**COMPUTER VISION AND ML BASED PS**

**1.**

**Gaussian Filter:**

The Gaussian filter smooth’s an image by reducing high-frequency noise and enhancing pixel continuity. It achieves this by blurring the image based on the spatial arrangement of pixels. Commonly employed for image blurring and noise reduction, particularly in preprocessing stages prior to advanced analyses like edge detection and image segmentation

**Sobel Filter:**

The sobel is the gradient based filter used for the edge detection. It calculates the gradient the magnitude at each pixel in the image, highlighting regions of high intensity change. Sobel filtering is widely used in edge detection tasks to identify edges, contours, and boundaries in images. It's a fundamental step in many computer vision algorithms.

**Laplacian Filter:**

The Laplacian filter utilizes the second derivative to accentuate changes in image intensity. Unlike the Sobel filter, it doesn't discern orientation. Primarily employed for image sharpening and edge detection, the Laplacian filter highlights areas of substantial intensity variation. This enables the identification of fine details and enhances image features by emphasizing regions of significant intensity change.

**Median Filter**:

The median filter replaces each pixel value with the median value of its neighbourhood, effectively reducing salt-and-pepper noise while preserving edges and fine details. It is commonly used for noise reduction in images. The median filter is robust to extreme pixel values and does not blur edges as much as Gaussian filtering.

**Bilateral Filter**:

The Bilateral filter, akin to the Gaussian filter, is employed to smooth images. However, it goes beyond solely considering the spatial arrangement of pixels; it also accounts for pixel brightness. This dual consideration allows it to preserve both spatial proximity and intensity similarity. Widely utilized for noise reduction, the bilateral filter retains crucial image features like edges and textures.

**2.**

Adaptive filters adjust their parameters based on local variations in images, enabling them to adapt to the varying characteristics of different regions within an image. Unlike fixed filters, which apply the same filter kernel across the entire image, adaptive filters dynamically modify their kernel or filter coefficients based on the local content of the image. This adaptation allows them to better capture the complex and varying structures present in natural images.

**Advantages:**

Adaptive filters possess the capability to adjust to the local statistics of an image, thereby enhancing their ability to preserve crucial image details and structures. This adaptability plays a pivotal role in effectively denoising images while safeguarding important features. By dynamically adjusting parameters based on local characteristics, adaptive filters demonstrate greater robustness compared to fixed kernel filters. Notably, their ability to preserve edges and significant image features is a distinctive property, ensuring more accurate image processing results. Utilizing adaptive filters consistently yields images of higher fidelity and precision compared to those processed using fixed kernel filters.

**3.**

In Convolutional Neural Networks (CNNs), filter kernels play a crucial role in feature extraction. A filter kernel is a small matrix of weights that slides across the input image or feature map. As it slides, it performs convolution operations, extracting features by computing dot products between the kernel weights and the corresponding input values.

* **Filter kernel**: Filter kernels act as feature detectors, capturing patterns and structures present in the input data. Each filter kernel specializes in detecting specific patterns or features, such as edges, textures, shapes, or more complex structures. By applying multiple filter kernels across the input data, CNNs can extract a diverse set of features at different spatial locations and scales.
* **Kernel Size:** Larger kernel sizes capture more global features and spatial relationships, while smaller kernel sizes focus on local details and fine structures. Increasing the kernel size can lead to more complex feature extraction but may also increase computational cost and risk over fitting, especially in deeper layers.
* **Kernel Weights:** The weights of the kernel determine the specific patterns or features that the kernel detects. During training, the network learns optimal weights for each kernel through backpropagation and gradient descent, adjusting them to minimize the loss function.

**4.**

Filter design plays a crucial role in real-time computer vision applications due to its impact on computational efficiency, accuracy, and adaptability to varying environmental conditions

**Computational Efficiency**:

In real-time applications, computational efficiency is crucial to ensure prompt processing of video streams or live camera feeds. Efficient filters are designed to minimize resource usage, including CPU and memory. This design enables real-time processing even on resource-constrained devices like embedded systems, mobile devices, or edge computing platforms.

**Accuracy**:

Accuracy is important factor because it directly influence the quality and reliability of computer vision task such image recognition, segmentation and classification. Accurate filter design is essential for robust performance across diverse scenarios, lighting conditions, and environmental factors.

**Adaptability to Varying Environmental Conditions**:

The real environment is constantly changing, with variations in lighting, weather, and object appearance. To ensure consistent performance and reliability, filters need to adapt to these dynamic conditions. Adaptive filters are well-suited for this task, as they employ techniques such as dynamic parameter tuning, learning-based approaches, and context-aware processing. By adapting to changing conditions, adaptive filters can maintain reliability and provide consistent performance.

**6.**

[**https://www.kaggle.com/code/pohrselvan/training-vs-inference/edit**](https://www.kaggle.com/code/pohrselvan/training-vs-inference/edit)

**7.**

<https://www.kaggle.com/code/pohrselvan/traffic-signal/edit>