LEARNING DEEP AND COMPACT MODELS FOR GESTURE RECOGNITION

Koustav Mullick and Anoop M. Namboodiri

Center for Visual Information Technology (CVIT),
International Institute of Information Technology, Hyderabad, India.

ABSTRACT

We look at the problem of developing a compact and accurate model for gesture recognition from videos in a deep-learning framework. Towards this we propose a joint 3DCNN-LSTM model that is end-to-end trainable and is shown to be better suited to capture the dynamic information in actions. The solution achieves close to state-of-the-art accuracy on the ChaLearn dataset, with only half the model size. We also explore ways to derive a much more compact representation in a knowledge distillation framework followed by model compression. The final model is less than 1 MB in size, which is less than one hundredth of our initial model, with a drop of 7% in accuracy, and is suitable for real-time gesture recognition on mobile devices.

1. INTRODUCTION AND RELATED WORK

Gesture recognition is one of the key components in natural human-computer interfaces, especially for mobile devices, where interfaces like keyboard and mouse are impractical. However, the problem is challenging due to several factors like changes in background, lighting and variations in the performance and speed of the gestures. The users may have different appearance, pose and positioning relative to the camera as well as differences in the way they perform any specific gesture. In this work, we develop a solution that is accurate and robust enough to handle these variations, while being compact and fast to suit resource limited devices.

Deep Learning (DL) has had a major impact in the fields of computer vision, natural language processing and speech recognition over the past few years. However, one of the major issues with DL models are the large number of parameters involved which makes training time and memory requirements quite high. This makes it difficult to employ them in embedded devices. Keeping this in mind, we would like to explore ways to reduce the parameter space of such models without compromising a lot on performance.

1.1. Action Recognition in Videos

Human action in video sequences consists of spatial information within individual frames as well as temporal information of motion across the frames. CNN models are very good at capturing the spatial information within an image as demonstrated by their success on ImageNet classification. Previous works have tried to follow two approaches in integrating temporal information into these frameworks. The first one is to extract features from videos that capture temporal information into images and then use image classification models. These include ideas such as dynamic image [1] that produces a single image from a video obtained by applying rank pooling on the raw image pixels of a video, or the use of optical flow images [2].

A second approach is the use of deep-learning models that are better suited for capturing temporal information such as 3D-Convolutional Neural Network (3D-CNN) [3] proposed by Ji *et al.*, as well as Long Short Term Memory (LSTM) recurrent networks [4], which can use either pre-computed image features such as Bag of words and SIFT [5], regular CNN features including optical-flow [6] or 3D-CNN features [7]. Tran *et al.* [8] proposed a simple, yet effective approach for spatio-temporal feature learning using deep 3D-CNN trained in a supervised fashion on a large scale video dataset. [6] performed activity recognition by passing RGB and optical flow frames through pre-trained CNN networks and fed the learned features to an LSTM for final recognition. Varol *et al.* [9] learned video representation using 3D-CNN and demonstrated the impact of different low-level features as input.

Most of the works discussed above either use pre-trained CNN models for feature extraction or trains the CNN on parts of videos, separately from the model that performs actual classification (like LSTM or SVM). We would like to develop an end-to-end trainable model that can automatically learn both spatial and temporal features in a complete gesture. Towards this we combine the strengths of 3D-CNNs that are good at capturing spatial and short-term temporal features and LSTMs that can capture long-term temporal signatures of actions.

1.2. Knowledge Distillation

The second aspect that we explore in this work is to develop models that are efficient in terms of memory and speed. Knowledge distillation, originally introduced by Caruana *et al.* [10], introduced a process of transferring knowledge from

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

one model to another. In order to obtain a faster inference, Hinton *et al.* [11] proposed a compression framework which trains a student network, from the softened output of a larger model or an ensemble of wider networks, called teacher network. The idea is to allow the student network to capture not only the information provided by the true labels, but also the finer structure learned by the pre-trained teacher network. This approach has been used for speech recognition [12], image classification [13], and network compression [11].

In this work we show that the use of 3D-CNN with LSTM can improve the accuracy of action recognition in videos without increasing the model size, while the use of knowledge distillation allows us to develop a compact model that can be further compressed with minimal impact on accuracy.

2. OUR APPROACH

We describe our framework in three stages. As baseline models we use a 3D-CNN and an LSTM variant of RNN to classify each gesture. For 3D-CNN each gesture video needs to be represented using a fixed number of frames. For LSTM the raw frames are flattened and passed through a linear layer to obtain features that are then fed to the LSTM.

Next we present a framework that combines the 3D-CNN with LSTM. The 3D-CNN acts as an encoder at sub-gesture level and provides representations for groups of few frames. These are then fed as sequences to the LSTM to get the final prediction. This can be thought of as learning a two-step hierarchy, where gestures are made up of sub-gestures. The 3D-CNN learns representations at sub-gesture level followed by the LSTM which makes prediction by building on the sub-gesture features. As many gestures have considerable overlap in their sub-gestures (raising or lowering of hand), having a two-level learning of temporal evolution of the gesture helps in better modeling the gesture. Also, all gestures will not span across the same number of frames. This is a limitation while using a CNN for classification as it requires a fixed length input. A recurrent network allows us to process arbitrary length videos.

Finally, we use our trained baseline CNN as a teacher and use its softmax output to train much smaller variants of our joint CNN and LSTM models. The benefits of this approach is demonstrated in the experiments section. In the subsequent sub-sections we detail out the architectures used in each of the aforementioned approaches.

2.1. Baseline 3D-CNN

Each gesture is represented using 32 consecutive frames, each of dimension 64×64 and two channels: depth and grayscale. For videos longer than 32 frames, the central 32 frames are extracted, while shorter videos are zero-padded. The model consists of six convolution layers, followed by two fully-connected layers (FCs). Each layer is followed by ReLU activation. Convolution layers are additionally followed by batch normalization. We do not use any pooling layer and sub-sampling is performed using padded convolution by varying the strides. For down-sampling by a factor of two, convolution is performed with filter size 4, stride 2 and padding 1 on each side. Whereas, to keep the size constant during convolution we use filter size 3, with stride 1 and padding 1 on each side. Figure 1 illustrates the model.

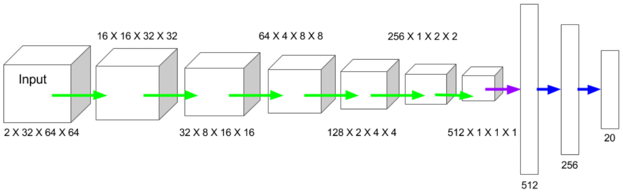

Fig. 1. Baseline 3D-CNN architecture. Each block shows dimensions in the format (*channels* × *number of frames* × *height* × *width*). Green: Conv layers, Purple: Flattening, Blue: FC layers.

2.2. Baseline LSTM

Each video is divided into chunks of 4 frames of dimension 64×64, for grayscale and depth channels respectively. After flattening, it is passed through a 512 dimensional linear layer, followed by ReLU activation. Sequences of 512 dimensional feature vectors are fed into the LSTM, containing 256 units, for final classification.

2.3. Joint 3D-CNN and LSTM

The joint model consists of an encoder 3D-CNN and an LSTM for sequence classification. The encoder takes blocks of $4 \times 64 \times 64$ frames of two channels (depth and grayscale) as input and produces a feature vector of it. The LSTM takes those vectors sequentially for an entire gesture and gives the final prediction. Architecture of the 3D-CNN is similar to the one used in our baseline model, except the number of frames in each block (4 *versus* 32), and the LSTM consists of 256 units. The complete model is shown in Figure 2, which is trained end-to-end.

2.4. Knowledge Distillation from Baseline 3D-CNN Model to Joined Model

We train our baseline 3D-CNN model till convergence and obtain the softmax output for each training sample. As suggested in [14], for each gesture we obtain softened version of softmax using:

$$P_i = \frac{e^{\frac{z_i}{T}}}{\sum_{j=1}^{c} e^{\frac{z_j}{T}}}, \forall i \in \{1, ...c\}, \tag{1}$$

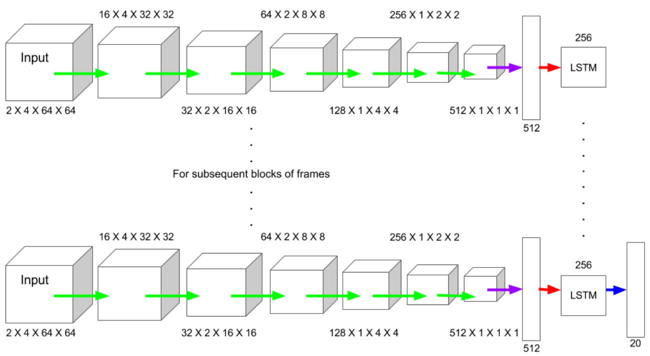

Fig. 2. Joined 3D-CNN and LSTM architecture (best viewed in color). Each block shows dimensions in the format (*channels* $\times$ *number of frames* $\times$ *height* $\times$ *width*). Green: Conv layers, Purple: Flattening, Red: LSTM connection at each time-step, Blue: FC layers.

where c is the number of classes and T is the temperature, set depending on how "soft" we want the distribution to be.

We train smaller variants (referred to as *medium* and *small*) of our joint model obtained by reducing the number of feature maps (channels) in each layer by two and four times, respectively. The aim is to minimize the following loss function:

$$L = \alpha L^{(soft)} + (1 - \alpha) L^{(hard)}, \tag{2}$$

where $L^{(soft)}$ is the cross-entropy loss between pre-trained teacher's and student's softened softmax output, $L^{(hard)}$ is the cross-entropy loss between the actual class label and model output, and α is a weighting parameter (set as 0.5 in our experiments).

3. EXPERIMENTS AND RESULTS

3.1. Dataset

The Chalearn 2014 Looking at People Challenge (track 3) [15] dataset is a vocabulary of 20 different Italian cultural/ anthropological signs, performed by 27 different users with variations in surroundings, clothing, lighting and gesture movement. The dataset consists of 7,754 gestures for training, 3,362 gestures for validation and 2,742 gestures for testing. The videos are recorded using Microsoft Kinect and consist of RGB, depth, user mask and skeleton/joint information for each frame of video as shown in Figure 3.

Each video may contain multiple gestures, performed one after another. The training and validation data consists of soft segmented annotation for the gestures in the videos, but no such annotation is available for the testing data. Since the main aim of this work is gesture classification, not segmentation, we use the validation data as our testing data and report accuracies on the same here-on. To maintain parity, we compare against the accuracies obtained on the validation data by other methods also. Further, we randomly choose 2,000 videos from training set to act as our validation data.

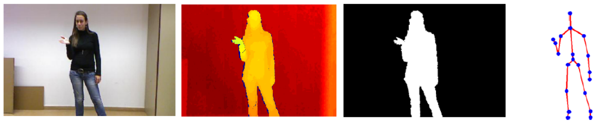

Fig. 3. Example frame modalities from the dataset.

Input mode	Accuracy(%)
Hand	87.33
Upper-body	87.59
Combined	90.13

Table 1. Baseline 3D-CNN on different input modalities.

3.2. Input

For each video frame we use the depth and convert the RGB frames to grayscale and concatenate them to obtain two-channel inputs for our models. Since all the gestures occur in the upper-body region, we extract it using the skeleton information and resize it to 64×64. Thus inputs to our models are of dimension $2 \times T \times 64 \times 64$, where T is 32 for the baseline 3D-CNN model and 4 for the joint model (as mentioned in section 2.1 and 2.3 respectively).

Similar to [17], using skeleton information, we crop out the highest hand region for each gesture, resize to 64×64 and use them as input. Table 1 shows accuracies obtained with our baseline 3D-CNN model while using hand, upper-body and both combined, as input. Since the combination of upper-body and hand gives the best performance, all following experiments use that as input (Figure 4). We also perform rotation, translation and zooming on the frames for data augmentation.

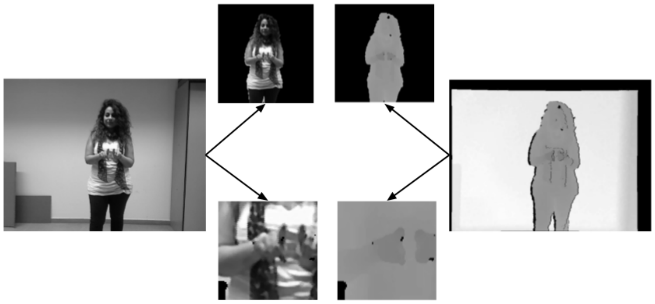

Fig. 4. Left and right images are example grayscale and depth frames from the dataset. Center rows show the upper-body and hand input frames for our models obtained from it.

	Model	# of parameters (in millions)	Trained using	Accuracy(%)
Original	3D-CNN + LSTM	18.37	class labels	93.18
Teacher	3D-CNN	18.82	class labels	90.13
Student	3D-CNN + LSTM *(medium)*	**4.59**	class labels	86.18
			class labels and softmax output of *teacher*	**88.35**
	3D-CNN + LSTM *(small)*	**1.15**	class labels	81.50
			class labels and softmax output of *teacher*	**86.05**

Table 3. Knowledge Distillation from baseline 3D-CNN to CNN + LSTM.

Method	**# of parameters** (in millions)	**Single-precision**		**Half-precision**	
		Model size (MB)	**Accuracy(%)**	**Model size** (MB)	**Accuracy(%)**
1. *Teacher* 3D-CNN	18.82	72	90.13	36	89.5
2. *Original* 3D-CNN + LSTM	18.37	71	93.18	35.5	93.18
3. *Student* 3D-CNN + LSTM	1.15	4.5	86.05	2.25	85.98
4. **Sparse model of (3)**	**0.25**	**1.12**	**86.05**	**0.635**	**85.98**

Table 4. Reduction is size along with performance impact of the student model and sparse model.

Method/Model	**Accuracy(%)**
Baseline LSTM	86.6
Baseline 3D-CNN	90.1
3D-CNN + LSTM (ours)	**93.2**
Wu *et al.* [16]	87.9
Pigou *et al.* [17]	91.4
Neverova *et al.* [18]	**96.8**

Table 2. Accuracies obtained using our model compared with *state-of-the-art* methods.

3.3. Results and Analysis

Table 2 shows the accuracies obtained using our models. The baseline LSTM performs the worst since the spatial and temporal information are not conserved while flattening out. Baseline 3D-CNN preserves such information across dimensions and performs better, but it is not suitable for capturing long-term dependencies. The best accuracy is obtained using the joint model which takes advantage of both 3D-CNN, for feature learning, and LSTM, for classification.

We compare our CNN + LSTM model against other *state-of-the-art* methods on the same dataset in Table 2. Wu *et al.* [16] described a novel method called Deep Dynamic Neural Networks, a semi-supervised hierarchical dynamic framework based on a Hidden Markov Model (HMM) and learned high-level spatio-temporal representations using a Deep Belief Network (DBN) and 3D-CNN, suited to the input modality. Pigou *et al.* [17] used an approach similar to our baseline 3D-CNN approach, but their architecture is very different from our CNN. ModDrop [18] used a multi-scale and multimodal deep learning approach. It performed careful initialization of individual modalities and fused multiple modalities at several spatial and temporal scales. Their architecture is a big cascade having individual branches for each hand, each modalities, and articulated pose based features. Our simpler architecture works directly with the raw pixels, without the need to perform any pre-training or initializations.

The effect of Knowledge Distillation is analyzed in Table 3. Reducing the number of parameters limits the model's capacity to learn from the class labels directly, but it can be alleviated when trained according to Equation 2, using soft outputs from a larger pre-trained teacher network. We get almost at-par accuracy with the teacher, but at a much reduced training cost, making it both computationally and memory efficient. Further, training with Adam optimizer makes most of the parameters of the student have very low weights, which allows for further compression to obtain a sparse model. Removing weights having magnitude below 2^{-100} got rid of $\sim 905K$ parameters out of $1.15M$, of our small student network with no drop in performance. Table 4 compares this sparse model with our other models.

4. CONCLUSION

We show how joint 3D-CNN and LSTM model for gesture recognition from videos, leverages the best of both 3D convolution and recurrent network to model the sequential evolution of information in a video, while allowing to process arbitrary length videos.

We also show how information can be distilled from a larger model to models with $16\times$ and $4\times$ fewer parameters. Size of models could be further reduced using a sparse representation as discussed above. This not only benefits training time but also makes it possible to use them in low-memory and low-power embedded devices.

5. REFERENCES

[1] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould, "Dynamic Image Networks for Action Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[2] Karen Simonyan and Andrew Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *Neural Information Processing Systems*, 2014.

[3] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3D Convolutional Neural Networks for Human Action Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[4] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 1997.

[5] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt, "Action Classification in Soccer Videos with Long Short-term Memory Recurrent Neural Networks," in *International Conference on Artificial Neural Networks*, 2010.

[6] Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.

[7] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt, "Spatio-Temporal Convolutional Sparse AutoEncoder for Sequence Classification," in *British Machine Vision Conference*, 2012.

[8] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," in *International Conference on Computer Vision*, 2015.

[9] Gül Varol, Ivan Laptev, and Cordelia Schmid, "Long-term Temporal Convolutions for Action Recognition," *arXiv preprint arXiv:1604.04494*, 2016.

[10] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil, "Model compression," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the Knowledge in a Neural Network," in *Neural Information Processing Systems, Deep Learning and Representation Learning Workshop*, 2014.

[12] William Chan, Nan Rosemary Ke, and Ian Lane, "Transferring Knowledge from a RNN to a DNN," in *INTERSPEECH*, 2015.

[13] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio, "FitNets: Hints for Thin Deep Nets," in *International Conference on Learning Representations*, 2015.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Dark knowledge," `http://www.ttic.edu/dl/dark14.pdf`, [Online; accessed 1-Feb-2017].

[15] Sergio Escalera, Xavier Baró, Jordi Gonzàlez, Miguel A. Bautista, Meysam Madadi, Miguel Reyes, Víctor Ponce-López, Hugo J. Escalante, Jamie Shotton, and Isabelle Guyon, "ChaLearn Looking at People Challenge 2014: Dataset and Results," in *European Conference on Computer Vision Workshops*, 2015.

[16] Di Wu, Lionel Pigou, Pieter-Jan Kindermans, Nam Do-Hoang Le, Ling Shao, Joni Dambre, and Jean-Marc Odobez, "Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[17] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks," in *European Conference on Computer Vision Workshops*, 2015.

[18] Natalia Neverova, Christian Wolf, Graham W. Taylor, and Florian Nebout, "ModDrop: Adaptive Multi-modal Gesture Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.