Image Processing and Computer Vision (CSC6051/MDS6004) Assignment 1

Zhiyou Zhao
222041044
222041044@link.cuhk.edu.cn

Abstract

In this assignment, all three tasks are accomplished successfully. For task 3 especially, a well-structured, comprehensive yet flexible codebase is build from scratch and accessible on my github ***repository***[1]*. Training process of experiments are presented on the public* ***Kaggle Notebooks***[2] *for reference.*

In this report, I will present my experiment results conclusions in clear and precise way. And in the Supplementary materials, the footprints of my work is entailed.

1. Task 1

Implementation Utilizing two library functions:

- `cv2.getRotationMatrix2D`
- `cv2.warpAffine`

Results Following is code output:

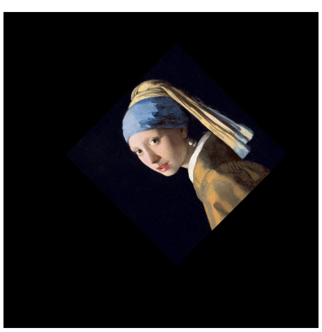


Figure 1: $s = 0.5, \theta = 45, t_x = 50, t_y = -50$

2. Task 2

Implementation Divided into two main steps:

- `get_camera_intrinsics`
- `project_cube`

The former computes the projection matrix based on provided geometrical information and the latter performs the projection based on the matrix.

Results Following is the code output:

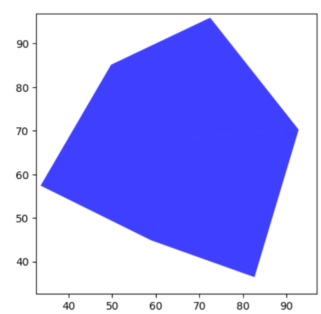


Figure 2: Same parameters as the provided example

3. Task 3

3.1. Implementation

Data pipeline Considering the large dataset size and simplicity of the task, no augmentation is performed. The only pipelins applied are `Resize` and `Normalize`. For masks, the data format of three datasets are quite different, so **three custom mask loading pipelines** are implemented to correctly obtain the desired mask format.

Model The overall PortraitNet structure is adopted. For the backbone network, I used **ResNet34** and **ResNet50** for easy ablation. For the decoder Network, the **U-Net** like **Upsampling** and **Skip-connections** are preserved, but replacing the D-block by Conv2d for better performance since the need to run on mobile devices is stripped.

[1]Code available at: https://github.com/zhiyozhao/tmp
[2]Kaggle Notebooks: MattingHuman, MattingHuman-S, EasyPortrait

Loss For simplicity, the model supports $n\_classes$ parameter, where we set it to 1 for binary segmentation and to C for multi-class segmentation. So the losses used are `nn.BCEWithLogitsLoss` and `nn.CrossEntropyLoss`.

Metrics The **IoU** and **mIoU** metrics are implemented, essentially the same thing, they are implemented separately to acommondate the two types of segmentation tasks involved. As for the computation, since all input are binary masks, the `logical_and` and `logical_or` are used to compute intersection and union.

Traing/Testing Loop A simplist training/validating script is implemented, including extra functionalities: load_config, logging, tensorboard, lr_schedule, best_ckpt_save. As result, all training tasks in the assignment can be performed simply by chaning configs. The results will be presented in following sections.

Experiment Setup All experiments use `CosAnealingLRScheduler` with inital $lr = 0.001$, and same batchsize 48, except for EG1800 with batchsize 64. While size of different datasets varies, I set different epochs for them so that optimization steps remains similar.

3.2. Results

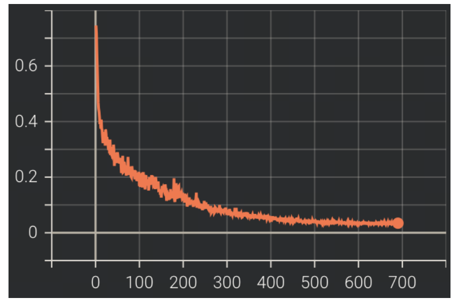
Overview All experiments and evaluation results on corresponding test set are summarized in Tab.1.

Method	Metric
EG1800 + ResNet50	93.5% IoU
MattingHuman + ResNet50	97.9% IoU
MattingHuman + ResNet34	98.0% IoU
EasyPortrait + ResNet50	72.2% mIoU

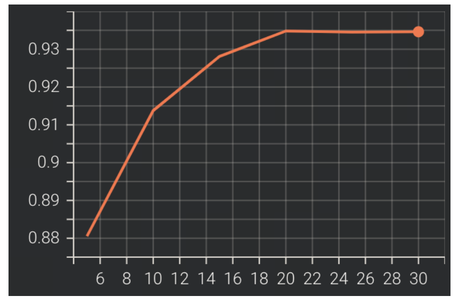
Table 1: Evaluation results on corresponding testing sets

EG1800 + ResNet50 This is a small dataset (1G), and is trained on my local machine for 30 epochs. The training curve and testing IoU is shown in Fig.3. We can see achieve test IoU of **0.935%** without any tuning.

MattingHuman + ResNet50 The MattingHuman dataset is huge, about 20 times larger than the original EG1800 dataset. Due to resource limit, I only used half of it for training and testing (about 1.4w training images). The training process and testing result is shown in Fig.4. Thanks to larger training set, an IoU of 97.9% can be easily achived within 3 hours of training.

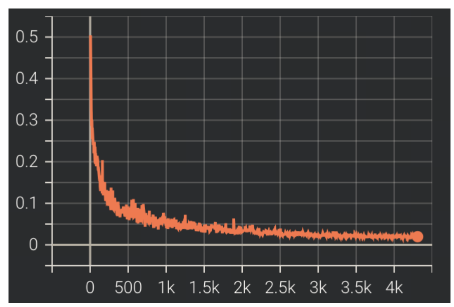


(a) Traning Loss



(b) Testing IoU

Figure 3: Results on EG1800

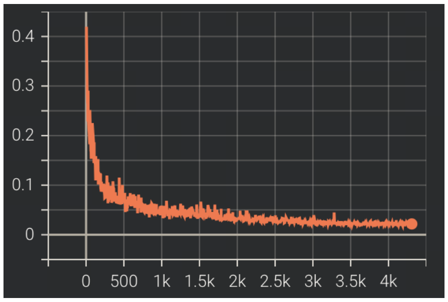


(a) Traning Loss

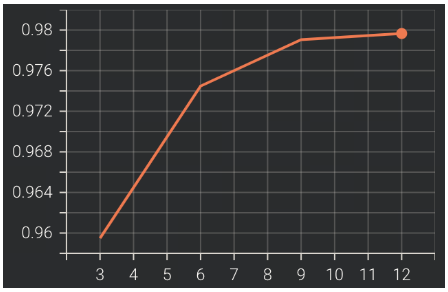
(b) Testing IoU

Figure 4: Results on MattingHuman

MattingHuman + ResNet34 The purpose of this experiment is to investigate the impact of model size on large datasets. Results are shown in Fig.5. Surprisingly, no considerable performance are observed compared to the Resnet50 backbone. More on this later.

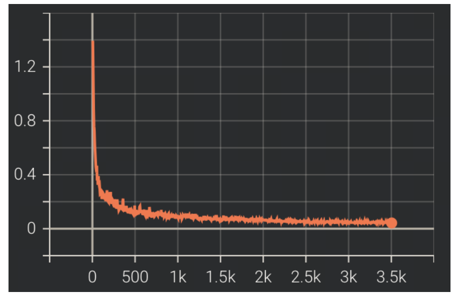


(a) Traning Loss

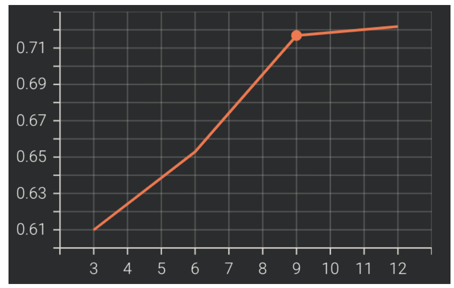


(b) Testing IoU

Figure 5: Results on MattingHuman (ResNet34)



(a) Traning Loss



(b) Testing IoU

Figure 6: Results on EasyPortrait

EasyPortrait + ResNet50 Different from previous binary segmentation tasks, this task is doing multi-class segmentation. So the loss and metrics used are changed accordingly. The training results are shown in Fig.6. One notable thins is that, the mIoU, which is the mean IoU on all classes is much lower compared to IoU metric on other experiments. After 12 epochs, we only achieved 72.2% mIoU. What's worse, from the curves, we can see the model is converging. More on this later.

3.3. Comparison & Analysis

Different Model Performance on EG1800 Models with different size and training data are tested on the EG1800 dataset for comparison, the results are shown in Tab.2. We can conclude that:

Order	Method	Metric
1	EG1800 + ResNet50	93.52% IoU
2	MattingHuman + ResNet50	95.90% IoU
3	MattingHuman + ResNet34	95.85% IoU

Table 2: Evaluation results on EG1800 testing set

- **Larger training dataset bring an considerable performance improvement** on IoU and much lower training loss even when tesing on cross domain dataset.
- On the large MattingHuman dataset, ResNet34 and ResNet50 backbone achieve similar performance. Maybe it can be attributed to my adoption of optimized segmentation architectures. However, a more realistic reason is that **this segmentation task with 224 input already can be handled by ResNet34 easily**, not mentioning ResNet50.

Different Model Visualization on Real-life Images I pick up three images from **XiaoHongShu** based on my personal preference and the magic of recommendation system. The visulization results from two different models trained on EG1800 and MattingHuman are shown in Fig.7 and Fig.8. Some interesting observations:

- Desplite the high IoU score on testing set, the random-choice testing results are not good at all. **The generalization ability is poor**
- The edges of the segmentation results are generally non satisfactory. **BCELoss is not enough to learn sharp and consistent edges**, some edge optimization like the original PortraitNet did is necessary.
- **Models trained on the 10x larger MattingHuman dataset are generally better**, especially on **Pengyuyan**. And that is important.

4. Conclusion

Well done homework!!! Congrats!!!

(a) Look Nice

(b) The elevator girl???

(c) Something grows on his head:-(

Figure 7: Results from EG1800 + ResNet50

(a) Slightly better

(b) The elevator girl???

(c) Right this time:-)

Figure 8: Results from MattingHuman + ResNet50