Received July 12, 2019, accepted July 25, 2019, date of publication August 2, 2019, date of current version August 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932726

Quality Recovery for Image Recognition

MOTOHIRO TAKAGI[1], (Member, IEEE), AKITO SAKURAI[2], AND MASAFUMI HAGIWARA[1], (Senior Member, IEEE)

[1]Department of Electrical Engineering, Keio University, Yokohama 223-8522, Japan
[2]Institute of Advanced Sciences, Yokohama National University, Yokohama 240-8501, Japan

Corresponding author: Motohiro Takagi (motohiro.takagi@ieee.org)

ABSTRACT This paper proposes a quality recovery network (QRNet) that recovers the image quality from distorted images and improves the classification accuracy for image classification using these recovered images as the classifier inputs, which are optimized for image quality loss and classification loss. In certain image classification tasks, classifiers based on deep neural networks achieve higher performance compared to those realized by humans. However, these tasks are based on images that are not distorted. To address distorted images, the classifier is fine-tuned with distorted images for practical applications. However, fine-tuning is insufficient for classifying images that include multiple distortion types with severe distortions and often requires the classifier to be retrained for adapting to distorted images, which is a time-consuming process. Therefore, we propose QRNet that generates recovered images for input to the classifier. To address multiple severe distortions, the proposed network is trained using multiple distortion-type images with our proposed loss, which comprises the image quality and classification losses. Moreover, by training the proposed network with multiple classifiers, the recovered images can be easily classified by a new classifier that is not used for training. The new classifier can classify the recovered images without retraining for adapting to distorted images. We evaluate our proposed network with classifiers on public datasets and demonstrate that it improves the classification accuracy for distorted images. Moreover, the experimental results demonstrate that our proposed network with the new classifier improves the classification accuracy.

INDEX TERMS Autoencoder, convolutional neural network, deep neural network, image quality.

I. INTRODUCTION

Image processing applications that use deep neural networks (DNNs) achieve high performance in several tasks such as image classification [1], [2] and semantic segmentation [3], [4]. However, these tasks assume undistorted images as the inputs and therefore, for training the datasets as well. In practical applications, several distortions such as compression distortion, motion blur, and image sensor noise exist. These distortions reduce the classification accuracy of image classification tasks [5], [6]. Furthermore, humans continue to outperform classifiers based on DNNs in the classification of distorted images [7]. Thus, as the classification of distorted images by DNNs has not yet been resolved, improving the classification of distorted images is crucial.

One solution to address these distortions is to fine-tune a DNN-based classifier with distorted images. Fine-tuning is one of the best solutions because of its ease of use. In [8], [9], fine-tuning has shown good performance for distorted images. Although fine-tuning is an effective method for classifying distorted images, the usage of fine-tuning solutions is limited; for instance, fine-tuning the classifier with distorted images is insufficient when the distortion is severe. For adapting the classifier to distorted images, fine-tuning often requires the classifier to be retrained. When a new classifier structure based on the DNN emerges, the classifier with this new structure also needs to be retrained for adapting to distorted images. However, retraining is a time-consuming process, and changes in the network structure occur often because of the rapid progress of DNNs.

Conventionally, several methods have been utilized to improve the performance for distorted images [10], [11]. However, these methods are insufficient when the distortion is severe; in addition, they assume that the network structure does not change. When a better performing classifier appears for a certain application, its network structure is different. To classify the distorted images accurately, this better-performing classifier needs to be trained with distorted

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

Original image Blurred image QRNet

FIGURE 1. **Quality recovered image from the blurred image. Original image (left), blurred image (center), quality recovered image (right). The quality recovered image was reconstructed using the proposed network, QRNet.**

images to address the classifier changes. Currently, there is no method for addressing classifier changes without training the classifier for distorted images.

In view of the above, we propose a new convolutional neural network (CNN) named quality recovery network (QRNet) that can recover the image quality for image classification. Figure 1 shows an example image recovered by QRNet, which is an encoder-decoder network. QRNet recovers the image quality from distorted images that include multiple distortions and various distortion intensity levels. The recovered image is then used as the classifier input. By optimizing QRNet for a classifier, QRNet can generate recovered images that are easy to classify. To realize the optimization of QRNet for a classifier, we propose a novel loss function for training QRNet. This loss function is composed of two terms: quality loss and classification loss. With the optimization of the proposed loss function, QRNet recovers the image quality and improves the classification accuracy by classifying the recovered image. QRNet can be optimized for image quality and classification accuracy.

Furthermore, we propose a quality recovery method for a new classifier. The recovered images generated by QRNet are input to the new classifier that is not used for training QRNet; i.e., the recovered images are not optimized for the new classifier. QRNet is trained with multiple classifiers except the new classifier, and this QRNet improves the accuracy of the new classifier.

The main contributions of this paper can be summarized as follows:

1) Proposal of an encoder-decoder network, QRNet, with a novel loss function that can recover the image quality from distorted images to improve the classification accuracy.
2) Extension of the proposed loss to address the changes in the network structure, and demonstration of the ease of classification with a new classifier using the recovered images generated by QRNet trained with multiple classifiers.

This paper is organized as follows: Section II discusses the related work. Section III presents the proposed QRNet based on an encoder-decoder network that considers the loss of the classifier. Section IV describes the application of our proposed method for assessing public datasets. Finally, Section V summarizes the paper and discusses the potential for future developments.

II. RELATED WORK

A. DISTORTION EFFECT ON IMAGE CLASSIFICATION

Dodge and Karam [6] investigated the influence of distortion on image classification. Their investigation demonstrated that VGG16 [2] was more robust against distortion compared to AlexNet [1] and GoogLeNet [12]. In the investigation, Gaussian noise and Gaussian blur affected the classification accuracy significantly, compared to other distortions such as contrast, JPEG, and JPEG2000 compression. Jo and Bengio [13] investigated whether neural networks learn semantic concepts. They used Fourier filtering to construct datasets and evaluated the classification errors for the datasets. Their experiments demonstrated that the recognition accuracy deteriorated considerably when high frequency components were removed. This result supports Dodge and Karam's investigation [6]. Based on these investigations, we focus on the Gaussian noise and blur in this paper.

B. CLASSIFICATION OF DISTORTED IMAGES

There are several methods to classify distorted images [10], [11]. Most studies on classifying distorted images are based on [6], and our research is also inspired by this investigation.

Zhou *et al.* [8] examined the effect of distortion and presented the effectiveness of fine-tuning and retraining with distorted images. They showed that retraining achieves higher accuracy than fine-tuning. However, fine-tuning is more practical. Vasiljevic *et al.* [9] showed that fine-tuning with a mixture of clean and blurred images can almost recover the classification performance to the original classification performance. Fine-tuning is an effective method for reducing the classification error of distorted images. However, our experiments demonstrate that fine-tuning is insufficient for images that are severely distorted.

To improve the robustness against distorted images, certain training algorithms have been proposed. Stability training [14] is a training method that renders neural networks more robust against distortion. This method trains the network with clean and perturbed images such that the output with clean images becomes closer to that with perturbed images. Stability training achieves higher performance compared to conventional neural networks.

For adversarial examples [15], BANG training [16] has been proposed. Adversarial examples are important for the security of image classification applications. To prevent adversarial attacks in image classification applications, BANG training increases the gradients of the correct samples in a minibatch when updating the weights. It can flatten the decision space around the correct samples for robust training of neural networks. Networks trained by the BANG are robust against adversarial examples. Although these training methods are useful for specific distortions, addressing severe distortions is difficult using these methods.

DeepCorrect [10] corrects filters that degrade the performance for distorted images and achieves higher performance than fine-tuning. However, this correction is not applicable to

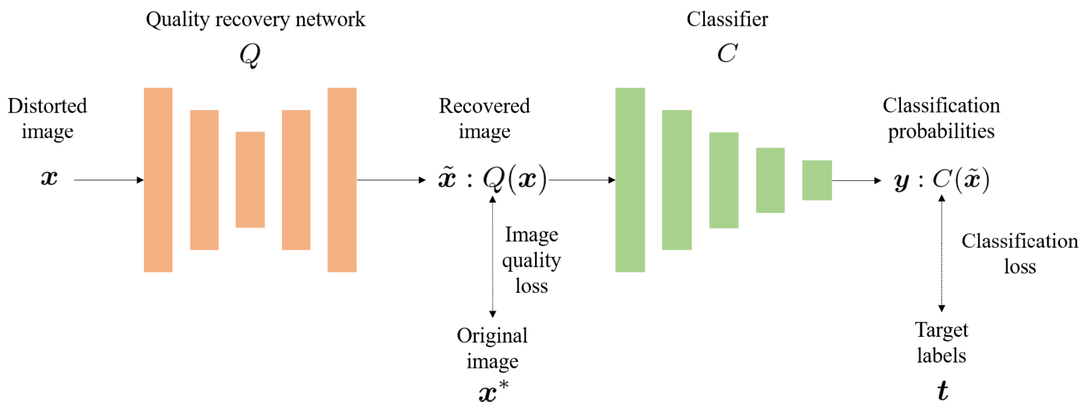


FIGURE 2. Quality recovery network (QRNet) with a classifier. QRNet generates a recovered image that is used as the classifier input. The classifier classifies the recovered image.

MixQualNets [11]. Yim and Sohn [17] proposed a dual channel model comprising two CNN models. One CNN model extracts the features from the original image, whereas the other extracts the features from a denoised image. These features are combined and input to the fully connected layer of the dual channel model. However, the denoising method has a limitation; their model requires a new preprocessing method for a new distortion.

Dodge and Karam [11] proposed MixQualNets that have network structures that can address distortion. MixQualNets have a clean network for original images and two expert networks for distorted images. These two expert networks are prepared for Gaussian noise and Gaussian blur, respectively. They also include a gating network for predicting the weights that weight the output of the clean and expert networks. MixQualNets achieve higher performance compared to a network optimized for a single distortion: they use VGG16 [2] as the base model. The structures of the clean and expert networks are similar to that of VGG16. To change the base model, these network structures must be reconstructed, and the networks must be retrained. Their model has many parameters because MixQualNets comprise four networks. Hence, their model is expensive, and requires considerable memory and time for training when the network structure is changed.

Hossain *et al.* [18] proposed a network that uses discrete cosine transform (DCT) to improve the robustness against distortion. They incorporated the proposed DCT module into the first layer of the network, which significantly improved the classification performance. However, as the DCT module is in the classification network, retraining is needed for application to classification tasks. Sun *et al.* [19] proposed the feature quantization method to enhance the robustness of neural networks against image distortion. They integrated a floor or power function into the networks and obtained good performance for various types of distortions. However, their method requires the integration of the floor or power operation into the network.

C. IMAGE TRANSFORM FOR CLASSIFICATION

Sharma *et al.* [20] proposed an image enhancement method that improves the image classification performance. Filters that enhance images were optimized with end-to-end training for classification. Although their method achieved higher performance compared to conventional neural networks, they did not consider distortion.

Palacio *et al.* [21] proposed a method that can measure and understand neural networks. They used an autoencoder that was fine-tuned with a pretrained classifier whose parameters were fixed. The pretrained classifier was trained on the ImageNet [22]. They used AlexNet [1], VGG16 [2], Inception V3 [23], and ResNet [24] as the pretrained classifiers. When training the autoencoder, the gradients were backpropagated from the classifier. They analyzed the reconstructed images and found that high performing image classifiers utilized less than 10% of the original input information. The reconstructed images generated by their method preserved the signals that were required for classification. Our study was inspired by their work. We considered their transformation method effective for recovering the signals in distorted images as well.

In this paper, we propose an encoder-decoder network that recovers the image quality for classification. Our proposed network trained with a classifier performs well with multiple distortion levels and distortion types. Moreover, our proposed network trained with multiple classifiers improves the classification accuracy for a new classifier that is not used for training the proposed network. In contrast to previous works, our proposed network can perform well even with a new classifier without retraining.

III. IMAGE QUALITY RECOVERY NETWORK

A. NETWORK STRUCTURE

Figure 2 depicts our proposed framework. QRNet Q has an encoder-decoder architecture that learns the mapping from distorted image $\boldsymbol{x}$ to recovered image $\tilde{\boldsymbol{x}}$, $Q : \boldsymbol{x} \rightarrow \tilde{\boldsymbol{x}}$. Distorted image $\boldsymbol{x}$ is generated from original image $\boldsymbol{x}^* \in \mathcal{X}$. QRNet Q is trained to output the quality recovered image. The recovered image is classified by classifier C that outputs the class probabilities $\boldsymbol{y}$. QRNet learns the network parameters with the image quality loss and classification loss. The image quality loss is calculated from distorted image $\boldsymbol{x}$ and original image $\boldsymbol{x}^*$. The classification loss is calculated from the class probabilities $\boldsymbol{y}$ and target labels $\boldsymbol{t}$ expressed by a one-hot vector. QRNet recovers the image quality from various distortions for a classifier. Most quality recovery methods deal with a single distortion or are not optimized for the classifier [25]–[29]; however, QRNet can deal with multiple distortions and various distortion intensity levels. Furthermore, the recovered image generated by QRNet can be easily classified by the classifier.

The quality recovery network structure is developed based on encoder-decoder architectures [4], [30], [31]. We adopt an encoder-decoder architecture similar to U-Net [31] that has a skip connection structure for preserving image information. Table 1 shows the structure of QRNet. QRNet consists of 34 layers, including concatenation and dropout layers. The network encoder has a repeated structure with two 3×3 convolutions and a 2×2 maxpooling operation. After convolution, we use rectified linear units (ReLU) for activation. The convolutional filters are initialized using the He initializer [32]. In contrast, the network decoder has a repeated structure with a two-dimensional (2D) transposed convolution with stride 2, a concatenation and two 3×3 convolutions. After convolution, we also use rectified linear units (ReLU) as the activation function. The output of the network is activated by the sigmoid function. The concatenation concatenates the output of the 2D transposed convolution and the output of the encoder convolution that has the same dimension as that of the 2D transposed convolution. Dropout [33] is applied for improving the generalization. To optimize the network for a classifier, the output of QRNet is used for classification. Furthermore, QRNet is optimized using the classification loss of the classifier.

We used VGG16 [2] as the classifier. As VGG16 is robust against distortion compared to the other networks such as AlexNet [1] and GoogLeNet [12], MixQualNets [11] use VGG16 as the base network. Hence, we also adopt VGG16 as the classifier to evaluate the image classification accuracy.

B. OPTIMIZATION ENCODER-DECODER NETWORK WITH SINGLE CLASSIFIER

QRNet recovers image quality from distorted images and uses these recovered images as the classifier inputs. The recovered images need to be easy to classify. Hence, our network is optimized with two losses: quality loss and classification loss. The quality loss represents the quality of the generated image by QRNet, whereas the classification loss represents the classification error.

The quality of the recovered image is expressed as the error between the original and recovered images. We use the L1 distance as the quality loss because it encourages less blurring:

$$L_q = E_{\boldsymbol{x}^*,\boldsymbol{x}}[||\boldsymbol{x}^* - Q(\boldsymbol{x})||_1] \tag{1}$$
$$= E_{\boldsymbol{x}^*,\tilde{\boldsymbol{x}}}[||\boldsymbol{x}^* - \tilde{\boldsymbol{x}}||_1]. \tag{2}$$

TABLE 1. QRNet structure.

Layer	Channels	Filter size	Stride
Conv.	16	3×3	-
Conv.	16	3×3	-
Maxpool.	-	2×2	-
Conv.	32	3×3	-
Conv.	32	3×3	-
Maxpool.	-	2×2	-
Conv.	64	3×3	-
Conv.	64	3×3	-
Maxpool.	-	2×2	-
Conv.	128	3×3	-
Conv.	128	3×3	-
Maxpool.	-	2×2	-
Conv.	256	3×3	-
Conv.	256	3×3	-
Transposed Conv.	128	2×2	2
Concat.			
Conv.	128	3×3	-
Dropout	-	-	-
Conv.	128	3×3	-
Transposed Conv.	64	2×2	2
Concat.			
Conv.	64	3×3	-
Dropout	-	-	-
Conv.	64	3×3	-
Transposed Conv.	32	2×2	2
Concat.			
Conv.	32	3×3	-
Conv.	32	3×3	-
Transposed Conv.	16	2×2	2
Concat.			
Conv.	16	3×3	-
Conv.	16	3×3	-
Conv.	3	1×1	-
Sigmoid.	-	-	-

It is expected that images with less blur will have better accuracy.

The classification loss represents the classification error that is generally expressed by the softmax cross entropy. In this study, we assume that classifier C classifies input recovered image $\tilde{\boldsymbol{x}}$ as class k using a CNN architecture such as VGG16. Therefore, we also assume that C includes the softmax layer for classification. The softmax cross entropy is calculated by the classifier. We use the softmax cross entropy as the classification loss:

$$L_r = -\sum_{k=1}^{K} t_k \log y_k \tag{3}$$

t_k is the binary label of target class k. y_k is the probability of target class k. K is the number of classes. When minibatch learning is used, equation (3) is expanded based on the mini-batch size.

For CNN optimization, we use the Adam optimizer [34]. For learning the parameters of the quality recovery network, we propose two learning methods. One method fixes the parameters of the classifier trained with the original images, while the other simultaneously learns the parameters of QRNet and the classifier trained with the original images. When the parameters of the classifier are fixed,

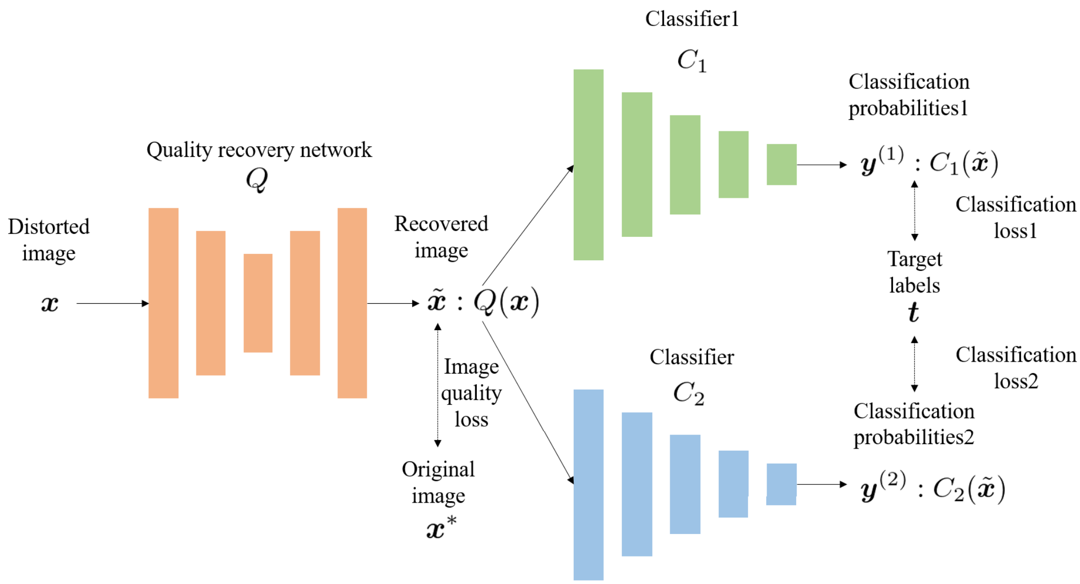


FIGURE 3. Multiple quality recovery network (MQRNet) with classifiers. MQRNet learns the network parameters with the classification losses of multiple classifiers. After learning, MQRNet generates the recovery images for the other classifiers that are not used during training.

the classifier trained with the original images is only used to obtain the classification loss for backpropagation. We train QRNet alone. When the parameters of the classifier trained with the original images are not fixed, the parameters of QRNet are trained using the image quality loss and classification loss of the classifier that is simultaneously trained; i.e., the classifier trained with the original images is trained using distorted images with QRNet. In this case, QRNet and the classifier are optimized in an end-to-end manner. We use the pretrained VGG16, trained using the ImageNet [22], as the classifier. Most of the parameters of VGG16 are fixed and used for classification. However, the three fully connected layers before the softmax layer are fine-tuned to adapt to distorted images. We train the parameters of QRNet and the fully connected layers of VGG16.

Finally, the proposed method minimizes the following equation:

$$Loss_{single} = L_q + \lambda L_r, \tag{4}$$

where λ is a hyper parameter.

C. OPTIMIZATION WITH MULTIPLE CLASSIFIERS

When there is a new classifier structure, the classifier with this new structure needs to be trained with distorted images for adapting it to such images. However, adapting the new classifier to distorted images is time-consuming. If QRNet is applied to the new classifier, the training time can be considerably reduced. To apply QRNet to the new classifier, we propose a loss for training the network with multiple classifiers. By learning the parameters of QRNet with the losses of multiple classifiers, we expect the proposed method to perform well, even for a classifier that has not been learned. Figure 3 depicts the framework for training with multiple classifiers. In this case, QRNet is trained with multiple classifiers, C_1 and C_2. The classification loss is obtained from each classifier output, $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$. When QRNet is trained with two classifiers, it requires two softmax cross entropies for optimizing the network parameters. The class probabilities of each classifier $\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}$ and the target labels $\boldsymbol{t}$, as represented in equation (3), are used for calculating the classification loss for learning the parameters of QRNet. QRNet with multiple classifiers has the same architecture as that with a single classifier. We propose a loss function as follows:

$$Loss_{multi} = L_q + \sum_{i}^{M} \lambda_i L_{C_i}, \tag{5}$$

where λ_i is a hyper parameter. L_{C_i} is the classification loss of classifier C_i. M is the number of classifiers.

In this paper, QRNet is trained with two classifiers for application to a new classifier that has not been used to train the QRNet. We call QRNet trained with multiple classifiers a multiple QRNet (MQRNet). This MQRNet is evaluated using a new classifier that is not used for training MQRNet. We use VGG16 [2] and Inception V3 [23] for training MQRNet, and use ResNet [24] as the new classifier for evaluating the classification accuracy.

D. TRAINING WITH DISTORTED IMAGES

In image classification, there are multiple distortions and various distortion intensity levels, and dealing with them is important for maintaining the classification accuracy. To address multiple distortions and various distortion intensity levels, we train QRNet with multiple distorted images.

Algorithm 1 Training With Distorted Images

Input: $\boldsymbol{b}_{\boldsymbol{x}^*} \in \mathcal{X}, \boldsymbol{b}_{\boldsymbol{t}} \in \mathcal{T}, Q, C$
for all $\boldsymbol{b}_{\boldsymbol{x}^*} \in \mathcal{X}$ **do**
  Select distortion d randomly from $\mathcal{D}$.
  Select parameter $\boldsymbol{p}$ randomly from $\mathcal{P}_d$.
  $\boldsymbol{b}_{\boldsymbol{x}} \leftarrow G(\boldsymbol{b}_{\boldsymbol{x}^*}, d, \boldsymbol{p})$
  $\boldsymbol{b}_{\boldsymbol{x}_{train}} \leftarrow concatenate(\boldsymbol{b}_{\boldsymbol{x}^*}, \boldsymbol{b}_{\boldsymbol{x}})$
  $\boldsymbol{b}_{\boldsymbol{x}_{target}} \leftarrow concatenate(\boldsymbol{b}_{\boldsymbol{x}^*}, \boldsymbol{b}_{\boldsymbol{x}^*})$
  $L_q \leftarrow D(Q(\boldsymbol{b}_{\boldsymbol{x}_{train}}), \boldsymbol{b}_{\boldsymbol{x}_{target}})$
  $\boldsymbol{b}_{\boldsymbol{t}_{target}} \leftarrow concatenate(\boldsymbol{b}_{\boldsymbol{t}}, \boldsymbol{b}_{\boldsymbol{t}})$
  $L_r \leftarrow S(C(\boldsymbol{b}_{\boldsymbol{x}_{train}}), \boldsymbol{b}_{\boldsymbol{t}_{target}})$
  $Q \leftarrow BackProp(L_q, L_r, C)$
end for

Initially, QRNet is pretrained with the original images that are not distorted. It is then trained with the original and multiple distorted images. During training, we use a batch that contains an equal number of original and distorted images.

Algorithm 1 shows the training procedure for multiple distortions when fixing the classifier parameters. Batch $\boldsymbol{b}_{\boldsymbol{x}^*}$ is first extracted from original images $\mathcal{X}$. Distortion $d \in \mathcal{D}$ is then randomly added to $\boldsymbol{b}_{\boldsymbol{x}^*}$. $\mathcal{D}$ is the set of distortion types including Gaussian noise and blur. The distortion parameter $\boldsymbol{p} \in \mathcal{P}_d$ is also selected randomly. $\mathcal{P}_d$ is the set of distortion parameters for distortion d. When Gaussian noise is selected, $\boldsymbol{p}$ is the mean and standard deviation pair. In this paper, $\boldsymbol{p}$ is selected among five distortion levels for each distortion. In each batch, the distortion is randomly selected. A distorted batch $\boldsymbol{b}_{\boldsymbol{x}}$ is generated by a distorted image generator G. The inputs for batch training $\boldsymbol{b}_{\boldsymbol{x}_{train}}$ are generated by concatenating the original batch $\boldsymbol{b}_{\boldsymbol{x}^*}$ and distorted batch $\boldsymbol{b}_{\boldsymbol{x}}$. The target images $\boldsymbol{b}_{\boldsymbol{x}_{target}}$ for reconstruction are generated by concatenating the target images of the original and distorted batches. QRNet is trained using the concatenated batches $\boldsymbol{b}_{\boldsymbol{x}_{train}}, \boldsymbol{b}_{\boldsymbol{x}_{target}}$ by backpropagation with the proposed image quality loss. To obtain the image quality loss L_q, D calculates the loss between the recovered images $Q(\boldsymbol{b}_{\boldsymbol{x}_{train}})$ and $\boldsymbol{b}_{\boldsymbol{x}_{target}}$ based on equation (2). The target labels of batch $\boldsymbol{b}_{\boldsymbol{t}} \in \mathcal{T}$ corresponding to $\boldsymbol{b}_{\boldsymbol{x}^*}$ are also concatenated. $\mathcal{T}$ is a set of target labels. The concatenated target labels $\boldsymbol{b}_{\boldsymbol{t}_{target}}$ are used for calculating the classification error with classifier C, and the gradients are backpropagated based on the classification loss. To obtain the classification loss L_r, S calculates the loss between the recovered images $C(\boldsymbol{b}_{\boldsymbol{x}_{train}})$ and $\boldsymbol{b}_{\boldsymbol{t}_{target}}$ based on equation (3). Finally, QRNet Q is updated using backpropagation with losses L_q and L_r. The classification loss L_r is backpropagated from classifier C to QRNet Q.

By randomly selecting the distortion type and parameter, QRNet is trained robustly against distortion. Furthermore, using the original and distorted images simultaneously as a batch, it is possible to realize distortion restoration without degrading the reconstruction performance of the original image.

IV. RESULTS

A. DATASETS AND PARAMETERS

The proposed method is evaluated using the Caltech101 [35] and Caltech256 [36]. We randomly split the data as 80% for training and 20% for testing. We use 20% of the training data for validation. We divide the data into these datasets once.

In this paper, we consider two distortion types, Gaussian noise and Gaussian blur. These distortions severely affect CNNs [5]. We add random Gaussian noise to each pixel. The standard deviation of the Gaussian noise σ_n ranges between (0, 100]. The standard deviation of the Gaussian blur σ_b ranges between (0, 10]. The kernel size of the Gaussian blur is defined as four times minus one of σ_b. For evaluation, we set the parameters at five levels. For the Gaussian noise, we use $\{20, 40, 60, 80, 100\}$ as σ_n. For the Gaussian blur, we use $\{2, 4, 6, 8, 10\}$ as σ_b. Therefore, the kernel size of Gaussian blur is $\{7, 15, 23, 31, 39\}$.

In this paper, we use Adam [34] to optimize the networks. We set the learning rate as 0.001, dropout rate as 0.2, batch size as 32, and the maximum number of epochs as 200. In each epoch, we use all the training data with distortions and use early stopping when the validation loss has not been updated after five epochs.

In equation (4), we evaluated different λ values $\{0.1, 0.01, 0.001\}$ on the validation data; however, we obtained similar results. Hence, $\lambda = 0.01$ is used in equation (4). In equation (5), we apply VGG16 and Inception V3 as classifiers with MQRNet to learn the parameters. We use the same value for λ_1 and λ_2 as λ. λ_1 for VGG16 is set to 0.01. λ_2 for Inception V3 is also set to 0.01.

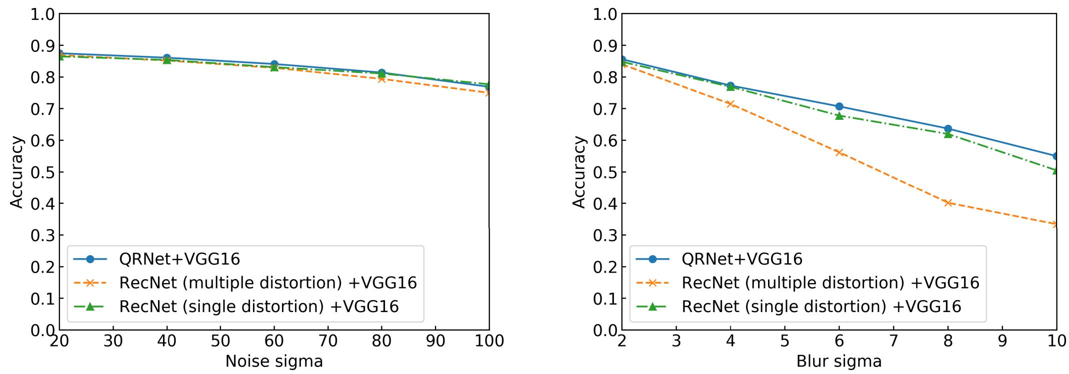
B. PERFORMANCE FOR DISTORTED IMAGES

1) RECOVERY PERFORMANCE FOR DISTORTED IMAGES

Table 2 shows the performance comparison with respect to distortion restoration in terms of the classification accuracy on the Caltech datasets. "VGG16" indicates the classification accuracy of VGG16 trained with the original images that classifies distorted images without reconstruction. A reconstruction network (RecNet) is used for comparison. "RecNet (single) +VGG16" and "RecNet (multiple) +VGG16" indicate the classification accuracy of VGG16 trained with the original images that classifies the recovered images reconstructed by RecNets. "RecNet (single)" and "RecNet (multiple)", which have the same structure as QRNet, are used as the reconstruction networks and trained in two ways. "RecNet (single)" is trained with single distortion images. Two RecNets are trained for two distortion types, respectively. "RecNet (multiple)" is trained with multiple distortion images. We train RecNet using a procedure similar to that mentioned in Section III-D. "RecNet (single)" is trained with one distortion type and various distortion intensity levels that are selected randomly in each batch. "RecNet (multiple)" is trained with two distortion types and various distortion intensity levels that are selected randomly in each batch. These RecNets are trained with the L1 distance alone, and do

TABLE 2. Recovery performance for distorted images.

Dataset	Distortion Types	VGG16	RecNet (single) + VGG16	RecNet (multiple) + VGG16	QRNet + VGG16
Caltech101	Gaussian Noise	0.642	0.828	0.819	**0.832**
	Gaussian Blur	0.259	0.684	0.571	**0.705**
	Average	0.450	0.756	0.694	**0.768**
Caltech256	Gaussian Noise	0.346	0.496	0.473	**0.521**
	Gaussian Blur	0.098	0.328	0.242	**0.346**
	Average	0.222	0.412	0.357	**0.434**



(a) Gaussian noise image (b) Blurred image

FIGURE 4. Recovery performance at each distortion level on the Caltech101 dataset.

not apply the classification error for learning the parameters. "QRNet+VGG16" indicates the classification accuracy of VGG16 trained with the original images that classifies the recovered images reconstructed by QRNet. QRNet is trained using the procedure mentioned in Section III-D.

We recover the images from the distorted images using RecNets and QRNet. The recovered images are then classified by VGG16 trained with the original images. The result of VGG16 for distorted images is poor. With Gaussian blur, in particular, the accuracy of the original VGG16 is 0.259 for the Caltech101. After recovering the images using QRNet, the accuracy is 0.705, as indicated in Table 2; the accuracy improves by 0.446 compared to that of VGG16 without reconstruction. The results of RecNets also show improvements in the accuracy. However, for the Caltech256, the results of RecNet trained with multiple distortions tend to be less accurate compared to those of RecNet trained with a single distortion. The accuracy achieved by QRNet is higher than those of RecNets. Thus, QRNet with our proposed loss function can generate images that improve the image classification accuracy. In addition, QRNet can recover the image quality from multiple distortions regardless of the dataset.

In our experiments, we used a GeForce GTX TITAN X to evaluate our model. QRNet with VGG16 takes 167 s per epoch for training on the Caltech101 dataset. This includes the time required to add distortions. In addition, QRNet with VGG16 takes 8.29 ms per image to classify a distorted image.

TABLE 3. Performance of small QRNet on the Caltech101 dataset.

Gaussian noise	Gaussian blur	Average
0.828	0.694	0.761

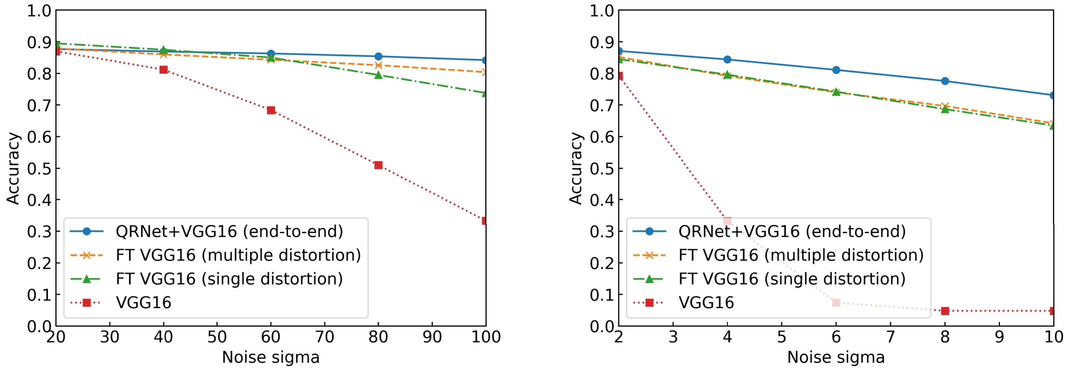
In our environment, VGG16 takes 5.51 ms per image to classify a distorted image. Although QRNet requires more time for inference than VGG16, it significantly improves accuracy for distorted images.

Figure 4 shows the recovery performance at each distortion level on the Caltech101 dataset. Although the accuracy of VGG16 with RecNet trained with multiple distortions is lower than that of VGG16 with RecNet trained with a single distortion at high distortion levels, the former maintains the accuracy. The accuracy of VGG16 with RecNet trained with a single distortion is low compared to that of QRNet trained with multiple distortions at high distortion levels. Thus, QRNet is robust against multiple distortions at high distortion levels.

To evaluate the depth of the network, we constructed a small QRNet. Table 3 shows the accuracy of small QRNet. As mentioned in Section III-A, QRNet has two 3×3 convolutions on the encoder and decoder sides. We changed the two 3×3 convolutions to one 3×3 convolution. This QRNet has a smaller structure than the proposed QRNet. We call this small QRNet. Small QRNet is slightly less accurate than the proposed QRNet. This indicates that the depth of the

TABLE 4. Performance of QRNet with end-to-end learning.

Dataset	Distortion Types	VGG16	Fine-tuned VGG16 (single)	Fine-tuned VGG16 (multiple)	QRNet + VGG16 (end-to-end)
Caltech101	Gaussian Noise	0.642	0.831	0.842	**0.861**
	Gaussian Blur	0.259	0.741	0.745	**0.807**
	Average	0.450	0.786	0.793	**0.834**
Caltech256	Gaussian Noise	0.346	0.545	0.533	**0.582**
	Gaussian Blur	0.098	0.402	0.414	**0.497**
	Average	0.222	0.474	0.473	**0.539**



(a) Gaussian noise image (b) Gaussian blurred image

FIGURE 5. Performance of the end-to-end framework on the Caltech101 dataset.

network could be reduce. However, the proposed QRNet is more accurate. Hence, in practice, it would be desirable to adjust the network structure in terms of speed and memory.

2) PERFORMANCE WITH END-TO-END LEARNING

Table 4 lists the performance of QRNet with the fine-tuning framework and the proposed end-to-end learning framework. We compare the performance of the fine-tuned classifiers and the classifier trained with end-to-end learning for distorted images. "Fine-tuned VGG16 (single)" indicates VGG16 trained with single distortion images. "Fine-tuned VGG16 (multiple)" indicates VGG16 trained with multiple distortion images. These VGG16s are fine-tuned with a batch that concatenates the original and distorted images. "QRNet + VGG16 (end-to-end)" indicates QRNet with VGG16. QRNet is trained by fine-tuning VGG16 simultaneously. The performance of the fine-tuned classifier with a single distortion is not always better than that with multiple distortions. In most cases, the performance of the fine-tuned classifier with multiple distortions is better compared to that with a single distortion.

For both datasets, QRNet with end-to-end learning performs well for blurred images in particular. Although blur has a strong influence on the CNNs [6], QRNet can recover the image quality for image classification and improve the accuracy. In our proposed loss function, the classification error is backpropagated from the classifier, which is effective in recovering signals that enable better image classification.

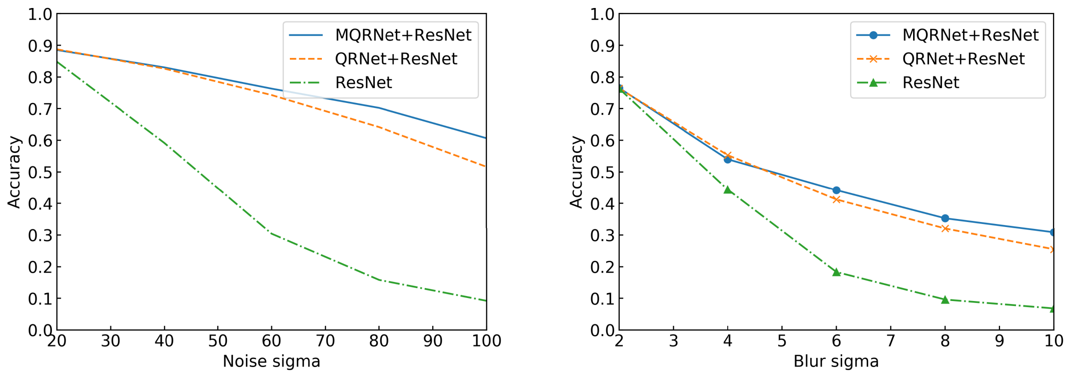
In end-to-end learning, QRNet with a VGG16 that is able to be fine-tuned takes almost the same time for training and inference as QRNet with a VGG16 that has fixed parameters, as mentioned in Section IV-B1.

Figure 5 shows the performance of the end-to-end framework for the Caltech101 at each distortion level. The accuracies of the fine-tuned classifiers ("FT VGG16") with a single distortion and multiple distortions are better compared to the model trained with the original images alone. However, their accuracies are lower than that of QRNet with end-to-end learning at high distortion levels. The accuracy of QRNet with end-to-end learning is good for both Gaussian noise and Gaussian blurred images. Thus, our proposed method with end-to-end learning is more effective than the fine-tuning method.

C. ROBUSTNESS AGAINST CLASSIFIER CHANGE

We evaluate the robustness of the proposed method against classifier changes. We train MQRNet with VGG16 and the Inception V3, as per equation (5). MQRnet is trained using the proposed loss composed of multiple terms: the quality loss and classification losses of the classifiers (VGG16 and inception V3). MQRNet recovers image quality from distorted images, and the recovered images are input to an evaluation network for classification. We use ResNet as the evaluation network; it was trained using the original images.

Figure 6 shows the performance of MQRNet for the Caltech101 dataset. "ResNet" indicates the performance of



(a) Gaussian noise image (b) Blurred image

FIGURE 6. Performance of Multiple QRNet at each distortion level on the Caltech101 dataset.

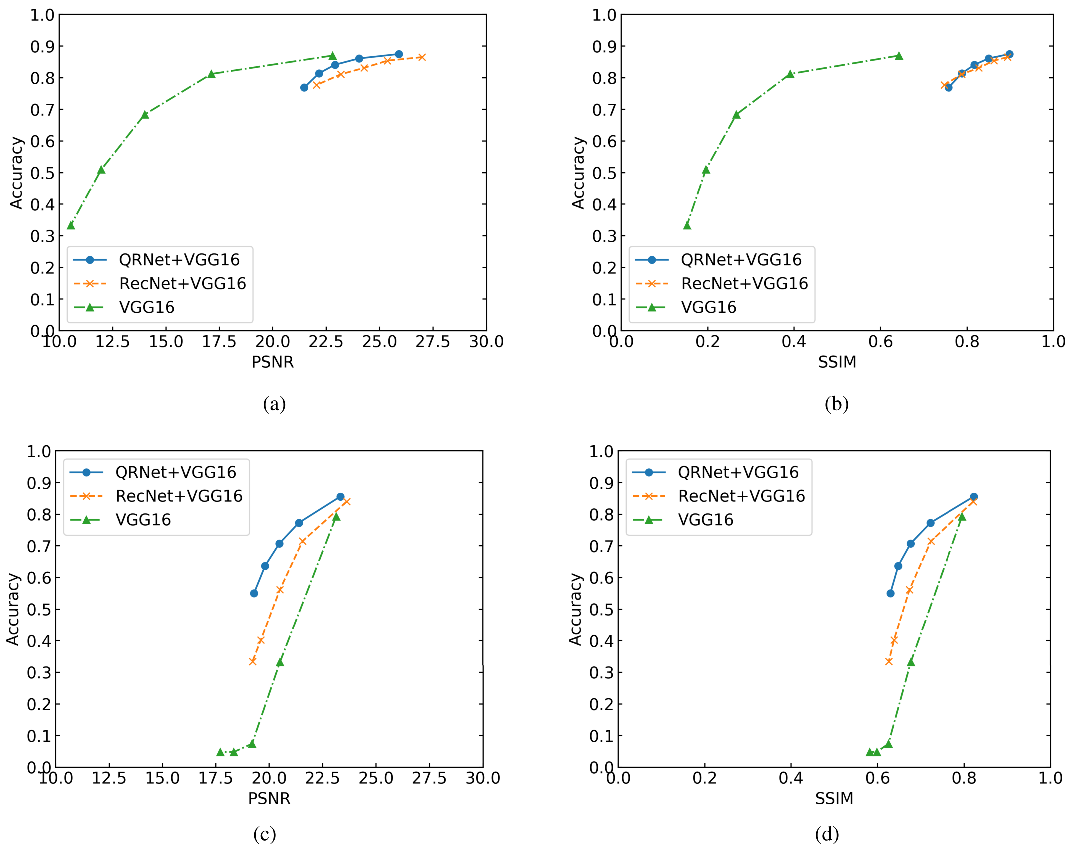


FIGURE 7. Relationship between the image quality metrics and QRNet performance at each distortion level on the Caltech101 dataset: (a) Relationship between the PSNR and accuracy for Gaussian noise images, (b) Relationship between the SSIM and accuracy for Gaussian noise images, (c) Relationship between the PSNR and accuracy for Gaussian blurred images, and (d) Relationship between the SSIM and accuracy for Gaussian blurred images.

ResNet whose input is the original image. "QRNet+ResNet" indicates the performance of ResNet whose input is the recovered image reconstructed by QRNet. QRNet is trained with VGG16 alone. "MQRNet+ResNet" indicates the performance of ResNet whose input is the recovered image reconstructed by MQRNet. MQRNet is trained with VGG16 and Inception V3. Figure 6 (a) shows the results for Gaussian noise images, whereas Figure 6 (b) shows the results for Gaussian blurred images. For both distortions, MQRNet performs well at high distortion levels. Although QRNet is trained with VGG16 alone, it is effective for network changes. The performance of ResNet is poor for severe distortion. The performance of ResNet whose input is the recovered image by QRNet is significantly better than that of ResNet whose

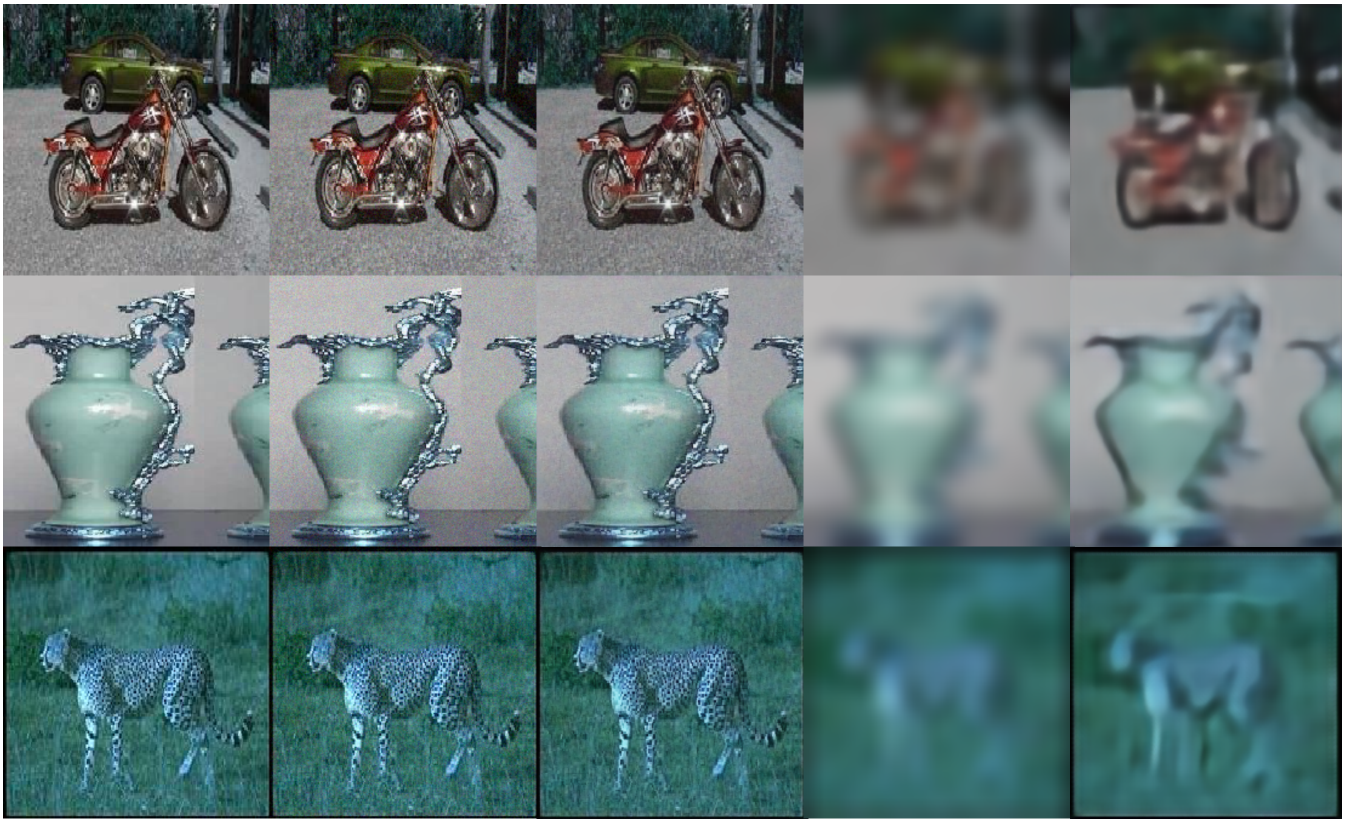


FIGURE 8. Samples of the quality recovered images from the distorted images. The quality recovered images are reconstructed using the proposed network, QRNet.

input is the original image. Furthermore, the performance of ResNet whose input is the recovered image by MQRNet is better than that of ResNet whose input is the recovered image by QRNet. As MQRNet is trained with multiple classifiers, it maintains the accuracy even with classifier changes. In addition, ResNet is not fine-tuned for distorted images. This means that MQRNet requires no time for training a classifier with distorted images.

D. RELATIONSHIP BETWEEN THE IMAGE QUALITY AND CLASSIFICATION ACCURACY

Although Gaussian noise and Gaussian blur reduce the classification accuracy, it is not clear how they actually affect CNNs. Images can be analyzed using the image quality metrics. Humans can easily classify objects if the image quality is high. Therefore, we measure the relationship between the image quality metrics and classification accuracy. Recently, a no-reference image quality metric based on CNN was proposed [37]. This metric represents the subjective image quality. However, the accuracy of this metric depends on training datasets, and we can use original images to calculate full-reference image quality metrics. Full-reference image quality metrics have a high correlation with the subjective image quality. Hence, we use the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [38] as the image quality metrics.

Figure 7 shows the relationship between the image quality metrics and the classification accuracy. The classification accuracy of each line is the classification result with the same classifier. Figures 7 (a) and 7 (b) show the relationship between the image quality metrics and the classification accuracy for Gaussian noise images on the Caltech101 dataset. VGG16 whose parameters are learned from the original images is used as the classifier. "QRNet+VGG16" indicates the classification result with reconstruction by the proposed QRNet. "RecNet+VGG16" indicates the classification result with reconstruction by RecNet. "VGG16" indicates the classification result without reconstruction. The accuracy of the classifier that classifies the recovered images generated by QRNet is higher than that of the classifier without reconstruction even if the PSNR is low. Therefore, QRNet recovers image quality and accuracy simultaneously. In Figure 7 (a), the relationship between the PSNR and classification accuracy is not linear. The relationship between the PSNR and classification accuracy is different for each classifier. This shows that the PSNR is not correctly mapped to the classification accuracy. In addition, the quality of the

TABLE 5. Performance of QRNet in recovering the image quality on the Caltech101 dataset.

Metrics	Gaussian noise		Gaussian blur	
	distorted image	recovered image	distorted image	recovered image
PSNR	15.29	**23.30**	19.77	**20.85**
SSIM	0.330	**0.823**	0.656	**0.700**

recovered images generated by QRNet and RecNet has a similar relationship between the image quality and classification accuracy in terms of the SSIM. However, the relationship between the SSIM and the classification accuracy of VGG16 without reconstruction is different from this relationship. Thus, the SSIM is also not correctly mapped to the classification accuracy.

Figures 7 (c) and 7 (d) show the relationship between image quality metrics and classification accuracy for Gaussian blurred images on the Caltech101 dataset. Gaussian blur affects the classification accuracy more than Gaussian noise. However, VGG16 with QRNet maintains higher accuracy compared to VGG16 without reconstruction even when the image quality is low. Furthermore, VGG16 with QRNet maintains higher accuracy compared to VGG16 with RecNet whose accuracy is degraded by low image quality. This shows that QRNet is robust with respect to the image quality in terms of the classification accuracy. With the same image quality as that of the recovered images generated by QRNet, the recovered images generated by RecNet are classified with low classification accuracy for Gaussian blurred images. Therefore, the image quality metrics are also not correctly mapped to the classification accuracy for Gaussian blurred images.

Table 5 shows the image quality metrics of the distorted images and the recovered images generated by QRNet. For Gaussian noise, QRNet improves the image quality considerably by appropriately removing noise. In addition, for Gaussian blur, QRNet improves the image quality. However, the degree of improvement is less compared to that of Gaussian noise. With Gaussian blur, it is difficult to recover the image quality. However, QRNet recovers image quality suitably for classifying blurred images.

Figure 8 shows samples of the recovered images by QRNet. Gaussian noise is removed, and the images are smoother than the distorted images. The edges of the object that are important for classification remain clear. For Gaussian blur, QRNet recovers the edges of the objects. The edges of the recovered images generated by QRNet are enhanced compared to those of the distorted images. Thus, QRNet performs denoising, and recovers the image quality without losing the information for classification. Moreover, QRNet enhances the signals such that the edges considered important for classification remain clear.

V. CONCLUSION

We have proposed QRNet that recovers the image quality for image classification and have demonstrated that the proposed QRNet improves the image classification accuracy by classifying these recovered images using the Caltech database. QRNet is trained with the proposed loss function that includes two terms: image quality loss and classification loss. The parameters of QRNet are optimized with these losses, and QRNet recovers the image quality of distorted images for image classification. Using these recovered images for classification, the classification accuracy is improved. Moreover, QRNet trained with multiple classification losses from multiple classifiers can improve the accuracy of a new classifier that is not used for training QRNet. By learning from the losses of multiple classifiers, our model improves the performance for a new classifier.

In this paper, we focused on the image classification task. However, by backpropagating the recognition loss to QRNet, our method can be easily extended to other tasks such as object detection [39], semantic segmentation [31], and instance segmentation [40]. Furthermore, by including adversarial examples for training, there is the possibility of preventing adversarial attacks to realize secure image classification applications. Because an index that represents the image quality of the classifier is necessary, we intend to investigate the image quality metric in our future study.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015, pp. 1–14.

[3] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[5] S. Dodge and L. J. Karam, "Understanding how image quality affects deep neural networks," in *Proc. Int. Conf. Quality Multimedia Exper. (QoMEX)*, Jun. 2016, pp. 1–6.

[6] S. Dodge and L. J. Karam, "A study and comparison of human and deep learning recognition performance under visual distortions," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul./Aug. 2017, pp. 1–7.

[7] S. Dodge and L. J. Karam, "Human and DNN classification performance on images with quality distortions: A comparative study," *ACM Trans. Appl. Percept.*, vol. 16, no. 2, Mar. 2019, Art. no. 7.

[8] Y. Zhou, S. Song, and N.-M. Cheung, "On classification of distorted images with deep convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 1213–1217.

[9] I. Vasiljevic, A. Chakrabarti, and G. Shakhnarovich, "Examining the impact of blur on recognition by convolutional networks," Nov. 2016, *arXiv:1611.05760*. [Online]. Available:https://arxiv.org/abs/1611.05760

[10] T. S. Borkar and L. J. Karam, "DeepCorrect: Correcting DNN models against image distortions," May 2017, *arXiv:1705.02406*. [Online]. Available: https://arxiv.org/abs/1705.02406

[11] S. F. Dodge and L. J. Karam, "Quality robust mixtures of deep neural networks," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5553–5562, Nov. 2018.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[13] J. Jo and Y. Bengio, "Measuring the tendency of CNNs to learn surface statistical regularities," Nov. 2017, *arXiv:1711.11561*. [Online]. Available: https://arxiv.org/abs/1711.11561

[14] S. Zheng, Y. Song, T. Leung, and I. J. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4480–4488.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2014, pp. 1–10.

[16] A. Rozsa and M. Günther, and T. E. Boult, "Towards robust deep neural networks with BANG," in *Proc. IEEE Win. Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 803–811.

[17] J. Yim and K.-A. Sohn, "Enhancing the performance of convolutional neural networks on quality degraded datasets," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov./Dec. 2017, pp. 1–8.

[18] M. T. Hossain, S. W. Teng, D. Zhang, S. Lim, and G. Lu, "Distortion robust image classification using deep convolutional neural network with discrete cosine transform," Nov. 2018. *arXiv:1811.05819*. [Online]. Available: https://arxiv.org/abs/1811.05819

[19] Z. Sun, M. Ozay, Y. Zhang, X. Liu, and T. Okatani, "Feature quantization for defending against distortion of images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7957–7966.

[20] V. Sharma, A. Diba, D. Neven, M. S. Brown, L. Van Gool, and R. Stiefelhagen, "Classification-driven dynamic image enhancement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4033–4041.

[21] S. Palacio, J. Folz, J. Hees, F. Raue, D. Borth, and A. Dengel, "What do deep networks like to See?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3108–3117.

[22] 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.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[25] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2016, pp. 2802–2810.

[26] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning image restoration without clean data," Mar. 2018, *arXiv:1803.04189*. [Online]. Available: https://arxiv.org/abs/1803.04189

[27] B. N. Narayanan, R. C. Hardie, and E. J. Balster, "Multiframe adaptive Wiener filter super-resolution with JPEG2000-compressed images," *EURASIP J. Adv. Signal Process.*, no. 1, Apr. 2014, Art. no. 55.

[28] H. R. Shahdoosti and S. M. Hazavei, "Combined ripplet and total variation image denoising methods using twin support vector machines," *Multimed. Tools Appl.*, vol. 77, no. 6, pp. 7013–7031, Mar. 2018.

[29] H. R. Shahdoosti and Z. Rahemi, "Edge-preserving image denoising using a deep convolutional neural network," *Signal Process.*, vol. 159, pp. 20–32, Jun. 2019.

[30] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2014, pp. 1–14.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention* (Lecture Notes in Computer Science), vol. 9351. Cham, Switzerland: Springer, Oct. 2015, pp. 234–241.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015, pp. 1–15.

[35] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop (CVPR)*, Jun. 2004, p. 178.

[36] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

[37] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1733–1740.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[39] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2015, pp. 91–99.

[40] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

MOTOHIRO TAKAGI (M'14) received the B.E. and M.E. degrees in administration engineering from Keio University, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree in information and computer science with Keio University.

He joined Nippon Telegraph and Telephone Corporation (NTT), in 2011. He is a Researcher with the NTT Media Intelligence Laboratories. His research interests include machine intelligence, machine learning, computer vision, and natural language processing.

AKITO SAKURAI received the B.E. and M.E. degrees in mathematical engineering and information engineering from The University of Tokyo, Tokyo, Japan, in 1975 and 1977, respectively, the M.S. degree in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1999, and the Ph.D. degree in engineering from The University of Tokyo.

From 1977 to 1998, he was with Hitachi Ltd. Then, he moved to the Japan Advanced Institute of Science and Technology, in 1998. He served as a Professor with the School of Knowledge Science. From 2001 to 2018, he was a Professor with Keio University, when he has retired and been appointed as a Professor Emeritus. He is currently an IAS Professor with the Institute of Advanced Sciences, Yokohama National University. His research interests include a wide range of topics from theoretical aspects to applications of machine learning and specifically neural networks.

MASAFUMI HAGIWARA (M'88–SM'05) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1982, 1984, and 1987, respectively.

Since 1987, he has been with Keio University, where he is currently a Professor. From 1991 to 1993, he was a Visiting Scholar with Stanford University. His research interests include neural networks, fuzzy systems, and affective engineering.

He is a member of IEICE, IPSJ, JSAI, SOFT, IEE of Japan, Japan Society of Kansei Engineering, and JNNS. He received the IEEE Consumer Electronics Society Chester Sall Award, in 1990, Author Award from the Japan Society of Fuzzy Theory and Systems, in 1996, Technical Award and Paper Awards from Japan Society of Kansei Engineering, in 2003, 2004, and 2014, Best Research Award from Japanese Neural Network Society, in 2013, and the Best Paper Award from SOFT, in 2018. He was the President of the Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT).