**Image processing: Thresholding**

Thresholding is a kind of image segmentation method which is used to simplify the image( or video ) by pixels manipulation.

In thresholding, we convert the image which can either be colored or in greyscale into binary image( only black and white color is used to represent any pixel ). But in computer vision, thresholding is done on grayscale images only. So initially, the image has to be converted in grayscale color space.

Every pixel will either have 0 to represent black or maximum value to represent white color.

Two methods are used in openCV for thresholding:

1. Simple thresholding
2. Adaptive thresholding

Let’s discuss them one by one:

Simple thresholding:

In this technique, the same threshold value is applied for every pixel.

cv2.threshold(source, thresholdValue, maxVal, thresholdingTechnique) function is used to perform simple thresholding with openCV. The function has four arguments:

1. source: it signifies the source image which has to be processed.
2. thresholdValue: the value of threshold.
3. maxVal: it signifies the maximum value which can be assign to a pixel.
4. thresholdingTechnique: the type of thresholding to be performed. There are five simple thresholding technique:
5. cv2.THRESH\_BINARY
6. cv2.THRESH\_BINARY\_INV
7. cv2.THRESH\_TRUNC
8. cv2.THRESH\_TOZERO
9. cv2.THRESH\_TOZERO\_INV

Python code:

#importing libraries

import cv2

img = cv2.imread('flower.jpg',0);

#different thresholding techniques on the image

retval, thresh1 = cv2.threshold(img, 125, 255, cv2.THRESH\_BINARY);

retval, thresh2 = cv2.threshold(img, 125, 255, cv2.THRESH\_BINARY\_INV);

retval, thresh3 = cv2.threshold(img, 125, 255, cv2.THRESH\_TRUNC);

retval, thresh4 = cv2.threshold(img, 125, 255, cv2.THRESH\_TOZERO);

retval, thresh5 = cv2.threshold(img, 125, 255, cv2.THRESH\_TOZERO\_INV);

cv2.imshow('ORIGINAL', img);

cv2.imshow('THRESH\_BINARY', thresh1);

cv2.imshow('THRESH\_BINARY\_INV', thresh2);

cv2.imshow('THRESH\_TRUNC', thresh3);

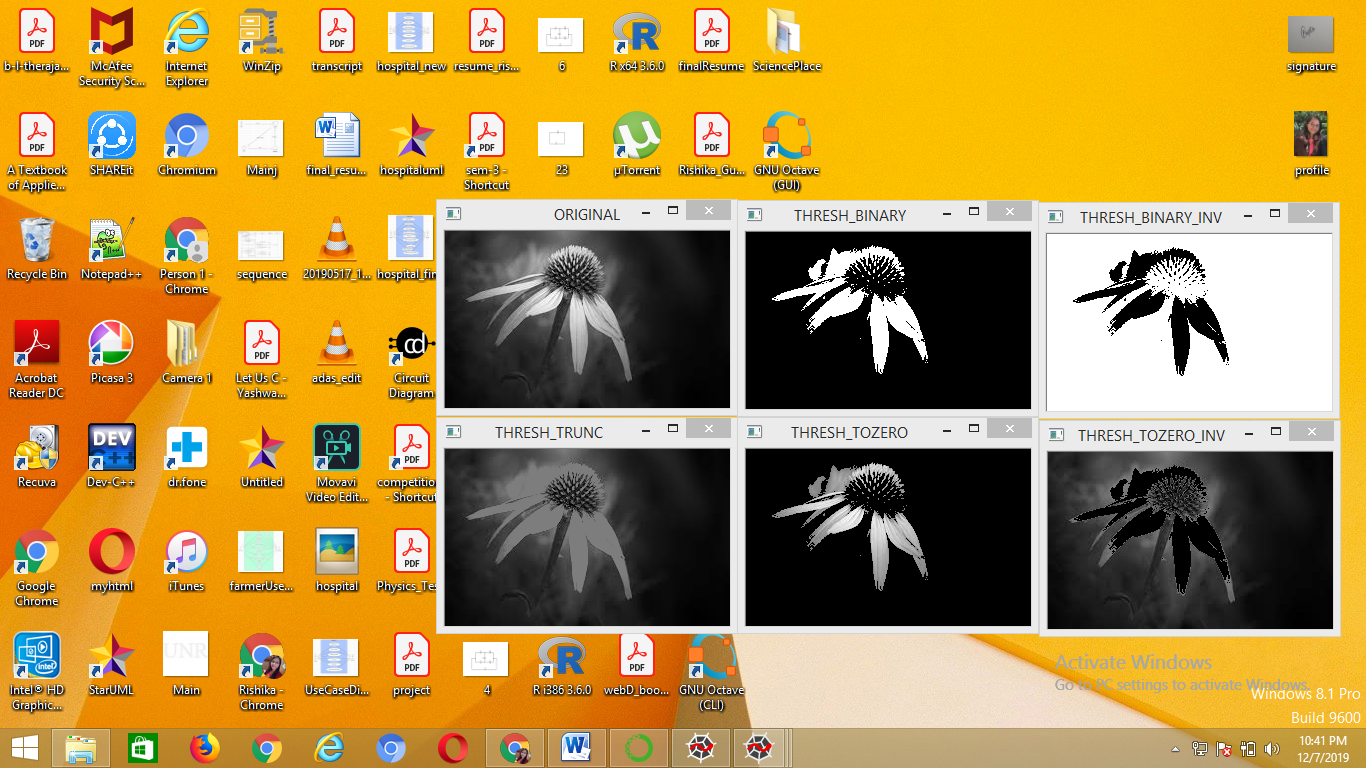
cv2.imshow('THRESH\_TOZERO', thresh4);

cv2.imshow('THRESH\_TOZERO\_INV', thresh5);

#wait for 10 seconds

cv2.waitKey(10000);

cv2.destroyAllWindows();

Output:

Adaptive thresholding:

In this technique, we can have different threshold values for different regions.

cv2.adaptiveThreshold(source, maxVal, adaptiveMethod, thresholdingTechnique, blocksize, constant) function is used in openCV to perform adaptive thresholding. The function has six arguments:

1. source: it signifies the source image which has to be processed.
2. maxVal: it signifies the maximum value which can be assign to a pixel.
3. adaptiveMethod: it signifies the way to calculate threshold value. There are two ways for calculating threshold value:
4. cv2.ADAPTIVE\_THRESH\_MEAN\_C
5. cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C
6. thresholdingTechnique: the type of thresholding to be performed. There are five simple thresholding technique:
7. cv2.THRESH\_BINARY
8. cv2.THRESH\_BINARY\_INV
9. cv2.THRESH\_TRUNC
10. cv2.THRESH\_TOZERO
11. cv2.THRESH\_TOZERO\_INV
12. blocksize: it signifies the size of region used to calculate threshold value.
13. constant: it signifies the constant value which subtracted from the mean or weighted mean.

Python code:

#importing libraries

import cv2

img = cv2.imread('flower.jpg',0);

thresh10 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C,cv2.THRESH\_BINARY, 101, 2);

thresh11 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C,cv2.THRESH\_BINARY\_INV, 101, 2);

thresh21 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C,cv2.THRESH\_BINARY, 101, 2);

thresh22 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C,cv2.THRESH\_BINARY\_INV, 101, 2);

cv2.imshow('ORIGINAL', img);

cv2.imshow('BINARY\_ADAPTIVE\_THRESH\_MEAN\_C', thresh10);

cv2.imshow('BINARY\_INV\_\_ADAPTIVE\_THRESH\_MEAN\_C', thresh11);

cv2.imshow('BINARY\_ADAPTIVE\_THRESH\_GAUSSIAN\_C', thresh21);

cv2.imshow('BINARY\_INV\_ADAPTIVE\_THRESH\_GAUSSIAN\_C', thresh22);

#wait for 10 seconds

cv2.waitKey(10000);

cv2.destroyAllWindows();

Output:

