Supplementary 1. Development of the Deep-Learning System

Multi-task Convolutional Neural Networks

For this study, we designed two versions of a multi-task CNN to process the OCT data obtained from different OCT devices. We built a 3D multi-task CNN for analyzing 3D volume-scans imaged by Cirrus OCT, and a 2D multi-task CNN for analyzing a series of 2D B-scans imaged by Spectralis OCT and Triton OCT.(1) For both 3D and 2D CNNs, we employed the residual network (ResNet) as the backbone. For the 3D CNN, we used a 3D version of ResNet-34 with the last fully connected layer removed as the feature extraction module; we halved the number of feature maps in the original setting to reduce computational costs. For the 2D CNN, we employed ResNet-18 with the last fully connected layer removed as the feature extraction module.

The fully connected layer with softmax activation accepted features from the feature extraction module and output class probabilities in the other two modules. We trained both our 3D network and our 2D network by minimizing the objective function L, as shown below:

$$\mathrm{L}_{DME} = -\sum_{c=1}^{C} y_{DME,c} \log(p_{DME,c} \mid x, \theta),$$

$$\mathrm{L}_{ABN} = -(y_{ABN} \log(p(y_{ABN} = 1 \mid x, \theta)) + (1 - y_{ABN}) \log(p(y_{ABN} = 0 \mid x, \theta))),$$

$$\mathrm{L} = \mathrm{L}_{DME} + \mathrm{L}_{ABN} + \lambda \|W\|_2^2$$

Regarding the equations above, L_{DME} is the cross-entropy loss for the DME classification task, and L_{ABN} is the binary cross-entropy loss for the abnormality classification task. $\theta = 8$ represents the model parameters, x is the input image, and y_{DME} and y_{ABN} are the corresponding labels for DME and abnormality respectively. C is the number of classes (i.e., 3) for the DME classification task. A regularization term W was added to the objective function to prevent the

problem of overfitting. λ controlled the trade-off between the loss terms and the regularization term, with a set value of 3×10^{-5}.

Our CNNs were implemented using ***Keras*** package (https://keras.io/) and Python, working on a workstation equipped with a 3.5 GHz Intel® Core™ i7-5930K CPU and Nvidia GeForce GTX Titan X GPUs. We set the learning rate at 0.0001, and optimized the weights of the networks using the Adam stochastic gradient descent algorithm.

The predictions made by the 3D CNN are at the volume-scan level, whereas those made by the 2D CNN are at the B-scan level. To obtain subsequent "volume-scan" level results for Spectralis OCT and Triton OCT using the 2D CNN, we applied a presence-based strategy: (1) If any B-scans are predicted as CI-DME, the whole scan is classified as CI-DME; (2) if (1) does not hold and at least one B-scan is predicted as non-CI-DME, the whole scan is classified as non-CI-DME; (3) if both (1) and (2) do not hold, the whole scan is classified as non-DME.

Data pre-processing and data augmentation

For the 3D volume-scans obtained from Cirrus OCT, we first applied Otsu's method,(2) a traditional automatic image thresholding algorithm, to obtain the region of interest (ROI) in the volumetric data (i.e., the retinal area). Applying Otsu's method caused the network to focus more on learning discriminative patterns without disturbances from other irrelevant areas, dramatically reducing the computational cost. We set the input size of our 3D DL system to 128 × 512 × 512, applying random cropping or zero padding to the ROIs during the pre-processing phase.

For the 2D OCT B-scans obtained from Spectralis OCT and Triton OCT, we set the input sizes of our 2D CNN system to 496 × 496 × 3 and 992 × 1024 × 3, respectively. We also applied

resizing to the images obtained from Spectralis OCT, given that the device used two scanning protocols.

We further applied standardization and normalization on all input data (both 3D volume-scans and 2D B-scans). Specifically, we standardized the input data to have zero mean and unit variance, and then normalized them to the range [0, 1],. In deep-learning scenario, it is very common to randomly split the dataset into training, testing and validation sets when the dataset is large since the overfitting problem can be prevented with adequate training samples. In this study, as shown in **Table 1** we constructed three large-scale OCT datasets (i.e., with 3,788 volumes in Cirrus OCT dataset, 30,515 B-scans in Spectralis OCT dataset, and 39,443 B-scans in Triton OCT dataset, respectively). We divided the primary dataset of the 3 OCT devices at random for training (60%), testing (20%) and primary validation (20%). It is worth mentioning that the random sampling was performed at patient level in order to prevent leakage and biased estimation of the performance on the primary validation sets.

During the training phase, we used data augmentation techniques (i.e., random flipping at all axes) to enrich the training samples, which alleviated the overfitting problem.

Heatmap Generation

We used the established Class Activation Map (CAM)(3) technique to generate heatmaps, highlighting the discriminative object parts detected by the CNN model. The CAM of a particular class implies the informative image regions used by the CNN to identify that class. We backward-projected the weights of the output layer (predicted class scores) on to the convolutional feature maps ahead of the global average pooling layer. Each value at the spatial location denotes the unit of the presence of a specific visual pattern. Therefore, the CAM is

exactly the weighted linear sum of these visual patterns at different spatial locations. As the CAM had a smaller size than that of the input image due to the convolution and pooling operations, we up-sampled the CAM to the original input size. In this manner, we could readily identify the image regions most relevant to the particular class.

For better visualization, we used the Python library Matplotlib(4) to plot the CAM overlapping the corresponding original input B-scans by adjusting the transparency. In particular, for the OCT volume-scan imaged by the Cirrus device, we first produced the heatmaps of the B-scan images, then utilized Fiji—a popular image processing package bundling a range of plugins(5)—to ensemble B-scan heatmaps from the same volume into a 3D heatmap.

References:

1. Wang X, Tang F, Chen H, Luo L, Tang Z, Ran AR, Cheung CY, Heng PA: UD-MIL: Uncertainty-driven Deep Multiple Instance Learning for OCT Image Classification. IEEE journal of biomedical and health informatics 2020;

2. Otsu N: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics 1979;9:62-66

3. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. p. 2921-2929

4. Hunter JD: Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 2007;9:90-95

5. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY, White DJ, Hartenstein V, Eliceiri K, Tomancak P,

Cardona A: Fiji: an open-source platform for biological-image analysis. Nature methods 2012;9:676-682