Technical Report

TEAM: OPTIMA-MUW
Dmitrii Lachinov, Botond Fazekas, Taha Emre, Hana Jebril

Full report download link:
https://drive.google.com/file/d/1fk8gcyyoufwWYWYrrF58ZGT_h74GW-zL/view?usp=sharing

(1) Dataset(s) used for training and model selection;

GOALS 2022 training dataset.
GOALS 2022 validation dataset (for unsupervised training).

(2) Description of the criterion applied to partition the data into training and validation;

In all of the experiments, we used the same cross-validation split. We randomly split the training dataset into 5 groups with stratification by class.
The spits are in the appendix.

(3) A figure with the schematic representation of the proposed method;

Classification:

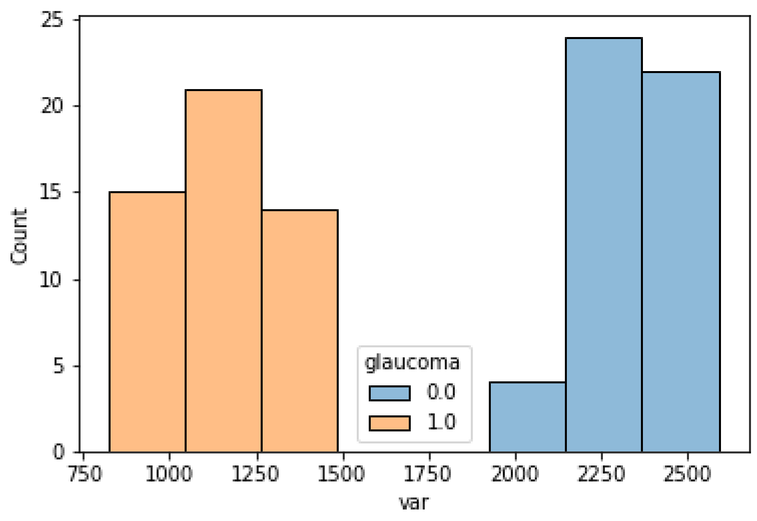
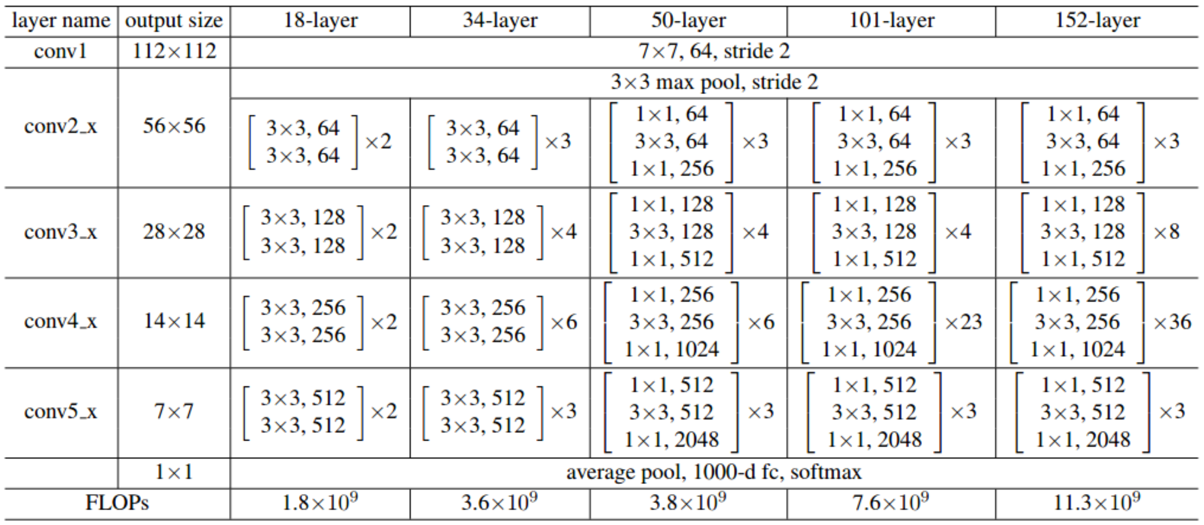


Fig. 1. Histogram of image intensity variance



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 2. ResNet-18 classification network, without any modification, the image is taken from [3].

<u>Segmentation:</u>

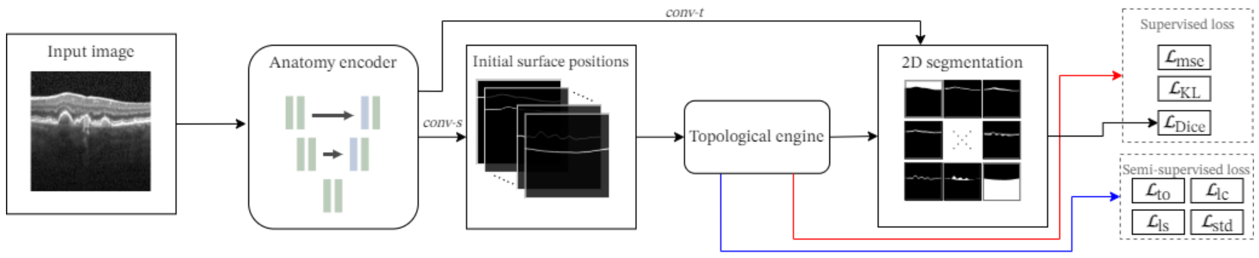


Fig. 3. Architecture of the layer segmentation network, the encoder is either UNet or UNet++.

(4) Input data format, such as Full image or Only the cropped patches?

Classification:
Each B-scan is cropped for 200 pixels from the bottom in order to remove black bands, and resized to 224x224 pixels.

Segmentation:
Vertically and horizontally cropped patches for training, vertically cropped, horizontally full image for inference.

(5) Full description of any preprocessing strategy, including any color mapping, normalization, contrast enhancement, etc.;

Classification:
Each B-scan is cropped for 200 pixels from the bottom in order to remove black bands, and resized to 224x224 pixels. (**same as section 4**) Also we tried zca-normalization to decorrelate images, but we did not notice a significant difference during training.

Segmentation:
We first resized the images to 128x128. Then we trained an SD-LayerNet [1], our submission to MICCAI 2022, to create an initial segmentation of the layers. This initial segmentation was used to get an estimate of the position of the RNFL in each of the scans. The scans were then cropped vertically starting at the IB-RNFL position - 10px and had 416px height. The contrast was adjusted using the MONAI AdjustContrast function with gamma = 1.2 for the inference part on the validation and test set. The image intensity values were normalized to the range [-1; 1] using histogram normalization.

(6) Description of neural network architecture. If a standard network, describe every modification;

Classification:
We use feature-based and deep-learning-based solutions.
One distinctive feature that correctly classifies the presence of Glaucoma in the training and validation images is the variance of image intensities. Thus, we apply the variance threshold if the dataset we infer can be split into two distinct clusters (Fig. 1). Otherwise, we use a ResNet-18. We used a small network since the data and labels are limited. We only changed the input channels of the first convolutional layer to 1 (instead of 3) in order to accommodate gray-scale B-scans.

Segmentation:
We used SD-LayerNet [1] (Figure 3) for the segmentation, although without the reconstruction loss term, and with a dice loss applied on the 2D segmentation. During a training iteration, 3 labeled scans were picked from the training set and 3 unlabeled data were picked from the validation set. During the unsupervised training the correct topological ordering, the surface continuity, and the maximum slope terms were enforced. During the supervised training among the former constraints, the DICE loss was minimized on the 2D pixel-wise segmentation and the Mean Squared Error on the 1D layer position regression. As an anatomy encoder either a U-Net was used with efficientnet-b4 (which had pre-trained weights on the imagenet, as coming from segmentation-models-pytorch), or a U-Net++, which used efficientnet-b4 as well.

(7) Programming language (Python, Matlab, ...) and deep learning framework (PaddlePaddle, Keras, Tensorflow, Pytorch, ...);

Python 3.8, Pytorch 1.11, PytorchLightning 1.6.4, MONAI 0.9.0, segmentation-models-pytorch 0.0.3

(8) Training details, including: optimizer, learning rate, batch size, etc;

Classification:
We used AdamW optimizer with betas 0.9 and 0.95 and weight decay 0.00004. The initial learning rate is set to 4e-6 and decayed using a cosine annealing scheduler. The first 5 epoch is

used as a warm-up period. The network trained for 200 epochs with batch size of 128 using binary cross-entropy loss. The loss is scaled 10-to-1 for the positive class. In order to regularize gradients, we applied gradient norm clipping.

Segmentation:
We used the Ranger [2] optimizer, with default hyper-parameter settings, besides the learning rate. For the learning rate, we used a cosine annealing scheduler with an initial LR of 0.005, which was reduced to 1e-12 in 350 epochs. We trained each of the 5 cross-validation folds for 350 epochs, with a batch size of 6. We applied Stochastic Weight Averaging during the training.

(9) Full description of any data augmentation techniques (transformations, parameters, etc.);

Classification:
For the training set, we used the following random augmentations: translation between 0-5% of the width, random rotation between -10 - +10 degrees, and random horizontal flip with 0.5 probability. For test and validation sets, we did not use any augmentations.

Segmentation:
We randomly adjusted the contrast of the image using the MONAI RandAdjustContrastd method, with a probability of 1.0, and a random gamma between 0.99 and 1.4. During the inference, we used a fixed contrast adjustment with gamma = 1.2.
In order to simulate vessels, we randomly chose 3 vertical bands of random size from 10 to 20 AScans and multiplied them by a random value in the range [0.5,1.5] with a probability of 1.0. Another 3 random bands are summed with a random value in the range [-1,1].
The scans were horizontally flipped with a probability of 0.3.

(10) Full description of any postprocessing of the model outputs or any process to produce the final outputs;

Classification:

We used 5 folds cross-validation in the training step. The best epoch is chosen w.r.t the calculated score from the provided formula. The network achieved the perfect score on the validation, and we did not use any ensemble across the folds.

Segmentation:

During the training, we selected the best-performing models from each of the cross-validation folds. After that, the total score was calculated on the validation set, coming from each of the folds. We trained several models with different configurations and the internal leaderboard was kept with the best performing models. The 6 best performing models were used to create an ensemble for the final submission. For the ensemble creation, the inverse standard deviation of the models' layer output probability map (1/std) was used as a weighting: The higher standard deviation of the prediction (e.g. where the prediction was more uncertain) was weighted less.

(11) [Link to Github repository with the code] is OPTIONAL!

(12) References.

[1] B.Fazekas, A. Guilherme, D. Lachinov, S. Riedl, J. Mai, U. Schmidt-Erfurth, H. Bogunovic SD-LayerNet: Semi-supervised retinal layer segmentation in OCT using disentangled representation with anatomical priors - MICCAI 2022 https://arxiv.org/abs/2207.00458
[2] https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer
[3] He, Kaiming et al. "Deep Residual Learning for Image Recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016): 770-778.

APPENDIX 1. Training data splits

id	split_id
1	0
2	4
3	0
4	3
5	1
6	3
7	4
8	4
9	4
10	2
11	0
12	3
13	2
14	3
15	3
16	1
17	1
18	2
19	1
20	4
21	2

22	2
23	1
24	4
25	1
26	2
27	4
28	3
29	3
30	3
31	1
32	1
33	0
34	4
35	1
36	2
37	4
38	2
39	0
40	4
41	1
42	0
43	0
44	4
45	3

46	4
47	3
48	2
49	2
50	1
51	0
52	0
53	2
54	4
55	3
56	4
57	2
58	4
59	0
60	2
61	0
62	2
63	2
64	3
65	3
66	3
67	0
68	1
69	1

70	1
71	4
72	0
73	3
74	1
75	4
76	3
77	1
78	1
79	4
80	2
81	0
82	1
83	1
84	0
85	4
86	3
87	2
88	3
89	0
90	0
91	2
92	2
93	2

94	4
95	0
96	0
97	0
98	3
99	3
100	1