IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 7, July 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

Violence Detection Using 3D Convolutional Neural Network

C.M .Suvarna Varma M.Tech, MBA, Ph.D[1], Y. SivaKumari[2], T.Pushpa[3], T.Yogitha[4], V.Manogna[5]

Associate Professor, Dept. of Information Technology, Vasireddy Venkatadri Institute of Technology, Guntur, India[1]

UG Student, Dept. of Information Technology, Vasireddy Venkatadri Institute of Technology, Guntur, India[2,3,4,5]

ABSTRACT: In the past decade, with the growth and advancements in the field of computer vision, an enormous amount of modern techniques has emerged and gained much attention among researchers due to their vast surveillance applications. For instance, in 2017, about 954,261 CCTV cameras were installed in public in South Korea, which was an increase of 12.9% compared to the previous year.

The worldwide utilization of surveillance cameras in smart cities has enabled researchers to analyse a gigantic volume of data to ensure automatic monitoring. An enhanced security system in smart cities, schools, hospitals, and other surveillance domains is mandatory for the detection of violent or abnormal activities to avoid any casualties which could cause social, economic, and ecological damages. Automatic detection of violence for quick actions is very significant and can efficiently assist the concerned departments. A triple-staged end-to-end deep learning violence detection framework. First, persons are detected in the surveillance video stream using a lightweight Convolutional neural network (CNN) model to reduce and overcome the voluminous processing of useless frames. Second, a sequence of 16 frames with detected persons is passed to 3D CNN, where the spatiotemporal features of these sequences are extracted and fed to the Soft max classifier. Furthermore, we optimized the 3D CNN model using an open visual inference and neural networks optimization toolkit developed by Intel, which converts the trained model into intermediate representation and adjusts it for optimal execution at the end platform for the final prediction of violent activity. After detection of a violent activity, an alert is transmitted to the nearest police station or security department to take prompt preventive actions. We found that our proposed method outperforms the existing state-of-the-art methods for different benchmarkdatasets..

I. INTRODUCTION

In the past decade, with the growth and advancements in the field of computer vision, an enormous amount of modern techniques has emerged and gained much attention among researchers due to their vast surveillance applications. For instance, in 2017, about 954,261 CCTV cameras were installed in public in South Korea, which was an increase of 12.9% compared to the previous year.The purpose of these cameras is to ensure security in public places. For this purpose, we focus on the detection of violence using these cameras. Violence is an abnormal behavior and an activity that involves some physical force to damage something, to kill or hurt a human or an animal; these actions can be identified through a smart surveillance system which could be used to prevent these events before further fatal accidents. One of the main functions of surveillance systems deployed on a large scale in different areas, such as schools, streets, parks, and medical centers, is to facilitate the authorities by alerting them to the violentactivity.

However, the response of human operators monitoring the surveillance footage is very slow, causing loss of human life and property; thus, there is a demand for an automated violence detection system and gaining interest in the computer vision society. Many techniques based on deep features and handcrafted features have emerged.

II. RELATED WORK

HHMM also called Hierarchical Hidden Markov Model.

It allows both observed events & hiddenevents.

Itis a StatisticalMarkovmodelinwhichsystemisassumedto be with Unobservable to recognize violent activities by taking Sequentialdata.

Recognizes violent scenes via detecting motion acceleration, motion magnitude, degree of motion, Sound & video modalities for detection of violentactivities. These are represented inHistograms.

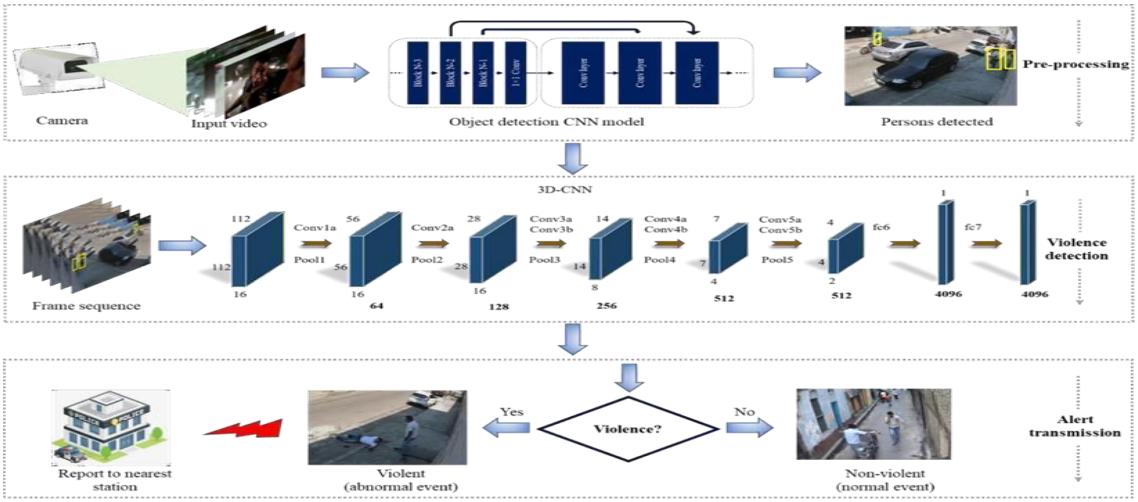
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

III. PROPOSED SYSTEM

Our proposed method in which is used to detect violent activity using an end- to-end deep learning framework. First, the camera captures the video stream VI, which is directly passed to a trained MobileNet CNN model to detect the people. When a person in the video stream is detected, the sequence Š of 16 frames is passed to the 3D CNN model for spatiotemporal features extraction. These features are fed to the Softmax classifier CS to analyze the activity features at the end and give predictions. An alert is sent to the nearest security department when violence is detected so that theycan take immediate action accordingly.



-Fig.1.framework for proposed system

In the first phase, a video stream from a surveillance camera is acquired in which persons are detected. The second phase extracts deep features by feeding a selected sequence of frames to a 3D CNN model which detects the violent activity. Lastly if a violent detected then we report this information to nearest police station.

3.1 Pre-processing

Person detection is an essential step in our proposed method to ensure efficient processing before the violence detection step. In this section, we detect the persons in the video stream for efficient processing. Instead of processing the whole video stream, we process only those sequences that contain persons by avoiding unimportant frames. The video stream is fed into the Mobile Net-SSD CNN model for person detection. We used this CNN architecture because it helps the system to restrict for latency and size. Mobile Net possesses depth wise separable convolutions to detect objects instead of regular convolutions. If depth wise and point wise convolutions are counted separately, there are 28 layers, where every layer is followed by nonlinearity batch norm and ReLU except the final fully connected layer. The first convolutional layer contains a stride of two with a filter shape of $3 \times 3 \times 3 \times 32$ and has an input size of $224 \times 224 \times 3$; its next depth wise convolution has one stride, the filter shape is $3 \times 3 \times 32$, and the input size is $112 \times 112 \times 32$. The Mobile Net is mainly used for classification while its SSD version is used to locate the multi box detector, and their combination performs object detection. For this purpose, the SSD is added at the end of the network, which performs feed forward convolution and produces a fixed-size group of bounding boxes, to ensure the presence and detection of object instances in those boxes via extracting the features map and applying the convolution filters. The boundary box is composed of a predicted class with a probability for each class. The class with thehighest probability indicates the object, while zero represents no object indication. A demonstration of person detection in some samples of the hockey fight dataset is shown inFigure2.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

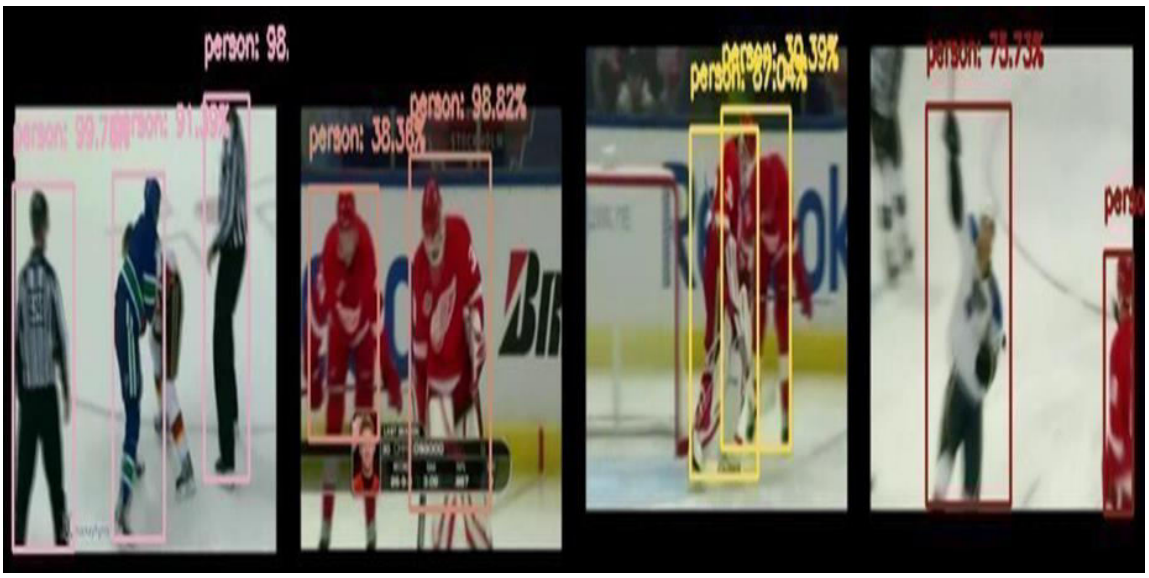


Fig2: Persons detected in the frames of both classes from video clips of the hockey fight data set using mobile net SSD

3.2 Learning with 3D CNN

A 3D CNN is well-suited to extract spatiotemporal features and can preserve the temporal information better owing to its 3D convolution and pooling operation. In addition, in 2D CNNs, there is spatial information only, while a 3D CNN can capture all temporal information regarding the input sequence. Some of the existing methods use 2D ConvNets to extract the spatial correlation in video data, which possess temporal correlation. For instance, in the 2D CNN processes multiple frames, and all the temporal feature information is collapsed. The 3D convolution operates by convolving a 3D mask on the cube designed via assembling attached frames. The obtained feature maps from the convolution layer are linked to multiple attached frames in the prior layer, capturing the motion information.

Only one type of feature is extracted by 3D convolutional mask from the frame cube since the weights of the kernel are replicated in the entire cube. In Figure3.3, the feature maps of the 3D CNN obtained from two layers conv3a and conv5a are provided. The input sequence is taken from the violence category in the movies' dataset. A principle for CNN is to increase the amount of feature maps in late layers by creating several kinds of features from the same feature maps. The input data to this network is a sequence of frames. Before starting the training process, the volume mean of training and testing data is calculated. The architecture of the network is fine-tuned to obtain these sequences as inputs. The final prediction at the Softmax layer is calculated as belonging to the violent or non-violentclass.

3.3 Data Preparation and Usage

This section specifies the preparation of data and their usage for learning violence activity patterns. First, violence dataset Dˇ was used, containing Nˇ number of short video clips with different duration's. Each video dataset contains two categories: i.e., violent class and non-violent class. Before the learning process, the whole dataset Dˇ was divided into a sequence of 16 frames Š with an 8-frame overlay between the two successive clips. Subsequently, having obtained the frames, we split the whole data into training and testing sets. For this purpose, we used 75% and 25% of data for training and testing, respectively. Once the training and testing data were obtained, we generated a file list containing the paths of training list LTr = {S1, S17, S33, . . ., SN} and testing list LTe = {S1, S17, S33, . . . , SN}.The subscript of S is the starting frame number in the sequence where each path is given in the list, pointing towards the extracted frames in the directories.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

3.4 C3D Network Architecture

Inspired by the performance of 3D CNN in, we also fine-tuned the 3D CNN model proposed in. A starting version of the C3D model was developed in 2014 with a version of Caffe. This network consisted of eight convolutions: five pooling and two fully connected layers with a Soft max output layer. Each convolutional layer has 3 × 3 × 3 kernels with one stride, and all the pooling layers are max pooling with a 2 × 2 × 2 kernel size except for the first pooling layer where kernel size is 1 × 2 × 2 with two strides, preserving the time-based information. The number of filters in each convolution is 64, 128, 256, for first, second, and third layers, respectively. The kernels for each convolution have a defined temporal depth, with size D. The kernel size and padding used to apply the convolution were kept as 3 and 1, respectively. Two fully connected layers (fc6 and fc7) contained 4096 neurons and the Soft max layer containing N number of outputs depended on the classes of the dataset. In our case, the output is only two because we have only two classes: i.e., violent andnon-violentscenes.Theoveralldetailedarchitectureisillustratedin Figure.3.

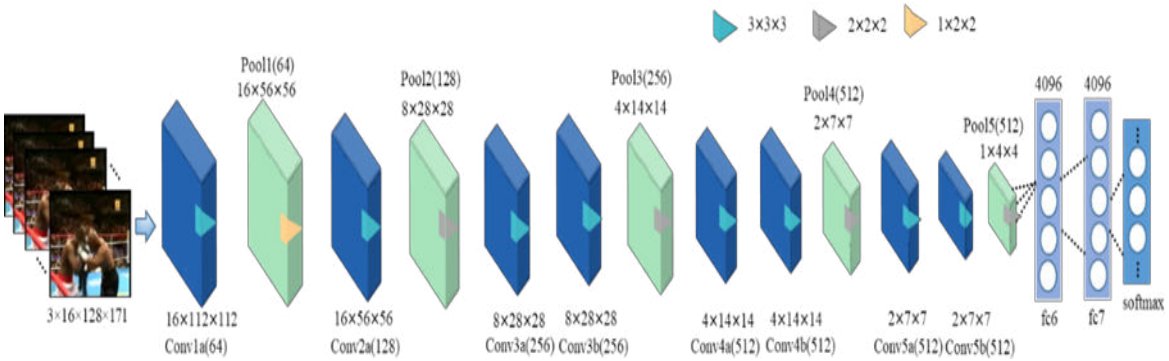


Fig.3: The architecture of C3D containing eight convolutional, five max pooling, and two fully connected layers, followed by Soft Max layer (Output). The three color cones
Represent the different size of filters for each layer.

This architecture of a 3D convolutional network obtained the short sequence of 16 frames as an input of size 128 × 171, but we usedrandom crops of size 3 × 16 × 112× 112 from the original input sequence at the time of training to avoid the over fitting problem and to achieve effective learning. After this, the sequence of frames is followed by 3D convolution and pooling operations. When training is performed, the network acts as a generic feature extractor. In fact, diverse features are learned at each layer of hierarchy in the network. The bottom's activation layers contain smaller receptive fields making it sensitive towards patterns, such corners, edges, and shapes, while the top activation layers contain larger receptive fields learning high-level and global features to collect complex invariance's. Finally, the output label is violent or non-violent.

3.5 Model Optimization

Modeloptimizationistheprocessusedtogenerateanoptimalandfine-tuned designmodel based on some prioritized constraints while keeping the model strength, efficiency, and reliabilitymaximized. Optimizing the model enables CNN network inference at the end and speeds up the process by using pre-optimized kernels and functions. Inspired by these strategies, we used an open source toolkit known as OPENVINO provided by the Intel Corporation. This toolkit extends the work process across the hardware by maximizing its performance. It works on Intel hardware and takes pre-trained models, such as Caffe, ONNX,MXNet,andTensorFlow,asinputsandconvertstheseintoanIRusinga model optimizer. The model optimizer is used to enable a transition between the training and deployment floor to adjust the model for optimal execution on the end platform. Figure shows the flow and process of the model optimization, taking the trained model as input and producing an intermediate model. At the end platform, this output is deployed for furtheranalysis.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

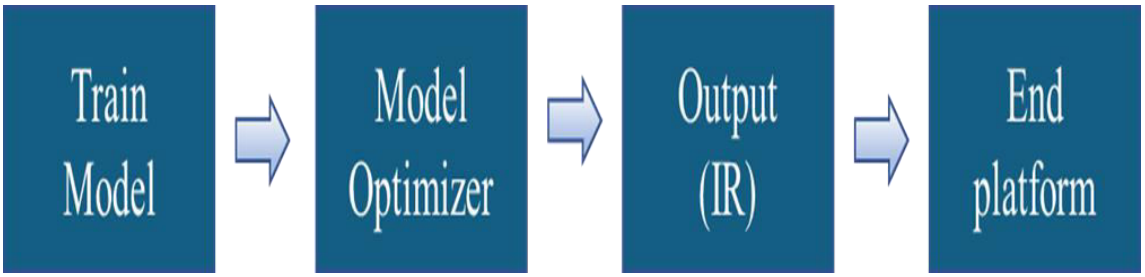


Fig.4.The flow of converting the trained model into intermediate representation (IR) format using model optimizer, where the IR format of the model is further
Used at the end platform

IV. CNN ALGORITHM AND BACKPROPAGATION ALGORITHM

3.3 CNN Algorithm and Back propagation algorithm

3.1.1 Forward pass

output of neuron of row k, column y in the l th convolution layer and k th feature pattern:

$$O_{x,y}^{(l,k)} = \tanh\left(\sum_{t=0}^{f-1}\sum_{r=0}^{k_h}\sum_{c=0}^{k_w} W_{(r,c)}^{(k,t)} O_{(x+r,x+c)}^{(l-1,t)} + Bias^{(l,k)}\right) \quad (3.1)$$

among them, f is the number of convolution cores in a feature pattern。

output of neuron of row x, column y in the l th sub sample layer and k th feature pattern:

$$O_{x,y}^{(l,k)} = \tanh\left(W^{(k)}\sum_{r=0}^{S_h}\sum_{c=0}^{S_w} O_{(x\times S_h+r,y\times S_w+c)}^{(l-1,k)} + Bias^{(l,k)}\right) \quad (3.2)$$

the output of the j th neuron in l th hide layer H :

$$O_{(l,j)} = \tanh\left(\sum_{k=0}^{s-1}\sum_{x=0}^{S_h}\sum_{y=0}^{S_w} W_{(x,y)}^{(j,k)} O_{(x,y)}^{(l-1,k)} + Bias^{(l,j)}\right) \quad (3.3)$$

among them, s is the number of feature patterns in sample layer.

output of the i th neuron l th output layer F

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

$$O_{(l,i)} = \tanh(\sum_{j=0}^{H} O_{(l-1,j)} W^{l}_{(i,j)} + Bias^{(l,i)}) \quad (3.4)$$

3.1.2 Back propagation

output deviation of the k th neuron in output layer O:

$$d(O_k^O) = y_k - t_k \quad (3.5)$$

input deviation of the k th neuron in output layer:

$$d(I_k^O) = (y_k - t_k)\varphi'(v_k) = \varphi'(v_k)d(O_k^o) \quad (3.6)$$

weight and bias variation of k th neuron in output O:

$$\Delta W^O_{k,x} = d(I_k^O) y_{k,x} \quad (3.7)$$

$$\Delta Bias_k^O = d(I_k^O) \quad (3.8)$$

output bias of k th neuron in hide layer H:

$$d(O_k^H) = \sum_{i=0}^{i<17} d(I_i^O) W_{i,k} \quad (3.9)$$

input bias of k th neuron in hide layer H:

$$d(I_k^H) = \varphi'(v_k)d(O_k^H) \quad (3.10)$$

weight and bias variation in row x, column y in the m th feature pattern ,a former layer in front of k neurons in hide layer H

$$\Delta W^{H,k}_{m,x,y} = d(I_k^H) y^m_{x,y} \quad (3.11)$$

$$\Delta Bias_k^H = d(I_k^H) \quad (3.12)$$

output bias of row x, column y in m th feature pattern ,sub sample layer S

$$d(O^{S,m}_{x,y}) = \sum_{k}^{170} d(I^H_{m,x,y}) W^{H,k}_{m,x,y} \quad (3.13)$$

input bias of row x, column y in m th feature pattern ,sub sample layer S

$$d(I^{S,m}_{x,y}) = \varphi'(v_k)d(O^{S,m}_{x,y}) \quad (3.14)$$

weight and bias variation of row x, column y in m th feature pattern ,sub sample layer S

$$\Delta W^{S,m} = \sum_{x=0}^{fh}\sum_{y=0}^{fw} d(I^{S,m}_{\lfloor x/2 \rfloor \lfloor y/2 \rfloor}) O^{C,m}_{x,y} \quad (3.15)$$

among them, C represents convolution layer.

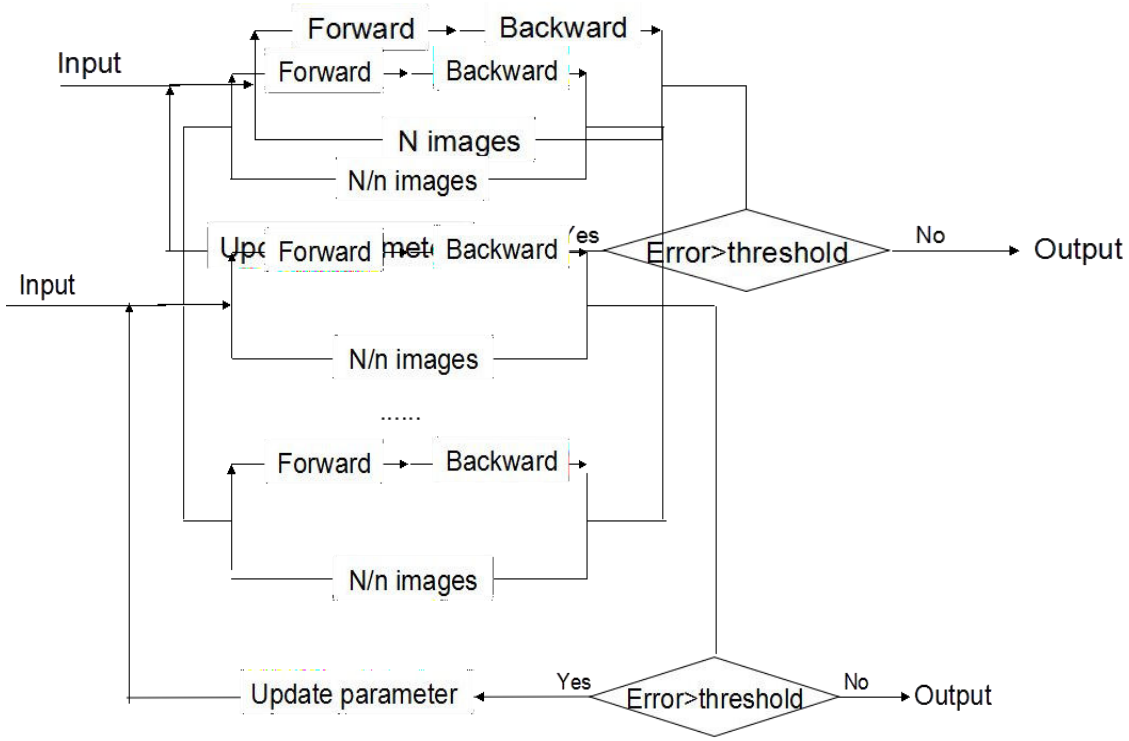
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

V. PARALLEL STRATEGY AND EFFICIENCY

This analysis is based on a hypothesis that both serial and parallel method have same number of training .In serial realization method, the total execution time is N times of the sum of t1 and t2



Timeofserialexecution:$(t_1 \; t_2) \; N \; t_3 \; t_{serial}$

Timeofparallel execution:$\max\{(t_1' \sqcup t_2')\} \sqcup (N/n) \sqcup t_3' \sqcup t_{parallel}$

speed-upratio=$t_{serical}/t_{parallel}$

Speed-up efficiency=speed-upratio/nN:num of images

n:num of nodes

t_1:time of forward pass fortraining a picture

t_2:time of backward propagation for training a picture

t_3:time for updating weight and bias of convolution neural network

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

V. EXPERIMENTAL RESULT

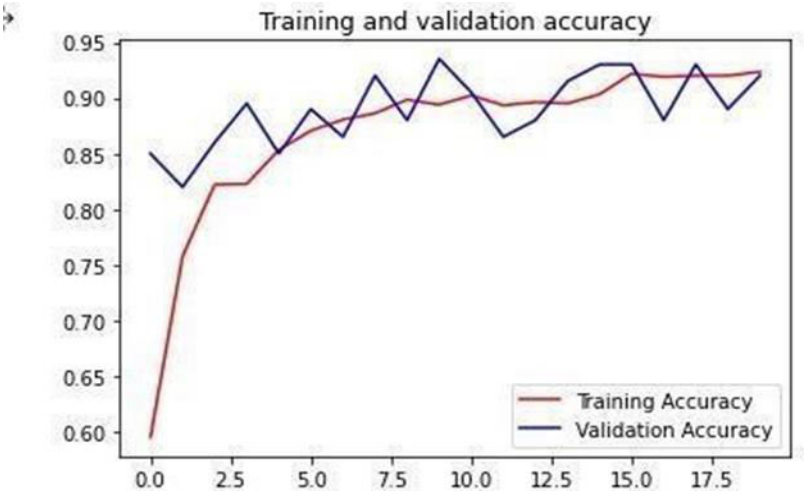
Real Life Vilenece Dataset is data set used to train and test the model

```
+ Code  + Text

!unzip \*.zip

Archive:  violenceandnonviolencevideos.zip
  inflating: Real Life Violence Dataset/NonViolence/NV_1.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_10.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_100.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_1000.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_101.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_102.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_103.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_104.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_105.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_106.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_107.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_108.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_109.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_11.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_110.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_111.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_112.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_113.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_114.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_115.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_116.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_117.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_118.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_119.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_12.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_120.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_121.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_122.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_123.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_124.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_125.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_126.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_127.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_128.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_129.mp4
  inflating: Real Life Violence Dataset/NonViolence/NV_13.mp4
```

Training and Validation Accuracy:



It is reported the average value of accuracy on all the classes that, to the contrary of the loss, increases with the increases of the data sets. Such values of loss and accuracy do not differ much among the training and test dataset, allowing us to affirm that the model does not turn out to be over fitted while training. The training set is used to train the model, while the validation set is only used to evaluate the model's performance.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907061 |

VI. CONCLUSION

In this paper, a three-staged end-to-end framework is proposed for violence detection in a surveillance video stream. In the first stage, persons are detected using an e*ffi*cient CNN model to remove unwanted frames, which results in reducing the overall processing time. Next, frames sequences with persons are fed into a 3D CNN model trained on three benchmark datasets, where the spatiotemporal features are extracted and forwarded to the Softmax classifier for final predictions. Finally, an OPENVINO toolkit is used to optimize the model to speed up and increase its performance at the end platform. Experimental results over various benchmark datasets confirm that our method is the best fit for violence detection in surveillance and achieved better accuracy than several employed techniques. In the future, we intend to ensure our system is implemented over resource-constrained devices. Furthermore, we plan to propose edge intelligence for violence recognition work in the IoT using smart devices for quickresponses.

REFERENCES

1. W. Sultani, C. Chen, and M. Shah, "Real-world Anomaly Detection in Surveillance Videos," Center for Research in Computer Vision (CRCV), University of Central Florida (UCF),2018.
2. I. S. Gracia, O. D. Suarez, G. B. Garcia, and T.-K. Kim, "Fast fight detection," PloS one, vol. 10, no. 4, p. e0120448,2015.
3. L. Tian, H. Wang, Y. Zhou, and C. Peng, "Video big data in smart city: Background construction and optimization for surveillance video processing," Future Generation Computer Systems,2018.
4. C. Dhiman and D. K. Vishwakarma, "A review of state-of-the-art techniques for abnormal human activity recognition," Engineering Applications of Artificial Intelligence, vol. 77, pp. 21–45, 2019.
5. Greenfield, M. Change in the Number of Closed-Circuit Television (CCTV) Camerasin Public Places in South Korea.Avaliable online: https://www.statista.com/statistics/651509/south-korea-cctv-cameras/(accessed on 25 April 2018).
6. Nievas, E.B.; Suarez, O.D.; García, G.B.; Sukthankar, R. Violence detection in video using computer vision techniques. In Proceedings of the International Conference on Computer Analysis of Images and Patterns, Seville, Spain, 29–31 August 2011; pp. 332– 339.
7. Khan, S.; Muhammad, K.; Mumtaz, S.; Baik, S.W.; de Albuquerque, V.H.C. Energy- Efficient Deep CNN for Smoke Detection in Foggy IoT Environment. IEEE Internet Things J.**2019**.
8. Sajjad, M.; Khan, S.; Muhammad, K.; Wu, W.; Ullah, A.; Baik, S.W. Multi-grade brain tumor classification using deep CNN with extensive data augmentation. J. Comput. Sci. **2019**, 30,174–182.
9. Sajjad, M.; Khan, S.; Hussain, T.; Muhammad, K.; Sangaiah, A.K.; Castiglione,A.;
10. Esposito, C.; Baik, S.W. CNN-based anti-spoofing two-tier multi-factor authentication system. Pattern Recognit. Lett. **2018**.
11. Mahadevan, V.; Li, W.; Bhalodia, V.; Vasconcelos, N. Anomaly detection in crowded scenes. In Proceedings of the 2010 IEEE Conference on Computer Visionand Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 1975– 1981.
12. Hassner, T.; Itcher, Y.; Kliper-Gross, O. Violent flows: Real-time detection of violent crowd behavior.In Proceedings of the 2012 IEEE Computer SocietyConference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, USA, 16–21 June 2012;pp.1–
13. Huang, J.-F.; Chen, S.-L. Detection of violent crowd behavior based on statistical characteristics of the optical flow. In Proceedings of the 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Xiamen, China, 19– 21 August 2014; pp.565–569.
14. H. Dar, M. I. Lali, H. Ashraf, M. Ramzan, T. Amjad, and B. Shahzad, "ASystematic Study on Software Requirements Elicitation Techniques and its Challenges in Mobile Application Development," IEEE Access, vol. 6, pp.63859–63867,2018.62
15. R. Piryani, D. Madhavi, and V. K. Singh, "Analytical mapping of opinion mining and sentiment analysis research during 2000--2015," Information Processing & Management, vol. 53, no. 1, pp. 122–150,2017.
16. R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," Neurocomputing, vol. 275, pp. 66–72, 2018.
17. P. C. Ribeiro, R. Audigier, and Q. C. Pham, "RIMOC, a feature to discriminate unstructured motions: application to violence detection for video-surveillance," Computer vision and image understanding, vol. 144, pp. 121–143, 2016

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 7.542

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details