IIAI30013

Computer Vision

HW4: Segmentation

Instructor: YuanFu Yang

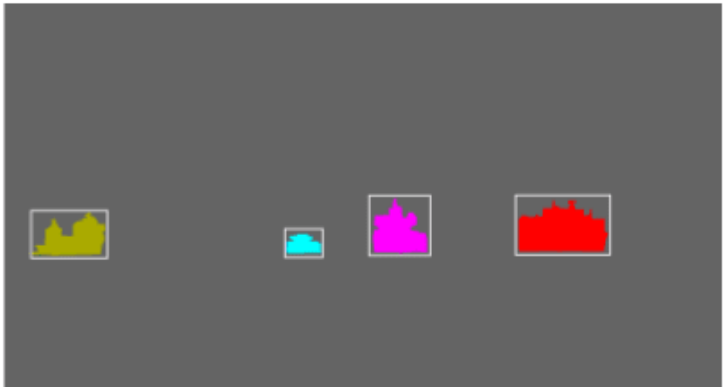
yfyangd@nycu.edu.tw

HW4: Segmentation

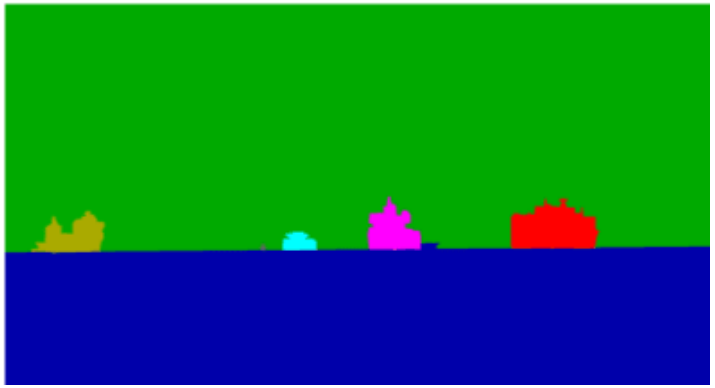
Image

Semantic Segmentation

Instance Segmentation



Panoptic Segmentation



HW4: Segmentation

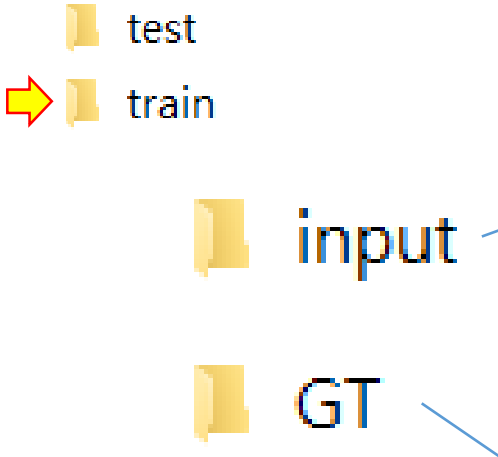
- Our homework this time is **Semantic Segmentation**:
  - Component Segmentation of Sofa

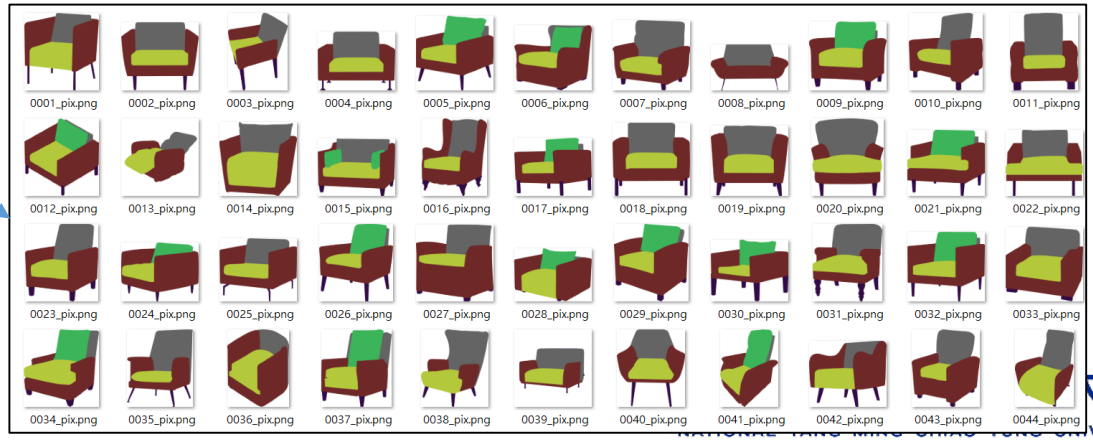
Input

Output

HW4: Segmentation

- Training Set:
  - 400 images with GT.





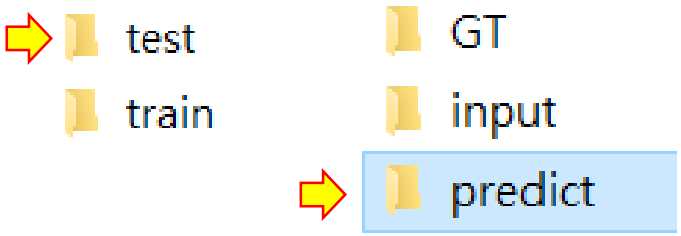
HW4: Segmentation

- Test Set:
  - 10 images with GT.

Please refrain from using ground truth data for training as it constitutes **cheating.**

HW4: Segmentation

- Submission:
  - Submit 10 segmentation predictions generated by your trained model and save them to the "predict" folder.

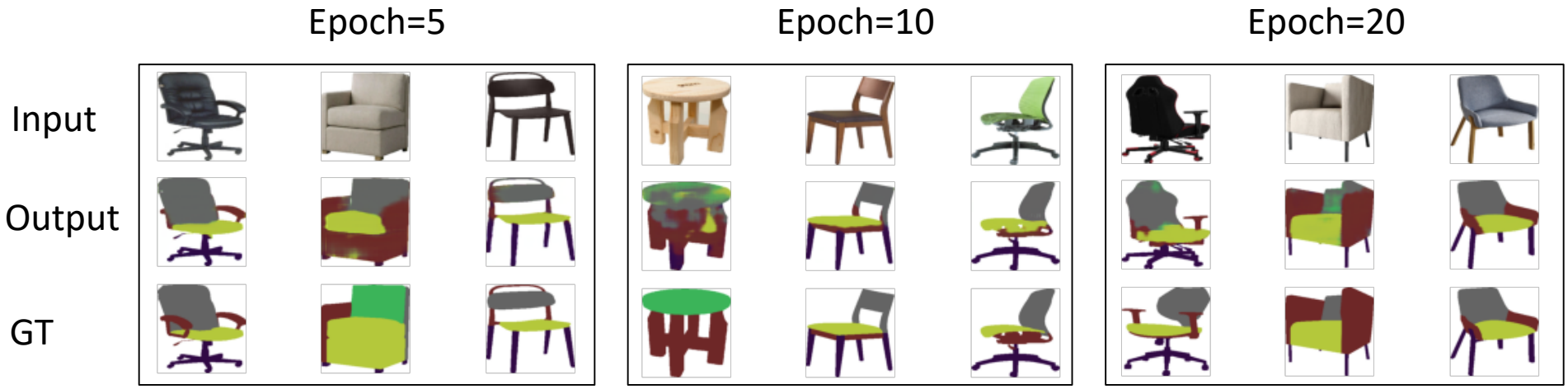


HW4: Segmentation

- Homework due: 5/28 23:59
- Late submissions will incur a penalty of one point for each day overdue.
- The assignment allows a maximum extension of 3 days (it will not be accepted if submitted later than 3 days).
- Submit files:
  1) Code file (.py or .ipynb) containing your implementation.
  2) 3 images of Validation results at epoch 5, 10, and 20.
  3) Test results consisting of 10 segmentation images.
- This assignment can be carried out using Colab or completed on your PC.

HW4: Segmentation

- Example of Validation Result



HW4: Segmentation

- Rules:

  (1) Participants are allowed to download pre-trained models for fine-tuning.

  (2) Participants may also train a new model from scratch for prediction.

  (3) Higher scores will be awarded to participants who build their models from scratch.

  (4) The segmentation result itself will not affect the scoring.

  (5) Bonus (+10): Implement an **IoU** function to calculate the IoU value for each segmentation results and the overall **mIoU**.

HW4: Segmentation

- Evaluation Metrics – IoU
  - ✓ IoU: Intersection over Union.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

HW4: Segmentation

- Example
  - PSNR: 17.71
  - IoU: 扶手: 97.13%, 椅腳: 95.99%, 椅子底: 85.40%, 椅墊: 97.28%

- 椅墊: (60,180,90) (綠色)
- 扶手: (110,40,40) (咖啡色)
- 椅腳: (50,10,70) (黑色)
- 椅子底: (180,200,60) (黃色)
- 椅背: (100,100,100) (灰色)

```
# select random example
ix = randint(0, len(X1), 1)
src_image, tar_image = X1[ix], X2[ix]
gen_image = model.predict(src_image)
print(ix,PSNR(gen_image, tar_image)[0])
plot_images(src_image, gen_image, tar_image)
```

```
[30] 17.713072
```


```
r=60;g=180;b=90
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅墊:",result[1])

r=110;g=40;b=40
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("扶手:",result[1])

r=50;g=10;b=70
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅腳:",result[1])

r=180;g=200;b=60
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅子底:",result[1])

r=100;g=100;b=100
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅墊:",result[1])
```

```
椅墊: 0
扶手: 0.9713475751746482
椅腳: 0.9599682392642096
椅子底: 0.8546839299314547
椅墊: 0.9728260869565217
```

HW4: Segmentation

- Example
  - PSNR: 17.21
  - IoU: 椅腳: 77.18%, 椅子底: 98.49%

- 椅墊: (60,180,90) (綠色)
- 扶手: (110,40,40) (咖啡色)
- 椅腳: (50,10,70) (黑色)
- 椅子底: (180,200,60) (黃色)
- 椅背: (100,100,100) (灰色)

```
# select random example
ix = randint(0, len(X1), 1)
src_image, tar_image = X1[ix], X2[ix]
gen_image = model.predict(src_image)
print(ix,PSNR(gen_image, tar_image)[0])
plot_images(src_image, gen_image, tar_image)
```

```
[1002] 17.214153
```


```
r=60;g=180;b=90
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅墊:",result[1])

r=110;g=40;b=40
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("扶手:",result[1])

r=50;g=10;b=70
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅腳:",result[1])

r=180;g=200;b=60
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅子底:",result[1])

r=100;g=100;b=100
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅墊:",result[1])
```

```
椅墊: 0
扶手: 0
椅腳: 0.771875
椅子底: 0.9849449088145896
椅墊: 0
```

HW4: Segmentation

- Example
  - PSNR: 11.40
  - IoU: 椅腳: 95.80%, 椅子底: 90.30%, 椅墊: 94.92%

- 椅墊: (60,180,90) (綠色)
- 扶手: (110,40,40) (咖啡色)
- 椅腳: (50,10,70) (黑色)
- 椅子底: (180,200,60) (黃色)
- 椅背: (100,100,100) (灰色)

```
# select random example
ix = randint(0, len(X1), 1)
src_image, tar_image = X1[ix], X2[ix]
gen_image = model.predict(src_image)
print(ix,PSNR(gen_image, tar_image)[0])
plot_images(src_image, gen_image, tar_image)
```

```
[624] 11.400866
```


```
r=60;g=180;b=90
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅墊:",result[1])

r=110;g=40;b=40
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("扶手:",result[1])

r=50;g=10;b=70
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅腳:",result[1])

r=180;g=200;b=60
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅子底:",result[1])

r=100;g=100;b=100
cnt=0;s_iou=0
result=iou(r,g,b,gen_image,tar_image)
print("椅墊:",result[1])
```

```
椅墊: 0
扶手: 0
椅腳: 0.9580444273240883
椅子底: 0.9030664147652957
椅墊: 0.94921875
```

HW4: Segmentation

- Result
  - 椅墊: 76.76%

椅墊: (60,180,90) (綠色)

In [31]:
```
r=60;g=180;b=90
cnt=0;s_iou=0
for k in test:
    k=np.array([k])
    src_image, tar_image = X1[k], X2[k]
    gen_image = model.predict(src_image)
    result=iou(r,g,b,gen_image,tar_image)
    cnt+=result[0]
    s_iou+=result[1]

s_iou/cnt
```

```
0.7676898467979737
```

  - 扶手: 77.64%

扶手: (110,40,40) (咖啡色)

In [32]:
```
r=110;g=40;b=40
cnt=0;s_iou=0
for k in test:
    k=np.array([k])
    src_image, tar_image = X1[k], X2[k]
    gen_image = model.predict(src_image)
    result=iou(r,g,b,gen_image,tar_image)
    cnt+=result[0]
    s_iou+=result[1]

s_iou/cnt
```

```
0.776427139455019
```

HW4: Segmentation

- Result
  - 椅腳: 67.86%

椅腳: (50,10,70) (黑色)

In [33]:
```
r=50;g=10;b=70
cnt=0;s_iou=0
for k in test:
    k=np.array([k])
    src_image, tar_image = X1[k], X2[k]
    gen_image = model.predict(src_image)
    result=iou(r,g,b,gen_image,tar_image)
    cnt+=result[0]
    s_iou+=result[1]

s_iou/cnt
```

```
0.6786286786968682
```

  - 扶手: 75.83%

椅子底: (180,200,60) (黃色)

In [34]:
```
r=180;g=200;b=60
cnt=0;s_iou=0
for k in test:
    k=np.array([k])
    src_image, tar_image = X1[k], X2[k]
    gen_image = model.predict(src_image)
    result=iou(r,g,b,gen_image,tar_image)
    cnt+=result[0]
    s_iou+=result[1]

s_iou/cnt
```

```
0.7583794512032871
```

HW4: Segmentation

- Result
  - 椅背: 60.20%

椅背: (100,100,100) (灰色)

```
In [35]: r=100;g=100;b=100
         cnt=0;s_iou=0
         for k in test:
             k=np.array([k])
             src_image, tar_image = X1[k], X2[k]
             gen_image = model.predict(src_image)
             result=iou(r,g,b,gen_image,tar_image)
             cnt+=result[0]
             s_iou+=result[1]

         s_iou/cnt

Out[35]: 0.6020309908219154
```

  - Overall IoU: 71.66%

```
(0.7676898467979737+0.776427139455019+0.6786286786968682+0.7583794512032871+0.6020309908219154)/5

0.7166312213950127
```

IIAI30013

Computer Vision

HW4: Segmentation

Q & A

Instructor: YuanFu Yang

yfyangd@nycu.edu.tw