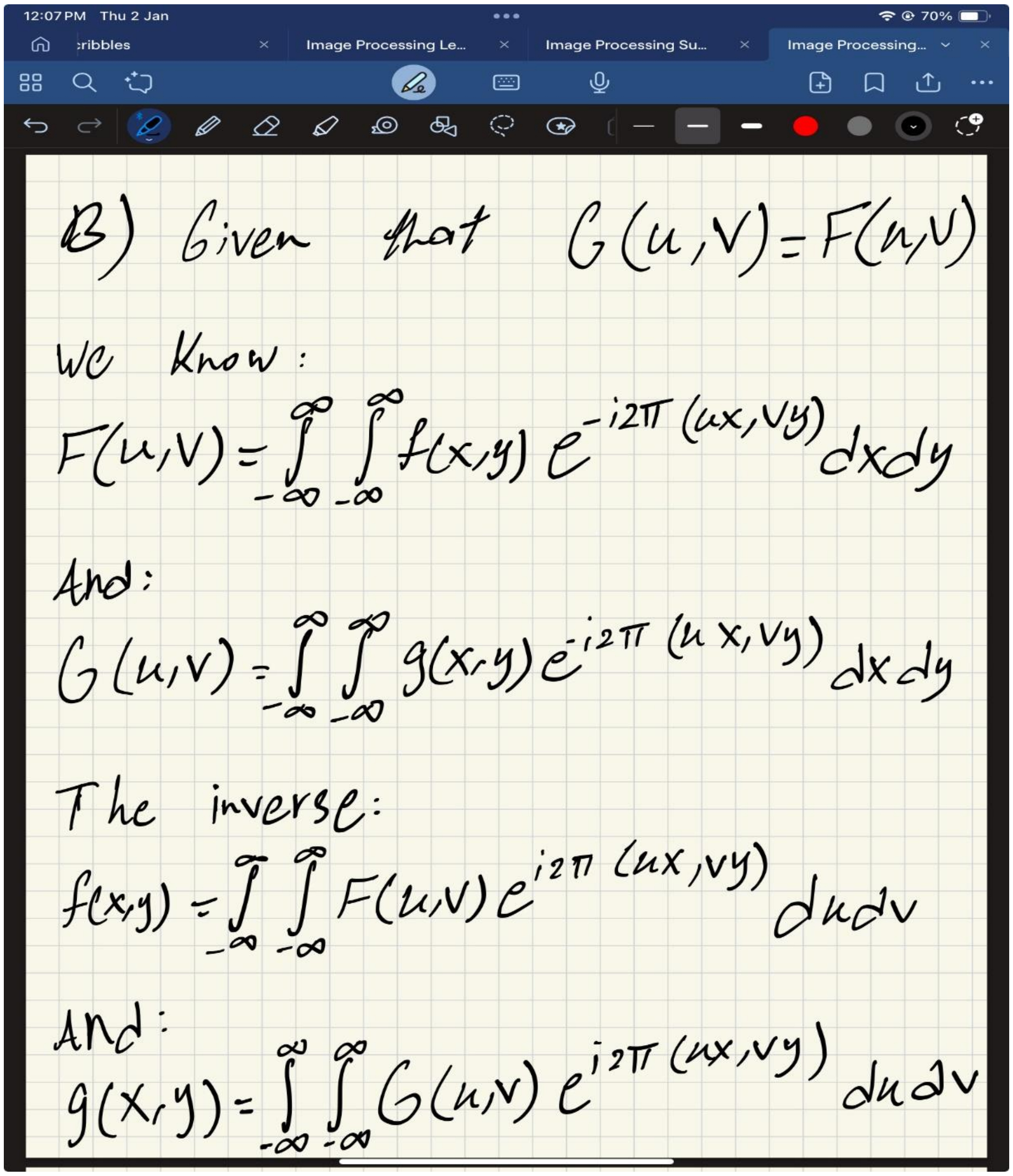
Problem 1 – Understanding Fourier:

A)

Geometric operations with interpolation are generally better for most practical applications. It is faster, easier to use, and sufficient for most tasks.

While the Fourier transform is more suitable for specialized cases for example when the precision in the frequency domain is critical, geometric operations in most cases are preferred due to their efficiency and simplicity and they are for general use.

B)



B) Given that $G(u,v) = F(u,v)$

We Know:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i2\pi(ux,vy)}\, dx\,dy$$

And:

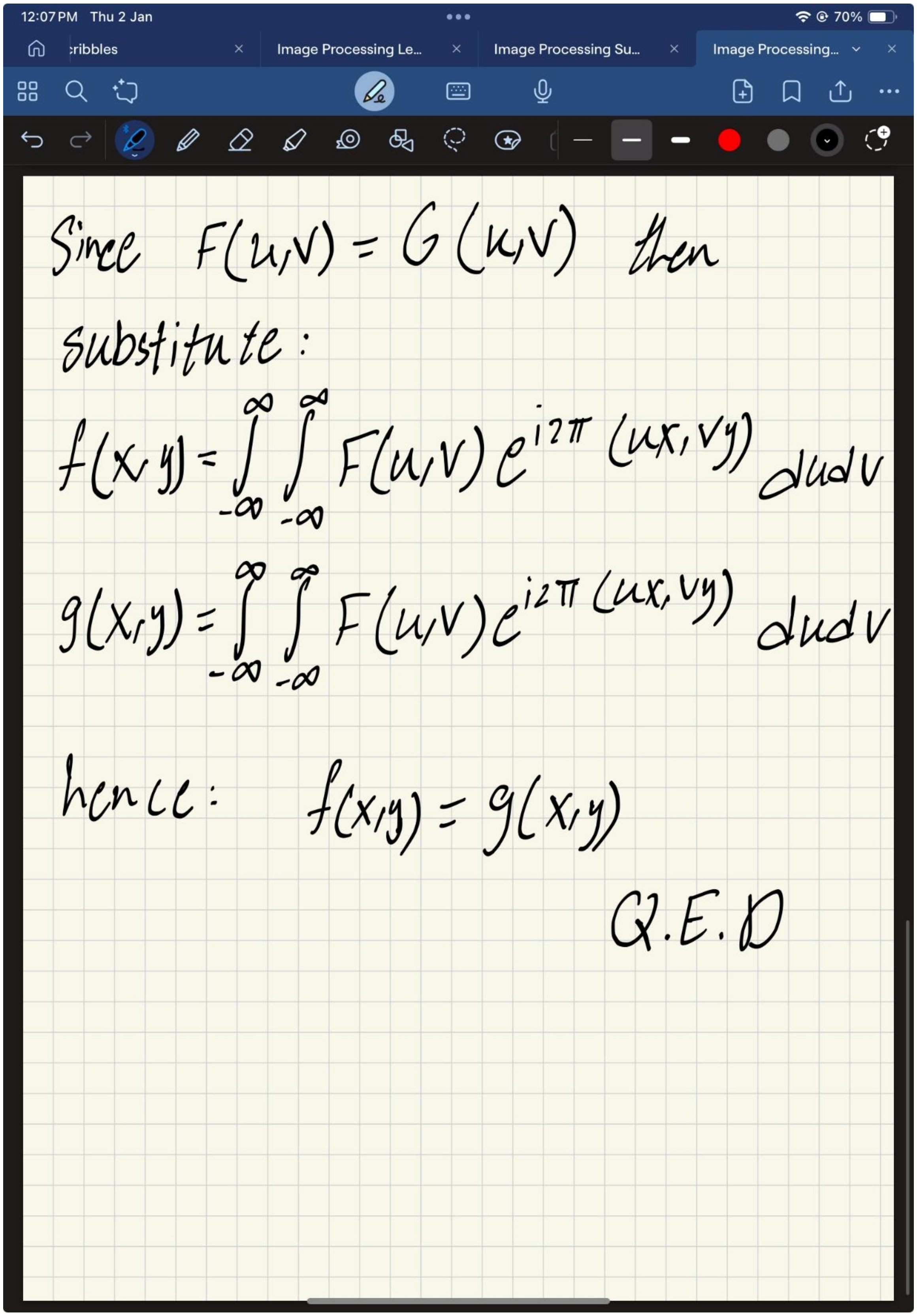
$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\, e^{-i2\pi(ux,vy)}\, dx\,dy$$

The inverse:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{i2\pi(ux,vy)}\, du\,dv$$

And:

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(u,v)\, e^{i2\pi(ux,vy)}\, du\,dv$$



Since $F(u,v) = G(u,v)$ then

substitute:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{i2\pi(ux,vy)}\, du\,dv$$

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{i2\pi(ux,vy)}\, du\,dv$$

hence: $f(x,y) = g(x,y)$

Q.E.D

Problem 2 – Scaling the Zebra:

First method:

1. Apply zero padding to the Fourier-transformed image

2. Expand the image size by increasing spatial dimensions.

3. Adjust pixel brightness by multiplying values by four to maintain intensity.

In the first method, zero padding is added to the Fourier-transformed image. This means extra low-frequency components with a value of zero are introduced, which helps enlarge the image, the image size increases from (H, W) to (2H, 2W) because zero padding spreads out the frequency data. However, this makes the pixels appear dimmer since the overall intensity spreads over a larger area, and to keep the image's brightness consistent with the original, the pixel values are multiplied by four. This ensures the average color intensity remains the same even after scaling.

Second method:

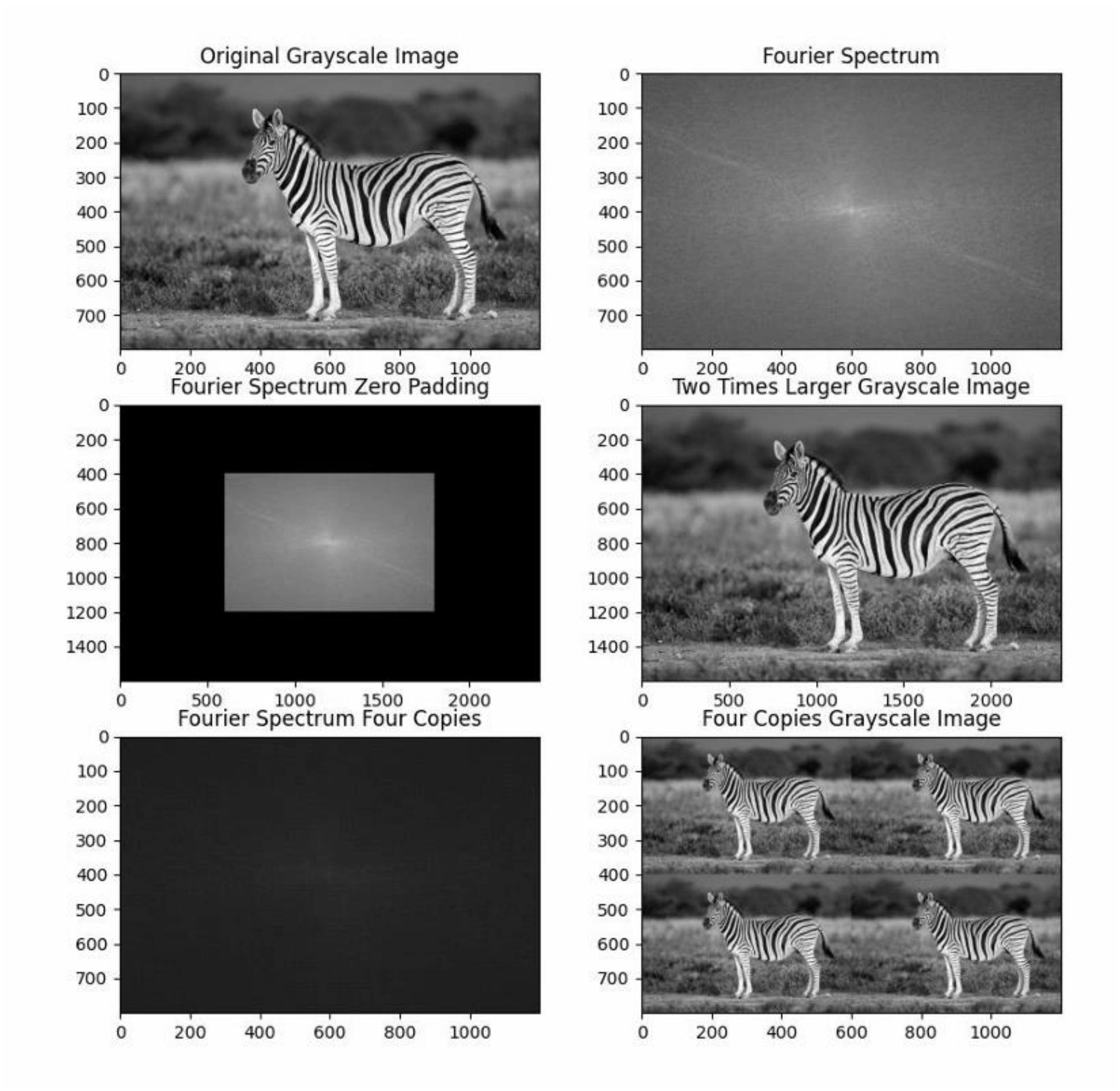
1.Double the frequency in the x and y directions.

2. Expand the image size to create four copies of the original.

3.Adjust pixel brightness by multiplying values by four.

In the second method, we double the frequency in both spatial directions (x, y). This allows us to replicate the original zebra image multiple times, the new image area becomes four times larger—from (H, W) to (4H, 4W). This is achieved by inserting zeros between pixels in the Fourier-transformed image, similar to the first method, the image's brightness reduces due to the increased size. To correct this, we multiply all pixel values by four, ensuring the brightness matches the original image.

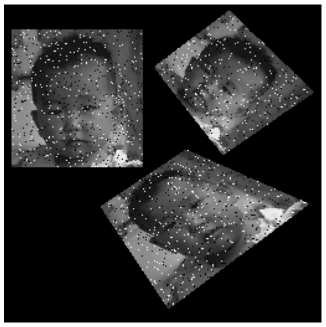
The first method enlarges the original image while maintaining its structure, whereas the second method creates a larger image by replicating the original four times. Both methods require brightness adjustments, but their purposes and scaling approaches are distinct.

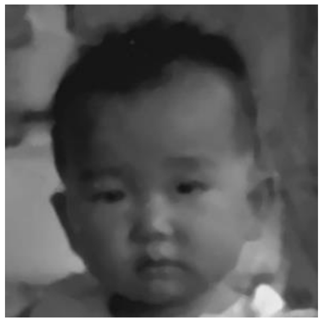


Problem 3 –

1. The first problem – the baby:

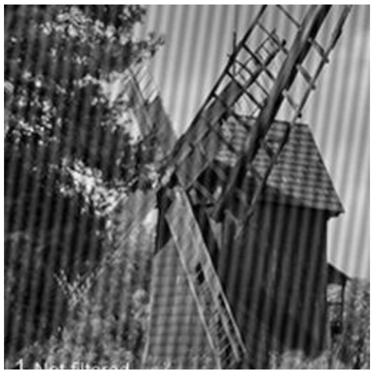
We're using three images, similar to what we did in HW2, and we apply transformations to align each image of the baby into a complete one. By stacking the images, we can reduce the salt-and-pepper noise, though it doesn't fully eliminate it. Finally, we use a median filter to improve the result further.

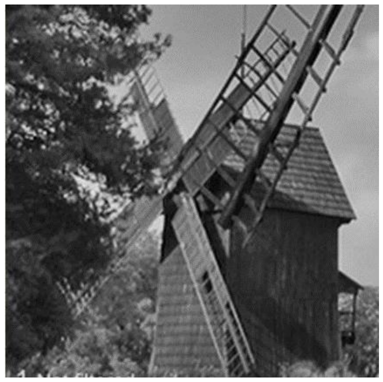




2. The second problem – the windmill:

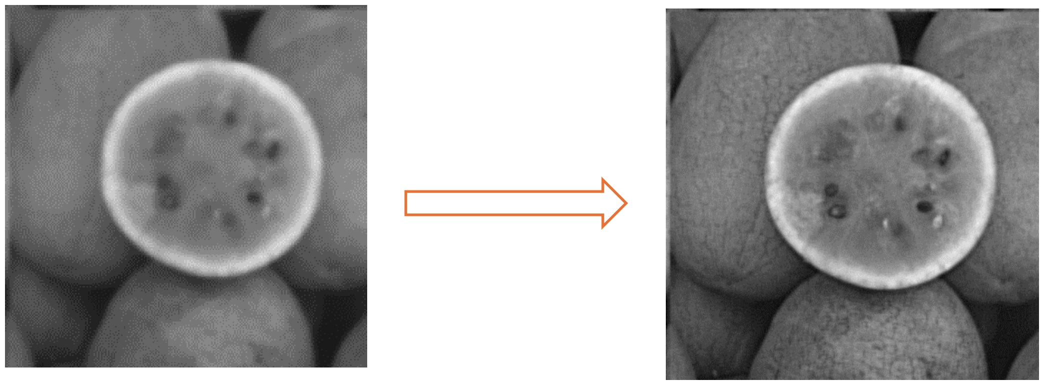
The image contains noise that repeats regularly, so we eliminated the noise by transforming the image into the Fourier domain, canceling out the noise, and then converting it back to its original state.





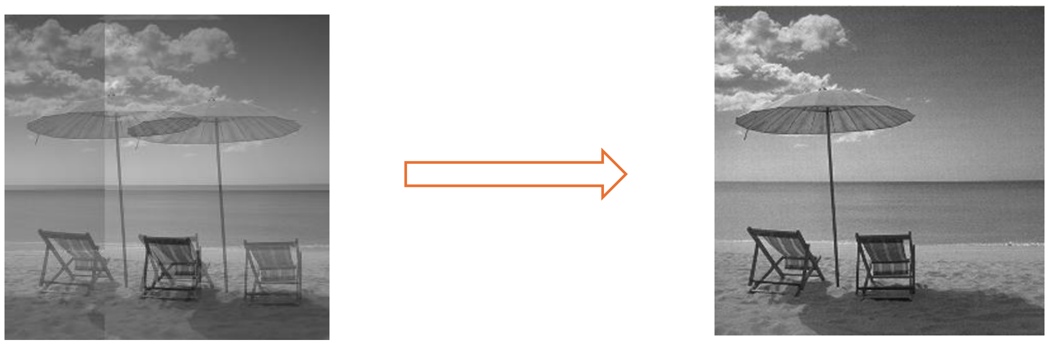
3. **The third problem – the water melon:**

   Just like in the tutorial, we applied a sharpening filter to the original image to achieve the result.



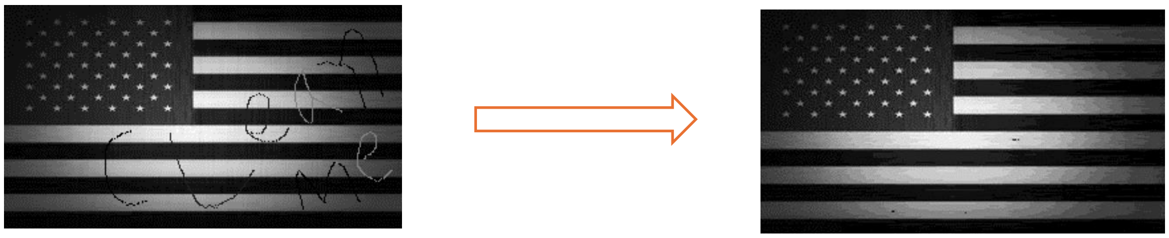
4. **The 4th problem – the umbrella:**

In the tutorial, we learned how to compare a duplicate image to the original one. We calculate the Fourier transform of the original image, making sure to handle division by zero carefully. Then, we divide the Fourier transform of the noisy image by the Fourier transform of the original to help reconstruct the image



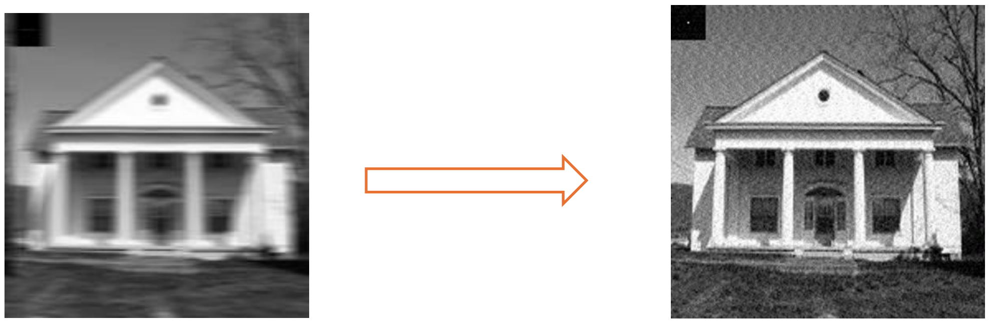
5. The 5th problem – the USA flag:

To remove the salt-and-pepper noise, especially in bigger chunks, we use a median filter. This is applied to each pixel and its neighboring pixels in the same row. We do this because we want to keep the flag's stripes looking clean. The filter is only applied where it's necessary, so the final result isn't affected too much



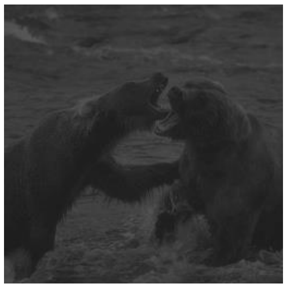
6. The 6th problem – the house:

We did something similar with the umbrella picture, but instead of working with just 2 images, we used 10. The distance between the copies was different for each one. For the umbrella, we followed the same process: we compared a noisy version to the original by calculating the Fourier transform. We made sure to avoid division by zero, and then divided the Fourier transform of the noisy image by the original to help restore the image.



7.The 7th problem – the bears:

To improve the image, we applied histogram equalization to spread out the gray values more evenly. This allowed us to adjust the brightness and contrast to make the image look clearer and closer to the target



$\Rightarrow$