Special Paper
방송공학회논문지 제26권 제7호, 2021년 12월 (JBE Vol. 26, No. 7, December 2021)
https://doi.org/10.5909/JBE.2020.26.7.868
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

An Analysis on the Properties of Features against Various Distortions in Deep Neural Networks

Jung Heum Kang[a)], Hye Won Jeong[a)], Chang Kyun Choi[a)], Muhammad Salman Ali[a)], Sung-Ho Bae[a)‡] , and Hui Yong Kim[a)‡]

Abstract

Deploying deep neural network model training performs remarkable performance in the fields of Object detection and Instance segmentation. To train these models, features are first extracted from the input image using a backbone network. The extracted features can be reused by various tasks. Research has been actively conducted to serve various tasks by using these learned features. In this process, standardization discussions about encoding, decoding, and transmission methods are proceeding actively. In this scenario, it is necessary to analyze the response characteristics of features against various distortions that may occur in the data transmission or data compression process. In this paper, experiment was conducted to inject various distortions into the feature in the object recognition task. And analyze the mAP (mean Average Precision) metric between the predicted value output from the neural network and the target value as the intensity of various distortions was increased. Experiments have shown that features are more robust to distortion than images. And this points out that using the feature as transmission means can prevent the loss of information against the various distortions during data transmission and compression process.

Keyword: Convolutional neural networks, Feature compression, Data transmission, Image restoration, Instance segmentation

Ⅰ. Introduction

Features in Deep Neural Networks (DNN) refer to regional properties and attributes of condensed information after an input image passed through the DNN which consists of hierarchical convolution layers. In complex vision tasks, such as Object Detection and Instance Segmentation, there is a separate module to serve feature extraction. And this module is also called as the backbone network. Most of the well-known complex vision task DNN networks such as Faster-RCNN[1], RetinaNet[2], Mask-RCNN[3] use

a) Department of Computer Science and Engineering, Kyung Hee Univ.
‡ Corresponding Author : Sung-Ho Bae
E-mail: shbae@khu.ac.kr
Tel: +82-31-201-2593
ORCID: https://orcid.org/0000-0002-3389-1159
‡ Corresponding Author : Hui Yong Kim
E-mail: hykim.v@khu.ac.kr
Tel: +82-31-201-3760
ORCID: https://orcid.org/0000-0001-7308-133X
※ This work was supported by Institute of Information Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00011, Video Coding for Machine).
※ This research was a result of a study on the NIPA.
・Manuscript received October 25, 2021; Revised November 29, 2021; Accepted November 29, 2021.

the same ResNet[4] FPN (Feature Pyramid Network)[5] architecture as a backbone network. Normally, we only regarded the Image as compression targets. However, using common backbone architecture points out that we can also treat the feature map as a means of transmission like image. Transmitting features has some advantages. One is as resources in mobile devices become more powerful and energy-efficient, doing feature extraction calculations on front-end mobile can achieve computational power offloading with reducing energy consumption in the back-end data center.[6] The other one is that it can prevent privacy issues by dealing with non-visual information on the information stolen situation.[7] As we treat feature data as transmission target, we should compress the feature before we transmit to the back end cloud. Based on these concerns, discussions on standardization about feature compression is now progressing.[8] Before we consider the method, we want to explore the feature response properties against various distortions so that we can make a better decision to research direction. We perform extensive experiments of comparing the performance of instance segmentation task, in the case of adding distortions to the Image and Feature map. We use five distortion algorithms with changing a noise intensity. These noises contain quantization distortion, blur distortion, noise distortions, and SR (Super Resolution) distortions, and coding distortion. Our experimental results prove that distortions applied in the feature domain show better performance on the vision task evaluation than the distortions applied in the image domain. We also analyzed the reason why the distortion applied in feature domain showed a better performance than distortion applied in image domain in terms of DNN approach.

II. Related works

As deep learning research advanced, there is a big performance gain on advanced computer vision task such as object detection and instance segmentation. And as the amount of video data collected in various application environments such as CCTV, IOT(Internet of Things) devices, smart devices and autonomous vehicles are increasing exponentially, the demand for automation without human supervision requirements are gradually increasing. Since there is the limitation to human monitoring vast amounts of data. To do the transmission process, compressing the transmission data to reduce the resource and fastening transmission time by decreasing the bits of the data stream is a normal way. Therefore, in order to efficiently compress video data while maintaining the task evaluation performance on the machine, video compression technique which focused on the performance of the task evaluation network is required. So, Video coding for Machine AHG was formed in July 2019 in ISO/IEC JTC1 SC29 WG11 (MPEG) with the purpose of establishing technical standards for compressed bit-streams of video or features extracted from multiple tasks while maintaining the machine vision performance. There is two benchmark task to evaluate, one is object detection task, and other is Segmentation task. Object detection task aims to precisely detect the target object in the single image. And Instance segmentation is a task to detect every different instance in the image even these instances are included in the same class. It is derived from object detection task but considered much more complex problem, since its goal is to recognize the instance separately and mark the edge elaborately. RCNN[9] is the milestone paper of using deep learning on object detection task. They firstly try to use DNN models to classify whether each proposal box is a seeking object or not. And it highly exceeds the previous State of the Art performance. However, in this early stage of research, it costs a lot of times to training each Network for each regional proposal. Fast-RCNN[10] propose to training the whole classifying model in once with sharing single convolution feature map. End-to-end training with multi-task loss training method helps to decrease the training time and inference time.

After that, Faster-RCNN[1] deploy DNN models to propose the region which is called RPN (Region Proposal Network) and previously done by selective search algorithm. It makes possible to train the whole task at once and achieve a better performance in shorter time. Mask-RCNN[3] is designed for instance segmentation task. It uses Faster-RCNN as base skeleton model, and additionally have a mask branch that predicts segmentation mask. All of the explained models commonly use same backbone network architecture like ResNet[4] or MobileNet[11] to extract the feature.

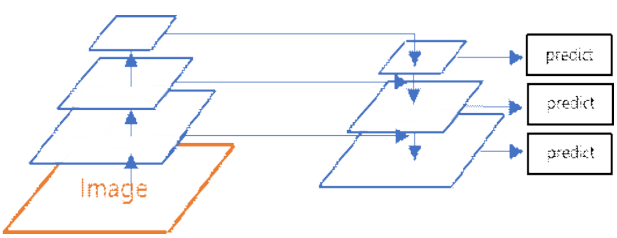


Fig. 1. The architecture of Feature Pyramid Network for detection tasks

Fig.1 is architecture of FPN[5] which performs as fast as extracting single feature map but more accurate. It deploys a top-down DNN architecture with lateral connection to build high level semantic feature maps at all scales. It is commonly used as a generic feature extractor in several applications.

Ⅲ. Method

In this section, we introduce 5 methods to generate the distortion and noise. We try to generate the noise that is naturally and frequently made in the real world. And we generate the distortions which were commonly used methods in data processing domains such as quantization, resizing, and coding noise when data pass through compressor. To generate the feature, we use a pretrained backbone which is based on ResNet-50[4] Feature Pyramid Network[5]. We use coco2017 image dataset[12] and mentioned pretrained model to generate p2-p6 type features. And this extracted feature was used as the target of distortions. And the quality of the distorted feature and distorted image is evaluated with Mask-RCNN model in the instance segmentation task with mAP measurement. Condition of measuring the mAP is as following:

Average Precision (AP) @[IoU = 0.50:0.95 | area = all | maxDets = 100])

1. Gaussian Blur

Gaussian blur is one of the most frequently used to get the blur image. It generate the random values which follows the probabilities of gaussian distribution. It can be simply obtained by convolution with gaussian kernel filter. In gaussian blur generation, σ decides the level of distortion. In two dimensional image, it is the product of two gaussian distribution functions in terms of x axis and y axis.

2. Gaussian Noise

Gaussian noise is often used to product the noisy image. It can be obtained by add the random noise made by gaussian kernel to original image. In gaussian noise generation, σ decides the level of noise.

3. Resizing Distortion

Resize of the image is the most common trick in the computer vision domain to exceed the performance of the DNN models. And SR is a promising image restoration task, to make the image seems more clearly in terms of human vision. However, as we are interested in making the spatial feature size small, we aim to make distorted LR(Low-Resolution) data. Which is the opposite direction of the SR task. Equation of making the LR image/feature map is as following.

$I_{LR} = ((I_{HR} \otimes k) \downarrow_s)$ → generate LR image

$I'_{LR} = ((I_{LR} \otimes k) \uparrow_s)$ → resize it to original size

In these equations, k denotes blur kernel, and $\downarrow_s$, $\uparrow_s$ represents scale factor. First term is the equation to make the LR image and second term is the equation to make LR image into original size.

4. Quantization Distortion

Quantization method is one of the most common method to reduce the memory and latency in deep learning research.[13] We use uniform quantization method that step size is equal in every interval. Equation of uniform quantization using in experiment is as following.

$$F_q = \partial\left(\frac{(F-min)\times(2^t-1)}{max-min}\right)\times\left(\frac{max-min}{2^t-1}\right)+min$$

In this term, t denotes the quantization bit-depth, and F_q denotes quantized feature value, and F denotes feature value.

5. Coding Noise Distortion

Fig.2. shows the process of adding a compression noise on image. We simply put the image into the compressor to get the coding noise. As you can see Fig. 3, Features are represented as floating point type (float32) with high dimension during the forward operation of deep learning

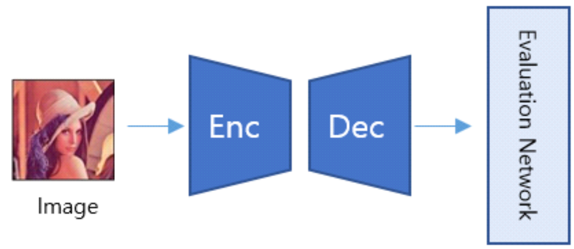


Fig. 2. Scenario of adding image compression noise

model. To put the feature to compressor, we firstly rearrange the 256 channel feature to 1 channel feature map.

Fig.4. shows the way to make a K channel feature into spatially rearranged 1 channel feature map. As feature is consist of 256 channels, we use K_w and K_h as 16. Secondly, we quantize the feature map to 8bit integer range since compressors such as VVC, VVenc, HEVC, have the limitation of accepting float type as input. After we get the feature from the decoder, we de-quantize values back to

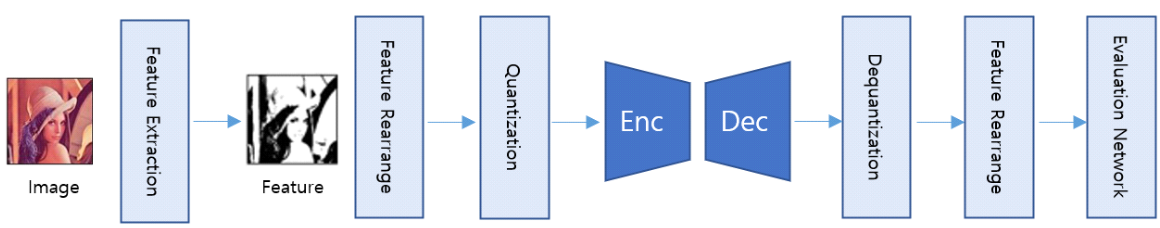


Fig. 3. Scenario of adding feature compression noise

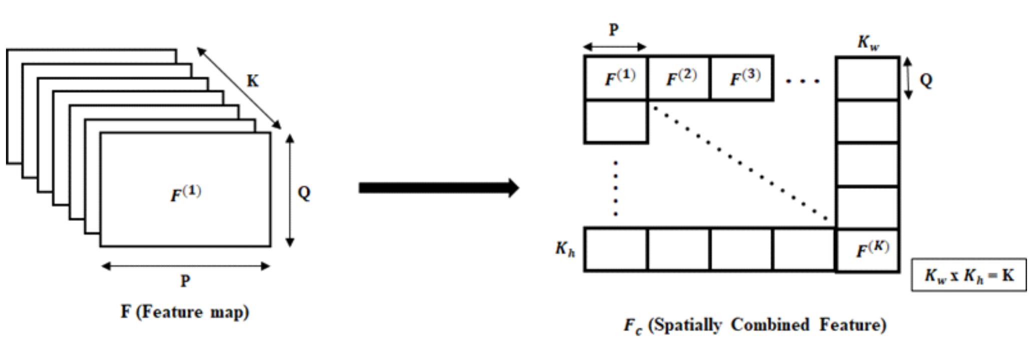


Fig. 4. Feature channel spatially rearrange

the original feature range and use the spatial feature rearrangement back to the original feature shape. And we put coding-noised distorted data to task evaluation network.

Ⅳ. Experimental Result & Analysis

In VCM feature compression standardization meeting, there is benchmark datasets used to check and compare the performance of the proposal method. It contains COCO image dataset[12], OpenImage[17] with many versions, and Tsinghua-Tencent 100k dataset. We select coco2017 image dataset[12] for the experiments as we deploy the pretrained Mask-RCNN[3] model to test the score. This model uses a ResNet-50[4] Feature Pyramid Network[5] architecture as backbone network to get p2-p6 features of the coco2017 image dataset[12]. After then, this model makes a region proposal with RPN and selects the best region proposal with proposal layer. After aligning the region with the ROI(Region of Interest) align layer, the feature goes separately to the classification branch, bbox regression branch, and mask branch. When every process has been done, rescale the generated mask with regarding the image size. The baseline accuracy for the Mask-RCNN on Instance segmentation with non-distorted coco2017 image dataset marks mAP 40.3.

1. Setup

With the methods we mentioned on section Ⅲ-5, we preprocess the extracted feature map to treat it like an image input (same value range, same channel, same distortion). First, as feature has many channels, we must rearrange the feature like normal image channel (3 for RGB image or 1 for black image) so that noise method can be applied the same way on the feature as it is applied on the image. Secondly, as range of feature is not the same as image, we must make the feature values into 0-255 range before we apply the compression distortions. We use the 8bit uniform quantization method as we explained above to get the quantized natural number range and we apply dequantization method to turn back into floating value before it goes into the evaluation network. In image restoration tasks such as SR and Denoising, we use the PSNR(Peak Signal- to-Noise Ratio) and SSIM(Structural Similarity Index Measure) as comparison metric to refer image quality.[14][15] We measure these metrics in feature domain (denotes SSIM/ PSNR[F]) and in image domain (denotes SSIM/ PSNR[I]) to compare distorted data with original one. Experiments which contain random factor like gaussian random noise are repeated 3 times and median value is reported.

2. Result on Gaussian blur

We set the sigma from 1 to 3. And we set the size of gaussian blur window as 21 and 5. Window size 5 experiment is held on feature domain since the size of the feature is smaller than the image at least 4 times more, and check whether this effects a lot on the performance. For more specific detail, the Image size is [800x1088] and biggest p2 feature size is [200x272], and smallest p6 feature size is [13x17]. So we check the feature result with a size 5 window to match the relative size considering the data size.

Table 1. The experimental results on distorted coco dataset with gaussian blur method

Condition	SSIM/PSNR[F]	SSIM/PSNR[I]	mAP
Feature(σ=1,ws=21)	0.545/26.23	-	34.8
Image(σ=1,ws=21)	0.690/32.38	0.572/28.35	35.7
Feature(σ=1,ws=5)	0.552/26.32	-	34.9
Feature(σ=1.5,ws=21)	0.372/23.78	-	29.7
Image(σ=1.5,ws=21)	0.578/29.47	0.375/25.87	32.7
Feature(σ=1.5,ws=5)	0.432/24.46	-	31.5
Feature(σ=3,ws=21)	0.142/20.42	-	15.4
Image(σ=3,ws=21)	0.356/25.16	0.176/22.82	18.9
Feature(σ=3,ws=5)	0.344/23.28	-	29.2

As shown in Table.1. we observe that Gaussian blur decrease more performance in feature domain. As we mentioned above, we use the FPN which makes feature size double smaller as feature pass through the FPN layer. So, we analyze that the feature's small spatial size makes Gaussian blur distortion in the feature affect more significantly. Result shows that in weak distortion(σ=1, σ=1.5) condition, distortion in feature domain still affect more significantly. However, in strong distortion(σ=3) condition, distortion in the image domain decreases more significantly compared with the similar relative window size.

3. Result on Gaussian Noise

As we mentioned before, we make feature value into 0-255 and then add the noise to make the distortion strength as same as possible.

Table 2. The experimental results on distorted coco dataset with gaussian noise method

Condition	SSIM/PSNR[F]	SSIM/PSNR[I]	mAP
Feature(σ=1)	0.942/46.47	-	39.2
Image(σ=1)	0.653/31.33	0.583/24.06	33.2
Feature(σ=3)	0.805/37.33	-	39.2
Image(σ=3)	0.416/26.11	0.317/17.35	20.1
Feature(σ=5)	0.675/33.12	-	39.0
Image(σ=5)	0.321/24.12	0.119/15.33	10.5

As shown in Table 2, Gaussian noise affect more in image domain. Result shows that feature is robust in random noise even SSIM and PSNR decrease like Image domain. It means that despite distorted feature and original target feature being different, the machine can well recognize the distorted feature like the original one.

4. Result on Resizing Distortion

We use the bicubic down-sample method to get the LR image and again use the bicubic up-sample to make the same size. We set the scale factor as 2 and 4, and 8 in this experiment.

As a result of Table 3, we observe that SR noise affects more in the Image domain. However, there is a big decrease in mAP performance in the feature domain either. We analyze that like Gaussian blur experiment, small spatial size of feature affects largely regarding SR noise. However, it maintains performance better than the distortion in the image domain.

Table 3. The experimental results on distorted coco dataset with resizing distortion method

Condition	SSIM/PSNR[F]	SSIM/PSNR[I]	mAP
Feature(RF=2)	0.755/31.97	-	36.4
Image(RF=2)	0.685/31.87	0.630/29.18	33.6
Feature(RF=4)	0.366/26.49	-	22.9
Image(RF=4)	0.389/25.40	0.282/23.32	18.1
Feature(RF=8)	0.151/23.95	-	5.8
Image(RF=8)	0.216/21.86	0.146/20.46	5.6

5. Result on Quantization Distortion

In this experiment, we set quantization bit depth as 8,4,2. And use uniform quantization which mentioned on above.

Table 4. The experimental results on distorted coco dataset with Quantization distortion method

Condition	SSIM/PSNR[F]	SSIM/PSNR[I]	mAP
Feature(Qbit=8)	0.997/58.92	-	39.3
Image(Qbit=8)	0.992/59.16	0.989/57.58	39.1
Feature(Qbit=4)	0.774/34.31	-	38.8
Image(Qbit=4)	0.800/36.08	0.729/37.27	37.8
Feature(Qbit=2)	0.211/18.91	-	37.1
Image(Qbit=2)	0.372/25.03	0.312/21.93	20.4

As shown in Table 4, we observed that quantize noise affect more in Image domain. There is no big decrease of mAP performance in feature domain. We analyze that even there is a big change in properties between original and distorted features such as mean square distance between values, and mean or covariance of feature values, the rank of the feature values is not being changed. So, we assume that preserve of feature rank helps to maintain the performance. We try to conduct the experiment about evidence for this assumption, but there is no good ways to change the rank of feature while preserve other properties.

6. Result on Coding Noise

In this experiment, we use the VVenC to encode and decode the feature. We conduct the experiment on several different Quantization Parameter (QP) to check the result depending on different strength of coding noise distortion.

Table 5. The experimental results on distorted coco dataset with coding noise distortion method

Condition	SSIM/PSNR[F]	SSIM/PSNR[I]	mAP
Feature(QP=32)	0.501/29.60	-	35.5
Image(QP =32)	0.628/30.55	0.522/27.26	35.5
Feature(QP=40)	0.394/27.72	-	32.0
Image(QP=40)	0.500/27.77	0.429/26.37	29.8
Feature(QP=47)	0.138/24.51	-	14.0
Image(QP=47)	0.227/22.09	0.208/23.35	8.6

As we can see in Table 5, it shows that feature is stronger against coding noise effect. However, there is a performance decrease in higher QP. We analyze the result was caused by the compressor's spatial inference method called intra-coding method, which infers the scene by using the spatial information of the input.[16] We assume that the feature's small spatial size is the reason for the degradation of the performance.

Ⅴ. Conclusion

The result shows that except in gaussian distortion noise, distortion on the feature domain shows more robust properties compared with Image. We assume this robustness is caused by the amount of information in the feature. The feature map which is extracted by FPN contains many channels with various feature types and it has abundant information. Unless external random noise or distortion spoils all the critical feature values, vision task evaluation performance still works fine. However, feature size is smaller than the image size, so it loses more information when spatial related distortion is added. And we found out that simply comparing the similarity of distorted feature with original feature did not determine the performance drawn by the task evaluation network. We also conduct the mix of distortions experiment and find out that mixing of two feature-less-influenced noises still performs well, even image performance is decreasing more. In conclusion, the feature shows advantages of enduring many distortions than the image. We analyze this is due to plenty of information in feature. However, feature can be strongly affected by spatial distortion since the size of feature is smaller than the image size.

References

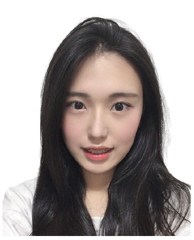
[1] Ren, S., He, K., Girshick, R., & Sun, J. ''Faster r-cnn: Towards real-time object detection with region proposal networks.'' Advances in neural information processing systems, 28, pp.91-99, 2015
[2] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. " Focal loss for dense object detection. '' In Proceedings of the IEEE international conference on computer vision. p. 2980-2988, 2017.
[3] He, K., Gkioxari, G., Dollár, P., & Girshick, R. "Mask r-cnn.'' In Proceedings of the IEEE international conference on computer vision, pp. 2961-2969, 2017.
[4] He, K., Zhang, X., Ren, S., & Sun, J. "Deep residual learning for image recognition.'' In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.
[5] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. "Feature pyramid networks for object detection.'' In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117-2125, 2017.
[6] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars,and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge." in Proc. 22nd ACM Int.Conf. Arch. Support Programming Languages and Operating System, pp. 615–629, 2017
[7] Bajić, Ivan V., Weisi Lin, and Yonghong Tian. "Collaborative intelligence: Challenges and opportunities.'' ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021.
[8] Duan, L., Liu, J., Yang, W., Huang, T., & Gao, W. "Video coding for machines: A paradigm of collaborative compression and intelligent analytics.'' IEEE Transactions on Image Processing, 29, 8680-8695, 2020.

[9] Girshick, R., Donahue, J., Darrell, T., & Malik, J. " Rich feature hierarchies for accurate object detection and semantic segmentation.'' In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587, 2014.
[10] Girshick, R. "Fast r-cnn.'' In Proceedings of the IEEE international conference on computer vision, pp. 1440-1448, 2015.
[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, arXiv:1704.04861.
[12] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. "Microsoft coco: Common objects in context.'' In European conference on computer vision, pp. 740-755, 2014.
[13] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A.,et.al, "Quantization and training of neural networks for efficient integer-arithmetic-only inference.'' In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2704-2713, 2018.
[14] Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J. H., & Liao, Q. "Deep learning for single image super-resolution: A brief review. '' IEEE Transactions on Multimedia, 21(12), 3106-3121, 2019.
[15] Fan, L., Zhang, F., Fan, H., & Zhang, C. "Brief review of image denoising techniques.'' Visual Computing for Industry, Biomedicine, and Art, 2(1), 1-12, 2019.
[16] Lainema, J., Bossen, F., Han, W. J., Min, J., & Ugur, K. "Intra coding of the HEVC standard.'' IEEE transactions on circuits and systems for video technology, 22(12), 1792-1801, 2012.
[17] Kuznetsova, Alina, et al. "The open images dataset v4." International Journal of Computer Vision 128.7, 1956-1981, 2020.

Introduction Authors

Jung Heum Kang

- May. 2020 ~ : He pursuining the M.S. degree with the Department of Computer Science and Engineering, Kyung Hee Universiity, Yongin, South Korea.
- May. 2016 ~ Feb. 2020 : He received the bachelor's degree from the Department of Computer Science and Engineering, Ajou University, Suwon, South Korea.
- ORCID : https://orcid.org/0000-0002-2890-8446
- Interests : Image Processing, Model Compression, Video Coding, Computer Vision, Machine Learning, nternational Standardization



Hye Won Jeong

- May. 2017 ~ : She pursuining the bachelor's degree from the Department of Software Convergence, Kyung Hee University, Yongin, South Korea.
- ORCID : https://orcid.org/0000-0003-1230-870X
- Interests : Image Processing, Video Coding, Computer Vision, Machine Learning, International Standardization

Chang Kyun Choi

- Sep. 2019 ~ Aug. 2021 : He received the M.S. degree from the Department of Computer Science and Engineering, Kyung Hee Universiity, Yongin, South Korea.
- May. 2013 ~ Aug. 2019 : He received the bachelor's degree from the Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea.
- ORCID : http://orcid.org/0000-0003-0127-556X
- Interests : Image Processing, Video Coding, Computer Vision, Machine Learning, International Standardization

Introduction Authors

Muhammad Salman Ali

- May. 2018 ~ : He pursuining the M.S. degree leading to Ph. D. degree with the Department of Computer Science and Engineering, Kyung Hee University, South Korea.
- May. 2013 ~ Feb. 2017 : He received the bachelor’s degree in computer science fromt the National University of Sciences and Technology (NUST), Islamabad, Pakistan.
- ORCID : http://orcid.org/0000-0002-8548-3827
- Interests : Interpretation of Deep Neural Networks, Image Processing, Video Coding, Machine Learning

Sung-Ho Bae

- Sep. 2017 ~ : He has been an Assistant Professor with Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea.
- Jul. 2016 ~ Aug. 2017 : He was a Postdoctoral Associate with Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), MA, USA.
- Feb. 2011 ~ Aug. 2016 : He received M.S. and Ph. D. degrees with Department of Electrical and Electronic Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea.
- May. 2004 ~ Feb. 2011 : He received the B.S. degree with he Department of Computer Science and Engineering/Electronics from Kyung Hee University, South Korea.
- ORCID : https://orcid.org/0000-0002-3389-1159
- Interests : Adversarial Attack, Model Compression, Semi-supervised Learning and Data Augmentation, Explainable DNN, Neural Architecture Search for Hardware Implementation, International Standardization

Hui Yong Kim

- Mar. 2020 ~ : He has been an Associate Professor in Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea.
- Sep. 2019 ~ Feb. 2020 : He has been an Associate Professor in Department of ICT Convergence Engineering, Department of Electronic Engineering, Sookmyung Women's University, Seoul, South Korea.
- Nov. 2005 ~ Aug. 2019 : He has been an Group Director/Responsible Researcher in Korea Electronics and Telecommunications Research Institute (ETRI) Realistic AV Research Group.
- Aug. 2014 ~ Sep. 2013 : He has been an Visiting Scholar of Media and Communication Research Institute in Univ. of Southern California, USA.
- Sep. 2006 ~ Aug. 2010 : He has been an affiliated professor in Department of Mobile Communication and Digital Broadcasting Engineering, Graduate School of Science and Technology (UST)
- Aug. 2003 ~ Oct. 2005 : He has been a Multimedia Team Leader in Adpark Technology Research Institute
- Mar. 1998 ~ Feb. 2004 : He received Ph. D. degrees with Department of Electrical and Electronic Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea.
- Mar. 1996 ~ Feb. 1998 : He received M.S. degrees with Department of Electrical and Electronic Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea.
- Mar. 1990 ~ Aug. 1994 : He received B.S. degrees with Department of Electrical and Electronic Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea.
- ORCID : https://orcid.org/0000-0001-7308-133X
- Interests : Image Processing, Video Coding, Computer Vision, Machine Learning, International Standarization