# Image segmentation Report

The code uses python 2.7 version SDK and interpreter, OpenCV 3.0.0 and NumPy.

The zip file submitted contains:

image-segmentation.py (code file)

image1.jpg, image2.jpg, image3.jpg, image4.jpg, image5.jpg (input images)

Report\_PartIII.txt

Readme.txt

k=20 and k=5 folder contains some experiment results

To Compile and run the program:

The values of k is assigned to 20. The folder Experimental results consists of output of images when value of k was 20.

**SETUP:**

Ensure that PyCharm 2.7, OpenCV 3.0.0 and NumPy 1.6.1 are installed on the system.

For ubuntu terminal, follow thses steps:

http://www.pyimagesearch.com/2015/06/22/install-opencv-3-0-and-python-2-7-on-ubuntu/

Installation steps was followed as above. Post installation type as follows:

$workon cv

$python

>>> import cv2

>>> cv2.\_\_version\_\_'3.0.0'

For windows terminal, follow these steps:

http://stackoverflow.com/questions/32660114/how-to-install-opencv-on-windows-and-enable-it-for-pycharm-without-using-the-pac/32664411#32664411

Steps to follow:

1. Install Python 2.7.10
2. Install Pycharm(If you have not done it already)
3. Download and install the OpenCV executable.
4. Add OpenCV in the system path(%OPENCV\_DIR% = /path/of/opencv/directory)
5. Goto C:\opencv\build\python\2.7\x86 folder and copy cv2.pyd file.
6. Goto C:\Python27\DLLs directory and paste the cv2.pyd file.
7. Goto C:\Python27\Lib\site-packages directory and paste the cv2.pyd file.
8. Goto PyCharm IDE and goto DefaultSettings>PythonInterpreter.
9. Select the Python which you have installed on Step1.
10. Install the packages numpy,matplotlib and pip in pycharm.
11. Restart your PyCharm.
12. PyCharm now has OpenCV library installed and working.

**EXECUTION:**

1. Once all the setup is complete, do the below step for executing the code.

2. Open the terminal and set the directory to the one containing the python code and input files.

3. Type the following command on the command line to execute the python code:

python image-segmentation.py <clustering k value> <input image name with file extension> <output file name with extension>

example: python image-segmentation.py 20 image4.jpg output\_image4.png

**REPORT:**

Language: Python version 2.7

Packages used: cv2, numpy, pip, sys

The experiments conducted for the sample five images followed the process below:

All 5 images were clustered for k=5 and k=20. The resulting images are tabulated in the table below: as we see the clustering results with k as 20 are more defined than those at k as 5.

The experimental results folder contains some of the observed changes when value of k was kept at 20.

The difference in the quality in both the k=5 and k=20 is significant.

|  |  |
| --- | --- |
| **Clustering k= 5** | **Clustering k=20** |
|  |  |
| **Clustering k=5** | **Clustering k=20** |
|  |  |
|  |  |

Clustering K=5

Clustering k=20



**Conclusion:**

We see how the change in value of k changes the output we receive and the changes in the result image due to clustering are significant.

Thus clustering can show us the different features of the same image and can be used for investigation and research.