Segmentation Perspectives: From Skin Lesions to Retinal Blood Vessels

Samer Bujana, Rasmus Clausen, Alba Gonzalo, Robert Jarvi, Leonardo Rodovero

INTRODUCTION

Sometimes the difference between two projects is not big enough to invest time in more than one model. Following this idea, in this project we have two different datasets that we are trying to apply segmentation to, one with retinal vessels, and one with skin lesions. The tasks at hand are so similar that we only developed one model that is going to be used for both datasets. We will do this twice, once with an encoder-decoder, and then with a U-net.

DATASETS

RETINAL VESSELS

- 20 Training images with manual labels and target masks
- 20 Test images with target masks but without manual labels
- Original image resolution 565×584 resized to 256x256
- 70-15-15 Train/Validation/Test split

SKIN LESIONS

- 200 images, not already split
- Image and target mask provided for each
- Original image resolution 576×767 resized to 128x128
- 70-15-15 Train/Validation/Test split

MODEL ARQUITECTURES

ENCODER-DECODER

× 4
- Conv2d (3 × 3)
- Padding = 1
- BatchNorm2d
- ReLU
- MaxPooling2d (2 × 2)

ENCODER

→ × 1
- Conv2d (3 × 3)
- Padding = 1
- ReLU

BOTTLENECK

→ × 4
- Upsample
- Conv2d (3 × 3) Padding = 1

DECODER

U-NET

Feature Size
256 × 256
128 × 128
64 × 64
32 × 32
16 × 16

× 3
- Conv2d (3 × 3) Padding = 1
- BatchNorm
- ReLU
- Conv2d (3 × 3) Padding = 1, Stride =1
- BatchNorm
- ReLU
- Conv2d (3 × 3) Padding = 2, Stride =2

ENCODER

→ × 3
- Conv2d(3 × 3) Padding = 1
- BatchNorm2d
- ReLU

BOTTLENECK

→ × 3
- ConvTransposed2d (2 × 2) ; Stride = 2
- Skip Connections
- Conv2d (3 × 3) ; Padding = 1
- BatchNorm
- ReLU
- Conv2d (3 × 3); Padding = 1
- BatchNorm
- ReLU
- Conv2d (3 × 3) ; Padding = 1

DECODER

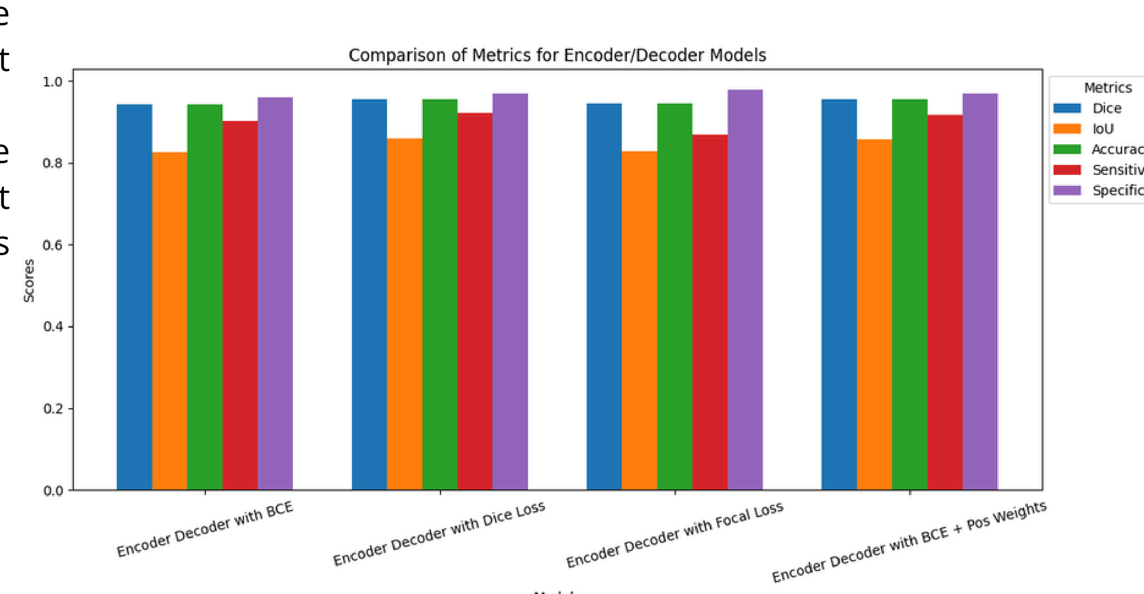
PERFORMANCE METRICS

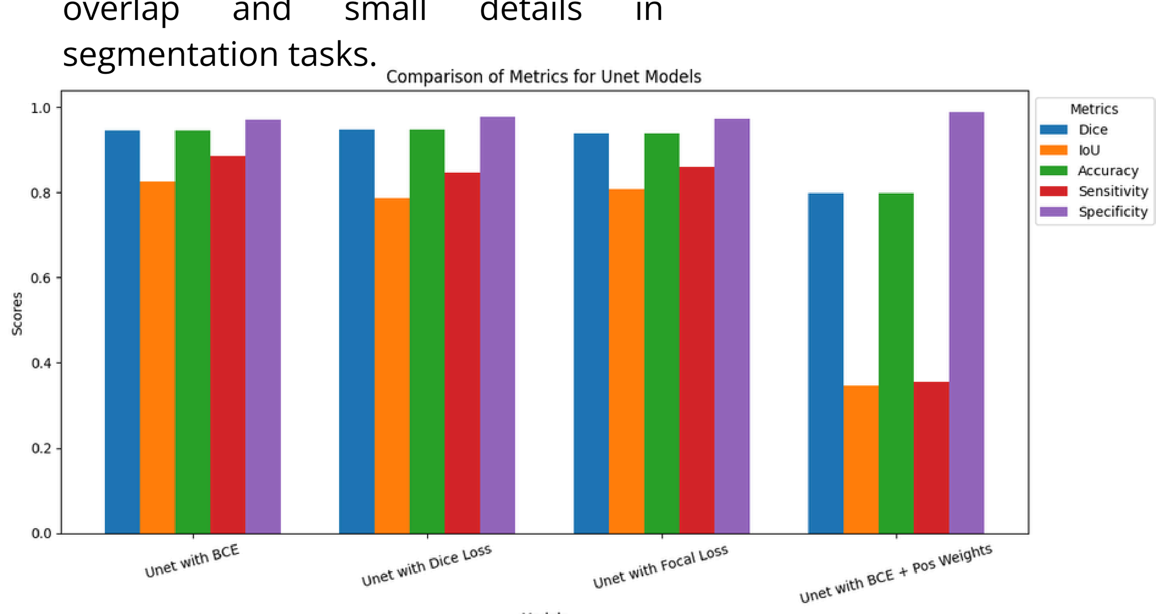
Metric	Strengths	Weaknesses	Retinal Segmentation	Skin Lesion Segmentation
Dice Overlap	Sensitive to small structures; highlights boundary mismatches	Overestimates with large background; less intuitive than IoU	+ Captures narrow vessels - Overestimates on sparse vessels	+ Captures irregular lesion shapes - Inflated results with small lesions
IoU	Intuitive, widely used; good for overlap measurement	Insensitive to boundary errors; underestimates sparse structures	+ Direct performance measure - Misses thin vessel details	+ Clear overview of segmentation quality - Misses minor lesion boundaries
Accuracy	Easy to interpret; general overview	Misleading with class imbalance (large background)	+ Simple overview - Background dominates	+ Useful for varied lesion sizes - Inflated with small lesions
Sensitivity	Good for detecting structures; reduces false negatives	Overly optimistic if false positives allowed. Incomplete without specificity	+ Detects narrow vessels - More false positives	+ Crucial for lesion detection - High false positives on normal skin
Specificity	Measures correct background identification, reduces false positives.	Incomplete without sensitivity for balanced evaluation.	+ Reduces background misclassification - May miss vessels	+ Ensures correct normal skin identification - Misses small lesions

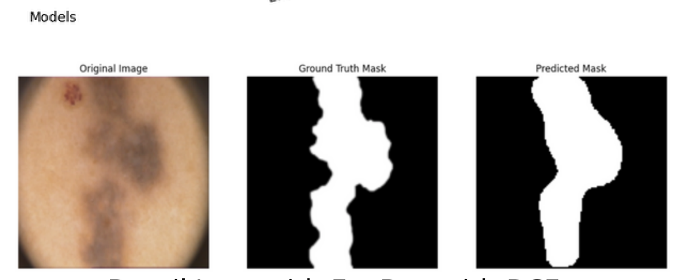
SKIN LESION ABLATION STUDY

We to evaluate the performance of the two models used when using different loss functions.
These comparisons also helped to see if there was any type of improvement on the segmentation or on class imbalance.

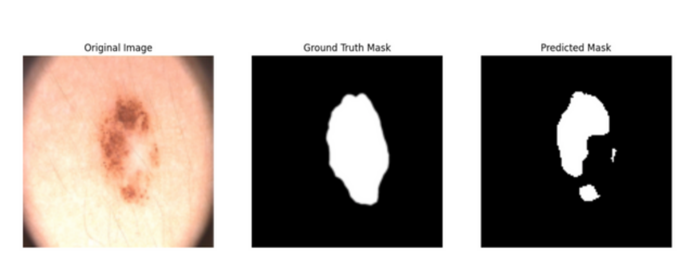
- **BCE:** Struggles with imbalanced data.
- **BCE + Positive Weights:** Better recall for minority classes.
- **Focal Loss:** Focuses on difficult areas.
- **Dice Loss:** Effective capturing region overlap and small details in segmentation tasks.







Detail Loss with EncDec with BCE

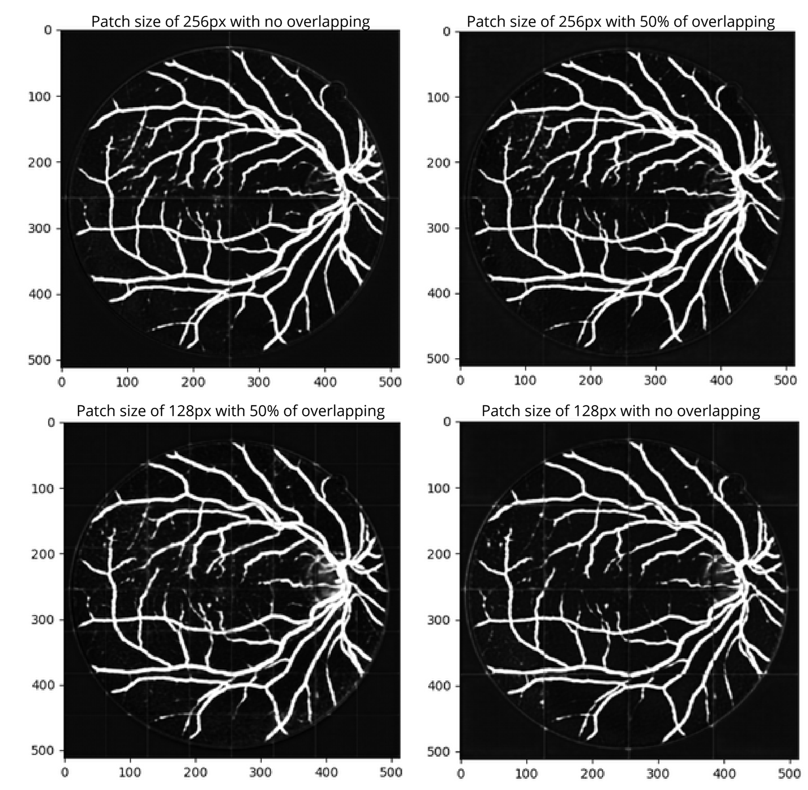


Specificity of UNet with Focal Loss

PATCHES APPROACH

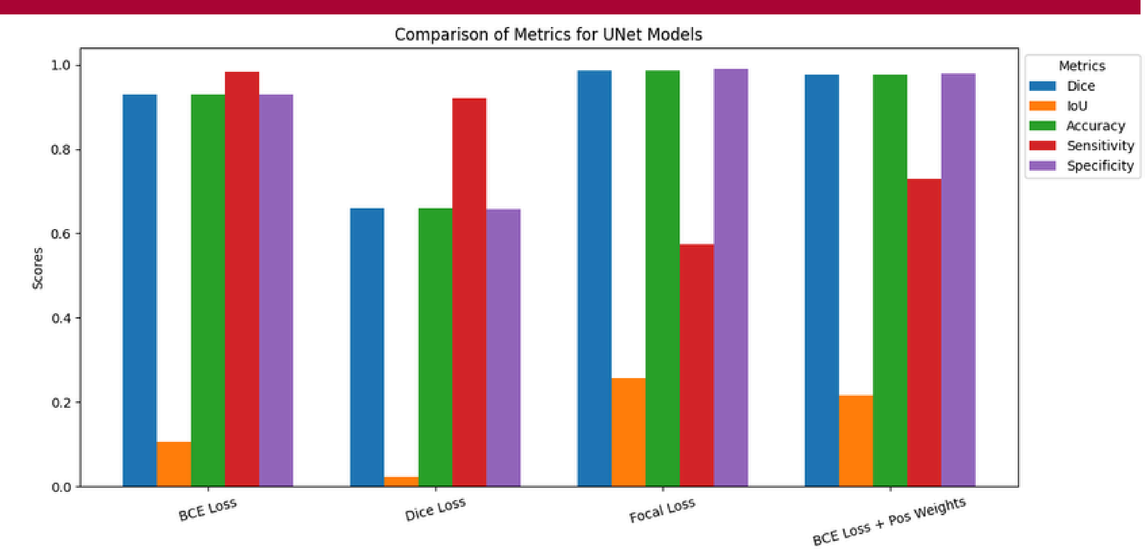
Large sized images are a common problem in deep learning. These high-resolution images need to be downscaled multiple times to be processed, and in the way a lot of information is lost. One of the best solutions to this problem is patch training, this method ensures a faster model and more detailed predictions.

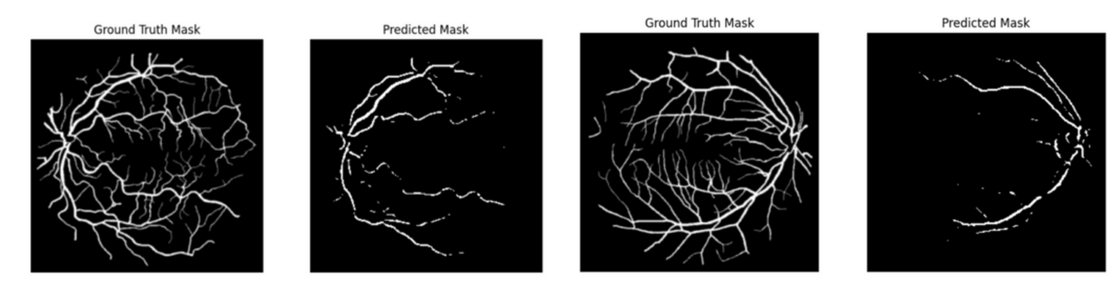
The purpose of this technique is to split the images in different patches, reducing the size of the images we are working with, and in the end, we just patch them back together to get the end result.



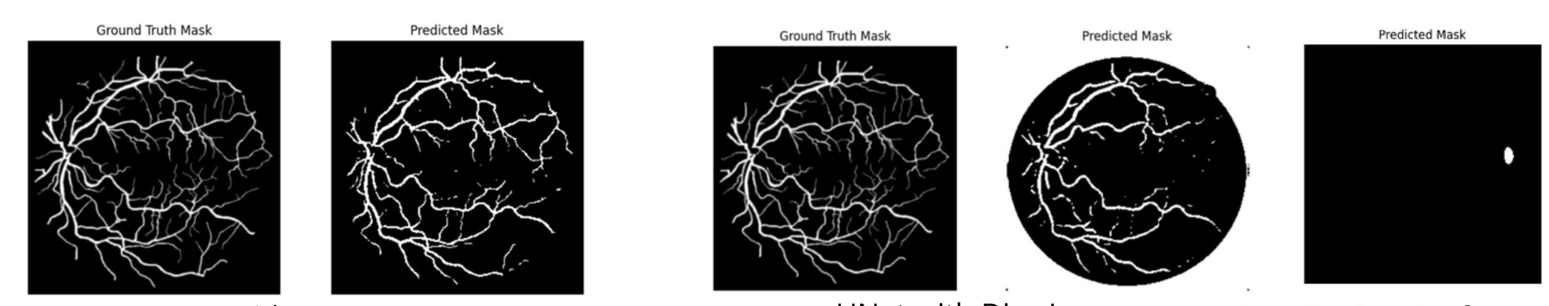
RETINAL ABLATION STUDY

- **BCE Loss** worked best on the Retinal Dataset
- EncDec model was not able to perform segmentation without skip layers for details
- **Dice Loss** metrics less accurately reflect the model's performance due to FOV problems
- **Focal Loss** was expected to work better on harder examples but performed worse, requires more parameter tuning
- **BCE + Positive weights** similarly was expected to help with class imbalance but struggled with details





BCE + Positive Weights
Focal Loss alpha = 0.4, gamma = 2.5

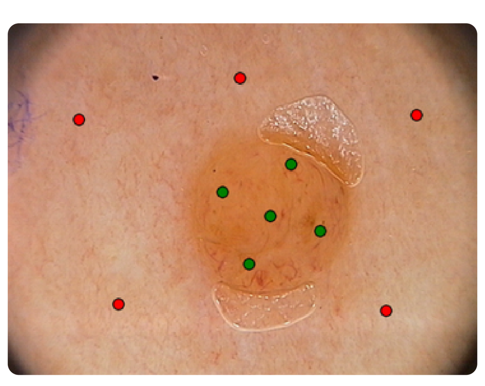


UNet with BCE Loss
UNet with Dice Loss
Best EncDec Performance

WEAKLY SUPERVISED

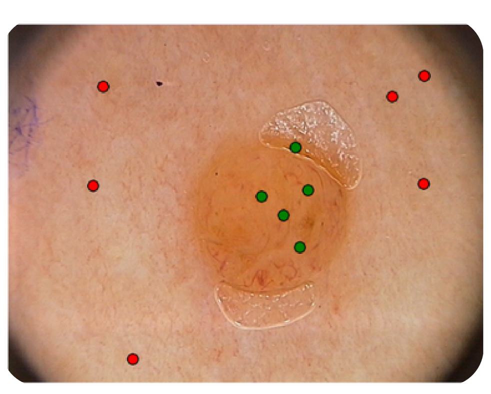
Strategy 1: K-Means Clustering for Targeted Points

- We applied K-means clustering separately to lesion and background regions to select well-distributed positive and negative points.



Strategy 2: Randomized Sampling with Boundary Margin and Variance.

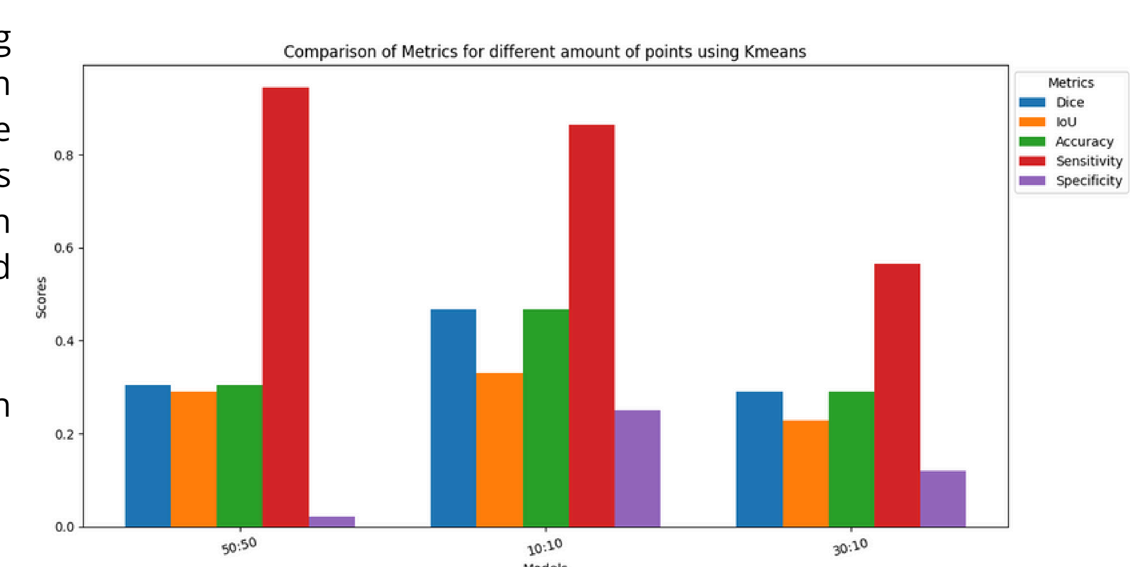
- Points were randomly placed within the lesion and background, with a margin to prevent background points from appearing too close to the lesion boundary.
- We removed point variance as it was not feasible to find widely distributed points in images with large lesion or background areas.

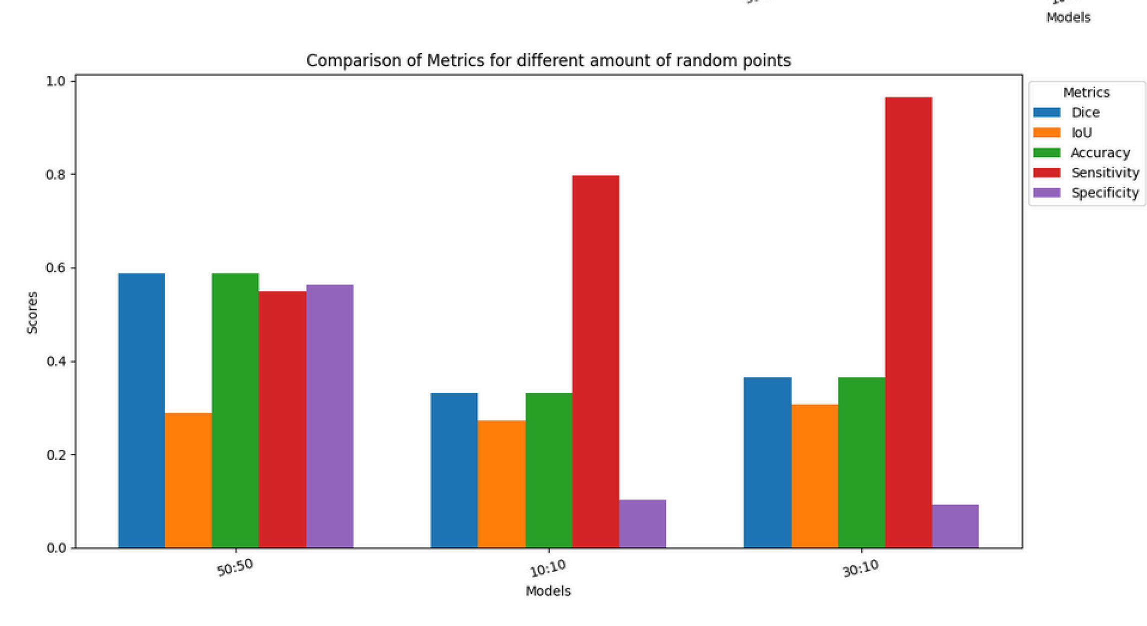


CLICKS COUNT ABLATION

We compared K-means Clustering and Random Sampling for the skin lesion task. We chose to compare positive and negative splits [[10,10], [50,50], [30,10]], for each strategy to see how each method perform.

This was done on the skin lesion dataset.





In the end 50:50 split for Random sampling performed best. But not as well as our fully supervised methods.

References and use of AI:
Bearman, A., Russakovsky, O., Ferrari, V. and Fei-Fei, L. (2015). What's the Point: Semantic Segmentation with Point Supervision. arXiv (Cornell University). doi:https://doi.org/10.48550/arxiv.1506.02106.
Use of AI: Aided in debugging and code generation, explaination of topics.