# HAND POSTURE RECOGNITION BASED ON BOTTOM-UP STRUCTURED DEEP CONVOLUTIONAL NEURAL NETWORK WITH CURRICULUM LEARNING

*Takayoshi Yamashita*

Chubu University
yamashita@cs.chubu.ac.jp

*Taro Watasue*

Tome R&D
watasue@omm.ncl.omron.co.jp

## ABSTRACT

Hand posture recognition has tremendous potential in the field of natural user interactions. There were many advances in research in recent years but there are still limitations regarding its usage in unfavorable live situations where hand posture variation, illumination change or background complexity are an issue. In cases like these, recognizing the hand posture is a difficult task. As such, we considered reducing the difficulty of the task by using curriculum learning with intermediate information. We proceeded to divide the complex architecture of the hand posture recognition task into two easier ones: 1) Extraction of the hand shape under clutter background with illumination change, 2) Recognition of the hand posture from a binary image. In order to do so, we propose here a bottom-up structured deep convolutional neural network incorporating a special layer for binary image extraction. Our proposed method also employs state-of-the art techniques for deep learning to obtain generalization. As a result, we achieved better recognition performances of the hand posture under clutter background compared to the baseline method.

***Index Terms***— deep learning, curriculum learning, hand posture, binarization, maxout

## 1. INTRODUCTION

Hand posture recognition has tremendous potential for natural user interactions. There are numerous application examples, such as remote-less control for TVs, PCs, medical devices or industrial equipment, and it can also be used for communication in robotics. Hand posture recognition has significantly progressed through advanced research for many years [**?**] [**?**] [**?**] [**?**] but there are still limitations to its practical real-environment application with regards to issues of robustness and hand shape variations. The approach for hand posture recognition can be categorized into a device-based approach and a vision-based approach. There is a long history of development for the device-based approach, and many methods of hand posture recognition employ a wearable glove-like device, where each finger joint is color-identified to facilitate extraction. As an example, Wang [**?**] used Hausdorff-like distances as metric to find the nearest neighbor in a database for posture recognition from a single camera. However, even if this method is robust in indoor scenes, it sacrifices flexibility and is limited for generalization. For the vision-based approach, Kolsch [**?**] proposed a detection based method employing a Viola-Jones detector. This method can recognize six types of hand posture and supports rotation up to 15 degrees. Among others, Bretzner [**?**] proposed a multi-scale color feature to recognize hand posture and Liu [**?**] also used a color feature to extract skin blob along with several geometric constraints to separate the fingers and recognize the hand posture using these geometric features through SVM of each posture. However, the existing vision-based approach has issues with posture variation and background complexity: the hand shape can be viewed as a kind of non-rigid object where there can be many variations for a same posture, and even in indoor scenes there can be numerous kinds of background, including some where its color will be close to the color of the skin. On one hand, the detection method can overcome the issues of skin color and slight background variation if it operates with grayscale images, but it require preparation with detectors in order to cover the angles of the hand posture. On the other hand, the method using skin color can overcome the posture variation, but it will not be robust to skin color variation and background complexity.

We attempted to address these issues with a deep learning approach. The deep learning approach is studied in various fields and lead to remarkable benchmark performances in areas like object recognition or speech recognition. Likewise, a deep convolutional neural network is especially suitable for image processing, and it roughly mimics the nature of the mammalian visual cortex. The parameters and their optimization, however, can be complex for such network. In order to address the issues mentioned above for hand posture recognition, it is primordial that the layer structure displays sufficient robustness. We propose here a method with a bottom-up structured deep convolutional neural network in order to break down these complex issues into a constellation of easier ones. This bottom-up network will extract the hand segments and then recognize hand postures from a grayscale image in-

stead of just recognizing the hand postures directly. The hand can display numerous variations and its appearance is affected by lighting conditions under clutter background, making it a difficult task to recognize its posture directly. However, the bottom-up network will generate a binary map through its first several layers in order to ignore background complexity and changes in illumination. The then just-made-easier task of hand posture recognition is done by giving a binary map to the later layers.

The rest of this paper is organized as follows: The convolutional neural network is briefly reviewed in Section 2, our proposed approach is described in Section 3 and the experiment results for hand posture recognition is given in Section 4. We discuss about the effectiveness of the bottom-up structure in Section 5 and present our conclusions in Section 6.

## 2. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks [**?**] have an alternate succession of convolutional layers and subsampling layers, based on the knowledge of local receptive fields discovered by Hubel [**?**]. There are several types of layers, such as input layers, convolutional layers, pooling layers or classification layers. The input layer has, besides the raw data, edge and normalized data as input. The convolutional layer has $M$ kernels with size ($Kx \times Ky$) and filters them in order to input data. The filtered responses from all the input data are then subsampled in the pooling layer. Scherer [**?**] found that max-pooling can lead to faster convergence and improved generalization while Boureau [**?**] analyzed theoretical feature pooling. Max-pooling can output the maximum value in certain regions such as a $2 \times 2$ pixel. The convolutional layer and pooling layer are laid alternatively in order to create the deep network architecture. Finally, the output feature vectors from the last pooling layer are used in the classification layer. The classification layer will output the probability of each class through softmax connection of all the nodes with weights in the previous layer. Unlike Belief neural networks, convolutional neural networks assume supervised learning where filters are randomly initialized and updated through backpropagation [**?**] [**?**].

Krizhevsky [**?**] applied deep convolutional neural networks to object recognition benchmark in order to classify 1000 different classes and obtained top-level performances. The network consisted of five convolutional layers with max-pooling and three fully connected layers, with softmax used on the final classification layer. The fully connected layers linked the neurons with all the neurons in the previous layers, and each connection had weight and bias. Dropout was used to reduce the overfitting in the fully connected layers. Schmidhuber [**?**] had great success using multiple deep convolutional neural networks with different preprocessing, such as contrast normalization or histogram equalization. These existing methods all focus on the architecture of the deep network in order to achieve significant performance results.

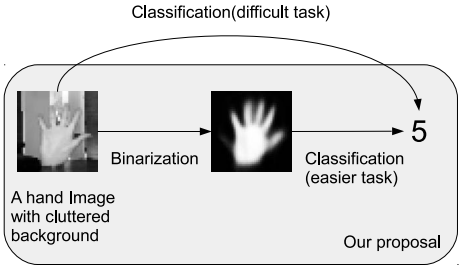


**Fig. 1**. Our approach.

## 3. PROPOSED METHOD

As mentioned above, hand posture recognition in natural grayscale image is a extremely difficult task. we divide hand posture recognition task into two tasks as shown in Fig.**??**, The binarization task is a hand segmentation from background. Then, the hand posture are classified from a binarized map could become a easier task than directly classifying hand posture. To realize the our idea, we propose a bottom-up structured deep convolutional neural network to recognize hand posture classes. This network has two characteristics: 1) It prevents overfitting. 2) It turns difficult tasks into easier ones by using intermediate information. First we will introduce the architecture of our proposed method and expand on how to prevent overfitting for the input layers, pooling layers and fully connected layers. Then we will describe the binarization layer inspired by Gulcehre's [**?**] notion of curriculum learning for networks.

### 3.1. Architecture of proposed method

The framework of our proposed method is shown in Fig.**??**. Our network contains eight layers. The first four are two sets of convolutional and pooling layers, followed by a fully connected layer, a binarization layer, a second fully connected layer, and finally a classification layer. The first convolutional layer has 32 kernels of size $5 \times 5$ with a stride of a pixel. The network receives the hand shape in grayscale image of $40 \times 40$ size. The output of the first convolutional layer is $36 \times 36 \times 32$. In the following pooling layer, each unit selects a maximum output from neighbors of a $2 \times 2$ pixel, then outputs subsamples to $18 \times 18 \times 32$. The max-pooling can lead to faster convergence and improve generalization []. However, unlike Krizhevsky and Schmidhuber, we employ maxout following max-pooling [**?**]. While the max-pooling selects the maximum output from the same map produced by a kernel and subsamples the map size to half, the maxout selects the maximum value from several maps and reduces their number. We divide the maps into $4$ groups and use maxout to obtain an output of $18 \times 18 \times 8$. The second convolutional layer takes as input the output of the first convolutional and pooling layer set with max-pooling and maxout. This layer has $32$ kernels of size $5 \times 5 \times 8$, and max-pooling and maxout is applied to its output. The size of the max-pooling and maxout is the

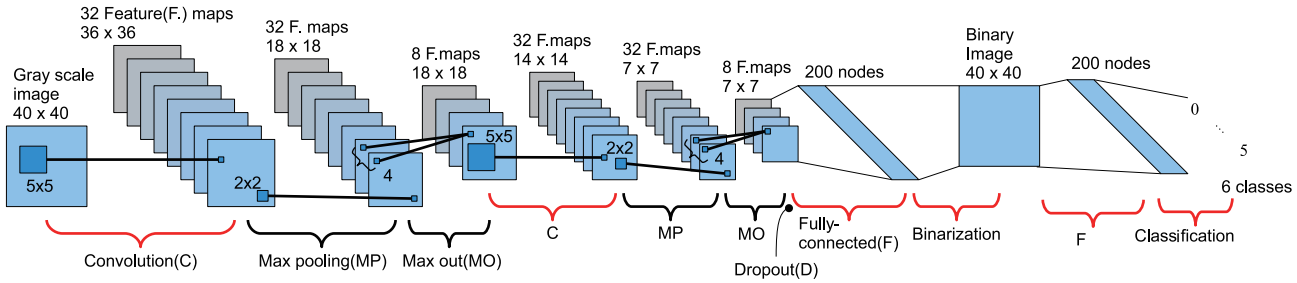


**Fig. 2**. Framework of our proposed method.

same as in first pooling layer, and the output is $7 \times 7 \times 8$. The fully connected layer receives 392 dimensional feature vectors by flattening the above output. It learns with dropout and output 200 dimensional feature vectors. These then are input in the binarization layer to obtain a binary image of the hand shape. This layer outputs the probability of each pixel of being part of a hand or not. The next fully connected layer takes the output of the binarization layer and output 200 dimensional feature vectors. The process in this layer is similar to decoding between the binarization layer and the previous fully connected layer if the same number of dimensional feature vectors is set. Finally, the classification layer produces a distribution of each class labels by using softmax. At the learning stage, all the parameters are optimized through back-propagation.

### 3.2. Input Layer

The network is trained with a huge amount of data to reduce overfitting. As it is difficult to collect a huge amount of labelled individual data, we use a data augmentation method to increase the amount of data from a limited dataset [**?**], but we apply elastic distortions to both the original grayscale image and the binary label data.

### 3.3. Pooling Layer

We employed maxout to avoid the many flaws of a traditional activation function design [**?**]. This results in a high representation compared to the ReLU nonlinearity proposed by [**?**]. The common activation function $h_i$ in $\boldsymbol{h} = (h_1, ...h_i, i \in I)$ is defined as Eq. (**??**)

$$h_i = \sigma(x^T W_i + b_i). \tag{1}$$

The $\sigma(\cdot)$ is the sigmoid function, $x$ is the input vector, $W_i$ is the weight and $b_i$ is the bias. The maxout selects the maximum value from several filtered output kernels as in Eq.(**??**)

$$h_i = \max_{j \in [1,k]} z_{ij}, \tag{2}$$

$$z_{ij} = x^T W_{ij} + b_{ij} \tag{3}$$

Unlike max-pooling, maxout selects the maximum output from several maps.

### 3.4. Fully Connected Layer

Dropout is an efficient method to reduce overfitting and improve generalization [**?**]. Dropout randomly samples hidden units with a probability of 50%. The features of the selected units are then used for optimization during each iteration of the learning process. Dropout is also used for learning of the fully connected layer.

### 3.5. Binarization Layer

Bengio [**?**] suggests that learning deep architectures is easier when some hints are given about the function that the intermediate information should compute. Following this, Gulcehre [**?**] proposes a method that divides difficult tasks into simpler detection and classification tasks using intermediate information. Inspired from these, we use a binarization layer to obtain intermediate information. Our goal is to recognize hand postures with shape variation and illumination change under clutter background. We simplify that difficult task by dividing it into two easier ones, a binarization and a classification task. While the binarization layer outputs as intermediate information the probability for each pixel of being part of a hand in order to ignore the illumination change and background complexity, the classification network uses this intermediate information in its learning process.

### 3.6. Network Learning

All parameters $\boldsymbol{W}$ such as the kernel elements, the weights between the units and the bias, are randomly initialized. We use backpropagation to update the parameters $\boldsymbol{W_{t+1}}$ in iteration $t+1$ as defined in Eq.(**??**),

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \epsilon_{t+1} \frac{\partial L}{\partial \boldsymbol{W}} \tag{4}$$

The update efficiency will decrease depending on the update time $t$ as defined in Eq.(**??**)

$$\epsilon_t = \frac{\epsilon_0 \tau}{\max(t, \tau)} \tag{5}$$

The $\epsilon_0$ is initially update-efficient and $\tau$ is the defined parameter. We use the L2 norm for the loss function $L$ in the binarization layer. However, softmax is used for the loss function $L$ in the classification layer learning.

**Fig. 3**. Example of evaluation dataset including 6 classes based on the number of fingers up.

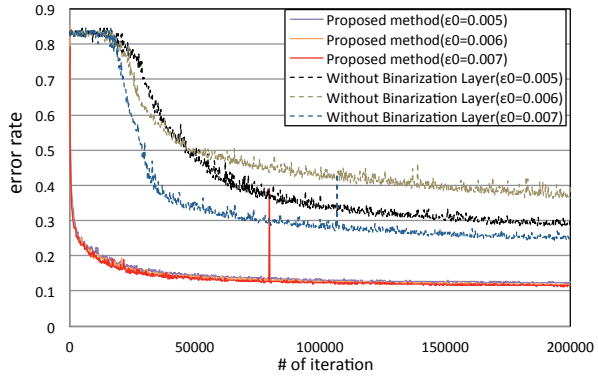


**Fig. 4**. Error rate for the iterations of each initial update coefficient.

## 4. EXPERIMENT

The hand postures are divided in six classes, as shown in Fig. **??**. Each hand posture class is identified by the number of up fingers displayed, with "0" for a closed fist and "5" for an open palm. Each learning image $I_o$ has a class label $y$ and a binary image $I_b$. We use 7500 images as seed for each class and obtain 200000 images after elastic distortions. The total iteration of updating parameters in convolutional neural network is 200000 with 10 mini-batches. The updating parameters, $\tau$ and $\epsilon_0$, are set to 20000 and 0.006 respectively. We learn the network on a NVIDIA GT640 2GB GPU. To evaluate the learned network, we collect images including shape variation in the same class and illumination changes under clutter background. The dataset contains 1600 images for each class. We compare our proposed method with the network that has not binarization layer as baseline. We also evaluate the performance for the initial update coefficient $\epsilon_0$. The performance results of our proposed method and of the one with no binarization layer are shown in Fig. **??**. We set the initial update coefficient from 0.005 to 0.007. Our proposed method obtain better results than the baseline with all update coefficients. While the baseline is sensitive to the update coefficients, our proposed method obtain similar performance even when changing the coefficient. In addition, our proposed method converges earlier than the baseline. It divides difficult tasks into easier ones using intermediate information based on the idea of curriculum learning. This result in a method that is not coefficient-sensitive and that achieve faster convergence.

**Table 1**. Confusion matrix: (a) is our proposed method, (b) is with no binarization layer.

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 91.2% | 8.1% | 0.4% | 0.1% | 0.1% | 0.2% |
| 1 | 9.8% | 84.6% | 5.0% | 0.3% | 0.2% | 0.0% |
| 2 | 0.3% | 6.3% | 83.6% | 8.3% | 1.3% | 0.2% |
| 3 | 0.0% | 0.2% | 8.3% | 84.3% | 7.1% | 0.1% |
| 4 | 0.1% | 0.1% | 0.9% | 7.5% | 90.0% | 1.4% |
| 5 | 0.0% | 0.0% | 0.1% | 0.1% | 0.9% | 99.0% |

(a)

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 80.3% | 14.3% | 4.5% | 0.6% | 0.2% | 0.0% |
| 1 | 17.5% | 64.9% | 15.5% | 1.0% | 1.0% | 0.1% |
| 2 | 1.6% | 13.9% | 67.5% | 12.7% | 4.1% | 0.2% |
| 3 | 0.3% | 1.1% | 17.3% | 62.3% | 18.3% | 0.8% |
| 4 | 0.2% | 0.7% | 4.1% | 9.8% | 80.8% | 4.4% |
| 5 | 0.1% | 0.2% | 0.1% | 0.3% | 1.8% | 97.6% |

(b)

**Fig. 5**. Output of binarization layer: the 1st and 3rd columns are original images and the 2nd and 4th are output images.

The confusion matrix is shown in Table **??**. As a whole, our proposed method obtain better performances for all classes compared to the baseline.

## 5. DISCUSSION

The output of the binarization layer with 130000 iterations and $\epsilon_0 = 0.007$ is shown in Fig.**??**. Even with a grayscale image input, the network manage to extract the hand region under clutter background. Moreover, it is robust to not only background but also to illumination change. The binarization layer succeeds in extracting the hand region and simplifyd the difficult recognition task from a grayscale image to the much easier one from a binary image.

## 6. CONCLUSION

We propose a bottom-up structured deep convolutional neural network. This network is based on the notion of curriculum learning that divided difficult tasks into easier ones with intermediate information. We use a binarization layer to obtain the intermediate information and output binary images under clutter background. As a result, the network recognize the hand posture better than the baseline method with no binarization layer. We will apply the proposed method to other segmentation based object recognition as a future work.

## 7. REFERENCES


[1] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex", Journal of Physiology,pp.160:106–154, 1962.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation", In Parallel Distributed Processing:Explorations in the Microstructures of Cognition, vol.1, pp.318–362. MIT Press, 1986.

[3] Y. LeCun, L. Bottou, Y. Bengio and P.Haffner, "Gradient-basedlearning applied to document recognition", In Proceedings of the IEEE, Vol.86, No.11, pp. 2278–2324, 1998.

[4] L. Bretzner, I. Laptev and T. Lindeberg, "Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering", In IEEE Conference on Automatic Face and Gesture Recognition (FGR2002), pp.423–428, 2002.

[5] P.Y. Simard, D. Steinkraus, and J.C. Platt, "Best practices for convolutional neural networks applied to visual document analysis", In International Conference on Document Analysis and Recognition, Vol.2, pp. 958–962, 2003.

[6] M. Kolsch and M. Turk, "Robust hand detection", In IEEE Conference on Automatic Face and Gesture Recognition (FGR2004), pp.614–619, 2004.

[7] C. Manresa, J. Varona,R. Mas and F. Perales, "Hand Tracking and Gesture Recognition for Human-Computer Interaction", Electronic letters on computer vision and image analysis Vol.5, No.3 , pp.96-104, 2005.

[8] A. Just, Y. Rodriguez and S. Marcel, "Hand Posture Classification and Recognition using the Modified Census Transform", In IEEE Conference on Automatic Face and Gesture Recognition (FGR2006), pp.351–356, 2006.

[9] H. Francke, J. Ruiz-del-Solar and R. Verschae, "Real-time hand gesture detection and recognition using boosted classifiers and active learning", In the 2nd Pacific Rim conference on Advances in image and video technology (PSIVT07), pp.533–547, 2007.

[10] Q. Chen, N. D. Georganas and E. M. Petriu, "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features", In IEEE Conference on Instrumentation and Measurement Technology Conference (IMTC2007), pp.1–6, 2007.

[11] S. Duffner and C. Garcia, " An online backpropagation algorithm with validation error-based adaptive learning rate", In International Conference on Artificial Neural Networks (ICANN), Vol.1, pages 249–258, 2007.

[12] R. Y. Wang and J. Popovic, "Real-time hand-tracking with a color glove", ACM Transaction on Graphics, Vol.28, No.3, p.63, 2009.

[13] D. Scherer, A. Muller, S.Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition", In International Conference on Artificial Neural Networks, 2010.

[14] Y. Boureau, F. Bach, Y. LeCun and j. Ponce, "Learning Mid-Level Features For Recognition", In IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2010.

[15] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification", arxiv preprint arXiv:1202.2745, 2012.

[16] L. Liu, J. Xing, H. Ai, X. Ruan, "Hand Posture Recognition Using Finger Geometric Feature", In International Conference on Pattern Recognition (ICPR 2012), pp.565–568, 2012.

[17] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet classification with deep convolutional neural networks", In Advances in Neural Information Processing Systems 25(NIPS2012), pp.1106-1114, 2012.

[18] Y. Bengio, "Evolving culture vs local minima", arXiv preprint arXiv:1203.2990, 2012.

[19] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", arXiv preprint arXiv:1207.0580, 2012.

[20] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville and Y. Bengio, "Maxout networks", arXiv preprint arXiv:1302.4389, 2013.

[21] C. Gulcehre and Y. Bengio, "Knowledge matters: Importance of prior information for optimization", arXiv preprint arXiv:1301.4083, 2013.