**Skin Lesion Segmentation from Dermoscopic Images**

by

Anand Venkateshwaran

Ashley Almeida

Manikandan Gopalakrishnan

Shivanand Kadadi

Vinuth Tulasi

Report

Submitted in partial fulfillment of the requirements for the

DLFA Program

Centre for Continuing Education

Indian Institute of Science

Bangalore – 560 012 India

# Abstract

Skin cancer is the most common cancer among all cancers. Current practice, specialists rely on visual inspection with manual Dermoscopic assessment to determine the onset of malignancy. Since early and accurate diagnosis is crucial, easier cheaper and frequent access to diagnostic tools becomes necessary. Motivation here is to solve the skin lesion segmentation, which is an important process in skin diagnostics, using deep neural networks which have been proven to be reliable. Our proposed solution can be extended to diagnose skin cancer and final complete solution can be integrated with a smart phone application or with desktop application making the diagnosis tools more accessible. In this project, we have made use of the ISIC challenge 2018 dataset to build our models.

We have experimented with over 6 models and built an ensemble of the best performing models. We have found our performance scores to be at par and also over-performed the benchmark scores of the ISIC 2018 challenge. MFSNet is the new emerging model and showcases a lot of promise.

**Acknowledgments**



We would like to extend a hearty thanks to our Mentor Ahmed who has diligently spent a lot of time and guided us through the capstone project



And, also to Prof. Dr. Venkatesh Babu for his insights and his direction that has helped us successfully research and build these models

**Table of Contents**

[Abstract 2](#_Toc127174758)

[Chapter I: Introduction 5](#_Toc127174759)

[Problem Statement 5](#_Toc127174760)

[Purpose of the Study 5](#_Toc127174761)

[Research Questions 6](#_Toc127174762)

[Assumptions and Limitations of the Study 6](#_Toc127174763)

[Overview 6](#_Toc127174764)

[Chapter II: Method/Experiment 7](#_Toc127174765)

[Introduction 7](#_Toc127174766)

[Dataset 7](#_Toc127174767)

[Data Preprocessing, Feature Engineering and Visualization 8](#_Toc127174768)

[Choice of Model 10](#_Toc127174769)

[Performance of the Model and Metrics 18](#_Toc127174770)

[Overall project and improvements and applications and results 26](#_Toc127174771)

[Chapter III: Results 27](#_Toc127174772)

[Introduction 27](#_Toc127174773)

[Summary 27](#_Toc127174774)

[Conclusion 29](#_Toc127174775)

[References 30](#_Toc127174776)

# 

# Chapter I: Introduction

## Problem Statement

Skin cancer is the most common cancer among all cancers. The malignant skin lesions consist of the melanocytic lesion, i.e., melanoma. Although melanoma is the least common type of skin cancer, it is the most aggressive and deadly cancer. Therefore, timely diagnosis is critical for its treatment before the onset of malignancy. In current medical practice, skin cancer specialists primarily examine the patients on visual inspection with manual measurement tools called Dermoscopic assessment to determine the skin lesions. Relying on self-vigilance and medical inspection by human vision risk life and survival rate as it is difficult to identify the type of lesions with the naked eye.

## Purpose of the Study

Since early diagnosis of melanoma is crucial, easier cheaper and frequent access to diagnostic tools becomes necessary. Motivation here is to solve the skin lesion segmentation, which is an important process in skin diagnostics, using deep neural networks which have been proven to be reliable. Our proposed solution can be extended to diagnose skin cancer and final complete solution can be integrated with a smart phone application or with desktop application making the diagnosis tools more accessible. The diagnostic tool can be used by the end user for regular self-diagnosis, or by semi trained/rural technicians to identify and trigger the further diagnosis.

## ****Research Questions****

Which AI/ML models currently exist which can help us in the field of medical images? Which are these models perform best given the current dataset? Can we identify a model which uses less compute and can be used in the final solution?

## Assumptions and Limitations of the Study

The ISIC 2018 [1][2] dataset provided images with a resolution size on 512x512x1. However, due to resource constraints we were unable to execute and train our models with this resolution. To resolve this, we downsized the images to 128x128x1.

|  |  |  |
| --- | --- | --- |
| Training Data | Ground Truth | Test Data |
| 2594 images | 12970 response masks (5 for each image). | 1000 images. |

## Overview

In the subsequent sections of the project, we will have a look at the Exploratory data analysis we conducted, the choices of models we had. We will provide a very brief explanation of the model and its uniqueness compared to the other models. We will then have a look at the performance of these models, compare them and arrive at our conclusion. We will also look at what additional research and methodologies can be used to improve these models.

# 

# Chapter II: Method/Experiment

## Introduction

In the medical field, digital imaging and pre-processing has been increasingly used. This helps doctors detect various disorders quickly and easily. In the case of skin cancer, it is one of the most prevalent types of disease in both adults and children. Melanoma is a malignant type of skin cancer that develops through the irregular growth of pigmented skin cells called melanocytes. It can develop anywhere on the epidermal layer of the skin and presumably may also affect the chest and back and propagate from the primary site of cancer. In this project, we will make use of Dermoscopic Images and segment the lesion boundaries which can be used in the diagnosis and classification of early cancer.

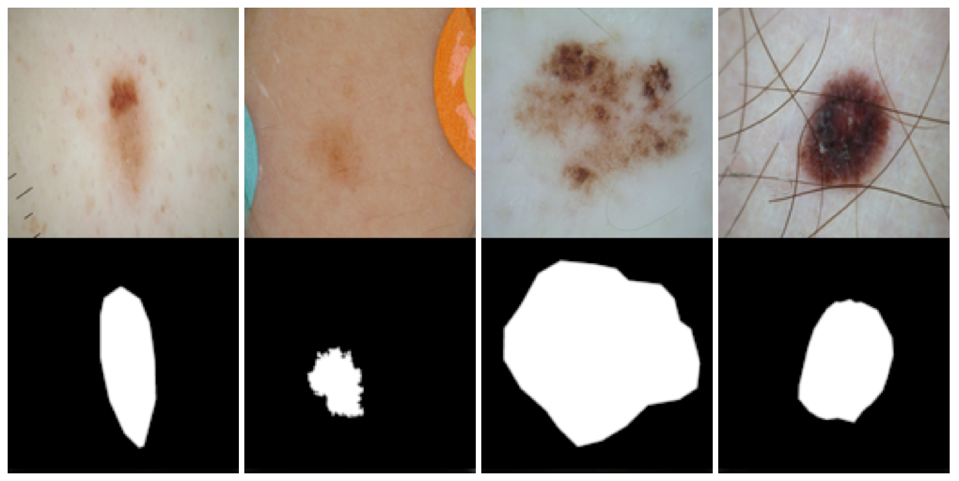
## Dataset

The success of achieving our objective depends on the dataset we use. For the project we have used the ISIC 2018 dataset [1][2]. The input data are Dermoscopic lesion images in JPEG format. All lesion images are named using the scheme ISIC\_.jpg, which is a 7-digit unique identifier. EXIF tags in the images have been removed. Every lesion image contains exactly one primary lesion; other fiducial markers, smaller secondary lesions, or other pigmented regions. The response data are binary mask images in PNG format, indicating the location of the primary skin lesion within each input lesion image. Mask images are named using the scheme ISIC\_\_segmentation.png, which matches the corresponding lesion image for the mask. Mask images are encoded as single-channel (grayscale) 8-bit PNGs.

|  |  |  |
| --- | --- | --- |
| Training Data | Ground Truth | Test Data |
| 2594 images | 12970 response masks (5 for each image). | 1000 images. |

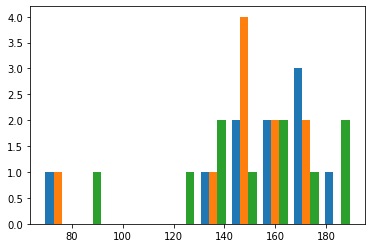
## Data Preprocessing, Feature Engineering and Visualization

We started the data pre-processing by downloading the dataset from the source. We can see some sample images and the ground truth in the image 3.1 below.



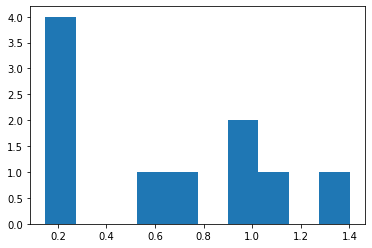
**Figure 2.1**: Sample images and ground truth

As part of the Exploratory Data Analysis (EDA) we ensured that all the images are of the same size. We then checked for any null values and outliers and removed those.



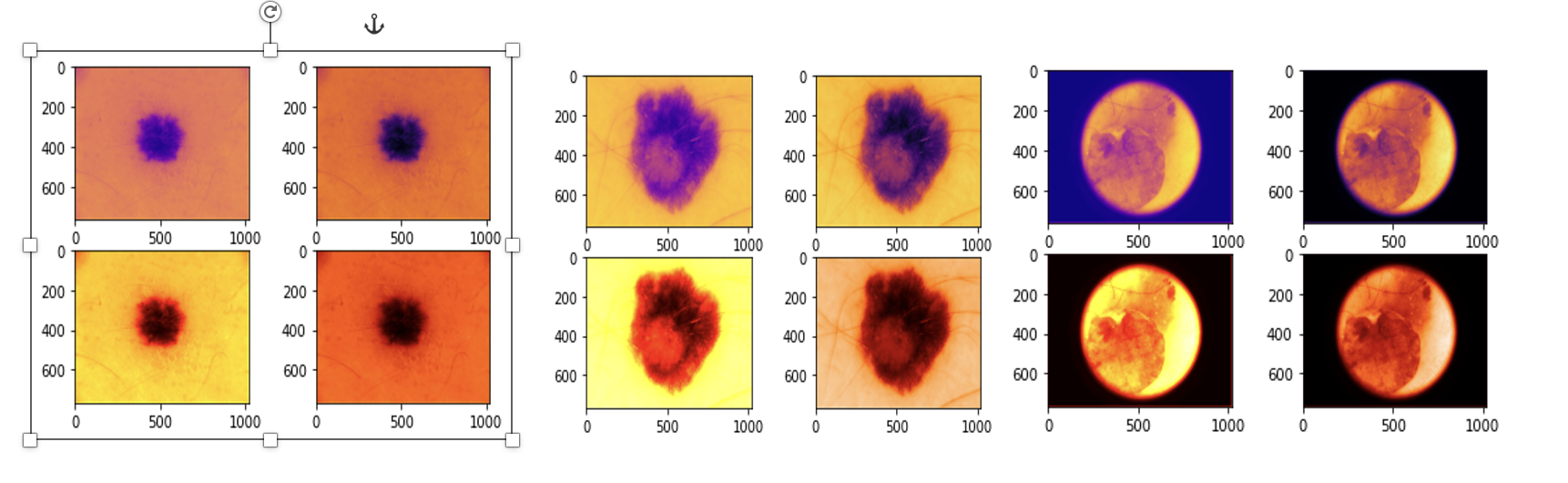
**Figure 2.2:** Color Intensity of Images

In the figure above we have checked for the distribution of the color intensity of the images.



**Figure 2.3** Lesion Area Distribution

In the figure above we have checked the Lesion Area distribution among all images. We can observe that most of the images are generally found around 0.2 mm in size.



**Figure 2.4:** Image Contour Maps

In the above graphs we have attempted to visualize the lesions using different contour maps. We can observe that some contour maps make the lesions more visible than the others.

## Choice of Model

As part of this project, we have attempted to use several models and then compare their performances. Basis the performance we have then chosen to do an ensemble of 2 top performing models.

* UNet
* UNet++
* R2U-Net
* Attention U-Net
* SegNet
* MFSNet
* Ensemble Method (Two best Algorithms)

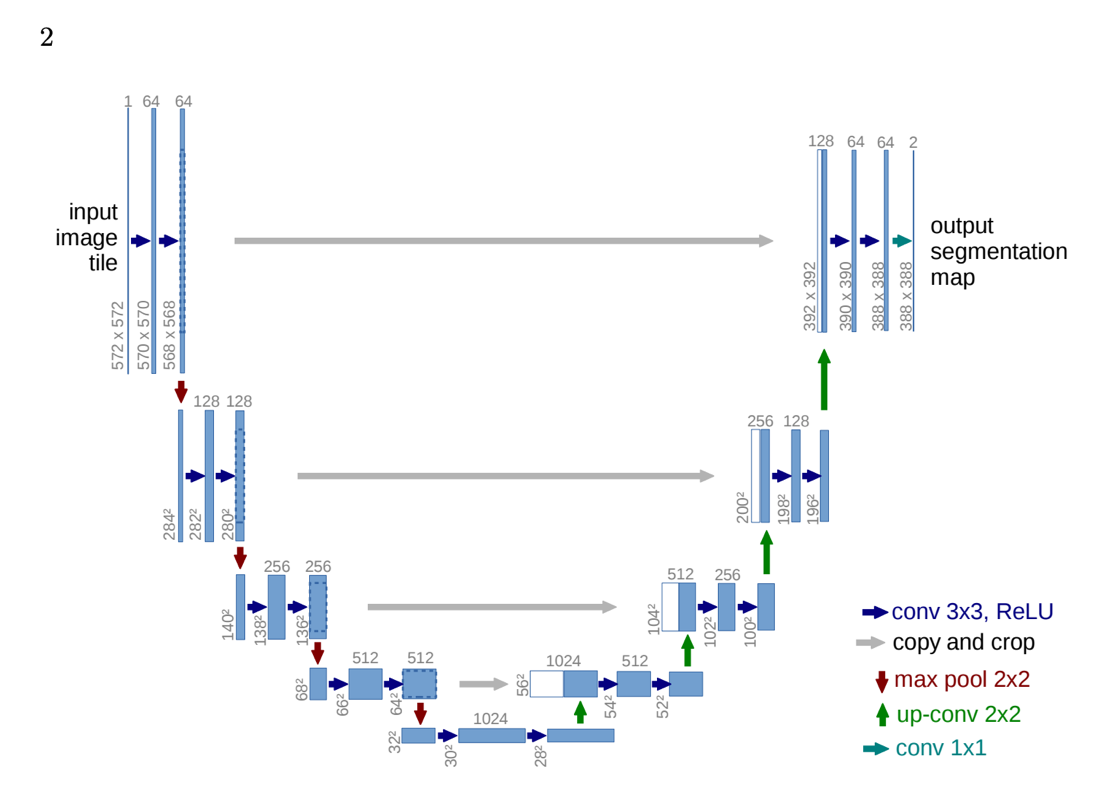
We have experimented with 2 transfer learning models. EfficientNet [10] and VGG16 [11]

**UNet**:

The U-Net architecture stems from the so-called “fully convolutional network” first proposed by Long, Shelhamer, and Darrell.[4]

The main idea is to supplement a usual contracting network by successive layers, where pooling operations are replaced by upsampling operators. Hence these layers increase the resolution of the output. A successive convolutional layer can then learn to assemble a precise output based on this information.[3]

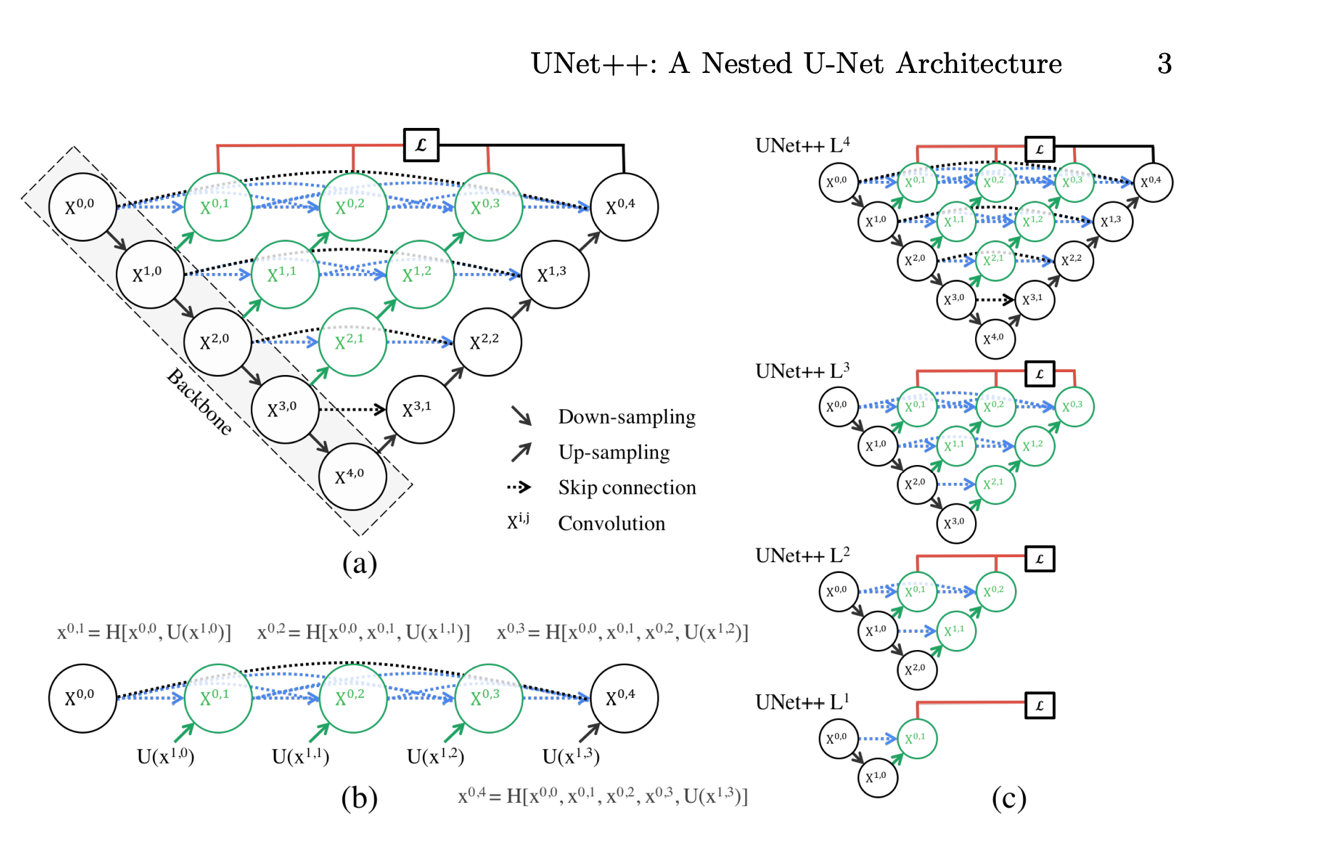
One important modification in U-Net is that there are a large number of feature channels in the upsampling part, which allow the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting part, and yields a u-shaped architecture. The network only uses the valid part of each convolution without any fully connected layers.[2] To predict the pixels in the border region of the image, the missing context is extrapolated by mirroring the input image. This tiling strategy is important to apply the network to large images, since otherwise the resolution would be limited by the GPU memory.



**Figure 2.5** UNet Architecture

**UNet ++:**

UNet++ [5] differs from the original U-Net in three ways: 1) having convolution layers on skip pathways (shown in green), which bridges the semantic gap between encoder and decoder feature maps; 2) having dense skip connections on skip pathways (shown in blue), which improves gradient flow; and 3) having deep supervision (shown in red), which enables model pruning and improves or in the worst case achieves comparable performance to using only one loss layer.

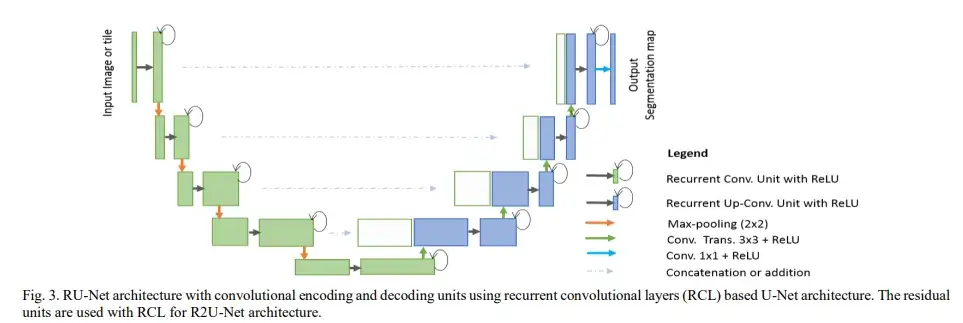


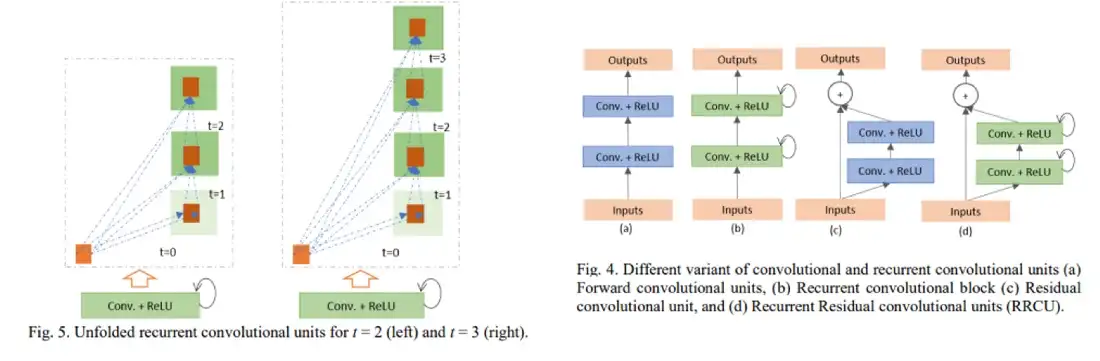
**Figure 2.6** UNet ++ Architecture

**R2UNet**:

R2UNet too tries to overcome the challenges in UNet. The Three main components of the network are

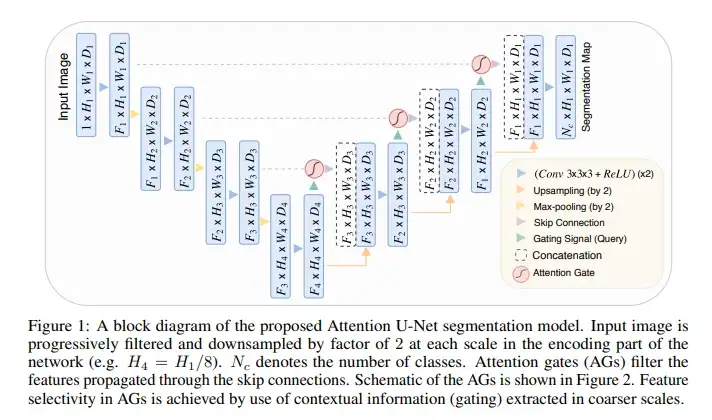
1. Skip connections: Inspired from Unet++, re-designed skip pathways modify the connection between encoder and decoder.
2. Backbone: Usage of recurrent residual convolutions layers (RRCL) over the simple convolutional layers
3. Deep Supervision: The added dense skip connections enable the network to merge the architectures of various depths into a single architecture

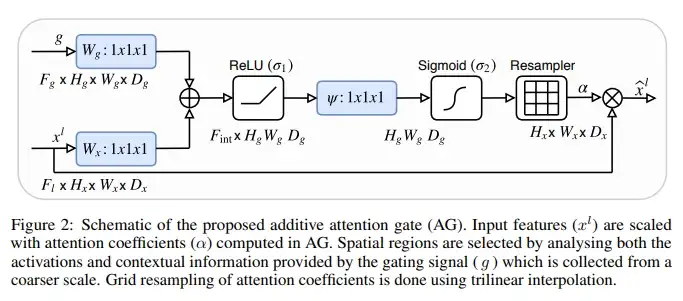


** Figure 2.7** R2UNet Architecture

**Attention** **UNet**:

This Attention UNet [7] network borrowed the idea of an attention mechanism from NLP and used it in skip connections. It gave the skip connections an extra idea of which region to focus on while segmenting the given object. This works great even with very small objects due to the attention present in the skip connections.

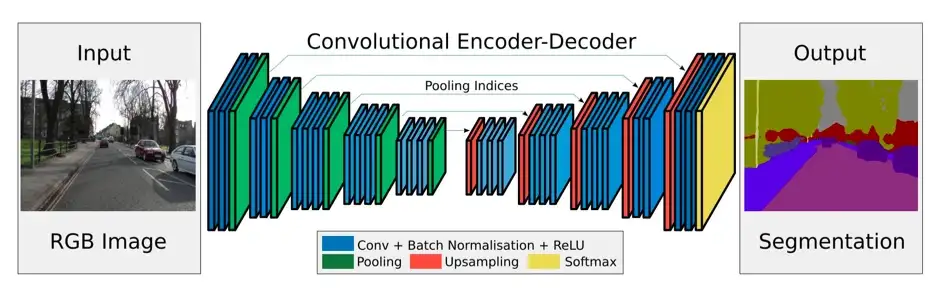
****

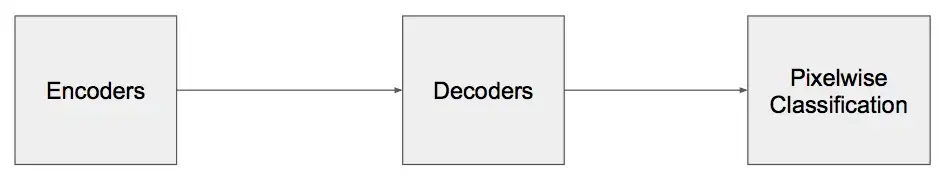
****

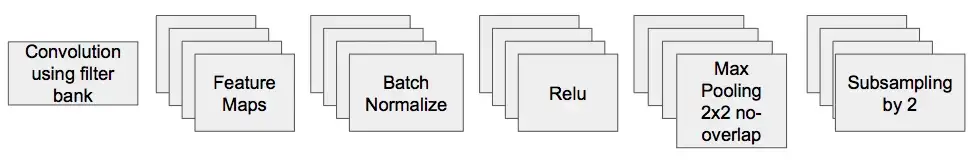
**Figure 2.8** Attention UNet Architecture

**SEGNet**:

SEGNet [8]:This core trainable segmentation engine consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. The architecture of the encoder network is topologically identical to the 13 convolutional layers in the VGG16 network [9] . The role of the decoder network is to map the low-resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. The novelty of SegNet lies is in the manner in which the decoder upsamples its lower resolution input feature map(s). Specifically, the decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to upsample. The upsampled maps are sparse and are then convolved with trainable filters to produce dense feature maps.



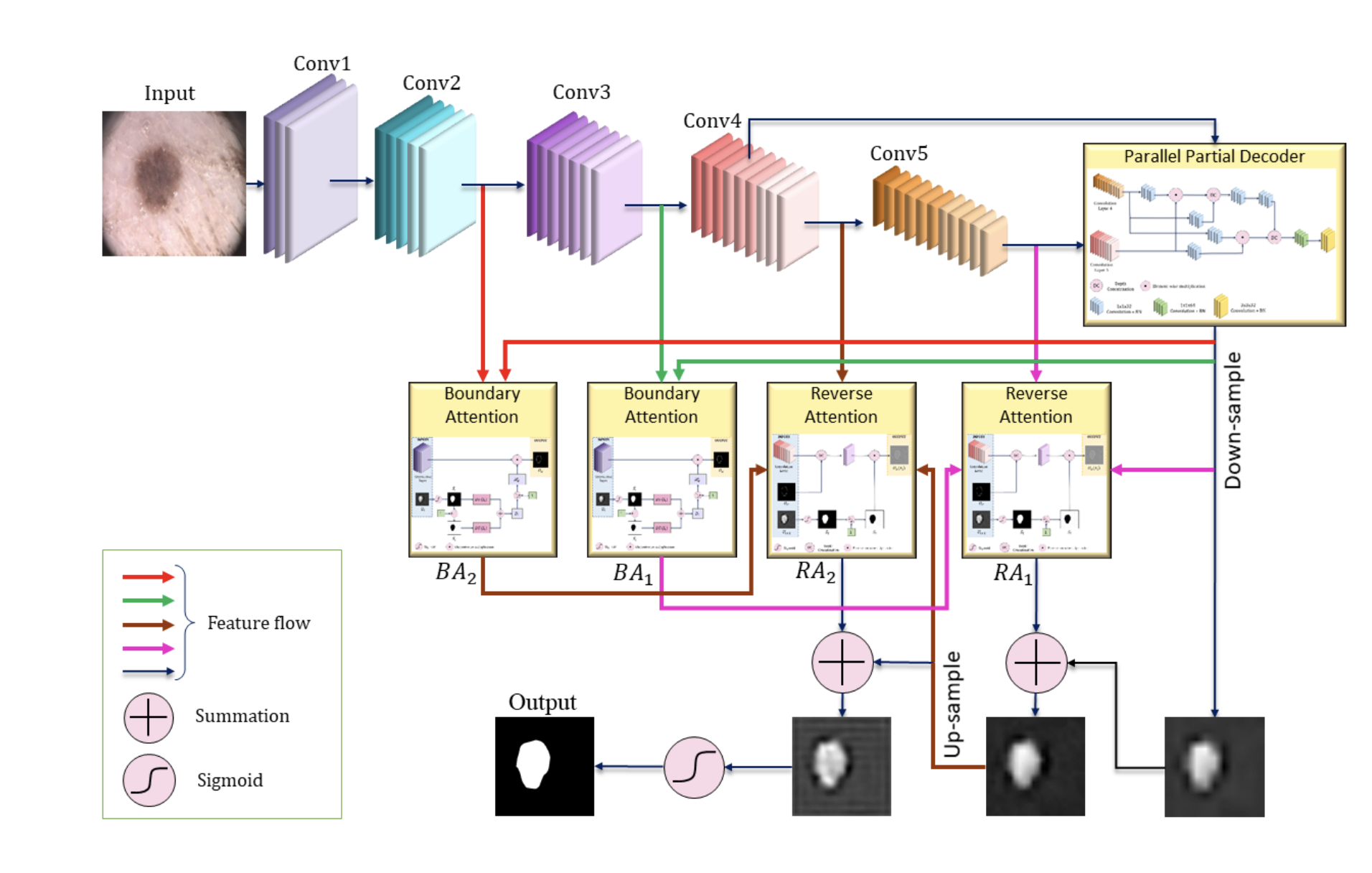




**Figure 2.9** SegNet Architecture

**MFSNet**:

The proposed framework, called MFSNet (Multi-Focus Segmentation Network), uses differently scaled feature maps for computing the final segmentation mask using raw input RGB images of skin lesions. In doing so, initially, the images are preprocessed to remove unwanted artifacts and noises. The MFSNet employs the Res2Net backbone, a convolutional neural network (CNN), for obtaining deep features used in a Parallel Partial Decoder (PPD) module to get a global map of the segmentation mask. In different stages of the network, convolution features and multi-scale maps are used in two boundary attention (BA) modules and two reverse attention (RA) modules to generate the final segmentation output.

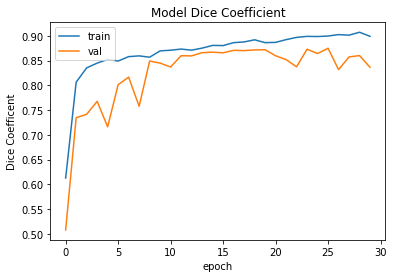
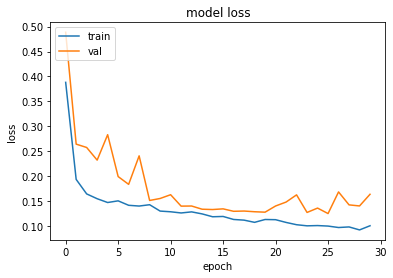


**Figure 2.10** MFSNet Architecture

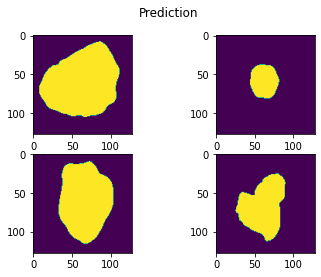
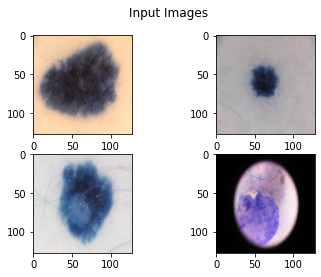
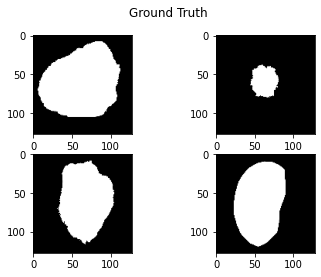
## Performance of the Model and Metrics

The models were developed in python using relevant libraries viz. Numpy, Tensorflow and Scipy.

We have trained all our models for 30 epochs. We will now look at the performance and metrics on each model.

**UNet**

**Figure 2.11:** UNet- Model Loss

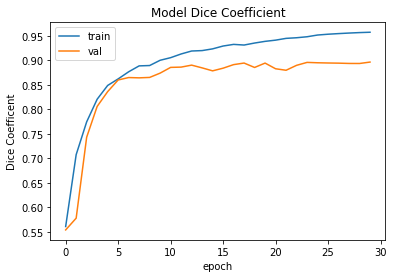
****

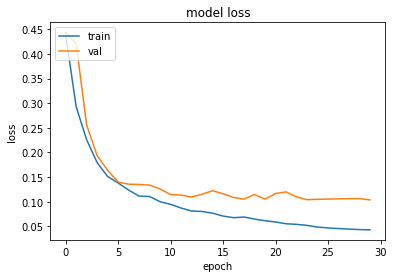
**Figure 2.12 :** UNet-Input image, Ground truth and Prediction

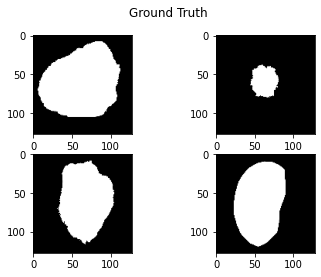
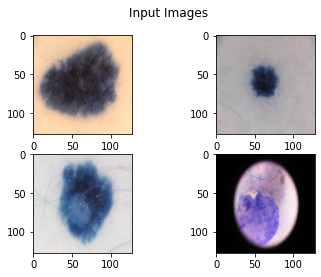
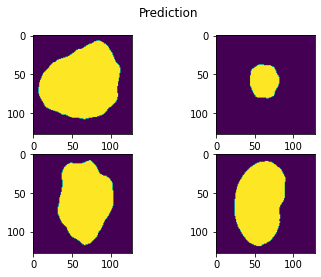
**UNet Model Performance:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1** | **IOU / Jaccard** |
| **VGG16** | 0.94977 | 0.9623 | 0.83299 | 0.85349 | 0.77792 |
| **EfficentNet** | 0.9564 | 0.96603 | 0.9099 | 0.87968 | 0.80303 |

**Figure 2.13 :** UNet Performance

**UNet ++**

**Figure 2.14:** UNet ++ Training performance

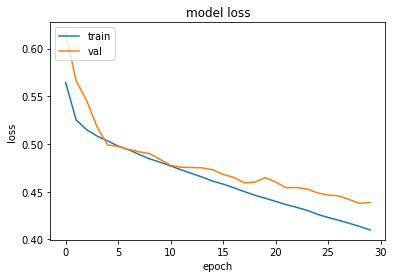
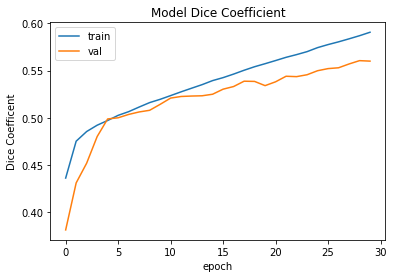


**Figure 2.15 :** UNet++ -Input image, Ground truth and Prediction

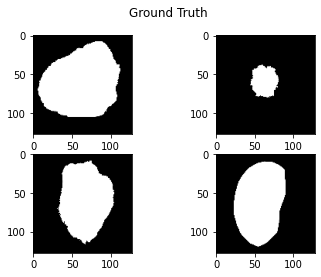
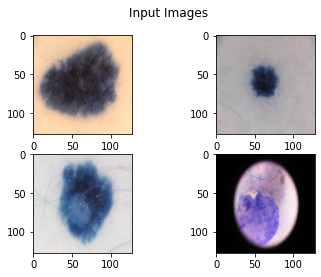
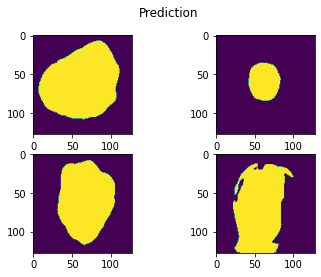
**UNet ++ Model Performance:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1** | **IOU / Jaccard** |
| **VGG16** | 0.96516 | 0.97202 | 0.92123 | 0.90895 | 0.84321 |
| **EfficentNet** | 0.95478 | 0.96213 | 0.86056 | 0.84608 | 0.75032 |

**Figure 2.16 :** UNet ++ model Performance

**Attention UNet**

**Figure 2.17:** Attention UNet Training performance

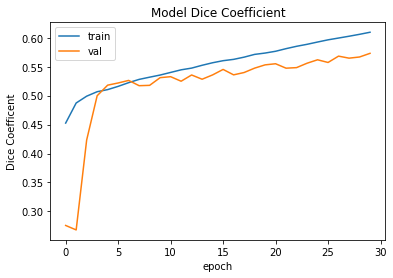
****

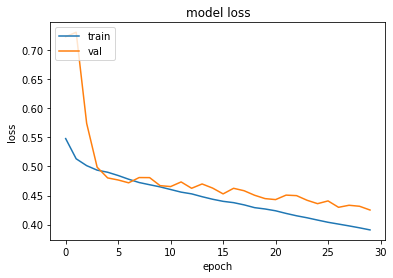
**Figure 2.18:** Attention UNet -Input image, Ground truth and Prediction

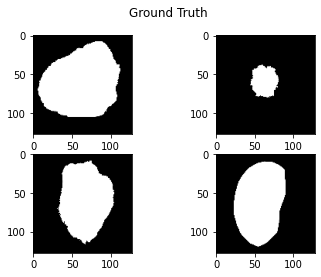
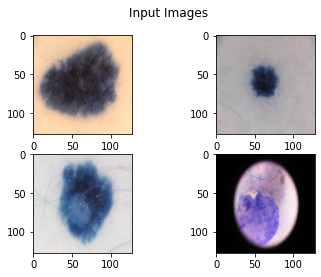
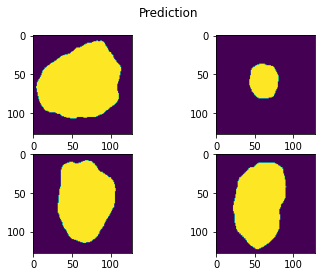
**Attention UNet Model Performance:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1** | **IOU / Jaccard** |
| **VGG16** | 0.92737 | 0.95303 | 0.92215 | 0.78256 | 0.67925 |
| **EfficentNet** | 0.93091 | 0.95052 | 0.89672 | 0.80228 | 0.7024 |

**Figure 2.19 :** Attention UNet Performance

**SegNet**

** Figure 2.20:** SegNet Training performance

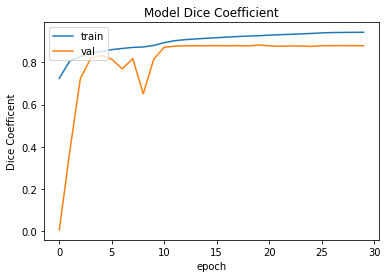


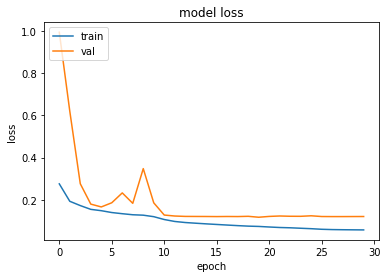
**Figure 2.21 :** SegNet -Input image, Ground truth and Prediction

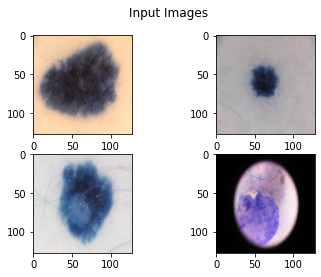
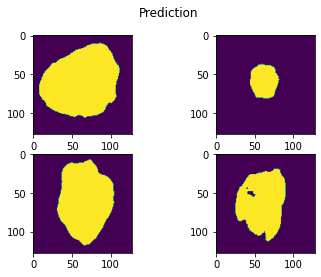
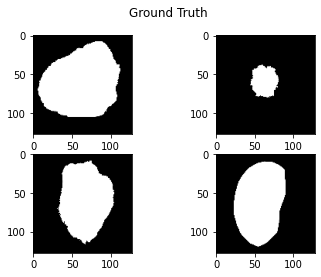
**SegNet Model Performance:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1** | **IOU / Jaccard** |
| **VGG16** | 0.94933 | 0.9611 | 0.91213 | 0.85237 | 0.7611 |
| **EfficentNet** | 0.95456 | 0.96306 | 0.89664 | 0.86838 | 0.78651 |

**Figure 2.22 :** SegNet Performance

**R2UNet**

** Figure 2.23:** R2UNet Training performance

****

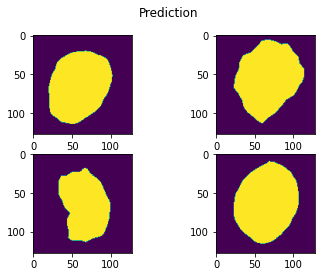
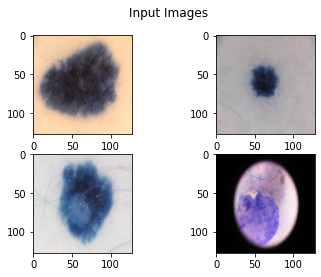
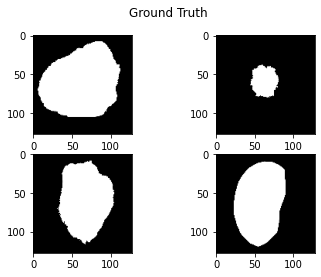
**Figure 2.24 :** R2UNet -Input image, Ground truth and Prediction

**R2UNet Model Performance:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1** | **IOU / Jaccard** |
| **VGG16** | 0.95905 | 0.9673 | 0.90419 | 0.88879 | 0.81766 |
| **EfficentNet** | 0.95832 | 0.96779 | 0.91463 | 0.88678 | 0.81557 |

**Figure 2.25 :** R2UNet Performance

**MFSNet**

****

**Figure 2.26 :** MFS -Input image, Ground truth and Prediction

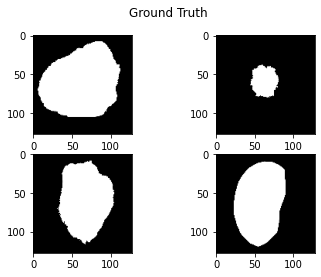
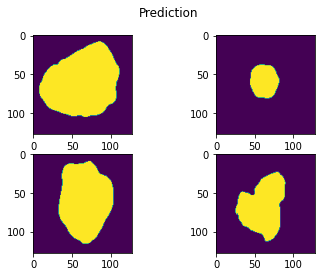
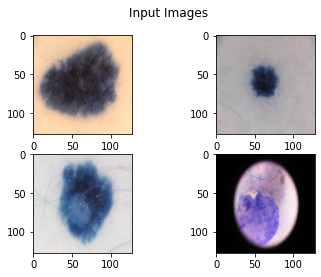
**MFSNet Model Performance:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1** | **IOU / Jaccard** |
| **VGG16** | 0.95328 | 0.96527 | 0.91641 | 0.83963 | 0.74453 |
| **EfficentNet** | 0.95418 | 0.83725 | 0.90676 | 0.84691 | 0.75187 |

**Figure 2.27 :** MFSNet Performance

**Ensemble Models**

We performed an Ensemble using 2 models UNet and UNet ++.

****

**Figure 2.28 :** Ensemble -Input image, Ground truth and Prediction

**Ensemble Model Performance:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1** | **IOU / Jaccard** |
| **VGG16** | 0.95323 | 0.96466 | 0.85087 | 0.86999 | 0.79665 |
| **EfficentNet** | 0.95828 | 0.91177 | 0.84282 | 0.86244 | 0.77443 |

**Figure 2.29 :** Ensemble Performance

## Overall project and improvements and applications and results

In order to run our models, we have used Google Colab Pro Plus. However due to the resource constraints we had to downsize the original image size from 512x512x1 to 128x128x1.

This enabled us to run the models successfully without crashing.

# Chapter III: Results

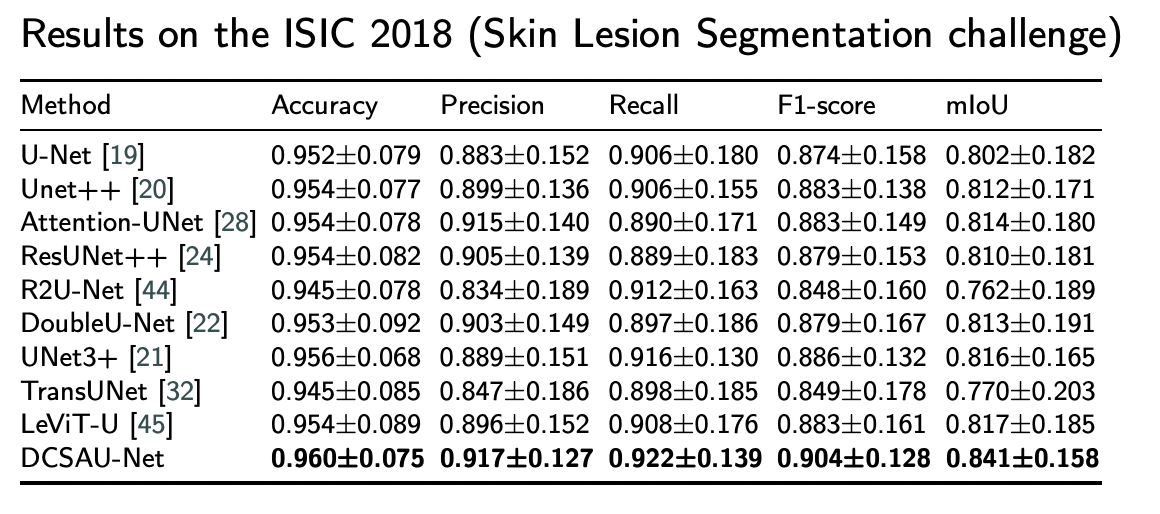
## Introduction

The main purpose of our project is to be able to create a model which can successfully draw the lesion boundaries from Dermoscopic images and identify the onset of skin cancer. This model should be easily adaptable in mobile devices so that it is easily usable by medical representatives.

## 

## Summary

In order to achieve our objective we have used the ISIC 2018 dataset containing Dermoscopic images. We have trained multiple models on the training dataset and we have compared the results to the original benchmarks achieved in the ISIC 2018 dataset challenge.



**Figure 3.1:** ISIC 2018 Benchmark Scores

**Results Summary of our models run on the training dataset**

**Performance using VGG16**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Unet** | **Unet ++** | **AttentionNet** | **R2UNet** | **SegNET** | **MFSNet** | **Ensemble** |
| **Accuracy** | 0.94977 | 0.96516 | 0.92737 | 0.95905 | 0.94933 | 0.95328 | 0.95323 |
| **Precision** | 0.9623 | 0.97202 | 0.95303 | 0.9673 | 0.9611 | 0.96527 | 0.96466 |
| **Recall** | 0.83299 | 0.92123 | 0.92215 | 0.90419 | 0.91213 | 0.91641 | 0.85087 |
| **F1** | 0.85349 | 0.90895 | 0.78256 | 0.88879 | 0.85237 | 0.83963 | 0.86999 |
| **IOU / Jaccard** | 0.77792 | 0.84321 | 0.67925 | 0.81766 | 0.7611 | 0.74453 | 0.79665 |

***Coloured cells signify highest values among all models***

**Figure 3.2:** VGG16 Performance Scores

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Unet** | **Unet ++** | **AttentionNet** | **R2UNet** | **SegNET** | **MFSNet** | **Ensemble** |
| **Accuracy** | 0.9564 | 0.95478 | 0.93091 | 0.95832 | 0.95456 | 0.95418 | 0.95828 |
| **Precision** | 0.96603 | 0.96213 | 0.95052 | 0.96779 | 0.96306 | 0.83725 | 0.91177 |
| **Recall** | 0.9099 | 0.86056 | 0.89672 | 0.91463 | 0.89664 | 0.90676 | 0.84282 |
| **F1** | 0.87968 | 0.84608 | 0.80228 | 0.88678 | 0.86838 | 0.84691 | 0.86244 |
| **IOU / Jaccard** | 0.80303 | 0.75032 | 0.7024 | 0.81557 | 0.78651 | 0.75187 | 0.77443 |

**Performance using EfficientNet**

***Coloured cells signify highest values among all models***

**Figure 3.3:** EfficientNet Performance Scores

We can see that from the above table our models are at par and have also beaten the benchmarks in most cases. When we used VGG16 UNet++ outperforms all models on most of the benchmark scores. However when we have used EffecientNet R2UNet outperforms all models.

Comparing the scores achieved using VGG16 and EfficientNet, UNet++ when used along with the VGG16 transfer learning network clearly outperforms all models.

## Conclusion

Basis our model performance we can conclude that our models have successful results.

The models have improved over the benchmarks in most scenarios.

We would like to state that dues to a resource constraint we had to reduce the size of the input images. This could be a factor for the difference in performance. In order to mitigate this problem, we could additionally try methods like bilinear interpolation to enhance the images to 512 size and not increase the need for higher compute.

We have experimented with MFSNet and this is an evolving model which has promising results. However we need to finetune it to extract good performance.

There is a need for constantly monitoring the developments in image processing and models using convolutional networks so as to improve the models that have been used. Better models will improve early detection of skin cancer.

# References

[1] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, Allan Halpern: "Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)", 2018; https://arxiv.org/abs/1902.03368

[2] Tschandl, P., Rosendahl, C. & Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci. Data 5, 180161 doi:10.1038/sdata.2018.161 (2018).

[3] U-Net: Convolutional Networks for Biomedical Image Segmentation. lmb.informatik.uni-freiburg.de. Retrieved 2018-12-24.

[4] Shelhamer E, Long J, Darrell T (April 2017). "Fully Convolutional Networks for Semantic Segmentation". IEEE Transactions on Pattern Analysis and Machine Intelligence. 39 (4): 640–651.

[5] Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J. Unet++: A nested u-net architecture for medical image segmentation. Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support, Springer; 2018, p. 3–11. - PMC – PubMed

[6] Mubashar, M., Ali, H., Grönlund, C. et al. R2U++: a multiscale recurrent residual U-Net with dense skip connections for medical image segmentation. Neural Comput & Applic 34, 17723–17739 (2022). https://doi.org/10.1007/s00521-022-07419-7

[7] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel RueckertAttention U-Net: Learning where to look for Pancreas

[8] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

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

[10] Tan, Mingxing and Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” *ArXiv* abs/1905.11946 (2019): n. pag.

[11] Tammina, Srikanth. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. International Journal of Scientific and Research Publications (IJSRP). 9. p9420. 10.29322/IJSRP.9.10.2019.p9420.