CHEAT SHEET FOR IMAGE PROCESSING

By CodeMaster.ninja

cv2.imread()	loads an image from the specified file
cv2.namedWindow()	used to create an empty window with a suitable name and size to display images
cv2.imShow()	used to create a window and set the image with a given window name
cv2.waitKey()	for blocking the script (wait for a key press)
cv2.destroyAllWindow()	for destroying the displayed image
IMAGE_WRITE_JPEG_QUALITY	used to set the quality of the image (compression size)
cv2.cvtColor()	To convert the image to a different color space
cv2.calcHist()	calculate the histogram of the image as a 1D array
cv2.equalizeHist()	equalize the histogram of the image to increase the contrast

costume func : MSE(image1, image2)	# Calculate the squared difference between the two images
	squared_diff = (image1 - image2) ** 2
	# Calculate the mean of the squared differences
	mse = np.mean(squared_diff)

gaussianNoise	np.random.randn(image.shape[0], image.shape[1]) * 2
saltPepperNoise	np.random.choice([0, 255], size=image.shape, p=[0.95, 0.05])
cv2.add(img1,img2)	this function is used to combine to images (ex : noise + original image)

meanFilter(image,kernel_size)	# Create a kernel with ones and divide each element by the kernelsize
	kernel = np.ones((kernel_size, kernel_size), np.float32) /
	(kernel_size ** 2)
	# Apply the kernel to the image using convolution
	filtered_image = cv2.filter2D(image, -1, kernel)

medianFilter(image, kernel_size)	# Apply the median filter to the image
	filtered_image = cv2.medianBlur(image, kernel_size)

gaussianFilter(image, kernel_size, sigma)	# Apply the Gaussian filter to the image
	filtered_image = cv2.GaussianBlur(image, (kernel_size, kernel_size), sigma)