Mujtaba Shahid Faizi

BSCS-5A

#131818

Lab 2 of Computer Vision

Task 1:

**CODE:**

**import** cv2  
**from** skimage.util **import** random\_noise  
**from** matplotlib **import** pyplot  
  
image = cv2.imread(**'lena.jpg'**,0)  
  
img = random\_noise(image, mode=**'gaussian'**, seed=**None**, clip=**True**)  
pyplot.imshow(img, cmap=**'gray'**)  
pyplot.imsave(**"Task1\_Gaussian.jpg"**, img, cmap=**'gray'**)  
  
img = random\_noise(image, mode=**'s&p'**, seed=**None**, amount=0.15,salt\_vs\_pepper=0.25)  
pyplot.imshow(img)  
pyplot.imsave(**"Task1\_Salt&pepper.jpg"**, img, cmap=**'gray'**)  
  
img = random\_noise(image, mode=**'speckle'**, seed=**None**, mean=0.1,var=0.02)  
pyplot.imshow(img, cmap=**'gray'**)  
pyplot.imsave(**"Task1\_Speckle.jpg"**, img, cmap=**'gray'**)  
  
 **SCREENSHOT:**

*(Images are already saved in the lab 2 folder)*

Task 2:

**CODE:**

**import** cv2  
**import** numpy **as** np  
  
**def** median(img, ksize=3, title=**'Median Filter Result'**, show=1):  
 *# Median filter function provided by OpenCV. ksize is the kernel size.* img = cv2.medianBlur(img, ksize)  
 **return** img  
  
**def** mean(image):  
 *# apply the 3x3 mean filter on the image* kernel = np.ones((3,3),np.float32)/9  
 processed\_image = cv2.filter2D(image,-1,kernel)  
 **return** processed\_image  
  
img1 = cv2.imread(**'Task1\_Gaussian.jpg'**,0) *# Only for grayscale image*img2 = cv2.imread(**'Task1\_Salt&pepper.jpg'**,0) *# Only for grayscale image*img3 = cv2.imread(**'Task1\_Speckle.jpg'**,0) *# Only for grayscale image*img=median(img1)  
cv2.imshow(**'Median filter'**, img)  
cv2.waitKey(0)  
cv2.imwrite(**'Task2\_Gaussian\_Median-Filter.png'**,img)  
  
img=mean(img1)  
cv2.imshow(**'Mean filter'**, img)  
cv2.waitKey(0)  
cv2.imwrite(**'Task2\_Gaussian\_Mean-Filter.png'**,img)  
  
img=median(img2)  
cv2.imshow(**'Median filter'**, img)  
cv2.waitKey(0)  
cv2.imwrite(**'Task2\_Salt&pepper\_Median-Filter.png'**,img)  
  
img=mean(img2)  
cv2.imshow(**'Mean filter'**, img)  
cv2.waitKey(0)  
cv2.imwrite(**'Task2\_Salt&pepper\_Mean-Filter.png'**,img)  
  
img=median(img3)  
cv2.imshow(**'Median filter'**, img)  
cv2.waitKey(0)  
cv2.imwrite(**'Task2\_Speckle\_Median-Filter.png'**,img)  
  
img=mean(img3)  
cv2.imshow(**'Mean filter'**, img)  
cv2.waitKey(0)  
cv2.imwrite(**'Task2\_Speckle\_Mean-Filter.png'**,img)

**SCREENSHOT:**

*(Images are already saved in the lab 2 folder)*

Task 3:

**CODE:**

**import** numpy **as** np  
**import** cv2  
**import** imutils  
  
*# load the puzzle and waldo images*puzzle = cv2.imread(**"waldo1.jpg"**)  
waldo = cv2.imread(**"twaldo.png"**)  
(waldoHeight, waldoWidth) = waldo.shape[:2]  
  
result = cv2.matchTemplate(puzzle, waldo, cv2.TM\_CCOEFF)  
(\_, \_, minLoc, maxLoc) = cv2.minMaxLoc(result)  
*# the puzzle image*topLeft = maxLoc  
botRight = (topLeft[0] + waldoWidth, topLeft[1] + waldoHeight)  
roi = puzzle[topLeft[1]:botRight[1], topLeft[0]:botRight[0]]  
  
*# construct a darkened transparent 'layer' to darken everything  
# in the puzzle except for waldo*mask = np.zeros(puzzle.shape, dtype=**"uint8"**)  
puzzle = cv2.addWeighted(puzzle, 0.25, mask, 0.75, 0)  
  
*# put the original waldo back in the image so that he is  
# 'brighter' than the rest of the image*puzzle[topLeft[1]:botRight[1], topLeft[0]:botRight[0]] = roi  
*#draw a blue rectangle for more highlight*cv2.rectangle(puzzle, topLeft, botRight, 255, 5)  
*# display the images*cv2.imwrite(**'Task3.png'**,puzzle)  
cv2.imshow(**"Puzzle"**, imutils.resize(puzzle, height=650))  
cv2.imshow(**"Waldo"**, waldo)  
cv2.waitKey(0)

**SCREENSHOT:**

