

# New blender-based augmentation method with quantitative evaluation of CNNs for hand gesture recognition

Huong-Giang Doan<sup>1</sup>, Ngoc-Trung Nguyen<sup>2</sup>

<sup>1</sup>Department of Control and Automation, Electric Power University, Hanoi, Vietnam

<sup>2</sup>Department of Research Management and International Cooperation, Electric Power University, Hanoi, Vietnam

---

## Article Info

### Article history:

Received Jun 15, 2022

Revised Dec 12, 2022

Accepted Dec 20, 2022

---

### Keywords:

Augmentation

Blender

Convolution neuron network

Generative adversarial network

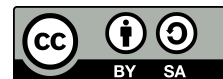
Hand gesture recognition

---

## ABSTRACT

In this study, we extensively analyze and evaluate the performance of recent deep neural networks (DNNs) for hand gesture recognition and static gestures in particular. To this end, we captured an unconstrained hand dataset with complex appearances, shapes, scales, backgrounds, and viewpoints. We then deployed some new trending convolution neuron networks (CNNs) for gesture classification. We arrived at three major conclusions: i) DenseNet121 architecture is the best recognition rate through almost evaluated red, green, blue (RGB) and augmentation datasets. Its performance is outstanding in most original works; ii) blender-based augmentation help to significantly increase 9% of accuracy, compared to the use of a RGB cues; iii) most CNNs can achieve impressive results at 97% accuracy when the training and testing datasets come from the same lab-based or constrained environment. Their performance is drastically reduced when dealing with gestures collected in unconstrained environments. In particular, we validated the best CNN on a new unconstrained dataset. We observed a significant reduction with an accuracy of only 74.55%. This performance can be improved up to 80.59% by strategies such as blender-based and/or GAN-based data augmentations to obtain an acceptable result of 83.17%. These findings contribute crucial factors and make fruitful recommendations for the development of a robust hand-based interface in practice.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



---

## Corresponding Author:

Ngoc-Trung Nguyen

Department of Research Management and International Cooperation, Electric Power University

235 Hoang Quoc Viet, Hanoi, Vietnam

Email: [trungnn@epu.edu.vn](mailto:trungnn@epu.edu.vn)

---

## 1. INTRODUCTION

Hand gestures have proffered a natural and efficient way for human machine interaction. Until now, hand gestures have been deployed in certain applications such as hand-3D-Object manipulation [1], hand-robot interactions [2], [3], hand-based surgery assistance [4], and hand-in-home appliance controlling [5], [6]. Such applications always require a high accurate recognition rate as well as a low computational cost. This problem has been actively studied for more than two decades and has achieved impressive results in lab-based environments. In real-world environments, the relevant techniques still face many challenges, such as complex background, view point variation, low hand resolution, and high hand degrees of freedom. Hand gesture recognition has been widely researched in literature work. Readers can refer to comprehensive surveys in this field [7]. Pisharady *et al.* [8], surveyed handcrafted feature recognition approaches. Recently, Al-Shamayleh *et al.* [9] focused on recent vision-based gesture systems and deep neural networks for static hand gesture

recognition. These surveys covered a large number of studies published in the research community. The deep learning-based approaches [10], [11] have been shown to outperform handcrafted feature-based approaches [5], [12]-[14] in most relevant tasks of hand such detection [15], pose estimation [5], [7], and gesture recognition [16], [17]. The convolution neuron network (CNN) architectures [18]-[20] require a very large dataset [21], [22] to train models while existing hand gesture datasets have not adapted for this demand. In addition, even through traditional augmentation method (such as flip, rotate, scale, strength images) or generative adversarial network (GAN) method [6] that could not increase enough large hand gesture dataset.

Recently, a comprehensive survey in [23] extensively lists gesture-based interfaces for a variety of applications. Many efficient CNN architectures have been proposed and trained on large datasets. For instance, CNNs are often trained using ImageNet with 1,000 classes and large-scale visual recognition competitions [24]. However, the ImageNet dataset does not contain objects from specific recognition tasks, such as hand or human body gestures. Therefore, the pre-trained CNN models have not been implemented on hand gesture datasets. In this study, pre-trained models of selected CNNs were fine-tuned using common public hand gesture datasets. We will investigate how to improve the performance of these networks for static hand-gesture recognition tasks. In this study, we consider the role of blender images on augmentation hand gesture dataset that is compared with traditional augmentation methods such as: rotate, flip, scale. Moreover, we quantitatively investigate new trending deep learning techniques, which are commonly utilized for object recognition tasks in general, but have never been studied deeply for static hand gesture recognition. To resolve this problem, firstly, a Hand-In-Wild dataset was collected. Secondly, we examine three trending neural networks, such as: MobileNet [25], DenseNet [26], and ResNet [19] for gesture classification. Based on the performance comparison results, we conclude that DenseNet achieves the best performance. Finally, we investigate the role of blender-based augmentation impacts to CNN performance.

In summary, the contributions of this study are four-fold, such as: i) we propose a new method to reach a new red, green, blue (RGB) hand gesture dataset in multiple viewpoints and multiple skill and shape patterns in virtual environment; ii) we examine and analyze transfer learning problem that impacts to the performance of a CNN model; iii) the CNNs model can achieve impressive results on the lab-based collected datasets but it could not efficient in unconstrained Hand-In-Wild dataset; iv) we recommend new techniques to improve the recognition performance on Hand-In-Wild datasets. Some limitations and suggestions for future works are also discussed.

The remainder of this paper is organized as follows. Section 2 firstly explains the proposed evaluation scheme and we then describe our new Hand-In-Wild dataset and Blender dataset. The experimental results and discussions are analyzed in section 3. Finally, section 4 concludes the paper and proposes research directions for future works.

## 2. PROPOSED METHOD

A new hand gesture framework is proposed to evaluate the performance of hand gesture recognition models as well as the efficiency of the blender-based augmentation data on recognizing hand gestures that is illustrated in Figure 1. This framework comprises three main blocks. The first block includes three RGB hand gesture datasets, such as: MICAHandPose [5], EPUHandPose1 [27], and new unconstrained Hand-In-Wild gesture dataset. The second block contains corresponding blender-based hand gesture datasets, such as: MICABlender, EPUBlender1, and EPUBlender3. The third block composes three deep neural networks: MobileNet, ResNet, and DenseNet. As denoted in section 1, these CNNs were selected because they recently achieved notable results on different computer vision tasks and presented their advantages on either GPU-based or embedded devices. We indicate deployment of deep neural networks for hand gesture recognition in section 2.1. In the next part, we present hand gesture datasets utilized to evaluate in this research. Finally, we implement protocols for the quantitative analyzes of the CNN models according to different factors; and robustness of the best model through evaluations in different conditions of the collected datasets (e.g., with complex background; in the unconstrained scene), which are reported in section 2.

### 2.1. Deployment of CNNs for hand gesture recognition

In this study, investigation of pre-trained CNN models for hand gesture recognition as MobileNet, ResNet, and DenseNet are reported. More particularly, specific versions of these neuronal networks, such as: ResNet50, MobileNetV2, and DenseNet121 are considered. While all these networks were trained on more than one million images from the ImageNet dataset [24] in order to classify images into 1,000 object

categories such as keyboard, mouse, pencil, many animals and so on. These models have not been trained for classifying hand gestures. Thanks to a large scale of ImageNet dataset, these networks have learned rich features which represent a wide range of images with image input size (224x224 pixels). In the following step, we re-train all above deep learning models following two strategies: i) retrain the last fully connected (FC) layer; and ii) retrain entire (All) layers of CNN model. Which are deployed by various learning rates. This work is not only performed on original RGB hand gesture datasets but also combined with augmentation hand gesture datasets (e.g., GAN-based synthetic images and/or blender-based images). We utilize each data stream to train aforementioned models. Because images are captured by various cameras at divergent resolutions, we firstly normalize images of RGB to the same size (224x224 pixels) for all models. More particularly, specific versions of the selected CNNs such as Resnet50, MobileNetV2 and DenseNet121 are considered. Then, we set the parameters for training the CNNs with input images (224x224 pixels); Batch size is 32; Number of epochs from 80 to 160; We applied two transfer learning strategies which are fine-tuning all layers or only the FC layer on pre-trained CNN models; three CNN networks are deployed, such as: Resnets50, MobileNetV2 and DenseNet121; loss function is applied by cross entropy; optimizer function is Adam; learning rate are from  $10^{-4}$  to  $10^{-6}$ . The process of fine-tuning the CNNs is evaluated and reported in detail in section 3.

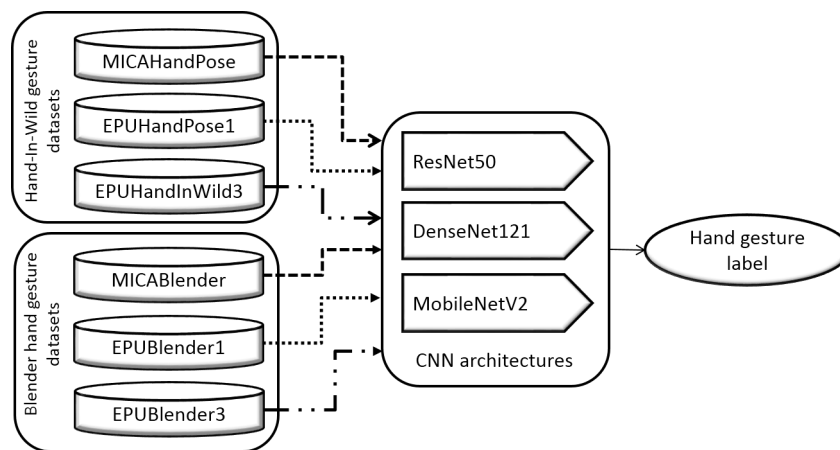


Figure 1. Our propose framework

## 2.2. Hand-In-Wild datasets and blender-based datasets

### 2.2.1. Framework for capturing datasets

In this paper, a framework is proposed to capture new hand gesture datasets as illustrated in Figure 2. Seven hand gestures ( $G_1$  to  $G_7$ ) are defined as illustrated in Figure 3 that utilizes for capture three dataset types, such as: RGB camera-based dataset; blender-based dataset; and Sensor-based dataset. These datasets are divided into two streams: Figure 4 shown original RGB hand gestures (hand images from RGB cameras) especially as shown Figure 4(a) and synthetic hand gestures (hand cues from blender program and hand glove) as shown in Figure 4(b). These datasets are presented in detail in the following section 2.2.2 and section 2.2.3.

### 2.2.2. Unconstrained hand gesture dataset

Almost published hand gesture datasets (e.g., MICAHandPose [5] and EPUHandPose1 [27]) which are formally implemented with some constraints, such as: a hand is raised in front of body and forwarded to camera, a hand is captured with simple background, camera is fixed, and so on. Consequently, hand shape images are simple and do not conclude entire hand poses of the real. In this research, we captured new datasets of seven hand gestures in Figure 3.

An unconstrained RGB Hand-In-Wild gesture dataset is captured as shown in a yellow stream of Figure 2. In this dataset, a participant implements twenty-five times for a gesture at various places and different complex scenes. Different RGB cameras are utilized to capture these datasets. These cameras are mobile and capture images with various resolutions. Figure 4(a) illustrated hand posture images  $G_1$  of Subject1. This dataset is named by EPUHandInWild3 and published in [28].

### 2.2.3. Blender-based hand gesture datasets

In this work, we collected a blender-based dataset of seven hand shapes ( $G_j; (j = (1, N), N = 7$  in Figure 3). It is assigned by the EPUBlender3 dataset. In addition, we also captured other blender-based datasets whose hand gestures are similar to gestures in MICAHandPose dataset and EPUHandPose1 dataset. These datasets are named by MICABlender dataset and EPUBlender1 dataset. As a results, there is a one-to-one correspondence between the gestures in the dataset pairs, such as: MICAHandPose and MICABlender; EPUHandPose1 and EPUBlender1; EPUHandInWild3 and EPUBlender3. The blender-based datasets are captured as illustrated in blue flows of Figure 2. Doan *et al.* [11], explained a hand glove was designed that composed five flex sensors and accelerator sensor to change the finger's curvatures and hand's movement. In this work, this electronic glove is continuously utilized to connect to a Blender program. Where finger's curvature cues of sensors could flexibly adjust to the change of the 3D hand patterns ( $PT_i, (i = (1, M), M = 5)$  in the Blender program. We created five hand patterns in the Blender environment as shown in the bottom of Figure 5. These 3D hand patterns are different in shapes, skin colors, and nail colors.

Thanks to the Blender application that generates various virtual cameras to capture object images in front of its viewpoint. These virtual cameras are easy to change its direction, distance to objects, its lens and so on. We then also capture 2D RGB images of a 3D hand model in various viewpoints, hand shapes, hand appearances and scales. Therefore, twenty-four virtual cameras are created around a 3D hand pattern in the Blender environment as illustrated in Figure 5. Cameras include different parameters with others, such as: positions, lens, scales and so on).

Given five Blender 3D hand models ( $PT_i, (i = (1, M), M = 5)$  in bottom of Figure 5, twenty-four virtual RGB cameras ( $C_k, (k = (1, K), K = 24)$  in the Blender environmental as show in Figure 5, and the electronics glove. It is note that only eight frontal viewpoint cameras are used ( $C_1, C_2, C_6, C_9, C_{10}, C_{11}, C_{12}, C_{17}$ ) for capturing MICABlender and EPUBlender1 datasets while EPUBlender3 dataset captured all twenty-four virtual cameras. Ten subjects are invited to wear the electronic glove and implement seven hand postures ( $G_j; (j = (1, N), N = 7)$ , each subject randomly chooses a 3D hand pattern ( $PT_i, (i = (1, M), M = 5)$  to show a hand posture in the Blender program, and data are saved under Blender subjects ( $BS_l, l = (1, L), L = 10)$  that is considered as virtual participants. At the same time, we capture both twenty-four 2D RGB hand images from twenty-four Blender cameras that are named by a EPUBlender3 dataset. In addition, information of five flex sensors and accelerator sensors are also stored and assigned by MICASensor, EPUSensor1, and EPUSensor3 datasets. Although these Sensor-based datasets are not utilized in this paper, it will be exploited in our next research. Figure 4(a) shows twenty-four Blender 2D hand images which are captured by hand gesture ( $G_1$ ) with hand pattern ( $PT_1$ ) and the first Blender subject ( $BS_1$ ). Figure 4(a) shows that multiple cameras of the Blender environment provide divergent high quality RGB hand poses in several viewpoints. Which hand gesture dataset could be enriched; efficient quantitative evaluation of this augmentation strategy is presented in detail in section 3.

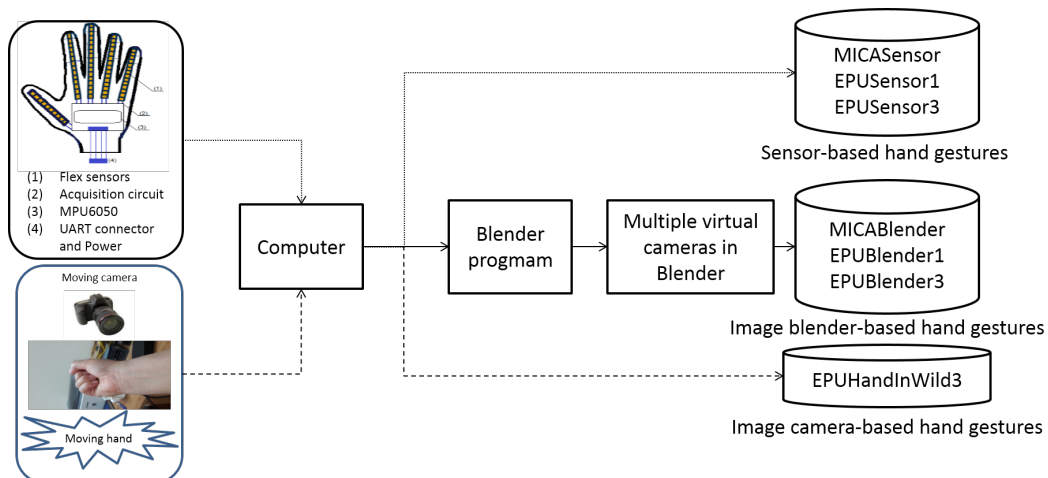


Figure 2. Framework to capture camera-based and blender-based datasets



Figure 3. Designed hand shapes

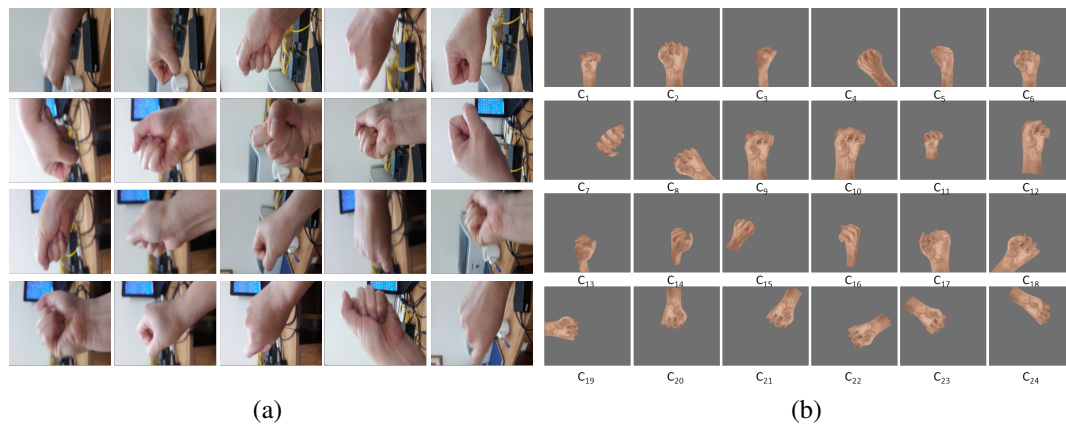


Figure 4. Hand gesture EPUBlender3 and EPUHandInWild3 (a) unconstrained hand gesture images from mobile cameras and (b) multiple views hand gesture images from Blender



Figure 5. Setup multiple virtual cameras in Blender environmental

### 2.3. Evaluation protocols and measurements

In this study, we apply single-dataset and combine-dataset evaluation protocols as illustrated in Figure 6. For the single-dataset protocol, we follow “One-leave-subject-out” for the subject independence test as the upper panel of Figure 6. It means in each dataset, only one subject is utilized for testing and remaining people in this dataset will be used for training models. Experiments are rolled for every subject in each dataset to ensure that every person could be tested. Hand gestures of a subject are used for testing that does not appear in the training phase. For the augmentation-based hand gesture recognition evaluation, it is the same with the single-dataset protocol in the testing phase but it differs above method that is performed to consider the affectation of the retrained CNN models with combination between the original RGB dataset and the augmentation dataset in training phase.

In this evaluation protocol, we used two augmentation datasets, such as: HRC\_GAN-based datasets [6] and blender-based datasets. The augmentation dataset evaluation scheme is illustrated in the lower panel

of Figure 6. As shown, one subject in the original RGB dataset is used for testing, remaining subjects in the original RGB dataset compose with the blender-based dataset and/or the HRC\_GAN-based dataset to pre-train a CNN model. This scheme examines that: How is the role of blender-based images on Hand-In-Wild hand gesture recognition? Is it efficient to retrain various CNN models with hand gesture augmentation data? How is the difference between various hand gesture augmentation data?

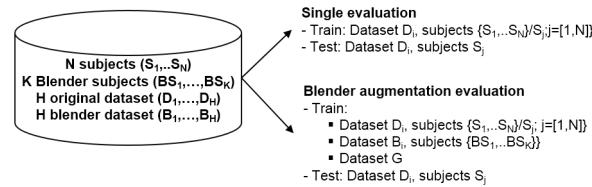


Figure 6. Evaluation protocol

### 3. EXPERIMENTAL RESULTS

The experiments are conducted to indicate the following problem: i) which is the most suitable CNN architecture for hand posture recognition. We also analyze the effectiveness of the transfer learning model and other factors that can impact the performance; ii) how good is a combination of augmentation (blender-based and/or HRC\_GAN-based) versus original RGB dataset?; iii) how much is the performance degradation rate of the best model when deployed in unconstrained environments?. The evaluation schemes are written in Python on a Pytorch deep learning framework and run on a workstation with NVIDIA GPU 11G.

#### 3.1. Transfer learning strategies of CNN architectures on hand posture datasets

In this evaluation, three single hand gesture datasets (MICAHandPose [5], EPUHandPose1 [27], and EPUHandInWild3) are utilized. “Leave-one-subject-out” protocol is applied to divide data as presented in detail in the previous section section 2.3. Figure 7 shows experimental results for hand gesture recognition on three deep convolutional neural networks as MobileNet, ResNet and Densenet, respectively. In this evaluation, we implemented two retrained strategies for CNN models which concludes retraining of all layers (All) and last layer (FC) of CNNs. Learning rate is applied at  $10^{-6}$  for this evaluation. This figure shows that: i) ResNet50 and DenseNet121 have the best and the same accuracy which are far larger than MobileNetV2 on all original hand gesture datasets; ii) ning the CNN models on all layers obtain higher accuracy than that deploying on the last layer (e.g., FC6 layer), as shown in Figure 7 for all three hand gesture datasets. The effectiveness of fine-tuning is clearly confirmed. For instance, the recognition rate was 81.29% without fine-tuning and increased up to 97.27% with fine-tuning on the MICAHandPose dataset for the DenseNet121 model. Therefore, the recognition rate of a CNN can be improved significantly by deploying fine-tuning on a small hand dataset.

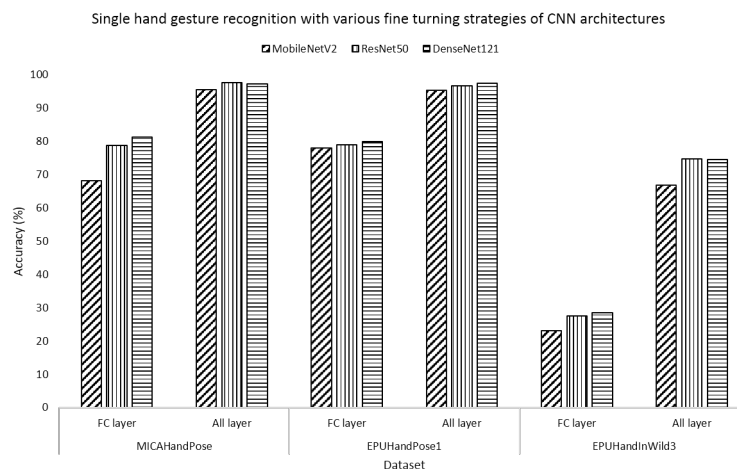


Figure 7. Hand gesture recognition with different model retrain strategies

### 3.2. Single hand gesture recognition of various CNN architectures

In this evaluation, we examine the effect of learning rate ( $lr$ ) values on retraining of the deep learning models when the transfer learning step is performed on a hand gesture database. Three learning rate values are deployed at  $10^{-6}$ ,  $10^{-5}$ , and  $10^{-4}$ . Figure 8 illustrates converging process of three CNN models on various learning rate values of EPUHandInWild3 dataset. This figure shows that transfer learning of the model at  $lr = 10^{-4}$  (accuracy values of models in Figure 8(a) and Loss values of models in Figure 8(b)) and  $lr = 10^{-5}$  (accuracy values of models in Figure 8(c) and loss values of models in Figure 8(d)) are achieved after 7 epochs and 10 epochs, respectively. Reduction speeds of in both cases are far faster than of  $lr = 10^{-6}$  (accuracy values of models in Figure 8(e) and Loss values of models in Figure 8(f)) that is stable after 80 epochs. Furthermore, the validation graph in Figure 8(a) and Figure 8(b) are not only fast movements but also not stable. Compared with Figure 8(e) and Figure 8(f), both accuracy and loss lines converge with a moderate speed and are highly stable. These results show that learning rate value at ( $lr = 10^{-6}$ ) is reliable and the CNN models could learn more features of hand gestures.

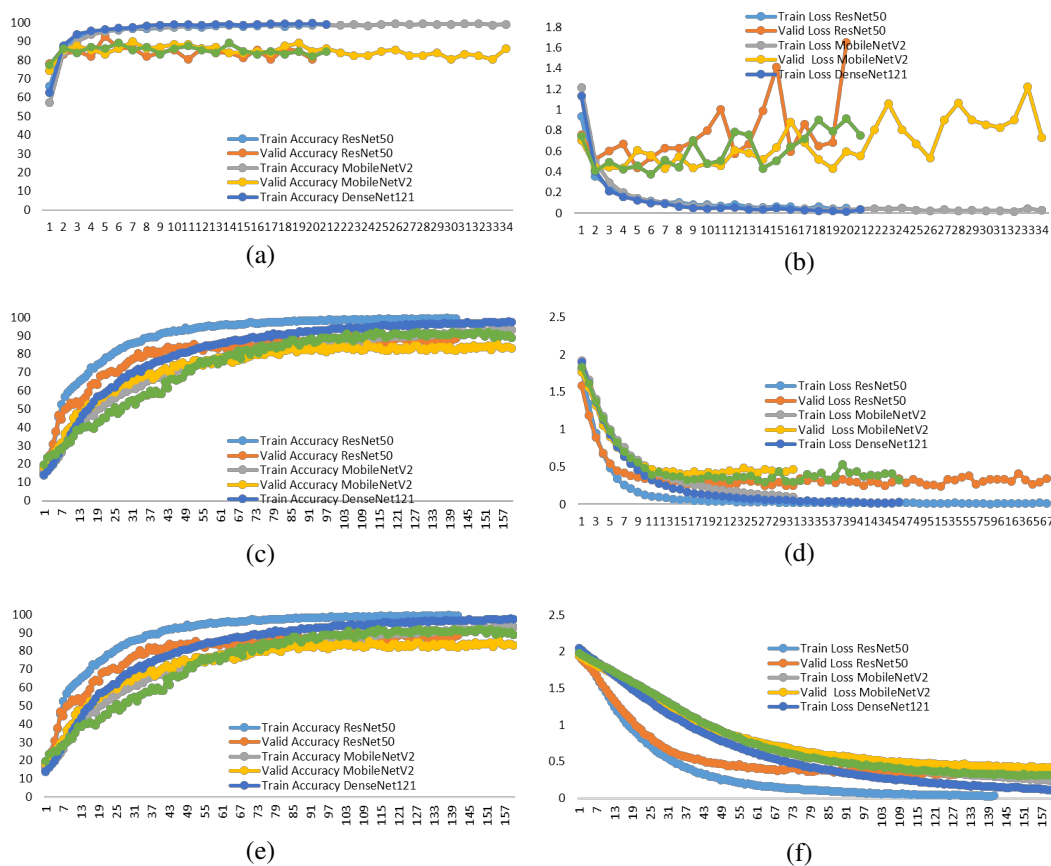


Figure 8. Accuracy and Loss values with different learning rates (a) accuracy value of various CNN models at  $lr=10^{-4}$ , (b) loss value of various CNN models at  $lr=10^{-4}$ , (c) accuracy value of various CNN models at  $lr=10^{-5}$ , (d) loss value of various CNN models at  $lr=10^{-5}$ , (e) accuracy value of various CNN models at  $lr=10^{-6}$  and (f) Loss value of various CNN models at  $lr=10^{-6}$

Figure 9 illustrates hand gesture recognition accuracy of the CNN models on three camera-based datasets and corresponding blender-based datasets at divergent learning rate values as:  $10^{-4}$ ,  $10^{-5}$  and  $10^{-6}$ . This result indicates that:

- These evaluations one against indicates that CNN models are better at  $lr = 10^{-6}$ .
- Existing a big gap hand gesture recognition results between a Hand-In-Lab dataset (MICAHandPose and EPUHandPose) and a Hand-In-Real dataset (EPUHandInWild3).

- Blender data promises a good hand gesture recognition accuracy with the best result belonging to the DenseNet121 architecture. With two constrained hand gesture datasets, average results of MICABlender dataset and EPUBlender1 dataset achieve upto 100%. Moreover, efficient blender-based hand gesture datasets with data augmentation role will be investigated in the next section.

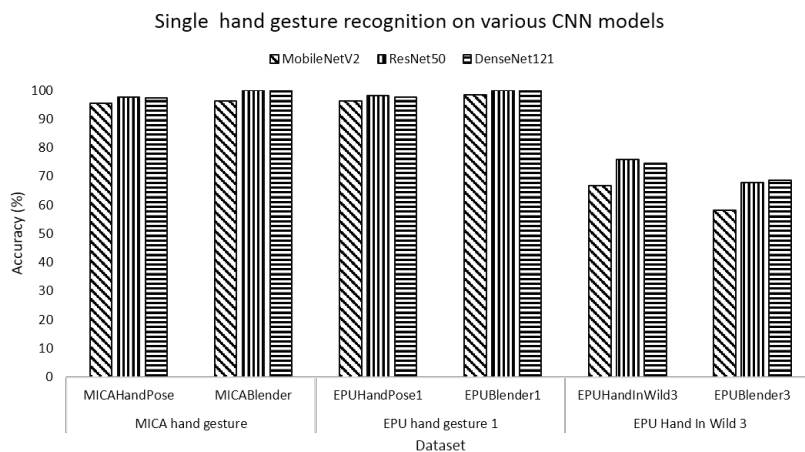


Figure 9. Single hand recognition accuracy of the CNN architectures at  $Lr = 10^{-6}$

### 3.3. Efficient of Hand-In-Wild gesture recognition with augmentation methods

In this section, we combine the camera-based datasets with the HRC\_GAN-based datasets, and/or the blender-based datasets in three different strategies in order to train the CNN models. This training and testing protocol was presented in section 2.3 (the bottom part of Figure 6). Firstly, combinations between Camera-based datasets and blender-based datasets are deployed on various learning rates of three CNN models as illustrated in 3rd columns of Figure 10. This result will be compared with the single dataset evaluation in section 3.2 (1st and 2nd columns in Figure 10). This result indicates that:

- Learning rate also brings the best recognition results on entire CNN architectures (MobileNetV2, ResNet50, and DenseNet121) at  $Lr = 10^{-6}$  on entire Original RGB images, Blender synthetic images, and both of them.
- For blender-based augmentation, DenseNet121 architecture obtains the highest accuracy at 80.58% while MobileNetV2 accounts the smallest values at 70.96% and ResNet50 network achieves slightly smaller results than the DenseNet121 method.
- Hand gesture recognition obtains the best accuracy with blender-based augmentation on MobileNetV2, ResNet50 and DenseNet121 at 70.96%, 78.62%, and 80.58%, respectively (3rd columns of Figure 10). Which are extremely better than single hand gesture recognition results of original RGB dataset at 64.91%, 70.19%, and 73.39%, respectively (1st columns of Figure 10). Combination between Camera-based dataset with blender-based dataset achieves far higher accuracy than single hand gesture dataset.
- The best recognition accuracy belongs to the DenseNet128 model with blender-based augmentation data at 80.58% that is dramatically improved 6.03% than without augmentation data at 74.55%. As a result, DenseNet121 is utilized for the next examination because of its outstanding performance on the above datasets.

Secondly, we also compare the efficiency of other augmentation strategies (GAN-based and blender-based) on the DenseNet121 network and using various Hand-In-Wild gesture datasets. A glance at the Table 1, it indicates that:

- In second column, while result of HRC\_GAN-based method is better on EPUHandInWild3 dataset at 76.89% (2,34% higher than the original data) and far smaller on MICAHandPose dataset and EPUHandPose1 dataset at 89.16% and 91.89%, respectively. It is because the HRC\_GAN method provided chaotic hand images in other viewpoints of hand gesture while the original MICAHandPose dataset and EPUHandPose1 dataset contain one viewpoint hand image. This augmentation data could be an



unnecessary disturbance of the training data. It inverses with the original EPUHandInWild3 dataset that consists of hand gestures in any viewpoints. Therefore, it is suitable for unconstrained and complex Hand-In-Wild dataset.

- In third column, hand gesture recognition results with blender-based augmentation methods (at 100%, 100%, and 80.59%, respectively) are higher than with Original Hand-In-Wild gestures (at 97.27%, 97.56%, and 74.55%, respectively) on entire three datasets (MICAHandPose, EPUHandPose1, and EPUHandInWild3, respectively).
- Last column indicates hand gesture accuracies of three Hand-In-Wild datasets with both HRC\_GAN data and Blender data. These results showed that augmentation of these flows brings the best hand gesture accuracies on EPUHandInWild dataset at 83.17%. But these results are still smaller accuracy of MICAHandPose dataset and EPUHandPose1 dataset at 91.25% and 93.63%, respectively. Combination data of HRC\_GAN data and Blender data promise to achieve the large augmentation dataset for improving the efficiency of the CNN hand gesture recognition which the data requirements are diverse in terms of viewpoints and various in appearances, difference in scales and so on.

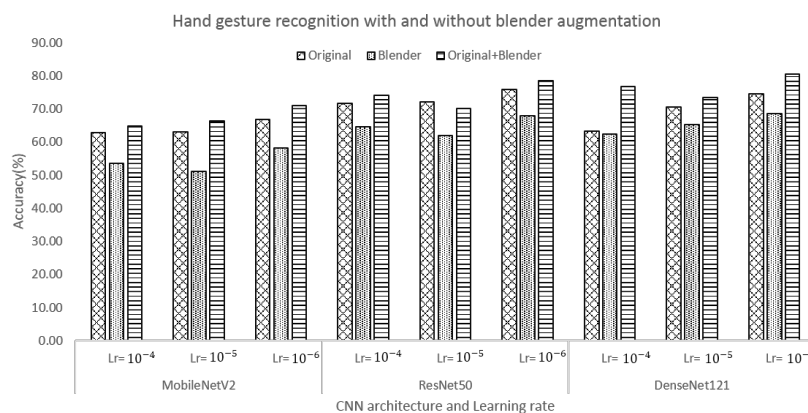


Figure 10. Hand gesture recognition results on single camera-based and blender-based augmentation

Table 1. Impact of hand augmentation on recognition accuracy (%) of DenseNet-121 model

	Original	Original+HRC_GAN	Original + Blender	Original + HRC_GAN + Blender
MICAHandPose	97.27%	89.16%	<b>100%</b>	91.25%
EPUHandPose1	97.56%	91.89%	<b>100%</b>	93.63%
EPUHandInWild3	74.55%	76.89%	80.59%	<b>83.17%</b>

Through this experiment, we confirmed the advantages of using augmentation data (blender-based images and HRC\_GAN-based images) to boost the performance of CNN models. These improvements provided valuable values. In particular, the blender-based method is easily setup and can capture multi-viewpoint, scale, shape, skin RGB hand data. One can clearly take advantage of blender images to deploy gesture-based applications. As shown above, accuracy of the CNN models can be higher than 83% when combining unconstrained hand gestures with Blender synthetic and GAN generator images, which is acceptable in some contexts. Moreover, this result is far from deploying a real application. In such a situation, the idea is to utilize online learning that helps to update the model with new hand gestures collected from a natural scene or increase the virtual camera in a Blender environment with complex objects in scenes. This scheme can achieve better performances than deploying a common augmentation strategy.

#### 4. CONCLUSIONS AND DISCUSSION

This paper presents a comparative analysis of some recent deep neural networks for static hand gesture recognition on various Hand-In-Wild datasets. Among the evaluated models (ResNet-50, MobileNet-V2, DenseNet-121), DenseNet121 has the best recognition accuracy. Its performance is superior to that of the original works by at least 9%. We also investigated other factors that can affect the performance of DenseNet121,

such as transfer learning, learning rate, augmentation data methods. We found that there exists a notable gap in recognition rates with CNN model deployment in unconstrained environments. In constrained dataset cases, GAN-based augmentation method is useless even with reduced recognition accuracies. Evaluation results on the unconstrained Hand-In-Wild dataset shows great interest in combining original RGB dataset and HRC.GAN-based augmentation to increase accuracy by 2%. In most cases, using blender-based augmentation data can achieve an accuracy rate above 9%. This performance is remarkable and promises a feasible solution for deploying gesture-based applications in practice. This remark opens up new research directions that require further investigation on data augmentation, multi-view gesture analysis, and online learning. Once these bottlenecks are resolved, the development of a gesture-based interface in practical applications is straightforward.




## REFERENCES

- [1] P. Bourdot, T. Convard, F. Picon, M. Ammi, D. Touraine, and J.-M. Vézien, "VR-CAD integration: multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models," *Computer-Aided Design*, vol. 42, no. 5, pp. 445–461, 2010, doi: 10.1016/j.cad.2008.10.014.
- [2] G. Gonzalez and J. Wachs, "Pose Imitation constraints for collaborative robots," 2020, [Online]. Available: <http://arxiv.org/abs/2009.10947>.
- [3] V.-T. Nguyen, T.-H. Tran, T.-L. Le, R. Mullot, and V. Courboulay, "Using hand postures for interacting with assistant robot in library," *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*, 2015, pp. 354–359, doi: 10.1109/KSE.2015.18.
- [4] N. Madapana, G. Gonzalez, and J. Wachs, "Gesture agreement assessment using description vectors," *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 2020, pp. 40–44, doi: 10.1109/FG47880.2020.00043.
- [5] H.-G. Doan, V.-T. Nguyen, H. Vu, and T.-H. Tran, "A combination of user-guide scheme and kernel descriptor on RGB-D data for robust and realtime hand posture recognition," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 103–113, 2016, doi: 10.1016/j.engappai.2015.11.010.
- [6] H.-G. Doan, "Multiple views and categories condition GAN for high resolution image," in *Artificial Intelligence in Data and Big Data Processing. ICABDE 2021. Lecture Notes on Data Engineering and Communications Technologies*, Cham: Springer, 2022, pp. 507–520, doi: 10.1007/978-3-030-97610-1\_40.
- [7] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015, doi: 10.1007/s10462-012-9356-9.
- [8] P. K. Pisharady, P. Vadakkepat, and A. P. Loh, "Attention based detection and recognition of hand postures against complex backgrounds," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 403–419, Feb. 2013, doi: 10.1007/s11263-012-0560-5.
- [9] A. S. Al-Shamayleh, R. Ahmad, M. A. M. Abushariah, K. A. Alam, and N. Jomhari, "A systematic literature review on vision based gesture recognition techniques," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28121–28184, 2018, doi: 10.1007/s11042-018-5971-z.
- [10] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 1–7, doi: 10.1109/CVPRW.2015.7301342.
- [11] H.-G. Doan and N.-T. Nguyen, "End-to-end multiple modals deep learning system for hand posture recognition," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 27, no. 1, pp. 214–221, 2022, doi: 10.11591/ijeecs.v27.i1.pp214-221.
- [12] H. Tang, H. Liu, W. Xiao, and N. Sebe, "Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion," *Neurocomputing*, vol. 331, pp. 424–433, 2019, doi: 10.1016/j.neucom.2018.11.038.
- [13] P. Sykora, P. Kamencay, and R. Hudec, "Comparison of SIFT and SURF methods for use on hand gesture recognition based on Depth Map," *AASRI Procedia*, vol. 9, pp. 19–24, 2014, doi: 10.1016/j.aasri.2014.09.005.
- [14] A. Dixit and T. Kasbe, "Multi-feature based automatic facial expression recognition using deep convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 25, no. 3, pp. 1406–1419, 2022, doi: 10.11591/ijeecs.v25.i3.pp1406-1419.
- [15] A. Mujahid *et al.*, "Real-time hand gesture recognition based on deep learning YOLOv3 model," *Applied Sciences*, vol. 11, no. 9, p. 4164, 2021, doi: 10.3390/app11094164.
- [16] A. G. Mahmoud, A. M. Hasan, and N. M. Hassan, "Convolutional neural networks framework for human hand gesture recognition," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2223–2230, 2021, doi: 10.11591/eei.v10i4.2926.
- [17] A. E. Minarno, W. A. Kusuma, and Y. A. Kurniawan, "Human activity recognition for static and dynamic activity using convolutional neural network," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 6, pp. 1857–1864, 2021, doi: 10.12928/telkomnika.v19i6.20994.
- [18] A. G. Howard *et al.*, "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, [Online]. Available: <http://arxiv.org/abs/1704.04861>.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017, doi: 10.1609/aaai.v31i1.11231.
- [21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
- [22] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732, doi: 10.1109/CVPR.2014.223.




- [23] T. Vuletic, A. Duffy, L. Hay, C. McTeague, G. Campbell, and M. Greal, "Systematic literature review of hand gestures used in human computer interaction interfaces," *International Journal of Human-Computer Studies*, vol. 129, pp. 74–94, 2019, doi: 10.1016/j.ijhcs.2019.03.011.
- [24] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.
- [25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear Bottlenecks," 2018, [Online]. Available: <http://arxiv.org/abs/1801.04381>.
- [26] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
- [27] D. H. Giang, "Improving hand posture recognition performance using multi-modalities," *EPU Journal of Science and Technology for Energy*, vol. 26, no. 26, pp. 40–49, 2021.
- [28] "Unconstrained Hand-in-Wild gesture dataset." [Online]. Available: <https://zenodo.org/record/6616066>.

## BIOGRAPHIES OF AUTHORS



**Huong-Giang Doan**    received B.E. degree in Instrumentation and Industrial Informatics in 2003, M.E. in Instrumentation and Automatic Control System in 2006 and Ph.D. in Control Engineering and Automation in 2017, all from Hanoi University of Science and Technology, Ha Noi, Vietnam. She can be contacted at email: [giangdth@epu.edu.vn](mailto:giangdth@epu.edu.vn).



**Ngoc-Trung Nguyen**    received B.E degree in Power System in 2003, M.E in Electrical Engineering in 2006, all from Hanoi University of Science and Technology, Hanoi, Vietnam; received Ph.D in Electrical Engineering from University of Palermo, Palermo, Italy, in 2014. He can be contacted at email: [trungnn@epu.edu.vn](mailto:trungnn@epu.edu.vn).