Violence Detection in Video with Deep

Github

NIKHITA KANDIKONDA PRANEETH TAMVADA

DATS_6501_10

1. Introduction

Action recognition in video sequences is a challenging problem due to the similarity of visual contents, changes in the viewpoint for the same actions, camera motion with action performer, scale and pose of an actor, and other such factor. A topic of increasing interest in video processing is the characterization of multimedia content regarding the presence of certain human actions. Specially, detection of violent scenes receives considerable attention in surveillance systems. This can be used in various situations like providing people with safer public spaces or detect situations where violence is considered inappropriate for the audience (e.g., children). Generally, human action is a movement of body parts by interacting with objects in the environment. A video is a collection of frames which can be easily understood by human brain but, it requires machine some better techniques to understand the action across frames.

There is a growing interest of learning feature representations from raw data with deep neural networks. Some of the most exciting network structures such as Convolution Neural Network for image classification or image-based object localization. For video classification, these techniques are extended to work on the temporal dimension by stacking frames over time. Despite the recent developments in deep learning field, very few techniques have been proposed to tackle the problem of violence detection from videos. Most of the developed models depend on handcrafted techniques which fail to generalize and require prior understanding. The proposed models do not suffer from these limitations and they can be inputted with raw pixel values without much complex pre-processing. Owing to these reasons, we choose to develop a deep neural network for performing violent video recognition.

Our contributions can be summarized as follows:

- We develop various an end-to-end trainable deep neural network model for performing video classification for violence.
- We use transfer learning techniques to understand and extract features.
- We compare 3 different models and show that a recurrent neural network capable of encoding localized spatio-temporal changes is capable to better detect violence in videos.
- We validate the effectiveness of the proposed method using three widely used benchmarks for violent video classification.

2. Related Works

The earlier work was based on hand-crafted features with some actions in a scene with simple background. The low-level features from the video data are extracted and then passed to a

classifier such as support vector machine (SVM), decision tree, and KNN for action recognition. Recent research includes methods that use the visual content, audio content or both. In this section, we will be concentrating on methods that use the visual cues alone since audio data is generally unavailable with surveillance videos or is not very clear. Existing techniques can be classified into Inter-frame changes [28, 5, 4, 8] and Local motion in videos [6, 3, 7]. Recently models using long short term memory (LSTM) RNNs [16] have been developed for addressing problems involving sequences such as machine translation, speech recognition, caption generation and video action recognition. A multi-resolution CNN framework for connectivity of features in time domain is proposed by [21] to capture local spatio-temporal information. A two-stream CNN architecture is proposed by [22] in which first stream captures spatial and temporal information between frames and second one demonstrates the dense optical flow of multiple frames. The feature maps of pre-trained model are analyzed by Bilen et al. [23] for video representation named as dynamic image.

3. Proposed Models

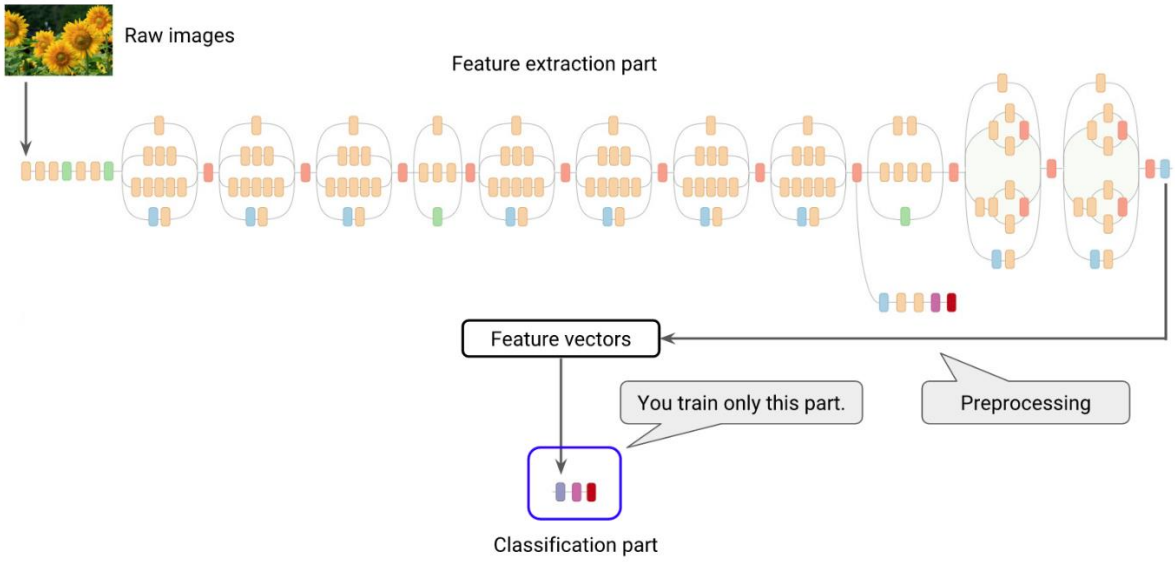
The major goal is to develop an end-to-end trainable deep neural network for classifying videos into violent and non-violent. In this section we introduce the proposed models in detail. A video can be classified into spatial and temporal components. Spatial component describes individual frame appearance, information about scenes and objects in the video. Whereas temporal component describes motion across the frames, conveys the movement of the objects and the camera or observer. We have hence build two models that classify based on spatial component and one model that classifies based on temporal component.

3.1 Spatial Modelling

Conventional CNN architectures take images as the inputs and contain alternating convolutional and pooling layers and a few dropout layers for regularization which are further topped by a few fully-connected (FC) layers. Convolution Neural Networks have well evolved in the past few years and we have state of the art networks which have been trained on large datasets and have well generalized to detect edges and other important features in images. We use the technique of transfer learning to utilize one such powerful image recognition framework, Inception and extract the feature vectors from the last pooling layer. Videos are a collection of multiple frames. To standardize videos from different source, a fixed number of frames are chosen from the video and are fed into Inception model which returns a feature vector for each image. These feature vectors are used in the models below for spatial learning. Figure 1 illustrates this transfer learning model.

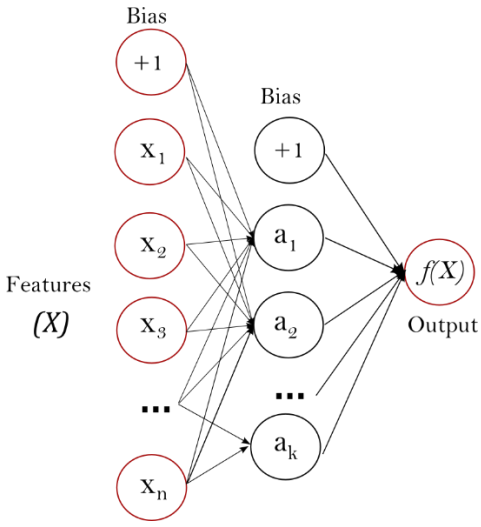
3.1.1 CNN with MLP (ConvMLP)

The features extracted from inception model for N frames are flattened to get a 40 vectors of 2048 dimension. Though this model ignores the sequence of the video, the hypothesis is, MLP will be able to infer the temporal features from the sequence, without having to know it's a sequence. Figure 2 demonstrates the MLP network. The flattened vectors are fed into MLP as individual input vectors and after trying different layers and activation functions, a two layer MLP with 512 nodes in each layer and activation function ReLu performs the best.



Source: https://codelabs.developers.google.com/codelabs/cpd102-txf-learning

Figure 1



Source: http://scikit-learn.org/stable/modules/neural_networks_supervised.html

Figure 2

3.1.2 CNN with LSTM (ConvLSTM)

Video is a sequence data of multiple frames. To model this sequence, we use a special type of Recurrent Neural Network (RNN) called Long Short Term Memory (LSTM) model which avoids vanishing gradient problem traditional RNNs suffer from. This network is capable of learning long term dependencies with input, output, and forget gates that control the long-term sequence pattern identification. Dur8ing training, the sigmoid units learn where to open and close the gates.

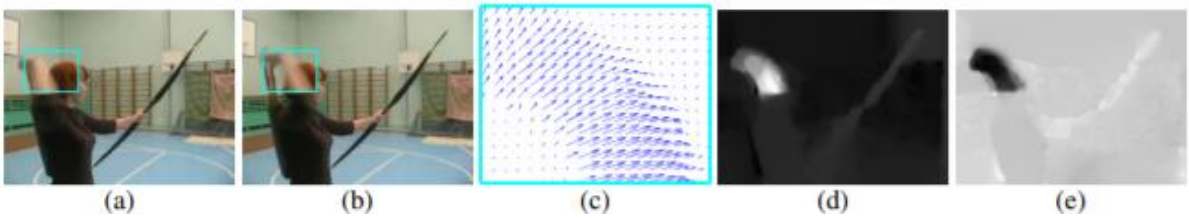
Eq. 1 to Eq. 7 explain the operations in LSTM unit, where the input $is\ x_t$ at time t, f_t is the forget gate at time t, which keeps a knowledge of the previous frame whose information needs to be cleared further in the network. The output gate o_t keeps information about the upcoming step. The input of the current frame and state of the previous frame s_{t-1} are combines with an activation function of 'tanh' which forms the recurrent unit g. Tanh activation and memory cell c_t are used to calculate hidden state of a RNN. As the action recognition does not need the intermediate output of the LSTM, we made final decision by applying softmax classifier on the final state of the RNN network.

$$i_t = \sigma((x_t + s_{t-1})W^i + b_i) \quad (1)$$
$$f_t = \sigma((x_t + s_{t-1})W^f + b_f) \quad (2)$$
$$o_t = \sigma((x_t + s_{t-1})W^o + b_o) \quad (3)$$
$$g = \tanh((x_t + s_{t-1})W^g + b_g) \quad (4)$$
$$c_t = c_{t-1} \cdot f_t + g \cdot i_t \quad (5)$$
$$s_t = \tanh(c_t) \cdot o_t \quad (6)$$
$$final\_state = soft\max(Vs_t) \quad (7)$$

A two-layer LSTM with2048 and 512 units respectively are used to train the model.

3.2 Temporal Modelling with optical flow data

This model is used to develop a ConvNet model which recognizes the temporal component of the videos. The input if the model is formed by stacking optical flow displacement fields between several consecutive frames. The input is capable of explicitly describing the motion in video frames which makes it easier as the model does not have to implicitly estimate the motion.



Source : http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

Figure 3

From figure1, we can see a dense optical flow that can be seen as a set of displacement vectors between any two pairs of consecutive frames at t and t+1 times. $d_t(u, v)$ denotes the displacement vector at a point (u,v) in frame t. d_t^x and d_t^y are the horizontal and vertical components of he vector fields. Stacked flow channels $d_t^{x,y}$ represents motion across a sequence of frames of L consecutive frames which results in 2L input channels. In this implementation we have used optical flow for a sparse feature set using the iterative Lucas-Kanade method with pyramids.

In this method, first we identify well-textured features within the target region. Once we identify these features we then calculate their optical flow using a two-frame gradient-based method developed by Lucas and Kanade. This methods calculated the vector of displacement for individual features rather than tracking them within the entire frame.

Architecture: Above we have described the ways of combining multiple optical flow displacement fields into a single volume $I_t \in$ R w×h×2L. As we know a ConvNet requires a fixed-size input, we crop input frames if a video to (244,244) and generate an optical flow of dimension (244,244,2L) to pass it to the net as input.

4. Datasets

The performance of all the proposed method are evaluated on three standard public datasets namely, Hockey Fight Dataset[1], Movies Dataset [1] and Violent-Flows Crowd Violence Dataset [2]. These videos are captured using various sources like mobile phones, CCTV cameras and high-resolution video cameras.

- Hockey Fights Dataset [1]: This dataset is created by collecting videos of ice hockey matches and contains 500 fighting and non-fighting videos.
- Movies Dataset [1]: These videos are based on the extraction of violent events in movies. The non-fight sequences are collected from other publicly available action recognition datasets and contains 100 violent and non-violent videos.
- Violent-Flows Dataset [2]: This is a crowd violence dataset with a number of people taking part in the violent events. There are 246 videos in this dataset.

We have used various datasets to scale and make our model robust to detect different violent scenes.

5. Data Preprocessing

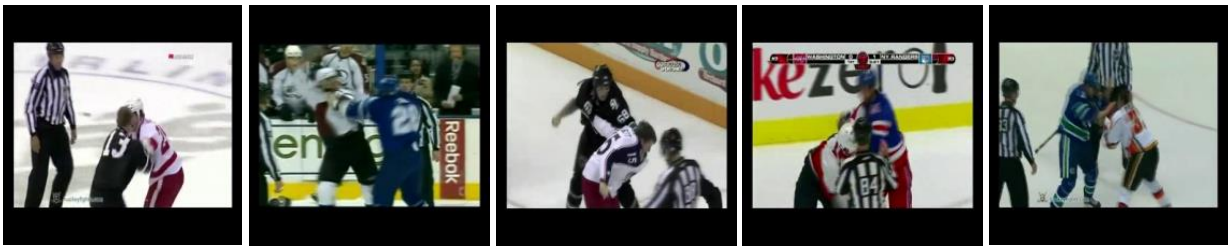
We are using video data from different sources and hence we must pre-process the data and standardize videos from all sources. We used OpenCV to extract frames from video and resized each frame to 224*244. Every video defers in the number of frames and hence, 40 frames are chosen by skipping equidistant frames. These frames are saved as images for CNN transfer learning and numpy array for optical flow leaning.

First, we used transfer learning methods to extract feature vectors from Inception model as we have small dataset and constraint on computational complexities for spatial classification then secondly, we calculated optical flow for temporal classification using OpenCV. Keras framework is used for transfer leaning and training the models.

6. Experimental Evaluation

In this section, the proposed technique is experimentally evaluated, and the results are discussed. The combination of 3 datasets was split into train and test and validation set. A few sample images from each data set are shown in figure 4. The networks are trained using RMSprop algorithm with a learning rate of 0.0001 and a batch size of 32. Early stopping technique was used to stop training the model where the validation loss does not improve for 5 iterations. To optimize the training of models, a generator is created which yields a batch of data and helps save memory required to train the model. Spatial models are run for 100 iterations. Temporal model is run for 5 iterations due to computational limitations. Table 1 is a summary of test accuracies of each model and figures 5-7 demonstrate the plots of train and validation accuracies and loss curves. Due to early stopping, ConvMLP

Hockey Dataset

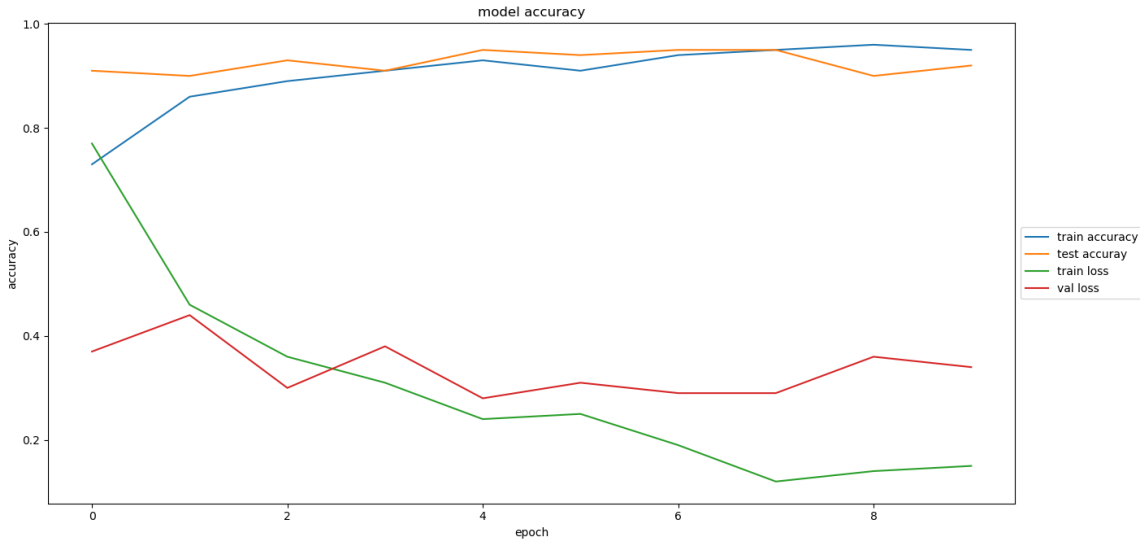


Movies Dataset

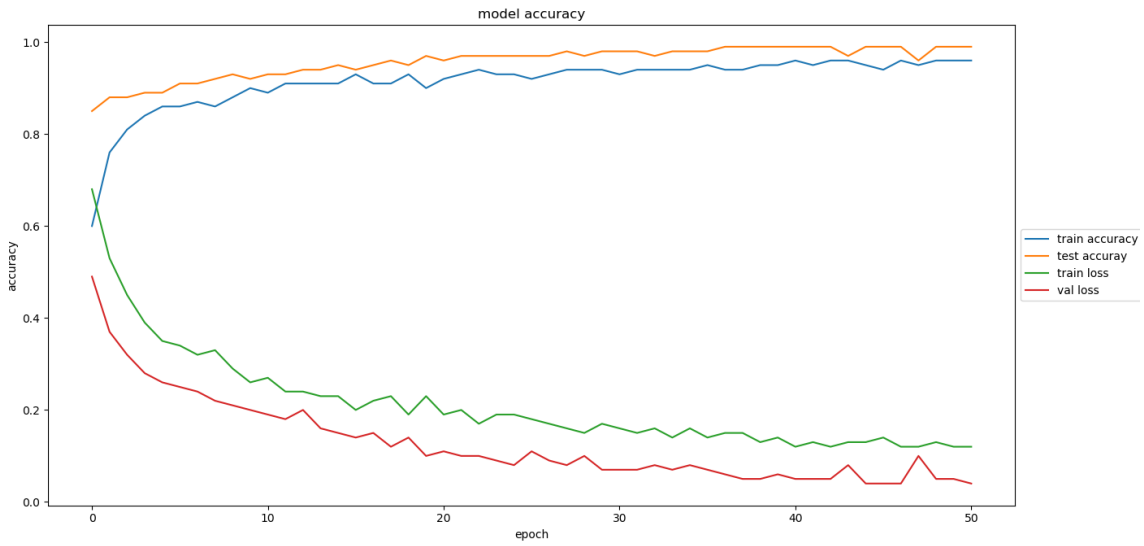
Violent Flow Dataset

Figure 4

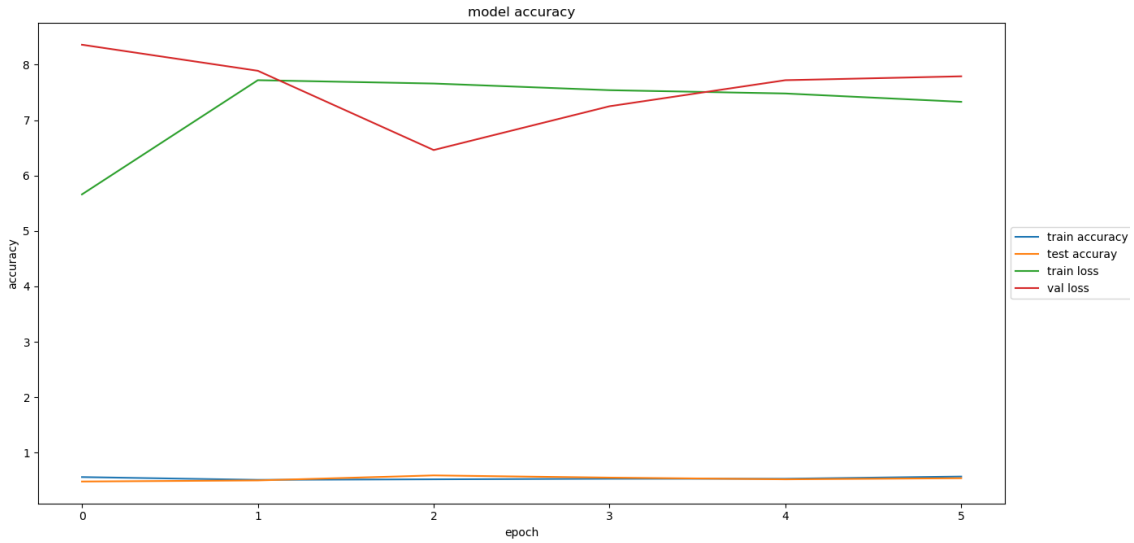
Accuracy and loss plots for ConvMLP model



Accuracy and loss plots for ConvLSTM model



Accuracy and loss plots for Temporal Optical Flow model



Model	Test Accuracy
ConvMLP	92.4
ConvLSTM	96.08`
Temporial Optical Flow	54.3

Table 1

In this analysis we consider aggressive behavior as violent. In the hockey dataset, the fight videos consist of players colliding and hitting each another. So, we can check if one player moves closer to another and classify as violent scene. But the non-violent videos also have players hugging each other which is not violent but players are moving close to each other and these videos could be mistaken as violent. The proposed method can avoid this as it encodes motion of localized regions (motion of limbs, reaction of involved persons, etc.). But this model does not work well on Violence Flow dataset because in most of the violent videos, only a part of the crowd is involved in aggressive behavior and the rest are spectators. The network hence classifies such videos as non-violent.

7. Conclusion

This paper presents three end to end trainable models to detect Violence in videos. The proposed models consist of two feature extraction processes, one with transfer learning through Inception and the other using Optical Flow of the Video. The proposed methods are evaluated on three different datasets and resulted in improved performance due to variety od datasets. The results show that the ConvLSTM model is capable of generating a better video representation compared to LSTM with less number of parameters, thereby avoiding overfitting. With greater computational capabilities, we can test temporal model and compare the results to spatial models.

8. Individual Contributions

Task	Contributor
Data Preprocessing and Model selection	Praneeth and Nikhita
ConvMLP model and Temporal Optical flow model	Praneeth
ConvLSTM	Nikhita

References

[1] E. Bermejo, O. Deniz, G. Bueno, R. Sukthankar. Violence Detection in Video using Computer Vision Techniques. Proceedings of Computer Analysis of Images and Patterns, 2011.

[2] C.H. Demarty,C. Penet, M. Soleymani, G. Gravier. VSD, a public dataset for the detection of violent scenes in movies: design, annotation, analysis and evaluation. In Multimedia Tools and Applications, May 2014. (pdf)

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, ‘‘Large-scale video classification with convolutional neural networks,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1725–1732.

[4] K. Simonyan and A. Zisserman, ‘‘Two-stream convolutional networks for action recognition in videos,’’ in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 568–576.

[5] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, ‘‘Dynamic image networks for action recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 3034–3042.

[28] N. Vasconcelos and A. Lippman. Towards semantically meaningful feature spaces for the characterization of video content. In ICIP, 1997.

[4] L.-H. Chen, H.-W. Hsu, L.-Y. Wang, and C.-W. Su. Violence detection in movies. In International Conference on Computer Graphics, Imaging and Visualization (CGIV), 2011. 2

[5] C. Clarin, J. Dionisio, and M. Echavez. Dove: Detection of movie violence using motion intensity analysis on skin and blood. Technical report, University of the Philippines, 01 2005

[8] O. Deniz, I. Serrano, G. Bueno, and T.-K. Kim. Fast violence detection in video. In International Conference on Computer Vision Theory and Applications (VISAPP), 2014

[6] A. Datta, M. Shah, and N. D. V. Lobo. Person-on-person violence detection in video data. In ICPR, 2002.

[7] F. D. De Souza, G. C. Chavez, E. A. do Valle Jr, and A. d. A. Ara´ujo. Violence detection in video using spatio-temporal features. In Conference on Graphics, Patterns and Images (SIBGRAPI), 2010.

[3] D. Chen, H. Wactlar, M.-y. Chen, C. Gao, A. Bharucha, and A. Hauptmann. Recognition of aggressive human behavior using binary local motion descriptors. In International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 2008

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[21] E. B. Nievas, O. D. Suarez, G. B. Garc´ıa, and R. Sukthankar. Violence detection in video using computer vision techniques. In International Conference on Computer Analysis of Images and Patterns. Springer, 2011.

[22] V. P˘atr˘aucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In ICLR Workshop, 2016.

[23] S. Pfeiffer, S. Fischer, and W. Effelsberg. Automatic audio content analysis. In ACM International Conference on Multimedia, 1997.