CS7.404. Digital Image Processing
Monsoon-2022
Assignment-1
Posted on: 08/08/2022
Due on: 01/09/2022, 23:59 hrs IST

- All your code should be in the `src` directory and images in `imgs` directory.
- Do not use any external library other than **`numpy`** for implementing any of the tasks. You can however use external libraries for I/O operations and plotting. If you are not sure if a library is allowed for a particular task, clarify with your TAs.
- You will be evaluated on correctness and how vectorized your code is - with correctness being the priority.
- Write modular code with relevant docstrings and comments for you to be able to use functions you have implemented in future assignments.
- All theory questions and observations must be written in a markdown cell of your jupyter notebook.
- All academic integrity policies apply. Check the course web page for more clarity.
- Start the assignment early, push your code regularly and enjoy learning!

---

1 Michael Scott's Storage Conundrum (15 Points)

In the popular sitcom **"The Office (US)"**, Michael Scott, the world's best boss (a title given to him by himself) likes to capture the little moments that he has with the employees of the Dunder Mifflin Company. He has planned a trip for the employees and would like to take many pictures during the trip. Help the IT guy of the Dunder Mifflin company, answer Michael Scott's queries.

1. **(4 Points)** Before he can trust you, Michael Scott wants to verify if you actually know how images are stored. He points to a coloured image of size 5.76 MB with an aspect ratio of 4:3. He wants you to calculate the height and width of the image in terms of number of pixels. Give him the right answer so that he can build trust in you.
2. **(6 Points)** Now that he has gained trust in you, he tells you that during the trip, he will be capturing images of square shape, with a width of 5200 pixels. He also estimates that he would take a maximum of 2000 images during the trip. He wants to know the *minimum* number of memory cards of size 64 GB he should carry in order to be able to store these images without having to delete any of them if the images were

(a) Black and White images (0 for black and 1 for white)

(b) Gray scale images

(c) Colored images

3. **(5 Points)** Andy and Dwight want to advertise Dunder Mifflin's low prices using a bill board. They made a poster on their desktop. Because they did not know about the concept of resolution - they were satisfied with how it looked on their desktop - to them it was high resolution. However they were able to see individual pixels when they printed the bill board. What might have gone wrong? Michael Scott wants you to explain to them in detail what they might have done wrong with examples.

Note: For the above questions assume that the raw bytes of the images are being stored without any kind of compression. (In reality this is not the case. Image compression techniques will be covered later in the course). Also assume standard intensity range of $0 - 255$ per pixel per channel, unless otherwise mentioned. For the above problems write down the calculations in your jupyter notebook with sufficient intermediary steps.

---

2 Threat Level Midnight (20 Points)

Michael Scott is a well-known name in and around Dunder Mifflin. He's trying to make a promotional poster for his upcoming blockbuster movie, **Threat Level Midnight**. As expected, he has no idea what he's doing. He needs your help to take an image of him on a green screen and graft it onto a fake background image. This task is commonly known as **chroma-keying**. How does the green screen help in this case?

(a) foreground image

(b) background image

(c) result

1. **(5 Points)** Write a function that takes a color image and finds the most frequently occurring color from the image.

2. **(15 Points)** Write a function `mergeImage` which takes two images `fg` and `bg` that extracts the foreground object and places it in the background and returns the resultant image. Test your implementation using the provided foreground (foreground.png) and background (background.png) images.

---

3 Broken Copier and Linear Transforms (30 Points)

Dunder Mifflin recently bought a new fancy and expensive copier for office use, but setting it up was too much trouble. Pam was left in charge of getting it up and running. After spending over four hours, she successfully got almost everything to work. The copier is expensive because it is supposed to be able to transform greyscale levels of images before it copies them. But no matter how hard Pam tried, she couldn't get it to do that. She's not very good at Digital Image Processing so she needs your help doing this.

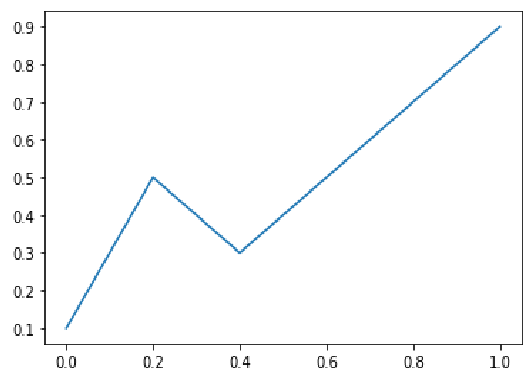
1. **(10 Points)** Write a function `piecewiseLinTransform` that implements a piecewise linear transform defined by

$$g(z) = K_1^i \cdot z + K_2^i \; ; \; a^i \leq z \leq b^i$$

   where, z is the input intensity (generally ranging from $0 - 255$) and i corresponds to the i^{th} piecewise linear transform. The function takes an input greyscale image, coefficients K_1 and K_2 , and intervals $[a, b]$ for each linear segment and should output the transformed image.

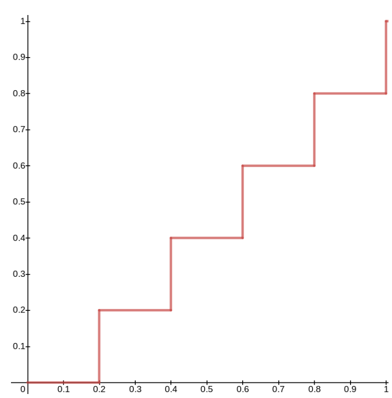
2. **(10 Points)** Run the images through `piecewiseLinTransform` on pam1.png and pam2.png. The piecewise functions to be used are given for each image.

(a) pam1.png



(b) plot1.png

(c) pam2.png



(d) plot2.png

3. **(10 Points)** Choose a photo of your interest, convert it to grayscale, and run `piecewiseLinTransform` using any *custom* piecewise linear function that you like. Plot the function used.

   Make sure to explain all your steps in detail so that Pam can do this on her own next time.

---

4 Slice Off the Unnecessary (50 Points)

It's festival time and you are given the task of slicing breads for the party thrown by Michael Scott. While slicing the breads, you could not help but remember about the concept of *bit slicing*. You start to explain it to the employees of the Dunder Mifflin company. Show them what bit slicing means by answering the following questions:

Note: Convert all color images to gray-scale images before running any of the algorithms.

1. **(10 Points)** Write a function `bitSlice` that takes in `im` - an 8-bit image and `k` - the bit-plane number where $1 \leq k \leq 8$. The function then returns the k^{th} bit-plane of the image.

2. **(10 Points)** Show all the bit planes for `rupee.jpg`, `securityThreat.jpg` and `pepper.jpg`. Label each of the images with their corresponding bit-plane number.

(a) `rupee.jpg`

(b) `securityThreat.jpg`

(c) `pepper.jpg`

3. **(10 Points)** Write a function `imgBitQuantizer` that takes in `im` - an 8-bit image and `k` the number of bits to quantize the image to and returns the quantized image. You may as well use the `bitSlice` that you coded up for 4.1.

4. **(15 Points)** Use `imgBitQuantizer` to show the quantized images when you take first k bits - starting from most significant bit on `rupee.jpg` , `securityThreat.jpg` and `pepper.jpg`. Label the plots and sub-plots (if any) appropriately. Write your observations explaining the plots from 4.2 and 4.4.

5. **(Bonus)(5 Points)** Can you go beyond the call of your duty and give an application of bit-plane slicing that is evident from 4.2 and 4.4? Explain in detail.(Hint: Read the title of the question.)

5 Demystifying Magical Transformations (65 Points)

After a successful trip, Michael Scott was visibly unhappy. He had just reviewed the photos that were taken and observed that many of the photos were not taken with proper lighting. Being an image processing expert, you know that the images can be made better. Make the images better and bring back the smile on Michael Scott's face.

1. **(5 Points)** The Gamma Transform (sometimes also called the Power-Law Transform) is given by the equation $s = c \times r^{\gamma}$, where c is a constant, r is the original pixel intensity, γ is the variable to be set based on requirement and s is the transformed pixel intensity. Answer the following questions:

   (a) **(1 Point)**What is the range of acceptable γ values?

   (b) **(2 Points)**Assuming $c = 1$, will the equation always give a transformed pixel intensity within the 8-bit range? If not, how would you modify c and/or the above equation to ensure it?

   (c) **(2 Points)** Explain the effect of $\gamma < 1$, $\gamma = 1$ and $\gamma > 1$ on various pixel intensities. Provide evidence of the same by plotting the curves for the gamma function.

2. **(10 Points)** Write a function `powerLawTransform` that takes in an image `im` and the value of `gamma` and returns the transformed image.

3. **(20 Points)** Among the many poor images, `scrantonAreal.png` and `tripGrpPhoto.png` are couple of them. Run `powerLawTransform` on both of them with appropriate γ values and enhance the images. Show results with multiple γs and empirically prove your answer to 5.1.c.

4. **(30 Points)** From the set of transforms taught in class, can you find any other magical transform that can enhance `scrantonAreal.png` very similar to the gamma transform? If you can, demystify it by

(a) **(5 Points)** Expressing the transform in mathematical form with all terms defined. Also mention the values of any constants present.

(b) **(5 Points)** Plot the transform with x-axis representing the input pixel intensity and y-axis representing the output pixel intensity.

(c) **(10 Points)** Implement the transform.

(d) **(10 Points)** Now run the magical transform that you have just implemented on `scrantonAreal.png` and compare the results with the results of the gamma transform. Write your observations clearly.

(a) `scrantonAreal.png`

(b) Original Areal Shot of the Scranton Area

(c) `tripGrpPhoto.png`

(d) Original Group Photo

6 The Man With the Most Experience (30 Points)

Ever since Creed was promoted from Head of Quality Assurance to the Interim Manager for Dunder Mifflin, his extraordinary brain power has been threatening to drive the company into the ground. Since no one else seems to care, it has fallen on Pam to make sure that the company doesn't lose any major clients. She comes up with the idea to try and distract Creed with something since he has the attention span of a chihuahua. She tells him that corporate has asked him to perform some tasks with a couple of pictures. Can you help Creed complete them? Explain all your steps in detail.

1. **(15 Points)** First, write a function to perform histogram equalization on a given input greyscale image. Apply the function on picture.jpg and display the input image and the resultant image. Plot the histograms for input and resultant image and explain the changes you observe for each image.

2. **(15 Points)**

   (a) Write a function that takes an input image and a reference image and applies histogram matching on the input image by matching the histogram with that of the reference image. Apply this function on Creed.png and Texture.jpeg with Creed.png as the input image and Texture.jpeg as the reference image.

   (b) Play around with 2 different reference images of your own and show the results.

(a) `Same_picture.png`

(b) `Creed.png`

(c) `Texture.jpeg`

7 Don't Neglect Your Neighbours (240 Points)

For the employees of Dunder Mifflin, your skills in Digital Image Processing is just mind blowing! They never thought one could manipulate the image in the direction one wanted using post taking the image with just a few lines of code. You are excited to showcase your knowledge because what you have just shown them till now is just the tip of the ice-berg. You are yet to show them miracles on color images. But before that, you want to introduce them to the concept of **Spatial Filters**. Answer the following questions in order to explain them the concepts.
Note: Assume kernels are square in shape, unless otherwise mentioned.
Tip: To avoid repetitive code of similar tasks, read through the entire question and plan your code.

7.1 Linear Spatial Filters (150 Points)

The implementation of *linear spatial filters*, requires moving a kernel centered at each pixel of the image and computing the sum of the products of the co-efficients of the kernel and the corresponding image pixels. **Note:** For all practical purposes, assume that the kernel size k is odd, unless otherwise mentioned.

7.1.1 Mean Filter (50 Points)

1. **(10 Points)** Implement a function `meanFilter` that takes in as input an image `img` and the kernel size `k`, and implements the *simple average filter* on the image. Ensure that all the elements of your kernel sums upto 1.

2. **(10 Points)** During the process of applying the `meanFilter`, notice that many individual filtering operations have common spatial locations. Write a function `effMeanFilter` that exploits this information and implements an efficient version of the *simple average filter*. Inputs to the `effMeanFilter` function are the same as the inputs to the `meanFilter` function.

3. **(10 Points)** The employees of the Dunder Mifflin Company do not know programming and hence are not really convinced that your solution to 7.1.b is efficient. But they are good at understanding plots. Prove to them that your function `effMeanFilter` is indeed more efficient than your original function `meanFilter` through a plot. Vary both the image size and kernel size. For a rough idea of what is expected , check out the plot below.

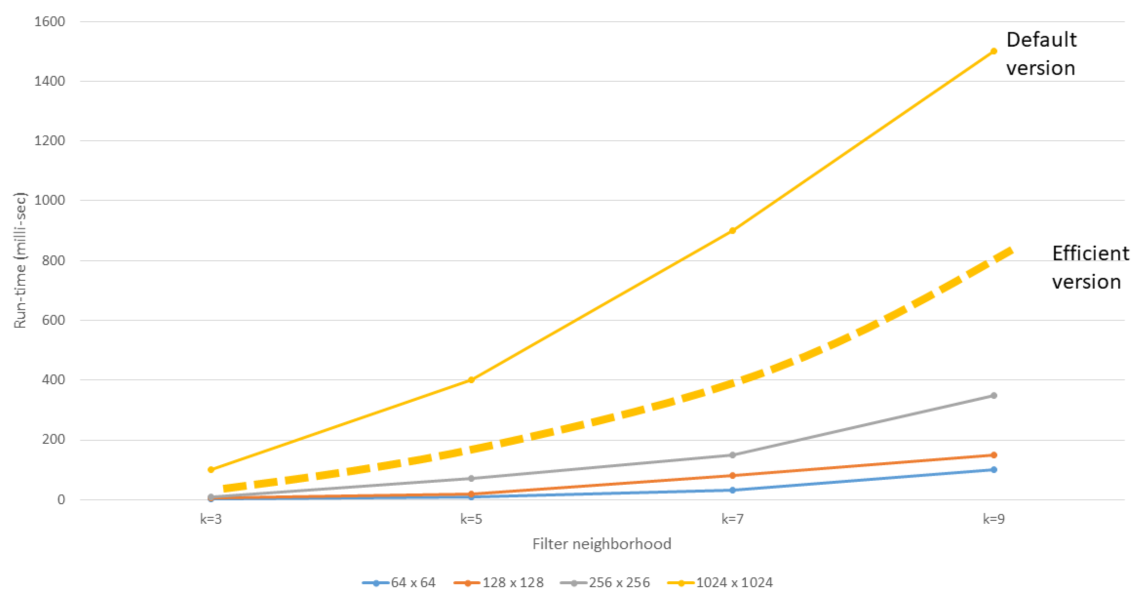


Figure 6: Comparison of Default vs Efficient Mean Filter

4. **(20 Points)** Perform *simple average filter* on `testPattern.jpg` for kernel sizes of $k = 5, k = 9, k = 15$ and $k = 35$. Also plot their corresponding histograms. Briefly explain what you observe?

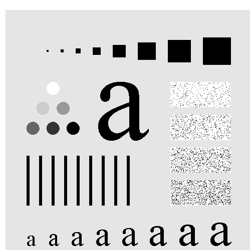


Figure 7: `testPattern.jpg`

7.1.2 Gaussian Filter (100 Points)

Use the following table if necessary.

Range of x	Area of Gaussian Covered
$[-1.5\sigma, 1.5\sigma]$	86.64%
$[-2.5\sigma, 2.5\sigma]$	98.76%
$[-3.5\sigma, 3.5\sigma]$	99.95%

1. **1D Gaussian(30 Points)**

   (a) **(20 Points)**Write a function `approx1DGaussFilter1` that takes in the kernel length of a 1-D array k, and returns the 1-D Gaussian Filter. Also return the standard deviation, σ of the Gaussian that is an approximate fit to the filter (compute this from the mathematical expression of a 1D Gaussian). Plot a graph between the 1-D kernel length k and the estimated standard deviation. Use the concept of Pingala's (most frequently called Pascal's) triangle - which is basically a triangular array of binomial co-efficients.

   (b) **(10 Points)** Write a function `approx1DGaussFilter2` that takes in the standard deviation, σ of the Gaussian kernel. The function must return the kernel with size k. Choose k such that the values of kernel cover 86.64%, 98.76% and 99.95% of the Gaussian being modeled. Use the concept of Pingala's (most frequently called Pascal's) triangle - which is basically a triangular array of binomial co-efficients.

2. **2D Gaussian (70 Points)**

   (a) **(10 Points)** Implement a function `gaussKernelGenerator` that takes in the size of the kernel `k`, and the variance `var` and returns a $k \times k$ Gaussian kernel using the 2D mathematical representation of the Gaussian function.

   (b) **(10 Points)** Write a function `approx2DGaussFilter` that takes in the standard deviation σ and returns a 2D Gaussian - which is generated from a 1D Gaussian. Ensure that the values of the kernel produced captures 98.76% of the Gaussian. Plot the percentage of the kernel co-efficients that are effectively equivalent to being zero for the purpose of filtering of images as you vary σ. (Hint: $\texttt{int}\left(\frac{x}{y}\right) = 0$ where $x < y$).

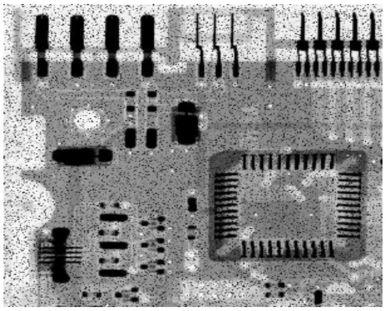
   (c) **(20 Points)** Write a function `gaussianFilter` that takes in an image `im` , the variance of the Gaussian kernel `var` and the kernel size `k` and returns Gaussian filtered image. Use `gaussKernelGenerator` to generate the Gaussian Kernel required. Run `gaussianFilter` on `testPattern.jpg` for kernel sizes of $k = 5, k = 9, k = 15$ and $k = 35$. Also, compare the outputs with the outputs of 7.1.1.4. Also comment on the effect of variance.

   (d) **(Bonus) (10 Points)** Analyze the error between the Gaussian generated by the Pingala's triangle vs the Gaussian generated without approximation. Feel free to explore how you want to display the error. One way (you can choose other ways too) to do so is to show a heat map (checkout `cmap="hot"` parameter of `matplotlib.pyplot.plot` function) of the absolute error between the two kernels. Make appropriate assumptions and state them clearly.

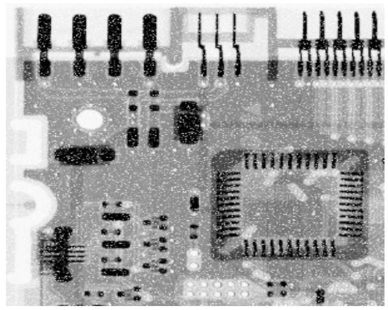
   (e) **(Bonus)(20 Points)** Empirically prove that applying a 2D Gaussian is same as applying the horizontal and vertical 1D Gaussians individually in a sequential manner in any order (i.e. horizontal filter followed by vertical or vice versa).

7.2 Non Linear Filters (40 Points)

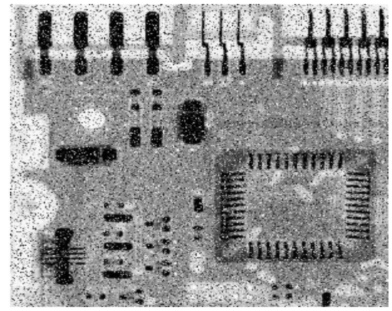
1. **(30 Points)** Use appropriate filters to **de-noise** each of the following images. Choose appropriate kernel size and explain the choice of your filter.



(a) `noisy1.jpeg`



(b) `noisy2.jpeg`



(c) `noisy3.jpeg`

2. **(10 Points)** Write a function `effMedianFilter` that implements an efficient median filter (exploiting the same observation as made in 7.1.1.2).

7.3 Edge Detection (50 Points)

1. **(20 Points)** Consider **Roberts**, **Prewitt**, **Sobel** and **Laplacian** filters that were discussed in the class. Display the magnitude and direction images obtained after applying these filters on `office.jpg`. Make your observations by comparing their outputs.

   (a) **Roberts:**

   $$M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

   (b) **Prewitt:**

   $$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

   (c) **Sobel:**

   $$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

   (d) **Laplacian:**

   $$L_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad L_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2. **(10 Points)** What will the 5×5 variants of Sobel and Prewitt filters look like? Apply these larger filters on `office.jpg` and make observations upon comparing their outputs with the corresponding smaller filters.

3. **(20 Points)** Using Gaussian sampling to generate noise and add to the input image (you could choose any image you want as well). Study the effect of applying the above filters on noise-affected inputs. In particular mention clearly which of the filters are more robust to noise and which are less.

8 Processing the Galaxies (40 Points)

Everyone respects and revers scientists, and there is a sense of even more respect for space scientists of organizations like `ISRO` or `NASA` just because of their sheer audacity to try things we only imagined.

Using the simple tools that you have just explained, show the Dunder Mifflin employees how simple concepts are fundamental to extremely complex tasks, and that they are one step closer in understanding what their heroes do. In particular show them step-by-step how would you go from filtering out the unnecessary details in `galaxies.jpeg` to produce the required galaxies as shown. Explain to them each choice that you make clearly with evidence wherever required.

(a) `galaxies.jpeg`

(b) `Processed Galaxies`