Introduction

Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision. Edge detection allows users to observe the features of an image for a significant change in the Gray level. The software tool is used MATLAB.

Approaches

Image edge detection using inbuilt operators are of two types which is gradient and gaussian. Gradient based operator which computes first order derivations in a image like Sobel, Prewitt and Robert operator. Gaussian based operator which computes second order derivations in a image like canny edge detector. First read the image and converts it to grayscale, and then applies various edge detection algorithms to it. Here I am used Canny,Log,Sobel,Prewitt,Robert and Zerocross alogrithms.Each subplot in the figure shows the result of applying a specific edge detection algorithm to the grayscale image. This allows to compare the effectiveness and characteristics of different edge detection techniques.

Thresholding is a fundamental technique in image processing and computer vision, commonly used in various applications including edge detection, object segmentation, and feature extraction. Sets a threshold value (in my case,0.5) for edge detection. This value might need adjustment based on the specific characteristics of the image and the desired edge detection results and also added my signature with it.

Support Vector Machine based edge segmentation combines edge detection with machine learning to identify object boundaries. By training an SVM classifier on labeled edge pixels extracted from images, it learns to distinguish between edge and non-edge regions. The SVM's decision boundary separates these regions in the feature space, effectively segmenting edges from the background. Feature vectors representing edge pixels typically include gradient magnitude and orientation information. SVM-based edge segmentation can yield accurate and reliable results, contributing to various computer vision applications.

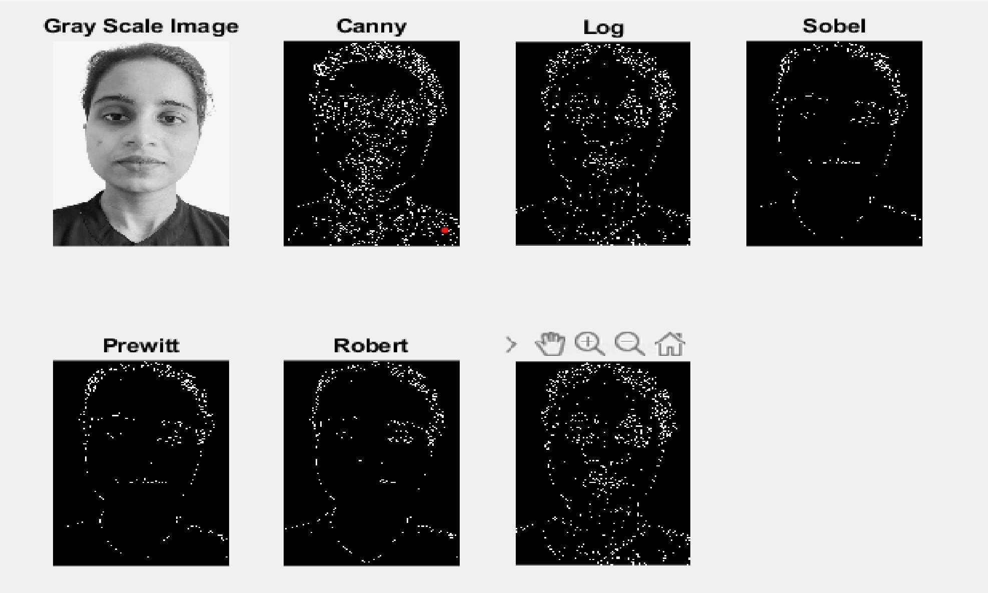
Segmenting the face using K-means cluster is an unsupervised learning algorithm widely used for image segmentation tasks. K-means clustering operates on the color space of the image, such as RGB or HSV. By treating each pixel's color values as feature vectors, K-means iteratively assigns pixels to clusters represented by centroids, aiming to minimize the within-cluster sum of squares. Once convergence is achieved, pixels within the same cluster are considered part of the same facial region. It is a visualization of the original image along with its segmentation based on k means clustering, where different regions with similar characteristics are assigned different colours.

Segmenting the face using Deep learning Techniques involves leveraging the power of convolutional neural networks (CNNs) to automatically learn and extract features for accurate facial segmentation. The network learns to map input facial images to pixel-wise segmentation masks. Here I am used DeepLabv3+ architecture. It is a state-of-the-art semantic segmentation architecture. It uses a deep neural network with a ResNet backbone to predict pixel-wise class labels. In the output image we have seen an effective approach to partitioning images into meaningful regions based on pixel similarities, facilitating various visual analytics problems.

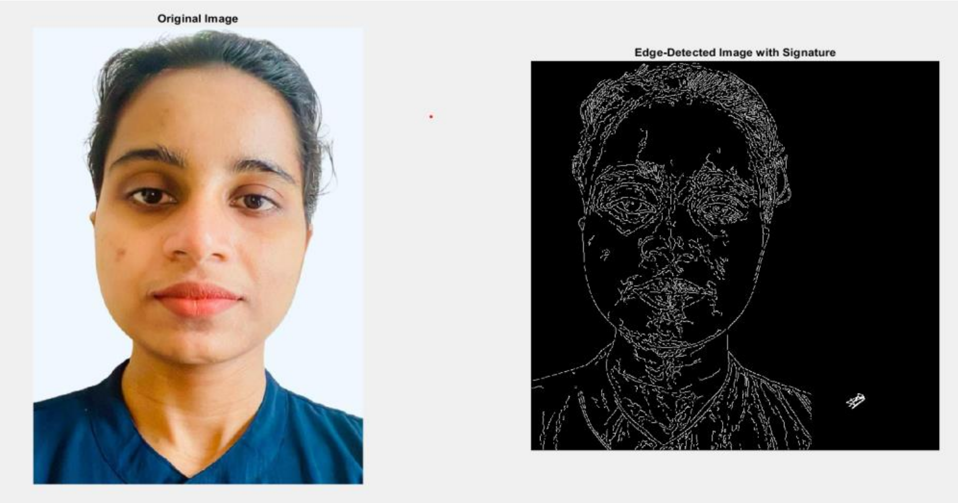
Here I implement a simple edge detection algorithm based on gradient magnitude calculation. The function iterates over each pixel of the input image, excluding the last row and column, to calculate the edge strength using a gradient-based approach. Specifically, it computes the absolute differences between pixel intensities in neighbouring locations along both the horizontal and vertical directions. These differences are then summed to obtain the edge strength at each pixel. while this edge detection approach is simple and easy to implement, it may not capture all edge details accurately and may produce noisy results, especially in complex images

Experimental Results

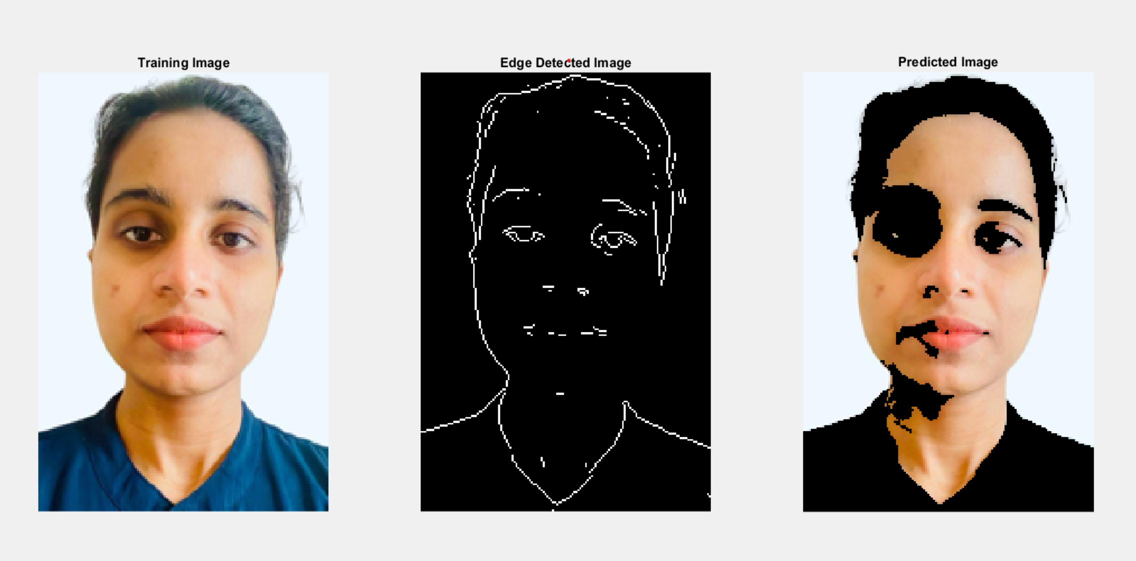
- Edge detection using inbuilt operators



- Thresholding



- Segmentation using SVM



- Segmentation using K-means

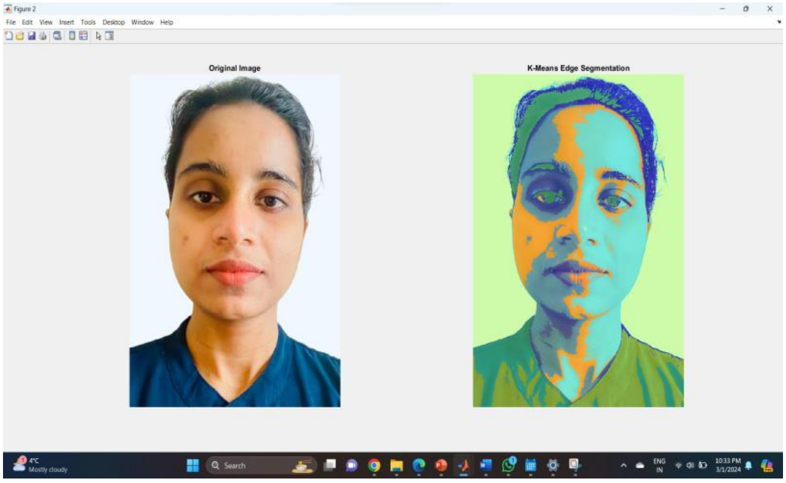
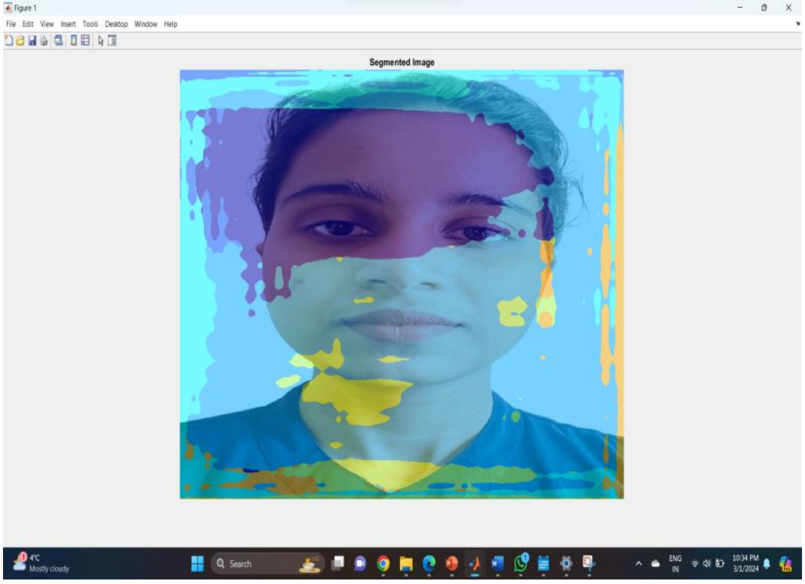


Figure 2
File Edit View Insert Tools Desktop Window Help
Original Image
K-Means Edge Segmentation
4°C
Mostly cloudy
Search
ENG
IN
10:33 PM
3/1/2024

- Segmenting the face using Deep learning Technique



- My Method

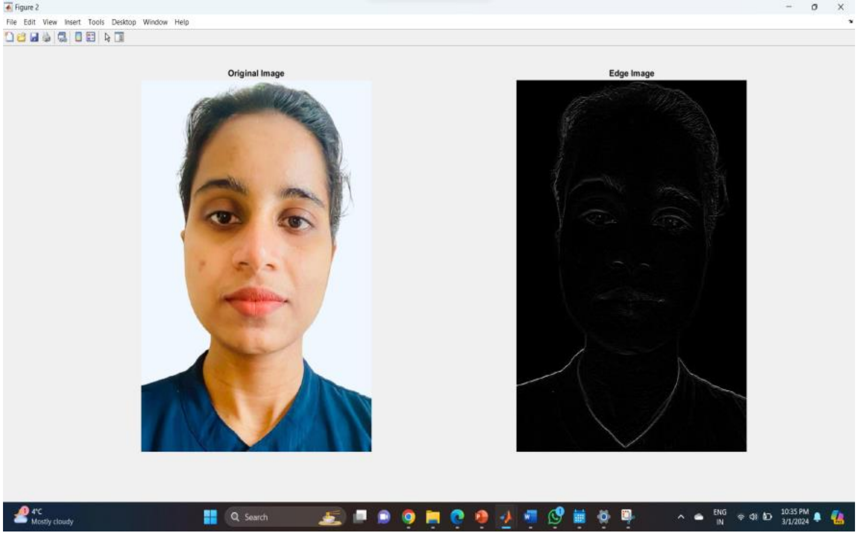


Figure 2
File Edit View Insert Tools Desktop Window Help
Original Image
Edge Image
4°C
Mostly cloudy
Search
ENG
IN
10:35 PM
3/1/2024