# Movement Detection of Real-Time CCTVImages Using OpenCV and CNN

**Priyanshu Singh, Sneha Dwivedi**

*School of Computer Science & Engineering, Galgotias UniversityPlot No. 2, Yamuna Expy, Opposite, Buddha International Circuit, Sector 17AGreater Noida, Uttar Pradesh 203201 India*

***Abstract*** -In daily life, we get to see CCTV cameras almost everywhere. Themain aim of these cameras is to increase security and reduce unwanted events. Apparently, most of these events occur when the operator is not watching thefootage, and it’s hard to find events in such long recorded footage. This project will use OpenCV Library to detect themovement of objects in the footage.

The footage will be checked frame by frame hence the difference in theprobabilities of the object will clearly state that there is movement or not. This will make it easier for the operator todetect the movement according to the desired event using some queries. For instance, if the operator wants to see the footage when there were 3 persons in it then the algorithm will find that instance only. This project can play a vital role in investigations that involve fetching data from the CCTV footage.

## I. INTRODUCTION

There are vast amounts of data collected by cameras all over the world but robots can only analyze and understand a small portion of it. This study predicts vehicle motion characteristics using real-world recordings from a car's front-camera. Such motion detection tasks are useful foranalyzing pre-recorded videos from devices such as dash cams, providing knowledge of current vehicle status, assisting other downstream tasks such as driver intention inference, [1, 4], and caneasily be extended to other moving objects when combined with the results of other computer vision tasks such asobject detection.

Optical flow is gathered using OpenCV'sbounding boxes of relevant objects (such as cars and people) produced using Faster-RCNN, as well as RGB channels 0and 1. These data are then combined and sent into a Convolutional Neural Network(CNN), which generates the three most critical characteristics of vehicle motion for each frame: forward speed, forward acceleration, and angular velocity, which defines the vehicle's travel direction.

## II. HOW A COMPUTER READS AND IMAGE

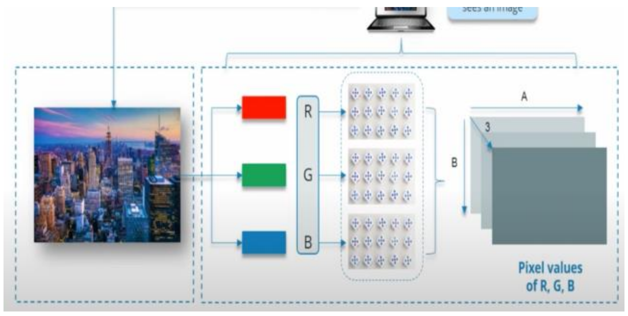


Figure 1. RGB configuration of an image

Suppose we have a colorful image over here, but how a computer will read this image. Basically there will be three channels Red, Green and Blue (RGB). The pixel values for each of these channels are different [9]. So, when we say that the image size is B X A X 3, it means that there are B rows, A columns and 3 channels.

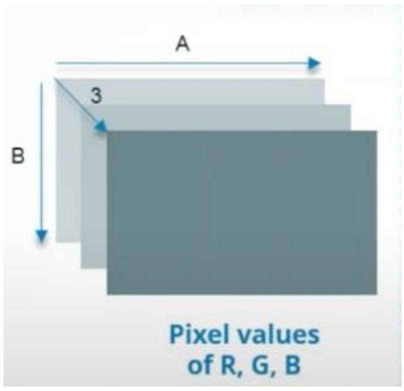


Figure 2. Image Free Diagram

## III. WHY CAN’T WE USE FULLY CONNECTED NETWORKS FOR IMAGE CLASSIFICATION

Consider an image with 28 X 28 X 3 pixels. When we pass this image into a fully connected network then the total number of weights required in the first hidden layer will be 2352. But in real life the images are not that small. The images that we see in daily life are definitely above 200 X 200 X 3 pixels. If we feed this image to a fully connected network then the total number of weights required in the first hidden layer will be 120,000. That means we have to deal with such a big amount of parameters and we’ll need that much amount of neurons.This can eventually lead to *Overfitting*.

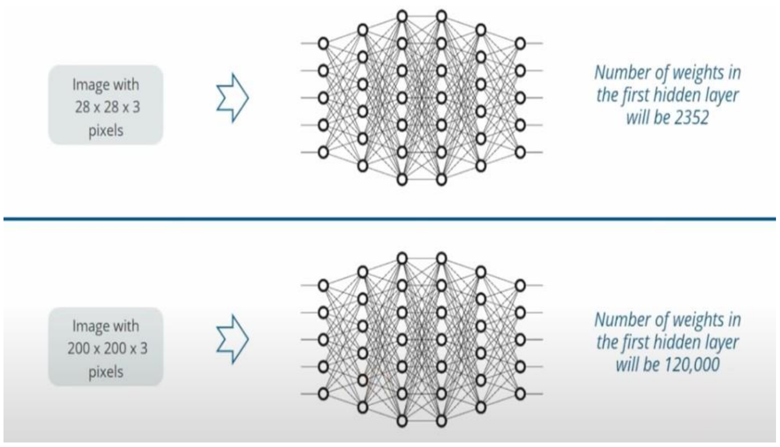


Image with 28 x 28 x 3 pixels
Number of weights in the first hidden layer will be 2352
Image with 200 x 200 x 3 pixels
Number of weights in the first hidden layer will be 120,000

Figure 3. Fully Connected Neural Network

## IV. WHY WE NEED CONVOLUTIONAL NEURAL NETWORKS

In CNN the neuron in a layer will only be connected to a small region of the layer before it instead of all of the r

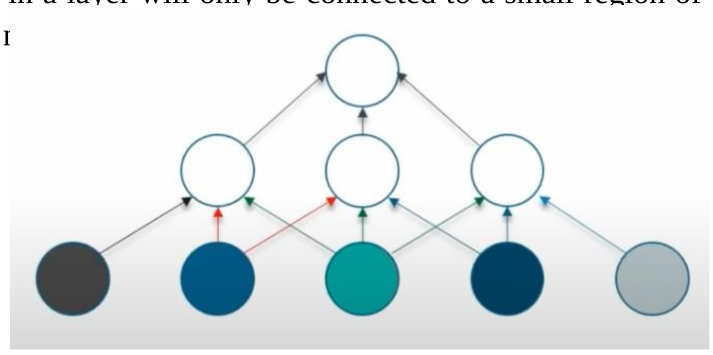


Figure 4. CNN Diagram

Because of this behavior we need to handle less amounts of weights and we need less amount of neurons as well.

CNN is a special type of feed-forward artificial neural network inspired from the visual cortex which is a small region of the human brain. The visual cortex has small regions of cells that are sensitive to specific regions of the visual field. Some individual neuronal cells in the brain respond (or fires) only in the presence of edges of a certain orientation [2]. For example, some neurons fire when exposed to vertical edges and some when shown horizontal or diagonal edges.

## V. HOW CNN WORKS?

Usually convolutional neural networks have 4layers: Convolution layer, ReLU Layer, Pooling Layer, Fully Connected Layer.

To understand this we will take an example of a classifier which can classify an image into an X or an O.
There can be some trickier cases.

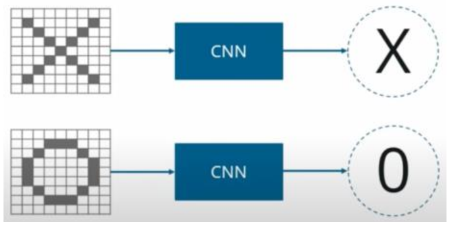


Figure 5. XO Classification

There will be some deformations because X and O images won’t always have the same images.

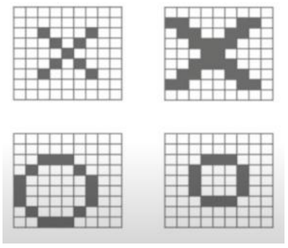


Figure 6. Deformed images of X and O

Now what we're gonna do is, we know that the computer understands an image using numbers at each pixel. In our example, we have considered that a black pixel will have value 1 and a white pixel will have -1 value.

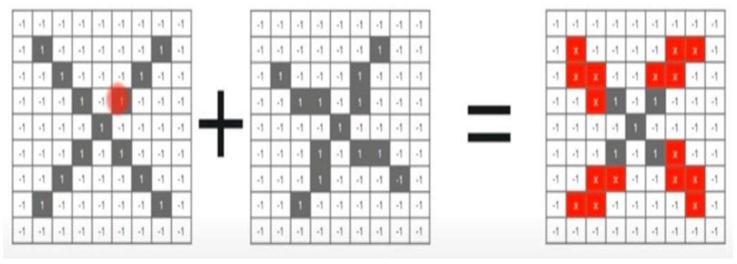


Figure 7. Comparison with normal method

When we'll use normal techniques to compare these images, we'll get to see something like this.

We know that the computer is unable to compare the deformed image of X. This is because it is trying to compare the deformed image of X with the proper image of X as a whole.

Now let's try to use CNN.

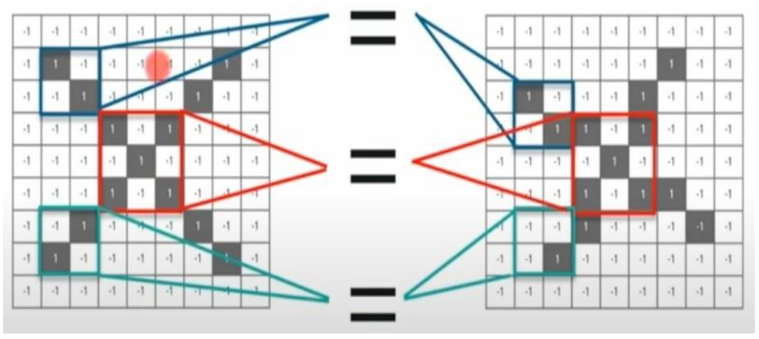


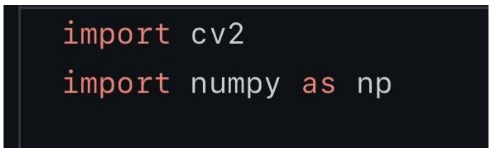
Figure 8. Features of image

Now what we do in CNN, we take small patches of the images. These patches/ pieces are nothing but the features of the images. CNN compares the image piece by piece. CNN sees similarity far better than whole-image matching systems by detecting rough feature matches in about the same location in two pictures [6].

## VI. MOVEMENT DETECTOR - THE PROJECT WORKING

### a. Importing Libraries

Import the necessary libraries to be utilized for example cv2 (OpenCV), NumPy. As the video is contained pictures, and OpenCV will be utilized to preprocess pictures for example beginning the web camera, to peruse video as a pile of edges, manage the shading designing of the casing, track down the gaussian haze of the picture, estimation of the contrast between outlines, picture thresholding, enlargement, shape identification, and bouncing boxes over the movement distinguished.

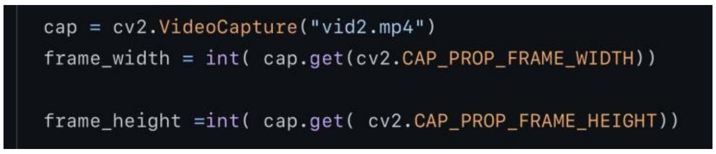


```
import cv2
import numpy as np
```

Figure 9. Importing cv2 and numpy packages

**b. Assignment of Static frame**

To perform movement recognition, we will contrast each casing and the principal outline that will be [3],named as a static casing. So everytime each edge will be contrasted andthe static edge and in view of thedistinction movement in the casings will be recognized. At first, the static casing will be [5]. None and will introduce the principal outline once the web camera opens.

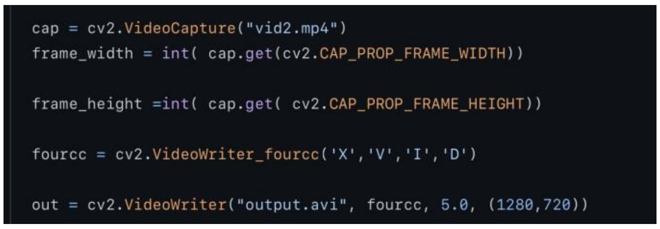


```
cap = cv2.VideoCapture("vid2.mp4")
frame_width = int( cap.get(cv2.CAP_PROP_FRAME_WIDTH))

frame_height =int( cap.get( cv2.CAP_PROP_FRAME_HEIGHT))
```

Figure 10. Applying Static Framing

**c. Create video capture Object**

To catch a live stream from the web camera utilizing OpenCV the initial step is to make a VideoCapture object by passing 0 as a contention to mean the camera to utilize. Set the window size to a specific aspect. Then, we will run while circle, to catch each casing from the web camera and run until the camera will be shut utilizing the waitkey.

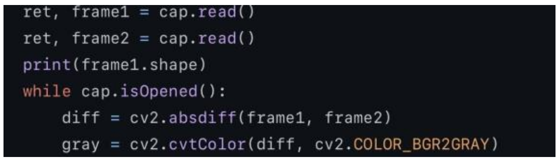


```
cap = cv2.VideoCapture("vid2.mp4")
frame_width = int( cap.get(cv2.CAP_PROP_FRAME_WIDTH))

frame_height =int( cap.get( cv2.CAP_PROP_FRAME_HEIGHT))

fourcc = cv2.VideoWriter_fourcc('X','V','I','D')

out = cv2.VideoWriter("output.avi", fourcc, 5.0, (1280,720))
```

Figure 11. Video object creation for serving into the pipeline

**d. Read the frames and convert intograyscale**

Presently read the edges utilizing object instated above and further convert it into the grayscale design.

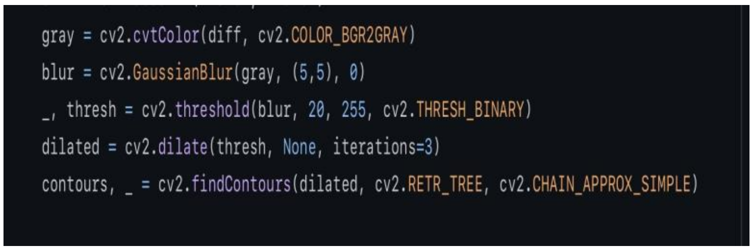


```
ret, frame1 = cap.read()
ret, frame2 = cap.read()
print(frame1.shape)
while cap.isOpened():
    diff = cv2.absdiff(frame1, frame2)
    gray = cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)
```

Figure 12. Frame Reading and Grayscaling

**e. Image Blurring**

Image Blurring : Presently the following stage is to eliminate the commotion from each edge and fundamentally eliminate high frequencies from the pictures utilizing GaussianBlur work by CV2.Pass the tallness and width of the lowpass channel bit as a contention alongside the standard deviation inthe two ways. Additionally, pass the grayscale picture as a contribution to the capacity. In the wake of obscuring the picture, movement in the successive edges will be recognized without any problem.

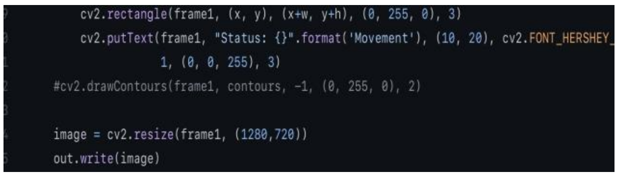


```
gray = cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (5,5), 0)
_, thresh = cv2.threshold(blur, 20, 255, cv2.THRESH_BINARY)
dilated = cv2.dilate(thresh, None, iterations=3)
contours, _ = cv2.findContours(dilated, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

Figure 13. Image Blurring

**f. Update Static Frame**

Presently its chance to refresh the static edge with the principal outline from the live gushing as no progressions in the underlying level. This casing will be utilized as a kind of perspective to ascertain the outright distinction between each casing.



```
        cv2.rectangle(frame1, (x, y), (x+w, y+h), (0, 255, 0), 3)
        cv2.putText(frame1, "Status: {}".format('Movement'), (10, 20), cv2.FONT_HERSHEY_
                    1, (0, 0, 255), 3)
    #cv2.drawContours(frame1, contours, -1, (0, 255, 0), 2)

    image = cv2.resize(frame1, (1280,720))
    out.write(image)
```

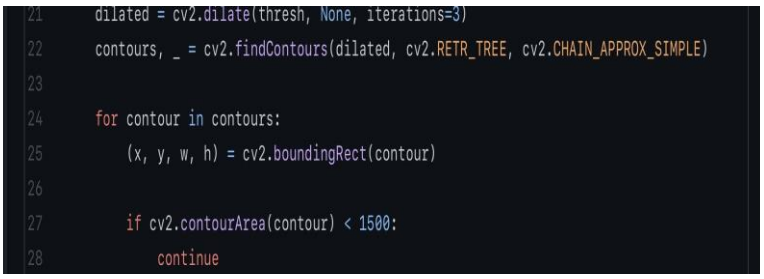
Figure 14. Updation of static Frame

g. **Calculate the difference in theframes and detect motion**

Calculate the difference in the frames and detect motion: Ascertain the outright contrast between the staticedge and each casing from the live streaming. Subsequently the thing that matters is more noteworthy than 30, apply thresholding to the grayscale in order to any place the movement will be distinguished that the area will be white and later enlarge that thresholded picture.

h. **Calculate the difference in theframes and detect motion**

Find Contours: Presently the locale of movement has been changed over to the white utilizing thresholding, which can be treated as shapes, so use track down forms to distinguish directions of the moving articles. Aside from the movement location, when we will get the directions of themoving items to plot the square shape (jumping box) on them and putthe text "Movement Detected".

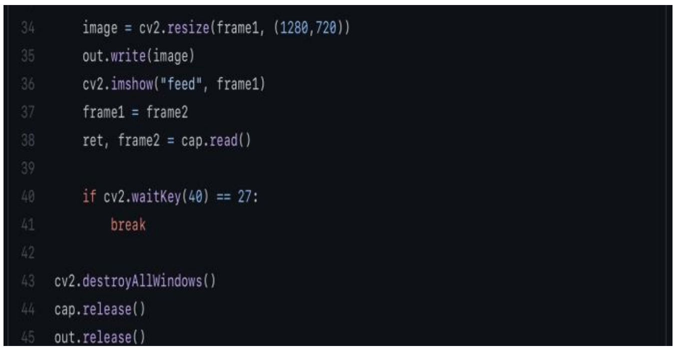


```
    dilated = cv2.dilate(thresh, None, iterations=3)
    contours, _ = cv2.findContours(dilated, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    for contour in contours:
        (x, y, w, h) = cv2.boundingRect(contour)

        if cv2.contourArea(contour) < 1500:
            continue
```

Figure 14. Calculating difference in frames and motion

i. **Display frames in live stream**

Display frames in live stream: Presently the movement has been distinguished in the video, so the subsequent stage is for live gushing with aftereffects of movement location. Put the code to deliver the camera and to annihilate every one ofthe windows.



```
    image = cv2.resize(frame1, (1280,720))
    out.write(image)
    cv2.imshow("feed", frame1)
    frame1 = frame2
    ret, frame2 = cap.read()

    if cv2.waitKey(40) == 27:
        break

cv2.destroyAllWindows()
cap.release()
out.release()
```

Figure 15. Frame display on live stream

OpenCV a PC vision library upholds many capacities and has numerous applications including facial acknowledgment, object location, following human movements, following articles, camera developments, movement identification, and so forth In contrast with different apparatuses, effectively open to everybody as its open-source and its speed is another element.

Many organizations are utilizing OpenCV because of its constant application as it very well may be incorporated with various dialects and a portion of the equipment gadgets for example raspberry pi and some others.

Out of its tremendous applications, this blog has been outlined to cover one of that for example Movement Detection dependent on foundation deduction from the live streaming,where each new edge will be keep deducted from the absolute first edge after picture smoothening and assuming any locale has a worth not exactly some edge, that means the movement recognition in the edges [7]. Later utilizing shapes, plot the jumping boxes on the district where movement has been recognized and put the text around there.

## VII. Methodologies

A pre-trained CNN model, like VGG16, ResNet50, or InceptionV3, which has demonstrated high performance in image classification tasks, is used in the suggested method. These models have already been trained on big datasets, including the millions of photos in ImageNet. In order to extract features from the input image, a feature extractor is used, which is a pre-trained model. A fully connected layer that has been trained to identify whether or not there is movement in the input image receives the feature extractor's output and processes it.
An image dataset containing both motion and still images is needed for the suggested strategy. There are training and testing sets for the dataset. As a feature extractor, the pre-trained model is utilised to pull out features from the training set's images. The output of the feature extractor is passed into the fully connected layer, which uses backpropagation training to determine whether or not the input image has movement.
When the model has been trained, it can be used to identify motion in live CCTV footage. The pre-trained model receives the input image, which is then utilised to detect whether there is movement in the image using the output of the fully connected layer. The location of the movement is marked in an alert if there is movement in the image.

## VIII Results:

A dataset of CCTV photos with and without motions was used to test the suggested approach. 500 photos with motions and 500 without movements were included in the collection. A training set of 800 photos and a testing set of 200 images were created from the dataset. For the test set, the proposed technique had a 98% accuracy rate. Also, real-time CCTV photos were used to evaluate the suggested method, and it was successful in detecting and locating movements there.

## IX. CONCLUSION AND FUTURE WORK

We explain how to anticipate vehicle mobilitystatus using movies obtained by vehicle-mounted front cameras in this paper. We extract a wide range of information from each image, do image preprocessing, and test several input types and CNN designs. Our model can achieve remarkable performance on all three motion characteristics and draw correct inferences about the vehicle's condition after hyperparameter modification. Our quantitative findings are further confirmed and supplemented by the video visualization [10, 11].

We could improveseveral areas of our product if we had more time and money. Because the dataset onlycontains 24 videos, our approaches cannot be generalized on a large scale. It may also be preferable to train a different model for each motion feature rather than combining losses or training the same model with three objectives, which was a proposal that was not taken into consideration owing to time restrictions [8]. Finally, more critical inferences regarding vehicle status and driverintent may be established by merging additional computer vision tasks such as sign and road recognition.

## X. REFERENCE

[1] H. Lin. Vehicle speed detection and identification from a single motion blurred image. In 7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005), 5-7 January 2005, Breckenridge, CO, USA, pages 461–467, 2005.

[2] H. Lin, K. Li, and C. Chang. Vehicle speed detection from a single motion blurred image. Image Vision Comput., 26(10):1327–1337, 2008.

[3] D. J. Dailey, F. W. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. IEEE Transactions on Intelligent Transportation Systems, 1(2):98– 107, 2000.

[4] S. Pumrin and D. Dailey. Roadside camera motion detection for automated speed measurement. In Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on, pages 147–151. IEEE, 2002.

[5] T. N. Schoepflin and D. J. Dailey. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. IEEE Transactions on Intelligent Transportation Systems, 4(2):90–98, 2003.

[6] A. R. YG, S. Kumar, H. Amaresh, and H. Chirag. Realtime speed estimation of vehicles from uncalibrated viewindependent traffic cameras. In TENCON 2015-2015 IEEE Region 10 Conference, pages 1–6. IEEE, 2015.

[7] C. Pornpanomchai and K. Kongkittisan. Vehicle speed detection system. In Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on, pages 135–139. IEEE, 2009.

[8] “PDCA12-70 data sheet,” Opto Speed SA, Mezzovico, Switzerland.

[9] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR, abs/1506.01497, 2015.

[10] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.

*[11] G. Farneback. Two-frame motion estimation based on poly- ¨ nomial expansion. In Proceedings of the 13th Scandinavian Conference on Image Analysis, SCIA’03, pages 363–370, Berlin, Heidelberg, 2003. Springer-Verlag.*