ECS795P Deep Learning and Computer Vision, 2018

Course Work 1: Image Super-resolution Using Deep Learning

Introduction

Aim: To obtain practical knowledge and hands-on understanding of the concepts of image super-resolution, deep learning using convolutional neural networks (CNN) and peak signal-to-noise ratio (PSNR).

Start: Download **CW1_ECS795P.zip** from the course website at http://www.eecs.qmul.ac.uk/~sgg/_ECS795P_/.

Tasks: three subtasks are involved:

1. Coding: to add your code blocks in the required sections; (40% of this CW)
2. Demonstrating: to answer THREE questions randomly selected from below during the lab demo session in WK11; (30% of this CW)
3. Report: to complete the questions in report. (30% of this CW)

Platform: Tensorflow

1. Understanding image super-resolution

Objective: To become familiar with the image super-resolution problem setting.

The questions to think over:

1. What are the concepts of image size and image resolution?
2. What is gray-scale or single-channel image super-resolution?
3. What is Ground Truth image?
4. How to measure the quality of the output high-resolution images?

The excises to conduct:

1. To read the image named *butterfly_GT.bmp*
2. To show the size of this image
3. To convert the image from the RGB colour space into the gray-scale space (Tip: use *scipy* package to read image, set a *breakpoint* to watch the image values, *import pdb, pdb.set_trace()*)
4. To shrink the current image by 3 times with bicubic interpolation algorithm (Tip: *imresize* is a Tensorflow build-in function)
5. To enlarge the current image by 3 times with bicubic interpolation algorithm

2. Understanding deep learning by convolutional neural network

Objective: To understand the principles of deep convolutional network.

The questions to think over:

1. What are the parameters of a CNN?
2. What is the target of CNN model training?
3. What is the actual behaviour in testing stage with a CNN?
4. What is feature map?
5. How to perform convolution filtering?

The excises to conduct:

1. To load the pre-trained model named *model.npy*
2. To set and show the weights of the first convolutional layer
    - To set the filter number
    - To set the filter size
    - To show the value of the 1st filter in command window
    - To show the bias of the 10th filter in command window
3. To set and show the weights of the second convolutional layer
    - To set the filter number
    - To set the filter size
    - To show the value of the 5th filter in command window
    - To show the bias of the 6th filter in command window
    - To show the channel number of the input
4. To set show the weights of the third convolutional layer
    - To set the filter number
    - To set the filter size
    - To show the value of the 1st filter in command window
    - To show the bias of the 1st filter in command window
5. To perform 2-d convolution filtering on a 2-d matrix with a given filter (Tip: *conv2d and relu* is a Tensorflow build-in function)

3. Image super-resolution using deep convolutional network

Objective: To perform image super-resolution with deep convolutional neural network and evaluate its performance.

The questions to think over:

1. How to use a trained SRCNN to perform image super-resolution (testing stage)?
2. What are the input and the output of SRCNN?
3. How to conduct qualitative and quantitative comparison between two different methods?
4. What is the typical numerical measure metric for quantitative analysis?
5. What is the maximum power of imaging signal (i.e. pixel) and noise signal, e.g. image of uint8 type?

The excises to conduct:

1. To get and show the Ground Truth image
2. To get and show the low-resolution image
3. To get the input image into the SRCNN
4. To call SRCNN to super-resolve the input image
5. To get and show the output high-resolution image by SRCNN
6. To compute the PSNR of the high-resolution image against the Ground Truth image
7. To get the high-resolution image with bicubic interpolation algorithm (baseline result)
8. To compute the PSNR of the baseline result (Tip: use the python module: *skimage.meause.compare_psnr*)
9. To compare the results of the two methods in terms of PSNR