Mangalore University

Project Report On

Skin Lesion Segmentation

Carried out and submitted
To

DEPARTMENT OF POST-GRADUATE STUDIES AND RESEARCH IN COMPUTER SCIENCE

In Partial Fulfilment for the Award of Degree in Master of Computer Science during the Academic Year 2024-2025

Submitted By:
Bhawya Devi
P05AZ23S038006
MSC Computer Science

Under the Guidance of:
DR. B. H. SHEKAR
Professor
Department of Computer Science
Mangalore University

Department of Post-Graduate Studies and Research in Computer Science
Mangalore University
Mangalagangothri, Mangalore – 574 199

Mangalore University

DEPARTMENT OF POST-GRADUATE STUDIES AND RESEARCH IN COMPUTER SCIENCE

Mangalore University
Mangalagangothri, Mangaluru – 574 199

CERTIFICATE

This is to certify that the project work entitled **“Skin Lesion Segmentation”** has been successfully carried out in the Department of Post- Graduate Studies and Research in Computer Science by **Ms. Bhawya devi (Reg. No: P05AZ23S038006),** student of third semester Master of Computer Science, under the supervision and guidance of **DR. B.H SHEKAR,** Professor, Department of Post-Graduate Studies and Research in Computer Science, Mangalore University. The project is partial fulfilment of the requirements for the award of **Master of Computer Science** by **Mangalore University** during the academic year **2024-2025.**

Internal Guide

Submitted for semester Mini Project and Domain Knowledge Seminar viva-voce examination held on February 2025

Internal Examiner **External Examiner**

INTRODUCTION

The detection and segmentation of skin lesions play a crucial role in early diagnosis and treatment of skin cancer. Automated lesion segmentation assists dermatologists by providing accurate boundary identification, which is essential for effective medical analysis. In this context, the integration of advanced deep learning techniques, particularly the U-Net architecture, presents a promising solution for precise and efficient skin lesion segmentation.

This project introduces a novel approach for real-time skin lesion segmentation using the U-Net model. The U-Net framework, renowned for its high accuracy in medical image segmentation, offers a compelling solution for automatically identifying and segmenting skin lesions from dermoscopic images.

The motivation behind this project stems from the growing need for reliable and automated systems to assist in early skin cancer detection. Traditional methods for lesion segmentation often struggle with accuracy and consistency due to variations in lesion shapes, sizes, and skin tones. By leveraging deep learning and U-Net architecture, this project aims to address these challenges and enhance the effectiveness of automated dermatological analysis.

The proposed approach involves training a U-Net model using the PH2 dataset, which contains a diverse range of dermoscopic images with expert-annotated ground truth masks. The dataset ensures the model's ability to generalize across different lesion types, lighting conditions, and skin tones, improving its segmentation accuracy.

Through extensive training and fine-tuning, the U-Net model learns to accurately differentiate between lesions and healthy skin. The integration of preprocessing techniques such as image resizing, normalization, and data augmentation (flipping, rotation, etc.) further enhances the model's robustness and generalization capabilities.

During the inference phase, the trained U-Net model is deployed to process dermoscopic images, providing precise lesion segmentation with high accuracy and efficiency. The system's performance is evaluated using standard segmentation metrics such as IoU (Intersection over Union), Dice Score, Precision, and Recall, demonstrating its effectiveness across different lesion types and image conditions.

In summary, this project represents a significant step towards the development of intelligent, AI-driven medical imaging solutions. By harnessing the power of deep learning and U-Net architecture, the proposed approach offers a scalable and efficient solution for automated skin lesion segmentation, with wide-ranging applications in dermatology, healthcare, and medical research.

ADVANTAGES:

1. **High Accuracy in Medical Image Segmentation**

   The U-Net model is specifically designed for biomedical image segmentation and provides high accuracy in distinguishing skin lesions from healthy skin. Its encoder-decoder architecture and skip connections help retain fine details, ensuring precise segmentation.

2. **Automation and Faster Diagnosis**

   Traditional manual segmentation by dermatologists is time-consuming and subject to human errors. Using U-Net automates the process, enabling faster and more efficient diagnosis, which is crucial for early detection of melanoma and other skin diseases.

3. **Effective Handling of Limited Data**

   Unlike many deep learning models that require vast amounts of labeled data, U-Net is known for performing well with **s**mall medical datasets by leveraging data augmentation techniques. This makes it highly suitable for medical imaging applications, where labeled datasets are often limited.

4. **Robust Against Variations in Lesions**

   Skin lesions vary in size, shape, color, and texture, making segmentation a challenging task. U-Net's convolutional layers and skip connections help the model generalize better, allowing it to segment lesions effectively across different skin tones, lighting conditions, and lesion types.

5. **Real-Time Application Feasibility**

   Once trained, the U-Net model can quickly process new images, making it suitable for real-time applications in dermatology clinics and AI-assisted diagnostic tools. The model's efficiency allows for quick integration into telemedicine platforms and mobile healthcare applications.

DISADVANTAGES:

1. **Computational Complexity and High Resource Requirement**

   Training U-Net requires high computational power, making it challenging for resource-limited environments. Its large number of parameters increases memory usage and processing time, affecting real-time deployment.

2. **Dependency on High-Quality Data**

   The performance of U-Net depends significantly on high-quality annotated datasets. If the training dataset contains noise, poor annotations, or imbalanced lesion types, the model may struggle to generalize, leading to incorrect segmentation results in real-world applications.

3. **Sensitivity to Image Artifacts and Variability**

   U-Net can sometimes be over-sensitive to artifacts such as hair, shadows, reflections, and image noise in dermoscopic images. If not properly preprocessed, these artifacts may cause incorrect segmentations, reducing the model's reliability.

4. **Potential Overfitting**

   Due to the small dataset sizes in medical imaging, deep learning models like U-Net are prone to overfitting, where they perform well on training data but fail on unseen data. Regularization

LITERATURE SERVEY

Duan Wang, "Skin Lesion Segmentation of Dermoscopy Images Using U-Net", Beijing University of Posts and Telecommunications, June 2023.

This paper focuses on using the U-Net model for accurate skin lesion segmentation. It highlights the drawbacks of traditional methods and demonstrates how U-Net's encoder-decoder architecture with skip connections improves segmentation accuracy. The study evaluates the model on public datasets, showing superior IoU and Dice coefficient scores. The paper concludes that U-Net is highly effective for medical image segmentation, with potential improvements through attention mechanisms, data augmentation, and computational optimizations.

Olaf Ronneberger, Philipp Fischer,Thomas Brox ,"U-Net: Convolutional Networks for Biomedical Image Segmentation"

This paper introduces the U-Net architecture, a fully convolutional neural network (CNN) designed specifically for biomedical image segmentation. U-Net consists of a contracting (encoder) path for feature extraction and an expanding (decoder) path for precise localization, with skip connections that help retain spatial information. The model is trained on a small dataset using data augmentation and achieves high accuracy with fewer training samples.The study demonstrates U-Net's effectiveness in medical image segmentation tasks, particularly in delineating cell structures and lesions, outperforming traditional segmentation methods. The paper highlights U-Net's efficiency, robustness, and ability to work with limited annotated data, making it widely used in medical imaging applications.

Lina Liu, Lichao Mou, Xiao Xiang Zhu, Mrinal Manda, "Skin Lesion Segmentation Based on Improved U-net" IEEE Access, 11 October 2019.

This paper presents an enhanced U-Net model for more accurate and efficient skin lesion segmentation. The study focuses on addressing the limitations of the traditional U-Net, such as difficulty in capturing fine lesion boundaries and handling variations in lesion size and shape.The improved model incorporates advanced feature extraction techniques, attention mechanisms, and optimized loss functions to enhance segmentation accuracy. The paper evaluates the proposed method on publicly available skin lesion datasets, showing improved Dice coefficient and IoU scores compared to the standard U-Net.The findings suggest that the improved U-Net model is more effective for automated skin lesion detection, making it highly valuable for early diagnosis and medical image analysis.

Prashant Brahmbhatt, Siddhi Nath Rajan, "Skin Lesion Segmentation using SegNet with Binary CrossEntropy", International Conference on Artificial Intelligence and Speech Technology (AIST2019) 14-15th November, 2019.

This paper explores the use of SegNet, a deep learning model, for automated skin lesion segmentation. The study employs Binary CrossEntropy loss to optimize the model's performance in distinguishing lesion regions from healthy skin.

The research highlights the advantages of SegNet's encoder-decoder architecture, which preserves spatial information and enhances segmentation accuracy. The model is evaluated on publicly available skin lesion datasets, demonstrating promising results in terms of Dice coefficient and IoU scores.The findings suggest that SegNet, combined with Binary CrossEntropy loss, provides an efficient solution for medical image segmentation, aiding in early skin disease detection and diagnosis

Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu, "Transformers in Medical Imaging: A Survey "

This paper provides a comprehensive review of transformer-based models in medical imaging.The study discusses how transformers, originally designed for natural language processing (NLP), have been successfully adapted for medical image analysis, including segmentation, classification, detection, and reconstruction. It compares transformer architectures with traditional CNN-based methods, highlighting their advantages, such as long-range dependency modeling and improved feature representation.

Additionally, the paper reviews various medical imaging applications, challenges in using transformers (e.g., computational cost and data efficiency), and future research directions. The survey emphasizes the growing role of transformers in advancing medical AI, offering insights into their potential impact on healthcare and diagnostic automation.

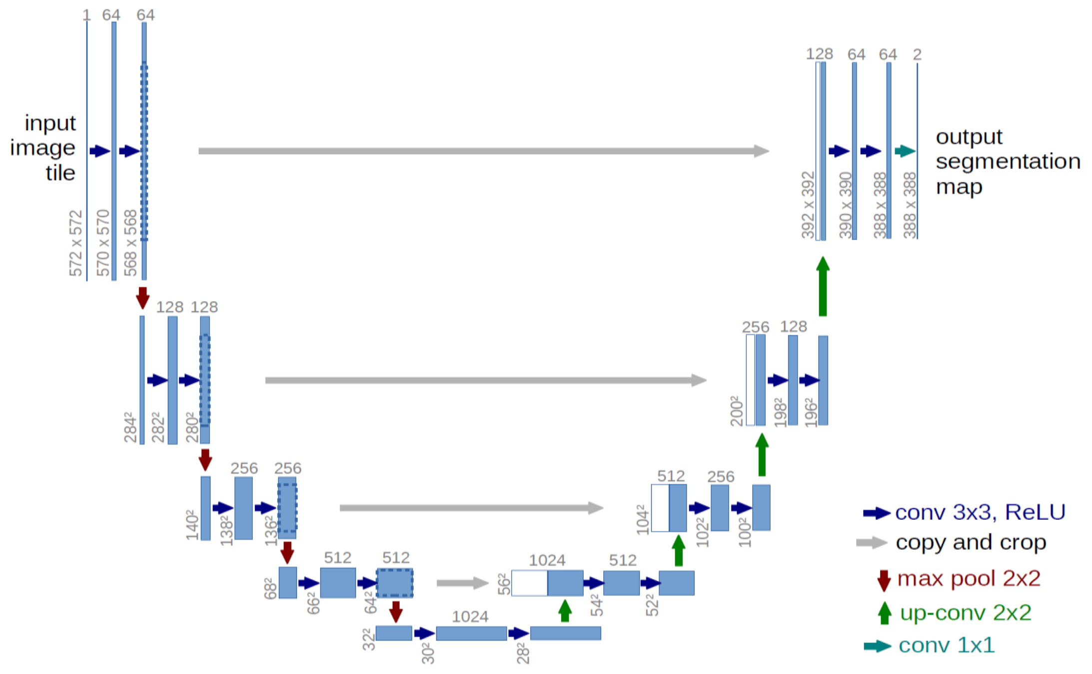
U-Net Architecture

The U-Net is a convolutional neural network initially developed for biomedical image segmentation at the University of Freiburg, Germany. It is an extension of the fully convolutional network proposed by Long, Shelhamer, and Darrell.

The U-Net architecture enhances the segmentation accuracy by incorporating upsampling operators instead of pooling operations, which increases the output resolution. Additionally, the network employs a symmetric u-shaped structure. This design choice enables the network to capture and propagate context information effectively. By utilizing a large number of feature channels in the upsampling part, the network can extract and utilize rich contextual information, leading to more accurate segmentations.

Unlike traditional networks that rely on fully connected layers, the U-Net does not use them and instead extrapolates missing context in the border region by mirroring the input image. This approach enables the network to handle large images efficiently.

Overview of UNet



The U-Net architecture consists of an encoder-decoder structure with skip connections, enabling it to capture both high-level and low-level features.

1. **Encoder:** The encoder is responsible for capturing high-level features from the input image.
    - It starts with a series of convolutional layers with a ReLU activation function, followed by padding to maintain spatial dimensions.
    - Convolutional layers apply a set of filters to extract features.
    - After each convolutional layer, another convolutional layer with the same number of filters is applied to capture more complex features.
    - MaxPooling is performed to downsample the feature maps and reduce spatial dimensions.
    -
2. **Bridge:** The bridge connects the encoder and decoder through skip connections.
    - It consists of additional convolutional layers with ReLU activation.
    - The bridge helps in preserving spatial information by concatenating the feature maps from the encoder to the corresponding decoder layers.
    -
3. **Decoder:** The decoder generates the final segmentation map using the concatenated feature maps from the bridge.
    - It starts with upsampling to increase the spatial dimensions of the feature maps.

- Convolutional layers with ReLU activation are applied to refine the feature maps.
- The upsampled feature maps are concatenated with the corresponding feature maps from the encoder.
- More convolutional layers are applied to further refine the features.
- The decoder ends with a convolutional layer with a sigmoid activation function to produce the final segmentation map.

The U-Net architecture leverages skip connections to combine both high-level and low-level features, allowing it to capture fine-grained details while maintaining contextual information. This makes it effective for tasks such as image segmentation, where precise delineation of object boundaries is important.

METHODLOGY

SOFTWARE AND HARDWARE REQUIREMENTS:

Software requirement:

1. Deep Learning Framework
2. UNet Implementation
3. Data Labelling Tools
4. Training Environment
5. Development Environment

Hardware Requirement:

1. RAM: 8GB
2. CPU: FASTER
3. GPU:(OPTIONAL)
4. STORAGE:500GB

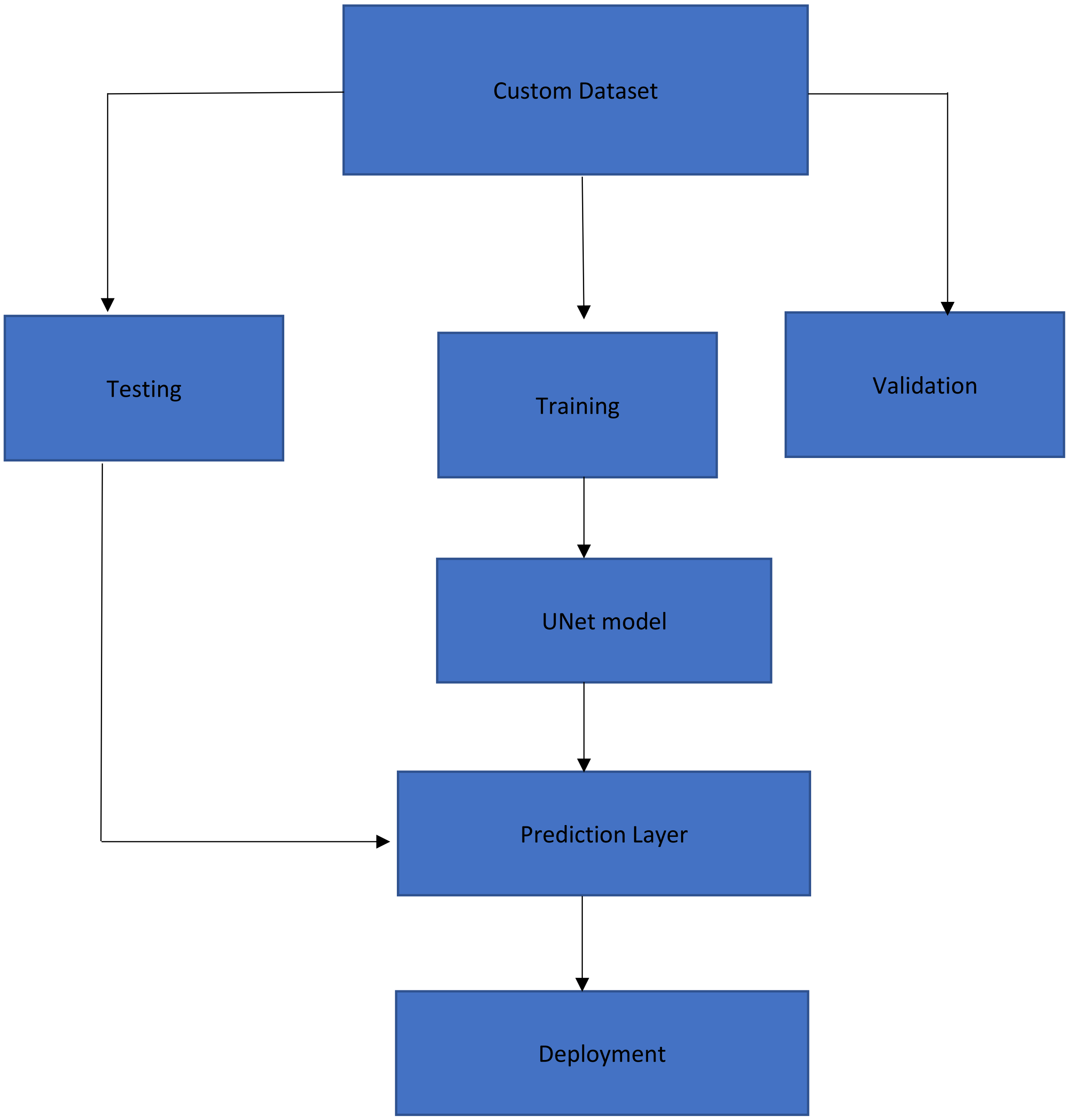
Steps to Develop the Model:

1. **Data Preparation**: Preprocess the PH2 dataset, including resizing images, normalizing pixel values, and augmenting data to improve model generalization. Split the dataset into training, validation, and test sets.
2. **Model Training:** Train the U-Net model using the preprocessed dataset. The training process involves passing images through the encoder-decoder architecture, computing the loss function (e.g., Dice loss, binary cross-entropy), and updating model parameters using an optimization algorithm like Adam.
3. **Model Evaluation**: Evaluate the trained U-Net model on the validation set using metrics such as IoU (Intersection over Union), Dice coefficient, precision, recall, and accuracy. Fine-tune the model based on performance results.
4. **Deployment**: Once the model achieves satisfactory segmentation accuracy, deploy it for real-world applications such as medical diagnosis assistance or automated lesion analysis in healthcare systems.

5. **Testing and Monitoring:** Test the deployed model with unseen dermoscopic images and continuously monitor its performance. Periodically fine-tune the model to enhance its accuracy and robustness.

System Design:

This section describes the methodology proposed in this study. This study starts with inputting the real-time images, followed by the implementation of Convolutional Neural Network (CNN) for recognizing the emotion. Succeeding,the recognized emotion will be displayed. Figure 1 shows the proposed flowchartin this study. Further explanation will be elaborate in the sub section accordingly.



Tools and Platforms used:

1. **Google Colab:** Google Colaboratory, or Colab, is an online Jupyter Notebook environment that allows users to write and execute Python code through a browser. It provides free access to GPUs and TPUs, making it ideal for training deep learning models efficiently.

2. **U-Net:** U-Net is a deep learning model designed for biomedical image segmentation. It follows an encoder-decoder architecture with skip connections, enabling precise localization while capturing high-level features. It is widely used for medical image analysis, including skin lesion segmentation.

3. **TensorFlow/Keras:** TensorFlow is an open-source deep learning framework, and Keras is its high-level API that simplifies building and training neural networks. It is used for implementing and optimizing the U-Net model for skin lesion segmentation.

4. **OpenCV:** OpenCV is an open-source computer vision library used for image preprocessing tasks such as resizing, denoising, and contrast enhancement. It helps improve input image quality for better segmentation performance.

5. **NumPy:** NumPy is a Python library for numerical computing and array manipulations. It plays a crucial role in handling and processing medical image datasets efficiently.

6. **Pandas:** Pandas is a data manipulation and analysis library in Python. It is used for loading, organizing, and analyzing metadata associated with skin lesion images, such as patient details, lesion types, and annotations.

7. **Matplotlib:** Matplotlib is a Python visualization library used for plotting images, loss curves, accuracy graphs, and segmentation outputs. It helps in analyzing model performance and visualizing segmented lesions effectively.

DATASET:

For this project, we will be training the U-Net model on the PH2 dataset, which is widely used for skin lesion segmentation in medical imaging.

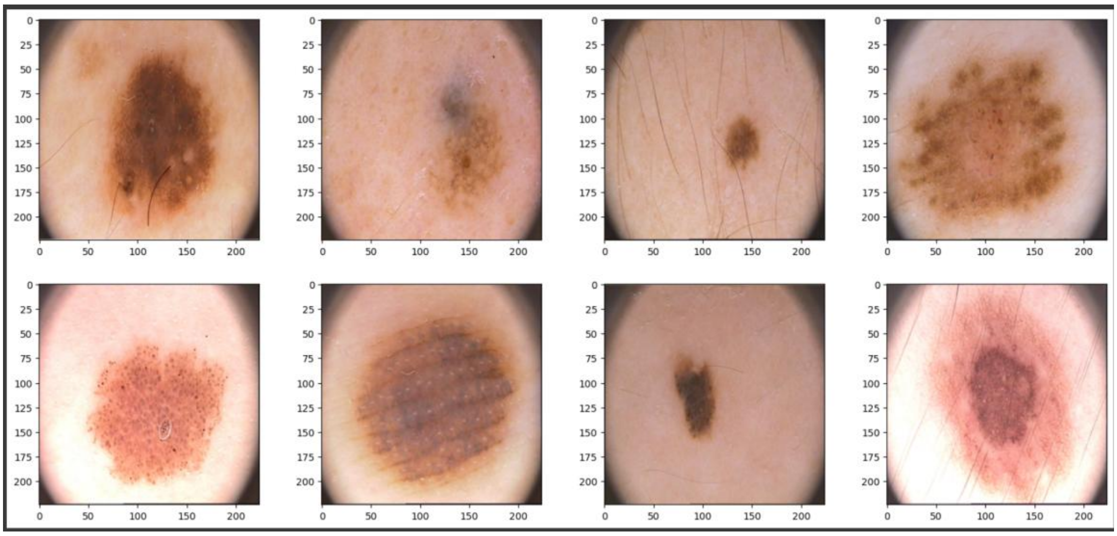
The dataset is publicly available and contains dermoscopic images of skin lesions with corresponding ground-truth masks. The images are preprocessed to ensure consistency in resolution and quality before training.

To facilitate dataset handling, we preprocess the images and masks, ensuring proper normalization, resizing, and augmentation techniques to improve model generalization. The dataset is split into training, validation, and testing sets to evaluate the model's performance effectively.

The dataset used for this project consists of:

- **Training Images:** 360
- **Test Images:** 155
- **Validation Images:** 67
- **Annotation Format:** Binary masks for lesion segmentation

Sample Images :



EXPERIMENTS AND RESULT:

Importing libraries :

```
from keras.models import Model, Sequential
from keras.layers import Activation, Dense, BatchNormalization, concatenate, Dropout, Conv2D, Conv2DTranspose, MaxPooling2D, UpSampling2D, Input, Reshape, SpatialDropout2D
from keras.callbacks import EarlyStopping
from tensorflow.keras import backend as K
from keras.optimizers import Adam
import tensorflow as tf
import numpy as np
import pandas as pd
import glob
import PIL
from PIL import Image
import matplotlib.pyplot as plt
import cv2
%matplotlib inline

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from warnings import filterwarnings

filterwarnings('ignore')
np.random.seed(101)
```

Loading Dataset :

```
filelist_trainx = sorted(glob.glob('/content/drive/MyDrive/Project/skin/trainx/*.bmp'), key=numericalSort)

# **Resizing Images to (128,128)**
IMG_WIDTH = 224
IMG_HEIGHT = 224
X_train = np.array([np.array(Image.open(fname).resize((IMG_WIDTH, IMG_HEIGHT))) for fname in filelist_trainx])

filelist_trainy = sorted(glob.glob('/content/drive/MyDrive/Project/skin/trainy/*.bmp'), key=numericalSort)
Y_train = np.array([np.array(Image.open(fname).resize((IMG_WIDTH, IMG_HEIGHT))) for fname in filelist_trainy]) # Resizing trainy images as well
```

Splitting the Dataset into Train and Test set :

```
x_train, x_test, y_train, y_test = train_test_split(X_train, Y_train, test_size = 0.25, random_state = 101)
```

Model training

```
dropout_val=0.50
if K.image_data_format() == 'channels_first':
    inputs = Input((INPUT_CHANNELS, 224, 224))
    axis = 1
else:
    inputs = Input((224, 224, INPUT_CHANNELS))
    axis = 3
filters = 32

conv_224 = double_conv_layer(inputs, filters)
pool_112 = MaxPooling2D(pool_size=(2, 2))(conv_224)

conv_112 = double_conv_layer(pool_112, 2*filters)
pool_56 = MaxPooling2D(pool_size=(2, 2))(conv_112)

conv_56 = double_conv_layer(pool_56, 4*filters)
pool_28 = MaxPooling2D(pool_size=(2, 2))(conv_56)

conv_28 = double_conv_layer(pool_28, 8*filters)
pool_14 = MaxPooling2D(pool_size=(2, 2))(conv_28)

conv_14 = double_conv_layer(pool_14, 16*filters)
pool_7 = MaxPooling2D(pool_size=(2, 2))(conv_14)

conv_7 = double_conv_layer(pool_7, 32*filters)
```

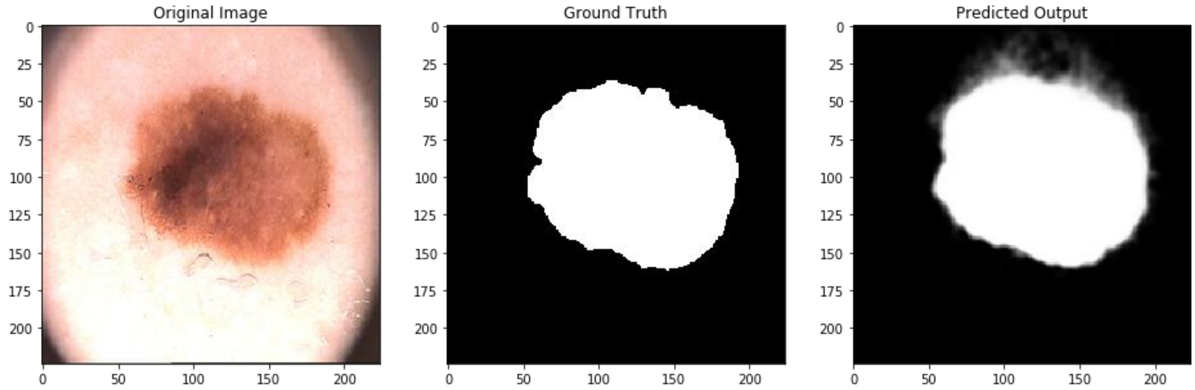
```
up_14 = concatenate([UpSampling2D(size=(2, 2))(conv_7), conv_14], axis=axis)
up_conv_14 = double_conv_layer(up_14, 16*filters)

up_28 = concatenate([UpSampling2D(size=(2, 2))(up_conv_14), conv_28], axis=axis)
up_conv_28 = double_conv_layer(up_28, 8*filters)

up_56 = concatenate([UpSampling2D(size=(2, 2))(up_conv_28), conv_56], axis=axis)
up_conv_56 = double_conv_layer(up_56, 4*filters)

up_112 = concatenate([UpSampling2D(size=(2, 2))(up_conv_56), conv_112], axis=axis)
up_conv_112 = double_conv_layer(up_112, 2*filters)

up_224 = concatenate([UpSampling2D(size=(2, 2))(up_conv_112), conv_224], axis=axis)
up_conv_224 = double_conv_layer(up_224, filters, dropout_val)

conv_final = Conv2D(OUTPUT_MASK_CHANNELS, (1, 1))(up_conv_224)
conv_final = Activation('sigmoid')(conv_final)
pred = Reshape((224,224))(conv_final)
```

```
model_0 = Model(inputs, pred, name="UNET_224")
model_0.compile(optimizer= Adam(learning_rate = 0.003), loss= [jaccard_distance]
                 , metrics=[iou, dice_coe, precision, recall, accuracy])
```

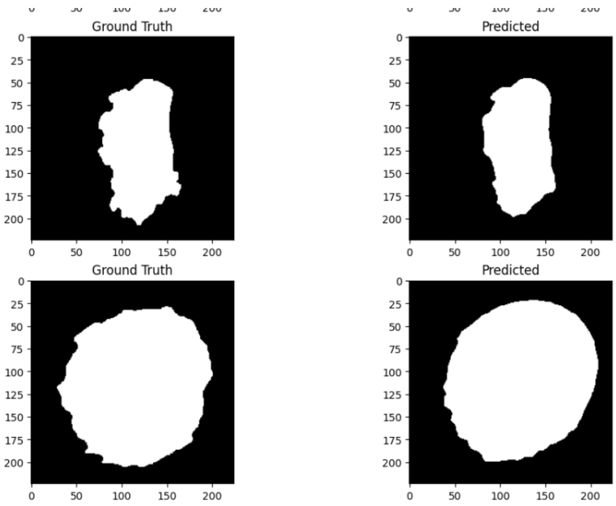
```
model_0.load_weights('unet_1_epoch.h5')
```

TESTING

Detection on unseen images:



After Enhancement :

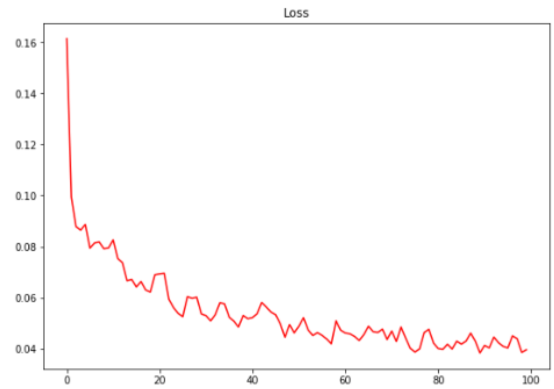


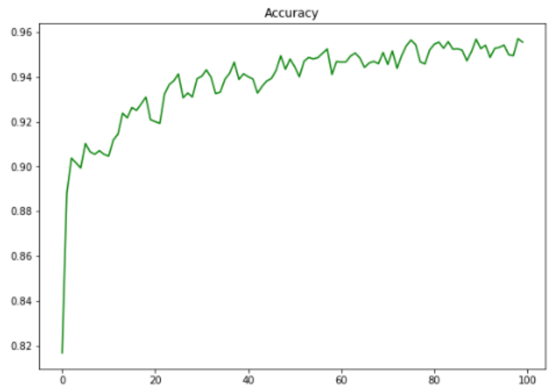
RESULT

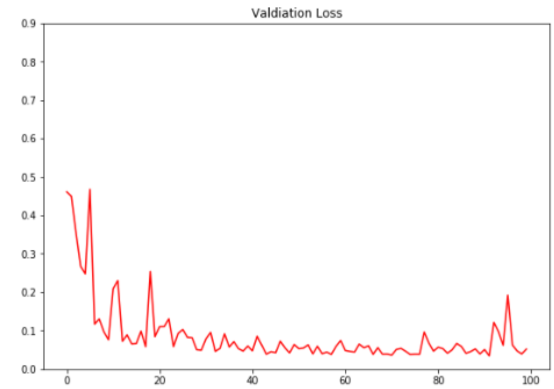
UNet model:

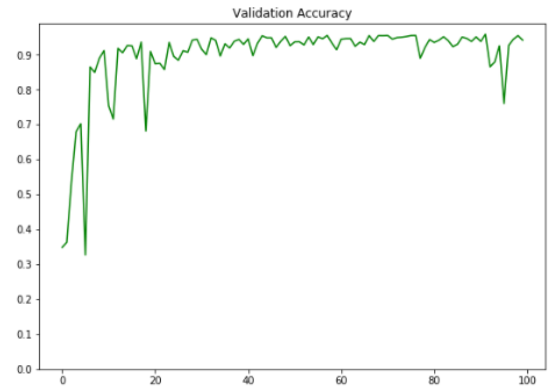
EVALUATION MATRIX:

```
-------------On Train Set-------------------------

360/360 [==============================] - 3s 9ms/step
________________________
IOU:         |    94.92  |
Dice Coef:   |    89.58  |
Precision:   |    87.83  |
Recall:      |    96.30  |
Accuracy:    |    94.55  |
Loss:        |    5.08  |
________________________

-------------On Test  Set-------------------------

50/50 [==============================] - 0s 5ms/step
________________________
IOU:         |    93.85  |
Dice Coef:   |    87.95  |
Precision:   |    84.86  |
Recall:      |    95.95  |
Accuracy:    |    93.57  |
Loss:        |    6.15  |
________________________

-------------On validation Set--------------------

90/90 [==============================] - 0s 5ms/step
________________________
IOU:         |    94.81  |
Dice Coef:   |    89.19  |
Precision:   |    86.24  |
Recall:      |    96.81  |
Accuracy:    |    94.18  |
Loss:        |    5.19  |
________________________
```









CONCLUSION:

The U-Net model used for skin lesion segmentation achieved a moderate to good performance in segmenting dermoscopic images, indicating its effectiveness in medical image analysis. The model successfully distinguishes between lesion and non-lesion areas, providing precise segmentation to aid in skin disease diagnosis.

However, challenges exist in accurately segmenting complex and irregular lesions, as performance metrics may indicate scope for improvement in handling variations in shape, texture, and illumination. Enhancing the model with data augmentation, fine-tuning hyperparameters, or integrating additional post-processing techniques could further boost accuracy and generalization.

Additionally, comparisons with other segmentation models like FCN and SegNet suggest that U-Net excels in capturing fine-grained details due to its skip connections, making it an ideal choice for medical image segmentation.

In conclusion, while U-Net provides strong segmentation capabilities, further optimization in data processing, model architecture, or training strategies can lead to even better lesion detection and improved clinical applicability.

FUTURE SCOPE:

1. Early Skin Cancer Detection:

- Improved segmentation models can assist dermatologists in early-stage melanoma detection, leading to better patient outcomes.
- AI-powered skin lesion segmentation can reduce manual diagnostic workload in hospitals and clinics.
-

2. Personalized Treatment Plans:

- AI-based segmentation can help track disease progression, assisting doctors in tailoring treatments based on lesion characteristics.
- Automated lesion growth monitoring can help predict cancerous transformations in skin abnormalities.

3. Real-Time Mobile Applications:

- Development of mobile apps with AI-driven skin lesion detection can empower users to self-check skin conditions.
- Wearable devices with imaging capabilities can continuously monitor skin health and provide early alerts.

4. Integration with Robotics and Surgery:

- AI-based skin lesion segmentation can assist in robotic-assisted skin surgeries by providing precise lesion boundaries.
- Autonomous robotic systems can use segmentation outputs to improve laser-based treatments and excision surgeries.

5. Multi-Class Segmentation:

- Extending the model to different types of skin lesions (benign, malignant, melanoma, etc.) can improve diagnostic efficiency.

6. Integration with Clinical Data:

- Combining segmentation results with patient history, dermoscopic metadata, and genetic factors can enhance AI-driven diagnostic systems.

REFERENCE:

[1] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu, "*Transformers in Medical Imaging: A Survey* "

[2] Prashant Brahmbhatt, Siddhi Nath Rajan, "*Skin Lesion Segmentation using SegNet with Binary CrossEntropy*", International Conference on Artificial Intelligence and Speech Technology (AIST2019) 14-15th November, 2019.

ONLINE REFERENCES:

PH2 Dataset: https://www.kaggle.com/datasets/athina123/ph2dataset

https://github.com/hashbanger/Skin_Lesion_Segmentation/blob/master/skin-lesion-segmentation-using-unet.ipynb