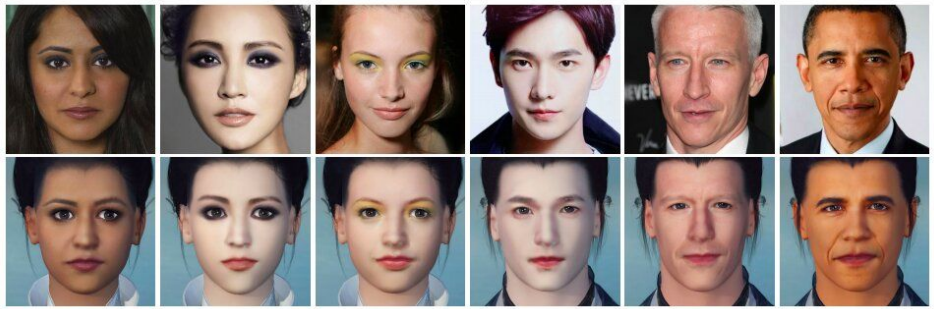
한국인 사진을 스케치로 변환시키기

목차

프로젝트 소개

GAN 활용 사례



프로젝트 소개

- 한국인 가상 인물 사진과 각각에 대응되는 스케치 이미지를 학습시켜 한국인의 사진을 입력받으면 스케치 이미지로 변환시키는 프로젝트
- 스케치 이미지는 Pix2Pix 모델 활용

데이터 소개 및 전처리

한국인 가상인물 사진 이미지와 각각에 대응되는 스케치 이미지

데이터 소개 및 전처리

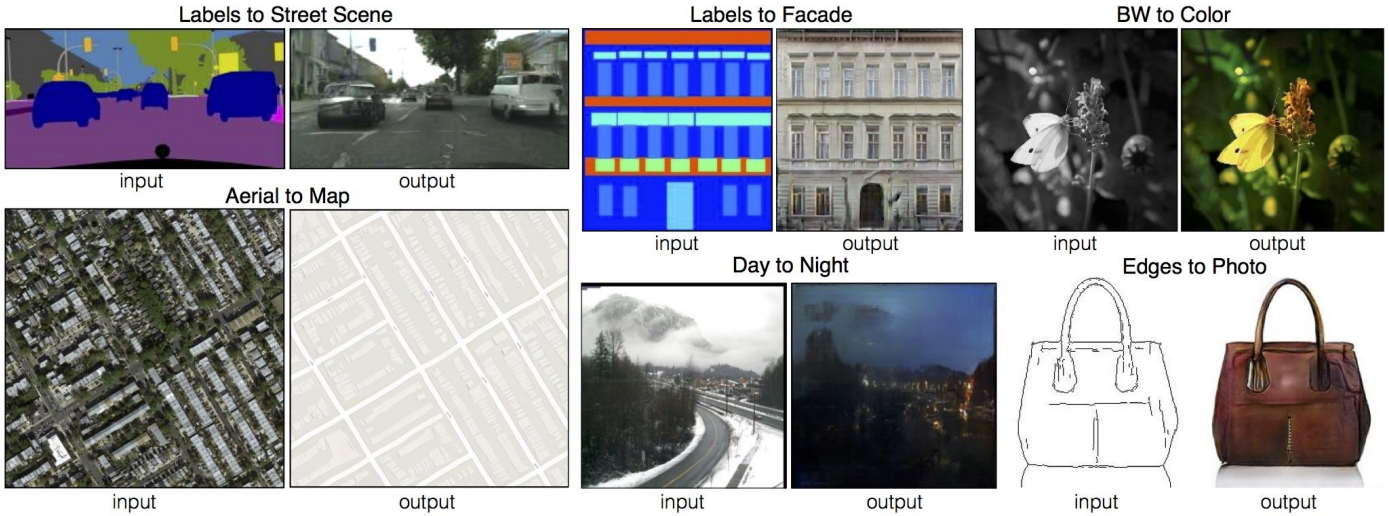
데이터 전처리

이미지를 불러올 때 정규화 진행

```
# image load
def load_image(filename, size=(256,256)):
    # image load
    pixels = tensorflow.keras.preprocessing.image.load_img(filename, target_size=size)
    # numpy array로 변환
    pixels = tensorflow.keras.preprocessing.image.img_to_array(pixels)
    # [0,255]에서 [-1,1]로 scaling
    pixels = (pixels - 127.5) / 127.5
    # 1 sample로 reshape
    pixels = expand_dims(pixels, 0)
    return pixels
```

모델 소개 및 학습 결과

Pix2pix 모델 소개



모델 소개 및 학습 결

Pix2pix 모델 구성

Pix2pix는 GAN으로 생성자와

판별자로 구성

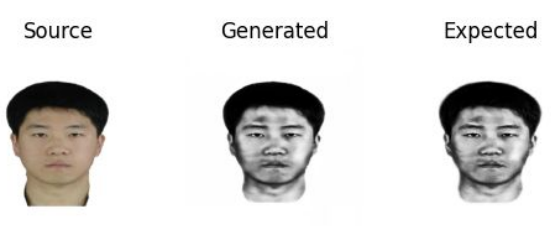
Model: "model_1"

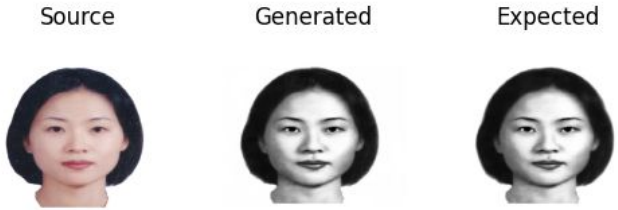
Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[(None, 256, 256, 3)]	0	[]
conv2d_6 (Conv2D)	(None, 128, 128, 64)	3136	['input_3[0][0]']
leaky_re_lu_5 (LeakyReLU)	(None, 128, 128, 64)	0	['conv2d_6[0][0]']
conv2d_7 (Conv2D)	(None, 64, 64, 128)	131200	['leaky_re_lu_5[0][0]']
batch_normalization_4 (BatchNormalization)	(None, 64, 64, 128)	512	['conv2d_7[0][0]']
leaky_re_lu_6 (LeakyReLU)	(None, 64, 64, 128)	0	['batch_normalization_4[0][0]']
conv2d_8 (Conv2D)	(None, 32, 32, 256)	524544	['leaky_re_lu_6[0][0]']
batch_normalization_5 (BatchNormalization)	(None, 32, 32, 256)	1024	['conv2d_8[0][0]']
leaky_re_lu_7 (LeakyReLU)	(None, 32, 32, 256)	0	['batch_normalization_5[0][0]']
conv2d_9 (Conv2D)	(None, 16, 16, 512)	2097664	['leaky_re_lu_7[0][0]']
batch_normalization_6 (BatchNormalization)	(None, 16, 16, 512)	2048	['conv2d_9[0][0]']
leaky_re_lu_8 (LeakyReLU)	(None, 16, 16, 512)	0	['batch_normalization_6[0][0]']
conv2d_10 (Conv2D)	(None, 8, 8, 512)	4194816	['leaky_re_lu_8[0][0]']
batch_normalization_7 (BatchNormalization)	(None, 8, 8, 512)	2048	['conv2d_10[0][0]']
leaky_re_lu_9 (LeakyReLU)	(None, 8, 8, 512)	0	['batch_normalization_7[0][0]']
conv2d_11 (Conv2D)	(None, 4, 4, 512)	4194816	['leaky_re_lu_9[0][0]']
batch_normalization_8 (BatchNormalization)	(None, 4, 4, 512)	2048	['conv2d_11[0][0]']
leaky_re_lu_10 (LeakyReLU)	(None, 4, 4, 512)	0	['batch_normalization_8[0][0]']
conv2d_12 (Conv2D)	(None, 2, 2, 512)	4194816	['leaky_re_lu_10[0][0]']
batch_normalization_9 (BatchNormalization)	(None, 2, 2, 512)	2048	['conv2d_12[0][0]']
leaky_re_lu_11 (LeakyReLU)	(None, 2, 2, 512)	0	['batch_normalization_9[0][0]']
conv2d_13 (Conv2D)	(None, 1, 1, 512)	4194816	['leaky_re_lu_11[0][0]']
activation_1 (Activation)	(None, 1, 1, 512)	0	['conv2d_13[0][0]']
conv2d_transpose (Conv2DTranspose)	(None, 2, 2, 512)	4194816	['activation_1[0][0]']
batch_normalization_10 (BatchNormalization)	(None, 2, 2, 512)	2048	['conv2d_transpose[0][0]']
dropout (Dropout)	(None, 2, 2, 512)	0	['batch_normalization_10[0][0]']
concatenate_1 (Concatenate)	(None, 2, 2, 1024)	0	['dropout[0][0]', 'leaky_re_lu_11[0][0]']
activation_2 (Activation)	(None, 2, 2, 1024)	0	['concatenate_1[0][0]']
conv2d_transpose_1 (Conv2DTranspose)	(None, 4, 4, 512)	8389120	['activation_2[0][0]']
batch_normalization_11 (BatchNormalization)	(None, 4, 4, 512)	2048	['conv2d_transpose_1[0][0]']
dropout_1 (Dropout)	(None, 4, 4, 512)	0	['batch_normalization_11[0][0]']
concatenate_2 (Concatenate)	(None, 4, 4, 1024)	0	['dropout_1[0][0]', 'leaky_re_lu_10[0][0]']
activation_3 (Activation)	(None, 4, 4, 1024)	0	['concatenate_2[0][0]']
conv2d_transpose_2 (Conv2DTranspose)	(None, 8, 8, 512)	8389120	['activation_3[0][0]']
batch_normalization_12 (BatchNormalization)	(None, 8, 8, 512)	2048	['conv2d_transpose_2[0][0]']
dropout_2 (Dropout)	(None, 8, 8, 512)	0	['batch_normalization_12[0][0]']
concatenate_3 (Concatenate)	(None, 8, 8, 1024)	0	['dropout_2[0][0]', 'leaky_re_lu_9[0][0]']
activation_4 (Activation)	(None, 8, 8, 1024)	0	['concatenate_3[0][0]']
conv2d_transpose_3 (Conv2DTranspose)	(None, 16, 16, 512)	8389120	['activation_4[0][0]']
batch_normalization_13 (BatchNormalization)	(None, 16, 16, 512)	2048	['conv2d_transpose_3[0][0]']
concatenate_4 (Concatenate)	(None, 16, 16, 1024)	0	['batch_normalization_13[0][0]', 'leaky_re_lu_8[0][0]']
activation_5 (Activation)	(None, 16, 16, 1024)	0	['concatenate_4[0][0]']
conv2d_transpose_4 (Conv2DTranspose)	(None, 32, 32, 256)	4194560	['activation_5[0][0]']
batch_normalization_14 (BatchNormalization)	(None, 32, 32, 256)	1024	['conv2d_transpose_4[0][0]']
concatenate_5 (Concatenate)	(None, 32, 32, 512)	0	['batch_normalization_14[0][0]', 'leaky_re_lu_7[0][0]']
activation_6 (Activation)	(None, 32, 32, 512)	0	['concatenate_5[0][0]']
conv2d_transpose_5 (Conv2DTranspose)	(None, 64, 64, 128)	1048704	['activation_6[0][0]']
batch_normalization_15 (BatchNormalization)	(None, 64, 64, 128)	512	['conv2d_transpose_5[0][0]']
concatenate_6 (Concatenate)	(None, 64, 64, 256)	0	['batch_normalization_15[0][0]', 'leaky_re_lu_6[0][0]']
activation_7 (Activation)	(None, 64, 64, 256)	0	['concatenate_6[0][0]']
conv2d_transpose_6 (Conv2DTranspose)	(None, 128, 128, 64)	262208	['activation_7[0][0]']
batch_normalization_16 (BatchNormalization)	(None, 128, 128, 64)	256	['conv2d_transpose_6[0][0]']
concatenate_7 (Concatenate)	(None, 128, 128, 128)	0	['batch_normalization_16[0][0]', 'leaky_re_lu_5[0][0]']
activation_8 (Activation)	(None, 128, 128, 128)	0	['concatenate_7[0][0]']
conv2d_transpose_7 (Conv2DTranspose)	(None, 256, 256, 3)	6147	['activation_8[0][0]']
activation_9 (Activation)	(None, 256, 256, 3)	0	['conv2d_transpose_7[0][0]']

Total params: 54,429,315
Trainable params: 54,419,459
Non-trainable params: 9,856

모델 소개 및 학습 결과

Pix2pix 모델 학습 결과





후기

좋았던 점

- Pix2pix 모델을 직접 활용해봄으로써 GAN의 구조에 대해 이해하는데 도움이 되었다

아쉬운점

- 더 많은 epoch 학습 후 비교해보고 싶었으나 Colab의 경우 GPU 백엔드 사용량 제한이 걸리고 local 환경에서 학습 시킬 경우 학습 시간이 길어져 하지 못함

감사합니다