IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 3, September 2024, pp. 3240~3249
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i3.pp3240-3249

Quantitative strategies of different loss functions aggregation for knowledge distillation

Huong-Giang Doan[1], Ngoc-Trung Nguyen[2]
[1]Faculty of Control and Automation, Electric Power University, Hanoi, Vietnam
[2]Department of Personnel and Organization, Electric Power University, Hanoi, Vietnam

Article Info

Article history:

Received Dec 23, 2023
Revised Feb 19, 2024
Accepted Feb 28, 2024

Keywords:

Convolution neuron network
Deep learning
Knowledge distillation
Student-teacher model
Transfer learning

ABSTRACT

Deep learning models have been successfully applied to many visual tasks. However, they tend to be increasingly cumbersome due to their high computational complexity and large storage requirements. How to compress convolutional neural network (CNN) models while still maintain their efficiency has received increasing attention from the community, and knowledge distillation (KD) is efficient way to do this. Existing KD methods have focused on the selection of good teachers from multiple teachers, or KD layers, which is cumbersome, expensive computationally, and requires large neural networks for individual models. Most of teacher and student modules are CNN-based networks. In addition, recent proposed KD methods have utilized cross entropy (CE) loss function at student network and KD network. This research focuses on the quantifiable evaluation of teacher-student model, in which knowledge is not only distilled from training models that have the same CNN architecture but also from different architectures. Furthermore, we propose combination of CE, balance cross entropy (BCE), and focal loss functions to not only soften the value of loss function in transferring knowledge from large teacher model to small student model but also increase classification performance. The proposed solution is evaluated on four benchmark static image datasets, and the experimental results show that our proposed solution outperforms the state-of-the-art (SOTA) methods from 2.67% to 9.84% at top 1 accuracy.

Corresponding Author:

Ngoc-Trung Nguyen
Department of Personnel and Organization, Electric Power University
No. 235 Hoang Quoc Viet Street, Co Nhue 1 Ward, Bac Tu Liem District, Hanoi City, Vietnam
Email: trungnn@epu.edu.vn

1. INTRODUCTION

Convolution neuron network has become a successful solution in many fields, such as natural language processing (NLP) [1], [2], and computer vision (CV) [3]–[5]. In many reviews, convolutional neural network (CNN) architectures are increasingly complex, but in return, their efficiency is increasing [6], [7]. The complex architecture of CNN models with a large number of parameters will lead to high computational cost and large memory storage. This limits the ability to deploy the CNN models on limited-resource devices. In order to boost the performance of a simple and small-size CNN model that can be deployed for robot applications and/or mobile devices, the knowledge distillation (KD) technique has been considered to transfer the trained parameters from a complicated model to a light model. This KD process is called the student-teacher framework, where the large model plays the role of teacher and the light model is the student [8]–[11].

The first KD method was proposed by [12], and expanded by [8]. The KD method is used for model compression and knowledge transfer. It could be deployed as an online or offline distillation model, with a single teacher or multiple teachers, supervised or unsupervised learning. In many reviews [8], [9], [13], KD methods are divided into two main categories: i) feature distillation [10], [11] which compresses knowledge at feature levels; ii) model distillation [14], [15] with the parameters transferred between two models of teacher and student through loss functions. For the feature KD method, it boosts the performance of a small student model with the supervision of the output feature maps from a complex teacher model. Some recent researches introduced a single teacher or multiple teachers to provide more supervision to a student network. Guan *et al.* [10] tackled both the efficiency and effectiveness of KD with feature aggregation to imitate the multiple teachers in the single teacher framework. According to Guan *et al.* [10], the differentiable feature aggregation (DFA) method is used to extract informative supervision from feature maps of multiple teachers by a two-stage DFA search. Heo *et al.* [16] investigated the design aspects of feature distillation methods for achieving network compression. In this work, the distillation loss is designed to create a synergy among various aspects: teacher and student transformations, distillation feature position, and distance function with distillation loss. It composes a feature transform with a designed margin rectified linear unit, a distillation feature position, and a partial Euclid distance function to skip redundant information, giving adverse effects to the compression of the student. Liu *et al.* [17] merges the transfer learning task and the model compression task into one stage that distills and transfers knowledge at the feature map level, releasing inconsistency between teacher and student. However, the multiple-teacher distillation method requires costly computation. In addition, comparisons between distributions of features are large, complex works, and there is no generalization for layer feature distillation.

For the loss function distillation, Yao *et al.* [14] proposed a graph few shot learning algorithm that incorporates prior knowledge learned from auxiliary graphs. A transferable metric space characterized by a node embedding and a graph-specific prototype embedding function is shared between auxiliary graphs and the target, facilitating the transfer of structural knowledge. This paper used reconstruction loss for training auto-encoders. According to Zhang *et al.* [15], an object relational graph-based encoder, that captures more detailed interaction features to enrich visual representation. In this work, a teacher recommended a learning method to make full use of the successful external language model to integrate the abundant linguistic knowledge into the caption model. The training criterion is based on the cross-entropy loss that temperature used to smooth output distribution. Siam *et al.* [18] proposed a teacher-student learning paradigm to teach robots about their surrounding environment. Two stream motions and appearances teacher network provide pseudo-labels to adapt to an appearance student network. The student network is able to segment the newly learned objects into other scenes, whether they are static or dynamic actions. The model distillation approach is highly generalized and simpler than data distribution comparison, thus reducing time costs. According to Siam *et al.* [18], the cross-entropy loss is also viewed as a mean value to decrease the divergence between the true distribution and the predicted one. However, most of the studies have been focused on analyzing the influence of the teacher model on a student, the role of many teachers versus fewer teachers, or studies using only cross entropy (CE) loss functions. In addition, research concentrated on the same type of CNN network for teacher and student, where teacher is the higher version and student is the smaller version (e.g., Resnet101 vs Resnet18, or DenseNet121 vs Densenet169.). In summary, the contributions of this research are two-fold: i) a new method for transferring knowledge between the teacher model and the student model is proposed, in which we investigate the influence of CE, balance cross entropy (BCE) and focal loss (FC) functions on transfer learning between a teacher model and a student model; ii) we quantitatively evaluate the effect of the proposed solution on KD between the same CNN architecture styles and the different CNN architecture styles. Some limitations and suggestions for future work are also discussed.

The remainder of this paper is organized as follows: section 2 firstly explains the proposed evaluation scheme. The experimental results and discussions are analyzed in section 3. Finally, section 4 concludes the proposed research directions for future works.

2. METHOD

Our proposed framework is illustrated in Figure 1, which comprises three main parts: The first part includes a teacher model as shown in Figure 1(a). It is a complex network with a large number of trained parameters. The second part is a student model whose network is more simple than the teacher network as shown in Figure 1(b). The third part comprises KD is shown in Figures 1(c) and 1(d). Given a target dataset, the progress of KD is firstly retrained and/or fine-tuned for the teacher network to obtain a teacher model. This complex model is then utilized to transfer to the student network where the teacher network is blocked on all layers and the student network is transferred with the adjusted parameters through the KD loss function (the third part concludes Figures 1(c) and 1(d)). In this paper, we propose a new loss function combination of the KD framework that is presented in detail in the next sections.

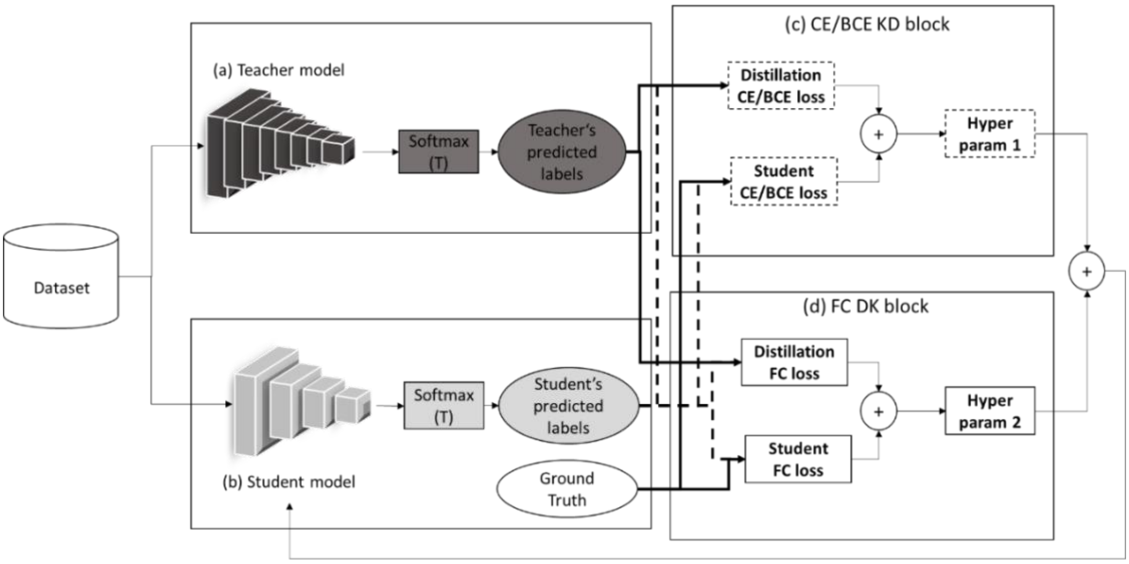


Figure 1. Our propose KD framework, (a) pretrained complex of teacher model, (b) simple of student model, (c) CE/BCE KD block, and (d) FC KD block

2.1. Cross entropy, balance cross entropy, and focal loss functions

For a machine learning or deep learning model, a loss function is used to optimize and adjust parameters when the model is trained. The objective function is the best way to minimize the value of the loss function. The lower the loss, the better the model. In this section, we briefly survey some common loss functions such as CE, BCE, and FC.

The CE method [19] was proposed as an adaptive importance sampling procedure for the estimation of rare-event probabilities that uses the CE function [19] A or Kullback-Leibler divergence [20] as a measure of closeness between two sampling distributions. The CE loss function is an important loss function that is widely used with a skewed dataset. It is pegged to an understanding of the softmax activation function. $\hat{y}$ is the predicted probability distribution, and y_i is the ground truth probability distribution. The cross-entropy loss function is shown in (1):

$$L_{CE}(y,\hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y_i}) \tag{1}$$

Moreover, previous research indicated that the CE function is not good for imbalanced data. Thus, the CE function [21] is improved by weighted CE. It is widely used for skewed datasets in which both positive and negative examples are weighted by α as presented in (2):

$$L_{BCE}(y,\hat{y}) = -\alpha \sum_{i=1}^{C} y_i \log \hat{y_i} \tag{2}$$

Where $\alpha = \frac{1}{\epsilon + f_i}$, ϵ is constant which helps to avoid zero denominator. However, the BCE method is not efficient for serious imbalances of data between categories in the training dataset. But it does not really change the gradient descent of the loss function yet. While the model is trained on the severely imbalanced sample, the gradient descent value is largely influenced by the dominant class. It is necessary to adjust more radically to increase the influence of minorities on gradient descent. The FC function [22] is proposed as illustrated in (3) which continues to inherit BCE and adjusts for gradient descent.

$$L_{FC}(y,\hat{y}) = -\alpha \sum_{i=1}^{C} y_i (1-\hat{y_i}) \log\hat{y_i} \tag{3}$$

The FC loss obtains the best effects on a highly-imbalanced dataset, down-weighing the contribution of easy samples and enabling the model to learn hard samples. It is clear that the FC function of $(1-\hat{y_i})$ is added in the BCE formula. However, this element is effective in adjusting output labels on the loss function and gradient descent. In this work, we propose to combine the loss functions of CE and FC functions, or BCE function and FC function, in the KD framework. They will be presented in the next section.

2.2. Loss function combination strategy for the knowledge distillation method

Given Ii image, its ground truth label is y_i. The Ii image is passed through the teacher model and the student model. The predicted result of the teacher model is $y_{pred}^{teacher} = \widehat{y^t}$. The predicted result of the student model is $y_{pred}^{student} = \widehat{y^s}$. KD was first introduced [8], it applied the original CE function as shown in (4):

$$L(\mathrm{y},\widehat{y^s},\widehat{y^t}) = L_{CE}^{S}(\mathrm{y},\widehat{y^s}) + \lambda L_{CE}^{KD}\left(\widehat{y^t},\widehat{y^s}\right) = -\sum_{i=1}^{C} y_i \log\widehat{y_i^s} - \lambda\sum_{i=1}^{C}\widehat{y_i^t}\log\widehat{y_i^s} \tag{4}$$

Moreover, KD utilizes CE as (4) is a hard logit probability distribution. Thus, it is then added with temperature scale normalization for a soft logit probability distribution as illustrated in (5) and (6). This transformation slightly reduces the output probability, in which the large value is not too close to 1 and the small value is not approaching 0. This function corresponds to Figure 1(c):

$$L^{CE}(\mathrm{y},\widehat{y^s},\widehat{y^t}) = (1-\lambda)L_{CE}^{S}(\mathrm{y},\widehat{y^s}) + \lambda L_{CE}^{KD}\left(\widehat{y^t},\widehat{y^s}\right) \tag{5}$$

$$L^{CE}(\mathrm{y},\widehat{y^s},\widehat{y^t}) = -(1-\lambda)\sum_{i=1}^{C} y_i\log\widehat{y_i^s} - \lambda T^2\sum_{i=1}^{C}\frac{\widehat{y_i^t}}{T}\log\frac{\widehat{y_i^s}}{T} \tag{6}$$

KD uses BCE loss with temperature scale normalization as shown in (7) and (8). This function corresponds to Figure 1(c):

$$L^{BCE}(\mathrm{y},\widehat{y^s},\widehat{y^t}) = (1-\lambda)L_{BCE}^{S}(\mathrm{y},\widehat{y^s}) + \lambda L_{BCE}^{KD}\left(\widehat{y^t},\widehat{y^s}\right) \tag{7}$$

$$L^{BCE}(\mathrm{y},\widehat{y^s},\widehat{y^t}) = -(1-\lambda)\alpha\sum_{i=1}^{C} y_i\log\widehat{y_i^s} - \lambda T^2\alpha\sum_{i=1}^{C}\frac{\widehat{y_i^t}}{T}\log\frac{\widehat{y_i^s}}{T} \tag{8}$$

The KD framework utilizes FC with temperature scale normalization as illustrated in (9) and (10). This function corresponds to Figure 1(d):

$$L^{FC}(\mathrm{y},\widehat{y^s},\widehat{y^t}) = (1-\lambda)L_{FC}^{S}(\mathrm{y},\widehat{y^s}) + \lambda L_{FC}^{KD}\left(\widehat{y^t},\widehat{y^s}\right) \tag{9}$$

$$L^{FC}(\mathrm{y},\widehat{y^s},\widehat{y^t}) = -(1-\lambda)\alpha\sum_{i=1}^{C} y_i(1-\widehat{y_i^s})\log\widehat{y_i^s} - \lambda T^2\alpha\sum_{i=1}^{C}\frac{\widehat{y_i^t}}{T}(1-\frac{\widehat{y_i^s}}{T})^{\lambda}\log\frac{\widehat{y_i^s}}{T} \tag{10}$$

This paper also evaluates the effect of a combination between FC and CE or between FC and BCE. We propose a new loss function that is composed of two hyper parameters as shown in (11) and (12). This function relates to both Figures 1(c) and 1(d).

$$\begin{aligned} L^{CE-FC}(\mathrm{y},\widehat{y^s},\widehat{y^t}) &= \beta_1 L^{CE}\left(y,\widehat{y^s},\widehat{y^t}\right) + \beta_2 L^{FC}(\mathrm{y},\widehat{y^s},\widehat{y^t}) \\ &= \beta_1\left[-(1-\lambda)\alpha\sum_{i=1}^{C} y_i\log\widehat{y_i^s} - \lambda T^2\alpha\sum_{i=1}^{C}\frac{\widehat{y_i^t}}{T}\log\frac{\widehat{y_i^s}}{T}\right] \\ &\quad + \beta_2\left[-(1-\lambda)\alpha\sum_{i=1}^{C} y_i(1-\widehat{y_i^s})\log\widehat{y_i^s} - \lambda T^2\alpha\sum_{i=1}^{C}\frac{\widehat{y_i^t}}{T}(1-\frac{\widehat{y_i^s}}{T})^{\lambda}\log\frac{\widehat{y_i^s}}{T}\right] \end{aligned} \tag{11}$$

$$\begin{aligned} L^{BCE-FC}(\mathrm{y},\widehat{y^s},\widehat{y^t}) &= \beta_1 L^{BCE}\left(y,\widehat{y^s},\widehat{y^t}\right) + \beta_2 L^{FC}(\mathrm{y},\widehat{y^s},\widehat{y^t}) \\ &= \beta_1\left[-(1-\lambda)\sum_{i=1}^{C} y_i\log\widehat{y_i^s} - \lambda T^2\sum_{i=1}^{C}\frac{\widehat{y_i^t}}{T}\log\frac{\widehat{y_i^s}}{T}\right] \\ &\quad + \beta_2\left[-(1-\lambda)\alpha\sum_{i=1}^{C} y_i(1-\widehat{y_i^s})\log\widehat{y_i^s} - \lambda T^2\alpha\sum_{i=1}^{C}\frac{\widehat{y_i^t}}{T}(1-\frac{\widehat{y_i^s}}{T})^{\lambda}\log\frac{\widehat{y_i^s}}{T}\right] \end{aligned} \tag{12}$$

2.3. Teacher and student networks

In this research, Resnet50 [23] or DenseNet121 [24] is utilized as a teacher model, and Resnet18 [23] is used as a student model. These CNN networks are trained on the ImageNet dataset. A for the KD training, two steps are deployed, as follows: in the first one, a large CNN network was transferred by a new dataset to archive a large pre-trained model. In the second one, both the large pre-trained model and a small CNN network are utilized for training the KD network using the above new dataset. The large model is blocked and all layers of the small CNN network are fine-tuned. We deployed two cases including: i) teacher is Resnet50 network, student is Resnet18 network (KD Resnet18-Resnet50 framework); ii) teacher is DenseNet121 network, student is Resnet18 network (KD Resnet18-DenseNet121 framework). The setup of Resnet18, Resnet50, DenseNet121 and the KD frameworks are presented in detail in Table 1.

In addition, some factors of FC function in (9)-(12) are also used as follows: T=6.0, α=0.95, λ=2.0. In particular, two hyper parameters β1 and β2 are considered and chosen in detail in section 3. The CNN models are obtained at 100 epochs on both the training sub-dataset and the validation sub-dataset.

Table 1. The setup details of CNN architectures

Parameter	Resnet18	Resnet50	DenseNet121	KD
Learning rate	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Batch size	64 images	64 images	64 images	64 images
Optimizer	Adam	Adam	Adam	Adam
Loss function	CE	CE	CE	CE, BCE, FC
Transfer learning	All layer	All layer	All layer	All layer (Resnet18), No Layer (Resnet50) All layer (Resnet18), No Layer (DenseNet212)
Input image	224x224 pixels	224x224 pixels	224x224 pixels	224x224 pixels

2.4. Evaluation protocols and measurements

In this study, four benchmark datasets are used to evaluate the KD method, including: CIFAR-100 [25], EPUHandInWild3 [26], Kinect Leap [27], and Creative Senz3D [28]. The CIFAR-100 dataset has 100 categories, each class contains 500 training images and 100 testing images. The training part is divided into the training sub-dataset and a validation sub-dataset with the ratio of 80% and 20%, respectively. A CIFAR-100 dataset is evaluated in N=10 times. Each evaluation obtained accuracy ($Acc_{\ i}$). The final result is then averaged as shown in (13):

$$Acc_{CIFAR100} = \frac{\sum_{i=1}^{N=10} Acc_i}{N} \tag{13}$$

Remaining datasets conclude EPUHandInWild3 [26], Kinect Leap [27], and Creative Senz3D [28] we follow ”One-leave-subject-out” for the subject independence test. That means that one subject is tested and the remaining subjects are used for training models as presented in our previous research [7]. In this paper, each testing subject is evaluated in N=10 times. Experiments are rolled out for every subject in each dataset to ensure that every person is tested. Hand gestures of a subject are used for testing that does not appear in the training phase. The number of subjects in each dataset is M, Each subject obtains accuracy at a certain time (Acc_i^j, i , i = (1, ..., N), j = (1, ..., M)). Final accuracy is computed as illustrated in (14):

$$Acc = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} Acc_i^j}{N*M} \tag{14}$$

3. EXPERIMENTAL RESULTS

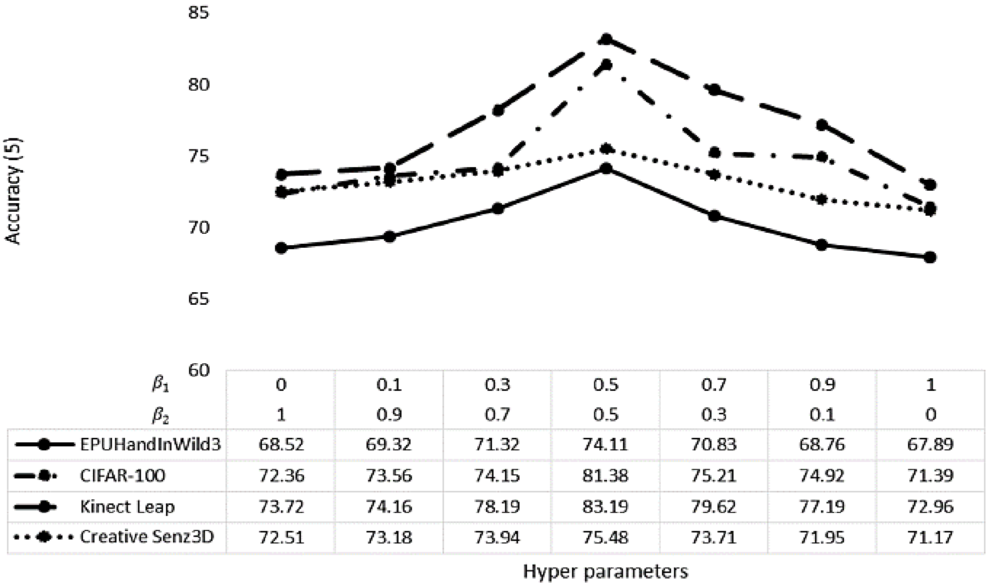
The experiments are conducted to indicate which is the best loss function for recognition in the KD framework. We also analyze the role of a combination between optimal functions, including: CE-FC loss, BCE-FC loss. Our method is compared with some state-of-the-art (SOTA) KD solutions, such as: KD [8], FDA [10], Margin [16], and AB [29]. The experiments are performed on the same CNN architectures (KD Resnet18-Resnet50 framework) and different CNN styles (KD Resnet18-DenseNet121 framework). The evaluation schemes are written in Python on a Pytorch deep learning framework and run on a workstation with an NVIDIA GPU 11G.

3.1. Evaluation of hyper parameters on recognition by knowledge distillation method

In this section, we evaluate the recognition accuracy of the same types of CNN architectures (Resnet50 is a teacher and Resnet18 is a student) with various values of hyper parameters of our proposed KD methods. The first loss combination is the CE-FC loss in (11); the results are illustrated in Figure 2(a). The second one is the BCE-FC loss in (12) which is shown in Figure 2(b). Figure 2 shows that the best accuracy is at $\beta_1 = \beta_2 = 0.5$ and the worst at $\beta_1 = 1$ and $\beta_2 = 0$. It means that the combinations of CE loss and FC loss are better than a single loss function. Furthermore, the best accuracy results at $\beta_1 = \beta_2 = 0.5$ of BCE-FC loss (74.11%, 81.38%, 83.19%, 75.48% for EPUHandInWild3, CIFAR-100, Kinect Leap, Creative

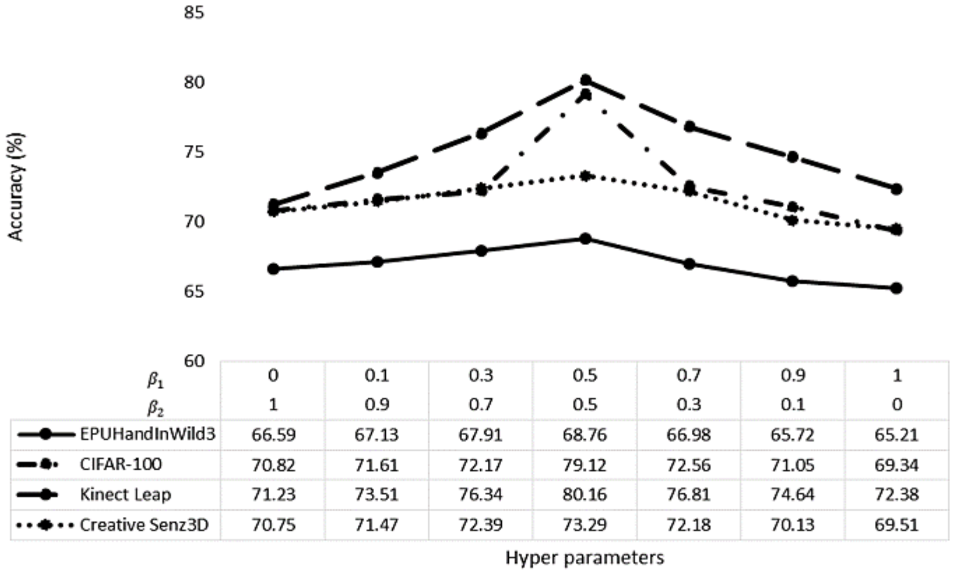
Senz3D, respectively) are higher than CE-FC strategy (68.76%, 79.12%, 80.16%, 73.29% for EPUHandInWild3, CIFAR-100, Kinect Leap, Creative Senz3D, respectively) on the entire four benchmark datasets. Thus, we will apply $\beta_1 = \beta_2 = 0.5$ in the remaining evaluations.

Figure 3 shows four confusion matrices of experimental results, such as Resnet18 in Figure 3(a); Resnet50 in Figure 3(b); KD of Resnet18-Resnet50 with CE-FC loss in Figure 3(c); KD framework of Resnet18-Resnet50 with BCE-FC loss function in Figure 3(d). This figure shows that the accuracy of the KD method of Resnet18-Resnet50 model with BCE-FC loss dramatically increases in entire categories after KD by Resnet50 model, such as: (91.8%, 84.0%, 73.2%, 52.6%, 80.7%, 78.6%, 57.9%) of BCE-FC loss and (83.5%, 80.0%, 71.1%, 49.2%, 77.8%, 72.2%, 48.3%) of CE-FC loss for (G1, G2, G3, G4, G5, G6, G7, respectively). The results show that the recognition accuracy of the Resnet18 network significantly increases when Resnet18 is used for KD by teachers model (DensetNet121 or Resnet50). However, knowledge of the Resnet50 model is better transferred than knowledge of the DensetNet121 model. Thus, we will use the Resnet50 model for KD in the remaining evaluations.



β_1	0	0.1	0.3	0.5	0.7	0.9	1
β_2	1	0.9	0.7	0.5	0.3	0.1	0
EPUHandInWild3	68.52	69.32	71.32	74.11	70.83	68.76	67.89
CIFAR-100	72.36	73.56	74.15	81.38	75.21	74.92	71.39
Kinect Leap	73.72	74.16	78.19	83.19	79.62	77.19	72.96
Creative Senz3D	72.51	73.18	73.94	75.48	73.71	71.95	71.17

(a)



β_1	0	0.1	0.3	0.5	0.7	0.9	1
β_2	1	0.9	0.7	0.5	0.3	0.1	0
EPUHandInWild3	66.59	67.13	67.91	68.76	66.98	65.72	65.21
CIFAR-100	70.82	71.61	72.17	79.12	72.56	71.05	69.34
Kinect Leap	71.23	73.51	76.34	80.16	76.81	74.64	72.38
Creative Senz3D	70.75	71.47	72.39	73.29	72.18	70.13	69.51

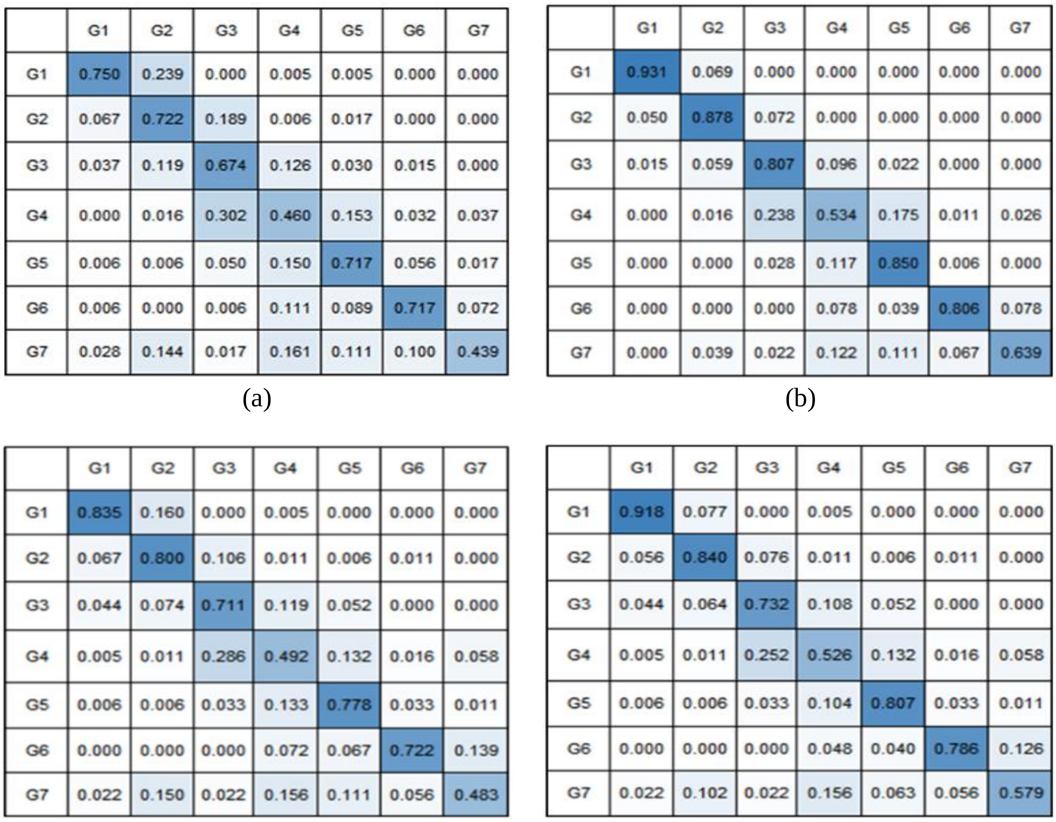
(b)

Figure 2. The accuracy with various strategies combination of hyper parameters, (a) combination between BCE and FC functions, and (b) combination between CE and FC functions

3.2. Comparison of knowledge distillation methods

In this section, we perform our proposed KD methods for the same network architecture types and the different network architecture types at hyper parameter values $\beta_1 = \beta_2 = 0.5$. In addition, we perform evaluations on three CNN models Resnet18, Resnet50, and DenseNet121, with the results are shown in the first, second and third columns in Figure 4, respectively. Four KD models with different optimization functions in (11) and (12) that correspond to KD models of Resnet18-Resnet50 with CE-FC loss, Resnet18-DenseNet121 with CE-FC loss, Resnet18-Resnet50 with BCE-FC loss, Resnet18-DenseNet121 with BCE-FC loss are also evaluated, with the results illustrated from the fourd columns to the seventh columns in Figure 4, respectively. Evaluations are deployed on four benchmark datasets. The results in Figure 4 are Top 1 recognition accuracy, and they demonstrate that:

- The KD Resnet18-Resnet50 models and Resnet18-DenseNet121 models (from the fourth column to the seventh column in each group of Figure 4) archive higher accuracy than the original Resnet18 model (from the first column to the third column in each group of Figure 4) on entire four benchmark datasets. It is evident that our KD method is efficient in transferring knowledge from the teacher model to the student model.
- The KD models transfer between the same style of network architecture (Resne18-Resnet50) obtains larger accuracy than different architecture network styles (Resnet18-DenseNet121), such as: the fourth columns compare with the fifth columns, and the sixth columns compare with the seventh columns in Figure 4. It shows that KD model of the similarity architectures is more efficient than the difference models.
- The KD methods use the BCE-FC loss function (the sixth and the seventh columns in Figure 4) obtain higher accuracy than KD with the CE-FC loss function (the fourth columns and the fifth columns in Figure 4) on both the same architecture styles as well as different architecture styles.
- The KD Resnet18-Resnet50 model with BCE-FC loss (the sixth columns) archives the highest accuracy at 81.38%, 74.11%, 83.19%, and 75.48% for CIFAR-100, EPUHandInWild3, Kinect Leap, and Creative Senz3D, respectively. It means that the combination between BCE loss function and FC loss function gives the best transfer from the teacher model (Resnet50) to the student model (Resnet18).



	G1	G2	G3	G4	G5	G6	G7
G1	0.750	0.239	0.000	0.005	0.005	0.000	0.000
G2	0.067	0.722	0.189	0.006	0.017	0.000	0.000
G3	0.037	0.119	0.674	0.126	0.030	0.015	0.000
G4	0.000	0.016	0.302	0.460	0.153	0.032	0.037
G5	0.006	0.006	0.050	0.150	0.717	0.056	0.017
G6	0.006	0.000	0.006	0.111	0.089	0.717	0.072
G7	0.028	0.144	0.017	0.161	0.111	0.100	0.439

(a)

	G1	G2	G3	G4	G5	G6	G7
G1	0.931	0.069	0.000	0.000	0.000	0.000	0.000
G2	0.050	0.878	0.072	0.000	0.000	0.000	0.000
G3	0.015	0.059	0.807	0.096	0.022	0.000	0.000
G4	0.000	0.016	0.238	0.534	0.175	0.011	0.026
G5	0.000	0.000	0.028	0.117	0.850	0.006	0.000
G6	0.000	0.000	0.000	0.078	0.039	0.806	0.078
G7	0.000	0.039	0.022	0.122	0.111	0.067	0.639

(b)

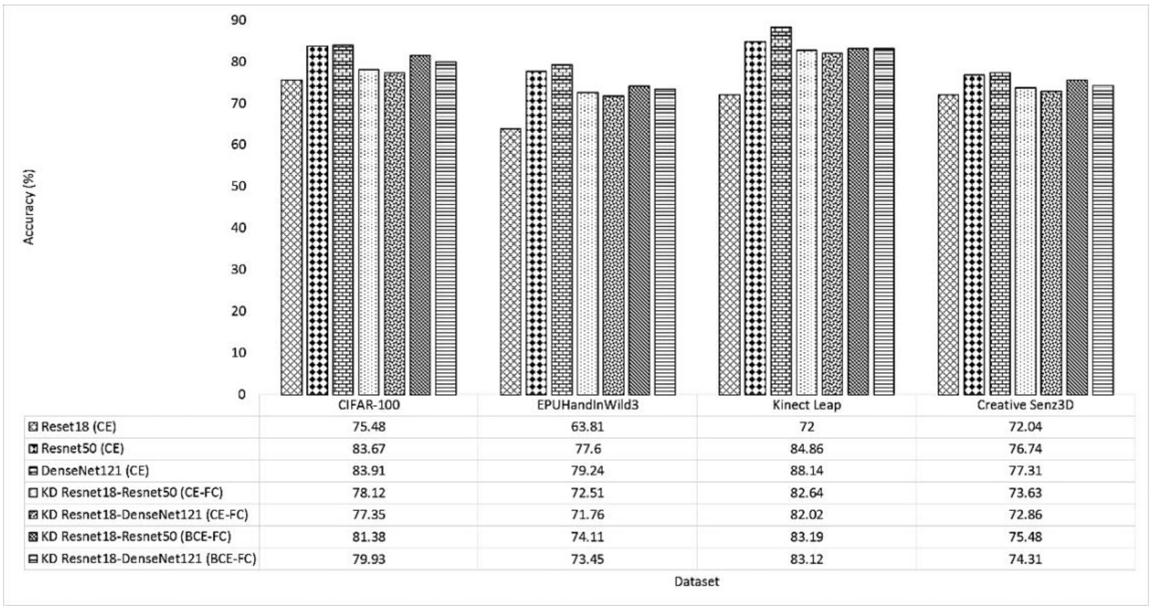
	G1	G2	G3	G4	G5	G6	G7
G1	0.835	0.160	0.000	0.005	0.000	0.000	0.000
G2	0.067	0.800	0.106	0.011	0.006	0.011	0.000
G3	0.044	0.074	0.711	0.119	0.052	0.000	0.000
G4	0.005	0.011	0.286	0.492	0.132	0.016	0.058
G5	0.006	0.006	0.033	0.133	0.778	0.033	0.011
G6	0.000	0.000	0.000	0.072	0.067	0.722	0.139
G7	0.022	0.150	0.022	0.156	0.111	0.056	0.483

(c)

	G1	G2	G3	G4	G5	G6	G7
G1	0.918	0.077	0.000	0.005	0.000	0.000	0.000
G2	0.056	0.840	0.076	0.011	0.006	0.011	0.000
G3	0.044	0.064	0.732	0.108	0.052	0.000	0.000
G4	0.005	0.011	0.252	0.526	0.132	0.016	0.058
G5	0.006	0.006	0.033	0.104	0.807	0.033	0.011
G6	0.000	0.000	0.000	0.048	0.040	0.786	0.126
G7	0.022	0.102	0.022	0.156	0.063	0.056	0.579

(d)

Figure 3. Confusion matrix of EPUHandInWild3 dataset with Resnet50 and Resnet18 networks, (a) Resnet18, (b) Resnet50, (c) CE-FC KD, and (d) BCE-FC KD



	CIFAR-100	EPUHandInWild3	Kinect Leap	Creative Senz3D
▨ Reset18 (CE)	75.48	63.81	72	72.04
▨ Resnet50 (CE)	83.67	77.6	84.86	76.74
▤ DenseNet121 (CE)	83.91	79.24	88.14	77.31
□ KD Resnet18-Resnet50 (CE-FC)	78.12	72.51	82.64	73.63
▨ KD Resnet18-DenseNet121 (CE-FC)	77.35	71.76	82.02	72.86
▧ KD Resnet18-Resnet50 (BCE-FC)	81.38	74.11	83.19	75.48
▤ KD Resnet18-DenseNet121 (BCE-FC)	79.93	73.45	83.12	74.31

Figure 4. Top 1 accuracy of Resnet18 model with and without uses KD

In addition, we also evaluate the Top 1 and Top 5 accuracy of the CIFAR-100 dataset for four compressed models as illustrated in Table 2. The result shows that:

- Using BCE-FC loss is dramatically higher than CE-FC loss for the Top 5 in accuracy at 96.19% vs 89.51% for the same Resnet architecture styles (Resnet18-Resnet50) and 93.64% vs 87.96% for different CNN architecture styles (Resnet18-DenseNet121). It is apparent that BCE-FC loss is more soft digit than CE- FC loss.
- The KD Resnet18-Resnet50 model with BCE-FC loss function accounts for the highest accuracy with both Top 1 and Top 5 accuracy at 81.38% and 96.64%, respectively. This result once again shows that knowledge is best transferred on the same CNN architecture styles (Resnet18-Resnet50) with BCE-FC loss.

Table 2. Top 1 and Top 5 accuracy (%) of CIFAR-100 dataset with various the KD models

	Top 1	Top 5
KD Resnet18-Resnet50 (CE-FC)	78.12	89.51
KD Resnet18-DenseNet121 (CE-FC)	77.35	87.96
KD Resnet18-Resnet50 (BCE-FC)	81.38	96.19
KD Resnet18-DenseNet121 (BCE-FC)	79.93	93.64

3.3. Comparison our method with SOTA knowledge distillation methods

In this section, we compare the efficiency of the proposed teacher-student methods with SOTA KD methods (KD [8]–the first rows, FDA [10]–the second rows, Margin [16]–the third rows, and AB [29]–the ford rows in Table 3)). Our best option of BCE and FC function at $\beta_1 = \beta_2 = 0.5$ (the fifth rows of Table 3) is compared with SOTA methods. All methods are deployed in 10 times and averaged as presented in section 2.4. Table 3 shows the accuracy of four various benchmark datasets. Table 3 illustrates that our proposed method outperforms SOTA KD frameworks on the entire four benchmark datasets at 81.30%, 74.11%, 83.19%, and 75.48% in accuracy for CIFAR-100, EPUHandInWild3, Kinect Leap, and Creative Senz3D, respectively.

Table 3. Comparison accuracy (%) between our method with SOTA teacher and student methods

	CIFAR-100	EPUHandInWild3	Kinect_Leap	Creative senz3d
KD [8]	76.12	65.39	74.21	73.16
FDA [10]	77.92	67.24	78.74	73.58
Margin [16]	72.27	64.27	76.49	72.81
AB [29]	76.45	66.13	75.26	72.98
Our KD Resnet18-Resnet50 (BCE-FC)	81.38	74.11	83.19	75.48

4. CONCLUSION

This paper presents a comparative analysis of KD methods with two similar CNN architecture styles (Resnet18-Resnet50) and two different CNN architecture styles (Resnet18-DenseNet121). In addition, we also investigated the efficiency of various loss functions on the KD models. Our proposed method is evaluated on four benchmark static datasets. Among the evaluated models, the same architecture styles of Resnet18-Resnet50 in combination with BCE-FC loss archives the best recognition accuracy. Its performance is superior to that of other SOTA works by 2.67% for Creative Senz3D dataset and 9.84% for EPUHandInWild3 dataset at Top 1 accuracy. Especially, our KD method obtains 96.19% of the Top 5 accuracy for the CIFAR-100 dataset. This performance of the proposed solution is remarkable and promises deployment in other KD models.

REFERENCES

[1] F. Yuan *et al.*, "Reinforced multi-teacher selection for knowledge distillation," *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 16, no. 16, pp. 14284–14291, 2021, doi: 10.1609/aaai.v35i16.17680.

[2] R. Sovia, S. Defit, Yuhandri, and Sulastri, "Development of natural language processing on morphology-based Minangkabau language stemming algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 1, pp. 542–552, 2023, doi: 10.11591/ijeecs.v31.i1.pp542-552.

[3] H. Tan, X. Liu, M. Liu, B. Yin, and X. Li, "KT-GAN: knowledge-transfer generative adversarial network for text-to-image synthesis," *IEEE Transactions on Image Processing*, vol. 30, pp. 1275–1290, 2021, doi: 10.1109/TIP.2020.3026728.

[4] L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov, "Knowledge distillation: a good teacher is patient and consistent," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2022, pp. 10915–10924, doi: 10.1109/CVPR52688.2022.01065.

[5] H. G. Doan, H. Q. Luong, T. O. Ha, and T. T. T. Pham, "An efficient strategy for catastrophic forgetting reduction in incremental learning," *Electronics*, vol. 12, no. 10, 2023, doi: 10.3390/electronics12102265.

[6] T. Vuletic, A. Duffy, L. Hay, C. McTeague, G. Campbell, and M. Grealy, "Systematic literature review of hand gestures used in human computer interaction interfaces," *International Journal of Human Computer Studies*, vol. 129, pp. 74–94, Sep. 2019, doi: 10.1016/j.ijhcs.2019.03.011.

[7] H. G. Doan, "Multiple views and categories condition GAN for high resolution image," in *Lecture Notes on Data Engineering and Communications Technologies*, Springer International Publishing, 2022, pp. 507–520, doi: 10.1007/978-3-030-97610-1_40.

[8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv-Statistics*, pp. 1-9, 2015.

[9] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: a survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021, doi: 10.1007/s11263-021-01453-z.

[10] Y. Guan *et al.*, "Differentiable feature aggregation search for knowledge distillation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer International Publishing, 2020, pp. 469–484, doi: 10.1007/978-3-030-58520-4_28.

[11] R. Miles and K. Mikolajczyk, "Understanding the role of the projector in knowledge distillation," *Arxiv-Computer Science*, pp. 1-9, 2022.

[12] C. Bucilă, R. Caruana, and A. N. -Mizil, "Model compression," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD06, ACM, 2006, pp. 535–541, doi: 10.1145/1150402.1150464.

[13] L. Wang and K. J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: a review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3048–3068, 2022, doi: 10.1109/TPAMI.2021.3055564.

[14] H. Yao *et al.*, "Graph few-shot learning via knowledge transfer," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 6656–6663, Apr. 2020, doi: 10.1609/aaai.v34i04.6142.

[15] Z. Zhang *et al.*, "Object relational graph with teacher-recommended learning for video captioning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2020, pp. 13275–13285, doi: 10.1109/CVPR42600.2020.01329.

[16] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, 2019, pp. 1921–1930, doi: 10.1109/ICCV.2019.00201.

[17] G. Liu, Y. Shang, Y. Yao, and R. Kompella, "Network specialization via feature-level knowledge distillation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, 2023, pp. 3368–3375, doi: 10.1109/CVPRW59228.2023.00339.

[18] M. Siam *et al.*, "Video object segmentation using teacher-student adaptation in a human robot interaction (HRI) setting," in *IEEE International Conference on Robotics and Automation*, IEEE, 2019, pp. 50–56, doi: 10.1109/ICRA.2019.8794254.

[19] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, 1997, doi: 10.1016/S0377-2217(96)00385-2.

[20] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951, doi: 10.1214/aoms/1177729694.

[21] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, 2015, pp. 1395–1403, doi: 10.1109/ICCV.2015.164.

[22] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020, doi: 10.1109/TPAMI.2018.2858826.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[24] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, IEEE, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Computer Science, University of Toronto*, pp. 1–58, 2009.

[26] H. G. Doan and N. T. Nguyen, “New blender-based augmentation method with quantitative evaluation of CNNs for hand gesture recognition,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 2, pp. 796–806, 2023, doi: 10.11591/ijeecs.v30.i2.pp796-806.

[27] G. Marin, F. Dominio, and P. Zanuttigh, “Hand gesture recognition with jointly calibrated leap motion and depth sensor,” *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14991–15015, 2016, doi: 10.1007/s11042-015-2451-6.

[28] A. Memo and P. Zanuttigh, “Head-mounted gesture controlled interface for human-computer interaction,” *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 27–53, 2018, doi: 10.1007/s11042-016-4223-3.

[29] B. Heo, M. Lee, S. Yun, and J. Y. Choi, “Knowledge transfer via distillation of activation boundaries formed by hidden neurons,” *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, vol. 33, no. 01, pp. 3779–3787, 2019, doi: 10.1609/aaai.v33i01.33013779.

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, Hanoi, Vietnam. She can be contacted at email: 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.