Learning Hand Features for Sign Language Recognition

Course Project: Deep Learning for Perception - Spring 2015

Kaushik Patnaik, Payam Siyari, Vinodh Krishnan, Vivek Nabhi
Georgia Institute of Technology, College of Computing

Abstract

Our project aims towards learning spatio-temporal features from images. We use 3D convolutional networks to learn such features and use it to recognize words in Amercian Sign Language. We compare our results with a baseline 2D convolutional neural network model to illustrate the advantages of learning spatio-temporal features over spatial features.

1 Introduction

Around 360 million people all over the world suffer from hearing loss and Sign Language, is the most popular means for this group to communicate. However, majority of people without hearing loss do not understand sign language. Thus, a lot of research and effort is put into **translating sign language into spoken language**. However this is a diffcult task as -

1. Sign Languages are **not** visual representation of spoken languages.
2. Sign languages have **their own syntactic and grammatical structures.**
3. They are both **spatial and temporal in nature.**
4. Gestures are identified by **hand position, finger shape, parts of the hand and also facial expressions.**
5. **Data background** is ever changing.
6. **Data occlusion** in the signed videos is also a concern.

2 The Dataset and Preprocessing

Dataset

- **American Sign Language Lexicon Video Data set (ASLLVD)** [1]
- Video sequences in uncompressed raw format for >3,300 words
- Collected using 4 synchronized cameras capturing different views of the signer
- **High fidelity in the hand and face regions** as they contain most information about the signed word
  - Automatic skin segmentation
  - Frames cropped to the skin
  - Normalized to ensure uniform brightness

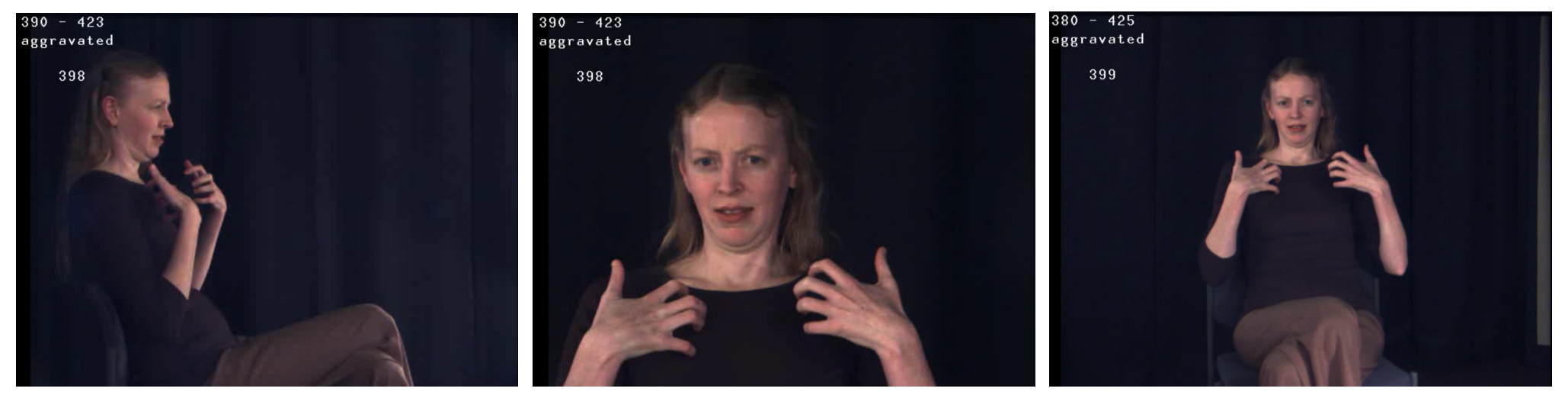

Figure 1: Side, zoomed and front views of the signer as in the ASLLVD dataset

In this work:

- A sample of this data set containing 10 classes (words)
- Each word had around **5-6 videos** showing the full resolution **front view and the side view**
  - We only use *front views* to ensure training set uniformity.

Preprocessing

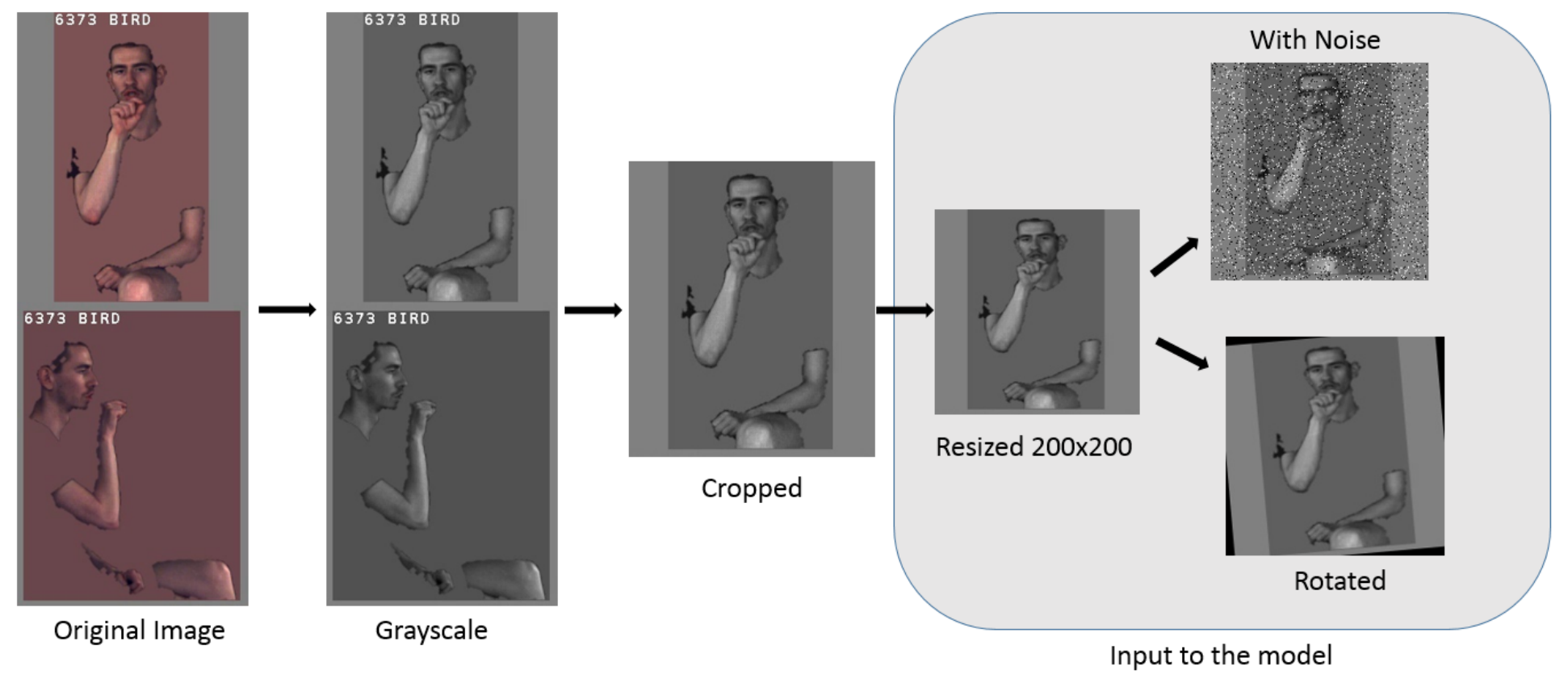

Figure 2: Data Pre-processing flow

3 Model Architecture

3.1 Spatial Model: 2D Convolutional Neural Network (2D-CNN)

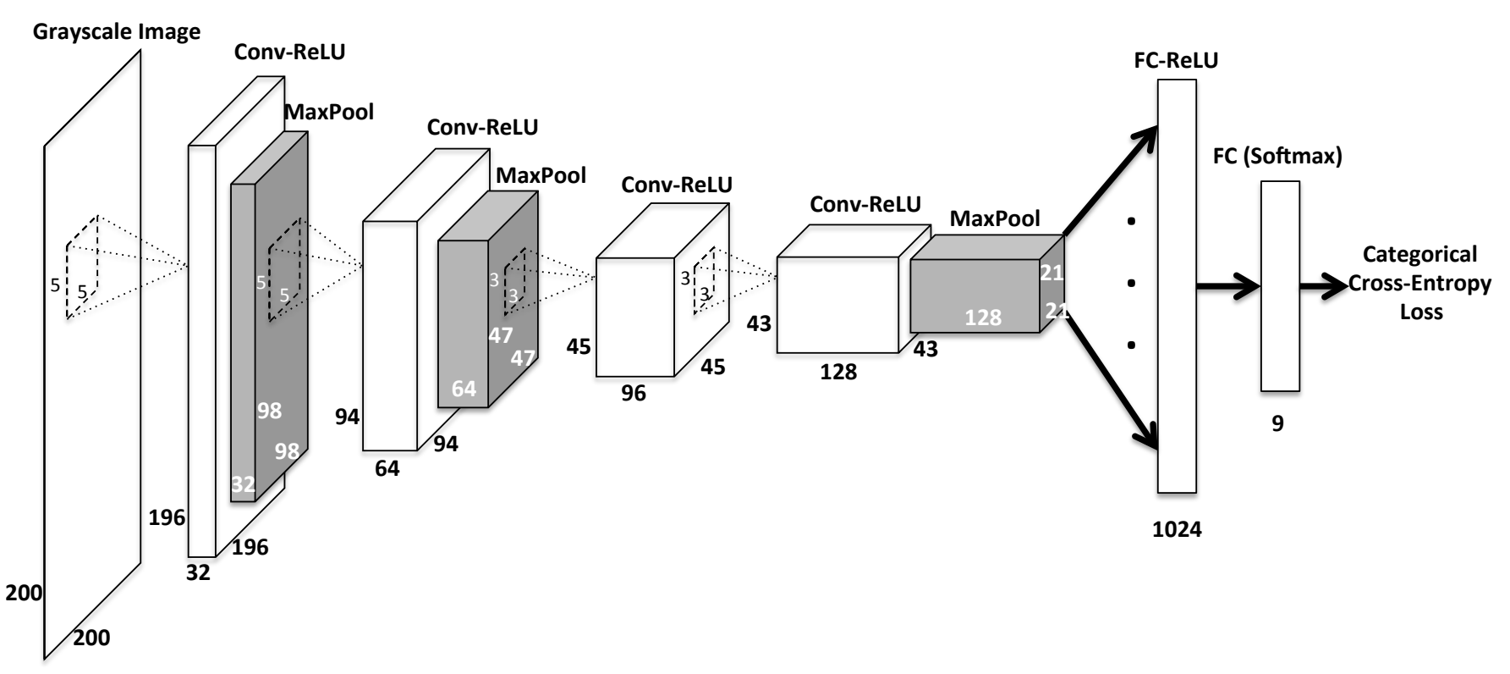

Figure 3: 2D-CNN Model

- Our baseline model:
  - Most common CNN architectures, e.g. [5]: $Input \rightarrow [[CONV \rightarrow ReLU]^N \rightarrow MaxPool]^M \rightarrow [FC- > ReLU]^K \rightarrow FC$
  - Convolution is only applied **across space**
  - **Softmax** is used at the last layer
  - **Categorical Cross-entropy** is the optimization objective

3.2 Spatio-Temporal Model: 3D Convolutional Neural Network (3D-CNN)

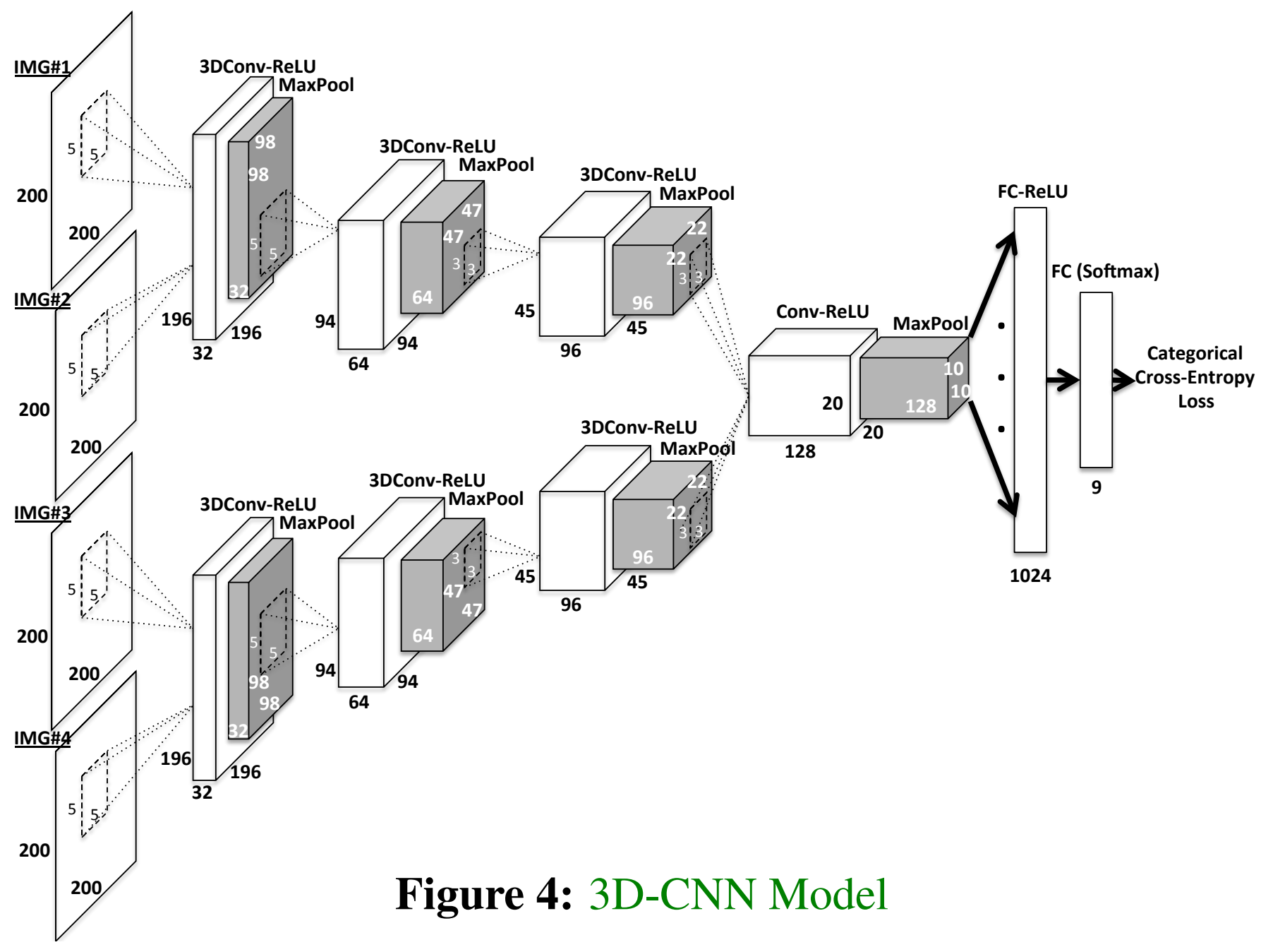

Figure 4: 3D-CNN Model

- Similar to the work in [3], "Slow Fusion"architecture [4]:
  - Using **cubic kernels** to support convolution across time
  - **Generalization of 2D-CNN**, across time
  - **Same spatial architecture, class score function and loss** as in 2D-CNN

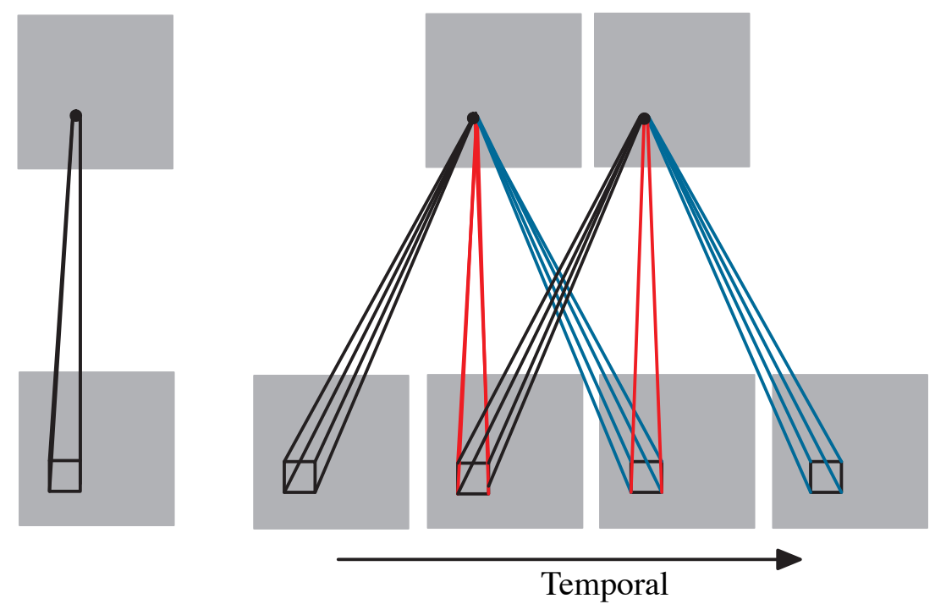

Figure 5: [3]: "2D-Convolution (left) vs. 3D-Convolution. In the right figure, the size of the convolution kernel in the temporal dimension is 3, and the sets of connections are color-coded so that the shared weights are in the same color. In 3D convolution, the same 3D kernel is applied to overlapping 3D cubes in the input video to extract motion features. "

4 Experimental Evaluation

4.1 Implementation Details

- Both models trained using **Backpropagation**, incorporating **RMSProp** [6]
- Implemented in Theano [2]
- Measuring Top-1, Top-3 and Top-5 classification accuracy
  - Single-frame classification in 2D-CNN
  - Sequence classification in 3D-CNN

4.2 Results

Model	2D CNN	3D CNN
Top-1	0.276	0.211
Top-3	0.4225	0.556
Top-5	0.439	0.556

Table 1: Comparison of classification accuracy of 2D and 3D-CNN models (Best results for each measure)

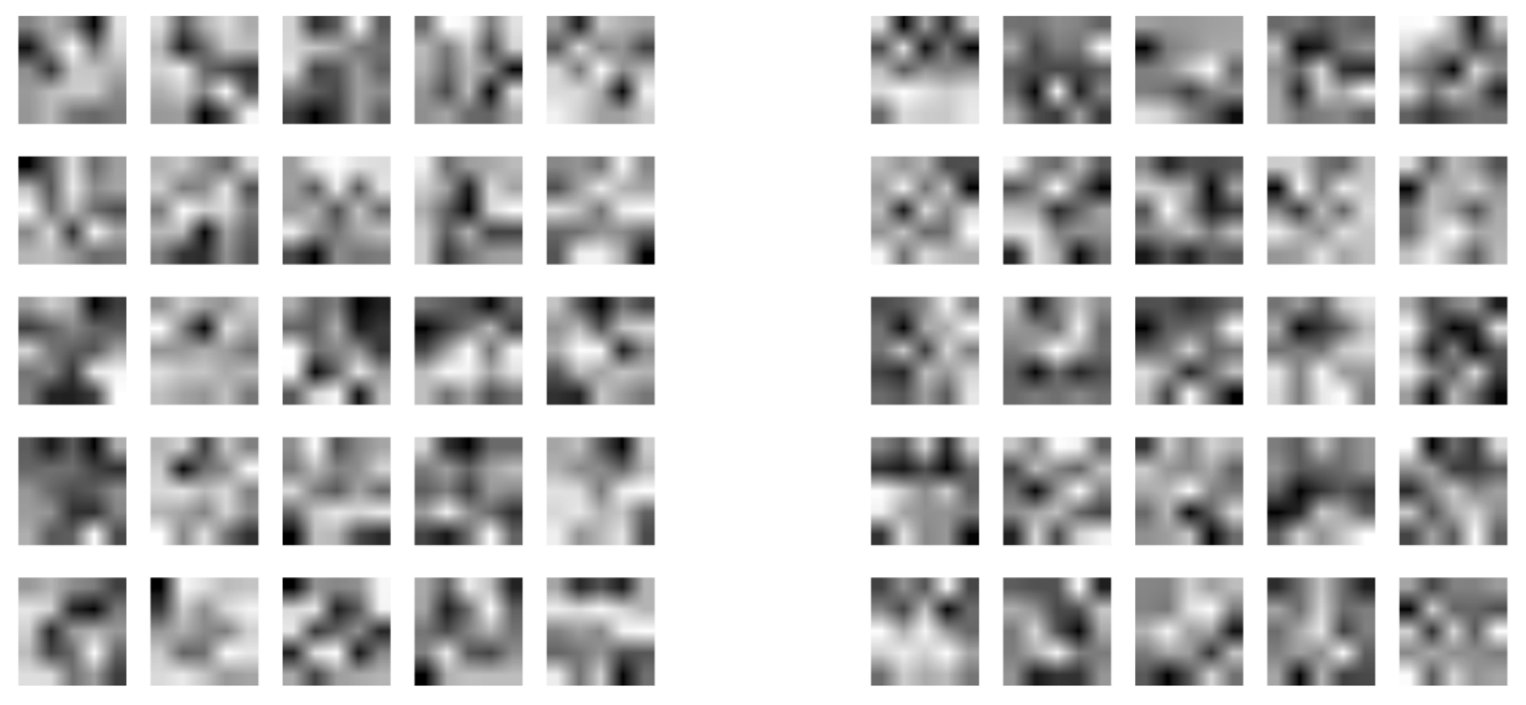

Figure 6: Visualization of Features in 3D CNN

Conclusions and Future Research

- 3D convolutions are able to learn spatio-temporal features but need better initializations. Current initialization schemes involve 2D CNNs learnt from the same data
- Current models do not involve max-pooling across time, this leads to time variant features
- For the particular task, weights can be biased to regions where the hands exist
- Comparison with alternate spatio-temporal learning methods such as CNN-RNN to be carried out

References

[1] `http://secrets.rutgers.edu/dai/queryPages/search/search.php`.

[2] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[3] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013.

[4] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1725–1732. IEEE, 2014.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[6] T. Tieleman and G. Hinton. Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning, 2012.

Acknowledgements

We thank Networking and Telecommunication Group at College of Computing, GeorgiaTech, for providing us the server computers.