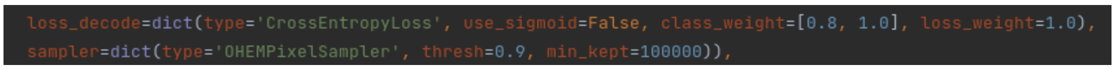
1. Overall process

All the tagged data of the rematch are used to divide all images into 50% (40% training +1% verification). The MMSegmentation framework is used to train separately, and the best results are divided. The training data of the fold is rotated 180° and the rotation picture and label mask image are saved, and the training is retrained. Finally, post-processing is performed in the inference stage, and the final result is converted into a binary graph for submission.

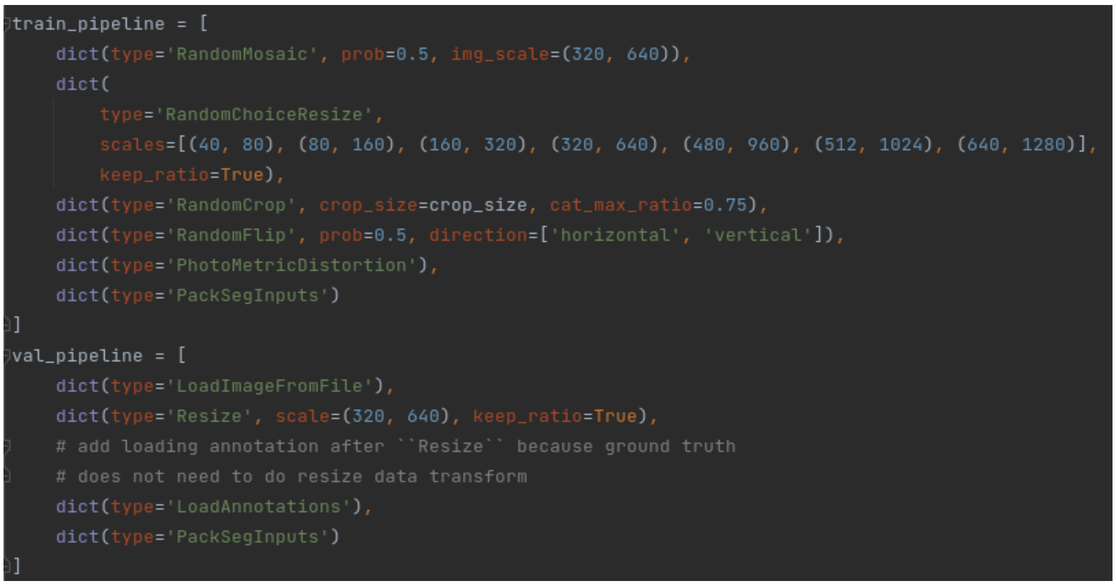
2. Details of deployment experiments

Model: Encoder-Decoder Based Models, the backbone part uses ConvNeXt-xlarge (loads the pre-training model provided by MMClassification), the decode_head part adopts the UperHead structure (from UperNet), the auxiliary_head part adopts FCNHead, and the decode_head and auxiliary_head parts use Class Balanced Loss and Online Hard Example Mining (OHEM). The code is as follows:

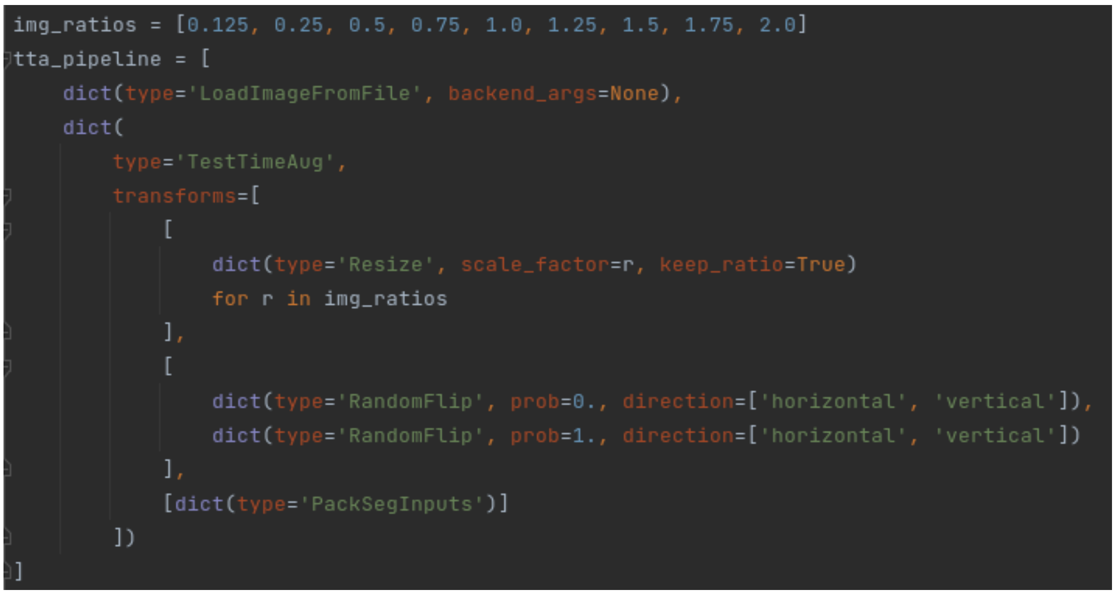


```
loss_decode=dict(type='CrossEntropyLoss', use_sigmoid=False, class_weight=[0.8, 1.0], loss_weight=1.0),
sampler=dict(type='OHEMPixelSampler', thresh=0.9, min_kept=100000)),
```

Data augmentation and training strategies: Training set data augmentation uses random Mosaic data augmentation (executed with probability 0.5, generated image size is 320*640), random scale scaling (maintaining aspect ratio scaling as one of 40*80, 80*160, 160*320, 320*640, 480*960, 512*1024, 640*1280), cropping (executed with probability 1, 320*640), random vertical and horizontal flips (executed with probability 0.5), PhotoMetricDistortion (applying photometric distortion to the image in order, with the probability of applying each transformation being 0.5). The verification set data enhancement uses scaling (320*640), and the test set inference is used to test data enhancement (tta). The enhancement method is to first scale the pictures with the scale of [0.125, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0], and then perform (unchanged, vertical flip, horizontal flip) on each picture, aggregate the inference results, and finally aggregate the inference results of all pictures. The code is as follows:



```
train_pipeline = [
    dict(type='RandomMosaic', prob=0.5, img_scale=(320, 640)),
    dict(
        type='RandomChoiceResize',
        scales=[(40, 80), (80, 160), (160, 320), (320, 640), (480, 960), (512, 1024), (640, 1280)],
        keep_ratio=True),
    dict(type='RandomCrop', crop_size=crop_size, cat_max_ratio=0.75),
    dict(type='RandomFlip', prob=0.5, direction=['horizontal', 'vertical']),
    dict(type='PhotoMetricDistortion'),
    dict(type='PackSegInputs')
]
val_pipeline = [
    dict(type='LoadImageFromFile'),
    dict(type='Resize', scale=(320, 640), keep_ratio=True),
    # add loading annotation after ``Resize`` because ground truth
    # does not need to do resize data transform
    dict(type='LoadAnnotations'),
    dict(type='PackSegInputs')
]
```



```
img_ratios = [0.125, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0]
tta_pipeline = [
    dict(type='LoadImageFromFile', backend_args=None),
    dict(
        type='TestTimeAug',
        transforms=[
            [
                dict(type='Resize', scale_factor=r, keep_ratio=True)
                for r in img_ratios
            ],
            [
                dict(type='RandomFlip', prob=0., direction=['horizontal', 'vertical']),
                dict(type='RandomFlip', prob=1., direction=['horizontal', 'vertical'])
            ],
            [dict(type='PackSegInputs')]
        ])
]
```

The preset training iteration is 80,000, the first 1,500 iterations adopt linear learni ng rate, and the subsequent polynomial attenuation learning rate (PolyLR). The ea rly stop strategy is used to save the model with the highest average Dice on the ve rification set.

Post-processing: During inference, each picture will generate a probability matrix of inference results, with a size of (2, 320, 640), that is, each pixel point has two pr obabilities (it is the tooth area/not the tooth area). For the probability matrix of eac h map, all pixel points are traversed. If the maximum probability of two probability of a point is less than 0.6 (meaning that the model predicts the point inaccurately b ecause the probability of not the tooth area =1-the probability of the tooth area, whi ch means that the model is close to the two predictions). Therefore, for conservativ e reasons, change the pixel value of this point to 0 (background), and the remainin g points are assigned a value of 255 (tooth).

3. Results display

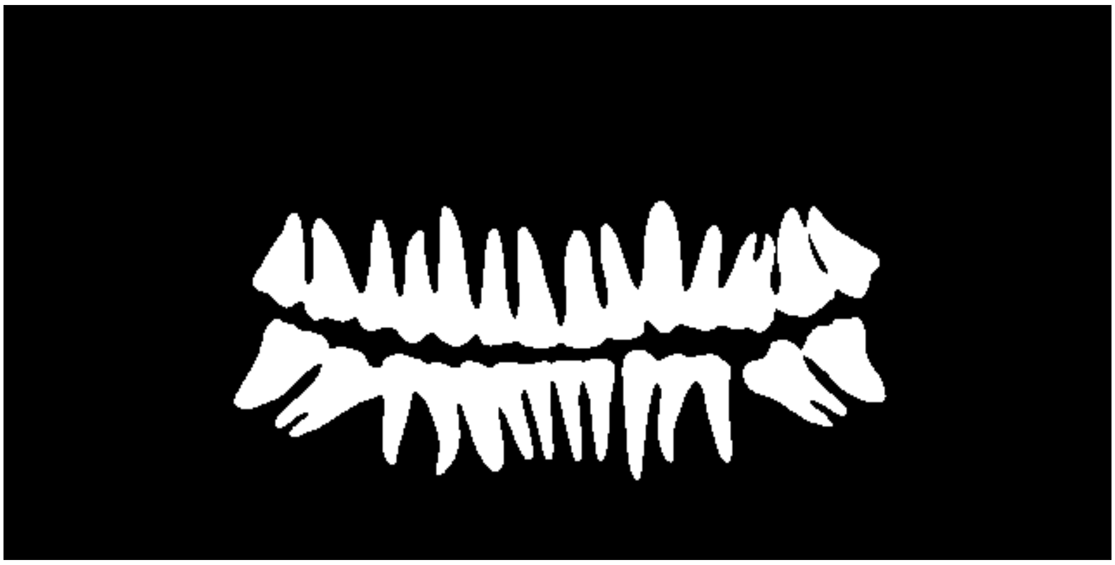


Figure 1. test_0.png

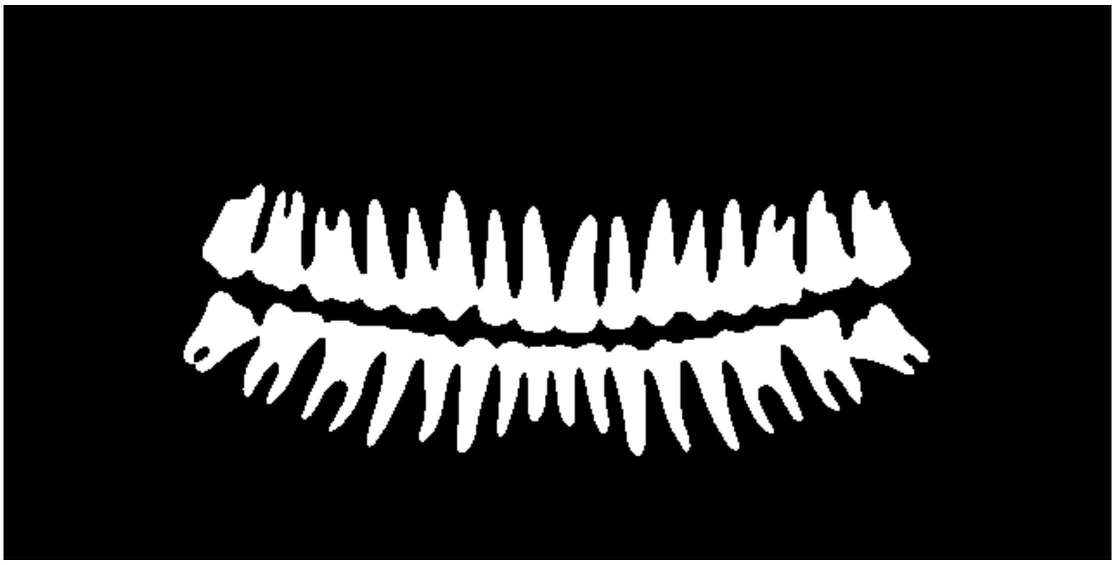
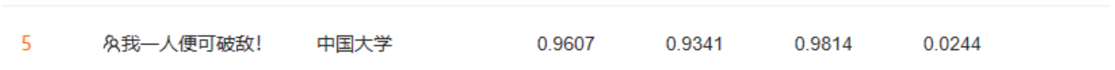


Figure 2. test_1.png

Evaluation indicator results:



5	我一人便可破敌！	中国大学	0.9607	0.9341	0.9814	0.0244