An Efficient Image Compression Method Based On Neural Network: An Overfitting Approach

Yu Mikami, Chihiro Tsutake, Keita Takahashi, Toshiaki Fujii

Nagoya University, Japan

Thank you for watching this video.

An Efficient Image Compression Method Based On Neural Network: An Overfitting Approach

Yu Mikami, Chihiro Tsutake, Keita Takahashi, Toshiaki Fujii

Nagoya University, Japan

In this video, we'd like to talk on "An Efficient Image Compression Method Based On Neural Network: An Overfitting Approach".

Our achievement

Fujii Laboratory

Image compression performance:
comparable to conventional neural network-based methods
using overfitting method, while reducing network size

Two contributions

1. Formulation of loss function including entropy of network parameters
2. Construction of encoding method for network parameters

First of all, we'd like to show our achievement.

Our achievement

Fujii Laboratory

Image compression performance:
comparable to conventional neural network-based methods
using overfitting method, while reducing network size

Two contributions

1. Formulation of loss function including entropy of network parameters
2. Construction of encoding method for network parameters

We achieved the image compression performance comparable to conventional neural network-based methods using the overfitting method, while reducing the network size.

Our achievement

Image compression performance:
comparable to conventional neural network-based methods
using overfitting method, while reducing network size

Two contributions

1. Formulation of loss function including entropy of network parameters
2. Construction of encoding method for network parameters

We formulated the loss function including the entropy of the network parameters,
and we constructed an encoding method for the network parameters.

Background

Then, we'd like to introduce the background of our study.

Image compression methods

Fujii Laboratory

Past

Present

- DCT-based method
  Ex) JPEG [1] (1992), BPG (2014)

- Wavelet-based method
  Ex) JPEG 2000 (1997~)

- Neural network-based method
  Ex) Autoencoder [2] (2006~)

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1 (1992)
[2] G. E. Hinton et.al., "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507 (2006)

Image compression is a fundamental task in image processing and is used in a variety of applications.

Image compression methods

Fujii Laboratory

Past

Present

- DCT-based method
  Ex) JPEG [1] (1992), BPG (2014)
- Wavelet-based method
  Ex) JPEG 2000 (1997~)
- Neural network-based method
  Ex) Autoencoder [2] (2006~)

[1] G. K. Wallace, “The JPEG still picture compression standard,” *IEEE Trans. Consum. Electron.*, vol. 38, no. 1 (1992)
[2] G. E. Hinton et.al., “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507 (2006)

There are various image compression methods, typical of which are DCT-based and Wavelet-based methods.

Image compression methods

Fujii Laboratory

Past

Present

- DCT-based method
  Ex) JPEG [1] (1992), BPG (2014)

- Wavelet-based method
  Ex) JPEG 2000 (1997~)

- Neural network-based method
  Ex) Autoencoder [2] (2006~)

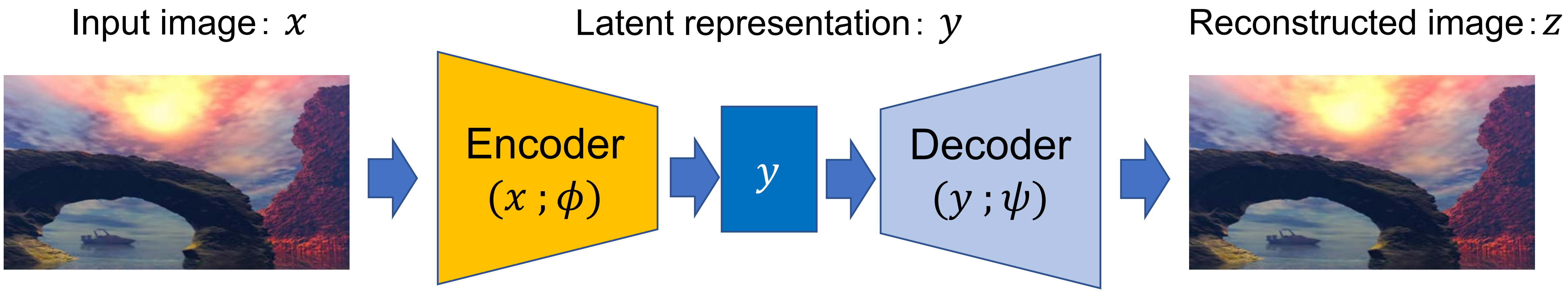
[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1 (1992)
[2] G. E. Hinton et.al., "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507 (2006)

Recently, neural network-based methods such as autoencoder have attracted a lot of attention. We focus on this method using an autoencoder.

Autoencoder-based method [3,4]

Fujii Laboratory

Outline of autoencoder-based method



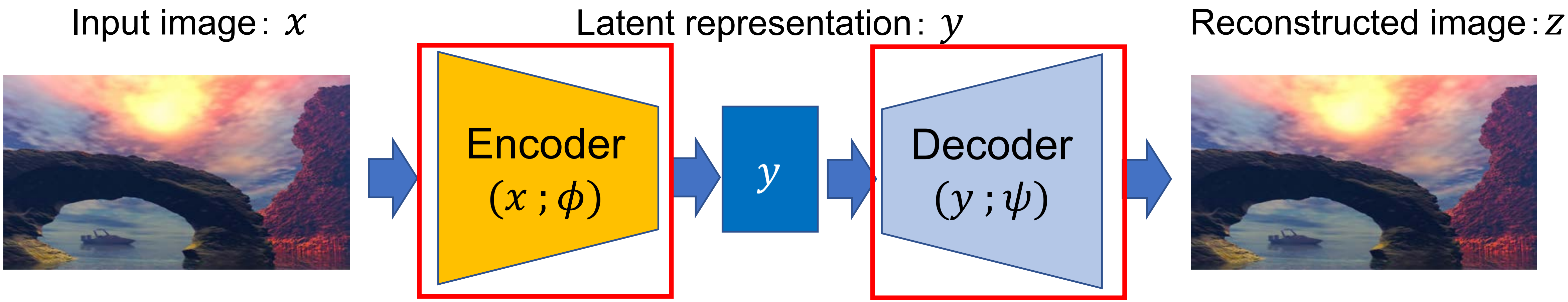
- Input image (x): converted to reconstructed image (z) through Encoder and Decoder
- Latent representation (y): compressed representation of input image (x)

[3] F. Mentzer et.al., "Conditional probability models for deep image compression," *CVPR*, (2018)
[4] J. Balle et.al., "End-to-end optimized image compression", ICLR, (2017)

We'd like to explain the autoencoder for the neural-network-based image compression method.

Autoencoder-based method [3,4]

Outline of autoencoder-based method



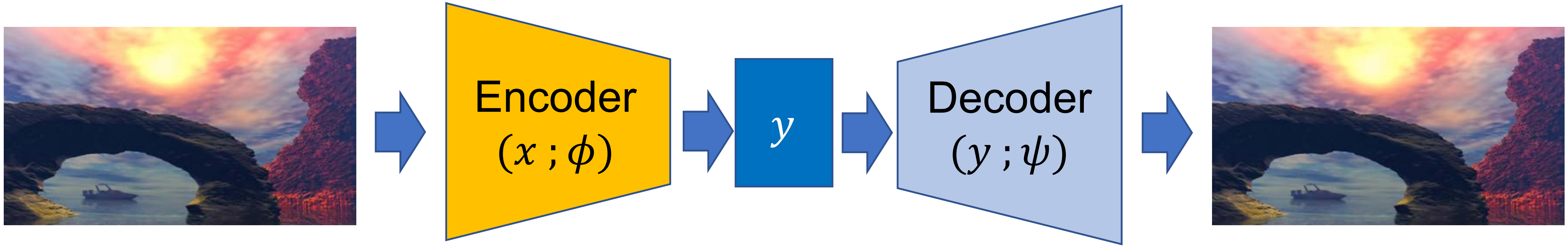
- Input image (x): converted to reconstructed image (z) through Encoder and Decoder
- Latent representation (y): compressed representation of input image (x)

[3] F. Mentzer et.al., "Conditional probability models for deep image compression," *CVPR*, (2018)
[4] J. Balle et.al., "End-to-end optimized image compression", ICLR, (2017)

The autoencoder is a network including two parts: an encoder and a decoder.

Autoencoder-based method [3,4]

Outline of autoencoder-based method



- Input image (x): converted to reconstructed image (z) through Encoder and Decoder
- Latent representation (y): compressed representation of input image (x)

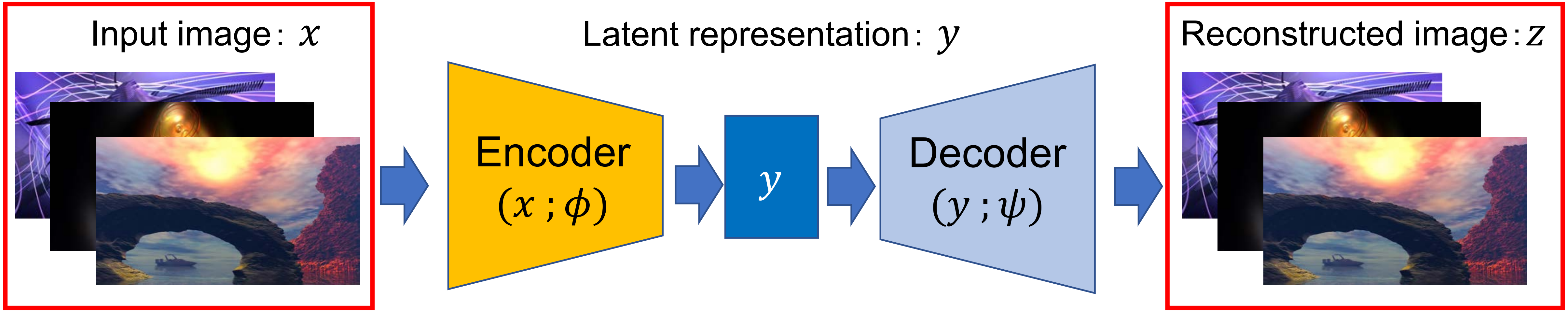
[3] F. Mentzer et.al., "Conditional probability models for deep image compression," *CVPR*, (2018)
[4] J. Balle et.al., "End-to-end optimized image compression", ICLR, (2017)

At an encoder, the input image is first converted to a latent representation that has a small data size. Then it is transformed to the reconstructed image z at a decoder.

Autoencoder-based method [3,4]

Fujii Laboratory

Training of autoencoder



Optimized for large number of images (Data sets : χ)

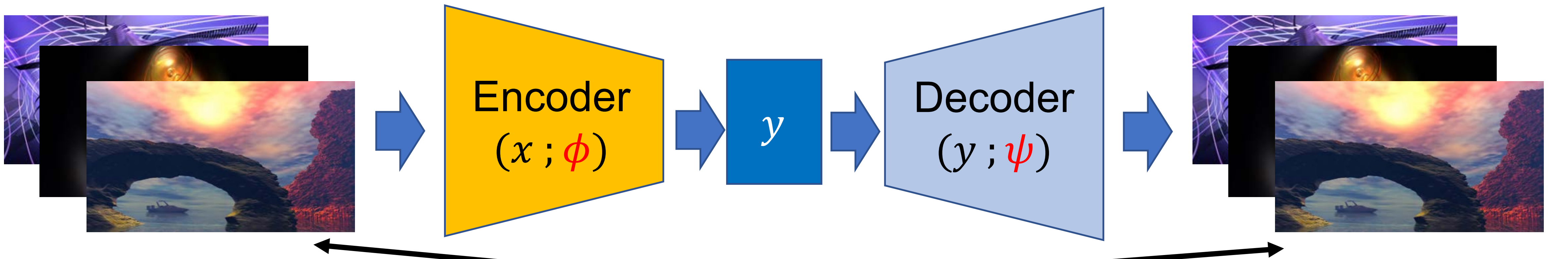
[3] F. Mentzer et.al., “Conditional probability models for deep image compression,” *CVPR*, (2018)
[4] J. Balle et.al., “End-to-end optimized image compression”, ICLR, (2017)

Next, we explain the training procedure for an autoencoder.
In the training, the autoencoder is trained using a large number of images.

Autoencoder-based method [3,4]

Training of autoencoder

Input image: x Latent representation: y Reconstructed image: z



$$\min \|x - z\|_2^2$$

→Reconstructed image $z \approx$ Input image x

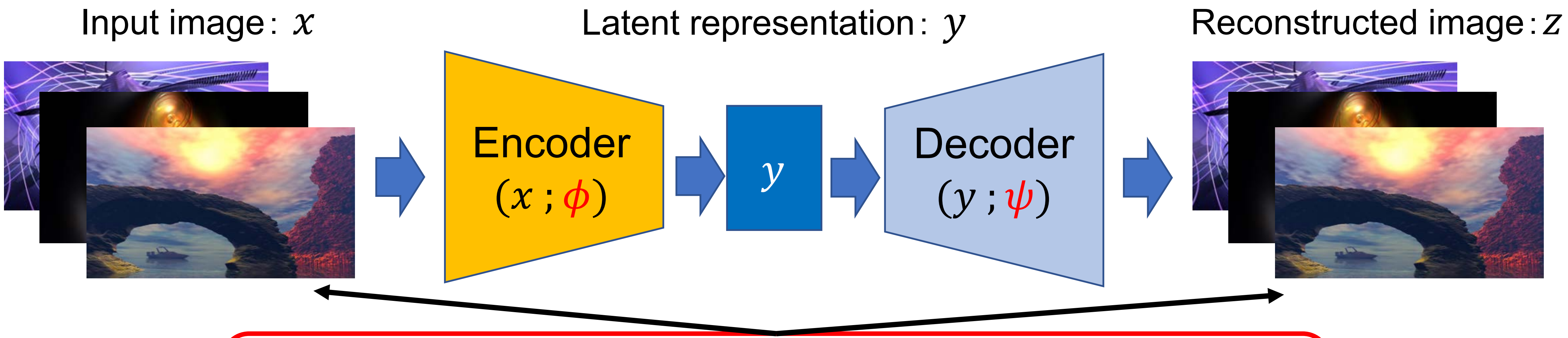
[3] F. Mentzer et.al., "Conditional probability models for deep image compression," *CVPR*, (2018)
[4] J. Balle et.al., "End-to-end optimized image compression", ICLR, (2017)

The parameters of the network are updated so that the error between the input image x and the reconstructed image z is reduced.

Autoencoder-based method [3,4]

Fujii Laboratory

Training of autoencoder



$$\min ||x - z||_2^2$$

→Reconstructed image $z \approx$ Input image x

[3] F. Mentzer et.al., “Conditional probability models for deep image compression,” *CVPR*, (2018)
[4] J. Balle et.al., “End-to-end optimized image compression”, ICLR, (2017)

By doing so, the output z of the network will be similar to the input image x.

Autoencoder-based method [3,4]

Testing of trained autoencoder

Input image: x

Encoder $(x\,;\phi)$ → y → Decoder $(y\,;\psi)$

Latent representation: y

Reconstructed image: z

Optimized for common implicit features in training images
→Applicable to unseen images

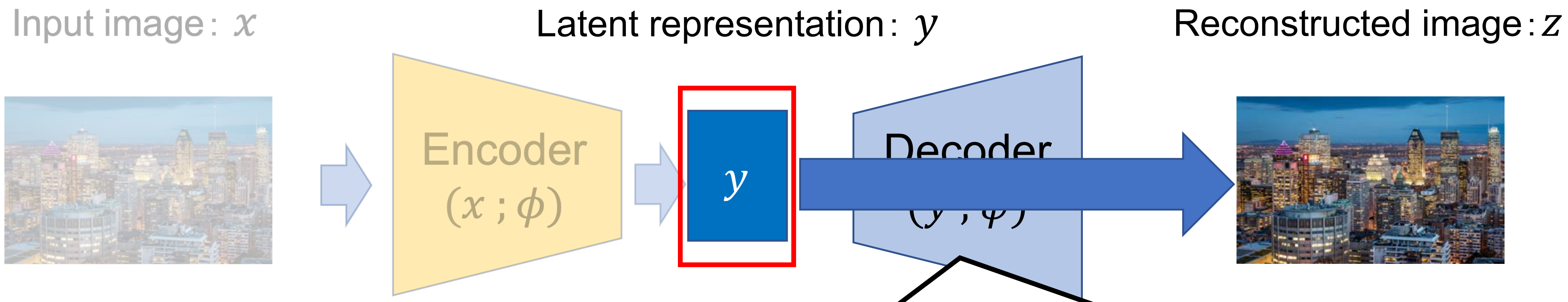
[3] F. Mentzer et.al., "Conditional probability models for deep image compression," *CVPR*, (2018)
[4] J. Balle et.al., "End-to-end optimized image compression", ICLR, (2017)

A network trained on a large number of images can represent general image features. Thus, the network can be applied to test images that were not used for training.

Autoencoder-based method [3,4]

Fujii Laboratory

Testing of trained autoencoder



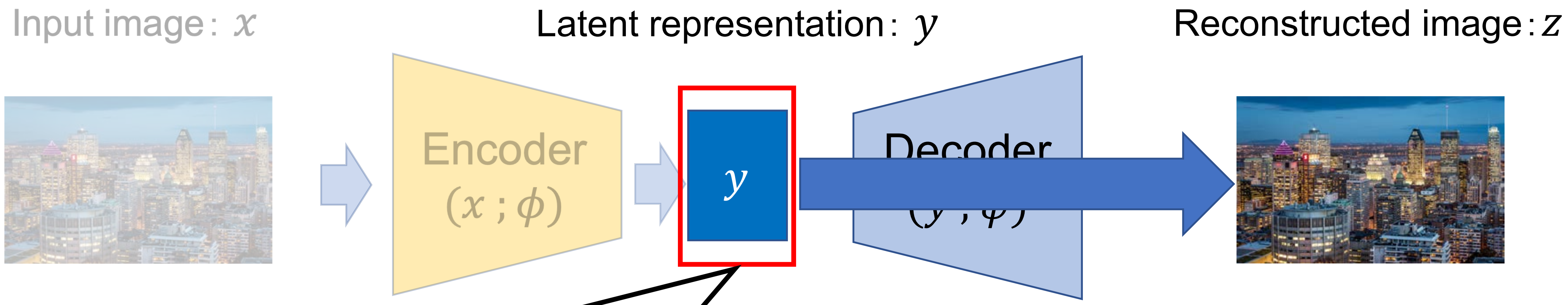
Latent representation (compressed representation) → Target image

[3] F. Mentzer et.al., "Conditional probability models for deep image compression," *CVPR*, (2018)
[4] J. Balle et.al., "End-to-end optimized image compression", ICLR, (2017)

When we attempt to transmit an image efficiently, we transmit the latent representation, and the receiver can reproduce a target image from the latent representation through the decoder.

Autoencoder-based method [3,4]

Testing of trained autoencoder



Transmitted information: Latent representation only

[3] F. Mentzer et.al., “Conditional probability models for deep image compression,” *CVPR*, (2018)
[4] J. Balle et.al., “End-to-end optimized image compression”, ICLR, (2017)

Therefore, the transmitted information is only a latent representation.
This is the outline of the autoencoder-based image compression method.

Autoencoder-based method

Fujii Laboratory

- Large number of images for training
- Large computational cost for training
- Large network architectures unsuitable for limited computational resources (e.g. memory)

Here we'd like to indicate three drawbacks of the autoencoder-based method.

Autoencoder-based method

Fujii Laboratory

- Large number of images for training
- Large computational cost for training
- Large network architectures unsuitable
  for limited computational resources (e.g. memory)

First, as discussed thus far, conventional methods using the autoencoder require a large number of images for training, whose collection is a very laborsome task.

Autoencoder-based method

Fujii Laboratory

- Large number of images for training
- Large computational cost for training
- Large network architectures unsuitable for limited computational resources (e.g. memory)

Second, when a network size is large, the training process becomes complex.

Autoencoder-based method

Fujii Laboratory

- Large number of images for training
- Large computational cost for training
- Large network architectures unsuitable for limited computational resources (e.g. memory)

Third, when we use digital devices with poor computational resources such as smartphones, a model with a large network size is not suitable.

Autoencoder-based method

Fujii Laboratory

- Large number of images for training
- Large computational cost for training
- Large network architectures unsuitable for limited computational resources (e.g. memory)

Smaller network size

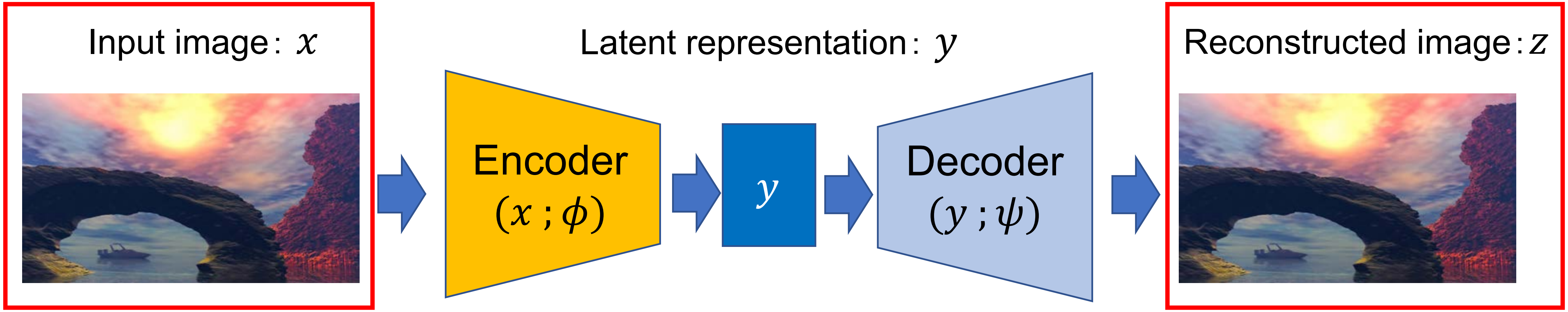
i.e. reduction of numbers of layers and channels while keeping compression performance

Therefore, we attempt to build a model with a smaller network size.
Specifically, we reduce the numbers of layers and channels.

Proposed Method

In this section, we explain the proposed method that can overcome abovementioned drawbacks.

Proposed method



~~Large number of images (Data sets : χ)~~ → Single target image

First, the proposed method does not use a large number of images as in the conventional methods. It uses only a single image to be compressed as the training data.

Proposed method

Fujii Laboratory

Input image : x

Latent representation : y

Encoder $(x\,;\phi)$ → y → Decoder $(y\,;\psi)$

Reconstructed image : z

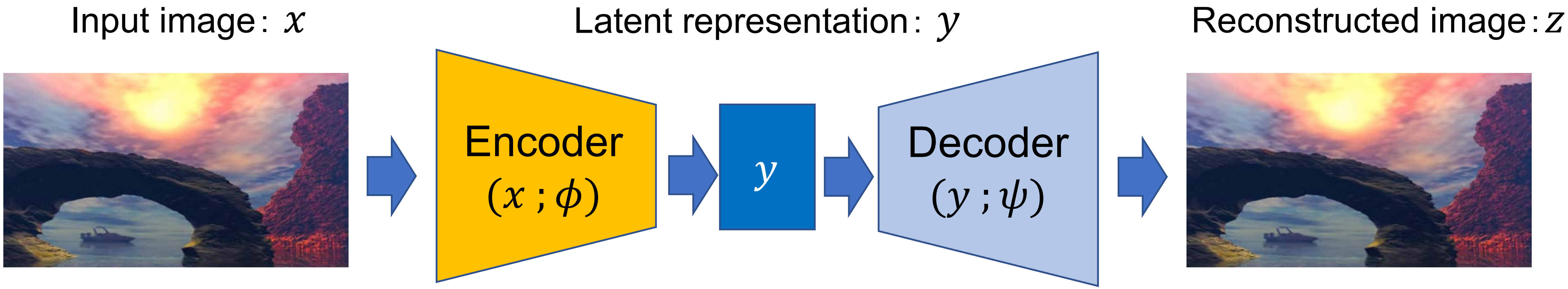
Overfitted

High performance

(smaller reconstruction error, higher compression performance)

Because it is trained on a single target image, this model is overfitted.

Proposed method



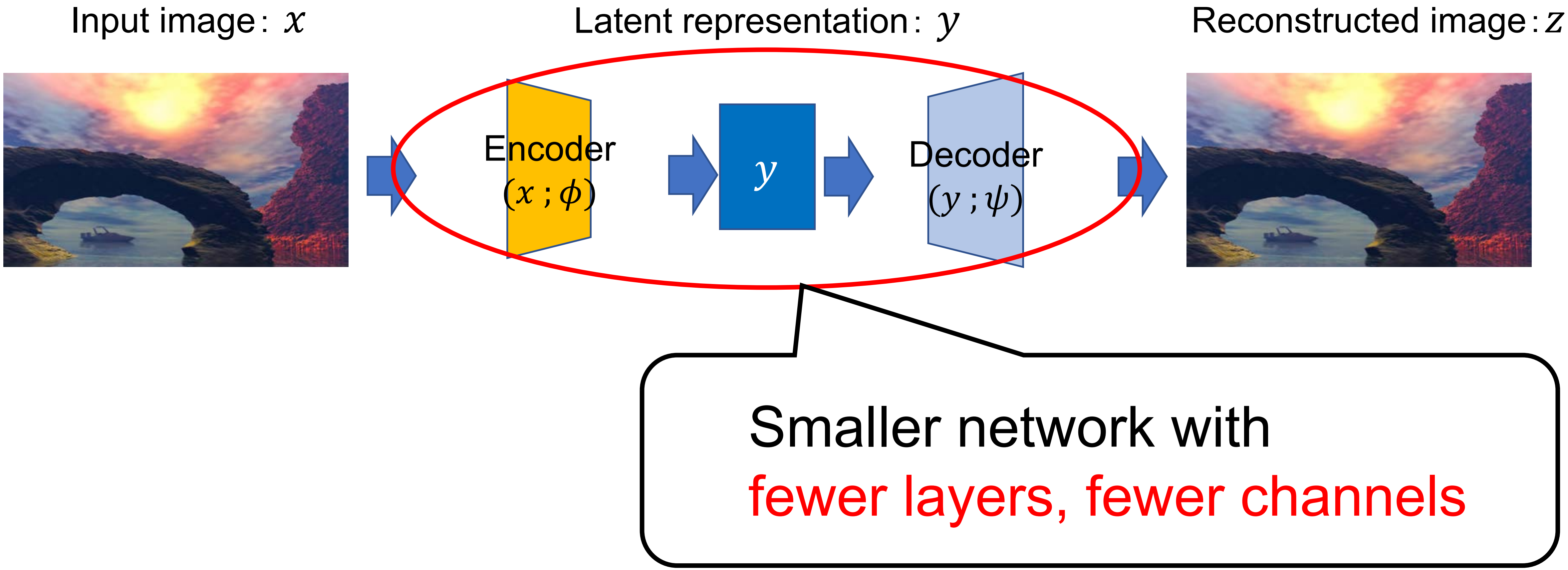
Overfitted

High performance

(smaller reconstruction error, higher compression performance)

Therefore, the network is optimized only for the image, which produces small reconstruction error and improves the compression performance.

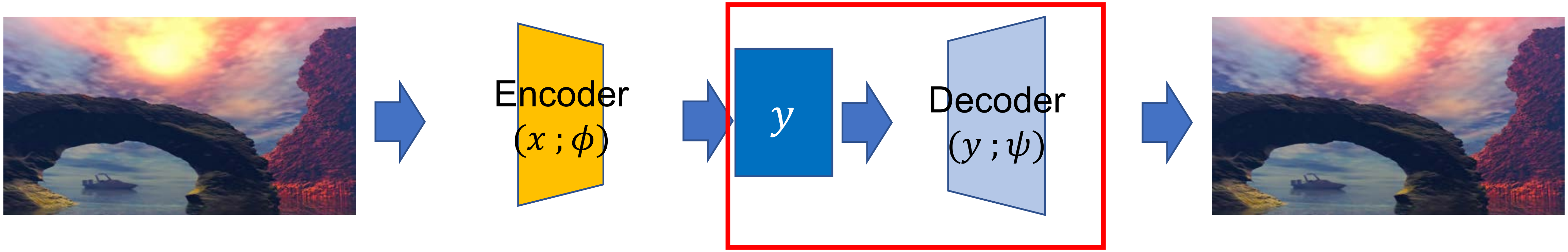
Proposed method



Thanks to the overfitting, we can reduce the number of layers and channels while keeping the moderate performance.

Proposed method

Input image: x Latent representation: y Reconstructed image: z



Transmitted information:
latent representation + decoder weights

In addition to the latent representation, the decoder weights have to be transmitted in the proposed method. An actual encoding method will be elaborated later on.

Loss function

Fujii Laboratory

Conventional Method [4]

$$\mathrm{L}(\phi,\psi) = \sum_{x\in\chi}\{D(x,z) + \lambda_{y'}\underbrace{\mathrm{R}(y')}_{\text{Latent Loss}}\} \quad (1)$$

χ	: Training dataset
D()	: Distortion function
R()	: Rate function
$y' = Q(y)$	: Quantized y
$\lambda_{y'}$	: Lagrangian multiplier

Proposed method

$$\mathrm{L}(\phi,\psi) = D(x,z) + \lambda_{y'}R(y') + \lambda_{\psi'}\underbrace{\mathrm{R}(\psi')}_{\text{Weight Loss}} \quad (2)$$

$\psi' = Q(\psi)$	: Quantized ψ
$\lambda_{\psi'}$	: Lagrangian multiplier

Next, we introduce the loss function in the proposed method.

Loss function

Fujii Laboratory

Conventional Method [4]

$$L(\phi,\psi) = \sum_{x\in\chi}\{D(x,z) + \lambda_{y'}\underbrace{R(y')}_{\text{Latent Loss}}\} \quad (1)$$

χ : Training dataset
D() : Distortion function
R() : Rate function
$y' = Q(y)$: Quantized y
$\lambda_{y'}$: Lagrangian multiplier

Proposed method

$$L(\phi,\psi) = D(x,z) + \lambda_{y'}R(y') + \lambda_{\psi'}\underbrace{R(\psi')}_{\text{Weight Loss}} \quad (2)$$

$\psi' = Q(\psi)$: Quantized ψ
$\lambda_{\psi'}$: Lagrangian multiplier

We extend the loss function in [4], where D represents a distortion function w.r.t. original and reconstructed images, and R is a rate function w.r.t. a latent representation.

Loss function

Fujii Laboratory

Conventional Method [4]

$$\mathrm{L}(\phi,\psi) = \sum_{x\in\chi}\{D(x,z) + \lambda_{y'}\underbrace{\mathrm{R}(y')}_{\text{Latent Loss}}\} \quad (1)$$

χ	: Training dataset
$\mathrm{D}()$	: Distortion function
$\mathrm{R}()$	: Rate function
$y' = Q(y)$	: Quantized y
$\lambda_{y'}$	: Lagrangian multiplier

Proposed method

$$\mathrm{L}(\phi,\psi) = D(x,z) + \lambda_{y'}R(y') + \lambda_{\psi'}\underbrace{\mathrm{R}(\psi')}_{\text{Weight Loss}} \quad (2)$$

$\psi' = Q(\psi)$	: Quantized ψ
$\lambda_{\psi'}$	: Lagrangian multiplier

y' represents a quantized version of a latent representation y, and $\lambda_{y'}$ is a Lagrangian multiplier that controls the balance between the rate and distortion.

Loss function

Fujii Laboratory

Conventional Method [4]

$$\mathrm{L}(\phi,\psi) = \sum_{x\in\chi}\{D(x,z) + \lambda_{y'}\underbrace{\mathrm{R}(y')}_{\text{Latent Loss}}\} \quad (1)$$

- χ : Training dataset
- $\mathrm{D}()$: Distortion function
- $\mathrm{R}()$: Rate function
- $y' = Q(y)$: Quantized y
- $\lambda_{y'}$: Lagrangian multiplier

Proposed method

$$\mathrm{L}(\phi,\psi) = D(x,z) + \lambda_{y'}R(y') \boxed{+ \lambda_{\psi'}\underbrace{\mathrm{R}(\psi')}_{\text{Weight Loss}}} \quad (2)$$

- $\psi' = Q(\psi)$: Quantized ψ
- $\lambda_{\psi'}$: Lagrangian multiplier

In the proposed method, because we have to consider rate of decoder weights, we append a new rate term w.r.t. the weights to Eq. (1).

Loss function

Fujii Laboratory

Conventional Method [4]

$$\mathrm{L}(\phi,\psi) = \sum_{x\in\chi}\{D(x,z) + \lambda_{y'}\underbrace{\mathrm{R}(y')}_{\text{Latent Loss}}\} \qquad (1)$$

χ	: Training dataset
$\mathrm{D}()$	: Distortion function
$\mathrm{R}()$	: Rate function
$y' = Q(y)$	: Quantized y
$\lambda_{y'}$	: Lagrangian multiplier

Proposed method

$$\mathrm{L}(\phi,\psi) = D(x,z) + \lambda_{y'}R(y') \boxed{+ \lambda_{\psi'}\underbrace{\mathrm{R}(\psi')}_{\text{Weight Loss}}} \qquad (2)$$

$\psi' = Q(\psi)$	: Quantized ψ
$\lambda_{\psi'}$	: Lagrangian multiplier

Here, ψ' indicates a quantized version of original weights ψ. Eq. (2) is the proposed loss function.

Convolution process

Fujii Laboratory

Convolution for a spatial position (u, v)

$$I'(u, v, ch') = \sum_{ch} \sum_{i} \sum_{j} I(u + i, v + j, ch) \cdot k(i, j, ch, ch') \tag{3}$$

ch/ch' : input/output channel of the layer
k : convolutional filter

Because decoder weights are represented by real numbers, we have to quantize them to reduce their data amount. To this end, we here give a brief overview of the convolution in the autoencoder.

Convolution process

Fujii Laboratory

Convolution for a spatial position (u, v)

$$I'(u, v, ch') = \sum_{ch} \sum_{i} \sum_{j} I(u + i, v + j, ch) \cdot k(i, j, ch, ch') \quad (3)$$

ch/ch' : input/output channel of the layer
k : convolutional filter

If we consider a two-dimensional convolution, the operation is generally formulated by Eq. (3).

Convolution process

Fujii Laboratory

Convolution for a spatial position (u, v)

$$I'(u, v, ch') = \sum_{ch} \sum_{i} \sum_{j} I(u + i, v + j, ch) \cdot k(i, j, ch, ch') \qquad (3)$$

ch/ch' : input/output channel of the layer
k : convolutional filter

Convolution: computed for each output channel ch'

Decoder weights: quantized for each output channel ch'

Because the convolution is computed for each output channel, we quantize weights for each output channel as well.

Convolution process

Fujii Laboratory

Convolution for a spatial position (u, v)

$$I'(u, v, ch') = \sum_{ch} \sum_{i} \sum_{j} I(u + i, v + j, ch) \cdot k(i, j, ch, ch') \qquad (3)$$

ch/ch' : input/output channel of the layer
k : convolutional filter

Convolution: computed for each output channel ch'

Decoder weights: quantized for each output channel ch'

In other words, the filter is adaptively quantized for each output channel.

Quantization Process

Fujii Laboratory

Quantization Process : $W' = \mathrm{Q}(W)$

$$W' = \mathrm{round}\left(\frac{W}{q}\right)q \qquad (4)$$

$$q = 2^N \qquad (5)$$

$$N = \mathrm{round}(\log_2(C \cdot \overline{w})) \qquad (6)$$

W / W' : Weight before (/after) applying quantization
q : Quantization step
$\overline{w}$: Absolute average of weights
C : Control parameter (const)

For notational convenience, decoder weights and biases are commonly represented by W, but the following quantization is applied to weights and biases individually.

Quantization Process

Fujii Laboratory

Quantization Process : $W' = \mathrm{Q}(W)$

$$W' = \mathrm{round}\left(\frac{W}{q}\right) q \qquad (4)$$

$$q = 2^N \qquad (5)$$

$$N = \mathrm{round}(\log_2(C \cdot \overline{w})) \qquad (6)$$

W / W'	: Weight before (/after) applying quantization
q	: Quantization step
$\overline{w}$	: Absolute average of weights
C	: Control parameter (const)

Our quantization is a simple scalar quantization as defined in Eq. (4), where q is the quantization step size.

Quantization Process

Fujii Laboratory

Quantization Process : $W' = \mathrm{Q}(W)$

$$W' = \mathrm{round}\left(\frac{W}{q}\right)q \tag{4}$$

$$q = 2^N \tag{5}$$

$$N = \mathrm{round}(\log_2(C \cdot \overline{w})) \tag{6}$$

W / W'	: Weight before (/after) applying quantization
q	: Quantization step
$\overline{w}$	: Absolute average of weights
C	: Control parameter (const)

The key point of our quantization process is that it is adaptively determined from weights for each output channel.

Quantization Process

Quantization Process : $W' = \mathrm{Q}(W)$

$$W' = \mathrm{round}\left(\frac{W}{q}\right) q \quad (4)$$

$$q = 2^N$$

$$N = \mathrm{round}(\log_2(C \cdot \bar{w}))$$

Controlling the ratio of '0' and '1' in binary representation

W / W'	: Weight before (/after) applying quantization
q	: Quantization step
$\bar{w}$	: Absolute average of weights
C	: Control parameter (const)

The quantization step q is parameterized by N, where C is constant and $\bar{w}$ is the means of absolute values of weights.

Quantization Process

Quantization Process : $W' = \mathrm{Q}(W)$

$$W' = \mathrm{round}\left(\frac{W}{q}\right)q \qquad (4)$$

$$q = 2^N$$

$$N = \mathrm{round}(\,\log_2(C \cdot \bar{w})\,)$$

Controlling the ratio of '0' and '1' in binary representation

W / W'	: Weight before (/after) applying quantization
q	: Quantization step
$\bar{w}$	: Absolute average of weights
C	: Control parameter (const)

By using our quantization, the probability of ‘0’ and ‘1’ will be biased for each channel, so that they can be efficiently encoded by a suitable entropy coding technique.

Model Summary

Fujii Laboratory

	Conventional Method [4]	Proposed Method
Embed image information into…	Latent representations	Latent representations and network parameters
Training data	Large number of images	A single target image
Trained model has…	Generalization performance	Target image specific
Information for decoding	Latent representations	Latent representations and decoder weights
Loss function $\mathrm{L}(\phi,\psi)$	$\sum_{f\in\chi}\{D(x,z)+\lambda_{y'}\mathrm{R}(y')\}$	$D(x,z)+\lambda_{y'}R(y')+\lambda_{\psi'}\mathrm{R}(\psi')$

We present a summary of the conventional and proposed methods.

Model Summary

Fujii Laboratory

	Conventional Method [4]	Proposed Method
Embed image information into…	Latent representations	Latent representations and network parameters
Training data	Large number of images	A single target image
Trained model has…	Generalization performance	Target image specific
Information for decoding	Latent representations	Latent representations and decoder weights
Loss function $\mathrm{L}(\phi, \psi)$	$\sum_{f \in \chi} \{ D(x, z) + \lambda_{y'} \mathrm{R}(y') \}$	$D(x, z) + \lambda_{y'} R(y') + \lambda_{\psi'} \mathrm{R}(\psi')$

The proposed method is trained on a single target image for which the network is optimized.
This results in target image specific weights.

Model Summary

Fujii Laboratory

	Conventional Method [4]	Proposed Method
Embed image information into…	Latent representations	Latent representations and network parameters
Training data	Large number of images	A single target image
Trained model has…	Generalization performance	Target image specific
Information for decoding	Latent representations	Latent representations and decoder weights
Loss function $\mathrm{L}(\phi, \psi)$	$\sum_{f \in \chi} \{ D(x, z) + \lambda_{y'} \mathrm{R}(y') \}$	$D(x, z) + \lambda_{y'} R(y') + \lambda_{\psi'} \mathrm{R}(\psi')$

Because we have to transmit decoder weights to reconstruct an image, we appended a rate term w.r.t. weights to the loss function in [4].

Experiment

We evaluated our method through experiments.

Quantitative / Qualitative evaluation

Fujii Laboratory

- Image compression through autoencoder
  - Test Data : 8K images (a) and (b)
  - Metric (Distortion) : Peak signal-to-noise ratio (PSNR)
  - Metric (Rate) : Bits per pixel (bpp)
  - Baseline : Ref. [4]
- Comparison with
  - Ref. [4]
  - JPEG

(a)

(b)

We conducted an experiment to verify the effectiveness of our method.

Quantitative / Qualitative evaluation

Fujii Laboratory

- Image compression through autoencoder
  - Test Data : 8K images (a) and (b)
  - Metric (Distortion) : Peak signal-to-noise ratio (PSNR)
  - Metric (Rate) : Bits per pixel (bpp)
  - Baseline : Ref. [4]

- Comparison with
  - Ref. [4]
  - JPEG

(a)

(b)

We used two 8K images (a) and (b) as test data. We employed PSNR and bpp as distortion and rate metrics, respectively.

Quantitative / Qualitative evaluation

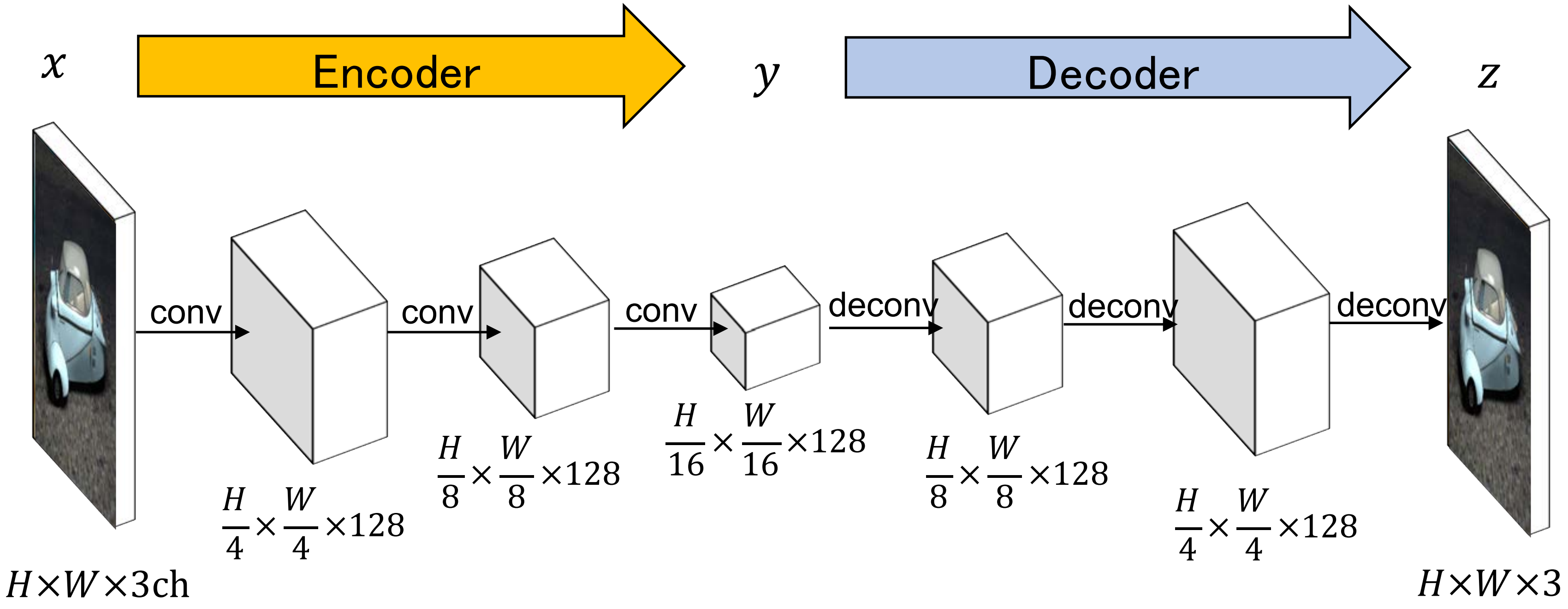
- Image compression through autoencoder
  - Test Data : 8K images (a) and (b)
  - Metric (Distortion) : Peak signal-to-noise ratio (PSNR)
  - Metric (Rate) : Bits per pixel (bpp)
  - Baseline : Ref. [4]
- Comparison with
  - Ref. [4]
  - JPEG

(a)

(b)

We implemented our method on an autoencoder-based method [4]. To verify the effectiveness of our method, it was compared with JPEG and the original technique [4].

Network Architecture (Ref. [4])



- Encoder : 3 convolutional layers
- Decoder : 3 convolutional layers
- Latent representation : 128 channels

In Ref. [4], both the encoder and decoder include three convolutional layers, and each tensor has 128 channels.

Network Architecture（Ref. [4]）

Fujii Laboratory

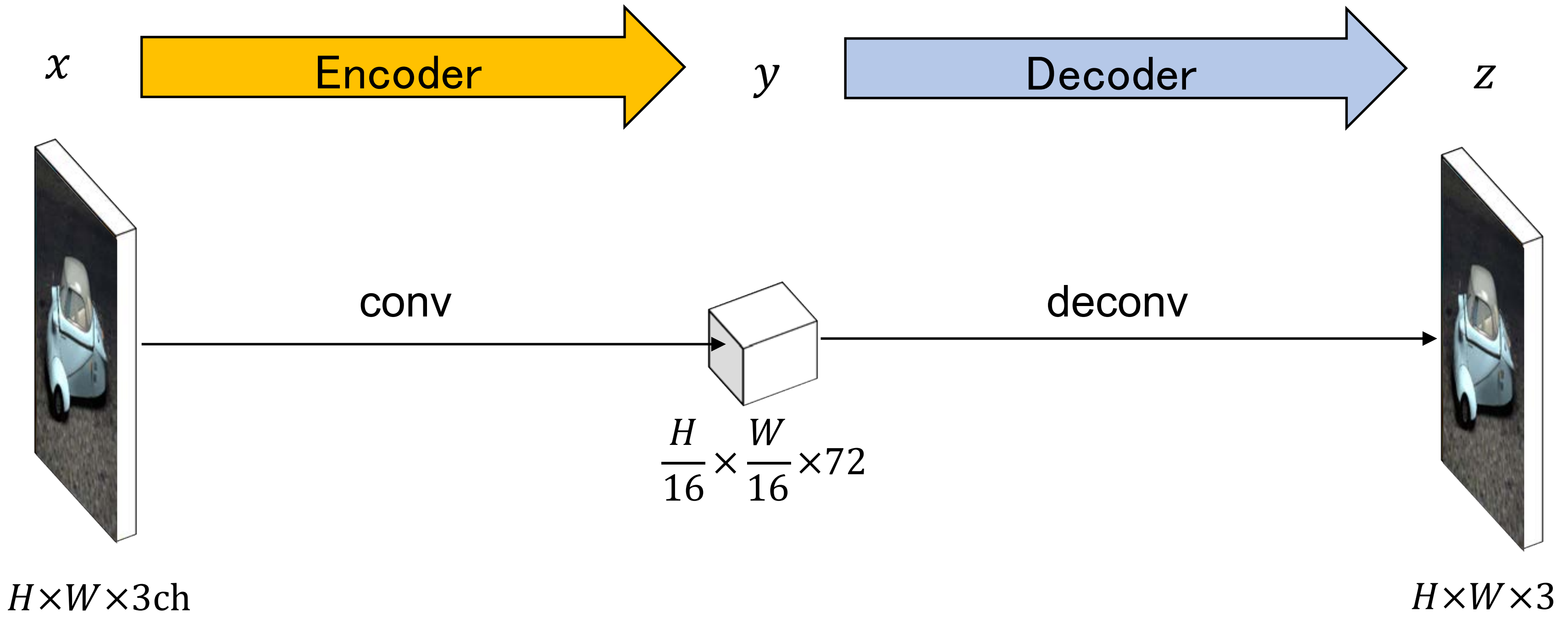
- Network architecture in Ref. [4]

Layer	Kernel size	Stride size	Channel (in/out)	Accumulated Strides (in/out)	Input
Conv-1+GDN-1	9	4	3/128	1/4	Input Image
Conv-2+GDN-2	5	2	128/128	4/8	Conv-1+GDN-1
Conv-3	5	2	128/128	8/16	Conv-2+GDN-2
Deconv-1+IGDN-1	5	2	128/128	16/8	Conv-3
Deconv-2+IGDN-2	5	2	128/128	8/4	Deconv-1+IGDN-1
Deconv-3	9	4	128/3	4/1	Deconv-2+IGDN-2

This table summarizes the network architecture of Ref. [4].

Network Architecture (Ours)

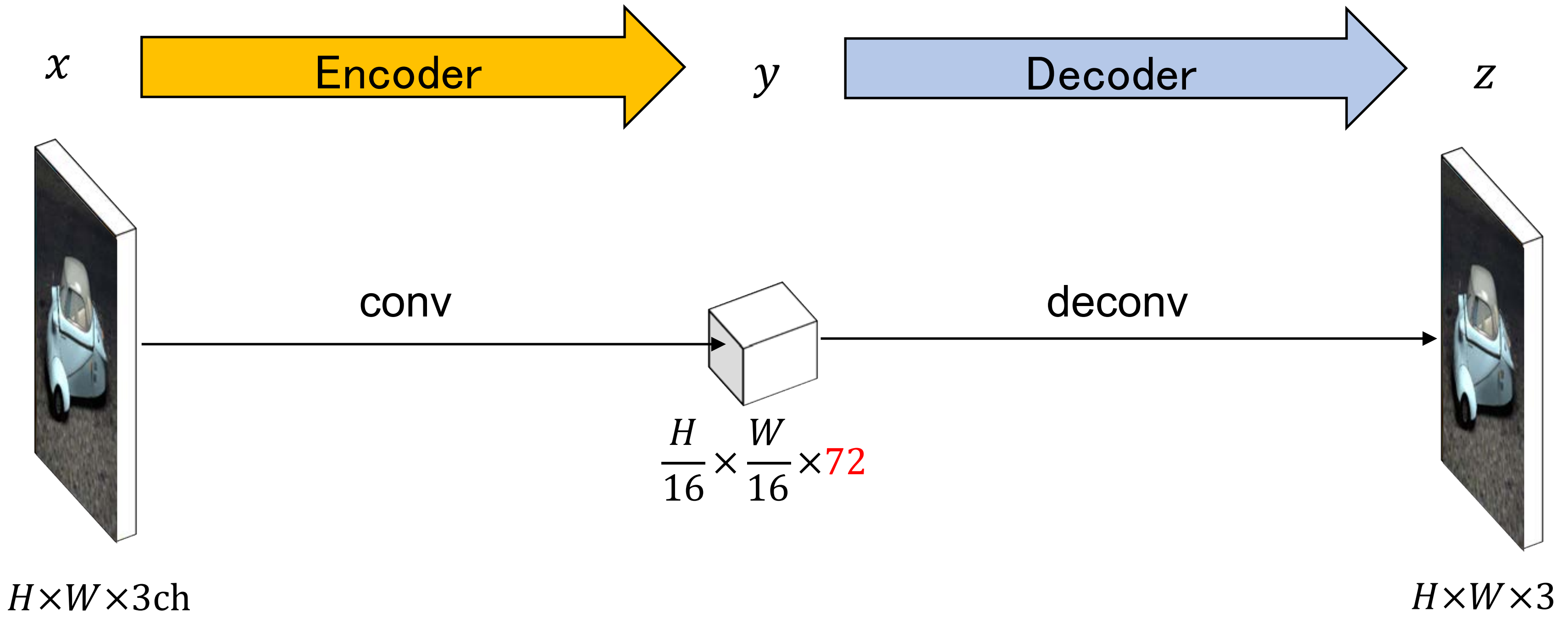
Fujii Laboratory



Layer	Kernel size	Stride size	Channel (in/out)	Accumulated Strides (in/out)	Input
Conv-1+GDN-1	33	16	3/72	1/16	Input Image
Deconv-1+IGDN-1	33	16	72/3	16/1	Conv-1+GDN-1

Meanwhile, our method was implemented by using a smaller network than Ref. [4].

Network Architecture (Ours)



Layer	Kernel size	Stride size	Channel (in/out)	Accumulated Strides (in/out)	Input
Conv-1+GDN-1	33	16	3/72	1/16	Input Image
Deconv-1+IGDN-1	33	16	72/3	16/1	Conv-1+GDN-1

Specifically, we used one convolutional layer and 72 channels for the latent representation.

Training Details (Ours)

Fujii Laboratory

- Our Training Details
  - Epoch : 130,000 (a), 70,000 (b)
  - Batch size : 1
  - Optimizer : Adam (learning rate=1e-4)
  - Quantization parameter C : 0.01

(a)

(b)

In the proposed method, the training epochs were set to 130,000 and 70,000 for (a) and (b), respectively. The quantization parameter C was set to 0.01.

Training Details (Ours)

Fujii Laboratory

- Different Lambda Parameters Sets $(\lambda_{y'}, \lambda_{\psi'})$

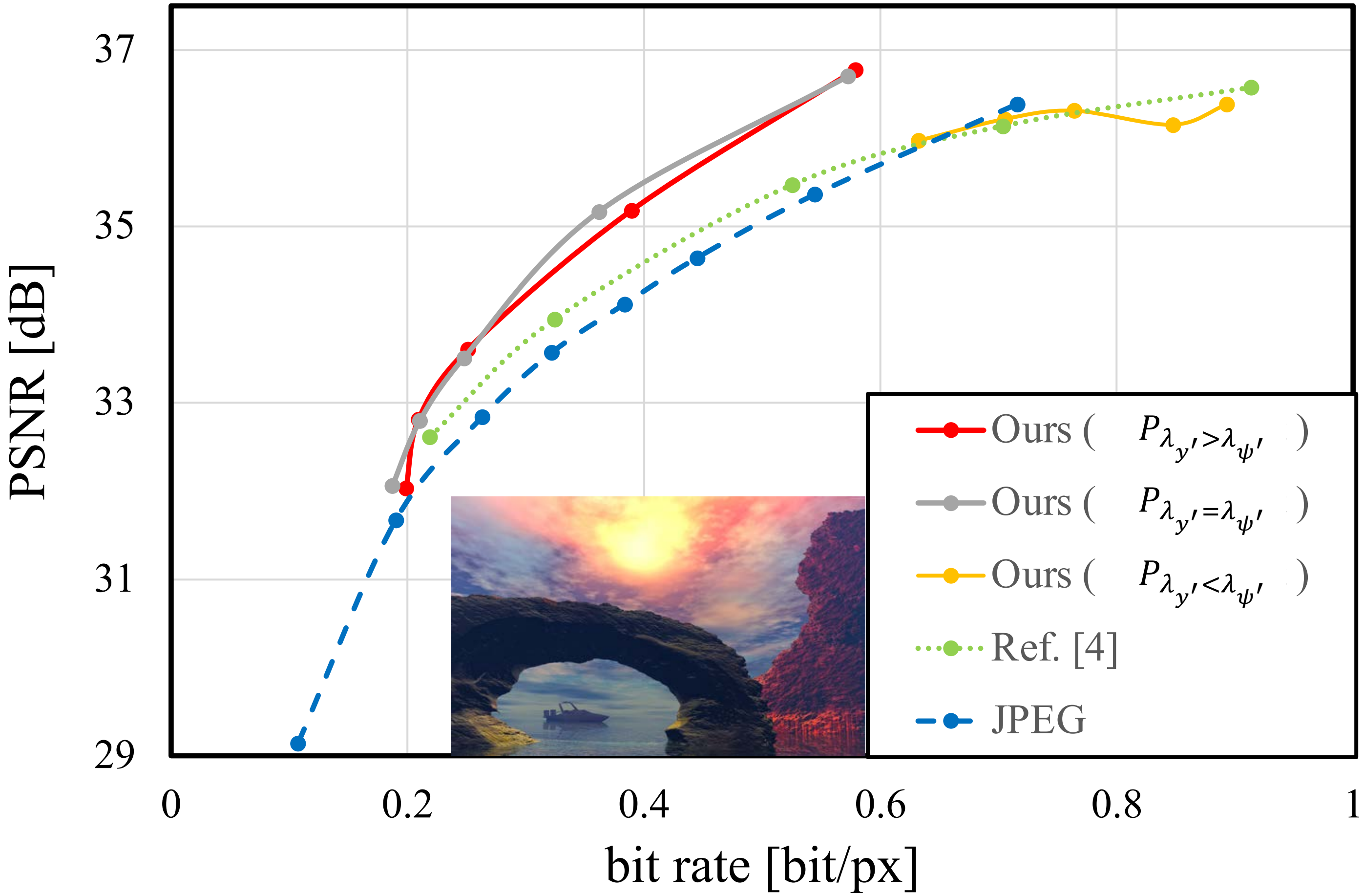
$$\mathrm{L}(\phi, \psi) = D(x, z) + \lambda_{y'} R(y') + \lambda_{\psi'} \mathrm{R}(\psi')$$

Parameter Sets	$(\lambda_{y'}, \lambda_{\psi'})$
$P_{\lambda_{y'} > \lambda_{\psi'}}$	(100, 10), (250, 25), (500, 50), (750, 75), (1000, 100)
$P_{\lambda_{y'} = \lambda_{\psi'}}$	(100, 100), (250, 250), (500, 500), (750, 750), (1000, 1000)
$P_{\lambda_{y'} < \lambda_{\psi'}}$	(10, 100), (25, 250), (50, 500), (75, 750), (100, 1000)

We used three different sets of parameters λ in the loss function, which can be categorized whether $\lambda_{y'}$ is larger than $\lambda_{\psi'}$.

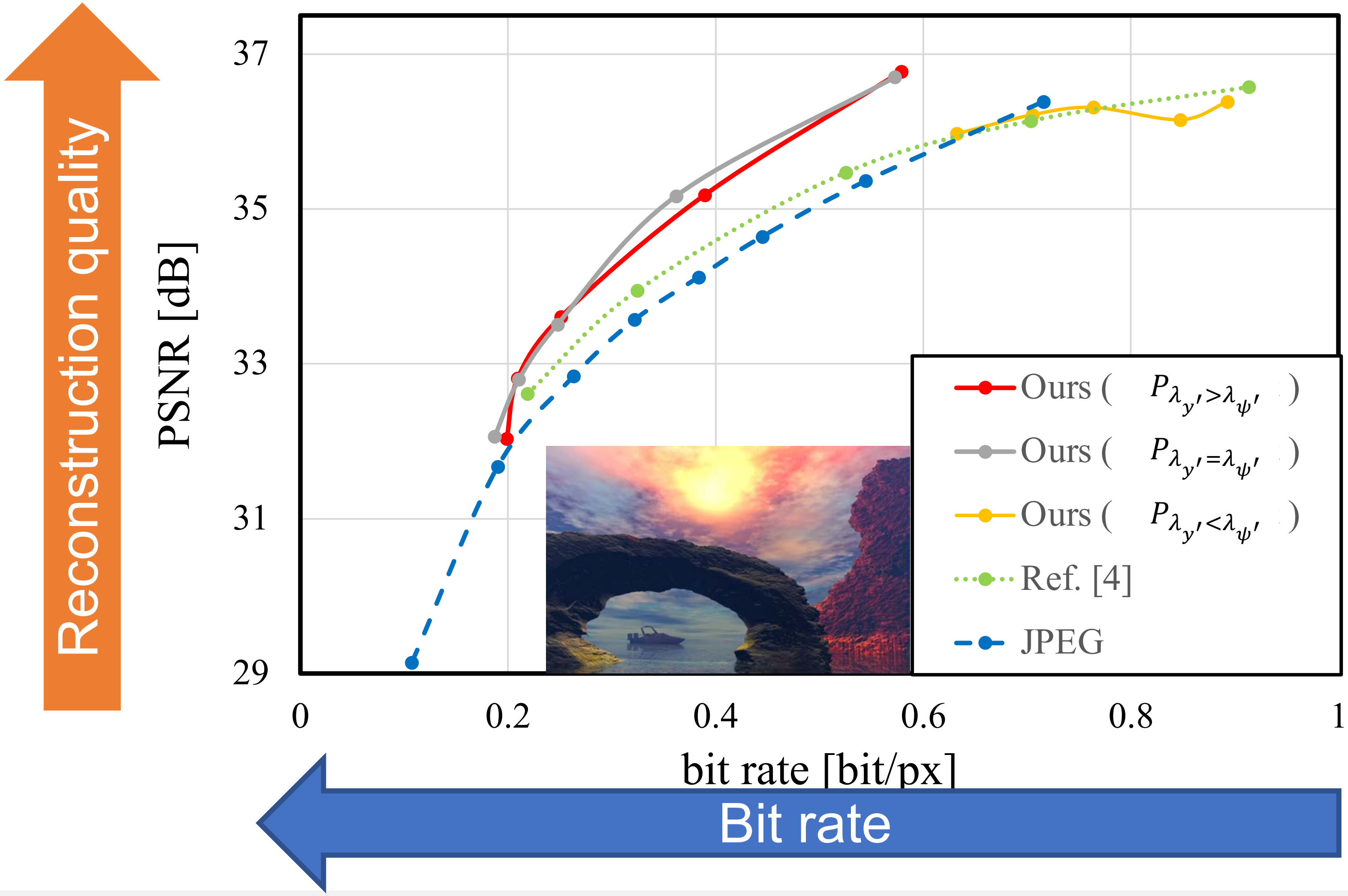
Rate-Distortion Curve

Fujii Laboratory



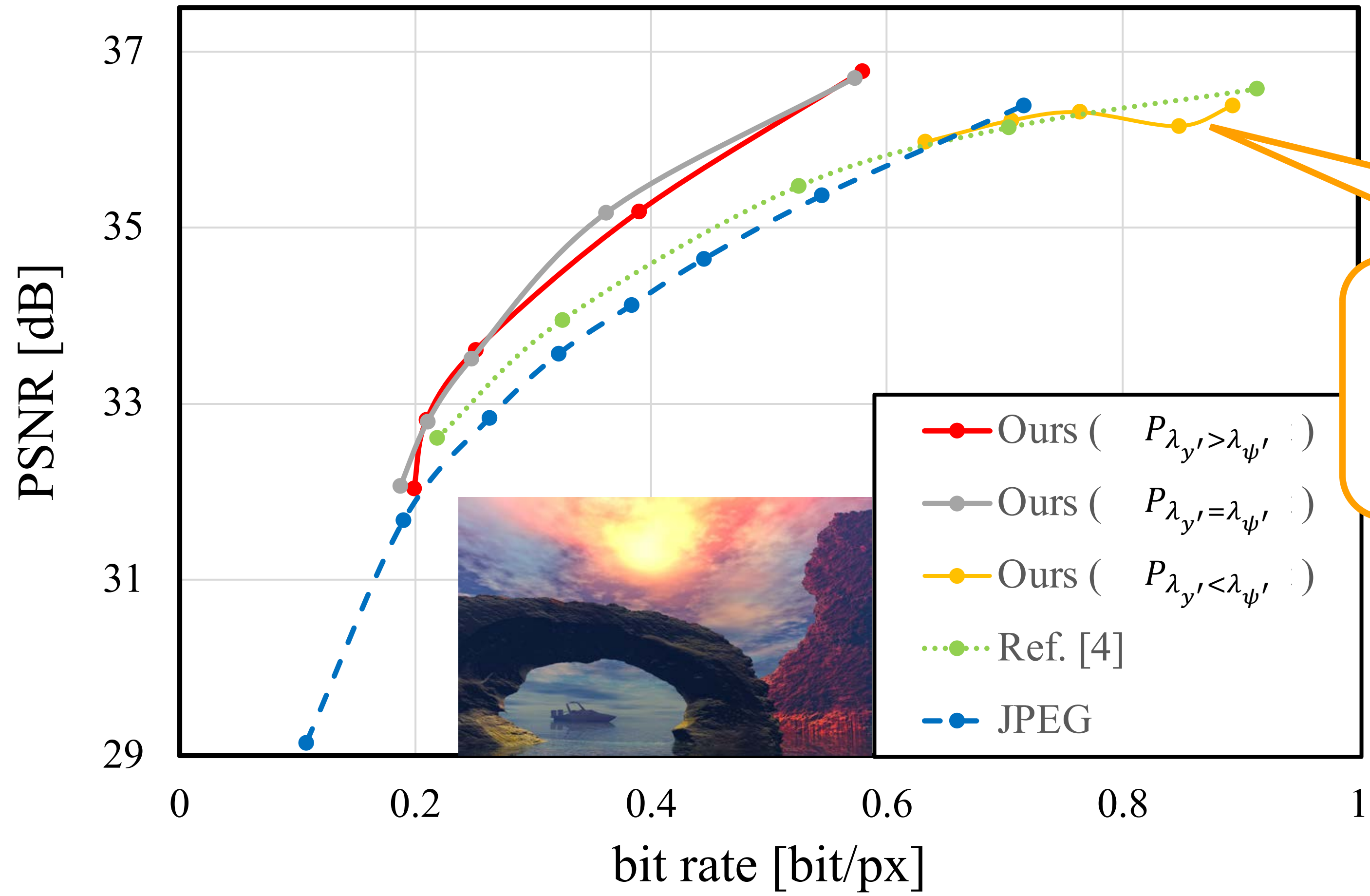
This graph shows rate-distortion curves of our method and two other methods.

Rate-Distortion Curve



The vertical axis represents the PSNR, where a higher value indicates better reconstruction quality.
The horizontal axis represents the bit rate, where a smaller value indicates better compression.

Rate-Distortion Curve

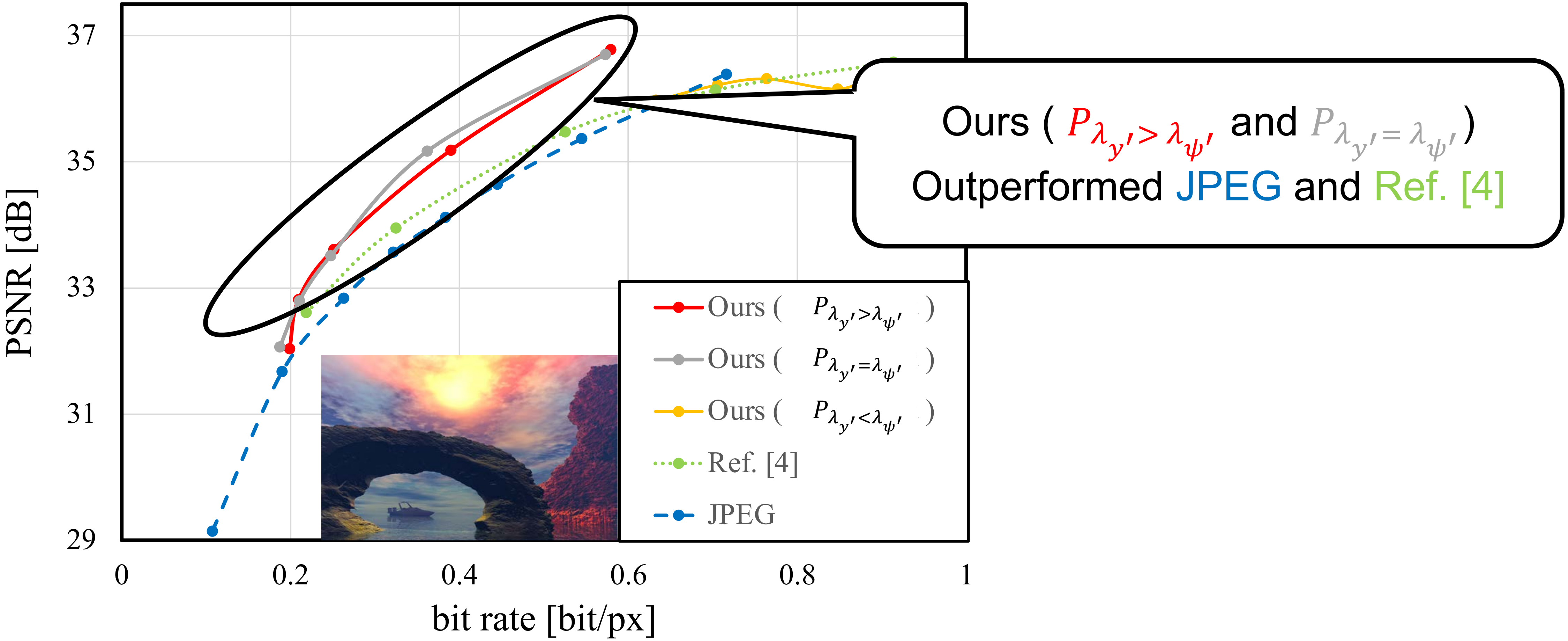


Ours ($P_{\lambda_{y'}<\lambda_{\psi'}}$)

Performance equivalent to JPEG and Ref. [4]

We can confirm that our method with $P_{\lambda_{y'}<\lambda_{\psi'}}$ achieved comparable performance to JPEG and Ref. [4].

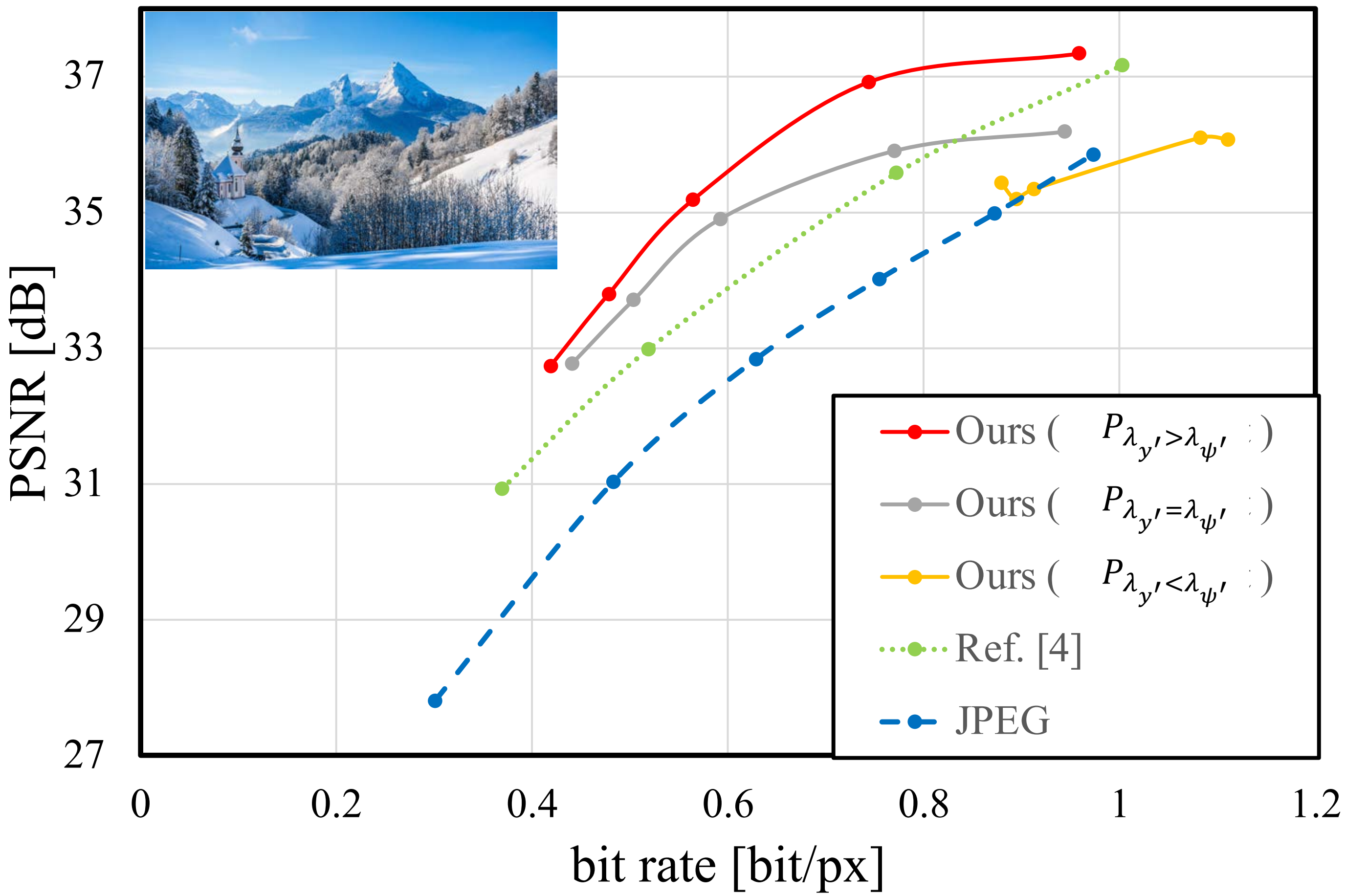
Rate-Distortion Curve



Our method with $P_{\lambda_{y'} \geq \lambda_{\psi'}}$ outperformed Ref. [4] and JPEG, even though our network was smaller than Ref. [4].

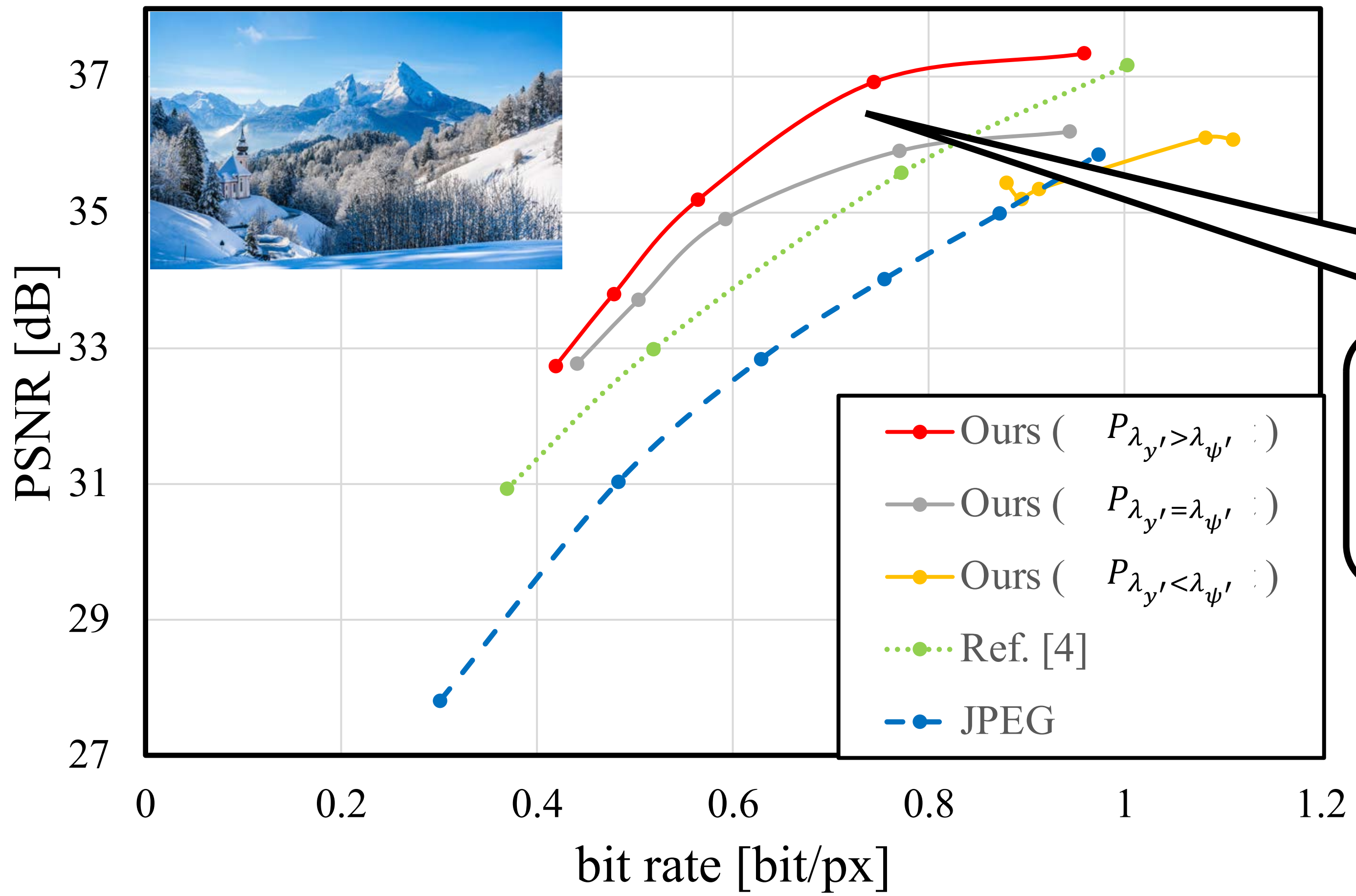
Rate-Distortion Curve

Fujii Laboratory



This is the result of the other test image.

Rate-Distortion Curve



Ours ($P_{\lambda_{y'}>\lambda_{\psi'}}$ and $P_{\lambda_{y'}=\lambda_{\psi'}}$)
Outperformed JPEG and
Equivalent to Ref. [4]

The proposed method achieved the better performance than JPEG and Ref. [4].

Reconstructed Images

Ground Truth
PSNR [dB] / bpp

JPEG
31.03 / 0.482

Balle et al. [4]
32.98 / 0.518

Ours
33.79 / 0.477

We also evaluate qualitative image quality. These are the images reconstructed from JPEG, Ref. [4], and our method.

Reconstructed Images

Ground Truth

PSNR [dB] / bpp

JPEG

31.03 / 0.482

Balle et al. [4]

32.98 / 0.518

Ours

33.79 / 0.477

These are enlarged views of the reconstructed images. Our method can preserve finer textures with a smaller bpp than the other methods.

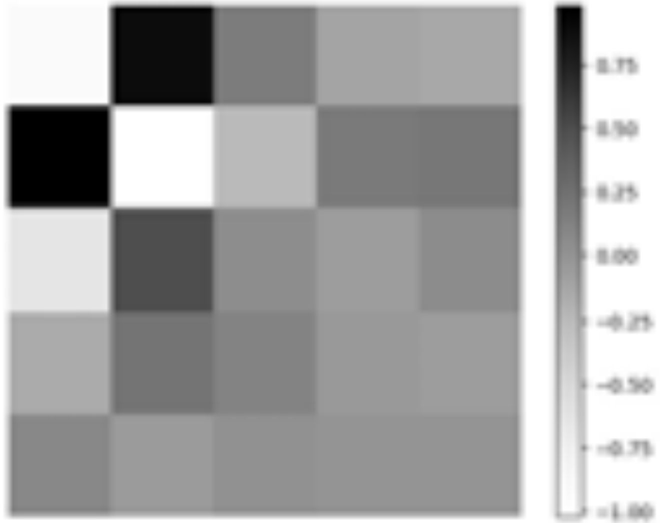
Visualization of Convolutional Filter

We then investigate why the fine image features could be represented efficiently in our method, by visualizing convolutional filters in our networks.

Visualization of Convolutional Filter

Fujii Laboratory

Ref. [4] (Generalization)



Filter size : 5x5

Ours (Overfitting)

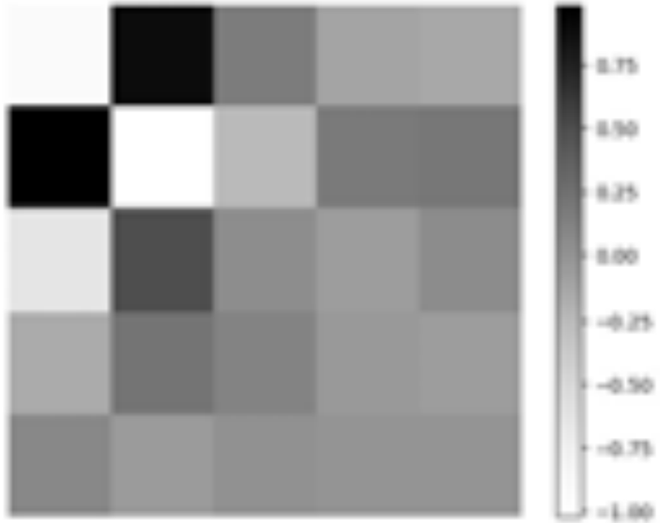
Filter size : 33x33

These images are the decoder's convolutional filters of Ref. [4] and our method.

Visualization of Convolutional Filter

Fujii Laboratory

Ref. [4] (Generalization)



Filter size : 5x5

Ours (Overfitting)

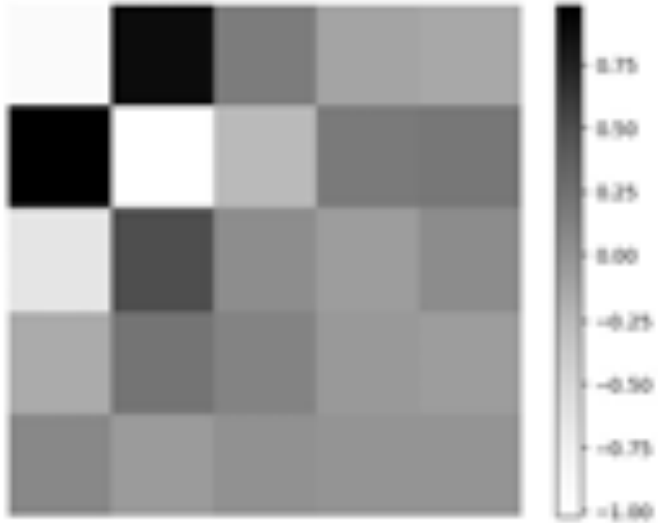
Filter size : 33x33

For Ref. [4], we visualize the filter deconvolving the latent representation with the largest variance over channels.

Visualization of Convolutional Filter

Fujii Laboratory

Ref. [4] (Generalization)



Filter size : 5x5

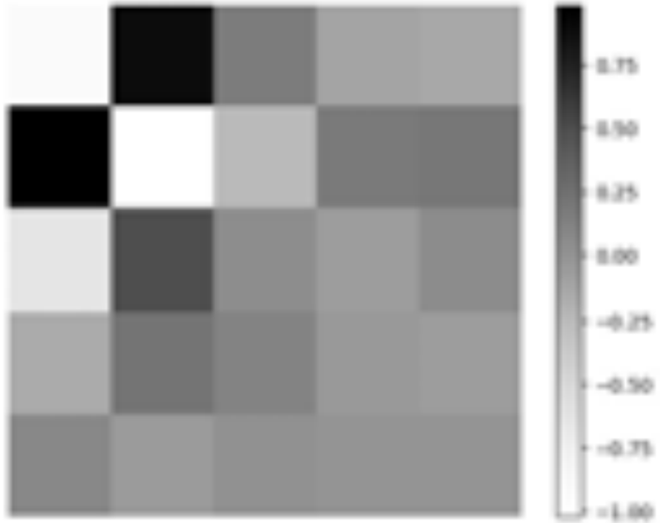
Ours (Overfitting)

Filter size : 33x33

For ours, we extract the filter with the largest variance over channels as well.

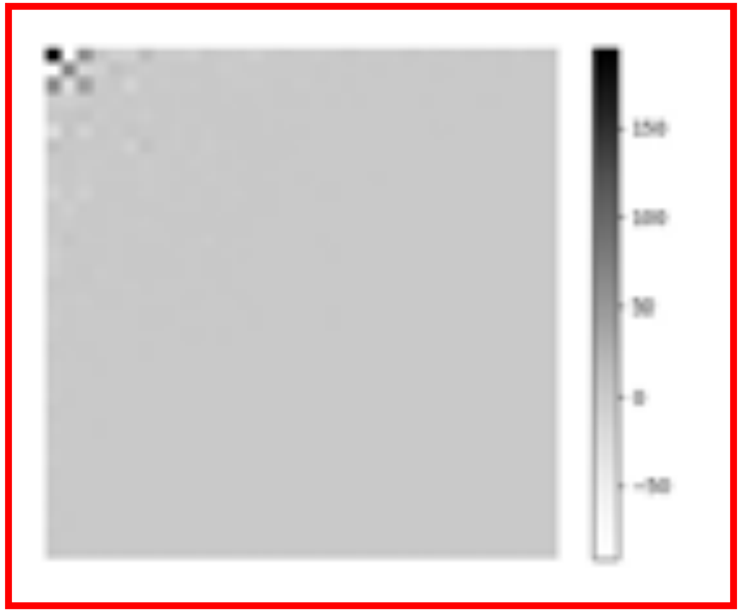
Visualization of Convolutional Filter

Ref. [4] (Generalization)



Filter size : 5x5

Ours (Overfitting)



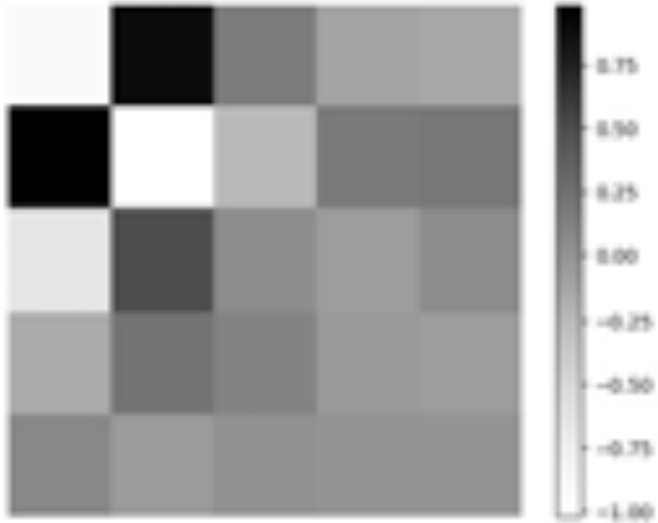
Filter size : 33x33

From the visualization results, it can be seen that the shape of our filter is narrower than that of Ref. [4].

Visualization of Convolutional Filter

Fujii Laboratory

Ref. [4] (Generalization)



Filter size : 5x5

Ours (Overfitting)

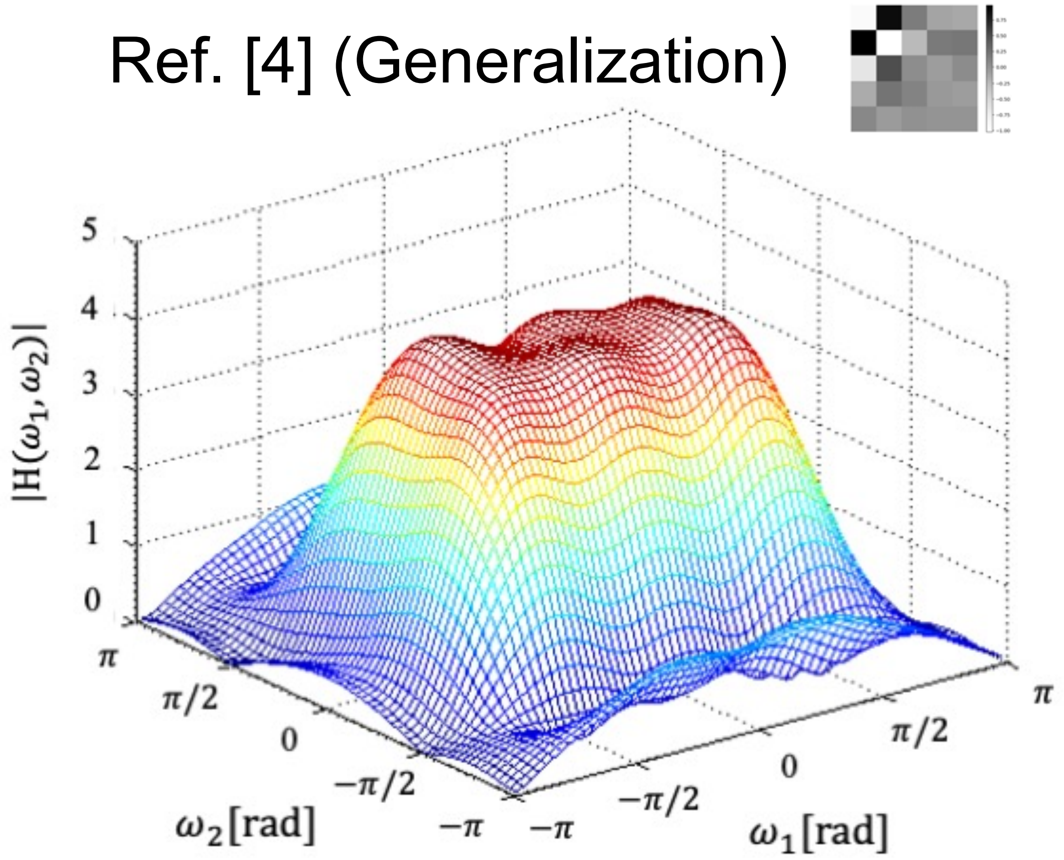
Filter size : 33x33

This indicates that our filter preserves the high-frequency components of the image, which correspond to textures.

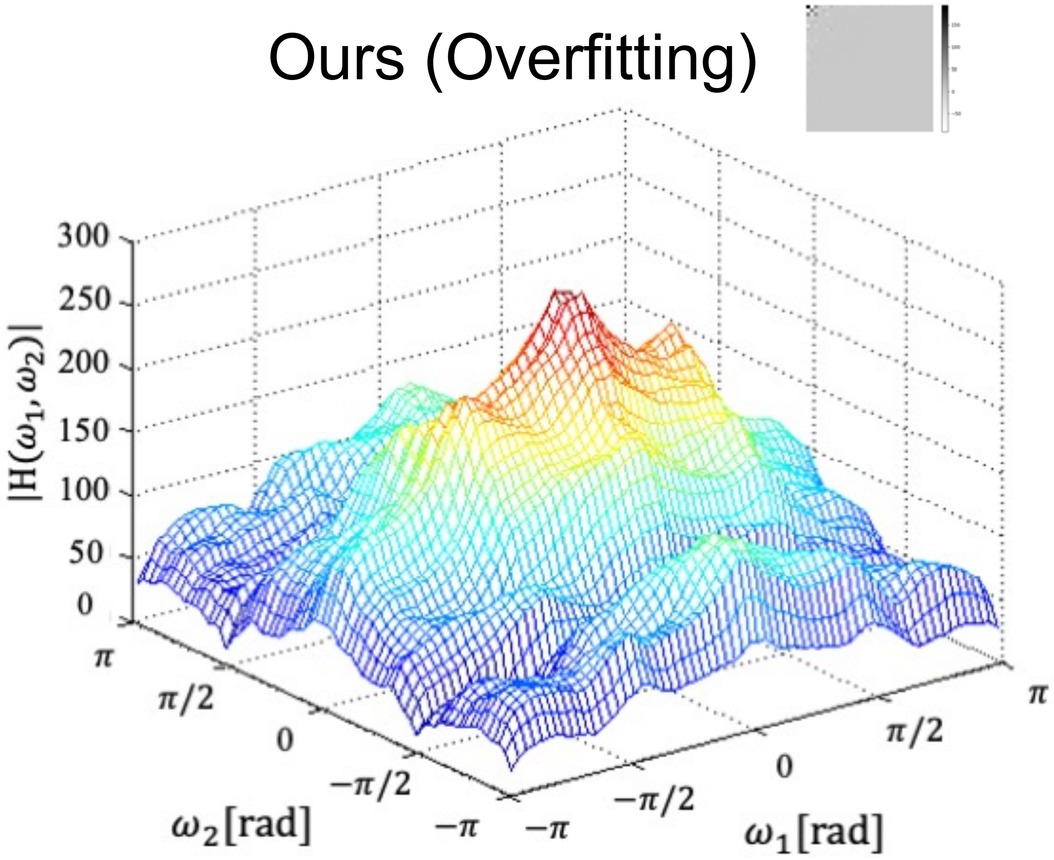
Fourier Transform of Filters

Fujii Laboratory

Ref. [4] (Generalization)



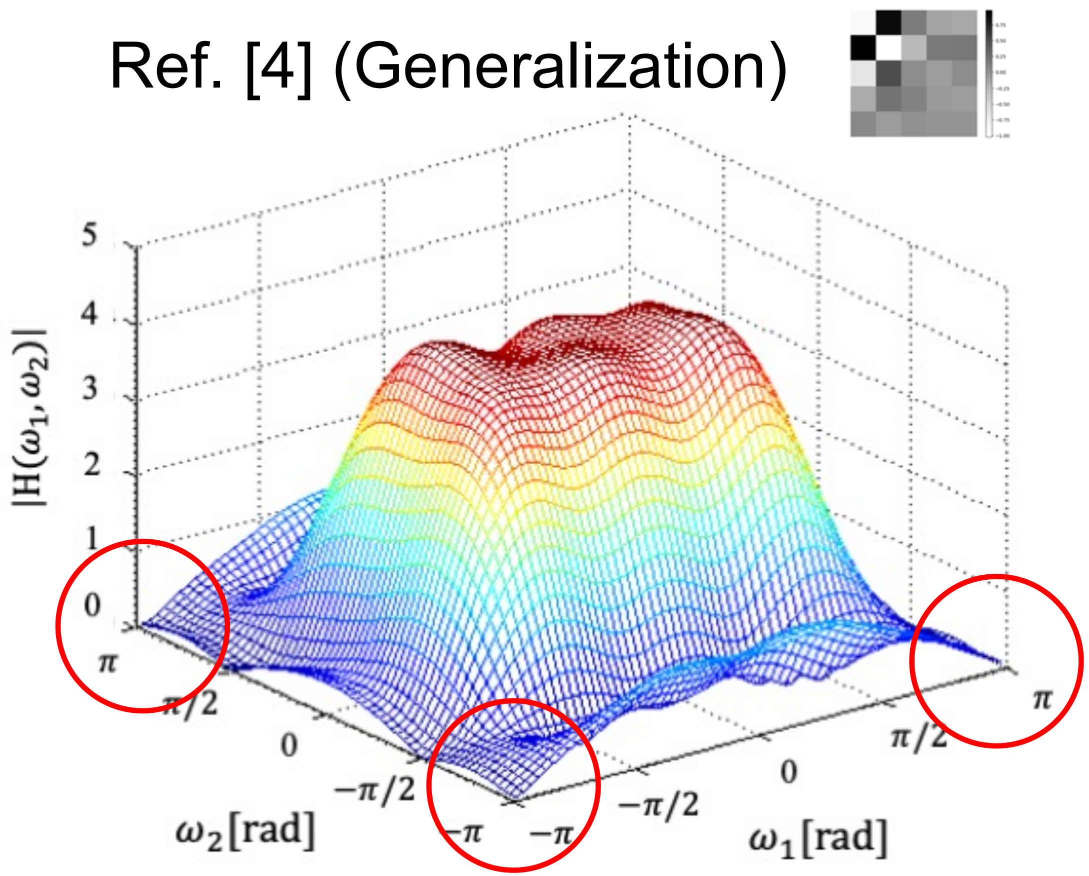
Ours (Overfitting)



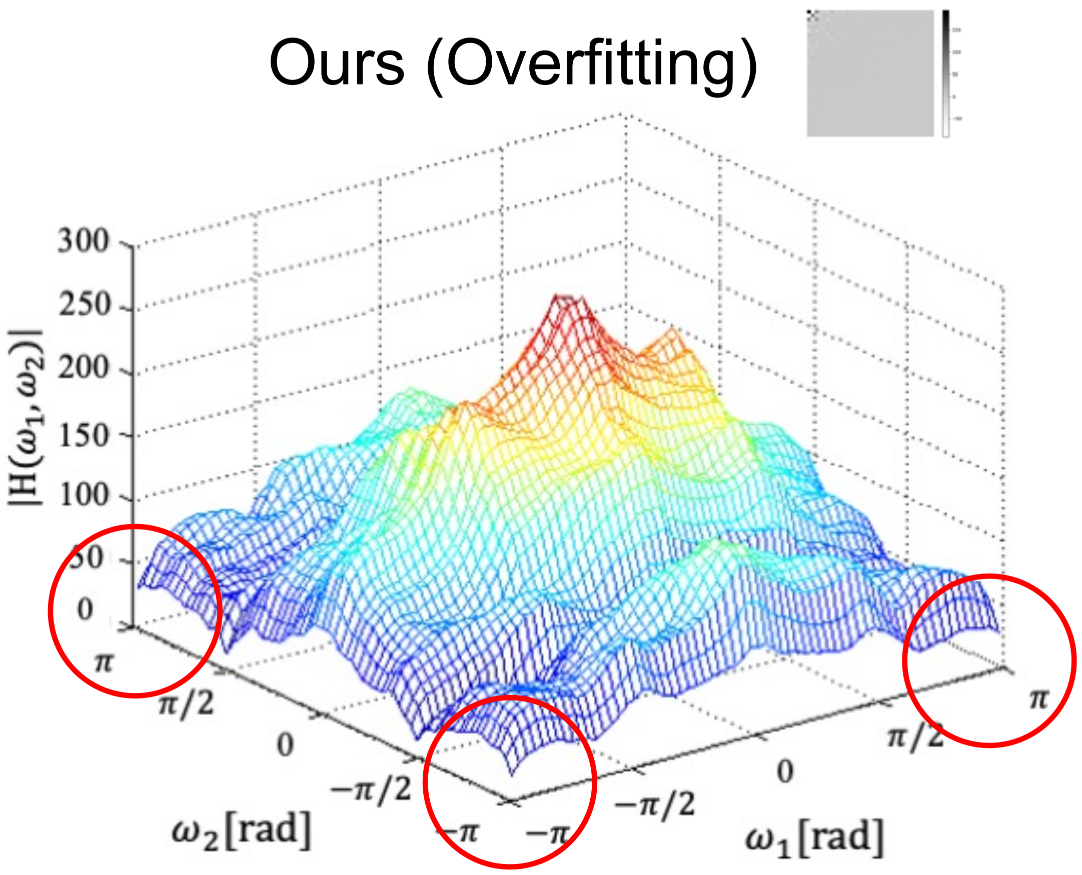
These graphs show the frequency responses of the convolution filters.

Fourier Transform of Filters

Ref. [4] (Generalization)



Ours (Overfitting)



From the high-frequency regions (circled by red), we can see that our filter pass more high-frequency components than Ref. [4].

Conclusion

Fujii Laboratory

- Image compression method using overfitting strategy and smaller network

- From experimental results…
  1. ”$\lambda_{y'} \geq \lambda_{\psi'}$” was optimal
  2. Convolutional filter passing more high-frequency components
  3. Better coding performance than generalized methods

This is the conclusion of this video. We proposed a overfitting method with a smaller network than conventional methods.

Conclusion

Fujii Laboratory

- Image compression method using overfitting strategy and smaller network

- From experimental results…
  1. ”$\lambda_{y'} \geq \lambda_{\psi'}$” was optimal
  2. Convolutional filter passing more high-frequency components
  3. Better coding performance than generalized methods

From the experiments, we found that $\lambda_{y'} \geq \lambda_{\psi'}$ was suitable for our method.

Conclusion

Fujii Laboratory

- Image compression method using overfitting strategy and smaller network

- From experimental results…
  1. "$\lambda_{y'} \geq \lambda_{\psi'}$" was optimal
  2. Convolutional filter passing more high-frequency components
  3. Better coding performance than generalized methods

From the visualization of the convolutional filter, we confirmed that our filter can pass the high-frequency components of images.

Conclusion

Fujii Laboratory

- Image compression method using overfitting strategy and smaller network

- From experimental results…
  1. "$\lambda_{y'} \geq \lambda_{\psi'}$" was optimal
  2. Convolutional filter passing more high-frequency components
  3. Better coding performance than generalized methods

These results show higher compression performance of our method than generalized method.

Conclusion

Fujii Laboratory

- Image compression method using overfitting strategy and smaller network
- From experimental results…
  1. "$\lambda_{y'} \geq \lambda_{\psi'}$" was optimal
  2. Convolutional filter passing more high-frequency components
  3. Better coding performance than generalized methods

That's all for our presentation. Thank you for watching this video.