15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Recognition of JSL Finger Spelling Using Convolutional Neural Networks

Hana Hosoe[†] and Shinji Sako[†‡]
[†]Department of Computer Science and Engineering
[†‡]Frontier Research Institute for Information Science
Nagoya Institute of Technology
466-8555, Gokiso-cho, Showa-ku Nagoya, Japan
{hosoe, sako}@mmsp.nitech.ac.jp

Bogdan Kwolek
Department of Computer Science
AGH University of Technology
Kawiory 21, 30-059 Krakow, Poland
bkw@agh.edu.pl

Abstract

Recently, a few methods for recognition of hand postures on depth maps using convolutional neural networks were proposed. In this paper, we present a framework for recognition of static finger spelling in Japanese Sign Language. The recognition takes place on the basis of single gray image. The finger spelled signs are recognized using a convolutional neural network. A dataset consisting of 5000 samples has been recorded. A 3D articulated hand model has been designed to generate synthetic finger spellings and to extend the real hand gestures. Experimental results demonstrate that owing to sufficient amount of training data a high recognition rate can be attained on images from a single RGB camera. The full dataset and Caffe model are available for download.

1. Introduction

There exists communication barrier between hearing impaired people and hearing people, which can be reduced by the use of adequate information technologies. There are few situations they could be correctly interpreted during interpersonal communication, still the most popular method of communication of the hearing impaired is sign language. Rapid progress in information technologies results in that lot of information such as subtitles for pictures, e-mail, and so on are provided visually. However, the hearing impaired seems to feel some difficulties to get information from the written Japanese. One of reason is that Japanese Sign Language (JSL) as their mother tongue has a quite different grammar system from Japanese spoken language.

Currently, translation to/from sign language is done by human and therefore it is expensive, due to requirement of experienced expert. While automatic speech recognition has now advanced, development of automatic sign language recognition is still in underway [4].

Sign language recognition aims to develop algorithms and methods to correctly identify a sequence of demonstrated signs and to translate its meaning [3]. In order to recognize sign, it is necessary to capture several features of sign language: manual makers such as handshape, hand orientation, location and movement expressing lexical meaning. In addition, non-manual signals such as facial expression and head position group sentences into different types. Despite significant progress [4,5], recognition of hand shapes is still a challenging problem because hand shape has high degree of freedom and undergoes considerable influence of individual distinctions in expressing of finger spellings.

This paper is devoted to hand shape recognition for automatic JSL translation. Such systems require input devices to capture the essential features of sign. There are two major approaches to capture hand postures and movement [3]. One approach is vision-based, and the other is sensor-based, where devices like data-glove are employed. The vision-based methods are unobtrusive, but they still are prone to errors due to capturing not only hand postures but also whole body motions. Moreover, when the posture recognition is on the basis of small hands at images the lighting conditions may affect the recognition performance considerably.

There are three main groups of methods utilized in estimation of hand poses, namely template-based methods, model-based methods and machine learning methods, as well as different combinations of the above mentioned techniques. Recently, a considerable attention is devoted to depth-based hand pose estimation [4,6,7,9]. One of the most promising results were obtained using convolutional neural networks (CNNs) on depth maps [5]. It is worth noting, that there are only few papers that report results obtained on the basis of CNNs [4].

Training CNNs from scratch, i.e. full training very often is connected with several difficulties [1]. It is a difficult task since it requires a large amount of labeled training data as well as a great deal of expertise to assure appropriate convergence. One of the promising approaches to cope with insufficient amount of properly labeled data for training consists in fine-tuning a CNN, which has been pre-trained in advance using, for example, a large set of labeled natural images. However, in many practical applications, including recognition of finger spelling, the differences between natural and hand images may advise against such a knowledge transfer. Therefore, in our approach we employ an articulated 3D hand model to render hands in the requested poses. This way, a considerable number of training images can be generated in short time and at low cost. Moreover, in this approach the ground-truth data are available. What's more, for the finger spellings that are difficult to recognize we can generate additional data to reflect the subtle differences between close words in JSL.

In this work, we propose an efficient framework to recognize the finger spellings from Japanese Sign Language using single RGB camera. For the whole alphabet consisting of 41 words we recorded 5000 images with hand gestures, which were expressed by ten performers. A few thousands

of images were generated using our 3D articulated hand model. A convolutional neural network has been then trained on such a dataset. Experimental results demonstrate that the proposed system for finger spelling recognition achieves high recognition accuracy.

To the best of our knowledge, CNNs were very rarely used for finger spelled gesture recognition using RGB cameras, with an exception of work [8]. One of main causes is lack of suitable datasets. Our work differs from form the above work in that we recognize far larger number of words. We also demonstrate how skinned hand models can be utilized to generate data for proper training of CNNs for effective hand pose recognition.

2. Japanese Sign Language

Japanese Sign Language (JSL) is a visual language used by deaf community in Japan. In general, there is no single standard JSL. JSL is distinct from spoken and written Japanese both in grammar and lexicon. The grammatical system shares many similarities with other sign languages that are used in other countries.

The manual system of finger spelling in JSL is a signed representation of written Japanese syllabary Hiragana. In general, finger spelling can be used to express the word and proper noun such as name or place that are not expressible by sign language.

The JSL finger spelling is expressed mainly by five fingers of the single hand and the direction of the hand. Most of finger spellings are expressed through static postures, but some of them are dynamic postures. In addition, dullness, half dullness, and long sound are represented by dynamic postures. In this study, we only focused on static finger spellings. There are 41 static finger spellings in JSL. They are listed bellow, see also Fig. 1. that depicts the considered static hand poses:

'a',' i', 'u', 'e', 'o',ka', 'ki','ku','ke','ko',
'sa','shi','su','se','so',ta','chi','tsu','te','to',
'na','ni','nu','ne','ha','hi','hu','he','ho','ma',
'mi','mu','me','ya','yu','yo','ra','ru','re','ro','wa'.

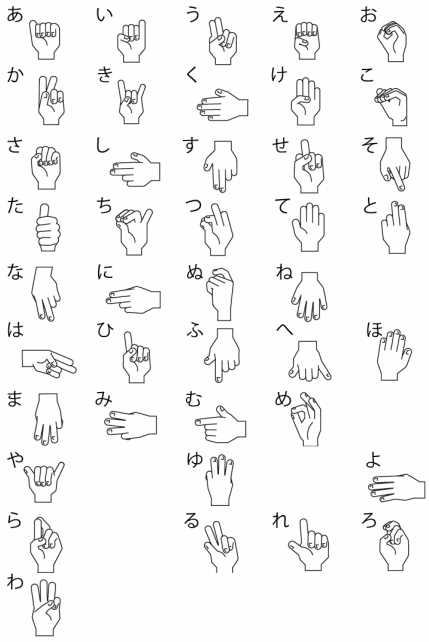

Figure 1. Finger spelling in JSL (static postures).

3. Hand Modeling

The human hand consists of 27 bones, 8 of which are located in the wrist. As shown on Fig. 2b, the 19 bones compose the palm and fingers. The bones in the skeleton constitute a structure of rigid bodies connected together by joints with one or more degrees of freedom.

A 3D hand model has been prepared in Blender software. The model has 26 degrees of freedom (DOF), see Fig. 2a, and consists of 21 bones, see Fig. 2b. The 3D mesh contains 1385 vertices and is composed of 2142 triangles. The 21 element skeleton (armature) is bound to such a 3D mesh. The root joint is placed in the wrist. The model has been textured using real photographs of the hand. The model has been exported to several data formats, including MD5 data format. This means that it can be animated in external programs.

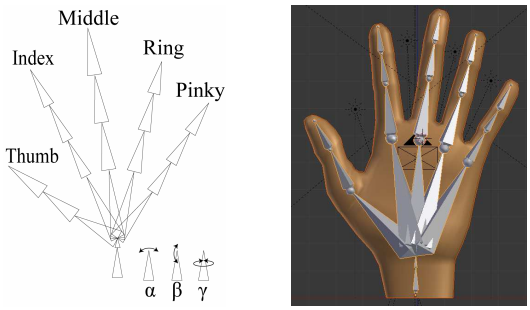

Figure 2. 26 DOF skinned hand model; kinematic tree (left), digital skeleton bound to the 3D mesh (right).

Figure 3. illustrates examples of synthetically generated hand poses for 'me', 're' and 'ra' words from the JSL. Posing of the hands in the requested poses has been done manually in Blender through moving and rotating the bones. The rendered images were then exported to .png data format. It is worth noting that after posing the hand to a pose representing a desired word, slight modifications from the basic pose can be obtained quickly and easily. Moreover, a technique called skeletal animation, which is based on interpolation allows us to obtain intermediate hand poses between such key frames, i.e. user defined hand poses.

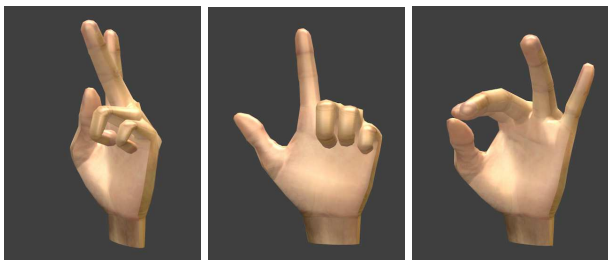

Figure 3. Examples of synthetic hands for 'me', 're' and 'ra'.

4. Dataset

The dataset has been recorded using an RGB color camera. The size of the input images was 640 x 480. The camera was located at the height of about 1 m from the

floor. Ten students of various nations attended in recording of the dataset. Each student was asked to reproduce as closely as possible the hand pose demonstrated on the image. The finger spelled words were executed one by one. Every tenth image of image stream acquired with 25 fps has been stored in the .png image format. During data recording the actors moved, rotated and changed slightly the hand pose to introduce some pose variations into the dataset. Every actor demonstrated the finger spelled word for about five seconds to record minimum ten images for each gesture. This way, for the whole alphabet consisting of 41 words we recorded 5000 images. Afterwards, skin color models were used to extract the hands on color images. In order to eliminate small skin-like pixels not being hands we executed connected components algorithm. The components, which area was below a preset threshold value were omitted in further hand segmentation. The binary images obtained in such a way were morphologically closed and then eroded to eliminate small holes in the hand areas. Finally, the binary masks were used to segment the hands in color images. The wrist positions were then manually selected in all images to place the wrists at predefined positions in sub-images of size 64 x 64. The hands extracted in such a way were then shifted, rotated and scaled to generate additional 1000 images. Figure 4. demonstrates sample images from our image collection. To preserve compatibility with images generated synthetically all images have uniform background.

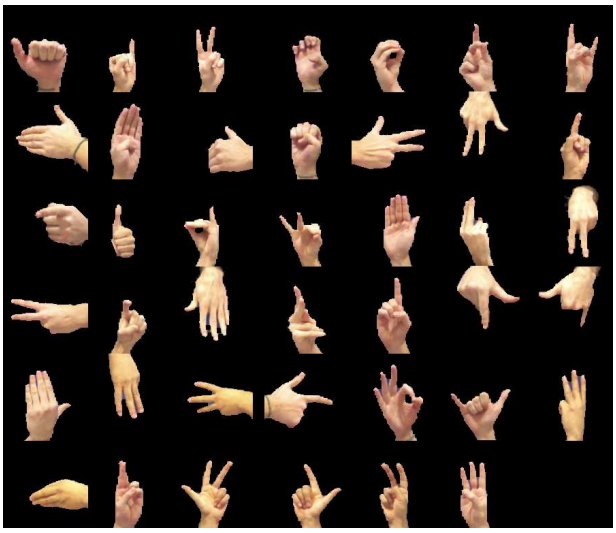

Figure 4. Sample images that were used to construct dataset.

The 3D hand model was then utilized to render the images of the hand in the requested poses. In skeletal animation, also known as s*kinning,* the way a 3D model animates is defined by its underlying skeleton (armature). A skeleton is defined by a hierarchy of special points called joints. Those joints are defined by their position and rotation. We utilized the Blender software to manually pose the hand's armature in order to create the requested words of JSL. For each finger spelled word we rendered ten images, which were saved in the .png image format. The hand images were then rescaled to images of size such that the dimensions of the hands were similar to hand sizes in the images acquired by the camera. In total 410 images generated in such a way were then rotated and scaled to in order to generate additional 1000 images.

The gesture animations were also stored in MD5 data format, which uses a skeletal system to do the animation. The MD5 model format was used by several commercial game projects including ID software's Doom 3. The mesh data and animation data are separated in distinct files, one for loading the MD5 model, and one for animation. These are ASCII files and are human readable. The MD5 file format represents rotations in unit quaternions, which are of norm one. A program in C++ language was prepared to parse the MD5 data format and to perform OpenGL-based rendering of the hand model. The MD5 animation data were then modified and then used for rendering 1000 color images. Afterwards, all color images were converted to grayscale images. Finally, they were rescaled to images of size 32 x 32 and then stored in the dataset. The whole dataset consists for 8000 grayscale images of size 32 x 32. The JSL finger spelling dataset is freely available at: http://home.agh.edu.pl/~bkw/research/data/mva/jsl.zip.

5. Convolutional Neural Networks

Convolutional neural networks (CNNs) have been used in the field of computer vision for decades [1,10]. Acknowledged as one of the top ten breakthroughs in recent years [2], CNNs have become a popular tool, now not only within the computer vision community but across various disciplines ranging from natural language processing to robotics. They belong to the family of Multi-Layer Perceptrons (MLPs) and may consist of four types of layers: convolutional, pooling, normalization, and fully connected layers. In a CNN convolutional layer, synaptic weights can be (shared) by certain neurons. One of the benefits of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units. The convolutional layer can be perceived as a set of local filters responsible for identifying certain characteristics of input images. To detect such local structures, each node in the convolutional layer is connected to only a small subset of spatially connected neurons in the input image. In order to permit seeking for identical feature, the connection weights are shared between the nodes in the convolutional layers. Each set of shared weights is called a convolution kernel. Thus, a convolutional layer consisting of n kernels learns to detect n local features, whose strength is then projected onto the resulting feature maps. In order to reduce the computational overload and to obtain a hierarchy of the features, the convolution layers are followed by pooling layers. The max pooling layers are capable of reducing the size of feature maps through picking the maximum feature response in overlapping or non-overlapping local neighborhoods. Through discarding the strict position of maximum responses this operation increases the translation invariance. This means that the same, i.e. pooled features will be active even when the image undergoes small shifts. A feature map in a sub-sampling layer is connected to one or more planes in the following convolution layer. In the last convolution layer, each plane has a connection with exactly one preceding feature map. This layer employs convolution masks, which are of the as its input feature maps. Thus, every plane in the last convolution layer will have single scalar output.

6. Experimental Results

A convolutional neural network has been trained on gray images of size 32 x 32. For such image size there is no considerable difference between the hand images acquired by the camera, see also Fig. 2 that contains images of size 64 x 64, and the synthetic images, which were generated on the basis of our 3d hand model, see also Fig. 3., and then rescaled to images of size 32 x 32. The grayscale images I0 of size 32 x 32 were utilized to train the CNN network consisting of six layers, denoted as C1, S2, C3, S4, C5, and C6. C1, C3, C5 and C6 are convolution layers, whereas S2 and S4 are the sub-sampling layers. The C6 has softmax output layer that serves as the output of the neural network. As we already mentioned, the input is an 32 x 32-pixel gray image. By applying twenty 5 x 5-pixel convolution kernels patches on I0, twenty 28 x 28-pixel feature maps are generated in C1 layer. Then, each feature map in C1 undergoes 2 x 2-pixel sub-sampling, resulting in twenty 14 x 14-pixel feature maps in S2. Executing fifty 5 x 5-pixel convolutions on the twenty feature maps in S2 results in fifty 10 x 10-pixel feature maps in C3. Finally, 2 x 2-pixel sub-sampling executed on the feature maps in C3 results in fifty 5 x 5-pixel feature maps in S4. The following C5 layer contains 500 4 x 4-pixel convolution kernels. The last C6 layer consists of 2 x 2-pixel convolution kernels that generate 41 softmax outputs. The S2 and C5 outputs are fed to ReLU layers, whereas S4 and C5 have 50% dropouts.

In a first stage of experimental evaluation, the recognition of 41 static hand poses was realized using real-images only, i.e. using a dataset consisting of 5000 hand images plus 1000 images that were obtained by shifting and rotating the hand images. On a dataset consisting of 80% training samples and 20% testing samples of the dataset we obtained almost 93% accuracy of classification. Afterwards, we extended the dataset about synthetic images. We divided the dataset into train data and test data such that the synthetically generated data appear only in training data. The accuracy on such an extended dataset was significantly better. On a test dataset consisting of 1185 samples, 1161 samples were classified correctly, and the classification accuracy was equal to 98%.

In the next stage we performed experiments with images of size 28 x 28 and 36 x 36. Slightly better results were obtained but it appears that the improvement is not statistically significant. Finally, we experimented with convolutional neural network consisting of larger number of layers. However, even though with the 3d hand-model based generated finger spelling the number of training samples is apparently not sufficient for the deep learning approach.

The convolutional neural network has been configured and learned using matConvNet library. The learning rate has been set to 0.001. The final convolutional neural network has been trained in 40 epochs using batch size set to 100. The model trained in such a way has been converted to Caffe format. The trained model and source code for evaluating of the classification accuracy is available at: http://home.agh.edu.pl/~bkw/research/data/mva. The data processing for extracting hand in color images, posing the wrist at requested poses in the sub-images of size 64 x 64 has been implemented in C++ with support of OpenCV library.

7. Conclusions

In this paper we demonstrated a framework for recognition of static finger spellings on images acquired from a single RGB camera. The recognition of hand gestures is done using a convolutional neural network, which has been trained using both real and synthetic images. We recorded 5000 images with static finger spellings from Japanese Sign Language. A few thousands of synthetic images for training were then generated from such a repository of images as well as on the basis of our skinned hand model. We demonstrated experimentally that high recognition rate can be obtained thanks to sufficient amount of properly prepared data for the training. In consequence we demonstrated experimentally, that CNN-based methods that rely only on color and even gray images can achieve promising results in recognition of JSL finger spelling.

There are a number of future research directions. First of all, we would like to generate more training samples using our articulated hand model. The integration of learned hand models with 3d model based techniques for estimating the hand pose using both the engineered and learned features would be one of the possibilities for further improving the recognition performance.

Acknowledgement. This work was supported by JSPS KAKENHI Fostering Joint International Research (15KK0008), NITech Grant for Global Initiative Projects as well as Polish National Science Center under a research grant 2014/15/B/ST6/02808 and a grant 11.11.230.124.

References

[1] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, P. Vincent: "The difficulty of training deep architectures and the effect of unsupervised pre-training," in Proc. Int. Conf. Artif. Intell. Stat., pp. 153–160, 2009.

[2] Y. LeCun, Y. Bengio, G. Hinton: "Deep learning," Nature 521, 436–444, 2015.

[3] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, X. Twombly: "Vision-based hand pose estimation: A review," *Comput. Vis. Image Underst.,* vol. 108, no. 1-2, pp. 52-73, 2007.

[4] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, D. Ramanan: "Depth-Based Hand Pose Estimation: Data, Methods, and Challenges," ICCV, pp. 1868-1876, 2015.

[5] J. Tompson, M. Stein, Y LeCun, K. Perlin: Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. *ACM Trans. Graph.,* 33, 2014.

[6] Z. Ren, J. Yuan, J. Meng, Z. Zhang: "Robust Part-Based Hand Gesture Recognition Using Kinect Sensor," in *IEEE Trans. on Multimedia*, vol. 15, no. 5, pp. 1110-1120, 2013.

[7] A. Kuznetsova, B. Rosenhahn: "Hand Pose Estimation from a Single RGB-D Image," ISVC (2), pp. 592-602, 2013.

[8] J. Nagi *et al*.: "Max-pooling convolutional neural networks for vision-based hand gesture recognition," *IEEE Int. Conf. on Signal and Image Proc. Appl.*, pp. 342-347, 2011.

[9] Ch. Xu and L. Cheng. "Efficient Hand Pose Estimation from a Single Depth Image," *Proc. IEEE Int. Conf. on Computer Vision*, pp. 3456-3462, 2013.

[10] B. Kwolek. "Face Detection Using Convolutional Neural Networks and Gabor Filters," *Proc. 15th Int. Conf. on Art. Neural Networks*, LNCS, Springer, pp. 551-556, 2005.