 backbone. This branch focuses on capturing local and hierarchical features through a sequence of convolutional and residual layers. The earlier layers extract low-level patterns, such as edges and textures, while the deeper layers capture more abstract, semantic information. These local feature maps are later combined with the global features from the Transformer branch to enhance segmentation performance.

```
from torchvision.models import resnet34
```

Figure 3.14: Importing of the pretrained RestNet 34.

2- Fusion Module (BiFusion Block)

We implement a BiFusion block which combines CNN and Transformer features with spatial and channel attention mechanisms, bilinear interaction, and residual connection, optionally with dropout.

```

class BiFusion_block(nn.Module):
    def __init__(self, ch_1, ch_2, r_2, ch_int, ch_out, drop_rate=0.):
        super(BiFusion_block, self).__init__()

        # channel attention for F_g, use SE Block
        self.fc1 = nn.Conv2d(ch_2, ch_2 // r_2, kernel_size=1)
        self.relu = nn.ReLU(inplace=True)
        self.fc2 = nn.Conv2d(ch_2 // r_2, ch_2, kernel_size=1)
        self.sigmoid = nn.Sigmoid()

        # spatial attention for F_l
        self.compress = ChannelPool()
        self.spatial = Conv(2, 1, 7, bn=True, relu=False, bias=False)

        # bi-linear modelling for both
        self.W_g = Conv(ch_1, ch_int, 1, bn=True, relu=False)
        self.W_x = Conv(ch_2, ch_int, 1, bn=True, relu=False)
        self.W = Conv(ch_int, ch_int, 3, bn=True, relu=True)

        self.relu = nn.ReLU(inplace=True)

        self.residual = Residual(ch_1+ch_2+ch_int, ch_out)

        self.dropout = nn.Dropout2d(drop_rate)
        self.drop_rate = drop_rate

    def forward(self, g, x):
        # bilinear pooling
        W_g = self.W_g(g)
        W_x = self.W_x(x)
        bp = self.W(W_g*W_x)

        # spatial attention for cnn branch
        g_in = g
        g = self.compress(g)
        g = self.spatial(g)
        g = self.sigmoid(g) * g_in

        # channel atttention for transformer branch
        x_in = x
        x = x.mean((2, 3), keepdim=True)
        x = self.fc1(x)
        x = self.relu(x)
        x = self.fc2(x)
        x = self.sigmoid(x) * x_in
        fuse = self.residual(torch.cat([g, x, bp], 1))

        if self.drop_rate > 0:
            return self.dropout(fuse)
        else:
            return fuse

```

Figure 3.15: Illustration of Fusion Module (BiFusion block).

3- Decoder (UpSampling + DoubleConv + Skip Connections + Conv 1X1)

- UpSampling Layer

```
class Up(nn.Module):
    """Upscaling then double conv"""
    def __init__(self, in_ch1, out_ch, in_ch2=0, attn=False):
        super().__init__()

        self.up = nn.Upsample(scale_factor=2, mode='bilinear', align_corners=True)
        self.conv = DoubleConv(in_ch1+in_ch2, out_ch)

        if attn:
            self.attn_block = Attention_block(in_ch1, in_ch2, out_ch)
        else:
            self.attn_block = None

    def forward(self, x1, x2=None):
        x1 = self.up(x1)
        # input is CHW
        if x2 is not None:
            diffY = torch.tensor([x2.size()[2] - x1.size()[2]])
            diffX = torch.tensor([x2.size()[3] - x1.size()[3]])

            x1 = F.pad(x1, [diffX // 2, diffX - diffX // 2,
                           diffY // 2, diffY - diffY // 2])

            if self.attn_block is not None:
                x2 = self.attn_block(x1, x2)
            x1 = torch.cat([x2, x1], dim=1)
        x = x1
        return self.conv(x)
```

Figure 3.16: Illustration of UpSampling phase.

- DoubleConv

```
class DoubleConv(nn.Module):
    def __init__(self, in_channels, out_channels):
        super().__init__()
        self.double_conv = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True),
            nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(out_channels)
        )
        self.identity = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, kernel_size=1, padding=0),
            nn.BatchNorm2d(out_channels)
        )
        self.relu = nn.ReLU(inplace=True)

    def forward(self, x):
        return self.relu(self.double_conv(x)+self.identity(x))
```

Figure 3.17: Illustration of DoubleConv phase.

- Conv 1X1

```
class Conv(nn.Module):
    def __init__(self, inp_dim, out_dim, kernel_size=3, stride=1, bn=False, relu=True, bias=True):
        super(Conv, self).__init__()
        self.inp_dim = inp_dim
        self.conv = nn.Conv2d(inp_dim, out_dim, kernel_size, stride, padding=(kernel_size-1)//2, bias=bias)
        self.relu = None
        self.bn = None
        if relu:
            self.relu = nn.ReLU(inplace=True)
        if bn:
            self.bn = nn.BatchNorm2d(out_dim)

    def forward(self, x):
        assert x.size()[1] == self.inp_dim, "{} {}".format(x.size()[1], self.inp_dim)
        x = self.conv(x)
        if self.bn is not None:
            x = self.bn(x)
        if self.relu is not None:
            x = self.relu(x)
        return x
```

Figure 3.18: Illustration of Conv 1X1 phase.

3.4 Model Training

During training, the model learns to adjust its weights by recognizing patterns in annotated medical images, while validation helps prevent overfitting. Our architecture combines a Transformer-based feature extractor with a CNN-based segmentation head, trained end-to-end with multi-scale supervision. This hybrid approach takes advantage of the Transformer's ability to capture global context and the CNN's strength in extracting local details working together to accurately delineate lesion boundaries.

3.4.1 Loss Function

We define a custom structure loss function that combines:

✓ Weighted Binary Cross-Entropy (wBCE) :

- Weights are dynamically adjusted based on edge regions (higher weights near boundaries).
- Helps the model focus on difficult segmentation areas.

✓ Weighted Intersection-over-Union (wIoU) :

- Measures overlap between predictions and ground truth.
- Also weighted to emphasize boundary regions.

✓ **Final Loss:**

$$\text{Loss} = 0.5 \times \text{Loss2} + 0.3 \times \text{Loss3} + 0.2 \times \text{Loss4}$$

where Loss2, Loss3, Loss4, are losses from different prediction scales.

3.4.2 Optimization Strategy

✓ **Optimizer :** Adam

- Learning rate: $7e-5$.
- Momentum terms:

$$\beta_1 = 0.5, \beta_2 = 0.999$$

✓ **Gradient Clipping:** Applied with a maximum norm of 5.0 to prevent exploding gradients.

3.4.3 Evaluation Metrics

Metric	Purpose
Dice Coefficient	Calculates overlap between ground truth and prediction
IoU (Jaccard Index)	Dice-like, but more harshly penalizes false positives
Pixel Accuracy	Standard classification accuracy

Table 3.3: Illustration of used evaluation metrics.

3.4.4 Training loop

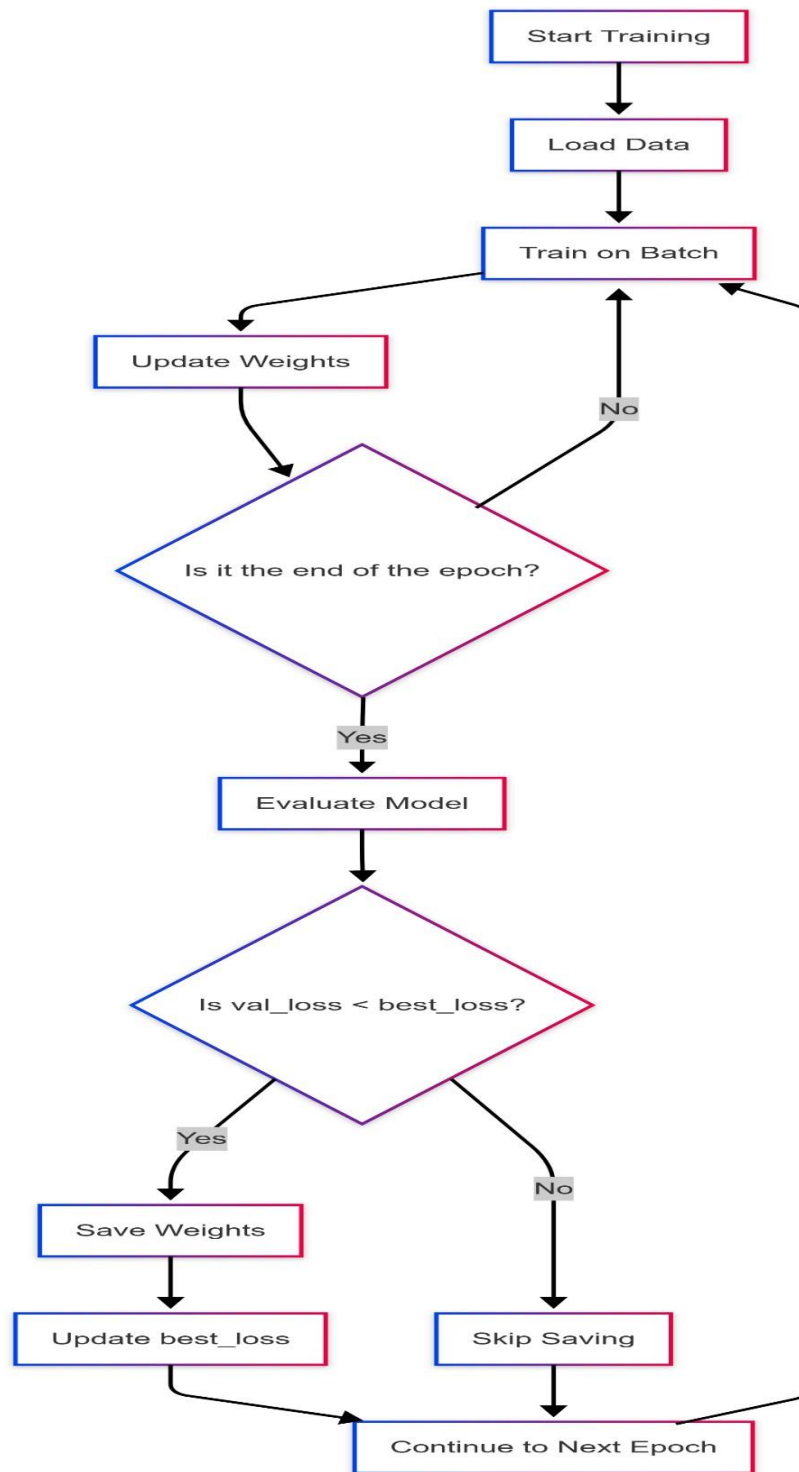


Figure 3.19: Training phase of our deep model.

This figure (Figure 3.19) illustrates a typical training process for our model. The process begins with loading the data, followed by training the model on batches of data and updating the model's weights iteratively. After each epoch (a complete pass through the entire dataset), the model is evaluated using a validation set to measure its performance. If the validation loss (val-loss) is lower than the best recorded loss (best-loss), the model's weights are saved, and best-loss is updated to the new value. If not, the weights are not saved, and the training proceeds to the next epoch. This cycle repeats until the training is complete, ensuring that only the best-performing model weights are retained based on validation performance. The process emphasizes iterative improvement and validation to achieve optimal model performance.

3.5 Experimental Results

In this section, we present the results of our model's automated segmentation across several samples from the dataset, evaluated at different training epochs.

3.5.1 Performance of Our Model

We evaluated our model's performance using several key metrics, including training and validation loss, accuracy, Dice coefficient, and Intersection over Union (IoU). The results, summarized below, highlight the model's effectiveness across these measures.

- **Loss:** Training loss converged at 0.30, whereas validation loss converged at 0.39 (Figure 3.20).
- **Accuracy:** Our model obtained very high classification accuracy of 99% for the training data and 97% for the validation data (Figure 3.21).
- **Dice Coefficient :** On the validation set, 0.89 Dice score was obtained, which reflects high overlap between predicted and ground truth segmentations (Figure 3.22).
- **IoU (Intersection over Union):** The model obtained an IoU value of 0.81 on the validation set, reflecting precise region-based segmentation (Figure 3.23).
- **These results validate that our model is:**
 - Stable training without overfitting.
 - High segmentation precision.
 - Strong generalization capability.



Figure 3.20: Loss Tracking.

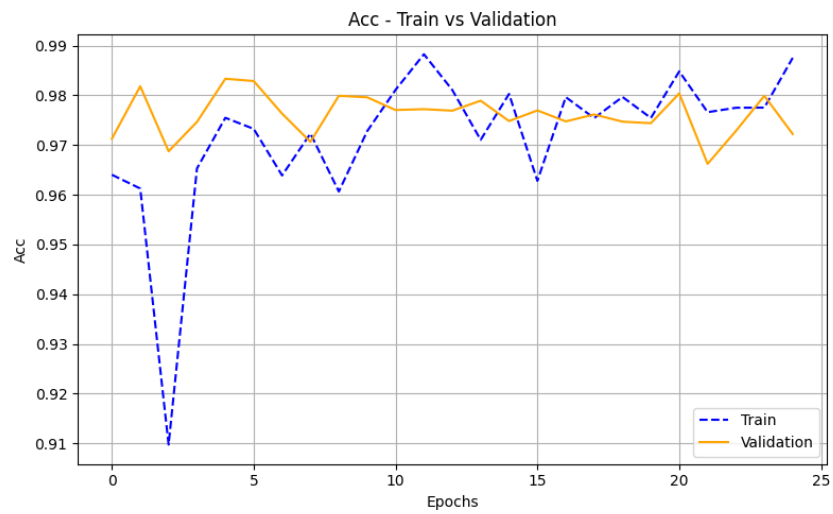


Figure 3.21: Accuracy Monitoring .

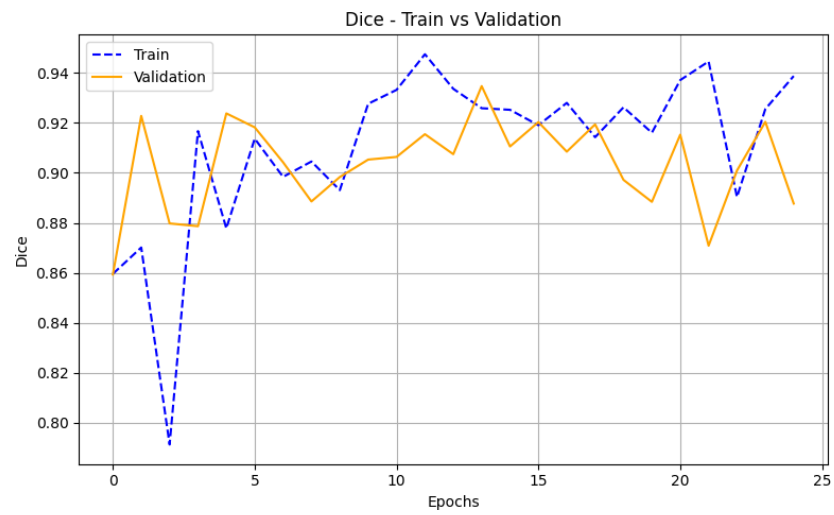


Figure 3.22: Dice Score Evaluation.

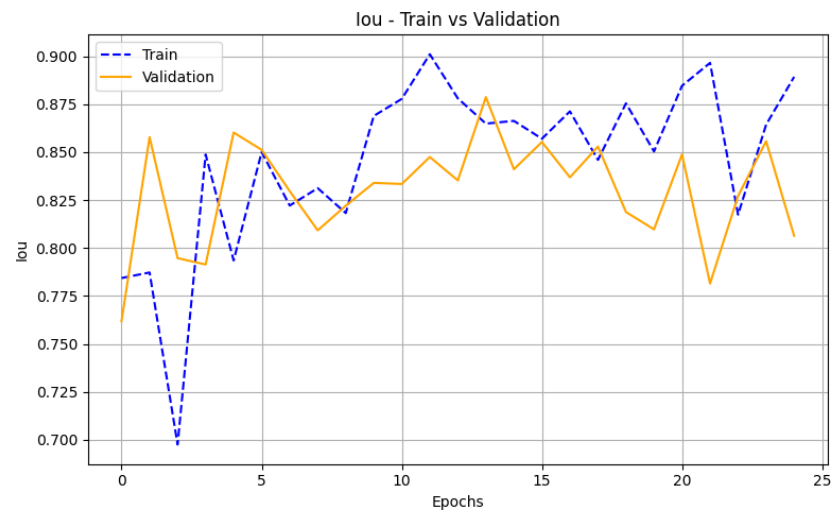


Figure 3.23: IoU Metric Evaluation.

3.5.2 Model prediction results

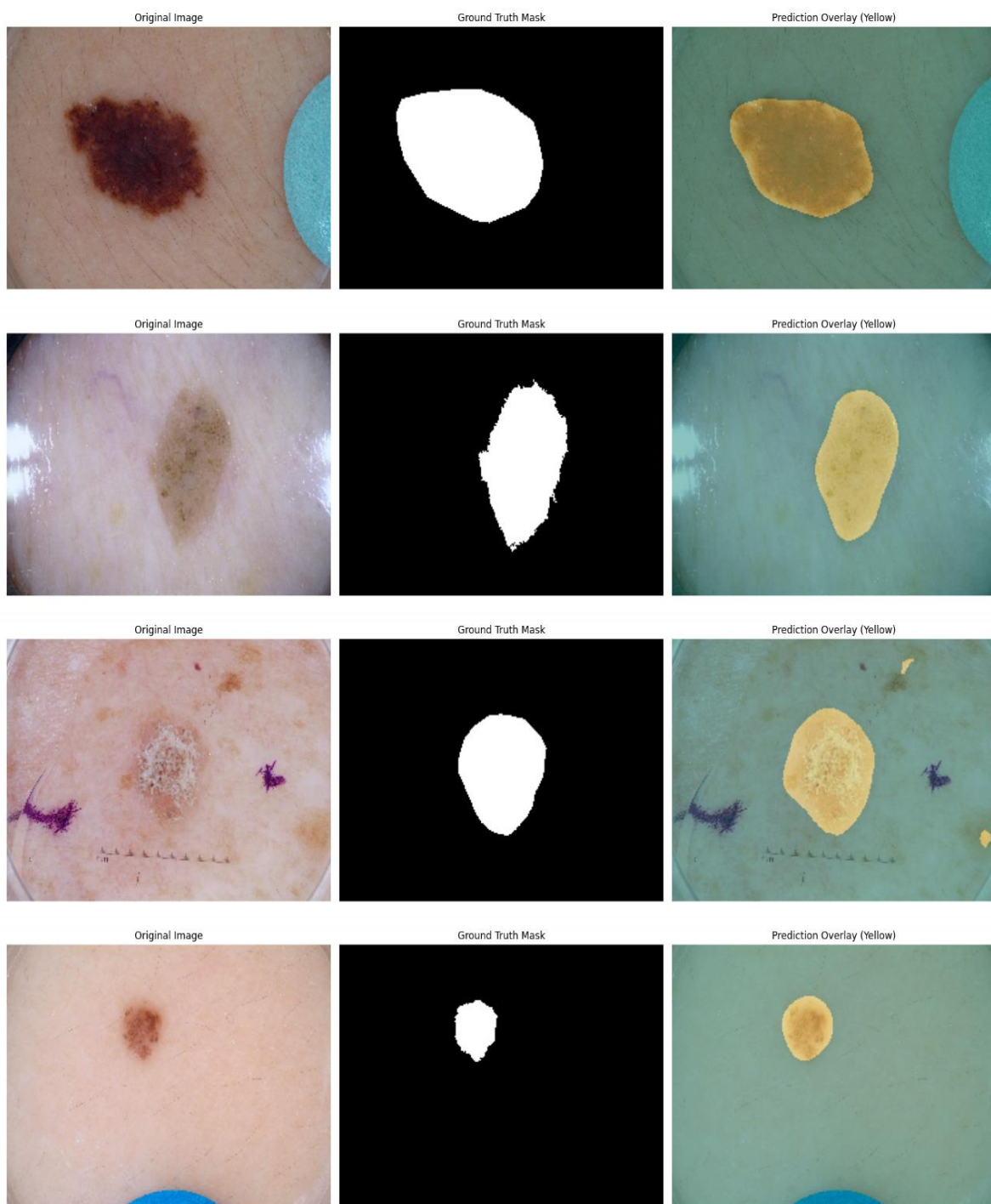


Figure 3.24: Qualitative Results.

3.5.3 Model prediction results analysis

Our model’s predicted overlays show strong alignment with the ground truth masks, particularly in cases where lesion boundaries are well-defined indicating robustness in segmenting large and clear features. Minor discrepancies tend to appear in regions with fine details or irregular patterns, which are common challenges in segmentation tasks and do not significantly affect the model’s overall accuracy.

To further enhance performance, expanding the training dataset with more diverse examples and refining the loss function could help improve edge detection. Quantitative metrics such as Intersection over Union (IoU) provide additional validation of these outcomes in numerical terms. Overall, the model demonstrates highly promising results, and with further fine-tuning, its performance could reach state-of-the-art standards.

3.5.4 Comparison with State-of-the-Art Models

To further assess the effectiveness of our proposed model, we compared its performance with several state-of-the-art segmentation models that have also been evaluated on the ISIC 2017 dataset. Table 3.4 provides a summary of their performance based on the Dice coefficient and Intersection over Union (IoU).

Model	Dice Coefficient	IoU	Accuracy
nnUnet [112]	0.921	0.801	0.957
DeepLabV3+ [113]	0.911	0.776	0.950
TransUNet [114]	0.919	0.790	0.955
Our model	0.89	0.81	0.97

Table 3.4: Comparison of skin lesion segmentation performance of different networks on ISIC 2017.

The evaluation results demonstrate the effectiveness of our proposed model in medical image segmentation for skin lesion detection. Compared to established models such as nnU-Net, DeepLabV3+, and TransUNet, our model achieves competitive performance with a Dice coefficient of 0.89, an IoU of 0.81, and the highest accuracy of 0.97. While nnU-Net shows the best Dice score (0.921), our

model outperforms all others in terms of IoU and overall accuracy, indicating a better balance between precision and recall across the segmentation task.

3.6 Discussion

The results of our experiments show that the proposed model performs well across several key evaluation measures. It achieved high accuracy, Dice score, and IoU, which indicates its strong ability to accurately segment different types of medical lesions. This good performance highlights the benefit of combining Convolutional Neural Networks (CNNs), which are good at capturing local details, with Transformer models, which are effective at understanding the global context. By merging these two approaches, the model becomes more accurate and can handle a variety of lesion shapes and sizes. Overall, these results show that the CNN-Transformer hybrid model is a promising solution for improving medical image segmentation tasks.

3.7 Conclusion

In this chapter, we detailed the full development process of our proposed skin lesion segmentation system, covering its architectural design, implementation, and experimental evaluation.

The core of our approach is a hybrid deep learning model that combines Convolutional Neural Networks (CNNs) with Transformer-based architectures, allowing the system to capture both local texture details and global contextual information.

The model was built using state-of-the-art tools and libraries and trained on the ISIC 2017 dataset following thorough preprocessing and fine-tuning. The observed results across multiple topics indicate highly promising outcomes when compared to existing deep learning models for other works.

General conclusion

In this work, we proposed a deep learning model to automatically segment skin lesions, addressing a key challenge in computer-assisted dermatology. The research began by explaining the medical background and the importance of detecting skin cancer early. We then reviewed the main concepts of deep learning, focusing on convolutional neural networks (CNNs), vision transformers (ViTs), and hybrid models that combine both. To overcome the limits of CNNs in capturing long-range information and the difficulty transformers face in detecting fine details, we designed a hybrid model using ResNet (a CNN) and DeiT (a transformer).

These components were combined through a BiFusion module, which helps integrate both local and global features.

When tested on the ISIC 2017 dataset, our model achieved better results than using CNNs or transformers alone, especially in segmentation accuracy, Dice coefficient, and Intersection over Union (IoU). These results show the strength of using a hybrid approach.

Despite the promising performance, there are still areas for improvement. These include:

- Improving accuracy on lesions with irregular and fine boundaries.
- Using larger and more varied datasets for training.
- Adding uncertainty estimation to help doctors understand prediction confidence.
- Using advanced data augmentation techniques.

In conclusion, building accurate skin lesion segmentation models is not only a technical success but also a critical step toward early and reliable skin cancer diagnosis. By improving how lesions are detected, these models can assist dermatologists in making better decisions, reducing diagnosis delays, and improving patient outcomes.

As skin cancer cases increase worldwide, improved segmentation tools can enhance screening efforts, increase access to care, and ultimately save lives through early detection.

References

- [1] Australasian College of Dermatologists, "Skin Structure and Function," [Online]. Available: <https://www.dermcoll.edu.au/atoz/skin-structure-function/>. [Accessed: 10-Avril-2025].
- [2] Skin Cancer Foundation, "Skin Cancer Information," [Online]. Available: <https://www.skincancer.org/skin-cancer-information/>. [Accessed: 24- Avril-2025].
- [3] "Skin Cancer," *Journal of Visualized Experiments (JoVE)*, [Online]. Available: <https://www.jove.com/science-education/v/14004/skin-cancer>. [Accessed: Avril 10, 2025].
- [4] "Skin Cancer Image Gallery," *American Cancer Society*, [Online]. Available: <https://www.cancer.org/cancer/types/skin-cancer/skin-cancer-image-gallery.html>. [Accessed: Avril 10, 2025].
- [5] "Living with cancer: Skin cancer diagnosis and treatment," *Mayo Clinic News Network*, [Online]. Available: <https://newsnetwork.mayoclinic.org/discussion/living-with-cancer-skin-cancer-diagnosis-and-treatment/>. [Accessed: Avril 10, 2025].
- [6] T. Fischer, "4 Different Types of Skin Cancer: Signs and Treatments," [Online]. Available: <https://www.drtrevanfischer.com/blog/4-different-types-of-skin-cancer-signs-and-treatments>. [Accessed: 11-Avril-2025].
- [7] Everyday Health, "What Are the Different Types of Skin Cancer?," [Online]. Available: <https://www.everydayhealth.com/skin-cancer/what-are-the-different-types-of-skin-cancer/>. [Accessed: 11-Avril-2025].
- [8] S. Hussain, I. Mubeen, N. Ullah et al., "Modern diagnostic imaging technique applications and risk factors in the medical field: A review," **IEEE/CAA Journal of Automatica Sinica**, vol. 8, no. 2, pp. 273–302, 2021, doi: [10.1155/2022/5164970] (<https://doi.org/10.1155/2022/5164970>).
- [9] R. M. Reilly, Ed., *Medical Imaging for Health Professionals: Technologies and Clinical Applications*, 1st ed. Hoboken, NJ: John Wiley & Sons, Inc., 2019.
- [10] H. Kittler, H. Pehamberger, K. Wolff, and M. Binder, "Diagnostic accuracy of dermoscopy," *The Lancet Oncology*, vol. 3, no. 3, pp. 159–165, Mar. 2002. [Online]. Available: [https://doi.org/10.1016/S1470-2045\(02\)00679-4](https://doi.org/10.1016/S1470-2045(02)00679-4)
- [11] "Dermoscopy Vitiligo Accurate Diagnosis Skin Endoscope Skin Detecting Instrument Vitiligo Diagnosis Analyzer Dermoscope," *Made-in-China.com*, [Online]. Available: <https://letymedical.en.madein-china.com/product/FOofynVJnAke/China-Dermoscopy-Vitiligo-Accurate-Diagnosis-Skin-Endoscope-Skin-Detecting-Instrument-Vitiligo-Diagnosis-Analyzer-Dermoscope.html>. [Accessed: 11-Avril-2025].

- [12] M. Rajadhyaksha, R. R. Anderson, and R. H. Webb, "Reflectance confocal microscopy of skin in vivo: From bench to bedside," *Lasers in Surgery and Medicine*, vol. 28, no. 5, pp. 404–413, 2001. [Online]. Available: <https://doi.org/10.1002/lsm.1185>
- [13] DermNet New Zealand, "Reflectance Confocal Microscopy," [Online]. Available: <https://dermnetnz.org/topics/reflectance-confocal-microscopy>. [Accessed: 12-Avril-2025]
- [14] A. Ulrich, D. Themstrup, and G. B. Jemec, "Optical coherence tomography in the diagnosis of skin cancer: a review," *Current Dermatology Reports*, vol. 8, no. 3, pp. 146–154, 2019. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6516952/>
- [15] Michelson Diagnostics, "Dermatological Optical Coherence Tomography (OCT) imaging available in Korea," VivoSight, 2023. [Online]. Available: <https://vivosight.com/dermatological-optical-coherence-tomography-oct-imaging-available-in-korea/>. [Accessed: 12-Avril-2025]
- [16] E. Wortsman, "Common applications of dermatologic sonography," *Journal of Ultrasound in Medicine*, vol. 31, no. 1, pp. 97–111, Jan. 2012. [Online]. Available: <https://doi.org/10.7863/jum.2012.31.1.97>
- [17] Plastic Surgery Key, "Ultrasound," *Plastic Surgery Key*, Feb. 2025. [Online]. Available: <https://plasticsurgerykey.com/ultrasound/>. [Accessed: 16-Avril-2025]
- [18] Y. Martínez-Escribano *et al.*, "Ultrasound in Skin Cancer: Why, How, and When to Use It?," *Cancers*, vol. 16, no. 19, 2024. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/39409920/>. [Accessed: 16-Avril-2025]
- [19] J. M. Kittler *et al.*, "The Accuracy of Skin Cancer Detection Rates with the Use of Dermoscopy," *J. Am. Acad. Dermatol.*, vol. 49, no. 1, pp. 69–74, 2003. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11460753/#sec9>. [Accessed: 20-Avril-2025]
- [20] American Academy of Dermatology, "Skin Cancer Statistics," AAD media resources, Apr. 2025. [Online]. Available: <https://www.aad.org/media/stats-skin-cancer>. [Accessed: 20-Avril-2025]
- [21] M. Dinnes *et al.*, "Sonography of the Primary Cutaneous Melanoma: A Review," *Ultrasound Med. Biol.*, vol. 37, no. 9, pp. 1487–1497, 2011. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/29180093/>. [Accessed: 20-Avril-2025]
- [22] A. Ahmed, G. Sun, A. Bilal, Y. Li, and S. A. Ebad, "Precision and efficiency in skin cancer segmentation through a dual encoder deep learning model," *Scientific Reports*, vol. 15, no. 1, Article 4815, 2025. [Online]. Available:

<https://www.nature.com/articles/s41598-025-88753-3>

- [23] G. Brancaccio, A. Balato, J. Malvey, S. Puig, G. Argenziano, and H. Kittler, "Artificial Intelligence in Skin Cancer Diagnosis: A Reality Check," *Journal of Investigative Dermatology*, vol. 144, no. 3, pp. 444–452, Mar. 2024. [Online]. Available: <https://doi.org/10.1016/j.jid.2023.10.004>
- [24] A. Alam, "What is Machine Learning?" ResearchGate, Aug. 2023. [Online]. Available: <https://www.researchgate.net/publication/373015635> [Accessed: Oct. 2023]. doi: 10.5281/zenodo.8231580
- [25] S. Dutt, S. Chandramouli, and A. K. Das, *Machine Learning*, 1st ed. Noida, India: Pearson India Education Services Pvt. Ltd., 2019. ISBN: 978-93-530-6669-7.
- [26] Malla Reddy College of Engineering & Technology, "Machine Learning [R17A0534] Lecture Notes B.Tech IV Year – I Sem (R17) (2020–21)," Dept. of Computer Science and Engineering, Secunderabad, India, 2020.
- [27] D. Paik and R. Naskar, "Types of Machine Learning – Lecture 9," CS460: Machine Learning, National Institute of Science Education and Research, Jan. 31, 2023.
- [28] B. Mahesh, "Machine Learning Algorithms – A Review," *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, Jan. 2020. doi: 10.21275/ART20203995.
- [29] T. O. Ayodele, "Types of Machine Learning Algorithms," in *New Advances in Machine Learning*, InTech, Feb. 2010. doi: 10.5772/9385. [Online]. Available: <https://www.researchgate.net/publication/221907660>
- [30] X. Ge, *Brief Introduction to Artificial Neural Networks*, reviewed by L. Oudre, Culture Sciences de l'Ingénieur, Jul. 2022.
- [31] *Deep Learning For Dummies®*, *Deep Instinct Special Edition*, John Wiley & Sons, Inc., Hoboken, NJ, 2018.
- [32] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [33] P. F. Christ *et al.*, "Automatic liver and tumor segmentation of CT and MRI volumes using cascaded fully convolutional neural networks," *arXiv preprint arXiv:1702.05970*, Feb. 2017.
- [34] I. A. Basheer and M. Hajmeer, "Artificial neural networks: Fundamentals, computing, design, and application," *J. Microbiol. Methods*, vol. 43, no. 1, pp. 3–31, 2000.
- [35] Dept. of Computational Intelligence (CSE-AIML, AIML, AI&DS), Malla Reddy College of Engineering & Technology (Autonomous), Maisammaguda, Dhulapally (Post Via Hakimpet), Secunderabad – 500100, Telangana, India. Affiliated to JNTUH, Hyderabad; Approved by AICTE; Accredited by NBA & NAAC ('A' Grade); ISO 9001:2015 Certified.
- [36] School of Electrical and Electronics, Department of Electronics and Communications, "Unit – I – Fundamentals of Artificial Neural Networks – SEC1609," *Course Material*, Sathyabama Institute of Science and Technology, Chennai, India, pp. 1–14.
- [37] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, Dec. 2015.

- [38] Purwono *et al.*, "Understanding of convolutional neural network (CNN): A review," *Int. J. Robot. Control Syst.*, vol. 2, no. 4, pp. 739–748, 2022. [Online]. Available: <http://pubs2.ascee.org/index.php/ijrcs>
- [39] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imaging*, vol. 9, pp. 611–629, Jun. 2018, doi: 10.1007/s13244-018-0639-9.
- [40] L. Wu and G. Perin, "On the importance of pooling layer tuning for profiling side-channel analysis," Delft University of Technology, The Netherlands, *arXiv preprint* arXiv:2106.06544, Jun. 2021.
- [41] C. A. Dias *et al.*, "Using the Choquet integral in the pooling layer in deep learning networks," in *Commun. Comput. Inf. Sci.*, vol. 912, pp. 153–164, 2018, doi: 10.1007/978-3-319-95312-0_13.
- [42] P. Gupta, "Activation functions in neural networks," *GeeksforGeeks*, Jul. 2021. [Online]. Available: <https://www.geeksforgeeks.org/activation-functions-neural-networks/>
- [43] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 12, pp. 310–316, Apr. 2020. [Online]. Available: <http://www.ijeast.com>
- [44] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.
- [45] J. Wang, N. I. R. Ruhaiyem, and P. Fu, "A comprehensive review of U-Net and its variants: Advances and applications in medical image segmentation," *Universiti Sains Malaysia*, 2023.
- [46] S. Bangar, "VGG Net architecture explained – The company: Visual Geometry Group," *Medium via Scribd*. [Online]. Available: <https://www.scribd.com/document/815573863/VGG-Net-Architecture-Explained-The-company-Visual-Geometry-Group-by-Siddhesh-Bangar-Medium>. [Accessed: May 21, 2025].
- [47] Ultralytics, "Residual Networks (ResNet)," *Ultralytics Glossary*, [Online]. Available: <https://www.ultralytics.com/glossary/residual-networks-resnet>. [Accessed: May 21, 2025].
- [48] Papers with Code, "DenseNet," *Papers with Code*, [Online]. Available: <https://paperswithcode.com/method/densenet>. [Accessed: May 21, 2025].
- [49] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, 2021. [Online]. Available: <https://doi.org/10.1186/s40537-021-00444-8>
- [50] F. A. Mohammed, K. K. Tune, B. G. Assefa, M. Jett, and S. Muhie, "Medical image classifications using convolutional neural networks: A survey of current methods and statistical modeling of the literature," *Mach. Learn. Knowl. Extr.*, vol. 6, no. 1, pp. 699–735, 2024. [Online]. Available: <https://www.mdpi.com/2504-4990/6/1/33>.
- [51] L.-Y. Wang, S.-G. Chen, and F.-T. Chien, "Robust deep convolutional neural network against image distortions," *APSIPA Transactions on Signal and Information*

- Processing*, vol. 10, p. e14, Oct. 2021. [Online]. Available: <https://doi.org/10.1017/ATSIP.2021.14>
- [52] Milvus, "How do neural networks optimize feature extraction?" *Milvus AI Quick Reference*, [Online]. Available: <https://milvus.io/ai-quick-reference/how-do-neural-networks-optimize-feature-extraction>. [Accessed: May 21, 2025].
- [53] H. Jia, J. Zhang, K. Ma, X. Qiao, L. Ren, and X. Shi, "Application of convolutional neural networks in medical images: A bibliometric analysis," *Quantitative Imaging in Medicine and Surgery*, vol. 14, no. 5, pp. 3501–3518, May 2024. [Online]. Available: <https://doi.org/10.21037/qims-23-1600>.
- [54] Zilliz, "What are the limitations of CNN in computer vision?" *Zilliz*, [Online]. Available: <https://zilliz.com/ai-faq/what-are-the-limitations-of-cnn-in-computer-vision>. [Accessed: May 21, 2025].
- [55] M. Khatri, Y. Yin, and J. Deogun, "Enhancing Interpretability in Medical Image Classification by Integrating Formal Concept Analysis with Convolutional Neural Networks," *Biomimetics*, vol. 9, no. 7, Art. no. 421, 2024. [Online]. Available: <https://doi.org/10.3390/biomimetics9070421>
- [56] Consensus, "What are the limitations of CNNs and CRNNs for hypertrophy detection in echocardiograms?," *Consensus*, [Online]. Available: https://consensus.app/search/what-are-the-limitations-of-cnns-and-crnn-for-hyp/mqf8DbCkREKDW_zKN1ZW8Q/. [Accessed: May 22, 2025].
- [57] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evol. Intell.*, vol. 15, no. 1, pp. 1–22, Jan. 2021, doi: 10.1007/s12065-020-00540-3.
- [58] Python Programming for Data Science, "Lecture 20 - Transformer Networks," *Fall 2023 Python Programming for Data Science*, [Online]. Available: https://fall-2023-python-programming-for-data-science.readthedocs.io/en/latest/Lectures/Theme_3-Model_Engineering/Lecture_20-Transformer_Networks/Lecture_20-Transformer_Networks.html [Accessed: May 22, 2025].
- [59] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv preprint arXiv:1706.03762*, Jun. 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [60] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training Data-efficient Image Transformers & Distillation through Attention," *arXiv preprint arXiv:2012.12877*, Dec. 2020. [Online]. Available: <https://arxiv.org/abs/2012.12877>
- [61] A. Raj, "Image Segmentation Using Vision Transformers (ViT): A Deep Dive with Cityscapes and CamVid Datasets," *Medium*, Oct. 26, 2024. [Online]. Available: <https://medium.com/@ankitrajsh/image-segmentation-using-vision-transformers-vit-a-deep-dive-with-cityscapes-and-camvid-datasets-fc1ccdca295b>
- [62] H. Anzum, M. N. S. Sammo, and S. Akhter, "Leveraging Data Efficient Image Transformer (DeiT) for Road Crack Detection and Classification," in *Proc. 2024 Int. Conf. on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*, Dhaka, Bangladesh, Mar. 2024, doi: 10.1109/iCACCESS61735.2024.10499539. [On

line]. Available: <https://www.researchgate.net/publication/380014684>.

[63] S. Dixit, "Swin Transformers for Semantic Segmentation: Part 1," *Medium*, Apr. 9, 2024. [Online]. Available: <https://medium.com/@srddev/swin-transformers-for-semantic-segmentation-part-1-bd85bad7e051>. [Accessed: May 22, 2025].

[64] C. Wei, S. Ren, K. Guo, H. Hu, and J. Liang, "High-Resolution Swin Transformer for Automatic Medical Image Segmentation," *Sensors*, vol. 23, no. 7, p. 3420, Mar. 2023. doi:10.3390/s23073420. [Online]. Available: <https://www.mdpi.com/14248220/23/7/3420>.

[65] W. Shi, P. Gao, and J. Xu, "SSFormer: A Lightweight Transformer for Semantic Segmentation," *arXiv preprint arXiv:2208.02034*, Aug. 2022. [Online]. Available: <https://arxiv.org/abs/2208.02034>.

[66] Hugging Face, "DeiT," *Transformers Documentation*, [Online]. Available: https://huggingface.co/docs/transformers/model_doc/deit. [Accessed: May 22, 2025].

[67] Toolify, "Advancements in Medical Imaging Segmentation: The Power of Transformers," *Toolify AI News*, Mar. 1, 2024. [Online]. Available: <https://www.toolify.ai/ai-news/advancements-in-medical-imaging-segmentation-the-power-of-transformers-2313313>. [Accessed: May 22, 2025].

[68] Q. Pu, Z. Xi, S. Yin, Z. Zhao, and L. Zhao, "Advantages of Transformer and Its Application for Medical Image Segmentation: A Survey," *BioMedical Engineering OnLine*, vol. 23, no. 14, Feb. 2024. doi: 10.1186/s12938-024-01212-4. [Online]. Available: <https://doi.org/10.1186/s12938-024-01212-4>.

[69] C. Wei, S. Ren, K. Guo, H. Hu, and J. Liang, "High-Resolution Swin Transformer for Automatic Medical Image Segmentation," *Sensors*, vol. 23, no. 3, pp. 1–21, 2024. doi: 10.3390/s23031056. [Online]. Available: <https://www.mdpi.com/1424-8220/23/3/1056>

[70] A. Khan, Z. Rauf, A. R. Khan, S. Rathore, S. H. Khan, N. S. Shah, *et al.*, "A recent survey of vision transformers for medical image segmentation," *arXiv preprint arXiv:2312.00634*, Dec. 2023. [Online]. Available: <https://arxiv.org/abs/2312.00634>.

[71] R. F. Khan, B.-D. Lee, and M. S. Lee, "Transformers in medical image segmentation: a narrative review," *Quantitative Imaging in Medicine and Surgery*, vol. 13, no. 12, pp. 8747–8767, 2023, doi: 10.21037/qims-23-542.

[72] M. M. Rahman, S. Shokouhmand, S. Bhatt, and M. Faezipour, "MIST: Medical Image Segmentation Transformer with Convolutional Attention Mixing (CAM) Decoder," *arXiv preprint arXiv:2310.19898*, Oct. 2023, accepted for publication at WACV 2024.

[73] M. Cossio, "Augmenting Medical Imaging: A Comprehensive Catalogue of 65 Techniques for Enhanced Data Analysis," *arXiv preprint arXiv:2303.01178*, Mar. 2023.

[74] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019, doi: 10.1186/s40537-019-0197-0.

[75] D. Bar-David, L. Bar-David, Y. Shapira, R. Leibu, D. Dori, A. Gebara, R. Schneur, A. Fischer, and S. Soudry, "Elastic deformation of optical coherence tomography images of diabetic macular edema for deep-learning models training: How far to go?" *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 11, pp. 1–9, Jul. 2023, doi: 10.1109/JTEHM.2023.3294904.

- [76] R. Singh, "Regularization in Machine Learning," *Medium*, Oct. 7, 2024. [Online]. Available: <https://medium.com/@RobuRishabh/regularization-in-machine-learning-79e2f87ce898>. [Accessed: May 22, 2025].
- [77] L. Harisha, "Dice Coefficient! What is it?" *Medium*, Feb. 19, 2023. [Online]. Available: <https://lathashreeh.medium.com/dice-coefficient-what-is-it-ff090ec97bda>. [Accessed: May 22, 2025].
- [78] SuperAnnotate, "Intersection over Union (IoU) for Object Detection," *SuperAnnotate Blog*, Jul. 20, 2023. [Online]. Available: <https://www.superannotate.com/blog/intersection-over-union-for-object-detection>. [Accessed: May 22, 2025].
- [79] D. Müller, I. Soto-Rey, and F. Kramer, "Towards a guideline for evaluation metrics in medical image segmentation," *BMC Research Notes*, vol. 15, no. 1, p. 52, Feb. 2022.
- [80] Cudo Compute, "Accuracy, Precision and Recall in Deep Learning," *Cudo Compute Blog*, Jul. 19, 2023. [Online]. Available: <https://www.cudocompute.com/blog/accuracy-precision-recall-in-deep-learning>. [Accessed: May 22, 2025].
- [81] P. Kashyap, "Understanding Precision, Recall, and F1 Score Metrics," *Medium*, Dec. 2, 2024. [Online]. Available: <https://medium.com/@piyushkashyap045/understanding-precision-recall-and-f1-score-metrics-ea219b908093>. [Accessed: May 22, 2025].
- [82] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, Sep. 1993, doi: 10.1109/34.232073.
- [83] N. Normand, "Hausdorff Distance," *McGill University, Computational Geometry Lab*, 1998. [Online]. Available: <https://cgm.cs.mcgill.ca/~godfried/teaching/cgprojects/98/normand/main.html>. [Accessed: May 22, 2025].
- [84] D. Bergmann and C. Stryker, "What is Loss Function?", *IBM Think*, Jul. 12, 2024. [Online]. Available: <https://www.ibm.com/think/topics/loss-function>. [Accessed: May 22, 2025].
- [85] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, 2016.
- [86] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4700–4708, 2017.
- [87] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Int. Conf. Learn. Represent. (ICLR)*, 2015.
- [88] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 6105–6114, 2019.
- [89] C. Szegedy et al., "Going Deeper with Convolutions," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1–9, 2015.
- [90] V.-C. Lungu-Stan, D.-C. Cercel, and F. Pop, "SkinDistilViT: Lightweight Vision Transformer for Skin Lesion Classification," *arXiv preprint arXiv:2308.08669*, 2023. [Online]. Available: <https://arxiv.org/abs/2308.08669>
- [91] K. Tang, J. Su, R. Chen, R. Huang, M. Dai, and Y. Li, "SkinSwinViT: A Lightweight

- Transformer-Based Method for Multiclass Skin Lesion Classification with Enhanced Generalization Capabilities,” *Applied Sciences*, vol. 14, no. 10, p. 4005, 2024. [Online]. Available: <https://www.mdpi.com/2076-3417/14/10/4005>
- [92] C. Karmakar, A. Khasnobish, and M. Nasipuri, “SkinViT: A Transformer-Based Method for Melanoma and Nonmelanoma Classification,” *PLOS ONE*, vol. 18, no. 12, p. e0295151, 2023. [Online]. Available: <https://doi.org/10.1371/journal.pone.0295151>
- [93] El-Shafai, W., & Yasin, H. (2024). *Boosting Skin Cancer Classification: A Multi-Scale Attention and Ensemble Approach with Vision Transformers*. <https://www.researchgate.net/publication/390830599>
- [94] A. Kumar, K. R. Kanthen, and J. John, “GS-TransUNet: Integrated 2D Gaussian Splatting and Transformer UNet for Accurate Skin Lesion Analysis,” *arXiv preprint arXiv:2502.16748*, Feb. 2025. [Online]. Available: <https://arxiv.org/abs/2502.16748>
- [95] E. K. Aghdam, R. Azad, M. Zarvani, and D. Merhof, “Attention Swin U-Net: Cross-Contextual Attention Mechanism for Skin Lesion Segmentation,” *arXiv preprint arXiv:2210.16898*, Oct. 2022. [Online]. Available: <https://arxiv.org/abs/2210.16898>
- [96] M. Ahmed, A. Khan, T. Alam, et al., “SkinEHDLF: A Hybrid Deep Learning Approach for Accurate Skin Cancer Classification in Complex Systems,” *Scientific Reports*, vol. 15, no. 1, pp. 1–14, 2025. [Online]. Available: <https://www.nature.com/articles/s41598-025-98205-7>
- [97] PlantUML, “State diagram,” [Online]. Available: <https://plantuml.com/state-diagram>. [Accessed: Jun. 8, 2025].
- [98] IBM, “Sequence diagrams,” [Online]. Available: <https://www.ibm.com/docs/en/rsm/7.5.0?topic=uml-sequence-diagrams>. [Accessed: Jun. 1, 2025].
- [99] Python Software Foundation, “The Python Language Reference,” [Online]. Available: <https://www.python.org/doc/essays/blurb/>. [Accessed: Jun. 1, 2025].
- [100] Google Research, “Google Colaboratory FAQ,” [Online]. Available: <https://research.google.com/colaboratory/faq.html>. [Accessed: Jun. 1, 2025].
- [101] PyTorch, “PyTorch Documentation,” [Online]. Available: <https://pytorch.org/docs/stable/index.html>. [Accessed: Jun. 1, 2025].
- [102] Torchvision, “Torchvision Documentation,” [Online]. Available: <https://pytorch.org/vision/stable/index.html>. [Accessed: Jun. 1, 2025].
- [103] Python Software Foundation, “os — Miscellaneous operating system interfaces,” [Online]. Available: <https://docs.python.org/3/library/os.html>. [Accessed: Jun. 1, 2025].
- [104] R. Wightman, “PyTorch Image Models,” [Online]. Available: <https://github.com/rwightman/pytorch-image-models>. [Accessed: Jun. 1, 2025].
- [105] Python Software Foundation, “argparse — Parser for command-line options,” [Online]. Available: <https://docs.python.org/3/library/argparse.html>. [Accessed: Jun. 1, 2025].
- [106] Pillow, “Pillow (PIL Fork) Documentation,” [Online]. Available: <https://pillow.readthedocs.io/>. [Accessed: Jun. 1, 2025].
- [107] OpenCV, “OpenCV Documentation,” [Online]. Available: <https://docs.opencv.org/>. [Accessed: Jun. 1, 2025].

- [108] NumPy, “NumPy Documentation,” [Online]. Available: <https://numpy.org/doc/>. [Accessed: Jun. 1, 2025].
- [109] Matplotlib, “Matplotlib Documentation,” [Online]. Available: <https://matplotlib.org/stable/contents.html>. [Accessed: Jun. 1, 2025].
- [110] Albumentations, “Albumentations Documentation,” [Online]. Available: <https://albumentations.ai/docs/>. [Accessed: Jun. 1, 2025].
- [111] Imageio, “Imageio Documentation,” [Online]. Available: <https://imageio.readthedocs.io/>. [Accessed: Jun. 1, 2025].
- [112] Isensee F., Jäger P.F., Kohl S.A.A., Petersen J., Maier-Hein K.H. Automated Design of Deep Learning Methods for Biomedical Image Segmentation. *Nat. Methods*. 2021;18:203–211. doi: 10.1038/s41592-020-01008-z .
- [113] Chen L.-C., Zhu Y., Papandreou G., Schroff F., Adam H. In: *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*. Ferrari V., Hebert M., Sminchisescu C., Weiss Y., editors. Springer International Publishing; 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation; pp. 833–851.
- [114] Chen J., Lu Y., Yu Q., Luo X., Adeli E., Wang Y., Lu L., Yuille A.L., Zhou Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv*. 2021 doi: 10.48550/arXiv.2102.04306.