Synthetic Training of Deep CNN for 3D Hand Gesture Identification

Chun-Jen Tsai†, Yun-Wei Tsai, Song-Ling Hsu, and Ya-Chiu Wu
Dept. of Computer Science,
National Chiao Tung University,
Hsinchu, Taiwan
† cjtsai@cs.nctu.edu.tw

Abstract—**In this paper, we present some experiments and investigations on a synthetically-trained neural network for the 3D hand gesture identification problem. The training process of a deep-learning neural network typically requires a large amount of training data to converge to a valid recognition model. However, in practice, it is difficult to obtain a large set of tagged real-data for the training purposes. In this paper, we investigate the plausibility of combining a large set of computer-generated 3D hand images with few real-camera images to form the training data set for the 3D hand gesture recognition applications. It is shown that by adding 0.09% of real images to the synthetic training data set, the recognition accuracy are raised from 37.5% to 77.08% for the problem of identifying 24 classes of hand gestures of an unknown user whose hand was not used in the training data set. In this paper, we have shown that the effect of the few real images to the trained CNN models mainly falls upon the fully-connected layers.**

Keywords—Deep-learning neural networks; convolutional neural networks; hand gesture identification; 3D hand models

I. INTRODUCTION

Hand gesture recognition is the key technology of many man-machine interface applications. Popular schemes for hand gesture recognitions can be classified into the glove-based [1], the depth-based [2][3], and the vision-based methods [4]. The glove-based techniques require the users to wear some special gloves with attached sensors. Although these devices provide robust hand gesture information, they can be cumbersome to use and too expensive for general consumers.

For the past few years, it is shown that the deep convolutional neural network (CNN) models work very well for solving the image recognition problems [5]. However, the most difficult task in constructing a deep CNN model is to collect a large training data set that contains different aspects of the key features of the target objects. In general, acquiring a large amount of real data in the field and tagging these data offline are demanding and costly. In addition, the features that are important for the deep CNN model to learn the appropriate weight parameters are usually not known in advance. Therefore, it is difficult to conduct the data acquisition process in a well-controlled manner such that the training data set only contains the crucial samples that can enhance the learned model.

As a result, a common industry practice is to acquire as many training data as possible and manually tag these data without the knowledge of the deep learning process. However, there are prices to pay for such an approach. First of all, the larger the amount of training data, the longer it takes to train the recognition model. Secondly, without knowing precisely what features in the training data are important for the deep CNN to converge to the right model, a large data set may often contains outliers that cause overfitting and performance degradation of the recognition model.

In this paper, we investigate the possibility of using a mixed collection of synthetically generated data and few real-world data to train a CNN model. The advantage of using synthetically generated data is that a synthetically generated data from a parameterized model can be tagged automatically and the distribution of the data in the feature space can be well-controlled. Although there are previous researches that adopt synthetic data for CNN training [6], the application context is quite complex and, therefore, is not easy to get insights into the effect of synthetic training data on the real-image recognition problem. The 3D hand gesture identification problem is used in this work to investigate the idea of training a CNN model using synthetic data. Surprisingly, by adding 0.09% of real images into the training data set of synthetic images, the recognition accuracy can be raised from 37.5% to 77.08% for the identification of 24 classes of hand gestures.

This paper is organized as follows. Section II formulates the hand gesture recognition problem. Section III discusses the design of the 3D model for the generation of the synthetic hand gesture training images. Section IV describes the deep CNN model and the training parameters used in this

978-1-5090-6536-3/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCAIRO.2017.40

work. Section V presents the experiments and some analyses on the results. And finally, section VI concludes this paper.

II. Problem Formulation

The 3D hand gesture identification problem can be solved in 2D domain where 2D-feature extraction and matching are used to determine the type of gestures. However, for the man-machine interaction application in 3D space, it is not good enough to simply determine the type of gesture from the observed 2D images. For example, the manipulation of 3D virtual objects in augmented virtual reality, fine remote control of drones using hand gestures, and 3D editing of virtual models in front of a stereo display all require the estimation of the 3D parameters of the user's hands from the 2D sensor images. Such 2D-to-3D estimation problems are ill-posed problems that cannot be solved without reasonable constraints.

For 3D hand tracking and recognition, researchers have been using a realistic 3D hand model as the constraint to solve the ill-posed problem [7][8]. These frameworks adopt an iterative analysis-by-synthesis approach. In short, some initial parameters of the 3D hand model are used to generate a synthetic 2D projection of the hand image. The synthetic 2D image is then compared against the camera image. The differences between these two 2D images are used to infer the required 3D model parameter updates such that the differences between the synthetic image and the real camera image are minimized. A major problem with such an approach is in the determination of the initial parameters. Without good initial guesses of these parameters, the iterative analysis-by-synthesis approach will not converge to a true solution. Using a deep CNN model to obtain the initial 3D hand pose would be an elegant way to find initial parameters for such frameworks.

III. Synthetic Training Data Design

In this paper, the 3D hand gesture identification is used to verify whether it is possible to train a deep CNN model with synthetic data. Therefore, it is important to design a 3D model of human hands such that the synthetically generated hand images are representative for the real images captured by a live camera. In this section, we present the design of the 3D hand model and tags.

A. The 3D Hand Model and Its Control Parameters

Each finger of a human hand has four joints: distal interphalangeal (DIP), proximal interphalangeal (PIP), metacarpophalangeal (MP), and abduction (ABD). The MP and the ABD joints are collocated at the same point that allows the base of each finger to move with two degrees of freedom (Fig. 1). Although each finger has four degrees of freedom, the angles each joint can move are constrained. There are two types of constraints: static and dynamic. The static constraints are the minimal and maximal angles of a joint moving alone. A set of static joint constraints are proposed in [9]. However, in this work, we have changed the static constraints of the ABT joints of the index, the middle, the ring, and the little fingers from ±60° to ±30° to make the 3D hand model look more realistic.

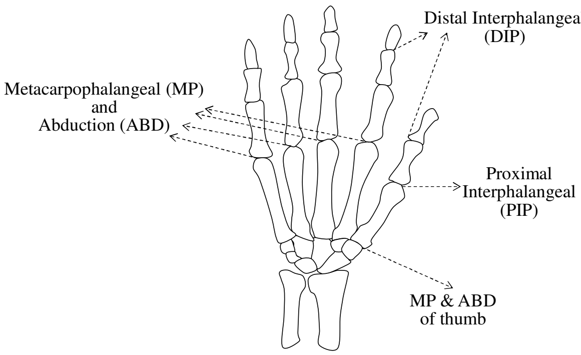


Fig. 1. The skeleton and the joints of a 3D hand.

The dynamic constraints on finger joints put additional limits on the possible angles of a joint due to the pose of other joints. Lee and Kunii [10] proposed some dynamic constrains among the finger joints. In this paper, we use most of the constraints proposed in [10] except for the dynamic constraints between the MP and the ABD joints. In this work, if the angles of the MP and the ABD joints are θ_{ABD} and θ_{MP}, respectively, the maximal dynamic angle of the ABD joint, $D_{\mathrm{MAX}}(\theta_{ABD})$, is constrained by the maximal static angle of the ABD joint, $S_{\mathrm{MAX}}(\theta_{ABD})$. Eq. (1) is used in this paper to enforce this constraint.

$$D_{MAX}(\theta_{ABD}) = \left(1 - \frac{\theta_{MP}}{S_{MAX}(\theta_{MP})}\right) \times S_{MAX}(\theta_{ABD}) \qquad (1)$$

In this work, the open source 3D graphics package Blender [11] is used to create the 3D skeleton of a hand following the dynamic joint constraints as described in [10] and by Eq. (1). The adoption of a realistic 3D hand model can be used to reduce the dimension of the solution space of the ill-posed 2D-to-3D recognition problem so that the inverse solution of a 3D hand pose can be obtained from the observation of a 2D camera image. For example, in Fig. 2, only a particular set of joint angles of the 3D model can produce the 2D camera image. Therefore, the observed 2D camera image alone is sufficient to estimate the twenty 3D hand model parameters. However, in this paper, the goal is to use a deep CNN to solve the recognition problem. Therefore, the 3D hand model is used for the automatic generation of tagged training data.

Before we discuss the design of the training data, we must first define the hand gesture classes (i.e., the tag) used in this paper. The hand gestures are divided into 24 classes. Each class describes whether each finger is straight, half-bent, or fully bent. There are totally 24 classes of gestures used in this paper. Sample frontal snapshots of all the hand

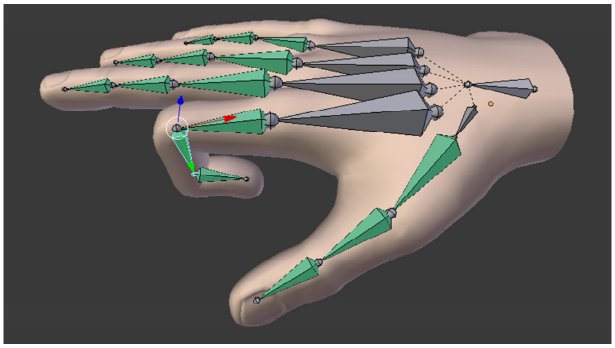


Fig. 2. The 3D hand model used in this paper posing a gesture. The tag of this gesture is #04 (see Fig. 10), and the tag represents the angles of the 20 finger joints.

gesture classes are shown in Fig. 10. The numbers of hand postures that belong to each gesture class are practically limitless. For a half-bent or fully-bent finger, at least one of the DIP, PIP, or MP angles is not zero. For a straight finger, only the ABD joint can have a non-zero value.

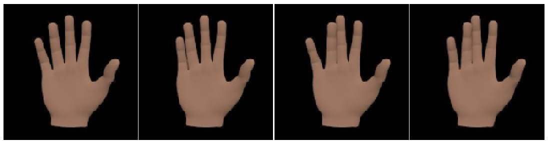
B. Design of the Synthetic Training Data

For the automatic generation of the tagged training data, we must define the controllable parameters of each finger classes. The tag of a training image is simply the ID of the gesture class it belongs to. In each gesture class, the following parameters are used to generate different training images that have the same tag: the angles of the ABD joints of each finger, the rotations of the hand with respect to the wrist, the position and the scale of the hand. These controllable parameters allow us to produce an enormous amount of training images for each gesture class (tag). However, to avoid long training time, only a few parameters for each class are used to generate the training images.

For the ABD joint angles, each finger has one to three different values depending on the agility of the finger. The joint angles and some samples of the training images generated using these control parameters are shown in Table I. There are totally 24 possible combinations of ABD joints.

For the rotation angles of the hand with respect to the wrist, we use –10°, 0°, and 10° for all three axes X, Y, and Z. Some samples of the training images that generated from these control parameters are shown in Fig. 3.

TABLE I. THE ABD PARAMETERS FOR THE TAGGED TRAINING IMAGES.

	thumb	index	middle	ring	little
ABD angles	-10°, 0°, 10°	-10°, 0°	0°	0°, 10°	0°, 10°
					

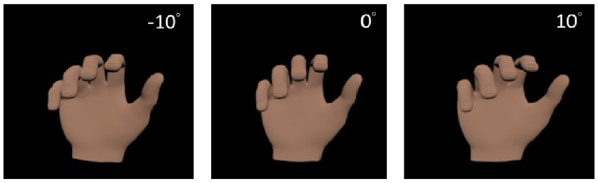


Fig. 3. Sample training images with wrist rotations of –10°, 0°, and 10° along the Z-axis.

For the scales and positions of the hand, the control parameters are the offset to the Blender camera coordinate systems along the Y and Z axes. Note that we use a 3D model and the perspective projection to generate the 2D images, moving the model along the Z axis effectively changes the scale of the projected 2D hand image. This training data design allows some scale and position variations of the hand in the cropped detection image. In some sense, this part of the training image allows some flexibility in a deep CNN recognition system where the region proposal techniques are used to select the candidates of recognition. Seven different y-z pairs used in this work and some training images generated by these control parameters are shown in Table II.

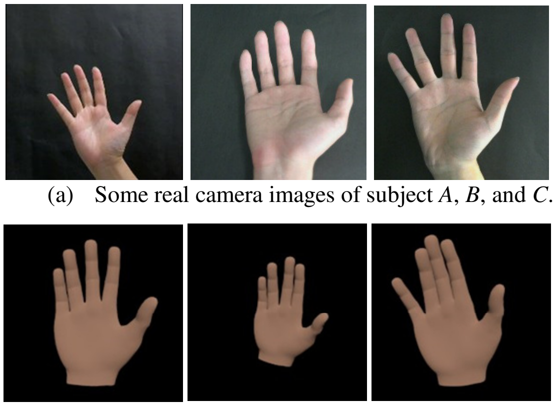
The total combinations of these controllable parameters are 24×27×7 = 4536 training images per gesture class. For hand gesture recognition applications of 32 classes, this is more than enough. Later, some experiments will show that for the 32 classes gesture recognition problem, each class only need 538 synthetic training images to achieve more than 90% accuracy rate for real image gesture recognition. However, if the application has to differentiate among fingers with different degrees of bending, the control parameters should be increased to generate even more training data.

IV. THE DEEP CNN USED IN THIS WORK

Since the goal of this paper is to verify the feasibility of using synthetically generated images to train a CNN model for a real image recognition task, we did not propose a new CNN model in this paper. Instead, we used the well-known

TABLE II. THE POSITION AND SCALE PARAMETERS FOR TRAINING IMAGES.

Y, z offsets (the numbers are in Blender units)			
0, 0	10, 3	10, 0	10, -3
20, 6	20, 0	20, -6	



(a) Some real camera images of subject *A*, *B*, and *C*.

(b) Synthetic images generated using Blender.

Fig. 4. Samples of the real and the synthetic images of a gesture.

AlexNet [5] network architecture provided in the Caffe project [12] for this work. In a training iteration of a CNN model, there are two stages, training and validation. The training stage is used to learn the weight parameters of the neuron connections and the validation stage is used to avoid overfitting. In this work, we do not have the validation stage because we use mainly the synthetic data for model training and the real data for model accuracy evaluation. It does not make sense to expect the validation stage to use synthetic data to determine whether the model is over-fitted for real data or not.

V. The Experimental Results

The setup of the experiments is as follows. A computer with Intel Core i7-4790K and 16GB memory running Ubuntu 16.04 LTS 64-bit is used as the target platform. An NVIDIA GTX 1060 graphic card with 6GB memory is used for the acceleration of the training process. The Caffe package with OpenCV 3.1.0, CUDA 8.0, and cuDNN 5.1 are installed. For the generation of the synthetic training images, Blender v 2.76b is used. For the acquisition of the real camera hand images, a Logitech C310 webcam is used. All the training images and real images are cropped and scaled to 200×200 pixels.

Both the synthetic training images and real training images have simple backgrounds so that the analysis and the interpretation of the experimental results can be focused on the hand area. In section V, we will show some test results when the background has some textures. Small scale and position differences of the hand images can be deal with by the trained CNN model because we have designed the training data to account for such variations as described in section III-*B*. The real-camera hand images are taken from three different test subjects, named *A*, *B*, and *C*. For the training phases, only subjects *A* and *B*'s hand images are added to the training data. For the accuracy tests, only subjects *B* and *C*'s hand images are used (in separate experiments). If subject *B*'s hand images are used for the training phase and the accuracy tests, different images of *B*'s are used in training and testing, respectively.

Some real and synthetic images are shown in Fig. 4. For each gesture class, there are 4536 synthetic training images. If the real images are also used for training, there are 4 real images per class, two from subject *A* and two from subject *B*. Hence, the size of the training set is 4540 images per class.

We have trained three different types of the AlexNet models. Model I begins with an empty AlexNet, and then it is trained with only the 4536×24 = 108,864 synthetic images. Model II uses Model I as a starting point, and then is incrementally trained with extra 4×24 = 96 real hand images. Model III begins with an empty AlexNet, like Model I, but then is trained by the mixture of 4540×24 = 108,960 synthetic and real images simultaneously.

The following subsections show the accuracy rates of the different models on the different test subjects. For each test, there are six real-camera test images per gesture class. The total number of real test images is 24×6 = 144. Note that none of these test images are used in the training process of the CNN models.

A. *Recognition rate when the test subject's hand data are also used for model training*

Fig. 5 shows the test results when the real camera images of the test subject *B*'s right hand are presented to the trained models. Note that both subject *A* and *B*'s hand images are used for the training of Model II and III. Nevertheless, the training images and the test images of subject *B*'s are different. Model I is only trained with the synthetic images so its recognition accuracy is quite low.

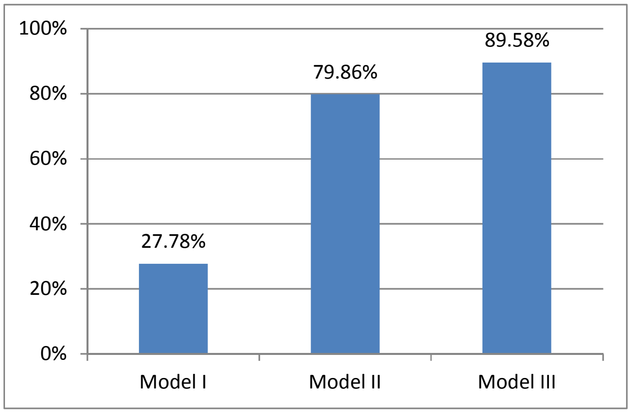


Fig. 5. The accuracy rates of the three differently trained models tested with the test subject *B*'s 144 hand images. Test subject *B*'s hand images are also used to train Model II and III.

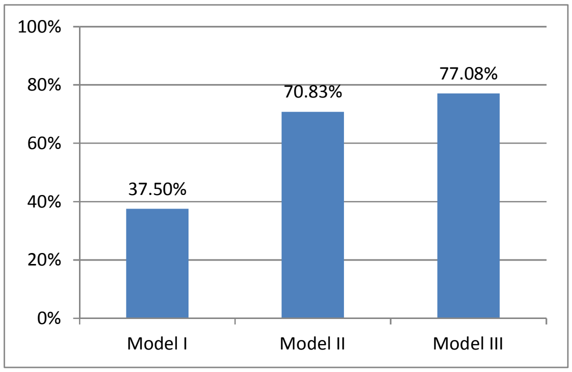


Fig. 6. The accuracy rates of the models tested using the test subject *C*'s 144 hand images. Test subject *C*'s hand images are not used to train the models.

B. Recognition rate when the test subject's hand data are not used for model training

Fig. 6 shows the test results when the real camera images of the test subject *C*'s right hand are presented to the trained models. Note that only subject *A* and *B*'s hand images are used for the training of Model II and III. Therefore, this experiment is a more realistic case since in practical applications, the target subjects' hand images are most likely not used for model training. Here, the recognition rate increases from 37.50% for Model I to 77.08% for Model III. Model I is only trained by the synthetic images while Model III is trained jointly by both the synthetic and the real images. Although there are only 0.09% of real images in the training data, these minor data must have some key features that cannot be captured from the majority of synthetic data. It is also interesting to see that Model II which are first pre-trained by the synthetic data, then trained incrementally by the 0.09% of real data only achieves an accuracy rate of 70.83%. This is a non-negligible drop from the accuracy achievable by Model III. This phenomenon may indicate that incremental training of a CNN model can get trapped in local minimum, depending on the implementation of the network solver.

C. Recognition rate when there are textures in the background of the test images

In practical applications, the background of a captured hand image can be removed by different techniques such as depths sensors, codebooks, or mixed Gaussian models. However, it is still interesting to see the performance of the CNN models when the inputs are real images with complex backgrounds. Note that these models are trained by images with no backgrounds. Fig. 7 shows some test images we used in this experiment and the recognition rate is shown in Fig. 8. With no surprise, the accuracy rates are quite low. However, the third background performs better than the first background in Fig. 7. The observation here is that the key features to the hand gesture recognition problem is the silhouette of the hands. A simple background that does not produce strong silhouette creates more problems to the recognition model. In addition to traditional computer vision techniques for background removal, it would be interesting to investigate whether it is possible to design a CNN model to do the job.

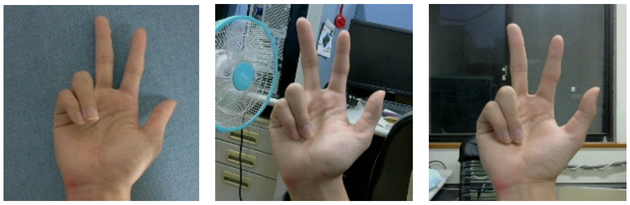


Fig. 7. Some test images with background textures. Again, there are totally 144 such images in this experiment.

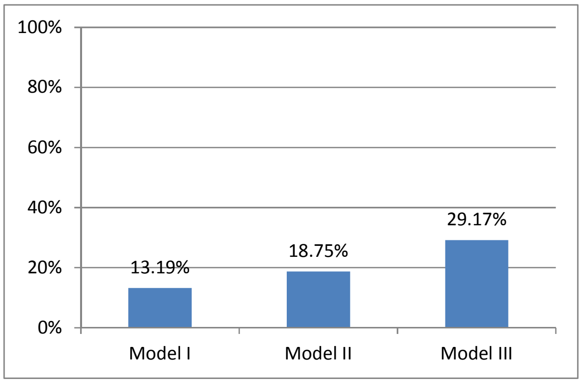


Fig. 8. The accuracy rate of the real images with backgrounds.

D. Some analyses on the experimental results

It is intriguing to see that by merely adding 0.09% of real images to the training data, the recognition rate can be raised from 37.50% to 77.08%, even if the test subject's images are not used in the training phases of the CNN model. In this section, we present some observations based on Model II, the model that is pre-trained with only the synthetic data, and then trained incrementally with the real data. The reason Model II is selected for analysis is that we can compare and contrast its weight parameters and trained kernels with those from Model I to see the effects of 0.09% of the real images to the synthetically pre-trained model.

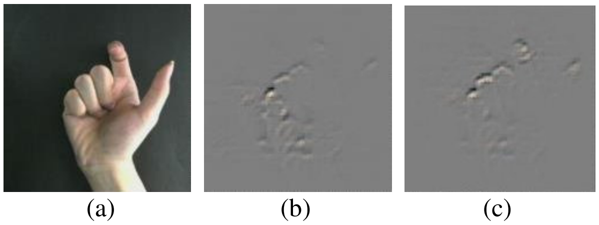


Fig. 9. (a) A sample gesture. (b) Its weights of the first FC layer of Model I. (c) Its weights of the same layer of Model II.

According to our experiments, the key differences are not in the five convolution layers of AlexNet. Using the DNN model visualization tool described in [13], one can see that the low-level and high-level features extracted by these convolution layers are basically the same with and without the real images. This observation can be verified by the fact that, even if the connection weights for the five convolutional layers are fixed after Model I, the performance of Model II are virtually the same. It is therefore speculated that the real images change how the high-level features are combined during the three fully-connected layers to trigger the outputs.

Fig. 9(a) shows a gesture that is not recognized by Model I but correctly recognized by Model II. Fig. 9(b) and 9(c) are the corresponding weights of the first fully connected layer in Model I and II, respectively. It can be seen that the real images raise the weights around the profiles of the fingers, especially the index finger, which causes the key difference between the two models.

VI. Conclusions

In this paper, we conduct some investigations on the usage of synthetic tagged data to train a deep CNN model for the 3D hand gesture recognition task. The experimental results show that the mixing of 99.91% synthetic data and 0.09% real data in the training data set produces a CNN model that can reach 77.08% recognition accuracy when the inputs are real hand images of a test subject not involved in the training phases. However, if only the synthetic data are used for training, the recognition rate will be as low as 37.5%. We also identify that the improvements come from the change in the weights of the fully-connected layers. That is, the real images training data affects how high-level features extracted from the convolutional layers are summarized in the fully-connected layers. We will perform further investigations on this phenomenon in our future work.

Acknowledgment

This work was funded by the Ministry of Science and Technology, Taiwan, under Grant # MOST 103-2221-E-009-206-MY3.

References

[1] Dipietro, L., A. M. Sabatini, and P. Dario, “A Survey of Glove-Based Systems and Their Applications,” *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38(4): 461-482, 2008.

[2] G. Marin, F. Dominio, and P. Zanuttigh, “Hand Gesture Recognition with Leap Motion and Kinect Devices,” *IEEE 23rd Int. Conf. on Image Processing (ICIP)*, pp.1565-1569, 2014.

[3] J. S. Supancic III, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, “Depth-Based Hand Pose Estimation: Data, Methods, and Challenges,” *IEEE Int. Conf. on Computer Vision (ICCV)*, 2015.

[4] J. J. Kuch and T. S. Huang, “Vision based Hand Modeling and Tracking for Virtual Teleconference and Telecollaboration,” *IEEE Int. Conf. on Computer Vision (ICCV)*, pp.666-671, 1995.

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *In Advances in Neural Info. Processing Systems*, 2012, pp. 1097-1105.

[6] S. R. Richter, V. Vineet, S. Roth, V. Koltun, “Playing for Data: Ground Truth from Computer Games,” *In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9906*.

[7] T. Heap and D. Hogg, “Towards 3D Hand Tracking using a Deformable Model,” *Proc. of the 2nd Int. Conf. on AFGR*, 1996.

[8] M. Kato, Y.-W. Chen, and G. Xu, “Articulated Hand Tracking by PCA-ICA Approach,” *Proc. of the 7th Int. Conf. on AFGR*, 2006.

[9] A. Gustus, G. Stillfried, J. Visser, H. Joerntell, and P. van der Smaqt, “Human Hand Modelling: Kinematics, Dynamics, Applications,” *Biological Cybernetics*, 106(11-12):741-55, Dec. 2012.

[10] J. Lee and T. Kunii, “Model-based Analysis of Hand Posture,” *IEEE Computer Graphics and Applications*, Vol 15, Issue 5, Sep. 1995, pp. 77-86.

[11] The Blender open source project, available on line at: https://www.blender.org/

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional Architecture for Fast Feature Embedding,” *Proc. of the 22nd ACM Int. Conf. on Multimedia*, Orlando, Florida, USA, 2014, pp. 675-678.

[13] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding Neural Networks through Deep Visualization,” *In Proc. of 32th ICML, Deep Learning Workshop*, Lille, France, July 6 – 11, 2015.

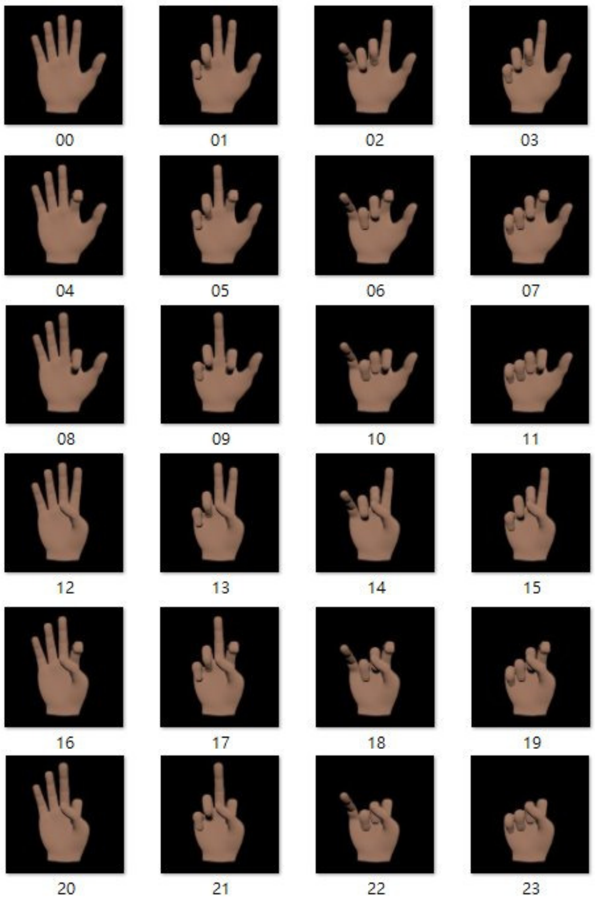


Fig. 10. The 24 gesture classes used in this paper. Only one snapshot per gesture class is shown here.