Real-time Sign Language Fingerspelling Recognition using Convolutional Neural Networks from Depth map

Byeongkeun Kang Subarna Tripathi Truong Q. Nguyen
Department of Electrical and Computer Engineering
University of California, San Diego
bkkang, stripathi, tqn001@ucsd.edu

Abstract

Sign language recognition is important for natural and convenient communication between deaf community and hearing majority. We take the highly efficient initial step of automatic fingerspelling recognition system using convolutional neural networks (CNNs) from depth maps. In this work, we consider relatively larger number of classes compared with the previous literature. We train CNNs for the classification of 31 alphabets and numbers using a subset of collected depth data from multiple subjects. While using different learning configurations, such as hyper-parameter selection with and without validation, we achieve 99.99% accuracy for observed signers and 83.58% to 85.49% accuracy for new signers. The result shows that accuracy improves as we include more data from different subjects during training. The processing time is 3 ms for the prediction of a single image. To the best of our knowledge, the system achieves the highest accuracy and speed. The trained model and dataset is available on our repository[1].

1. Introduction

Sign language recognition is important for natural and convenient communication between deaf community and hearing majority. Currently, most communications between two communities highly rely on human-based translation services. However, this is inconvenient and expensive as human expertise is involved. Therefore, automatic sign language recognition aims to understand the meaning of signs without the assistance from experts. Then it can be translated to sound or text based on end users' needs. We believe that sign language recognition is important for providing equal opportunity to every person and improving public welfare.

Sign language recognition is still a challenging problem despite of many research efforts during the last few decades [12, 1]. It requires the understanding of combination of multi-modal information such as hand pose and movement, facial expression, and human body posture. Also, sign language has at least thousands of words including very similar hand poses while gesture recognition generally includes a small set of well specified gestures. Moreover, even same signs have significantly different appearances for different signers and different viewpoints.

In this paper, we focus on static fingerspelling in American Sign Language (ASL) which is a small, but important part of sign language recognition. This is a small set of sign languages as shown in figure 1, but is used in many situations in conveying names, addresses, brands, and so on. Static fingerspelling is still challenging because of visually similar yet different signs. For example, some of the signs are only distinguished by the position of thumb. Also, a large variation occurs from different camera viewpoint and different signers.

Depth sensors enable us to capture additional information to improve accuracy and/or processing time. Also, with recent improvement of GPU, CNNs have been employed to many computer vision problems. Therefore, we take advantage of a depth sensor and convolutional neural networks to achieve a real-time and accurate sign language recognition system.

2. Related Works

Although gesture recognition only considers well specified hand gestures, some approaches are related to sign language recognition. Nagi *et al.* [8] proposed a gesture recognition system for human-robot interaction using CNNs. Van den Bergh *et al.* [14] proposed a hand gesture recognition system using Haar wavelets and database searching. The system extracts features using Haar wavelets and classifies input image by finding the nearest match in the database. Although both systems show good results, these methods consider only six gesture classes.

[1] https://github.com/byeongkeun-kang/FingerspellingRecognition

978-1-4799-6100-9/15/$31.00 ©2015 IEEE

Different sign languages are used in different countries or regions. There have been efforts towards sign language recognition systems other than ASL as well. Pigou *et al.* [9] proposed an Italian sign language recognition system using CNNs. Although they reported 95.68% accuracy for 20 classes, they mentioned that users in test set can be in training set and/or validation set. Liwicki *et al.* [7] described a British Sign Language recognition system that understands fingerspelled words from video. The system first recognized letters using Histogram of Gradients (HOG) descriptors. Then that recognized words using Hidden Markov Models (HMM). That system is different from recognizing a single fingerspelling. The dataset in use corresponded to a single signer.

ASL sentence recognition and verification has also been explored. Zafrulla *et al.* [16] proposed a system which recognizes a sentence of three to five words. The word should be one of 19 signs in their dictionary. They also used Hidden Markov Models on extracted features.

Our work belongs to the category of ASL fingerspelling recognition systems. Isaac *et al.* [3] proposed an ASL fingerspelling recognition system based on neural networks applied to wavelet features. They reported a recognition rate of 99%. However, they did not specify the size of the dataset and the number of different subjects. Later, Pugeault *et al.* [10] proposed a real-time ASL fingerspelling recognition system using Gabor filters and random forest. Their system recognizes 24 different ASL fingerspelling for alphabets. They collected dataset from five subjects and reported a recognition rate of 75% using both color and depth, 73% using only color, and 69% using only depth. Although Pugeault *et al.* [10] reported that combination of color and depth improves the recognition rate, we only use depth to achieve better consistency to illumination changes and skin pigment differences and to avoid calibration process for general users. Kuznetsova *et al.* [6] also proposed a real-time ASL fingerspelling recognition system using multi-layered random forest. They reported and Dong *et al.* [2] also analyzed that they achieved 87% accuracy for the subjects whom the system has been trained on and 57% accuracy for new subject. Very recently, Dong *et al.* [2] proposed an ASL alphabet recognition system. They first localized hand joint positions using random forest and hierarchical mode-seeking method. Then the system recognized ASL signs by applying random forest classifier to joint angle vector. They reported 90% accuracy for the subjects whom the system has been trained on and 70% accuracy for new signers.

This paper differs from previous works in several ways. First, to the best of our knowledge, ours is the first fingerspelling recognition system to classify total 31 alphabets and numbers compared with the state of the art approach to classify only 24 classes reported in the literature. Second, we extract features by fine-tuning convolutional neural network parameters which are pre-trained for image classification task using 1.28 millions of color images [11]. Moreover, it achieves both real-time and the state of the art accuracy across different users. Our contribution also includes providing publicly available dataset which is currently limited both in quantity and quality.

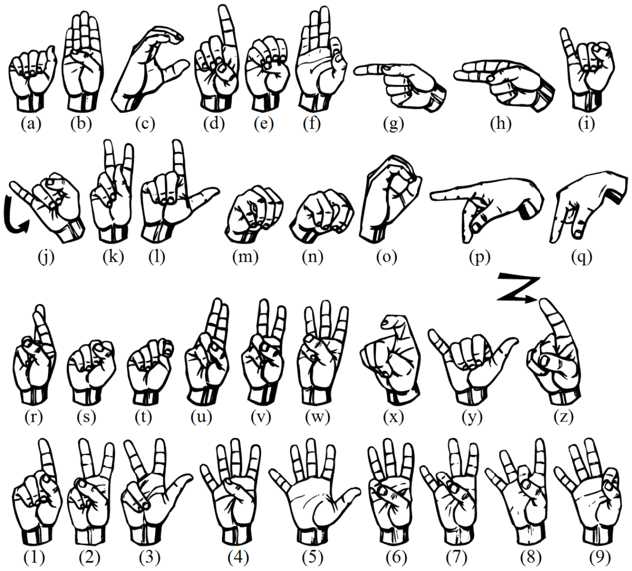

Figure 1. ASL fingerspelling alphabets and numbers [13]. We follow the real demonstrations of formal signs on [15] to collect our dataset. The demonstration images are also available on our repository.

3. Method

3.1. Dataset

We have collected 31,000 depth maps using a depth sensor, Creative Senz3D camera of the resolution of 320 × 240. The dataset consists of 1,000 images for each of the 31 different hand signs from five subjects. 31 hand signs include all the fingerspellings of both alphabets and numbers except J and Z which require temporal information for classification. Since (2/V) and (6/W) are differentiated based on context, we have only one class to represent both one alphabet and one number. Although some informal signs are clearer and easier to recognize, we follow formal signs to avoid ambiguity between signers [15]. To collect dataset from various viewpoint, the dataset is collected while subjects are moving their hand around both on image plane and along z-axis.

3.2. Hand Segmentation

We assume that the closest object from camera is the user's hand. This assumption is valid in fingerspelling and

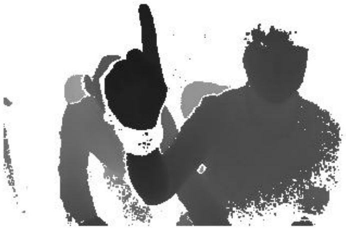

Figure 2. Captured image before pre-processing. The hand is convincingly the closest object according to the captured depth map and there is a clear depth void around the hand which can be exploited for hand segmentation using connected component from the area with the closest depth.

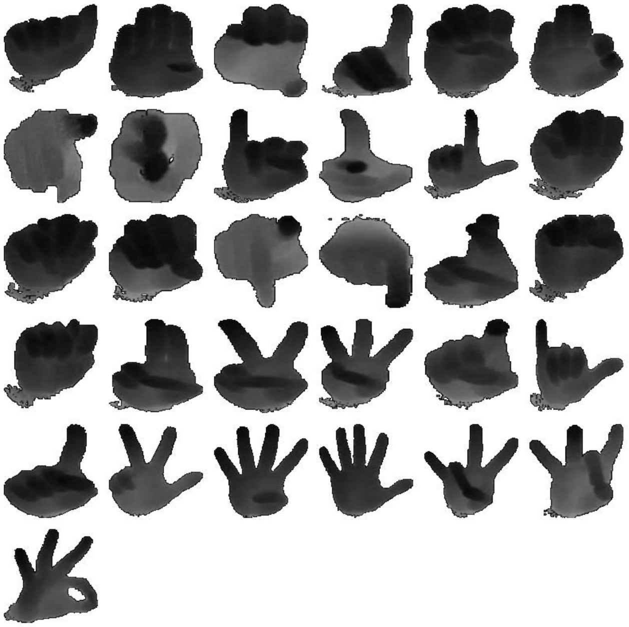

Figure 3. Examples of pre-processed dataset from A to Z and from 1 to 9. In real dataset, background is set to zero.

most of gesture recognition tasks. In addition, we use a black wrist band to get depth voids around wrist, since depth sensor cannot capture depth from black objects well. Figure 2 shows one example of the captured depth image, where the hand is convincingly the closest object according to the captured depth map and there is a depth void around the hand. Hand segmentation thus ends up in finding the connected components from this closest region of the depth image. This strategy provides a very simple and effective real-time hand segmentation. Figure 3 shows segmented hand depth image samples for the 31 signs including alphabets and digits generated using this method. We find a bounding box of hand region and scale it to 227 × 227. Then we include redundant 14 or 15 pixels at each edge of the bounding box to make it the same input size of 256 × 256 described by Krizhevsky *et al*. [5] and thus take the differences of different hand segmentation results into account.

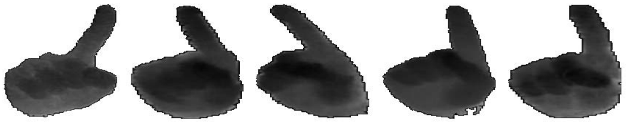

Figure 4. The collected data for the same meaning. It shows the importance of consideration of different signers and viewpoint variation.

3.3. Classification

Architecture : We use Caffe [4] implementation (CaffeNet) of the CNNs which is almost equivalent to AlexNet [5]. The architecture consists of five convolution layers, five max-pooling layers, and three fully connected layers. After each convolution layer or fully connected layer except the last one, rectified linear unit layer is followed. For details, we will upload the architecture and also readers can refer to CaffeNet/Caffe [4] and AlexNet [5].

Feature extraction : We extract a 4096-dimensional feature (final fully connected layer feature) vector from each pre-processed depth image using the aforementioned architecture. First, we subtract the mean image from each of the sample training/validation/test image. Then the mean-subtracted image is forward-propagated to extract features.

Training : We train and test neural networks in five different operating modes. These five cases can be looked upon from different perspectives. One way to look at it is from the pre-training perspective and the second way is how we deal with the training/testing data separation for different subjects. In the former case, we categorize the operating modes into two categories, namely re-training and fine-tuning. For re-training, the model is re-trained from randomly generated weights using the collected fingerspelling data. In fine-tuning, we pre-train the CNNs using a large ILSVRC2012 classification dataset [11]; then we fine-tune the network weights for fingerspelling classification with the same architecture except the last layer which is replaced by 31 output classes. From the subjects' data separation perspective, in one case, we do not separate the subjects in training, validation, and testing and in the second scenario, we use data from different subjects for training, validation, and testing.

4. Experimental Results

As mentioned in Sec. 3.3, we train and test for five experimental settings. The results are compared with other systems in table 1. Our system achieves 99.99% accuracy when training and validation data have samples corresponding to the test subject. In this experiment, 50%, 25%, and

Method	Class type	# of class	# of subj.	Test w/ diff.	Input	Accur.(%)
Nagi *et al.* [8]	Gesture	6	-	No	Color	96
Van den Bergh *et al.* [14]	Gesture	6	-	No	Color & Depth	99.54
Isaacs *et al.* [3]	Alphabets	24	-	-	Color	99.9
Pugeault *et al.* [10]	Alphabets	24	5	-	Color	73
Pugeault *et al.* [10]	Alphabets	24	5	-	Depth	69
Pugeault *et al.* [10]	Alphabets	24	5	-	Color & Depth	75
Kuznetsova *et al.* [6] (50/50)%	Alphabets	24	5	No	Depth	87
Kuznetsova *et al.* [6] (4/1)	Alphabets	24	5	Yes	Depth	57
Dong *et al.* [2] (50/50)%	Alphabets	24	5	No	Depth	90
Dong *et al.* [2] (4/1)	Alphabets	24	5	Yes	Depth	70
Ours (re-training) (50/25/25)%	Alph. & Digit	31	5	No	Depth	99.99
Ours (re-training) (3/1/1)	Alph. & Digit	31	5	Yes	Depth	75.18
Ours (re-training) (4/1)	Alph. & Digit	31	5	Yes	Depth	78.39
Ours (fine-tuning) (3/1/1)	Alph. & Digit	31	5	Yes	Depth	83.58
Ours (fine-tuning) (4/1)	Alph. & Digit	31	5	Yes	Depth	85.49

Table 1. Comparison. Gesture and ASL fingerspelling recognition systems are compared. Test with different subject means that the signer in test set is excluded from training/validation set. The corresponding results are highlighted in light gray. (a/b/c)% represents the portion of dataset for (training/validation/test). (a/b/c) and (a/b) represent the number of subjects in (training/validation/test) and (training/test).

	A	B	C	D	E	F	G	H	I	K	L
Method	M	N	O	P	Q	R	S	T	U	V	W
	X	Y	1	3	4	5	7	8	9	-	-
Pugeault *et al.* [10]	75	83	57	37	63	35	60	80	73	43	87
Color & Depth	17	23	13	57	77	63	17	7	67	87	53
	20	77	-	-	-	-	-	-	-	-	-
Ours	82.7	**94.9**	83.3	85.9	45.7	86.6	86.1	81.8	72.5	86.7	**93.4**
(re-training)	42.1	32.6	73.1	85.4	70.8	58.9	73.6	10.1	61.7	**93.0**	80.2
(3/1/1)	66.9	51.0	**98.6**	80.3	**92.7**	**95.7**	**92.7**	79.6	**92.6**	-	-
Ours	85.8	**95.4**	**91.7**	**91.4**	43.0	83.6	79.9	81.5	78.7	**94.4**	**98.4**
(re-training)	46.8	28.4	74.9	79.0	70.2	69.4	73.2	4.8	70.5	**98.1**	87.6
(4/1)	75.8	86.0	**97.7**	79.5	**92.2**	**96.3**	**98.1**	80.2	**97.6**	-	-
Ours	84.2	**94.3**	86.9	89.7	87.1	**92.3**	88.0	85.2	**90.5**	83.0	**99.7**
(fine-tuning)	62.2	59.0	69.4	82.9	82.6	80.9	57.6	55.3	**92.2**	**94.5**	83.7
(3/1/1)	72.0	76.9	**99.0**	77.9	**91.0**	**98.1**	**95.4**	82.0	**98.1**	-	-
Ours	89.5	**97.1**	**93.2**	88.4	85.7	**93.8**	84.9	86.1	**95.6**	**90.0**	**99.4**
(fine-tuning)	59.9	65.1	69.2	80.2	85.7	85.2	53.7	60.6	**96.1**	**98.2**	87.0
(4/1)	75.3	81.9	**99.0**	82.1	**92.1**	**97.5**	**95.9**	82.5	**99.3**	-	-

Table 2. Detailed results. Accuracy less than 50% is highlighted in light gray. Bold entries correspond to accuracy higher than 90%.

25% of dataset is used for training, validation, and testing. We achieve 75.18% and 78.39% for regular training (re-training) and 83.58% and 85.49% for fine-tuning. This shows that fine-tuning outperforms re-training about 7∼8% in this depth image dataset even though the nature of the pre-training dataset (ILSVRC2012) is different. For each case, the former represents the average result of training with three subjects, validation with one subject, and test with one subject. The latter represents the average result of training with four subjects and test with one subject. We considered all possible combinations of subjects for training, validation, and test and the final reported accuracy is the average of all. For the latter case, although we use the same training parameters (e.g. the number of iterations) for all combinations, the performance is improved about 2∼3%. It shows that by increasing the number of subjects in training, the system's performance for new subject has high possibility of improvement. The number of iterations in the training phase is fixed to 8000 and 4000 for re-training and fine-tuning respectively. Overall, our system achieves about 10∼15% improvement compare to previous state of the art result even with more number of classes and subjects. Moreover, the processing time is about 3 ms for the prediction of a single image using Nvidia GeForce GTX Titan. Table 2 shows accuracy for each alphabet or number. It shows that even with more number of classes, the accuracy of each class is higher than or similar to previous state of the art result. Therefore, it is obvious we will achieve better result with the same number of classes of comparing with the other works. On the other hand, the table shows that (E,M,N,T) has low accuracy for both our method and others since the letters are only differentiated by thumb position.

Previous gesture or ASL fingerspelling recognition systems considered only six gestures or 24 signs respectively. We increase the number of signs to 31 to accommodate most of the signs. Also, some previous methods did not separate subjects in training, validation, and testing. Experimenting with training and testing dataset from different subjects is important because then only the system's performance for a new subject can be measured. Therefore, we include experiments and report results where training and testing subjects are separated to demonstrate how the trained model can be used for anonymous subject. Lastly, although Pugeault *et al*. [10] reported that combination of color and depth improves the recognition rate, we only use depth to achieve better consistency to illumination changes and skin pigment differences and to avoid calibration process for general users.

5. Conclusions

We show the efficacy of using convolutional neural networks and a depth sensor for ASL fingerspelling recognition system. We collect and are in a process to share the dataset of depth images for ASL fingerspelling system. Our approach of classifying 31 signs of alphabets and numbers using depth image and CNNs achieves real-time performance and state-of-the-art accuracy even for different signers. We conclude that pre-training from auxiliary task of image classification from color images is helpful for apparently different type of input data such as depth image. The trained model and dataset is available on our repository `https://github.com/byeongkeun-kang/FingerspellingRecognition`.

References

[1] H. Cooper, B. Holt, and R. Bowden. Sign language recognition. *Visual Analysis of Humans*, pages 539–562, 2011. 1

[2] C. Dong, M. Leu, and Z. Yin. American sign language alphabet recognition using microsoft kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on*, pages 44–52, June 2015. 2, 4

[3] J. Isaacs and S. Foo. Hand pose estimation for american sign language recognition. In *System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on*, pages 132–136, 2004. 2, 4

[4] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding, 2013. "`http://caffe.berkeleyvision.org`". 3

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012. 3

[6] A. Kuznetsova, L. Leal-Taixe, and B. Rosenhahn. Real-time sign language recognition using a consumer depth camera. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 83–90, Dec 2013. 2, 4

[7] S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in british sign language. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 50–57, June 2009. 2

[8] J. Nagi, F. Ducatelle, G. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pages 342–347, Nov 2011. 1, 4

[9] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen. Sign language recognition using convolutional neural networks. In *Computer Vision - ECCV 2014 Workshops*, pages 572–578, 2015. 2

[10] N. Pugeault and R. Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1114–1119, Nov 2011. 2, 4, 5

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, April 2015. 2, 3

[12] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1371–1375, Dec 1998. 1

[13] A. S. L. University. Fingerspelling. "`http://www.lifeprint.com/asl101/fingerspelling/`". 2

[14] M. Van den Bergh and L. Van Gool. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 66–72, Jan 2011. 1, 4

[15] Wikipedia. American manual alphabet. "`http://en.wikipedia.org/wiki/American_manual_alphabet`". 2

[16] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the kinect. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, ICMI '11, pages 279–286. ACM, 2011. 2