Activity #5

Image Enhancement with AutoEncoder

Agenda

- 5.1 Data Preparation
- 5.2 Adding Noise
- 5.3 Autoencoder

Libraries

1. import numpy as np
2. import cv2
3. from matplotlib import pyplot as plt
4. from keras.models import Model, Input
5. from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, UpSampling2D
6. from tensorflow.keras.callbacks import EarlyStopping
7. from keras.preprocessing import image
8. from sklearn.model_selection import train_test_split
9. Import glob

5.1 DATA PREPARATION

5.1 Data Preparation

1

- # List all filename in face image path
  - filenames= glob.glob ()

2

- # Load image files and Resize (h,w,ch) -> h = w < 100 , ch = 3 (R,G,B)
  - img = image.load_img(fname, target_size, interpolation="nearest")
- # convert the image to an array
  - img = img.img_to_array(img)
- # Normalized color image
  - img = img/255
- all_imgs = img.append()

3

- # convert to numpy array
- all_images = np.array(all_images)

4

- # split data into train and validation data
- train_x, val_x = train_test_split(all_images, random_state=32, test_size=0.3)

5.2 ADD NOISE

4.2 Prepare input image from scratch

1

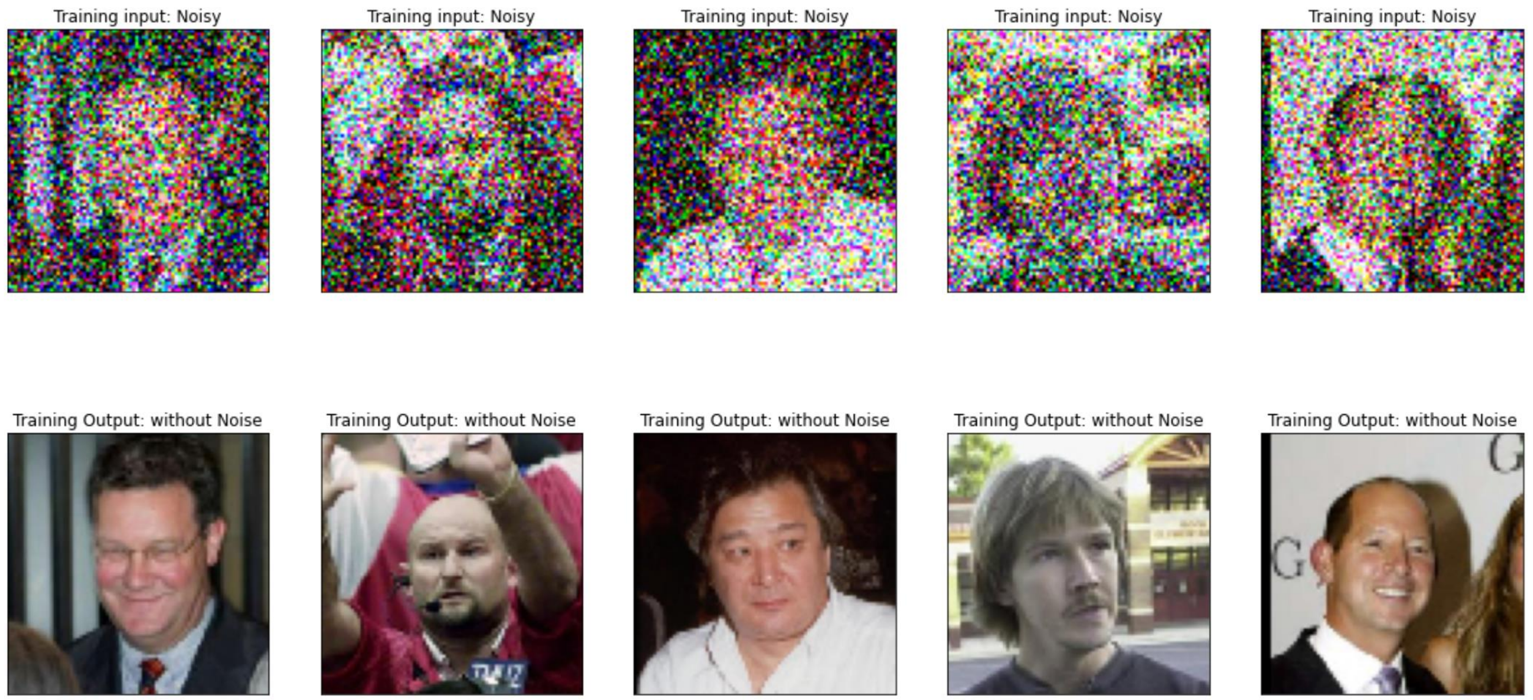
- # Add Noise
  - กำหนด noise factor (noise level)
    - # Add noise
      - noise_factor = scalar between 0 -1 ลองอย่างน้อย 2 ค่า

2

- # กำหนด noise parameter
  - Noise distribution: normal
  - Noise mean: zero mean (Nmean = 0)
  - Noise std: unit variance (Nstd = 1)

3

- # use np.random.normal to generate normal distribution (gaussian) noise
  - x_train_noisy = x_train + (noise_factor * np.random.normal(loc=Nmean, scale=Nstd, size=x_train.shape))
  - x_val_noisy = x_val + (noise_factor * np.random.normal(loc=Nmean, scale=Nstd, size=x_val.shape))


Training input: Noisy
Training input: Noisy
Training input: Noisy
Training input: Noisy
Training input: Noisy
Training Output: without Noise
Training Output: without Noise
Training Output: without Noise
Training Output: without Noise
Training Output: without Noise

5.2

Adding Noise

Choose to plot at least 3 images

5.3 AUTOENCODER MODEL

5.3 Autoencoder Architecture

1

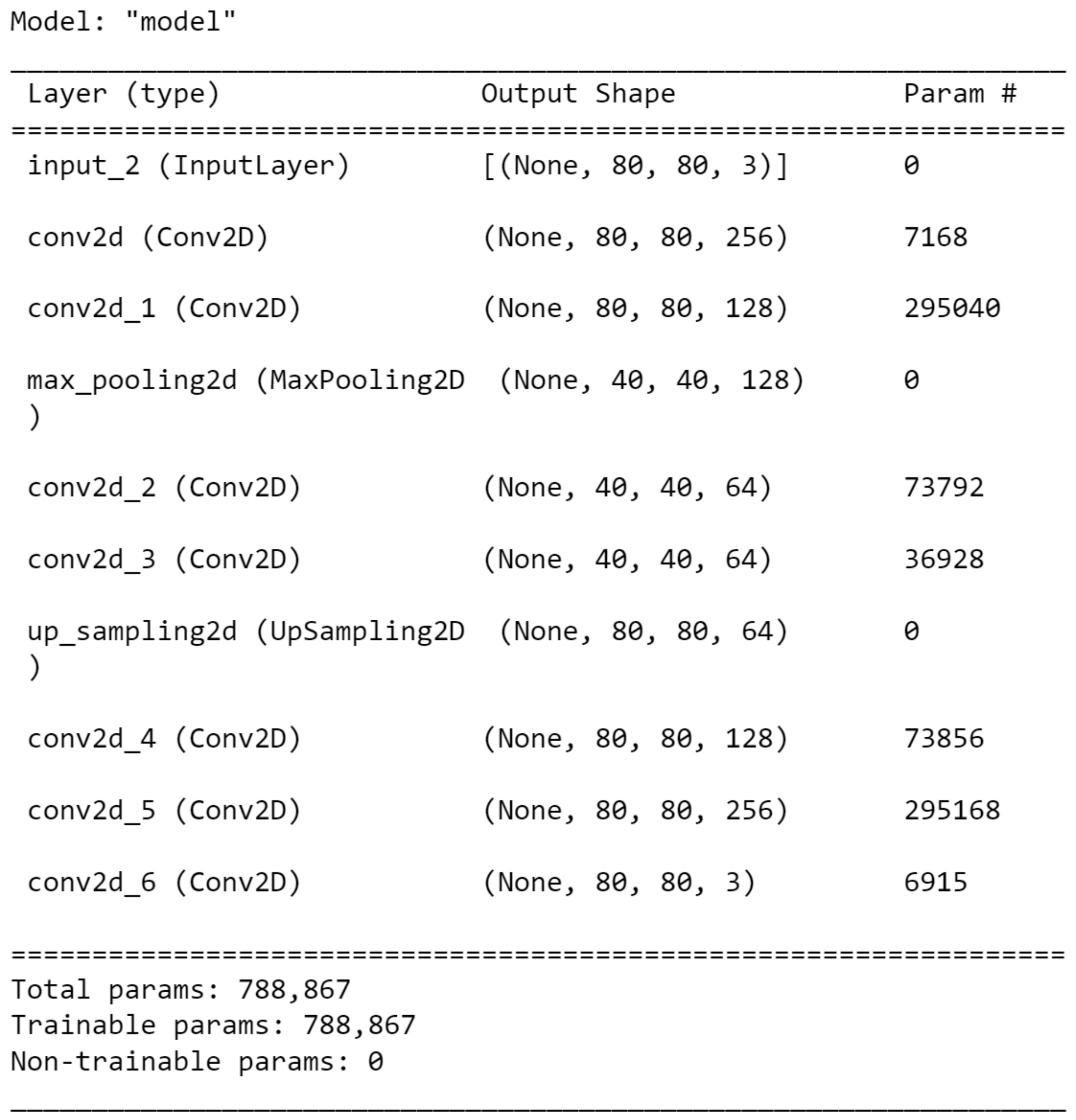
- # กำหนด Encoder Architecture
  - Input_img = Input(shape=(height, width, ch))
  - #encoding architecture
  - x1 = Conv2D(256, (3, 3), activation='relu', padding='same')(Input_img)
  - x2 = Conv2D(128, (3, 3), activation='relu', padding='same')(x1)
  - x2 = MaxPool2D((2, 2))(x2)
  - encoded = Conv2D(64, (3, 3), activation='relu', padding='same')(x2)

2

- # กำหนด Decoder Architecture
  - # decoding architecture
  - x3 = Conv2D(64, (3, 3), activation='relu', padding='same')(encoded)
  - x3 = UpSampling2D((2, 2))(x3)
  - x2 = Conv2D(128, (3, 3), activation='relu', padding='same')(x3)
  - x1 = Conv2D(256, (3, 3), activation='relu', padding='same')(x2)
  - decoded = Conv2D(3, (3, 3), padding='same')(x1)

3

- # กำหนด optimizer setting
  - autoencoder = Model(Input_img, decoded)
  - autoencoder.compile(optimizer='adam', loss='mse’) -> loss ใช้ Mean Square Error
  - autoencoder.summary()



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 80, 80, 3)]       0

 conv2d (Conv2D)             (None, 80, 80, 256)       7168

 conv2d_1 (Conv2D)           (None, 80, 80, 128)       295040

 max_pooling2d (MaxPooling2D  (None, 40, 40, 128)      0
 )

 conv2d_2 (Conv2D)           (None, 40, 40, 64)        73792

 conv2d_3 (Conv2D)           (None, 40, 40, 64)        36928

 up_sampling2d (UpSampling2D  (None, 80, 80, 64)       0
 )

 conv2d_4 (Conv2D)           (None, 80, 80, 128)       73856

 conv2d_5 (Conv2D)           (None, 80, 80, 256)       295168

 conv2d_6 (Conv2D)           (None, 80, 80, 3)         6915

=================================================================
Total params: 788,867
Trainable params: 788,867
Non-trainable params: 0
_________________________________________________________________
```

5.2

Autoencoder Model

5.3 Training Autoencoder Model

1

- # กำหนด Training parameter
  - epoch ทดลองอย่างน้อย 2 ค่า [2, 3, 4]
  - batch_size ทดลองอย่างน้อย 1 ค่า [8, 16, 32]

2

- # เริ่มการ training
  - history = autoencoder.fit (x_train_noisy, train_x,
  - epochs=2,
  - batch_size=32,
  - shuffle=True,
  - validation_data=(x_val_noisy, val_x),
  - callbacks=[early_stopper])

3

- # ทดสอบ autoencoder model ด้วย validation
  - predictions = autoencoder.predict(x_val_noisy)

4

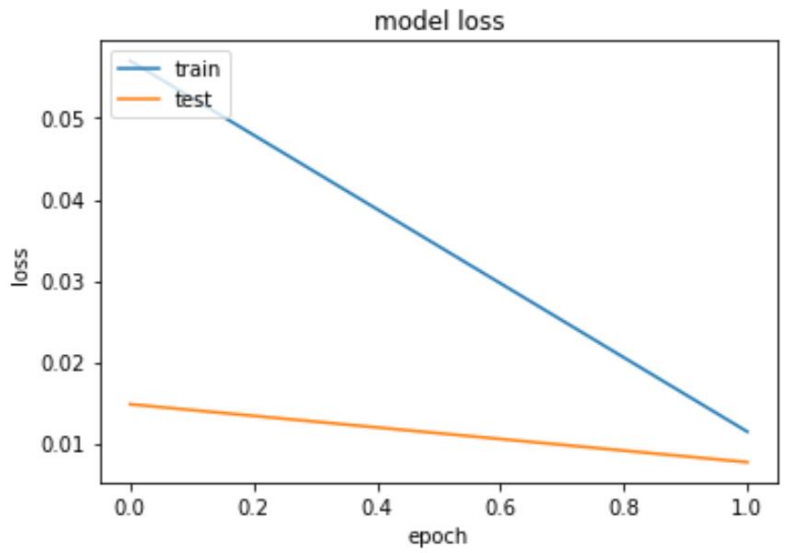
- View Loss from history
  - plt.plot(history.history['loss'])
  - plt.plot(history.history['val_loss'])
  - plt.title('model loss') plt.ylabel('loss')
  - plt.xlabel('epoch')
  - plt.legend(['train', 'test'], loc='upper left')
  - plt.show()

5

- # Display Result image

5.3

Autoencoder Model


model loss
train
test
loss
epoch