Action Recognition for Wearable Devices

Smriti Gupta sgupta23@umd.edu
Yow-Ting Shiue ytshiue@umd.edu
Preyash Parikh pparikh@umd.edu
Akshay kurhade akurhade@umd.edu

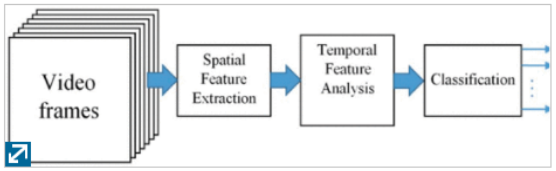
Abstract

The purpose of this work is to optimize the current action detection architectures and maintain a trade off between architectures performance and low fps for wearable devices. We will be checking current datasets with proven good performance for training deep convolutions networks. Some studies has been done over 3D architectures. Recently, convolutional neural networks with 3D kernels have been very popular in processing video related tasks. Some works have converted 2D CNN's into 3D CNN's to increase its performance over video datasets. Our work is about investigating the network modifications to maintain the performance over low fps.

1. Introduction

In the recent years, video content on Internet is growing rapidly. The main objective of video surveillance systems used in schools, prisons, psychiatric hospitals, administrative institutions, stadiums, public places, etc., is to monitor potentially various situations, e.g., abuse, assault, fighting, robbery, clash, etc., in order to resolve them. However, a large number of surveillance applications, as well need of preventing the distribution of content in the Internet is also gaining importance now. However, the procedure of action recognition content in real-world videos is almost a manual process [5]. Considering the increasing amount of both surveillance video and video on the Internet makes it hard to analyze this content manually[5]. It is necessary to develop effective action recognition models both for public safety purposes and for analyzing video content on the Internet. Currently, the area of human action recognition is not widely spread because training on video data is computationally expensive and there is not much data available. Human actions on video can be represented as spatio-temporal movements or dynamics of frame pixels, that correspond to visual objects on the video[5]. Thus, by identifying and analyzing these dynamics, it is possible to do action recognition. Human action recognition process is shown in Figure 1.[5]

The feature extraction block of the system receives video



ss

Figure 1. Human Action Recognition in Process [5]

frames and extracts spatial features of visual objects in a processing frame. This procedure can be typically performed with various spatial image descriptors: histogram descriptors, spectral descriptors, SIFT [4], SURF [5], HOG [6], CNN [7], etc. Extracted spatial features are processed in the block of temporal analysis of spatial features. This block analyzes temporal dynamics of spatial features sequence. This can be achieved with Hidden Markov Models, the analysis of the difference between two consecutive frames, recurrent neural networks, etc. Classification block performs classification of results of the temporal dynamics analysis. This operation can be performed with Support Vector Machines (SVM), neural networks, decision trees, etc. It is important to note that the feature extraction block and the block of temporal analysis can be combined into a single entity. In this case, the system performs simultaneous extraction of spatio-temporal features of the video being processed.[5]

3D CNN architectures has proven to be the best at solving general human action recognition task [6]. These neural network architectures perform both spatio-temporal feature extraction from videos and efficient classification of the extracted features. However, application of various 3D CNN architectures in the specific aggressive action recognition task has not been thoroughly studied. To train deep neural networks for classification tasks, it is necessary to use datasets, containing a large number of classes, and the classes themselves must include a large number of instances i.e., training datasets must be representative.

In this work we have used UCF-101 dataset [7] to evaluate the applicability of transfer learning for each model. Since the large video datasets became available, the primary trend for video recognition tasks is again to achieve higher accuracies by building deeper and wider architectures. Con-

sidering the fact that 3D CNNs achieve better performance for video recognition tasks compared to 2D CNNs [3], it is very likely that this 3D CNN architecture[8,9,10] search will continue until the achieved accuracies saturate. However, real-world applications still require resource efficient 3D CNN architectures taking runtime, memory and power budget into account. This work aims to fill this gap.

2. Related Work

Lately, there is a rising interest in building small and efficient neural networks [11, 12, 13, 14]. The common approaches used for this objective can be categorized under two categories: (i) Accelerating the pretrained networks, or (ii) directly constructing small networks by manipulating kernels. For the first one, [15, 16, 17, 18] proposes to prune either network connections or channels without reducing the performance of pre-trained models. Additionally, many other methods apply quantization [19, 20, 14] or factorization [19, 14, 15] for the same objective. However, our focus is on the second one for directly designing small and resource efficient 3D CNN architectures so that it can be used for wearable devices.

Current well-known resource efficient CNN architectures are all constructed with 2D convolutional kernels and benchmarked at ImageNet. SqueezeNet [1] reduced the number of parameters and computation while maintaining the classification performance. MobileNet [12] makes use of depthwise separable convolutions to construct lightweight deep neural networks. These architectures intensively make use of group convolutions and depthwise separable convolutions. Group convolutions are first introduced in AlexNet [18] and efficiently utilized in ResNeXt [21]. Depthwise separable convolutions are introduced in Xception [21] and they are the main building blocks for majority of lightweight architectures.

3D CNNs such as well-known C3D [23] require significantly more parameters and computations compared to their 2D counterparts which make them harder to train and prone to over-fitting. With the availability of large scalevideo datasets such as Sports-1M [24], Kinetics-400 [3], this problem is solved. Moreover, [25] proved that 3D CNNs achieve better accuracies compared to 2D CNNs for video classification task. Consequently, 3D CNN architecture search is an active area in research community to achieve higher accuracies.

Several 3D CNN architectures have been proposed recently Carreira et al. propose Inflated 3D CNN (I3D) [25], where the filters and pooling kernels of a deep CNN are expanded to 3D, making it possible to leverage successful ImageNet architecture designs and their pretrained models. P3D [26] and (2+1)D [27] propose to decompose 3D convolutions into 2D and 1D convolutions operating on spatial and depth dimensions, respectively. In [8], 3D versions of famous ImageNet architectures such as ResNet [9], Wide ResNet [28], ResNeXt [29] and DenseNet [30] are evaluated and it has been shown that ResNeXt achieves better results compared to others.

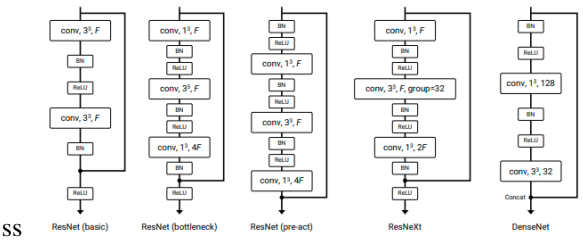
Capturing long-range dependencies is of main importance in deep neural networks. For sequential data (e.g.,in speech, language), recurrent operations [33, 23] are the dominant solution to long-range dependency modeling. For image data, long-distance dependencies are modeled by the large receptive fields formed by deep stacks of convolutional operations [14, 30]. In this paper [31], authors present non-local operations as a generic family of building blocks for capturing long-range dependencies. Inspired by the classical non-local means method [31] in computer vision, our non-local operation computes the response at a position as a weighted sum of the features at all positions. This building block can be plugged into many computer vision architectures. On the task of video classification, even without any bells and whistles, our non-local models can compete or outperform current competition winners on both Kinetics and Charades data-sets.

In this work, we are building our model over I3D convolutional network by adding some non-local layers any training the architecture for videos with low fps. As previous work in paper [31] claims that I3d with non-local layers performs better that its counterparts.

3. Approach

3.1. Video Classification Models

We focused on CNNs with 3D convolutional kernels, which have recently begun to outperform 2D CNNs through the use of large-scale video datasets. These 3D CNNs are intuitively effective because such 3D convolution can be used to directly extract spatio-temporal features from raw videos.The authors in [8] have discussed various RESNET based architectures like ResNet (basic), ResNet (bottleneck), ResNet (pre-act),ResNext, DenseNet shown in Figure 2 which is the baseline of our project.



SS

Figure 2. Block of each architectures with 3D convolutionals. F as the kernel size, and the number of feature maps of the convolutional filter are x × x × x and F, respectively. Shortcut connections of the architectures are summation except for those of DenseNet, which are concatenation [8]

3.1.1 Two-Stream Inflated 3D ConvNets

- The authors [32] in talk about Inflated ConvNets as, 2D k×k kernel can be inflated as a 3D t×k×k kernel that spans t frames. This kernel can be initialized from 2D models (pretrained on ImageNet): each of the t planes in the t×k×k kernel is initialized by the pre-trained k×k weights, rescaled by 1/t. If a video consists of a single static frame repeated in time, this initialization produces the same results as the 2D pre-trained model run on a static frame. We study two cases of inflations: we either inflate the 3×3 kernel in a residual block to 3×3×3 (similar to [25]), or the first 1×1 kernel in a residual block to 3×1×1 (similar to [33]). We denote these as I3D3×3×3 and I3D3×1×1. As 3D convolutions are computationally intensive, we only inflate one kernel for every 2 residual blocks; inflating more layers shows diminishing return. We inflate conv1 to 5×7×7. The authors of [25] have shown that I3D models are more accurate than their CNN+LSTM counterparts.

The I3D architecture is shown in Figure 3. The5 shallow layers are low-level plain 3D convolutional and max-pooling spatio-temporal feature extractors. The feature map that is obtained from the shallow layers is processed by consecutive Inception blocks and 3D max pooling layers. The output of this neural network is a three-dimensional convolutional layer of size (1x1x1). The output of this layer is flattened into a num_class dimensional vector, which is fed into a softmax classifier. This neural network allows processing video fragments of 16 frames each. The size of every single frame is 224x224 pixels. The I3D architecture model was chosen as a feature extractor [31, 5].

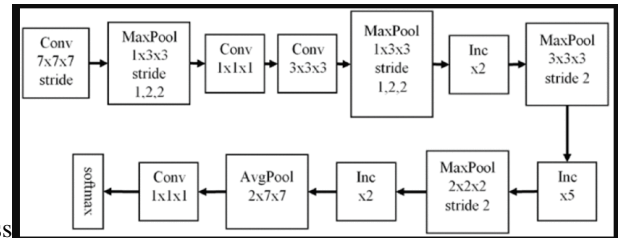


Figure 3. The architecture of I3D model[5].

3.2. Non-Local Layers

Non-local means [32] is a classical filtering algorithm that computes a weighted mean of all pixels in an image. It allows distant pixels to contribute to the filtered response at a location based on patch appearance similarity. This non-local filtering idea was later developed into BM3D (block-matching 3D) [10], which performs filtering on a group of similar, but non-local, patches. BM3D is a solid image denoising baseline even compared with deep neural networks [5]. Non-local matching is also the essence of successful texture synthesis [12], super-resolution [16, 31].

The authors in [31] defines non local blocks as:

$$z_i = W_z y_i + x_i \tag{1}$$

where x_i is the residual connection which allows us to insert a new non-local block. We inserted non-local layer blocks into I3d to turn them into non local net. These non-local I3D (NL I3D) models improve over their I3D counterparts (+1.6 point accuracy), showing that non-local operations and 3D convolutions are complementary[31].

3.3. Low rate of frames per second

To use the system for wearable devices we aim to reduce the FPS of the videos we are using to train the model. By modifying the architecture of I3D with the addition of non-local layers, we aim to maintain a trade-off between system's performance and low fps.

4. Experiments and Results

4.1. Experiment 1

First we use pretrained model ResNet 18 and checked the accuracy degradation with reducing FPS show in the below table. From the results we obtained, it was clear that accuracy does not change much with the decreasing FPS.

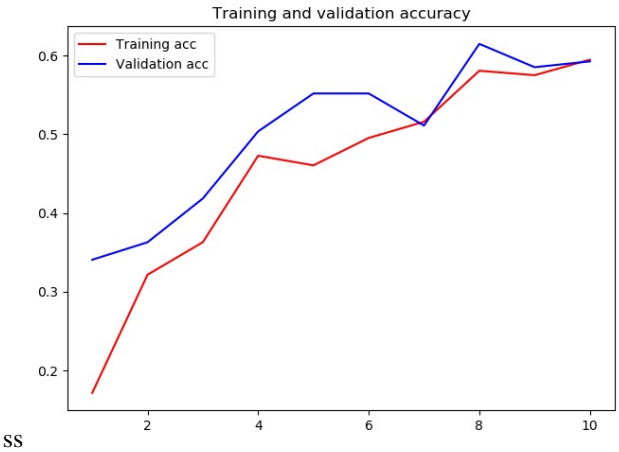
Accuracy with decreasing FPS

train fps	test fps	top1 acc
25	25	0.871531
25	20	0.869680
25	15	0.866508
25	10	0.853555
20	25	0.871002
20	20	0.871531
20	15	0.869416
20	10	0.855406
15	25	0.869416
15	20	0.870209
15	15	0.868623
15	10	0.859107
10	25	0.858578
10	20	0.860428
10	15	0.865451
10	10	0.859107

4.2. Experiment 2

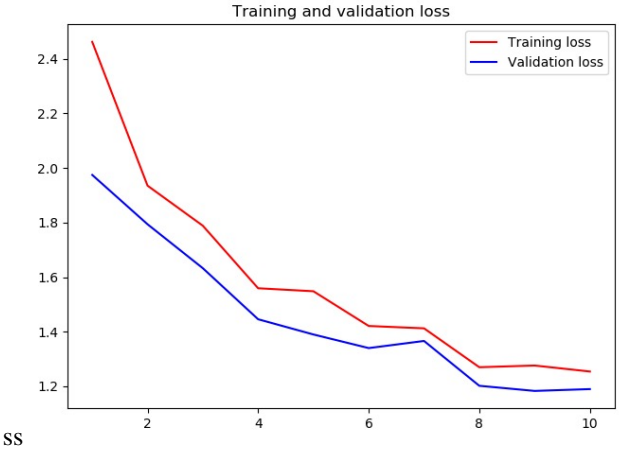
A few non-local layers were added in our existing network to check the performance for which we have the baseline model used was ResNet18 on the UCF-101 dataset reduced to 20fps. On adding non-local layer with last 3d convolutional layer of the model we achieved an accuracy of '0.823164'. Further, on adding non-local layer to all 4 layers(ResNet 18), we got an accuracy of '0.714236' on the training dataset. Tuning the hyper-parameters may have

yielded a better result. It can be inferred from the validation loss plots that the curves are yet to be saturated. If trained for higher epochs would yield better results which was not possible due to time and computational constraints.



ss

Figure 4. Training and validation accuracy



ss

Figure 5. Training and validation loss

4.3. Experiment 3

For the third experiment, we used non-local neural network with a base of ResNet18 pre-trained on the dataset hmdb51. We tested the model on UCF-101 with reduced fps in train and test datasets to 20fps. The accuracy obtained was 0.7326.

5. Conclusion

After this experiments we can conclude that reducing FPS does not affect the results to a noticable extent and can be used to reduce the bandwidth and space-time complexity for mobile applications. Addition of non-local layers did provide a boost but no conclusive results could be inferred, although the plots indicate that with better hyper-parameters and a bigger dataset better performance could have been achieved. We attempted to do an ensemble of pre-trained RESNET-18 model and Non-local NN (Base ResNet-18) but could not train it due to time constraint.

For future work, we plan to incorporate a dual stream I3D convolutional network to the existing ensemble, and also hope to achieve better results by optimizing the non-local layers in conjugate with the existing networks.

References

[1] O. Köpüklü, N. Kose, A. Gunduz and G. Rigoll, "Resource Efficient 3D Convolutional Neural Networks," 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019, pp. 1910-1919. doi: 10.1109/ICCVW.2019.00240 2

[2] C. Li, L. Zhu, D. Zhu, J. Chen, Z. Pan, X. Li, et al., "End-to-end Multiplayer Violence Detection based on Deep 3D CNN", Proceedings of the 2018 VII International Conference on Network Communication and Computing, pp. 227-230, 2018.

[3] A. Saif, M. Khan, A. Hadi, R. Karmoker and J. Gomes, "Aggressive Action Estimation: A Comprehensive Review on Neural Network Based Human Segmentation and Action Recognition", International Journal of Education and Management Engineering, vol. 9, no. 1, pp. 9-19, 2019.

[4] Joao Carreira,Andrew Zisserman†,Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,DeepMind Department of Engineering Science, University of Oxford,arXiv:1705.07750v3 [cs.CV] 12 Feb 2018

[5] A. Saveliev, M. Uzdiaev and M. Dmitrii, "Aggressive Action Recognition Using 3D CNN Architectures," 2019 12th International Conference on Developments in eSystems Engineering (DeSE), Kazan, Russia, 2019, pp. 890-895, doi: 10.1109/DeSE.2019.00165. 2

[6] G. Yao, T. Lei and J. Zhong, "A review of Convolutional-Neural-Network-based action recognition", Pattern Recognition Letters, vol. 118, pp. 14-22, 2019.

[7] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012. 1

[8] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and

imagenet? In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 6546–6555, 2018. 1, 3

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 1

[10] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri. Convnet architecture search for spatiotemporal feature learning. arXiv preprint arXiv:1708.05038, 2017. 1

[11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360, 2016 2

[12] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. 2

[13] Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. arXiv preprint arXiv:1807.11164, 5, 2018. 3

[14] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4820–4828, 2016 2

[15] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149, 2015 2, 3

[16] . Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In Advances in neural information processing systems, pages 1135–1143, 2015. 2

[17] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In Advances in neural information processing systems, pages 2074–2082, 2016. 2

[18] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440, 2016. 2

[19] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnornet: Imagenet classification using binary convolutional neural networks. In European Conference on Computer Vision, pages 525–542. Springer, 2016. 2, 3

[20] D. Soudry, I. Hubara, and R. Meir. Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In Advances in Neural Information Processing Systems, pages 963–971, 2014. 2

[21] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated ´ residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1492–1500, 2017. 2

[22] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1251–1258, 2017. 2

[23] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 4489–4497, 2015. 2

[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1725–1732, 2014. 2

[25] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6299–6308, 2017.

[26] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In proceedings of the IEEE International Conference on Computer Vision, pages 5533–5541, 2017. 2

[27] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 6450– 6459, 2018. 2

[28] S. Zagoruyko and N. Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016. 2, 3

[29] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated ´ residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1492–1500, 2017. 2

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017 2

[31] Xiaolong Wang1,Ross Girshick, Abhinav Gupta, Kaiming He, Non-local Neural Networks ,arXiv:1711.07971v3 [cs.CV] 13 Apr 2018 2

[32] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In Computer Vision and Pattern Recognition (CVPR), 2005. 1, 2, 3 2

[33]] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In Neural Information Processing Systems (NIPS), 2016. 2, 4 2

2, 3

3

2, 3