IEIE Transactions on Smart Processing and Computing, vol. 12, no. 1, February 2023
https://doi.org/10.5573/IEIESPC.2023.12.1.48

Discrete Cosine Transformed Images Are Easy to Recognize in Vision Transformers

Jongho Lee and Hyun Kim

Department of Electrical and Information Engineering, Research Center for Electrical and Information Technology, Seoul National University of Science and Technology / Seoul, Korea {jhlees, hyunkim}@seoultech.ac.kr

* Corresponding Author: Hyun Kim

Received December 6, 2022; Accepted December 23, 2022; Published February 28, 2023

** Regular Paper*

***Abstract*:** Deep learning models for image classification with adequate parameters show excellent classification performance because they can effectively extract the features of input images. On the other hand, there is a limit to the abilities of deep learning models to interpret images using only spatial information because an image is a signal with great spatial redundancy. Therefore, in this study, the discrete cosine transform was applied to an input image in units of an N×N block size to allow the deep learning model to employ both frequency and spatial information. The proposed method was implemented and verified by selecting a vision transformer using a 16×16 non-overlapping patch as a baseline and training various datasets of Cifar-10, Cifar-100, and Tiny-ImageNet from the very beginning without pre-trained weights. The experimental results showed that the top-1 accuracy is improved by approximately 3-5% for every dataset with little increase in computational cost.

***Keywords*:** Computer vision, Image classification, Deep learning, Discrete cosine transform (DCT), Vision transformer

1. Introduction

Recently, owing to developments in deep learning (DL), there have been remarkable performance improvements in the field of computer vision [1-4, 26-29]. Until now, most DL-based computer vision studies have been developed based mainly on model architectures [5, 6] and computational methods, such as convolution and self-attention [7, 8]. In the 2020s, self-attention-based vision transformers have tended to replace convolutional neural networks (CNNs) [1, 3, 5, 9]. The transformer model, which has been actively studied in the field of natural language processing (NLP), allows one image patch to act as a word in a sentence through patch embedding. This enables self-attention operations in the field of computer vision.

However, unlike a single word of great importance in a language, an image is only a signal of light. Thus, it has redundant information relative to the importance of words in sentences [4]. Therefore, if important information is extracted in advance from the image, it may help improve the accuracy of the DL model in computer vision. Frequency domain transform methods, such as the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Fast Fourier Transform (FFT), have been steadily used for extracting meaningful information from images. In modern DL models, it is also possible to use these frequency domain transform methods for this purpose [10-12] because computational methods for image recreation can achieve good performance with the DCT, DWT, and FFT, as well as with convolution and self-attention [13]. Several attempts to use DCT in DL models have been reported [14-16]. On the other hand, these studies were only performed to reduce the communication bandwidth and computational costs in CNN or NLP models.

This paper proposes a method to improve the accuracy of the vision transformer model using the DCT. In detail, an input image enters the vision transformer after applying a 2D DCT [10] in units with an N×N block size, allowing the DL models to utilize inputs with both spatial and frequency information. The proposed method improves the top-1 accuracy of the vision transformer by approximately 3-5% on the Cifar10 [17], Cifar-100 [17], and Tiny-ImageNet [18] datasets, while performing the 2D-DCT only once, immediately before patch embedding. In

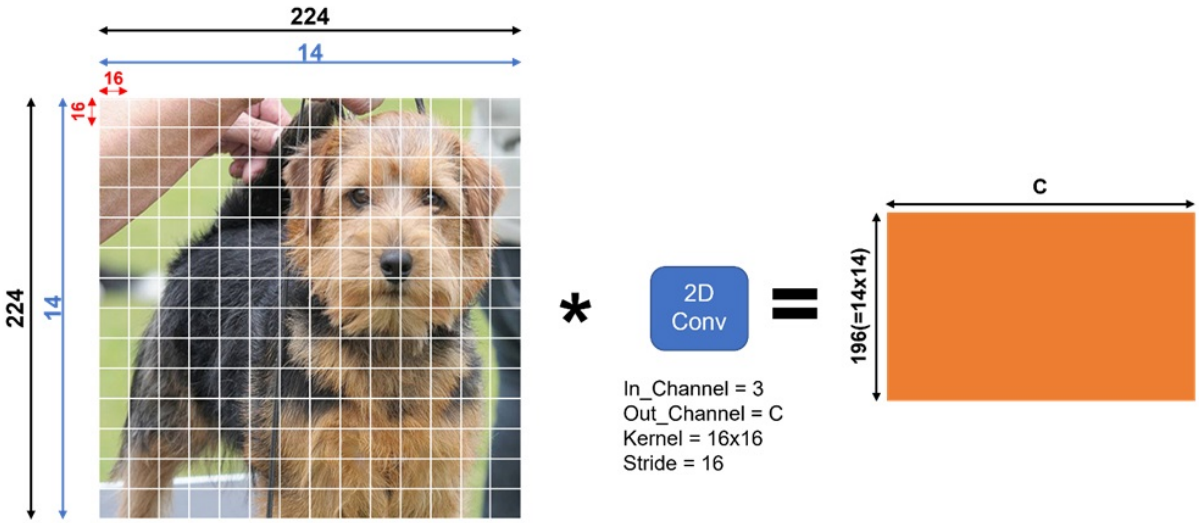

Fig. 1. Diagram showing the patch embedding process of a sample image of size 224×224 extracted from imageNet-1k dataset. The RGB image is cut into 196(=14×14) 16×16 size patches, and is released in the form of a matrix of 196×C size through a non-overlapping 2D convolution operation of the 16×16 size filter.

addition, as the proposed method can improve the performance of various vision transformer models [1], including tiny and small sizes, it has high compatibility and scalability for model sizes and datasets.

2. Background

2.1 Patch Embedding

Before the emergence of vision transformers [1] by Dosovitsky et al., CNN models [6, 19] were used widely in computer vision. Subsequently, the emergence of vision transformer DL models based solely on self-attention operations has become dominant, excluding the convolution structure [1, 3, 5, 9]. The concept of self-attention in computer vision is similar to self-attention in the field of NLP; however, it is possible to change the concept of image-patch in vision transformer models to that of sentence-word in NLP through the patch embedding process [1], as shown in Fig. 1. This concept is simple. It cuts the image into a non-overlapping 16×16 patch for linear projection and adds a class token. Through these ideas, vision transformers can achieve state-of-the-art performance not only in NLP but also in computer vision.

On the other hand, despite the contributions of patch embedding, most studies on computer vision tasks have focused on improving model architectures and computational methods, such as convolution, multi-layer perception, and self-attention [5-8, 30]. Accordingly, these studies cannot overcome the limitations of using only the spatial information of the image. As shown in Fig. 1, when patch embedding is performed, the patches cut into 16×16 pixels are converted to a 196×C-dimensional matrix through a 2D-convolution operation. Owing to the nature of the CNN, the weights of the filters applied to each patch are shared. Therefore, it may be helpful to match the uniform format for each patch rather than to use the original image of the pixel.

2.2 2D-Discrete Cosine Transform

The 2D-DCT is used widely in signal processing, and various fields, including image compression, such as JPEGs [20]. One of the advantages of 2D-DCT is that the image can be viewed from a frequency perspective. When the 2D-DCT (in units of N×N block size) is performed on the image, the upper-left side of the block has low-frequency information, whereas the lower-right side has high-frequency information. Fig. 2 presents the result from obtaining an image with frequency information for each block by performing the N×N block 2D-DCT on the RGB image with a resolution of 224×224 in Fig. 1.

An image is a signal in which spatial redundant information is captured. Therefore, if the 2D-DCT is performed on a block basis, the frequency and spatial information can be expressed in the local and global parts, respectively. The motivation of this study is that by exploiting these advantages, the vision transformer models without inductive bias can be better trained by inputting images with the 2D-DCT applied to vision transformer models. The 2D-DCT with the N×N size can be calculated as follows:

$$D_{i,j} = \frac{1}{\sqrt{2N}} \alpha(i)\alpha(j) \sum_{x=0}^{N} \sum_{y=0}^{N} I_{x,y} \cos\left[\frac{(2x!+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right], \quad (1)$$

where D represents one N×N-sized block. For example, the original image of a 224×224 resolution may be converted to 3136 4×4 size blocks, 784 8×8 size blocks, 196 16×16 size blocks, or one 224×224 block. In (1), i and j are the pixel indices of the blocks converted by the 2D-DCT, and x and y are the pixel indices of the original block I. α is a scale factor for ensuring that the transform is orthonormal and is given as follows:

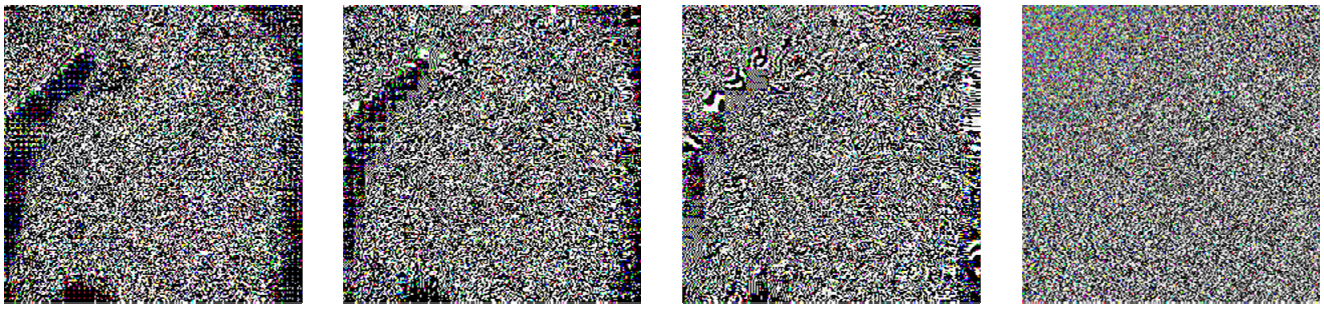

Fig. 2. Results of a 2D-discrete cosine transform (DCT) operation for the 224×224 size input image of Fig. 1 in block units of (a) 4×4; (b) 8×8; (c) 16×16; (d) 224×224.

$$\alpha(u)=\begin{cases}\frac{1}{\sqrt{2}} & if\ u=0\\ 1 & otherwise\end{cases}, \tag{2}$$

In this way, the original image can be converted to a frequency domain on a block basis, and within the block, the upper-left part can concentrate low-frequency energy. The lower right can have the relatively less important high-frequency energy.

2.3 Vision Transformer

Fig. 3(a) shows the overall structure of the vision transformer used for image classification. First, the input image resized to a 224×224 resolution was cut into 196 16×16 size patches through a patch embedding process. Subsequently, each patch was supplemented with position information through position embedding and class information by adding a class token. Because the multi-head attention [21]-based transformer encoder has the same input and output dimensions, it can go through several blocks depending on the size of the models (e.g., tiny and small). Multi-head attention is a method of self-attention in parallel by dividing the query, key, and value input by the number of heads. Through this method, the vision transformer identifies the relationships between patches and extracts the image features. Finally, an image classification task is performed through a multi-layer perceptron head after the previous processes.

3. Proposed Method

While the vision transformer model receives an original image in RGB format, this study proposes adding a new 2D-DCT patch embedding method at the input stage of the vision transformer model. In the proposed method, the original image was cut into N×N blocks, and the 2D-DCT is performed in units of blocks before patch embedding, making the local spatial information available within each block, as shown in Fig. 3. The location information is more critical in the 2D-DCT block than in the original image because the upper-left part of the 2D-DCT block has DC information representing the average pixel value. In contrast, the bottom-right part has high-frequency AC information. The proposed method improves performance by utilizing this additional information and has the advantage that it can be implemented with a minimal computational increase to the baseline (i.e., vision transformer) without changing the number of parameters. In addition, it can be applied to various vision transformer models in various structures (i.e., high compatibility and scalability).

This paper describes applying the proposed method through an example of a specific block size. When the 2D-DCT block size is 4×4, the original image of size 224×224 is divided by 3136 (= (224×224) / (4×4)) blocks and a 4×4 2D-DCT is then performed in parallel for each block. The 3136 blocks on which the 4×4 2D-DCT is performed are again combined into images of size 224×224. In this case, an additional 1.5M floating-point operations per second (FLOPs) is required for the 2D-DCT operation, compared to the case where the vision transformer's inference operation is performed on the RGB image with a resolution of 224×224. On the other hand, this is an insignificant increase considering that the computational amount of DeiT-Tiny and DeiT-Small is 1.3G FLOPs and 4.6G FLOPs [22], respectively. Even if the 2D-DCT block size increases to 8×8, only approximately 2.7M FLOPs are required (i.e., when performing the same process with the 784 8×8 blocks). In addition, because there is no dependency between each block, a fast parallel operation is possible through "CUDA" [23]. The subsequent processing method follows the operation process of the existing vision transformer described in Section 2.3. In other words, the proposed method is a straightforward but effective method that performs N×N block DCT on the input image without changing the structure of the existing vision transformer model and then inputs the DCT blocks to patch embedding.

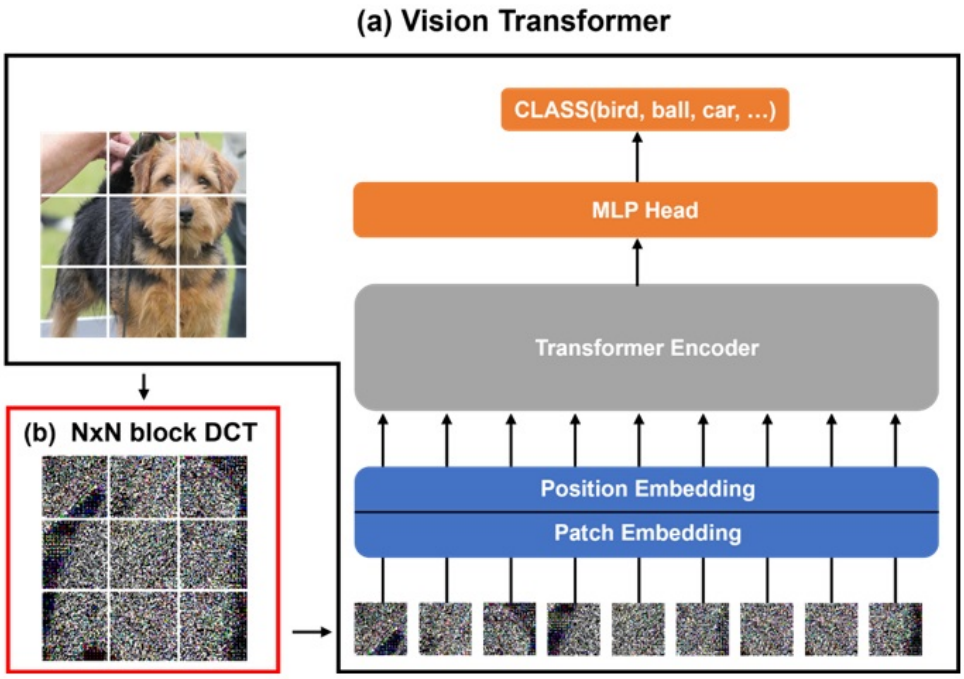

Fig. 3. Diagram showing the patch embedding process of a sample image of size 224×224 extracted from imageNet-1k dataset. The RGB image was cut into 196(=14×14) 16×16 size patches, and was released in the form of a matrix of 196×C size through a non-overlapping 2D convolution operation of the 16×16 size filter.

4. Experimental Results

4.1 Experiment Settings

A vision transformer was trained and validated in four Tesla-V100 GPUs using the Cifar-10 [17], Cifar-100 [17], and Tiny-ImageNet [18] datasets. The Cifar-10 and Cifar-100 datasets have 50,000 training images and 10,000 validation images with 10 and 100 classes, respectively. The Tiny-ImageNet dataset contains 100,000 training images and approximately 10,000 validation images for 200 classes. The most training strategy of DeiT [3] and the detailed environmental settings are as follows. Adam [31] was used as an optimizer, and set the momentum to 0.9 and weight decay to 0.05. All models were trained for 300 epochs using a batch size of 1,024 and a learning rate of 0.0005. All source codes are referred to the pytorch-based pytorch image models (Timm) library [24], and for the 2D-DCT operations, the torchJpeg library [25] is used. It should be noted that the DCT block size in all result tables is marked as "-" for vanilla vision transformers that do not use the DCT patch embedding.

4.2 Accuracy Evaluation

Table 1 lists the results of applying the proposed method to the vision transformer with the tiny and small models on the Cifar-10 dataset. As suggested, the model was trained by adding discrete cosine transformed images in units of N×N blocks before performing patch embedding for the vision transformer. When a 4×4 block size 2D-DCT was adopted on the Cifar-10 dataset, the top-1 accuracy increased by 4.09% and 0.36% for the tiny and small models, respectively. When a 16×16 block size 2D-DCT was adopted, the top-1 accuracy improved by 4.57% and 3.7% for the tiny and small models, respectively. When the block size was more than 16×16, the performance was inferior to the others because the detailed information was lost spatially. Therefore, experiments larger than 32×32 were not performed. In particular, when 2D-DCT was performed with an image size of 224×224, all spatial features of the image were lost because of the characteristics of 2D-DCT, as shown in Fig. 2(d).

Table 2 shows the same experiment for the Cifar-100 dataset with a tiny model and small model. When the 4×4 block size 2D-DCT was adopted, the top-1 accuracy

Table 1. Accuracy of the Proposed Method on the CIFAR-10.

Model	DCT block size	Top1-Acc.(%)	Top5-Acc.(%)
DeiT -Tiny	-	79.92	98.8
	2	84.42	99.18
	4	84.01	99.2
	8	84.27	99.21
	16	84.49	99.26
	32	82.7	99.21
DeiT -Small	-	79.73	98.76
	2	75.96	98.42
	4	80.09	98.73
	8	80.67	99.0
	16	83.43	99.07
	32	53.82	93.68

Table 2. Accuracy of the Proposed Method on the CIFAR-100.

Model	DCT block size	Top1-Acc.(%)	Top5-Acc.(%)
DeiT -Tiny	-	66.17	89.79
	2	68.95	91.36
	4	70.03	91.55
	8	69.64	91.34
	16	71.53	92.07
	32	67.15	90.35
DeiT -small	-	61.13	86.2
	2	61.19	85.34
	4	64.72	88.12
	8	66.31	88.59
	16	70.05	90.75
	32	38.47	38.76

Table 3. Accuracy of the Proposed Method on the Tiny-ImageNet.

Model	DCT block size	Top1-Acc.(%)	Top5-Acc.(%)
DeiT -Tiny	-	54.22	77.84
	2	57.24	79.88
	4	57.14	79.94
	8	56.81	80.17
	16	57.88	80.62
	32	52.45	76.75
DeiT -Small	-	48.78	73.7
	2	53.26	76.28
	4	54.15	76.94
	8	52.75	76.88
	16	54.27	77.66
	32	28.42	53.66

increased by 3.86% and 3.59% for the tiny and small models, respectively. When the 16×16 block size 2D-DCT was adopted, the top-1 accuracy increased by 5.36% and 8.92% for the tiny and small models, respectively.

Experiments were conducted on the Tiny-ImageNet dataset, and a relatively large dataset was used, as shown in Table 3. When the 4×4 block size 2D-DCT was adopted, the top-1 accuracy increased by 2.92% and 5.37% for the tiny and small models, respectively. When the 16×16 block size 2D-DCT was adopted, it increased by 3.66% and 5.49% for the tiny and small models, respectively.

As a result, when the 16×16 block size 2D-DCT patch embedding was applied, performance was increased most in all cases through the proposed method. This result was attributed to the patch being cut into 16x16 during the patch embedding process in DeiT. The increase in computational cost and decrease in speed was negligible. In addition, as the proposed model can be applied directly to most vision transformer models using patch embedding, its compatibility was excellent, making it an easy and general way to improve performance.

5. Conclusion

Thus far, modern DL models perform well because they can independently extract and process the information needed in the image. On the other hand, in this study, because an image is simply a light signal, the DL model can help better process an image by utilizing a modulation method for the traditionally studied frequency band. The proposed method shows remarkable performance improvements in all experimental cases, even though the computational cost and latency remain relatively unchanged. The proposed method can be applied directly to other transformer-affiliated models and can be extended to tasks such as object detection, instance segmentation, semantic segmentation, and depth estimation

Acknowledgments

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2023-RS-2022-00156295) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

References

[1] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” arXiv preprint arXiv:2010.11929, 2020. Article (CrossRef Link)

[2] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “Coatnet: Marrying convolution and attention for all data sizes,” in Proc. Advances in Neural Information Processing Systems, 2021, vol. 34, pp. 3965-3977. Article (CrossRef Link)

[3] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in Proceedings of the International Conference on Machine Learning, 2021, pp. 10347-10357. Article (CrossRef Link)

[4] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” arXiv preprint arXiv:2111.06377, 2021. Article (CrossRef Link)

[5] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012-10022. Article (CrossRef Link)

[6] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, “Vision Transformer with Deformable Attention,” arXiv preprint arXiv:2201.00520, 2022. Article (CrossRef Link)

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961-2969. Article (CrossRef Link)

[8] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
[9] Z. Liu et al., "Swin Transformer V2: Scaling Up Capacity and Resolution," arXiv preprint arXiv:2111.09883, 2021. Article (CrossRef Link)
[10] N. Ahmed and K. R. Natarajan T_ and Rao, "Discrete cosine transform," IEEE Transactions on Computers, vol. 100, no. 1, pp. 90-93, 1974. Article (CrossRef Link)
[11] J. Shin and H. Kim, "RL-SPIHT: Reinforcement Learning based Adaptive Selection of Compression Ratio for 1-D SPIHT Algorithm," IEEE Access, vol. 9, pp. 82485-82496, 2021. Article (CrossRef Link)
[12] H. Kim, A. No, and H.-J. Lee, "SPIHT Algorithm with Adaptive Selection of Compression Ratio Depending on DWT Coefficients," IEEE Transactions on Multimedia, vol. 20, no. 12, pp. 3200-3211, Dec. 2018. Article (CrossRef Link)
[13] Y. Rao, W. Zhao, Z. Zhu, J. Lu, and J. Zhou, "Global filter networks for image classification," in Proceedings of the Advances in Neural Information Processing Systems, 2021, vol. 34. Article (CrossRef Link)
[14] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the Frequency Domain," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020, pp. 1740-1749. Article (CrossRef Link)
[15] X. Shen et al., "DCT-Mask: Discrete Cosine Transform Mask Representation for Instance Segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2021, pp. 8720-8729. Article (CrossRef Link)
[16] C. Scribano, G. Franchini, M. Prato, and M. Bertogna, "DCT-Former: Efficient Self-Attention with Discrete Cosine Transform," arXiv preprint arXiv:2203.01178, 2022. Article (CrossRef Link)
[17] A. Krizhevsky, G. Hinton, and others, "Learning multiple layers of features from tiny images," 2009.
[18] Y. Le and X. S. Yang, "Tiny ImageNet Visual Recognition Challenge," 2015.
[19] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving," in Proc. IEEE/CVF Int. Conf. Computer Vision, 2019, pp. 502-511. Article (CrossRef Link)
[20] G. K. Wallace, "The JPEG still picture compression standard," IEEE Transactions on Consumer Electronics, vol. 38, no. 1, pp. xviii-xxxiv, 1992. Article (CrossRef Link)
[21] A. Vaswani et al., "Attention is all you need," in Proc. Advances in Neural Information Processing Systems, 2017, vol. 30. Article (CrossRef Link)
[22] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," ACM Computing Surveys (CSUR), 2021. Article (CrossRef Link)
[23] NVIDIA, P. Vingelmann, and F. H. P. Fitzek, CUDA, release: 10.2.89. 2020. [Online]. Available: Article (CrossRef Link)
[24] R. Wightman, PyTorch Image Models. GitHub, 2019. doi: 10.5281/zenodo.4414861. Article (CrossRef Link)
[25] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, "Quantization Guided JPEG Artifact Correction," 2020. Article (CrossRef Link)
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778. Article (CrossRef Link)
[27] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in Proceedings of the International conference on machine learning. PMLR, 2019, pp. 6105-6114. Article (CrossRef Link)
[28] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in Proceedings of the International Conference on Machine Learning. PMLR, 2021, pp. 10 096-10 106. Article (CrossRef Link)
[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," vol. 28, 2015. Article (CrossRef Link)
[30] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in Proc. IEEE/CVF Int. Conf. Computer Vision, 2017, pp. 764-773 Article (CrossRef Link)
[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017. Article (CrossRef Link)

Jongho Lee received his B.S. degree in Electrical and Information Engineering from Seoul National University of Science and Technology, Seoul, Korea, in 2020. Currently, he is a graduate student of Seoul National University of Science and Technology, Seoul, Korea. In 2020, he was a research student at the Korea Institute of Science and Technology (KIST), Seoul, Korea. In 2022, he was a visit student at the University of Wisconsin-Madison, Wisconsin, USA. His research interests are the deep learning and machine learning algorithms for computer vision tasks.

Hyun Kim received his B.S., M.S. and Ph.D. degrees in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 2009, 2011 and 2015, respectively. From 2015 to 2018, he was with the BK21 Creative Research Engineer Development for IT, Seoul National University, Seoul, Korea, as a BK Assistant Professor. In 2018, he joined the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul, Korea, where he is currently working as an Associate Professor. His research interests are the areas of algorithms, computer architecture, memory, and SoC design for low-complexity multimedia applications and deep neural networks.