9th place solution

posted in Intel & MobileODT Cervical Cancer Screening 8 months ago

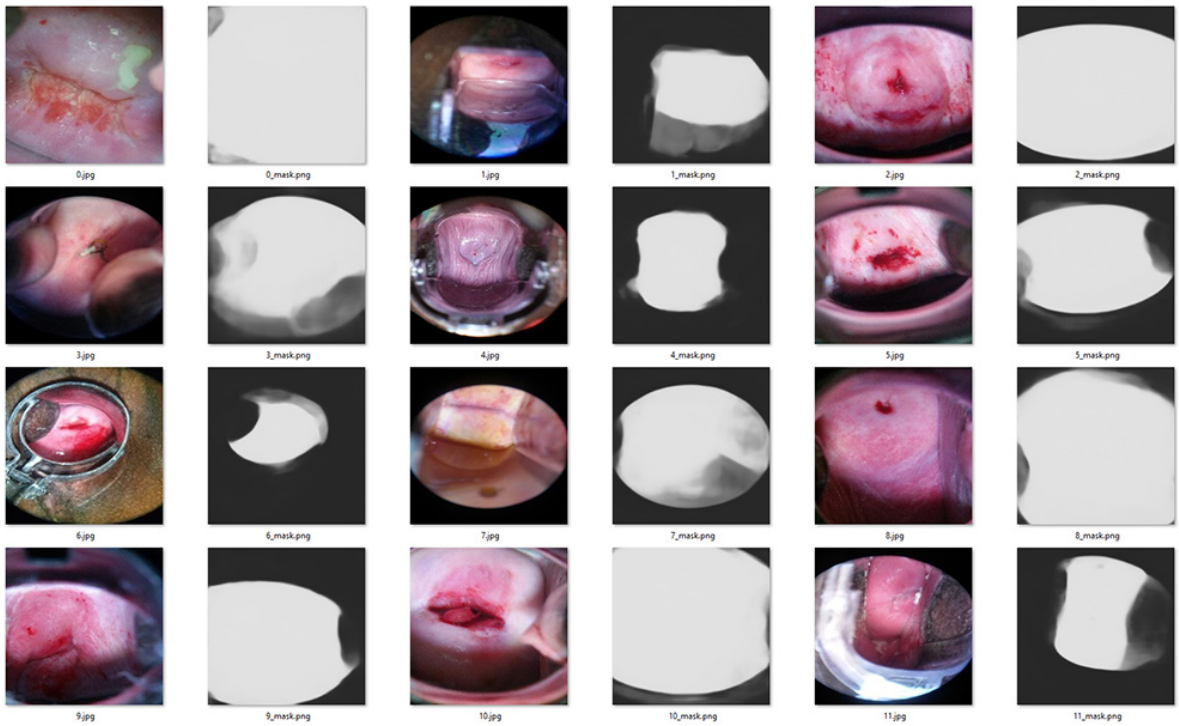
30

Used tools

Windows 10 + Python 3.4 + Keras 1.2 + Theano 0.9

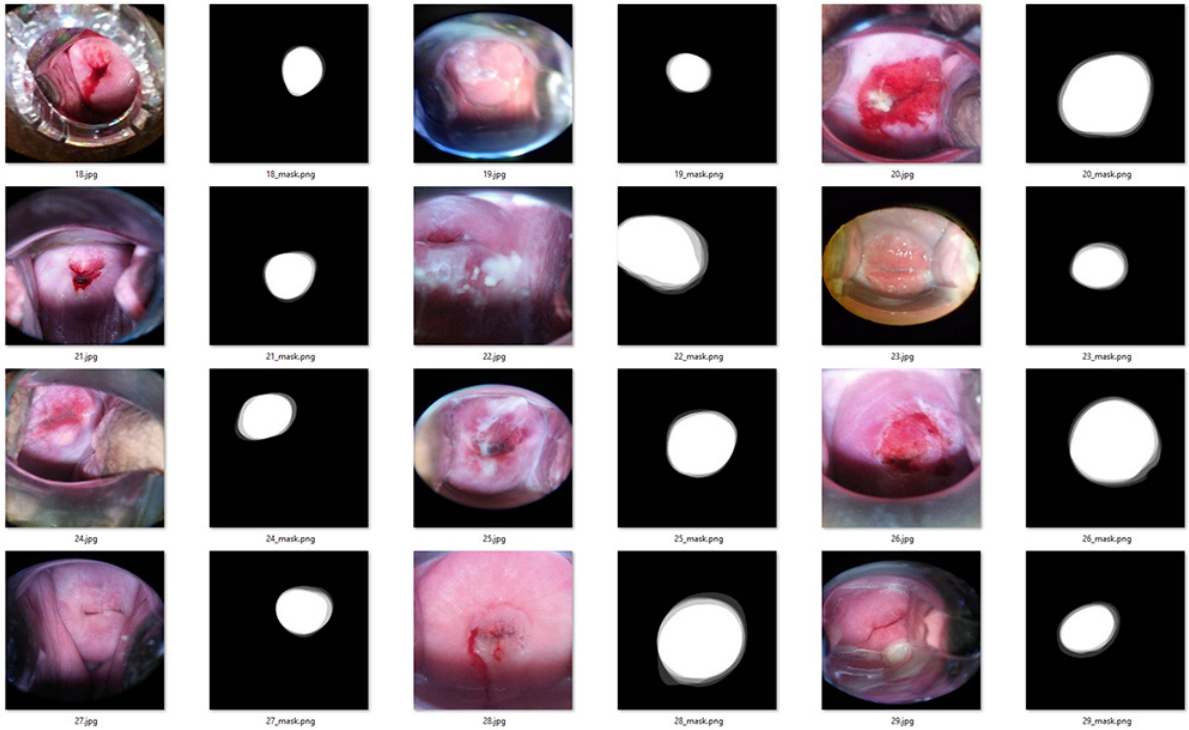
ZF_UNET_224 (version 1)

To find important parts of images I used my own version of UNET. First UNET was trained on segmentation of train images, which I made by hands with Sloth. In this segmentation I tried to cut all elements which are totally useless like speculum and everything outside it. The predictions look nice:



ZF_UNET_224 (version 2)

Second UNET was trained on rectangles provided by Paul. This net finds the main region of interest. The predictions look even better.



I didn't use these predictions directly. I only extract bounding box rectangles for each image to form rectangles.csv file which I used in later training process.

Zoo

I used the following set of CNNs: VGG16, VGG19, RESNET50, INCEPTION_V3, SQUEEZE_NET, DENSENET_161, DENSENET_121. The training process for them was mostly similar. 5 KFold validation, vary only learning rate parameters and batch size. For each model I obtain: train OOF predictions with same length as number of training images and test predictions.

Augmentations

I think augmentation was important key for this problem, since we had very small amount of data. I used:

1. Random crops based on rectangle.csv generated from UNETs. These crops were in very big range from UNET_v1 prediction to UNET_v2 predictions.
2. Random perspective transformation
3. Random rotations (mean color or border reflect at random) and mirroring
4. Lightning change
5. Rare random blur

Basically neural nets never see the same pictures during training.

Batch generator

For learning process I used fit_generator function from Keras, which I recommend to use for everyone. You only need to create your own batch_generator function. No need to store many different images in memory or on HDD. In my batch generator I add some fraction of images from "additional" folder ~25%. It greatly improves validation and leaderboard score. And we all know now it's because of leakage of test images in additional folder.

Ensemble

For ensemble I used XGBoost blender. Final solution was average on 500 XGBoost iterations with different seed and random run parameters.

Submission

The only difference for my 2 final submissions was to train with usage of "additional" images or without them. My final models used "train" and "test_stg1" images for training.

Submission v1 (with additional): Validation score: 0.57457 Private LB: 0.88856

Submission v2 (without additional): Validation score: 0.64254 Private LB: 0.83209

I believe low scores on private LB depends on bad data preparation, mostly because of mislabeling. We could clean it by hands, but in case private test set had the same labeling quality it would make everything worse.

Code

You can find my code as it was prepared for Kaggle on GitHub

polygons.json (765.26 KB)
ZF_UNET_v1_predictions.jpg (169.94 KB)
ZF_UNET_v2_predictions.jpg (161.75 KB)

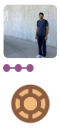
utility • (12th in this Competition) • 8 months ago • Options • Reply ▲ 3 ▼

How important was the usage of bounding boxes? In my experiments i was not able to get any significant improvements from this information. In the end, I just fine-tuned resnet and densenet=)

ZFTurbo • (9th in this Competition) • 8 months ago • Options • Reply ▲ 1 ▼

With bounding boxes training process was more stable for me, without big jumps and lower speed of overfitting. It's because with boxes I was able to make larger crops without affect of area of interest. In the beginning I used just 10% random crops from all sides of image.

Bounding boxes gave me improvements on stage 1 LB, but I don't remember exact value.



entanglement • (26th in this Competition) • 8 months ago • Options • Reply ▲ 1 ▼

@utility Yeah, it seems so that bounding boxes weren't much important. The only thing I did was to take a center crop assuming the cervix to be more or less at the center due to the MobileODT tool which has got a cylindrical tube that enters through the vaginal opening. And I was able to train quite stably on green-channel-replicated-thrice images.

ZadrraS • (1st in this Competition) • 8 months ago • Options • Reply ▲ 1 ▼

My experience was the opposite - for some reason I had a really hard time training anything on crops. It wrecked test time performance every time, even when I was sure I wasn't cropping out anything of relevance.

kubilai • (6th in this Competition) • 8 months ago • Options • Reply ▲ 0 ▼

bounding boxes gained about 0.1 for me on the stage 1 LB

eagle4 • 8 months ago • Options • Reply ▲ 1 ▼

ZFTurbo, Congrats once again! I have a question about the xgb blender: what are the inputs for the model?

ZFTurbo • (9th in this Competition) • 8 months ago • Options • Reply ▲ 3 ▼

The input for XGBoost blender are predictions on validation set from each neural net model. So neural net predictions used as features. Validation set has the same length as train since it combined from 5 Folds.

Scotty • (33rd in this Competition) • 7 months ago • Options • Reply ▲ 0 ▼

Congrats and thanks for sharing!
Just curious, does XGBoost blender perform better than a simple average blender in your case?

ZFTurbo • (9th in this Competition) • 7 months ago • Options • Reply ▲ 1 ▼

I stopped comparing in the early beginning of contest. In my first attempts XGBoost blender was better.