Spatiotemporal Multimodal Network for Dynamic Gesture Recognition

Hsuan-I Ho
hohs@student.ethz.ch

Chi-Ching Hsu
hsuch@student.ethz.ch

ABSTRACT

Human gesture recognition is among interesting topics of visual understanding, which benefits a variety of applications like user interface design and robotic perception. With the goal of deriving an effective model to recognize human gestures, we propose a deep neural network architecture for describing and recognizing important spatial-temporal information across different video data domains. Our proposed network is realized by using a technique of transfer learning, in which the network can be trained and evaluated on a relative small dataset while possible overfitting can be also mitigated. In our experiments, our method is able to achieve state-of-the-art accuracy on a challenging gesture recognition benchmark.

1 INTRODUCTION

Human gesture carries a wealth of information, aiding us in communication, negotiation and even expressing our feelings without words. Understanding and recognizing the meaning of human gesture is therefore important for applications in the area of human-computer interaction, computer vision and robotics. To recognize semantics of human gesture given a video sequence, gesture recognition can also be regarded as a task of video action recognition.

Traditional approaches for video action recognition utilized hand-crafted features, such as HoG [2] and HoF [11], with aggregation of interest points based on trajectories [20, 21]. With the development of deep learning, recent works successfully utilized deep convolutional neural networks (CNNs) for video action recognition [10, 14]. However, this improvement is brought on by more robust CNN image representations. Approaches to modeling the temporal structure are still naive and simple, e.g., subsampling several frames and performing average pooling to generate final predictions. Other deep learning based methods have been explored by utilizing recurrent neural networks (RNNs) [3, 15, 25], and other feature aggregation schemes [6, 22, 23] based on CNN features. Nevertheless, these methods introduce new computation overhead but not necessarily perform better than simple average pooling [22]. More recent works introduced the concept of 3D convolution [5, 9, 16, 17] and (2+1)D convolution [18], which enhanced the capability of CNNs to model the short-term temporal structure. However, the explosion of parameters not only increases overall computation time, possible overfitting might also occurs as well. In addition, evidence also shows that exploitation of additional depth maps, pre-computed optical flow can always improve the performance [1, 5, 22], which suggests that simply using features from RGB frames is insufficient to capture spatial-temporal information within video clips.

Inspired by [7, 12], with the goal of incorporating both short-term and long-term information from different domains of data, we introduce a spatiotemporal multimodal network. In specific, our model combines multiple 3D CNN feature extractors for capturing short-term information from RGB frames, depth maps, and skeleton data. Moreover, we further integrate a sequence-to-sequence model based on bidirectional long short term memory (biLSTM) and attention mechanism, which allows the exploitation of long-term temporal information for improved classification.

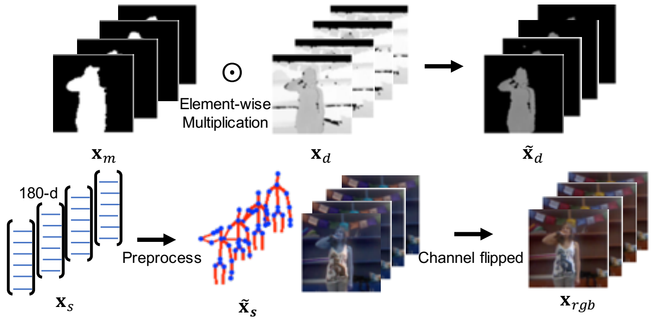

Figure 1: Example of a video clip from ChaLearn [4] dataset and our preprocessing procedure.

In summary, our contributions are listed as follows:

- We exploit 3D CNN feature extractors pretrained on a large scale dataset, and further encode the resulting feature maps for later classification. Not only the number of parameters during training can be reduced, possible overfitting due to limited training data can be also alleviated.
- Our network is able to observe long-term dependency between segments within a full video sequence by integrating informative spatial-temporal features with RNN-based components.
- In addition to the use of RGB video frames, we further extend our network by exploiting auxiliary features of depth maps and skeleton information, which is able to compensate for the shortfall of RGB features.

2 PROPOSED METHOD

2.1 Dataset and preprocessing

The provided dataset for gesture recognition is a cleaned version of the ChaLearn [4] dataset, which contains 5722/1765/2174 video clips in the training, validation and test set respectively. For each video which contains L ($50 \leq L \leq 150$) frames, we are provided with its RGB frames, depth maps, segmentation masks, skeleton information (denoted as $\{\mathbf{x}_{rgb}^{(i)}, \mathbf{x}_d^{(i)}, \mathbf{x}_m^{(i)}, \mathbf{x}_s^{(i)}\}_{i=0,...,L-1}$) and the corresponding action label y. The RGB frames, depth maps and segmentation masks are cropped into size of 80×80 pixels while skeleton information is a 180-dimensional feature per frame.

As shown in Fig. 1, we apply several prepossessing techniques based on characteristics of different data domains rather than simply

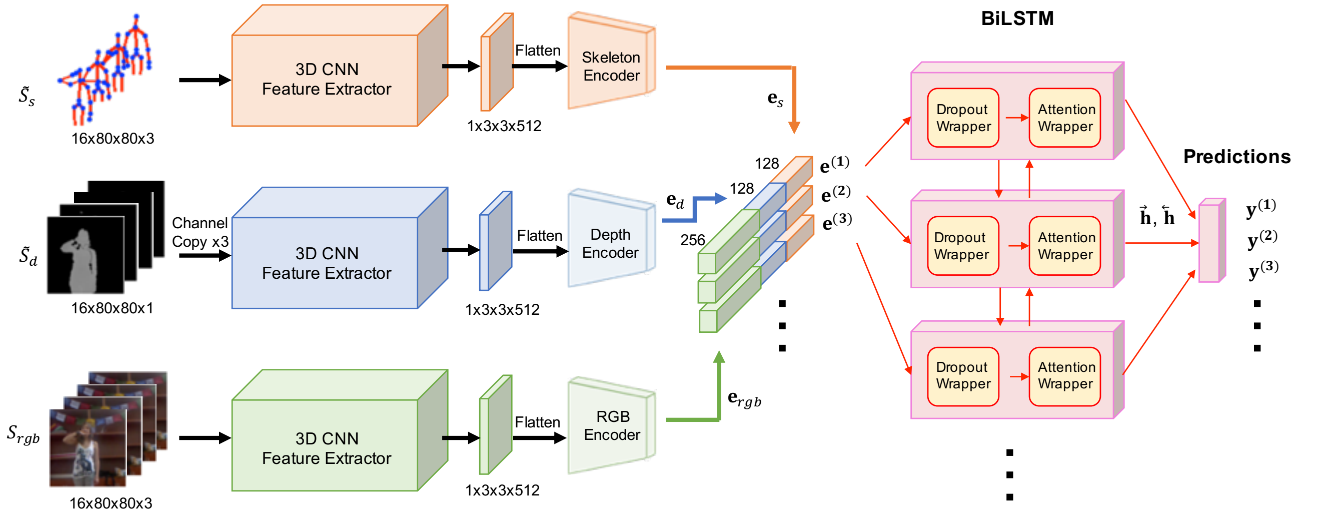

Figure 2: Our spatiaotemporal multimodel network for gesture recognition. We have three sub-models in green, blue, and oranges denote the RGB frames, depth maps and skeleton information, respectively. The input sequence is divided into a series of 16-frames clips to extract their spatial-temporal representations. The final prediction is made by considering all representations in order via a BiLSTM cell.

using the raw data as inputs. To be precise, we perform element-wise multiplication on $\mathbf{x}_d$ and $\mathbf{x}_m$ to eliminate noisy depth response near the edge of human body. Moreover, we depict skeleton features as 80×80 RGB skeleton frames $\tilde{\mathbf{x}}_s$ which are more suitable for the 3D CNN feature extractor. Finally we also adjust the channel order of $\mathbf{x}_{rgb}$ (i.e., from BGR to RGB) in correspondence to the pretrained weigths of our feature extractor.

2.2 Network architecture

The architecture of our proposed method is shown in Fig. 2, which consists of network components for video feature extraction and gesture recognition. Details of our network will be described in the following subsections.

2.2.1 Feature extraction. As suggested in [18], the architecture of 3D CNN have demonstrated the ability to extract semantic spatio-temporal feature given a short video segment (usually with a fixed length of 16 frames). Thus, we first split each video clip into segments with fixed-length for feature extraction, e.g., $\{S_{rgb}^{(j)} = \{\mathbf{x}_{rgb}^{(16j)}, \dots, \mathbf{x}_{rgb}^{(16j+15)}\}\}_{j=0,\dots,N-1}$, $N = \lceil \frac{L}{16} \rceil$. However, training such network on a small dataset is infeasible due to large number of parameters in 3D convolution kernels. We address this issue using the technique of transfer learning, which reduces the computational burden of training a network by initializing with pretrained weights.

To be more specific, we adopt the architecture of Convolutional 3D networks (i.e., C3D networks [17]) as our feature extractor, in which its weights are pretrained on Sport1M [10] video classification dataset. The feature map yielded from the pool5 layer of extractor serves as the input of a feature encoder, which further encodes the feature map into a compact representation for later classification.

In addition, to further enhance robustness of spatio-temporal feature and classification performance, we employ three separated feature extractors to capture domain specific information in different data domains. We note that the concatenation of the above features can be regarded as a new feature representation, i.e., $\{\mathbf{e}^{(j)} = \text{concat}(\mathbf{e}_{rgb}^{(j)}, \mathbf{e}_d^{(j)}, \mathbf{e}_s^{(j)})\}_{j=0,\dots,N-1}$.

2.2.2 Sequence based gesture recognition. From the above subsection, we see that the first stage of our network performs feature extraction across data domains to preserve domain specific information in the compact feature representation. Since the focus of our network is to perform gesture recognition, we finally introduce a sequence based classification network.

As pointed out by [7, 12], integration with RNN-based components allows one to observe long-term dependency between segments within a video sequence. Inspired by their work, we particularly exploit bidirectional long short term memory (biLSTM) to model long-term dependency in each video clip. The biLSTM cell takes a sequence of concatenated features $\{\mathbf{e}^{(j)}\}_{j=0,\dots,N-1}$ as inputs and returns both forward hidden states $H_{for} = \{\overrightarrow{\mathbf{h}}^{(j)}\}_{j=0,\dots,N-1}$ and backward hidden states $H_{back} = \{\overleftarrow{\mathbf{h}}^{(j)}\}_{j=0,\dots,N-1}$, which preserve semantic information across time periods. A following single-hidden-layer MLP can generate softmax predictions $\tilde{Y} = \{\tilde{\mathbf{y}}^{(j)}\}_{j=0,\dots,N-1}$ given H_{for} and H_{back} as inputs. The average cross-entropy loss between the softmax predictions and ground-truth label is calculated as:

$$\mathcal{L} = -\frac{1}{N} \sum_{j=0}^{N-1} y \cdot \log(\tilde{\mathbf{y}}^{(j)}), \tag{1}$$

where the ground-truth label y is converted to a 20 classes one-hot vector, and the loss is averaged over a sequence of 20-dimensional softmax predictions $\tilde{Y}$.

Due to the special characteristic of human gestures, e.g., two gestures could have totally different meanings with only slight differences, it would be desirable to apply a mechanism to distinguish and emphasize such differences. Thus, we further extend our sequence based network by integrating the attention mechanism [13, 19], which assigns importance weights to different segments instead of treating them equally for classification. We note that the attention wrapper can be easily applied and integrated to our network with only small computational overhead.

2.3 Implementation details

2.3.1 Network topology. The architecture of 3D CNN feature extractor is based on the architecture in [17], which consists of 8 convolution layers (with 64, 128, 256(×2), and 512(×4) 3×3×3 convolution kernels) and 5 spatial-temporal pooling layers. Based on their architecture, we additionally apply batch normalization [8] to each convolution layer to accelerate the training procedure. The flatten feature (4608-d) extracted from the pool5 layer serves as the input of our feature encoder, which has a two-layer fully connected structure (with 4608, and 256/128 Leakly ReLU [24] units). Each fully connected layer is followed by a dropout layer to prevent overfitting. Our sequence based classification network consists of a biLSTM with 256 hidden units in the both forward and backward cells. Both cells contain an additional dropout wrapper and attention wrapper for improved classification.

2.3.2 Parameter settings. To train our proposed model, we employ different learning rates when updating different network components. To be detailed, we train our network using Adam optimizer with a batch size of 4, first- and second-momentum of 0.9 and 0.99, and dropout rate of 0.3. The length of attention wrapper is set to 5 so that it fits the number of segments in video clip. The learning rate of the feature extractors is set to 10^{-5} while the other components is set to 5×10^{-5} in the training procedure. We choose "fixed learning rate" and "average loss" (i.e., Eq. 1) for updating parameters. It takes roughly 12 hours to train our network in total 40K iterations on a single NVIDIA Tesla V100.

3 EXPERIMENT

Table 1 summarizes the quantitative results of our method. We compare our spatiotemporal multimodal network with existing baseline models, and report their public scores after submission. Our method produced favorable results due to learning of long-term video dependency in multiple data domains. Approaches of 2D CNN and C3D [17] were not able to achieve comparable results due to lack of ability in modeling dependency between video frames/segments. We also observed that the use of "average loss", which outperformed the other two loss computation criteria could also benefit learning of long-term dependency in the LSTM cells.

	Methods	Accuracy
Baseline	2D CNN (Sample)	0.4765
	C3D [17]	0.7332
	Hard baseline	0.8326
Analysis of feature size / LSTM units	256 / 128	0.8565
	512 / 256	0.8804
	1024 / 512	**0.8896**
Analysis of loss critera	"last logit"	0.8261
	"average logit"	0.8666
	"average loss"	**0.8804**
Analysis of network design	RGB only	0.7690
	RGB + depth	0.7856
	W/o skeleton preprocessing	0.8031
	W/o pre-training	0.8233
	W/o attention wrapper	0.8565
	W/o dropout wrapper	0.8684
Ours	Full model	**0.8804**
	Ensemble	**0.9126**

Table 1: Performance evaluation and analysis of our network design and settings.

To further verify each design choice of the proposed network, we also present controlled experiments using variants of our model. First, we search for an appropriate feature size for the spatial-temporal features and LSTM hidden states (i.e., $\mathbf{e}$, $\overleftarrow{\mathbf{h}}$ and $\overrightarrow{\mathbf{h}}$). It shows that using the size of 1024/512 slightly improved the performance, while the number of parameters was increased by 16M. Therefore, to reduce computational overhead and training time, we choose 512/256 as our final parameter setting. In addition, we note that the exploitation of our data preprocessing outperformed other variants which either solely use data in certain domains or without any preprocessing. This again verifies the effectiveness of learning domain-specific features in a multimodal scheme. It can also be seen that training the network from draft prone to overfit the small amount of training data, and therefore could not yield satisfactory performance. Finally, our ensemble model is averaged over softmax predictions from multiple full models (i.e., models in different runs). It is clear that both our full model and ensemble model achieved the best performance among all variants. Thus, our network design and integration of the above components are desirable for gesture recognition.

4 CONCLUSION

We present spatiotemporal multimodal network which effectively incorporates features extracted from different data domains. We also introduce sequence based gesture recognition component to model long-term dependency between video segments and further improve the performance of our proposed network. In the experiment, we verify the effectiveness of our network design and demonstrate that our model perform satisfactory result against the other participants on the leaderboard of the dynamic gesture recognition challenge.

REFERENCES

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

[2] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, Vol. 1. IEEE Computer Society, 886–893.

[3] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2625–2634.

[4] Sergio Escalera, Xavier Baró, Jordi Gonzalez, Miguel A Bautista, Meysam Madadi, Miguel Reyes, Víctor Ponce-López, Hugo J Escalante, Jamie Shotton, and Isabelle Guyon. 2014. Chalearn looking at people challenge 2014: Dataset and results. In *European Conference on Computer Vision*. Springer, 459–473.

[5] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1933–1941.

[6] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. 2017. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 971–980.

[7] Pavlo Molchanov Xiaodong Yang Shalini Gupta and Kihwan Kim Stephen Tyree Jan Kautz. [n. d.]. Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks.

[8] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

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

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

[11] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. 2008. Learning realistic human actions from movies. In *CVPR 2008-IEEE Conference on Computer Vision & Pattern Recognition*. IEEE Computer Society, 1–8.

[12] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. 2018. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision* 126, 2-4 (2018), 430–439.

[13] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* (2015).

[14] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*. 568–576.

[15] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*. 843–852.

[16] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. 2015. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 4597–4605.

[17] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 4489–4497.

[18] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 6450–6459.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[20] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. 2013. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision* 103, 1 (2013), 60–79.

[21] Heng Wang and Cordelia Schmid. 2013. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*. 3551–3558.

[22] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*. Springer, 20–36.

[23] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. 2018. Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6026–6035.

[24] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).

[25] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4694–4702.