# Skin Cancer Segmentation

Supervisor: Prof. Francesco Tortorella

Tajwar Abrar Aleef
Yeman Brhane Hagos
Vu Hoang Minh

MAIA

June 19, 2017

# Outline

# Outline

# Introduction
## Skin Cancer

- Melanoma - the most dangerous form of skin cancer
- The rates of melanoma have been rising for the last 30 years
- Every year, about **5.4 million new cases** of skin cancer are diagnosed in the United States
- Early detection is the key to proper treatment of skin cancer
  - early-stage: **97%** five-year survival rate
  - final-stage: **14%** five-year survival rate

- Melanoma - the most dangerous form of skin cancer
- The rates of melanoma have been rising for the last 30 years
- Every year, about **5.4 million new cases** of skin cancer are diagnosed in the United States
- Early detection is the key to proper treatment of skin cancer
  - early-stage: **97%** five-year survival rate
  - final-stage: **14%** five-year survival rate

# Introduction

## The danger of Skin Cancer

- Melanoma - the most dangerous form of skin cancer
- The rates of melanoma have been rising for the last 30 years
- Every year, about **5.4 million new cases** of skin cancer are diagnosed in the United States
- Early detection is the key to proper treatment of skin cancer
  - early-stage: **97%** five-year survival rate
  - final-stage: **14%** five-year survival rate

- Melanoma - the most dangerous form of skin cancer
- The rates of melanoma have been rising for the last 30 years
- Every year, about **5.4 million new cases** of skin cancer are diagnosed in the United States
- Early detection is the key to proper treatment of skin cancer
    - early-stage: **97%** five-year survival rate
    - final-stage: **14%** five-year survival rate

- Detection of malignant melanoma in its early stages considerably reduces morbidity and mortality
- Early detection also **saves hundreds of millions of dollars** that otherwise would be spent on the treatment of advanced diseases
- Skin Cancer Detection is needed

- Detection of malignant melanoma in its early stages considerably reduces morbidity and mortality
- Early detection also **saves hundreds of millions of dollars** that otherwise would be spent on the treatment of advanced diseases
- Skin Cancer Detection is needed

- Detection of malignant melanoma in its early stages considerably reduces morbidity and mortality
- Early detection also **saves hundreds of millions of dollars** that otherwise would be spent on the treatment of advanced diseases
- Skin Cancer Detection is needed

- The International Skin Imaging Collaboration (ISIC) is an international effort to improve melanoma diagnosis
- The goal of the challenge is to help participants develop image analysis tools to **enable the automated diagnosis of melanoma** from dermoscopic images
- Lesion Segmentation challenge provides training data ($\sim$2000 images) with size of $3000 \times 2000$ pixels

ISBI 2017

- The International Skin Imaging Collaboration (ISIC) is an international effort to improve melanoma diagnosis
- The goal of the challenge is to help participants develop image analysis tools to **enable the automated diagnosis of melanoma** from dermoscopic images
- Lesion Segmentation challenge provides training data ($\sim$2000 images) with size of $3000 \times 2000$ pixels

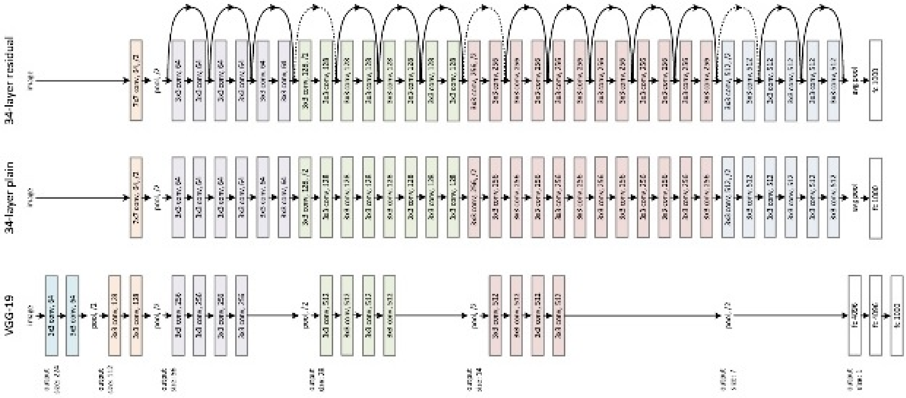
ISBI 2017

- The International Skin Imaging Collaboration (ISIC) is an international effort to improve melanoma diagnosis
- The goal of the challenge is to help participants develop image analysis tools to **enable the automated diagnosis of melanoma** from dermoscopic images
- Lesion Segmentation challenge provides training data ($\sim$2000 images) with size of $3000 \times 2000$ pixels

# Outline

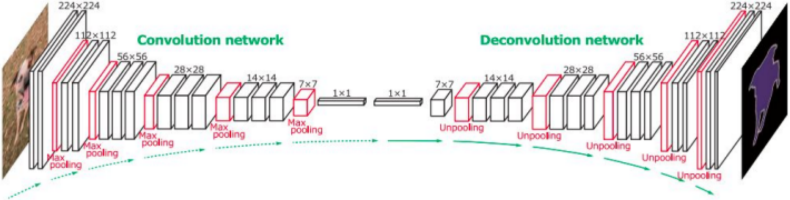
**Pattern analysis outperforms the rest!!!**

# Approaches
## Deep Residual Networks



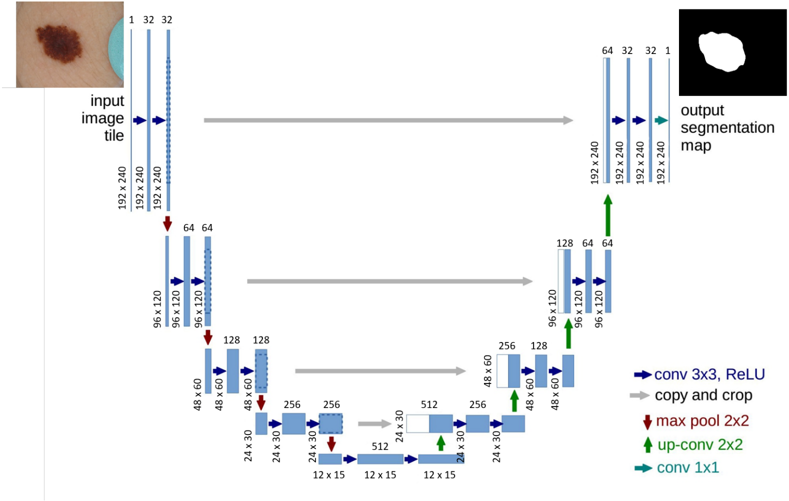
# Approaches

## Convolutional-Deconvolutional Neural Networks



# Our approach

## Modified U-Net



- Python and Matlab
- **Deeplearning4j** deep dearning for Java
- **Caffee** developed by Berkeley AI Research
- **TensorFlow** developed by Google
- **Keras** deep learning API with TensorFlow in the back end

# Approaches

## Frameworks and Programming languages

- Python and Matlab
- **Deeplearning4j** deep dearning for Java
- **Caffee** developed by Berkeley AI Research
- **TensorFlow** developed by Google
- **Keras** deep learning API with TensorFlow in the back end

# Approaches

## Frameworks and Programming languages

- Python and Matlab
- **Deeplearning4j** deep dearning for Java
- **Caffee** developed by Berkeley AI Research
- **TensorFlow** developed by Google
- **Keras** deep learning API with TensorFlow in the back end

Caffe

- Python and Matlab
- **Deeplearning4j** deep dearning for Java
- **Caffee** developed by Berkeley AI Research
- **TensorFlow** developed by Google
- **Keras** deep learning API with TensorFlow in the back end

# Approaches

## Frameworks and Programming languages

- Python and Matlab
- **Deeplearning4j** deep dearning for Java
- **Caffee** developed by Berkeley AI Research
- **TensorFlow** developed by Google
- **Keras** deep learning API with TensorFlow in the back end

# Outline

# Our Approach

## Pre-processing

- Image vectorization
- Mean subtraction and normalization

$$X = \frac{x - \mu}{\sigma} \quad (1)$$

where $x$ is input image, $\mu$ is mean of pixel values, $\sigma$ is standard deviation of pixels in the image, and $X$ is normalized image.

- Image resizing $192 \times 240$

# Our Approach
## Pre-processing

- Image vectorization
- Mean subtraction and normalization

$$X = \frac{x - \mu}{\sigma} \tag{1}$$

where $x$ is input image, $\mu$ is mean of pixel values, $\sigma$ is standard deviation of pixels in the image, and $X$ is normalized image.

- Image resizing $192 \times 240$

# Our Approach
## Pre-processing

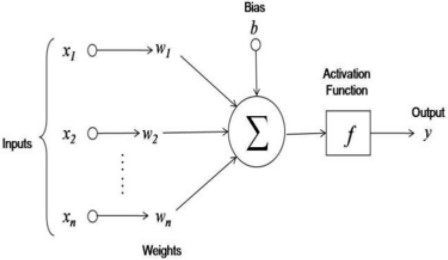
- Image vectorization
- Mean subtraction and normalization

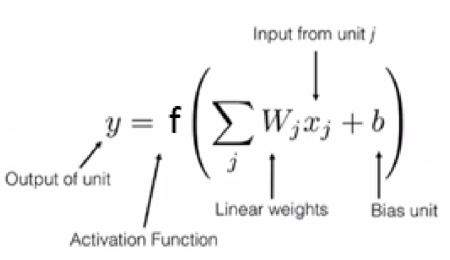
$$X = \frac{x - \mu}{\sigma} \tag{1}$$

where $x$ is input image, $\mu$ is mean of pixel values, $\sigma$ is standard deviation of pixels in the image, and $X$ is normalized image.

- Image resizing $192 \times 240$

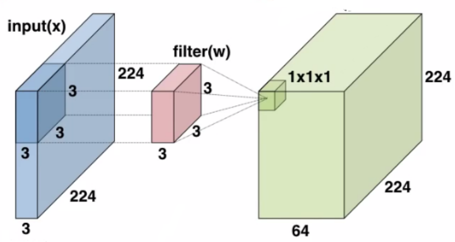
# Our Approach
## Mathematical Model of Neuron

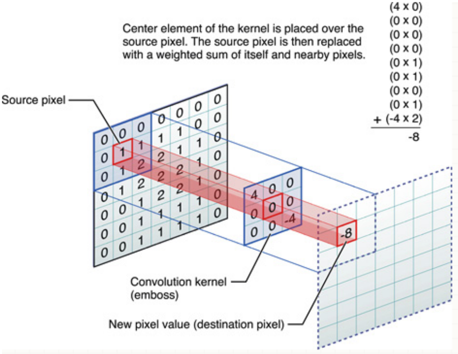


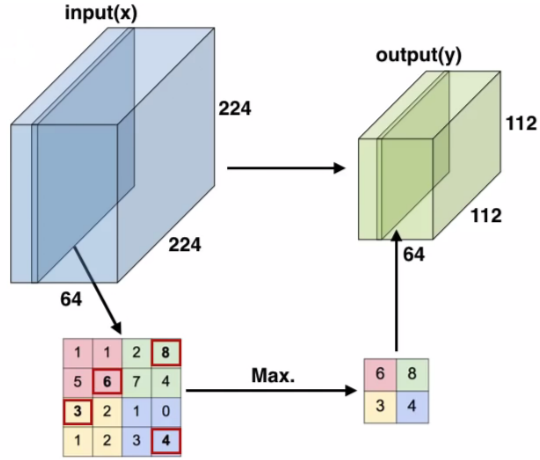
# Our Approach

## Convolutional Layer

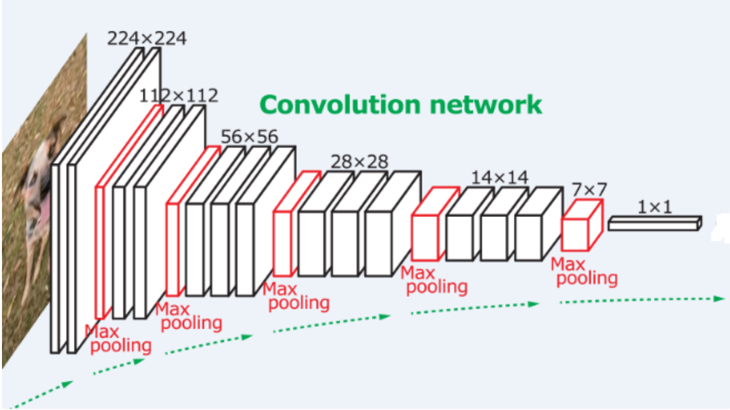


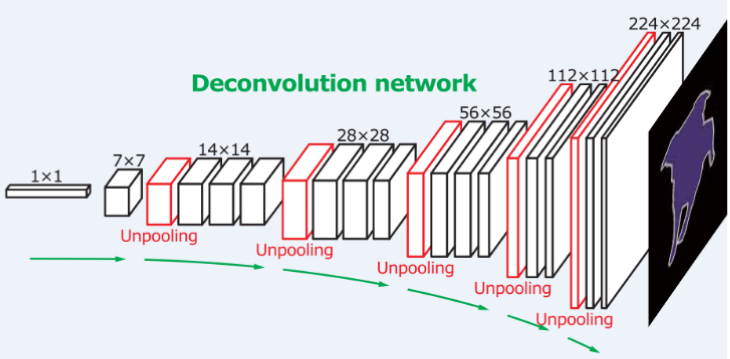
# Our Approach
## Max Pooling Layer

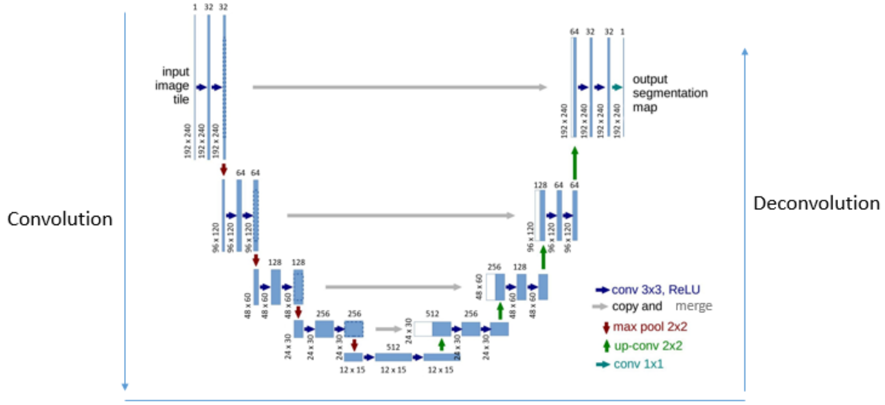


# Our Approach
## Convolutional Neural Network



# Our Approach
## Deconvolutional Neural Network





Convolution
Deconvolution
input image tile
output segmentation map
conv 3x3, ReLU
copy and merge
max pool 2x2
up-conv 2x2
conv 1x1
1 32 32
192 x 240
64 64
96 x 120
128 128
48 x 60
256 256
24 x 30
512
12 x 15
64 32 32 1
128 64 64

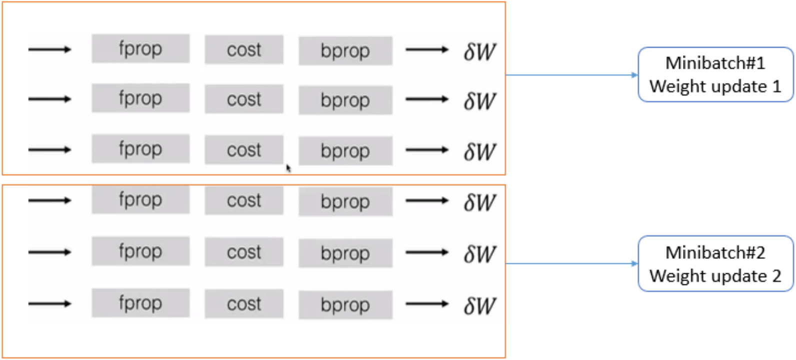
# Our Approach

## Training

# Our Approach
## Training

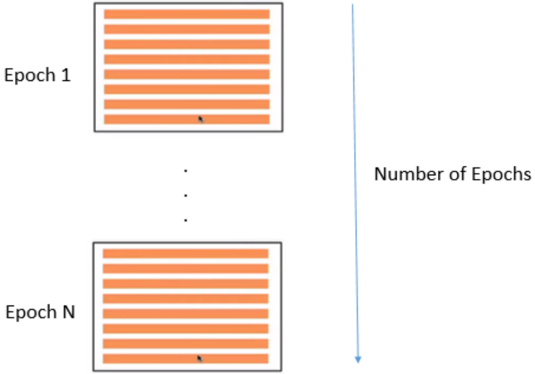
# Our Approach
## Training Mini Batch Size

- Number of training images in one forward and backward pass
- Mini batch-size = 10



# Our Approach
## Training Epoch

- How many times every image has been seen during training?
- Shuffling



# Our Approach
## Training Parameters

- **Learning rate**
  - Step size for which the weights of model are updated
  - $10^{-5}$
- **Loss function** (also called cost function)
  - Evaluate the penalty between the prediction and the ground truth label in every batch
  - Dice Coefficient is used as loss function
- **Optimizer**
  - Optimal set of hyper parameters for the model
  - Adam optimization algorithms

# Our Approach
## Training Parameters

- **Learning rate**
  - Step size for which the weights of model are updated
  - $10^{-5}$
- **Loss function** (also called cost function)
  - Evaluate the penalty between the prediction and the ground truth label in every batch
  - Dice Coefficient is used as loss function
- **Optimizer**
  - Optimal set of hyper parameters for the model
  - Adam optimization algorithms

- **Learning rate**
  - Step size for which the weights of model are updated
  - $10^{-5}$
- **Loss function** (also called cost function)
  - Evaluate the penalty between the prediction and the ground truth label in every batch
  - Dice Coefficient is used as loss function
- **Optimizer**
  - Optimal set of hyper parameters for the model
  - Adam optimization algorithms

# Our Approach
## Testing

- Test image
- Launch the trained model

# Outline

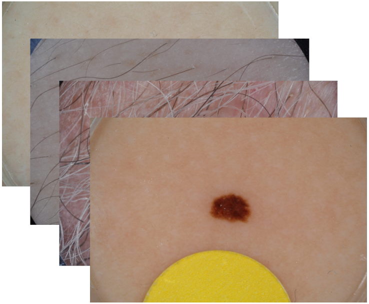
# Result and Discussion

## Test Bench

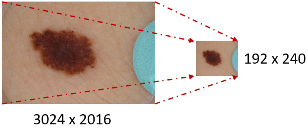
- Dell Inspirion 15 with 4 GB NVIDIA GEFORCE 960M GPU
- University of Cassino Computer with 12 GB NVIDIA TITAN X GPU
- 450 epochs take **81 hours of training**

150 Test sets of three classes:

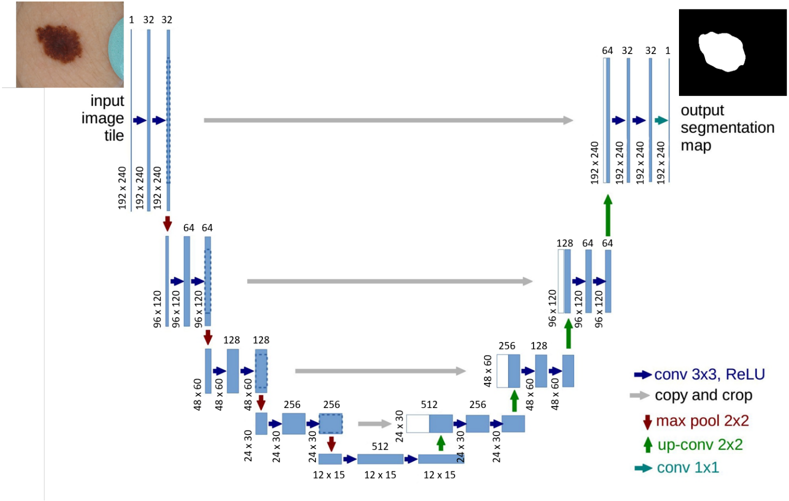
- 30 Melanoma – malignant skin tumor
- 78 Nevus – benign skin tumor (melanocytic)
- 42 Seborrheic keratosis – benign skin tumor



1. Images are resized to 192 x 240 pixels
2. Then images are vectorized and stored in numpy array file

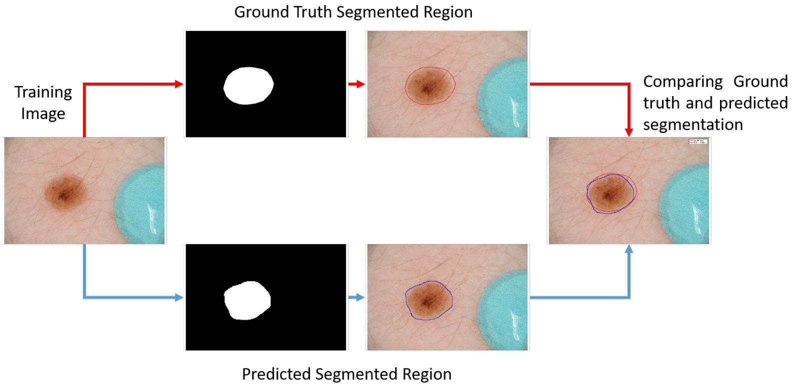


# Result and Discussion
## Test Data



# Result and Discussion

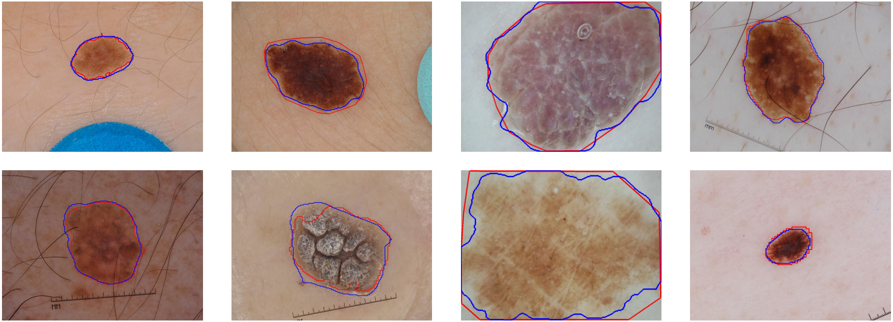
## Flowchart



# Result and Discussion

## Predicted segmentation & Ground truth mask on some of the test images

The red mask denotes the ground truth while the blue mask denotes the predicted mask from the convolutional neural network.

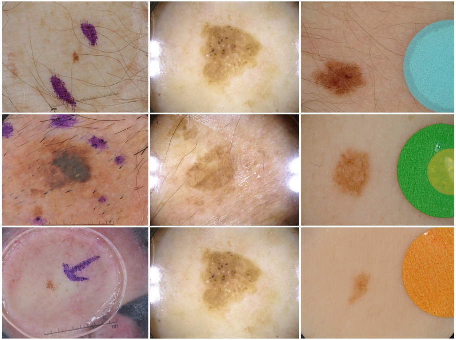


# Result and Discussion

## Problems of Dataset of Skin Lesion Segmentation Challenge on ISBI 2017

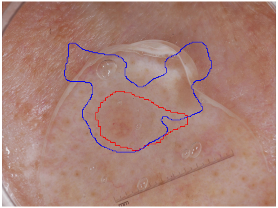
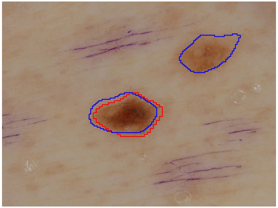
From the training set, many of the training images had artifacts

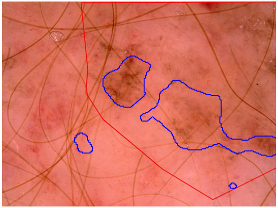
- Images with markings
- Images with bright lights at the edge
- images with color patches

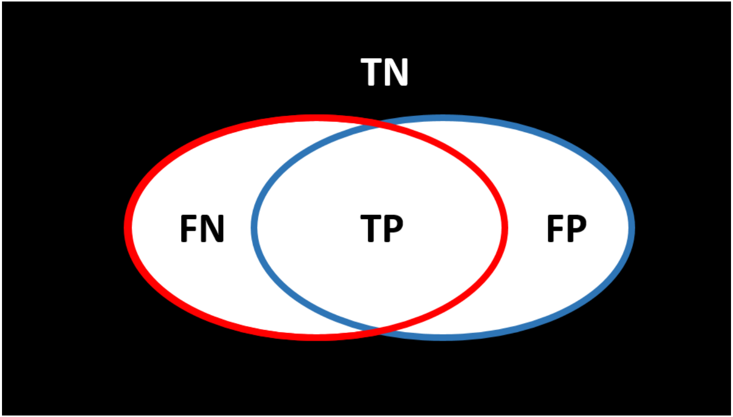


# Result and Discussion

## Some of the poorly segmented results from the testing set



TN
FN
TP
FP

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

$$\text{Jaccard Index} = \frac{\text{TP}}{\text{TP}+\text{FN}+\text{FP}}$$

$$\text{Dice Coefficient} = \frac{2 \times \text{TP}}{2 \times \text{TP}+\text{FN}+\text{FP}}$$

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

$$\text{Specificity} = \frac{\text{TN}}{\text{TN}+\text{FP}}$$

Table: Comparison of the results

| Methods | Accuracy | Dice Coefficient | Jaccard Index | Sensitivity | Specificity |
|---|---|---|---|---|---|
| **Our Approach** | **0.9403** | **0.8053** | **0.7116** | **0.8531** | **0.9308** |
| Yading Yuan | 0.9340 | 0.8490 | 0.7650 | 0.8250 | 0.9750 |
| Matt Berseth | 0.9320 | 0.8470 | 0.7620 | 0.8200 | 0.9780 |
| Lei Bi | 0.9340 | 0.8440 | 0.7600 | 0.8020 | 0.9850 |
| Euijoon Ahn | 0.9340 | 0.8420 | 0.7580 | 0.8010 | 0.9840 |
| Recod Titans | 0.9310 | 0.8390 | 0.7540 | 0.8170 | 0.9700 |
| Jeremy Kawahara | 0.9300 | 0.8370 | 0.7520 | 0.8130 | 0.9760 |
| Jahanifar Zamani | 0.9300 | 0.8390 | 0.7490 | 0.8100 | 0.9810 |
| INESC TECHNALIA | 0.9220 | 0.8240 | 0.7350 | 0.8130 | 0.9680 |
| Vic Lee | 0.9220 | 0.8100 | 0.7180 | 0.7890 | 0.9750 |
| Juana M.Guitierrez | 0.9150 | 0.7970 | 0.7150 | 0.7740 | 0.9700 |

# Outline

# Conclusion

## Problems Faced

Problems faced during the implementation of the CNN on Skin Cancer segmentation

- Time constraint: Training the dataset even without data augmentation takes a lot of time
- Dataset limited: Limited training set, most of the images has background artifact

Problems faced during the implementation of the CNN on Skin Cancer segmentation

- Time constraint: Training the dataset even without data augmentation takes a lot of time
- Dataset limited: Limited training set, most of the images has background artifact

Although, we were still able to get better percentage of accuracy compared to the other methods used on the same dataset, further improvements are still possible to yield even a better result:

- **Data Augmentation:** The limited data can be augmented using translation, rotation, flipping and elastic distortion
- **Cropping:** Cropping can be performed on the training set images with unnecessary objects, i.e unwanted hair, color patch, markings, bright spots etc.

# Q & A

Thank you for listening