CREATIVE
INNOVATION
REAMAKE

Background Removal 진행 상황 보고

개발 인턴 김경태

REMAKE DIGITAL AGENCY

CROLO

Parameter – Pymatting

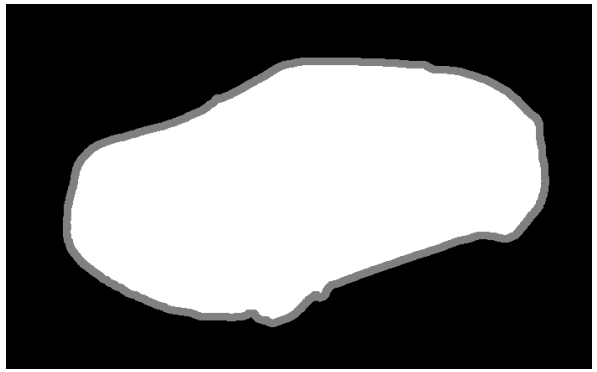
Pymatting + PointRend 값 조정 및 향상

value = 1e-3
alpha = estimate_alpha_cf(image, trimap, laplacian_kwargs={"epsilon": value},
cg_kwargs= {"maxiter":3000})

value2 = 0.5~0.7
foreground= estimate_foreground_ml(image, alpha, regularization = value2)

PointRend → Trimap → Pymatting

Pymatting



Inputs

ouput

PointRend → Trimap → Pymatting

Google Colab with Cuda 10.0

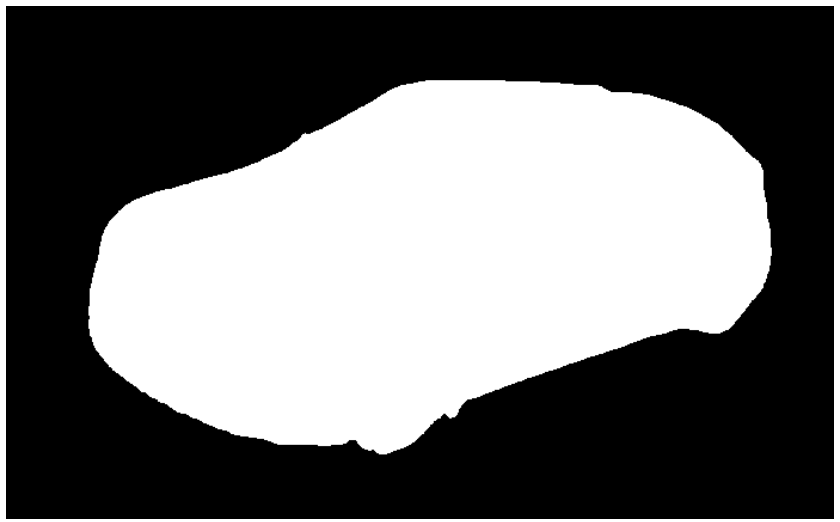
PointRend

```
cfg = get_cfg()
# Add PointRend-specific config
point_rend.add_pointrend_config(cfg)
# Load a config from file
cfg.merge_from_file("detectron2_repo/projects/PointRend/configs/InstanceSegmentation/pointrend_rcnn_R_50_FPN_3x_coco.yaml")
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5  # set threshold for this model
# Use a model from PointRend model zoo: https://github.com/facebookresearch/detectron2/tree/master/projects/PointRend#pretrained-models
cfg.MODEL.WEIGHTS = "https://dl.fbaipublicfiles.com/detectron2/PointRend/InstanceSegmentation/pointrend_rcnn_R_50_FPN_3x_coco/164955410/model_final_3c3198.pkl"
predictor = DefaultPredictor(cfg)
outputs = predictor(img)
```

```
[ ] outputs
```

```
{'instances': Instances(num_instances=1, image_height=589, image_width=626, fields=[pred_boxes: Boxes(tensor([[ 16.6632,  19.1460, 620.5416, 585.3265]], device='cuda:0')), scores: tensor([0.9886], device='cuda:0'), pred_classes: tensor([0], device='cuda:0')
         [False, False, False,  ..., False, False, False],
         [False, False, False,  ..., False, False, False],
         ...,
         [False, False, False,  ..., False, False, False],
         [False, False, False,  ..., False, False, False],
         [False, False, False,  ..., False, False, False]]], device='cuda:0'))}
```

PointRend → Trimap → Pymatting



PointRend → Trimap → Pymatting

Pymatting

```
!pip3 install pymatting
from pymatting import cutout

def gen_matted_image(img_path, trimap_path, cutout_path):
  try:
    cutout(
        # input image path
        img_path,
        # input trimap path
        trimap_path,
        # output cutout path
        cutout_path
    )
    return "success"
  except ValueError as e:
    return e
```

PointRend → Trimap → Pymatting

Pymatting

```
!pip3 install pymatting
from pymatting import cutout

def gen_matted_image(img_path, trimap_path, cutout_path):
  try:
    cutout(
        # input image path
        img_path,
        # input trimap path
        trimap_path,
        # output cutout path
        cutout_path
    )
    return "success"
  except ValueError as e:
    return e
```

Cutout is for application-oriented users

PointRend → Trimap → Pymatting

Pymatting

```
!pip3 install pymatting
from pymatting import *
import numpy as np

scale = 1.0
image_name = 'Hyundai (6).png'
image = load_image("/content/drive/My Drive/remake/matting/main-jpg/Automobile/" + image_name, "RGB", scale, "box")
# image = load_image("/content/drive/My Drive/remake/matting/main-jpg/Automobile/" + image_name, "RGB", scale, "box")

trimap = load_image("/content/drive/My Drive/remake/matting/trimap/Automobile/" + image_name, "GRAY", scale, "nearest")
# trimap = load_image("/content/drive/My Drive/remake/matting/trimap/Automobile/" + image_name, "GRAY", scale, "nearest")

value = 1e-1
alpha = estimate_alpha_cf(image, trimap, laplacian_kwargs={"epsilon": value}, cg_kwargs= {"maxiter":3000})
# alpha = estimate_alpha_cf(image, trimap)

# background = np.zeros(image.shape)
# background[:, :] = [0.5, 0.5, 0.5]
image.shape

foreground, background = estimate_foreground_ml(image, alpha, return_background=True)

new_image = blend(foreground, background, alpha)

new_image

# images = [image, trimap, alpha, new_image]
# grid = make_grid(images)
# save_image("/content/drive/My Drive/remake/matting/pymatting_image/" + image_name[:-4] + "_grid_" + str(value) + ".png", grid)

cutout = stack_images(foreground, alpha)
save_image("/content/drive/My Drive/remake/matting/pymatting_image/" + image_name[:-4] + "_cutout_" + str(value) + ".png", cutout)
save_image("/content/drive/My Drive/remake/matting/pymatting_image/" + image_name[:-4] + "_new_image_" + str(value) + ".png", new_image)
```

Low level

laplacian_kwargs : regularization strength

PointRend → Trimap → Pymatting

laplacian_kwargs : 1e-7

laplacian_kwargs : 1e-1

PointRend → Trimap → Pymatting

PointRend

PointRend → Trimap → Pymatting

Notion Page : Foreground segmentation >> BG removal algorithm

To do

Should trim the exact border of the matted image.

Test pymatting for trimaps with unknown area of several sizes.

Test pymatting adjusting parameters in low level.

To do

Should trim the exact border of the matted image.

Analyze pointrend and adjusting parameters

Learning pointrend

Generate Trimap in other ways

Test using other instance/semantic segmentation models.