IT117 Lab Report (Module 8)

Purpose of this Lab:

Computer Vision Applications with Python

Software tools:

NumPy, SciPy, and PIL/PILLOW

Time: 40 min

Introduction

The Python Imaging Library (PIL) adds image processing capabilities to your Python interpreter. This library supports many file formats, and provides powerful image processing and graphics capabilities.

Install and configuration

Mac/Linux users

Linux Anaconda users have build-in support for PIL and Pillow. If you are using Mac and have Xcode, things should be extremely easy.

1) Run xcode install, use this command or download updates from the app store:

xcode-select --install

2) Install everything needed for python (using brew), I believe you can use port as well:

brew install readline sqlite gdbm

brew install python --universal --framework

brew install libpng jpeg freetype

Unlink/ relink if needed i.e. if upgrading.

3) Install Pip and required modules:

easy\_install pip

sudo pip install setuptools --no-use-wheel --upgrade

4) Finally this works with no errors:

sudo pip install Pillow

Make sure you have installed numpy, scipy, pil/pillow

Check it by typing import each modules to see if there is any error message popping out. If not, you are done.

Windows users

Install numpy, scipy, and pillow should be easy.

Follow the instructions provided by the module website.

Image Processing

Finishing the following code using

1, Image load and show, conversion

>>> from PIL import Image

>>> imnew = Image.open('Images/fox.jpg')

>>> imnew.show()

>>> imnew = Image.open('Images/fox.jpg').convert('L')

>>> imnew.show()

//paste your output here





2, Image resize and rotate

To resize an image, call resize() with a tuple giving the new size

>>> out = imnew.resize((128,128))

>>> out.show()

to rotate an image, call counter clock angles and rotate() like this

>>> out = imnew.rotate(45)

>>> out.show()

//paste your output here



3, Image Blur and Sharpen

Blur is a comely used in image processing. The (grayscale) image I is convolved with a Gaussian kernel to create a blurred version

3.1 Using PIL ImageFilter() method

>>> from PIL import ImageFilter

>>> im = Image.open('Images/fox.jpg')

>>> im = im.filter(ImageFilter.BLUR)

>>> im.show()

>>> im = Image.open('Images/lena.jpg')

>>> im = im.filter(ImageFilter.SHARPEN)

>>> im.show()

//paste your output here

My SHARPENED IMAGE

  
MY Blurry Image



Note: Windows user may have difficulty visualizing the result. The reason is MS paint bug. The solution can be found here:

<http://stackoverflow.com/questions/7715501/pil-image-show-doesnt-work-on-windows-7>

You need to save the result first then use webbrowser to show it.

To save the image you need to learn a little bit more on image representation. A quick step-by-step tutorial can be found here

<http://stackoverflow.com/questions/14452824/how-can-i-save-an-image-with-pil>

3.2 Using numpy method and scipy.ndimage filters.

>>> from PIL import Image

>>> from numpy import \*

>>> from scipy.ndimage import filters

>>>

>>> im = array(Image.open('Images/plate.jpg').convert('L'))

>>> imnew = filters.gaussian\_filter(im, 5)

>>> pil\_im = Image.fromarray(imnew)

>>> pil\_im.show()

>>>> pil\_im.save('Images/blurredplates.jpg')

//paste your output here



3.3 Sharpen an image and then blur it

Sometimes we want to sharpen the image as well. ImageFilter provides two useful methods. Blur and sharpen, note that they both are capsized.

>>> im = Image.open('Images/house.jpg')

>>> im.show()

>>> im = im.filter(ImageFilter.SHARPEN)

>>> im.show()

>>> im = im.filter(ImageFilter.BLUR)

>>> im.show()

//paste your output here

original Image first





BLUR



3.4 Image filtering with different direction filters

>>> from PIL import Image

>>> from numpy import \*

>>> from scipy.ndimage import filters

>>>

>>> im = array(Image.open('Images/lena.jpg').convert('L'))

>>> imx = zeros(im.shape)

>>> filters.sobel(im, 1, imx)

>>>

>>> imy = zeros(im.shape)

>>> filters.sobel(im, 0, imy)

>>> imnew = Image.fromarray(imx)

>>> imnew.show()

>>> imnew = Image.fromarray(imy)

>>> imnew.show()

>>> magnitute = sqrt(imx\*\*2 + imy\*\*2)

>>> imnew = Image.fromarray(magnitute)

>>> imnew.show()

>>> im = Image.open('Images/lena.jpg').convert('L')

>>> im.show()

//paste your output here





