이미지 생성 모델

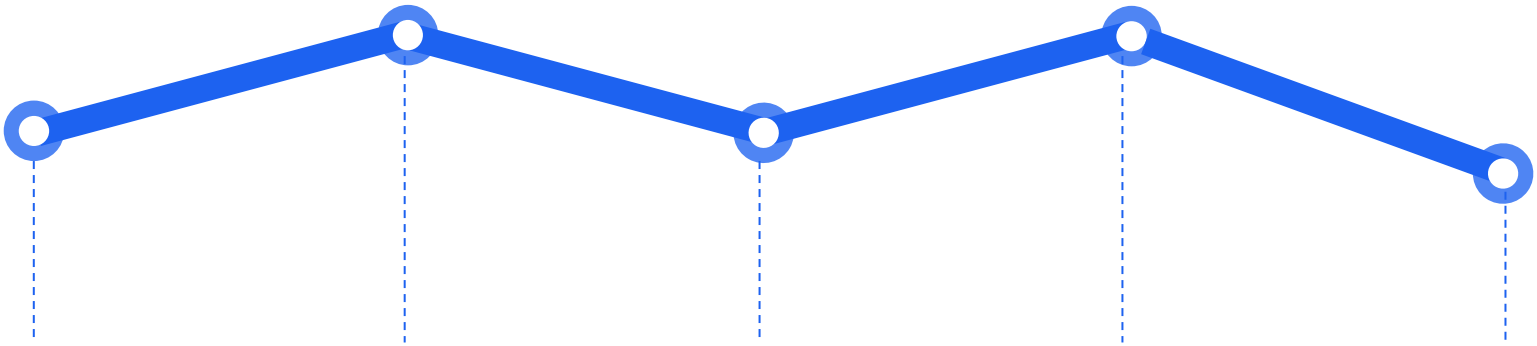
GAN 모델 - pix2pix
CNN 모델 - autoencoder

↓

다양한 형태의 이미지 생성

1)흑백이미지 채색화(coloring)
2)모네화풍으로 변환(monet)
3)만화화풍으로 변환(comic)

목차
CONTENTS

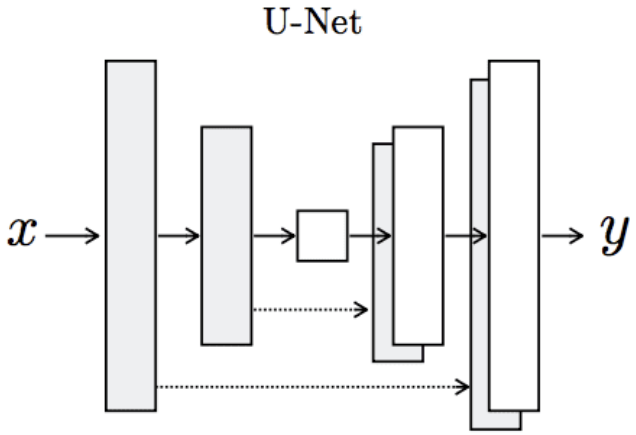


01 분석방향

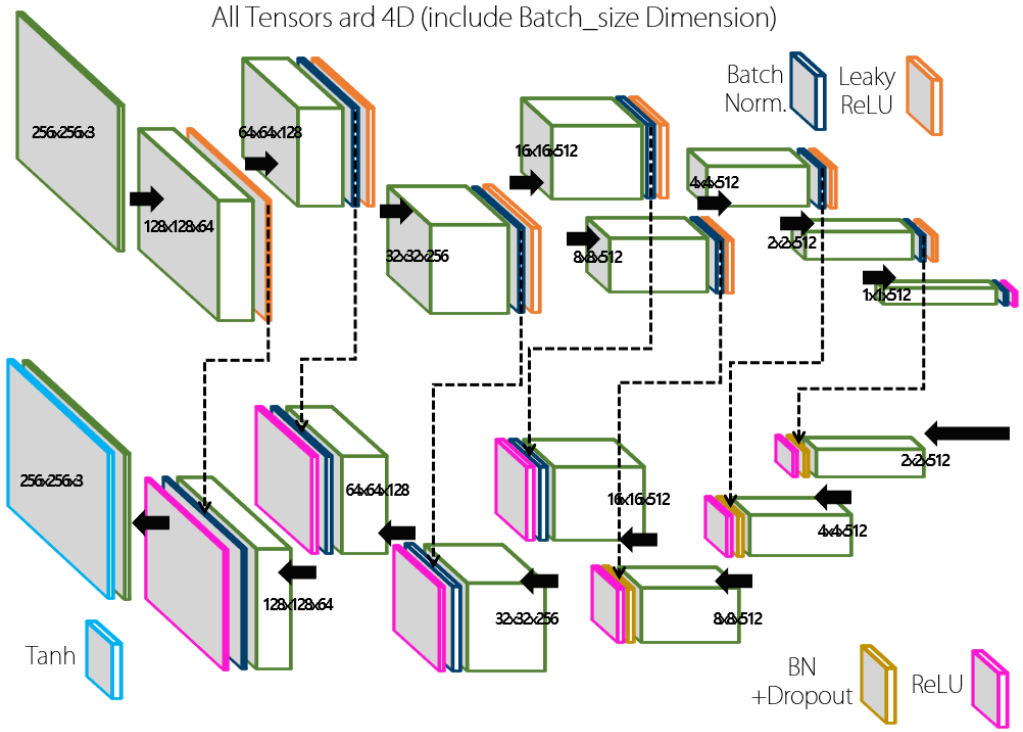
pix2pix를 활용한 coloring, monet화, comic화를 진행하였습니다.
MSE, PSNR, SSIM 평가지표를 이용해 결과값을 도출하였습니다.
추가적으로 CNN 모델 중 하나인 autoencoder 모델을 구현하여 pix2pix와 autoencoder 결과값을 비교 분석하였습니다.

1. pix2pix 모델 구조 - Generator와 Discriminator로 나눠진 구조

1) Generator



< Unet 구조 >



< Generator >

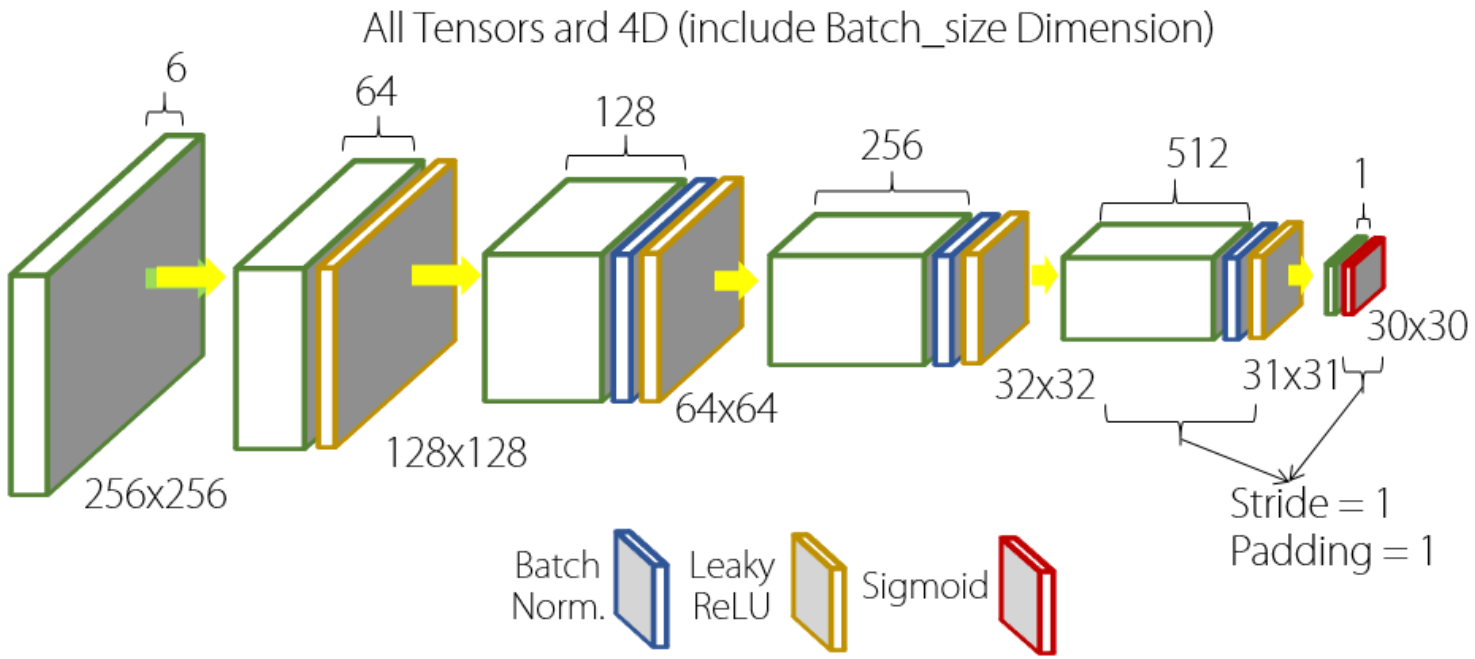
01 분석방향

pix2pix를 활용한 coloring, monet화, comic화를 진행하였습니다.
MSE, PSNR, SSIM 평가지표를 이용해 결과값을 도출하였습니다.
추가적으로 CNN 모델 중 하나인 autoencoder 모델을 구현하여 pix2pix와 autoencoder 결과값을 비교 분석하였습니다.

1. pix2pix 모델 구조

2) Discriminator

patchGAN 구조를 사용함.
: 서로 겹치는 patch로 나누고 각 patch들에 대하여 진짜/가짜를 구별하는 방식

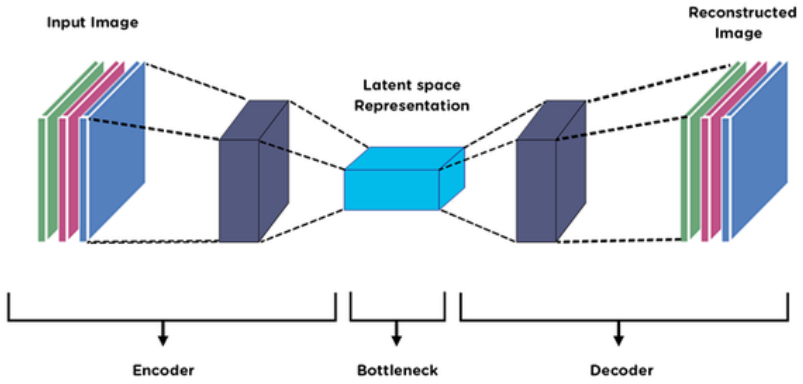


< Discriminator >

01 분석방향

이전까지 GAN 모델 중 하나인 pix2pix를 활용한 coloring을 진행하였습니다.
monet화, comic화를 추가로 진행하였으며, MSE, PSNR, SSIM 평가지표를 이용해 결과값을 도출하였습니다.
추가적으로 CNN 모델 중 하나인 autoencoder 모델을 구현하여 pix2pix와 autoencoder 결과값을 비교 분석하였습니다.

2. autoencoder 모델 구조 - Encoder + Decoder로 이뤄진 구조



Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 256, 256, 3)	0	
sequential_18 (Sequential)	(None, 128, 128, 128	3584	input_2[0][0]
sequential_19 (Sequential)	(None, 64, 64, 128)	147584	sequential_18[0][0]
sequential_20 (Sequential)	(None, 32, 32, 256)	296192	sequential_19[0][0]
sequential_21 (Sequential)	(None, 16, 16, 512)	1182208	sequential_20[0][0]
sequential_22 (Sequential)	(None, 8, 8, 512)	2361856	sequential_21[0][0]
sequential_23 (Sequential)	(None, 16, 16, 512)	2359808	sequential_22[0][0]
concatenate_8 (Concatenate)	(None, 16, 16, 1024)	0	sequential_23[0][0] sequential_21[0][0]
sequential_24 (Sequential)	(None, 32, 32, 256)	2359552	concatenate_8[0][0]
concatenate_9 (Concatenate)	(None, 32, 32, 512)	0	sequential_24[0][0] sequential_20[0][0]
sequential_25 (Sequential)	(None, 64, 64, 128)	589952	concatenate_9[0][0]
concatenate_10 (Concatenate)	(None, 64, 64, 256)	0	sequential_25[0][0] sequential_19[0][0]
sequential_26 (Sequential)	(None, 128, 128, 128	295040	concatenate_10[0][0]
concatenate_11 (Concatenate)	(None, 128, 128, 256	0	sequential_26[0][0] sequential_18[0][0]
sequential_27 (Sequential)	(None, 256, 256, 3)	6915	concatenate_11[0][0]
concatenate_12 (Concatenate)	(None, 256, 256, 6)	0	sequential_27[0][0] input_2[0][0]
conv2d_18 (Conv2D)	(None, 256, 256, 3)	75	concatenate_12[0][0]

Total params: 9,602,766
Trainable params: 9,600,206
Non-trainable params: 2,560

Encoder

Bottleneck

Decoder

01 분석방향

pix2pix를 활용한 coloring, monet화, comic화를 진행하였습니다.
MSE, PSNR, SSIM 평가지표를 이용해 결과값을 도출하였습니다.
추가적으로 CNN 모델 중 하나인 autoencoder 모델을 구현하여 pix2pix와 autoencoder 결과값을 비교 분석하였습니다.

평가 지표 : MSE(Mean Squared Error), PSNR(Peak Signal-to-noise ratio) 소개

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

$$\begin{aligned} PSNR &= 10\log_{10}\left(\frac{{MAX_I}^2}{MSE}\right) \\ &= 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20\log_{10}(MAX_I) - 10\log_{10}(MSE) \end{aligned}$$

```
def evaluate(model, dataset):
    avg_mse = 0
    it = 0
    psnr = 0

    for test_input, tar in dataset :
        it += 1
        print(it)
        gen_output = model(test_input, training=True)
        avg_mse += tf.reduce_mean(tf.compat.v1.squared_difference(tar, gen_output)).numpy()

    avg_mse /= it
    avg_mse = avg_mse * 255 * 255
    psnr = 10 * np.log10(255*255/avg_mse)
    return avg_mse, psnr
```

← 공식 함수화

```
evaluate(UNet_generator, tf.data.Dataset.zip((comic_dataset,face_dataset)))
```

← pix2pix 평가코드

```
evaluate(model, tf.data.Dataset.zip((comic_dataset,face_dataset)))
```

← autoencoder 평가코드

01 분석방향

pix2pix를 활용한 coloring, monet화, comic화를 진행하였습니다.
MSE, PSNR, SSIM 평가지표를 이용해 결과값을 도출하였습니다.
추가적으로 CNN 모델 중 하나인 autoencoder 모델을 구현하여 pix2pix와 autoencoder 결과값을 비교 분석하였습니다.

평가 지표 : SSIM(Structural Similarity Index) 소개

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(2\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

```python
def ssim(y_true , y_pred):
    u_true = np.mean(y_true)
    u_pred = np.mean(y_pred)
    var_true = np.var(y_true)
    var_pred = np.var(y_pred)
    std_true = np.sqrt(var_true)
    std_pred = np.sqrt(var_pred)
    c1 = np.square(0.01*7)
    c2 = np.square(0.03*7)
    ssim = (2 * u_true * u_pred + c1) * (2 * std_pred * std_true + c2)
    denom = (u_true ** 2 + u_pred ** 2 + c1) * (var_pred + var_true + c2)
    return ssim / denom
```

← 공식 함수화

02 Coloring 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

1) pix2pix 결과 정리

Epoch = 0 일 때,
(학습 전)

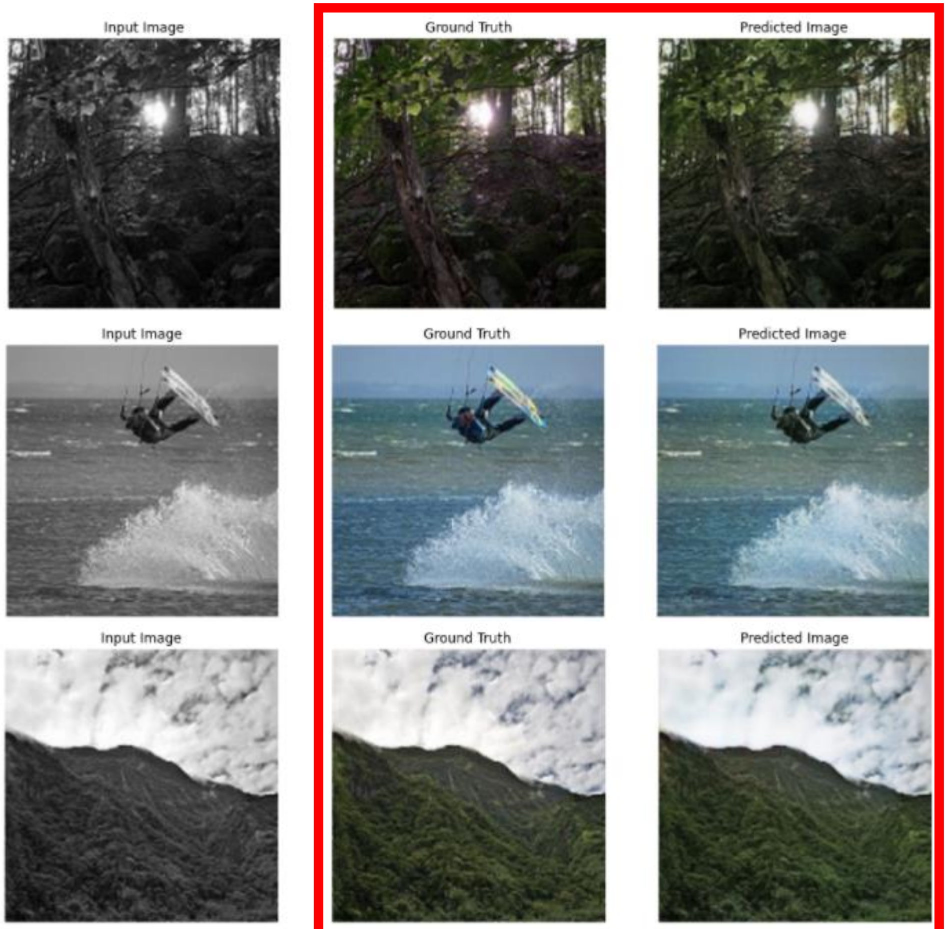
Epoch = 100 일 때,
(학습 후)

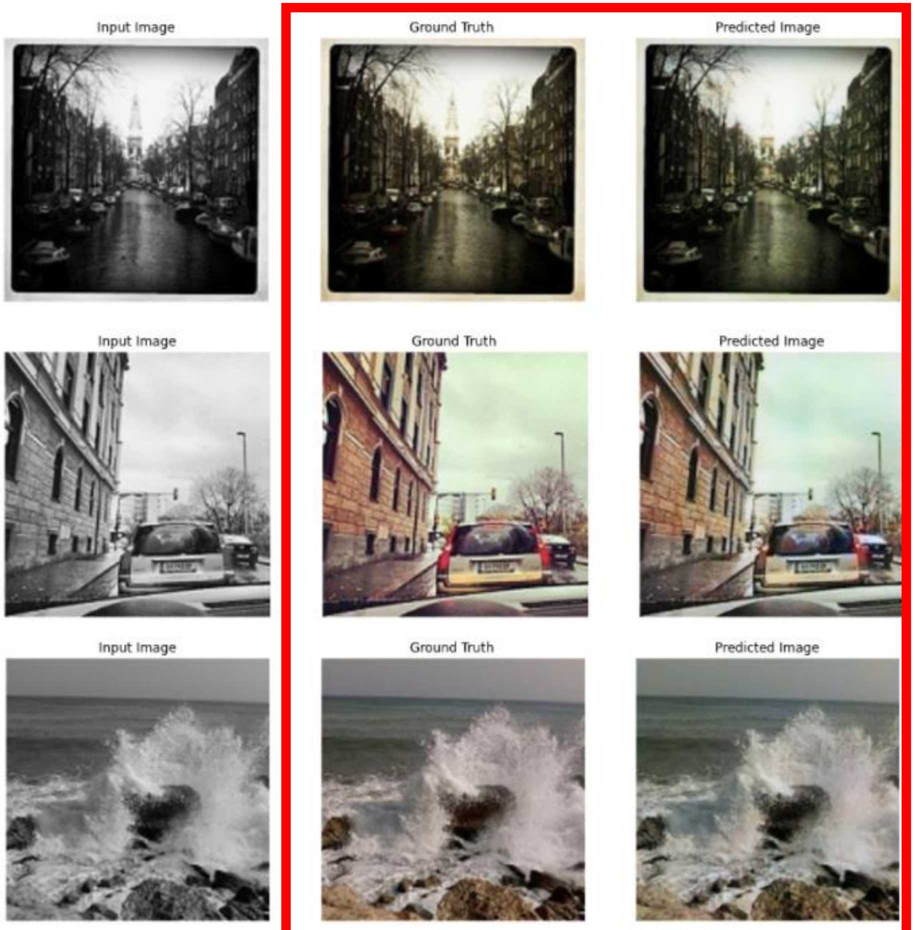
MSE, PSNR, SSIM 값 86.2874936813.23(MSE), 28.771325062742502(PSNR), 0.9982023899249365(SSIM)

02 Coloring 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

1) pix2pix 결과 정리(추가 결과 사진)





02 Coloring 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

2) Autoencoder 결과 정리

Epoch = 0 일 때, (학습 전)

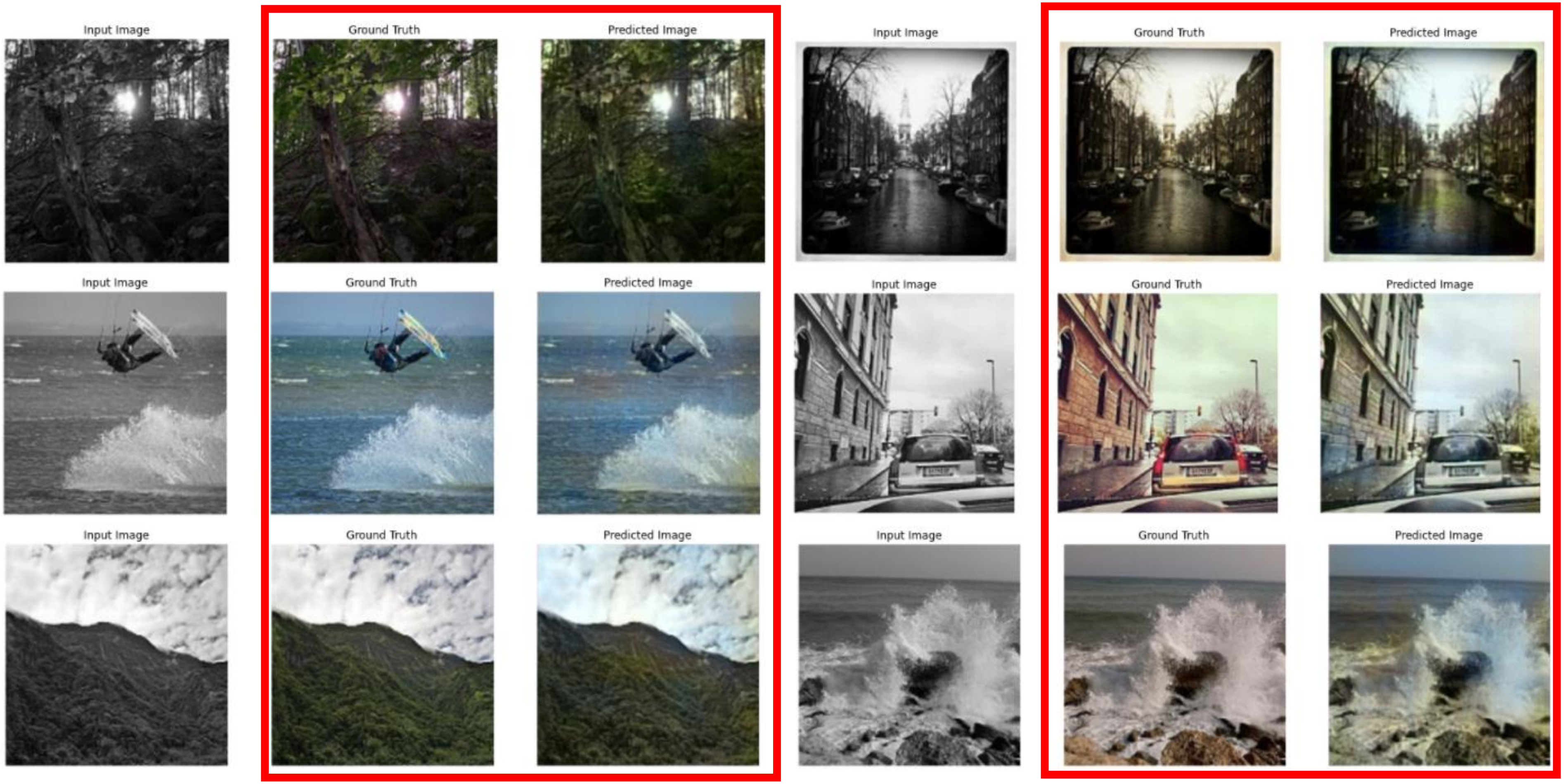
Epoch = 100 일 때, (학습 후)

MSE, PSNR, SSIM 값 510.79449485521764(MSE), 21.048341529086485(PSNR), 0.8416023859812365(SSIM)

02 Coloring 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

2) Autoencoder 결과 정리(추가 결과 사진)

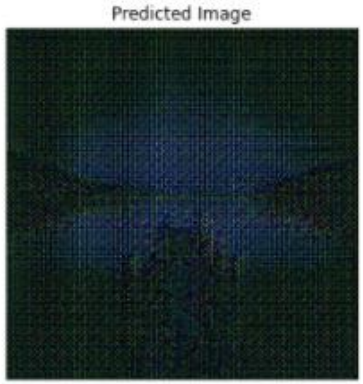


03 Monet 결과정리

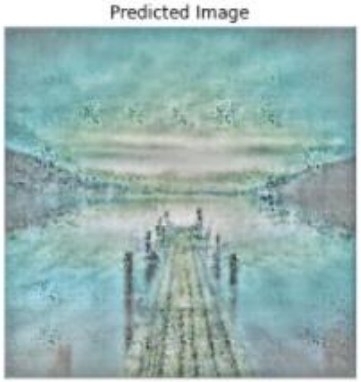
Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

1) pix2pix 결과 정리

Epoch = 0 일 때,
(학습 전)



Epoch = 100 일 때,
(학습 후)

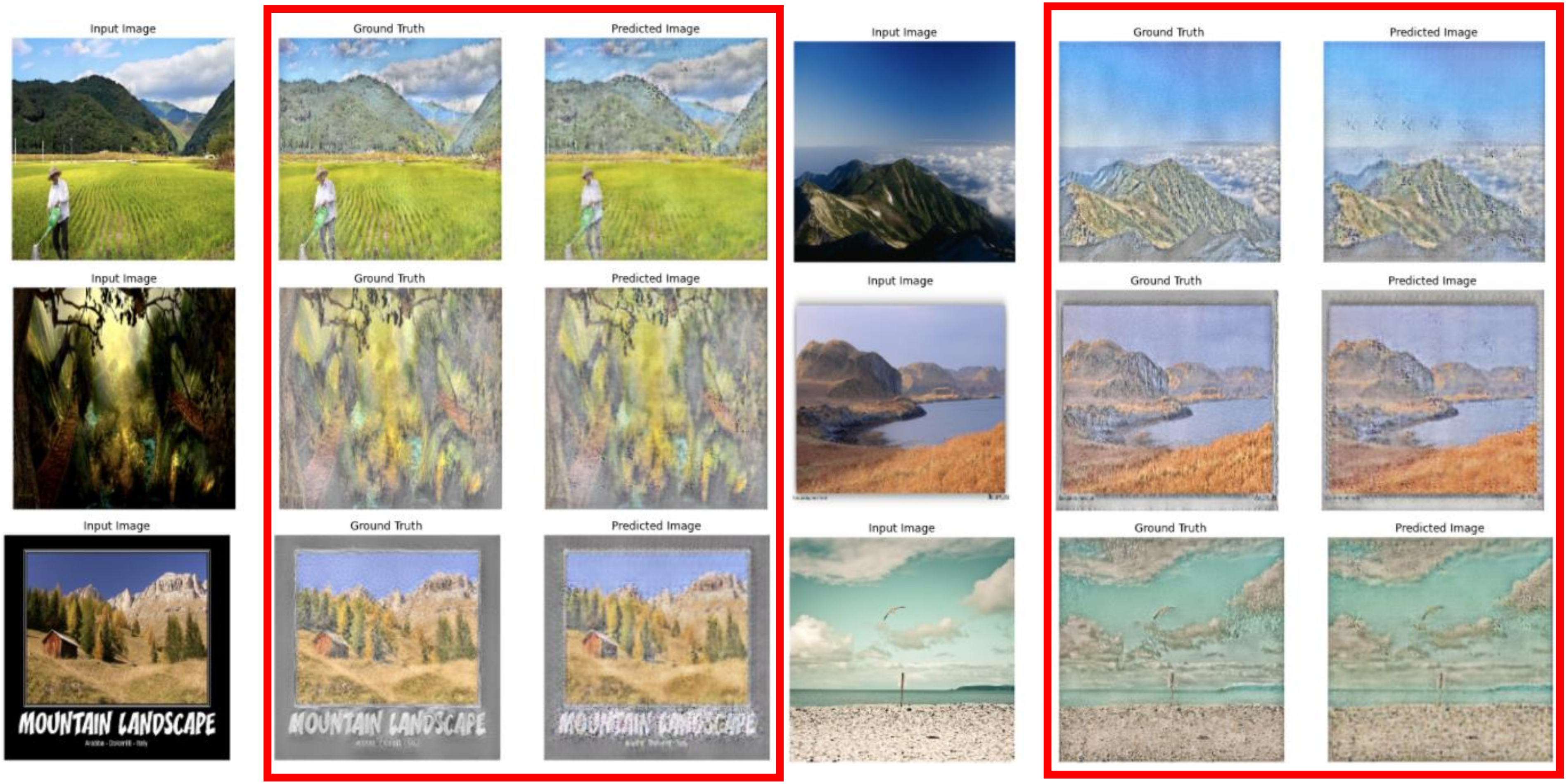


MSE, PSNR, SSIM 값 456.94185440428555(MSE), 21.53219421012434(PSNR), 0.8424010341117464(SSIM)

03 Monet 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

1) pix2pix 결과 정리(추가 결과 사진)



03 Monet 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

2) Autoencoder 결과 정리

Epoch = 0 일 때,
(학습 전)

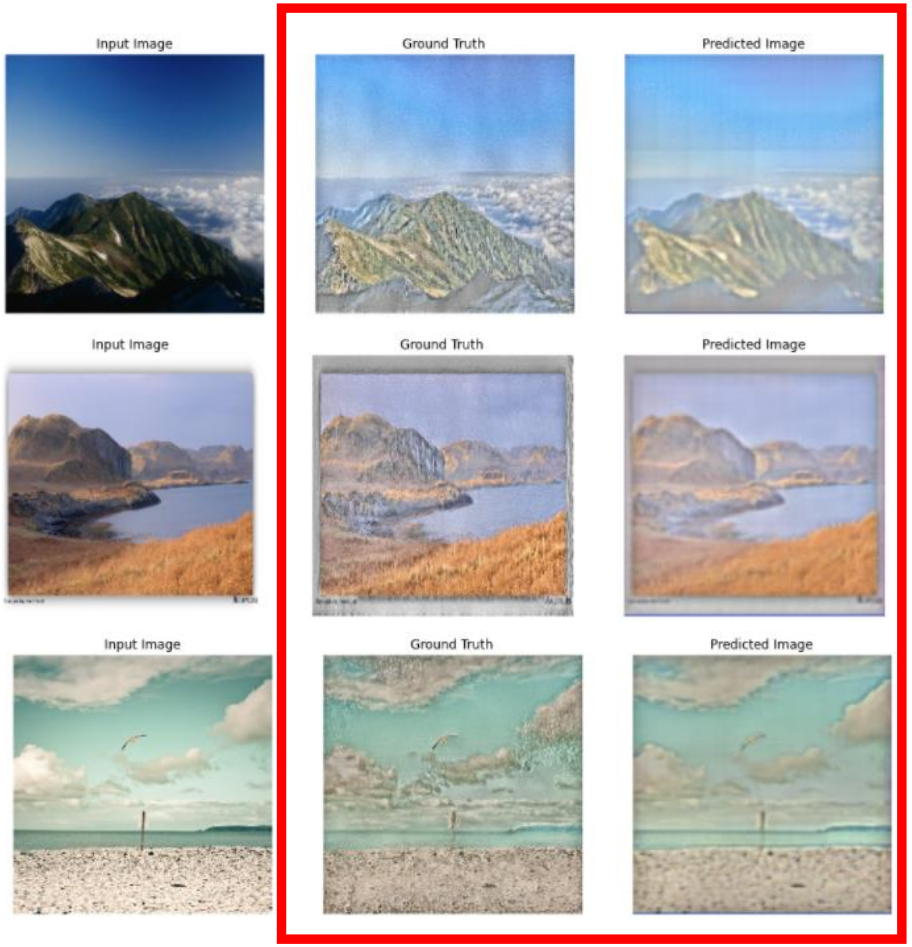
Epoch = 100 일 때,
(학습 후)

MSE, PSNR, SSIM 값 551.0595319897402(MSE), 20.718818418381936(PSNR), 0.6521010846215854(SSIM)

03 Monet 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

2) Autoencoder 결과 정리(추가 결과 사진)

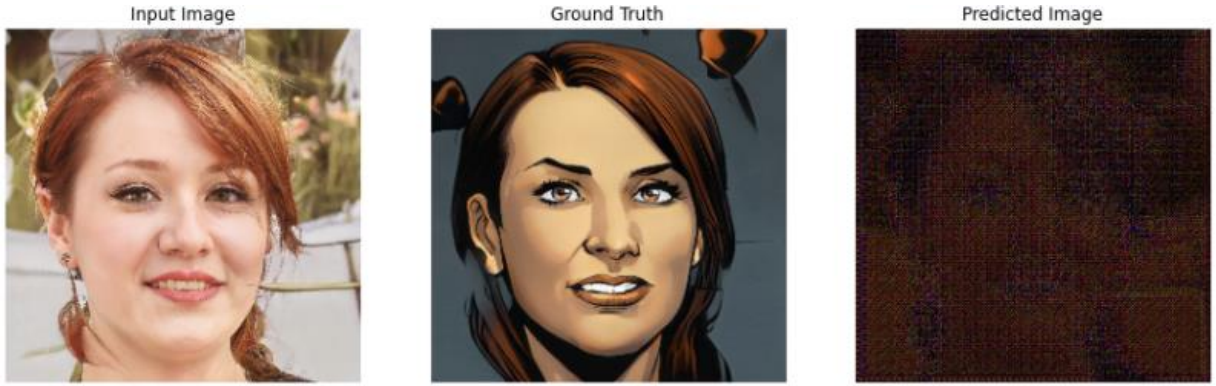


04 Comic 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

1) pix2pix 결과 정리

Epoch = 0 일 때,
(학습 전)



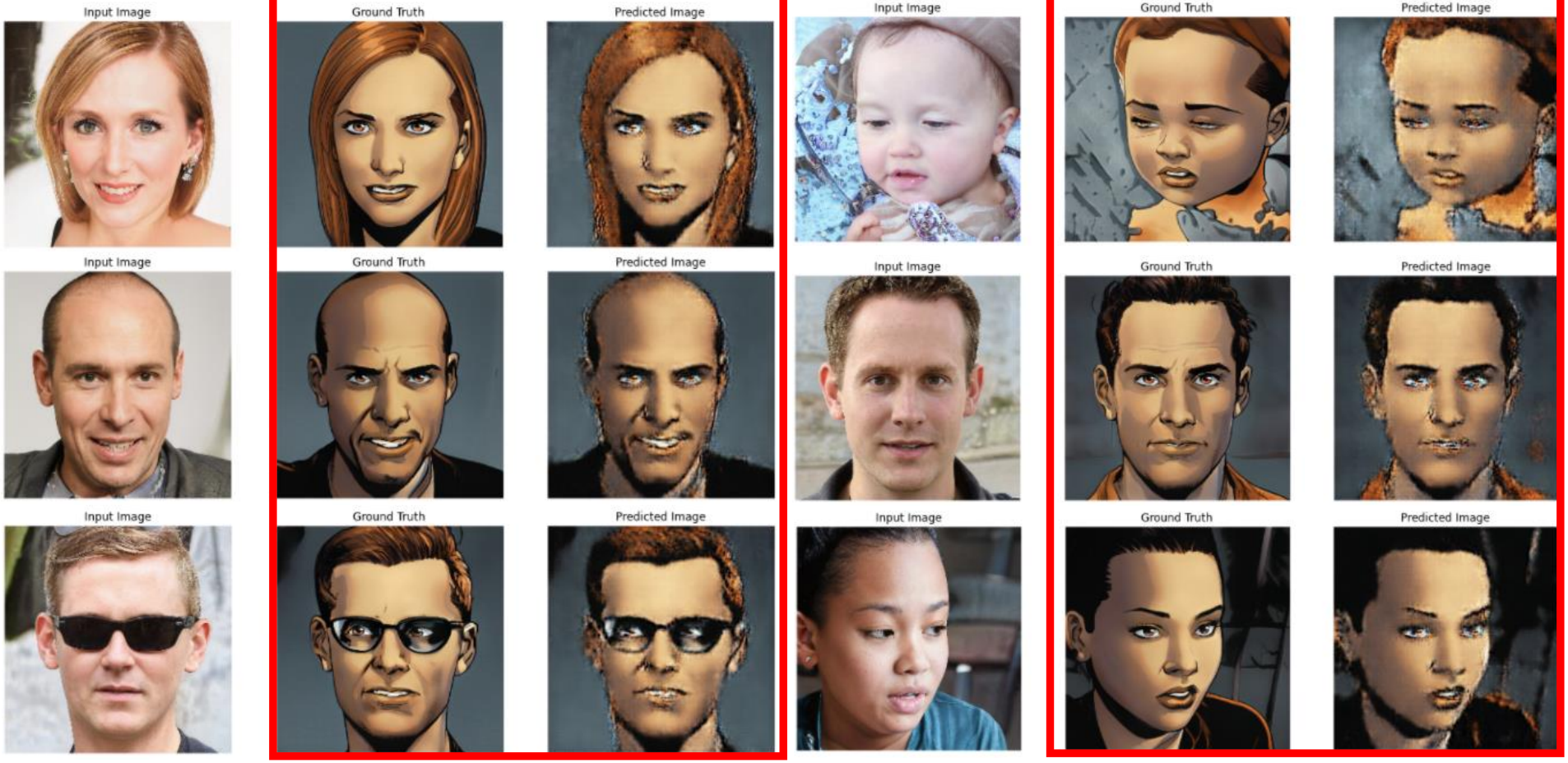
Epoch = 100 일 때,
(학습 후)

MSE, PSNR, SSIM 값 519.8645839921664(MSE), 20.971901289328482(PSNR), 0.8835949517239816(SSIM)

04 Comic 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

1) pix2pix 결과 정리(추가 결과 사진)

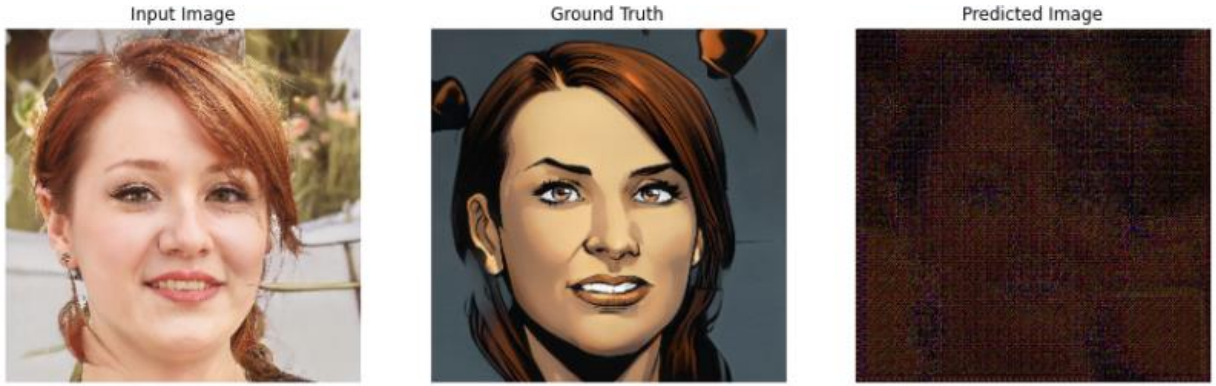


04 Comic 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

2) Autoencoder 결과 정리

Epoch = 0 일 때, (학습 전)



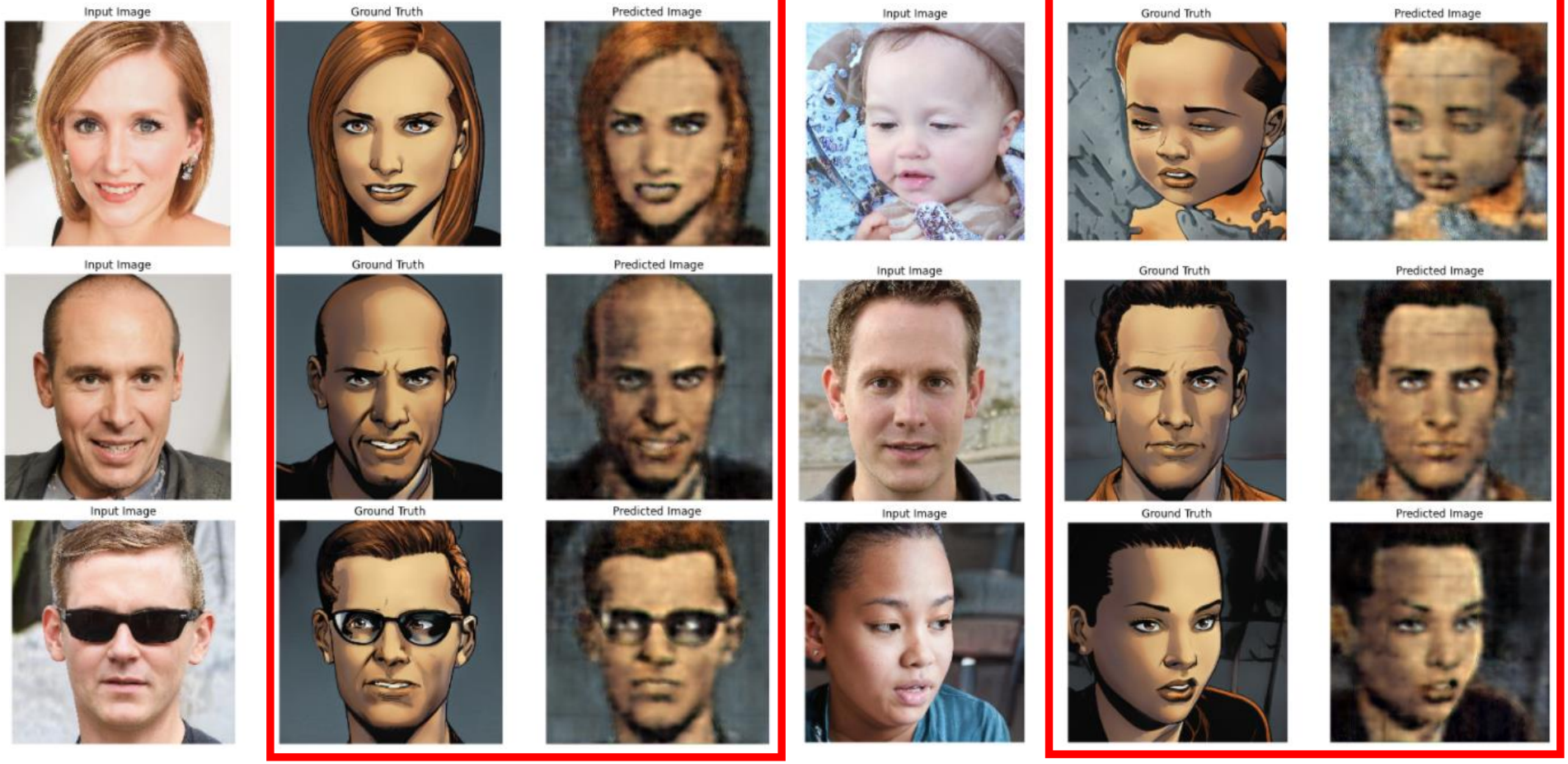
Epoch = 10 일 때, (학습 후)

MSE, PSNR, SSIM 값 696.8523634504527(MSE), 19.699395835308376(PSNR), 0.6852129498172398(SSIM)

04 Comic 결과정리

Pix2pix 및 Autoencoder 각각에 대한 최종적으로 도출된 사진과 MSE, PSNR, SSIM 값을 도출하였습니다.
Learning rate = 2e-4, beta_1 = 0.5, epoch = 100 을 공통적으로 적용하였습니다.

2) Autoencoder 결과 정리(추가 결과 사진)



05 전체 결과정리

Pix2pix 및 Autoencoder 모델에 대한 MSE, PSNR, SSIM 값을 표로 정리한 내용입니다.

평가지표

	MSE		PSNR		SSIM	
	PIX2PIX	AUTO ENCODER	PIX2PIX	AUTO ENCODER	PIX2PIX	AUTO ENCODER
Coloring	86.2874	510.7944	28.7713	21.0483	0.9982	0.8416
Monet	456.9418	551.0595	21.5321	20.7188	0.8424	0.6521
Comic	519.8645	696.8523	20.9719	19.6993	0.8835	0.6852

모든 평가 지표 측면에서 pix2pix의 성능이 좋게 나왔습니다.

주관적 평가

평가지표와 육안으로의 평가에 차이가 생길 수 있으나,
육안으로 보기에도 pix2pix를 통해 나온 결과물이 Autoencoder보다 화질이 좋았습니다.

한계점

- 학습데이터셋을 늘려 모델 평가 높이기
- 모델 구조 및 파라미터 수정을 통해 모델을 강화하기