Indian Institute of Technology Jodhpur

Image segmentation using pre-trained MobileNet as an Encoder

Submitted by

MITESH KUMAR (M23MAC004)

Colab Link: M23MAC004.ipynb

Pre-Processing:

- Define a load image function for image pre-processing. It extracts images and masks from 4 different folders and sorts them according to their name.
- Resize all images into 128 x 128 the images and convert them into tensors.

Model architecture:

- Pre-trained MobileNetV2 is used as an encoder for the feature extractor.
- The decoder contains 9 layers which is half of the number in the encoder.
- It upsamples the image size from 1280x4x4 to 1x128x128.
- To deal with the vanishing gradient there is a total of 9 skip connections between the encoder and the decoder.
- To avoid overfitting dropouts are applied.
- Model 1 is in which the encoder is frozen.
- Model 2 is fine-tuned.

Results:

Model 1:

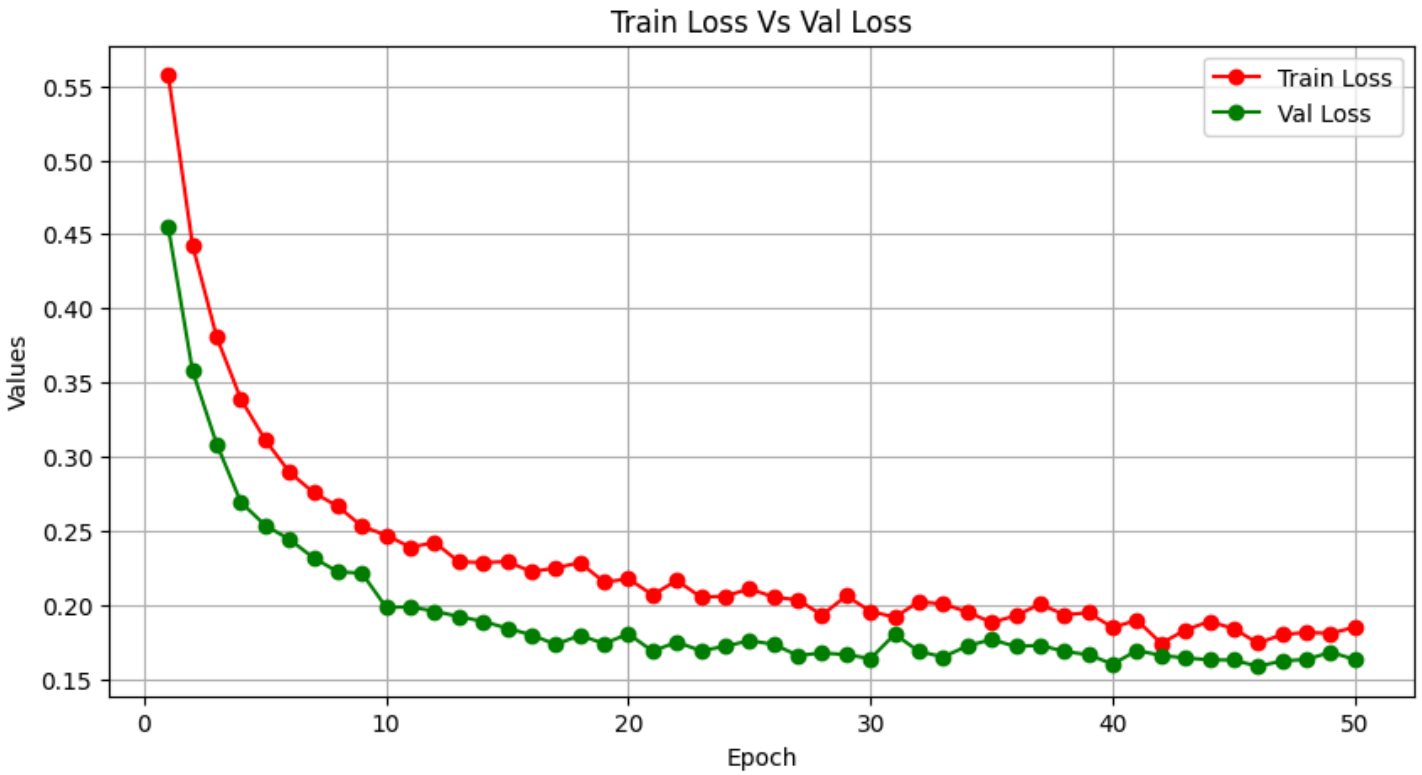
Enoder lr = 0 , decoder lr = 0.01

epoche = 50

Loss function = Binary cross entropy

Batch size = 45

Epoche	Training Loss	Validation Loss	Validation IoU	Validation Dice score
1	0.5574	0.4549	3178	0.4544
10	0.2469	0.1981	0.5434	0.6787
20	0.2174	0.1803	0.5893	0.7180
30	0.1955	0.1636	0.6358	0.7573
40	0.1846	0.1599	0.6497	0.7682
50	0.1848	0.1630	0.6454	0.7641



Number of trainable parameters: 1594314
Number of non-trainable parameters: 2223872

Model 2:

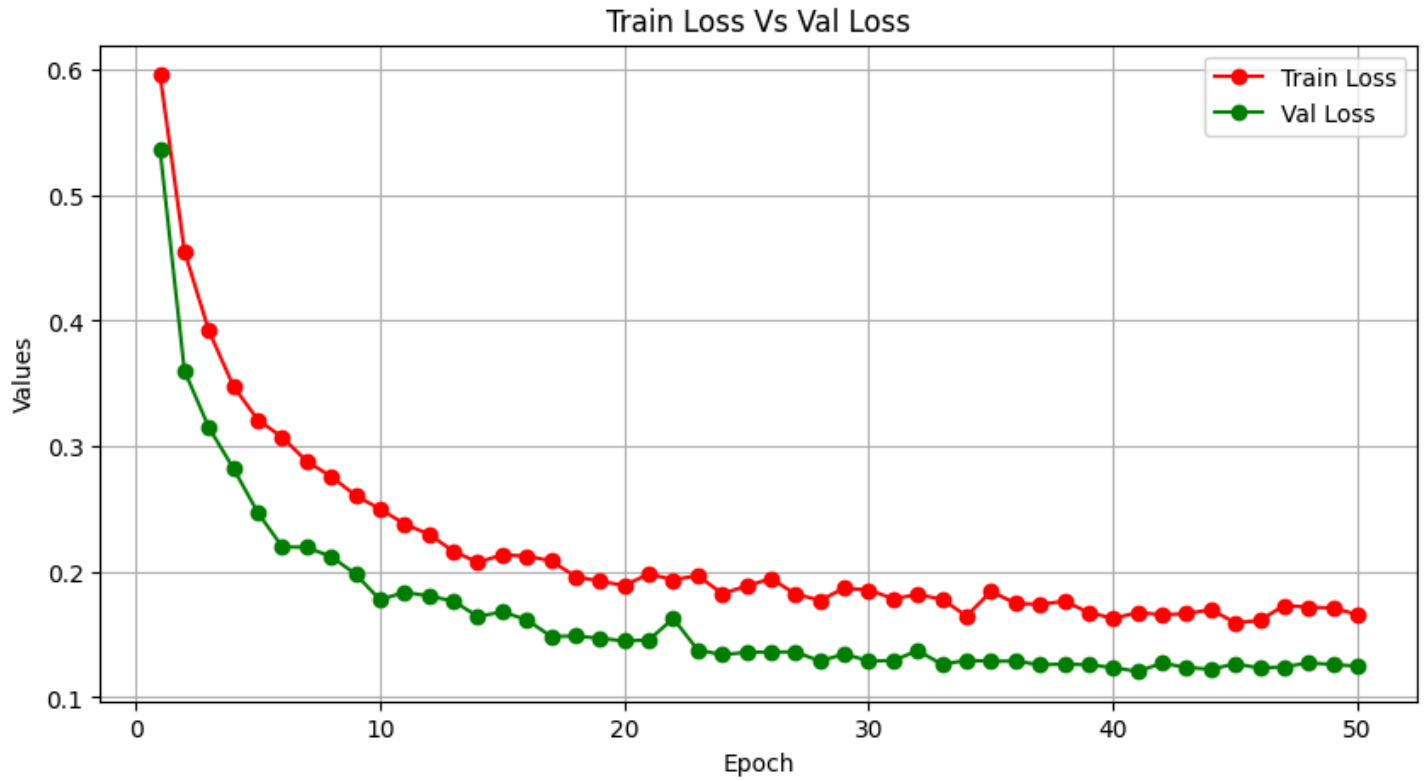
Enoder lr = 0.0001 , decoder lr = 0.01

epoche = 50

Loss function = Binary cross entropy

Batch size = 45

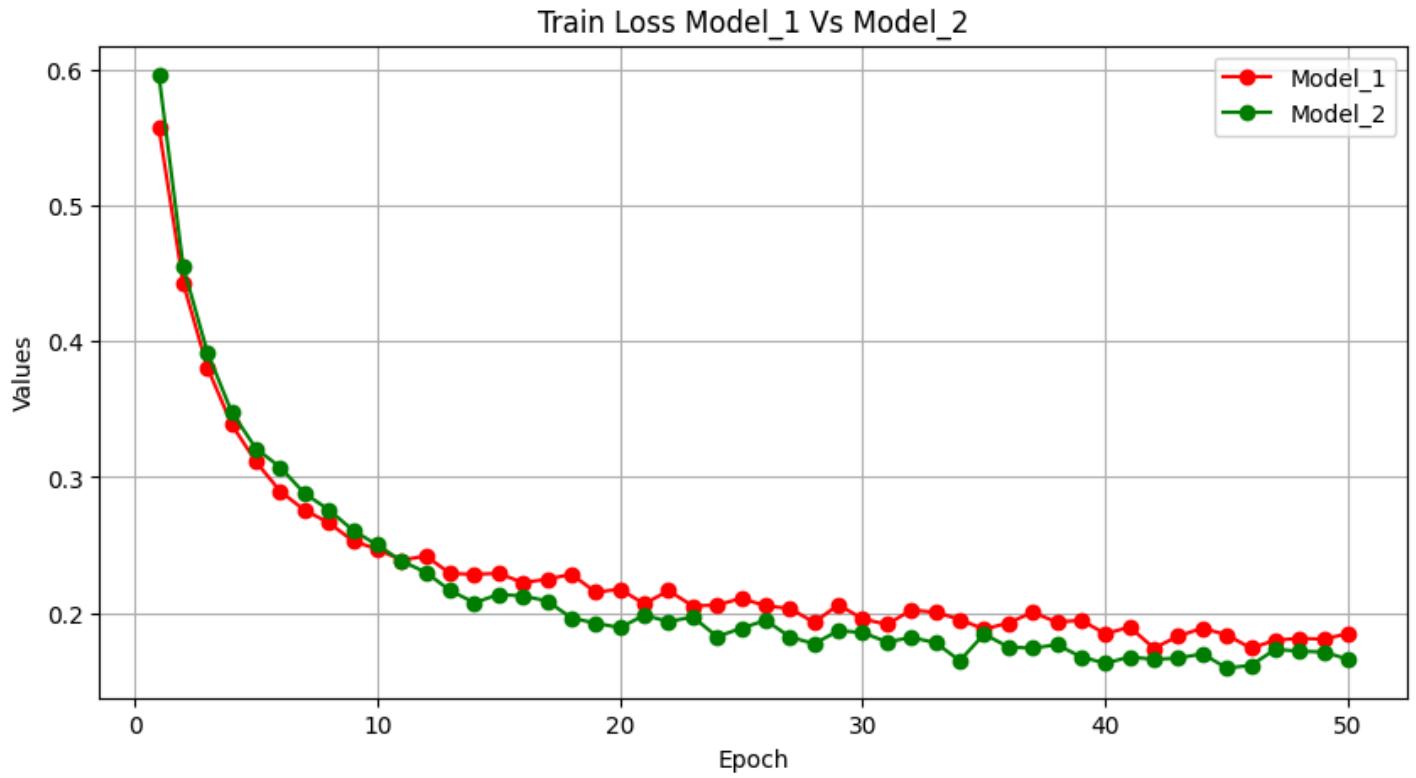
Epoche	Training Loss	Validation Loss	Validation IoU	Validation Dice score
1	0.5958	0.5364	0.3833	0.5191
10	0.2500	0.1777	0.5392	0.6793
20	0.1891	0.1448	0.6354	0.7606
30	0.1855	0.1290	0.6901	0.8024
40	0.1629	0.1237	0.7194	0.8264
50	0.1658	0.1248	0.7239	0.8295

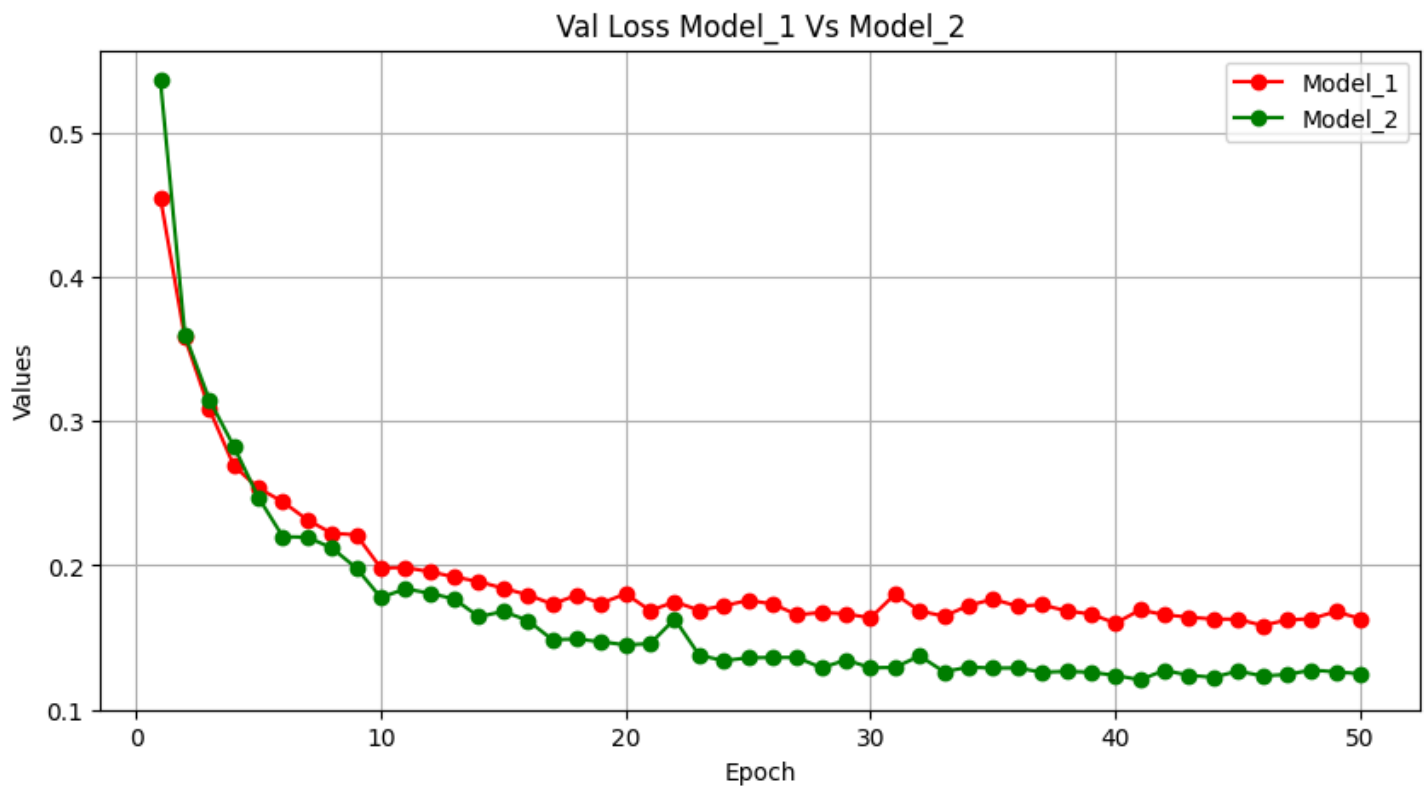


Number of trainable parameters: 3818186
Number of non-trainable parameters: 0

Comparison between models

Below are the graphs that represent variation in the losses of model 1 and model 2 for 50 epochs.


Train Loss Model_1 Vs Model_2
Model_1
Model_2
Values
Epoch
0.6
0.5
0.4
0.3
0.2
0
10
20
30
40
50


Val Loss Model_1 Vs Model_2
Model_1
Model_2
Values
Epoch
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50

Model Name	Learning rate	Training Loss	Validation Loss	Validation IoU	Validation Dice Score
Model 1	0 & 0.01	0.1848	0.1630	0.6454	0.7641
Model 2	0.0001 & 0.01	0.1658	0.1248	0.7239	0.8295

Model 1 = encoder is freeze
Model 2 = encoder is fine-tuned

Conclusion:
From the above two graphs, it is inferred that

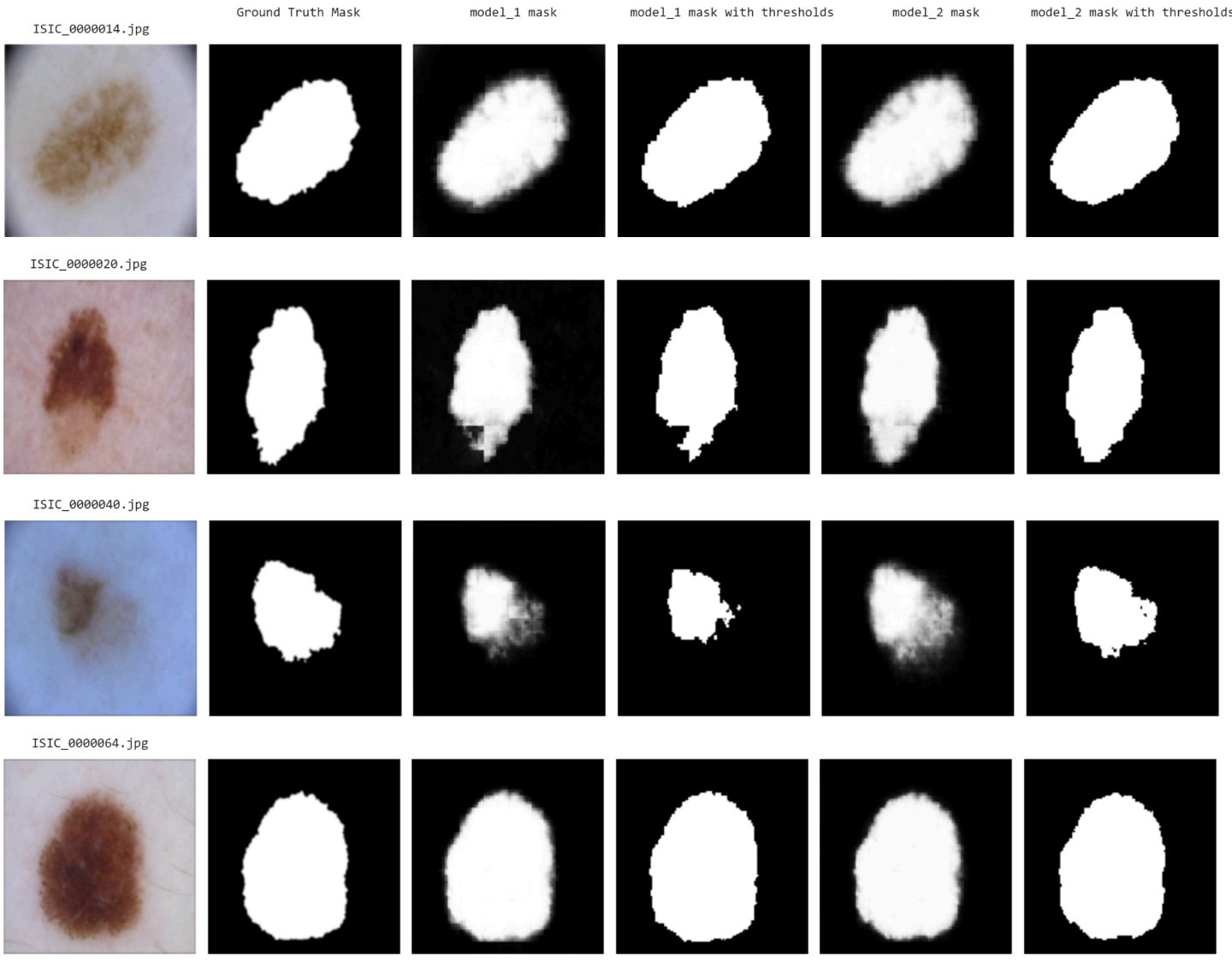
- the training and validation loss of model 2 is less than that of model 1.
- IoU and dice score is more than model 2 is more than that of model 1.
- So fine-tuning the encoder architecture to perform better results than the freezen one.
- Below are some sample images of predicted masks and the predicted mask of model 2 is better than model 1.

Sample Images, its true mask and predicted mask.
Below there are some sample images and corresponding masks.
Each row contains 6 images.
1st image = real image
2nd image = true mask

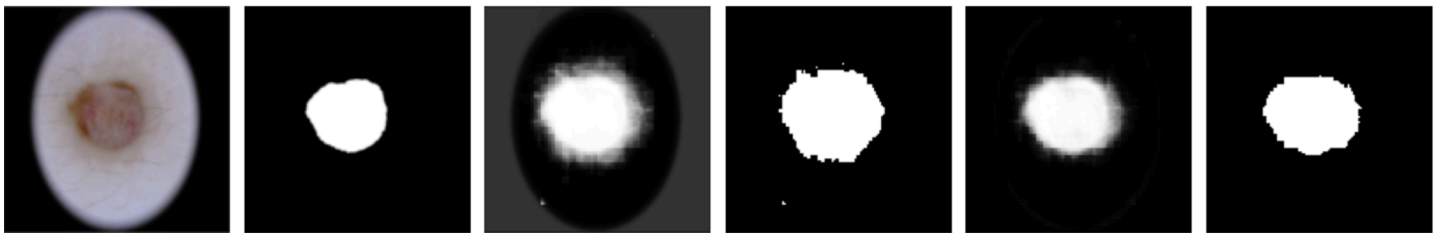
3rd image = model 1 predicted mask
4th image = model 1 predicted mask with thresholding of 0.5
5th image = model 1 predicted mask
6th image = model 2 predicted mask with thresholding of 0.5

Here thresholding means that the pixel values that are greater than 0.5 are assigned numerical values of 1 and the remaining are assigned value numerical values of 0.

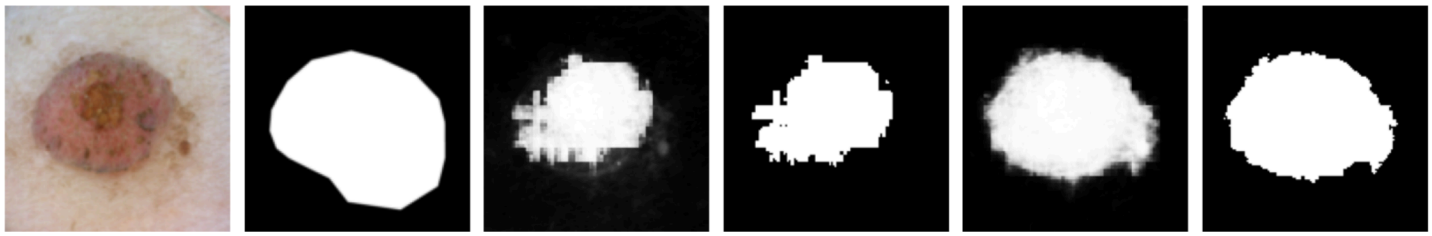
Below images



ISIC_0000071.jpg



ISIC_0000099.jpg



ISIC_0000003.jpg