(1) Average the images together

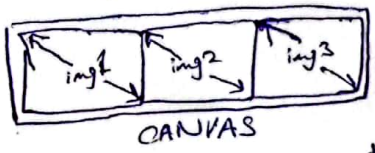
avg_img = (img1 + img2 + img3 + img4 + img5) / 5;

* uint8 can only have integers 0-255 saving memory but has limitations.
e.g. ~~uint8~~ uint8(200) + uint8(200) = 255 & not 400.

* Use im2double(img1) on images before applying any algorithm one them as this rescales the image to range(0,1) as its standard for many img processing algorithms.
So, avg_img = (im2double(img1) + im2double(img2) + im2double(img3))/3;

(2) Compare individual images:-

montage({list of imges}) e.g. montage({img1, img2, img3})



* Datatype double requires a lot more memory than im2 uint8 image

(3) img to Grayscale:-

img = im2gray(img1);

* Grayscale images require less memory, Faster to operate on and is used in more functions

(4) Reduce the img dimensions:-

reduce = imresize(img1, 0.75) # img is reduce to 75% of its original resolution

reduced = imresize(img1, [2000, 2000]); # specify the no. of rows & cols in op img - use this if app requires image to be of specific type. Resized image will be distorted
(rows, cols)

(5) Rotate image:-

r_img = imrotate(img1, -30) #Specify angle. -ve value specifies clockwise rotation

r_img1 = imrotate(img1, -30, "crop") # crop will make the o/p img same size as i/p img.