Violence Detection in Crowd using Deep Learning

Sanket Sheth
Graduate Student in Computer Science
Rochester Institute of Technology
Rochester, NY 14623
sas6792@g.rit.edu

Varun Mantri
Graduate Student in Computer Science
Rochester Institute of Technology
Rochester, NY 14623
vm9324@g.rit.edu

Ifeoma. Nwogu
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
ion@cs.rit.edu

Figure 1: The above images illustrates huge crowds as being violent and non-violent

Abstract

According to Integrated Health-care Strategies (IHS) survey there are about 245 million professionally installed video surveillance cameras world wide. However there are only a few trained professionals available to monitor footages from these cameras and make meaningful inferences in real time. In our project we try to solve this problem by suggesting a novel technique of classifying a video as violent / non violent by training Convolutional Neural Network (CNN) on crowd surveillance data set. Our objective is to classify a surveillance footage as violent or non-violent as quickly as possible without human intervention. We aim to use two salient approaches to train our model and plan to keep the one that performs the best.

1 Introduction

Surveillance cameras are widely used and available all over the world with constant supervision by humans to check for any anomalies, the main problem arises with the human part of this, with human supervision we get human error along with manipulation and also the need of the human in the first place.Our system proposes the detection of violence in a scene obtained from surveillance footage, as these videos do no consist of any audio tracks the system only can rely on visual features. The idea is to detect crowd based violence and with crowd comes the issue of too much motion and hence we dismiss the use of high level motion features and analysis and instead dive into changes observed in low level features for classification. Short frame sequences are used to classify the videos two ways using deep learning.The first method uses convolutional neural networks in it's pure form and the the other method uses the same CNN but with STIP(Space-Time Interest Points) as he input.The

videos are obtained from a annotated public database used in a simillar project as ours T. Hassner and Kliper-Gross [2012]

2 Past related work

Violence detection is subtask of **action recognition** can be frame-based or interest-point based, in case of motion based interest points the problem arises when there are too few interest points or like in the cases of crowds too much motion bag of words approach fails immensely. The frame based method is efficient but uses a search based approach which is not practical(Too slow) for real time detection.Liu et al. [2009] Dollar et al. [2005] Boiman and Irani Boiman and Irani [2005] proposed an approach that involved categorizing videos as violent by analyzing sudden changes in videos. Hendel et al Hendel et al. [2011] defined a probabilistic method to detect sudden changes by using space-time tubes containing an object moving in the scene. This method is know to under-perform with crowd videos. Another approach is to use dynamic features produced by a stochastic process which are stationary in space and time but crowds are not stationary but recently local binary pattern(LBP) have proven to be quite efficient. Crook et al. [2008] Zhao and Pietikainen [2007] Hassner, Yossi and Klipper-Gross T. Hassner and Kliper-Gross [2012] proposed a unique method for violence detection using their unique feature descriptor called as ViF (Violent Flows). They classified surveillance clips as violent/non-violent using ViF descriptors and Support Vector Machines (SVM). In our opinion, their work is by far the best when it comes to making predictions in real time and we plan to derive motivation from their work in our project.

3 Dataset

The dataset that is used is an annotated dataset that is a mixture of surveillance data and other in the wild videos acquired from youtube. The total number of videos is about 246 with half of them annotated as violent and other as non-violent.The shortest video is of 1 second and the longest is 6.52 seconds with an average duration of 3.6 seconds justifying our approach to work with a short number of frames. The videos are split into 5 different categories each displaying some type of crowd situation whether a sporting or other social gathering with many people with half displaying acts of violence and the other half displaying normal behavior. As the idea is to detect violent behavior in crowds and perform actions to stop it through surveillance cameras and other forms of monitoring. A rather in-depth motive is to understand crowd behavior from image data analysis.

The data is in video and the model is fed images that are the frames extracted from the video data with a total image count of about 21828 images with 9421 non-violent and 12407 violent images in separate folders marked as labels before preprocessing kicks in.The ratio of violent to non- violent data points is about 57:43 which is mildly biased towards the violent data. The training: testing split considered is 80:20.

4 Approach

4.1 Preprocessing

The first part of the approach is to use the video data set as several images. The reason behind it is that the features used are not temporal which that is, the features are more suitable for images also the other reason is of data, we begin with a modest number of videos, but converting them to images would let us work with a very rigorous dataset and help the model generalize better.To achieve this OpenCV was used with python scripting and each video was converted into many frames. An advantage that comes along with using images instead of videos is the discrepancy in the length of the videos which if used would have needed normalization that is converting each video to the same length as CNN requires a consistent size of the feature vector.

The second step is to select our features from the several images obtained, after contemplating with some spatial features like the histogram of orientations using descriptors like SIFT, but after experimentation we came to the conclusion that using orientation based features will surely give bad results as the data at hand contain drastic actions which would rattle the descriptors and the number

of interest points may be several to very few in number. Thus, we ended up selecting the intensities of the images as our features for the neural network.

The next step is to pre-process the data extracted for the convolutional neural network being used further down the process.Initially, we used a vectorized version of all images which was of the size (320 x 240) which was humongous along with our dataset and the network when training dropped into the problems of memory issues on a machine with fairly good specifications. Thus we took the decision of skewing the data at hand by converting each image used to a size of (64 x 48). This was achieved in Matlab and the images saved for further processing.

4.2 Architecture Explanation

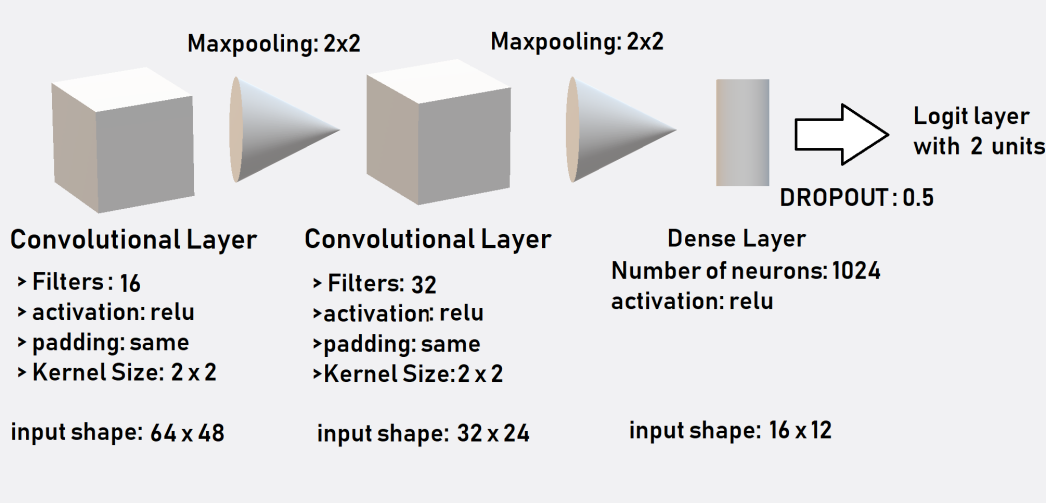

Figure 2: The above figure illustrates the CNN architecture used by us to classify videos as violent / non-violent

We used shallow network architecture for building our Convolutional Neural Network (CNN). Before finalizing on this architecture we experimented with adding more layers, but more layers resulted in drop of training accuracy. This can be accounted to the fact that deeper the layer, more the data requirement. In our case, we had fairly modest data set comprising of roughly 21 thousand images[after data cleaning]. Also deeper networks suffer from vanishing gradient problem. Finally, more the layers more the computations. By taking into account all the above factors and considering the trade offs we decided to stick with just two convolutional layers.

Between every convolutional layer we inserted a max pooling layer which helped in two ways:

- As we used, strides of length 2 and kernel dimensions of 2 x 2, for an input that is of dimensions 64 x 48 will be reduced to 32 x 24. This helps reduce computations significantly.
- It helps to generalize the CNN by letting go of some of the information. If we did not use any maxpooling layer, the training accuracy would increase but testing accuracy would go down.

We used one batch normalizer between the first pooling layer and the second convolutional layer. Batch normalizer makes sure that the input weights and bias to the next layer have 0 mean and unit variance. The primary use of batch normalization is to speed up training process by squashing the range of possible values for weights and bias to a normalized range. This however introduces noise and lowers training accuracy. Normalization can help reduce over-fitting. In our case, our training accuracy was already over 99% and we could do with some normalization to reduce over-fitting. Testing accuracy with and without batch normalization had a difference of about 2% with the model lacking batch normalization having lower accuracy of the two.

Finally, we used dropout layer with a dropout value of about 0.5. Dropout works by randomly switching certain proportion of neurons on and the rest off by multiplying by either 1 or 0. This

process is known to introduce multiplicative noise in training phase. It's again used to combat over-fitting and helps improve testing accuracy.

4.3 Work-flow explanation

Training phase The input images to the CNN were scaled down from 640 x 320 to 64 x 48 pixels. This was required as the unscaled images could not be trained on our hardware and the process ran into *out of memory errors*. The first Convolutional layer used 16 filters of 2 x 2 size. The output of these filters was forced to be kept same as the input by padding the borders before convolution. Output from this convolutional layer was passed through relu activation and into the maxpooling layer which reduced the tensor from 64 x 48 to 32 x 34 size. This tensor was fed into another convolutional layer with 32 filters of 2 x 2 dimensions. Again the output from this layer was fed into another maxpooling layer that reduced the tensor size from 32 x 24 to 16 x 12.

This tensor was given as input to the densely connected layer with 1024 neurons and relu activation at its output. Finally we used two neurons in the output layer with softmax activation.

Testing Phase After the model was trained, saved its progress using checkpoints. Using the tensorflows estimator we reloaded this trained model and passed previously unseen images comprising of both violence and non-violence scene into the trained model.

5 Experiments

There are 5 different versions for this model and the results for the experiments are mentioned here, for versions 1 and 2 the resulting testing accuracy is very bad hence their confusion matrix is not discussed. For version three, the number of epochs was set to 30 and a drop rate of 0.2 was used with no batch normalization which gave a classification rate of 78% on testing data. The true positives for violence data is far fewer than the non-violence positives, with great accuracy achieved for non-violent testing data. The confusion matrix for this version can be found below-

Label	Violence	Non-Violence
Violence	2418	1092
Non-Violence	206	2204

Table 1: Confusion Matrix for Testing Data for version 3

The next version that is 4 gave extremely good results with a classification rate of 82.75% where the number of epochs was equal to 100 with a drop out rate of 0.5 and no batch normalization implemented. The violence data in this case gave extremely good results while the results for non-violence data fell down a bit.The confusion matrix for this version is given below-

Label	Violence	Non-Violence
Violence	2956	554
Non-Violence	467	1943

Table 2: Confusion Matrix for Testing for Data version 4

Version 4 gave satisfactory results for the task at hand but we wanted to experiment with batch normalization and thus implemented that for version 5 giving us the best results thus far.

5.1 Results and discussions

The model with best result was version 5 which gave a classification rate of 84.52% which will move around the surrounding neighborhood of values based on the data sequence selected as a random shuffle is carried out to achieve a more grounded result. The training accuracy cam to about 99% which suggests the model to be over fitting but as the number of data samples are less comparatively over fitting seemed necessary, while using a larger data set overfitting will be unnecessary. The confusion matrix for training data is given below-

Label	Violence	Non-Violence
Violence	8894	0
Non-Violence	10	6998

Table 3: Confusion Matrix for Training Data for version 5

The confusion matrix for testing data is given below-

Label	Violence	Non-Violence
Violence	2858	652
Non-Violence	264	2146

Table 4: Confusion Matrix for Testing Data for version 5

Here, we can say that for the training portion the model over fits with zero false positives for violent data and nominal false negatives for non-violent data. In the case of testing data the results are the best of both version 3 and 4 with good results of version 3 in terms of non-violence data combined with good results for violence data from version 4. The main reason behind this is the use of batch normalization which brought about a great difference in terms of model accuracy.

6 Conclusion and future work

Crowd analysis is an emerging field of computer vision and with increasing amounts of surveillance cameras set up all over the world, detection of crowd behavior using this type of data is very crucial. The task accomplished here surpasses many research projects in the same domain, but as most of this systems are modeled for real-time feedback this results may be not comparable. The idea is to extend these results further and achieve better accuracy by tweaking hyperparameters and experimenting with the network architecture.

In terms of future work, there is scope to expand the model to incorporate functionalities with real-time data with implementations of spatial-temporal features to achieve a more functional system. Also, one other thing that can be done is develop the model into a windows or IOS system for law enforcement departments with real-time machine learning based monitoring of large crowds specifically it would prove very useful for countries with huge populations like India or China. Lastly, future research can be invested in expanding the domain of action detection from crowds and extend it to more diverse actions other than just violence and non-violence detection.

In conclusion, we can positively say that a high precision violence and non-violence system was implemented using deep learning concepts like convolutional neural network on video data. With further advancement and research going about in the field of crowd behavior analysis the system will only get better.

References

O. Boiman and M. Irani. Detecting irregularities in images and in video. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 462–469 Vol. 1, Oct 2005. doi: 10.1109/ICCV.2005.70.

P.A. Crook, V. Kellokumpu, G. Zhao, and M. Pietikainen. Human activity recognition using a dynamic texture based method. In *Proceedings of the British Machine Vision Conference*, pages 88.1–88.10. BMVA Press, 2008. ISBN 1-901725-36-7. doi:10.5244/C.22.88.

P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, Oct 2005. doi: 10.1109/VSPETS.2005.1570899.

Avishai Hendel, Daphna Weinshall, and Shmuel Peleg. *Identifying Surprising Events in Videos Using Bayesian Topic Models*, pages 448–459. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-19318-7. doi: 10.1007/978-3-642-19318-7_35. URL https://doi.org/10.1007/978-3-642-19318-7_35.

J. Liu, Jiebo Luo, and M. Shah. Recognizing realistic actions from videos 'in the wild';. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003, June 2009. doi: 10.1109/CVPR.2009.5206744.

Y. Itcher T. Hassner and O. Kliper-Gross. Violent flows: Real-time detection of violent crowd behavior. In *3rd IEEE International Workshop on Socially Intelligent Surveillance and Monitoring (SISM) at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2012. URL www.openu.ac.il/home/hassner/data/violentflows/.

G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, June 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1110.