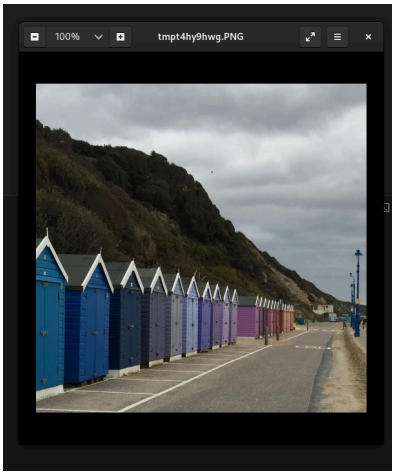
Advanced Mathematics Assignment 2024-2025

Jan 20th - Jan 24th

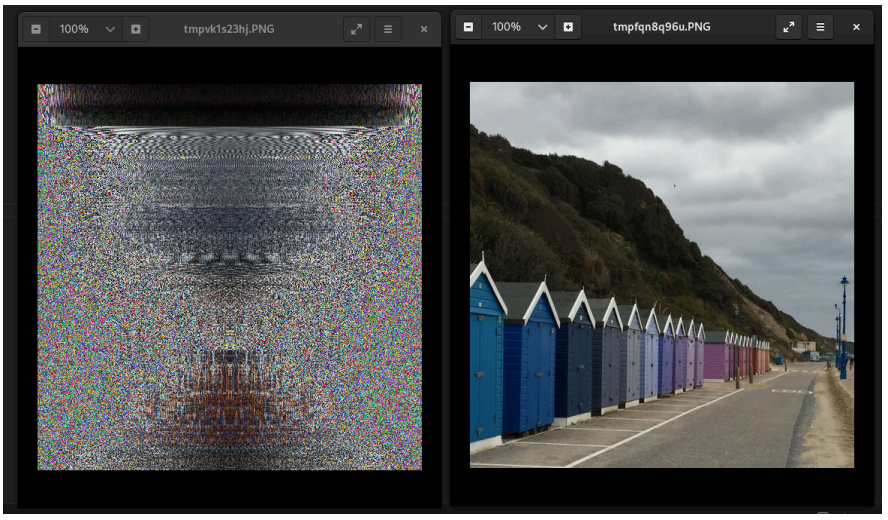
a) Write Code to load in an image of your choice, display it on screen and save it back to disk [5 Marks]



I used Pillow in Python to load the image, display it and export it. In preparation for parts b, c, d, and e I turned the Pillow image data into a numpy array and back again. I chose to use an image of Bournemouth Beach (Spurin 2023) for testing my code.

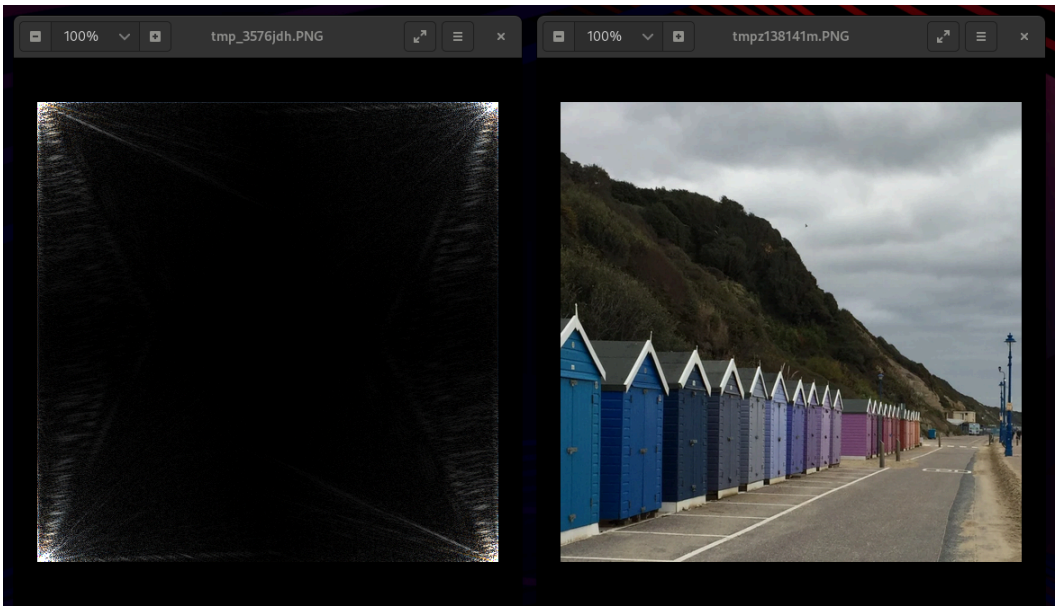
b) Write code to Fourier Transform each row of image. Show the result, and Fourier transform it back to the original image [20 Marks].

I used Fast Fourier Transform (with the ability to do Inverse Fast Fourier Transform by flipping the sign) (Stephenson 2025) (Maklin 2024) on each colour channel for each pixel in each row of the image. This gave me a coloured view of the transformed pixels which I could use to pick up colour trends. I called this function the CFFT (Color Fast Fourier Transform).

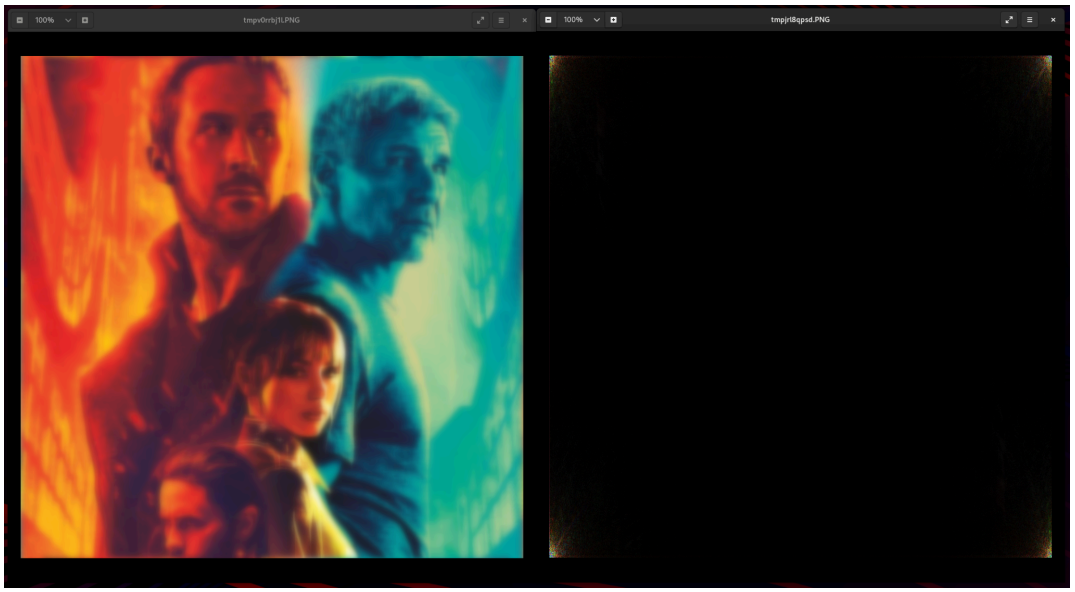


c) Switch the rows and Columns of the Fourier transform results, then Fourier transform the new rows to produce a full 2D Fourier transform. [15 Marks]

I created a function called FF2D_transform that could Fourier Transform 2D images between the different spaces. On the left, you can see the image in transform space, and on the right, you can see the image in normal space.

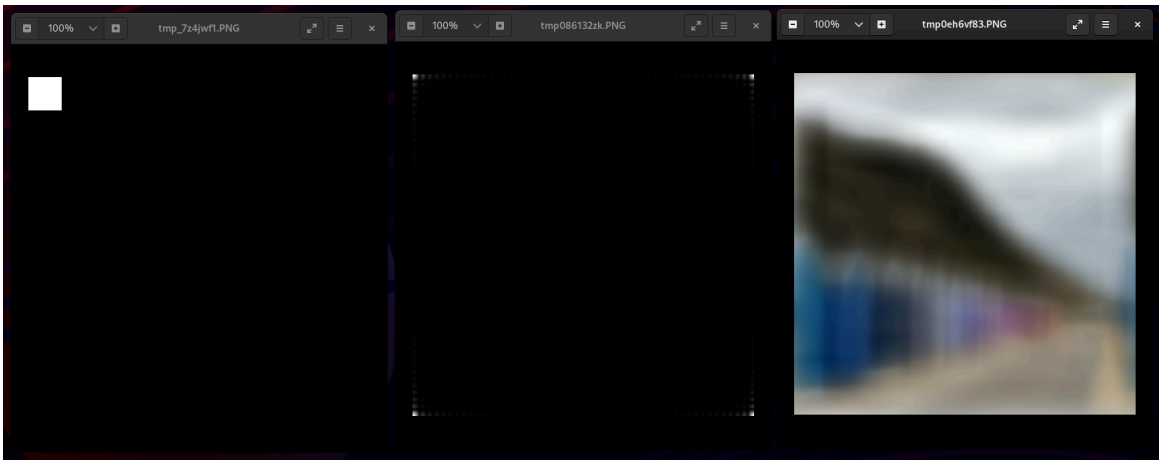


I was a bit disappointed that the CFFT wasn't very effective in this image. So I also tested using a more mono-chromatic picture (Zhang 2024) to show the colors better. You can see in the corners (right) a much more orange hue, and as the data spreads towards the centre it has a more blueish green. Being able to see this makes it possible to refine kernels and target specific details.



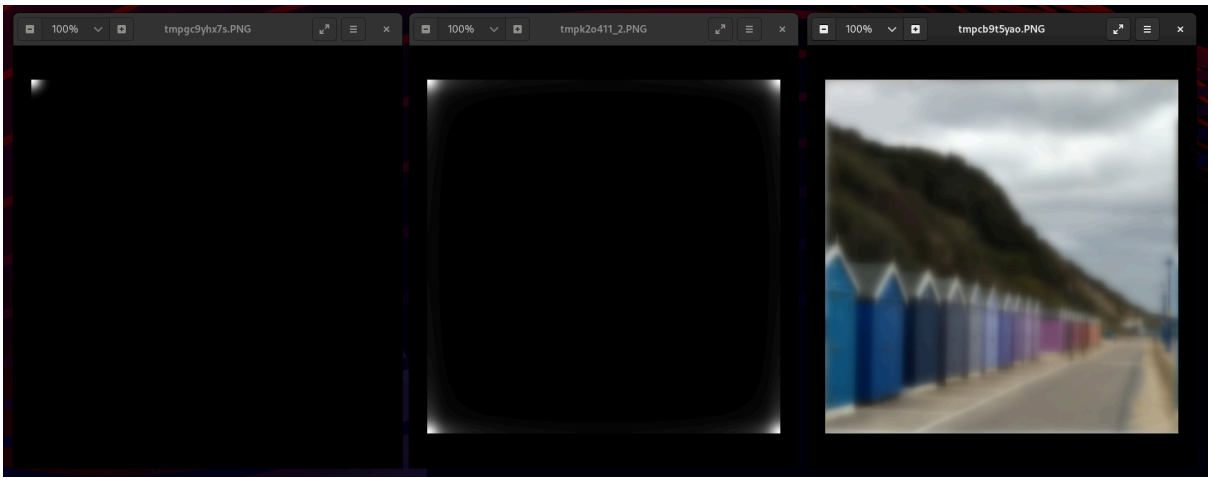
d) Create a second image containing a box convolution kernel. Transform both images, multiply them together and transform back. [20 Marks]

I created a second image with a box inside of it to use as a convolution kernel. Due to how the CFFT behaved, the top left corner is the "centre" of the image, which is why the box is in the corner (left). The middle is the CFFT of the box, which is then multiplied by the CFFT of the image (which is shown in part c). When ICFFT is undergone, the product of the two transforms shows a box-blurred image.



e) Choose a "better" convolution kernel and apply that to your original image to produce an aesthetically pleasing blur [10 Marks]

Following the equation of the 2D Gaussian Distribution (Vignesh 2023). I was able to implement a "better" convolution kernel for blurring. The left-most image is the Gaussian shape, the middle is the Fourier-transformed convolution, and on the right is the blurred image.



Reference list

Maklin, C., 2024. *Fast Fourier Transform Explained | Built In* [online]. builtin.com. Available from: https://builtin.com/articles/fast-fourier-transform [Accessed 20 Jan 2025].

OpenAI, 2022. *ChatGPT* [online]. ChatGPT. Available from: https://chatgpt.com/ [Accessed 20 Jan 2025].

Spurin, J., 2023. *101 Things to do in Bournemouth – Bournemouth Beach* [online]. Jane's Midlife Journey. Available from: https://janesmidlifejourney.uk/2023/04/06/101-things-to-do-in-bournemouth-bournemouth-beach [Accessed 20 Jan 2025].

Stephenson, I., 2025. *10.Fourier* [online]. Keynote. Available from: https://www.icloud.com/keynote/0D8pGkg7AINmA9MkZa7aO2kAw#10.Fourier [Accessed 21 Jan 2025].

Vignesh, V., 2023. *Gaussian Blurring — A Gentle Introduction* [online]. Medium. Available from: https://pub.towardsai.net/gaussian-blurring-a-gentle-introduction-e34aca1d9bbd [Accessed 20 Jan 2025].

Zhang, J., 2024. *Film Review: Blade Runner 2049* [online]. Strange Harbors. Available from: https://www.strangeharbors.com/blog/2017/10/10/film-review-blade-runner-2049 [Accessed 20 Jan 2025].