EXP NO: 6
DATE: 16.02.2024

VIDEO SURVEILLANCE USING COMPUTER VISION IN MATLAB

AIM:

To perform video surveillance using computer vision in Matlab.

SOFTWARE REQUIRED:

- MATLAB R2019A
- INSTALL usbwebcams.mlpkginstall

THEORY AND ALGORITHM OF VIDEO SURVEILLANCE:

Webcam

To acquire live video frames or images, we will be using the webcam function. Suppose you have multiple cameras connected to your system. You can specify the number or the name of the camera you want to use for the task.
The algorithm requires two consecutive frames implementing the snapshot function on the webcam object. We acquire these frames as an image.

Image Processing

Once we have the frames in hand, the next step is to find the absolute difference between the consecutive frames. For this, we will be making use of the imabsdiff function in MATLAB. The image obtained is still in RGB form, but it is better to carry out image transformations in Grayscale. It is simpler and more convenient. To transform a color image to a grayscale image, we make use of the rgb2gray function.

The next step is to use the Gaussian blurring filter of size 5x5 to reduce image noise and reduce details, and for this, we use the imgaussfilt function.
After image enhancement, we move on to Computer Vision Toolbox objects and morphological processing to satisfy our requirements. We will be using two computer vision toolbox objects; one is the vision.ForegroundDetector and the other one is the vision.BlobAnalysis.

The ForegroundDetector segment moving objects in a video frame from the background by comparing to a background model whether individual pixels are part of the background or the foreground. It outputs a binary mask, where the pixel value 1 corresponds to the foreground, and the value 0 corresponds to the background. By using background subtraction, you can detect foreground objects in an image taken from a stationary camera.

Three morphological functions will be implemented to remove the noise and fill in the holes:

- imopen
- imclose
- imfill

Connected groups of foreground pixels from morphological processing likely correspond to moving objects. To find such groups, we use the blob analysis System object called 'blobs' or 'connected components' and compute their characteristics, such as area, centroid, and the bounding box.

PROGRAM:

Frame Size: 100 x 100 pixels
Frames Per Second: 40
We use an additional MP3 file to alarm the user about the excessive movement of the object.

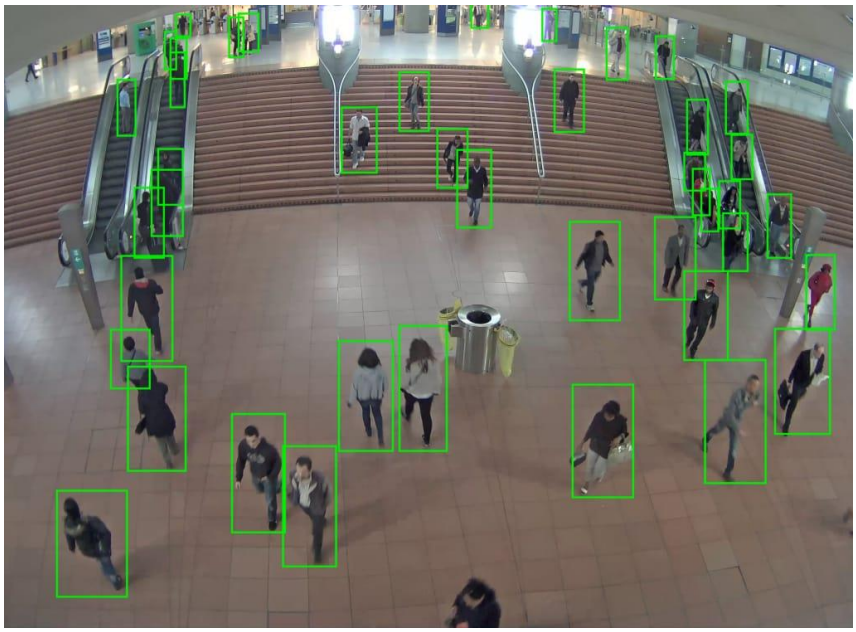
% Video Surveillance MATLAB code

```
clc;
clear;
close all;
 % Create objects for foreground detection and blob analysis
Detector = vision.ForegroundDetector('NumGaussians', 3, ...
        'NumTrainingFrames', 40, 'MinimumBackgroundRatio', 0.7);
blobAnalyser = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...
        'AreaOutputPort', true, 'CentroidOutputPort', true, ...
        'MinimumBlobArea', 400);
% Create the webcam object.
reader =webcam ();
% Create the video player object.
frame2=snapshot(reader);
frameSize = size(frame2);
videoPlayer = vision.VideoPlayer('Position', [100 100 [frameSize(2), frameSize(1)] +30]);
while ~isOpen(videoPlayer)
   frame1 =frame2; %first frame
   frame2 = snapshot(reader); %next frame
   diff=imabsdiff(frame1, frame2); %finding the difference between two frames
   gray=rgb2gray(diff); %RGB to gray image
   blur =  imgaussfilt(gray,'FilterSize',5); %gaussian smoothing filter
   % Detect  foreground.
   mask = Detector.step(blur);
   % Apply morphological operations to remove noise and fill in holes.
   mask = imopen(mask, strel('rectangle', [3,3]));
   mask = imclose(mask, strel('rectangle', [15, 15]));
   mask = imfill(mask, 'holes');
  % Perform blob analysis to find connected components.
  [a, centroids, bboxes] = blobAnalyser.step(mask);
  %stats = regionprops(mask, {'Area','BoundingBox'})
  %this should take the contour area and if it is greater than a
  % certain value it has to play the alarm
   if a>5000
      [y,fs]=audioread('SoundEffect.mp3');
      sound(y,fs);
   end
   % Draw the objects on the frame.
    frame1 = insertObjectAnnotation(frame1, 'rectangle', ...
                bboxes, 1);
   %when g key is pressed it stops analysing
   g = figure (1);
   isKeyPressed = ~isempty(get(g,'CurrentCharacter'));
   if isKeyPressed
      clear cam;
      release(videoPlayer);
      break;
   end
   imshow(frame1)
```

```
end

% Clean up.
clear reader;
Release (videoPlayer);
release (Detector);
```

OUTPUT:



RESULT:

Thus the video surveillance using computer vision and interfacing a webcam was succesfully executed using matlab.