Signal, Image and Video Processing (2021) 15:779–787
https://doi.org/10.1007/s11760-020-01796-z

ORIGINAL PAPER

A new 3D convolutional neural network (3D-CNN) framework for multimedia event detection

Kaavya Kanagaraj[1] · G. G. Lakshmi Priya[1]

Received: 2 August 2019 / Revised: 29 September 2020 / Accepted: 1 October 2020 / Published online: 19 October 2020

Abstract
Multimedia event detection has received a great deal of interest due to developments in video technology and an increase in multimedia data. However, complexities of video content such as noisy, overlapping, repeated interaction between individuals, and various scenes are becoming difficult for characterizing the subjects and concepts. In particular, Internet users find it difficult to search for a specified event. To solve the above problem, a method is proposed that best suits for event detection, demonstrating the 3D convolutional neural network (3D-CNN) structure to accomplish promising performance in multimedia event classification. To take an advantage of motion content of the event in the video, temporal axis is considered. Both the feature extraction and classification are incorporated in this model. Experiments are carried out on the Columbia Consumer Video benchmark dataset, and results are compared with other existing works.

Keywords Multimedia event detection · 3D convolutional neural network · Feature extraction · Classification · Mean average precision · Columbia consumer video

1 Introduction

With the proliferation of video content, advanced technology for indexing, filtering, searching, and mining the enormous amount of videos is increasingly needed. In social networks such as YouTube and Facebook, lakhs of user-generated videos are uploaded every week. The user-generated video resolution of cell phones, tablets, iPods, etc. is lower compared to professionally filmed video. It may undergo some jerkiness, partial occlusion, more people interaction, noisy environmental condition, etc. Due to this, the presence of visual feature becomes less, resulting in lack of semantic meaning. It challenges the problem of identifying the specific event of interest in the videos. As a result, overall need remains for the technology where the video content is automatically identified. This motivates us to focus on multimedia event detection task, where specific event of interest is retrieved from the videos.

Although multimedia events are complex and may include components of low level, such as human interaction, different scenes, ideas, and action, they can involve major variations in the intra-class. For example, the events such as 'dog' and 'wedding reception' involve the concepts like dog, wedding couple, ship/boat, dining, etc. Hence, by using single concept, it is difficult to interpret the individual event completely. Moreover, videos are often captured in real-world scenarios that last for seconds to few minutes in noisy environmental conditions. Because of those challenges, events are difficult to detect. The proposed work, i.e., a 3D convolutional neural network architecture for event detection tasks, is designed in such a way that it overcomes the above-mentioned difficulties.

The contribution of this paper is three-fold: First, this is the first work on multimedia event detection using 3D CNN, which is experimented with complex dataset. Second, the proposed identifies the salient point and its descriptors from few keyframes of the video. Third, since the architecture is based on sequential CNN model, experiments have been performed on tuning hyperparameters such as number of hidden layers, pooling types, and learning rate.

The remainder of the paper is structured as follows: Sect. 2 provides the related work. Section 3 explains the proposed work, Sect. 4 presents the experimental result and discussions, and finally, Sect. 5 provides conclusion of this paper.

✉ G. G. Lakshmi Priya
lakshmipriya.gg@vit.ac.in

[1] SITE, VIT, Vellore, Tamil Nadu, India

2 Related work

Due to the advancement of video data, huge number of video contents are generic. Therefore, a significant need arises for video classification. The video applications still challenging for the researchers are abnormal video event detection [1], multimedia event detection, surveillance event detection [2], semantic indexing [3], etc. The common way to tackle large-scale video tasks is the use of video descriptors where it identifies features that efficiently describe the visual content. These content can be searched by users by providing query. Various types of queries are text-based semantic queries, content-based queries, composite queries, spatiotemporal queries, etc. [4].

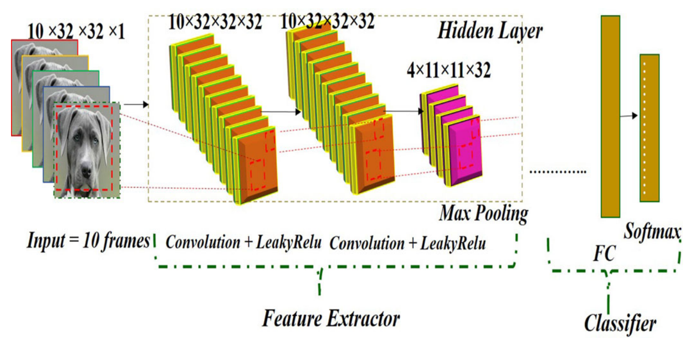
The features extracted from different modalities may be numerous [5]. They are text features, audio features, visual features, motion features, and so on. The textual features can be extracted from optical character recognition (OCR), automatic speech recognition (ASR), production metadata, etc. These features are highly domain oriented and complex in nature. Audio features are mel-frequency cepstral coefficient (MFCC), linear predictive coding (LPC), non-silence ratio, etc. Motion features are measured in terms of magnitude and direction, optical flows, histogram, and so on. However, it is a challenging task to provide better performance for large video content. Visual features include color, texture, shape. Improved dense trajectories (IDT) is the visual descriptor which includes both spatial and temporal contents. However, it suffers from high computational complexity and limited memory storage capacity.

Due to the above limitations of the existing features, convolutional neural network (CNN) [6] emerged. CNN has obtained its accuracy over other features, especially in the field of image classification and event detection [7]. CNN has also been shown to be invariant to certain differences such as lighting, pose, and surrounding clutter [8]. It is highly concerned with unprecedented capabilities for some high-level vision tasks, such as image classification [9], concept detection [10], and scene labeling [11]. In addition, the features learned from large networks trained on the ImageNet dataset [12] show a high generalization capability that delivers state-of-the-art performance beyond the standard task of image classification, e.g., action recognition datasets [13]. Some researchers used CNN either to extract feature or classification and few used for both purposes. Due to CNN's aforementioned benefits, the proposed work utilized CNN for event recognition task.

CNN, a deep model applied to the raw input image with the use of trainable filters and pooling operations. It obtains a layer-wise hierarchy of advanced complex features. Applying sufficient regularization [14, 15] obtains superior efficiency. Various pooling and encoding strategies have been used to reduce dimensionality of videos. The last fully connected layer, gives the overall content of the input. An image classification task, Google Net [16], VGG Net [17], has been proposed. However, the FC of the above two works does not make explicit spatial content; therefore, the spatial pyramid pooling (SPP) scheme be enhanced [18]. The paper [19], proposed CNN and the recurrent neural network (RNN) architecture, in which CNN extracts every frame's static feature and RNN fuses it to represent video in constant length. In order to select the feature from 3D images, Nie et al. [20] enhanced a CNN model. They used low-level features of CNN as their content and used RNN to obtain deeper features. On their extracted features, the nearest neighbor classifier is used to exhibit the viability of their proposed model. Song et al. [21] proposed the event detection model by extracting key segments by transferring conceptual information from Web images and videos. Richard et al. [22] proposed the CNN-based classifier for feature classification. SVM has been applied to the selected features to resolve the classification issue. They claimed the accuracy of the new features that learned from CNN is discernibly superior to other standard techniques. The paper [23] proposed two-stream CNN to capture action content. It provides better video representation by integrating both spatial and temporal contents from CNN. Although these works have achieved promising results on both action recognition and video classification, they have lost accuracy on complex event detection due to the absence of ordering of temporal interactions. To gain the advantage of temporal variation, 3D CNN emerged.

The 3D convolution (s, s, d) [19], where the matrix represents the kernel, where d is the temporal depth corresponding to the number of frames used as input and s is the spatial size of kernel. Furthermore, 3D convolution can be used to obtain spatiotemporal features directly from raw videos. Some of the research works of 3D CNN are as follows: The paper [24, 25] proposed a trained 3D CNN, otherwise referred as C3D. C3D was used for the sports event recognition by using the sports-1 M dataset. In the motive of enhancing the C3D, the length of temporal inputs has been expanded, which leads to improvised performance in the action recognition task. Some authors have experimented with the use of optical flow for 3D CNN (i.e.) by providing optical flow as input for 3D CNN to provide better performance [26]. The combination of both RGB and optical flow as input provided much better performance for image recognition task [17].

As a summary of the above literature, 3D CNN has been applied to overcome some existing difficulties faced by 2D CNN. The proposed work focused on 3D CNN, which is necessary for motion identification particularly in event detection application. As it incorporates the temporal content for the feature extraction, it yields a better classification, and therefore, the proposed work can achieve promising result when compared with other approaches.



Layer descriptions:

Layer (type)	*Output shape*	*Kernel Size*	*Stride*
Conv 3d_1 (Leaky ReLu_1)	10×32×32×32	3×3×3	1×1×1
Conv 3d_2 (Leaky ReLu_2)	10×32×32×32	3×3×3	1×1×1
Max Pooling_1 (Dropout_1:0.25)	4×11×11×32	3×3×3	3×3×3
Conv 3d_3 (Leaky ReLu_3)	4×11×11×64	3×3×3	1×1×1
Conv 3d_4 (Leaky ReLu_4)	4×11×11×64	3×3×3	1×1×1
Max Pooling_2 (Dropout_2: 0.25)	2×2×2×64	3×3×3	3×3×3
Conv 3d_5 (Leaky ReLu_5)	2×2×2×64	3×3×3	1×1×1
Conv 3d_6 (Leaky ReLu_6)	2×2×2×64	3×3×3	1×1×1
Max Pooling _3 (Droput_3: 0.25)	1×2×2×64	3×3×3	3×3×3
Fully connected Layer:(Batch	1×16		
Softmax Classifier			

Fig. 1 Architecture of the proposed work

3 Proposed work

The 3D CNN model is proposed to contribute better performance in Multimedia event detection. Generally, 2D CNN was used earlier, where spatial content was captured. In the proposed work, 3D CNN is used because of the need for temporal variations, which is important for motion recognition, particularly for event detection. This work embodies both feature extraction and classification in the same architecture. The proposed 3D CNN for the extraction of salient point and its descriptors from few randomly selected sequential frames is shown in Fig. 1. The detailed explanation of the proposed work is given below.

3.1 Architecture of 3D CNN for event detection

The proposed work models an architecture for event detection task. It incorporates motion content for event analysis. In this, the preprocessing step taken care of only normalization of the input image to the size of 32×32. The input to the model was a grayscale video of dimension $10 \times 32 \times 32 \times 1$. The number of video frames that are stacked is 10. The convolutional layers used $3 \times 3 \times 3$ filters with $1 \times 1 \times 1$ pixel strides. There are 32 filters for the first and second convolution layers and 64 filters for the remaining convolution layers. The description of the layers is explained below.

3.2 Preliminaries

The proposed CNN has six convolutional layers and one fully connected layer. The 3D CNN uses convolutional and max pooling layers and kernel size $3 \times 3 \times 3$. The kernel includes the motion content; therefore, it is of the form temporal axis, width of a frame, height of a frame. Finally, one fully connected layer is adopted.

The present study utilizes the Leaky ReLu and max pooling layer. Leaky ReLu's non-linearity for hidden neurons has shown good fitting capabilities than the sigmoid function. Max pooling is used to subsample the acquired feature representation after the convolution of the input by a learned filter. This is done by separating the representation into pools and choosing the maximum number in each pool. Max pooling has the benefit of minimizing the computational strain by decreasing the amount of connections in successive convolutional layers and adding translational/rotational invariance [19].

The sandwich design of convolutional/Leaky ReLu and proper weight initializations enhanced the learning process. Following the convolutional layers, max pooling layers with a pooling window of $3 \times 3 \times 3$ and pixel strides of $3 \times 3 \times 3$ are adopted. The sixth convolutional layer's pooled output is supplied to a fully connected layer with 16 neurons, and at last softmax classifier is utilized for classification.

In the proposed work, the dropout regularization with a dropout ratio of 0.25 was applied to the result of the max pooling layer, and probability of 0.5 is fed to the softmax layer. The model was trained using the RMSProp optimizer. The initial learning rate used was 0.001, and it decreased gradually after every 20 epochs. The exponential weighted average was 0.9.

3.3 Dropout

It comprises of setting zero for each hidden neuron's output with a likelihood 0.25. In this manner, the neurons that are ’left out’ do not contribute to the forward pass and are not involved in backpropagation. The neural network samples a distinct architecture each time an input is sent, but all of these architectures share weights. As the neuron does not depend on the existence of the specific other neurons, a complex co-adoption reduction is observed in this approach. Therefore, in combination with many distinct random subsets of the other neurons, it is necessary to know more powerful features that are essential. Without dropout, an advance occurrence of over-fitting is done. If the model is over-fitted with training data, the fully connected layers have a drop of 0.5, which means that at each iteration, 50% of the neurons will be dropped randomly.

3.4 Optimizer

The proposed work has been trained using RMSProp [27]. It is an adaptive learning rate that divides the current gradient by the moving average over the root mean squared of the weighted sum of recent gradients. RMSProp is an extension of Adagrad with the addition of momentum. It overcomes the step size vanishing problem of Adagrad. Hyperparameters of the RMSProp are: $\rho = 0.9$ and epsilon $= 1 \times 10^{-6}$ with an initial learning rate of 0.0001. The learning rate was linearly decreased after every epoch (an entire loop through the training set). Weight updates were performed after every batch size of 32. The loss function for all tested model is a categorical cross-entropy. The model that achieves the lowest training loss is considered best, and its efficiency is reported on the test set.

3.5 Softmax classification

The fully connected layer is preceded by the softmax classifier, where the softmax function is used for multi-class classification. In the softmax classification, the probability of the sample(s) belonging to class (c) is shown in the following equations.

$$Y\left(\frac{c}{s}\right) = \frac{Y(s/c) \times Y(c)}{\sum_{k=1}^{c} Y(s/k) \times Y(k)} \tag{1}$$

$Y(s/c)$ is the conditional probability. *Y(c)* is the class prior probability. *C* is the number of classes.

For clarity, define G_c as

$$G_c = In(Y(s, c)) \times Y(c)) \tag{2}$$

Lastly,

$$Y(c/s) = \frac{\exp(G_c(s))}{\sum_{k=1}^{c} \exp(G_k(s))}. \tag{3}$$

Equation (3) shows the working principle of softmax classifier. Finally, the output layer consists of 20 neurons, one for each class. This layer is activated by a softmax function, which ends up giving probabilities for the input which belongs to each of the 20 classes. The main advantage of our model is that it uses very few frame and small filters that are capable of effectively capturing the relevant content from the input and require significantly less computation than filters of large sizes. The dropout used prevents over-fitting by generalizing the model and hence improves performance on the test data set.

4 Experimental results and discussion

Experiments have been carried out on the basis of two objectives—(i) identifying the best features that perform the event detection task efficiently and (ii) justifying the performance of current work with some already existing works.

4.1 Dataset

The Columbia Consumer Video (CCV) dataset is used to assess the performance of the proposed 3D CNN architecture. This dataset includes 9317 videos with 20 event categories. The duration of the dataset is of 210 h with an average length of 80 s per video.

4.2 Evaluation metrics

mAP is used in the proposed work to calculate the relevant events retrieved for each query and to average the results for human understanding. mAP is a single-valued metric that approximates the area covered by the precision recall curve, particularly for content retrieval tasks. mAP is the AP mean for all event classes. Additionally, among various evaluation measures, mAP has been appeared to have good discrimination and cohesion. It has also been used as an official performance metric in the TRECVID evaluation since 2001 [28]. The computation of mAP is shown in Eq. (4).

$$\text{mAP} = \frac{1}{N} \sum_{m=1}^{N} \left[\frac{1}{N} \sum_{n=1}^{N_m} \frac{n}{CM(m, n)} \right] \tag{4}$$

where N = Total test videos, N_m = Number of relevant videos for $N(i)$, CM is the confusion matrix of mth row and nth column.

4.3 Experiments

The proposed work requires hyperparameter tuning to provide better result. Therefore, various experiments were held for tuning hyperparameters. Since hyperparameters play a critical role in CNN, in order to make the user better under-

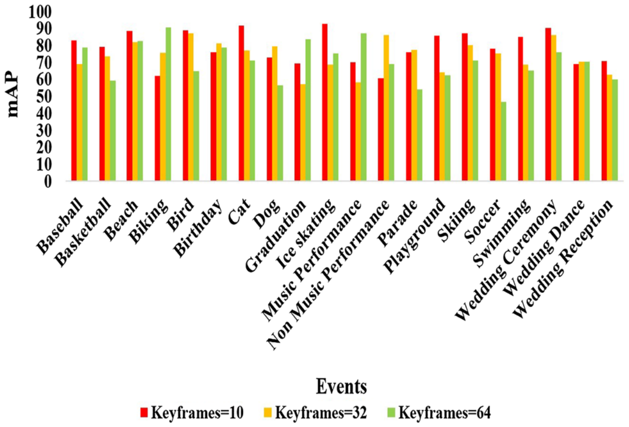


Fig. 2 Comparison of the total input frames per video

stand the proposed work, evaluation has been carried out for every step proceeded. The experiment has been carried out using 'Intel (R) Core (TM) i7-6700 k CPU @ 4.00 GHZ, 4001 MHZ, 4 Core (S), 8 Logical Processor (S).'

4.3.1 Evaluation on input frames

Since the dataset consists of 1 to 1.5 min videos, it is important to choose the total number of frames as input. The experiments were carried out on 10, 32, and 64 frames, and analysis showed that 10 frames were more accurate than any other number of frames. From the analysis, it is clear that the increasing number of frames reduces the salient point recognition. This is due to the fact that the presence of occlusions and jerkiness yields an irrelevant concept. The comparison of the total input frames per video is shown in Fig. 2.

4.3.2 Evaluation on hidden layers

In the proposed work, the term 'hidden layer' indicates the layer, consisting of [convolutional layer + leaky ReLu, convolutional layer + leaky ReLu, max pooling + dropout]. Experiments have been carried out to fix the number of hidden layers to yield better accuracy. As part of the analysis, the hidden layer greater than or less than 3 results in poor performance in event classification. The reason is the features extracted were not sufficient to classify events in less hidden layers, whereas in greater block size, i.e., addition of more hidden layers, beyond a certain threshold leads to finding irregularities in the data.

The comparison of hidden layers is shown in Table 1. Out of 20 events, the presence of single hidden layers yields better accuracy in only two events because extracted features are visually overlapped except for basketball and music performance events. In the case of two hidden layers, four events are classified, whereas when the hidden layers are increased to three, it results in classification of nine events. Here, the event parade and dog shows slight difference in accuracy than 4 hidden layers. Finally, when using 4 hidden layers, neglecting parade and dog events, only 3 events provided better classification accuracy. Therefore, as a result, three hidden layers were chosen for further experiments.

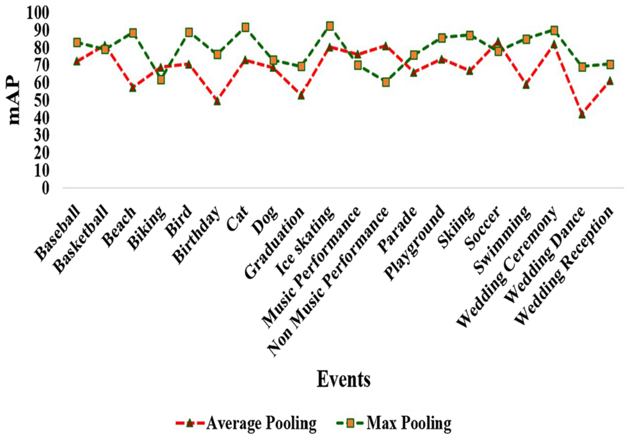


Fig. 3 Comparison of max pooling and average pooling in terms of mAP

4.3.3 Evaluation on aggregation functions

In this study, pooling is considered for feature aggregation. It has the advantage of reducing variance and computational complexity. Thus, by comparing two pooling techniques such as max and average pooling, max pooling gained more accuracy than average pooling. The reason is max pooling concentrates on extracting edge features which suits the event detection application. Average pooling cannot extract good features because it takes all into count and results in an average value that is not suitable for event detection tasks. In other words, average pooling is the much generalized computation. Comparison of average and max pooling in terms of mAP is shown in Fig. 3.

4.3.4 Evaluation on learning rate

Learning rate (lr) is the hyperparameter used to adjust the weight of the network to reduce the loss gradient. The low lr makes the training more reliable, but optimization has become time-consuming because the steps toward the minimum loss function are tiny. If the learning rate is high, then training cannot converge or even diverge. Weight changes can be so big that the optimizer overwhelms the minimum and makes the loss worse. The analysis of the learning rate for rmsprop optimizer is presented in Table 2. Analyzing the table, it is noted that lr of 0.001 achieves better mAP than the remaining learning rates taken for comparison. The lr of 0.01 classified only 4 events with better accuracy, namely dog, non-music, parade, and wedding dance. Similarly, the

Table 1 Comparison of hidden layers

Events	Hidden Layer = 4	Hidden Layer = 3	Hidden Layer = 2	Hidden Layer = 1
Baseball	69.01	**83.05**	80.01	72.16
Basketball	73.45	79.08	80.83	**81.49**
Beach	82.01	**88.69**	70.16	57.34
Biking	**75.48**	62.05	53.81	68.86
Bird	87.12	**89.01**	85.43	70.59
Birthday	**81.30**	76.13	68.65	49.68
Cat	77.16	**91.65**	88.01	72.88
Dog	**74.37**	73.00	50.16	68.71
Graduation	57.16	69.30	**72.59**	53.06
Ice skating	68.82	**92.63**	77.84	80.29
Music performance	58.19	70.1	62.63	**76.12**
Non music performance	**86.22**	60.61	56.03	81.01
Parade	**77.49**	75.9	49.88	65.96
Playground	64.05	**85.81**	81.00	73.44
Skiing	80.13	**87.11**	72.88	66.77
Soccer	75.45	78.03	**90.19**	83.39
Swimming	68.56	**84.97**	53.66	59.11
Wedding ceremony	86.02	**90.13**	75.11	81.91
Wedding dance	70.43	69.16	**80.12**	42.18
Wedding reception	62.89	70.59	**72.6**	61.07
MAP (%)	**73.76**	**78.85**	**71.07**	**68.30**

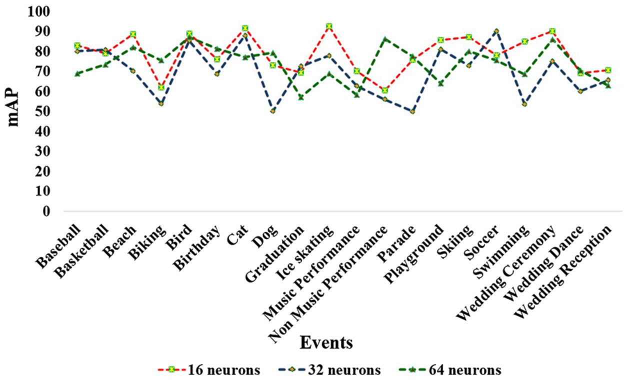


Fig. 4 Comparison of 16, 32, and 64 neurons in one-FC layer

lr of 0.0001 also classified 4 events which are highlighted in bold. Finally, the lr 0.001 provided better detection accuracy for remaining 12 events. Therefore, lr = 0.001 was chosen for further experiments.

4.3.5 Evaluation of number of FC layer

Experiments have been performed to fix neurons and the number of fully connected layers. The comparison of FC layers with neurons is shown in Figs. 4, 5, 6.

The graph shown in Fig. 4 illustrates the comparison of one FC layer with three different neurons. These comparisons were made in order to analyze the model to achieve

Table 2 Comparison of the lr for rmsprop optimizer

Events	lr = 0.01	lr = 0.001	lr = 0.0001
Baseball	71.48	**83.05**	78.63
Basketball	64.32	**79.08**	59.18
Beach	69.18	**88.69**	82.72
Biking	48.08	62.05	**90.56**
Bird	30.13	**89.01**	64.88
Birthday	59.96	76.13	**78.61**
Cat	77.56	**91.65**	71.13
Dog	**83.63**	73.00	56.36
Graduation	68.70	69.30	**83.49**
Ice skating	87.49	**92.63**	75.33
Music performance	72.42	70.1	**87.27**
Non music performance	**79.76**	60.61	69.01
Parade	**86.89**	75.9	54.11
Playground	44.12	**85.81**	62.23
Skiing	56.88	**87.11**	70.94
Soccer	65.19	**78.03**	46.62
Swimming	49.98	**84.97**	65.19
Wedding ceremony	88.16	**90.13**	76.13
Wedding dance	**75.79**	69.16	70.29
Wedding reception	67.57	**70.59**	59.91
MAP (%)	**74.98**	**78.85**	**70.12**

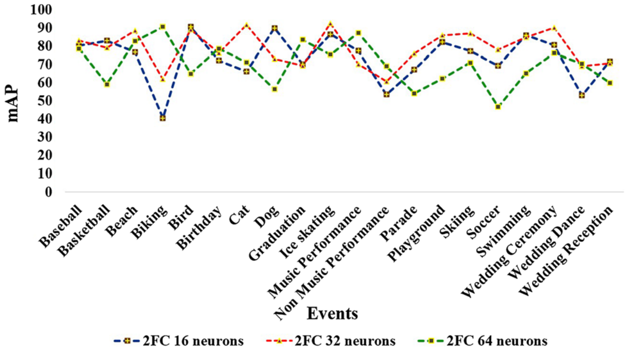


Fig. 5 Comparison of 16, 32, and 64 neurons in two-FC layer

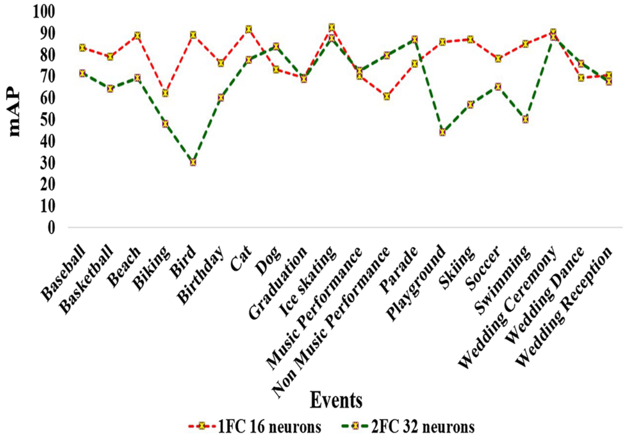


Fig. 6 Comparison of 16 and 32 of one-FC and two-FC layers

remarkable improvement. Although adding more FC layers is advantageous, it is a time-consuming approach because, in order to make the model fit better, considering minute differences in accuracy is meaningless. Experiments were conducted in one FC and two FC's in order to fix the neurons. The proposed model results more accuracy in 16 neurons than 32 and 64. The reason is that the data used are complex, which leads to expressing complicated functions when adding more neurons.

Generally, the conv layers extract the local content, while the FC layers represent global content. Although the addition of more FC layers provides for stronger improvement, the proposed model is better when using single FC than two FC. This is because, as the FC layer has increased, it leads to over-fitting of data.

Figures 7, 8, 9, 10 show the loss rate and accuracy of the current model during training and validation of various iteration rates such as 80, 100, 150, and 200. It is clearly visible that the accuracy in the plots of different iteration rates advanced gently as the number of iteration advances. Also, the loss rate lowered in the opposite side of the accuracy.

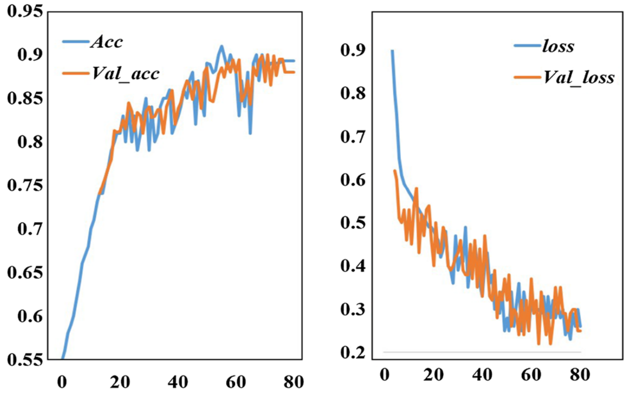


Fig. 7 Accuracy and loss for 80 epochs

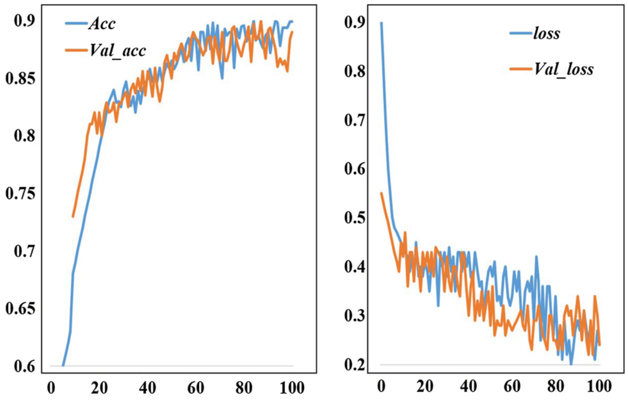


Fig. 8 Accuracy and loss for 100 epochs

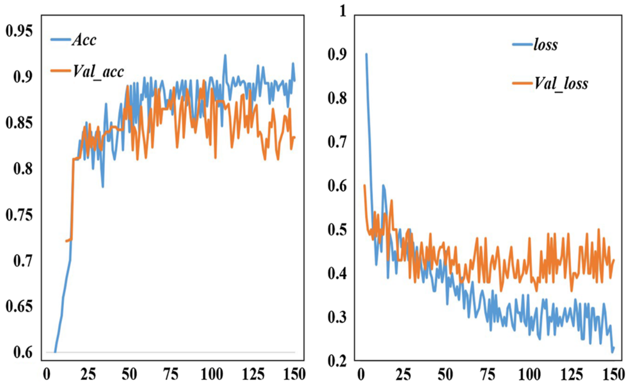


Fig. 9 Accuracy and loss for 150 epochs

The proposed work is compared with other existing works and has proved that it achieves better result than several existing works. Comparison of the proposed work with state-of-the-art methods is shown in Table 3.

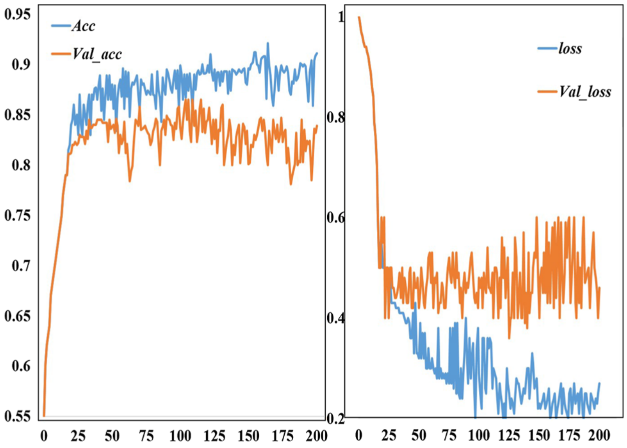


Fig. 10 Accuracy and loss for 200 epochs

Table 3 Comparison of the proposed work with state-of–the- art methods

Methods	mAP
2D CNN + VLAD +LSTM + SVM + Softmax [29]	69.1
rDNN [30]	73.5
2D CNN + ASLN + VLAD + SVM [31]	78.4
2D CNN + SVM [27]	75.1
Proposed work	**78.85**

4.3.6 Comparison of the proposed work with state-of-the-art methods

Let us discuss various baseline methods taken for comparison with the proposed work. The state-of-the-art method for video event detection is proposed [29] where two independent CNN's are modeled, one for semantic and other for scene feature extraction. Vector of locally aggregated descriptor (VLAD) is adopted to encode spatial information from the scene. To learn temporal information, the LSTM is adopted for fully connected deep features. Finally, the hierarchical late fusion method is employed to fuse the outputs of SVM and softmax classifier. This work shows that spatial VLAD performs better on scene-specific categories, such as ice skating, birthday party, wedding ceremony, and playground. In the case of LSTM, insufficient samples or non-exemplary videos in the dataset may not train an efficient model and ultimately lead to over-fitting. The proposed work provides a better result, without using separate approaches to the extraction and classification of features. Spatiotemporal features are extracted from the same model. This can be achieved by the usage of dropout and normalization parameter tuning, which avoids overfit.

Jiang, Yu-Gang, et al. [30] suggested regularized DNN (rDNN) for MED. It is proposed to exploit the rDNN feature and class relations. By imposing explicit forms of regularizations, the class relationships can be better exploited for improved video categorization performance. Nonetheless, minimizing the variation in information is not suitable for exploiting videos information in this complex dataset. This drawback is overcome by the proposed work by jointly learning both the feature representations and classification model to aggregate diversified information.

The other state-of-the-art method proposed for MED is based on attention based local saliency localization (ASLN), representation, and classification [31]. The ASLN is built to predict semantic saliency objects. By using salient objects and whole feature map, the frame-level features are extracted. VLAD is adopted to encode frame-level features for video representation. The SVM classifier is used for classification purposes. This approach achieves a mAP of 78.4, which is slightly lower than the proposed work. Contrary to this method, the proposed study used 3D CNN where temporal information is embedded with the involvement of few hyperparameters. Note that our pipeline is fully automated and does not require manual intervention. For concept retention, 3D CNN acts and extracts more information than 2D.

The video event recognition based on frame-level CNN descriptors is proposed [27]. The hierarchical concept score postprocessing method for mid-level video representation is discussed. The WordNet concept tree is considered to alleviate uncertainty of resulted concept scores. Next, a concept-wise power-law normalization method for feature normalization has been proposed. However, the highly correlated event classes are difficult to recognize. Due to the dispersion of power, a certain amount of mid-values is dispersed through the use of power law. However, concept-wise method is incorporated to improve performance, with unconstrained events in the dataset leading to less efficiency. In the proposed work, 3D CNN is used to overcome the problem of concept detection. As 3D CNN iteratively samples, its kernels until all voxels predictions are associated with it. The implementation is less efficient when parameters are not fine-tuned. In order to make it easier to understand, Table 1 shows the concepts extraction using tuning of hidden layers.

5 Conclusion

In this paper, the proposed 3D CNN for multimedia event detection task combines both the feature extraction and classification work in the same model. 3D CNN is used to integrate the temporal features, which is necessary for motion identification. The better setting of hyperparameters provides a good result, especially with the use of minimum input frames per video. Experiments are carried out using the standard benchmark Columbia Consumer Video dataset. The proposed work has yielded better accuracy compared to the state-of-the-art methods. In the future, we will continue to work with our model with more number of datasets.

Acknowledgement The authors thank the management for providing VIT SEED GRANT for carrying out this research work. Further, the authors thank the NVIDIA for providing the Titan Xp GPU card under the NVIDIA GPU Grant scheme.

References

1. Kangwei, Liu, Jianhua, Wan, Zhongzhi, Han: Abnormal event detection and localization using level set based on hybrid features. Signal Image Video Process. **12**(2), 255–261 (2018)
2. Saykol, E., et al.: Keyframe labeling technique for surveillance event classification. Opt. Eng. **49**(11), 117203 (2010)
3. Srikanth, D., Sakthivel, S.: Vantage Point Latent Semantic Indexing for multimedia web document search. Clust. Comput. **22**, 10587–10594 (2019). https://doi.org/10.1007/s10586-017-1135-6
4. Baştan, M., et al.: Bilvideo-7: an MPEG-7-compatible video indexing and retrieval system. IEEE MultiMed. **17**(3), 62–73 (2010)
5. Atrey, P.K., et al.: Multimodal fusion for multimedia analysis: a survey. Multimed. Syst. **16**(6), 345–379 (2010)
6. LeCun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
7. Deng, J. et al.: Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp. 248–255 (2009)
8. Ji, S., et al.: 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2012)
9. Krizhevsky, A., Ilya S., Geoffrey E. H.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 1097–1105 (2012)
10. Girshick, R., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
11. Farabet, C., et al.: Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1915–1929 (2012)
12. Deng, J., et al.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255 (2009)
13. Karpathy, A., et al.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)
14. Yu, K., Wei X., Yihong G.: Deep learning with kernel regularization for visual recognition. In Advances in Neural Information Processing Systems, pp.1889–1896 (2009)
15. Mobahi, H., Ronan C., Jason W.: Deep learning from temporal coherence in video. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 737-744. (2009)
16. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
17. Simonyan, K., Andrew Z.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556(2014)
18. He, K., et al.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
19. Yue-Hei Ng, Joe, et al.: Beyond short snippets: Deep networks for video classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4694–4702 (2015)
20. Nie, W., et al.: Convolutional deep learning for 3D object retrieval. Multimed. Syst. **23**(3), 325–332 (2017)
21. Song, H., et al.: Extracting key segments of videos for event detection by learning from web sources. IEEE Trans. Multimed. **20**(5), 1088–1100 (2018)
22. Socher, R., et al.: Convolutional-recursive deep learning for 3d object classification. Adv. Neural Inf. Process. Syst. 656–664 (2012)
23. Ye, H., et al.: Evaluating two-stream CNN for video classification. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. pp. 435–44 (2015)
24. Hinton, Geoffrey, Nitish Srivastava, and Kevin Swersky. “Lecture 6a overview of mini–batch gradient descent.” *Coursera Lecture slides* https://class . *coursera. org/neuralnets-2012-001/lecture,[Online* (2012)
25. Karpathy, A., et al.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2014)
26. Varol, Gül, Laptev, Ivan, Schmid, Cordelia: Long-term temporal convolutions for action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**(6), 1510–1517 (2017)
27. Soltanian, Mohammad, Ghaemmaghami, Shahrokh: Hierarchical Concept Score Postprocessing and Concept-Wise Normalization in CNN-Based Video Event Recognition. IEEE Trans. Multimed. **21**(1), 157–172 (2018)
28. Wang, H., et al.: Evaluation of local spatio-temporal features for action recognition. In: British Machine Vision Conference, London, United Kingdom (2009)
29. Zhao, Zhicheng, Song, Yifan, Fei, Su: Specific video identification via joint learning of latent semantic concept, scene and temporal structure. Neurocomputing **208**, 378–386 (2016)
30. Jiang, Y.-G., et al.: Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **40**(2), 352–364 (2017)
31. Zhao, Zhicheng, Xiang, Rui, Fei, Su: Complex event detection via attention-based video representation and classification. Multimed. Tools Appl. **77**(3), 3209–3227 (2018)