L-Università ta' Malta

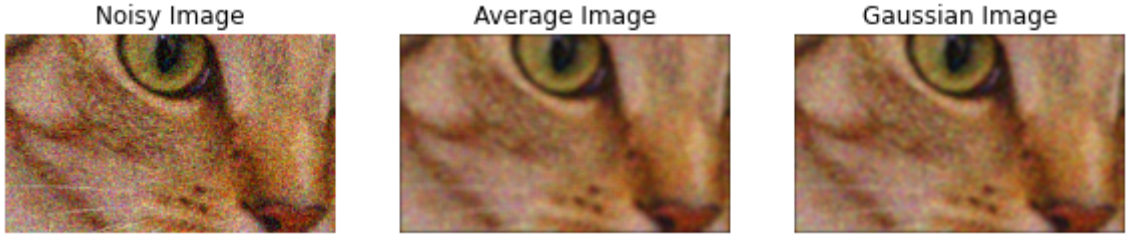
Convolution and Image Gradients

1. Load the `noisy.png` using the `imageio` package and show the image using the `matplotlib` library
2. From `scipy.signal`, import the `convolve2d` function which can be used to convolve kernels with an image.
3. Define a kernel of the average filter using the `numpy.ones` function.
4. Use the `convolve2d` function to convolve the average kernel with each channel of the image independently.
5. Plot the noisy and the image restored using the average filter using subplots as shown below.

6. Use the function below to generate a 5x5 Gaussian kernel.

```
def gauss_kernel(l=5, sig=1.):
    """\
    creates gaussian kernel with side length `l` and a sigma of `sig`
    """
    ax = np.linspace(-(l - 1) / 2., (l - 1) / 2., l)
    gauss = np.exp(-0.5 * np.square(ax) / np.square(sig))
    kernel = np.outer(gauss, gauss)
    return kernel / np.sum(kernel)
```

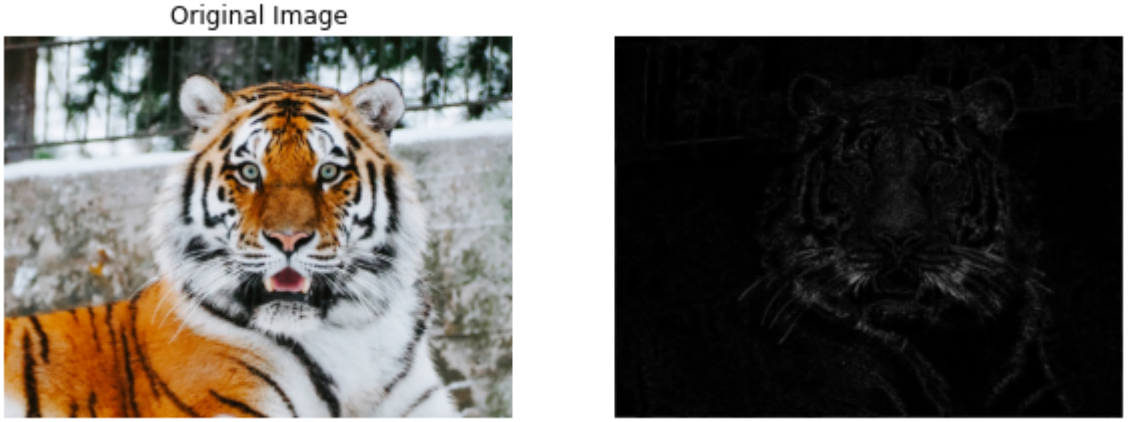
7. Plot the noisy image, the image restored using the average kernel and the image restored using a Gaussian kernel.



8. Use the following function to convert a colour image to grayscale.

```
def rgb2gray(rgb):
    return np.dot(rgb[...,:3], [0.299, 0.587, 0.144])
```

9. Use the `convolve2d` function to convolve the Laplacian kernel with the grayscale image.



10. Use the `convolve2d` function to generate the vertical and horizontal gradients using the Sobel operator.