Video Classification using Machine Learning

Shaunak Deshpande[1], Ankur Kumar[2], Abhishek Vastrad[3], Prof. Pankaj Kunekar[4]

[1,2,3]Student, Department of Information Technology, Atharva college of Engineering, Mumbai, Maharashtra India.
[4]Professor, Department of Information Technology, Atharva college of Engineering, Mumbai, Maharashtra India.

Abstract - *Society is rapidly accepting the utilization of cameras in an exceedingly large choice of locations and applications: Live traffic monitoring, automobile parking space surveillance, inside vehicles and intelligent spaces. These cameras offer data on a commonplace that require to be analyzed in an efficient manner. Unfortunately, most visual surveillance still depends on a operator to sift through this video. We will analyze any footage using the assistance of machine learning and classify the footage into two classes which are Dangerous activity and Safe Activity. The developed system makes use of the successful techniques like Deep Learning. Particularly, convolutional neural networks are accustomed detect features within the video images, meanwhile Recurrent Neural Networks are accustomed analyze these features and predict the possible activity within the video.*

***Key Words*: Convolution Neural Network, Recurrent Neural Network, Long Short-Term Memory, Human Behaviour, Inception v3**

1.INTRODUCTION

Human Behavior Analysis (HBA) is a significant field of focus in artificial intelligence. It has many fields of use such as video monitoring, environment-assisted living, smart shopping environments, etc. The availability of human video data is increasing dramatically, with the help of leading companies in this area. From the perspective of DL this job approaches HBA. Owing to the growth of computing capacity, Deep Learning techniques have been a great step in the classification context in the last few years. Strategies used are Convolutional Neural Networks (CNNs) for image comprehension, and RNNs for temporary comprehension like video or text. The following sections discusses them in detail.

1.1 Deep Learning

Deep learning is a machine learning branch that uses advanced, multi-level "deep" neural networks to build systems capable of detecting features from vast quantities of unlabeled training data. Due to the better performance against conventional methods such as decision trees, support vector machines, Bayesian networks, etc., there is a growing interest in the last decade for the use of deep learning. Over the past few years, the increase in computing capacity has made it possible to consider the biologically inspired algorithms developed decades ago.

1.2 Convolutional Neural Network

Convolutional Neural Networks are a particular form of Artificial Neural Networks that are designed to process a large number of input data (images, audio, or video). Due to the large amount of input data it would be very inefficient to remove the function with a normal Fully Connected (FC) network. In a broad sense, what CNN does is to minimize the details by looking at the individual regions of the data in order to extract specific features. CNNs are based on filters (kernels) that function like the weights of the Completely Connected ANN. The only difference with the weights of the FC is that a single convolutional filter is spread across all the input regions to produce one output. It is called Local Receptive Fields, and reduction of the amount of weights CNN can know is very beneficial. The way output is measured is by sliding the filter over the input. The product between each kernel element and the input element that it overlaps is computed at each position. All the items are then summed up to get the production at the current spot.

Raising the size of the feature maps is a common technique for reducing the number of parameters and the amount of computation in a CNN. The pooling layer works independently (normally following a convolutional layer) and uses the maximum or average role in the feature map to reduce the regions. A very successful method for detecting features is the combination of many convolutional and pooling layers in parallel. It is because various kernel sizes can be implemented in parallel, allowing basic and complex functionality to be identified at different layers of the network. The first network to implement the parallelism was AlexNet (2012). Google

Inception Network is a highly accurate Deep Neural Network for image recognition. This incorporates multiple layers in a module called the "Inception Module."

1.3 Recurrent Neural Network

The above portrayed techniques are intended to arrange autonomous information, however what happens when we manage time-arrangement information? To handle with these necessities, another sort of neural system was intended to demonstrate time succession information. These systems are called Recurrent Neural Systems (RNNs) and permits the data to persist, by having loops in it .In this we get a streaming input x_t, and thus a streaming output o_t, in every iteration, the output o_t, are another input for the subsequent iteration. Basic RNNs are useful to model small temporal dependencies. When coping with long sequences of information (in most real cases) a replacement variety of RNN called Long Short-Term Memory Networks are used.

1.4 LSTM Networks

LSTM networks, introduced by Hochreiter & Schmidhuber (1997), are RNN-based models capable of learning long-term dependencies. Vanilla RNNs have a awfully simple structure, like one perception with a singular tanh activation layer. In contrast, LSTM cells have a more complex structure, where rather than having one layer (tanh), there are four, interacting in an exceedingly very special way.

Video analysis provides more information for the task of recognition by incorporating a temporal dimension from which additional use can be made of motion and other information. At the same time, the task becomes much more computationally challenging for processing short video clips, as each video may contain hundreds to thousands of frames, not all of which are helpful. A naive solution would be to view video frames as still images, and to apply CNNs in order to identify each frame and average video level predictions. Nonetheless, because each individual video frame forms only a small part of the story of the video, such an approach would be using incomplete information and thus could easily confuse groups, particularly if fine-grained distinctions or portions of the video are unrelated to the interesting action.

We hypothesize therefore that learning a global explanation of the temporal evolution of the video is necessary for accurate classification of data. From a modeling perspective this is difficult as we have to model variable length videos with a fixed number of parameters.

The main purpose of this project is to design and implement an efficient deep learning solution able to predict and classify human behavior into two categories which are Safe Activity and Dangerous Activity by using combination of CNNs and RNNs architectures.

2. RELATED WORK

Last decade was a prominent time for researchers to present many hand-crafted and deep-net-based methods for action recognition. Earlier works were supported hand-crafted features for non-realistic actions videos. Since the proposed method is predicated on deep neural network (DNN), during this section, we are going to only review related works supported DNN. In recent years, different variants of deep learning models are proposed for human action recognition in videos and have achieved great performance for computer vision tasks. Ji et al. [4] applied 3D convolutional kernels on video frames in a very time axis to capture both spatial and temporal information. Karpathy et al. [5] directly tried to apply CNNs to multiple frames in each sequence and obtained the temporal relations by pooling, using single, late, early and slow fusion; however, the results of this scheme were just marginally better than those of one frame baseline. Simonyan and Zisserman [6] used a two-stream CNN framework where both feature types were included, with one stream taking RGB image frames because the input and also the other taking pre-computed stacked optical flows. Since optical flow contains only short-term motion information, adding it doesn't enable CNNs to find out long-term motion transitions. The extra stream significantly improved action recognition accuracy, indicating the importance of motion features. Tran et al. [21] avoided the necessity for pre-computing optical flow features through their 3D convolution (C3D) framework, which allows deep networks to find out temporal features in an end-to-end manner. However, C3D only covers a brief range of the sequence. Wang et al. [7] introduced a temporal segment network (TSN) architecture, where a sparse temporal sampling strategy is adopted to model long-term temporal structures. In [8], Feichtenhofer et al. study variety of how of fusing CNN towers so as to require advantage of this spatial-temporal information from the looks and optical flow networks. The CNN-based method only extracts visual appearance features and lacks the long-range temporal modeling capabilities, CNN-based method ignores the intrinsic difference between spatial and temporal domains as well.

Some researchers have presented methods by uniting the advantages of both hand-crafted and deeply learned elements, such as [9,10], and obtained good results. They combine the key factors of two positive video representations, namely improved trajectories[1] and ConvNets[8] in two streams. How to combine the advantages of these two types of functions to build good descriptors was an active research area. Some research attempts were made using depth imaging, such as dynamic images and maps of depth. The dynamic image network was introduced by Bilen et al.[2] to produce dynamic images for action videos. The order of video frames is used as the information for supervision; however, this approach lacks some knowledge regarding discrimination. Taylor et al.[11] used a Boltzmann constrained convolutional gated system to produce a flow field of the two adjacent frames in the video for action recognition, but this model could not produce a single map to represent a video. Rank pooling[12] and Fisher Vector[13] attempted to produce a motion map for the duration of desire. Those methods, however, are unable to model temporal dynamics between video frames. RNNs have been considered for video-based HAR RNN networks to offer strength to find and process hidden patterns in time-space data in order to model the temporal dynamics among video frames. Data is processed sequentially so that it gets input from the previous hidden state s_t-1 at each time t and gets new data x_t. By leveraging CNNs and RNNs for action recognition, most of the state-of-the-art methods[14–19] have proposed their own recurring networks, and have achieved amazing results. Nonetheless, there were vanishing gradient problems due to the large number of parameter calculations and neglect of initial input effect after a few layers. The solution to this problem is LSTM [15,17,22], which is capable of capturing long-term dependencies and maintaining knowledge of sequences over time by combining memory units. LSTM was initially introduced by [20]; It has been successfully adapted to a variety of sequential modeling projects, such as speech recognition, visual description and machine translation, and improved efficiency achieved. The inputs to the LSTM are the high-level features captured from a fully-connected CNN layer in most of those networks. Multiplicative gates are used by LSTM units to monitor access to the network propagating error signal.

Combination of CNN with RNN provides an effective representation for long-term motion and modeling of the sequential data, each of which incorporates a time relationship with adjacent points (RNN uses the extracted CNN features as inputs and models more robust, longer-range features.) The CNN network is ready to encode local temporal features within each video unit; it cannot model across the multiple units of a video series. We introduce LSTM to capture global sequence dependencies of the input video and cues on motion information. Fused spatio-temporal features are processed by Long short-term memory, which helps recognizing complex frame-to-frame hidden sequential patterns. After conducting extensive experiments, we observed that our method is incredibly effective for videos of assorted lengths and shows significant improvement in action recognition.

3. METHODOLOGY

3.1 ACTIVITY RECOGNITION SYSTEM

Upon evaluating the state-of-the-art of various behavior recognition systems, identifying the methodology of the tools we will be using, and the expensive process of downloading the two data sets, we begin to introduce our proposal. This system is a combination of two separate DL models, a CNN reads the video frames and extracts the features, and RNN reads those features, predicting the operation. This DL model is scheduled in Python, using the Keras framework (using the Tensorflow framework as backend).

Before proceeding with this training step, the data should be pre-processed so that the DL model fits properly.

3.2 MODEL

The state of the art of deep learning activity recognition suggests that the best way to tackle this problem is through a model with a Convolutional Neural Network, at the beginning, to extract the features of the video frames, followed by a Recurrent Neural Network that can model frame sequences. There are other Behavior Recognition DL models including a 3D CNN that uses an FC network. In this method, the whole video is fed to the 3D CNN at once, and this CNN is capable of extracting not only image characteristics but also motion or time characteristics. All these features are then fed into a network of vanilla FCs. The problem with this method is that predicting the operation requires all the footage.

Nevertheless, the activity can be predicted before the video has finished, and this method is better because it can predict the activity in real time (early prediction).

3.3 CONVOLUTION PART

It's not an easy task to achieve a 2D convolutional neural network with good performance in understanding images and generating its features (vector that summarizes an image's information). This is due to the difficulty of finding a good model, and the immense amount of time and data required for training. As a consequence, a common method in deep learning is to incorporate a pre-trained model to extract the features, and then then move the features on to the new model. There are several models that are pre-trained to recognize pictures. ImageNet is a database which has been organizing an annual challenge (ILSVRC) since 2010, testing object detection and image classification algorithms. Out of this rivalry several DL models have emerged since 2012: Alex. Net (2012), ZF Net (2013), VGG Net (2014), GoogLeNet (2014), Microsoft ResNet (2015). All these deep learning models (since 2012) have two core blocks in common: the image feature extractor using convolutions, the number and structure of the model-determined convolution layers. The second block is connected to the classification process, which in this case will be a feedforward neural network that takes a feature vector as its input and classifies the object type (the output size depends on the number of objects to be classified). This second element is similar to any (ILSVRC)challenge pattern. For this project, we will use the extraction feature part of a pre-trained model called transfer learning which focuses on storing knowledge and applying it to a specific but related question. The model we will be using is Inception v3 because it has excellent classification accuracy and low cost of computation. There are other models that achieve better performance like Inception ResNet v2 but they do have layers that require more computation and the increase in classification is not that much. One of the available models within the Keras system is the Inception v3 model (Figure 6). The manner in which initiation works is as follows. Instead of making a pyramid of convolutions (one behind another), Inception has, what they call modules of inception, groups of layers where the flow is not sequential. Several convolutions of various sizes are computed separately in these modules, and then concatenated into one row. This process allows extracting additional functionality. It also takes advantage of the 1 x 1 convolutions to reduce operations. The classification component is the second part of the Inception network, generated by a fully connected layer and a softmax output layer. This classification method is appropriate if we only need to classify an image at once, but if we need to classify an image stream, such as a picture, then we need RNNs.

3.4 RECURRENT PART

A Recurrent Neural Network is the best method, according to the state of the art, of deep learning to identify a sequence of inputs such as text, speech or video recognition. RNNs that can model data sequences via internal loops that provide input to the network. Long Short-Term Memory Networks are a form of RNNs that can "remember" important parts of the input sequence, regardless of the time it shows up (simple RNNs only remember recent parts of the sequence, they have short-term memories). In this model, it is suggested that an LSTM network adopt the function part of the network Inception. The size of the feature vector returned by the Inception v3 network is 2048, so it is suggested that an LSTM layer of the same size recall every single feature of the vector series (each LSTM cell will be fed with one feature). A Completely Connected 512 Neuron layer is attached after the LSTM layer.

4. IMPLEMENTATION

The proposed framework incorporate the utilization of two neural networks. They are Inception-v3 and LSTM.

Steps which are required are as follows:

Stage 1: Dataset building

- Videos of classifications dangerous activity and safe activity are gathered from the web from sources, for example, Youtube, Dailymotion, and so on.
- The videos then require to be labeled.
- Since there are only two classes, they would be represented as 0 and 1.
- The video files belonging to the above classes are renamed as 0_1, 0_2, 1_1, 1_2 etc.
- The video file for instance 0_1, here '0' represents the class dangerous activity, '1' represents the video file number for the dangerous activity class and '_' is used to separate those values.

Stage 2: Training

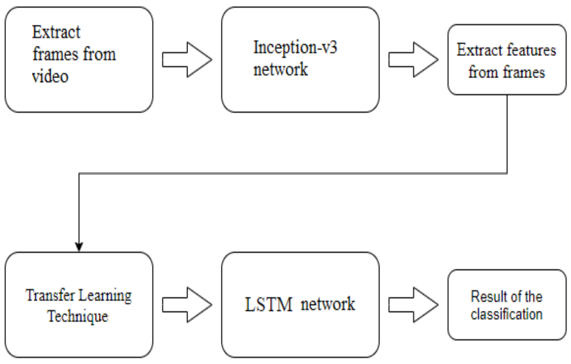
- 75 % of the dataset is utilized for training the model and 25% of the dataset is utilized for testing the model.
- Videos are converted into frames.
- Label for each video is extracted and stored in an array
- For each frame, image pixels are converted into numpy array which are then reshaped for the inceptionV3 model.
- The Frame is then preprocessed and features are extracted from the frame using the inception-V3 model.
- Using Transfer Learning technique features extracted from frames in a sequence with their respective labels is used as an input for the LSTM model.
- The LSTM model contains 2 hidden layers with relu and sigmoid activation respectively and output layer with 2 neurons with softmax activation to generate the classification for the video.

Stage 3: Testing

- Testing is performed on a single video
- Frames are extracted from the video

- For each frame, image pixels are converted into numpy array which are then reshaped for the inception-V3 model.
- The Frame is then preprocessed and features are extracted from the frame using inception-V3 model.
- The features are then stored in an array which is then converted in numpy array and then reshaped in the input shape for LSTM.
- Using the predict_class method of the LSTM model the classification of the video file is produced.

Figure -1: Block Diagram


Extract frames from video
Inception-v3 network
Extract features from frames
Transfer Learning Technique
LSTM network
Result of the classification

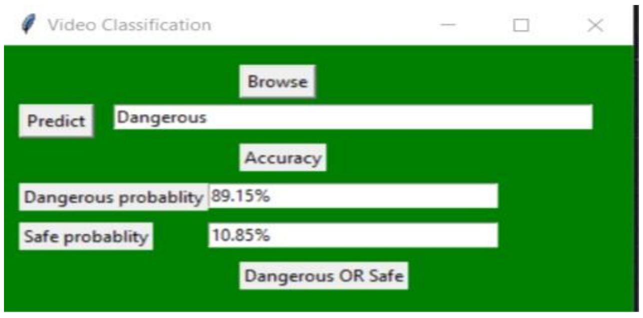
5. RESULTS

This section analyzes result obtained for the labeled dataset constructed in the implementation phase.

Evaluating the Results

Using the user interface video file is selected which needs to be classified. The following figure 2 shows the prediction result when the input video is of a burglary in a store, which is classified by our system as Dangerous activity. Footage of an interview is categorized into Safe Activity in Figure 3.

Figure -2: Dangerous Activity

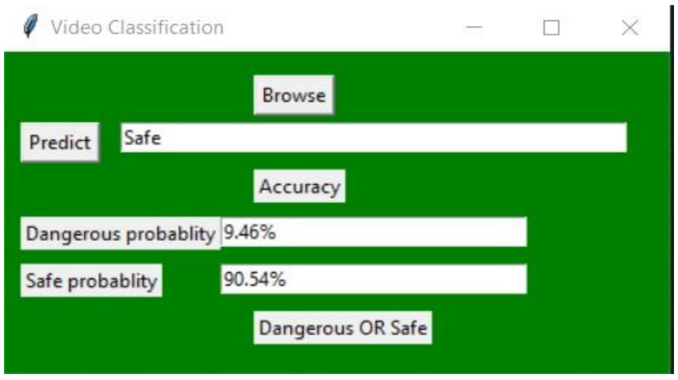


Dangerous

Dangerous probablity 89.15%

Safe probablity 10.85%

Figure -3: Safe Activity



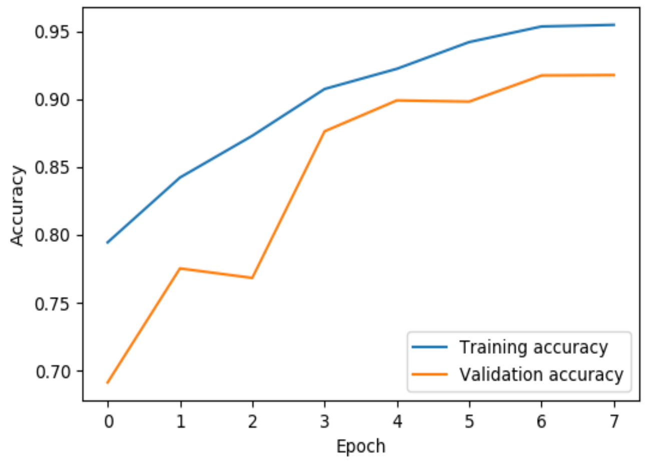
Safe

Dangerous probablity 9.46%

Safe probablity 90.54%

The fit() method which is used to train the model returns a History object. The history attribute of the History object is a dictionary that contains successive metrics and loss values.
Figure 4 shows the training and validation accuracy obtained while training the model at successive epoch.

Figure -4: Training and Validation accuracy of the model.



The training and validation accuracy obtained at last epoch are approximately 95 and 92 percent respectively.

6. CONCLUSIONS

The classification of video is troublesome because of numerous reasons, for example, shortage of video dataset, low accuracy etc. The main highlights of this paper are, data manipulation of distinct datasets to fit a Deep Learning model; transfer learning of a pretrained Deep Learning model (Inception V3) to our system; use of LSTM Recurrent Neural Networks. Since all day, every day manual checking of recordings is troublesome this framework could supplant such convention and analyze the footage with maximum accuracy. Video based online search, security surveillance, recognizing and removal of reuploaded copyright video are portions of its future scope.

As future work we aim to increase the system accuracy by taking advantage of the diversity that datasets such Activitynet offers to obtain an even more robust activity recognition system. A better model could be implemented by improving the fine tuning process, varying the number of layers, neurons, learning rate, etc.

REFERENCES

[1] Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 3551–3558.

[2] Bilen, H.; Fernando, B.; Gavves, E.; Vedaldi, A.; Gould, S. Dynamic image networks for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3034–3042.

[3] Chen, C.; Liu, K.; Kehtarnavaz, N. Real-time human action recognition based on depth motion maps. J. Real-Time Image Process. 2016, 12, 155–163. [CrossRef]

[4] Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35, 221–231. [CrossRef] [PubMed]

[5] Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Li, F.F. Large-scale video classification with convolutional neural networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.

[6] Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. arXiv 2014, arXiv:1406.2199.

[7] Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. arXiv 2016, arXiv:1608.00859.

[8] Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.

[9] Wang, L.; Qiao, Y.; Tang, X. Action recognition with trajectory-pooled deep-convolutional descriptors. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.

[10] Lu, X.; Yao, H.; Zhao, S. Action recognition with multi-scale trajectory-pooled 3D convolutional descriptors. Trans. Multimedia Tools Appl. 2017, 1–17. [CrossRef]

[11] Taylor, G.; Fergus, R.; LeCun, Y.; Bregler, C. Convolutional learning of spatiotemporal features. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 140–153.

[12] Fernando, B.; Gavves, E.; Oramas, J.M.; Ghodrati, A.; Tuytelaars, T. Modeling video evolution for action recognition. In Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5378–5387.

[13] Perronnin, F.; S´anchez, J.; Mensink, T. Improving the fisher kernel for large-scale image classification. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 143–156.

[14] Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. arXiv 2014, arXiv:1409.2329.

[15] Donahue, J.; Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 677–691. [CrossRef] [PubMed]

[16] Veeriah, V.; Zhuang, N.; Qi, G.J. Differential recurrent neural networks for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Santiago, Chile, 7–13 December 2015; pp. 4041–4049.

[17] Yue-Hei, J.; Hausknecht, M.; Vijayanarasimhan, S. Beyond short snippets: Deep networks for video classification. In Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4694–4702.

[18] Wu, Z.; Wang, X.; Jiang, Y. Modelling spatial-temporal clues in a hybrid deep learning framework for video classification. In Proceedings of the 23rd ACM

international conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 461–470.

[19] Ji-Hae, K.; Gwang-soo, H.; Byung-Gyu, K.; Debi, D. deepGesture: Deep learning-based gesture recognition scheme using motion sensors. Displays 2018, 55, 38–45.

[20] Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780. [CrossRef] [PubMed]

[21] Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. arXiv 2015, arXiv:1412.0767.

[22] Srivastava, N.; Mansimov, E.; Salakhutdinov, R. Unsupervised Learning of Video Representations using LSTMs. arXiv 2015, arXiv:1502.04681.