Journal of information and
communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 21(2): 139-144, Jun. 2023 Regular paper

Improving Adversarial Domain Adaptation with Mixup Regularization

Bayarchimeg Kalina[1] **and Youngbok Cho**[1]*

[1]Department of Information Security, Daejeon University, Daejeon 300-716, Republic of Korea

Abstract

Engineers prefer deep neural networks (DNNs) for solving computer vision problems. However, DNNs pose two major problems. First, neural networks require large amounts of well-labeled data for training. Second, the covariate shift problem is common in computer vision problems. Domain adaptation has been proposed to mitigate this problem. Recent work on adversarial-learning-based unsupervised domain adaptation (UDA) has explained transferability and enabled the model to learn robust features. Despite this advantage, current methods do not guarantee the distinguishability of the latent space unless they consider class-aware information of the target domain. Furthermore, source and target examples alone cannot efficiently extract domain-invariant features from the encoded spaces. To alleviate the problems of existing UDA methods, we propose the mixup regularization in adversarial discriminative domain adaptation (ADDA) method. We validated the effectiveness and generality of the proposed method by performing experiments under three adaptation scenarios: MNIST to USPS, SVHN to MNIST, and MNIST to MNIST-M.

Index Terms: Adversarial discriminative domain adaptation (ADDA), Domain-invariant, Mixup, Unsupervised domain adaptation (UDA)

I. INTRODUCTION

Adversarial discriminative domain adaptation (ADDA) was first introduced in [1] and is categorized as a feature-based domain adaptation method. In recent unsupervised domain adaptation (UDA) techniques, adversarial learning has been employed to learn invariant features across domains. Using a min-max two-player game in which generators master confusing domain discriminators, ADDA models learn discriminative representations and invariant features. Although it is highly successful at various tasks, such as image classification and semantic segmentation, it has two major shortcomings. First, domain classifiers only differentiate features as sources or targets, and do not consider task-specific decision boundaries between classes. Second, it attempts to perfectly align the feature distributions between different domains on the basis of the characteristics of each domain. To address existing UDA issues, category mixup regularization (CMR) was applied in this study to ADDA. We implemented it on the source and target domain samples to make the model predictions insensitive to perturbations. This approach helps the model learn meaningful representations across domains. Our main contributions are summarized as follows:

- We propose a category mixup-regularized learning technique to improve ADDA generality, which maps the source and target domains to a common latent code and transfers the learned knowledge from the annotated source domain to the label-free target domain.
- Experimental results show that regularization techniques enable deep-learning models to learn better discrimina-

Received 17 February 2023, **Revised** 23 April 2023, **Accepted** 16 May 2023
***Corresponding Author** Youngbok Cho (E-mail: ybcho@dju.ac.kr)
Department of Information Security, Daejeon University, Daejeon 300-716, Republic of Korea

Open Access https://doi.org/10.56977/jicce.2023.21.2.139 print ISSN: 2234-8255 online ISSN: 2234-8883

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright ⓒ The Korea Institute of Information and Communication Engineering

tive and domain-invariant representations, which effectively reduce large domain shifts, such as the SVHN to MNIST adaptation task.
- We validated the effectiveness and generality of the proposed method by evaluating it on three adaptation tasks. According to the experimental results, our improved ADDA method is a strong competitor to previous UDA methods.
- Regularization techniques using CMR also address the problem of training unstable deep networks.

The remainder of this paper is organized as follows: In Section 2, interpolation-based regularization and domain adaptation are described in related works, and in Section 3, the proposed method is presented in detail. In Section 4, the experimental setup and results are described, and the conclusions are presented in Section 5.

II. RELATED WORK

A. Interpolation-Based Regularization

Interpolation-based regularization [2,3] was recently proposed for supervised learning and can mitigate model uncertainty in adversarial training and vulnerability to adversarial samples. Consequently, [3] suggested that a mixup is a superior implementation for training models on virtual example sets as linear combinations of inputs and labels. This end is achieved by regularizing the output of the model for a convex combination of the two inputs. Mixup [4] was used to confirm the consistent predictions in the data distribution. In recent years, several versions of Mixup have been studied. Among them, manifold mixup [1] was proposed to interpolate latent space representations.

B. Domain Adaptation

Domain adaptation (DA) transfers knowledge from one domain with abundant labeled data to another domain with no labeled data. DA discovers shared latent representations across the source and target domains. It adapts them to minimize marginal and conditional inconsistencies in the embedded space. Recently, UDA methods have been considered for learning domain-invariant representations using adversarial training. Cycle-consistent adversarial domain adaptation (CyCADA) [5] transforms feature representations at both the pixel and feature levels while executing pixel and semantic consistency. The cycle-consistency term forces the cross-domain transformation to retain pixel information, whereas the semantic-loss term forces semantic consistency. Multi-adversarial domain adaptation (MADA) [6] exploits the generative interconnection between feature representations and label predictions to execute adversarial learning. Generate-to-adapt (GTA) [7] introduces a novel adversarial image generation method that learns a latent representation that reduces the domain shift between the source and target domains. Pixel-level adapt [8] transforms source-domain images into target-like images. Two main approaches have been attempted. The first attempt was to determine a mapping function from the source-to-target domain representation. The second attempt was to identify domain-invariant representations that were not restricted to one domain. Existing DAs learn the target encoders according to the target task. However, because it learns to separate the target and reduces the domain shift between real and fake images at the pixel level, it is less affected by the tasks.

III. THE PROPOSED METHODS

A. Mixup

Mixup [3] was proposed as a data- and domain-agnostic technique. Deep neural networks (DNNs) memorize the corrupted labels. To address this problem, a mixup is proposed that combines the features of different samples such that the network is not overconfident about the relationship between features and labels. This technique can be extended to various data modes such as computer vision, natural language processing, and speech recognition.

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda) x_j, \quad x_i, x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda) y_j, \quad y_i, y_j \end{aligned} \tag{1}$$

Equation (1) first expresses the original input vector and the one-hot label encoding based on it as follows: The values are in the range [0, 1], and are sampled from the beta distribution.

Neural networks share two common attributes for successful application. To reduce mean error, deep learning model is trained using a weighted combination of features and labels. This technique is known as empirical risk minimization (ERM) [3]. Subsequently, the performance of these state-of-the-art neural networks [3] is improved by increasing the amount of training data. The predictions of the neural networks trained with the ERM are unstable when tested on samples with different data distributions. In recent years, mixup regularization has been used to alleviate this problem when training neural networks using annotated data.

B. Adversarial Discriminative Domain Adaptation

A feature-based domain adaptation method called ADDA [1] ensures that target mapping reduces the distance between the source and target domains. Training robust deep networks using generative-adversarial loss reduces domain shifts and allows the generator network to produce synthetic samples across various domains without sharing weights with the discriminator. To fool discriminator networks, ADDA masters a novel feature representation. This feature representation is learned from two encoder networks. The source encoder is designed to yield superior features for learning tasks in the source domain. The task is mastered using a task network conditioned on a source encoder. The target encoder and discriminator are trained adversarially. The parameters of the four networks are optimized using a two-step algorithm, in which the source and task networks are fitted first, as shown by the following optimization problem:

$$\begin{gathered}\min_{\phi_s,F} L_{task}\left(F\left(\phi_s(X_s)\right), y_s\right)\\ \min_{\phi_t} -\log\left(D\left(\phi_t(X_t)\right)\right)\\ \min_{D} -\log\left(D\left(\phi_s(X_s)\right)\right) - \log\left(1 - D\left(\phi_t(X_t)\right)\right)\end{gathered} \tag{2}$$

In Eq. (2), (X_s, y_s) and (X_t) are the labeled source data and label-free target data, respectively. $\phi_s, \phi_t, F,$ and D represent the source encoder, target encoder, and discriminator network, respectively.

C. ADDA with Category Mixup Regularization

This section describes the training process for improved ADDA using CMR. Fig. 1 illustrates the network structure proposed in this study. Feature extractors learn domain-invariant features while extracting high-level representations of scenes from input images ("shape of digit 1 in front of images"). The discriminator determines whether the input data come from a source domain or target domain by analyzing its distribution. This step is performed without sharing weights with the feature extractor network. Digit classification is performed using a classifier. According to Fig. 1, the CMR mechanism is implemented in both the source and target domains to improve the latent representation. To improve the generality of deep-learning models, it is essential to use a consistent regularization technique. The CMR forces model prediction to be insensitive to perturbed inputs by separating feature representations. During the testing phase, we evaluate the improved ADDA generality on the target test data using a learned feature extractor and classifier. We provide more details on updating the gradients of the feature extractor, classifier, and discriminator networks in the Optimization Problem section.

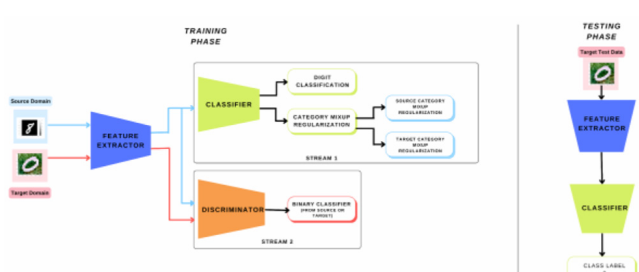


Fig. 1. Architecture of the proposed method.

D. Optimization Problem

Our method uses source and target images to update the gradients of the feature extractors, classifiers, and discriminators. We perform adversarial learning to learn invariant features across domains. For the source domain, because the label information is available, we employ mixed-source examples and their corresponding labels to implement consistent predictions.

$$L_{src}^{r}(F_s, C) = E_{\left(x_i^{src}, y_i^{src}\right),\left(x_j^{src}, y_j^{src}\right)\sim D_{src}} L\left(p_{\hat{x}^{src}}^{C}, \hat{y}^{src}\right) \tag{3}$$

where D_{src} denotes the source-domain distribution. The "r," "src" and "tgt" notations represent regularization, source domain, and target domain respectively. However, we do not have access to the label information for the target domain. Consequently, the mixup can be applied to the pseudo-labels. Specifically, we replace y_i^{tgt} and y_j^{tgt} with $p_{x_i^{tgt}}^{C}$ and $p_{x_j^{tgt}}^{C}$, which are the current predictions of the classifier network. We also need to assemble linear combinations, denoted as ($\hat{x}^{tgt}$, Mix_λ $(p_{x_i^{tgt}}^{C}, p_{x_j^{tgt}}^{C})$), of pairs of target samples (x_i^{tgt} and x_j^{tgt}) and their pseudo-labels $(p_{x_i^{tgt}}^{C}, p_{x_j^{tgt}}^{C})$. Subsequently, regularization is implemented by making $p_{\hat{x}^{tgt}}^{C}$ consistent with Mix_λ $(p_{x_i^{tgt}}^{C}, p_{x_j^{tgt}}^{C})$ using a penalty term:

$$\begin{gathered}L_{tgt}^{r}(F_t, C) =\\ E_{x_i^{tgt}, x_j^{tgt}\sim D_{tgt}}\, dis\left(p_{\hat{x}^{tgt}}^{C}, Mix_\lambda\left(p_{x_i^{tgt}}^{C}, p_{x_j^{tgt}}^{C}\right)\right)\end{gathered} \tag{4}$$

where D_{tgt} represents the target domain distribution and $dis()$ indicates the penalty term that punishes the disagreement between $p_{\hat{x}^{tgt}}^{C}$ and $Mix_\lambda(p_{x_i^{tgt}}^{C}, p_{x_j^{tgt}}^{C})$. The λ parameter set to 0.2 in both unlabeled and labeled CMR. The L1-Norm function is used as a penalty term during training. CMR can regularize the output distribution of networks by producing neighboring examples of the training and penalizing perpixel inconsistent predictions between the created neighboring and training examples, which executes class-aware knowledge of the target domain during the training phase.

$$L_{src}^{C} = L_C\left(F_s(X_{src}), Y_{src}\right) \tag{5}$$

$$L_{adv}^{D}(F_s, F_t, D) = E_{D_{src}} \log D\left(F_s(X_{src})\right) + E_{D_{tgt}} \log\left(1 - D\left(F_t(X_{tgt})\right)\right) \tag{6}$$

$$L_{total} = L_{src}^{C} + L_{adv}^{D} + \mu_1 L_{src}^{r} + \mu_2 L_{tgt}^{r} \tag{7}$$

where μ_1, μ_2 are tradeoff parameters that are set to 1,1 during training, respectively. Equation (5) indicates the classifier loss of the source domain samples, Eq. (6) calculates the adversarial loss, and finally, the total loss is calculated by Eq. (7). To build robust deep-learning models, consistent regularization is essential. By implementing mixup regularization in the categorical distributions of the source and target data, the model predictions are not affected by perturbations. By increasing the distance between features, mixup regularization enhances the latent-space discriminability.

IV. EXPERIMENTS AND RESULTS

In this section, we provide details regarding the experimental environment setup and results. Fig. 1 shows that our network consists of three components: a generator, a classifier, and a discriminator. All the networks were built using DNNs. The Google Colab platform was used for this experiment. The Google Colab platform had an Intel CPU with two cores at a clock speed of 2.30 GHz in the session. Google Colab offers 12 GB of free memory. The GPU was a Tesla T4, and the Linux operating system version was 20.04.5 LTS. All the experiments were implemented on the PyTorch platform, and the Torch version was 1.13.0+cu116. All the networks were trained from scratch using the Adam optimizer. The learning rate and batch size were set to 0.0001 and 64, respectively.

A. Effects of CMR In the ADDA Method

The aim of this study was to improve the ADDA method. Further investigation was conducted to determine how CMR enhances the generality of the ADDA methods. Table 1 shows that all adaptation tasks performed poorly after the category mixup regularizations were disconnected. Specifically, the performance of the M→MM and S→M adaptation tasks degraded significantly by 9% and 8%, respectively. The results also suggest that the proposed approach improves the generality of the ADDA method. Compared to previous state-of-the-art approaches, such as pixel-level adaptation [8], CyCADA [5], and MCD [9], our approach is considered a strong competitor. We further visualized the feature distribution of the target domain in the MNIST to MNIST-M (M→MM), MNIST to USPS (M→U), and SVHN to MNIST (S→M) unsupervised adaptation tasks using the T-SNE tool. The T-SNE visualizations are shown in Fig. 3. The T-SNE plots indicate that the features of the different classes were clearly distinguishable in the latent space. All the experiments show that the proposed consistent mixup regularization enables deep learning models to learn from meaningful representations.

Table 1. Comparison of the performance between our proposed method and existing UDA methods

Methods \ Tasks	M→MM	M→U	S→M
Pixel-Level Adapt [8]	98%	95%	---
CyCADA [5]	----	93%	89%
MCD [9]	----	93%	95%
ADDA without CMR	86%	92%	77%
Our proposed method	95%	94%	85%

B. Training Procedure of the Improved ADDA Method

The training algorithm of our improved ADDA approach, which uses the Adam optimizer, is presented as pseudocode in Algorithm 1. In every iteration, samples from the source domain are first mixed at the pixel level and fed into the feature extractor to obtain an embedding, which is then utilized by the classifier to predict the source label. The second step involves mixing the target samples at the pixel level and feeding them into the feature extractor to obtain the embedding. Because label information is unavailable, we calculate the L1-Norm as a penalty term. The CMR losses are calculated using Eqs. (3) and (4) for the source and target domains, respectively. In addition, the discriminator network gradients are updated using the $\boldsymbol{L_{adv}^{D}}$ objective, whereas the $\boldsymbol{L_{total}}$ objective updates the feature extractor and classifier network gradients. To validate the effectiveness and generality of the improved ADDA approach using CMR, we conducted experiments under three UDA scenarios: M→MM, M→U, and S→M.

C. Datasets

Three adaptation scenarios were considered: M→U, S→M, and M→MM. The MNIST dataset contains 60,000 handwritten digit images. SVHN contains 73,257 examples of digits and numbers in natural settings. and USPS contains 7,291 images. MNIST-M contains 60,000 MNIST images with color patches. Furthermore, the number of input channels was set to three for the experiments on the S→M and M→MM adaptation tasks. In contrast, the number of channels was set to 1 for the M→U adaptation task only. We evaluated the pretrained model on the target test datasets of MNIST, USPS, and MNIST-M at the end of the training phase.

Fig. 2. Examples from the MNIST, USPS, SVHN, and MNIST-M datasets.

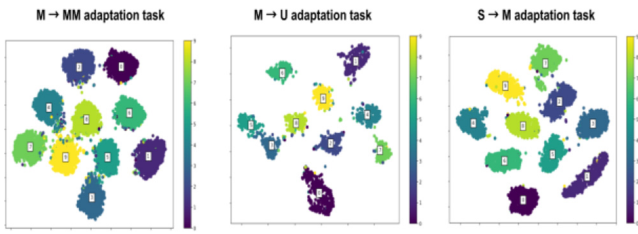


Fig. 3. T-SNE visualization of the feature distribution of target domain on the M→MM, M→U, and S→M adaptation tasks.

Algorithm 1. Training procedure of our proposed method

1) Inputs: Source domain: $\boldsymbol{D_{src}}$, target domain: $\boldsymbol{D_{tgt}}$, batch size: N, and training iterations: K.
2) Initialize α, $\boldsymbol{\mu_1}$, and $\boldsymbol{\mu_2}$ hyper-parameters.
3) for j in 1: K do
4) $(\boldsymbol{x_{src}}, \boldsymbol{y_{src}}) \leftarrow$ SAMPLING ($\boldsymbol{D_{src}}$, N) #random
5) $(\boldsymbol{x_{tgt}}) \leftarrow$ SAMPLING ($\boldsymbol{D_{tgt}}$, N) #random
6) λ parameter ← SAMPLING (Beta (α, α)) #random
7) $(\boldsymbol{x_{src}}, \boldsymbol{y_{src}}) \leftarrow$ Eq. (1) #return mixed images and soft labels.
8) $(\boldsymbol{x_{tgt}}) \leftarrow$ Eq. (1) #return only mixed images.
9) Compute $\boldsymbol{L^{D}_{adv}}$ adversarial loss; update D network by ascending along gradients $\nabla \boldsymbol{L^{D}_{adv}}$.
10) Compute $\boldsymbol{L_{total}}$; update F, C networks by descending along gradients $\nabla \boldsymbol{L_{total}}$.
11) end for

V. CONCLUSIONS

The big-data environment in the era of the "fourth Industrial Revolution" is changing significantly. Well-annotated datasets are required to improve deep-learning models. UDA learns domain-invariant features using adversarial learning. In this paper, we propose an innovative learning mechanism for the ADDA method. By generating neighboring training examples, mixup regularization can regularize the output distribution. It penalizes per-pixel inconsistent predictions between neighboring and training examples. This learning mechanism also considers the categorical distribution of the target domain during the training phase. Specifically, deep neural network training was stabilized after plugging the CMR into the ADDA. The T-SNE plots also indicated that the models could learn from distinctive representations. According to our comparison results, our proposed method can compete with existing UDA methods.

ACKNOWLEDGMENTS

This study was supported by a Daejeon University Research Grant (2021).

REFERENCES

[1] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 2962-2971, 2017. DOI: 10.1109/CVPR.2017.316.

[2] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, A. Courville, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, California, pp. 6438-6447, 2019. DOI: 1806.05236/arXiv.1806.05236.

[3] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *6th International Conference on Learning Representations*, Vancouver, Canada, pp. 1-13, 2018. DOI: 10.48550/arXiv.1710.09412.

[4] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, Macao, China, pp. 3635-3641, 2019. DOI: 10.24963/ijcai.2019/504.

[5] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 1989-1998, 2018. DOI: 10.48550/arXiv.1711.03213.

[6] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *32nd AAAI Conference on Artificial Intelligence*, New Orleans, USA, pp. 3934–3941, 2018. DOI: 10.48550/arXiv.1809.02176.

[7] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 8503-8512, 2018. DOI: 10.1109/CVPR.2018.00887.

[8] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 95-104, 2017. DOI: 10.1109/CVPR.2017.18.

[9] K. Saito, K. Watanabe, Y. Ushiku and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 3723–3732, 2018. DOI: 10.1109/CVPR.2018.00392.

[10] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in neural information processing systems 32 (NeurIPS 2019)*, Vancouver, Canada, pp. 5049-5059, 2019. DOI: 10.48550/arXiv.1905.02249.

[11] J. Na and W. Hwang, "Deep learning based domain adaptation: A survey," *2022 Korean Institute of Broadcast and Media Engineers*, vol. 27, no. 4, pp. 511-518, Jul. 2022. DOI: 10.5909/JBE.2022.27.4.511.

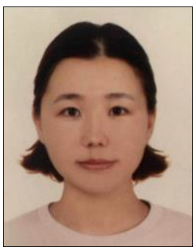
[12] K. T. Kim and J. Y. Choi, "Development of semi-supervised deep domain adaptation based face recognition using only a single training sample," *Journal of Korea Multimedia Society*, vol. 25, no. 10, pp. 1375-1385, Oct. 2022. DOI: 10.9717/kmms.2022.25.10.1375.

[13] L. Chen, H. Chen, Z. Wei, X. Jin, X. Tan, Y. Jin, and E. Chen, "Reusing the task-specific classifier as a discriminator: Discriminator-

free adversarial domain adaptation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, USA, pp. 7171-7180, 2022. DOI:10.1109/CVPR52688.2022.00704.

[14] M. Chen, S. Zhao, H. Lui, and D. Cai, "Adversarial-learned loss for domain adaptation," in *34th AAAI Conference on Artificial Intelligence*, New York, USA, pp. 3521-3528, 2020. DOI: 10.1609/aaai.v34i04.5757.

[15] T-D Troung, R.T.V. Chappa, X-B. Nguyen, N. Le, A.P.G. Dowling, and K. Luu, "OTAdapt: Optimal transport-based approach for unsupervised domain adaptation," in *2022 26th International Conference on Pattern Recognition*, Montreal, Canada, pp. 2850-2856, 2022. DOI: 10.1109/ICPR56361.2022.9956335.



Bayarchimeg Kalina

Bayarchimeg Kalina completed her bachelor's degree in computer science from the Information and Network Security Department of Mongolian University of Science and Technology. She is now a master's student in the Computer and Information Security Department of Daejeon University. Currently, she is interested in deep learning, meta-learning, and natural language processing.

Young-Bok Cho

Young-Bok Cho is an Associate Professor at the Department of Computer & Information Security, Daejeon University, Daejeon, South Korea. She received her M.S. and Ph.D. degrees in Computer Science from Chungbuk National University, in 2006 and 2012, respectively. She has been involved in national research and has been an expert adviser of SME in Korea to follow-up research projects and an evaluator of Korea R&D, a national research proposal. She is currently working in the department of information security, Daejeon University. Her research interests include network security, medical information security, and medical image processing.