

# ニューラルネットワークを用いた単一画像超解像の ウェーブレット変換による高速化

グエン マントウアン<sup>1</sup> 岩井 啓輔<sup>2</sup> 松原 隆<sup>2</sup> 黒川 恭一<sup>2</sup>

**概要**：本稿では、ニューラルネットワークを用いた単一画像超解像(SISR, Single Image Super-Resolution)にウェーブレット変換(WT, Wavelet Transform)を加えて高速化する手法を提案する。SISRは、一枚の低解像度画像から高解像度画像を作成する。機械学習の技術発展と共に、畳み込みニューラルネットワークを用いたEDSR(Enhanced Deep network SR)等、高精度のSISRが提案されている。しかし、これらのネットワークは非常に階層が深く処理に時間がかかる。WTは、画像を1つの低周波成分と3つの高周波成分に分解し、WIT(Wavelet Inverse Transform)はその4成分を結合して元の画像に復元する。4成分はサイズが元の画像サイズの1/4であり、スパース性を持つ。WTにより生成された成分をニューラルネットワークに学習させる事で、画像生成時間や学習時間を大幅に下げることが出来る。本稿では、WTを行う層(WT Layer)とWITを行う層(WIT Layer)を既存のEDSRと組み合わせて、より高速なEDWSR(Enhanced Deep Wavelet network SR)を提案する。その結果、EDWSRはEDSRとほぼ同等の精度を保ちながら、約3.6倍の高速化を実現した。学習スピードも約2倍に高速化した。

**キーワード**：単一画像超解像，ニューラルネットワーク，ウェーブレット変換，ウェーブレット変換層

## 1. はじめに

近年、ニューラルネットワークを用いた単一画像超解像(SISR, Single Image Super-Resolution)の処理が研究されている。2015年、3層の畳み込みニューラルネットワーク(CNN, Convolutional Neural Network)を用いたSISRであるSRCNN[1]が提案された。その後、さらに高い精度を持つEDSR(Enhanced Deep network SR)[2]等が提案されている。これらの手法は高精度であるが、ネットワークの階層が非常に深く、低速である。Kumarらは2017年にCNNを用いたSRCNNにウェーブレット変換(WT, Wavelet Transform)[3]を適用したCNNWSR(CNN Wavelet SR)[4]を提案した。それ以降、これを改良する提案[5,6]がされている。特に2020年、GadipudiらはCNNにWTを組み合わせたRWSR(Residual Wavelet SR)[7]を提案し、SRCNNより高精度化と高速化を実現した。

本稿では、SRCNNより高精度のSISRのネットワークにWTを組み合わせる事で、より高精度かつより高速になる手法を提案する。RWSRと同様にWTとウェーブレット逆変換(WIT, Wavelet Inverse Transform)を行うWT層とWIT層を作成し、他のSISRネットワークと組み合わせる。今回はこれを既存のSISRネットワークであるEDSRに組み合わせたEDWSR(Enhanced Deep Wavelet network SR)を提案する。

## 2. 関連事項

### 2.1 SISR

SISRとは、図1の様に一枚の画像を入力として受け取り、対応する高画質な画像を出力として生成する技術である。一枚の低画質な画像に対応する高画質な画像は複数存在するため、SISRは解が定まらない不良設定問題である。SISRは人工衛星による地球環境の観測画像や天体観測画

像に応用される[8]。そして、医療に不可欠なコンピュータ断層撮像技術、バーチャルリアリティ、そして画像通信による遠隔医療支援の各種画像の研究にも使用される[9]。ぼやけた画像に対して、ぼやけを改善し顔・物体の認識や解析を行う事を目的として使われている[10,11]。大量の画像データを作成する作業にもよく扱われる[12]。



図1 SISRのイメージ

SISRは観測画像では正しく表現されていない高周波成分を復元する手法であると考えられる。大きく2つの手法が用いられており、一つ目は信号処理的なバイキュービック法などの補間法である。二つ目は機械学習のニューラルネットワークに基づく手法である。

### 2.2 WT

ここでは、本研究で用いるWTについて述べる。ウェーブレット(Wavelet)は小さな(Let)波(Wave)という意味である。WTは20世紀初頭から発展したものであり、基本的に、小さい波を拡大縮小、平行移動して足し合わせる事で、与えられた入力の波形を表現する手法である。WTには連続WT(CWT, Continuous WT)と離散WT(DWT, Discrete WT)がある。本研究ではDWTを利用している。

DWTは1989年にMallatによって提案された。DWTは画像等の元信号を高周波成分と低周波成分に分解する。分解された低周波成分をまた高周波成分と低周波成分に分解するという処理を繰り返す事が出来る。一般的には、DWTはJPEG 2000のデータ圧縮に使われている。DWTは一度変換した情報を加工して逆変換する事で、ノイズの除去な

1 防衛大学校理工学研究所

2 防衛大学校情報工学科

どに応用出来る．また，画像の方向性エッジを特徴として計算する事も出来る．

### 2.3 2次元 DWT

本研究で用いている 2 次元画像の DWT について説明する．2 次元の離散データとして与えられる縦横  $M \times N$  の画像を  $I(x, y)$  と表す．この画像  $I(x, y)$  をレベル 0 のスケーリング係数  $s_{x,y}^{(0)}$  と見なす．まず，図 2 に示す様に水平方向に WT を行い，水平方向のスケーリング係数  $s_{x,y}^{(1,x)}$ ，ウェーブレット展開係数  $w_{x,y}^{(1,x)}$  を式(1)に示す．

$$\begin{aligned} s_{x,y}^{(1,x)} &= \sum_{i=0}^{M-1} \overline{h_{\phi}(i-2x)} s_{i,y}^{(0)} \\ w_{x,y}^{(1,x)} &= \sum_{i=0}^{M-1} \overline{h_{\psi}(i-2x)} s_{i,y}^{(0)} \end{aligned} \quad (1)$$

ここでの  $\overline{h_{\phi}}$ ,  $\overline{h_{\psi}}$  はスケーリング・ベクトル  $h_{\phi}$ , ウェーブレット・ベクトル  $h_{\psi}$  の複素共役である．本研究で使用するハール・ウェーブレットの  $h_{\phi}$ ,  $h_{\psi}$  は式(2)の様に表される．

$$h_{\phi}(a) = \begin{cases} \frac{1}{\sqrt{2}}, & a = 0, 1 \\ 0, & \text{その他} \end{cases}$$

$$h_{\psi}(a) = \begin{cases} \frac{1}{\sqrt{2}}, & a = 0 \\ -\frac{1}{\sqrt{2}}, & a = 1 \\ 0, & \text{その他} \end{cases} \quad (2)$$

次に，その  $s_{x,y}^{(1,x)}$  係数と  $w_{x,y}^{(1,x)}$  の係数に対して，それぞれ垂直方向に WT を行う．これを図 2 及び式(3)に示す．

$$s_{x,y}^{(1)} = \sum_{j=0}^{N-1} \overline{h_{\phi}(j-2y)} s_{x,j}^{(1,x)}$$

$$w_{x,y}^{(1,h)} = \sum_{j=0}^{N-1} \overline{h_{\psi}(j-2y)} s_{x,j}^{(1,x)}$$

$$w_{x,y}^{(1,v)} = \sum_{j=0}^{N-1} \overline{h_{\phi}(j-2y)} w_{x,j}^{(1,x)}$$

$$w_{x,y}^{(1,d)} = \sum_{j=0}^{N-1} \overline{h_{\psi}(j-2y)} w_{x,j}^{(1,x)} \quad (3)$$

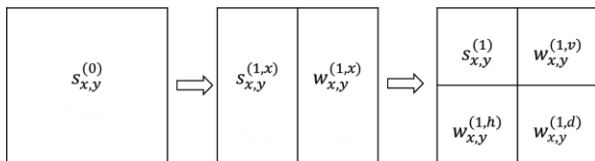


図 2 レベル 1 の DWT の処理手順

ここで， $s_{x,y}^{(1)}$  はレベル 1 のスケーリング係数， $w_{x,y}^{(1,h)}$  はレベル 1 の水平方向のウェーブレット係数， $w_{x,y}^{(1,v)}$  はレベル 1 の垂直方向のウェーブレット係数， $w_{x,y}^{(1,d)}$  はレベル 1 の対角方向のウェーブレット係数である．式(1)と式(3)をまとめて表現すると， $s_{x,y}^{(1)}$ ,  $w_{x,y}^{(1,h)}$ ,  $w_{x,y}^{(1,v)}$ ,  $w_{x,y}^{(1,d)}$  には低周波成分  $A(x, y)$ ，水平方向の高周波成分  $H(x, y)$ ，垂直方向の高周波成分  $V(x, y)$ ，対角方向の高周波成分  $D(x, y)$  が現れる．

$$\begin{aligned} A(x, y) &= \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} \overline{h_{\phi}(j-2y)} \overline{h_{\phi}(i-2x)} I(i, j) \\ H(x, y) &= \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} \overline{h_{\psi}(j-2y)} \overline{h_{\phi}(i-2x)} I(i, j) \\ V(x, y) &= \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} \overline{h_{\phi}(j-2y)} \overline{h_{\psi}(i-2x)} I(i, j) \\ D(x, y) &= \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} \overline{h_{\psi}(j-2y)} \overline{h_{\psi}(i-2x)} I(i, j) \end{aligned} \quad (4)$$

ここで，垂直方向の高周波成分のみを残し，その他の係数を 0 と置いて，式(5)の様な WIT によって，レベル 1 の垂直方向エッジが抽出出来る．また，低周波成分を 0 と置き WIT を行えば，レベル 1 のエッジが抽出出来る．画像に対してこの様な処理ができる事は WT の特徴である．

$$\begin{aligned} I(x, y) &= h_{\phi}(y-2j) h_{\phi}(x-2i) A(i, j) \\ &+ h_{\psi}(y-2j) h_{\phi}(x-2i) H(i, j) \\ &+ h_{\phi}(y-2j) h_{\psi}(x-2i) V(i, j) \\ &+ h_{\psi}(y-2j) h_{\psi}(x-2i) D(i, j) \end{aligned} \quad (5)$$

$s_{x,y}^{(1)}$  をさらに 4 つの成分に分解する事をレベル 2 の DWT と呼ぶ．以上の処理を繰り返す事が出来るため，DWT は多重解像度解析と呼ばれる．

本稿では，これ以降用いているレベル 1 の 2 次元ハール DWT を WT と省略する．WT により，図 3 に示す様に，画像は左上のぼやけた低周波成分 A，左下の水平方向の高周波成分 H，右上の垂直方向の高周波成分 V と右下の対角の高周波成分 D の 4 つの成分に分解される．各成分の縦横は元の画像の縦横の半分となる．



図 3 元画像とそのレベル 1 の DWT による 4 つの成分

## 3. 先行研究

SISR にはいくつかの実験法があるが，本研究では，高画質な正解画像 HRI(High-Resolution Image)を用意し，それを縮小した画像 LRI(Low-Resolution Image)を入力とし，そこから画像 SRI(Super-Resolution Image)を生成する手法を用いる．具体的には，HRI をバイキュービック法で LRI に縦横  $1/n$  に縮小し，この LRI から提案手法等を用いて，縦横  $n$  倍超解像した SRI を生成する．そして，それを HRI と比較し精度を測る．同様の手法を用いる関連研究について次に述べる．

### 3.1 SRCNN

SRCNN では，LRI からバイキュービック法で拡大した LRI' を用意し，LRI' を RGB チャンネルから YCbCr チャンネルに変換する．LRI' の輝度  $Y_{LRI'}$  チャンネルだけをニューラルネットワークの入力とする．図 4 の様に SRI の輝度  $Y_{SRI}$  チャンネルを出力する．最終的に LRI' の Cb, Cr と生成した  $Y_{SRI}$  から変換し RGB チャンネルの SRI を生成する．



図 4 SRCNN の  $Y_{SRI}$  生成

ネットワークは図 4 に示す様に 3 つの畳み込み層(Conv 層)から構成される。

学習にも  $Y_{LSI'}$  と  $HRI$  の輝度  $Y_{LSI'}$  と  $Y_{HRI}$  を用いる。この手法は最もシンプルで、精度がバイキュービック等より優れている。

### 3.2 CNNWSR

CNNWSR は SRCNN と同様に  $Y_{LRI'}$  を使用する。そして WT より  $Y_{LRI'}$  を 4 つの成分(A,H,V,D)に変換する。低周波の A 成分だけをニューラルネットワークの入力とし、図 5 の様に新しい 3 つの高周波成分(H,V,D)を予測する。低周波の 1 成分と予測した高周波の 3 成分を用いて WIT より  $Y_{SRI}$  を生成する。

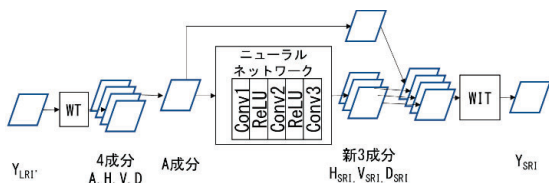


図 5 CNNWSR の手順

ネットワークは SRCNN と同様に 3 つの Conv 層から構成される。ただし、出力が 3 チャンネルであるため、Conv3 層は 1 フィルタから 3 フィルタになっている。

CNNWSR の学習では WT した  $Y_{LSI'}$  の A 成分、 $Y_{HRI}$  の 3 つの成分(H,V,D)を用いる。この手法は SRCNN より高速、高精度な SISR である。

### 3.3 RWSR

RWSR は CNNWSR と同様に  $Y_{LRI'}$  を使用し、これを 4 つの成分(A,H,V,D)に WT する。高周波の情報を失わない目的で、図 6 に示す様に  $Y_{LSI'}$  の WT で生成された 4 成分を全て使用し、ニューラルネットワークの入力とする。SRI の生成では、ニューラルネットワークで予測する 4 差分と入力の 4 成分の和を取り、 $Y_{SRI}$  の 4 成分を生成する。WIT より、それらを  $Y_{SRI}$  に変換する。

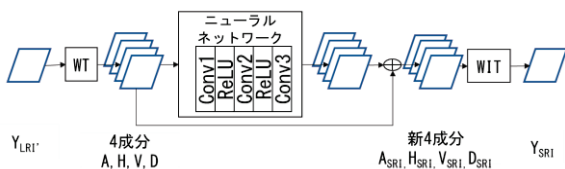


図 6 RWSR の手順

ネットワークは CNNWSR と同様に 3 つの Conv 層で構成される。ただし、4 チャンネルの入力と 4 チャンネルの出力を持つ。

学習では、 $Y_{LSI'}$  の WT による 4 成分、 $Y_{HRI}$  の WT による高周波の 4 成分との残差を用いる残差学習を利用する。RWSR は高速な手法であり、CNNWSR より高精度化する事が出来る。

## 4. 提案手法

本手法では、WT を用いた SISR を一般化し、より精度の高い既存 CNN と WT を組み合わせて、高速化と高精度化を目指す。

### 4.1 WT 層と WIT 層を組み合わせる手法

本手法は図 7 の様に、既存の CNN の前後に WT 層(WT Layer)と WIT 層(WIT Layer)を組み合わせ、新しいニューラルネットワークを構築する。WT 層と WIT 層はそれぞれ WT と WIT の処理を行う。

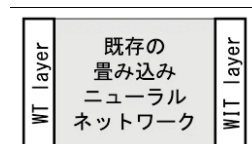


図 7 WT を用いる SISR のニューラルネットワーク

ここで、既存の CNN ではネットワークへの入出力が輝度 Y チャンネルではなく RGB チャンネルの場合もある。また、入力画像はバイキュービック法で拡大した  $LRI'$  ではなく拡大しない  $LRI$  の場合もある。本手法では入出力共に Y, RGB どちらも対応可能である。また、学習時の損失関数の対象は Y, または RGB のピクセルの画像画素値のどちらかである。

WT 層及び WIT 層の仕組みを次に示す。

WT 層は Y, RGB どちらのタイプの入力画像でも、そのまま入力する事が出来る。入力が Y の場合、WT により 4 成分(A,H,V,D)を生成し、この 4 成分を 1 つのテンソルにする。入力が RGB の場合、図 8 の様に、R,G,B それぞれのチャンネルに対して WT により 4 成分(A,H,V,D)を生成し、合計 12 成分を 1 つのテンソルにして次の Conv 層に送る。

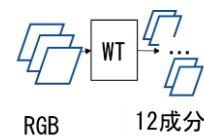


図 8 RGB チャンネル入力時の WT 層の仕組み

WIT 層では、入力のデータ形式に従って、Y または RGB を出力する。出力が Y の場合、WIT により以前の Conv 層から来る 4 成分のデータを持つ 4 チャンネルを Y チャンネルに変換する。出力が RGB の場合、図 9 の様に 12 成分のデータを持つ 12 チャンネルを 4 チャンネル毎に R,G,B チャンネルに WIT する。

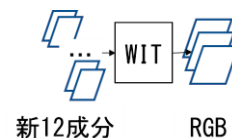


図 9 RGB チャンネル入力時の WIT 層の仕組み

本手法では、WT 層によりテンソルの各チャンネルは縦横のサイズが 1/2 倍になる。それに伴い、新しいネットワークの隠れ層は畳み込み処理により、全てのチャンネルに

において縦横のサイズが 1/2 倍になる。つまり、隠れ層のテンソル・サイズは 1/4 倍になる。それに従って、畳み込み演算や活性化関数による演算などの隠れ層の全体の演算量は 1/4 倍となる。WT 層と WIT 層の処理が増える事になるが、これは削減した演算量より非常に小さい。そのため、ネットワークの演算総量は WT を使用しないものより小さくなり、高速化される。

WT 層による 12 チャンネルのテンソル・データはスパース性を持ち、これを入力する Conv 層は自然にスパース化され、学習し易く速くなる事が期待される。

また、WT により LRI から特徴やエッジの貴重な構造情報がネットワークへ 12 チャンネルで補足的に提供される。その結果、学習により高精度な SRI の生成が出来ると考えられる。

#### 4.2 EDWSR

既存の SISR のニューラルネットワークは多数存在するが、本稿では、よく知られている EDSR を使用する。EDSR はネットワークが深層で、SRCNN よりはるかに精度が高く優れている手法である。ここでは、EDSR に提案手法を適用する EDWSR を提案する。

図 10 の全体像に示す様に、EDSR は RGB チャンネルの LRI を用いて RGB チャンネルの SRI を生成する。

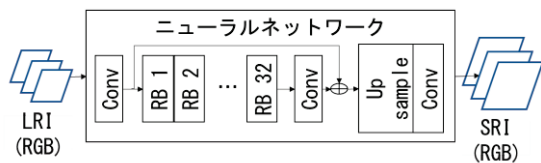


図 10 EDSR の SRI 生成

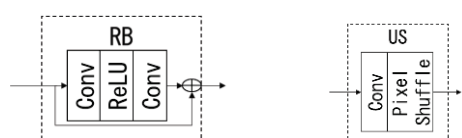


図 11 Residual Block(RB)と Up Sample(US)の構造

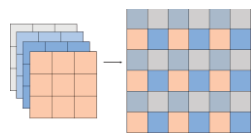


図 12 Pixel Shuffle の仕組み

EDSR のネットワークは図 10 の様に多数の Conv 層から構成される。図 10 中の RB(Residual Block)は図 11 に示す様に 2 つの Conv 層で構成される。この RB を 32 回繰り返す。ネットワークの最後に Up sample(US)でサイズを調整する。US は図 11 の様に、Conv 層と Pixel Shuffle を利用する。Pixel Shuffle はテンソルの再配置を行って、特徴マップを拡大する。図 12 に倍率を 2 とした場合の例を示す。2 次元情報なので、特徴マップの縦横をそれぞれ 2 倍拡大したい場合、US の Conv 層で畳み込みによって特徴マップのチャンネル数を 4 倍増加させる。図中の色に対応する様に、順番

に 4 チャンネル毎の特徴マップを 2x2 のグリッド内に再配置する。その結果、US は 3x3 のチャンネルから 6x6 のチャンネルに拡大する事が出来る。特徴マップの縦横をそれぞれ 3 倍拡大したい場合、2 倍の操作と同様に、Conv 層で特徴マップを 9 倍増加させ、順番に 9 チャンネル毎の特徴マップを 3x3 のグリッド内に再配置する。特徴マップの縦横をそれぞれ 4 倍拡大したい場合は、2 倍の操作を 2 回繰り返す。最後の Conv 層は縦横 3x3 のフィルタが 3 個ある。それ以外の Conv 層では縦横 3x3 のフィルタから 256 個作られる。全ての Conv 層はゼロ・パディングを付けられる。

EDSR の学習は二重の残差学習で構成されており、RGB チャンネルの LSI, HRI を用いる。RB はローカル・スキップ・コネクションと呼ぶローカルの残差を学習する。この RB を繰り返し、更にグローバル・スキップ・コネクションと呼ぶ全体の残差学習を採用する。これらのコネクションは低周波の情報を迅速にスキップして、細かい高周波の情報を多く取得する事が出来る。

次に、提案手法の EDWSR について説明する。EDWSR は EDSR と同様に RGB チャンネルの LRI を用いて RGB チャンネルの SRI を生成する。EDWSR のネットワークは図 13 に示す様に EDSR のネットワークの先頭に WT 層を、最後に WIT 層を追加する。WIT 層の直前にある最後の Conv 層においては、元々 RGB の 3 個のフィルタだったものを、WIT 層の 12 成分の入力に対応するために 12 個のフィルタにする。EDSR から追加された部分を図 13 の赤枠内に示す。

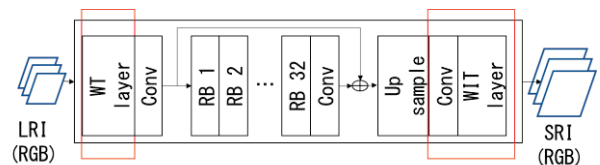


図 13 EDWSR のニューラルネットワーク

学習は LRI, HRI を用いて、式(6)の損失関数を最小値にする。損失関数は  $L_1$  ノルムを用いる

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N \|SRI^i - HRI^i\|_1 \quad (6)$$

ここで、 $N$ サイズのミニ・バッチ時、集合  $\{SRI^i, HRI^i\}_{i=1}^N$  は、LRI から予測された SRI と対応する正解 HRI の集合で、 $\theta$  は EDWSR のパラメータである。

#### 5. 実験

本実験では提案手法 EDWSR を実装した。EDSR も比較対象として実装した。各手法で、縦横の 2, 3, 4 倍拡大をそれぞれ学習し、学習時間を計測し、学習済みのニューラルネットワークを用いて画像生成する時間とその精度も計

測した。ここで、EDSR については公開されているコード<sup>1</sup>があり、それを使用して実装した。EDWSR もそれに基づいて、ニューラルネットワーク、WT 層、WIT 層のプログラムを作成し、実装した。

実験における学習では訓練データとして DIV2K[17]のデータ・セットを利用している。DIV2K には 2K の HRI とそれに対応する縦横 1/2, 1/3, 1/4 倍縮小の LRI が 800 セット含まれる。訓練の際、1 枚毎に入力 LRI から 48 x 48 サイズの部分領域(パッチ)とそれに対応する正解 HRI のパッチを取る。そのペア {LRI, HRI} に対して、それぞれ同じ鏡反転や 90°, 180°, 270°回転の画像変形処理をランダムに行い、16000 組の学習データを用意した。バッチ・サイズは 16 であり、1 エポックあたり 1000 バッチを 300 エポック訓練した。パラメータの総更新回数は  $3 \times 10^5$  回である。両手法共に損失関数 L1 と勾配法の最適化アルゴリズム ADAM を利用した。ADAM のパラメータ  $\beta_1$ ,  $\beta_2$ ,  $\epsilon$  はそれぞれ 0.9, 0.999,  $10^{-8}$  とした。学習率は  $10^{-4}$  を初期値とし、訓練の 200 エポック目からそれを半分にした。

画像生成時の精度評価については、PSNR(Peak Signal-to-Noise Ratio)を使用した。PSNR は、客観的な評価指標の一つであり、画像の圧縮等を行った時にその変換後の画像がどれ程劣化したかを評価する。PSNR は、0 から  $\infty$  までの範囲の値をとる。単位は dB である。これが高ければ高いほど超解像の精度が良い事を示す。本稿において、YCbCr チャンネルに変換した SRI と HRI それぞれの Y チャンネルを比較し PSNR で評価する。

評価データはよく使われている Set5[13], Set14[14], B100[15], Urban100[16]の 4 つのデータ・セットである。各データ・セットの評価は平均 PSNR, 総実行時間で行う。

### 5.1 実験環境

Python3.8, Pytorch1.8 を用いて実装した。学習には高速化のため、GeForce RTX 3090 を用いた。画像生成には Intel Core-i9 10900K を用いた。

### 5.2 結果

WT を用いない EDSR と WT を用いる EDWSR の実験結果を表 1, 2, 3 に示す。表 1 に 2, 3, 4 倍拡大する実験と、それぞれの学習時間を示す。表 2 には、Set5, Set14, B100, Urban100 をそれぞれ 2, 3, 4 倍拡大した各実験の精度 PSNR を示し、表 3 に各実験それぞれの CPU での画像生成の実行時間を示す。ここで、PSNR は各データ・セットの平均値であり、単位は dB である。実行時間は各データ・セット内の画像生成の総計である。単位は秒である。

表 1 EDWSR と EDSR の学習した時間の比較[hr.]

手法	EDSR	EDWSR	比較	
学習時間	x2	19.5	9.8	2.0 倍
	x3	22.6	10.4	2.2 倍
	x4	28.7	11.3	2.5 倍

表 2 EDWSR と EDSR の PSNR の比較[dB]

Data set	手法	EDSR	EDWSR	差分
Set5 [5 枚]	x2	38.15	38.18	+0.03
	x3	34.66	34.62	-0.04
	x4	32.47	32.36	-0.11
Set14 [14 枚]	x2	33.88	33.89	+0.01
	x3	30.50	30.49	-0.01
	x4	28.74	28.74	+0.00
B100 [100 枚]	x2	32.30	32.30	+0.00
	x3	29.19	29.18	-0.01
	x4	27.69	27.66	-0.03
Urban100 [100 枚]	x2	32.67	32.69	+0.02
	x3	28.55	28.47	-0.08
	x4	26.41	26.40	-0.01

表 3 EDWSR と EDSR の CPU での画像生成の実行時間の比較[sec]

Data set	手法	EDSR	EDWSR	比較
Set5 [5 枚]	x2	19.54	5.45	3.58 倍
	x3	9.12	2.47	3.33 倍
	x4	6.33	1.89	3.35 倍
Set14 [14 枚]	x2	109.52	28.38	3.86 倍
	x3	53.10	14.87	3.57 倍
	x4	35.28	9.87	3.57 倍
B100 [100 枚]	x2	532.50	139.77	3.81 倍
	x3	258.24	70.06	3.69 倍
	x4	175.64	48.89	3.59 倍
Urban100 [100 枚]	x2	2541.57	705.60	3.60 倍
	x3	1207.98	334.11	3.62 倍
	x4	814.69	224.30	3.63 倍

## 6. 考察

表 1 で、提案手法である EDWSR を既存手法 EDSR と比較すると、同じ  $3 \times 10^5$  回の更新訓練で、学習速度が 2 倍以上上昇した。特に、縦横 4 倍に拡大する実験では学習速度は 2.5 倍上がった。表 3 で、EDWSR の画像生成の実行時間を EDSR の実行時間と比べてみると、実行速度が 3.3 倍以上上昇した。最も画像が多い B100 と Urban100 を基準とすると、実行速度が 3.6 倍上がった。WT を組み合わせる事で、実行速度が向上した事が分かった。

表 2 の PSNR の差分を見ると、縦横 2 倍の拡大では、全データ・セットの精度が若干上昇した。3 倍の拡大では、精度は若干低下した。4 倍の拡大の場合、Set5 の精度が低下したが、他のデータ・セットは精度がほぼ同じである。平均をとると、EDWSR は EDSR と精度がほぼ同等であるといえる。これにより、高速化をしつつ、精度は EDSR と同等に維持された。しかし、精度については WT による画像構造のエッジ情報がまだ適切に扱われていないため、精度の改善にはならなかったと考えられ。本実験で用いたハール WT 以外の WT の使用の検討が必要である。

<sup>1</sup> <https://github.com/sanghyun-son/EDSR-PyTorch>

## 7. おわりに

ニューラルネットワークを用いた既存の SISR に対し、高精度化かつ高速化を目的として、WT の導入を一般化した新たな WT 層及び WIT 層を組み合わせる手法を提案した。WT を用いない SISR の EDSR に対し、WT 層及び WIT 層を加える EDWSR を提案し、実験した。実行結果から、EDWSR は EDSR と比べて高速である事を確認した。WT により、ネットワークの隠れ畳み込み層の特徴マップのサイズは 1/4 となり、更にデータがスパース性を持つ事で、学習速度は 2 倍になった。また、画像生成の速度も 3.6 倍となった。精度について、EDWSR は EDSR とほぼ同等であった。今後はハール以外の WT の検討や EDSR 以外のニューラルネットワークを用いた SISR へこの手法を行っていき、より高速かつより高精度な WT を用いた SISR の手法を提案していく。

## 参考文献

- [1] Dong, C., Loy, C.C., He, K., Tang, X., Image Super-Resolution Using Deep Convolutional Networks. <https://arxiv.org/pdf/1501.00092>.
- [2] Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., Enhanced Deep Residual Networks for Single Image Super-Resolution. <https://arxiv.org/pdf/1707.02921>.
- [3] 中野 宏毅, 山本 鎮男, 吉田 靖夫. ウェーブレット変換による信号処理と画像処理. 共立出版株式会社, 2000.
- [4] Kumar, N., Verma, R., Sethi, A., Convolutional neural networks for wavelet domain super resolution. *Pattern Recognition Letters*, 2017, vol. 90, p. 626-634.
- [5] Guo, T., Mousavi, H.S., Vu, T.H, Monga, V., Deep Wavelet Prediction for Image Super-Resolution. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, vol. 1, p. 1100-1109, DOI: 10.1109/CVPRW.2017.148.
- [6] Liu, J., Xie, Y., Song, H., Yuan, W., Ma, L., RESIDUAL ATTENTION NETWORK FOR WAVELET DOMAIN SUPER-RESOLUTION. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, DOI: 10.1109/ICASSP40776.2020.9053245.
- [7] Gadipudi, A., Deivalakshmi, S., Residual Wavelet Coefficients Prediction using deep Convolutional Neural Network for Single Image Super-Resolution. *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*. 2020, DOI: 10.1109/AISP48273.2020.9073328.
- [8] Yue, L., Shen, H., Li, J., Yuan, Q., Zhang, H., Zhang, L., Image super-resolution: The techniques, applications, and future. *Signal Processing*, 2016, vol. 128, p. 389-408. <https://doi.org/10.1016/j.sigpro.2016.05.002>.
- [9] Shi, W., Caballero, J., Ledig, C., Zhuang, X., Bai, W., Bhatia, K., de Marvao, A.M.S.M, Dawes, T., O'Regan, D., Rueckert, D., Cardiac Image Super-Resolution with Global Correspondence Using Multi-Atlas PatchMatch. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2013*, 2013, p. 9-16.
- [10] Wilman, W.W.Z., Pong, C.Y, Very Low Resolution Face Recognition Problem. *Transactions on Image Processing*, January 2012, vol. 21, p. 327-340, DOI: 10.1109/TIP.2011.2162423.
- [11] Sajjadi, M.S.M., Scholkopf, B., Hirsch, M., EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. <https://arxiv.org/pdf/1612.07919>.
- [12] Karras, T., Aila, T., Laine, S., Lehtinen, J., PROGRESSIVE

GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION. <https://arxiv.org/abs/1710.10196>.

- [13] Marco, B., Aline, R., Christine, G., Alberi, M.M., Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. *British Machine Vision Conference*, 2012, p. 135-144.
- [14] Roman, Z., Michael, E., Matan, P., On Single Image Scale-Up using Sparse-Representation. *The 7th international conference on Curves and Surfaces*, 2012, p. 711-730.
- [15] Timofte, R., Smet, V.D., Gool, L.V., A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution. *Lecture Notes in Computer Science*, 2015, vol. 9006, p. 111-126.
- [16] Huang, J.B., Singh, A., Ahuja, N., Single image super-resolution from transformed self-exemplars. *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, October 14, 2015, p. 5197-5206.
- [17] Agustsson, E., Timofte, R., NTIRE 2017 challenge on single image super-resolution: Datasets and study. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, DOI: 10.1109/CVPRW.2017.150.