ECNN: Activity Recognition Using Ensembled Convolutional Neural Networks

Aprameyo Roy
Dept. of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram, India - 695547
aproroy@gmail.com

Deepak Mishra
Dept. of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram, India - 695547
deepak.mishra@iist.ac.in

***Abstract*—Human Activity Recognition (HAR) has been a compelling problem in the field of computer vision since a long time. Our focus is to address the problem of trimmed activity recognition which is to identify the class of human activity in a video which is temporally trimmed to contain only those periods where human activity is present. In the past few years there has been a transition from handcrafted features for classification to deep convolutional neural networks which work on raw video data to extract features and classify human activities. 3D convolutional neural networks learn features from both the temporal as well as spatial dimensions and prove to be very powerful in finding correlations in signals containing spatiotemporal information. 3D-CNNs have been extremely successful in activity recognition. We explore the shortcomings of a 3D-CNN architecture and propose ensembling with a 2D-CNN to overcome these for a significantly better performance in activity recognition.**

***Index Terms*—Human Activity Recognition, 3D-CNN, 2D-CNN, spatio temporal, ensembling**

I. Introduction

Human activity recognition(HAR) has become a highly competitive and researched area in the field of computer vision owing to the recent advances in deep learning which make this problem more approachable and also its numerous applications like surveillance [1], assisted living [3], human-computer interaction [2] etc.

Convolutional Neural Networks(CNNs) have been very successful in image based feature extraction and classification. They have set benchmarks and still continue to be an integral part of deep architectures used in various applications like object recognition [4] [5] and video classification [6].

CNNs can exploit 2D data like images very effectively, however activity recognition requires understanding of an additional dimension, i.e. the temporal domain. To exploit the success of 2D-CNN on three dimensional data, 3D-CNN was proposed by S Ji [7] et. al. 3D-CNN is capable of capturing the spatial as well as temporal information effectively from a given video and thus proved to be extremely useful in tasks like activity recognition. With further improvements on the 3D-CNN architecture, state-of-the art results were obtained in activity recognition by training from scratch and also transfer learning [8] [9].

A major shortcoming of the 3D-CNN is the high training time and number of weights that is to be updated even for a relatively shallow architecture due to three dimensionality of the convolutional kernel. In our research, we try to exploit the fact that we are working on trimmed activity recognition where each frame contains important information about the activity present in the video. We try to recognize the activity by training a 2D-CNN with single frames randomly selected per training video from the dataset to assess the importance of each frame in trimmed activity recognition.

We also assess the performance of a 3D-CNN architecture on UCF-101 dataset [10]. Although 3D-CNN provides good accuracy in recognition overall, it falters in some classes and we posit that such errors caused by 3D-CNN can be offset by ensembling it with a 2D-CNN which contains only spatial information. We suggest that a 3D-CNN trades off important spatial information to be better overall in spatiotemporal feature extraction. This is because of the update of kernel weights while moving along frames. A 2D-CNN works well when used together with a 3D-CNN and competes well with the current state-of-the-art architecture in activity recognition which has been trained only on the given dataset.

In Section II we discuss the previous work done in HAR including the most recent involving 3D-CNN. Section III(A) pertains to the detailed description of 3D-CNN architecture used in our model. It elaborates on the working of 3D-CNN and preprocessing of the input data. Section III(B) discusses the architecture of the 2D-CNN employed in our model and also the preprocessing of the data for its input. Section III(C) discusses the final model which ensembles the two CNNs together and is used for classification of the test data. In Section(IV) we compare our results to other models and also elaborate on the implementation details.

II. Related Work

Activity recognition requires utilization of spatial as well as temporal information for superior performance. Classical approach to solve such a problem is to to extract local spatial features from an individual frame by the means of interest points [11] [12] which tends to follow the motion of object in the video [13]. The extracted features are encoded to represent the video in a vector space. Descriptors like Harris3D [15], SIFT3D [16] and improved Dense Trajectories (iDT) [14] which is the current state-of-the-art for handcrafted features

978-1-7281-1895-6/19/$31.00

can be used to describe the interest points. These vectors along with the action labels are then fed to a classifier for classification.

Recently, deep learning models have been successful in tackling the problem of HAR very efficiently. CNNs [23] have given extraordinary results in visual challenges [4] [24]. More recently in the area of HAR, two stream CNNs [17] was proposed which fuse information of spatially and temporally extracted data by training two separate CNNs on raw frames and optical flow displacement fields of consecutive frames as input of spatial and temporal streams respectively. As an extension, spatiotemporal residual networks [19] have been developed which is the current state-of-the-art in HAR involving end to end training only on the dataset being evaluated. These networks introduce connections between the spatial and temporal streams which help in hierarchically learning spatiotemporal features.

3D-CNN [7] provides an effective way to automatically capture the spatiotemporal information from a video by using 3D convolution operations to arrive at features best suited to classify different actions. Several improvements and augmentations [8] [18] have been performed to the original 3D-CNN architecture to further improve its results. However the best results are obtained by inflated two stream CNNs [9] which inflate the existing architecture and initialize weights of kernels using weights of pretrained models from the ImageNet challenge. It is then trained end to end on the Kinetics [20] dataset which is a massive dataset with over 400 classes. This is hugely time consuming and resource intensive, however the method proposed by us gives significantly good results without training on a secondary dataset much complex than the primary test dataset and surpasses any other model aiming to do the same.

III. Proposed Method

In this section we first establish the ability of a 3D-CNN architecture to recognize human activities on the UCF-101 dataset. This dataset contains 101 classes of human activities with a variety of trimmed videos for each activity. Subsequently the results on the same dataset using a 2D-CNN with a single frame from each video as input is also obtained.

A model is proposed, which fuses the results of the 3D-CNN and 2D-CNN in classifying the test videos. The 2D-CNN helps in reinforcing the 3D-CNN with important spatial information which it loses in the process of getting better at overall spatiotemporal understanding. The two seprately trained CNNs produce a prediction matrix conatining the index of training video and its class probabilities for 101 classes. We do a dot product of these matrices and the final matrix is sorted according to highest probabilities for the final prediction vector [Fig. 2].

A. Primary 3D-CNN architecture

A 3D-CNN is a logical extension of 2D-CNN which works with three dimensional data like video which has an additional temporal dimension in addition to the X and Y co-ordinates. The convolution operation takes place between the video and a 3D filter [Fig. 1] which is able to capture features both in the spatial as well as temporal domain for classification purposes.

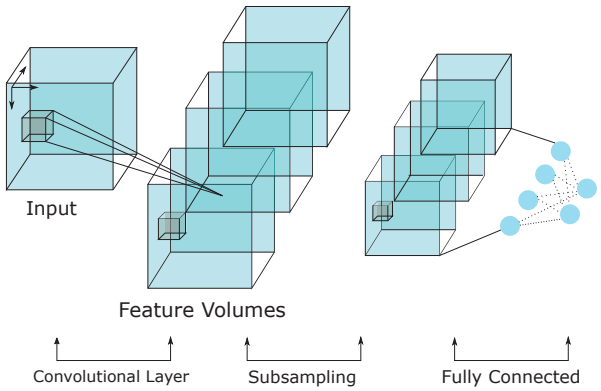

Fig. 1. Basic 3D CNN architecture: the 3D filter is convolved with the video in three dimensions as indicated by the arrows to produce feature volumes. After subsampling and flattening the features are fed to a fully connected layer for classification

The architecture takes in an input of a stack of frames. These frames have been gathered from the training videos sequentially as shown in [Fig. 2]. We work with the first 60 frames of each video. The videos are of different durations and videos with less than 60 frames has new frames of all pixel values 255 appended to the stack. The frames are resized to (56x56) for computational ease. The input dimension is (60x56x56x3) due to the RGB channels.

The architecture uses dropout [21] after the second and fourth convolutional layers and uses ReLU and softmax as its activation and loss functions respectively. The convolution operation [7] takes place with filters of dimensions (3x3x3) as it gives best results among all other receptive field dimensions [8].

UCF-101 has been divided into three folders/splits containing different training videos for ease of processing. We run the training for 40 epochs per train split and use the weights saved in the previous training split to initialize weights for training on the subsequent split. We have used a softmax activation function and the loss function as categorical crossentropy. The learning rate is set to 0.001 and the batch size to 20.

We observe that the overall accuracy on the test dataset of the UCF-101 after complete training from scratch is **90.04%** and the training time is 10.5 hours on our system which we set as a standard to calculate differences in time.

This performance is significantly better than several handcrafted feature based architectures but is still behind the state-of-the art. On carefully studying the confusion matrix of the test data, we observe that this model performs well overall but falters for a few classes like [Basketball, Bowling, Boxing-PunchingBag] whose classification presision are considerably lower than all the other activities. The lowest being 0.42. The misclassifications due to these activities take up the highest percentage among all other classes. On probing the videos, we find a visual similarity in the environments between the

classes which is the suspected reason for the dip in precision. We explore the ability of a 2D-CNN to offset this error and simultaneously test its ability in HAR as a standalone architecture.

B. 2D-CNN architecture for reinforcing 3D-CNN results

The 2D-CNN architecture is obtained by flattening the 3D-CNN architecture so that the features extracted are compatible for fusion. Similar to the 3D-CNN, this model also has two dropout layers and same activation and loss functions.

A random frame is selected from each of the training splits and train the network end to end. Each frame selected is resized to (128x128) to preserve spatial detail. The input dimensions are (128x128x3). The learning rate is 0.001 and batch size is 150. Each split is trained for 50 epochs.

It is observed that the duration of training is approximately one hour on the standard system (Section IV) and test accuracy is **86.6%** and minimum precision value for a class is **0.72** which is a feasible tradeoff in performance against the time it takes to train. In just one hour, a 2D-CNN is able to predict with such high accuracy what the 3D-CNN takes almost ten hours.

The effectiveness of a 2D-CNN on trimmed activity recognition is established as it utilizes the important information from a single frame. We observe that the 2D-CNN does not suffer from biases like the 3D-CNN. The variance of precision values in 2D-CNN is less than that of 3D-CNN, therefore we can reinforce the results of the 3D-CNN by that of 2D-CNN.

Assigning $Precision_{3D}$ as the classification precision for a particular class using standalone 3DCNN and $Precision_{2D}$ is the classification precision for a particular class using standalone 2DCNN.

$$Average(Precision_{3D}) \geq Average(Precision_{2D}) \quad (1)$$

$$Min(Precision_{3D}) \ll Min(Precision_{2D}) \quad (2)$$

The average of $Precision_{3D}$ or accuracy of 3DCNN is greater than average $Precision_{2D}$, however, as mentioned before, the variation in precision values for various classes is higher in 3DCNN than 2DCNN and thus prone to classification errors biased against a few classes.

These two pieces of information are used to augment the previous 3D-CNN architecture by ensembling it with the 2D-CNN. We fuse the class probabilities to obtain final prediction on the test data.

C. Ensembled CNN Architecture

Fusing the results of the 2D-CNN and 3D-CNN by a simple operation of dot product, we are able to obtain a prediction vector containing class probabilities for each test video corresponding to a class. We have called this ensembling of the two CNNs. Ensembling is useful as the final prediction depends upon the overall spatiotemporal understanding of the 3D-CNN alongwith very important spatial fetatures as extracted by the the 2D-CNN. These two reinforce each other to offset each others shortcomings.

A dot product is chosen as it weights each of the predictions by the 3D-CNN with that of 2D-CNN and no bias is created toward any class. On sorting and assigning the index of the highest probability value after the dot product operation as the predicted class, we are able to obtain accuracy of **96.3%** which is better than the current-state-of-the art architecture trained only on the given dataset.

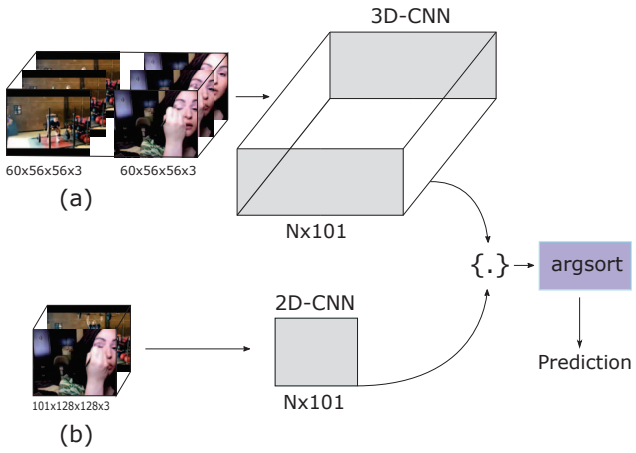

Fig. 2. Ensembled CNN architecture: (a) represents the stack of frames per video being fed into the 3D-CNN whose output is a matrix of dimensions *Nx101* where N is the number of test videos (b) Represents randomly selected frames per test video being fed to 2D-CNN whose output is a matrix of *Nx101*

IV. Implementation Details and Results

Training was done for both the architectures on a standard system with a Nvidia GTX 1060 with 6GB VRAM, 40GB of RAM and 12-Core Intel Xeon processor. We use Keras on the TensorFlow backend for creating the model.

TABLE I
Comparison of Architectures in Human Activity Recognition

Model	Accuracy
Improved Dense Trajectories(IDT) [21]	86.4%
3D-CNN [7] Pretrained(Sports 1M)	82.3%
Two Stream CNN [17]	88%
Two Stream + IDT [22]	93.5%
Spatiotemporal Residual Networks [19]	94.6%
ECNN	**96.3%**
Inflated Two Stream CNN(Kinetics Pretrained) [9]	98%

The proposed model outperforms the current state-state-of-the art model which trains on the dataset without any pre-training in terms of accuracy and training complexity. The ensembled architecture also outperforms a 3D-CNN pretrained

on the Sports 1M dataset. Pretraining is useful in situations where we have access to large computing resources and datasets. However, we cannot afford to pretrain our network on a dataset as complex and large as Kinetics due to computational power limitations. However, ensembling any network to be used for trimmed activity recognition with a 2D-CNN would improve accuracy at minimal computational cost.

V. Conclusion

In trimmed activity recognition each frame has useful information. To exploit this fact we use a 2D-CNN which is adept at learning spatial features in conjunction with a 3D-CNN. 3D-CNN when trained on the dataset tends to improve overall spatiotemporal understanding of the data due to 3D convolution operations on the stack of frames. The 3D filter moves in the temporal direction which leads the weights to update after each stride. The 3D-CNN thus trades off some spatial understanding for overall spatiotemporal understanding. To plug this deficit in spatial understanding we employ a 2D-CNN to reinforce it and the final prediction is made after taking both results into account.

Ensembling can be used with any 3D-CNN architecture to improve its classification accuracy at very low computational cost since a single frame is required from each video for training the 2D-CNN. It provides superior results over other architectures which are significantly more complex and resource intensive.

References

[1] Lin, Weiyao, Ming-Ting Sun, Radha Poovendran, and Zhengyou Zhang. "Activity recognition using a combination of category components and local models for video surveillance." IEEE Transactions on Circuits and Systems for Video Technology 18, no. 8 (2008): 1128-1139.

[2] Rodomagoulakis, Isidoros, Nikolaos Kardaris, Vassilis Pitsikalis, E. Mavroudi, Athanasios Katsamanis, Antigoni Tsiami, and Petros Maragos. "Multimodal human action recognition in assistive human-robot interaction." In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, pp. 2702-2706. IEEE, 2016.

[3] Van Kasteren, Tim, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. "Accurate activity recognition in a home setting." In Proceedings of the 10th international conference on Ubiquitous computing, pp. 1-9. ACM, 2008.

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

[5] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[6] Karpathy, Andrej, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725-1732. 2014.

[7] Ji, Shuiwang, Wei Xu, Ming Yang, and Kai Yu. "3D convolutional neural networks for human action recognition." IEEE transactions on pattern analysis and machine intelligence 35, no. 1 (2013): 221-231.

[8] Tran, Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning spatiotemporal features with 3d convolutional networks." In Computer Vision (ICCV), 2015 IEEE International Conference on, pp. 4489-4497. IEEE, 2015.

[9] Carreira, Joao, and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724-4733. IEEE, 2017.

[10] Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild." arXiv preprint arXiv:1212.0402 (2012).

[11] Wang, Heng, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. "Evaluation of local spatio-temporal features for action recognition." In BMVC 2009-British Machine Vision Conference, pp. 124-1. BMVA Press, 2009.

[12] Dollár, Piotr, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. "Behavior recognition via sparse spatio-temporal features." In Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, pp. 65-72. IEEE, 2005.

[13] Zhu, Guangming, Liang Zhang, Lin Mei, Jie Shao, Juan Song, and Peiyi Shen. "Large-scale isolated gesture recognition using pyramidal 3d convolutional networks." In Pattern Recognition (ICPR), 2016 23rd International Conference on, pp. 19-24. IEEE, 2016.

[14] Wang, Heng, and Cordelia Schmid. "Action recognition with improved trajectories." In Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 3551-3558. IEEE, 2013.

[15] Laptev, Ivan. "On space-time interest points." International journal of computer vision 64, no. 2-3 (2005): 107-123.

[16] Scovanner, Paul, Saad Ali, and Mubarak Shah. "A 3-dimensional sift descriptor and its application to action recognition." In Proceedings of the 15th ACM international conference on Multimedia, pp. 357-360. ACM, 2007.

[17] Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." In Advances in neural information processing systems, pp. 568-576. 2014.

[18] Wang, Xuanhan, Lianli Gao, Jingkuan Song, and Hengtao Shen. "Beyond frame-level cnn: Saliency-aware 3-d cnn with lstm for video action recognition." IEEE Signal Processing Letters 24, no. 4 (2017): 510-514.

[19] Feichtenhofer, Christoph, Axel Pinz, and Richard Wildes. "Spatiotemporal residual networks for video action recognition." In Advances in neural information processing systems, pp. 3468-3476. 2016.

[20] Kay, Will, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola et al. "The kinetics human action video dataset." arXiv preprint arXiv:1705.06950 (2017).

[21] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15, no. 1 (2014): 1929-1958.

[22] Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. "Convolutional two-stream network fusion for video action recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933-1941. 2016.

[23] LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks 3361, no. 10 (1995): 1995.

[24] Jain, Arjun, Jonathan Tompson, Yann LeCun, and Christoph Bregler. "Modeep: A deep learning framework using motion features for human pose estimation." In Asian conference on computer vision, pp. 302-315. Springer, Cham, 2014.