



Assignment

Data Science - Image Processing

Important, please read first!

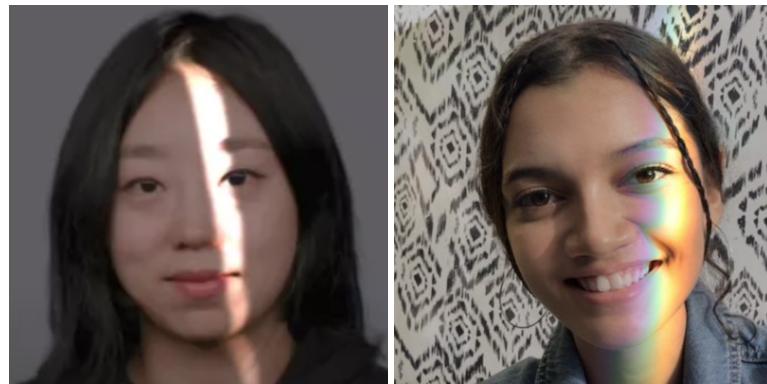
- This assignment should be submitted via DUO. The deadline is **2pm on Thursday, February 4th 2021**.
- **Use only Python 3.7.4 with OpenCV 4.1.1., also available on AppsAnywhere.**
- Submit a compressed archive (.zip) having the mask(s) generated in problem 1 (if you are not generating them in code) in addition to **three** files: a python file with your four functions (named problem1, problem2, ... and taking appropriate parameters), a .txt file with example commands to execute each one of your four functions and a report file in .pdf format no longer than 5 pages including images; at the top of the first page of the document identify yourself only using your CIS username.
- For the results submitted in the report, only use the input images provided in DUO.
- The main purpose of the report is to (i) include evidence of testing, (ii) explain why you have chosen a particular filter or technique, (iii) briefly discuss the computational complexity of each filter, (iv) discuss anything else requested for a particular problem (e.g., problem 4), (v) comment on anything else that you consider important.
- The marks available for correct implementations/answers to each question are indicated. Partial credit will be given for good attempts.
- Your marked work will be returned on the **week beginning March 3rd**.
- The Image Processing sub-module contributes **25% of the total Data Science module mark** and is only assessed by this coursework.
- A FAQ section in DUO will be updated with questions as they arise.

Imagine you are a newly hired engineer at Instagram. As part of your training you were asked to implement a few Instagram filters, to showcase your understanding of basic image processing concepts and image transformation techniques, in addition to your creativity.

PROBLEM 1: LIGHT LEAK AND RAINBOW LIGHT LEAK - 25 MARKS:

Implement a filter that generates the light leak [10 marks] and rainbow light leak ("flower crown") [15 marks] effects shown below. First darken the input image appropriately and then apply a light leak mask (identical size to the input image) that can be either generated by hand in your program of preference (e.g., GIMP), or generated using code. The same or different masks can be used for both effects. Implement appropriate controllable blending between the image and mask. The face will always be at the center of the image. Showcase your results by using the input images provided in DUO. The effect does not need to look exactly like the examples, of course; masks can be arbitrary but should look as if light is leaking off a window.

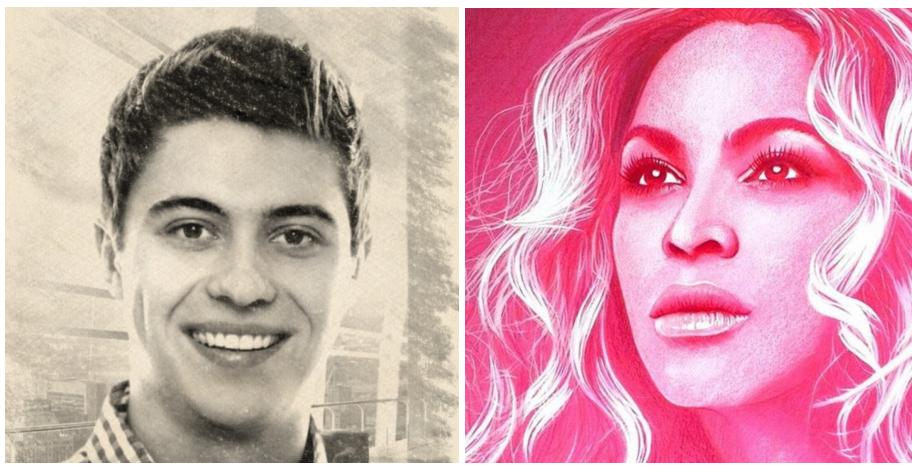
Adjustable parameters: (i) darkening coefficient, (ii) blending coefficient between the input image and the mask, (iii) mode, i.e., simple or rainbow



PROBLEM 2: PENCIL/CHARCOAL EFFECT - 25 MARKS:

Implement a filter that applies a pencil/charcoal effect. The effect should be applied on a greyscale version of the input images provided in DUO, using adjustable blending strength. Generate the effect by blending an input image with an appropriate custom-generated noise texture [15 marks] that has an appropriate motion blur effect applied to it [5 marks] to create pencil-stroke-like effects. Develop a coloured pencil effect by applying the effect on two of the three RGB output channels [5 marks] (input image should remain greyscale, i.e., input RGB channels will hold identical information) and apply a different noise texture to each channel. Showcase your results by using the input images provided in DUO.

Adjustable parameters: (i) blending coefficient between the input image and the noise texture to adjust stroke strength, (ii) mode: monochrome or coloured pencil



PROBLEM 3: SMOOTHING & BEAUTIFYING FILTER - 25 MARKS:

Develop a filter that first smooths out an image [5 marks] and then applies Instagram-like colour grading [20 marks] to beautify it further. Justify in the report your decisions, e.g., why you picked the image smoothing (blurring) filter that you used and why you set its parameters the way you did for the input

images in DUO. For beautification you should use lookup table transformations (colour curves) that map input colours to output colours. For this you can use the univariate spline from scipy that takes a few reference values and finds a method to modify the other values in the requested range with respect to the reference values, or you can hardcode your own lookup table if you feel particularly creative. You can apply the same curve to all channels, a selection of channels, or have a different curve per channel. For example, consider changing the colour temperature of an image: to obtain a "warm" image, you could increase the values of the red channel and decrease the values of the blue channel for all the pixels in the image. Or for a "cold" image, you could do the opposite: increase the values for the blue channel and decrease the values for the red channel. Don't forget to give a creative name to your filter!

Adjustable parameters: (i) blurring amount (variable number of parameters depending on the filter you choose to implement)



PROBLEM 4: FACE SWIRL - 25 MARKS:

Develop a filter that applies a geometric transformation and in particular image warping to create a swirl effect. A geometric transformation in this context is a 2D mapping $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ acting upon pixel coordinates in the source image, to transform them to new coordinates in the target image. Image swirling is a non-linear image deformation that creates a whirlpool effect.

For geometric transformations on images we typically apply reverse mapping: for each pixel in the output, we compute its corresponding position in the input image. Doing it the other way around, may yield -in some cases- empty pixels in the output. With reverse mapping each location on the output image has exactly one corresponding location in (or in some cases, outside) the input image, and even if that position is non-integer, interpolation can be used to compute the corresponding image value.

Tasks:

1. Write a function to perform a swirl geometric transformation on a digital image. Hint: research and use polar coordinates! Use your code to experiment with two different interpolation strategies. Test both nearest neighbor interpolation and bilinear interpolation. There is no need to automatically detect the face, consider the face being at the center of the image. [15 marks]

Parameters: (i) strength of swirl (swirl angle), (ii) the radius of the swirl extent.

2. Implement pre-filtering, i.e., low-pass filtering (blurring) the source image prior to transformation and demonstrate that this process can help suppress aliasing artifacts in the transformed image. [5 marks]

3. Perform an image warp, and then perform the inverse transformation on the result to yield an image resembling the source image. Visualise the difference between the result and the original source image via image subtraction and comment on the error. [5 marks]

