Aryan Shirke 21BCS111

1. Generic filter with User-Defined Coefficients:
- Header: cv::filter2D
- Prototype: void filter2D(InputArray src, OutputArray dst, int ddepth, InputArray kernel, Point anchor=Point(-1,-1), double delta=0, int borderType=BORDER_DEFAULT)
- Usage: This function applies a custom linear filter to an image using the provided kernel. It allows you to define your own convolution kernel for operations like blurring, sharpening, edge detection, etc.

2. Sobel filter:
- Header: cv::Sobel
- Prototype: void Sobel(InputArray src, OutputArray dst, int ddepth, int dx, int dy, int ksize=3, double scale=1, double delta=0, int borderType=BORDER_DEFAULT)
- Usage: Sobel filter is used for edge detection by computing the gradient magnitude of an image. It takes the derivatives of the image in the x and y directions.

3. LoG (Laplacian of Gaussian):
- Header: N/A (implemented using Gaussian filter followed by Laplacian filter)
- Usage: LoG is used for detecting edges and blobs in

an image. First, the image is blurred using a Gaussian filter to reduce noise, then the Laplacian operator is applied for edge detection.

4. Gaussian Filter:
- Header: cv::GaussianBlur
- Prototype: void GaussianBlur(InputArray src, OutputArray dst, Size ksize, double sigmaX, double sigmaY=0, int borderType=BORDER_DEFAULT)
- Usage: Gaussian filter is used for noise reduction and blurring. It applies a Gaussian blur to the input image with a specified kernel size and standard deviation.

5. Median Filter:
- Header: cv::medianBlur
- Prototype: void medianBlur(InputArray src, OutputArray dst, int ksize)
- Usage: Median filter is used for removing salt-and-pepper noise from images. It replaces each pixel's value with the median value of the neighboring pixels within a specified kernel size.

6. Prewitt Filter:
- Header: N/A (implemented using cv::filter2D

with predefined kernels)
- Usage: Prewitt filter is used for edge detection. It calculates the gradient magnitude of an image in both x and y directions using predefined kernels.

7. Sharpening Filters:
- Unsharp Masking: Implemented using a combination of blurring and addition.
- High Boost Filtering: Implemented using a combination of unsharp masking and original image.

8. Plotting Image Histogram:
- Header: N/A (usually implemented using external libraries like Matplotlib)
- Usage: Histogram of an image can be plotted using histogram plotting functions in libraries like Matplotlib. OpenCV provides functions for calculating histograms.

9. Calculating Image Histogram:
- Header: cv::calcHist
- Prototype: void calcHist(const Mat* images, int nimages, const int* channels, InputArray mask, OutputArray hist, int dims, const int* histSize, const float** ranges, bool uniform=true, bool accumulate=false)

- Usage: calcHist calculates the histogram of an image. It is useful for image analysis and understanding the distribution of pixel intensities.

10. Histogram Equalization:
- Header: cv::equalizeHist
- Prototype: void equalizeHist(InputArray src, OutputArray dst)
- Usage: Histogram equalization is used to enhance the contrast of an image by redistributing the intensity values. It improves the global contrast of the image.

11. Histogram Matching:
- Header: N/A (usually implemented using custom algorithms)
- Usage: Histogram matching is used to transform the intensity distribution of an image to match a specified histogram. It helps in matching the appearance of two images.

12. Edge Detection:
- Header: N/A (implemented using functions like Sobel, Canny, etc.)
- Usage: Edge detection algorithms like Sobel, Canny, etc., are used to detect edges in images. These edges

represent significant changes in intensity and are crucial for image analysis tasks.

13. Finding Derivatives along X and Y Axis:
- Header: N/A (implemented using functions like Sobel, Scharr, etc.)
- Usage: Derivatives along the x and y axes can be found using functions like Sobel, Scharr, etc. These derivatives are useful for gradient-based image processing operations.

14. Other Functions Related to Image Enhancement:
- OpenCV provides various other functions for image enhancement, including contrast adjustment, gamma correction, color
space transformations (e.g., RGB to grayscale), resizing, cropping, and more. These functions are crucial for various image processing tasks like object detection, recognition, segmentation, etc.