2.1 IntroductionTraditional cameras are not flexible and also unaware of their environment. They cannot  
provide constant and standard image quality in every environment because of their fixed  
(non-programmable) internal architecture. Along with image quality, image pre-processing  
is also crucial. The images need to be pre-processed before they can be used to generate  
useful results. Therefore, ample amount of work done only to improve image quality and  
making pre-processing faster.  
The authors of [1] use Field Programmable Gate Array (FPGA) architecture for making  
pre-processing faster. As the FPGA architecture has the ability to perform parallel  
processing, it will shorten the processing time and the efficiency will increase.

2.2 Edge ComputingThe pre-processing generally occurs at server side. Server solution is simple and effective  
but it is costly and time consuming. Edge computing eliminates the need to send image data  
to server-side for processing. Provides a way to process the images directly on the camera  
by taking advantage of sensors and the use of end devices to take over the load of processing  
[3].  
In [4] the detection and recognition tasks for surveillance are executed locally by edge  
devices. Only when devices are not able to execute the recognition task, a recognition  
request is sent to the server.  
  
2.3 Opencv

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

OpenCv is a very powerful tool used to analyze images and video files. Thresholding as a technique of image processing was chosen for the implementation of tracking in video streams. The choice to script using OpenCv – Python was because Python on its own does not support video processing. There is so far no video processing library in Python. OpenCv thus provided the necessary platform to achieve image processing. The following flowchart was used for this implementation

2.4 Summary

Face Recognition is a technology in computer vision. In Face recognition / detection we locate and visualize the human faces in any digital image.

It is a subdomain of Object Detection, where we try to observe the instance of semantic objects. These objects are of particular class such as animals, cars, humans, etc. Face Detection technology has importance in many fields like marketing and security

Cascade Classifiers and Haar Features:

Cascade Classifiers and Haar Features are the methods used for Object Detection.

It is a machine learning algorithm where we train a cascade function with tons of images. These images are in two categories: positive images containing the target object and negative images not containing the target object.

There are different types of cascade classifiers according to different target objects. In our project, we will use a classifier that considers the human face to recognize it as the target object.

Haar Feature selection technique has a target to extract human face features. Haar features are like convolution kernels. These features are different permutations of black and white rectangles. In each feature calculation, we find the sum of pixels under white and black rectangles.

### Haar-cascade Detection in OpenCV:

OpenCV provides pre-trained models on Haar features and [Cascade classifiers](https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html). These models are located in OpenCV installation.

Image processing applications are widely used nowadays which need to analyse and process  
a large number of images and video streams in real-time. This real-time constraint can be  
handled if the live camera feed is pre-processed right at the edge device in our case a Raspberry Pi device . The raspberry pi board coupled with an image sensor is used to develop a camera that is adaptive and can pre-process the incoming live camera stream by adjusting it parameters, i.e (brightness, saturation, contrast).