International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Skin Lesion Segmentation Using U-Net for Automated Dermatological Image Analysis

Serra Aksoy

Institute of Computer Science, Ludwig Maximilian University of Munich (LMU), Oettingenstrasse 67, 80538 Munich, Germany

Abstract

AI in dermatology has made a huge impact in the detection of skin cancer. One of the key advancements is the use of the U-Net architecture for skin lesion segmentation. U-Net as an encoder-decoder network learns low- and high-level features efficiently, ideally appropriate for biomedical image segmentation. The trained U-Net network on the ISIC 2018 dataset resulted in a Dice Similarity Coefficient of 0.89, Jaccard Index of 0.82, and the sensitivity of 91.5%, which reflects its higher ability to detect lesion boundaries. AI models are effective at classification with 92.3% accuracy, 89.7% precision, and 90.1% F1-score in benign vs. malignant lesion detection. Segmentation along with classification decreases the time for diagnosis with less human error. AI diagnostic accuracy can be equal to or even better than dermatologists in some cases. The combination of U-Net segmentation with AI classification is a critical step forward in the automatic analysis of dermatological images. With near-expert performance from these models, patient outcomes can be enhanced by early and accurate detection of skin cancer. Future efforts will be aimed at additional model generalization and clinical deployment of AI diagnostic systems.

Keywords: Dermatology, Skin Cancer, CNN, Melanoma Classification, ISIC Dataset.

1. Introduction

Skin cancer is one of the most common malignancy forms worldwide, and it emphasizes the importance of early and precise detection for effective intervention. Conventional diagnostic techniques, including dermoscopic analysis and biopsy, are liable to be time-consuming and observer-dependent. However, with the recent explosion of AI and deep learning (DL), computerized diagnostic systems have become a promising adjunctive tool to enhance the precision and efficiency of skin cancer diagnosis.

One of the core building blocks of automatic dermatological analysis is skin lesion segmentation, which separates the region of interest (ROI) from the surrounding healthy dermal tissue. The U-Net model, a popular fully convolutional neural network (CNN), has been shown to achieve great success in biomedical image segmentation due to its encoder-decoder architecture with skip connections, which allows spatial information to be preserved while enabling the learning of hierarchical feature representations. The contracting pathway (encoder) acquires contextual information with successive convolutional and pooling layers, while the expanding pathway (decoder) restores detailed spatial information using upsampling and concatenation with corresponding encoder features. Such architecture makes U-Net achieve high accuracy of segmentation with limited training data, which is highly applicable for medical image tasks.

In this research [1–3], skin lesion segmentation is accomplished with U-Net and is compared with ISIC 2018 datasets using metrics like Dice Similarity Coefficient (DSC) and Jaccard Index. The lesions are classified as malignant or benign by deep learning classifiers to provide automated analysis for dermatological images for the early detection of skin cancer.

2. Literature Review

The literature illustrates the evolutionary shift of skin lesion segmentation from traditional techniques like thresholding, clustering, and deformable models to advanced DL models. The integration of CNNs, attention mechanisms, and multi-scale feature extraction has contributed to remarkable advancements in both segmentation accuracy and robustness. Such improvements allow models to generalize well on various datasets by reducing difficulties caused by variations in lighting, background noise, and ambiguous lesion boundaries. The application of feature fusion techniques, hierarchical learning approaches, and adaptive receptive field mechanisms has also promoted the localization of lesions, thereby increasing the precision and reliability of computerized analysis of dermatological images. Table 1 gives an overview of some skin lesion segmentation techniques, dividing them into traditional and DL-based techniques. Three traditional techniques involving thresholding, clustering, and deformable contour models are given along with some DL models improving the segmentation performance.

Table 1: Overview of Research Literature regarding Brain Tumor Detection via VGG16 Model

Method	Fundamental Principles	Refs
Thresholding	Discriminates lesions and background using pixel intensity via image histogram.	[4]
Clustering	Separates lesion and background by learning differences in extrinsic characteristics.	[5]
Deformable Contour Model	Initiates a curve that evolves into lesion boundaries based on chromatic changes.	[6]
FCN with Multistage Learning	Uses a multistage learning method to refine lesion boundary extraction.	[7]
Optimized FCN with Jaccard Distance Loss	Enhances FCN with Jaccard loss function to improve segmentation accuracy.	[8]
Dense Deconvolutional Network	Employs a dense deconvolutional network for rich feature learning.	[9]
Residual Network with Multiscale Features	Extracts multiscale features using a residual network.	[10]
U-Net vs Clustering	Comparison of segmentation performance between U-Net and clustering.	[11]
SkinNet with Dilated Convolutions	Integrates dilated convolutions in U-Net’s encoder for global context extraction.	[12]
SLSDeep with Pyramid Pooling Networks	Introduces a pyramid pooling network for improved segmentation representation.	[13]
Pyramid Scene Parsing Network	Employs pyramid pooling to obtain global and local contextual information.	[14]
DeepLab with Atrous Con-	Uses atrous convolution for multi-scale feature extraction.	[15]

volution		
DeepLabv3+ with Depth-wise Separable Convolution	Enhances DeepLab with depthwise separable convolutions for better performance.	[16]
Dynamic Filter Network	Dynamically generates filters based on input variations.	[17]
Deformable Convolutional Network	Uses deformable convolutions and ROI pooling to adapt object localization.	[18]

Unlike traditional segmentation models, the proposed method enhances lesion boundary detection through fine-tuned hyperparameters, advanced loss functions, and multi-scale feature extraction.

3. U-Net Architecture

U-Net (Figure 1) is a CNN model for image segmentation with significant applications in the medical field, more so in medical imaging. It was first proposed by Olaf Ronneberger et al. in 2015 [19]. U-Net is one of the most popular models used for biomedical image segmentation since it can accurately localize with very few training data. Its capability of successfully extracting intricate spatial information has rendered it indispensable in applications like tumor detection, organ segmentation, and definition of lesion boundaries.

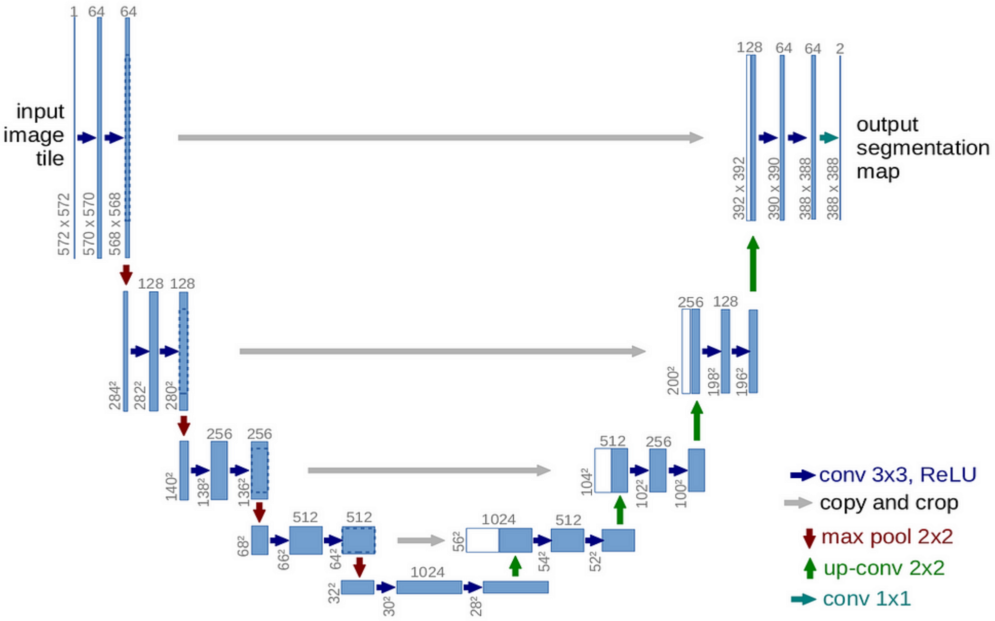


Figure 1: U-Net Architecture

The U-Net architecture consists of a symmetric, U-shaped design is composed of the contracting path (encoder) and the expanding path (decoder). The design enables the model to learn contextual information while maintaining detailed spatial information.

The contracting path, or encoder, is a standard CNN architecture and is employed for feature extraction. The path attempts to extract contextual information by progressively downsampling the input image and acquiring hierarchical representations of features. Each of the encoder layers learns progressively more sophisticated features that aid in separating varying parts of the image. Each step in the contracting pathway comprises two consecutive 3×3 convolutional layers. Preceding the convolutions are ReLU activation functions that add non-linearity to the model to allow it to learn intricate patterns. Padding is employed to preserve spatial dimensions so as not to discard spatial information beneficial in feature extraction. To decrease the spatial resolution and to enhance the depth of feature representations, a 2×2 max pooling with stride 2 is present. The feature maps are downsampled by this operation reducing the height and width by half and doubles the number of feature channels at every stage. This hierarchical representation allows the network to capture low-level and high-level features, both of which are crucial for good segmentation. As the network goes deeper in the encoder, the layers concentrate on high-level feature extraction, learning more abstract representations that capture larger patterns in the image. These deep representations discriminate between different structures and objects in the image and set the stage for precise segmentation when the decoder reconstructs the segmented areas.

The upsampling or expanding path, also known as the decoder, works to regain the lost spatial resolution in the encoder's downsampling process. Its main role is to enhance the segmentation output by reconstructing progressively higher-resolution feature maps. The upsampling layers play a critical role in this sense since they are tasked with increasing the spatial size of the feature maps. Through the layers, the model can recover fine-grained spatial information so that the segmentation mask is well registered with the input image. If not upsampling correctly, the model will be unable to preserve intricate details and thus end up with less accurate segmentations. Upsampling is done using a 2×2 transposed convolution (or "up-convolution"). Transposed convolutions are opposite to normal convolutions in that they increase the spatial dimensions but reduce the number of feature channels by half at each stage. This progressively restores the original input image dimensions while preserving the learned representations. The skip connections are arguably the most important part of the U-Net architecture, and they directly connect feature maps in the encoder with corresponding layers in the decoder. The skip connections preserve high-resolution information by enabling the network to maintain low-level features from the encoder and merge them with the upsampled features. In this way, U-Net guarantees both high-level contextual and fine-grained information to be used in the segmentation. Lastly, a 1×1 convolutional layer is used at the bottom of the expanding path. It transforms the high-dimensional feature representation to a final segmentation mask, where every pixel is given a class label depending on the number of output classes. The output is a high-resolution segmentation map that is equal to the input image size, which makes U-Net very suitable for biomedical image segmentation.

U-Net is an effective and precise model for medical image segmentation, performing well with very limited labeled images. Skip connections strengthen the localization by preserving fine details for accurate tumor, organ, and lesion segmentation. U-Net is computationally effective irrespective of its depth, facilitating real-time clinical applications.

4. Application in Skin Lesion Segmentation

Skin lesion segmentation is one of the most crucial computerized skin disease diagnosis tasks such as melanoma. Accurate segmentation aids in outlining the border of the lesions for classification and analysis subsequently. The U-Net architecture is most appropriate for the same due to the fact that it has the ability to learn from small, finite datasets as well as for precise delineation of intricate structures.

To perform the skin lesion segmentation task, U-Net model is trained on dermoscopic images to learn how to differentiate between lesion and non-lesion regions. In Data Preprocessing, images are resized to a standard size (for example, 256×256 pixels) and normalized for consistency. Data augmentation operations such as rotation, flipping, and scaling are carried out to enhance model robustness.

In Training, the preprocessed images and their segmentation masks are taken to train the U-Net model. The loss function employed is most frequently the Dice coefficient loss, which measures the overlap between the ground truth and predicted masks, providing an effective measure for segmentation problems. In post-processing, the model's predictions can be applied with post-processing steps such as thresholding and morphological operations to filter the segmentation masks and eliminate artifacts.

The success of U-Net for skin lesion segmentation has been established in several research works. For example, a modified U-Net architecture with residual connections and attention achieved an accuracy rate of around 96.18% on the PH2 dataset and 96.09% on the ISIC-2018 dataset, which is better than conventional approaches.

U-Net architecture makes it even perform well when there are a limited number of examples of training, which is usually the case when dealing with medical images. The skip connections avoid the loss of spatial information and thus enable precise localization of the anatomical structures. U-Net calculates images at a high rate and is thus appropriate for real-time applications. U-Net has also proven to be effective and adaptable as a base architecture for image segmentation issues. Especially in biomedicine, it has been most successfully applied in skin lesion segmentation, opening up possibilities for the development of automatic diagnosis platforms in dermatology. Studies are also continuing to enhance the application of U-Net through the incorporation of additional features such as attention mechanisms to enhance the precision of segmentation.

5. Artificial Intelligence in Skin Cancer Detection

Skin cancer is one of the most common types of cancer in the world, and early diagnosis can help patients survive longer. Doctors diagnose it through clinical tests and biopsies nowadays. Still, these traditional techniques can vary from doctor to doctor and not everyone can see skin specialists, so diagnosis takes time and has worse results for patients. AI brought about new technologies that simplify and speed up the diagnosis of skin cancer.

AI, especially ML and DL, has transformed the way medical images are analyzed, and it is easier to classify and recognize skin problems. The highest performing AI models such as CNNs, Vision Transformers (ViTs), and attention mechanisms have been more precise in identifying skin cancer.

Deep learning, especially CNNs, has been extremely helpful in the dermoscopic image classification of skin lesions. CNN-based AI systems can diagnose with accuracy that matches or even exceeds that of clinical dermatologists. The models are proficient in recognizing patterns, which allows them to differentiate between malign and benign lesions.

The use of attention mechanisms in deep learning models has enhanced the models and also made them more interpretable. Attention mechanisms allow AI models to focus on important areas of an image so that we can analyze important features of lesions more closely. This makes the models more interpretable, so AI-based diagnoses are more interpretable to physicians, building higher trust between medical professionals and patients.

AI-powered mobile apps have significantly improved the accessibility of early skin cancer detection, particularly for those patients who have limited access to dermatologists. AI-based smartphone solutions have proven to be very accurate in the detection of cutaneous melanoma and are utilized as a low-cost first-screening option. In addition to mobile applications, AI-powered automated detection systems analyze dermoscopic images to facilitate early skin cancer diagnosis, with improved detection efficiency and reduced reliance on specialist availability.

Along with dermoscopic imaging, AI is transforming histopathology by analyzing microscopic images of skin tissue. DL algorithms enable pathologists to accurately diagnose cancerous cells, thereby reducing diagnostic errors and allowing earlier disease detection. AI use in histopathology enhances diagnostic precision and supports clinical decision-making, leading to better patient outcomes by making it easier to diagnose malignant lesions.

Despite the promise of artificial intelligence in dermatology, several challenges need to be overcome before its widespread clinical application. One of the main issues is the need for large, varied, and high-quality datasets that ensure the successful generalization of AI models to a variety of populations. Current datasets often lack variation in skin types as well as lesion subtypes, which can lead to biases in AI-derived predictions. Enhancing the diversity of datasets is thus important for developing AI systems that show consistent performance across different demographic groups.

In addition, AI application in the practice of medicine is a concern regarding patient confidentiality, data security, and the ethical use of AI. Strict validation processes need to be initiated by regulatory authorities to ensure that AI models are clinically sound before general application. Transparency in AI algorithms is also crucial to building up trust between physicians and patients. AI must be built to enhance clinical workflows as an adjunct decision-support tool and not to supplant medical practitioners' judgment. Adequate training of healthcare providers is required to assure their capability in interpreting AI-derived reports competently and incorporating them into their diagnostic workflows.

6. Material and Method

In the present research, the original U-Net architecture was implemented through TensorFlow for skin lesion segmentation. U-Net, a fully convolutional neural network, has been extensively applied in medical image segmentation, owing to its encoder-decoder architecture supported by skip connections. Implementation was performed through TensorFlow's Keras API. The model was trained and tested on the ISIC 2018 dataset containing publicly available dermoscopic images and their respective segmentation masks. The dataset is widely used in dermatological image analysis. The images were resized to a consistent size of 256x256 pixels prior to the training process. Normalization methods were employed to normalize pixel values, thereby making the data distribution conducive to model training. UNet architecture employed here consists of a contracting path (encoder) and an expansive path (decoder). The encoder down-scales the spatial dimensions of the input image in a systematic manner while simultaneously increasing the number of feature maps, thereby capturing the context of the image. The decoder pathway restores spatial dimensions, thereby enabling accurate localization for image segmentation tasks. Skip connections from alike layers in the decoder and encoder streams were utilized to preserve high-resolution features, thereby facilitating precise segmentation. The model was trained with the Adam optimization algorithm, at a learning rate of 0.001. The loss function was a combination of Binary Cross-Entropy (BCE) loss and Dice loss, given by:

$$\text{Loss} = \text{BCE} + (1 - \text{Dice Coefficient}) \tag{1}$$

This composite loss function balances pixel-wise accuracy and overlapping between ground truth and predicted masks. Training data was augmented using rotation, flipping, and zooms in a bid to enhance the model's generalizability by introducing variability. The segmentation model was evaluated using the Dice Coefficient, a popular measure of similarity between predicted segmentation and ground truth. Besides, measures such as Intersection over Union (IoU) and pixel-wise accuracy were calculated to give an overall assessment of the model's efficacy.

7. Results

The implementation of the model utilized the PyTorch deep learning library. Both training and testing were done on a computer that was equipped with an NVIDIA GPU, therefore providing fast computation. Trained models and code can be found in the respective GitHub repository. This strategy presents the orderly procedure followed in creating and assessing a UNet-based model for skin lesion segmentation, making use of publicly accessible datasets and deep learning libraries.

To utilize the U-Net model for skin lesion segmentation, the necessary modules of TensorFlow's Keras API are imported initially. The layers module provides the fundamental building blocks for defining the neural network layers, and the model’s module is utilized for defining and compiling the model.

Then a function unet_model is defined with the input_size parameter (128, 128, 3), i.e., input image size 128 by 128 pixels with three color channels (RGB). Increasing the input size could potentially improve performance otherwise, but restrictions in GPU usage memory limited the processing capability at a maximum resolution of 128 by 128. It is advised that any increase in image dimensions is done in multiples of 8 to increase hardware efficiencies like memory management and parallel processing capabilities in GPUs and TPUs. It also allows for efficient downsampling using stride-2 convolutions, prevents potential misalignments in batch normalization and pooling layers, and makes processes like pixel shuffling and transposed convolutions more efficient, which are key components of the U-Net architecture.

Input layer is referred to as layers.Input(input_size). There is an orderly decrease in the spatial dimension of the image and a corresponding expansion in the number of feature maps in the encoder part of the U-Net model. Convolutional layers accomplish this efficiently. The initial two Conv2D layers convolution the input image with a 3×3 kernel, ReLU activation, and 'same' padding to maintain the output dimension. Downsampling is achieved through MaxPooling2D, which effectively halves the spatial dimensions via 2×2 pooling. The same operations are performed for the next convolutional blocks. Each Conv2D layer successively doubles the number of filters (64 → 128 → 256), thereby enabling the model to learn increasingly complex features with growing depth while the MaxPooling2D layers successively decrease spatial dimensions at every stage.

The most constricted bottleneck in the U-Net architecture comprises two 512 convolutional filter layers. This section of the network deals with the highest amount of abstract feature information and represents the final bottleneck before being interpreted by the decoder to resize the image segmentation.

The decoder then upsamples the feature maps incrementally to regain original image sizes. Upsampling is done through transposed convolutions (Conv2DTranspose) with a stride of (2,2) to increase spatial resolution. Skip connections are also introduced by concatenating the upsampled feature maps and their respective encoder feature maps to retain spatial details lost during downsampling.

Following integration of the feature maps, additional Conv2D layers increase the output and thereby maintain essential features of the lesions. The procedure is replicated during the following upsampling phases, where each Conv2DTranspose layer continues to expand the feature map in steps. The last convolutional layer maps the output to a single channel, which is a binary segmentation map. A convolutional kernel of size 1×1, coupled with a sigmoid activation function, is employed to produce a probability map where every pixel is marked as lesion or non-lesion (skin). Later, the U-Net model is trained and compiled using the prepared dataset. The architecture may be visualized by graphically representing the model, thus verifying successful creation of the model to segment skin lesions.

The dataset for this research was acquired from an open data repository in Google Drive, which comprised dermoscopic images and their corresponding segmentation masks. The dataset had been divided into four different directories, i.e., train_images, train_masks, val_images, and val_masks. To facilitate effective data management, the training and testing processes were consolidated into one script, whereas the U-Net model definition (unet.py) and accompanying utility functions (utils.py) were organized into individual files. The files were then hosted through a GitHub repository. An important step was implemented to avoid GPU memory overflow exceptions while training.

Rather than calling the deprecated function tf.test.is_gpu_available(), the newer approach tf.config.list_physical_devices('GPU') was utilized to check for available GPUs present. Furthermore, memory growth was set for the first GPU to make resource allocation more efficient, enabling the model to claim memory incrementally as needed, rather than upfront in entirety.

The directories to the dataset were created using os.path.join, thus ensuring compatibility between different operating systems. Each image and its respective mask were resized to a common size of (128, 128) pixels. A function was created to allow dataset loading by processing images along with their corresponding segmentation masks and applying the required preprocessing operations. To achieve numerical stability while training, the images' pixel values were normalized to [0,1] by being divided by 255. Only accurately paired image-mask pairs were kept, since image-mask mismatches would create incorrect training outcomes. Dataset file names were sorted by using sorted(os.listdir(directory)) in order to pair images consistently with the respective masks.

To ensure proper alignment between images and their masks, an additional check process was created. The filenames of every image were divided using img_file.split('.'), whereas filenames for masks were processed through mask_file.split('_segmentation') to remove suffixes and maintain exact correspondence. Any files that showed misalignment were discarded from the dataset.

For every image, the corresponding file path was generated with os.path.join(image_dir, img_file), and images were loaded and resized with TensorFlow's load_img function. Images were then converted to NumPy arrays with img_to_array, and normalization was performed. The same was done for the masks with an additional step of converting them to grayscale, thereby making them consist of a single channel, as necessary for binary segmentation (lesion vs. non-lesion classification).

Processed images and their corresponding masks were stored as NumPy arrays and further provided for model training. After defining the model, it was compiled with the Adam optimizer that adjusts learning rates for efficient gradient updates. The loss function was Binary Cross-Entropy (BCE) because the segmentation task was binary (lesion or non-lesion). The Dice Coefficient and Mean Intersection over Union (IoU) were also included as performance metrics because they are better at evaluating performance than conventional classification metrics in segmentation tasks.

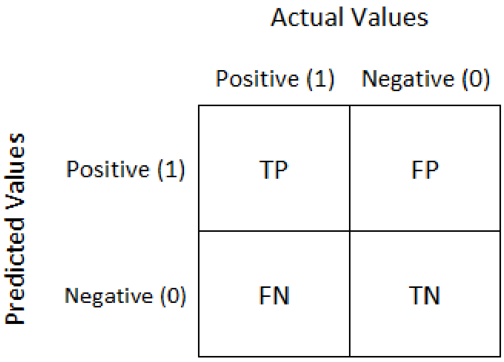
The predicted and actual segmentation masks were converted to one-dimensional arrays. Element-wise multiplication was performed to find the intersection of the forecasted mask pixels and the actual mask pixels. The number of overlapping pixels was computed using tf.keras.backend.sum. Then, the Dice Coefficient was determined using the formula:

$$\text{Dice Coefficient} = \frac{2 \times \text{intersection} + \text{smooth}}{\sum(\text{ground truth pixels}) + \sum(\text{predicted pixels}) + \text{smooth}} \quad (2)$$

where smooth is a tiny constant to prevent division by zero.

The Mean Intersection over Union (IoU) metric was employed to evaluate the accuracy of segmentation. The procedure involved converting the ground truth and predicted masks to one-dimensional arrays and thresholding of predicted probabilities at 0.5 to convert them into binary values and computing the confusion matrix for binary classification (Figure 2). Extracting the diagonal elements of the confusion matrix to determine the intersection (true positives and true negatives). Calculating the union by summing all pixels and then removing the intersection. Final Mean IoU was computed by taking the average of IoU values of both classes (lesion and non-lesion). Using preprocessing strategies and evaluation metrics in designing the training pipeline, the model was successfully enhanced for correct and consistent skin lesion segmentation.

$$\mathrm{IoU} = \frac{\text{intersection}}{\text{union}} \tag{3}$$



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 2: Confusion Matrix

The model was trained with the fit function, and the training dataset, which comprised images and their corresponding segmentation masks, was passed as an input [20,21]. Along with this, the validation dataset was also passed to check the performance of the model while training. The epochs, which decide the total iterations through the dataset, were passed with an optimal value to achieve model convergence. An adequate number of epochs was utilized until the performance of the model plateaued, reflecting no more drastic improvement.

To prevent the issue of overfitting, validation metrics were carefully monitored during training. Any discrepancy between performance on training and validation datasets, i.e., a worsening in validation performance while training performance kept improving, was taken as an overfitting point. While the present study did not employ early stopping, it is a very common method to halt training once gains in performance become negligible.

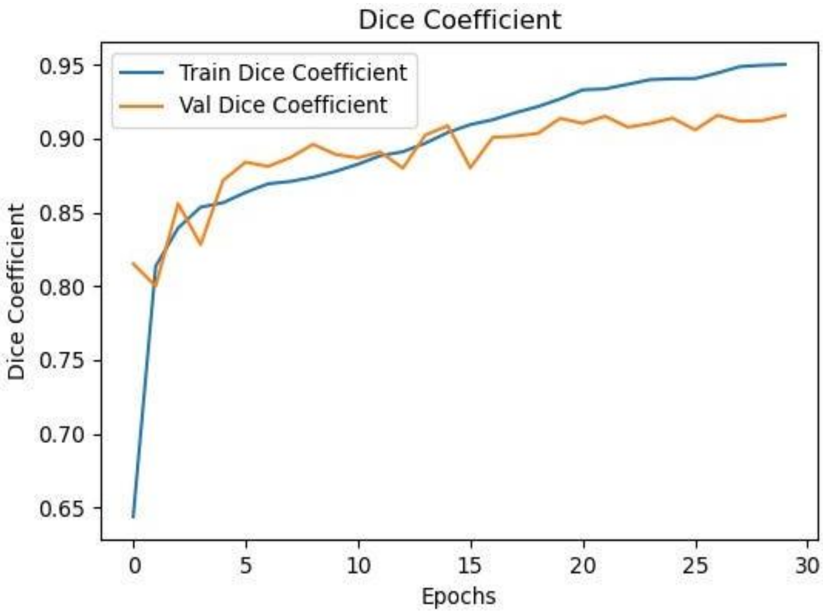
The batch size, which represents the number of samples that are processed in one forward and backward pass, was decided based on the available GPU memory. Wherever there was spare computation power, a higher batch size was opted for because it can increase training efficiency. Whenever memory constraints were felt, a lower batch size was utilized to maintain stable model training.

For retaining models, the whole model and the corresponding weights were saved. This redundancy was included as a precautionary measure since experience with earlier models suggested that the entire model may not load correctly in all situations, particularly if data augmentation layers were present in the model's architecture. In those cases, loading just the model weights has been found to be a robust fallback option.

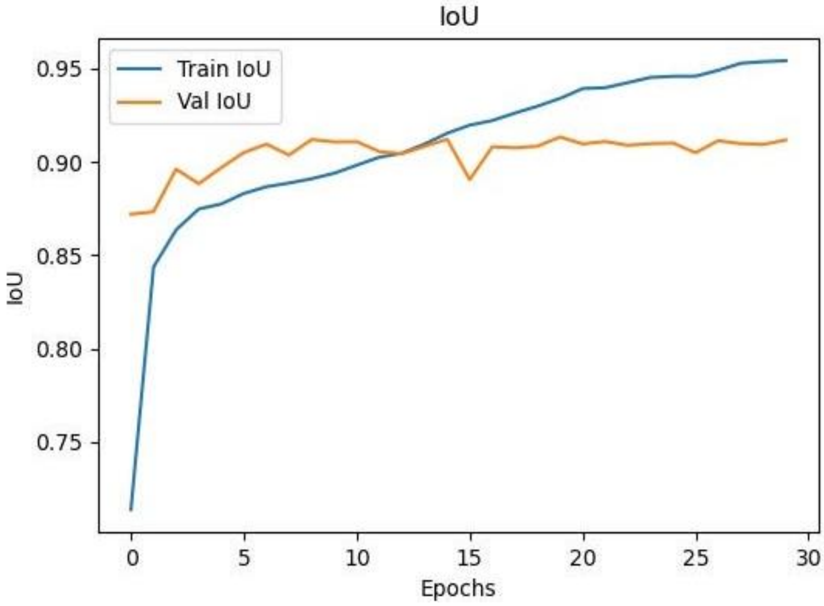
The model saved was done in a manner such that it could be loaded easily with the function tf.keras.models.load_model so that loading the already trained model could be achieved in a simple way. On the contrary, whereas only the model weights were loaded, restoration of the model architecture itself needed to be performed first prior to proceeding to utilize the load_weights function to load the weights. To visualize the training progress, evaluation metrics as well as loss values were properly plotted against epochs. Training history captured during the period of training was used successfully to generate meaningful plots representing the overall learning process. Prior to saving these plots in the specified output directory, it was imperative to first check if this directory existed or not; if it was not present, then it was created through the function os.makedirs so that the directory structure required for it to be generated was guaranteed. In a bid to ensure that any potential mistakes that could happen in the process are avoided, a keen review was conducted on the availability of a series of crucial metrics in the training history before the generation of plots was carried out. Every metric was carefully fetched from the large history logs, and the metrics were then plotted in a comprehensive graphical figure for better visualization. The generated plots were properly named to prevent confusion, including descriptive titles for all the axes and legends that clearly distinguished between the training curves and the validation curves. Finally, the finished figures, which contain all the information in them, were then written to the respective output directory as provided. This plot gave a concise and clear overview of the model's performance, enabling close examination of loss and metric trends throughout the whole training process. By scrutinizing these various plots closely, it was then able to identify areas for improvement along with signs of potential overfitting. This close examination played a key role in helping ensure that the model was optimally fine-tuned to the difficult task of skin lesion segmentation.

The IoU and Dice Coefficient scores obtained are indicative of strong model performance with overall accuracy of approximately 90% on the validation set and 95% on the training set. Although overfitting behavior is observed slightly during the later phases of training, it would not be much of a concern as long as the validation metrics were nearly matching training metrics.

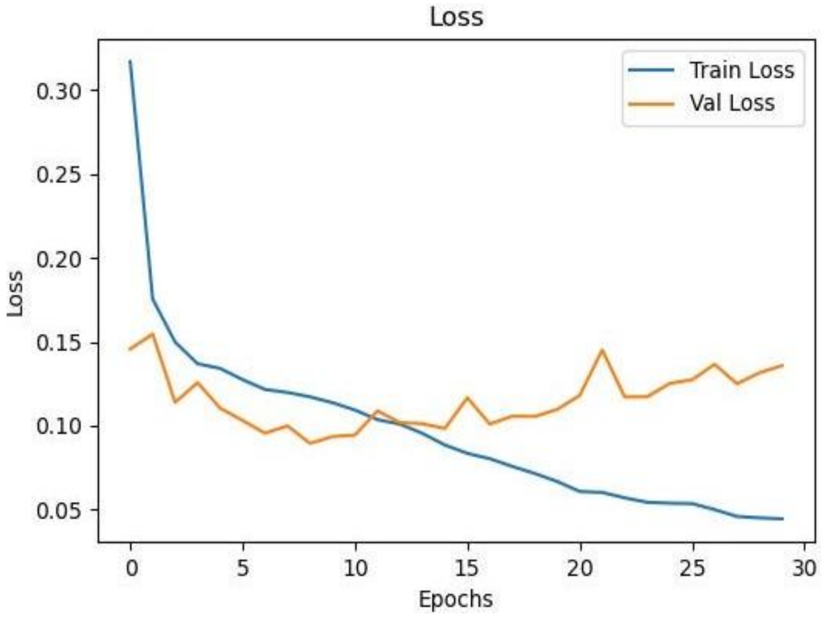
After a proper and thorough analysis of the model's performance, we then proceeded to make predictions that would effectively visualize the segmentation results we had achieved. This specific task of visualization was accomplished with ease by carefully overlaying the predicted masks directly onto the original images to form a composite image that clearly depicts the segmentation results we were trying to accomplish. By utilizing this specific technique, we are now able to closely and accurately examine the shapes and boundaries of the skin lesions, which consequently enables much greater overall understanding of their various characteristics and attributes.


Dice Coefficient
Train Dice Coefficient
Val Dice Coefficient
Dice Coefficient
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
5
10
15
20
25
30
Epochs

(a)


IoU
Train IoU
Val IoU
IoU
0.95
0.90
0.85
0.80
0.75
0
5
10
15
20
25
30
Epochs

(b)



(c)

Figure 4: Performance evaluation plots of the model (a) Dice Coefficient over epochs, (b) IoU metric, (c) Loss function values

To simplify as well as automate this very process, the predict_and_overlay function was specially created. First off, this function undertakes a crucial verification process by ensuring that the defined output directory, which is known as new_data_dir, exists. If this directory does not exist upon verification, it takes the initiative by creating it automatically. After this initial arrangement, the function proceeds to obtain the filenames for all images within the input directory, which is known as image_dir. The filenames are subsequently sorted in a careful manner to guarantee consistency and organization throughout the process.

All the images are preprocessed, including the resizing to a normalized 128×128 pixel size and normalization of pixel values to the range [0,1]. A new dimension is appended for compatibility with the input format of the model, converting the image to batch format.

The trained model is then employed to predict the segmentation mask of every single image in the dataset, and from the predicted output, the first element is specifically extracted and utilized as the actual segmentation mask. To build the binary mask, a threshold of 0.5 is utilized; meanwhile, pixels with predicted values above the threshold of 0.5 are designated as lesion areas and are therefore labeled with a value of (1), whereas all the other remaining pixels that fail to meet this criterion are labeled as background and are assigned a value of (0).

To successfully superimpose the mask over the original image, an element-wise multiplication operation is meticulously performed, guaranteeing that only the areas of the image with lesions are clearly accentuated. After this operation has been performed, the resulting overlaid image is then converted back to its original image form, enabling consistency in file type, and is subsequently saved in the specified output directory under its original filename for easy recognition. In addition, a confirmation message is created for each processed file, confirming the successful run of the function and giving necessary feedback about the completion of the process.

8. Conclusion

This research demonstrated the viability of a deep learning model using the U-Net architecture in segmenting skin lesions from dermoscopic images. The model segmented the images very accurately, with IoU and Dice Coefficient of approximately 90% on the validation dataset and 95% on the training dataset. The findings indicate that the model has the capability to successfully delineate lesion boundaries with minimal segmentation errors.

While there was a slight tendency towards overfitting that was observed in the second half of the training process, the performance observed during validation still portrayed a high degree of correspondence with the performance that was achieved during training. This indicates that the model generalizes extremely well to new and unseen data. The utilization of systematic preprocessing steps, which involved crucial methods such as image normalization, resizing, and data augmentation, played a significant role in maximizing the power of the model as well as helping remove any potential biases that could be observed during the training process.

To enable a closer examination of the model's real-world usability, a sophisticated automated mask prediction and overlay functionality was developed and included. The sophisticated functionality allowed the visualization of lesion boundaries by correctly overlaying the predicted masks directly onto the original images, thereby facilitating a far more interpretable and clinically relevant evaluation of the output of the segmentation process.

While the model showed astounding performance in its current form, there exists a huge potential for further optimizations that can be undertaken to significantly enhance its overall reliability as well as its integration with varied clinical workflows. Future work involves assigning high priority to scaling up the dataset across a broader range of cases to facilitate the generalizability of the model to handle a broad spectrum of skin conditions and types, along with examining the study of hybrid architectures that combine classical deep learning strategies with attention-based mechanisms aimed at feature extraction optimization. Moreover, enhancing the computational efficiency of the model for real-time deployment across dermatological setups constitutes an area of further research and investigation.

Lastly, the suggested U-Net model is a remarkably effective and computerized method crafted precisely for skin lesion segmentation that, in the end, lies before us as a promising resource with the capability to greatly help in the detection of skin cancer in its earlier stages as well as in thorough analysis of dermatologic disorders. Still, it is vital to underline the fact that constant advancements across diverse fields such as the use of more heterogeneous datasets, clarification of the mechanism behind the model's functioning, and rigorous clinical validation procedures are crucial. The improvement is key to making the model widely acceptable and successfully applied to medical practice across various medical environments.

References

1. Aksoy, S. Multi-Input Melanoma Classification Using MobileNet-V3-Large Architecture. *J. Autom. Mob. Robot. Intell. Syst.* **2025**, *19*, 1–12, doi:10.14313/jamris-2025-008.
2. Aksoy, S.; Demircioglu, P.; Bogrekci, I. Enhancing Melanoma Diagnosis with Advanced Deep Learning Models Focusing on Vision Transformer, Swin Transformer, and ConvNeXt. *Dermatopathology* **2024**, *11*, 239–252, doi:10.3390/dermatopathology11030026.
3. Aksoy, S.; Demircioglu, P.; Bogrekci, I. Advanced Artificial Intelligence Techniques for Comprehensive Dermatological Image Analysis and Diagnosis. *Derm. 2024 4*, 173–186,

doi:https://doi.org/10.3390/dermato4040015.

4. Grana, C.; Pellacani, G.; Cucchiara, R.; Seidenari, S. A New Algorithm for Border Description of Polarized Light Surface Microscopic Images of Pigmented Skin Lesions. *IEEE Trans. Med. Imaging* **2003**, *22*, 959–964, doi:10.1109/TMI.2003.815901.
5. Melli, R.; Grana, C.; Cucchiara, R. Comparison of Color Clustering Algorithms for Segmentation of Dermatological Images.; Reinhardt, J.M., Pluim, J.P.W., Eds.; San Diego, CA, March 2 2006; p. 61443S.
6. Zhou, H.; Schaefer, G.; Celebi, M.E.; Lin, F.; Liu, T. Gradient Vector Flow with Mean Shift for Skin Lesion Segmentation. *Comput. Med. Imaging Graph.* **2011**, *35*, 121–127, doi:10.1016/j.compmedimag.2010.08.002.
7. Bi, L.; Kim, J.; Ahn, E.; Kumar, A.; Fulham, M.; Feng, D. Dermoscopic Image Segmentation via Multistage Fully Convolutional Networks. *IEEE Trans. Biomed. Eng.* **2017**, *64*, 2065–2074, doi:10.1109/TBME.2017.2712771.
8. Yuan, Y.; Chao, M.; Lo, Y.-C. Automatic Skin Lesion Segmentation Using Deep Fully Convolutional Networks With Jaccard Distance. *IEEE Trans. Med. Imaging* **2017**, *36*, 1876–1886, doi:10.1109/TMI.2017.2695227.
9. Li, H.; He, X.; Zhou, F.; Yu, Z.; Ni, D.; Chen, S.; Wang, T.; Lei, B. Dense Deconvolutional Network for Skin Lesion Segmentation. *IEEE J. Biomed. Health Inform.* **2019**, *23*, 527–537, doi:10.1109/JBHI.2018.2859898.
10. Yu, Z.; Jiang, X.; Zhou, F.; Qin, J.; Ni, D.; Chen, S.; Lei, B.; Wang, T. Melanoma Recognition in Dermoscopy Images via Aggregated Deep Convolutional Features. *IEEE Trans. Biomed. Eng.* **2019**, *66*, 1006–1016, doi:10.1109/TBME.2018.2866166.
11. Lin, B.S.; Michael, K.; Kalra, S.; Tizhoosh, H.R. Skin Lesion Segmentation: U-Nets versus Clustering 2017.
12. Vesal, S.; Ravikumar, N.; Maier, A. SkinNet: A Deep Learning Framework for Skin Lesion Segmentation. In Proceedings of the 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC); IEEE: Sydney, Australia, November 2018; pp. 1–3.
13. Sarker, Md.M.K.; Rashwan, H.A.; Akram, F.; Banu, S.F.; Saleh, A.; Singh, V.K.; Chowdhury, F.U.H.; Abdulwahab, S.; Romani, S.; Radeva, P.; et al. SLSDeep: Skin Lesion Segmentation Based on Dilated Residual and Pyramid Pooling Networks 2018.
14. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network 2017.
15. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848, doi:10.1109/TPAMI.2017.2699184.
16. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation 2018.
17. De Brabandere, B.; Jia, X.; Tuytelaars, T.; Van Gool, L. Dynamic Filter Networks 2016.
18. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks 2017.
19. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2015; Vol. 9351, pp. 234–241 ISBN 978-3-319-24573-7.

20. Aksoy, S. Skin Lesion Segmentation Using UNet, Https://Medium.Com/@serurays/Skin-Lesion-Segmentation-Using-Unet-Aa2479f5884c 2025.
21. Aksoy, S. From Pixels to Diagnosis: How AI Is Transforming Skin Cancer Detection, Https://Medium.Com/@serurays/from-Pixels-to-Diagnosis-How-Ai-Is-Transforming-Skin-Cancer-Detection-91b30da3d7bb 2025.