

# コンピュータビジョン 最終課題

5123F053 清水 駿太

2023年8月10日

## 1 プロジェクト名

低解像度航空写真の高解像度化

## 2 プロジェクトの背景と目的

航空写真は、主に測量を目的とし、自然や都市の環境変化を広域的に観測するために活用されている。そして、画像からできるだけ詳細な視覚情報を抽出することが求められる。しかし、そのためには高解像度画像が必要となる。高解像度画像はデータサイズが大きく、大量の画像を撮影する必要のある航空写真ではストレージの圧迫が起こる。また、撮影の条件により、全ての写真が高解像度で取れるわけではないという問題がある。そのため、本プロジェクトでは、深層学習を用いた低解像度画像を高解像度化するモデルを構築することで、小さいデータサイズの画像からより多くの有用な視覚情報が得られるようになることを目指す。

## 3 背景知識

### 3.1 置み込み層での演算

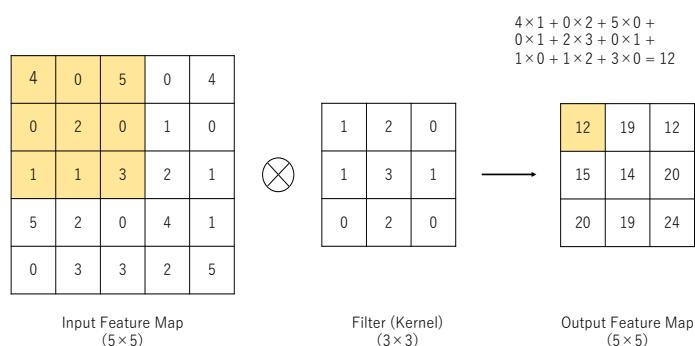


図 1. 2 次元特徴マップに対する置み込み演算

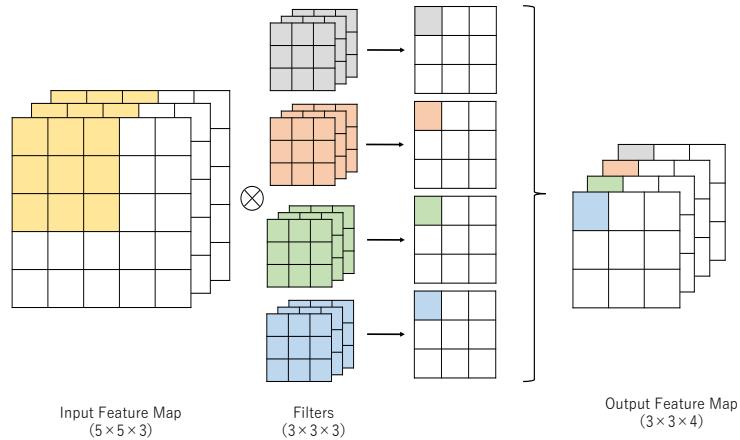


図 2. 3 次元特徴マップに対する畳み込み演算

畳み込み層では画像の特徴抽出を目的とした畳み込み演算が行われる。畳み込み層を基本構成要素とするネットワークを畳み込みニューラルネットワークという。畳み込み層では、 $3 \times 3$ などのフィルタ（カーネル）を用いて畳み込み演算を行う。入力特徴マップに対してフィルタの適用位置をスライドさせ、スライドさせる度に入力とフィルタが重なっている部分の要素同士の積和を取る。そして、それぞれの計算結果をフィルタの位置に基づいた 2 次元配列の要素とし、特徴マップを得る。畳み込みニューラルネットワークでは、フィルタの各要素がネットワークの重みとなっており、それを学習によって更新する。また、フィルタのスライド幅をストライドという。通常、畳み込み演算の出力特徴マップは入力のサイズよりも小さくなる。そして、ネットワークの層が深くなると特徴マップのサイズは更に小さくなり、最終的には演算を行えなくなってしまう。このようなデータサイズの縮小を防ぐために、パディングという操作を行う。パディングは、入力特徴マップの周囲に 0 などの固定値を追加することで入力特徴マップのサイズの拡張を行い、畳み込み演算時のデータサイズの縮小を防ぐことができる。図 1 に 2 次元の特徴マップを入力とした畳み込み演算の例（ストライド 1、パディング無し）を示す。

畳み込み層では多くの場合、縦 H、横 W、チャンネル C の 3 次元の特徴マップを扱う。入力が RGB 画像であればチャンネル数は 3 となる。3 次元の特徴マップを入力とする畳み込み演算では、フィルタも 3 次元形状になっており、フィルタのチャンネル数は入力特徴マップのチャンネル数と等しくなっている。畳み込み演算はフィルタを 2 次元平面方向にスライドさせ、特徴マップとフィルタの要素同士の積和を取ることで特徴マップを生成する。ここで、1 つのフィルタのみを用いた畳み込み演算で出力される特徴マップは 2 次元となるが、一般的な畳み込み層では特徴マップに対して複数のフィルタを適用する。よって、それぞれのフィルタについて畳み込み演算を行い、特徴マップが生成するため、最終的な畳み込み層の出力は、フィルタの個数分のチャンネル数を持つ 3 次元の特徴マップとなる。このように、畳み込み層では、フィルタ数を増やすことで、チャンネル方向に深い特徴マップを生成できる。図 2 に 3 次元の特徴マップを入力としたときの畳み込み演算の概要（ストライド 1、パディング無し）を示す。

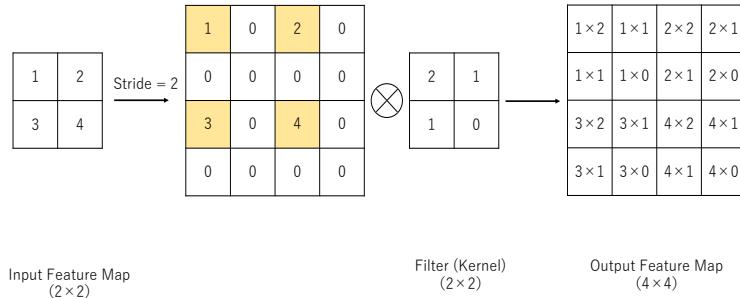


図 3. 転置畳み込み演算

### 3.2 転置畳み込み層での演算

転置畳み込み層では、入力された特徴マップに対してアンプーリング (unpooling) という操作を指定したストライドの大きさに従って行う。一般的にストライドは 2 を用いられる。アンプーリングでは、学習パラメータを用いずに特徴マップの拡大操作を行う。アンプーリングには、”bed of nails”, ”nearest neighbor”, ”max unpooling” の 3 つの手法が主に用いられる。図 3 は”bed of nails”を用いて入力特徴マップをアンプーリングしている。このときに、ストライドは特徴マップの拡大率に相当する。そして、拡大された特徴マップに対してフィルタを適用し、演算を行う。この演算では、フィルタを適応した領域内に存在する入力特徴マップの値（ゼロパディングにより埋め込まれていない値）とフィルタの各要素の積を取ることでゼロパディングした領域の値を補間する。これにより、拡大された特徴マップが得られる。

### 3.3 プーリング

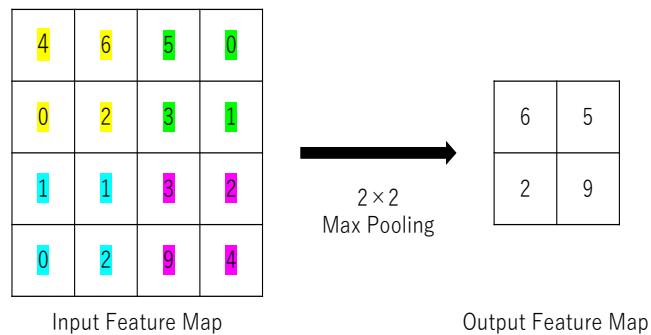


図 4. マックスプーリングの例

複数の畳み込み層からなるネットワークでは、プーリング層を追加し、プーリングという演算処理によって特徴マップの空間的サイズを小さくすることが多い。プーリングは、特定サイズの領域 ( $2 \times 2$ 、 $3 \times 3$  など) を  $1 \times 1$  の領域に縮小させる操作を行う。このときに、入力と出力のチャンネル数は維持される。

プーリングは大きく 2 種類に分けられる。1 つ目は、局所領域内における値のサンプリングを行う、マックスプーリングとアベレージプーリングである。マックスプーリングは、特定サイズの領域の最大値をとることでサイズ縮小を行い、アベレージプーリングでは、特定サイズの領域の平均値をとる。図 4 は  $4 \times 4$  のサイズの特徴マップに対して、 $2 \times 2$  の領域を指定したマックスプーリングを行うことにより、特徴マップのサイズを  $2 \times 2$  に縮小させる処理の例である。2 つ目は、局所領域を用いずに特徴マップ全体に対してを処理を行うグローバルプーリングである。グローバルプーリングは、特徴マップ全体の最大値をとるグローバルマックスプーリングと平均値をとるグローバルアベレージプーリングがある。

### 3.4 エンコーダ・デコーダ型ネットワーク

エンコーダ・デコーダ型ネットワークとは、エンコーダとデコーダという 2 つの構造を持つネットワークである。画像データを対象とする場合、エンコーダは畳み込み層などによる複数のダウンサンプリングレイヤから構成され、高次元の画像情報をダウンサンプリングし、特徴抽出を繰り返しながら低次元データに圧縮する。

デコーダは、転置畳み込み層などの複数（エンコーダで用いられた畳み込み層の数と等しいことが多い）のアップサンプリングレイヤから構成される。エンコーダの出力である低次元データをアップサンプリングレイヤで拡大を繰り返すことで高次元データに変換する。そして、目的の画像を出力として得る。

このネットワーク構造を持つ深層学習モデルは、異常検知、画像生成、画像変換など様々なタスクに用いられている。

### 3.5 U-net

U-net は 2015 年に O.Ronneberger ら [1] によって提案されたセマンティックセグメンテーションのモデルである。現在においても応用研究など多くの場面で活用されており、セマンティックセグメンテーションの代表的なモデルとして知られている。

U-net は、エンコーダ・デコーダ型のネットワーク構成になっている。エンコーダは、2 つの畳み込み層から成るインプットブロックと 1 つのプーリング層・2 つの畳み込み層から成るダウンサンプリングブロックを 4 つ持った構造となっている。デコーダは、1 つの転置畳み込み層・2 つの畳み込み層からなる 4 つのアップサンプリングブロックと 1 つの畳み込み層（出力層）から構成される。

U-net が優れたセグメンテーションの精度を出せる大きな要因として、エンコーダ・デコーダ間でスキップ接続の存在が挙げられる。U-net におけるスキップ接続は、デコーダのアップサンプリングブロックの転置畳み込み層からの出力に対して、同じ階層にあるエンコーダのダウンサンプリングブロックからの出力をチャンネル方向に連結させる。これにより、エンコーダ側の特徴マップが持つ特徴マップの空間的情報を補うことが可能となり、アップサンプリングブロックでの処理によって低次元データを高次元データに変換する際の精度を高めることができる。

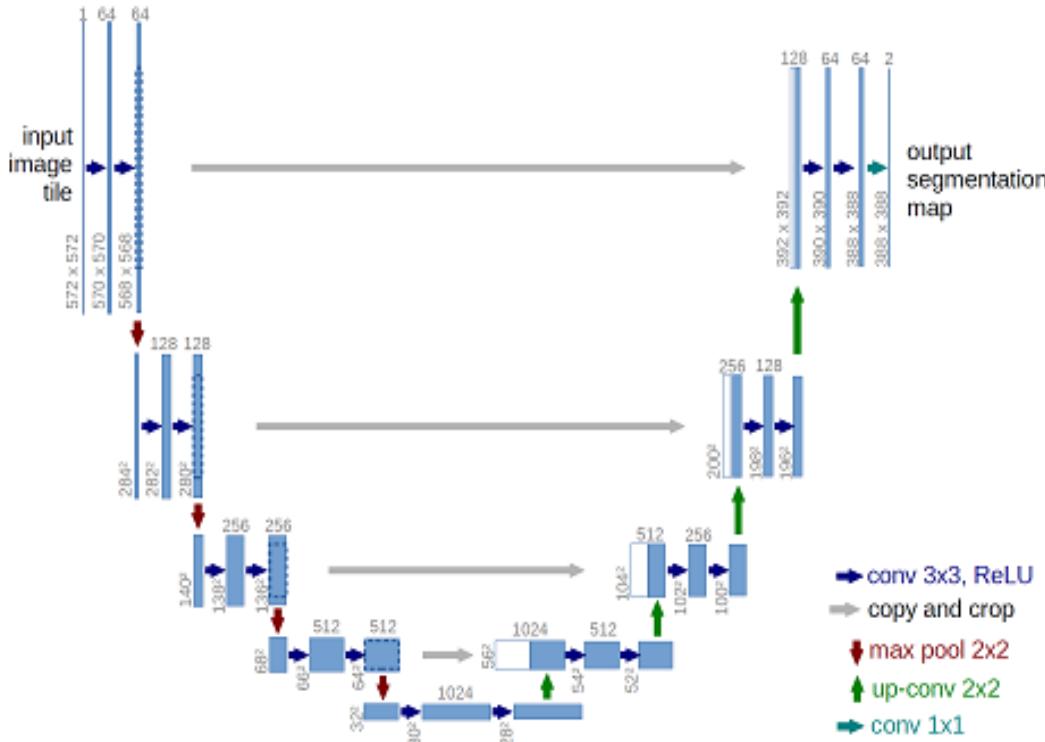


図 5. U-net

## 4 提案モデル

本プロジェクトでは、深層学習を用いた低解像度航空写真の高解像度化を実現するために、U-net ベースのエンコーダ・デコーダ型ネットワークを構築する。図 6 にネットワークの構造を示す。U-net が持つネットワーク構造を活かし、セマンティックセグメンテーションのみならず、画像変換やオートエンコーダとしての異常検知など様々なタスクへの応用が行われている。超解像も画像変換の一種であり、出力の高解像度画像では入力画像の画像特徴やその位置は維持される必要がある。そのため、U-net のネットワーク構造を基にモデルを構築することが効果的であると考えた。

提案モデルのエンコーダの構造は U-net と同じものを用いる。デコーダは 1 つの転置畳み込み層と 2 つの畳み込み層から成る 4 つのアップサンプリングブロックと 2 つの畳み込み層から成るアウトプットブロックモデルによって構成される。本モデルでは、畳み込み層や転置畳み込み層での出力特徴マップに対してバッチ正規化と活性化関数 ReLU による処理を行なっている。しかし、アウトプットブロックについては、1 つ目の畳み込み層からの出力に対してバッチ正規化をおこなった後、活性化関数 tanh を適用する。また、2 つ目の畳み込み層に対しては U-net のデコーダの最終層に倣って、畳み込み層の出力をそのままモデルの出力とする。

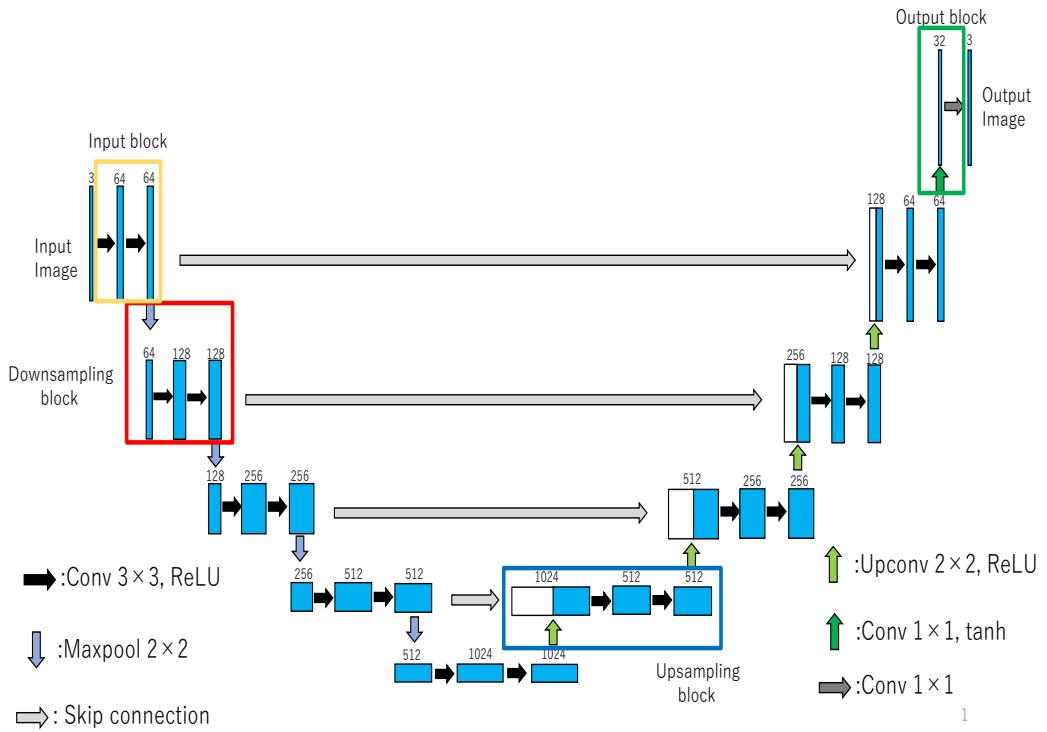


図 6. 提案モデル

## 5 実験

### 5.1 データセット

本プロジェクトでは、国土地理院 [2] が作成した GSI データセットを用いる。このデータセットは、国土地理院が行った「AI を活用した地物自動抽出に関する研究」の成果として作成されたものであり、航空画像の中から「道路」「建物」「水田」など計 18 クラスの地物についてのセグメンテーションを実施するためのものである。このデータセットは、18 クラスごとに元画像と正解のセグメンテーション画像が用意されている。本プロジェクトでは、18 クラスのうち「道路」「普通建物」「駐車場」「墓地」のデータセットに収録されていた元画像のみを合計 6800 枚を使用する。

元画像は  $572 \times 572$  のサイズになっている。全ての画像に対して、画像サイズを  $1/4(143 \times 143)$  に縮小した後に、 $572 \times 572$  に再拡大することで解像度が  $1/4$  になった画像を得る。そして、解像度が  $1/4$  になった画像を入力とし、元の解像度の画像を正解の高解像度画像とする画像ペアのデータセットを作成する。図 7 にデータセットに含まれる画像の一例を示す。

### 5.2 実装環境

本プロジェクトで行なったモデルの学習や推論の実行環境について表 1 に示す。



低解像度画像（入力）

高解像度画像（正解）

図 7. データセットの画像例

表 1. 実装環境の詳細

CPU	Intel Core i9-7900X
RAM	64GB
GPU	NVIDIA GeForce GTX1080ti(11GB)
OS	Ubuntu 20.04.5 LTS
使用言語	Python 3.9.0

### 5.3 学習の設定

提案モデルの学習を行なったときの設定を表 2 に示す。データセットは学習用とテスト用で 9:1 に分割した。

表 2. 学習の設定事項

学習用データ数	6120
テスト用データ数	680
バッチサイズ	2
エポック数	50
ロス関数	平均二乗誤差 (MSE)
最適化手法	Adam
学習率	$1 \times 10^{-5}$
重み減衰	$1 \times 10^{-5}$

## 6 結果

### 6.1 評価指標

本プロジェクトで行なった実験結果について定量的評価と定性的評価を行う。定量的評価指標には PSNR(Peak Signal-to-Noise Ratio) を用いる。PSNR は画質の劣化度合いを測る指標であり、画質の主観的評価との相関関係が高い。PANR を算出するためには 2 枚の画像が必要になり、値が高い程高画質であると言える。今回の評価では、入力画像と正解画像のペアと出力画像と正解画像のペアを用いて算出する。この時に、入力画像を用いて算出された PSNR の値より、出力画像を用いた方が高ければ、提案モデルの有効性を示す根拠になる。PSNR は式 1 のように計算され、単位は dB (デシベル) である。ここで  $Max_I$  とは画像が取りうる画素値の最大値で、MSE は 2 つの画像の平均二乗誤差の値である。

$$PSNR = 10 \times \log_{10} \frac{Max_I^2}{MSE} \quad (1)$$

また、定量的評価には、画像データサイズも取り入れる。入力の低解像度画像よりも出力画像の方がデータサイズが大きくなつていれば、高解像度化されていることが言える。画像データサイズの単位は kB (キロバイト) である。定性的評価では、画像の見た目に関する主観的評価を行う。定性的評価として、図 8,9 に入力画像、出力画像、正解画像を示し、定量的評価指標についても記載する。

図 8, 9 より、PSNR や画像データサイズの両指標において入力画像より出力画像の方が値が大きくなっている。また、全てのテストデータに対して、両指標を算出し、平均値を算出した結果を表 3 に示す。表 3 より、両指標とも出力画像が入力画像を上回っていることがわかる。

定性的評価について、図 8, 9 の入力画像と出力画像の比較より、物体の輪郭が鮮明になっていることが分かる。正解画像と比べると小さい物体の形状や路面表示など細かい模様を正しく再現できていないことが分かる。

表 3. 各画像における PSNR, 画像データサイズの平均値

	PSNR[dB]	Data size[kB]
入力画像（低解像度）	28.148	390.73
出力画像	28.954	414.34
正解画像（高解像度）	-	516.11

## 7 考察

本プロジェクトでは、低解像度の航空写真を高解像度化するという目的は果たせたが、正解画像との差を見ると精度は十分とは言えず、モデルの改善の余地があると考える。提案モデルでは、画像中の物体の輪郭については精度高く表現することができていた。これは、U-net のネットワーク構造に基づいて導入したスキップ接続が有効に働いたと考えられる。しかし、図 10 のように画像中の小さい物体や細かい模様に対する物体表現の再現が不十分であることが課題である。これは、低解像度画像で失われた物体の特徴をデコーダで補うためのネットワークの深さが足りないことやスキップ接続時に同じ階層の画像特徴のみしか得られていないことが原因として考えられる。

## 8まとめと今後の課題

低解像度の航空写真を高解像度化することはできたが、正解画像に近いレベルまでは到達していない。また、物体形状や細かさによって性能に大きな差が生まれることが確認できた。

今後の課題として、提案モデルの改良が挙げられる。今回のモデルの性能を改善するために、デコーダを深くするなどの複雑化とスキップ接続により得られる画像特徴の情報を多くすることが考えられる。これにより、細かい物体形状やより低解像度な画像に対しても有効に働くようになるが予想される。具体的には、スキップ接続の部分にも学習可能なパラメータを持ち、エンコーダの各階層から得た特徴をデコーダの出力連結することができる U-net++[3] の構造を応用することができると考えられる。

また、より多くのデータを用いて、様々な解像度の画像に対応できるようにモデルを学習することでによる汎用性の高いモデルの実現も必要だと考える。



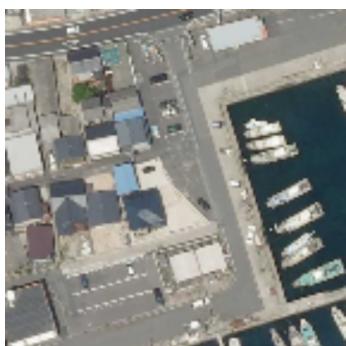
PSNR[dB]: 22.571  
Data size[kB]: 460.07



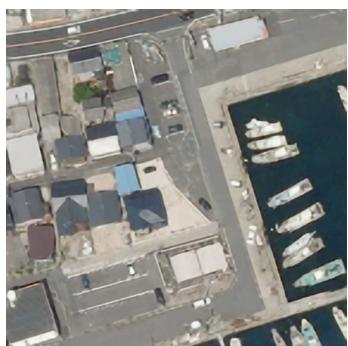
PSNR[dB]: 24.548  
Data size[kB]: 493.87



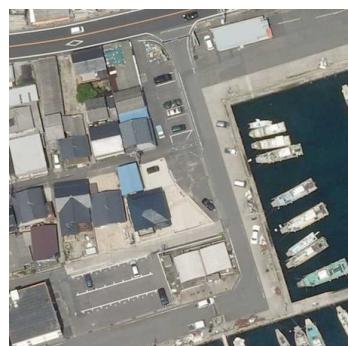
Data Size[kB]: 571.61



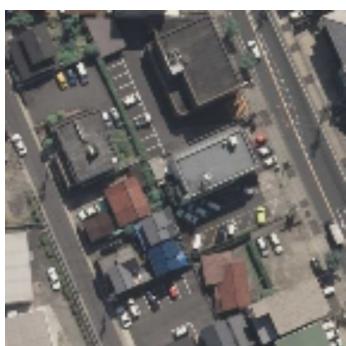
PSNR: 24.588  
Data size: 407.04



PSNR: 26.651  
Data size: 436.34



Data size: 505.50



PSNR: 25.772  
Data size: 428.96



PSNR: 27.588  
Data size: 468.68



Data size: 552.02

入力画像

出力画像

正解画像

図 8. 実験結果の定量的評価と定性的評価 1



PSNR[dB]: 24.081  
Data size[kB] 497.68



PSNR[dB]: 25.465  
Data size[kB]: 506.98



Data size[kB]: 625.20



PSNR: 23.037  
Data size: 522.13



PSNR: 24.965  
Data size: 541.33



Data size: 634.70



PSNR: 30.296  
Data size: 383.23



PSNR: 30.638  
Data size: 377.75



Data size: 523.00

入力画像

出力画像

正解画像

図 9. 実験結果の定量的評価と定性的評価 2



図 10. 上手く高解像度化できていない例

## 参考文献

- [1] Ronneberger, O., Philipp, F., Thomas, B.:U-Net: Convolutional Networks for Biomedical Image Segmentation, *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pp.234-241, (2015).
- [2] 国土地理院 : GSI データセット [https://gisstar.gsi.go.jp/gsi-dataset/index\\_ja.html](https://gisstar.gsi.go.jp/gsi-dataset/index_ja.html) (参照 2023-08-10).
- [3] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N. et al.: Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE transactions on medical imaging*, pp.1856-1867, (2019)