IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 July 2022, accepted 21 August 2022, date of publication 29 August 2022, date of current version 2 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3202172

RESEARCH ARTICLE

Edge Computing-Assisted DNN Image Recognition System With Progressive Image Retransmission

MUTSUKI NAKAHARA[1], (Life Member, IEEE), MAI NISHIMURA[2], (Member, IEEE), YOSHITAKA USHIKU[2], (Member, IEEE), TAKAYUKI NISHIO[3], (Senior Member, IEEE), KAZUKI MARUTA[4], (Senior Member, IEEE), YU NAKAYAMA[5], (Member, IEEE), AND DAISUKE HISANO[1], (Member, IEEE)

[1]Graduate School of Engineering, Osaka University, Suita, Osaka 565-0871, Japan
[2]OMRON SINIC X Corporation, Bunkyo, Tokyo 113-0033, Japan
[3]School of Engineering, Tokyo Institute of Technology, Ookayama, Meguro, Tokyo 152-8550, Japan
[4]Department of Electrical Engineering, Tokyo University of Science, Katsushika, Tokyo 125-8585, Japan
[5]Institute of Engineering, Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-8588, Japan

Corresponding author: Daisuke Hisano (hisano@comm.eng.osaka-u.ac.jp)

This work was supported in part by JSPS KAKENHI under Grant JP20H04178, Grant JP20K21789, and Grant JP21H03399; and in part by GMO Foundation, Japan.

ABSTRACT Deep learning-based image recognition systems have rapidly evolved. Due to the extensive processing load of the deep neural network (DNN) on graphic processing units (GPUs), the DNN model is deployed on the cloud server. Images or videos are forwarded from user terminals through the network to the server. In recent years, edge computing has gained popularity as a means of reducing the data traffic in the backbone network. However, the last one-mile access network between an edge server and user terminals will still be congested because a large amount of data such as video/image files must be forwarded. In particular, when computer vision applications such as image recognition are loaded in the edge network, a large amount of data is forwarded although the edge server always may not need the high-definition image. This paper proposes an image compression and progressive retransmission scheme for deep learning-based image recognition systems to reduce image data traffic and alleviate network congestion. The proposed method introduces an entropy-based threshold calculated from posterior probabilities from a deep learning model's output layer. Entropy is an extremely effective metric because it can be used as an indicator independent of the number of classification labels in the DNN model. The thresholding can control the image retransmission and reduce traffic while maintaining image recognition accuracy. We implement the proposed scheme on the edge server and reveal the relationship between the data compression and the recognition accuracy through simulation evaluation. As a result, we indicate that an entropy-based threshold reduces the overall ambiguity of the accuracy of image recognition. Moreover, when a higher accuracy recognition model with more accuracy is combined with a retransmission scheme, it becomes the more effective.

INDEX TERMS Edge computing, computer vision, image recognition, retransmission system.

I. INTRODUCTION

Deep neural network (DNN) has continued improving image recognition estimation accuracy in recent years. DNN-based image recognition has been aggressively used on Internet of things (IoT) applications. In the IoT, User terminals, such as smartphones and mobile sensors, collect data and send it to network servers, i.e., cloud, that analyze them and provide various services [1], [2]. Although the cloud computing is now one of the mainstream technologies, there are some challenges, such as increased latency and load between the user terminals with increasing IoT applications that require real-time processing and the number of user terminals. Therefore, edge computing technologies that relocate processing resources to the areas near the user terminals have been studied [3], [4], [5], [6], [7]. Edge computing can provide

The associate editor coordinating the review of this manuscript and approving it for publication was Fahmi Khalifa.

low-latency services because the user terminals and edge servers terminate all processing without the support of cloud servers. Note that this paper assumes that the applications of edge computing networks also include networks in not only 5G/6G networks but also small areas such as shopping centers, stadiums, etc. In particular, when edge computing provides computer vision applications such as image recognition and object detection using DNN, data reduction forwarded through the network is a critical issue [8], [9] as shown in Fig. 1. Self-driving cars [10], [11], surveillance camera analysis [12], [13], and traffic navigation services at tourist attractions are all examples of computer vision applications. The use of edge computing will grow in tandem with the expansion of services.

Furthermore, the amount of data collected from users increased year after year. The network between the edge server and user terminals will be congested in the near future. As a result, our goal is to avoid decreasing throughput, increasing latency, and network congestion on the edge computing system that supports computer vision applications. In particular, this study focuses on DNN-based image recognition systems. Popularly, image compression such as JPEG encoding is a straightforward technique to reduce data traffic. However, when the image is compressed at a large rate, the image compression degrades the estimation accuracy of the DNN [14]. There is a trade-off relationship between image compression and the DNN estimation accuracy. Moreover, the proper compression rate depends on images. In addition, DNN models have significantly evolved, and new models have been proposed. Therefore, the compression method is expected to be independent of the DNN model.

This paper proposes an edge-assisted image recognition method with image compression and progressive retransmission to overcome the above problem. The proposed method improves the trade-off between network efficiency and recognition accuracy in edge-assisted image recognition. The proposed edge estimates the recognition accuracy and requests the retransmission to the user terminal when the accuracy is estimated to be low. Then, the user terminal retransmits the higher-quality image. The proposed method can guarantee the estimation accuracy of image recognition.

The contribution of this paper is as follows;

- Entropy can be employed as on an indicator of the retransmission decision. An entropy-oriented decision is independent of the number of labels of DNN because the top-k output from the last layer of DNN is not used.Thus, entropy is the generic indicator.
- The proposed edge-assisted image recognition system reduces the network traffic. Using progressive JPEG provides more traffic reduction, and we confirm the effectivity with the simulation analysis.
- The proposed retransmission scheme processes can be operated independently of the estimation model of DNNs, so the proposed scheme can be applied to any image recognition models based on DNNs.

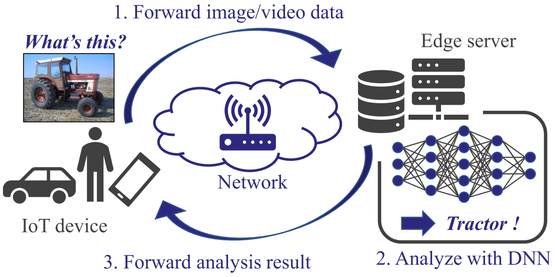


FIGURE 1. **Edge computing based image recognition.**

This paper evaluates the estimation accuracy and entropy of the image recognition by changing the image compression rate. We compare the proposed scheme with the image compression system without the retransmission in the entropy view point and the data size reduction.

This research is an expanded version of our previous work, which was published in IEEE VTC-Fall 2021 [15]. We extended the previous work in the following three aspects:

- AlexNet is replaced in IEEE VTC-Fall version with two major image recognition models, ResNet and EfficientNet, which are widely used in practical image recognition systems.
- The detail of the explanation and the discussion of the proposed system was extended, and the simulation data was obtained with the various parameters.
- The entropy was introduced as the retransmission decision indicator.

In the following chapters, Section II introduces related works, and Section III describes the proposed method. The experimental evaluations are presented in Section IV. Finally, we conclude in Section V.

II. RELATED WORK

This section introduces the representative schemes for edge computing to conduct the DNN and the advantage of our proposed scheme. Several schemes have been reported on the edge computing system, with the DNN covering the IoT applications, to avoid congestion between the edge server and user terminals. For example, the image compression scheme is useful before transmitting the image to the edge server at the user terminals.

J. Ren et al. proposed an image compression scheme for object detection based on the region of interest (ROI) [16]. The ROI refers to the area that includes the target object to be recognized. The proposed scheme sets a lower compression rate for the background region. As part of a related study, we proposed multiple ROI transmission schemes and reduced the number of background images transmissions in a narrow bandwidth, and high packet loss [17]. Li *et al.* proposed an image compression scheme focusing on the difference in the required image quality of each application [18]. The proposed scheme adaptively selected the JPEG compression rate between the edge server and the user terminals based on the argent designed by reinforcement learning. The above works employed the traditional compression methods, and the DNN was used to update the compression rate. In addition, JPEG encoding may not be optimal for DNN-based image recognition because the compression is tailored to human vision. Therefore, a method has been proposed to reconfigure JPEG encoding for DNNs [19].

Besides image compression, a new method called split computing has been proposed for enabling network-efficient edge-assisted image recognition [20]. In split computing, the DNN model is split into a head network and tail network, deployed to a user terminal and edge server, respectively. The user terminal inputs its obtained image with the head network, and the output of the hidden layer is forwarded to the edge server. Then, the server processes the rest with the tail network. Split computing can reduce traffic and latency by introducing a bottleneck architecture to the head network. Matsubara *et al.* have studied an efficient way to train the head network to reduce network traffic without degrading the model performance [21]. Itahara *et al.* studied a model tuning method to improve the model robustness against compression and network-induced packet losses [22]. However, the user terminal must have enough computing power to handle the head network to apply this split computing. In contrast, the IoT devices such as network cameras and wearable sensors often do not have such computation power.

The followings are the key features of our proposed scheme:

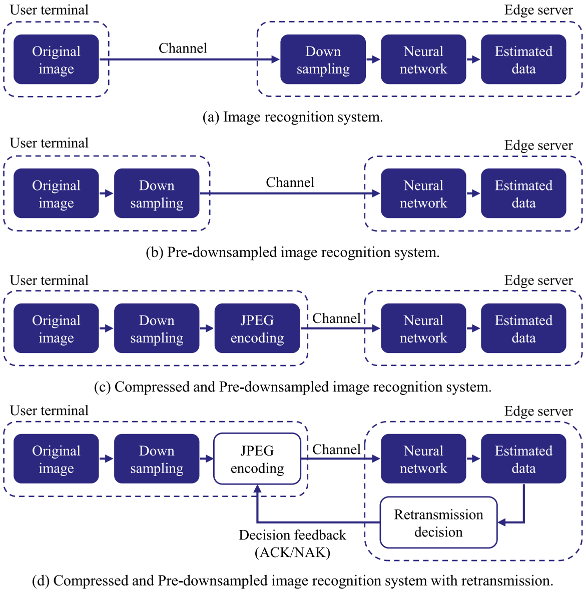
- When the estimation model was a classifier, the proposed scheme did not refer to or retrain the DNN model. The proposed scheme can be carried out even when the model is updated. The proposed scheme employs the entropy calculated by the posterior probability distribution output from the softmax of the DNN as the image retransmission decision indicator. Any estimation model can be used as long as the posterior probability distribution is obtained.
- The proposed scheme does not affect the related work introduced in this section. As a result, we can employ both schemes at the same time.

The next section describes the principle of operation of the proposed scheme in detail.

III. PROPOSED EDGE-ASSISTED IMAGE RECOGNITION

A. OVERVIEW

This section describes the concept of the proposed scheme. Fig. 2 (a)–(d) shows the candidate for the edge computing system for traffic reduction. Fig. 2 (a) is the configuration of the normal image recognition with edge computing. The user terminal has the original image and sends it to the edge server. Prior to recognizing the image with the DNN, the edge server uses a downsampling method to match the image size with the input size of the DNN. For example, the input size in ResNet, a popular model of the DNN, is 224 × 224 pixels. While Fig. 2 indicates only downsampling, the image is up-sampled when the image size is smaller than the input size.



(a) Image recognition system.

(b) Pre-downsampled image recognition system.

(c) Compressed and Pre-downsampled image recognition system.

(d) Compressed and Pre-downsampled image recognition system with retransmission.

FIGURE 2. Candidate for edge computing system for the traffic reduction.

In Fig. 2 (b), the user terminal conducts the downsampling in advance. We anticipate a reduction in the image size. Furthermore, the system has no effect on image recognition accuracy. Meanwhile, the data size is equal to the total number of pixels multiplied by 24 bits. As a result, we anticipate greater traffic reduction when the user terminal performs JPEG encoding before transmission, as shown in Fig. 2 (c). However, when lossy compression is used, the recognition accuracy decreases.

Fig. 2 (d) shows the proposed retransmission scheme. In the proposed scheme, users downsample and compress images before sending them to an edge server. The edge server uses the DNN to recognize the images. The edge server sends the Image-NAK retransmission request message when the estimation accuracy falls below the predefined threshold. The Image-NAK-received user terminals reset the compression rate to a lower value and resend the images to the edge server. When the edge server achieves sufficient accuracy, it transmits an acknowledgment message known as Image-ACK and ends the forwarding process. In addition, the edge server terminates the process when the number of image retransmissions reaches a certain threshold. It is worth noting that ACK and NAK messages of TCP connection are communicated in the network. The ACK and NAK are different messages of Image-ACK and Image-NAK.

B. IMAGE COMPRESSION FORMAT

We introduce two types of image compression format; The first is a baseline JPEG encoding standardized by ISO/IEC JTC 1/SC 29. The other is a progressive JPEG format. The progressive JPEG stores the binary data in order, starting with the image's lower resolution (frequency) components. In other words, the image can be opened even when the binary data is cut from the beginning to the middle. The shorter the binary data is cut, the coarser the image. Meanwhile, the

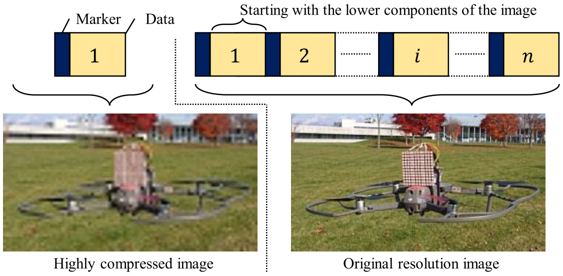

Marker
Data
Starting with the lower components of the image
1
1
2
i
n
Highly compressed image
Original resolution image

FIGURE 3. Example of progressive JPEG format.

standard JPEG format cannot open the image when the binary data is cut in the middle. The progressive JPEG embeds markers in the binary data. The compression rate is calculated at the marker position. For example, Fig. 3 shows the highly compressed image with a first marker position from the beginning of the binary data and the original image with all binary data. In Fig. 3, we assume that the maximum number of markers is n. Higher-frequency components are absent from the highly compressed image. The compression rate of the proposed scheme is controlled by referring to the markers. In this paper, the marker number is referred to as the compression step.

C. OPERATION OF USER TERMINALS

1) BASELINE JPEG CASE

In advance, user terminals perform downsampling and convert the image format into standard JPEG. The user terminal sets the quality in the range of 0–100% of the JPEG to compress the image. The level of quality is predetermined. When the user terminal receives the Image-NAK, it recompresses the image using the next designated compression rate. This phase is repeated until Image-ACK is received or the retransmission limit is reached.

2) PROGRESSIVE JPEG CASE

User terminals conduct the downsampling and convert the image format into progressive JPEG in advance. Algorithm 1 shows the operation of the user terminals in the proposed scheme. The image in the progressive JPEG format is converted to binary data D_{orig}. The algorithm then reads the current compression step σ_c. c is the compression step number. The initial compression step is assumed to σ_i. The reception of Image-NAK suggests that the forwarding process has already been performed several times. Thus, the algorithm extracts the binary data D_p at positions from the previous compression step σ_c to the currently designated compression step σ_f. When the Image-NAK has never been received, and this is the first time the image is being forwarded, the user terminals forward the binary data from the beginning of the data to the initial compression step σ_i.

While the user terminal continues to send the compressed image to the edge server until the image recognition is complete, overlapped data are not forwarded due to the progressive transmission. From this perspective, the proposed scheme contributes to the traffic reduction.

Algorithm 1 User Terminals in Progressive Retransmission

```
Input: Binary data $D_{orig}$ with or without NAK $N_f$
Output: Requested partial binary data $D_p$
  Read: Current compression step $\sigma_c$
  if Receive NAK $N_f$? then
    Extract: Binary data $D_p$ between comp. steps $\sigma_c$ and $\sigma_f$ in $D_{orig}$
    Update: $\sigma_c \leftarrow \sigma_f$
  else
    Extract: Binary data $D_p$ until comp. steps $\sigma_i$ in $D_{orig}$
    Update: $\sigma_c \leftarrow \sigma_i$
  end if
  return Requested partial binary data $D_p$
```

D. OPERATION AT EDGE SERVER SIDE

Algorithm 2 shows the operation of the edge server. The edge server combines the binary data D_p just received with the data D has already received and composes the image y. The image y is input into the prediction model of the DNN. Note again that we use a pre-trained DNN model, get only the output of the DNN model, and calculate the posterior probability. That is, we need not retrain the DNN model. Here, the threshold was required be set for the retransmission. Entropy and top-k error are introduced as the decision indicator. The server calculates the entropy $E(y)$ using the posterior probability $p(x_i|y)$ from the output layer.

$$E(y) = -\sum_{i=1}^{L} p(x_i|y) \log p(x_i|y), \tag{1}$$

where y is the input image, x_i is the i-th label, L is the total number of labels. In the top-k case, the top-k error is expressed as,

$$E(y) = 1 - \sum_{i=1}^{k} p(x_i|y). \tag{2}$$

Up to k-th the posterior probability are summed. The prediction model is combined with the softmax layer to convert the logits into pseudoposterior probability. When $E(y)$ is less than the E_{th} threshold, the edge server requests retransmission to the user terminal. The entropy threshold is predetermined, and the maximum number of retransmissions is also limited.

The next section confirms the entropy by varying the compression steps with the ResNet and the EfficientNet, which are the typical image recognition models.

IV. INVESTIGATION OF ENTROPY AND TOP-K ERROR PROPERTIES

A. SETUP

The proposed scheme must set the following parameters in advance to retransmit the compressed images.

- JPEG quality and compression steps in initial transmission and retransmissions,
- Top-k and entropy threshold for the decision of prediction accuracy.

Algorithm 2 Edge Server

```
Input: Binary data $D_p$
Output: DNN output and image-ACK, or only image-NAK
  Read: The number of data receptions $N_r$
  Read: Entropy threshold $E_{th}$
  Read: Maximum number of retransmission requests $N_r^{max}$
  Read: Binary data already received $D$
  if Progressive transmission then
    Append: $D \leftarrow D + D_p$
  else
    Update: $D \leftarrow D_p$
  end if
  Reconstruct: Image $y$ from $D$
  DNN-based Recognize: Posterior probability $p(x_i|y)$
  Calculate: Entropy $E(y)$
  if $E(y) < E_{th}$ or $N_r \geq N_r^{max}$ then
    Initialize: $N_r$
    return DNN output and image-ACK
  else
    Update: $N_r$
    return Image-NAK $N_f$
  end if
```

These parameters depend on the dataset and prediction model. This paper introduced ImageNet datasets [23]. ImageNet dataset includes 1,200,000 train images, 50,000 validation images, and 100,000 test images of 1,000 class. We used the test dataset for the experiment for setting the threshold of the retransmission and compression step in this section. The validation dataset was used for the experiment to evaluate the feasibility of the proposed scheme in the next section. The reason why we separated the dataset into test and validation was to avoid the overfitting. The applied prediction models of the DNN were,

- ResNet-50,
- EfficientNet-B7

The input sizes of ResNet-50 and EfficientNet-B7 are 224 × 224 pixels and 600 × 600 pixels, respectively. The prediction model is provided by Tensorflow library. We used the provided and pre-trained model. That is, we conducted no additional learning and no change in the layer structure. The proposed scheme conducts the downsampling and JPEG encoding as preliminary treatment. Thus, this section studied the relationship of the JPEG quality, compression step, and data size versus top-1, top-5, and entropy. In addition, NVIDIA RTX3090 was used for the machine specification, including 24-GB GPU, and AMD Ryzen 7 3700X. All of the simulations were carried out on this machine.

B. BASELINE JPEG RESULTS

Fig. 4 (a) shows normalized data size when changing the JPEG quality. The normalized data size is the total data size of the compressed images in the test dataset divided by that of the original images. The solid line and the color region display the average value and standard deviation, respectively. The data size changed nonlinearly against the quality. Fig. 4 (b)–(d) shows the top-1 output, the top-5 output, and the entropy when changing the JPEG quality. The solid and the dashed lines indicate average and median values, respectively. Both cases of ResNet-50 and EfficientNet-B7 changed the slopes by around $10-20\%$. We used the mean or median value as the threshold for the decision of the prediction accuracy. Table 1 summarizes the prediction threshold values. In addition, we used the values around inflection points as the threshold of the retransmission.

TABLE 1. Prediction threshold in baseline JPEG case. There are 12 settings.

Model type	Statistic type	Threshold type	Threshold value
ResNet	Mean	Top-1	79.65%
		Top-5	94.62%
		Entropy	1.0702 bit
	Median	Top-1	92.69%
		Top-5	99.76%
		Entropy	0.5066 bit
EfficientNet	Mean	Top-1	70.25%
		Top-5	79.02%
		Entropy	3.0753 bit
	Median	Top-1	77.43%
		Top-5	81.23%
		Entropy	2.8067 bit

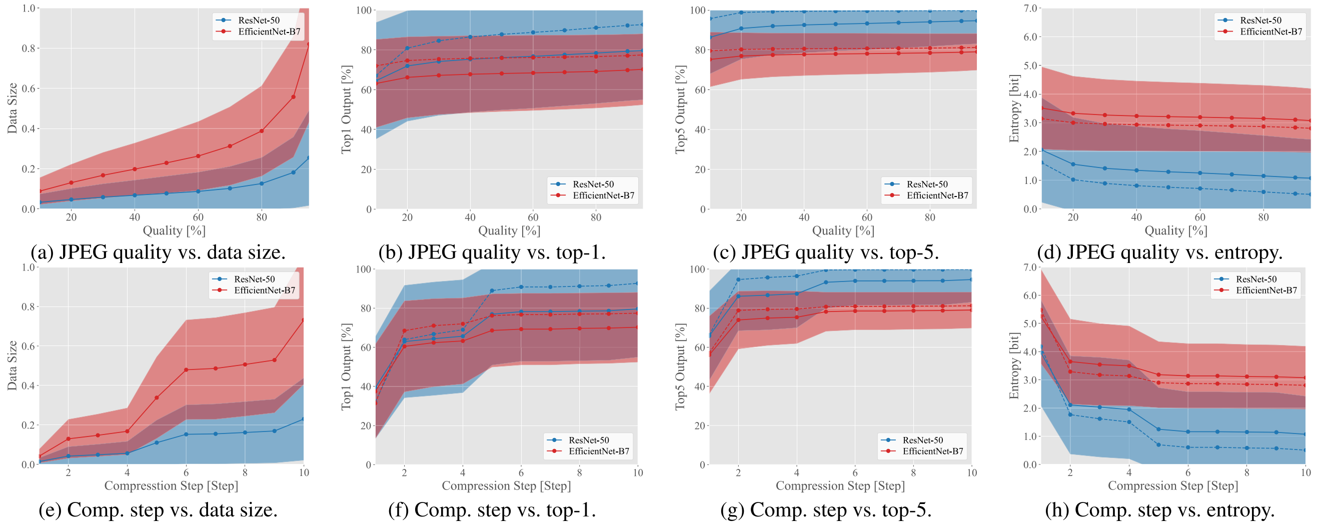
C. PROGRESSIVE JPEG RESULTS

Fig. 4 (e) shows the normalized average data size changed by the compression step of the progressive JPEG image. We converted the baseline JPEG format of the ImageNet test dataset into the progressive JPEG format and then set the JPEG quality to 95%. When the compression step is 2, 4, and 6, the data size was steeply changed. Fig. 4 (f)–(h) shows top-1 output, top-5 output, and entropy. Fig. 4 (e)–(h) were changed at the same inflection points. We set the threshold for the retransmission to mean or median value at compression step = 10. Table 2 summarizes the prediction threshold values.

V. EXPERIMENTAL EVALUATION

A. SETUP

We evaluated the proposed scheme. This section used validation datasets of ImageNet. Table 3 shows the evaluation items. We prepared ten items. Indexes (1)–(4) used the baseline JPEG format, and Indexes (5)–(10) used the progressive JPEG format. We employed a two-pattern threshold of mean or median as shown in Tables 1 and 2 on the retransmission decision. In addition, we applied the top-1 error, the top-5



(a) JPEG quality vs. data size.
(b) JPEG quality vs. top-1.
(c) JPEG quality vs. top-5.
(d) JPEG quality vs. entropy.
(e) Comp. step vs. data size.
(f) Comp. step vs. top-1.
(g) Comp. step vs. top-5.
(h) Comp. step vs. entropy.

FIGURE 4. Variations of data size, top-1, top-5, and entropy when changing JPEG quality (above) and compression step (bottom).

TABLE 2. Prediction threshold in progressive JPEG case. There are 12 settings.

Model type	Statistic type	Threshold type	Threshold value
ResNet	Mean	Top-1	79.65%
		Top-5	94.62%
		Entropy	1.0700 bit
	Median	Top-1	92.69%
		Top-5	99.76%
		Entropy	0.5066 bit
EfficientNet	Mean	Top-1	70.25%
		Top-5	79.02%
		Entropy	3.0753 bit
	Median	Top-1	77.43%
		Top-5	81.23%
		Entropy	2.8067 bit

error, and the entropy as the criterion for the retransmission decision. The fourth, fifth, and sixth columns indicate the JPEG quality or the compression step. The fourth column shows the case of the initial transmission (indicated as "Trans."). In the initial transmission, images are forwarded using JPEG quality or step as shown in fourth column of Table 3. This paper set the maximum number of retransmissions to twice. The fifth and sixth columns mean the first and second retransmissions cases (indicated as "1st retrans." and "2nd retrans."), respectively. For example, in the progressive JPEG case of the index (5), the user terminal forwards the binary data from zero to one step as the initial transmission. The binary data from two to four-step is forwarded in the first retransmission. Finally, in the second retransmission, the user terminal forwards the data from 5 to σ_{max}.

We prepared the comparison data in the baseline JPEG format without retransmission. The JPEG quality was changed from 10% to 95%. We simulated the relationship between the forwarded data size and the top-1 error, the top-5 error, and the entropy. If having a smaller data size and a smaller error, the proposed schemes have an advantage over the baseline JPEG transmission without the retransmission. The second experiment evaluated the number of retransmissions for all indexes, as shown in Table 3. These evaluations employed ResNet and EfficientNet on the prediction model.

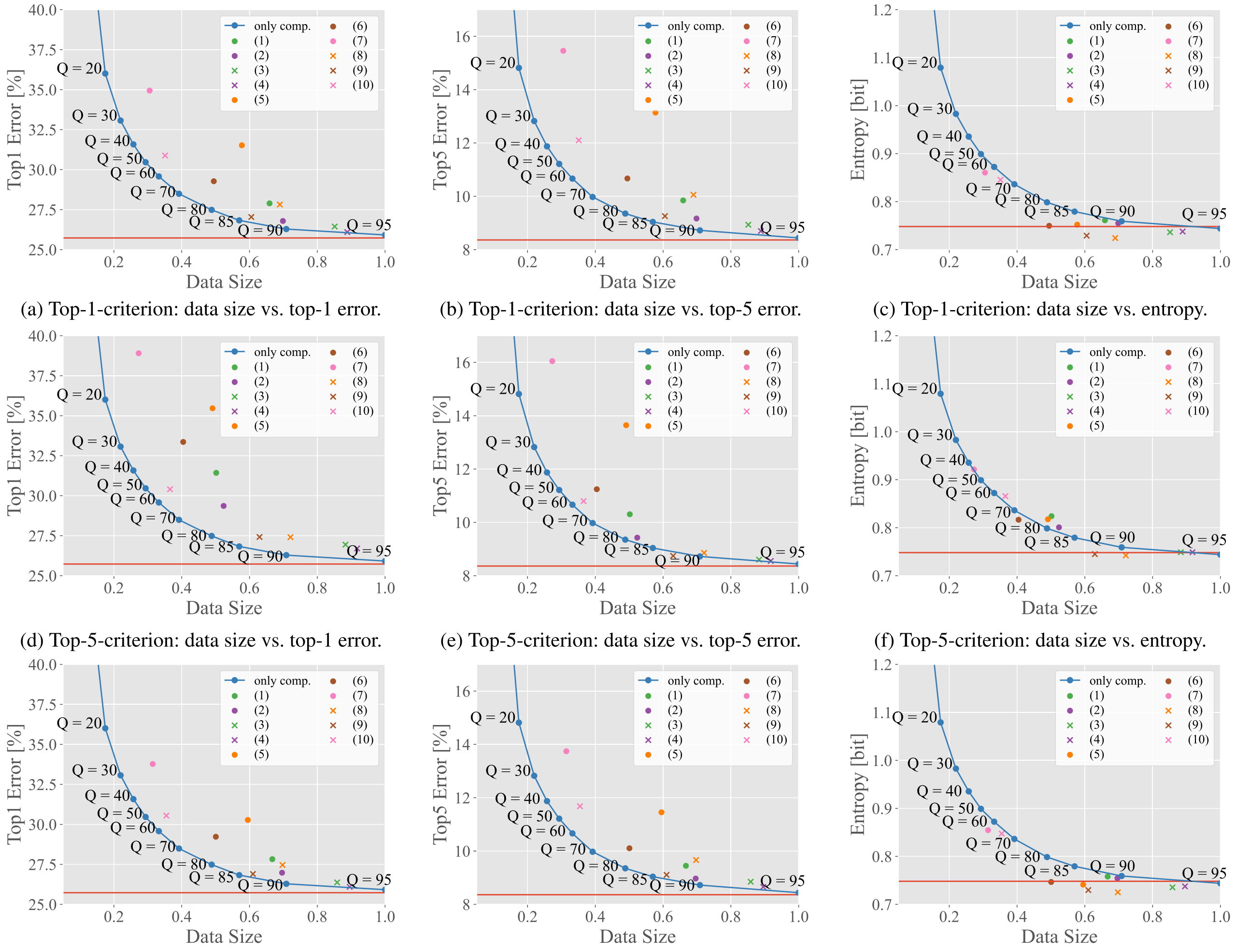
The proposed system is a novel topic for edge computing systems since it adds only a retransmission process that does not affect the DNN model. Thus, it is difficult to compare the proposed system with the related work. To fundamentally evaluate the effectivity, we compared it with the baseline JPEG. We used the published DNN model without the change, e.g., retraining or fine-tuning.

This verification assumed the ideal communication channel. In other words, the channel has no packet loss characteristics. When considering a practical communication channel, packet loss and forwarding latency affect the retransmission delay directly; however, we deal with this problem as a further study. In this paper, we reveal the prime potential of the proposed method.

B. RESULTS

1) DATA SIZE VS. ERROR

Fig. 5 shows the result of using ResNet. In Fig. 5 (a)–(c), we used the top-1 error for the retransmission decision. (d)–(f) and (g)–(i) used the top-5 error and the entropy, respectively. The blue line indicates the result of the compression transmission without retransmissions. Variable Q is the JPEG quality. The red line is the result in the typical prediction case with ResNet. From Fig. 5, except for (c), (f), and (i), the points of the proposed scheme were mapped on the right side against the blue line. It indicates the effectiveness of the proposed system is low. Note that index (9)



(a) Top-1-criterion: data size vs. top-1 error.

(b) Top-1-criterion: data size vs. top-5 error.

(c) Top-1-criterion: data size vs. entropy.

(d) Top-5-criterion: data size vs. top-1 error.

(e) Top-5-criterion: data size vs. top-5 error.

(f) Top-5-criterion: data size vs. entropy.

(g) Entropy-criterion: data size vs. top-1 error.

(h) Entropy-criterion: data size vs. top-5 error.

(i) Entropy-criterion: data size vs. entropy.

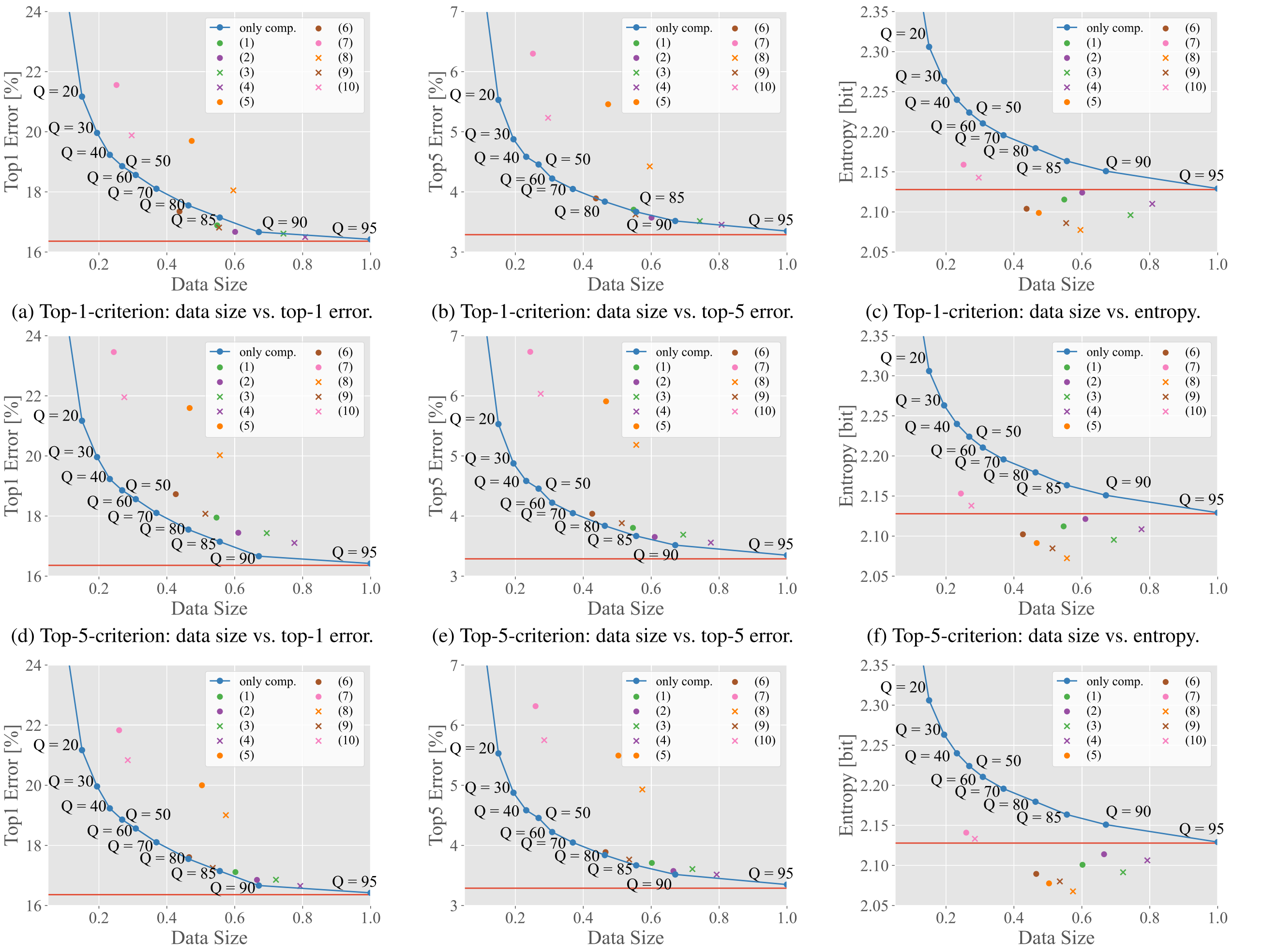
FIGURE 5. **ResNet results.**

is slightly effective in Fig. 5 (e). Meanwhile, the proposed scheme is effective in the entropy evaluation. In particular, all the points with the proposed scheme were mapped on the left side against the blue line when the entropy criterion decision was employed, as shown in Fig. 5 (i). Notably, the results with the proposed scheme were improved than the typical ResNet result indicated with the red line. The lower entropy means that it is possible to include the correct label near the top even when the correct label is out of top-5.

Fig. 6 shows the results employed EfficientNet. The red line is the typical EfficientNet results without the additional compression. The error is more minor because the EfficientNet accuracy is better than the ResNet. For this reason, the proposed scheme was effective in the cases of the top-1 error and the top-5 error, unlike the ResNet case. The entropy case was improved than the ResNet case. That is, the better the accuracy of the model, the more effective the proposed method is.

2) NUMBER OF RETRANSMISSIONS

Fig. 7 shows the breakdown of the number of retransmissions in all validation data. The horizontal axis is the index, as shown in Table 3. The vertical axis is the ratio of the number of retransmissions. For example, the index (1) in Fig. 7 (a) includes 40% of the validation datasets with the initial transmission, 20% of those with the first retransmission, and 40% of those with the second retransmissions. Overall, the baseline JPEG cases were more likely to be accepted without the retransmission, while the progressive JPEG cases were more likely to need the retransmission. In addition, the EfficientNet case contained around 40% to 50% of the second retransmission. This is because the EffficientNet accuracy is better than the ResNet. The ResNet had the larger number of the second retransmission. While the retransmission scheme is operated effectively, an increase in the retransmissions causes an increase in latency. Thus, the number of retransmissions should be limited when the proposed scheme is employed on mission-critical systems.



(a) Top-1-criterion: data size vs. top-1 error. (b) Top-1-criterion: data size vs. top-5 error. (c) Top-1-criterion: data size vs. entropy.

(d) Top-5-criterion: data size vs. top-1 error. (e) Top-5-criterion: data size vs. top-5 error. (f) Top-5-criterion: data size vs. entropy.

(g) Entropy-criterion: data size vs. top-1 error. (h) Entropy-criterion: data size vs. top-5 error. (i) Entropy-criterion: data size vs. entropy.

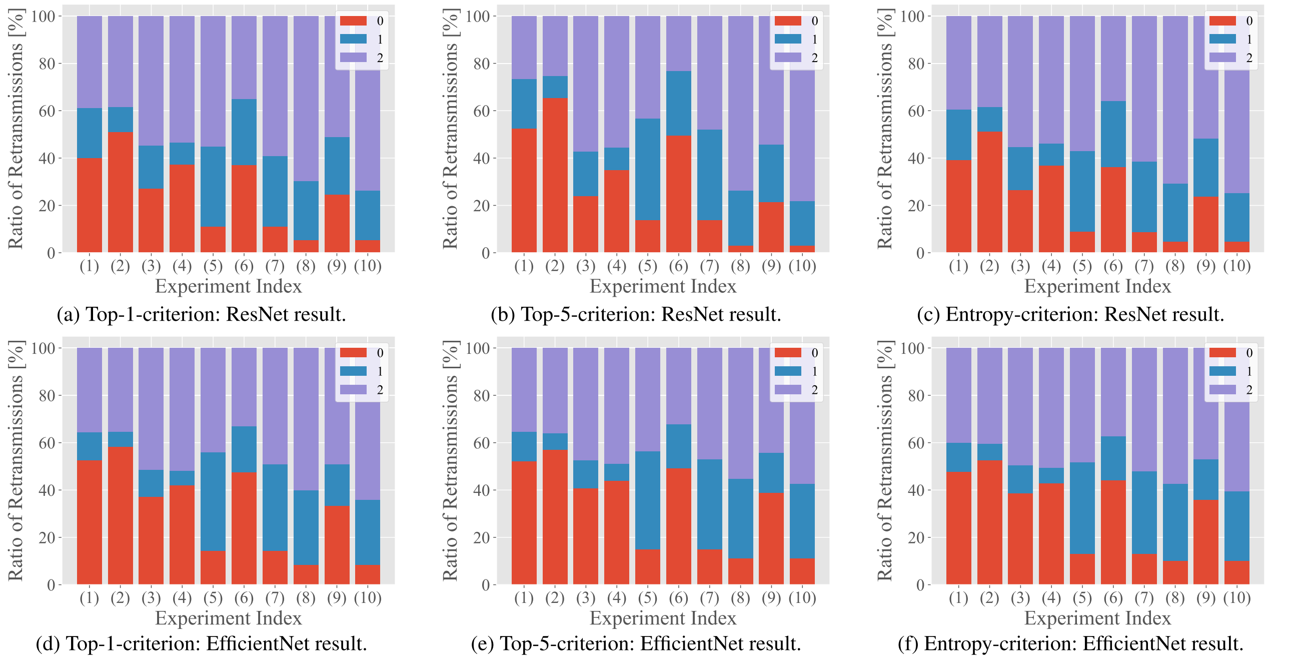
FIGURE 6. EfficientNet results.

TABLE 3. Evaluation patterns.

Index	JPEG type	Decision	Trans.	1st retrans.	2nd retrans.
(1)	Baseline	Mean	10%	40%	95%
(2)			20%	50%	95%
(3)		Median	10%	40%	95%
(4)			20%	50%	95%
(5)	Progressive	Mean	1 step	4 step	σ_{max} step
(6)			2 step	5 step	σ_{max} step
(7)			1 step	2 step	5 step
(8)		Median	1 step	4 step	σ_{max} step
(9)			2 step	5 step	σ_{max} step
(10)			1 step	2 step	5 step

VI. LIMITATION OF PROPOSED SYSTEM

The proposed scheme aims to the system of image classification. It calculates the entropy from the classification results and guarantees the accuracy of the classification. The proposed scheme cannot be directly employed in object detection from an image, including multiple objects and segmentation. For the object detection [24], as a new method, we can reset ROI from the accuracy of the detected object and request to retransmit the image that includes the minimum required pixels. Meanwhile, an indicator by using the accuracy of object detection is needed. These problems are future works. For the segmentation, it may be easy to apply the proposed scheme to a segmentation method using belied map [25]. We will calculate the entropy from the conditional random field (CRF). The segmentation using attention [26] requires an image with multiple size. That is, a high resolution is needed. Pre-compression on the user terminal side, as in the proposed method, may not be suitable for the segmentation scheme. In the future work, we plan to extend the proposed method to not only these object detection and segmentation, but also video data.



(a) Top-1-criterion: ResNet result.

(b) Top-5-criterion: ResNet result.

(c) Entropy-criterion: ResNet result.

(d) Top-1-criterion: EfficientNet result.

(e) Top-5-criterion: EfficientNet result.

(f) Entropy-criterion: EfficientNet result.

FIGURE 7. Breakdown of the number of retransmissions in all test images.

VII. CONCLUSION

This paper proposed edge-assisted image recognition systems with progressive retransmission to reduce image data traffic and alleviate network congestion. We introduced a threshold based on entropy metric calculated from posterior probabilities from a deep learning model's output layer. We implemented the proposed scheme on the edge server. In this paper, we first calculated the practical threshold in the cased of top-1, top-5, and entropy when using ResNet and EfficientNet. We simulated the proposed image recognition system with baseline and progressive JPEG images using the calculated thresholds. The simulation results revealed the relationship between the data compression and the recognition accuracy. In the ResNet case, while top-1 and top-5 results were not exceeded the baseline compression method, the entropy result was drastically improved. Moreover, in the EfficientNet case, the proposed system indicated an improvement compared with the baseline method. This result implied that the higher the accuracy of the original DNN model, the better the proposed method also returns results. Further studies include employing the proposed scheme on more advanced computer vision applications such as object detection and experiments using commercially-supported edge systems.

REFERENCES

[1] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[2] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022–23040, 2020.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st, Ed., MCC workshop Mobile cloud Comput.*, 2012, pp. 13–16.

[4] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[6] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.

[7] J. Nakazato, M. Kuchitsu, A. Pawar, S. Masuko, K. Tokugawa, K. Kubota, K. Maruta, and K. Sakaguchi, "Proof-of-concept of distributed optimization of micro-services on edge computing for beyond 5G," in *Proc. IEEE 95th Veh. Technol. Conference: (VTC-Spring)*, Jun. 2022, pp. 1–6.

[8] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, Jan. 2020.

[9] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[10] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim, and C. S. Hong, "Deep learning based caching for self-driving cars in multi-access edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2862–2877, May 2021.

[11] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[12] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018.

[13] G. Cho, Y. Shinyama, J. Nakazato, K. Maruta, and K. Sakaguchi, "Object recognition network using continuous roadside cameras," in *Proc. IEEE 95th Veh. Technol. Conference: (VTC-Spring)*, Jun. 2022, pp. 1–5.

[14] W. Yuan, K. Maruta, Y. Nakayama, D. Hisano, and K. Sakaguchi, "Image size reduction by road-side edge computing for wireless relay transmission and object detection," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2022, pp. 959–960.

[15] M. Nakahara, D. Hisano, M. Nishimura, Y. Ushiku, K. Maruta, and Y. Nakayama, "Retransmission edge computing system conducting adaptive image compression based on image recognition accuracy," in *Proc. IEEE 94th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2021, pp. 1–5.

[16] J. Ren, Y. Guo, D. Zhang, Q. Liu, and Y. Zhang, "Distributed and efficient object detection in edge computing: Challenges and solutions," *IEEE Netw.*, vol. 32, no. 6, pp. 137–143, Nov./Dec. 2018.

[17] E. Takeshita, A. Sakaguchi, D. Hisano, Y. Inoue, K. Maruta, Y. Hara-Azumi, and Y. Nakayama, "Stochastic image transmission with CoAP for extreme environments," 2022, *arXiv:2205.01852*.

[18] H. Li, Y. Guo, Z. Wang, S. Xia, and W. Zhu, "AdaCompress: Adaptive compression for online computer vision services," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 2440–2448.

[19] Z. Liu, T. Liu, W. Wen, L. Jiang, J. Xu, Y. Wang, and G. Quan, "DeepN-JPEG: A deep neural network favorable JPEG-based image compression framework," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.

[20] Y. Matsubara, M. Levorato, and F. Restuccia, "Split computing and early exiting for deep learning applications: Survey and research challenges," 2021, *arXiv:2103.04505*.

[21] Y. Matsubara, R. Yang, M. Levorato, and S. Mandt, "Supervised compression for resource-constrained edge computing systems," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 2685–2695.

[22] S. Itahara, T. Nishio, Y. Koda, and K. Yamamoto, "Communication-oriented model fine-tuning for packet-loss resilient distributed inference under highly lossy IoT networks," *IEEE Access*, vol. 10, pp. 14969–14979, 2022.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

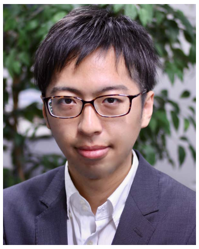
[25] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*.

[26] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3640–3649.

MUTSUKI NAKAHARA (Life Member, IEEE) received the B.E. degree in mechatronics engineering course from the National Institute of Technology, Wakayama College, Wakayama, Japan, in 2020, and the M.E. degree in electrical, electronic and infocommunications engineering from Osaka University, Osaka, Japan, in 2022. His research interests include computer vision, machine learning, and sensor networks. He was a recipient of IEEE VTS Tokyo/Japan Chapter Young Researcher's Encouragement Award, in September 2021.

MAI NISHIMURA (Member, IEEE) received the B.E. and M.S. degrees from Kyoto University, in 2013 and 2015, respectively. In 2015, she joined the NTT Research Laboratories, where she was involved in researches on low-level computer vision. From 2017 to 2019, she worked as a Senior Software Engineer at Fixstars Inc. She is currently a Research Engineer at OMRON SINIC X. Her research interests include 3D computer vision and GPU-accelerated machine learning.

YOSHITAKA USHIKU (Member, IEEE) received the B.E., M.A., and Ph.D. degrees from The University of Tokyo, Japan, in 2009, 2011, and 2014, respectively. In 2014, he joined NTT CS Laboratories, Japan, where he was involved in research on image recognition. From 2016 to 2018, he was a Lecturer with The University of Tokyo. He has been a Principal Investigator at OMRON SINIC X and a Chief Research Officer at Ridge-I, since 2018 and 2019, respectively. His research interests include cross-media understanding through machine learning, mainly for computer vision and natural language processing. He received ACM Multimedia Grand Challenge Special Prize, in 2011, ACM Multimedia Open Source Software Competition Honorable Mention, in 2017, and NVIDIA Pioneering Research Awards, in 2017 and 2018.



TAKAYUKI NISHIO (Senior Member, IEEE) received the B.E. degree in electrical and electronic engineering and the master's and Ph.D. degrees in informatics from Kyoto University, in 2010, 2012, and 2013, respectively. He was an Assistant Professor with the Graduate School of Informatics, Kyoto University, from 2013 to 2020. From 2016 to 2017, he was a Visiting Researcher with the Wireless Information Network Laboratory (WINLAB), Rutgers University, USA. He has been an Associate Professor with the School of Engineering, Tokyo Institute of Technology, Japan, since 2020. His current research interests include machine learning-based network control, machine learning in wireless networks, and heterogeneous resource management.

KAZUKI MARUTA (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in engineering from Kyushu University, Japan, in 2006, 2008, and 2016, respectively. From 2008 to 2017, he was with the NTT Access Network Service Systems Laboratories. From 2017 to 2020, he was an Assistant Professor with the Graduate School of Engineering, Chiba University. From 2020 to 2022, he was a Specially Appointed Associate Professor with the Academy for Super Smart Society, Tokyo Institute of Technology. He is currently an Associate Professor at the Department of Electrical Engineering, Tokyo University of Science. His research interests include MIMO, adaptive array signal processing, channel estimation, medium access control protocols, and moving networks. He is a Senior Member of IEICE. He received the IEICE Radio Communication Systems (RCS) Active Researcher Award, in 2014, the APMC 2014 Prize, the IEICE RCS Outstanding Researcher Award, in 2018, and the IEEE ICCE Excellent Paper Award, in 2021.

YU NAKAYAMA (Member, IEEE) received the B.A., M.E., and Ph.D. degrees in agriculture, environmental studies, and information and communication engineering from The University of Tokyo, Tokyo, Japan, in 2006, 2008, and 2018, respectively. In 2008, he joined the NTT Access Network Service Systems Laboratories, NTT Corporation. He is currently an Associate Professor with the Institute of Engineering, Tokyo University of Agriculture and Technology. He is also the President of the neko 9 Laboratories, which is a nonprofit organization in Tokyo. His research interests include adaptive networks, network architecture, packet switching, and sensor networks. He is a member of IEICE and IPSJ.

DAISUKE HISANO (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electrical, electronic and information engineering from Osaka University, Osaka, Japan, in 2012, 2014, and 2018, respectively. In 2014, he joined the NTT Access Network Service Systems Laboratories, Yokosuka, Japan. Since October 2018, he has been an Assistant Professor with Osaka University. His research interests include optical-wireless converged networks, optical communication, all-optical signal processing, visible light communication, edge computing, and application of deep learning to optical communication. He is also a member of the IEICE. He was a recipient of IEEE Kansai Section Young Professionals Awards, in 2021.