

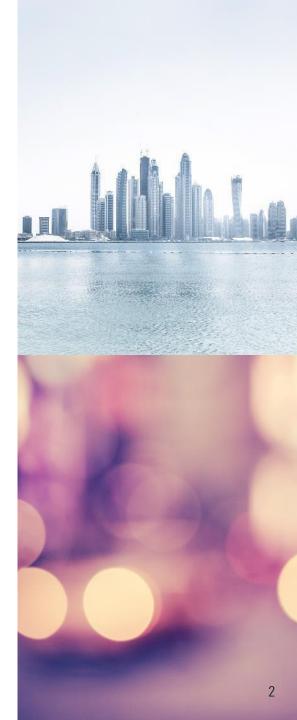
LAB #5

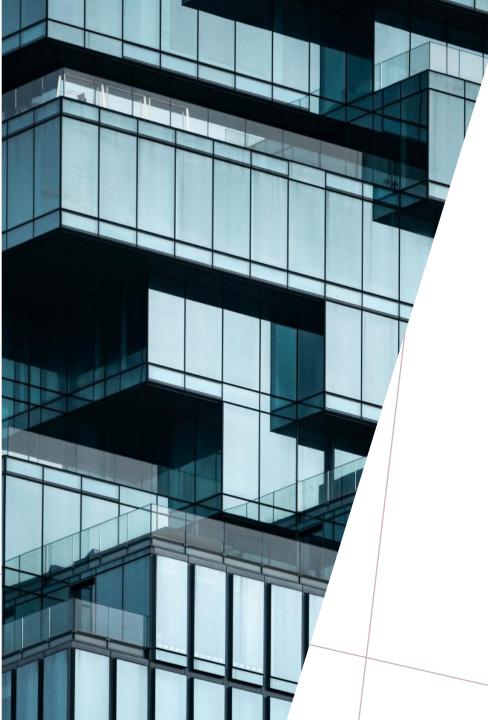
การทคลอง

5.1 Effects of interpolation techniques in Image Resizing

5.2 Image Augmentation using KERAS ImageDataGenerator

5.3 Training Model using ImageDataGenerator





5.1 EFFECTS OF
INTERPOLATION TECHNIQUES
IN IMAGE RESIZING

LIBRARIES

- # Import Libraries
- import cv2
- import numpy as np
- from matplotlib import pyplot as plt
- from keras import Model, Input
- import keras.utils as image
- from keras.wrappers.scikit_learn import KerasRegressor
- from tensorflow.keras.preprocessing.image import ImageDataGenerator
- from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, UpSampling2D
- from tensorflow.keras.callbacks import EarlyStopping
- from tensorflow.keras import optimizers
- from tensorflow.keras.datasets import fashion_mnist
- from sklearn.model_selection import train_test_split

5.1 Effects of interpolation techniques in Image Resizing

1

- # Read image file
 - imread("Grid Image.jpg")

- # Define resize factor
 - Reduce_factors = [2, 4, 5, 7, 8, 10, 15] # อย่างน้อย 3 ค่า
 - Scale_factors = 1/ Reduce_factors
- # Define interpolation method
 - inter methods = [cv2.INTER NEAREST,cv2.INTER LINEAR, cv2.INTER CUBIC, cv2.INTER AREA]

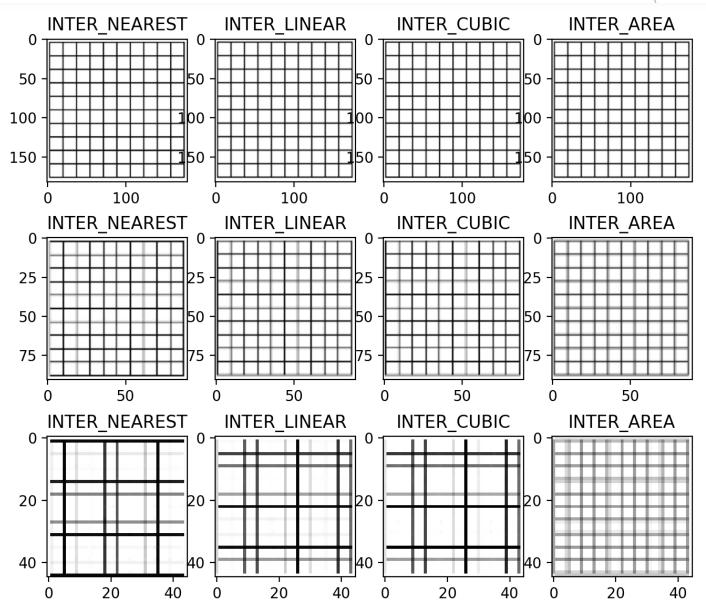
2

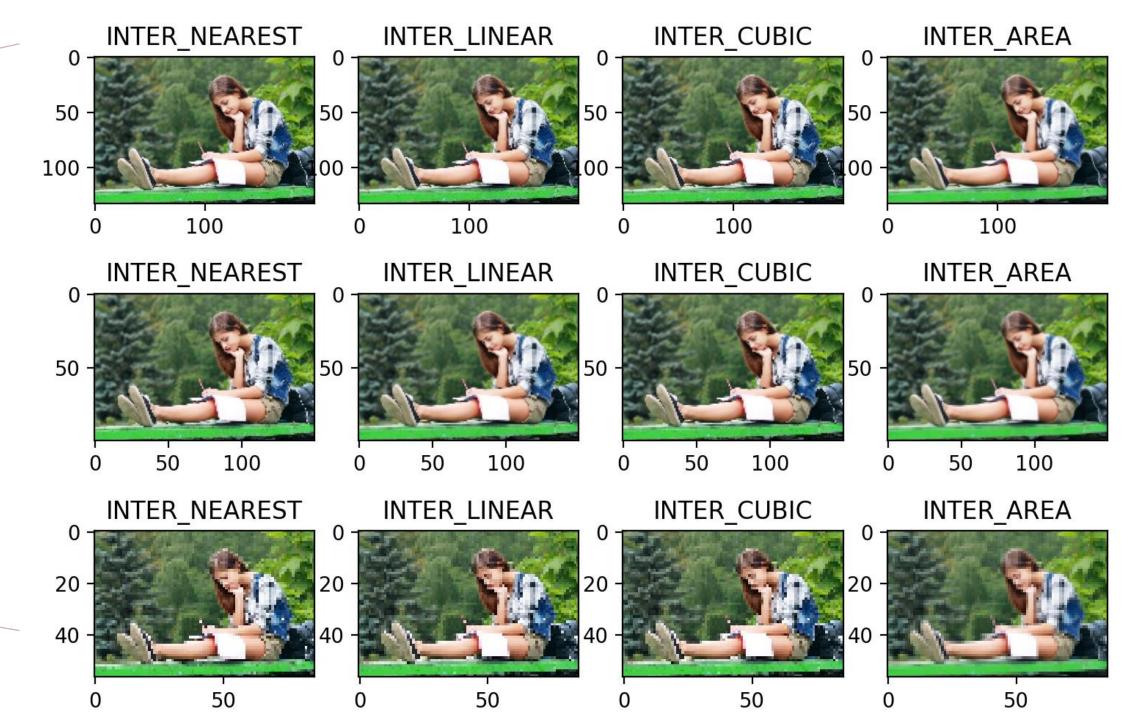
- Display results using each scale factors & interpolation methods
 - Cv2.resize(Scale factors, inter methods)

3

50 -20

5.1 IMAGE RESIZE







5.2 IMAGE AUGMENTATION USING KERAS IMAGEDATAGENERATOR

Read image

Imread()

• # Define fill method

fill method = ['constant', 'nearest', 'reflect', 'wrap']

•# Define parameters

Npic / rotation_range / width_shift_range / height_shift_range / shear_range / zoom_range / horizontal_flip

Prepare Gaussian Noise Function

def add_gaussian_noise(img)

#img_noisy = img+gaussion_noise

return img_noisy

4

7.2 IMAGE AUGMENTATION USING KERAS IMAGEDATAGENERATOR

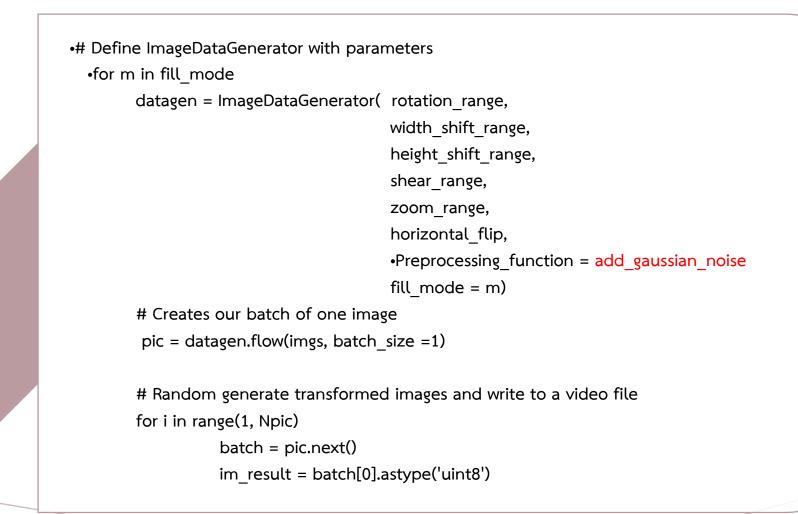




IMAGE DATA GENERATOR

(PREPROCESSING_FUNCTION=NONE)

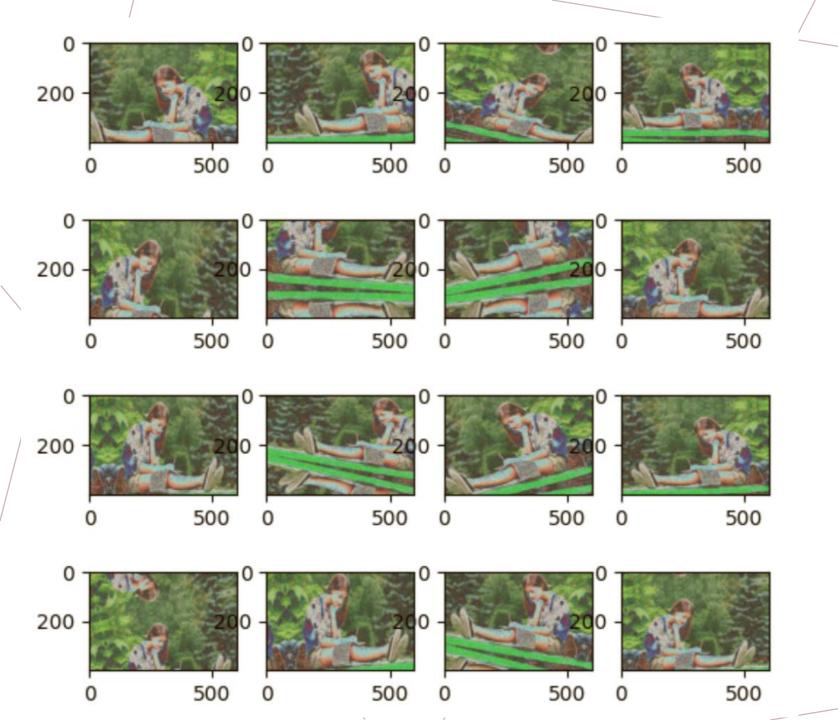


IMAGE DATA GENERATOR

(PREPROCESSING_FUNCTION=ADD_NOISE)



5.3 TRAINING AUTOENCODER USING IMAGE AUGMENTATION AND ADD_NOISE

Read image from fashion_mnist dataset

• (x_train, _), (x_test, _) = fashion_mnist.load_data()

• x_train = x_train/255. x_test =x_test/255.

x_train, x_test= train_test_split(x_train, random_state, test_size)

x_train, x_val = train_test_split(x_train, random_state, test_size)

load images from dataset

normalized image intensity

create train data, test data

create train data, validation data

Prepare Gaussian Noise Function

- add_gaussian_noise
 - กำหนด noise (mean, std, noise_factor)

•# Define ImageDataGenerator with parameters

datagen = ImageDataGenerator(rotation range,

width_shift_range,

height shift range,

shear range,

zoom range,

horizontal flip,

fill mode = m

Preprocessing function = add gaussian noise)

2

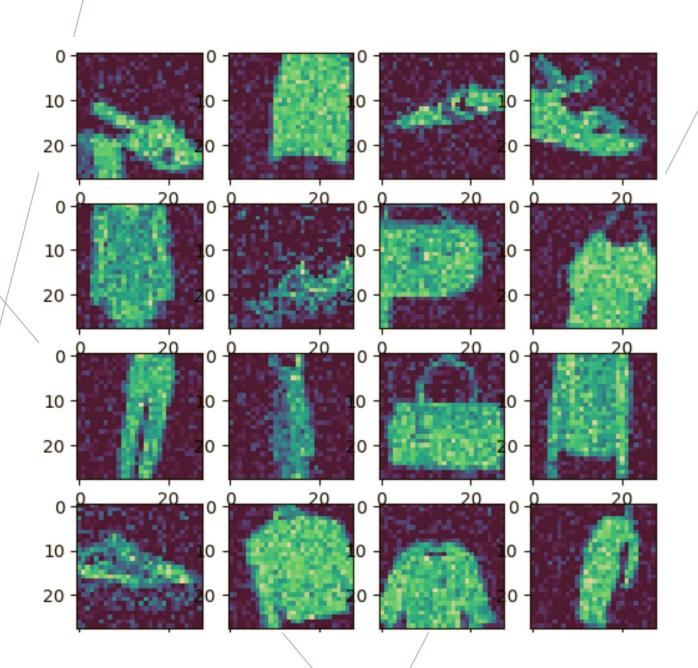


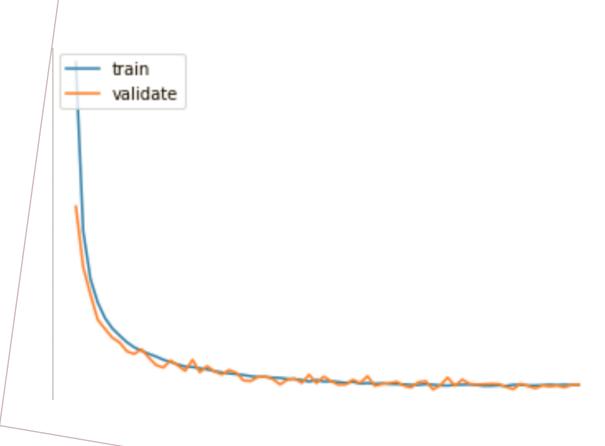
IMAGE DATA GENERATOR

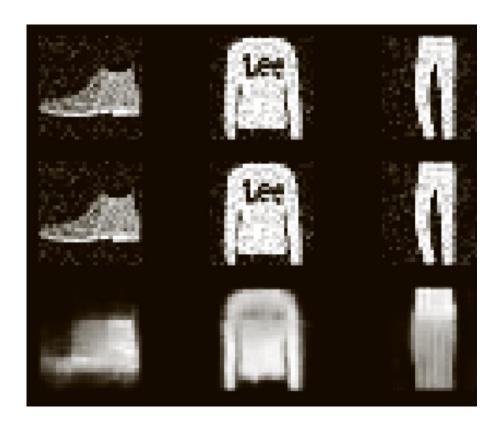
(PREPROCESSING_FUNCTION=ADD_NOISE)

5.3 TRAINING AUTOENCODER USING IMAGE AUGMENTATION WITH NOISE

- # construct the autoencoder model
- autoencoder = create_model(optimizer=opts, learning_rate=lr)
 - callback = EarlyStopping(monitor='loss', patience=10)
 - history = autoencoder.fit_generator(datagen.flow(x_train, x_train, batch_size=bs),
 - epochs=eps, steps_per_epoch = x_train.shape[0]//bs,
 - validation_data=datagen.flow(x_val, x_val, batch_size=bs),
 - callbacks=[callback],verbose=1)
 - # test batch_size , epoch, learning_rate,
- Plot graph to show training and validation loss
- •predict_test = autoencoder.predict(x_test_noisy)
- •Imshow() to see prediction images

EPOCH=70





Sample Footer Text

เงื่อนไขการทดลอง

- ทดสอบ noise std, noise_factor อย่างน้อย 2 คู่
- ทดลองค่า epoch, batch_size, learning_rate, จำนวน convolution node เลือกอย่างละ 2 แบบ เป็นอย่างน้อย
- แสดงผลลัพธ์ในรูปแบบ
 - Graph Training vs Validation Loss
 - รูปภาพ เปรียบเทียบ Original, Noisy, and Predicted images จำนวนอย่างละ 5 ภาพ เป็นอย่าง น้อย

Sample Footer Text 2/7/20XX