A Review on the Effects of Fourier Transforms and Convolution in Image Filtering

John K. Goebel[1]

[1]Joint Department of Biomedical Engineering, University of North Carolina-Chapel Hill and North Carolina State University, Raleigh, NC

Fourier Transform, Convolution, Motion Blur, Filtering, Frequency Domain, Spatial Domain

ABSTRACT: This review explores the roles of the Fourier Transforms and Convolution in image processing. Despite the wide range of techniques available, these mathematical operational remain fundamental in transitioning from the spatial to frequency domains, significantly enhancing image analysis and filtering capabilities. An examination of the theoretical foundations of Fourier Transform and Convolution is presented with examples to demonstrate their use. Specifical emphasis is placed on their efficiency in deblurring and noise reduction, particularly in removing motion blur from images underneath the presence of noise. Furthermore, the paper also discusses the impact of these techniques in contemporary applications, such as high-resolution medical imaging. By analyzing these fundamental tools, the review aims to provide a deeper understanding of their critical roles and implications in the advancement of image processing technologies.

INTRODUCTION

Image processing has proved to be extremely important for applications ranging from medical diagnostics to digital media. Digital image processing involves methods that manipulate digital images by utilizing computer algorithms[1]. This field spans a range of techniques, from basic pixel processing to linear shift-invariant systems and even nonlinear image filters[2]. The roots of digital manipulation lies in its role as a preprocessing step for various applications, including facial recognition, object detection, and image compression[1]. From these images, image processing aims to dispense of the unimportant information and enhance the important information of the image.

Both the input and output in this process are images, which are defined by the variation in brightness of their pixel values. For filtering and analysis, an image is converted into a numeric matrix of its pixels. This transformation is referred to as digitization[3]. Analyzing and understanding this numerical matrix requires the use of image models and transforms. Image models serve to quantitatively characterize the data within the image, while image transforms are instrumental in examining this data across different domains, including the transform domain[3]. These techniques are essential for various image filtering applications such as compression, enhancement, and noise reduction[3].

In image filtering, many methods have proven to be more straightforward to develop and analyze in the frequency domain. The Fourier Transform serves as an efficient tool for transitioning from the spatial to the frequency domain[2]. Similarly, convolution and its inverse process, deconvolution, are critically important for removing motion blur from images. The Fourier Transform and Convolution are intricately intertwined, offering significant utility when used together. Their efficiency has improved with the advancement of fast algorithms for computing the Fourier Transform and its inverse[2]. When dealing with large filters for image convolution, it is often more efficient to perform this operation in the frequency domain. Additionally, this approach allows for a better understanding of a filter's effects. Any filter in the spatial domain can be analyzed to determine its impact on the frequencies in the frequency domain[2].

The broader objective of this review is to better explore how the Fourier Transform and Convolution mathematical operations play a key role in image processing

THEORETICAL FOUNDATION

The core techniques for image filtering, a subset of image processing, refer to the procedures used to manipulate digital pictures by changing their pixel values. Often, algorithms apply mathematical operations to help with noise reduction, blurring, feature extraction, etc[5].

Fourier Transform

The 2D Fourier Transform is essential in image processing, as it decomposes an image, denoted as $f(x, y)$ where x and y are spatial coordinates, into its constituent frequencies, example shown in Figure 1. This transformation shifts the representation from the spatial domain, characterized by pixel brightness and location, to the frequency domain, represented through the frequency and amplitude of sinusoidal components. Low frequencies represent gradual, smooth transitions in the image, while high frequencies align with rapid changes and fine details[3]. This shift simplifies tasks such as image filtering, where operations like blurring (low-pass filtering) or edge enhancement (high-pass filtering) become more efficient. The Fourier Transform also uncovers hidden periodicities not readily visible in the spatial domain[2].

The Fourier Transform captures both amplitude and phase of sinusoidal components, representing frequencies with both positive and negative numbers[2]. For discrete images, the Discrete Fourier Transform (DFT) is applied:

$$\mathcal{F}[u,v] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n]e^{-i2\pi(\frac{um}{M}+\frac{vn}{N})}$$

where u and v are frequencies along the x and y-directions, respectively, and where M and N are the dimensions of the image[2].

The Inverse Discrete Fourier Transform (IDFT) is given by:

$$f[m,n] = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\mathcal{F}[u,v]e^{i2\pi(\frac{um}{M}+\frac{vn}{N})}$$

The Inverse Discrete Fourier Transform (IDFT) is used to reconstruct the original image from its frequency domain representation. Essentially, it reverses the process of the Discrete DFT. The purpose of the IDFT is to enable the analysis and manipulation of images in the frequency domain (such as filtering or enhancement) and then convert them back into their original in the spatial form.

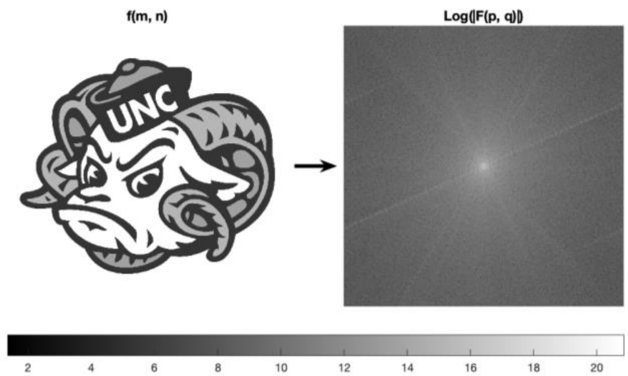

Figure 1. Picture depicting the UNC Tar Heel in the spatial domain and the displayed log of its Fourier Transform in the frequency domain. The strong lines emanating from the center of the Fourier transform indicate high frequencies along these lines. These frequencies correlate to the edges throughout the ram due to the sharp contrast between the different and distinct color transitions in the image. Due to the relative complexity of the image, it is extremely difficult to correlate the frequencies and fully interpret its transform. Humans are not used to having to interpret signals in frequency domain[2]. The frequency key at the bottom identifies the range of frequencies of the Fourier transform. The min frequency occurs at 0.

Convolution

Convolution in image filtering is a mathematical operation of combining two functions to form a third function. More specifically, convolution is an integral that defines the amount of overlap of one function as it is shifted over another function[7]. The convolution for 2D image filtering can be represented as:

$$g(x,y) = (f * h)(x,y)$$

$$g(x,y) = \sum_{m=-a}^{a}\sum_{n=-b}^{b} f(x-m, y-n)\cdot h(x,y)$$

In this equation, f is the image, h is the kernel (or filter), and g is the convolved image. The values of a and b are determined based on the size of h, and it is noted that the kernel h is usually smaller than the image, f [2]. The convolution computation involves flipping the kernel h both horizontally and vertically and then sliding it across the image f, computing the sum of products at each position to form the output image g[8]. All pixels from the summation form the image g.

The importance of convolution in image filtering relates the ability to perform operations in the frequency domain, which can computationally be more efficient than in the spatial domain. Thus, convolution is fundamental for image filtering, allowing for blurring, sharpening, and edge detection operations, all which result from using different kernels.

Convolution's significance in relation to the Fourier Transform stems from one key relation, the Convolution Theorem. This theorem states that the convolution in the spatial domain is equivalent to multiplication in the frequency domain[7]. Mathematically:

$$\mathcal{F}\{(f * h)(x,y)\} = \mathcal{F}\{f(x,y)\}\cdot\mathcal{F}\{h(x,y)\}$$

For two functions $f(x,y)$ and $h(x,y)$, their convolution in the spatial domain is equal to the production of their Fourier Transforms in the Frequency domain. This theorem allows for the mass simplification of image operations with large kernels by utilizing simple multiplication in the frequency domain[6].

FOURIER AND CONVOLUTION IN PRACTICE

Fourier analysis allows for image representation in terms of frequencies, rather than individual pixel values. The conversion of images to the frequency domain eliminates the often-inefficient process of looking at each single pixel. Typical filters applied in the frequency domain include low pass, high pass, and Gaussian filters. These filters are applied to images in the frequency domain via multiplication. Ideal low pass of high pass filters seeks to discriminate or enhance certain frequencies of the image. High frequencies correspond to rapid changes in intensity or color values in an image which low frequencies correspond to the slow changes in intensity or color values. High frequencies can often be mapped to the fine details of an image such as edges or sharp transitions[5]. Low frequencies are often found in the smooth areas of an image such as gradual transitions. Manipulating such frequencies is an efficient way to alter the contents of an image.

A low pass filter, shown in Figure 2, allows for the passage of low frequencies and eliminates the higher frequencies. In the spatial domain this translates to the reduction of sharp edges or noise and the continuance of smooth variations[10]. Typically, low pass filters are utilized to reduce high frequency noise components or to create a blurring effect by smoothing out the image with the enhancement of the low frequencies.

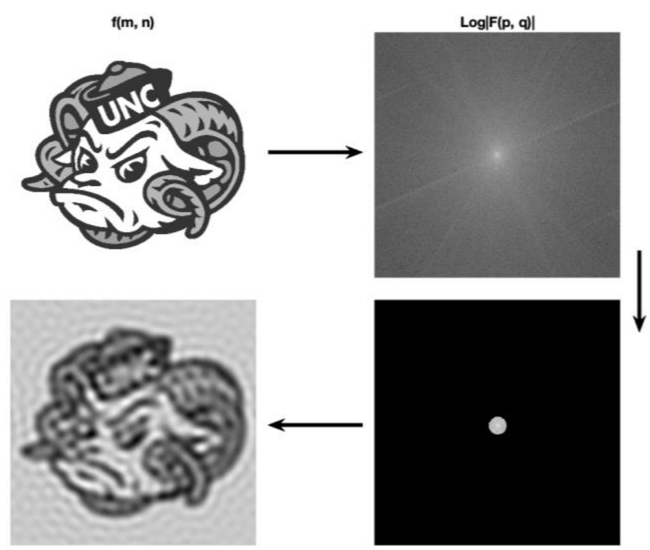

Figure 2: Application of a Low Pass Filter to the Fourier Transform of the UNC Ram Image. Originally, the ram's head features many

sharp edges and transitions, particularly in the horns and facial area. By applying the low pass filter in the frequency domain, all frequencies except for the low ones are eliminated. The resulting image, obtained after performing the inverse Fourier transform exhibits a smoother appearance, with diminished sharp contrasts, exemplifying the smoothing effect of low pass filtering.

A High Pass filter, Figure 3, operates opposite of that compared to the Low Pass filter. The High Pass filter allows for the transmission of the higher frequencies and reduces the low frequencies. Its application typically results in the accentuation of edges or fine details and decreases or eliminates the gradual intensity shifts[10]. A High Pass filter is typically utilized for specific edge enhancement for object detection is computer vision applications or image sharpening[10].

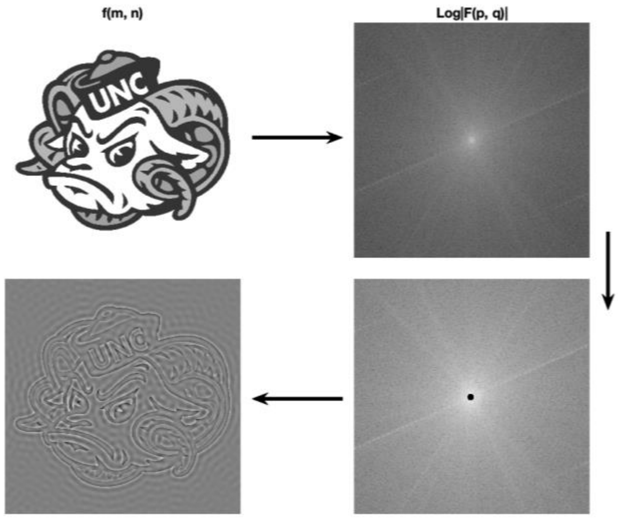

Figure 3: Application of a High Pass Filter to the Fourier Transform of the UNC Ram Image. The ram's head is filled by various solid colors and are essentially the only smooth transitions in the image. Thus, upon application of the high pass filter in the frequency domain, all frequencies except for the high ones are eliminated. The resulting image, obtained after performing the inverse Fourier transform exhibits almost a uniform color with the sole features of the image being the edges, with increased sharp contrasts.

An additional type of low pass filter smooths images utilizing the Gaussian function and is known as the Gaussian filter. Application of the Gaussian filter results in more gradual transitions compared to typical low pass filters[2]. Typical utilization for gaussian filters includes smoothing, blurring, or noise reduction.

The Gaussian filter is defined by its use of the Gaussian function, Figure 4. This filter smoothly blurs images with a bell-shaped curve. This leads to more subtle transitions compared to other low pass filters, avoiding the abrupt changes. What sets the Gaussian filter apart is its ability to reduce noise and fine details without introducing common artifacts like ringing[2]. This makes it particularly useful in pre-processing for computer vision, where maintaining the main features of an image is essential. Application of the Gaussian filter results in more gradual transitions compared to typical low pass filters[2].

Its flexibility is improved by the ability to control the extent of blurring through the standard deviation parameter of the Gaussian function[11]. Gaussian filters are often employed in edge detection and reducing noise to generate more accurate results for edge detection algorithms.

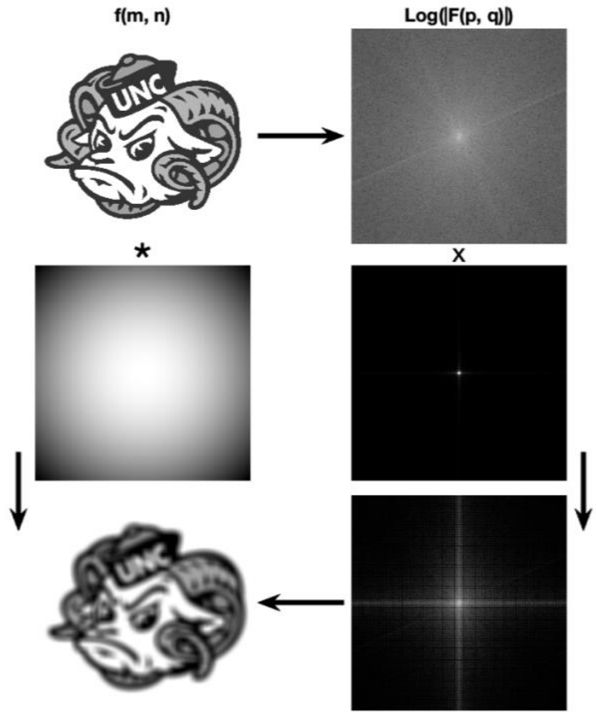

Figure 4: A Gaussian smoothing filter is applied to the Fourier Transform of the UNC Ram Image. Like a low pass filter, the gaussian-shaped pulse filter removes the high frequency noise. The gaussian filter smooths the ram image by decreasing and eliminating the high frequencies that define the contrasts and sharp features of the image. The original image is displayed in the top left. The first path to the final image involves the convolution of the original image with the Gaussian filter in the spatial domain, exhibited by the left column. The second path to the final image involves multiplication of the gaussian filter and the original image in the frequency domain, and then the computation inverse Fourier transform to arrive at the final image.

DECONVOLUTION AND MOTION BLUR

Motion Deblur Deconvolution

For image processing, blurring can be modeled as the convolution of an image $f(x, y)$ with a Point Spread Function (PSF) $h(x, y)$, example shown in Figure 5. Typically, a point squared function is estimated utilizing gyroscopes and accelerometers[2]. Utilizing the previously defined Convolution Theorem, the blurry image $g(x, y)$ with randomly generated Noise, $n(x, y)$, is mathematically expressed:

$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

In the frequency domain, after taking the Fourier Transform of both sides of the equation, the convolution operation in the spatial domain becomes a multiplication operation.

The functions correspond to their respective Fourier Transformations $G(u, v), F(u, v), H(u, v), and\ N(u, v)$. Utilizing this relationship, recovery of the original image, $F(u, v)$, in the frequency domain is given by:

$$\bar{F}(u, v) = \frac{G(u, v)}{H(u, v)}$$

where $\bar{F}(u,v)$ is given instead of $F(u,v)$ as the exact original image in the frequency domain cannot be recovered due to the unknown noise[8]. The recovered image $\bar{f}(x,y)$ is then found by taking the inverse Fourier Transform of $\bar{F}(u,v)$. Typically issues with this method arise due to the amplification of noise in the deconvolution process. The PSF function, simulating the motion blur, behaves like a low pass filter and amplifies the high frequncies[2]. As the added noise, $n(x,y)$, is high in frequency, this leads to its amplification and the need for some type of noise suppression. This concept is shown in Figure 6, where the Motion Blur Recovered Image is just random noise and not at all comparable to the original image.

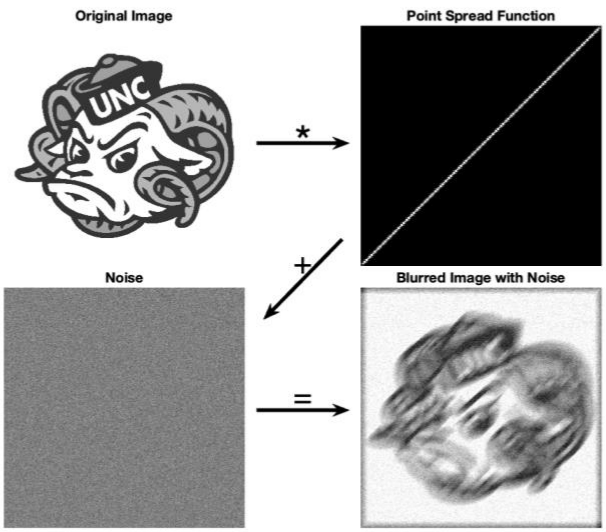

Figure 5: The process above shows the construction of a blurred UNC Ram image with noise. First, the Original Image is convolved with the PSF, representing the motion blur. After convolution, noise is introduced to the image in the form of random high frequencies. The resulting image is a combination of a blur, with a known kernel, and a combination of unknown high frequencies meant to represent noise. The introduced external noise will prove to present significant complications in the recovery of the initial image through normal deconvolution.

Wiener Deconvolution

Wiener deconvolution is a method that aims to reduce additive noise, $n(x,y)$. The implementation of the deconvolution methods remains highly similar to the typical motion deblur deconvolution with the exception of the addition of a constant K, which represents the noise-to-signal ratio. Often, K is estimated as some constant value to recover the original image with some graininess[2]. The deconvolution formula of the image in the frequency domain is given by:

$$\bar{F}(u,v) = \frac{H(u,v) \cdot G(u,v)}{|H(u,v)^2| + K}$$

The Wiener filter suppresses the noise by reducing the frequencies at locations where the PSF has relatively small values[9]. This process then hopes to avoid the amplification of noise typically resulting from basic motion deblur deconvolution.

Ultimately both the Motion Deblur Deconvolution and Wiener Deconvolution both utilize the Fourier Transformation to aid in mathematical manipulations and computations by operating in the frequency domain. A clear difference in their effectiveness with additional noise is shown in Figure 6.

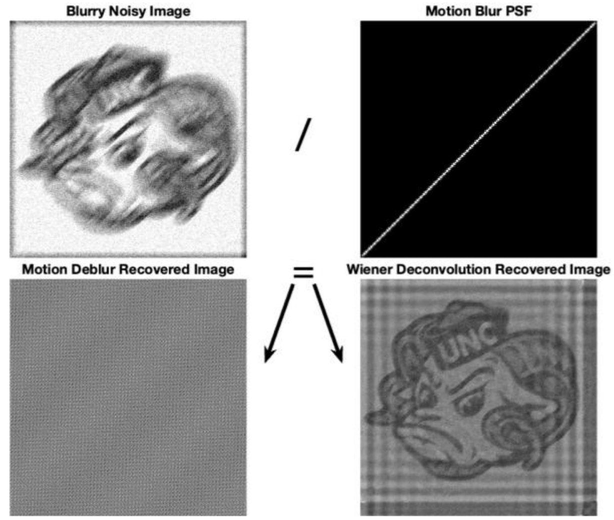

Figure 6: This set of images demonstrates the process of attempting to recover the original image from the blurred and noisy version created in Figure 5. The 'Motion Deblur Recovered Image' was produced using standard deconvolution, which involved dividing the Fourier Transform of the Blurry Noisy image by that of the Motion Blur PSF. However, this approach significantly amplified the high-frequency random noise post-inverse Fourier Transform (IFT), resulting in the recovered image being unrecognizable. This also highlights the inadequacy of standard deconvolution in noisy conditions. Conversely, the second method, employing Wiener deconvolution, substantially improved the quality of the recovered image, making . Due to the random nature of the noise, and the absence of an exact noise kernel, precise determination of the K value in Wiener deconvolution was challenging. This resulted in residual high-frequency noise, shown as streaks across the image, exemplifying the effects of random noise.

RECENT ADVANCES AND FUTURE PROSPECTS

Recently, the use of the Fourier Transform and Convolution has seen a major increase in its applicability with Artificial Intelligence (AI) image and video classification. Utilizing these fundamental mathematical methods alongside the ever-improving high-performance computing and algorithms, machine learning has allowed for more intelligent filtering techniques[12].

Other developments pertain to the field of medical imaging, where Fourier transform techniques are extremely important tools for MRI scans. These techniques allow for more detailed images and ultimately aid in a more accurate medical diagnosis. However, despite their usefulness, The Fourier Transform and Convolution methods continue to face difficulties when dealing with noise and image distortions. Potential solutions and ongoing research related to addresses these difficulties are aiming at combining traditional Fourier transformation methods with more recent non-linear image processing techniques[12].

CONCLUSION

The primary benefit of utilizing the Fourier Transform within image processing is its ability to transition components from the spatial to frequency domain. In the frequency domain, manipulation of frequency components with linear operations proves to be a much more efficient method than in the spatial domain for tasks such as edge enhancement or noise reduction. Convolution also maintains a key role in image processing, especially for image filtering techniques. Within the frequency domain, complex spatial domain integrals become simple products in the frequency space. Ultimately, convolution significantly applies to crucial edge detection and feature extraction for enhancing image quality throughout much of image processing.

However, it must be important to recognize the limitations in these techniques with their handling of non-linear image distortions. Ongoing research seeks to address these issues with machine learning and adaptive filers to uphold its integrity in many applications. The continual evolution of these fundamental mathematical tools and their adaptations with new applications and improvements proves their immortality in the field of image processing.

AUTHOR INFORMATION

Corresponding Author

John K. Goebel, Chapel Hill, NC, kenjo@ad.unc.edu

ABBREVIATIONS

DFT, Discrete Fourier Transform; IDFT, Inverse Discrete Fourier Transform; Point Spread Function, PSF; Artificial Intelligence

REFERENCES

(1) Kundu, R. Image processing: Techniques, types, & applications [2023] https://www.v7labs.com/blog/image-processing-guide#what-is-image-processing (accessed Nov 26, 2023).

(2) Nayar, S. K. Image Processing II https://fpcv.cs.columbia.edu (accessed Nov 26, 2023).

(3) Chellappa, R.; Rosenfeld, A. Image Processing. In *Encyclopedia of Physical Science and Technology*; Meyers, R. A., Ed.; Academic Press: San Diego, California, 2002; pp. 595–630.

(4) Kumar, N. Digital Image Processing Basics https://www.geeksforgeeks.org/digital-image-processing-basics/ (accessed Nov 26, 2023).

(5) Kulkarni, S. Understanding Image Filtering Techniques in Image Processing https://www.imageprovision.com/articles/understanding-image-filtering-techniques-in-image-processing(accessed Nov 26, 2023)

(6) Basavarajaiah, M. 6 basic things to know about convolution https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411 (accessed Nov 26, 2023).

(7) Colton, J. Convolutions https://physics.byu.edu/faculty/colton/docs/phy441-fall19/lecture-42-convolutions.pdf (accessed Nov 27, 2023).

(8) Bai, K. A comprehensive introduction to different types of convolutions in deep learning https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215 (accessed Nov 26, 2023).

(9) Deblur Images Using a Wiener Filter https://www.mathworks.com/help/images/deblurring-images-using-a-wiener-filter.html (accessed Nov 28, 2023).

(10) Minhaz. Lowpass, Highpass, Bandreject and Bandpass Filters in Image Processing. *Minhaz's Blog*, 2021.

(11) Fisher, R.; Wolfart, E.; Perkins, S.; Walker, A. Gaussian smoothinghttps//homeages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm#:~:text=Brief%20Description,(%60bell%2Dshaped%27)%20hump. (accessed Nov 28, 2023).

(12) Sarker, I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*2021, *2*.

APPENDIX

GitHub Repository for Created and Utilized Code
https://github.com/jogoebel/MATH528FINAL.git