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

koustav.mullick@research.iiit.ac.in
anoop@iiit.ac.in

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Summary

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.

Motivation

- Gesture recognition is one of the key components in natural human-computer interfaces, especially for mobile devices.
- Challenges: Background inconsistencies, user-level variations in gesturing, different user appearance, pose.
- Existing Approaches
  - Distill the video into an image using: 1) Features that capture temporal information [1], or computing optical flow [7], and use image classification models.
  - Use of models better suited to capture temporal information: 1) 3D-CNN [4] and 2) recurrent networks such as LSTM [3].
- Combining 3D-CNN with LSTM leads to models that are accurate and robust enough to handle the complex variations present in the videos.
- Using knowledge distillation, we develop compact models, that can be further compressed, with minimal impact on accuracy to make them suitable for mobile devices.

Our Approach

Baseline Models

As baseline models we use a 3D-CNN and an LSTM variant of RNN to classify each gesture.

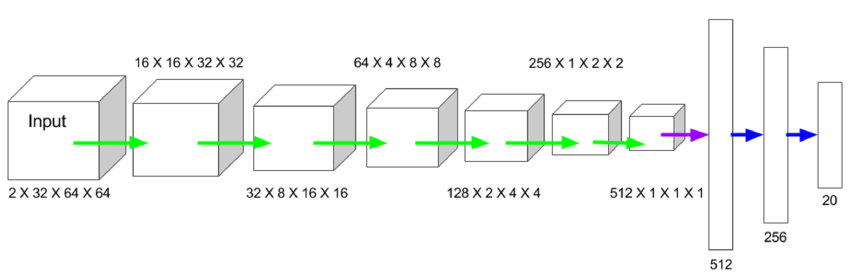

Figure 1: 3D-CNN architecture

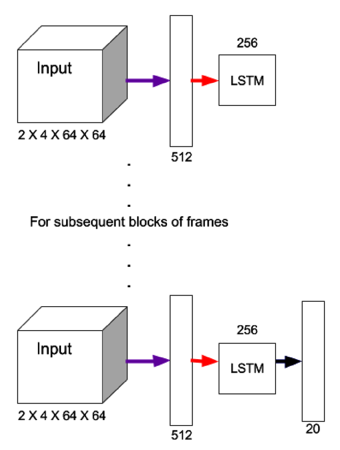

Figure 2: LSTM architecture

Joined 3D-CNN and LSTM

Next we combine the 3D-CNN with LSTM. The 3D-CNN acts as an encoder for groups of few frames, which are fed as sequences to the LSTM to get the final prediction.

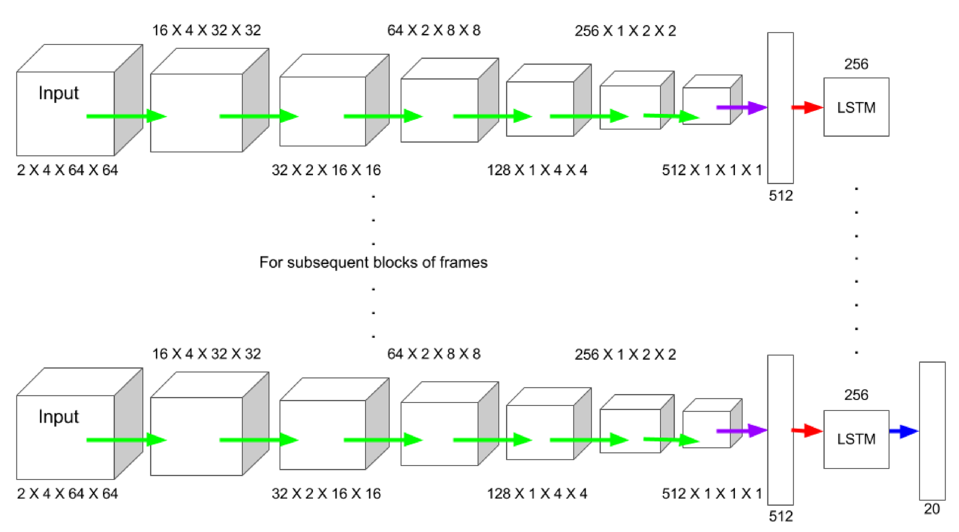

Figure 3: Joined 3D-CNN and LSTM architecture

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

We use our trained baseline CNN as a teacher to train much smaller variants of our joined 3D-CNN and LSTM models. Softened softmax output for each training sample is obtained from the trained 3D-CNN architecture using:

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

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

Smaller variants of the joined model are trained using 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).

Dataset

The Chalearn 2014 Looking at People Challenge (track 3) [2] dataset:

- Vocabulary: 20 different Italian cultural/ anthropological signs.
- Number of users: 27 users with variations in surroundings, clothing, lighting and gesture movement.
- Recording Device: Microsoft Kinect. Data contains RGB, depth, user mask and skeleton/joint information for each frame of video.

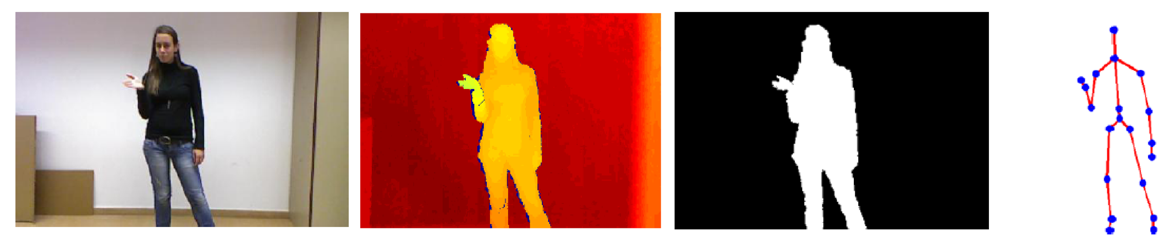

Figure 4: Example frame modalities from the dataset

- For each video frame we use the depth and grayscale to obtain two-channel inputs for our models.
- Upper-body region and the highest hand region for each gesture are cropped out using skeleton information.
- We also perform rotation, translation and zooming on the frames for data augmentation.

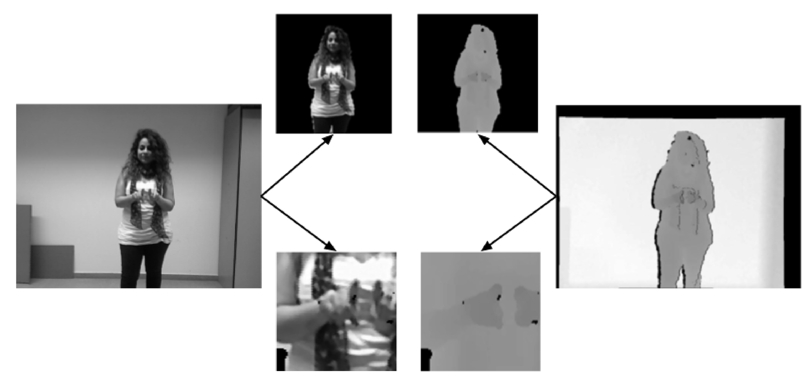

Figure 5: Input frames to our models

Results

Method/Model	Accuracy(%)
Baseline LSTM	86.6
Baseline 3D-CNN	90.1
3D-CNN + LSTM (ours)	**93.2**
Wu *et al.* [8]	87.9
Pigou *et al.* [6]	91.4
Neverova *et al.* [5]	**96.8**

Table 1: Accuracies obtained using our model compared with *state-of-the-art* methods

	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 2: Knowledge Distillation from baseline 3D-CNN to CNN + LSTM

- Training with Adam optimizer compresses the model further by pushing most of the parameters of the student towards very low weight.
- 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 accuracy.

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 3: Reduction is size along with performance impact of the student model and sparse model.

Conclusions

- 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.
- Information can be distilled from a larger model to models with $16\times$ and $4\times$ fewer parameters. To the best of our knowledge, this is the first work exploring the knowledge distillation framework for videos.
- The model size could be further reduced using a sparse representation. This benefits training time and also makes it possible to use them in low-memory and low-power embedded devices.

References

[1] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic Image Networks for Action Recognition. In *CVPR*, 2016.

[2] S. Escalera, X. Baró, J. Gonzàlez, M. A. Bautista, M. Madadi, M. Reyes, Víctor Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon. ChaLearn Looking at People Challenge 2014: Dataset and Results. In *ECCV Workshops*, 2015.

[3] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997.

[4] S. Ji, W. Xu, M. Yang, and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. In *IEEE TPAMI*, 2013.

[5] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. ModDrop: Adaptive Multi-modal Gesture Recognition. In *IEEE TPAMI*, 2015.

[6] L. Pigou, S. Dieleman, P. Kindermans, and B. Schrauwen. Sign Language Recognition Using Convolutional Neural Networks. In *ECCV Workshops*, 2015.

[7] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NIPS*, 2014.

[8] D. Wu, L. Pigou, P. Kindermans, N. Le, L. Shao, J. Dambre, and J. Odobez. Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition. In *IEEE TPAMI*, 2016.