Violence Detection using Embedded GPU

Sagar R. Tharali, Gaurav S. Wakchaure, Durvesh S. Shirsat, Navin G. Singhaniya
Department of Electronics Engineering
Ramrao Adik Institute of Technology
Nerul, Navi Mumbai
Email: sagart2207@gmail.com, gauravwakchaure1998@gmail.com, durvesh301198@gmail.com, navin.singhaniya@rait.ac.in

Abstract—**Since the CCTV cameras been introduced in this world, society has started to depend heavily on the usage of this technology for the high security purposes in most of the public and private areas. It is convenient to use these CCTV footages in courts as evidence and has been beneficial many times. But these footages are given priority and checked later when the incident has already taken place and that too after some period of time and not in real-time of happening. The screening of the multiple CCTV footages on a single monitor is done with very less efficiency as the ratio of number of CCTV footages to that of number of surveillance staff is very high. Also, the human unreliable supervision due to many reasons like tiredness from physical or mental effort, worker boredom, or discontinuous observation make the surveillance more inefficient. To address the issue and automatically detect the violent scenes using surveillance cameras and Embedded GPU in real-time we have developed this project for the benefit of our society. As the alert is generated in real-time, the security can take action immediately to prevent any further damage or mishappening in the crowd. Our primary objective is to automatically differentiate between violent activities and non-violent activities through CCTV surveillance cameras and automatically display the security alert on the screen as soon as any violent activity is captured and thus ensuring the safety of our society.**

I. Introduction

Violence is any intentional act causing injury to another person by way of bodily contact. It can be domestic violence or workplace aggression. These crimes may happen in any crowded places like schools, hospitals, courts, malls, offices, religious places or any other public place. For ensuring the security in these places we heavily depend on the usage of CCTV cameras. In most places, we have the facilities to install CCTV cameras but do they really prove useful to avoid unusual activity eventhough they might be small like hitting one another. CCTV footages are used as an evidence in court but not often. It is also not considered in real-time at the time of happening. The footages are observed and assessed by humans but due to many worker's weariness, boredom, or discontinuous observation makes the scrutiny unreliable. We check footages after everything has happened. This can be avoided if we get an alert and take an action while happening these incidents.

The latest report on annual crime data of India, Crime in India 2017 by the National Crime Records bureau has revealed that there has been an increase of 3.6 % in criminal cases in 2017 as compared to that of 2016 when 50 lakhs of cognisable offences were lodged across the India. There was a 9 % raise in kidnapping and abduction cases in 2017, with 95,893 cases registered against 88,008 in 2016, declared by NCRB. Giving growth to the need of surveillance cameras there has been increase of manufacturing CCTV. According to industrial experts, it is estimated that over a million of surveillance systems were sold every month a couple of years ago. But now it has increased to two million. The Indian market is continuously making the growth of around 20-25% annually. Industries sources have estimated that the security and surveillance market was worth Rs 8,200 crore in year 2017, which reached 11,000 crore in year 2018 and it is expected that it may even touch to Rs 20,000 by the year 2020. But was installing the cameras has solved the problem completely? Although there has been decrease but still many incidents have been encountered and actions were taken after everything has happened. An issue with having so much of data of cameras is that they have captured too much of data for effective human observations. Due to daily increase of this data it becomes difficult for human observer to detect interested scenes from video frames. Thus limiting the benefits of recordings of surveillance cameras.

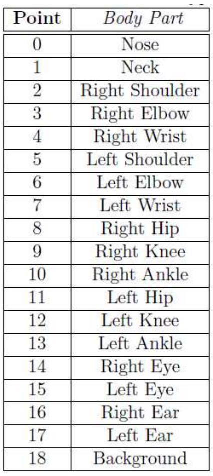
To address this issue and automatically detect the violent scenes using surveillance cameras and Embedded GPU we are aiming to develop and apply Openpose[9] and real time action recognition algorithm[1] in this project for the benefit of our society. Our idea is based on capturing crowded scenes and extracting the visual data and feeding the collected data to action recognition algorithm[1]. Our primary objective is that by using visual surveillance CCTV cameras we can ensure the maximum safety of our society by giving an alert only if any violent activities are happening avoiding the regular human actions which are not threatening to human life. The appearance of human body and behaviour of the people is different for violent activities when compared to that of people exhibiting normal activities. We would be taking live video frames captured by CCTV and extract some features. This extracted data will be compared with our database which will consist of classified frames representing both violent and non-violent behaviour. As soon as result falls in violent category a popup alert will be displayed on surveillance screen. Hence to develop a model that is based on action recognition[1] which can learn the human behaviour and actions through sequences of poses and classifying them into violent and non-violent activity is the sole purpose of this project.

II. Graphical Processing Unit

GPU can render images more quickly as compared to that of CPU because of its capability to do parallel computing at the same time since it has been featured with parallel processing architecture. A CPU also has a higher clock speed, which means it can perform any task or calculations faster than a GPU so it is often better equipped to perform basic computing tasks but not for image processing. CPU and GPU architectures also differ in their number of cores. The core is basically the processor within the processor. Each core processes its own tasks or threads. While CPUs can run only two threads per core whereas GPUs can run four to ten thread per core. Due to the available feature of massive parallel construction, GPUs can run a software algorithm much faster than any other processor could do. As the performance of hardware partially depends on software, the speed and controllability of the hardware is also increased by the use of GPUs as its software support. GPUs also give good results in floating-point operations. A GPU with 384 cores can perform 384 floating-point operations per cycle. Hence, for applications consisting of intensive image processing, GPU is considered as an ideal processing unit. GPUs also offer better backward compatibility. Nowadays, many latest signal and image processing algorithms considers GPUs for computation. Designing with Direct Memory Access (DMA) and very fast memory techniques enabled it to stream high volume data to GPU without consuming clock cycles. Further, GPUs are supported with a wide variety and wide array of free math function libraries and open development tools.

III. Tensorflow OpenPose Algorithm

Pose Estimation is the process of detecting position and orientation of a object. In case of human body detection the prime focus is on the major joints/parts of the body(Ears, Eyes, Nose, Neck, Shoulders, Wrists, Elbows, Hip, Knees, Ankles). In case of a single person pose detection it is comparatively easy process where localizing keypoints and joining the pairs of relatable parts. When there are multiple people in a image frame, the algorithm produces keypoints for each individual. The next step is to figure out which keypoints belong to which individual. The database used by the Openpose algorithm[8] is based model trained on the COCO dataset which uses 18 point skeleton. The keypoints representation is given as below:



Point	*Body Part*
0	Nose
1	Neck
2	Right Shoulder
3	Right Elbow
4	Right Wrist
5	Left Shoulder
6	Left Elbow
7	Left Wrist
8	Right Hip
9	Right Knee
10	Right Ankle
11	Left Hip
12	Left Knee
13	Left Ankle
14	Right Eye
15	Left Eye
16	Right Ear
17	Left Ear
18	Background

Fig. 1. COCO keypoints. [8]

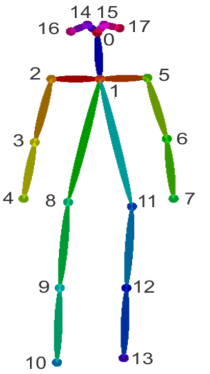


Fig. 2. OpenPose skeleton. [9]

The Openpose model takes a color image as input and produces a confidence map array for each keypoint alongwith Part Affinity Heatmaps for each key-point pair.[26] The algorithm is further divided into stages as explained below:

1) Stage 0: This stage is for creation of feature maps for each input image and consists of ten layers of VGG network. [8]
2) Stage 1:This stage uses 2-branch CNN where each branch does the following:
    a) Predicting the set of 2-D confidence maps of body parts is the main objective of this branch. It is a grayscale image which depicts high value at the prominent position of a body part. In this project the first 5 frames correspond to confidence map's matrices.[8]
    b) Predicting the set of vector fields for respective Part Affinity Fields is the objective of this branch. It illustrates the association between different body parts.[3]

IV. Design Algorithm

The Real Time Action Recognition[1] algorithm divides the process into Preprocessing, Feature Extraction, ML algorithm. Initially the video inputs from camera module are provided which needs to be converted to images. Those images contain certain series of images which define the required action. Only those are considered and OpenPose[8] algorithm is applied.

A. Preprocessing

The skeleton data obtained is stored as a raw data. It must be preprocessed before extracting the features. The following steps are to be followed to process the raw data:

1) Scaling of the coordinates
The output obtained from Openpose algorithm has different unit for x and y coordinates. It is necessary that these units are scaled to same units. The value of height to width ratio is different for each frames. Scaling helps to obtain regular pattern in further processing.[1]

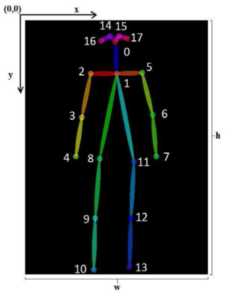


Fig. 3. Scaling of coordinates. [1]

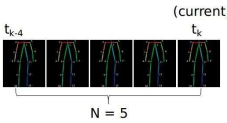


Fig. 4. Concatenation of frames. [1]

2) Remove all invalid joints
The purpose of this project work is to classify between normal activities and violent activities. Initial classification will not require some joints in the skeleton (for example head joints). Those joints must be manually removed to make meaningful feature extraction.[1]

3) Discarding invalid frames
During the frames extraction there are chance of getting frames with no openpose skeleton or highly distorted skeleton output. Such frames are invalid for future processsing and would cause deflections in desired model designing.[1]

4) Filling up of useful missing joints
In some frames the Openpose algorithm might fail to detect a complete human skeleton. It may cause blanks in the joint positions. These values has to be filled with some valid values in order to maintain a fixed-size feature for Classification procedure. One of the solutions is to discard the frame which can cause the algorithm to not detect the person if it not facing the camera or standing sideways. The algorithm uses relative positions from previous frame that was detected. With this method the missing positions can be filled.[1]

B. Feature Extraction

At the end of the previous steps the joint positions are ready to use for feature extracting[1]. According to the application of our project following features must be computed:

1) Concatenation of Joints in N frames
This feature gives out the possible action formation with N frames concatenated altogether.

2) Average Skeleton Height
Due to removal of invalid joints in the previous steps the height of the skeleton frames may vary in height. This step helps it to normalize into average height.

3) Velocity of Body to Neck
It provides the velocity of complete body within the frame.

4) Normalized positions
The position of the subject in the skeleton frames may vary and hence a normalized position frames are extracted for better results.

5) Velocity of joints
Basic human actions can be easily classified by tracking the movements of the limbs. This feature helps extract the velocity of the joints with normalized coordinates.

6) Joint Angles
This will extract the Joint angles required for classification purpose.

7) Limb Lengths
This will extract the Limb lengths required for classification purpose.

C. Neural Networks(NN)

Neural networks consist of neurons with weight and prejudices learned. Each neuron receives several entries, take a weighted sum on them, pass it through a activation function and respond with an output. There are several layers of input and output on NN. The hidden neurons enable the network to learn complex tasks by progressively extracting more meaningful features from input patterns. The basic building block of NN is the convolution layer, which applies the convolution operation as input and passes it to the next layer. Humans can merely recognize any action which come in their sight. Though the same action may be recognized in multiple ways but atlast the action may remain the same as recognized by everyone. The task of recognizing will done using computer vision in case of computers. Computer vision depends on the images captured with certain pixel resolution. In this project approach we obtained the skeleton image data and skeleton text data of violent and non-violent actions. Neural Networks begins with taking a skeleton image with certain pixels that represent a bunch of neurons correponding the pixels sizes. Each neuron holds the number value, ranging from 0(background) to 1(skeleton limbs). This forms the first layer of neural networks. The last layer here is the activity is violent or non-violent that is two neurons. In the middle there are hidden layers which get activated depending upon the preceding layer. Such a structure helps in training the dataset to classify the activities.

V. Implementation

We created our dataset of images by recording action videos. Images containing the precise action postures were sorted manually and fed to the Openpose algorithm[8] followed by the real time action recognition algorithm[1].

- Obtain joints position from OpenPose algorithm.[9]
- Tracing each person in the image frame and distance between the joints of two skeletons is used for differentiating two skeletons.[1]
- Filling in a person's joints on the basis of previous frame or series of frames.[1]
- Extract features according to the (x,y) joint positions.Uses 5 frames(0.5s window size) for this purpose.[1]
- Extract features such as body velocity, normalized joint positions and joint velocities.[1]
- Apply PCA technique to reduce feature dimension. Classify by various machine learning algorithms.[1]
- Filtering the prediction scores between two frames and add appropriate label to the person if the score is greater than 0.8.

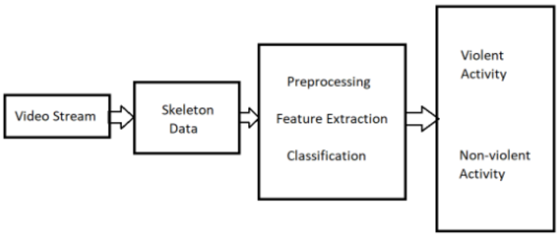


Fig. 5. Workflow.

The obtained model design will be saved as pickle file. The creation of portable and size restricted model design was fulfilled by pickle file. The overall process consisted of many objects and classes which were successfully calibrated and decalibrated with this module. Such file occupies vey less space on a disk. Merits of Pickle Module:

- Same attributes for different Classes
  It has ability to keep a track of data that has been already serialized. So this is helpful for referencing for future use.
- Object Sharing
  Pickle file stores the object once and makes sure that the references are pointing towards master copy. It proves useful for objects that are mutable.
- On-the-go defined classes
  Defining the user needed functions and classes and placing them in the correct module where its objects are located.

VI. Results

The final recognition accuracies for a selective features extracted is shown in the table below. The result shows that the accuracy of all 5 models are higher than 90%. The speed of the algorithms was tested on a laptop with a Geforce MX 150 GPU. The project framerate runs at about 7 fps when the image is resized to 432x368. The fps is seen to decrease for high resize values. The time cost for feature extraction and classification is less than 0.01s per frame for all classifiers, since the models are all relative shallow.

TABLE I. Recognition Accuracy of five models.

Method	*Training Accuracy*	*Testing Accuracy*
Deep Neural Networks	100%	97%
Decision Tree	93%	92%
k-Nearest Neighbors	96%	93%
Linear SVM	92%	90%
Random Forests	100%	96%

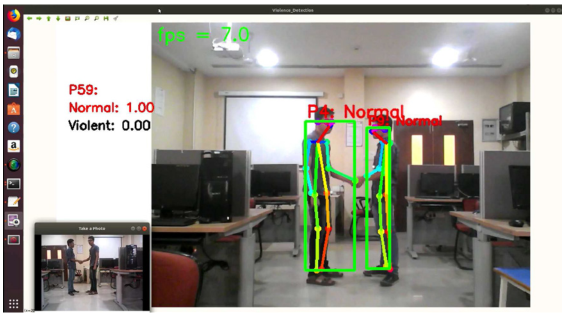


Fig. 6. Action: Non-violent

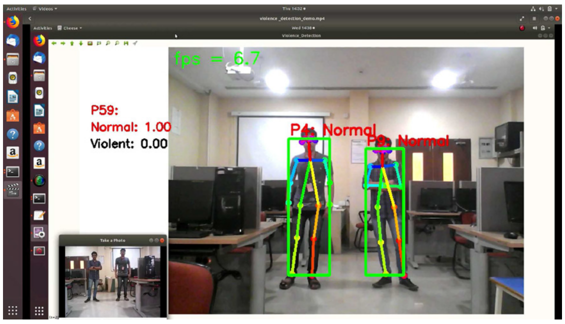


Fig. 7. Action: Non-violent

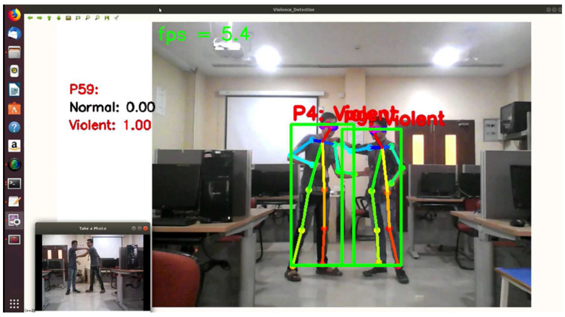


Fig. 8. Action: Violent

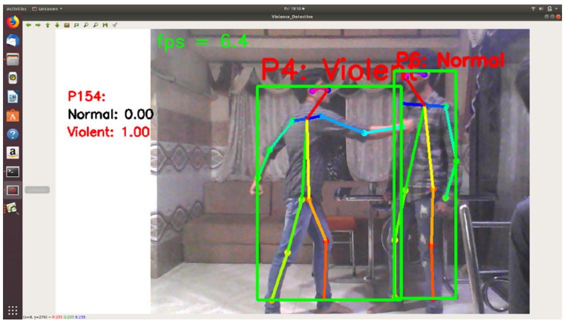


Fig. 9. Action: Violent

VII. Conclusion

We implemented a Violence Detection system that can detect types of actions as Violent and Non-violent. We obtained satisfactory results on our dataset which comprised of violent and non-violent activities. The recognition accuracy was up to 97% on the training set composed of more than 10000 samples. This detection system was then tested on real world videos captured through camera modules. It achieved stable and accurate recognition performance on a video similar to the training set and achieved relatively good result on other videos. Considering realistic issues like surveillance, human-computer interaction, the approach towards building a complex dataset is a state of art application with its broad range of applications. All the small steps taken in this field would lead to robust automatic human actions recognition system in future.

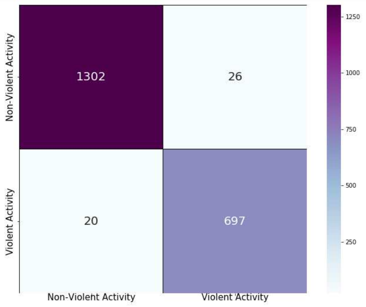


Fig. 10. Confusion Matrix.

References

[1] Feiyu Chen, Real-time Action Recognition Based on Human Skeleton in Video, Final Project of EECS-433 Pattern Recognition, Teacher: Prof. YingWu,https://github.com/felixchenfy/Data-Storage/blob/master/EECS-433-Pattern-Recognition/FeiyuChen_Report_EECS433.pdf

[2] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd Edition, Prentice Hall, 2008, ISBN-13: 9780131687288.

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in CVPR, 2017.

[4] Lillo, I., Soto, A., and Niebles, J. C. (2014). Discriminative hierarchical modeling of spatio-temporally composable human activities, in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Columbus, OH), 812819.

[5] Yang, W., Wang, Y., and Mori, G. (2010). Recognizing human actions from still images with latent poses, in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (San Francisco, CA), 20302037.

[6] Thurau, C., and Hlavac, V. (2008). Pose primitive based human action recognition in videos or still images, in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Anchorage, AK), 18.

[7] Tran, K. N., Kakadiaris, I. A., and Shah, S. K. (2012). Part-based motion descriptor image for human action recognition. Pattern Recognit. 45, 25622572. doi:10.1016/j.patcog.2011.12.028

[8] An example of the skeleton representation obtained using the OpenPose library www.researchgate.com

[9] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in CVPR, 2017.

[10] Deep Neural Network for Image Classification www.datascience-enthusiast.com

[11] Sedai, S., Bennamoun, M., and Huynh, D. Q. (2013a). Discriminative fusion of shape and appearance features for human pose estimation. Pattern Recognit. 46, 32233237.doi:10.1016/j.patcog.2013.05.019

[12] Rodriguez, M. D., Ahmed, J., and Shah, M. (2008). Action MACH: a spatio-temporal maximum average correlation height filter for action recognition, in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Anchorage,AK), 18.

[13] Chaitra B H, Anupama H S, Cauvery N K ”Human Action Recognition using Image Processing and Artificial Neural Networks” International Journal of Computer Applications (0975 8887) Volume 80 No.9, October 2013

[14] Maji, S., Bourdev, L. D., and Malik, J. (2011). Action recognition from a distributed representation of pose and appearance, in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Colorado Springs, CO), 31773184.

[15] L. Weilun, H. Jungong, and P. With, Flexible human behavior analysis framework for video surveillance applications, Int. J. Digital Multimedia Broadcast., vol. 2010, pp.920121-1920121-9, Jan. 2010.

[16] T. Kohonen, Self-Organizing Maps, Springer, Berlin, Heidelberg, 1995.

[17] G.Strang, The discrete cosine transform, SIAM Review, vol. 41, No.1, pp.135 - 147, 1999.

[18] Sigal, L., Isard, M., Haussecker, H. W., and Black, M. J. (2012a). Loose-limbed people: estimating 3D human pose and motion using non-parametric belief propagation. Int. J. Comput. Vis. 98, 1548. doi:10.1007/s11263-011-0493-4

[19] A tutorial on principal component analysis J Shlens - arXiv preprint arXiv:1404.1100, 2014

[20] Classification of Knowledge Based Image using Decision Tree Algorithm Vadthe Narasimha, B.Satyanarayana, K. Krishnaiah, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878,Volume-8, Issue- 1C2, May 2019

[21] RGBD Human Action Recognition using Multi-Features Combination and K-Nearest Neighbors Classification www.researchgate.net/publication/320804220

[22] Human Action Recognition using SVM and KNN Classifiers International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 8616 Volume 5, Issue 8 August 2016

[23] R. Poppe, A survey on vision-based human action recognition, Image and Vision Computing, vol. 28, pp. 976990, 2010.

[24] R. Messing, C. Pal, and H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in ICCV, 2009.

[25] 2017 CVPR: Quo vadis, action recognition? a new model and the kinetics dataset

[26] 2017 CVPR: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields 2007.