Supplementary Table 2. Model training configuration.

Parameter	Description
batch_size: 4	The number of samples (images) in a single batch. One batch is processed in one training iteration.
iters: 250000	The total number of training iterations. In each iteration, the model is updated based on one batch of data.
model: type: SegNeXt **backbone:** type: MSCAN_L	Parameters defining the segmentation model architecture. SegNeXt is the type of model used for image segmentation. The MSCAN_L backbone is a neural network responsible for feature extraction from the input image.
num_classes: 9	The number of classes in the segmentation task.
decoder_cfg: channels: 256 ham_channels: 256 ham_kwargs: MD_R: 16 dropout_rate: 0.1 align_corners: False	Parameters configuring the model decoder, responsible for generating the final prediction maps. channels is the number of channels in the decoder layers. ham_channels refers to the number of channels in the HAM (Hierarchical Attention Mechanism) module. MD_R is the reduction ratio in the HAM module, controlling the level of feature compression. dropout_rate is the dropout coefficient, which helps prevent overfitting. align_corners is a parameter controlling whether the image edges are aligned during interpolation (e.g., during resizing).
transforms: - type: Resize target_size: [890, 1024] - type: RandomHorizontalFlip - type: RandomDistort - type: Normalize mean: [0.485, 0.456, 0.406] std: [0.229, 0.224, 0.225]	A list of transformations applied to the data before being fed into the model. First, resizing the images. Second, randomly flipping the images horizontally. Third, randomly distorting the images. Finally, normalizing the images based on the mean and standard deviation for the RGB channels.

optimizer: type: AdamW weight_decay: 0.01 custom_cfg: - name: head lr_mult: 10.0 - name: norm weight_decay_mult: 0.0	Parameters configuring the optimizer used for model training. AdamW is the type of optimizer, an extended version of Adam with regularization weights based on weight_decay. Additional parameters are set for specific model layers. The head layers have a different learning rate, while the norm layers have the regularization coefficient (weight_decay_mult) set to zero.
lr_scheduler: type: PolynomialDecay warmup_iters: 1500 warmup_start_lr: 1.0e-6 learning_rate: 0.00006 end_lr: 0.0 power: 1.0	Parameters configuring the learning rate schedule during training. PolynomialDecay is the type of schedule, which decreases the learning rate according to a polynomial decay function. warmup_iters is the number of initial iterations (warmup) during which the learning rate gradually increases from a very small value to the target value. warmup_start_lr is the initial, very low learning rate used during the warmup phase. learning_rate is the rate at which the model is trained after warmup. end_lr is the final learning rate towards which the schedule converges. power controls how fast the learning rate decreases (a value of 1.0 means linear decay).
loss: types: - type: CrossEntropyLoss coef: [1]	Parameters defining the loss function. In this case, a single loss function is used (based on cross-entropy) with a weight of 1.