

Assignment 6

Thresholding and Segmentation

1. **Otsu's Thresholding:** Apply Otsu's thresholding to the image 'noisy_leaf.jpg' to obtain the optimal threshold value to separate the foreground and background effectively as shown below. (Apply gaussian blur before thresholding). Save the thresholded image.



2. **Adaptive thresholding:** The image 'page.jpg' has illumination variation across it. Apply both 'global' and 'local' thresholding and compare and save the resulting images.
3. **Segmentation with Grabcut:** OpenCV samples (../OpenCV/samples/python/) contain a sample **grabcut.py** which is an interactive tool using grabcut. Use it to segment Messi out of the image 'messi.jpg' and save the result.
youtube video on using the tool: <https://www.youtube.com/watch?v=kAwxLTDDAwU>
4. **K-means clustering for color quantization:** Apply k-means clustering to the image 'building.png' to cluster different color defined regions together. Here the value of 'k' determines the number of colors to which the image is quantised. Save the results of clustering with k=3, 5 and 8.

For example, with input image:



K = 3 results in :



And K = 8 results in :



These questions can be easily tried out with the help of the following references:

1. <https://www.meccanismocomplesso.org/en/opencv-python-the-otsus-binarization-for-thresholding/>
2. https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html
3. https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_grabcut/py_grabcut.html
4. https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_ml/py_kmeans/py_kmeans_opencv/py_kmeans_opencv.html