

Yasemin Keskin's Portfolio



☎ +90 536 303 50 64

✉ keskinyasemin@gmail.com

🌐 [linkedin.com/in/keskinyasemin](https://www.linkedin.com/in/keskinyasemin)

🔗 github.com/keskinyasemin

Eastern Mediterranean University

M.Sc. Computer Engineering

B.Sc. Information Systems Engineering

B.Sc. Electrical and Electronic Engineering

Graduate in Computer Engineering with a M.Sc. degree. Work experience mainly based in the academic field focusing on computer vision, image processing and machine learning. Video game designer and character artist in free time. Enthusiastic about applying expertise and creativity to lead cutting-edge projects within the industry.

This portfolio includes my recent projects in Advanced Digital Signal Processing, Digital Image Processing and Computer Vision, which are listed below:

- Image Restoration using Wiener Filter
- Image Filtering in Frequency Domain
- Spoof Detection on Photo Attacks
- Car License Plate Recognition
- Autonomous Underwater Vehicle

Image Restoration using Wiener Filter

This project is focused on image restoration using Wiener filter to estimate the original image from the distorted image. The Wiener filter is based on minimum mean square error (MMSE) estimation algorithm, which aims to minimize the mean squared error between the estimated image and the original image. This filter is able to adapt various images and noise properties by using the power spectral density (PSD) of the image and the noise. The filtering process is done in the frequency domain to visualize the signal and the noise separation using MATLAB.

Lena image is used for testing the Wiener filtering. The original image, an image with PSF (motion blur) distortion at 10° and additive noise with variance of 0.02 and a mean of 0 are shown below in Figure 1.



Figure 1: Image Distortion

Fourier transform is applied to PSF filter and the distorted image. The effect of PSF filter to the original image can be clearly seen in Figure 2.

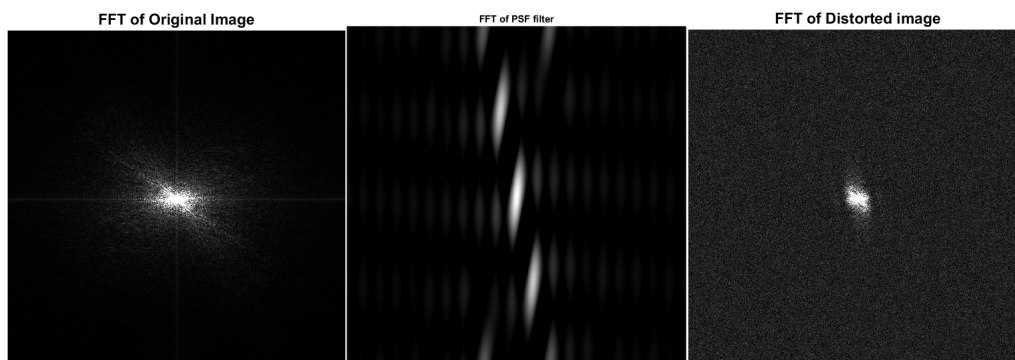


Figure 2: Fourier Spectrum of Original Image, PSF Filter and Final Distorted Image

The autocorrelation of the distorted image and the noise are calculated, and their Fourier transform are taken in order to obtain their PSD's. Then, noise to signal ratio is calculated by dividing PSD of noise by PSD of distorted image, which is used for the Wiener equation. The results of the Wiener filter method is shown in Figure 3.

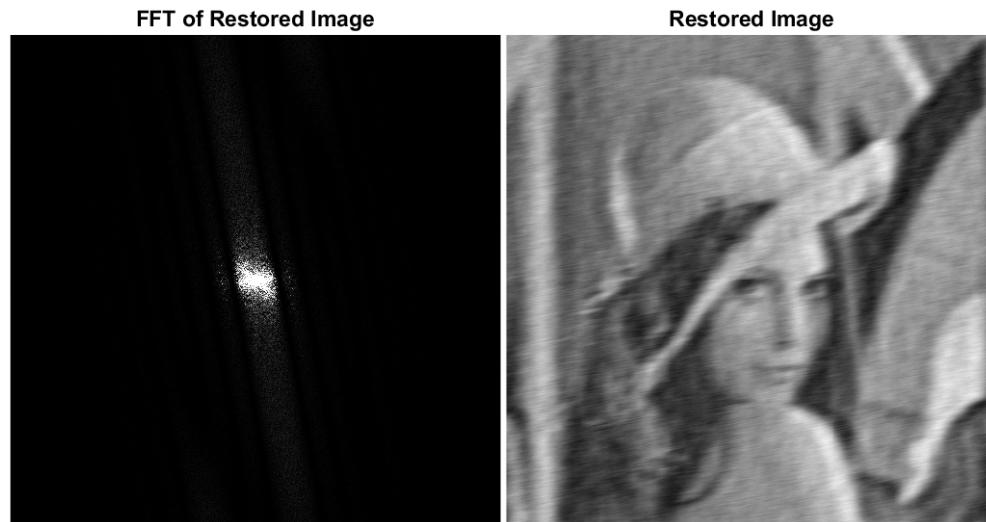


Figure 3: Restored Image

The frequency domain representation of the restored image exhibits sharper edges and smaller black motion distortion lines compared to the distorted image, indicating successful restoration and reduction of noise. This suggests that the filter effectively estimated the image in certain areas. While the restored image displays increased detail compared to the distorted version and mitigates motion blur effects, it still retains some degree of blurriness relative to the original image. This showcases image restoration processes, where the aim is to enhance image quality while maintaining overall coherence and clarity.

The comparison between the distorted and restored images are evaluated with PSNR and MSE metrics, where the original image is used as a reference. The increased PSNR value indicates improved image quality, while the slightly reduced MSE value further confirms the enhancement in image quality as shown in Table 1.

Table 1: Evaluation of Wiener Filter

Images\Metrics	PSNR	MSE
Distorted Image	15.9611	0.0253
Restored Image	16.7579	0.0211

Image Filtering in Frequency Domain

Image filters can be applied to smooth or sharpen images, enhancing certain features for viewers. In this project, the frequency domain of Ideal, Butterworth and Gaussian methods will be implemented for low pass, band pass and high pass filters to well-known grayscale image Lena (512×512) as shown in Figure 4. Cut off frequency for high pass and low pass frequencies are set to 75, and for band pass the frequency is set between 50 and 150.

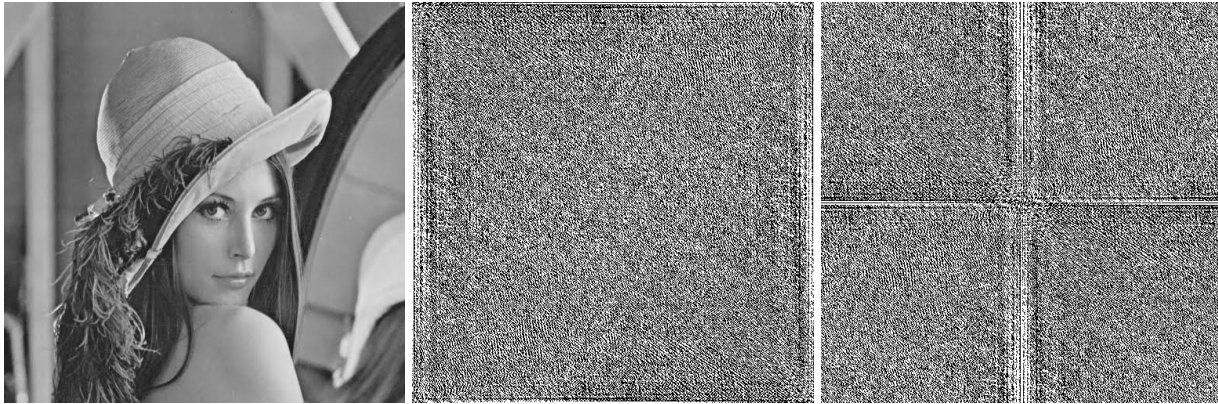


Figure 4: Lena, Fourier Spectrum of Lena and Inverse Fourier Transform of Lena respectively

In image processing, ideal, Butterworth, and Gaussian filters serve different purposes. Ideal filters offer a sharp frequency cutoff, effectively removing specific frequencies, though they may introduce potential artifacts known as ringing. Butterworth filters provide a wider transition band, preserving image quality while attenuating unwanted frequencies. Gaussian filters offer smoother results in noise reduction, blurring and edge detection while maintaining image quality.

The applications of low-pass, band-pass, and high-pass filters in image processing reveal their distinct characteristics and suitability for various tasks. The application of these filters are shown in Figure 5.

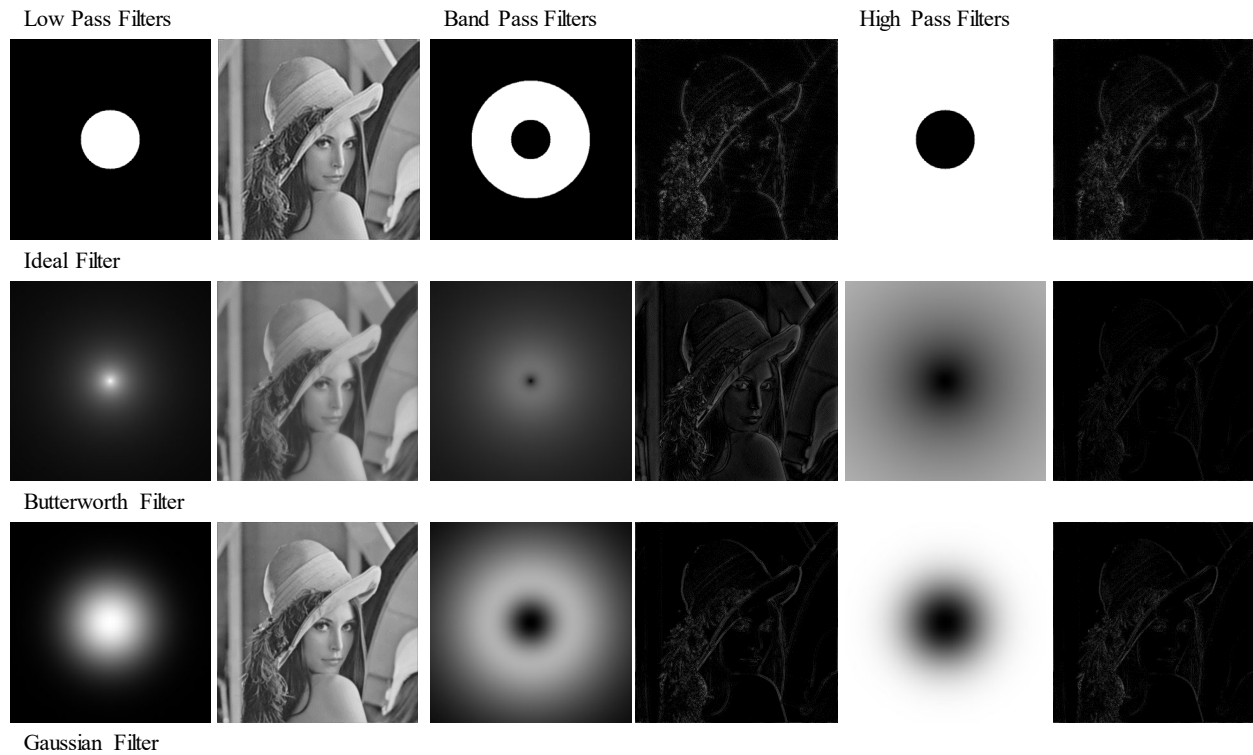


Figure 5: Ideal, Butterworth and Gaussian Filter Application for Low Pass, Band Pass and High Pass Filters

In low pass filtering, high-frequency signal is removed which results in blurred image. The ideal filter introduces a sinusoidal ring effect around edges due to its sharp cutoff. The Butterworth filter, results in a more blurred image. In contrast, the Gaussian filter maintains clearer details compared to the other filters, despite some blurring relative to the original image.

Band pass filtering aims to capture specific frequency bands, often highlighting edges or certain features in an image. When an ideal band pass filter is applied, the edges appear rough, indicating a less precise capture of details. The Butterworth filter, captures more features beyond just the edges, potentially due to its wider transition band. On the other hand, the Gaussian filter captures edges more accurately while presenting a smoother result than the ideal filter.

In high pass filtering, low-frequency noise is eliminated, enhancing edges and fine details. Both the ideal and Gaussian filters captured the edges while Butterworth filter loses more information in low frequency, resulting in a darker appearance compared to the other filters.

As a conclusion, image filters are valuable applications for noise reduction, image smoothing, edge detection and feature extraction.

Spoof Detection on Photo Attacks

Spoof detection in photo attacks involves analyzing image properties to differentiate between original and forged content. Employing techniques such as image forensics and machine learning can help to detect photo spoofing attacks.

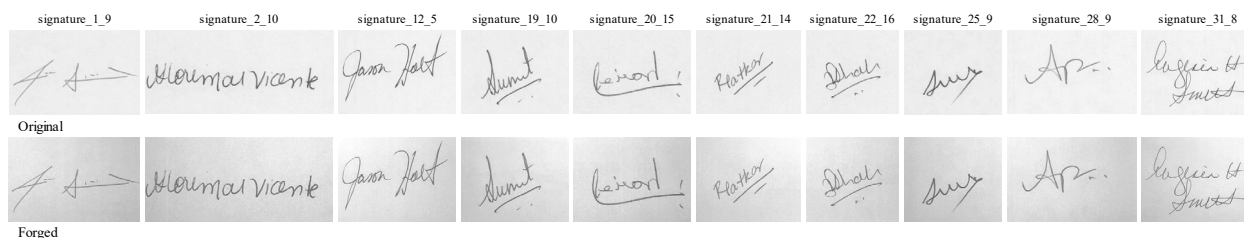


Figure 6: Original Signatures and Forged Signatures

The signatures from the CEDAR dataset and the forged signatures are shown in Figure 6. Forged images are created by scanning the printed copies of the signatures with the camera of Galaxy Note 8 using the Office Lens application.

One of the methods about fake biometric detection indicates that there is a quality difference between the original image and the forged image [1]. In this project, firstly, Gaussian filter with a kernel size of 3×3 and a standard deviation of 0.5 is applied as a low-pass filter to dataset. Then, the quality between the image and Gaussian image is calculated using PSNR, SSIM and SME quality measurements. Linear Discriminant Analysis (LDA) is applied to these measurements, which separates the original and forged classes. The data is classified by Random Forest Classifier. 20% of the dataset is separated as test data to measure the accuracy of the model.

Pair plot and correlation heatmap are employed for better insights in image quality metrics as shown in Figure 7. PSNR and SME values overlaps and SSIM value is a separable factor for original and forged signatures. PSNR has a strong correlation with SSIM while SSIM has weak correlation between SME.

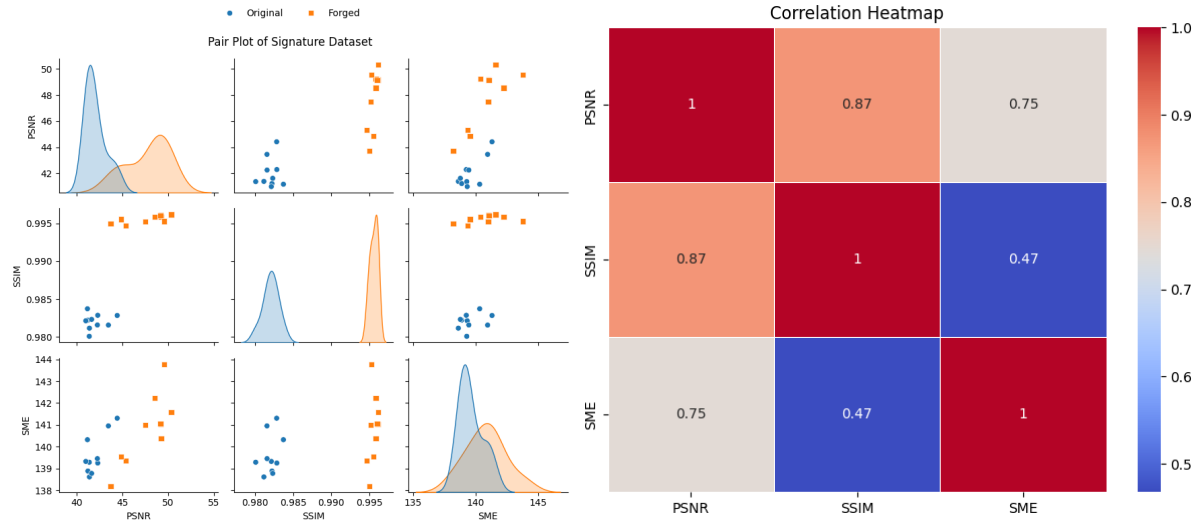


Figure 7: Image Quality Metrics Visualization

The confusion matrix after evaluation is shown in Table 2. The model predicted the test results with 100% accuracy for 4 test images.

Table 2: Confusion Matrix

Actual\Predicted	Original	Forged
Original	2	0
Forged	0	2

In conclusion, this report shows that the loss of quality produced by Gaussian filtering differs between original and forged signature images. With this result, we can easily check photo attacks on signature images using IQM parameters.

Car License Plate Recognition

License plate recognition helps us to identify the cars that have entered through the university gate and saves their plate number into our database along with the body temperature value of the driver and the local time. This section includes, capturing the image from camera and adjusting the image using image processing, and detecting the car plate, then reading and saving the detected car plate into a text file.

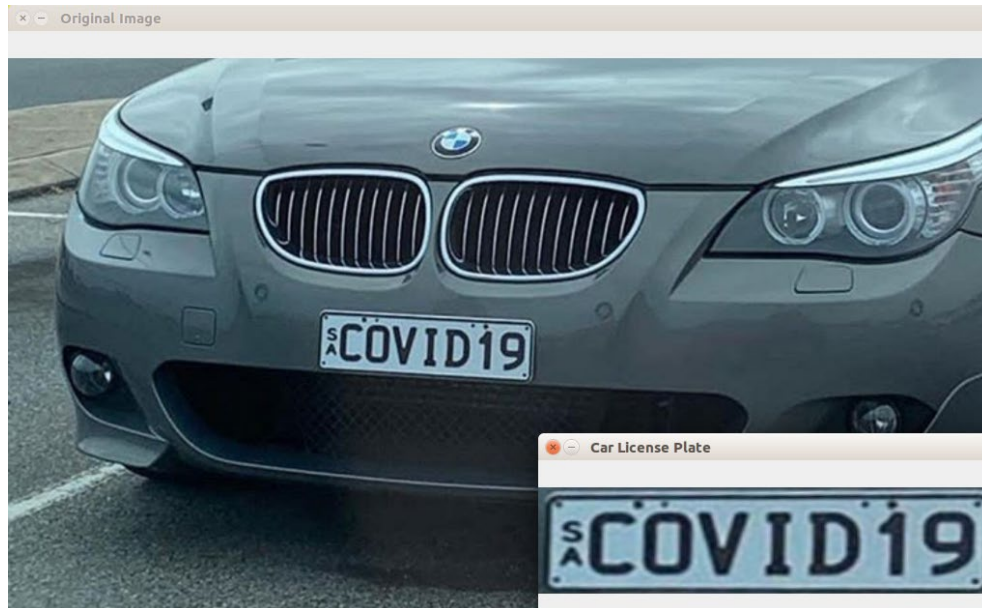


Figure 8: Car License Plate Detection

The Figure 8 above shows the result of the car license plate detection with Keras. As it can be seen, the output image is aligned perfectly horizontal even though car license plate is tilted in the original image. After cropping the image, the code continues with recognition of the characters in the plate.

OpenCV is used in car license plate character recognition that adjusts the image to make the characters readable by the pytesseract library. Pytesseract library is easier to implement in the code and runs faster compared to Keras models. So, it is efficient to use this library for car license plate recognition.



Figure 9: License Plate Image Processing

As depicted in Figure 9 above, the first step involves converting the image to grayscale, as it is easier to work with a single-layered image. Next, Gaussian blur is applied to reduce noise in the image. Subsequently, a threshold is applied to convert the image to a binary format. Then the algorithm finds the white areas that are separated from each other in this threshold image and labels each of them uniquely. Finally, fine adjustments are made to clear the background, resulting in clearer and more distinct characters. This process effectively separates and labels the characters in the image. The data of the car license plate and time is written in the text file. The miniature of university gate is tested as a simulation and the data recordings are saved as shown in Figure 10.

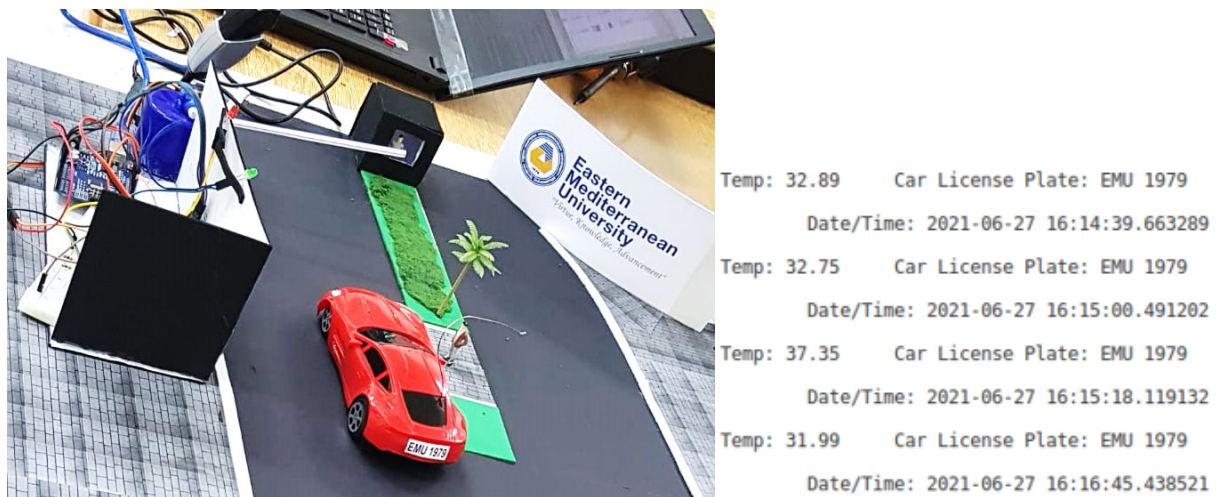


Figure 10: The Simulation and the Recording of the Data

Autonomous Underwater Vehicle

As a Software Team Member at EMU Aquabotics, I played a crucial role in advancing the capabilities of our Autonomous Underwater Vehicle (AUV) by implementing computer vision and image processing techniques using OpenCV, which enhanced the AUV's capability to navigate and interpret complex underwater environments. The impact of these enhancements resulted in our team's acceptance into prestigious competitions such as SAUVC 2020 [2] and Teknofest 2020 [3]. The news page on the Eastern Mediterranean University website honored us for our achievements and our team members, as shown in Figure 11.



Figure 11: Team Members of EMU Aquabotics

References

- [1] Galbally, J., Marcel, S., & Fierrez, J. (2014). Image quality assessment for fake biometric detection: Application to Iris, fingerprint, and face recognition. *IEEE Transactions on Image Processing*, 23(2), 710-724. <https://doi.org/10.1109/tip.2013.2292332>
- [2] EMU Aquabotics. (2020, February 25). *EMU Aquabotics Qualification Video for SAUVC 2020*. <https://aquabotics.emu.edu.tr/en/competitions/sauvc-2020>
- [3] Eastern Mediterranean University (2020, May 5). *The EMU "Aquabotics" robotics team comfortably pass the first stage of eliminations at the Teknofest 2020 underwater systems competition*. <https://www.emu.edu.tr/en/news/news/the-emu-aquabotics-robotics-team-comfortably-pass-the-first-stage-of-eliminations-at-the-teknofest-2020-underwater-systems-competition/1206/pid/3408>