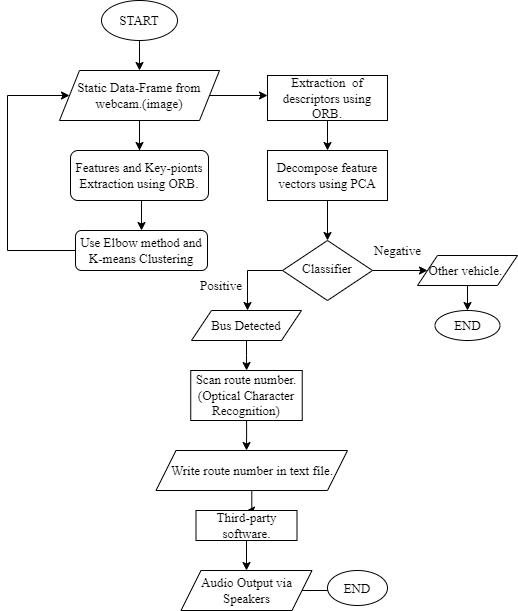
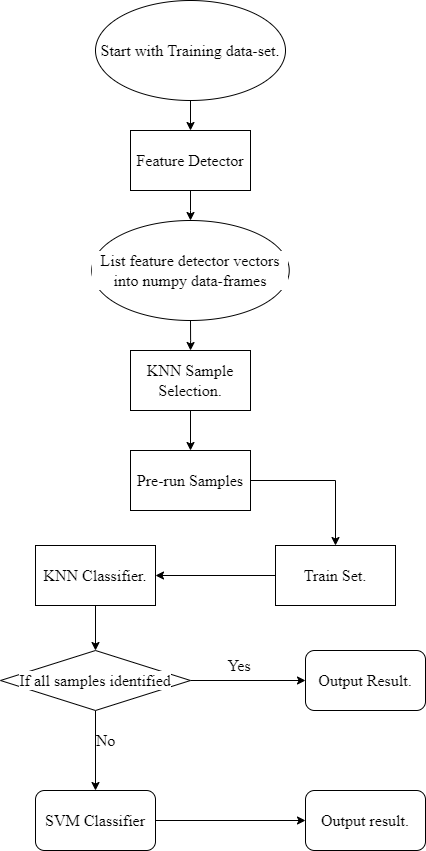
**Real-time object recognition and classification of objects through static data-frames is implemented in this work. This is a promising sector to implement innovative ideas related to computer vision applications. Visually impaired individuals undergo through difficult situations and complications faced while recognizing visual information as well as processing that information in a state of chaos, like a crowded area such as bus stop. Things get even harder for them when there is a need to step up and examine whether the bus or any transport they want to board into, mainly public transports, are actually routing from their boarding destination towards their target destination.**

**The chaotic situation makes it almost impossible for any blind person to summarize that information and board onto that transport medium correctly. There are hardly any aids for the visually impaired people that will actually help them classify between buses specifically. The idea of personal help can be held under consideration but that is not always liable and viable. This paper describes a novel approach for recognizing buses and providing the corresponding routes. The bus images are captured in real-time, then the token number is extracted and the user is informed about the route information.**

The project proposes the following flow for implementation of this idea :



Following is the flowchart for the back-end process which has been implemented :



After the classification of buses we implemented EasyOCR for reading the token numbers any other alternate Optical Character Recognition algorithm can be used here but it is advised to go for EasyOCR as it is easier to implement and gives almost 100% accuracy in character recognition.

The algorithm for the easy OCR is as follows :

Input: Bus image with route number pasted on it.

Output: Recognised text that is number from the image

*Stages of Initialization:*

1: **Start**

2: Convert RGB image to Grey image

3: Apply bilateral filter for noise reduction

4: Apply canny filter for edge detection

5: Finding contours

6: Apply Masking

7: Extracting text from the masked image using easy OCR

8: Running if-else statement to find bus route

9: **End**