

# 線形モデルに基づいたJPEG画像の品質向上のための 効率的な計算手法

高三佳己<sup>1\*</sup>, 中川 雅新<sup>1</sup>, 芦澤 恵太<sup>2</sup>, 山谷 克<sup>1</sup>

<sup>1</sup> 名城大学

<sup>2</sup> 静岡理工科大学

\*Email: 210781208@ccmailg.meijo-u.ac.jp

- ディープラーニングなどのAI技術の進歩に伴い, 学習した先験情報を有効に活用する逆問題アプローチへの関心が高まっている。
- 我々の研究は先験的な情報に頼ることなく, 数学的な性質のみに基づいて逆問題に取り組む手法に着目してきた。  
この手法の利点は汎用性と計算コストの低さである。
- 数値実験で実証するように, 先験情報を利用するアプローチは十分な学習を行った条件下では高精度の解を得ることができるが, 学習外のデータに適用した場合, その性能は著しく低下する。
- **本発表では**  
**局所線形性の仮定に基づいたJPEG品質向上の逆問題について,**  
**特に先験情報に依存しない手法に焦点を当てる。**

## 非可逆画像圧縮

- ・JPEG (JPEG ベースライン)

⇒ 様々なシチュエーションで使われている, 最も一般的

- ・JPEG2000

⇒ ・離散ウェーブレット変換, 圧縮効率が高い

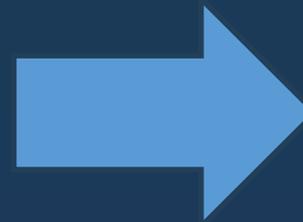
WebP, JPEG XL , JPEG XR , BPG, HEIF, など

---

**Point** JPEGベースラインは非可逆画像圧縮の中で最も一般的



JPEG画像  
(観測可能)



復元



オリジナル画像

**Point** より小さいコストで、より軽い処理で、正確な画像を作り出す

## 数学的な手法

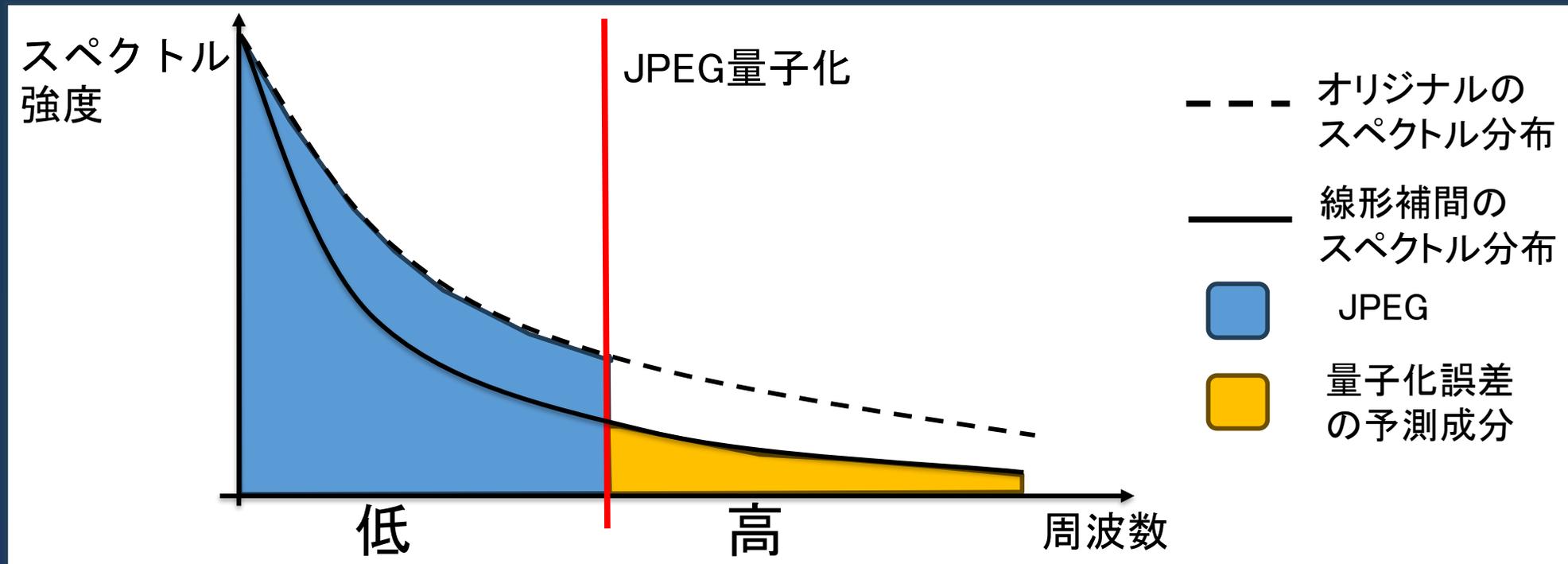
- N. Wiener(1949): Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications
- S. Geman, D. Geman(1984): Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of images
- Ashizawa, K., Yamatani, K. (2008) : Multiple prediction of DCT coefficients based on higher order derivative matching and its application to image compression.
- Morita, M., Kimura, Y., Nakagawa, M., Yamatani, K. (2023): Image Interpolation Technique by the PCA of the Gradient Distribution. In: ICIAM, Tokyo, <https://www.math.ucdavis.edu/~saito/confs/ICIAM23/morita.pdf>, last accessed 2024/5/25

## 機械学習を用いた手法

- Zhang, K., W. Zuo, Y. Chen, D. Meng, and L. Zhang, (2017): Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising.

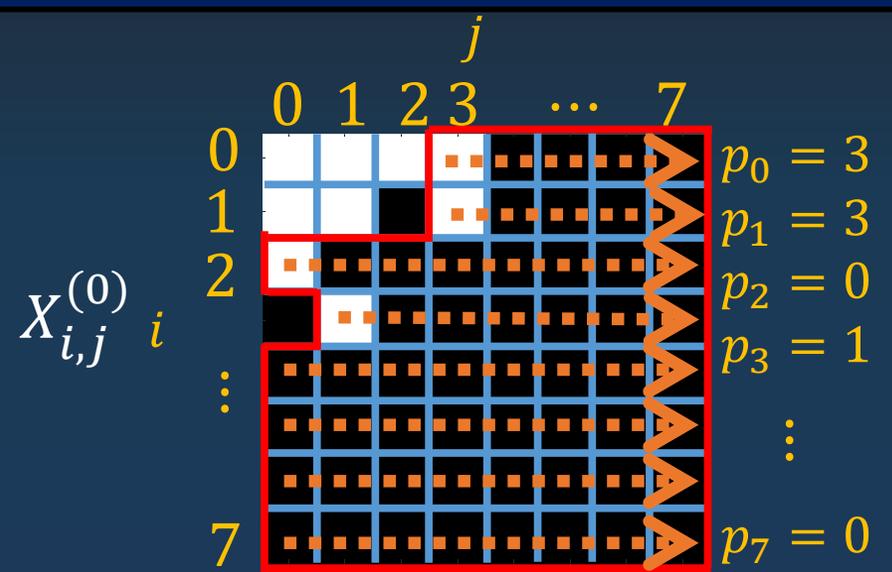
**Point** 数学的アプローチを用いたJPEGの品質向上に取り組んでいる

- 1) 自然画像の場合, スペクトル分布は漸近的に減衰する
- 2) 周波数成分の切り捨ては, 画像空間の局所的な線形性を失うことと等価

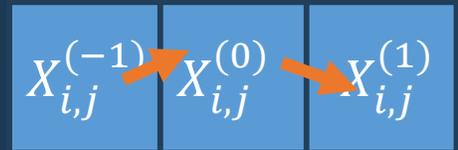


**Point** 切り捨てられたスペクトル分布を線形補間で復元する

# 復元手順

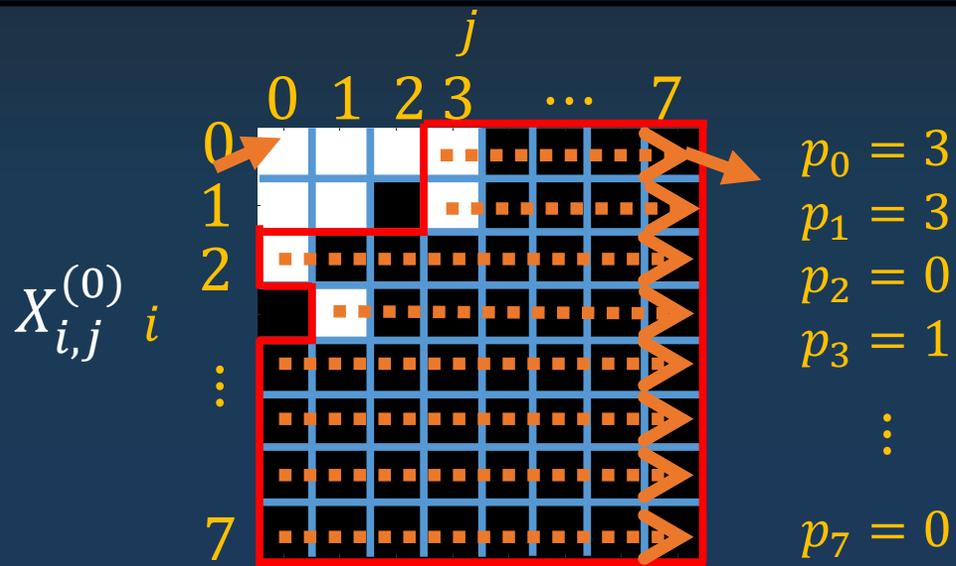


入力:  $X_{i,j}^{(-1)}$ ,  $X_{i,j}^{(0)}$ ,  $X_{i,j}^{(1)}$   
 (水平方向に隣接する3ブロック)

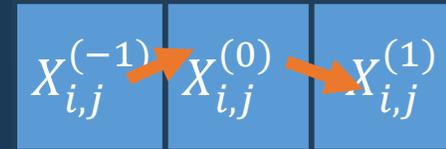


$X_{i,j}^{(-1)}$ ,  $X_{i,j}^{(0)}$ ,  $X_{i,j}^{(1)}$  からブロック境界の勾配を計算

# 量子化誤差の予測



入力:  $X_{i,j}^{(-1)}, X_{i,j}^{(0)}, X_{i,j}^{(1)}$   
(水平方向に隣接する3ブロック)



$X_{i,j}^{(-1)}, X_{i,j}^{(0)}, X_{i,j}^{(1)}$  からブロック境界の勾配を計算

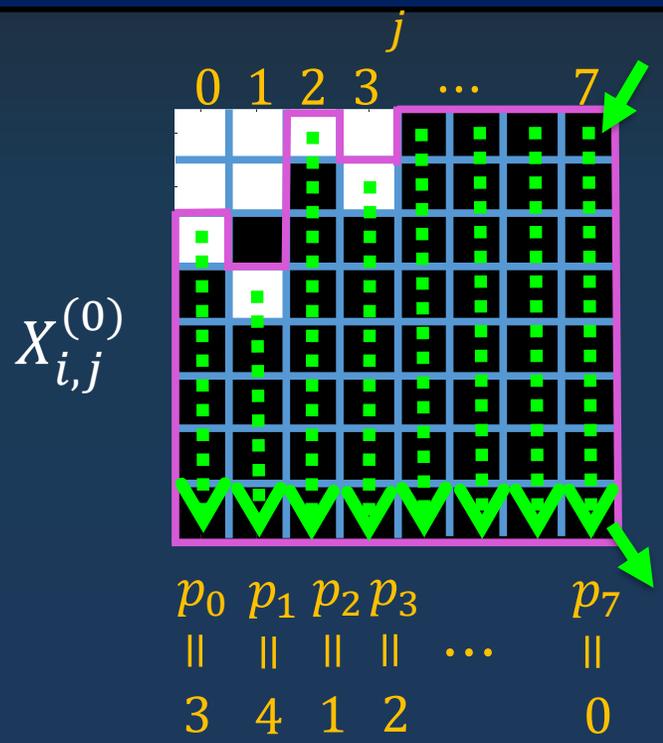
量子化誤差の予測式 :

$$\varepsilon_{i,j} = \left( \cos \frac{j\pi}{8} - 1 \right) X_{i,j}^{(0)} - \beta_j (L_i - (-1)^j R_i)$$

ブロック境界の勾配:  $L_i$  and  $R_i$   
予測した誤差:  $\varepsilon_{i,j}$

$$\{\beta_j\} = \{0.176, 0.045, 0.231, 0.208, 0.177, 0.139, 0.096, 0.049\}$$

**Point** 誤差  $\varepsilon$  はブロック境界の勾配の計算コストのみで推定する



入力:  $X_{i,j}^{(-1)}, X_{i,j}^{(0)}, X_{i,j}^{(1)}$   
(垂直方向に隣接する3ブロック)

$X_{i,j}^{(-1)}, X_{i,j}^{(0)}, X_{i,j}^{(1)}$  から  
ブロック境界の勾配  
を計算



量子化誤差の予測

$$\varepsilon_{i,j} = \left( \cos \frac{i\pi}{8} - 1 \right) X_{i,j}^{(0)} - \beta_i (T_j - (-1)^i B_j)$$

入力:  $T_i$  and  $B_i$   
出力:  $\varepsilon_{i,j}$

$$\{\beta_i\} = \{0.176, 0.045, 0.231, 0.208, 0.177, 0.139, 0.096, 0.049\}$$

**Point** 誤差  $\varepsilon$  はブロック境界の勾配の計算コストのみで推定する

# 実験の設定(データセットの説明)



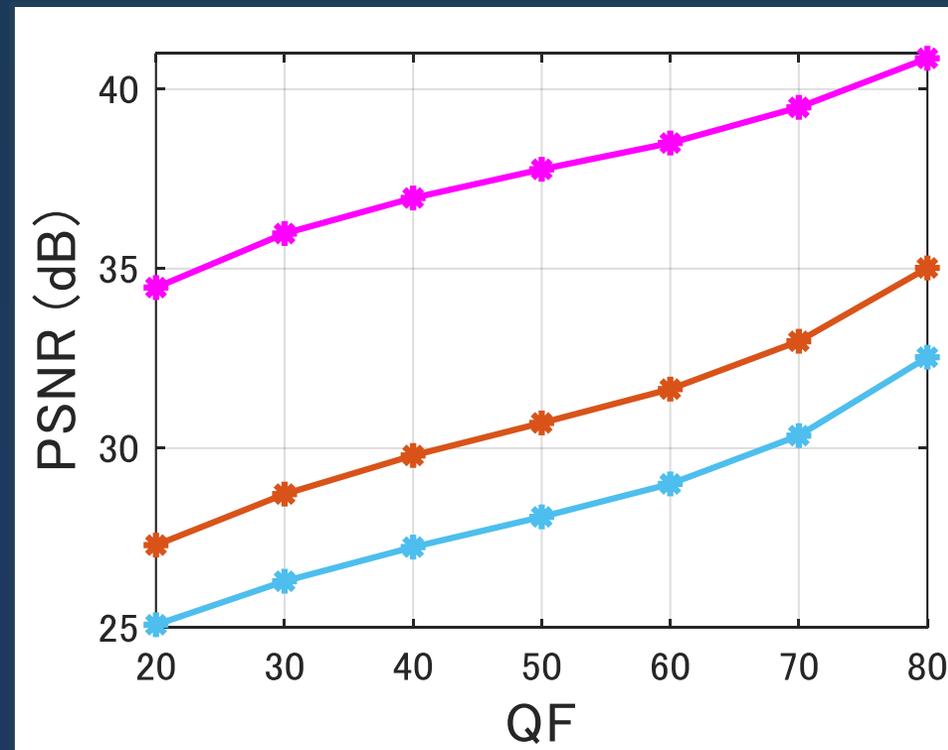
Kodak24: 24枚の512 × 768解像度の8bitグレースケール画像



# 実験の設定(圧縮parameterの説明)

QF: JPEG Quality Factorを8段階で指定して「入手可能なJPEG画像」とする

- ・量子化テーブルをパラメータ値QFで決定  
(bppを揃えても画像によっても劣化具合はさまざま)
- ・多くの画像でQF: 30でノイズが気になる, 一般的にはQF:70~80



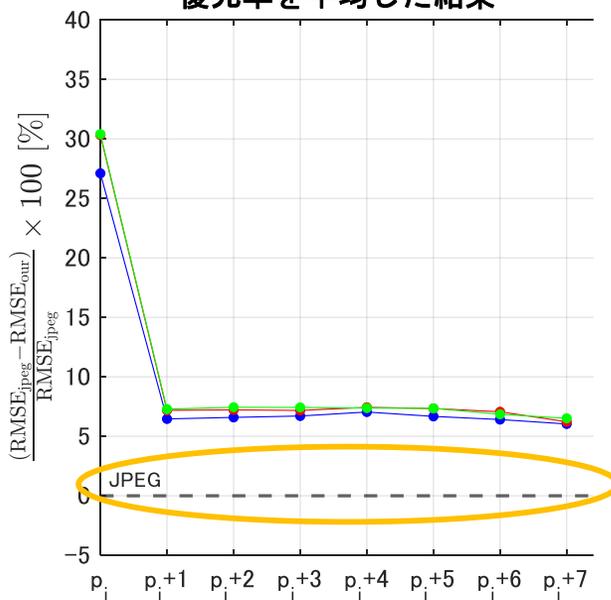
# 復元率 : JPEGで発生する誤差をどれくらい戻せるか

$$\frac{(RMSE_{JPEG} - RMSE_{JPEG-\varepsilon})}{RMSE_{JPEG}} \times 100 [\%]$$

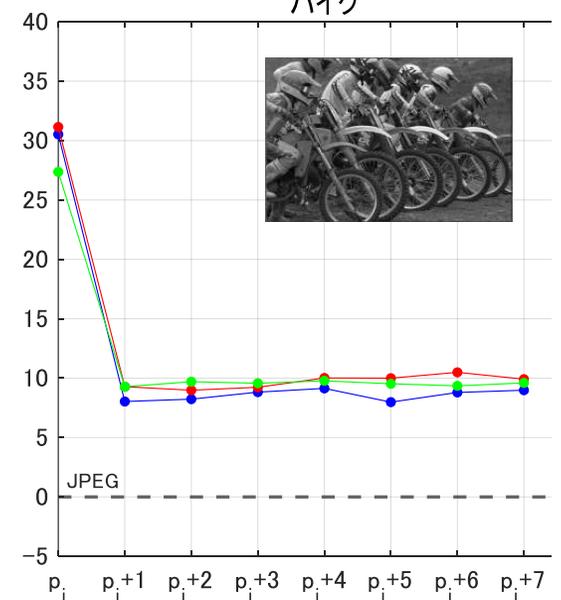
$RMSE_{JPEG}$ : JPEGの量子化誤差

$RMSE_{JPEG-\varepsilon}$ :  $\varepsilon$ による予測誤差

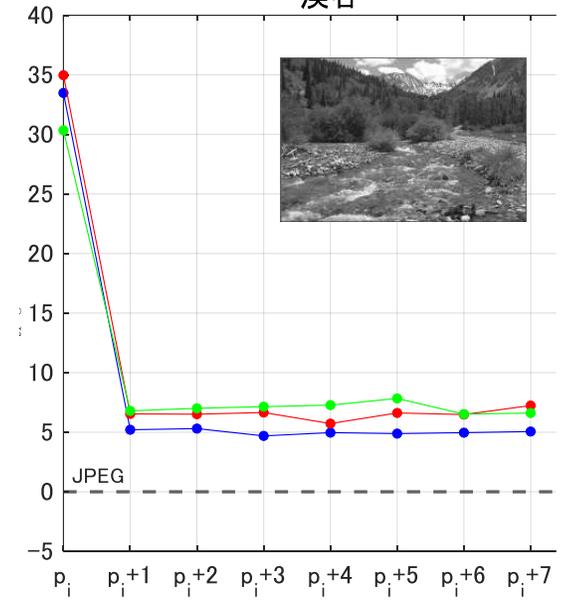
24枚の画像に対する復元率を平均した結果



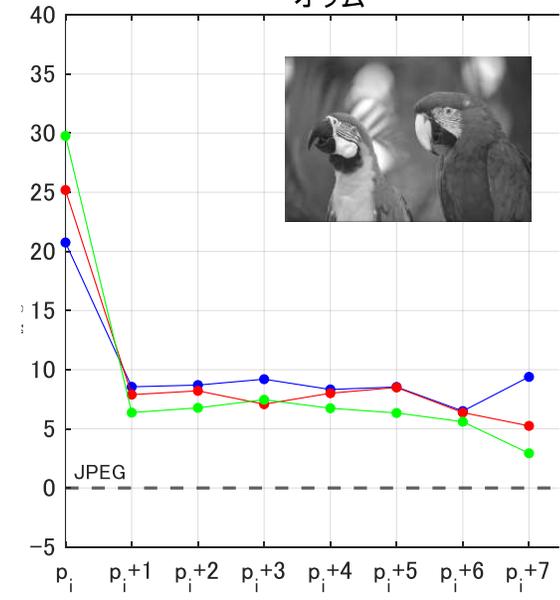
バイク



渓谷



オウム



—●— QF:20      —●— QF:50      —●— QF:80

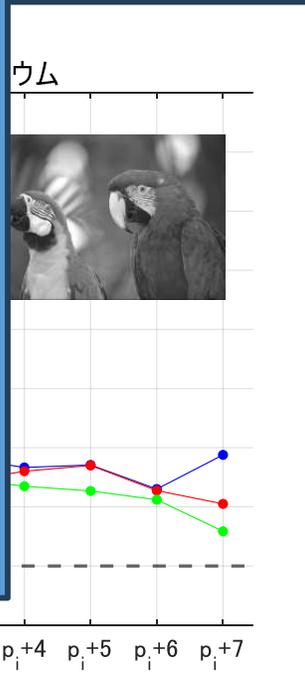
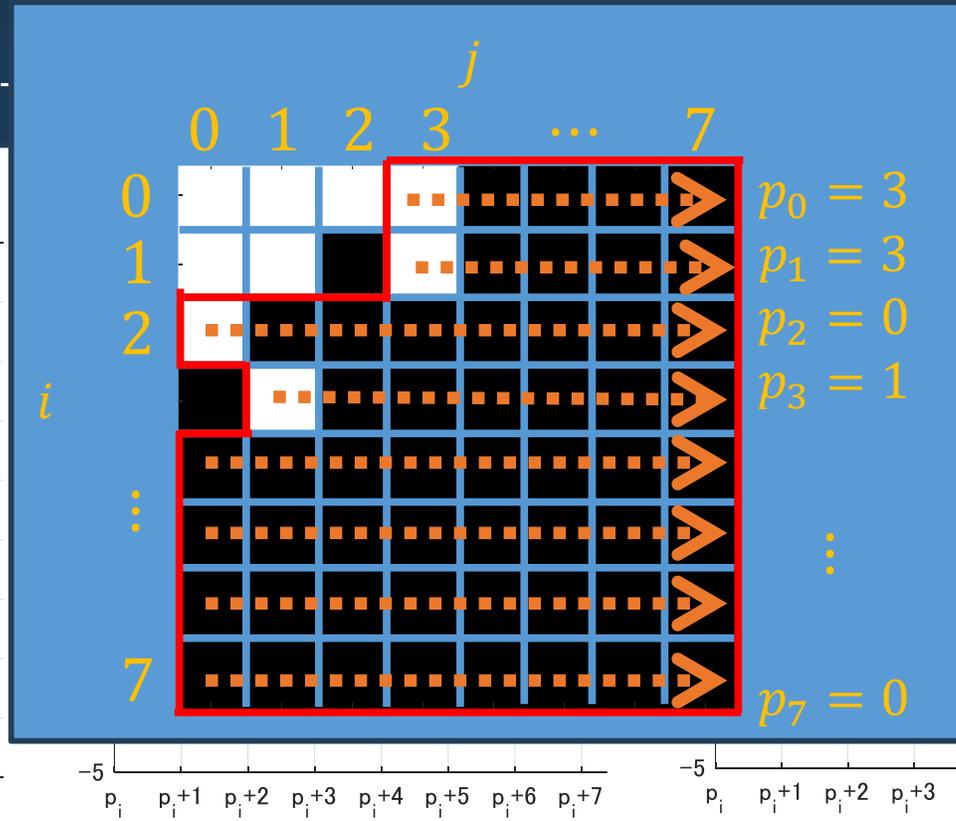
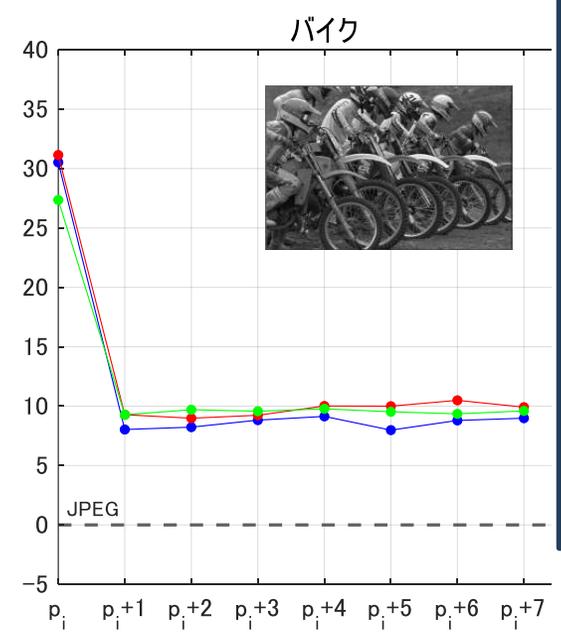
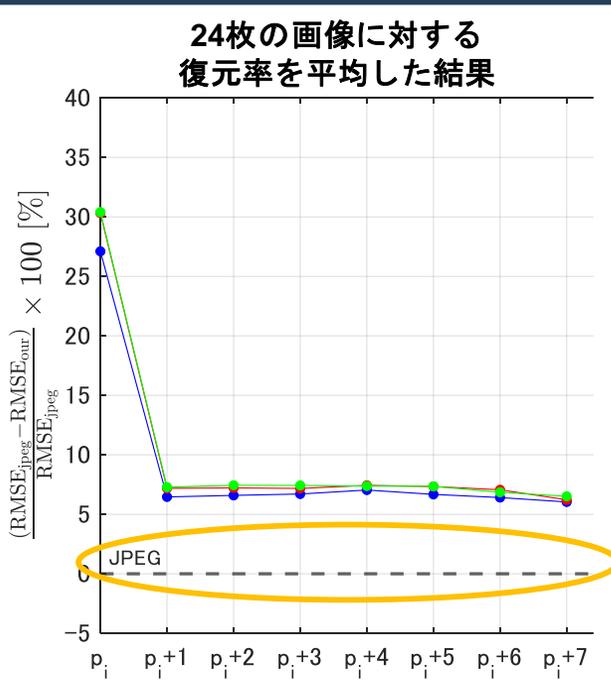
**Point**  $\varepsilon$ は, 画像とQFに関係なくDCT係数を復元した

# 復元率 : JPEGで発生する誤差をどれくらい戻せるか

$$\frac{(RMSE_{JPEG} - RMSE_{JPEG-\epsilon})}{RMSE_{JPEG}} \times 100 [\%]$$

$RMSE_{JPEG}$ : JPEGの量子化誤差

$RMSE_{JPEG-\epsilon}$



- QF:20
- QF:50
- QF:80

**Point**  $\epsilon$ は, 画像とQFに関係なくDCT係数を復元した

## 数学的な手法

### ● 今回の方式

- Morita, M., Kimura, Y., Nakagawa, M., Yamatani, K. (2023): Image Interpolation Technique by the PCA of the Gradient Distribution. In: ICIAM, Tokyo, <https://www.math.ucdavis.edu/~saito/confs/ICIAM23/morita.pdf>, last accessed 2024/5/25

### 1) OUR

#### 特徴

- ・ 先験情報を使わない
- ・ 線形モデリングを適用
- ・  $\varepsilon$ を使ってJPEGの量子化誤差を予測する方法

### 2) Linear modeling + PCA approach(LIP+PCA)

#### 特徴

- ・ 先験情報を使わない
- ・ 線形モデリングを適用
- ・ PCAを利用して画像空間のエッジを探る方法

## 機械学習を用いた手法

- Zhang, K., W. Zuo, Y. Chen, D. Meng, and L. Zhang, (2017): Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising.

### 3) DnCNN approach(DnCNN)

#### 特徴

- ・ オリジナル画像と対応するJPEG画像の残差を、20から80までのQFで学習する
- ・ 画像データセットは20,000枚の画像からなるTC12

## 1) CPU Time Ratio: Central Processing Units Time Ratio



*CPU Time Ratio* =

$$\frac{\text{Each approachで付加される処理の時間}}{\text{OURで付加される処理の時間}} \quad (\text{倍})$$

### 動作環境

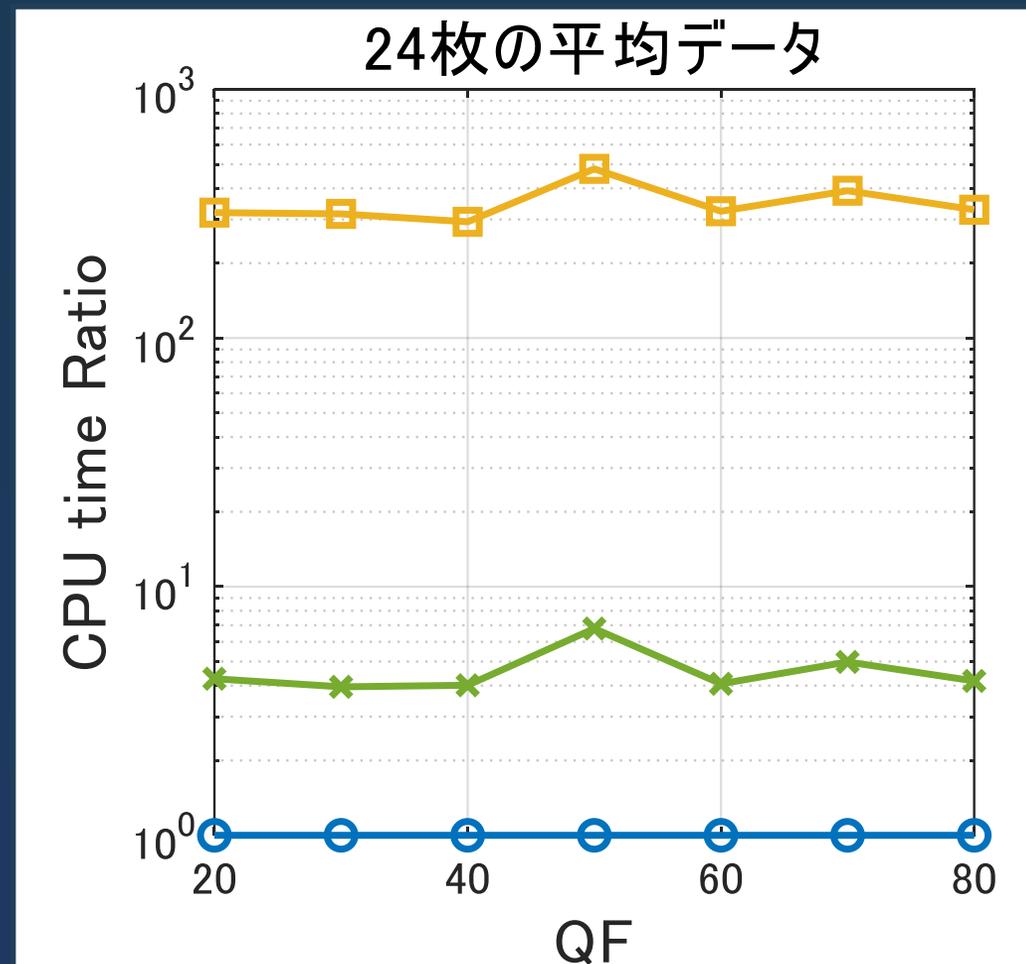
CPU: Intel(R) Core(TM) i7-10710U

Clock rates: 1.10GHz

OS: Windows 11 Pro

Memory: 16.0 GB

Language : MALAB R2024a



## 評価指標

2) PSNR: **画像全体**で見たときの相対誤差を測る

$$PSNR\ Gain_{OUR} = PSNR_{OUR} - PSNR_{JPEG} \quad [dB]$$

$$PSNR\ Gain_{PCA} = PSNR_{PCA} - PSNR_{JPEG} \quad [dB]$$

$$PSNR\ Gain_{DnCNN} = PSNR_{DnCNN} - PSNR_{JPEG} \quad [dB]$$

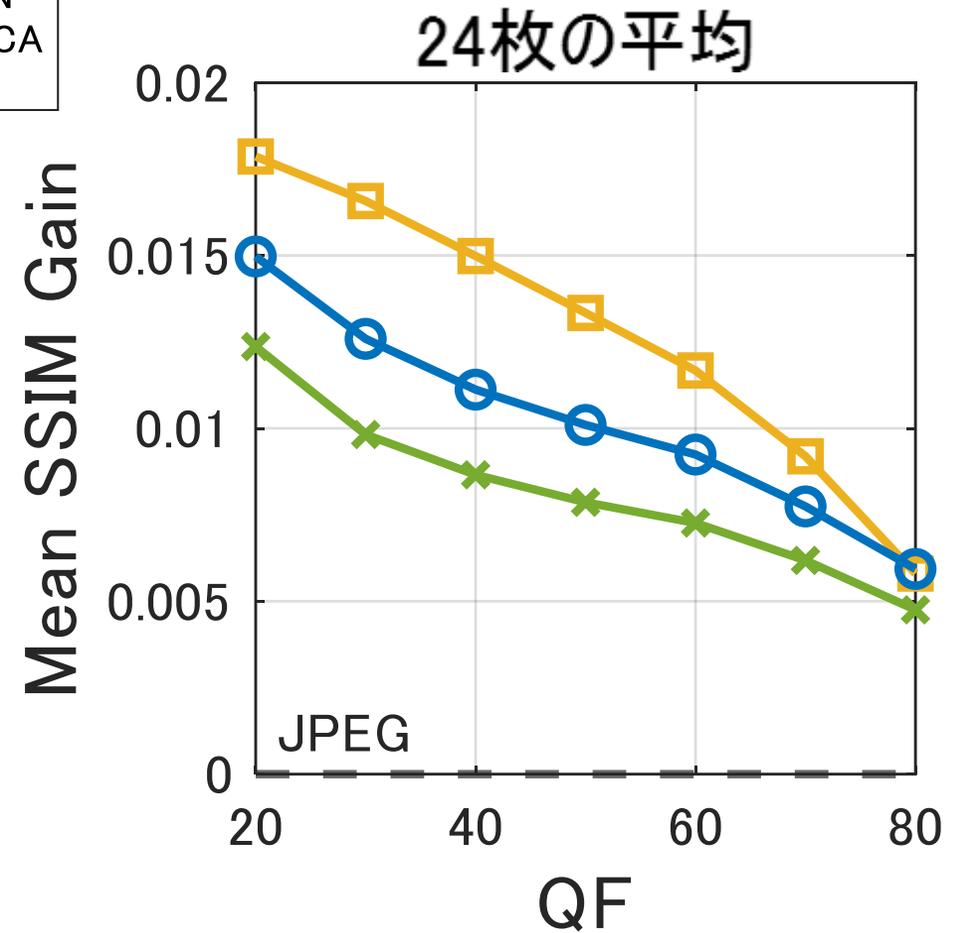
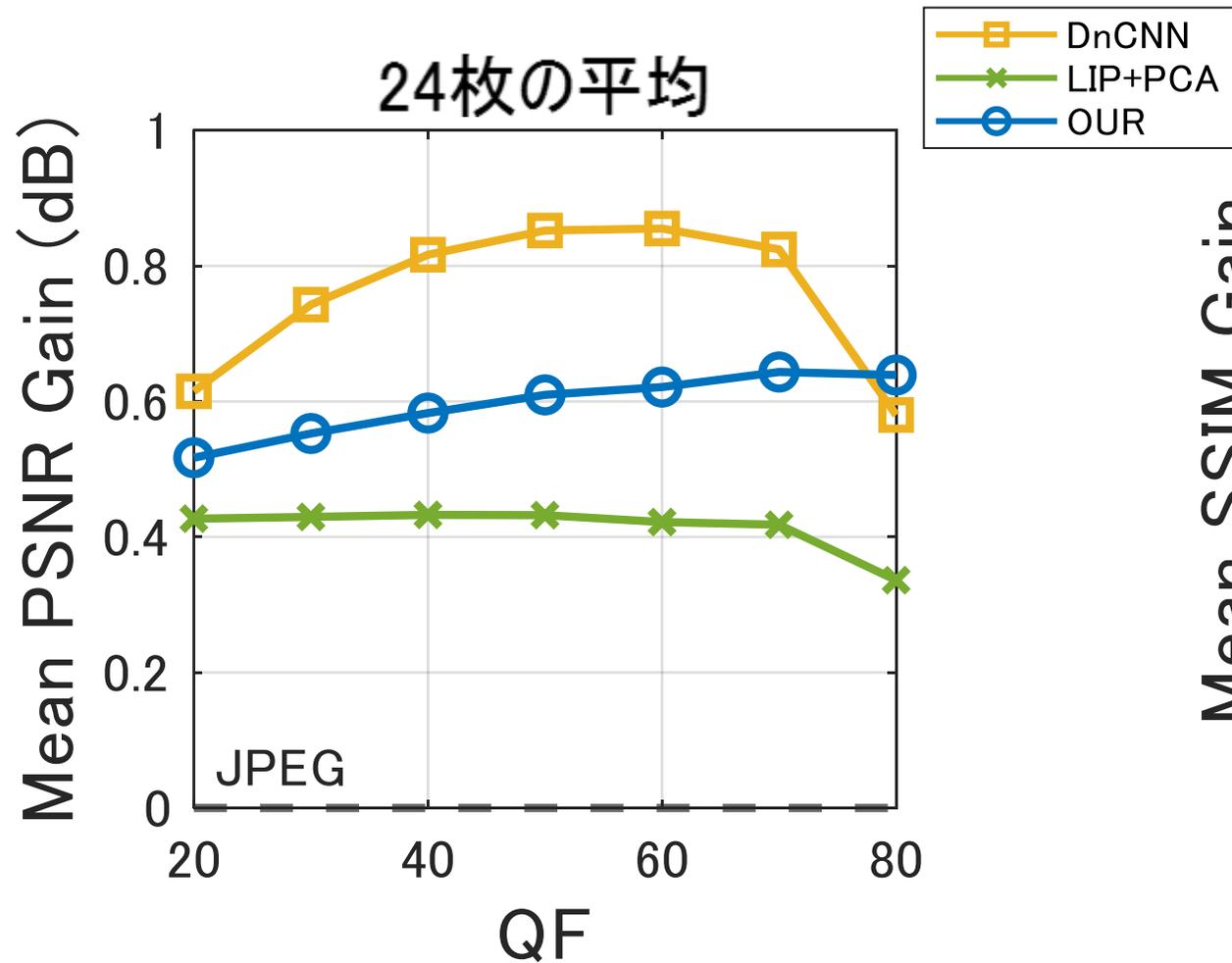
3) SSIM: **局所的**に画像を見た際の相対誤差を測る

$$SSIM\ Gain_{OUR} = SSIM_{OUR} - SSIM_{JPEG}$$

$$SSIM\ Gain_{PCA} = SSIM_{PCA} - SSIM_{JPEG}$$

$$SSIM\ Gain_{DnCNN} = SSIM_{DnCNN} - SSIM_{JPEG}$$

※画像処理の分野では主観評価に近いものとして多用される



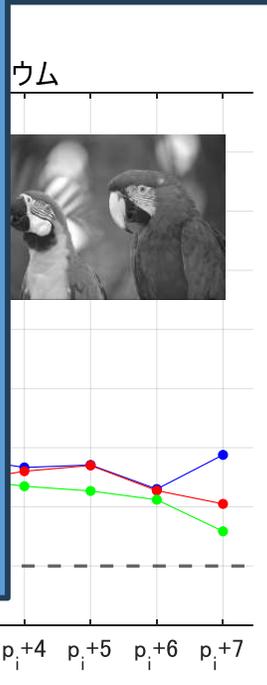
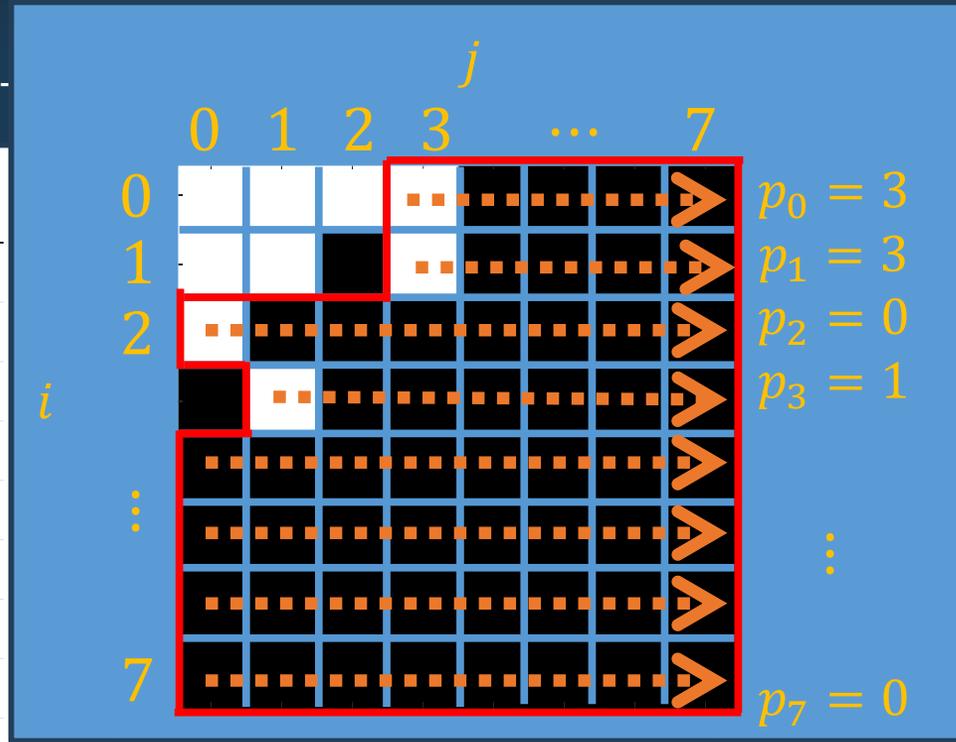
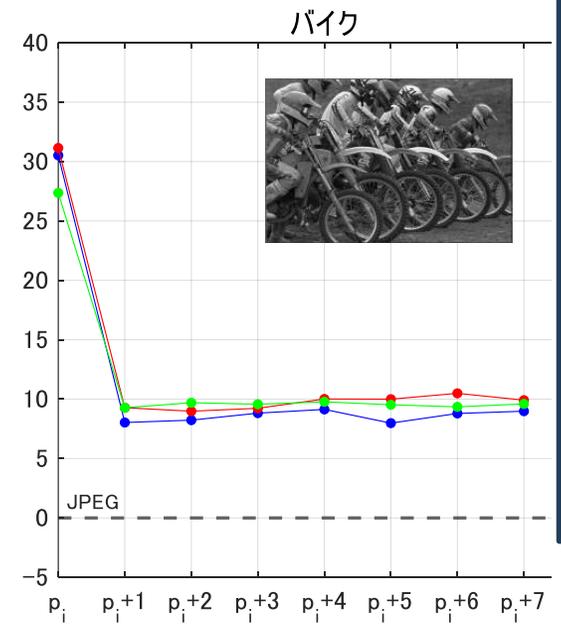
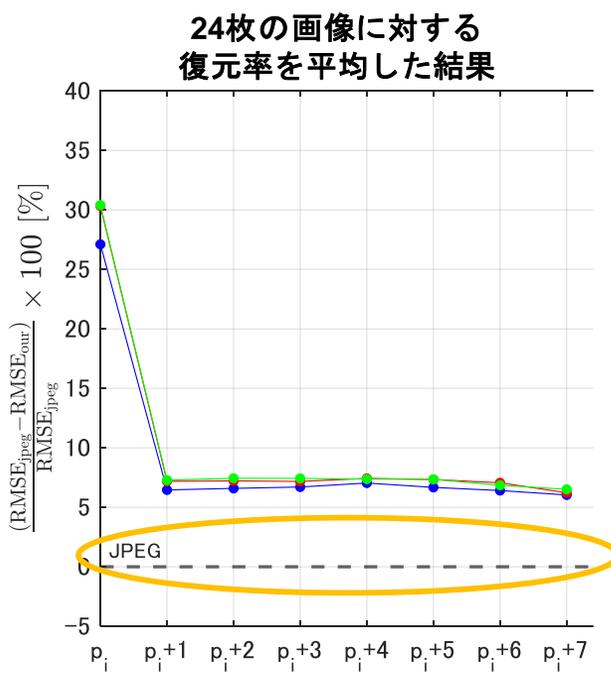
**Point** DnCNNは単一のモデルではすべてのQFでパフォーマンスを発揮できない

# 復元率 : JPEGで発生する誤差をどれくらい戻せるか

$$\frac{(RMSE_{JPEG} - RMSE_{JPEG-\varepsilon})}{RMSE_{JPEG}} \times 100 [\%]$$

$RMSE_{JPEG}$ : JPEGのDCT係数とオリジナルのDCT係数のRMSE

$RMSE_{JPEG-\varepsilon}$

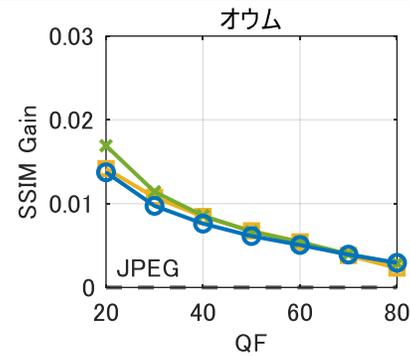
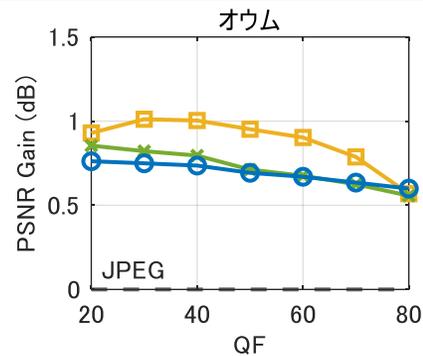


- QF:20
- QF:50
- QF:80

**Point**  $\varepsilon$ は, 画像とQFに関係なくDCT係数を復元した

# 数値実験(個別の画像)

オウム



・ オウムはPSNRとSSIMでの評価に差がある

・ PSNRは画像全体の相対誤差→エッジにおける誤差に影響されやすい

・ SSIMは局所的に誤差を正規化している→テクスチャとエッジの誤差が対等になる

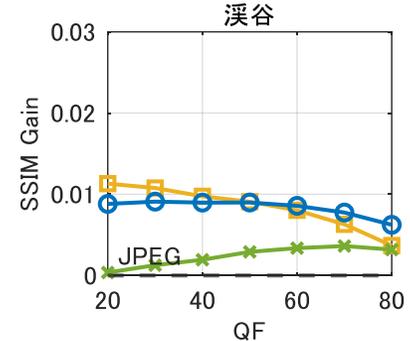
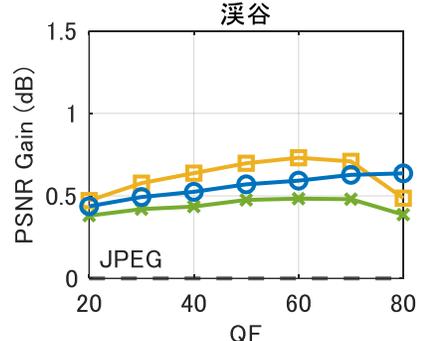
・ どちらも含まれているオウムの画像ではテクスチャの復元も重要となる

・ SSIMに着目するとOURはテクスチャの復元が得意なのではないか

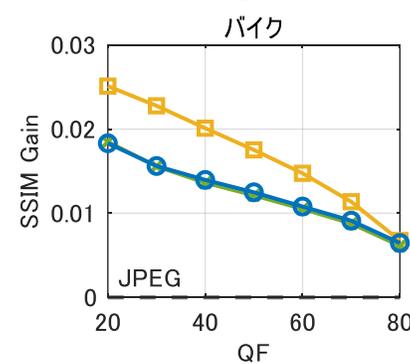
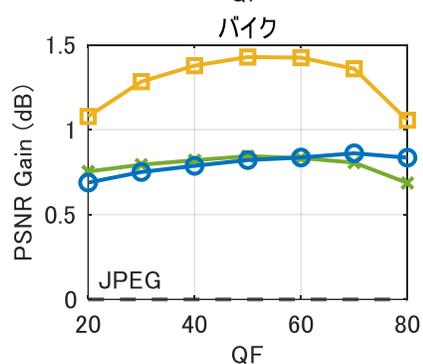
・ PSNRに着目するとエッジではDnCNNが優位である

→エッジブロックと非エッジブロックに分けて実験

渓谷



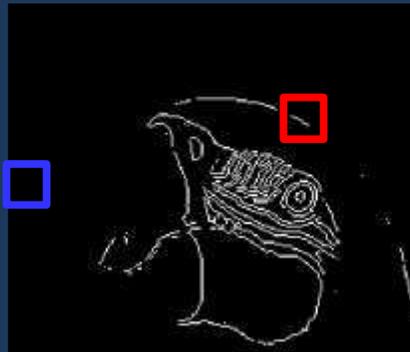
バイク



**Point** OURとDnCNNの差はエッジ部分に差が表れているのではないだろうか

# エッジブロックと非エッジブロックの分け方

24枚全体ではエッジブロック: 約30%



- edge関数  
@MATLABで判定
- sobelフィルタ
- ブロック内で1画素でもエッジ判定であったら  
**エッジブロック**と判定



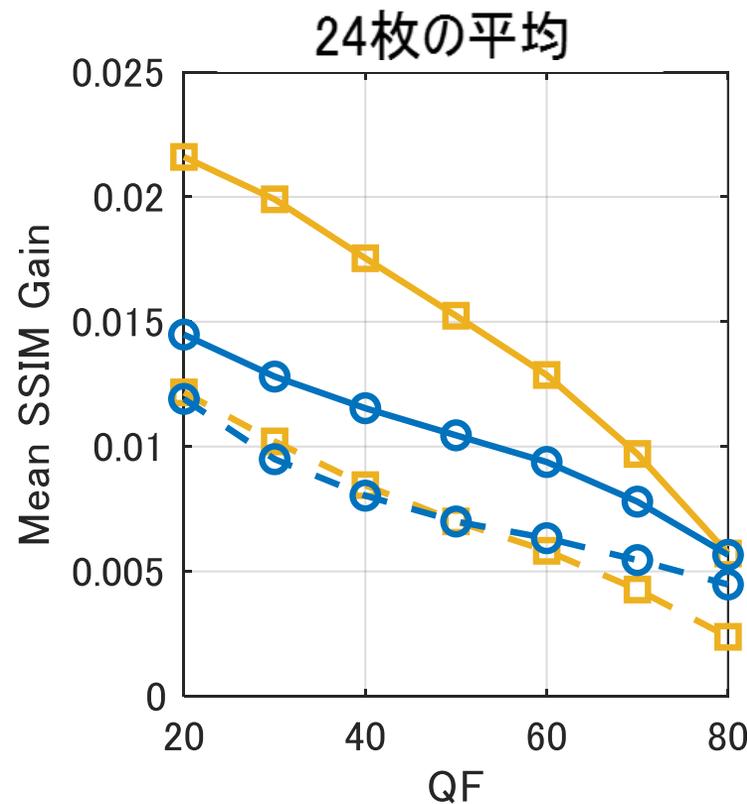
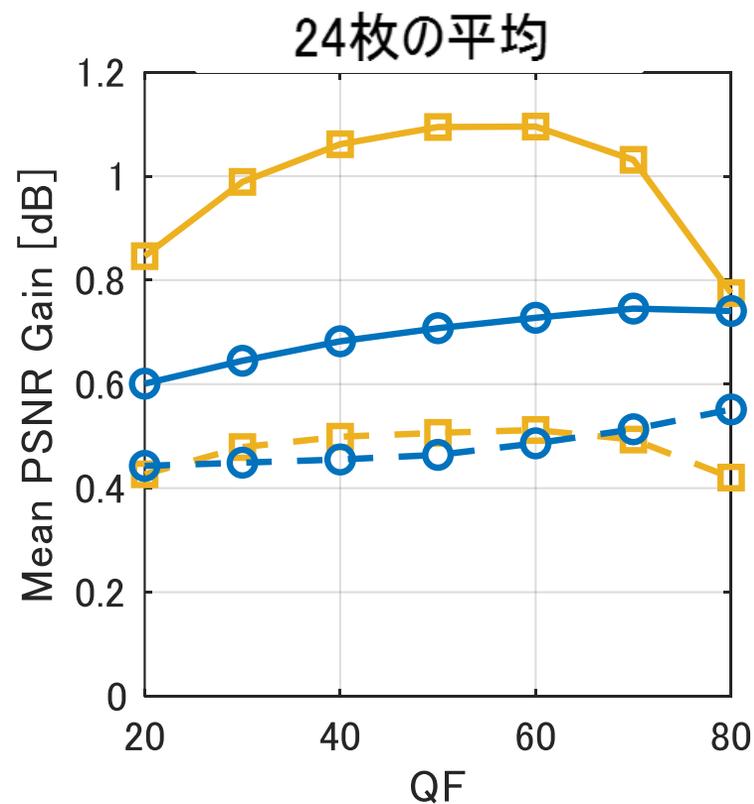
2705/6144  
(約44%)



2393/6144  
(約38%)



917/6144  
(約14%)

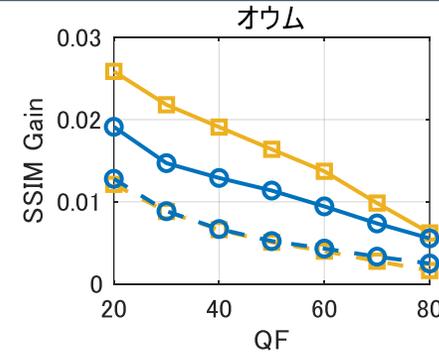
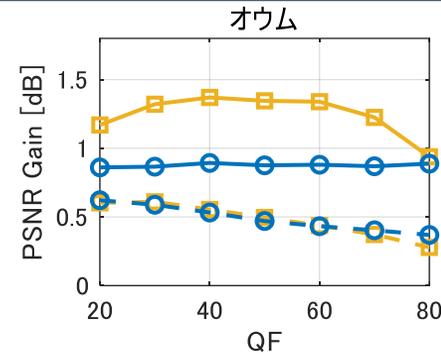


- ・ エッジブロックではOURとDnCNNの差が大きく開いている
- ・ 非エッジブロックではOURとDnCNNは差が現れなかった
- ・ OURとDnCNNの性能差はエッジにある

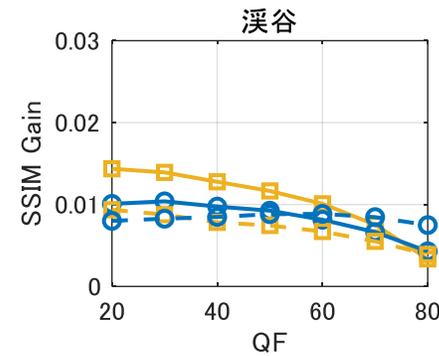
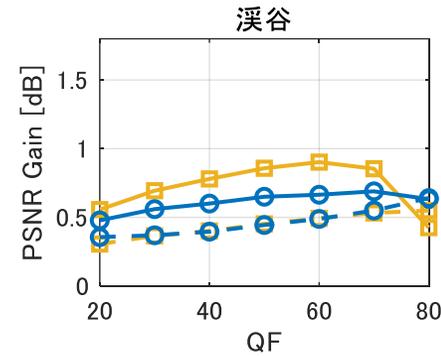
**Point** DnCNNとOURの性能差はエッジブロックにある

# エッジブロック・非エッジブロックの精度(個別の画像)

オウム



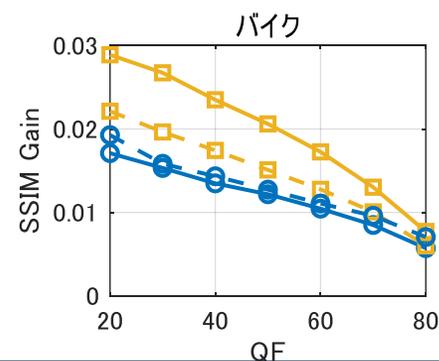
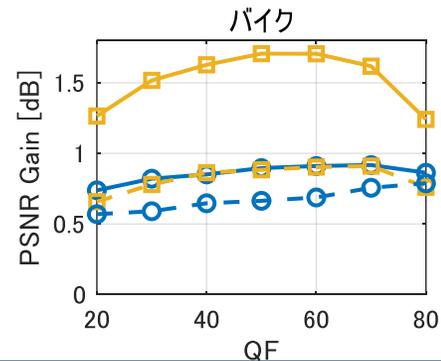
渓谷



・エッジがはっきりしているバイクとオウムではエッジブロックでの性能差が明確

・渓谷では特に高いQFではっきりと性能差が現れない  
→エッジブロックにもテクスチャが広がっている

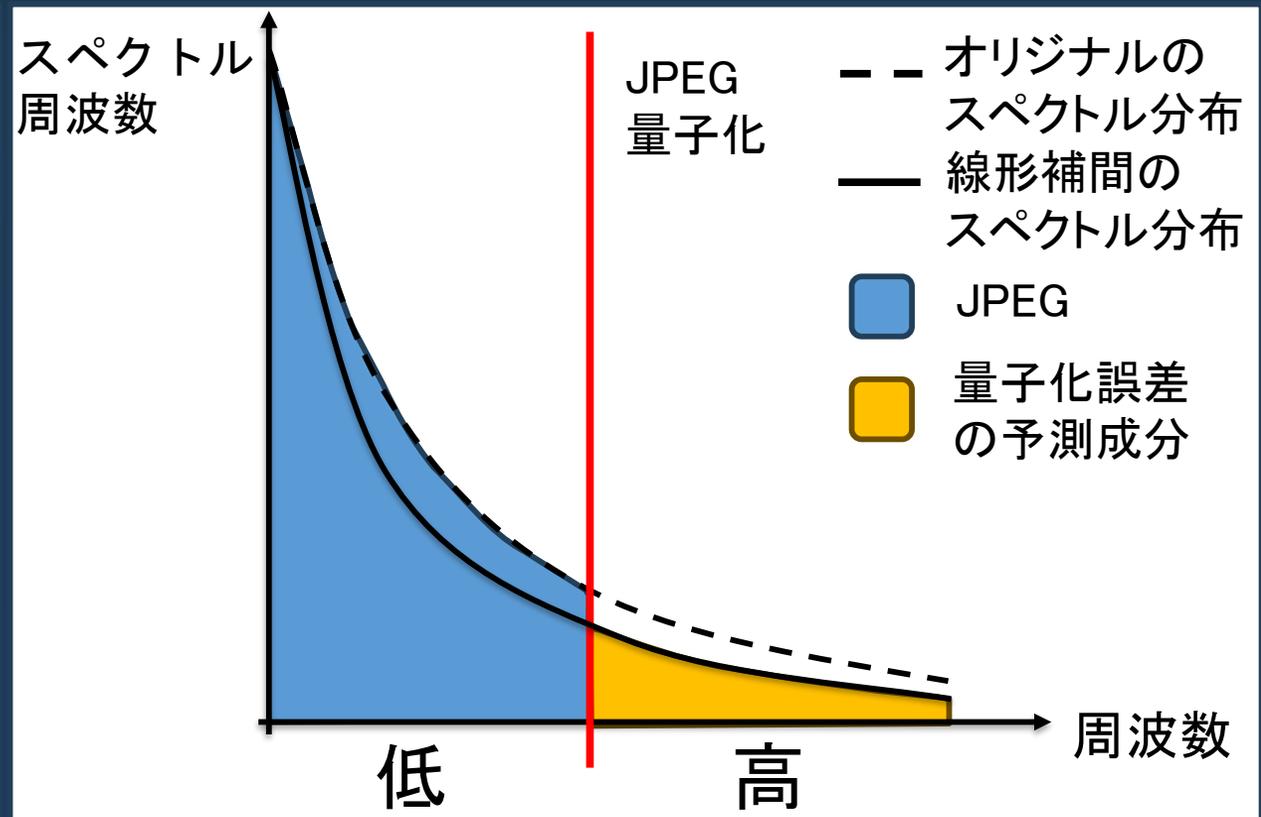
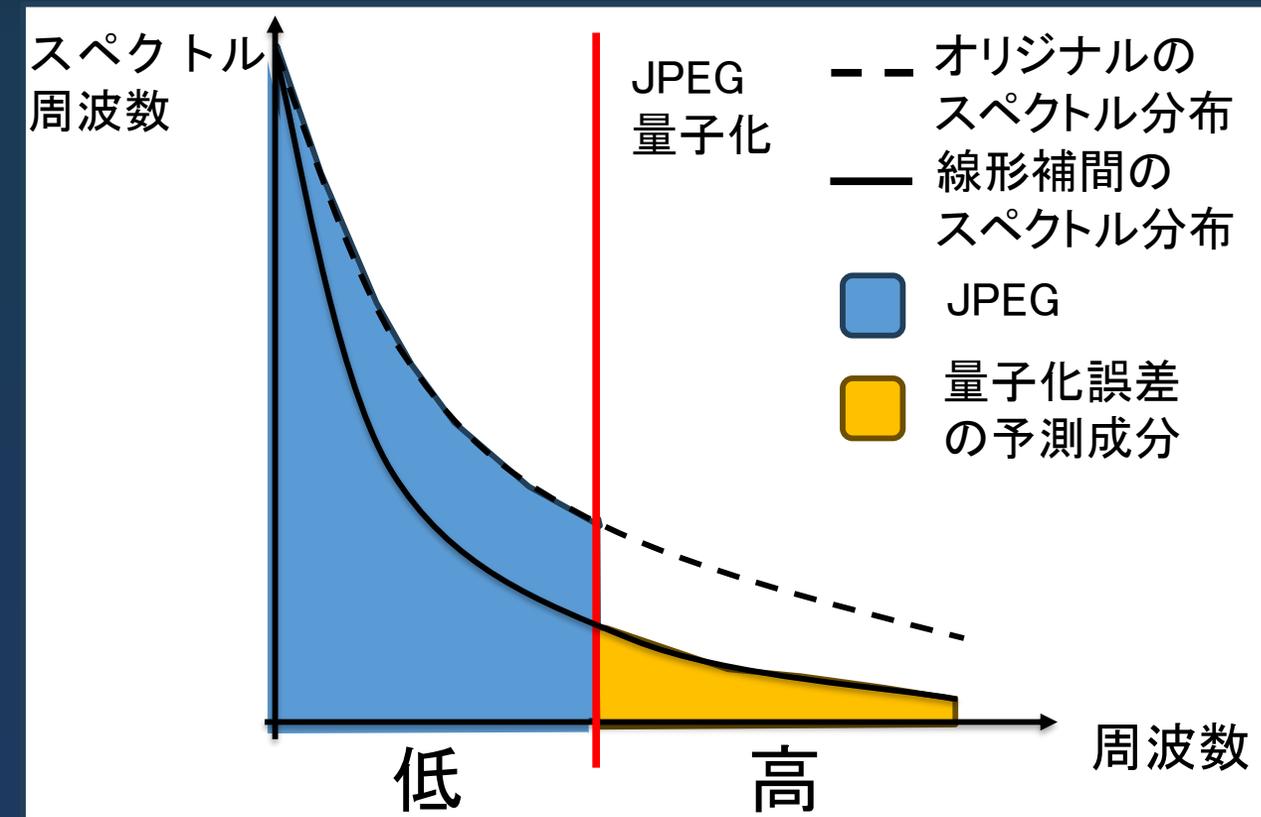
バイク



**Point** 非エッジブロックではDnCNNと同等の性能である

## エッジブロック

## 非エッジブロック



### Point

エッジ部分には先験情報を用いることが効果的であり、非エッジ部分には先験情報を用いることは効果的ではない

# 異なる解像度の (AI研究で良く用いられる) データセットへの適用

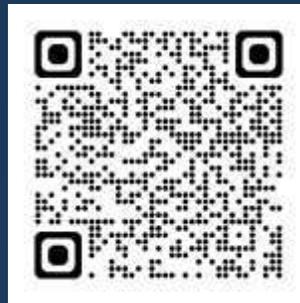
480 × 320



68種類が無償で入手可能

以降、BSD68と表記

768 × 512



24種類

Kodak24と表記

2K(1920x1080 他)



100種類

DIV2K100と表記

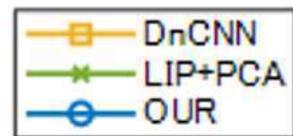
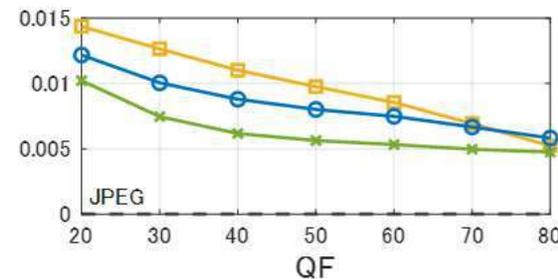
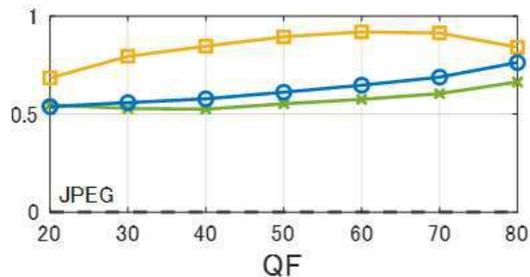
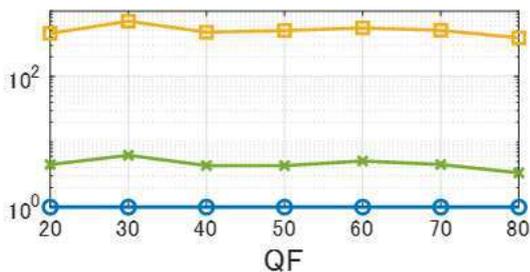
## 異なる解像度のデータセットへの適用

CPU time Ratio(sec)

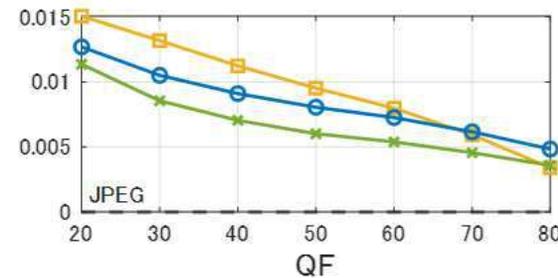
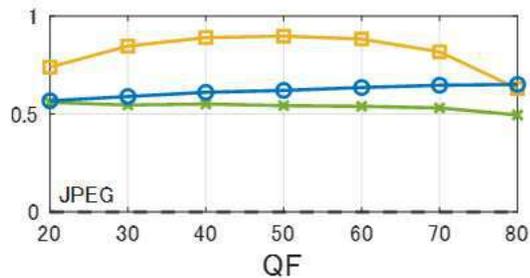
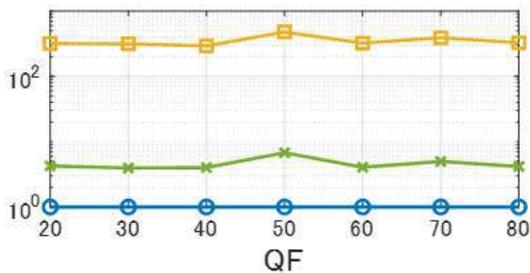
Mean PSNR Gain (dB)

Mean SSIM Gain

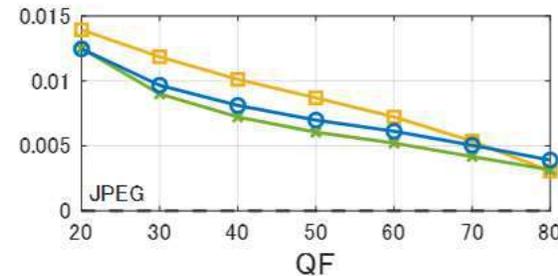
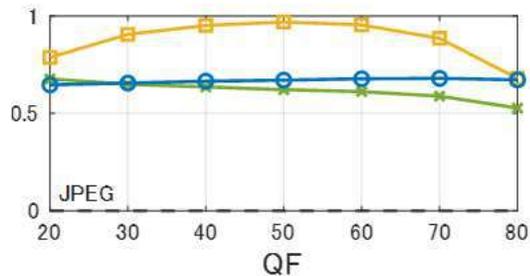
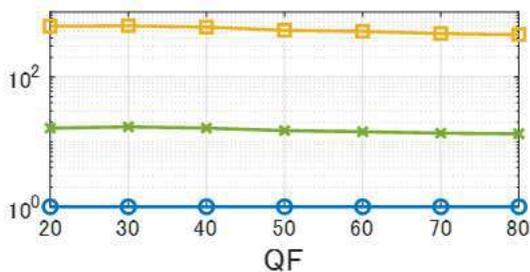
BSD68



Kodak24



DIV2K100



**Point** 異なる解像度でもkodak24と同様の結果が得られた

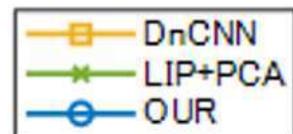
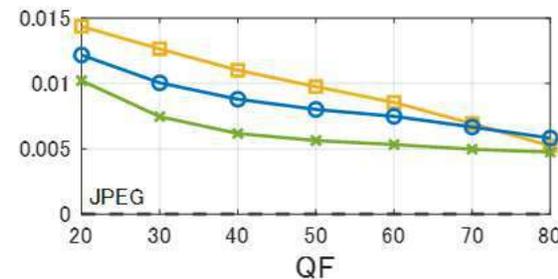
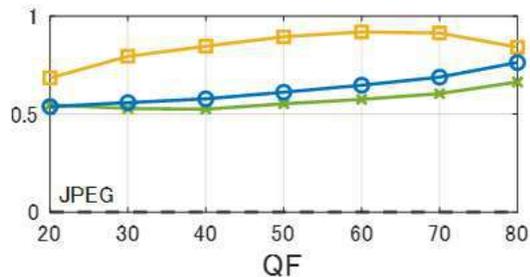
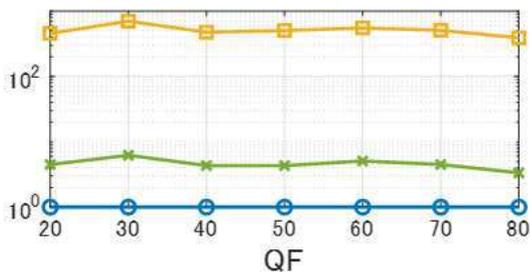
## 異なる解像度のデータセットへの適用

CPU time Ratio(sec)

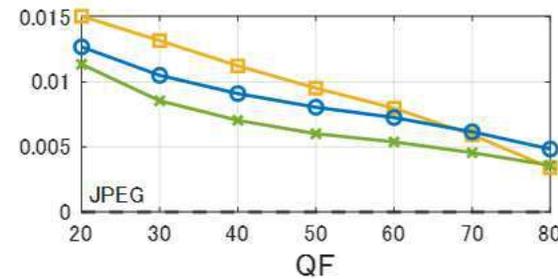
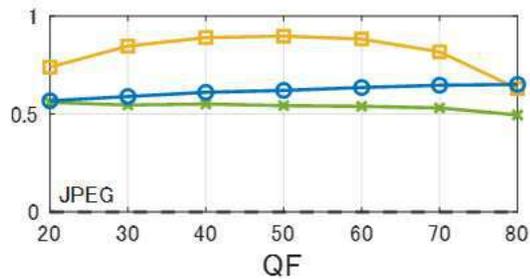
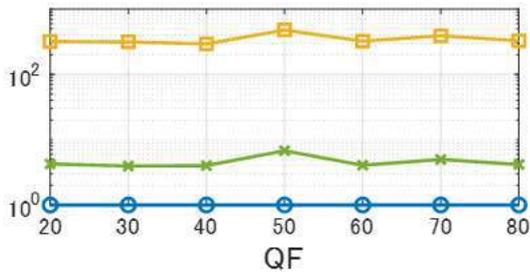
Mean PSNR Gain (dB)

Mean SSIM Gain

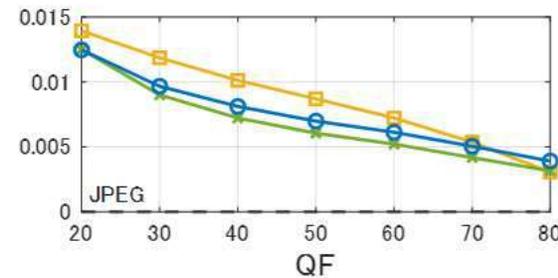
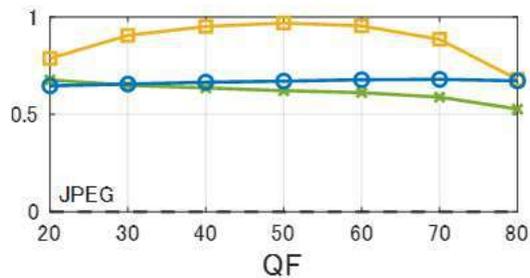
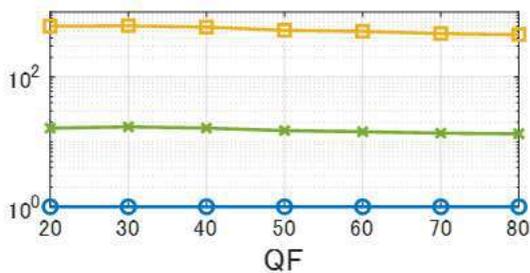
BSD68



Kodak24



DIV2K100



**Point** 異なる解像度でもkodak24と同様の結果が得られた

## <目的>

- ・JPEG画像からオリジナル画像に出来るだけ近い画像を少ない計算コストで得る事

## <我々が行ったこと>

- ・先験情報を用いない数学的な手法を適用
- ・画像空間内での局所的な線形性を仮定して、線形補間モデリングを適用
- ・ブロック境界の勾配情報からJPEG量子化誤差を予測するためのアルゴリズムを提案

## <結果>

- ・JPEG量子化誤差の予測に提案法が一定の効果を示すことが数値的に確認できた
- ・従来のアプローチと比較して、計算コストが最も低い
- ・エッジブロックでは先験情報を用いることが有効であることが分かった
- ・非エッジブロックでは先験情報を用いる手法と同等の精度が得られた

## <展望>

- ・今回復元しなかった領域に対するアプローチ
- ・OURを適用した画像を初期値とした機械学習への応用