به نام خدا

گزارش پروژه سیگنال

پروژه ۸

گروه ۱۰

احسان بهنامی ۴۰۰۱۰۰۸۴۳
سید علیرضا علمدار ۴۰۰۱۰۱۶۳۴
شهناز پوربرقی ۴۰۰۱۰۰۸۷۶

Before displaying the project outputs, we provide explanations about the requested noises and filters.

Impulse noise:

Impulse noise, also known as spike or burst noise, is a type of interference that occurs as sudden and brief disturbances in a signal. These disturbances can be random and are characterized by their short duration and high amplitude.

Here are some key points about impulse noise:

1. Nature of the Noise:

- Impulse noise is often caused by external factors, such as electromagnetic interference, power surges, or sudden electrical discharges.
- It can manifest as sudden spikes or glitches in audio signals, pixel distortions in images, or errors in data transmissions.

2. Effects on Signals:

- In audio signals, impulse noise can create popping or clicking sounds.
- In image signals, it may lead to pixelated or distorted areas in the image.
- In data transmissions, impulse noise can cause errors and affect the accuracy of the received information.

3. Mitigation Techniques:

- To reduce the impact of impulse noise, various techniques are employed depending on the context.
- In communication systems, error-correction codes and error-detection mechanisms can be used.
- In audio processing, filters may be applied to suppress or remove unwanted impulse noise.
- Surge protectors and shielding can help prevent impulse noise caused by power surges or electromagnetic interference.

4. Common Examples:

- In audio systems, a sudden pop or crackle in the sound can be caused by impulse noise.
- In digital imaging, a single bright pixel appearing suddenly in an image could be due to impulse noise.
- In data communications, corrupted bits or packets may result from impulse noise

Additive noise:

Additive noise is another type of unwanted interference that affects signals. Unlike impulse noise, which occurs suddenly and briefly, additive noise is characterized by being continuously present throughout the duration of the signal. It is called "additive" because it adds to the original signal, distorting it in the process.

Here are some key points about additive noise:

1. Nature of the Noise:

- Additive noise is often introduced externally and can be caused by various factors such as electronic components, atmospheric conditions, or other environmental influences.
- It is considered a random and continuous disturbance that affects the entire signal.

2. Effects on Signals:

- Additive noise introduces random variations to the signal, leading to a degradation in signal quality.
- In audio signals, it can manifest as a hiss, hum, or background noise.
- In images, it may appear as a grainy or speckled texture.
- In data communications, additive noise can lead to errors in the received data.

3. Mitigation Techniques:

- Filtering techniques, such as low-pass filters, can be used to reduce high-frequency components associated with additive noise.
- Signal processing algorithms, like noise reduction or equalization, can be employed to minimize the impact of additive noise in audio or image signals.
- In communication systems, error correction codes and modulation schemes may be used to improve the robustness of data transmission against additive noise.

4. Common Examples:

- In analog audio systems, the background hiss or hum heard when no audio is playing is an example of additive noise.
- In digital images, the graininess seen in low-light conditions can be caused by additive noise.
- In wireless communication, radio signals may experience additive noise due to atmospheric conditions or interference from other electronic devices.

Salt and pepper noise:

Salt and pepper noise is a specific type of image noise characterized by the appearance of random, isolated pixels in an image that have either very high intensity (resembling white salt grains) or very low intensity (resembling black pepper grains).
This type of noise can significantly degrade the visual quality of images.
Here are some key points about salt and pepper noise:

1. Causes:

- Salt and pepper noise often occurs during the process of image acquisition or transmission. It can be caused by faults in image sensors, communication channels, or other environmental factors.

2. Appearance:

- The noise manifests as randomly occurring bright pixels (salt) and dark pixels (pepper) scattered throughout the image.
- Salt and pepper noise can be particularly problematic in low-light conditions or when images are transmitted over noisy channels.

3. Effects on Images:

- The presence of salt and pepper noise can obscure important details in images and reduce overall image quality.

- The impact is especially noticeable in regions where the noise pixels are present, as they stand out from the rest of the image.

4. Mitigation Techniques:
    - Various image processing techniques can be used to reduce or eliminate salt and pepper noise.
    - Median filtering is a common method for removing this type of noise. It involves replacing each pixel's value with the median value of the surrounding pixels.
    - Adaptive filters and morphological operations are also employed to deal with salt and pepper noise while preserving image details.

5. Applications:
    - Salt and pepper noise can affect images in various domains, including medical imaging, satellite imagery, and digital photography.
    - In medical images, the noise can impact the accuracy of diagnostic tools, making noise reduction techniques crucial.
    - In digital photography, images captured in challenging conditions may exhibit salt and pepper noise, requiring post-processing for improvement.

Shot noise:

Shot noise, also known as photon noise in the context of light, is a type of random variation in a signal that arises from the quantized nature of particles, such as photons or electrons. This noise is particularly noticeable in situations where the number of particles is low, leading to statistical fluctuations in the observed signal.

Here are key points about shot noise:

1. Nature of the Noise:
    - Shot noise is a fundamental aspect of systems involving the emission or detection of discrete particles, such as light (photons) or electrons.
    - It results from the random arrival of these particles at a detector or sensor.

2. Characteristics:
    - The intensity of shot noise follows statistical properties described by a Poisson distribution, which is a probability distribution for the number of events occurring within a fixed interval.
    - The discrete nature of particles contributes to random fluctuations in the signal, leading to variations in intensity.

3. Effects on Signals:
    - In practical terms, shot noise manifests as random fluctuations in the observed signal intensity.
    - For example, in low-light conditions, where the number of photons is limited, shot noise becomes more pronounced and can affect the quality of an image or signal.

4. Applications:
    - Shot noise is encountered in various scientific and technological fields, including optics, electronics, and imaging.

- In photography, shot noise is apparent in low-light conditions, affecting the quality of images captured with limited light.

5. Mitigation Techniques:

- Complete elimination of shot noise is not possible due to its fundamental nature.
- Increasing the number of particles (e.g., using brighter light sources or increasing signal strength) can help reduce the relative impact of shot noise.
- Signal processing techniques, such as filtering or post-processing algorithms, may be applied to mitigate the effects of shot noise in certain applications.

Speckle noise:

Speckle noise is a type of noise that appears as granular or salt-and-pepper-like interference in images. It often occurs in images acquired by imaging systems that use coherent illumination, such as laser or ultrasound imaging. Speckle noise can degrade image quality and make it challenging to discern fine details. Understanding and mitigating speckle noise is crucial in various fields, including medical imaging, remote sensing, and industrial applications.
Here are key points about speckle noise:

1. Nature of the Noise:

- Speckle noise is a form of multiplicative noise that results from the interference of coherent waves. Coherent waves have a fixed phase relationship, leading to constructive and destructive interference patterns.

2. Appearance:

- Speckle noise manifests as a random pattern of bright and dark pixels or grains, resembling the appearance of grains in a speckled texture.
- It can significantly impact the visual quality of images, making them appear grainy or speckled.

3. Causes:

- Speckle noise is commonly encountered in imaging systems that use coherent illumination sources, such as lasers or ultrasound.
- The interference of coherent waves, combined with the random nature of the scattering or reflection of waves from surfaces, leads to the formation of speckle patterns.

4. Effects on Images:

- Speckle noise can obscure fine details in images, making it challenging to analyze or interpret visual information.
- In medical imaging, such as ultrasound or laser imaging, speckle noise can affect the clarity of diagnostic images.

5. Mitigation Techniques:

- Various techniques are employed to reduce or mitigate speckle noise:
- **Filtering**: Spatial filters, such as median filters or adaptive filters, can be used to smooth out speckle noise while preserving image details.
- **Statistical Approaches**: Statistical methods, including despeckling algorithms based on statistical modeling of speckle, are used to reduce noise.
- **Multi-Look Processing**: In remote sensing applications, acquiring multiple looks of the same scene and averaging them can help reduce the impact of speckle.

6. Applications:

- Speckle noise is encountered in a range of imaging applications, including medical ultrasound imaging, synthetic aperture radar (SAR) imagery, and laser-based imaging systems

Uniform additive noise:

Uniform additive noise is a type of noise that is characterized by the addition of a constant value, randomly selected from a uniform distribution, to each pixel in an image or each point in a signal. The uniform distribution means that each possible value within a given range has an equal probability of being chosen. This type of noise is common in various imaging and signal processing applications and can affect the overall quality and appearance of the data.
Here are key points about uniform additive noise:

1. Nature of the Noise:

- Uniform additive noise is considered an additive noise model because it involves adding a constant value to each pixel or point in a signal.
- The constant value is randomly selected from a uniform distribution, meaning that any value within a specified range is equally likely to be added.

2. Mathematical Representation:

- If X represents the original pixel value and U (a, b) is a random variable uniformly distributed between a and b, then the pixel value with uniform additive noise (Y) can be expressed as $Y = X + U(a, b)$.

3. Effects on Images/Signals:

- The impact of uniform additive noise is a consistent shift in pixel values across the entire image or signal.
- This type of noise can introduce a constant background offset, making it challenging to distinguish the original content, especially in regions with low contrast.

4. Mitigation Techniques:

- Clipping or Thresholding: Values outside the valid range may be clipped or thresholded to prevent them from exceeding certain limits.
- Filtering: Spatial filters or statistical filters can be applied to reduce the impact of additive noise while preserving the essential features of the image or signal.
- Preprocessing: Applying techniques such as normalization or background subtraction can help mitigate the effects of uniform additive noise.

5. Applications:

- Uniform additive noise can be encountered in various imaging scenarios, including digital photography, medical imaging, and remote sensing.
- In digital communication systems, additive noise may also be modeled as uniform to simulate certain types of channel impairments.

Gaussian noise:

Gaussian noise, also known as normal distribution noise, is a type of statistical noise that follows a Gaussian distribution (bell-shaped curve) when plotted in a histogram. It is a common form of random noise that affects signals, images, and data in various fields, such as communication systems, image processing, and scientific measurements.

Here are key points about Gaussian noise:

1. Nature of the Noise:

- Gaussian noise is characterized by random variations in signal amplitude that follow a Gaussian distribution.
- It is considered additive noise, meaning it is added to the original signal.

2. Mathematical Representation:

- If X represents the original pixel value and N $(0,\sigma^2)$ is a random variable following a Gaussian distribution with mean O and variance σ^2, then the pixel value with Gaussian noise (Y) can be expressed as Y = X + N$(0, \sigma^2)$.

3. Statistical Properties:

- The Gaussian distribution is fully described by its mean (average) and variance (spread).
- The central limit theorem suggests that the sum of a large number of independent, identically distributed random variables (such as pixel values with Gaussian noise) tends to follow a Gaussian distribution.

4. Effects on Images/Signals:

- Gaussian noise introduces randomness to the signal amplitudes, leading to a smooth distribution of noise values.
- It is often characterized by a relatively even distribution of bright and dark pixels, which can affect the overall appearance of the image.

5. Mitigation Techniques:

- **Filtering**: Gaussian noise can be reduced using various spatial filters or smoothing techniques, such as Gaussian smoothing or median filtering.
- **Statistical Methods**: Statistical methods, like mean or median filtering, can be effective in reducing the impact of Gaussian noise.
- **Signal Processing**: Techniques such as Wiener filtering or adaptive filtering may be applied to mitigate the effects of Gaussian noise.

6. Applications:

- Gaussian noise is encountered in a wide range of applications, including digital image processing, signal processing, and communication systems.
- In measurements and scientific experiments, Gaussian noise often represents the random errors associated with measurements.

Poisson noise:

Poisson noise is a type of statistical noise that arises from the inherent randomness in the number of events occurring within a fixed interval of time or space. It is particularly relevant in situations where events are rare and unpredictable. Poisson noise often manifests in imaging and measurement systems where the detection of discrete events, such as photons or particles, is involved.

Here are key points about Poisson noise:

1. Nature of the Noise:

- Poisson noise is a form of statistical noise that follows a Poisson distribution.
- It is a type of counting statistics noise, and its characteristics are closely tied to the probability of observing a certain number of events in a given interval.

2. Mathematical Representation:

- If X represents the true number of events expected to occur, then the number of observed events (Y) follows a Poisson distribution with mean X. The probability mass function is given by P(Y = k) = $\frac{e^{X}\, X^{k}}{k!}$.

3. Causes:

- Poisson noise is commonly encountered in situations where events are rare, discrete, and independent. Examples include the detection of photons in imaging systems or the occurrence of radioactive decay events.

4. Effects on Images/Signals:

- In imaging, Poisson noise often results in variations in pixel intensity due to the random arrival of photons.
- The impact of Poisson noise is more pronounced in low-light conditions, where the number of detected photons is limited.

5. Mitigation Techniques:

- **Noise Reduction Methods**: Techniques such as filtering or averaging can help mitigate the effects of Poisson noise by reducing its impact on the overall signal.
- **Statistical Techniques**: Statistical methods, such as maximum likelihood estimation, can be employed to estimate the true underlying signal and reduce the influence of Poisson noise.
- **Increase Signal Intensity**: In imaging applications, increasing the intensity of the signal (e.g., using more light) can help reduce the relative impact of Poisson noise.

6. Applications:

- Poisson noise is prevalent in various scientific fields, including medical imaging (such as positron emission tomography - PET), astronomy (in the detection of light from distant stars), and particle physics (in the detection of particle events).

Periodic noise:

Periodic noise is a type of repetitive and predictable variation that occurs in signals or images. It is characterized by the presence of distinct patterns or regular oscillations at specific frequencies within the signal. This type of noise can be introduced by various sources and can significantly impact the quality of signals and images.

Here are key points about periodic noise:

1. Nature of the Noise:

- Periodic noise exhibits a regular and repeating pattern in the signal or image.
- It is characterized by variations that occur at specific frequencies or harmonics.

2. Causes:

- Periodic noise can be introduced by various sources, including electromagnetic interference, power supply fluctuations, vibrations, and mechanical issues in imaging or measurement devices.
- Certain electronic components or faulty circuits may also introduce periodic noise.

3. Frequency Characteristics:

- Periodic noise can manifest at specific frequencies or harmonics, and its spectral analysis often reveals peaks at these frequencies.
- The frequency content of the noise is typically related to the source that introduces it.

4. Effects on Images/Signals:

- The presence of periodic noise can distort the original signal or image, making it challenging to extract accurate information.
- In images, periodic noise may result in noticeable patterns or artifacts, affecting image clarity.

5. Mitigation Techniques:

- **Filtering**: Frequency domain filters, such as notch filters or bandstop filters, can be used to target and suppress specific frequencies associated with periodic noise.
- **Frequency Analysis**: Identifying the frequency components of the noise through spectral analysis can guide the selection of appropriate filtering techniques.
- **Isolation and Elimination**: Addressing the root cause of the noise, such as shielding electronic components or stabilizing equipment, can be an effective way to eliminate periodic noise.

6. Applications:

- Periodic noise can impact various applications, including medical imaging, telecommunications, audio processing, and scientific measurements.
- In medical imaging, periodic noise may affect the quality of images obtained through MRI or CT scans.

Linear smoothing filters:

Linear smoothing filters, also known as linear filters or convolution filters, are widely used in image processing and signal processing to reduce noise, blur, or highlight certain features in a systematic way. These filters operate by applying a convolution operation between the filter kernel (or mask) and the input image or signal. The purpose of linear smoothing filters is to average or combine neighboring pixel values, resulting in a smoothed or blurred output.
Here are some commonly used linear smoothing filters:

1. Box Filter (or Mean Filter):

- The box filter assigns equal weights to all pixels within the filter kernel.
- It computes the average of pixel values in the neighborhood, effectively smoothing the image.
- The formula for the box filter is: $I_{smoothed}(x, y) = \frac{1}{k \times k} \sum_{i=-k/2}^{k/2} \sum_{i=-k/2}^{k/2} I(x + i, y + j)$

2. Gaussian Filter:

- The Gaussian filter uses a 2D Gaussian distribution as the filter kernel.

- It assigns higher weights to pixels near the center of the kernel, resulting in a smoother and more natural blur.
- The formula for the Gaussian filter is: $I_{smoothed}(x, y) = \sum_{i=-k/2}^{k/2} \sum_{i=-k/2}^{k/2} I(x + i, y + j)$
- Here, G(i, j) is the 2D Gaussian distribution.

3. Median Filter:

- The median filter replaces each pixel value with the median value of its neighborhood.
- It is particularly effective in preserving edges while reducing salt-and-pepper noise.
- This filter is non-linear, but it's often mentioned alongside linear smoothing filters due to its widespread use for noise reduction.

4. Bilateral Filter:

- The bilateral filter takes into account both spatial and intensity differences.
- It applies different weights based on both the spatial distance and the intensity difference between pixels.
- This filter is effective in smoothing images while preserving edges.

Linear smoothing filters are versatile tools used in various image and signal processing applications. The choice of filter depends on the specific requirements of the task at hand, including the level of smoothing desired, preservation of edges, and the type of noise to be addressed. These filters play a crucial role in enhancing image quality and facilitating further analysis or interpretation of signals and images.

Wiener filtering:

Wiener filtering is a signal processing technique used for the restoration or enhancement of signals, images, or other data that may be corrupted by additive noise. The goal of Wiener filtering is to minimize the mean square error between the estimated signal and the true signal by utilizing both the knowledge of the signal and the statistical properties of the noise.
Here are key points about Wiener filtering:

1. Mathematical Formulation:

- The Wiener filter is based on the Wiener-Hopf equation and is often expressed in the frequency domain. For a 1D signal, the Wiener filter H (f) is given by: $H(f) = \frac{S_f}{S_f + N_f}$ where S_f is the power spectral density of the signal, and N_f is the power spectral density of the noise.

2. Application in Image and Signal Processing:

- Wiener filtering is commonly used in situations where a signal is corrupted by additive noise, and some knowledge of the signal and noise characteristics is available.
- In image processing, Wiener filtering can be employed to reduce the impact of noise while preserving important details.

3. Parameters:

- The effectiveness of Wiener filtering depends on the accurate estimation of the signal and noise power spectral densities.

- The filter performance is also influenced by a parameter, often denoted as a, which allows for a trade-off between noise reduction and preservation of signal details.

4. Limitations:

- Wiener filtering assumes that the signal and noise are stationary, meaning that their statistical properties do not change with time.
- It is most effective when there is a good estimate of the signal and noise characteristics. In cases of rapidly changing signals or complex noise, the performance of Wiener filtering may degrade.

5. Implementation:

- In practice, Wiener filtering is often applied in the frequency domain using Fast Fourier Transform (FFT) operations.
- The Wiener filter can be implemented using convolution in the spatial domain or element-wise multiplication in the frequency domain.

6. Applications:

- Wiener filtering finds applications in various fields, including image processing, audio signal processing, communication systems, and medical imaging.
- In medical imaging, Wiener filtering is used to enhance the quality of images obtained through techniques such as magnetic resonance imaging (MRI) or positron emission tomography (PET).

Wiener filtering is a powerful tool for improving the quality of signals and images corrupted by additive noise. While it requires some knowledge about the signal and noise characteristics, when used appropriately, it can significantly enhance the fidelity of processed data.

Median filter:

The median filter is a non-linear digital signal processing technique used for smoothing or reducing noise in a signal or image. Unlike linear filters (such as the mean filter or Gaussian filter), the median filter is particularly effective in preserving edges while removing outliers or impulse noise.

Here are key points about the median filter:

1. Basic Operation:

- The median filter operates by sliding a window (also known as a kernel) over the signal or image.
- For each position of the window, the pixel values within the window are sorted, and the median value is chosen as the output for that position.

2. Noise Reduction:

- The median filter is robust to outliers and noise. It replaces each pixel value with the median of the neighboring values, making it less sensitive to extreme values.
- This property makes it effective for reducing salt-and-pepper noise or other types of impulse noise.

3. Preservation of Edges:

- Unlike linear smoothing filters, the median filter does not blur or smear edges in the image. It is capable of preserving sharp transitions and fine details.
- This makes the median filter suitable for applications where edge preservation is crucial.

4. Size of the Window:

- The effectiveness of the median filter is influenced by the size of the window or kernel.
- A larger window provides more smoothing but may blur smaller details, while a smaller window preserves finer details but may not reduce noise as effectively.

5. Application in Image Processing:

- Median filtering is commonly used in image processing applications, especially for reducing noise in images captured under challenging conditions.
- It is frequently employed in scenarios where preserving edge details is essential, such as in medical imaging or remote sensing.

6. Computational Efficiency:

- The computational complexity of the median filter is higher than that of linear filters, especially as the window size increases.
- Algorithms like the quickselect algorithm are often used to efficiently find the median without sorting the entire window.

7. Limitations:

- While effective for certain types of noise, the median filter may not be suitable for all noise reduction scenarios, particularly when the noise is not impulse-like.
- It may not perform well in the presence of correlated noise or when the underlying signal has a complex structure.

The median filter is a valuable tool in signal and image processing, especially when dealing with images affected by salt-and-pepper noise or other types of impulse noise.
Its ability to preserve edges makes it suitable for various applications where noise reduction without significant loss of detail is crucial.

Gaussian filter:

A Gaussian filter is a linear filter commonly used in image processing and signal processing to smooth, blur, or reduce noise in data. It derives its name from the fact that its kernel is based on the Gaussian distribution. Gaussian filters are particularly useful for applications where a smooth and continuous transition between intensities is desired.
Here are key points about the Gaussian filter:

1. Kernel and Mathematical Representation:

- The Gaussian filter uses a two-dimensional Gaussian distribution as its kernel.
- The 2D Gaussian function is defined as: $G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$
- The filter kernel is formed by evaluating this function at various points within a specified window.

2. Smoothing and Blurring:

- Gaussian filters are effective for smoothing and blurring images or signals.

- The amount of smoothing is controlled by the standard deviation (o) of the Gaussian distribution. A larger o results in more smoothing.

3. Preservation of Edges:

- While Gaussian filters introduce a certain degree of blurring, they tend to preserve edges better than some other smoothing filters (e.g., mean filters).
- The extent to which edges are preserved depends on the standard deviation of the Gaussian distribution.

4. Frequency Domain Interpretation:

- In the frequency domain, the Gaussian filter attenuates higher frequencies more than lower frequencies.
- This behavior is due to the Gaussian distribution's rapidly decaying tails.

5. Separable Property:

- The Gaussian filter has a separable property, meaning that a 2D convolution with a Gaussian kernel can be decomposed into two 1D convolutions along the rows and columns.
- This property reduces the computational complexity of applying a Gaussian filter.

6. Applications:

- Gaussian filters are widely used in image processing for tasks such as noise reduction, blurring, and image smoothing.
- They are also used in computer vision applications, feature extraction, and edge detection.

7. Implementation:

- Gaussian filtering can be implemented efficiently using convolution in both the spatial and frequency domains.
- The convolution operation involves sliding the Gaussian kernel over the image or signal and computing the weighted sum at each position.

8. Trade-off with Smoothing and Detail Preservation:

- The choice of the standard deviation (σ) in the Gaussian filter involves a trade-off. A smaller o preserves more fine details but provides less smoothing, while a larger σ results in more smoothing but may blur finer details.

Gaussian filters are versatile and widely used in various applications due to their effectiveness in smoothing images while preserving important features. The balance between smoothing and detail preservation can be adjusted by tuning the standard deviation parameter.

Now that we have provided explanations regarding the requested requirements, we will display the noisy images :

Noisy images :

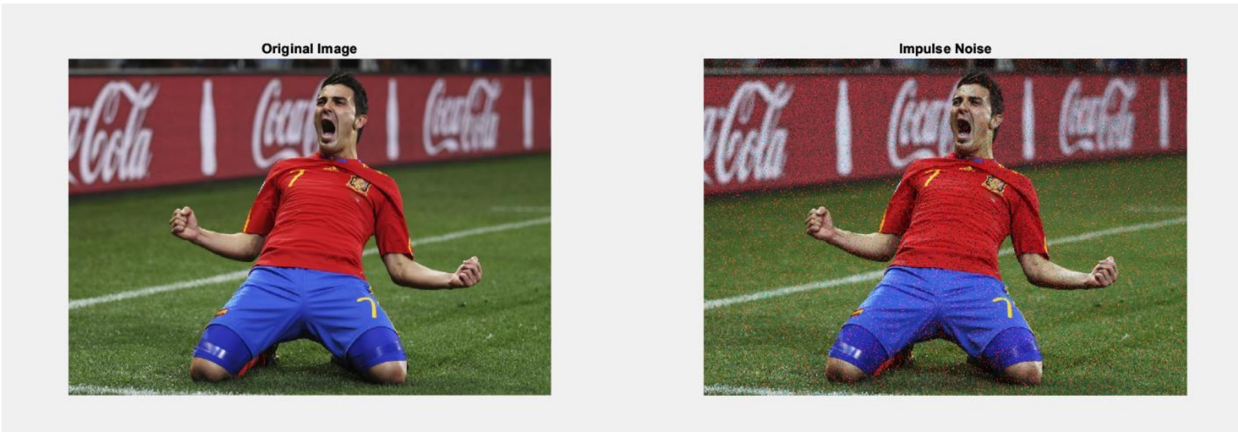


Image with Impulse noise

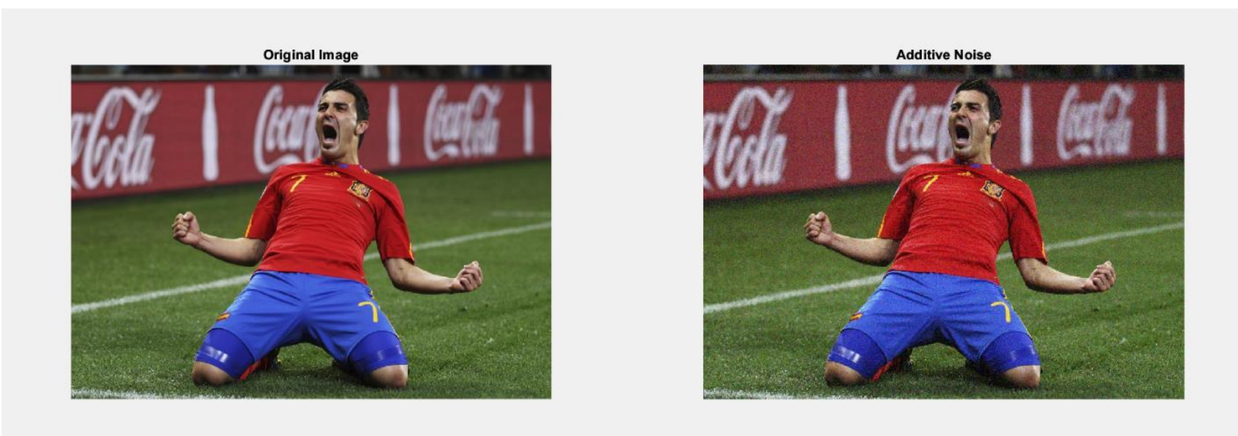


Image with Additive noise

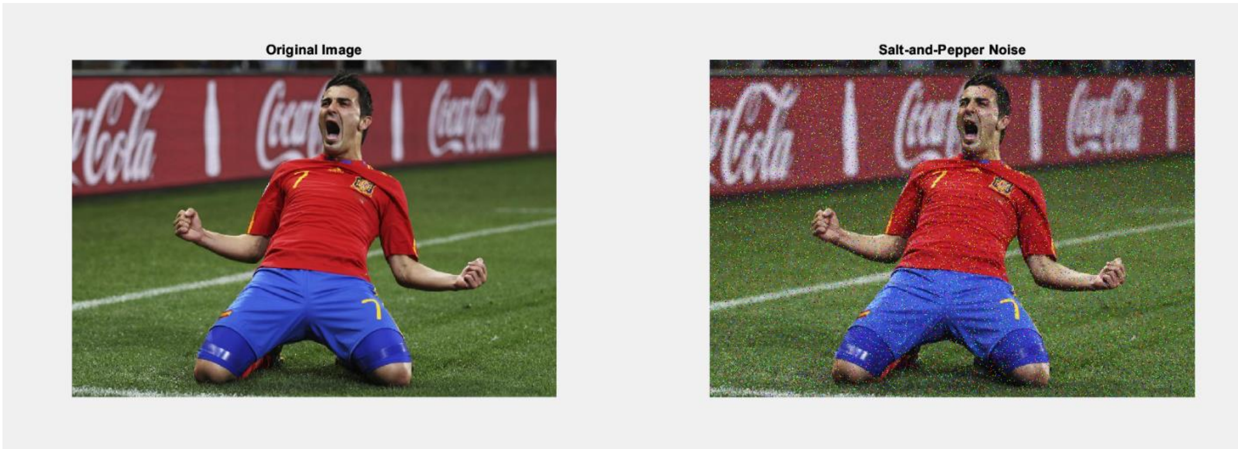


Image with Salt-and-Pepper noise

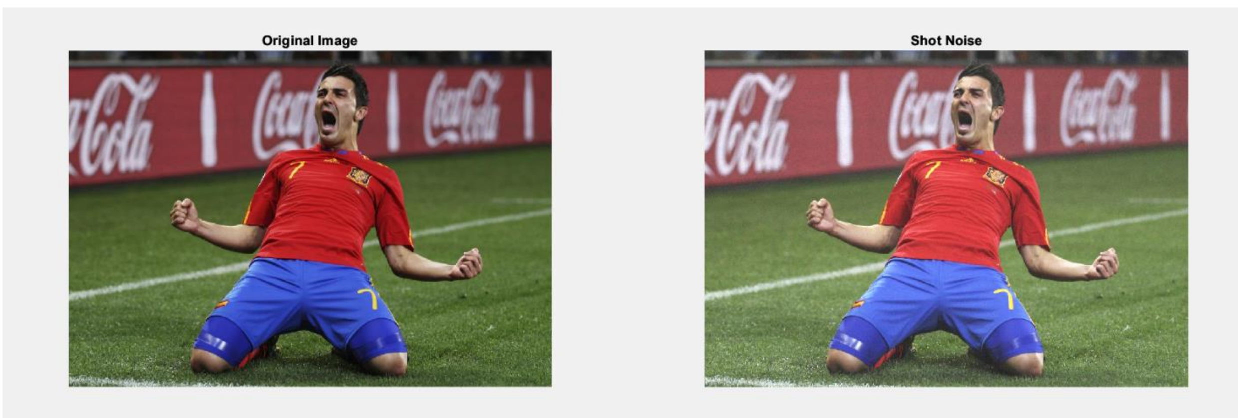


Image with Shot noise

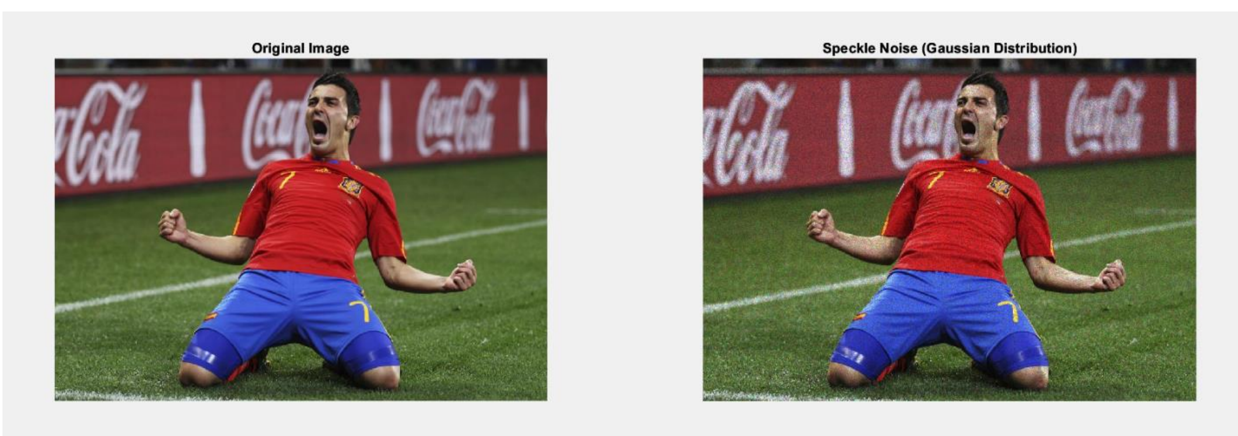


Image with Speckle noise(Gaussian Distribution)

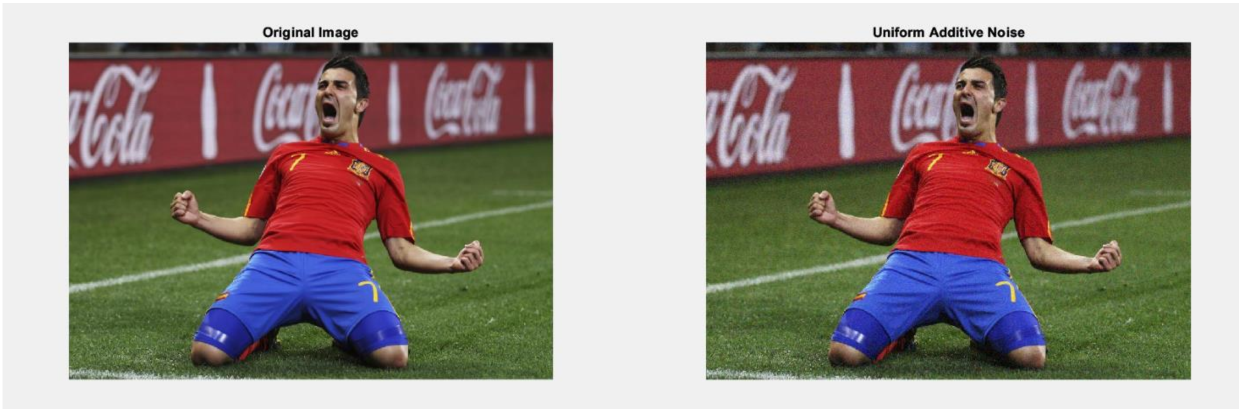


Image with Uniform Additive noise

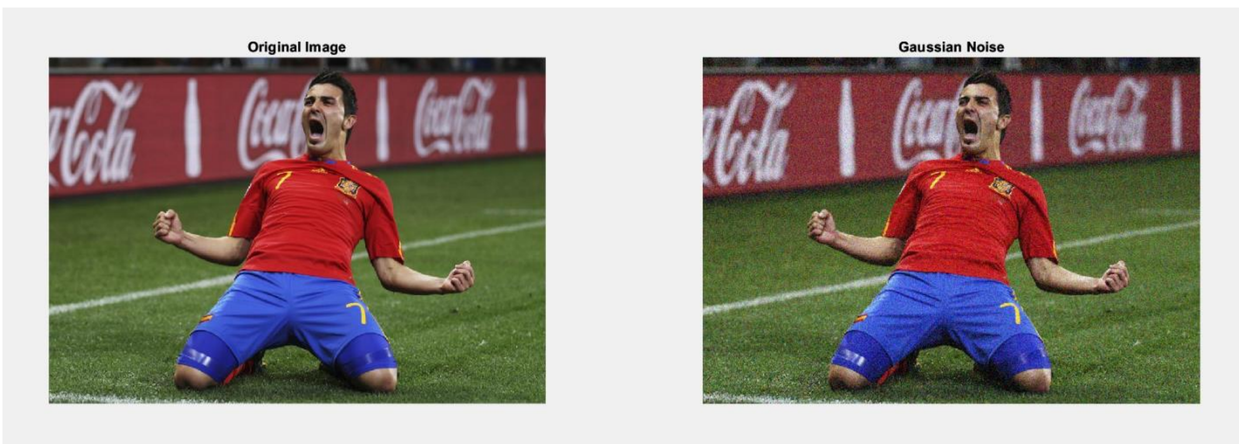


Image with Gaussian noise

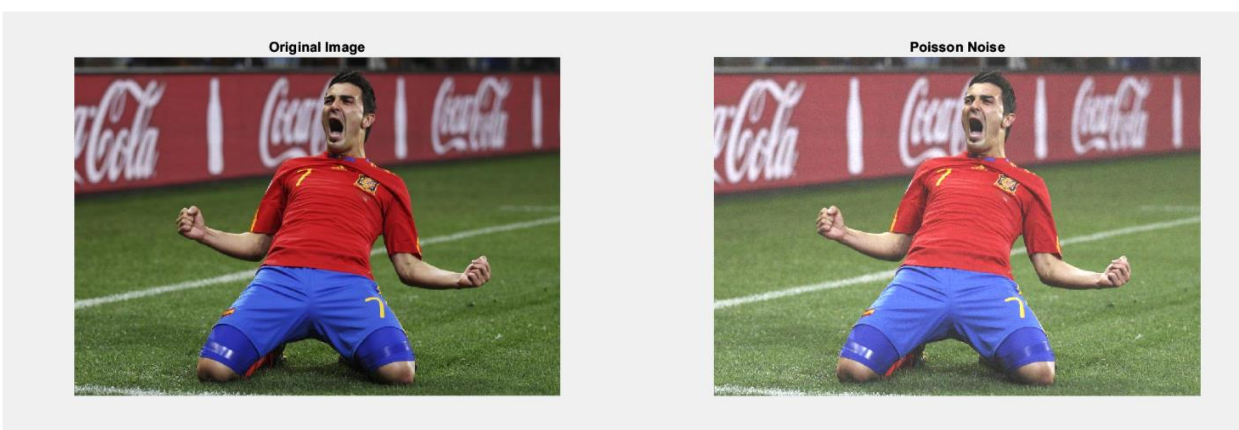


Image with Poisson noise

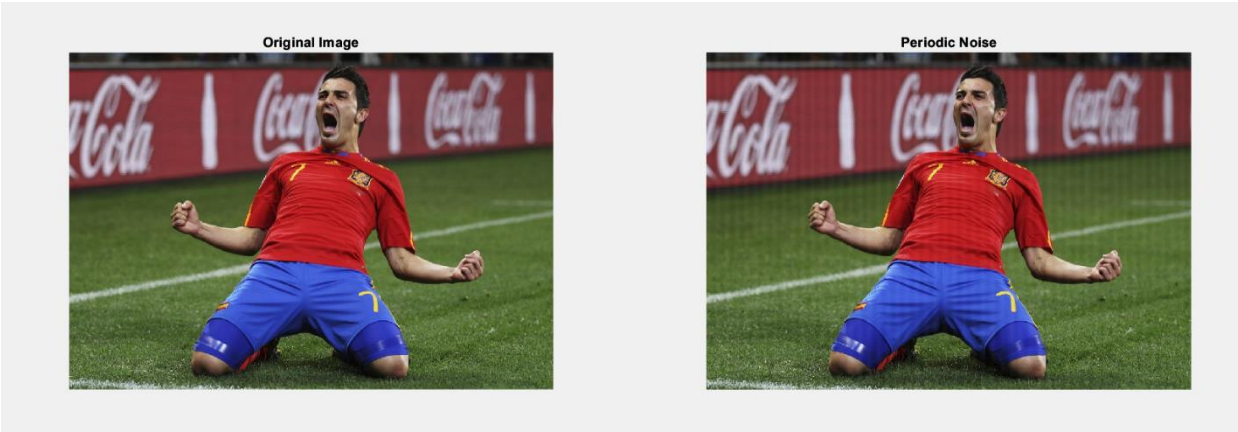


Image with Periodic noise

Explanation:

Considering the previously provided explanations, it is possible to recognize the nature of each of the noises and generate them. Then , the noises can be added to the original image.

For impulse noise, we first consider the noise intensity to be 0.1. Then, some pixels in the original image are randomly set to black or white (as specified in the code, half of the pixels are white and the other half are black).

For additive noise, we first consider the intensity to be 15. Then, based on that intensity, we generate noise using the rand function and add it to the original image to obtain the noisy image.

For salt and pepper noise, we consider the intensity to be 0.05. Additionally, we set the probability of salt and pepper to be 50%. Then, using the rand function, we randomly assign half of the pixels to white and the other half to black to create the noisy image.

For shot noise, we consider the intensity to be 30. Then, using the possrnd function of matlab, we generate noise and add it to the original image to obtain the noisy image. (It will be observed that Poisson noise is also constructed in this way since the nature of these two noises is the same.)

For speckle noise, we consider the intensity to be 0.15. After generating noise using the rand function, we multiply the noise with the original image to obtain the noisy image.

For uniform additive noise, we consider the intensity to be 20. Then, using the normal distribution, we generate noise and add it to the original image to obtain the noisy image.

For Gaussian noise, we consider the intensity to be 20. Then, using the rand function and the Gaussian distribution, we generate noise and add it to the original image to obtain the noisy image.

For Poisson noise, as mentioned earlier, we generate the noise in the same way as shot noise, with the only difference being that we consider the intensity to be 40 in this case.

For periodic noise, we generate a sinusoidal noise with an amplitude of 3 and a frequency of 0.05, then add it to the original image.

Now that the explanations regarding the noises and their generation methods have been provided, each of them will be filtered. The Filter functions are :

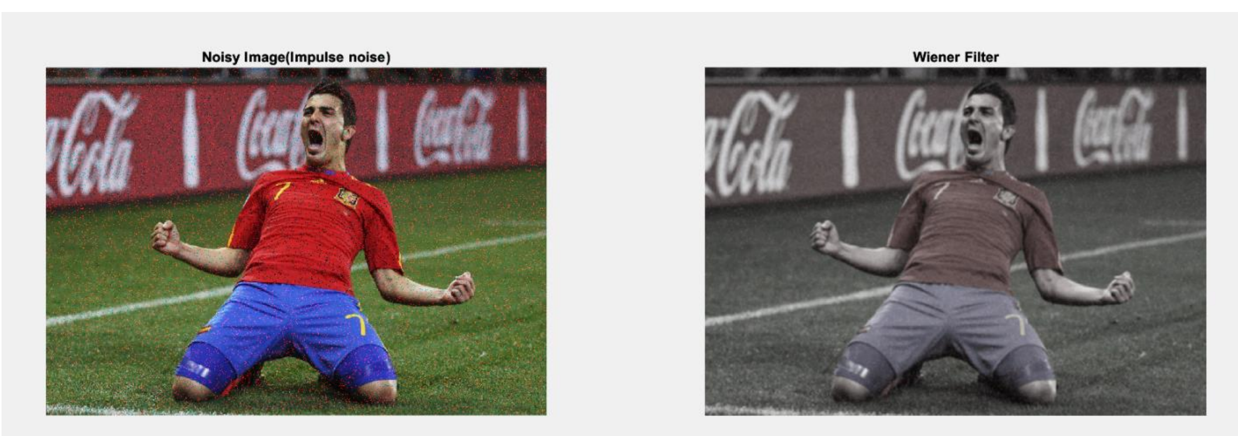
```
%% Filter Functions

function result = applyLinearFilter(image, kernel)
    [rows, cols, ~] = size(image);
    [kRows, kCols] = size(kernel);
    padRows = floor(kRows / 2);
    padCols = floor(kCols / 2);

    paddedImage = padarray(image, [padRows, padCols], 'replicate', 'both');

    result = zeros(size(image), 'like', image);

    for i = 1:rows
        for j = 1:cols
            window = double(paddedImage(i:i+kRows-1, j:j+kCols-1, :));
            result(i, j, :) = sum(sum(window .* kernel, 1), 2);
        end
    end
```

```
    result = uint8(result);
end

function result = applyWienerFilter(image, windowSize)
    [rows, cols, ~] = size(image);
    result = zeros(size(image), 'like', image);

    for i = 1:rows
        for j = 1:cols
            window = double(getNeighborhood(image, i, j, windowSize));
            localVariance = var(window(:));
            localMean = mean(window(:));
            noiseVariance = localVariance / 5;
            alpha = noiseVariance / (noiseVariance + localVariance);
            result(i, j, :) = localMean + alpha * (image(i, j, :) - localMean);
        end
    end

    result = uint8(result);
end

function result = applyMedianFilter(image, windowSize)
    [rows, cols, ~] = size(image);
    result = zeros(size(image));

    for i = 1:rows
        for j = 1:cols
            window = getNeighborhood(image, i, j, windowSize);
            result(i, j, :) = median(window(:));
        end
    end

    result = uint8(result);
end

function result = applyGaussianFilter(image, kernelSize, sigma)
    kernel = fspecial('gaussian', [kernelSize, kernelSize], sigma);
    result = applyLinearFilter(image, kernel);
end
```

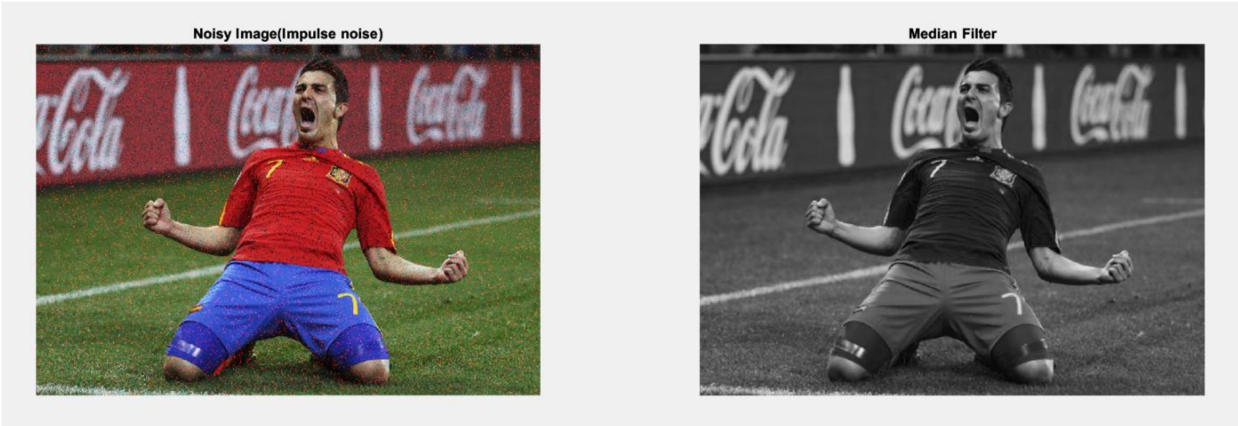
```
function window = getNeighborhood(image, i, j, windowSize)
    [rows, cols, ~] = size(image);
    halfSize = floor(windowSize / 2);

    iStart = max(i - halfSize, 1);
    iEnd = min(i + halfSize, rows);

    jStart = max(j - halfSize, 1);
    jEnd = min(j + halfSize, cols);

    window = image(iStart:iEnd, jStart:jEnd, :);
end
```

Now these are the Filtered images with 4 different filters :

Linear Smoothing Filter (Impulse noise)



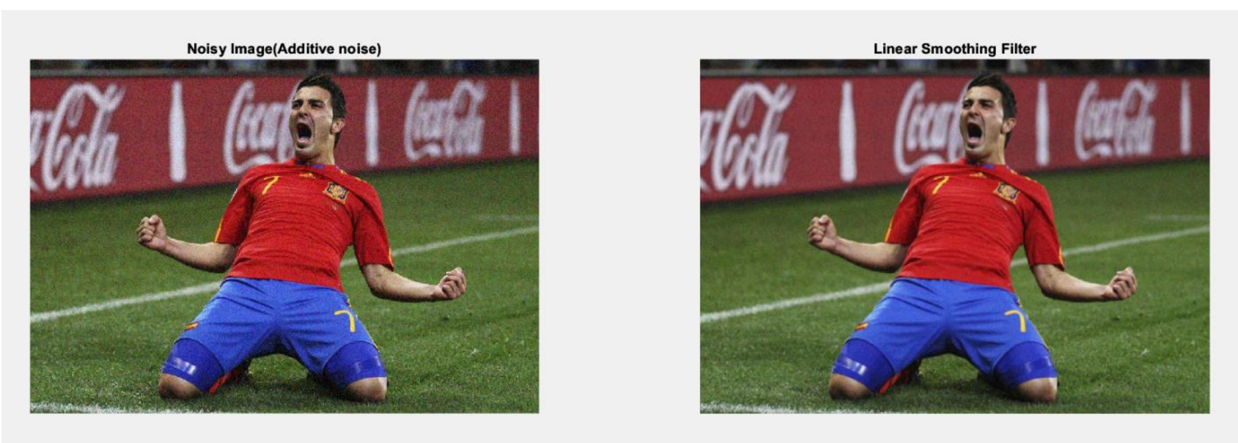
Wiener Filter(Impulse noise)



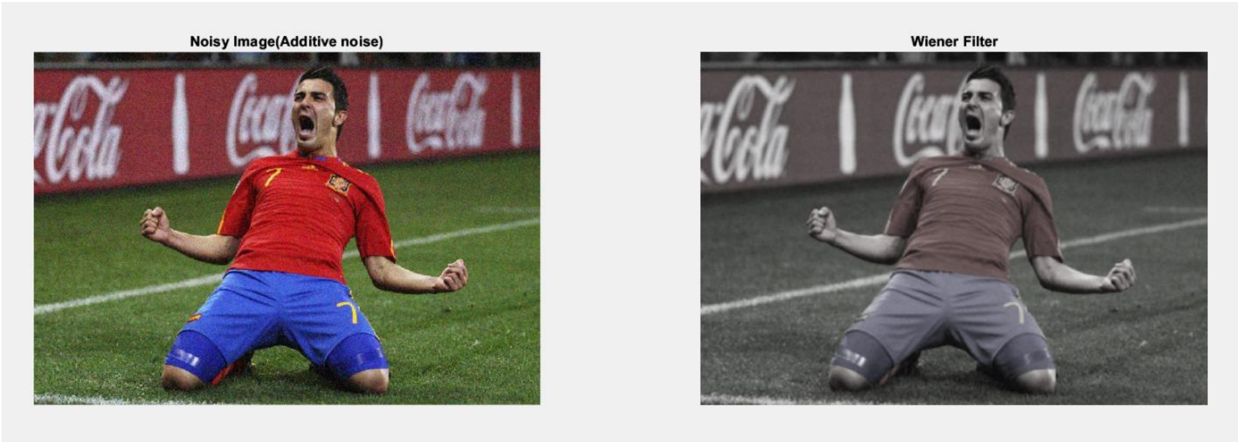
Median Filter(Impulse noise)

Gaussian Filter(Impulse noise)

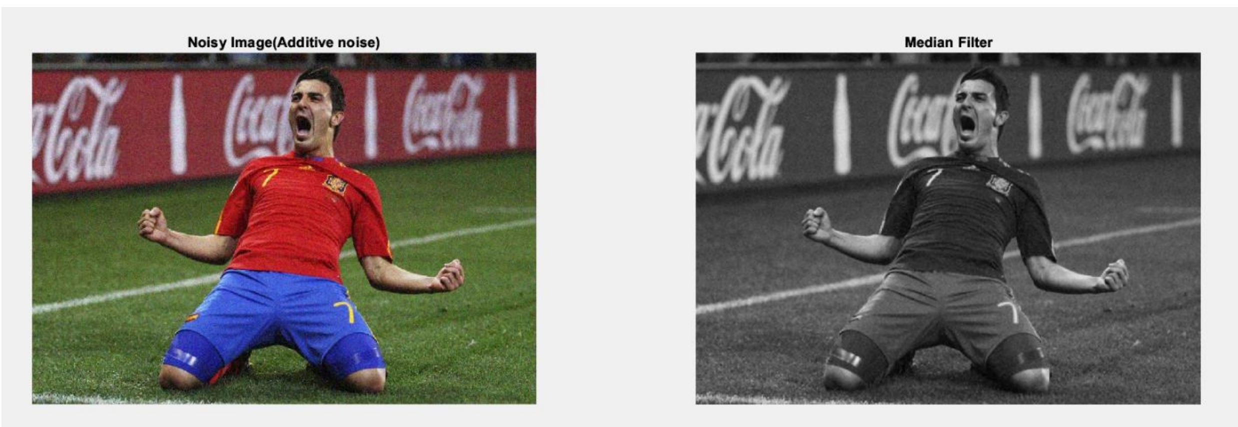
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



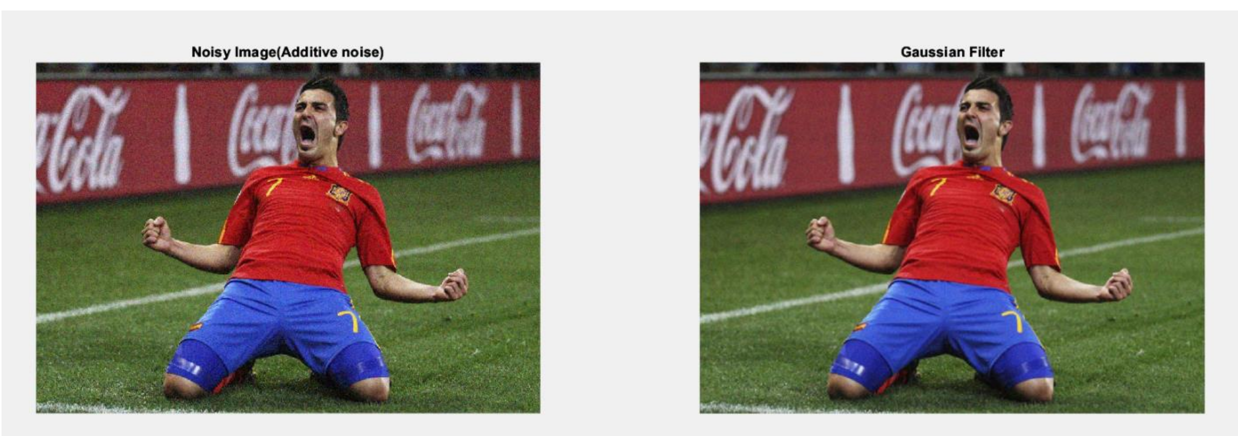
Linear Smoothing Filter(Additive noise)



Weiner Filter(Additive noise)

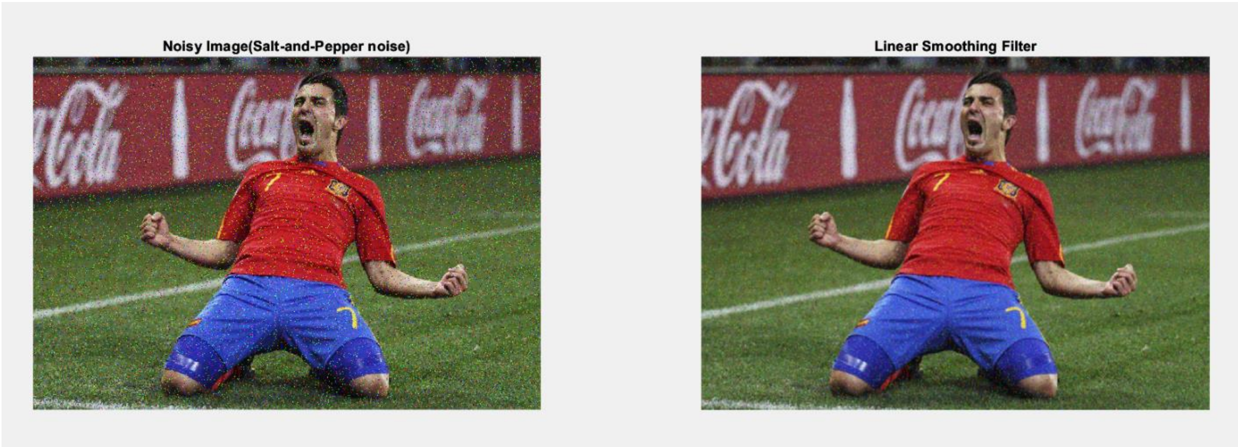


Median Filter(Additive noise)



Gaussian Filter(Additive noise)

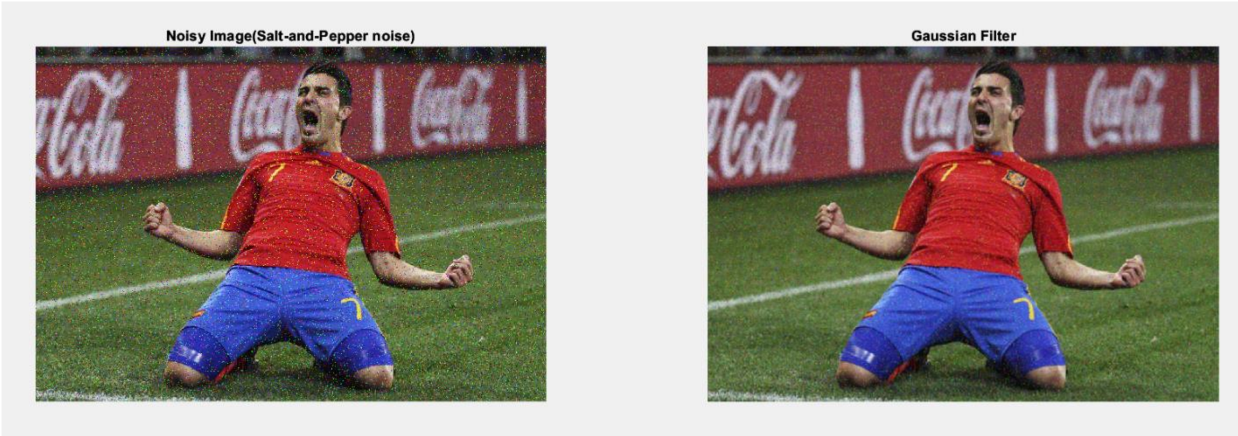
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



Linear Smoothing Filter(S&P noise)

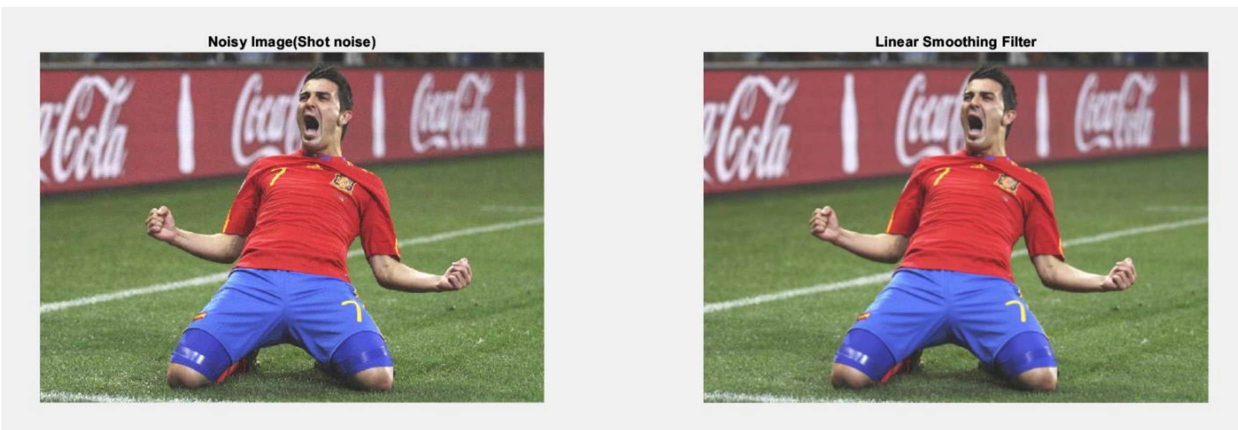
Weiner Filter(S&P noise)

Median Filter(S&P noise)

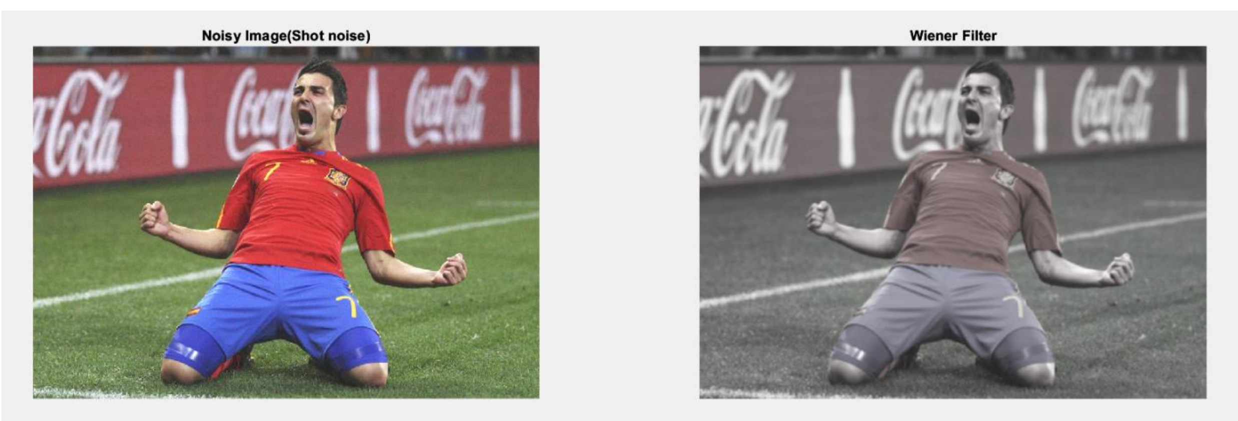


Gaussian Filter(S&P noise)

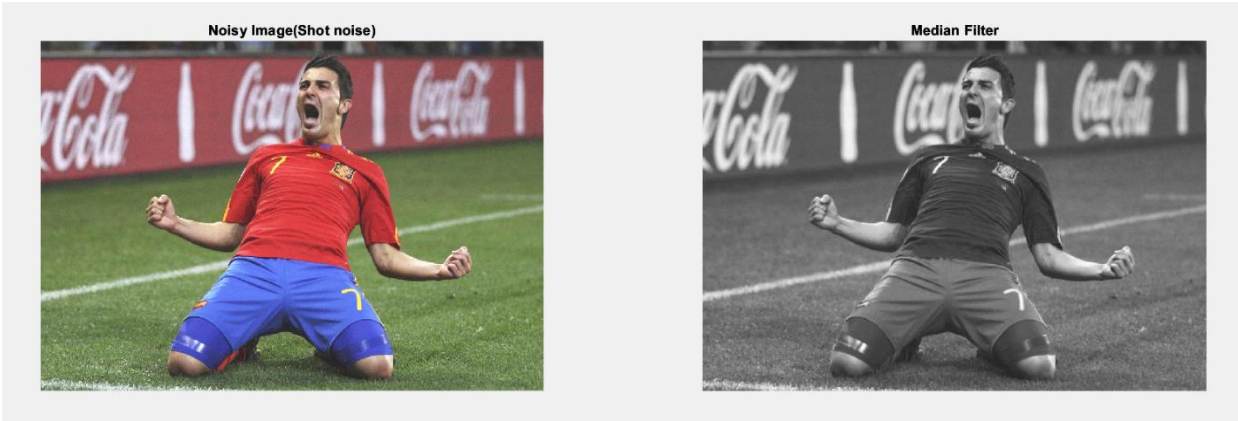
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



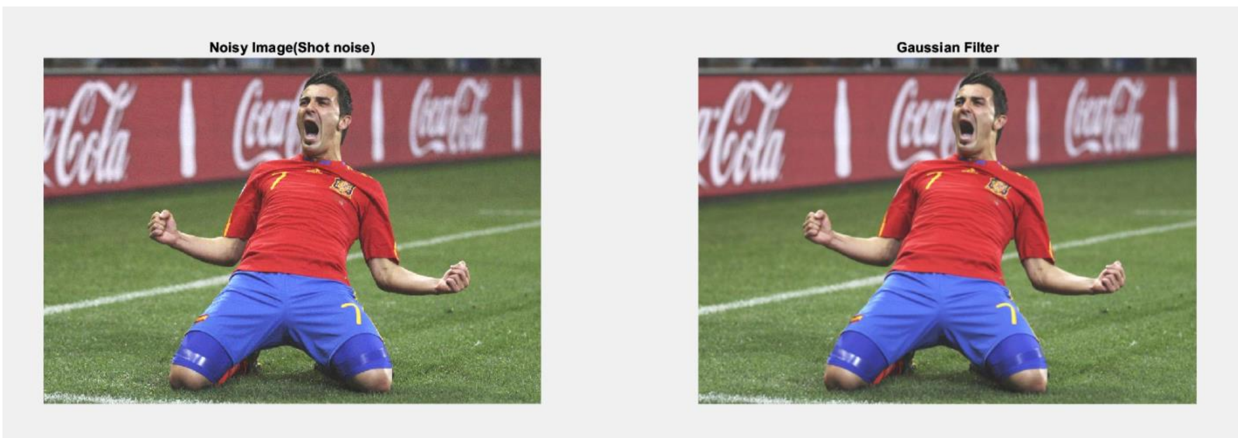
Linear Smoothing Filter(Shot noise)



Weiner Filter(Shot noise)

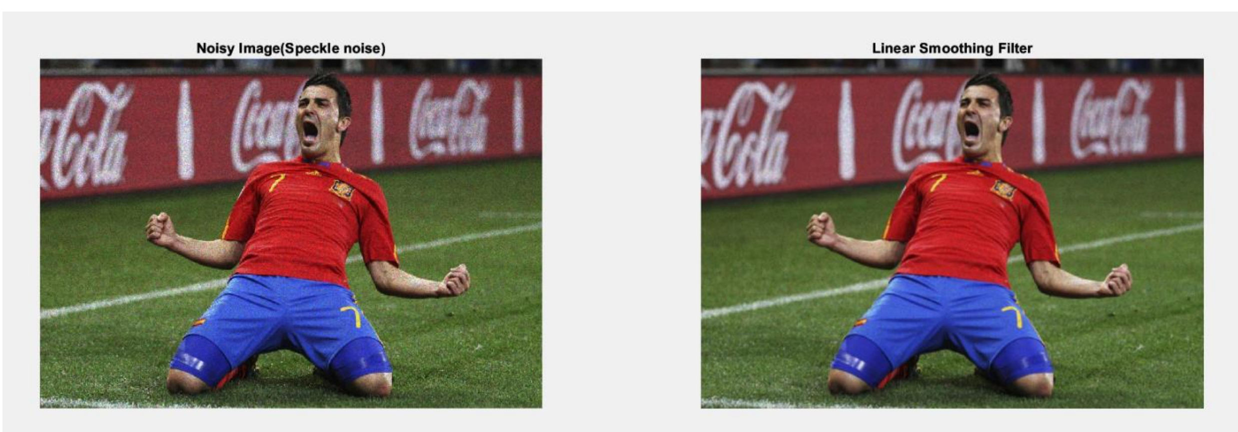


Median Filter(Shot noise)

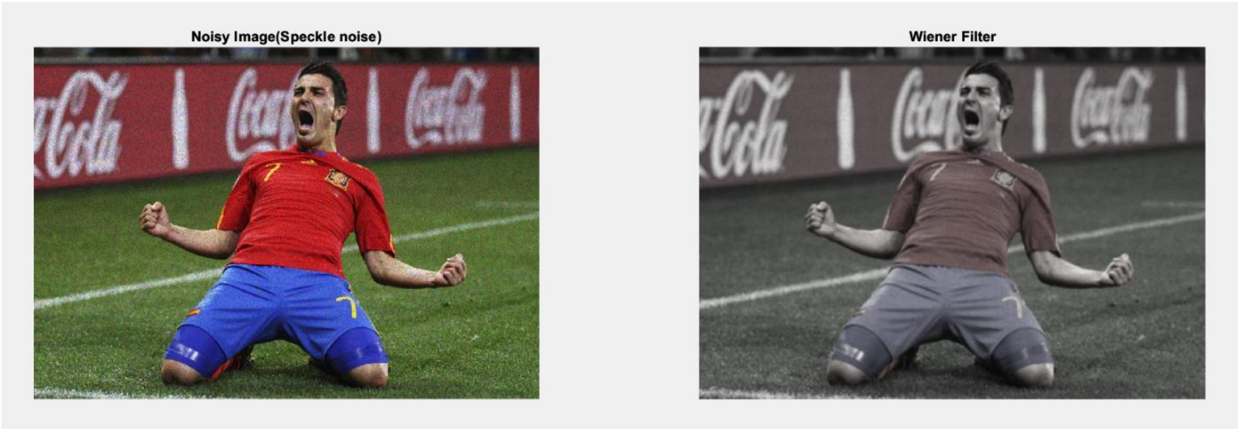


Gaussian Filter(Shot noise)

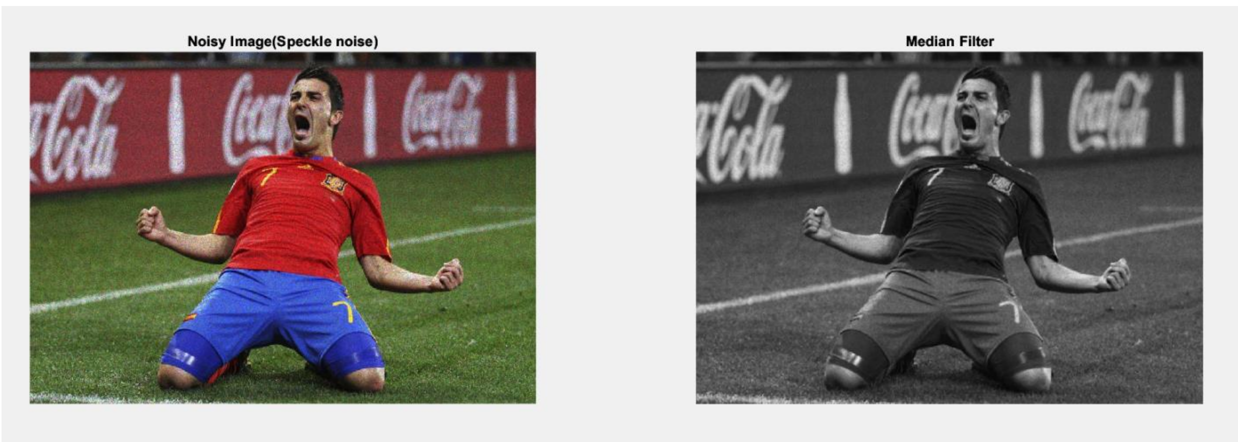
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



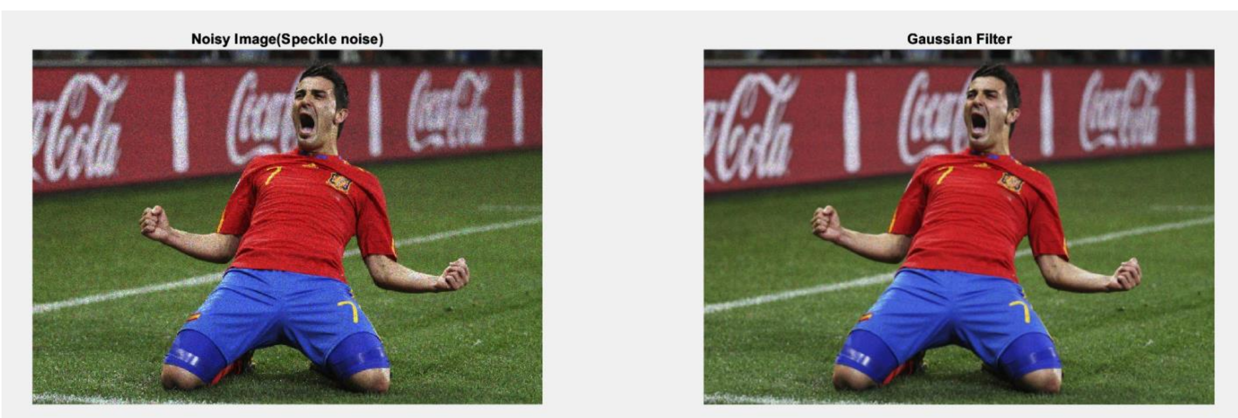
Linear Smoothing Filter(Speckle noise)



Weiner Filter(Speckle noise)

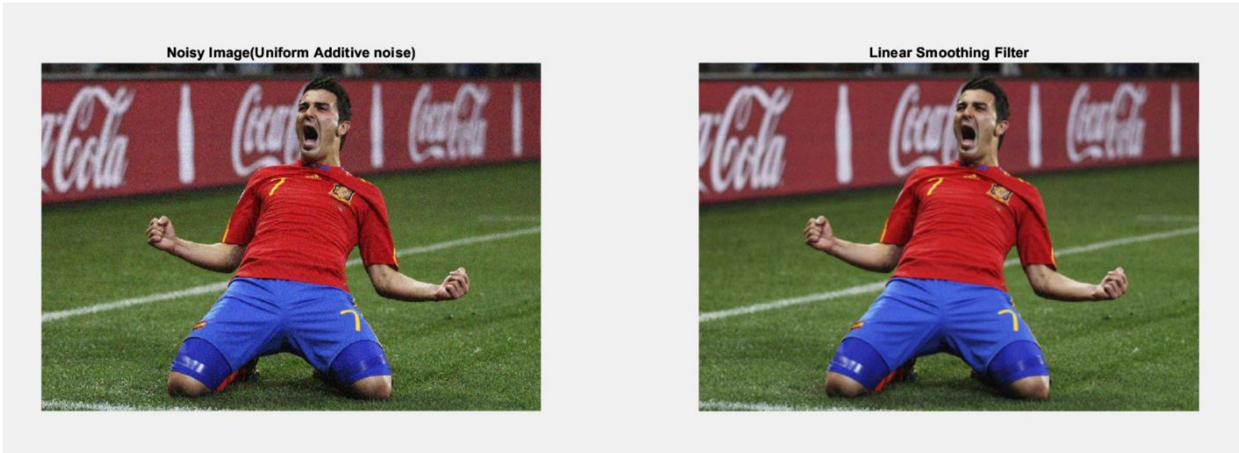


Median Filter(Speckle noise)

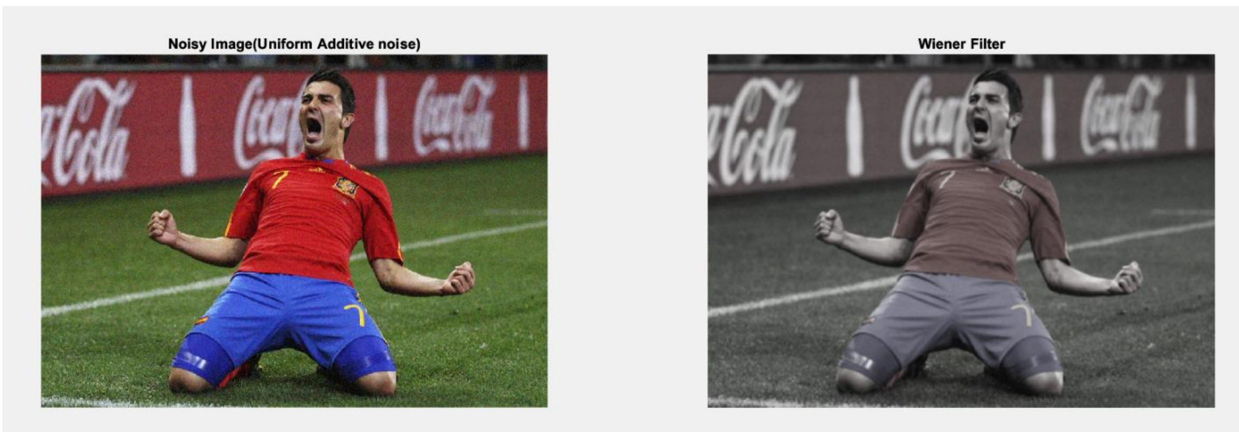


Gaussian Filter(Speckle noise)

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

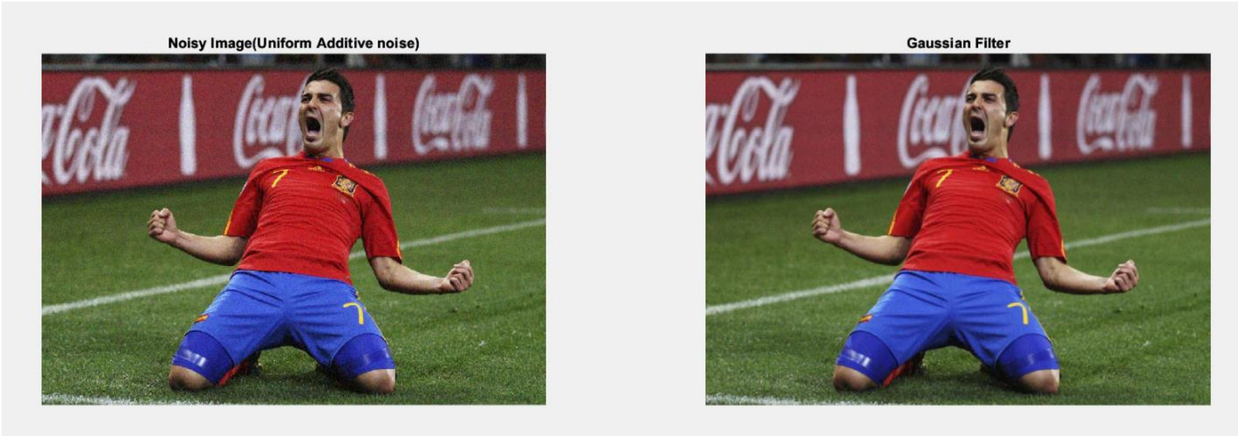


Linear Smoothing Filter(Uniform Additive noise)



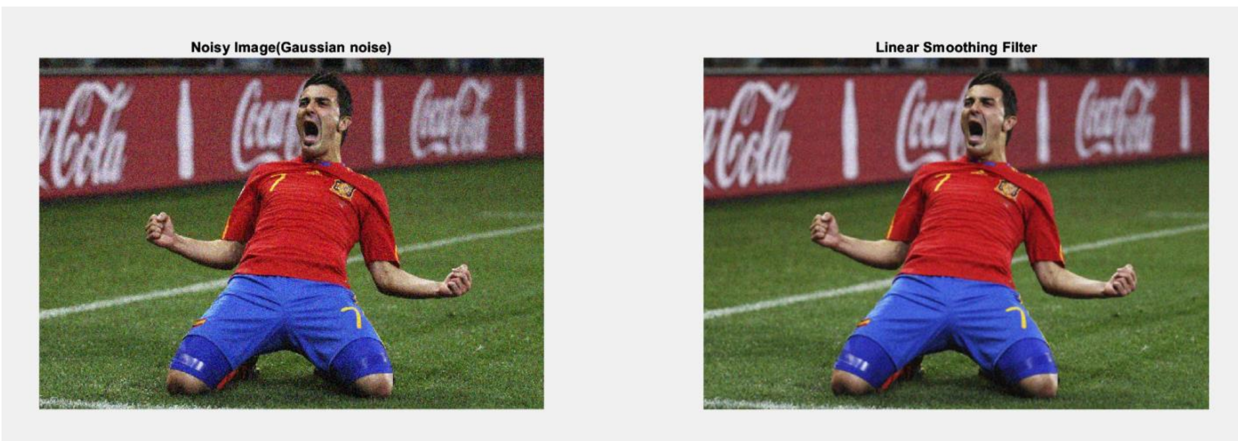
Weiner Filter(Uniform Additive noise)

Median Filter(Uniform Additive noise)

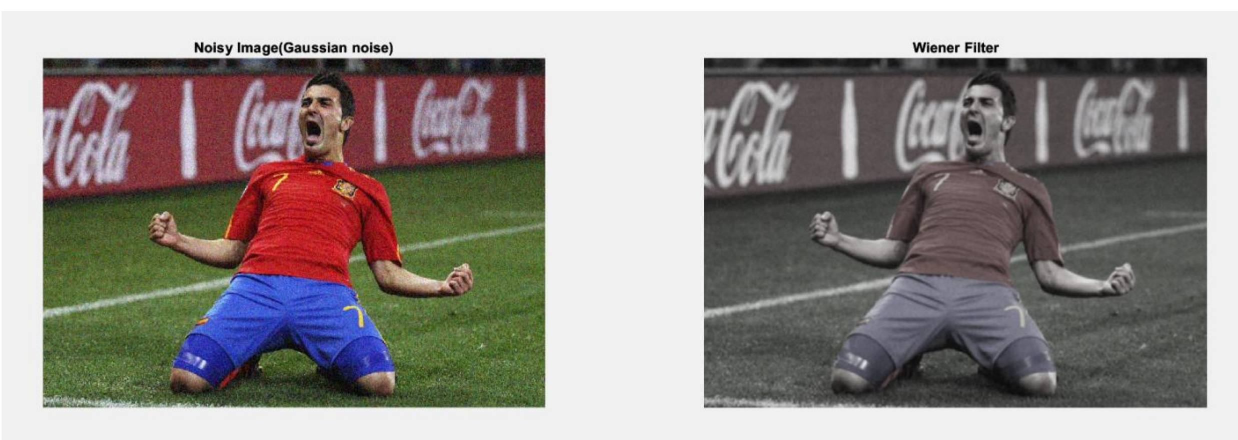


Gaussian Filter(Uniform Additive noise)

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

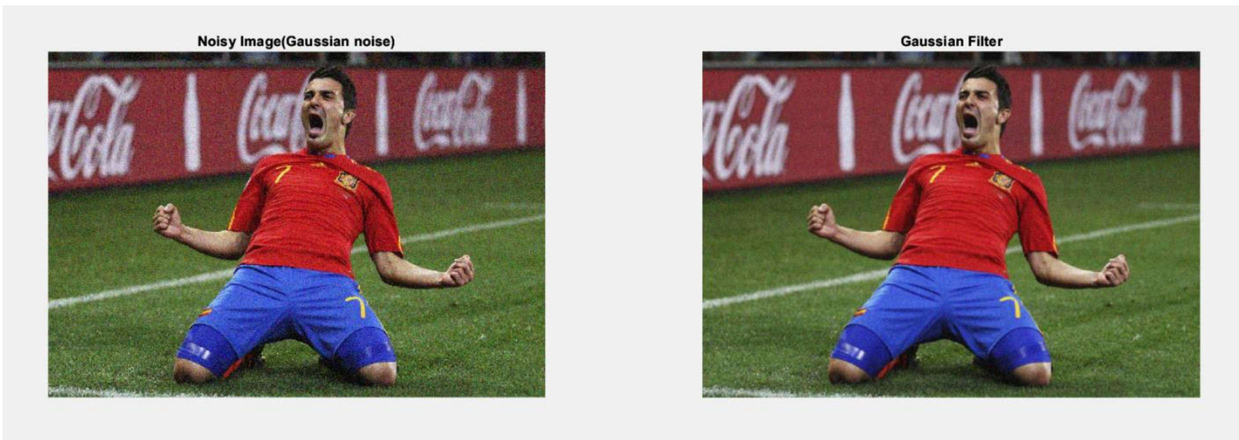


Linear Smoothing Filter(Gaussian noise)



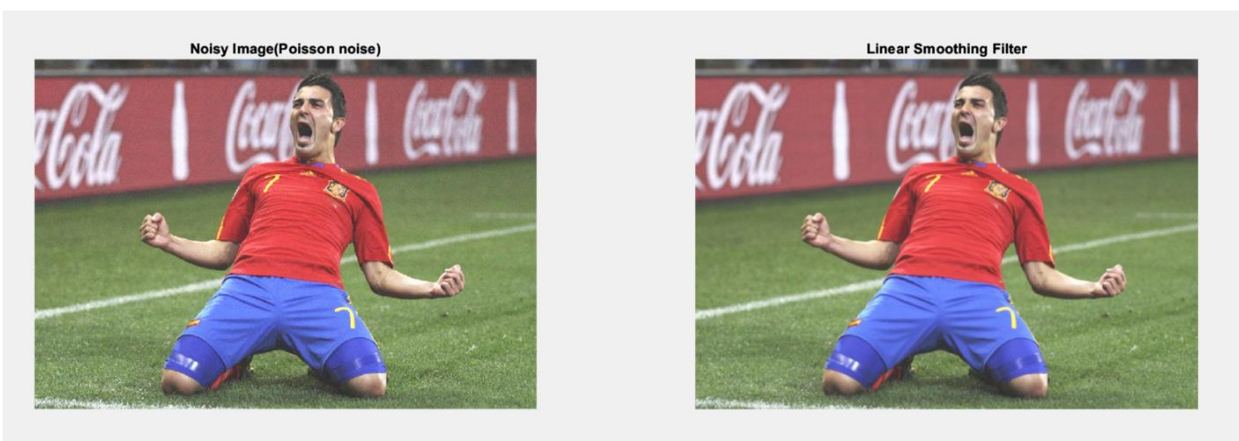
Weiner Filter(Gaussian noise)

Median Filter(Gaussian noise)

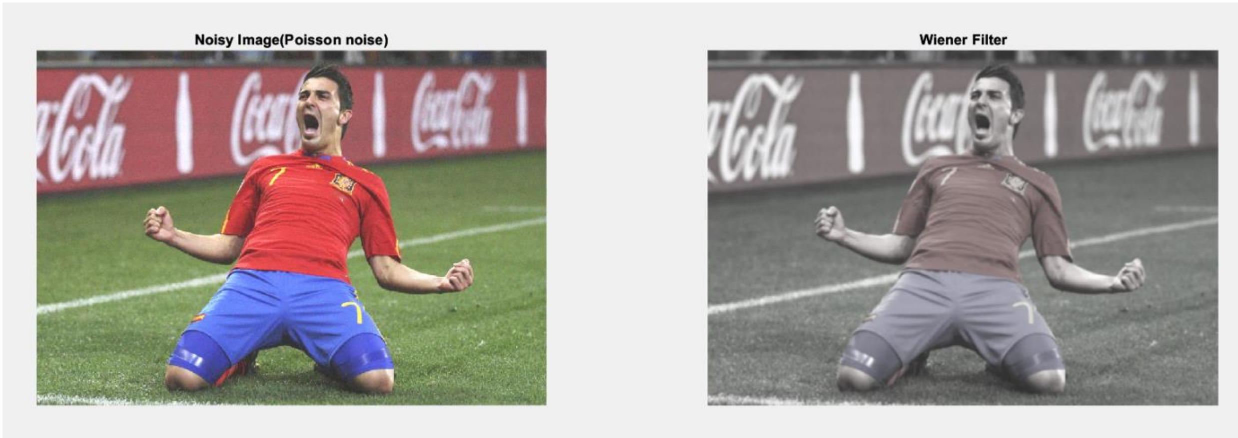


Gaussian Filter(Gaussian noise)

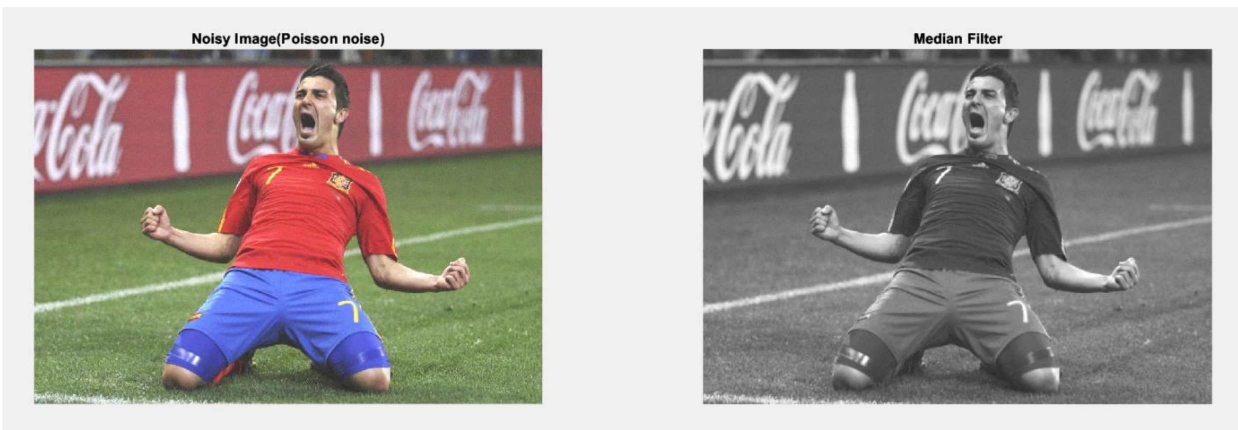
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



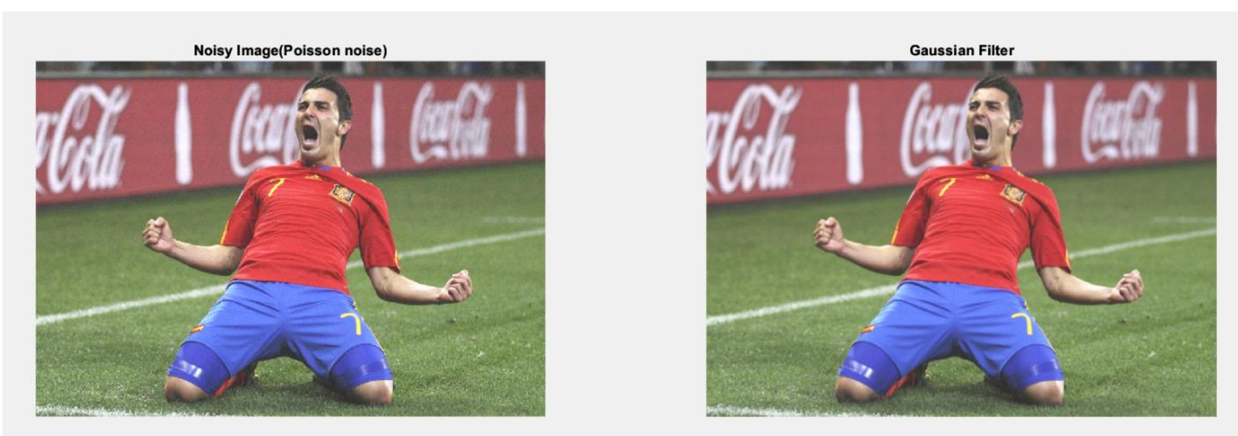
Linear Smoothing Filter(Poisson noise)



Weiner Filter(Poisson noise)

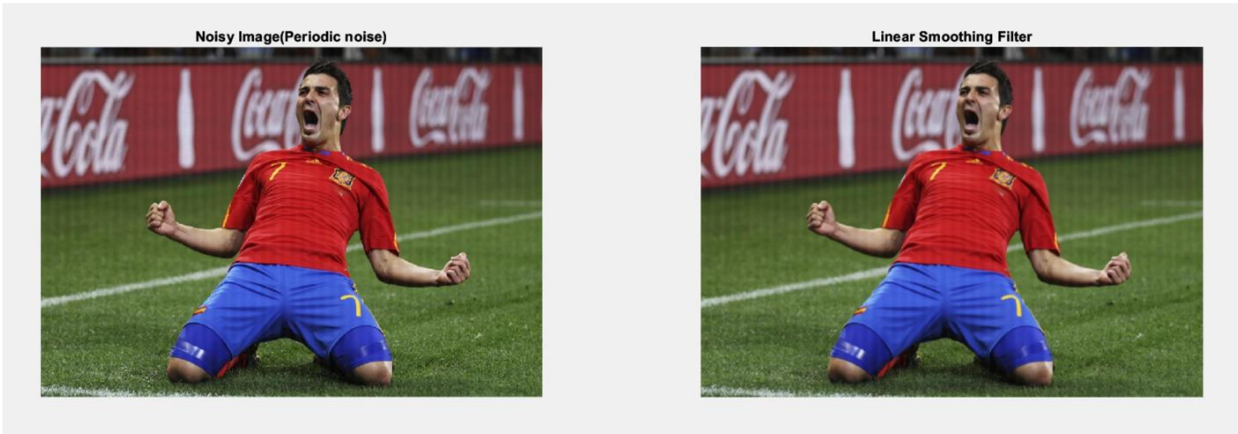


Median Filter(Poisson noise)

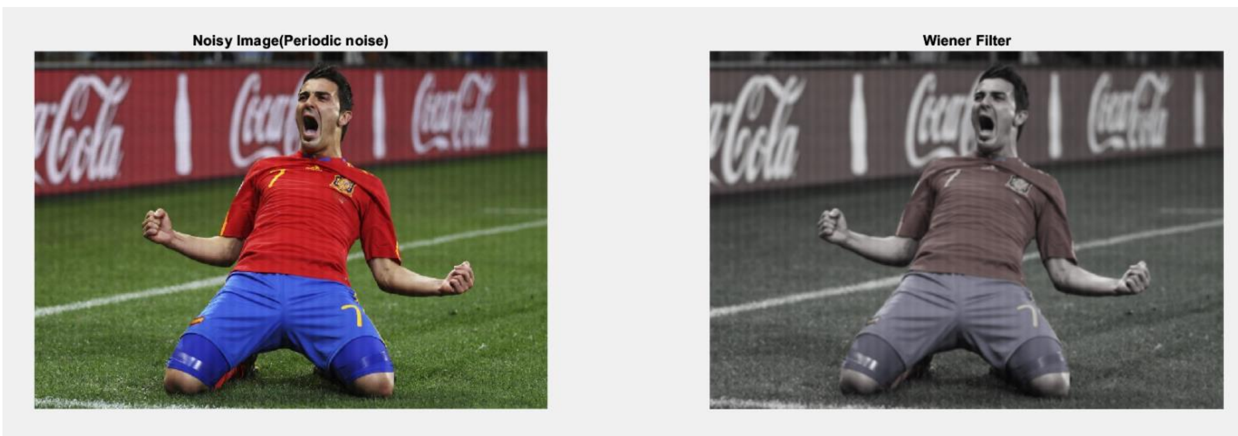


Gaussian Filter(Poissson noise)

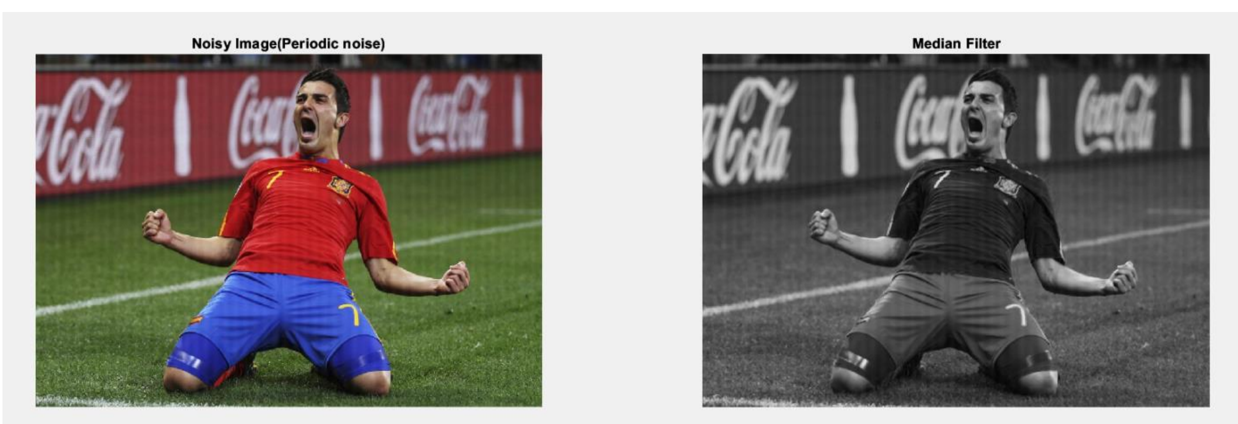
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&



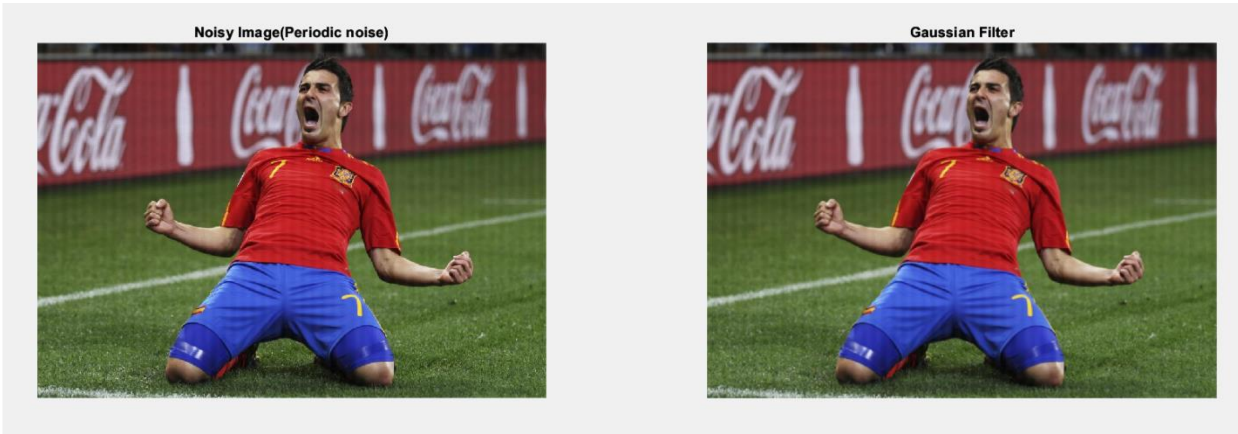
Linear Smoothing Filter(Periodic noise)



Weiner Filter(Periodic noise)



Median Filter(Periodic noise)



Gaussian Filter(Periodic noise)