Read in any image using:

img = imageio.imread('x.png',as\_gray=True)

plt.figure()

plt.imshow(img, cmap='gray')

1. Convolve the image using:

x = ndimage.convolve1d(img,dx,axis= 0)

gx\_I = ndimage.convolve(x,s)

y = ndimage.convolve1d(img,dx,axis= 1)

gy\_I = ndimage.convolve(y,s)

# alternative, you can use np.gradient()

gx\_I,gy\_I = np.gradient(img)[:2]

Then design a smooth gradient filter that just performs the convolution operation once and extracts the edge of the images? Plot the gradients images and the magnitudes, use subplot.

1. Given:

noisy = img + 0.4 \* img.std() \* np.random.random(img.shape)

# # adding salt and peper noise to the image

# # adding salt

num\_salt = np.ceil(0.05 \* img.size \* 0.5)

coords = [np.random.randint(0, i - 1, int(num\_salt))

                 for i in img.shape]

out=C.copy(l)

out[coords] = 255

# # adding pepper

num\_pepper = np.ceil(0.05\* img.size \* (1. - 0.05))

coords = [np.random.randint(0, i - 1, int(num\_pepper))

                   for i in img.shape]

out[coords] = 0

out=out.reshape(img.shape)

And

plt.figure()

plt.imshow(out, cmap=plt.cm.gray, vmin=40, vmax=220)

plt.axis('off')

plt.title('added gaussian noise', fontsize=20)

1. gaussian denoise? b. can you remove the noise by using Gaussian filter? try also box filter and median filter. nd.gaussian\_filter, etc. c. plot the denoised image. Plot as subplots.

1. a) Generate an image of a rotated rectangle. you can create a rectangle by filling center pixels 1 and the rest zero. ndimage.rotate can be used to rotate the image. b) Blur the image using a Gaussian filter. c) apply sobel filter to both x and y direction d) Display the original image, x-derivatives, y-derivatives. use np.hypot to compute the magnitude.