



**CNG 466 – FUNDAMENTALS OF IMAGE PROCESSING**  
**Fall 2019-2020**

**Assignment 1 - Image representation and interpolation**  
**(Includes Chapter 1 and 2)**

**Deadline:** Due on 10 October 2019, Thursday, beginning of class.

**Policy:** Bring a report (including your code) to the beginning of class. Upload your report and code (.m file) to ODTUClass. Submitted file should be archive file (.7z, .rar or .zip). Late submissions will not be accepted and graded as zero.

**Recommendation:** Initially, read and practise “CNG466 Additional Notes (MATLAB)” to get familiar to MATLAB and then use the book “Digital Image Processing Using MATLAB” to help you to complete this assignment.

**Tasks:**

**1.** Creating MATLAB files and folders.

- (a) Choose a photograph of you that clearly shows your face. Save it to an appropriate directory.
- (b) Open Matlab and change the directory to where you store the image.
- (c) Create a **M file** and write a code to do following steps (sections 3, 4 and 5).

**2. [10 points]** Reading and displaying an image.

- (a) Load the original image using the function **imread**.  
(To learn more about this function, type **help imread** to the command prompt)
- (b) Use function **figure** to create a figure.
- (c) Use function **imshow** to show the image with the created figure.
- (d) Give your own name as a title to the figure using function **title**.
- (e) Give the size of the image (using **size** function) as a x-y labels using **ylabel** and **xlabel** functions.

Example output:



3. [10 points] Image representation and manipulation as a matrix.

- (a) Crop the original image (loaded in section 2 a) around your eyes. Don't use the **imcrop** function, write your own code. You can use for loops or a code as simple as  $B = A(x1: x2, y1: y2)$  where A is the original image and B is the cropped image.
- (b) Use function **figure** to create a figure.
- (c) Show the image and add labels and title (as in step 2 c/d/e).

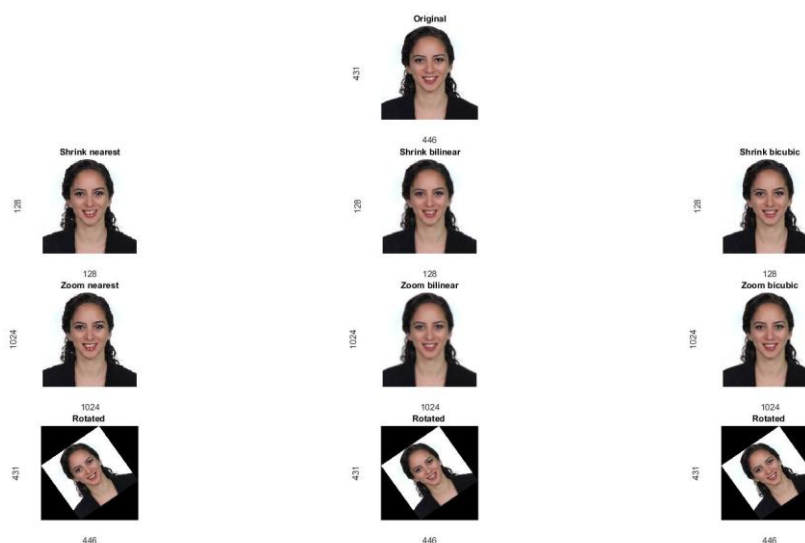
Example output:



4. [40 points] Image interpolation – shrinking, zooming and rotating

- (a) Shrink the original image (loaded in section 2 a) using function **imresize** to size 128\*128 with using nearest, bilinear and bicubic interpolation techniques.
- (b) Then zoom these shrunk images using function **imresize** to size 1024\*1024 with using nearest, bilinear and bicubic interpolation techniques.
- (c) Rotate the original image (loaded in section 2 a) by 35 degrees counterclockwise using function **imrotate** with using nearest, bilinear and bicubic interpolation techniques. Then resize the rotated image to its original size using function **imresize** and nearest interpolation technique.
- (d) Use function **figure** to create a figure and function **subplot** to **imshow** all these steps (section 4, a/b/c) on a same figure (with labels and title).
- (e) What do you see in this figure? Report if you recognize any problems with the resulting images?

Example output:



5. **[40 points]** Image interpolation –zooming

- (a) Zoom the shrunk image (loaded in section 4 a) using your own code (not using function **imresize**) to size 1024\*1024 with using nearest interpolation technique.
- (b) Show the image and add labels and title (as in step 2 c/d/e).