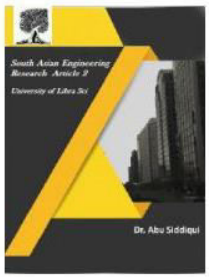
2581-4575

International Journal For Recent Developments in Science & Technology

Crossref

A Peer Reviewed Research Journal



ACTIVITY DETECTION IN UNCONSTRAINED VIDEOS USING CONVOLUTION NEURAL NETWORK AND BI-LSTM

B.H.DASARADHA RAM[1], M.VENKAT SAI[2], SK.CHANDINI[3], P.KAVYA SRI[4]

1 Associate Professor, NRI Institute of Technology, 2, 3, 4 Scholars, NRI Institute of Technology

Abstract:

Nowadays as computation power increases as increasing with the amount of data to process. As most of the data is generated every day is multimedia data. In many different media types video is one of them. Where finding what happens over a video as it streams or recorded using a computerized method is more helpful in CCTV surveillance. We're attempting to continually detect activities in the video as it's streamed, in an online system. As videos are sequential frames, we use CNN we extract a buffer length of features for a buffer length of video frames. These buffers then used to train Bi-LSTM, and so this Bi-LSTM model and CNN models are accustomed to detect particular activities in unconstrained or streaming videos using multiprocessing for speeding up the feature extraction.

Keywords: Activity Detection, unconstrained video, Bidirectional LSTM, CNN Features

Introduction:

Images and videos became ubiquitous on the web, which has encouraged the event algorithms that will analyze their semantic content for various applications, including search and summarization.CNN's are shown to search out powerful and interpretable image features, where the networks have access to not only the looks information present in single, static images but also their complex temporal evolution. There are several challenges to extending and applying CNN's during this setting. One of the key motivations, which attracts researchers to work in action recognition, is that the vast domain of its applications in surveillance videos, robotics, human-computer interaction, sports analysis, video games for the characters of the players, and management of web videos.

Action recognition using video analysis is computationally expensive as processing a short video may take protracted time because of its high frame rate.

Related Work:

Over the last decade, researchers have presented many hand-crafted and deep nets based on the approaches for action recognition. The sooner work was supported by hand-crafted features for non-realistic actions, where an actor accustomed to perform some actions during a scene with an easy background. Such systems extract low-level features from the video data, and so feed them to a classifier like a support vector machine (SVM), decision tree, and KNN for action recognition. g-based methods were also proposed in recent years. Deep learning has shown Besides hand-crafted features based approaches for action recognition, several deep learning significant improvement in many areas like to image classification, person re-identification, object detection, speech recognition, and

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

bioinformatics. As an example, an undemanding implementation of action recognition using deep networks is developed through a 3D Convolutional network. They applied 3D Convolutional kernels on video frames during a time axis to capture both spatial and temporal information. They also claimed that their approach can capture motion and optical flow information because frames are connected by fully connected layers at the top. A two-stream CNN architecture is proposed during which the primary stream captures spatial and temporal information between frames and therefore, the other demonstrates the dense optical flow of multiple frames. Our method extends the Bi-LSTM model proposed.

Methodologies:

Video Streaming is growing day by day and we can see everywhere the CCTV surveillance systems, those sources have very long lengthy videos where we can use computerized methods to detect activities in those videos using our proposed work, not only to detect activities but also to detect events which are very similar to activities but are not a single activity, for example, functions, marriages, ceremonies, parties, accidents, some other situations, etc. The computer storage and processing power were growing powerful year by year. We are uploading/streaming video content to the internet daily, organizing and delivering streaming content directly to the audience by the type of video, and what is happening in the video.

We assume videos as a sequence of frames that were used to detect certain events in it. Our proposed work uses two algorithms, the first one is CNN which is used to extract features from the video frames and the second one is Bi-LSTM which understands the relation between different scenes that were happening on frames and identifies the event or activity that it is trained to.

Our method consists of two models. The training of the models will be explained later in this paper. The two models CNN and any model that learns term dependencies like LSTM, RNN, Bi-LSTM, etc. Here we use Bi-LSTM as it is far better than the other two models. Also, our proposed method uses a deque to track out the N features from the frames and to pass to the Bi-LSTM model.

1) Convolutional Neural Network:

CNN architectures trained on videos have emerged, with the target of capturing and encoding motion information. a CNN model trained on the aspect of the frames with a CNN model trained on stacked optical flow features to match the performance of hand-crafted spatiotemporal features provides an extensive experimental evaluation of multiple approaches for extending CNN's into the video classification on a large-scale dataset of 1 million videos with 487 categories (which we release as Sports-1M dataset) and report significant gains in performance over strong feature-based baselines. Let f ∈ R D denote the primary video-level CNN feature vector and ˜f ∈ R D its normalized version. We have investigated three different normalizations: `1 normalization ˜f = f/kfk1; `2 normalizations ˜f = f/kfk2, which is commonly performed before training an SVM model; and root normalization ˜f = pf/kfk1 introduced in [1] and shown to boost the performance of SIFT descriptors.

We applied SVM classifiers to the video-level features, including linear SVM and nonlinear SVM with Gaussian radial basis

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

function (RBF) kernel exp2x-y and exponential χ 2 kernel exp{−γ P i (xi−yi) 2 xi+yi }. Principal component analysis (PCA) is applied SVM to cut back dimensions. An efficient approach to speeding up the runtime performance of CNNs is to alter the architecture to contain two separate streams of processing: a context stream that learns features on low-resolution frames and a high-resolution fovea stream that only operates on the center portion of the frame. Since all successful applications of CNNs in image domains share the availability of an oversized training set, we speculate that this can be often partly thanks to lack of large-scale video classification benchmarks.

The deeper CNN architecture yields consistently better performance resulting from the depths and tiny receptive field all told convolutional layers. We also observed that both hidden layers outperformed the output layer when the CNN architecture, normalization, and spatiotemporal pooling strategies are the identical

2) Making a Feature Extractor

Feature Extractor is nothing but CNN. Where CNN consists of many layers, a convolutional layer, and followed by a pooling layer, the output that is produced from those layers has lesser dimensionality. So, these can be considered as features.

In order to improve the performance, we need to train the scene related images to our CNN for closed level detection. To train CNN we can use State-of-Art CNNs to fasten up the training process. CNN is trained using a Supervised Learning method. So, it will consist of a dataset of labeled videos. The labeled videos are taken in a 3 level tree directory structure as shown in the below diagram.

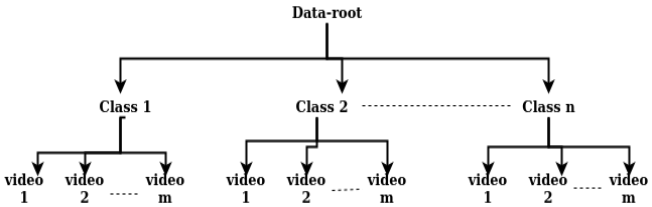
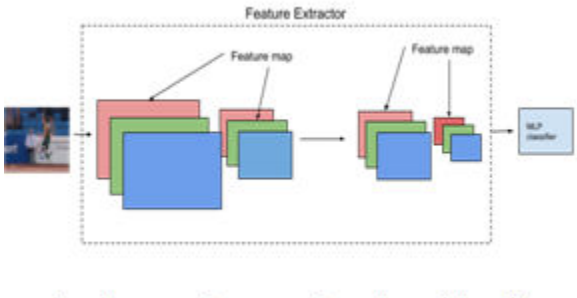


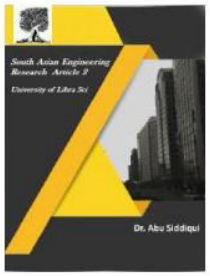
Fig 2.1: Showing how videos were taken

These videos are taken such that those videos will represent any activity or event that occurs over video. We selected videos from different sources for testing purposes. Also, we make sure the distribution of videos is balanced, if not the training may go wrong.

All the bottom ones are video files and other than the last level items were all directories that contain respective class video files. But CNN needs images to train on. So, we extracted the images from video files by traversing through each class directories and for each video in the same directories. We used the transfer learning method to improve both accuracy and training process with lesser data.



In order to train through transfer learning, we need to divide the model into the CNN part also called 'Feature Extractor' and a classifier. We use the feature map of pre-trained CNN in our new CNN and classifier weights are trained. However, the images from videos that will be not so clear. So, we need to train the feature extractor but using the pre-trained feature maps. The training images are loaded

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal


through directories.

3) LSTM and Bi-LSTM: Long short-term memory (LSTM) is a man-made recurrent neural network (RNN) architecture utilized in the sphere of deep learning. A typical LSTM unit is consists of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and also the three gates regulate the flow of information into and out of the cell.The cell is responsible for keeping track of the dependencies between the weather within the input sequence. The input gate controls the extent to which a latest new value flows into the cell, the forget gate controls the extent to which a worth remains within in the cell and so the output gate controls the extent to which the price within the cell is utilized to compute the output activation of the LSTM unit. The activation function of the LSTM gates is usually the logistic sigmoid function. There are connections into and out of the LSTM gates, a number of which are recurrent. The weights of those connections, which require to be learned during training, determine how the gates operate.

An RNN using LSTM units are trained during a supervised fashion, on a group of coaching sequences, using an optimization algorithm, like gradient descent, combined with backpropagation through time to compute the gradients needed during the optimization process, so onaltering each weight of the LSTM network in proportion to the derivative of the error (at the output layer of the LSTM network) with regard to corresponding weight.

A problem with using gradient descent for traditional RNNs is that error gradients vanish exponentially quickly with the dimensions of the interruption between important events. Bidirectional recurrent neural network(RNN) is really just putting two independent RNN's together. This structure allows the networks to possess both backward and forward information about the sequence at every occasion step.

Using bidirectional will run your inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that within the LSTM that runs backward you preserve information from the long term and using the two hidden states combined you are able in any point in time to preserve information from both past and future. Here in LSTM,(1). we use activation values, not just C (candidate values),(2).we even have 2 outputs from the cell, a replacement activation, and a spanking new candidate value.so to calculate the new candidate in LSTM it can control the memory cell through 3 different gates as we said before we've got 2 outputs from LSTM, the new candidate and an innovative new activation, in them we'd use the previous gates.

4) Collecting Dataset for Training Bi-LSTM:

Our videos will be the raw dataset to the Bi-LSTM. Videos are raw datasets now we need to process them into model understandable data. So, with the help of a feature extractor and a deque to collect sequences of features.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

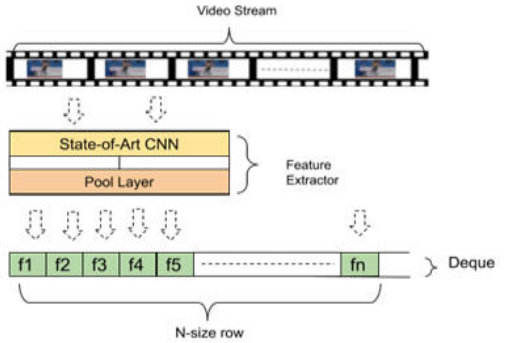


Fig 4.1: image showing how a part video is converted to a single row in the dataset

Videos were taken in a directory structure shown above fig2.1, we take every video of a class for each and every video we do feature extraction as shown in above fig4.1, then the video is converted to different data item sets by the following procedure

```
[1]   init dataset D
[2]   init deque Dq
[3]   init N
[4]   for frame in getnextframe(video)
[5]       features =
      getfeatures_featureextractor(frame)
[6]       push features into Dq
[7]       if length of Dq is N
[8]             sequence = list Dq
[9]             add sequence in dataset D
[10]              pop features from Dq
[11]      endloop
```

These collected features dataset is used to train the model, where the labels are automatically taken when label video processing starts. This collected dataset is used to train Bi-LSTM.

5) Activity and Event Detection:

Nowadays every machine has powerful processing power called GPUs and also cloud computing introduces us with huge processing power. All these computation power is through multiprocessing. Now in order to continuously classify or detect activities over video streaming, we use multiprocessing mechanisms, therefore the time taken to detect activity is greatly reduced. Activity and events will be detected in the video stream, which is taken directly as input for models, the stream is classified in 2 step process.

(1) Feature pulling (2) Action and event detection

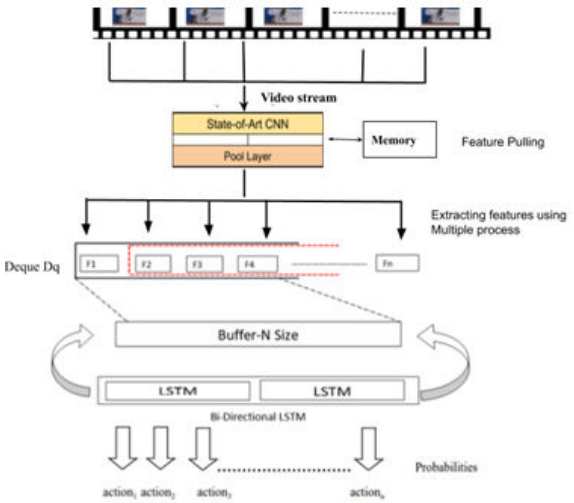


Fig 6.1: showing activity detection using multiprocessing power

Feature pulling: Future pulling is nothing but extracting features from a stream and for multiple frames using multiprocessing on multiple GPU's or CPU's. In order to extract multiple frames from video stream directly and are shared through a disk memory. When the procedure starts some bunch of processes is initialized and share a common model for feature extraction. These features are pushed to a shared deque Dq. Some synchronized methods can be used were to handle the sequencing problem for features from a video. We used a separate process that transfer frames to a process with a sequence number and it sends back the feature array to that process along with sequence number. That process repeatedly pushes the features into deque Dq and that process also handles the next step.

Action and event detection: Here a single process that iteratively grabs a buffer of features from deque Dq and feeds as input to the Bi-LSTM model

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

where it outputs a buffer of probabilities of actions or events. The buffer is of N-size in this step, the process regularly keeps check of the deque to fill out N-size whenever it reaches its size, the process do the following.

1) if length of Dq is N
2) buffer = list Dq
3) label = model_predict(buffer)
4) pop Dq
5) else
6) wait for features
7) features = get_features()
8) push features to Dq

Thus the actions were detected over a video stream constantly until the process is killed. We can keep track of actions detected over a video and to separate the video into chunks of videos occurring different actions and events.

Results:

We have taken a sports video from the internet and pass them as stream to our process. The actual inputs and the output produced is as shown below.

fig 6.2: Extracted videos from a video based on a video stres

Conclusion:

Nowadays as computation power increases as increasing with the amount of data to process. As most of the data is generated every day is multimedia data. In many different media types video is one of them. Where finding what happens over a video as it streams or recorded using a computerized method is more helpful in CCTV surveillance, video categorizing, etc. Making shorter videos for longer videos can be achieved through this algorithm, where longer videos are used to detect events and separate shorter videos where an event occurs over video automatically. As video length increases we need to increase the no of processes to work on feature extraction, thus we can speed up the amount of data is reduced as CNN features and also multiple Bi-LSTM models can be used to directly classify the multiple buffers. So, as we increasingly grow the no of the process with the no of buffers using multiple processes to do the action detection, the lesser the processing time is taken.

References:

[1]" Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features" AMIN ULLAH[1] , (Student Member, IEEE), JAMIL AHMAD[1] , (Student Member, IEEE), KHAN MUHAMMAD[1] , (Student Member, IEEE), MUHAMMAD SAJJAD[2] , SUNG WOOK BAIK[1] , (Member, IEEE)

[2] A. Nanda, D. S. Chauhan, P. K. Sa, and S. Bakshi, ''Illumination and scale invariant relevant visual features with hypergraph-based learning for multi-shot person re-identification,'' Multimedia Tools Appl., pp. 1–26, Jun. 2017, doi: https://doi.org/10.1007/s11042-017-4875-7

[3] K. Soomro, A. R. Zamir, and M. Shah. (2012). ''UCF101: A dataset of 101 human actions classes from videos in the wild.'' [Online]. Available: https://arxiv.org/abs/1212.0402

[4] S. Herath, M. Harandi, and F. Porikli, ''Going deeper into action recognition: A survey,'' Image Vis.

Comput., vol. 60, pp. 4–21, Apr. 2017.

[5] A. Nanda, P. K. Sa, S. K. Choudhury, S. Bakshi, and B. Majhi, ‘‘A neuromorphic person re-identification framework for video surveillance,’’ IEEE Access, vol. 5, pp. 6471–6482, 2017.

[6] “Exploiting Image-trained CNN Architectures for Unconstrained Video Classification” Shengxin Zha Northwestern University Evanston IL USA, Florian Luisier, Walter Andrews Raytheon BBN Technologies Cambridge, MA USA

[7] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2911–2918, June 2012.

[8] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In European Conference on Computer Vision (ECCV), pages 404–417. Springer, 2006.

[9] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 3286–3293. IEEE, 2014.

[10] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In Workshop on statistical learning in computer vision, ECCV, volume 1, pages 1–2, 2004

[11] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2555–2562. IEEE, 2013.

[12] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1):221– 231, 2013. 1, 2

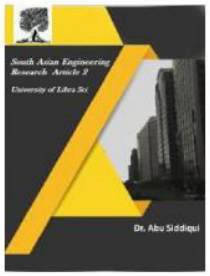
[13] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In Computer Vision–ECCV 2012, pages 425–438. Springer, 2012. 1

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014. 2, 7

[15] Large-scale Video Classification with Convolutional Neural Networks
Andrej Karpathy[1,2], George Toderici[1] , Sanketh Shetty[1] , Thomas Leung[1] , Rahul Sukthankar[1], Li Fei-Fei[2]

[16] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos “in the wild”. In CVPR, 2009. 2

[17] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In ECCV, pages 392–405. Springer, 2010. 2



[18] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. arXiv preprint arXiv:1403.6382, 2014. 1, 2

[19] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In ICPR, 2012. 2

[20] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization, and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013. 1, 2