

Recursive Deep Sign Retrieval : 画像圧縮における直交変換係数の符号情報を復元するニューラルネットワークの軽量化

Restoring Sign Information of Transformed Coefficients for Image Compression via
Recursive Deep Sign Retrieval

鈴木 圭* 都竹 千尋* 高橋 桂太* 藤井 俊彰*
Kei Suzuki Chihiro Tsutake Keita Takahashi Toshiaki Fujii

1 はじめに

1.1 背景

ブロック分割と直交変換に基づく画像圧縮では、変換係数に伴う正負の符号情報(以降、単に符号情報)の発生確率が等しいことが知られている。また、符号情報は無相関であると暗黙のうちに仮定されてきた。これら二つの条件下では、符号情報を圧縮することは不可能である。しかし、例えば画像圧縮の標準的な技術である JPEG において、符号情報が占めるビットストリームの割合は約 20% にも上る [1]。動画画像圧縮においても同様な傾向が見られ [2]、符号情報の圧縮は重要視されている。

符号情報の圧縮という挑戦的な課題に対して、これまでに幾つかのアプローチが提案されている。例えば、文献 [3] は符号情報がブロック間で相関をもつことを明らかにしている。この事実、条件付き確率に基づくエントロピー符号化で符号情報を圧縮できることを意味している。文献 [4, 5] は DCT 係数の絶対値を伝送し、復号済みの画素値から符号情報を復元した後に、復元誤差を符号化する手法を提案している。符号情報を高精度に復元することで、復元誤差に偏りが生じるため、その効率的な圧縮が可能となる。

文献 [4, 5] の方法論に基づいて、著者らは符号情報の復元を数理最適化問題として定式化し、復元誤差のさらなる圧縮を実現している。具体的には、符号情報の復元を phase retrieval 問題として定式化し [6]、これを deep neural network で解く手法を提案している [7]。この手法は、ネットワークのパラメータ数の増加に伴い符号情報の復元精度が向上する。しかし、符号化/復号器のストレージには上限があるため、少ないパラメータ数で符号情報を高精度に復元できることが望ましい。そこで本稿では、ネットワークの全ての隠れ層で共通のパラメータを用いる再帰的アーキテクチャ [8] に着目する。本研

究の目的を端的に述べると、このアーキテクチャを応用して、文献 [7] の符号情報の復元精度を維持しつつパラメータ数を大幅に削減することである。

1.2 記法

画像サイズを $I \times J$ 、ブロックサイズを $U \times V$ とする。二次元画像のあるブロックを $x_{s,t}$ 、ブロック内の画素値を $x_{s,t,u,v}$ と記述する。 s, t はブロックのインデックス、 u, v は画素のインデックスであり、各々 $0 \leq s \leq \lfloor I/U \rfloor - 1, 0 \leq t \leq \lfloor J/V \rfloor - 1, 0 \leq u \leq U - 1$ 、及び $0 \leq v \leq V - 1$ である。離散コサイン変換 (discrete cosine transformation, DCT) の正規直交基底を

$$c_{\omega_u, \omega_v, u, v} \sim \cos \frac{(u + 1/2)\omega_u \pi}{U} \cos \frac{(v + 1/2)\omega_v \pi}{V}$$

と記述する。 ω_u, ω_v は周波数インデックスであり、各々 $0 \leq \omega_u \leq U - 1, 0 \leq \omega_v \leq V - 1$ である。

数式を簡潔に記述するために、Einstein の縮約記法を用いる。単項式内に同一のインデックスが現れた場合、そのインデックスについて積和をとるという記述の規則である。つまり、積を計算するインデックスを明記しながら和の記号を省略できる。例えば、ブロック $x_{s,t}$ の DCT は

$$y_{s,t,\omega_u,\omega_v} = \sum_{u,v} x_{s,t,u,v} c_{\omega_u,\omega_v,u,v} \equiv x_{s,t,u,v} c_{\omega_u,\omega_v,u,v}$$

のように記述できる。ここで、 $y_{s,t,\omega_u,\omega_v}$ は DCT 係数である。この規則に従うと、逆 DCT は

$$x_{s,t,u,v} = y_{s,t,\omega_u,\omega_v} c_{\omega_u,\omega_v,u,v}$$

と記述できる。

2 文献 [6] の概要

2.1 符号化/復号器

文献 [6] を直感的に理解するために、符号情報の圧縮アルゴリズムを JPEG に実装した場合を考える。従っ

* 名古屋大学 大学院工学研究科 情報・通信工学専攻

て、ブロックサイズ $U \times V$ は 8×8 に固定される。JPEG で符号化/復号したブロックの画素値を $x_{s,t,u,v}$, DCT 係数を $y_{s,t,\omega_u,\omega_v}$ とする。DCT 係数の絶対値を $|y_{s,t,\omega_u,\omega_v}|$ と記述する。また、 $y_{s,t,\omega_u,\omega_v}$ の符号情報を $\text{sgn}(y_{s,t,\omega_u,\omega_v})$ と記述し、これを「原符号情報」と呼ぶ。次節の数理論最適化問題を解いて得られる符号情報を「復元符号情報」と呼び、 $\text{sgn}(\tilde{y}_{s,t,\omega_u,\omega_v})$ と記述する。この数理論最適化問題は、phase retrieval に基づいて $|y_{s,t,\omega_u,\omega_v}|$ から $\text{sgn}(y_{s,t,\omega_u,\omega_v})$ を復元するものである。以降、これを「sign retrieval (SR) 問題」と呼ぶ。図 1 に文献 [6] の符号化/復号器を示す。

符号化器：まず、DCT 係数の絶対値 $|y_{s,t,\omega_u,\omega_v}|$ を符号化する。次に、SR 問題を解き $\text{sgn}(\tilde{y}_{s,t,\omega_u,\omega_v})$ を得る。最後に、復元誤差

$$e_{s,t,\omega_u,\omega_v} = \text{sgn}(y_{s,t,\omega_u,\omega_v}) \oplus \text{sgn}(\tilde{y}_{s,t,\omega_u,\omega_v}) \quad (1)$$

を符号化する。ここで、 \oplus は XOR オペレータである。復元符号情報が原符号情報に近いほど、復元誤差に偏りが生じ、その効率的な圧縮が可能となる。

復号器：まず、絶対値 $|y_{s,t,\omega_u,\omega_v}|$ を復号し、SR 問題を解き $\text{sgn}(\tilde{y}_{s,t,\omega_u,\omega_v})$ を得る。次に、復元誤差 $e_{s,t,\omega_u,\omega_v}$ を復号し、復元符号情報との XOR を計算することで、原符号情報をロスレス復号する。最後に、原符号情報と絶対値 $|y_{s,t,\omega_u,\omega_v}|$ から JPEG 符号化/復号画像 x を得る。

2.2 SR 問題

Phase retrieval とは、信号の Fourier 変換の振幅情報から位相情報を求める問題のことであるが、実際のアプリケーションにおいては、位相が正しく復元された信号自体が重要視される。従って、phase retrieval は振幅情報から未知信号を復元する信号復元問題として定式化するのが定石である。Phase retrieval は未知信号を変数とする連立二次方程式として記述することができる。著者らの符号化シナリオにおける DCT と絶対値 $|y_{s,t,\omega_u,\omega_v}|$ の各々を、phase retrieval における Fourier 変換と振幅情報とみなすと、連立二次方程式は

$$y_{s,t,\omega_u,\omega_v}^2 = |x_{s,t,u,v} c_{\omega_u,\omega_v,u,v}|^2 \quad (2)$$

と立式できる。 x は JPEG 符号化/復号画像であり、phase retrieval における未知信号に相当する。

式 (2) は代数的に解くことが可能であり、両辺の平方根をとった後に逆 DCT を適用することで、

$$\tilde{x}_{s,t,u,v} = \pm |y_{s,t,\omega_u,\omega_v}| c_{\omega_u,\omega_v,u,v} \quad (3)$$

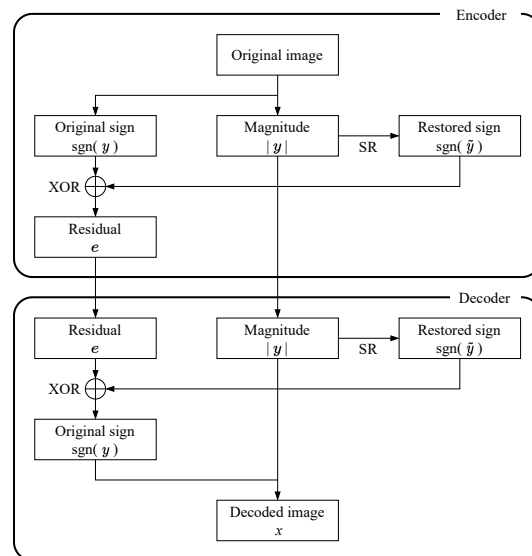


図 1: 符号化/復号器

が得られる。 \tilde{x} は式 (2) の解の候補であり、その個数は 2^{IJ} である*1。

解を一意に定めるには、 f を正則化関数として、式 (2) を正則化した

$$\min_{\tilde{x} \in \mathbb{R}^{I \times J}} f(\tilde{x}) \text{ s.t. } y_{s,t,\omega_u,\omega_v}^2 = |\tilde{x}_{s,t,u,v} c_{\omega_u,\omega_v,u,v}|^2 \quad (4)$$

を解けば良い。しかし、式 (3) の全ての \tilde{x} に対して $f(\tilde{x})$ を評価する必要がある。これは計算量が 2^{IJ} の組合せ最適化問題であるため、現代の計算機では解くことが不可能である。

文献 [9] では、正則化付き連立二次方程式を極めて効率的に解く近似解法が提案されている。この方法の本質を端的に述べると、連立二次方程式を不等式に書き換え、組合せ最適化問題から凸最適化問題に変換する点である。これを式 (4) に適用すると、

$$\min_{\tilde{x} \in \mathbb{R}^{I \times J}} f(\tilde{x}) \text{ s.t. } y_{s,t,\omega_u,\omega_v}^2 \geq |\tilde{x}_{s,t,u,v} c_{\omega_u,\omega_v,u,v}|^2 \quad (5)$$

が得られる。式 (4) と (5) の解は一致する可能性があり、その条件については文献 [9] を参照されたい。本稿では、式 (5) を「SR 問題」と呼ぶ。これを解いて \tilde{x} を復元した後に、その DCT 係数から符号情報を抽出したものが $\text{sgn}(\tilde{y}_{s,t,\omega_u,\omega_v})$ である。

2.3 SR アルゴリズム

文献 [6] では、正則化関数 f を次式で定義している。

$$f(\tilde{x}) = \|\Psi \tilde{x}\|_1 \quad (6)$$

*1 厳密には、直流成分と $y_{s,t,\omega_u,\omega_v} = 0$ の符号情報は自明であるため、解の候補数を減らすことができる。

ここで、 $\Psi\tilde{x}$ は \tilde{x} のウェーブレット変換、 $\|\cdot\|_1$ は ℓ_1 ノルムを表しており、これらは画像がウェーブレット領域でスパース性をもつという法則に基づいて採用している。

SR 問題の求解を目的として、Fienup の解法 [10]*2

$$\tilde{x}^{n+1} = \underset{\tilde{x} \in \mathbb{R}^{I \times J}}{\operatorname{argmin}} \|\Psi\tilde{x}\|_1 + \frac{\lambda}{2} \|\tilde{x} - z^n\|_2^2 \quad (7)$$

$$z^{n+1} = \underset{z \in \mathbb{R}^{I \times J}}{\operatorname{argmin}} \|\tilde{x}^{n+1} - z\|_2^2$$

$$\text{s.t. } y_{s,t,\omega_u,\omega_v}^2 \geq |z_{s,t,u,v} c_{\omega_u,\omega_v,u,v}|^2 \quad (8)$$

を考える。ここで、 λ は正則化パラメータである。 n は反復回数であり、式 (7) と (8) の反復の極限 $\lim_{n \rightarrow \infty} \tilde{x}^n$ が SR 問題の解である。式 (7) はウェーブレット縮退処理を用いることで解けるが、その詳細は文献 [6] を参照されたい。また、式 (8) の解法については後述する。以降、Fienup の解法で \tilde{x} を復元して、 $\operatorname{sgn}(\tilde{y}_{s,t,\omega_u,\omega_v})$ を抽出する一連の操作を「SR アルゴリズム」と呼ぶ。

3 DSR/RDSR アルゴリズム

3.1 DSR アルゴリズム [7]

ウェーブレット縮退はアフィン変換及び非線形関数によって構成され、これらは deep neural network における重み乗算、バイアス加算、及び活性化関数に類似した演算であることが知られている [11]。この類似性に基づいて、式 (7) を次式に置き換える。

$$\tilde{x}^{n+1} = g_{\theta^n}(z^n) \quad (9)$$

ここで、 θ^n は第 n 反復におけるネットワークの重みとバイアス、 $g_{\theta^n} : \mathbb{R}^{I \times J} \rightarrow \mathbb{R}^{I \times J}$ は θ^n でパラメタライズされた deep neural network を表す。Deep neural network を用いる最たる理由は、スパース性のような明示的な画像モデリングとは異なり、様々な画像データから暗黙的なモデルを学習できる点にあり、その有効性は広範な分野で確認されている。以降、式 (9) と (8) の反復で \tilde{x} を復元し、 $\operatorname{sgn}(\tilde{y}_{s,t,\omega_u,\omega_v})$ を抽出する一連の操作を「deep sign retrieval (DSR) アルゴリズム」と呼ぶ。

表 1 に DSR アルゴリズムのネットワーク構造を示す。Convⁿ は n に依存した重みとバイアスをもつ畳み込み層である。SR アルゴリズムと比較して、DSR アルゴリズムは符号情報を高速かつ高精度に復元できる点が特色

*2 この解法は交互最小化法としても知られているが、phase retrieval の分野では Fienup の解法と呼ばれる。著者らの方法論が phase retrieval から着想を得ていることから、本稿でも Fienup に敬意を表して後者の名前前で呼ぶことにする。

表 1: DSR のネットワーク構造 (stride は全て 1)

Layer	Kernel	In ch	Out ch	Act.	Input
Conv ₁ ⁿ	5 × 5	1	64	ReLU	z^n
Conv ₂ ⁿ	1 × 1	64	32	ReLU	Conv ₁ ⁿ
Conv ₃ ⁿ	3 × 3	32	1	Id.	Conv ₂ ⁿ

である。一方、各畳み込み層は n に依存したパラメータをもつため、ネットワーク全体のパラメータ数が極めて膨大となる問題がある。

3.2 RDSR アルゴリズム

畳み込み層のパラメータを全ての層で共有する再帰的アーキテクチャ [8] を採用し、符号情報の復元精度を維持しつつパラメータ数を大幅に削減することを目指す。ここでは、式 (7) を次式に置き換える。

$$\tilde{x}^{n+1} = g_{\theta}(z^n) \quad (10)$$

g_{θ} は n に依存しないパラメータ θ をもつ deep neural network である。ネットワークの構造は表 1 と同じものを採用する。反復回数の上限を N とした場合、このネットワークの総パラメータ数は DSR アルゴリズムの $1/N$ となる。以降、式 (10) と (8) の反復で \tilde{x} を復元して、 $\operatorname{sgn}(\tilde{y}_{s,t,\omega_u,\omega_v})$ を抽出する一連の操作を「recursive deep sign retrieval (RDSR) アルゴリズム」と呼ぶ。

ネットワークを学習する際のロス関数を定義する。まず、次の関数 m を定義する。

$$m_{|y|}(a_{s,t,\omega_u,\omega_v}) = \operatorname{median}(-|y_{s,t,\omega_u,\omega_v}|, a_{s,t,\omega_u,\omega_v}, |y_{s,t,\omega_u,\omega_v}|) \quad (11)$$

ここで、 $a_{s,t,\omega_u,\omega_v}$ は s, t, ω_u, ω_v で添字付けされた任意の DCT 係数を表す。また、 $|y_{s,t,\omega_u,\omega_v}|$ は復号済みの絶対値であり、既知である点に注意されたい。次に、関数 m を用いると、式 (8) の解は

$$z_{s,t,u,v}^{n+1} = m_{|y|}(\tilde{x}_{s,t,u,v}^{n+1} c_{\omega_u,\omega_v,u,v}) c_{\omega_u,\omega_v,u,v} \equiv \operatorname{proj}_{|y|}(\tilde{x}^{n+1}) \quad (12)$$

のように閉じた形で書くことが可能である。以降では、この演算を制約条件への「凸射影」と呼ぶ。関数 $\operatorname{proj}_{|y|}$ を用いると、RDSR アルゴリズムの復元画像 \tilde{x} は次の合成写像の出力として記述できる。

$$\tilde{x}(\theta) = \operatorname{proj}_{|y|} \circ g_{\theta} \circ \cdots \circ \operatorname{proj}_{|y|} \circ g_{\theta}(z^0) \quad (13)$$

最後に、RDSR のロス関数を次式で定義する。

$$L(\theta) = \sum_{x_{\text{train}} \in \mathcal{D}} \|\tilde{x}(\theta) - x_{\text{train}}\|_2^2 \quad (14)$$

\mathcal{D} は学習用の画像データセットである。

4 実験

4.1 実験条件

DSR と RDSR のネットワークを同一の条件で学習した。CLIC2021 で公開されている 585 枚の学習用画像をグレースケール化し、ランダムにクロップした 50,000 枚の画像 ($I \times J = 256 \times 256$) を収集した。以降では、これらの画像のことを D_{org} と呼び、式 (14) のロス関数の D として採用した*³。 D_{org} に含まれる画像に対して、 $QF = 50$ の量子化を適用して絶対値 $|y_{s,t,\omega_u,\omega_v}|$ を計算し、これを学習時の凸射影に用いた。初期画像 z^0 として、各ブロックを DCT 係数の直流成分で埋めたものを採用した。反復回数 n は最大で $N = 20$ とし、学習率 $2e - 4$ の Adam を用いて 50 エポック学習した。DSR と RDSR は共に JPEG 符号化/復号画像を復元するのに対し、学習用の画像データセットとして原画像を用いる理由については 4.6 節で後述する。

CLIC2021 で公開されている 60 枚のテスト画像をグレースケール化し、 $QF = 5, 10, \dots, 90, 95$ の量子化を適用した。このようにして得られた DCT 係数の符号情報を文献 [3, 4, 5] の手法, SR [6], DSR [7], 及び RDSR で圧縮した。DSR と RDSR アルゴリズムに関して、テスト時に様々な QF を用いる一方、学習時では $QF = 50$ に固定した理由については 4.6 節で後述する。

4.2 エントロピーの評価

いずれの手法も復号器で同一の画像が得られるため、歪性能に優劣はない。そこで、各手法の符号化対象となる情報に対して、符号情報あたりのエントロピー H *⁴を測り、レート性能のみを評価する。また、画素あたりの情報量 (bits per pixel, BPP) を次式で計算した。

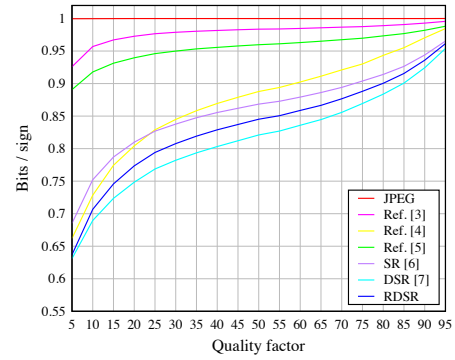
$$\text{BPP} = \frac{H \times \#\{y_{s,t,\omega_u,\omega_v} \neq 0, \omega_u \neq \omega_v \neq 0\}}{I \times J}$$

$\#\{\cdot\}$ は非零の交流成分の個数である。

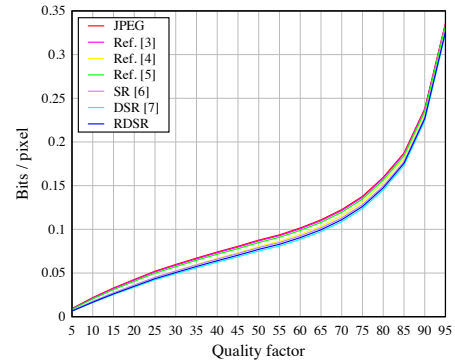
図 2(a) は各手法のエントロピー、(b) は BPP である。JPEG では符号情報に偏りが無いため、 QF によらずエントロピーは常に 1 ビットである。文献 [3, 4, 5, 6] の手法と RDSR を比較すると、全ての QF で RDSR が最も低いエントロピーを達成していることがわかる。また、DSR と RDSR を比較すると、後者は大幅にパラメータ数を削減しているにもかかわらず、エントロピーの増加

*³ DSR のロス関数は g_θ を g_{θ^n} に置き換えたものである。

*⁴ 文献 [3] の場合、原符号情報の条件付きエントロピーである。文献 [4, 5, 6], DSR, 及び RDSR の場合、復元誤差のエントロピーである。



(a) Bits/sign



(b) Bits/pixel

図 2: 符号情報あたりのエントロピーと BPP

表 2: 計算機環境

OS	Linux ubuntu 20.04 LTS
Language	Python 3.8.10 & PyTorch 1.11.0
Main memory	64 GB
CPU	Intel Core i9-10900K
GPU	NVIDIA GeForce RTX3090

表 3: 実行時間 [s]

[3]	[4]	[5]	SR [6]	DSR [7]	RDSR
0.91	88.18	6.33	85.69	4.61	4.63

はわずかであるため、再帰的アーキテクチャは有効であると言える。最後に、JPEG と RDSR を比較すると、エントロピーを 4-32 %, BPP を 1-8 % 削減できていることが確認できる。

4.3 計算コストの評価

計算機環境を表 2 に示す。DSR と RDSR の学習では表中の GPU を用いたが、他手法との公平な比較のため、全ての手法を CPU で実行した。各手法の実行時間を表 3 に示す。これは、実行開始から符号情報あたりのエントロピーを測り終えるまでの平均時間である。文献 [4] の手法では、復号済みの画素値から符号情報を総当たり的に復元する必要があるため、多くの実行時間が

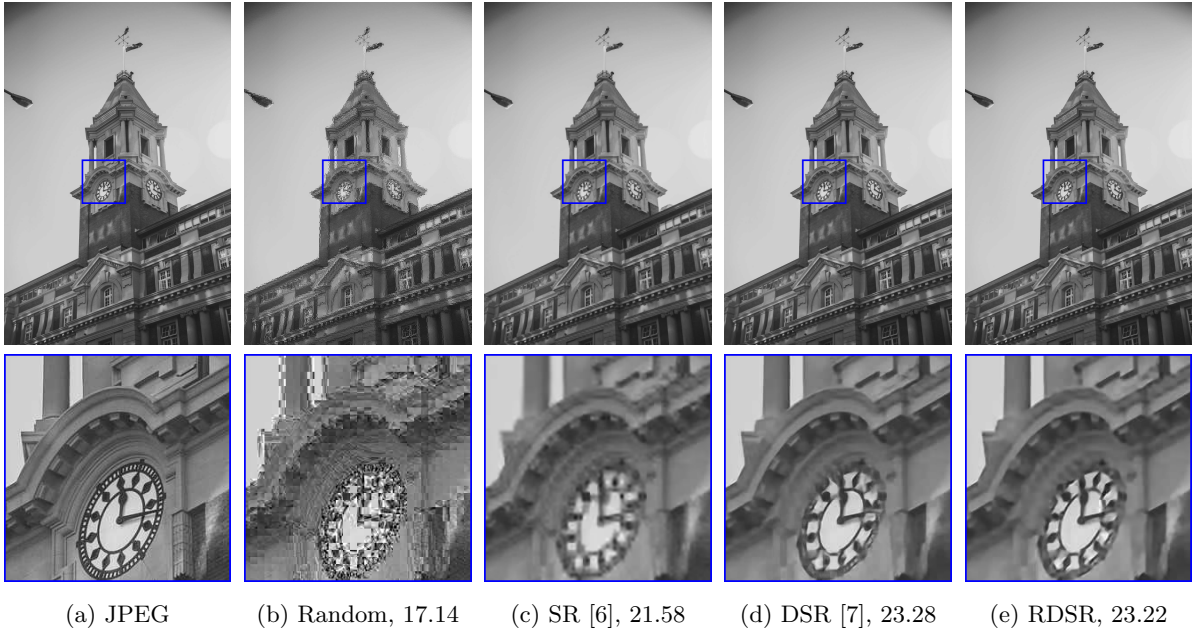


図 3: 復元画像と PSNR

かかる。また、SR アルゴリズムでは、高精度な復元符号情報を得るために少なくとも $N = 200$ 回の反復が必要であり、故に実行時間も長い。文献 [3, 5] の手法は非常に高速だが、前節で述べたように十分な圧縮率を得られていない。一方、DSR 及び RDSR はたった一回の推論のみで復元符号情報が得られるため、現実的な計算時間である。なお、DSR と RDSR のパラメータ数は各々 80,660 と 4,033 であり、パラメータ数を 1/20 に削減できている。以上の評価を纏めると、DSR の復元符号情報の精度を維持しつつ、パラメータ数を大幅に削減できている。本研究の目標が達成されたと言える。

4.4 復元品質の評価

SR [6], DSR [7], 及び RDSR アルゴリズムの復元画像 \hat{x} を図 3 に示す。図中の数値は JPEG 符号化/復号画像 (a) に対する PSNR である。(b) は符号をランダムに割り当てた画像であり、復元符号情報の誤差が画像領域でどれほどの影響があるかを目視するために併載した。(c), (d), 及び (e) は各手法の復元画像である。この結果から、DSR 及び RDSR は高精度に JPEG 符号化/復号画像を復元できていることが確認できる。PSNR の向上は復元符号情報の精度向上を意味し、これは復元誤差 $e_{s,t,\omega_u,\omega_v}$ のエントロピーの低減につながるため、図 2 の結果は妥当であると言える。なお、RDSR は DSR の 1/20 のパラメータ数であるにもかかわらず、ほぼ同等の復元品質を達成しており、再帰的アーキテクチャの有

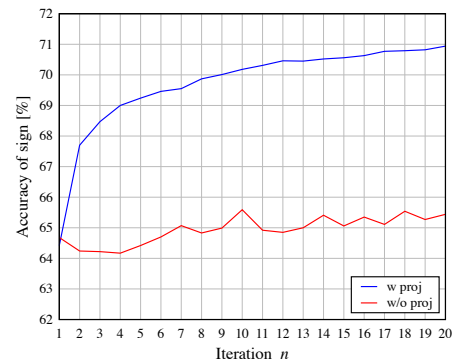


図 4: 凸射影と正解率の関係

効性が視覚的にも確認できる。

4.5 凸射影の評価

RDSR は複雑なテクスチャを含む高周波成分を復元できており、これはネットワーク内の凸射影による寄与が大きいと考えられる。ここでは、凸射影の有効性を検証するために、式 (12) の $\text{proj}_{|y|}$ を恒等写像に置き換えて学習を行なった。テスト時の QF を 50 とした場合の実験結果を図 4 に示す。横軸は反復回数 n 、縦軸は原符号情報 $\text{sgn}(y_{s,t,\omega_u,\omega_v})$ に対する復元ビット $\text{sgn}(\hat{y}_{s,t,\omega_u,\omega_v})$ の正解率を表す。凸射影を恒等写像に置き換えて復元した場合、 $n = 1$ を除いて大幅に正解率が下がっていることから、凸射影は極めて有効であると結論づけられる。

4.6 学習条件の調査

SR 問題を解いて得られる復元画像は JPEG 符号化/復号画像であるため、学習時に用いる画像データセット、

表 4: QF と学習用データセット

Dataset		Train QF		
D_{50}		25	50	75
Test QF	25	-	✓	-
	50	-	✓	-
	75	-	✓	-

Dataset		Train QF		
D_{org}		25	50	75
Test QF	25	✓	✓	✓
	50	✓	✓	✓
	75	✓	✓	✓

及び凸射影で用いる DCT 係数の絶対値も同じ QF で符号化/復号したものであるべきと考える。本節ではこの仮説の真偽の検証を目的として、様々な設定で学習を行なう。4.1 節の D_{org} に対して、QF = 50 の JPEG を適用して得られた符号化/復号画像群を D_{50} と記述する。また、学習時とテスト時に用いる DCT 係数 $|y_{s,t,\omega_u,\omega_v}|$ は QF = 25, 50, 75 のいずれかで量子化されたものとする。表 4 に本実験の設定をまとめる。チェックマークを記述した箇所の復元符号情報の正解率を計算した。

図 5 に結果を示す。横軸はテスト時の QF であり、縦軸は復元符号情報の正解率である。図中の線の色は表 4 のものに対応している。結果を端的に述べると、学習時に用いる画像データセットは D_{org} 、凸射影で用いる DCT 係数の絶対値は QF = 50 のものを用いれば、復元符号情報の正解率が最大化される。従って、QF ごとにネットワークを最適化する必要はないことがわかる。言い換えれば、QF = 50 で学習した単一のネットワークがあれば、あらゆる QF に対して最も効率よく符号情報を圧縮できることを示している。この結果より、本節の冒頭で述べた仮説は真である可能性が低いと言える。故に、4.1 節のような実験条件を採用した。

5 まとめ

本稿では、直交変換係数の絶対値のみから符号情報を復元するネットワークの軽量化手法を提案した。実験結果より、文献 [7] の符号情報の復元精度を維持しつつパラメータ数を大幅に削減することが達成された。今後の展望として、大規模データセットでの学習や、動画画像圧縮への応用が挙げられる。

参考文献

- [1] N. N. Ponomarenko, A. V. Bazhyna, and K. O. Egiazarian, "Prediction of signs of DCT coefficients in block-based lossy image compression," *Proc. SPIE 6497, Image Processing: Algorithms and Systems V*, 2007.
- [2] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas, "Transform coefficient coding in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1765–1777, 2012.
- [3] C. Tu and T. D. Tran, "Context-based entropy coding of block transform coefficients for image compression," *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1271–1283, 2002.
- [4] A. Nakagawa, J. Koyama, K. Kazui, and J. Katto, "Sign information predictive coding of transform coefficients in video compression," *IEICE Trans. Inf. & Syst.*, vol. J100-D, no. 9, pp. 819–830, 2017.
- [5] O. Miroshnichenko, M. Ponomarenko, V. Lukin, and K. Egiazarian, "Compression of signs of DCT coefficients for additional lossless compression of jpeg images," *Proc. IS&T Int'l Symp. on Electronic Imaging: Image Processing: Algorithms and Systems XVI*, pp. 385-1 – 385-5, 2018.
- [6] C. Tsutake, K. Takahashi, and T. Fujii, "An efficient compression method for sign information of DCT coefficients via sign retrieval," *IEEE Int. Conf. Image Process. (ICIP)*, 2021.
- [7] K. Suzuki, C. Tsutake, K. Takahashi, and T. Fujii, "Deep sign retrieval based on end-to-end learning," *PCSJ/IMPS 2021*, 2021.
- [8] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [9] T. Goldstein and C. Studer, "PhaseMax: convex phase retrieval via basis pursuit," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2675–2689, 2018.
- [10] E. Pauwels, A. Beck, Y. Eldar, and S. Sabach, "On Fienup methods for regularized phase retrieval," *arXiv:1702.08339*, 2017.
- [11] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

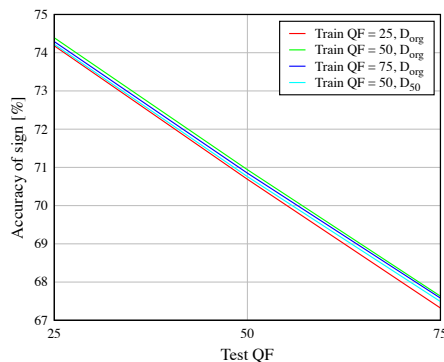


図 5: 学習条件と正解率の関係