LIVE-CCTV

PROJECT REPORT

Arnav Zutshi | Samiul Sheikh

15.10.21

VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE

MATUNGA, MUMBAI

ACKNOWLEDGEMENT

We would like to thank SRA, VJTI for providing us with the opportunity to work on this project. Working with the seniors was exciting and building something which we had no idea of within this span was something unimaginable. The collective guidance from the mentors ignited a spark within that is undying. In successfully completing this project, many people have helped me. We would like to thank all those who are related to this project.

SPECIAL THANKS TO OUR MENTORS:

1. **HARSH SHAH**
2. **KUSH KOTHARI**
3. **SAHETHI**

TEAM:

1. **ARNAV ZUTSHI**
   (arnzut1324@gmail.com)
2. **SAMIUL SHEIK**

TABLE OF CONTENT

1. PROJECT OVERVIEW

1.1 DESCRIPTION OF USE CASE AND PROJECT

The project has various applications in actuality:

1. The motion recorder method used in this project has various applications in the real world, this can be used in traffic cameras to prevent storage loss and save only relevant data.
2. With object classification it can also be used to identify the defaulters while driving so as to store their number plate and other data.
3. It can be used for defence purposes to detect intruders in border areas.

1.2 TECHNOLOGY USED

1. OpenCV Library:

OpenCV is a library of programming functions mainly aimed at real-time computer vision. It is an open-source computer vision and machine learning software library. This library will allow us to capture the video in front of the camera and will also allow us to detect the required object in front of the screen when used with various classifiers. This library helps in detection of a moving object in the video.

2. NumPy Library:

NumPy is an open-source software library. It allows us to convert images to arrays/matrices and perform various operations that are required to detect

the changes in the video. This library finds its application in equation solving as it has predefined classes that make solving equations given in the form of a matrix easier.

1.3 BRIEF IDEA

To detect any reasonable change in a live cctv to avoid large storage of data. Once we notice a change, our goal is to classify that object or person causing it. We would be using Computer vision concepts. Our major focus will be on Deep Learning and will try to add as many features in the process. The project uses the idea of Euclidean distance to calculate the difference between 2 frames in a video. It also uses Computer Vision to detect any object that is causing movement in the frame of the video and classify it.

2. INTRODUCTION

2.1 GENERAL:

The main idea of the project is to firstly detect any reasonable change that is occurring in real-time video and save that clip i.e to save only relevant data from a live feed of video. Then Computer Visions (Deep Neural Network) to detect what object is causing the change in the frames of the video. The object moving in the frame is given a bounding box with labels i.e the name of the object causing it to make it seem more genuine. Together summing all these ideas and concepts and applying them, we can build our project.

2.2 BASIC PROJECT DOMAINS

1. NEURAL NETWORKS
2. COMPUTER VISION
3. DEEP LEARNING
4. OBJECT CLASSIFICATION
5. MOTION DETECTION

2.3 THEORY

To detect motion i.e reasonable changes between 2 consecutive frames of a video, knowledge of image vectors and numpy library is required. This is for converting the image to an array and then applying various equations over it to calculate the difference between them. Furthermore, for object detection knowledge the OpenCV library's DNN module is used which is passed with a pre-trained configuration file and weights of yolov3. Mathematical concepts of Linear Algebra are very important to proceed with this project. For future application of this project adequate knowledge of Deep Learning and Neural Networks is required along with basic knowledge on Convolutional Neural Networks.

The project works best when further tuned with hyper-parameters and also instead of using the DNN module, a Network is used like darknet or ResNet for object classification.

3. METHODS AND STAGES OF PROGRESS

3.1 APPROACH

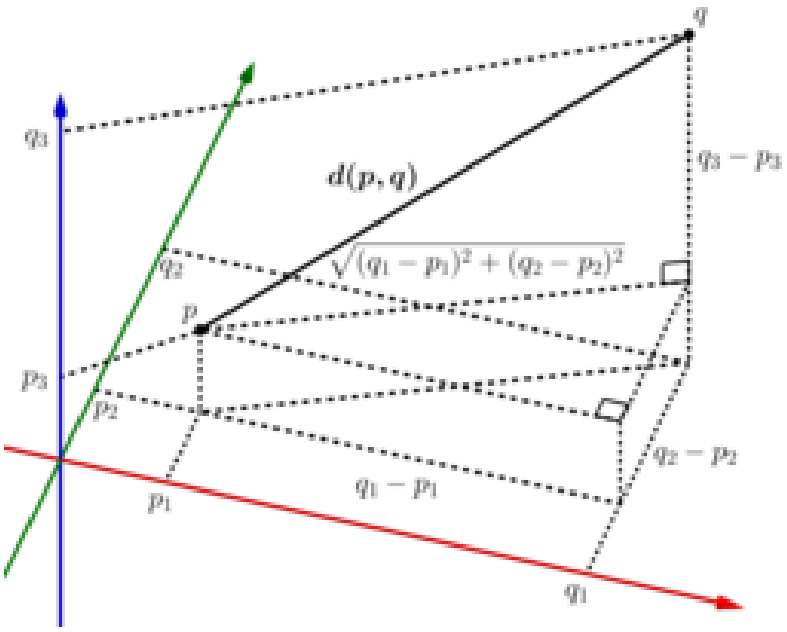
The approach of the project is as described before. First of all its aim is to detect reasonable changes in a video. Our project uses Euclidean distance to calculate the difference between the image vectors of the 2 consecutive frames of a video. The more the distance between the image vectors, the more reasonable the change is and vice-versa. Then the OpenCV library is used to detect objects in the frame and then enclose the object causing the change with a bounding box. The DNN module of OpenCV is used which takes Yolov3 weights and a configuration file as an input and loads them in the Neural Network. The DNN gives out outputs for each of the layers and the final output is a 85 element vector with the element 1-4 being the coordinates of the bounding box, the 5th element being the confidence of the network in that object detected and from 6-85 the scores from each of the 80 classes that can be detected.

3.2 FRAME SIMILARITY

The approach here is taking the Euclidean distance between the 2 image vectors of the consecutive frames of a video. We can also use other methods of detecting similarity like Cosine Similarity, Manhattan distance, Minkowski distance etc. but in this project as said Euclidean distance is implemented but code for other methods is also provided so that one can use it.

This idea was first implemented on 2 images to check if the results obtained are viable and accurate to be used in a video and therefore for this project.

A threshold is then put by carefully analysing the video inputs. Now if the distance between the 2 image vectors of the consecutive frames is less than the threshold the program does not print the distance and those frames are not recorded but as soon as the threshold is achieved then those frames where the change in distance is significant i.e actual changes are significant are saved.



The Euclidean distance if further integrated with CNN will give even better results for image similarity or difference.

3.3 OBJECT DETECTION & CLASSIFICATION

The program first loads the video (Either from the webcam of a custom video can be provided)

It then loads the YOLOv3 weights and configuration file to input them into the Neural Network using **cv2.dnn.readNet()**

Different classes are loaded from the .txt file coco.txt, this file contains 80 objects that are detected by the Neural Network.

The image is then normalized using **cv2.dnn.blobFromImage().** This also resizes the image and performs mean subtraction. Mean subtraction is used to help combat illumination changes in the input images in our dataset. We can therefore view mean subtraction as a technique used to aid our Convolutional Neural Networks.

Further, outputs of the layers of the Neural Network are gathered.

The output of the network is an 85 element long matrix. The first 4 elements are coordinates of the bounding box, the 5th element is the confidence of the object that is detected and the rest are the scores of each class.

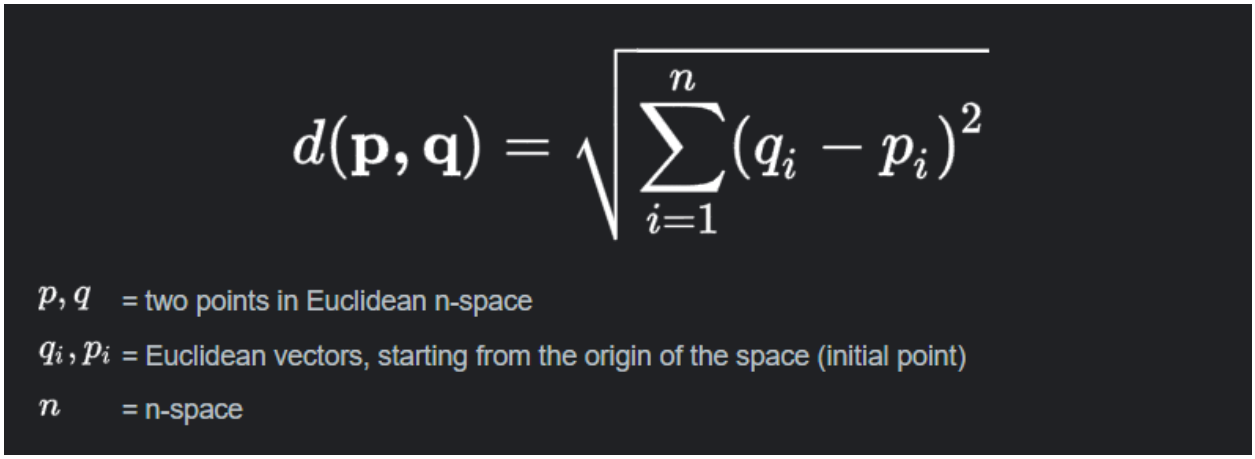
The program is made to proceed with the detection only if the confidence is greater than 0.5.

Further, the coordinates are extracted from the output and **cv2.Rectangle** is used to bound the object detected with a rectangle along with **cv2.putText** to mark the object with its corresponding label.

The Yolov3-tiny network is able to detect 80 objects at once.

3.4 EQUATIONS USED

EUCLIDEAN DISTANCE



The python code for the above equation using NumPy is

eucdis = np.linalg.norm(P - Q)

Where P and Q are image vectors.

The equation takes 2 image vectors and calculates the distance between them. Like said previously, the more similar the consecutive frames of a video are, the lesser the distance between them.

4. CONCLUSION AND FUTURE WORK

4.1 CURRENT EFFICIENCY AND ACCURACY

- **OBJECT DETECTION:**

  The project uses Yolov3 pre-trained weights and configuration file to detect objects in the frames of the video. The accuracy is very good but for the smooth functioning of the program a very high computational power is required. The layers of the pre-trained model give out output in the form of a matrix from which the essential details regarding the labels and the bounding box are extracted.

  The accuracy of the model will be much better if darknet or ResNet is used instead of the pre-existing OpenCV library, but the accuracy and efficiency given by this library is also very good

- **FRAME SIMILARITY:**

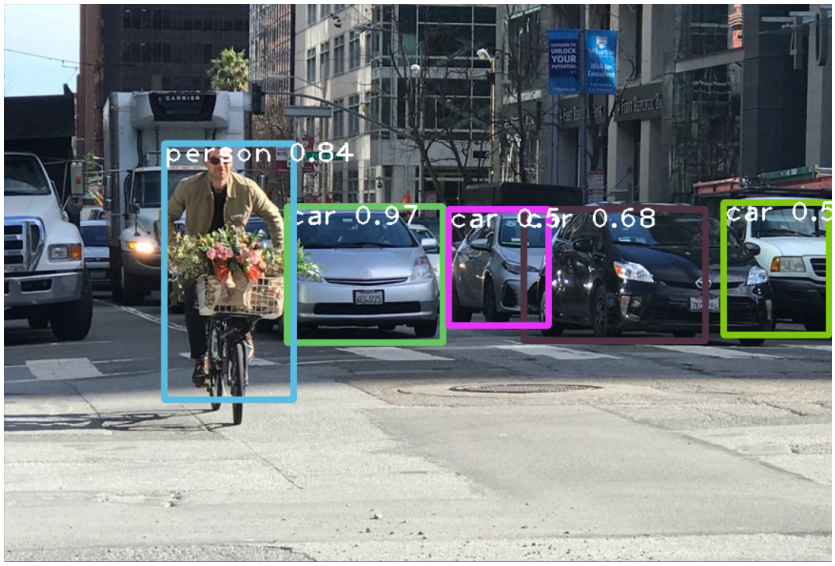
  The project uses Euclidean distance to calculate the similarity/difference between 2 consecutive frames of the video. The accuracy is not that good but it is satisfactory. If this idea is further integrated with CNN to get feature vectors of the frames by linear regression, the accuracy of image similarity will be very high.

  Once object classification is achieved then instead of a frame similarity method to calculate the difference and record, we will use

object classification to record only when the desired object is in the frame of the video.

4.2 ACHIEVEMENTS

1. We learned how to use the OpenCV library and implement it to detect the object in images and bound that moving object with a bounding box.
2. We learned the knowhow of the NumPy library, it's basic functions that can be implemented to images and arrays to get the desired output.
3. We learned about CNN and other Deep Learning algorithms to implement those in our program to classify and detect objects in the video.



The box is for any moving object in front of the camera.

4.3 FUTURE ASPECTS

The project as said previously helps store only relevant data and thus saves storage. The applications of this project are innumerable. After adding a trained model and data set and acquiring better object classification the data storage methods are further improved and tracking number plates of defaulter cars is also a very good usage of the idea. Also with further development, this will be very helpful for self-driving cars for classifying objects as it moves on the road. With sufficient ML and DL algorithms this will turn out to be very helpful in traffic surveillance.

5. REFERENCES

5.1 USEFUL LINKS AND RESEARCH PAPERS/HELPFUL COURSES

- Drive Link
- Coursera courses:
    1. Deep Learning and Neural Networks
    2. Improving Deep Neural Networks: Hyperparameter Tuning, Regulation and Optimization
    3. Convolutional Neural Networks
- Linear Algebra by 3 Blue 1 Brown
- Neural Networks by 3 Blue 1 Brown
- OpenCV Contour Detection
- GitHub Main Repository
- OpenCV Basics Tutorial
- Object Classification