Module 6

Segmentation

▼ Why do we need segmentation?

For an image to be interpreted accurately, pixels must be divided into groups that correspond to objects or parts of an object in that image.

▼ What is a region?

a group of connected pixels sharing similar characteristics

▼ What is image segmentation?

the process in which regions are identified and classified as belonging together

▼ What is thresholding?

- Simple thresholding works on the assumption that the pixels in an image that fall within a certain range of intensity values are assigned to one class and all the remaining pixels are assigned to the other class.
- If a pixel value is greater than a threshold value, it is assigned one value (perhaps white), else it is assigned another value (perhaps black).

▼ What is region-based segmentation?

- Regions are constructed by associating (or dissociating) neighboring pixels.
- Homogeneity of regions in characteristics such as gray level, color, texture, and shape is used as the grouping criteria in region-based segmentation.
- Each pixel is compared to its neighboring pixel using the homogeneity criteria. If the result is positive then that particular pixel is added to the region.
- There are three region-based segmentation methods:
  - region growing
  - region split and merge

        - watershed

▼ How does OpenCV do thresholding?

- OpenCV has a function **cv2.threshold()** to apply simple thresholding to an image.
- Thresholding is done on grayscale images.

▼ What are some thresholding methods in OpenCV?

- cv2.THRESH_BINARY
    - If the pixel value is greater than the set threshold, its value is set to 255 (white). Otherwise, its value is set to 0.
- cv2.THRESH_BINARY_INV
    - If the pixel value is greater than the set threshold, its value is set to 0 (black). Otherwise, its value is set to 255.
- cv2.THRESH_TRUNC
    - If the pixel value is greater than the set threshold, its value is set to be the same as the threshold value. All other pixel values remain the same.
- cv2.THRESH_TOZERO
    - If the pixel value is less than the set threshold, its value is set to 0. All other pixel values remain the same.
- cv2.THRESH_TOZERO_INV
    - If the pixel value is greater than the set threshold, its value is set to 0. All other pixel values remain the same.

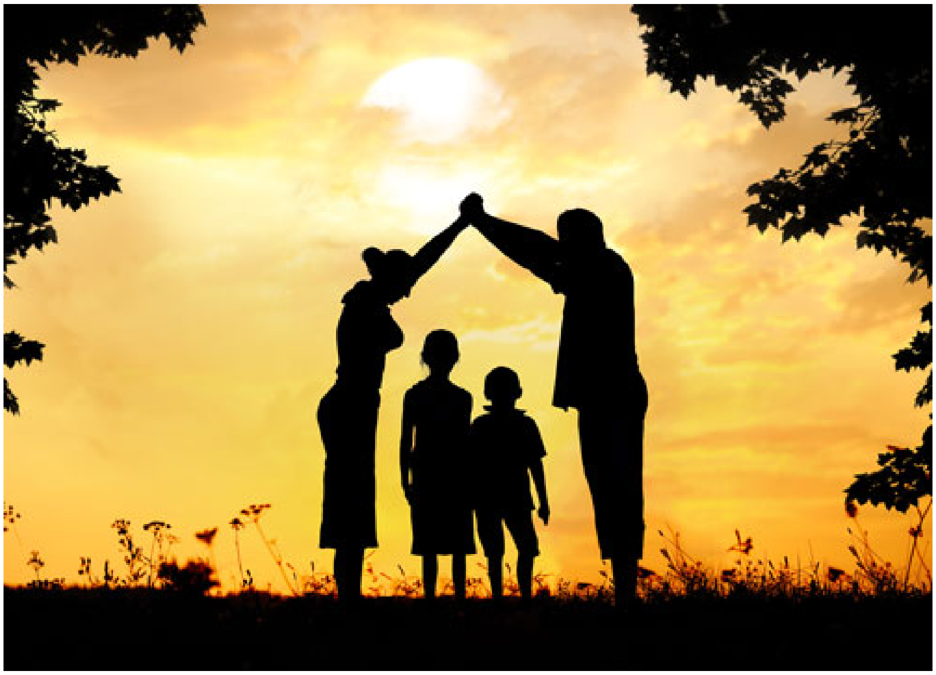
▼ OpenCV thresholding example

- The following code reads in the image and then applies the different simple thresholding methods.
- For each application, 120 is used as the threshold value. This is a global threshold value.

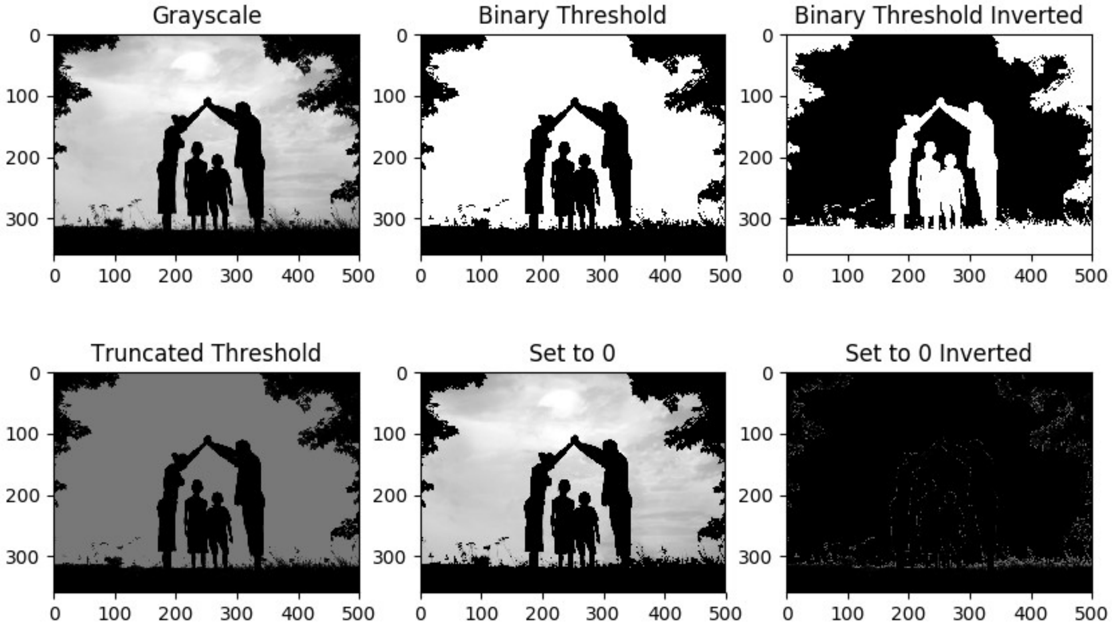
```
import cv2
import matplotlib.pyplot as plt

original = cv2.imread('family.jpg')

# to convert the image in grayscale
```

```
img = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)

# applying different thresholding techniques on the input image
# all pixels value above 120 will be set to 255
ret, thresh1 = cv2.threshold(img, 120, 255, cv2.THRESH_BINARY)
ret, thresh2 = cv2.threshold(img, 120, 255, cv2.THRESH_BINARY_INV)
ret, thresh3 = cv2.threshold(img, 120, 255, cv2.THRESH_TRUNC)
ret, thresh4 = cv2.threshold(img, 120, 255, cv2.THRESH_TOZERO)
ret, thresh5 = cv2.threshold(img, 120, 255, cv2.THRESH_TOZERO_INV)

fig, ((ax1, ax2, ax3), (ax4, ax5, ax6)) = plt.subplots(nrows = 2, ncols = 3, figsize = (10,10))

# when showing images in matplotlib, convert image from BGR to RGB
ax1.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
ax1.set_title('Grayscale')
ax2.imshow(cv2.cvtColor(thresh1, cv2.COLOR_BGR2RGB))
ax2.set_title('Binary Threshold')
ax3.imshow(cv2.cvtColor(thresh2, cv2.COLOR_BGR2RGB))
ax3.set_title('Binary Threshold Inverted')
ax4.imshow(cv2.cvtColor(thresh3, cv2.COLOR_BGR2RGB))
ax4.set_title('Truncated Threshold')
ax5.imshow(cv2.cvtColor(thresh4, cv2.COLOR_BGR2RGB))
ax5.set_title('Set to 0')
ax6.imshow(cv2.cvtColor(thresh5, cv2.COLOR_BGR2RGB))
ax6.set_title('Set to 0 Inverted')

plt.show()
```

Original:



After:



▼ What is adaptive thresholding?

- In order to make segmentation stronger to variations in an image, an algorithm should be able to **select an appropriate threshold automatically using the amount of intensity present in the image.**
- This type of thresholding algorithm is known as adaptive thresholding.
- OpenCV has a function **cv2.adaptiveThreshold()** to perform adaptive thresholding.