CS230

Human Activity Recognition for Healthcare Applications

Reetika Agarwal
SCPD
Stanford University
reetika@stanford.edu

Myra Kurosu Jalil
Dept of Mechanical Eng
Stanford University
mkurosu@stanford.edu

Rajorshi Paul
Dept of Mechanical Eng
Stanford University
rajorshi@stanford.edu

Abstract

Every year, more than 25% of the elderly population in the United States experience debilitating injuries, such as falls, which can lead to broken bones and major injuries. These injuries may go unreported, and the compounding health effects can significantly deteriorate quality of life [1]. Alternatives such as constant monitoring by trained personnel is impractical due to the cost, time, and restrictions this places on an individual's living accommodation. These issues motivate the development of a remote health monitoring system that tackle problems ranging from logging patient vitals to identifying high-risk accidents.

We train C3D, ResNet-18, ResNet-50, ResNet-101, ResNeXt-101 models on 9 healthcare-related actions from the NTU-RGB+D dataset. When the models are trained from scratch, the validation accuracy saturates at 53% for ResNet-50, ResNet-101 and ResNeXt-101. However, models pre-trained on Kinetics-600 have sufficient information to further train the deeper layers and can match existing state-of-the-art systems (84%). We further show that for real world deployment we can trade-off accuracy for resource efficiency, i.e., MobileNetV2 achieves 7% decrease in accuracy for 36x fewer paramters and 25x fewer floating point operations. Our work is publicly available and can be downloaded here: https://github.com/reetikaag/human-activity-recognition

1 Introduction

1.1 Description

Remote health monitoring can allow healthcare facilities, such as hospitals or assisted living facilities, to monitor patient health from a distance by automatically sending relevant data to healthcare providers. This can reduce the need for at-home caretakers and ensure immediate response in case of emergencies. Such a healthcare monitoring system would classify the actions of the patient, create logs of symptoms throughout the day, and alert the healthcare provider in the event of an anomaly. In this project, we train a neural network with nine medically-relevant actions for action recognition in patients. The input to our neural network is a video clip of a human action, and the output is a prediction of the action class (e.g. sneeze/cough, chest pain, falling down, etc).

1.2 Key Ideas

We explore three main trends in video classification research:
(i) 3D CNNs have been shown to outperform 2D CNNs as a natural way to encode spatiotemporal

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

information [2]. For the action classification task, we primarily focus on 3D CNNs which span from relatively shallow (C3D) to very deep networks (ResNet 3D/ResNeXt 3D).
(ii) Previous successes of 2D CNNs are owed to very deep networks that were pre-trained on ImageNet. Recently, [3] showed that similar pre-training of very deep 3D CNNs on large scale video datasets, such as Kinetics-600, can effectively train a neural network with a video representation. We build on their work by using deep 3D CNN architectures that are pre-trained on closely-related action classes.
(iii) 3D CNNs are computationally expensive due to the additional kernel dimension. In addition to classification accuracy, we use resource efficiency as a metric to compare different architectures. We evaluate pre-trained resource efficient 3D CNNs [4] such as MobileNetV2 [5], ShufflenetV2 [6] and SqueezeNet [7], which have fewer FLOP (floating point operations) than ResNets, yet can provide reasonable accuracy.

1.3 Challenges

The action classification task is challenging for many reasons. Firstly, some of the nine actions have subtle differences and closely resemble each other. An example is shown in Figure 2, where the action classes headache, chest pain and neck pain vary only in terms of a slight shift in hand placement. Secondly, the models are constrained to predict activity using only raw RGB images. The top three benchmarks on NTU-RGB+D dataset combine pose/raw depth and RGB images to boost performance [8]. Since such pose/depth information may not be always available and adds additional constraints to training and deployment, we limit our methods to rely only on a single modality.

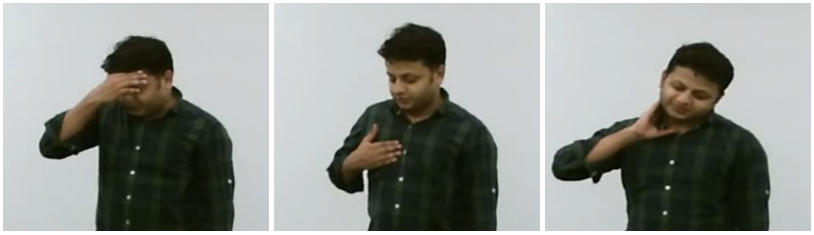


Figure 1: Sample frames from the NTU RGB dataset - headache, chest pain, neck pain

2 Related Work

Two-stream 2D CNNs using both RGB and optical flow was a popular method in many earlier works, owing to pre-training on ImageNet [9, 10]. Jie *et al.* proposed using 3D CNNs [11] for extracting spatiotemporal features and Tran *et al.* trained a 3D CNN architecture, known as C3D [12], on the sports 1M dataset. Carreira *et al.* introduced the Inflated 3D CNN (I3D), [2] achieving state-of-the-art performance in some benchmarks. For deep 3D CNN architectures, Hara *et al.* extended famous ImageNet architectures such as ResNet [13], Wide ResNet [14], and ResNeXt [15] to their 3D counterparts [3].

3 Dataset

For the proposed healthcare application, we acquired permission to use the NTU RGB+D dataset from the Rapid-Rich Object Research Lab (ROSE) at the Nanyang Technological University, Singapore [16]. This dataset consist of 114,480 video samples for 60 action classes. We focus on a subset that consists of 9 medically-relevant action classes. These action classes are: sneezing/coughing, staggering, falling down, headache, chest pain, back pain, neck pain, nausea/vomiting and fanning self. The ROSE dataset consists of 8532 1920x1080 RGB videos. Due to the small set of examples, we decided to omit the test set and perform an 80/20 training/validation split (6825 training/1707 validation videos). Stratified sampling is used to eliminate any sampling bias.

3.1 Data pre-processing

All videos are pre-processed to split them into individual jpg frames and scale the individual frames down from 1920x1080 to 427x240 pixels. The size of each video sample is 3 channels x N frames

x 240 pixels x 427 pixels, where N is number of frames per video and can vary between 33 to 222. Both spatial and temporal pre-processing are performed on the dataset. For spatial pre-processing, all frames of the video are center-cropped to 240 x 240 pixels to remove unnecessary background information. To prevent over-fitting, we randomly select a spatial scale from [1.0, 0.97, 0.94, 0.91, 0.88] in order to perform multi-scale cropping, as was done in [17]. A scale of 1 means that the output is 240 x 240 pixels, whereas a scale of 0.88 means the output size is 211 x 211 pixels. The scales are chosen by manually analyzing 100 video samples to ensure that cropping does not lead to removal of the target subject. Post-cropping, we spatially resize the resultant images to 112 x 112 pixels using bilinear interpolation. Finally, the input is normalized using the mean and standard deviation values of the ActivityNet dataset [18] for each color channel, and each sample is horizontally flipped with 50% probability during training.

Next, we perform temporal pre-processing to reduce the number of frames from 33-222 to 16. Since the action most distinctly occurs in the middle of the video, we pick 32 frames from the center of each video clip and then down-sample by 2 to generate the 16 frames. Looking at a 32-frame window is critical to capturing the full content of longer actions, such as staggering and falling down. The final size of the pre-processed video is 3 channels x 16 frames x 112 pixels x 112 pixels.

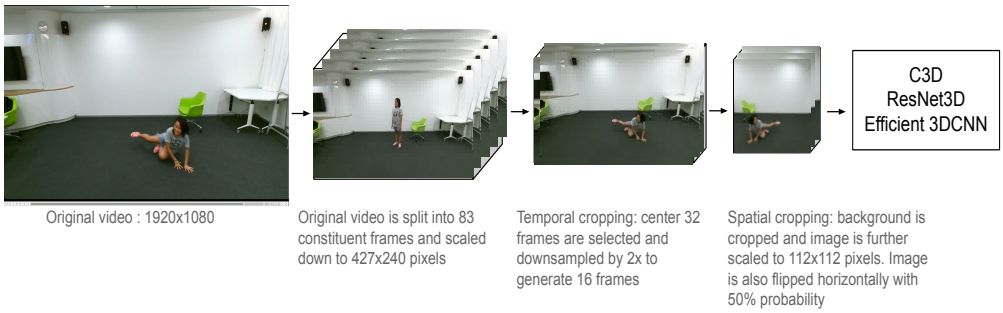


Figure 2: Data pre-processing pipeline

4 Methods

4.1 Network Architecture

We implemented three state-of-the-art models, which are described below. The detailed structure of all models are included in the github repository.

4.1.1 C3D

C3D [12] is widely seen as a de-facto standard for 3D CNNs. Given its popularity and widespread adoption, it is included as a benchmark for comparison with the other models. The architecture has 11 layers with 8 convolutions, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer with 9 neurons. Similar to [2], we used batch normalization after all convolutional layers to improve performance.

Figure 3: C3D architecture.C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer.

4.1.2 ResNet/ResNeXt 3D

ResNet 3D/ResNeXt 3D [19] is a deep architecture consisting of 18, 50, 101, 152 or 200 layers that has shown to outperform C3D, P3D, two-stream I3D [2, 20] and many other 2D CNN models in

action recognition tasks [3]. ResNets have skip connections, which allows for very deep networks. In this project, we explore the importance of deep networks by comparing the ResNet models with 18, 50 and 101 layers. For ResNext, we use 101 layers and a cardinality of 32 to compare it against ResNet-101.

4.1.3 Efficient 3D CNNs

These are light-weight deep 3D CNNs architectures that use group convolutions [21] and/or depthwise separable convolutions [22] to reduce computations. In this project, we focus on MobileNetV2, ShuffleNetV2 and SqueezeNet architectures, since these are deeper models and are expected to perform better than shallower resource-efficient models [4].

4.2 Implementation Details

Training: For all training, we use Stochastic Gradient Descent (SGD) with categorical cross-entropy loss. We choose a mini-batch size of 64 videos to balance memory usage and speed of training. Given the GPU limitations and large numbers of planned experiments, we limit the number of training epochs to 50, even though training for more epochs may improve the performance. The momentum, dampening and weight decay are set to 0.9, 0.9 and 1×10^3, respectively. We experimented with learning rates and chose 0.001 for the first 40 epochs until the validation loss saturates, and reduce it by a factor of 10 for the last 10 epochs. We compare results of both training from scratch and fine-tuning models pre-trained on Kinetics-600. For fine-tuning, we freeze the original network parameters and fine-tune only on the last fully connected layer.

Validation: For the validation set, we apply center crop to obtain a square image, rescale the image to 112 x 112 pixels, normalize and temporally crop the image to extract the 32 center frames, followed by downsampling by 2x. For all models, we report top 1 accuracy and top 2 accuracy, MFLOPs (floating points operations in units of 1×10^6) and number of trainable parameters.

5 Experiments and Results

Pre-training with representative datasets: In order to understand the effects of pre-training on our dataset, each model is both trained from scratch and fine-tuned, with pre-training on Kinetics-600 dataset. The results show significant improvement when pre-trained models are used. The accuracy of ShuffleNetV2 is improved by 19% and the accuracy of ResNet-101 is improved by 31%. This result replicates the result in [3] that pre-training on megascale video datasets allows spatio-temporal 3D CNNs to retrace the success of ImageNet and 2D CNNs.

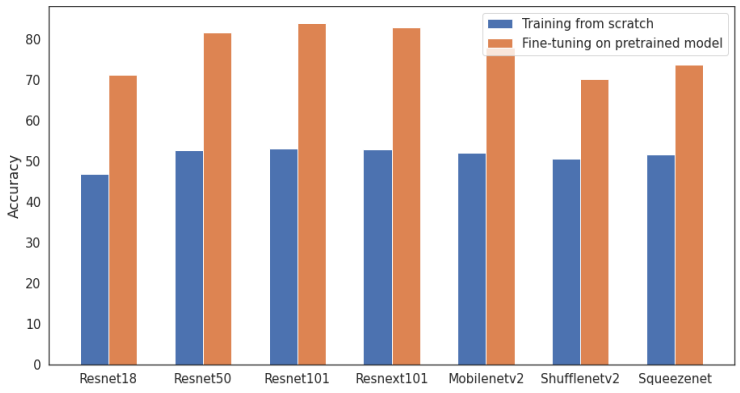


Figure 4: Accuracy comparisons between all models - trained from scratch vs pre-trained

Effect of model layers: We validate the relationship between the number of layers in ResNet and the video recognition accuracy. Table 1 shows that the accuracy of ResNet 3D improves by 12.5% as the model depth increases from 18 to 101, indicating that deeper CNNs perform better.

Accuracy vs model complexity: Table 1 shows that the architectures with more parameters and FLOPs, such as ResNet and ResNeXt-101, generally achieve higher accuracies. The exception is

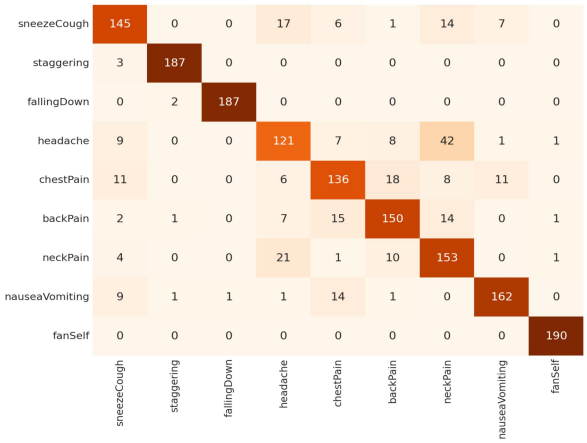
C3D, which achieves poor accuracy, due to the lack of pre-training. MobileNetV2 has inverted residual blocks, which excel at capturing dynamic motions, leading to its superior performance when compared to ShuffleNetV2 and SqueezeNet [5]. MobileNetV2 also achieves better performance than ResNet-18, despite having 14x fewer parameters and FLOPs. The results show that MobileNetV2 and SqueezeNet are better choices for applications that need light-weight neural network, since they provide lower complexity for a slight reduction of 6% in accuracy, compared to ResNet-101.

	Number of				Accuracy (%)	
Model	Layers	Non-linearities	Params	MFLOPs	Top 1	Top 2
C3D	11	10	78M	33164	51.6	69.8
ResNet-18	18	9	33.03M	8323	71.3	85.4
ResNet-50	50	17	46.22M	10128	81.5	93.0
ResNet-101	101	34	85.26M	13957	**83.8**	**94.2**
ResNeXt-101	101	34	47.54M	9650	83.0	94.4
3D-MobileNetV2	53	35	2.37M	560	77.8	90.8
3D-ShuffleNetV2	51	34	1.31M	194	70.2	86.8
3D-SqueezeNet	18	18	1.84M	921	73.6	89.6

Table 1: Accuracy and complexity comparison of 3D CNN models

Precision, Recall and Confusion Matrix: Figure 5 shows the confusion matrix for the best performing model, ResNet-101. As shown in Figure 1, the neck pain and headache actions closely resemble each other, and Figure 5 confirms that the model confuses between these two actions. In contrast, the falling down, staggering and fan-self actions have very distinct signatures, and the model rarely mis-classifies them.

In addition to accuracy, having a low recall is crucial to avoid missing an adverse health-related event. The more alarming actions of falling down/staggering have a very good recall at 98%, whereas chest pain has a recall of 72%.



	Precision	Recall
SneezeCough	79%	76%
Staggering	98%	98%
Falling down	99%	99%
Headache	70%	64%
Chest pain	76%	72%
Back pain	80%	79%
Neck pain	66%	81%
Nausea-Vomiting	90%	86%
Fanself	98%	100%

Figure 5: Confusion matrix, precision and recall of all action classes

Saliency Maps: We use saliency maps to visualize the parts of each image that maximize the class score. Figure 6 shows the original image, where we first average all the RGB channels, followed by an averaging over the 16 frames of the video. Below that is the corresponding saliency map which is also an average of the saliency map of all 16 frames. Localized actions such as sneeze-cough and fan-self show localized heat-map, whereas for more dynamic actions such as staggering, the importance is spread over more pixels.

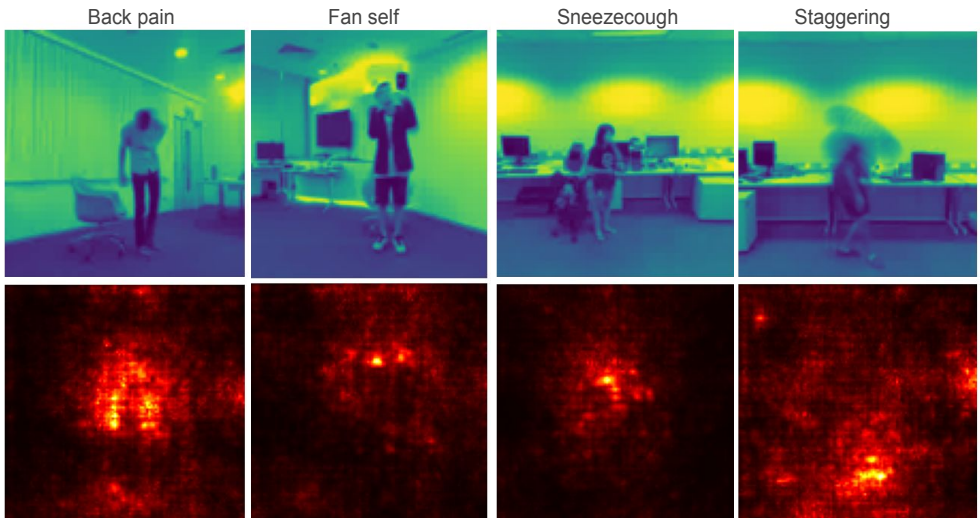


Figure 6: Saliency maps to visualize contribution of different parts of image to the class score

6 Discussion and Conclusion

We demonstrate the benefits of resource-efficient architectures, pre-trained 3D CNNs, and strategic pre-processing for action recognition task. Despite the subtle differences in some of the actions, our best-performing model, ResNet-101, achieves 84% accuracy on the validation set. The best-performing resource efficient model, MobileNetV2, achieves 78% accuracy with 36x fewer parameters and 25x fewer FLOPs than ResNet-101. Lastly, we achieve 50% top-1 accuracy and 67% top-2 accuracy on our own dataset.

7 Contribution

Reetika was responsible for running Efficient 3D CNN and C3D, and data pre-processing. Myra and Raj were responsible for running ResNet 3D, and collecting and testing on our dataset. All members contributed to the report and video. We would like to thank CS230 TAs for their guidance.

References

[1] CDC. Important Facts about Falls, 2017, Centers for Disease Control and Prevention. `https://www.cdc.gov/homeandrecreationalsafety/falls/adultfalls.html`, 2017.

[2] Joao Carreira and Andrew 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.

[3] Kensho Hara, Hirokatsu Kataoka, and Yutaka 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.

[4] Okan Köpüklü, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. Resource efficient 3d convolutional neural networks. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1910–1919. IEEE, 2019.

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[6] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[7] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[8] Action Recognition on NTU RGB+D. `https://paperswithcode.com/sota/action-recognition-in-videos-on-ntu-rgbd`.

[9] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

[10] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4768–4777, 2017.

[11] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[12] Tran Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[15] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[16] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019.

[17] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015.

[18] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.

[19] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[20] deepmind. kinetics-i3d. `https://github.com/deepmind/kinetics-i3d`, 2018.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

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