2019 年度　修士論文

深層学習を用いた 3 次元モデル超解像の学習安定化の検討

指導教員　椋木　雅之

宮崎大学大学院　工学研究科　工学専攻
機械・情報系コース　情報システム工学分野

学籍番号 T1703047
森　芳雄

概要

本論文では、3 次元モデルの超解像を行う 3D-Super Resolution Generative Adversarial Networks（3D-SRGAN）の学習を安定化させる方法の検討を行う。低解像度データから高解像度データを復元・生成する技術は超解像と呼ばれる。3 次元モデルの超解像を行う手法に 3D-SRGAN がある。3D-SRGAN の問題点として学習の不安定性があり、学習を続けていくと損失が最小の値に収束する前に学習が破綻することがある。学習の不安定性の原因として、ネットワーク構造の複雑さや学習手法の不適切さが考えられる。3D-SRGAN の層数を変更させ、パラメータの数を変化させることで学習の安定化の検討を行う。また、3 D-SRGAN に学習の安定化において有効性が知られている WGAN-GP の目的関数やハイパーパラメータの変更を取り入れることで、学習の安定化の検討を行う。従来手法と改良した手法で比較実験を行い、層数の変更は学習の安定化に影響しないこと、ハイパーパラメータの変更が学習の安定化に有効であることを示した。

目次

1. はじめに

3 次元モデルの表現方法にボクセル表現がある。ボクセル表現では、3 次元空間を格子状の細かい立方体(あるいは直方体)に分けて 3 次元モデルを表現する。ボクセル表現で高精細な 3 次元モデルを作成するには、多数のブロックを積み重ねる必要があり、手間が掛かる。粗い 3 次元モデルから高精細な 3 次元モデルを作成できれば、この手間を低減できる。このような粗いデータから高精細なデータを生成する処理を超解像[1]と呼ぶ。

3 次元ボクセルモデルの超解像を行う手法に 3D-Super Resolution Generative Adversarial Networks（3D-SRGAN）[2]がある。3D-SRGAN は深層学習を用いた生成モデルとして注目されている Generative Adversarial Network(GAN) [3]を用いている。GAN は Generator と Discriminator の 2 つのネットワークが互いに学習を行うことで能力を高めていく。データから特徴を学習することで、実在しないデータを作成することや、学習したデータに沿って入力データを変換することができる。しかし、GAN の学習は勾配消失問題等が原因で学習が安定しないことが知られている[4]。3D-SRGAN は GAN と同様に学習が安定せず、学習を続けていくと、収束計算が破綻することがある。

そこで、本研究では 3D-SRGAN の学習安定化について検討する。まず、ネットワーク構造が学習の安定性に影響するか調査する。3D-SRGAN のネットワーク構造を最適化して学習すべきパラメータの数を減らすことで、学習を安定化できるか調査する。次に、GAN の学習の安定化で有効性が知られている WGAN-GP[5]を 3D-SRGAN に取り入れ、学習の安定化の検討を行う。WGAN-GP は学習時の目的関数を変更することで学習の安定化を図っている。また、Discriminator と Generator の更新頻度や学習率等のパラメータ(ハイパーパラメータ)の変更も行っている。3D-SRGAN でも同様の変更を行うことで、学習の安定化を期待できる。

以下、2 章では、3D-SRGAN による 3 次元モデルの超解像について述べる。3 章では学習の安定化について議論する。4 章では 3D-SRGAN に変更を加えることで学習を安定化できるか、実験により評価する。最後に 5 章で、結論と今後の課題を述べる。

2. 3D-SRGAN による 3 次元モデルの超解像

2.1. 超解像の定義

超解像とは低解像度のデータから高解像度データを復元・生成する技術である。解像度とは、画素やボクセルが一定の長さの間にどれだけ存在しているかを表す量である。解像度が高いと、画像や 3 次元モデルを表現する格子が細かくなる。低解像度データは粗いデータとなり、高解像度データは高精細なデータとなる。

超解像は、低解像度データから高解像度データを生成する技術であるが、この問題は不良設定問題(ill-posed problem)[6]である。不良設定問題とは、解を求めるための必要な情報が一部欠けている問題のことである。低解像度データを高解像度データに変換するためには、低解像度データに存在しない部分のデータを生成する必要がある。この低解像度データに存在しない部分には、無数の生成パターンがありえる。データが周波数分解でき、サンプリング定理を満たす場合は、信号処理的アプローチで補間が行えるが、多くの実データでは、ノイズが含まれる上、デジタルデータでは量子化誤差も含まれるため、実用的には良い結果が得られない。そのため、一般に低解像度データから適切な高解像度データを生成することは難しい問題である。

2.2. 3D-SRGAN

3D-Super Resolution Generative Adversarial Networks(3D-SRGAN)は深層学習を用いた 3 次元モデル超解像の手法であり、画像の超解像で有効性が知られている Super Resolution Generative Adversarial Network(SRGAN)[7]を 3 次元モデルが扱えるように拡張したものである。SRGAN は Generative Adversarial Network(GAN)と呼ばれる生成モデルを応用している。GAN は Generator(生成器)と Discriminator(識別器)の 2 つのニューラルネットワークで構成されている。Generator はより学習データに近いデータを生成し、Discriminator は入力データが学習データか Generator が生成したデータかを識別する。Generator は Discriminator を騙すことができるように学習し、Discriminator は Generator が生成したデータを見破ることができるように学習する。このような学習により、最終的には Generator が学習データと同じようなデータを生成できることが期待される。この状態になると、Discriminator は学習データと生成データの識別ができなくため、Discriminator の正答率は約 50%となる。このような仕組みにより GAN は、学習データに非常に似た新たなデータを生成できる。3D-SRGAN は、GAN の持つこの特徴を利用して超解像を行う。そのため、自然な高解像度の 3 次元モデルを生成することができる。

2.3. SRGAN の特徴

3D-SRGAN の元となった SRGAN には以下の特徴がある。

(1) Content Loss[8]による主観的特徴の学習

(2) 2 つのネットワークによる敵対的な学習(GAN の利用)

(3) Residual Network(ResNet)[9]による深いネットワーク

(4) 逆畳み込みではなく Pixel Shuffler[10]を利用した画像拡大

1 つ目の特徴は、学習の際に Content Loss という Loss 関数を用いている点である。Content Loss には画像に何らかの操作を行っても、元の画像の大まかな見た目を維持しようとする性質がある。SRGAN では、Content Loss を利用することで、画像に拡大の操作を行っても主観的な特徴を変えることなく超解像を行うことができるようにしている。

2 つ目の特徴は、GAN を利用している点である。SRGAN では、GAN の持つ「学習データと見分けのつかないデータを生成する」性質を利用している。しかし、GAN は学習データに共通する性質を残した画像を生成するが、入力とは別の画像を生成してしまう。一方、上記の Content Loss は大まかな見た目を維持するだけであり、これだけではぼやけた画像が生成されてしまう。例えば、学習データが顔画像であった場合、入力に顔画像を与えると、GAN だけでは入力とは別人の顔画像が生成される。一方、Content Loss だけでは入力と同じ人の顔が生成されるが細部までは復元できない。そのため、SRGAN は Content Loss と GAN を適度に組み合わせて超解像を行っている。

3 つ目の特徴は、ResNet を利用している点である。ResNet は、Convolutional Neural Network(CNN)の層を深くしたニューラルネットワークの構造である。CNN では、層を深くする(層の数を多くする)とより高度で複雑な特徴を抽出できるとされている。しかし、単純に層を深くすると学習の性能が悪化してしまう[11][12]。ResNet では層毎に最適な出力を求めるではなく、層の入力と出力の

差を学習している。これにより、ResNet では学習性能を悪化させることなく深いネットワーク構造を実現できるため、高度で複雑な特徴を抽出することが期待できる。

4 つ目の特徴は、Pixel Shuffler 処理により画像拡大を行っている点である。深層学習では、低次元の特徴マップから高次元の特徴マップを生成する際に逆畳み込み処理が利用される。しかし、逆畳み込み処理では処理速度が遅く、精度があまり高くないと考えられている [13]。Pixel Shuffler は、入力の特徴マップの各ピクセルを並び替えて高解像度な特徴マップを出力する。処理がメモリコピーだけであるため処理速度が速く、オーバーラップがなく生成画像がぼやけにくい性質がある。

SRGAN はこれらの特徴を組み合わせることで、より本物に近い画像が生成でき、超解像の課題である可能な無数のパターンから自然なパターンを選択することを達成している

2.4. SRGAN から 3D-SRGAN への拡張

3D-SRGAN は SRGAN を 3 次元モデル超解像ができるように拡張したものであり、入出力データとしてボクセル表現の 3 次元モデルが扱えるようにしている。

3D-SRGAN では、扱うデータが 3 次元モデルとなっている。SRGAN は画像を扱うため 2 次元畳み込み層を用いるが、3D-SRGAN では代わりに 3 次元畳み込み層を利用している。2 次元畳み込み層でカーネルのサイズが$k \times k$の畳み込み処理をしていた場合、3 次元畳み込み層ではカーネルのサイズが$k \times k \times k$の畳み込み処理をする。

また、3 次元モデルのサイズを拡大する際には、Pixel Shuffler を 3 次元モデルに拡張した Voxel Shuffler を利用する。Pixel Shuffler は、もともと画像を拡大するための手法である。入力の特徴マップを並び替えて画像を拡大することができる。Voxel Shuffler は、Pixel Shuffler に奥行きを追加することで、3 次元モデルを拡大できるようにしている。

2.5. 3D-SRGAN のネットワーク構造

3D-SRGAN は GAN と同じく Generator と Discriminator のネットワークで構成されており、入出力データとしてボクセル表現の 3 次元モデルを扱う(図 1)。学習データには低解像度 3 次元モデルと高解像度 3 次元モデルのペアを使用する。Generator は低解像度 3 次元モデルから高解像度 3 次元モデルを生成し、Discriminator は入力された 3 次元モデルが学習データの高解像度 3 次元モデルなのか Generator が生成した 3 次元モデルなのかを識別する。この 2 つは敵対的関係にあり、それぞれの目的関数は、Generator は Discriminator を騙すように学習データと似た 3 次元モデルを生成することであり、Discriminator は学習データと生成された 3 次元モデルを見分けることである。3D-SRGAN の最終的な目的は、Discriminator が識別できないような 3 次元モデルを Generator が生成できることである。また、学習の結果得られた Generator が生成モデルとなる。

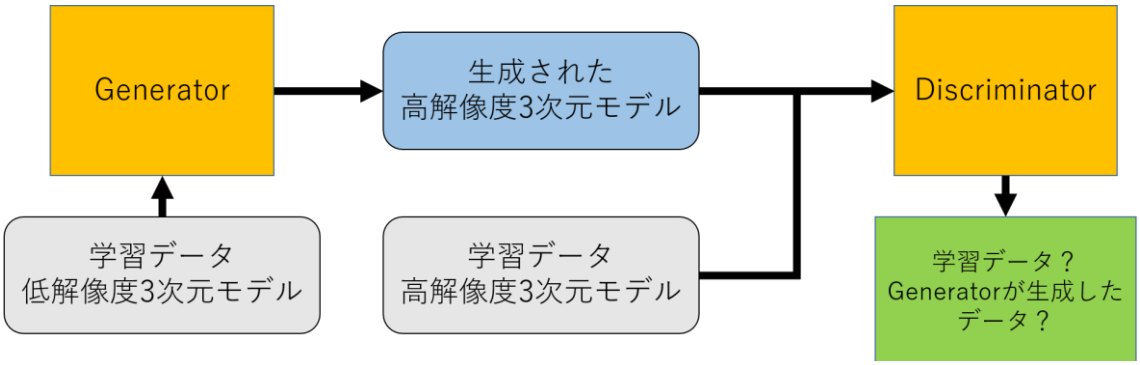
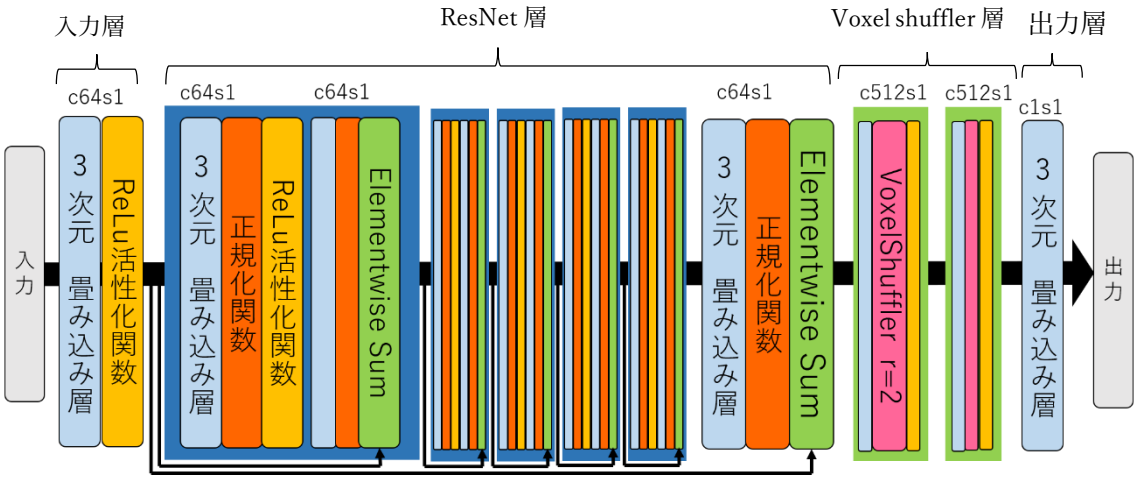


図 1. GAN の学習の概要図[5]



c: 出力チャネル数、s: ストライド数

図 2. Generator の構造

2.6. 3D-SRGAN の Generator の構造

Generator のネットワークの構造について説明する。Generator は図 2 のような構造になっている。以降、入力のチャネル数をC_{in}、出力のチャネル数をC_{out}、ストライドをs、カーネルのサイズをkとする。ここでストライドとは、畳み込み処理を行うときにカーネルを少しずつずらしていく間隔のことである。Generator の 3 次元畳み込み層のストライドは全てs=1 である。

入力には学習データの低解像度 3 次元モデルが与えられ、出力として高解像度 3 次元モデルを生成する。

Generator の構造は大きく入力層、ResNet 層、Voxel Shuffler 層、出力層の 4 つに分けられる。入力層では、入力で与えられた 3 次元モデルを 3 次元畳み込み層によって、複数のチャネルとして ResNet 層に与えている。ResNet 層では 3 次元モデルの特徴を抽出し、Voxel shuffler 層で 3 次元モデルを拡大している。最後に出力層で入力の特徴マップから 3 次元モデルを生成し、出力している。

3D-SRGAN の Generator では、3 次元畳み込み層を利用している。3 次元の畳み込み層では、カーネルのサイズが$k \times k \times k$の畳み込み処理をする。

3 次元畳み込み層の後の処理に、正規化関数や活性化関数がある。正規化関数では 3 次元畳み込み層の出力を正規化している。この処理をすることで学習速度を速くすることができ、学習の安定化を図ることができる[14]。Generator では活性化関数に ReLu 活性化関数(図 3)を使用しており、式(1)で表現できる。

$$f(x) = \max(0, x) \qquad (1)$$

次にそれぞれの層の処理を説明する。まず、入力層では、C_{in}=1、C_{out}=64、k=9 の 3 次元畳み込み層、ReLu 活性化関数を順番に処理する。

ResNet 層は 5 つの Residual Block で構成されている。Residual Block の中身は、C_{in}=64、C_{out}=64、k=3 の 3 次元畳み込み層、正規化関数、ReLu 活性化関数、C_{in}=64、C_{out}=64、k=3 の 3 次元畳み込み層、正規化関数の順番で構成されている。それぞれの Residual Block では入力を出力と加算している。即ち各 Residual Block はそれぞれの入力と出力の差を学習することになる。5 つの Residual Block の処理の後、C_{in}=64、C_{out}=64、k=3 の 3 次元畳み込み層、正規化関数で処理を行い、ResNet 層の入力を加算して出力している。

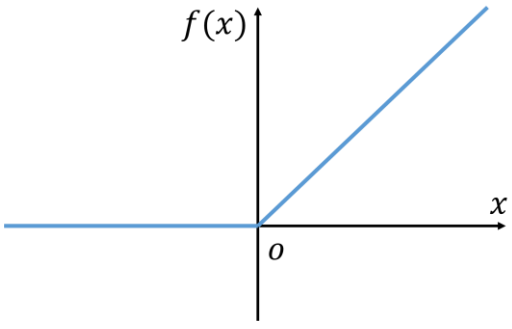


図 3. ReLu 活性化関数[5]

Voxel shuffler 層では、実際に 3 次元モデルのサイズを拡大していく(図 4)。そのために入力の特徴マップを並び替えて 3 次元モデルを拡大することができる Voxel Shuffler を利用する。Voxel shuffler 層は 2 つのブロックで構成されており、1 つの Block で入力の特徴マップから 2 倍に拡大している。1 つのブロックの中身はC_{in}=64、C_{out}=512、s=1、k=3 の 3 次元畳み込み層、C_{in}=512、C_{out}=64 の Voxel Shuffler 層、Relu 活性化関数を順に処理している。この 2 つのブロックを通過した出力は 4 倍に拡大されている。

最後にC_{in}=64、C_{out}=1、s=1、k=9 の 3 次元畳み込み層で処理し、高解像度 3 次元モデルを出力している。

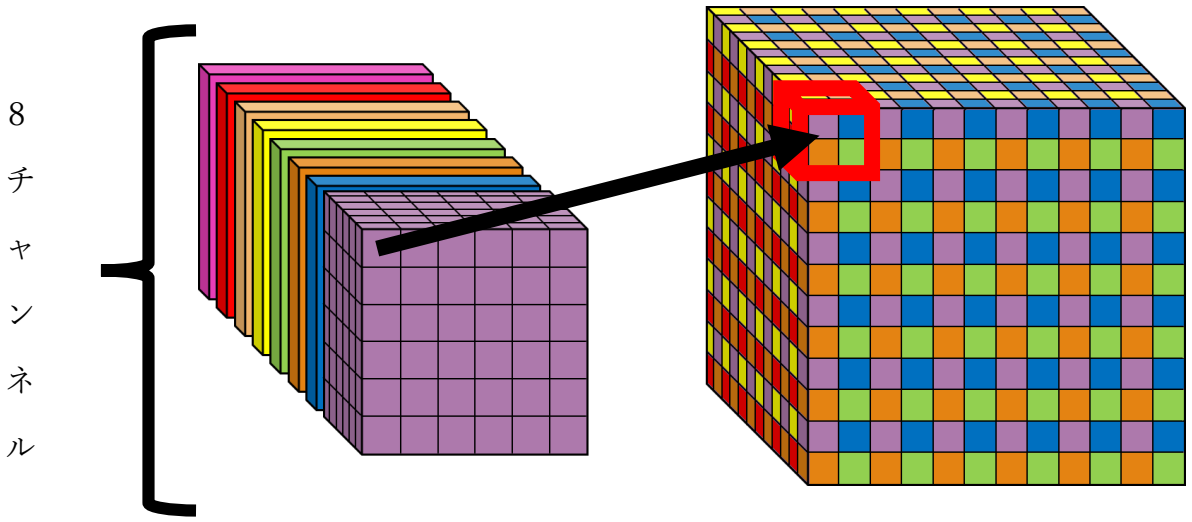
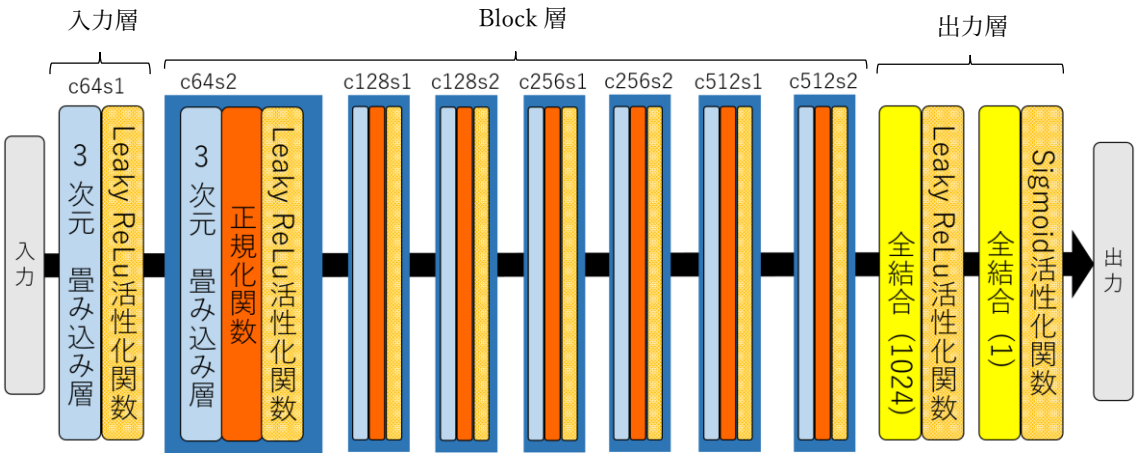


図 4. Voxel Shuffler



c: 出力チャネル数、s: ストライド数

図 5. Discriminator の構造と各パラメータ

2.7. 3D-SRGAN の Discriminator の構造

Discriminator は図 5 の構造になっている。Discriminator の構造は大きく入力層、Block 層、出力層の 3 つに分けられる。入力層では入力の 3 次元モデルから複数のチャネルを出力し、Block 層で 3 次元モデルの特徴を抽出し、出力層では入力の 3 次元モデルが学習データである確率が出力される。

3 D-SRGAN の Discriminator でも Generator と同様に 3 次元畳み込み層を利用し、3 次元モデルの畳み込み処理を行っている。

Discriminator では、活性化関数として Leaky ReLu 活性化関数(図 6)と Sigmoid 活性化関数(図 7)を使用しており、それぞれ式(2)(3)と表現できる。

$$f(x) = max(\alpha x, x) \ , 0 < \alpha < 1 \qquad (2)$$

$$f(x) = \frac{1}{1 + \exp(-\beta x)} \ , 0 < \beta \qquad (3)$$

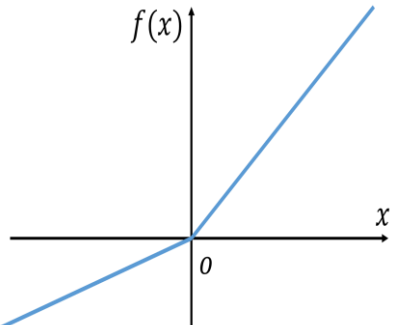


図 6. Leaky ReLu 活性化関数[5]

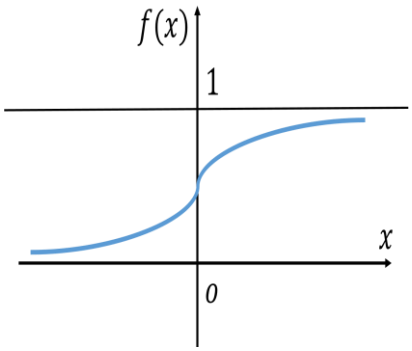


図 7. Sigmoid 活性化関数[5]

次にそれぞれの層の処理を説明する。まず、最初の層では、$C_{in} = 1$、$C_{out} = 64$、$s = 1$の 3 次元畳み込み層、Leaky ReLu 活性化関数の順に処理している。次の Block 層は 7 つのブロックで構成されている。1 つのブロックの中身は、3 次元畳み込み層、正規化関数、Leaky ReLu 活性化関数の順に処理している。また、それぞれの Block の 3 次元畳み込み層のパラメータは、層毎に順にC_{in}=64、64、128、128、256、256、512、C_{out}=64、128、128、256、256、512、512、s=2、1、2、1、2、1、2 となっている。3 次元畳み込み層のカーネルサイズは全てk=3 である。

7 つのブロックの処理後、C_{out} =1024 の全結合、Leaky ReLu 活性化関数、$C_{out} = 1$の全結合、Sigmoid 活性化関数を順番に処理している。最終的な出力の値は[0,1]の間の値となる。この値が、Discriminator の入力が学習データである確率を表している。

2.8. Generator の学習更新

Generator の Loss 関数l_G^{3DSR}は式(4)で表現できる。

$$l_G^{3DSR} = \frac{1}{m}(l_{con}^{3DSR} + 10^{-3} l_{Gen}^{3DSR}) \quad (4)$$

$$l_{con}^{3DSR} = \sum^{m} \frac{1}{r^3 WHD} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \sum_{z=1}^{rD} \left(I_{x,y,z}^{HR} - G(I^{LR})_{x,y,z}\right)^2 \quad (5)$$

$$l_{Gen}^{3DSR} = \sum^{m} -\log D\left(G(I^{LR})\right) \quad (6)$$

式(4)のl_{con}^{3DSR}は Content Loss、l_{Gen}^{3DSR}は Adversarial Loss、mはバッチサイズを表している。

Content Loss は、式(5)で計算される。W、H、Dはそれぞれ 3 次元空間の幅、高さ、奥行きを表している。式(5)では、学習データの高解像度 3 次元モデルI^{HR}と Generator が生成した高解像度 3 次元モデル$G(I^{LR})$の平均二乗誤差を計算している。Content Loss には、元のデータの見た目をある程度そのままにして、別の操作を加えることができる特徴がある。そのため、超解像しても元のデータの見た目から大きく変化する心配がない。

Adversarial Loss は、式(6)で計算される。$G(I^{LR})$は Discriminator で識別された結果、$D(\cdot)$は Discriminator の出力である。Adversarial Loss は、GAN で通常使われている Loss 関数で、学習データと見分けがつかないように学習をする特徴がある。Content Loss だけでは、見た目の変化の問題が解決するだけで、はっきりとした 3 次元モデルが生成されず、Adversarial Loss だけでは学習データの性質を保持しているが、入力とは別の 3 次元モデルが生成されてしまう。そのため、この 2 つを適切に組み合わせることで超解像を行っている（式(4)では$1 : 10^{-3}$）。

Generator の学習では、この Loss 関数の値が小さくなるよう、ネットワークのパラメータを逐次更新する収束計算を行う。

2.9. Discriminator の学習更新

Discriminator の Loss 関数l_D^{3DSR}は式(7)で表現できる。

$$l_D^{3DSR} = \frac{1}{m}\sum^{m}\left[\log D(I^{HR}) - \log\left(1 - D\left(G(I^{LR})\right)\right)\right] \quad (7)$$

mはバッチサイズを表している。$D(x)$は Discriminator の出力であり、入力データxが学習データの 3 次元モデルである確率を表している。そのため、Discriminator の識別がうまくいくと$D(I^{HR})$の値が大きくなり、$D\left(G(I^{LR})\right)$の値は小さくなる。

Discriminator の学習も、Generator と同様の収束計算により行う。

3. 学習の安定化

3.1. 学習の不安定性

GAN では、Discriminator を騙せるように Generator を学習することで生成データの分布を本物データに近づけていく。そのために Loss 関数が小さくなるよう、収束計算によりパラメータを更新していく。収束計算は、Loss 関数の勾配に基づいて行われるが、GAN の学習では Loss 関数の勾配が 0 になってしまう勾配消失や勾配が大きくなりすぎてしまう勾配爆発によって学習が不安定になってしまう問題がある。学習が安定しないと、Loss 関数が最小の値に収束する前に学習が破綻してしまう可能性がある。

図 8 は 3D-SRGAN で学習を行っている時の Generator の Loss 関数の値(Loss)を学習回数 1 回ごとに示したものである。学習回数 169 回付近で Generator の Loss が突然大きくなっており、勾配爆発を起こしたと考えられる。図 9 は Loss が大きくなった後の学習(学習回数 200 回)で得られた生成モデルを用いて超解像を行った例である。低解像度 3 次元モデルを超解像して高解像度 3 次元モデルのような 3 次元モデルを生成してほしいが、学習が破綻した後の生成モデルを用いて超解像を行うと、空間を埋めるような超解像 3 次元モデルが得られ、適切な超解像を行えていない。3D-SRGAN は ResNet 層の利用や正規化関数の導入など、安定化のための仕組みを取り入れているが、十分ではなく、学習が破綻することが多い。

このように学習が不安定だと、収束計算の途中で学習が破綻し適切な超解像を行える生成モデルが得られない問題が生じる。

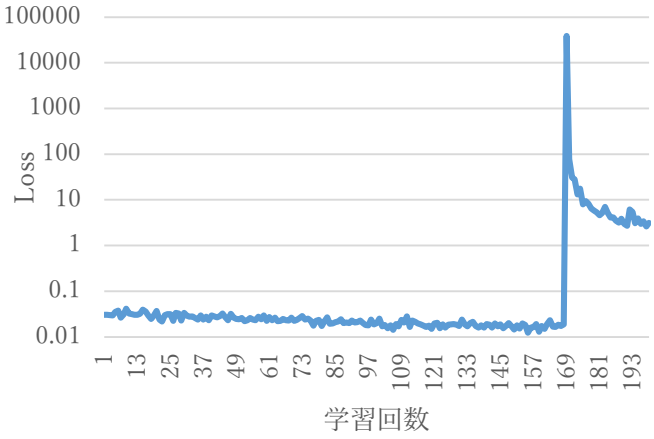
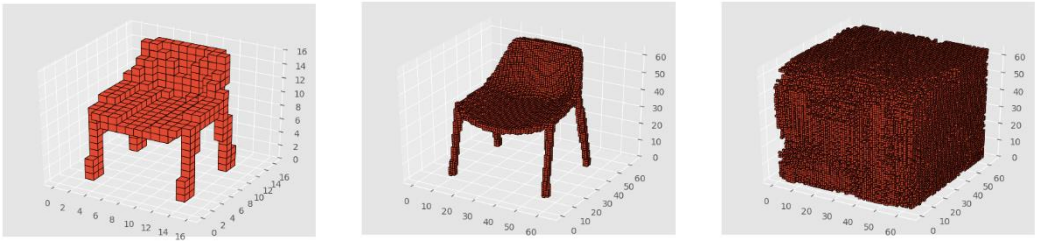


図 8. 3D-SRGAN の Generator の損失



低解像度 3 次元モデル　　高解像度 3 次元モデル　　超解像 3 次元モデル

図 9. 適切な超解像を行えていない例

学習が不安定になる原因としてはネットワークの構造が複雑であること、学習手法が不適切であることが考えられる。構造が複雑であると学習すべきパラメータが多くなり、一部で勾配消失や勾配爆発が起こり、学習が不安定になる。層数を減らすなどして構造を調整し、パラメータの数を適切に変化させることで学習の安定化を期待できる。また、学習手法が不適切である場合は、学習率等の学習のためのパラメータであるハイパーパラメータの最適化や目的関数を変更することで学習の安定化を期待できる。

3.2. WGAN-GP

学習の安定化に有効な手法に WGAN-GP[5]がある。WGAN-GP は GAN から目的関数の変更とハイパーパラメータの変更を行っている。

通常の GAN の目的関数は式(8)で表される。

$$\mathbb{E}_{x\sim\mathbb{P}_r}[\log D(x)] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}\left[\log\left(1 - D(\tilde{x})\right)\right] \quad (8)$$

一方、WGAN-GP の目的関数は式(9)で表される。

$$\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \underbrace{\lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2]}_{\text{Gradient Penalty}} \quad (9)$$

式(8)、(9)の$D(\cdot)$は Discriminator のネットワークの出力を表しており、$\tilde{x}$は生成データ、xは本物データを表している。Generator が本物データと似ているデータを生成できないと$D(\tilde{x})$は小さくなり、$D(x)$との差が大きくなる。Generator が学習データと似ているデータを生成できるようになると$D(\tilde{x})$は大きくなり、$D(x)$との差は小さくなる。Generator は$D(\tilde{x})$を最大化しようとし、Discriminator は$D(x)$と$D(\tilde{x})$の差を最大化しようとする。

通常の GAN の式(8)は勾配消失が起こりやすい形になっている[5]。WGAN-GP ではこれを式(9)の第 1、第 2 項のように変更することで、この問題を低減している。さらに、WGAN-GP では勾配消失や勾配爆発を防ぐため、勾配の制約として Gradient Penalty を用いている。Gradient Penalty により、勾配のノルムを一定値に近づけることで勾配消失や勾配爆発を防いでいる。

ハイパーパラメータの変更では、Generator の学習を抑制している。GAN では、Generator の学習が過度に進むと学習が不安定になる傾向がある。Generator と Discriminator の更新を 1:5 の頻度で行うことで、両者のバランスをとっていると考えられる。また、学習率は2×10^{-4}と比較的小さい値に設定している。学習率を小さく設定することで、Generator の学習が過度に進むことを抑制する効果があると考えられる。

表 1. データセット ModelNet10

クラス名	bathtub	bed	chair	desk	dresser
学習データ数	106	515	889	200	200
テストデータ数	50	100	100	86	86
クラス名	monitor	night_stand	sofa	table	toilet
学習データ数	465	200	680	392	344
テストデータ数	100	86	100	100	100

4. 実験

4.1. 層数の違いによる学習安定化の調査

4.1.1. 実験方法

この実験では、3D-SRGAN の構造を変化させることが、学習の安定化につながるか調査した。具体的には ResNet 層の Residual Block を 3 層、4 層、5 層、6 層、7 層に変化させて学習を行い、学習が安定化するか調査した。

学習、テストに用いる 3 次元モデルとして、データセット ModelNet10[15]を使用した。このデータセットは bathtub、bed、chair、desk、dresser、monitor、night_stand、sofa、table、toilet の 10 クラスの 3 次元モデルがあり、それぞれ学習データとテストデータが用意されている。表 1 にデータセット ModelNet10 の内訳を示す。

学習前の準備として、学習データとテストデータをそれぞれ、ボクセル数が $16 \times 16 \times 16$ と $64 \times 64 \times 64$ の 3 次元モデルに変換した。前者を低解像度 3 次元モデル、後者を高解像度 3 次元モデルと呼ぶ。それぞれの層数の 3 D-SRGAN で、chair の学習データの低解像度 3 次元モデルと高解像度 3 次元モデルのペアを 200 回学習させた。それぞれの層数で超解像を行い生成した 3 次元モデルを超解像 3 次元モデルと呼び、データセットの高解像度 3 次元モデルと区別する。

学習が安定化したかを確認するため、それぞれの Generator の Loss を確認し

た。また、主観的な評価として、それぞれの超解像 3 次元モデルを同じ角度から見た結果を画像として見比べた。さらに、chair のテストデータの高解像度 3 次元モデルと超解像 3 次元モデルの同じ座標間でのボクセルの有無の違いを誤差としてそれぞれ計算し、最も適切な超解像を行う生成モデルで比較した。

4.1.2. 結果・考察

図 10 に各層数での Generator の Loss の学習回数による変化を示した。縦軸は Generator の Loss を表しており、横軸は学習データのペアすべての学習を何回行ったか(学習回数)を表している。さらに、図 11、図 12 にそれぞれの手法で 50 回、100 回、150 回、200 回学習した生成モデルで chair のテストデータを超解像した結果を示した。表 2 に最も適切な生成モデルを使った超解像 3 次元モデルと高解像 3 次元モデルの誤差の平均を示す。小数点以下を四捨五入して示している。

まず図 10 からは 5 層では、学習回数 166 回程で Generator の Loss が大きくなっているのが分かる。層数を減らした 3 層は学習回数 160 回程、4 層は 130 回程で Generator の Loss が大きくなっていた。また、層数を増やした 6 層、7 層ではどちらも学習回数 145 回程で Generator の Loss は大きくなっていた。

図 11、12 からは主観的に見ると、3 層では学習回数 50 回、5 層では、150 回、4 層、6 層、7 層では 100 回のときそれぞれ最も適切に超解像を行えていた。また、5 層の学習回数 100 回のときは欠けている部分はあるが大きく形が崩れてはいなかった。3 層、5 層、7 層は Loss が大きくなった後の超解像した結果は空間を埋めるような 3 次元モデルとなっていた。4 層の Loss が大きくなった後の超解像した結果は、形が崩れている 3 次元モデルであった。6 層の Loss が大きくなった後の超解像した結果は 3 次元モデルが生成されていないものとなっていた。

表 2 の結果から、5 層が生成結果と高解像度 3 次元モデルの誤差が少ないが、4 層と 7 層も誤差は 5 層のときと近い値になっている。

いずれの層数でも、Loss は途中で大きくなっていた。超解像した結果も、Loss が大きくなった後は空間を埋めるような 3 次元モデルが生成されていたり、何も生成されていないものもあり、学習回数が多くなると適切な超解像が行えていなかった。層数を変えても不安定さは解消されていなかった。

層数を減らして学習するパラメータ数を減らせば、学習が安定化するならば、Loss が大きくなるまでの学習回数は、層数が小さいほど長くなると考えられるが、そのような傾向も見られなかった。また、誤差の平均は 5 層が最も少なくなっており、生成精度からは 5 層が最適であり、パラメータ数が多すぎるわけではなかった。

これらの結果から層数を変えても学習は安定化しないと考えられる。

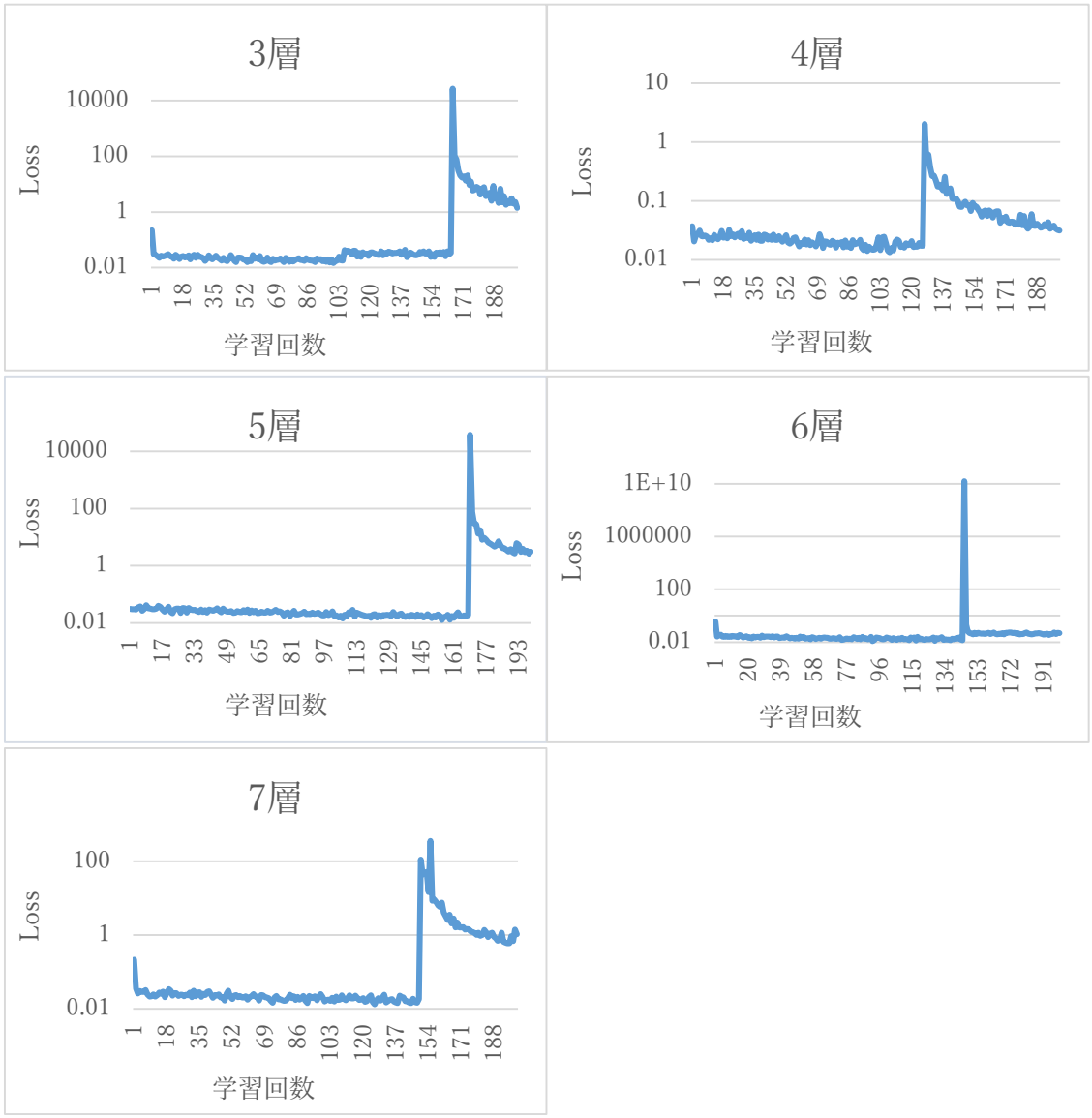


図 10. 各 ResNet 層数の Generator の Loss

図 11. 各 ResNet 層数での 3 次元モデル超解像(1)

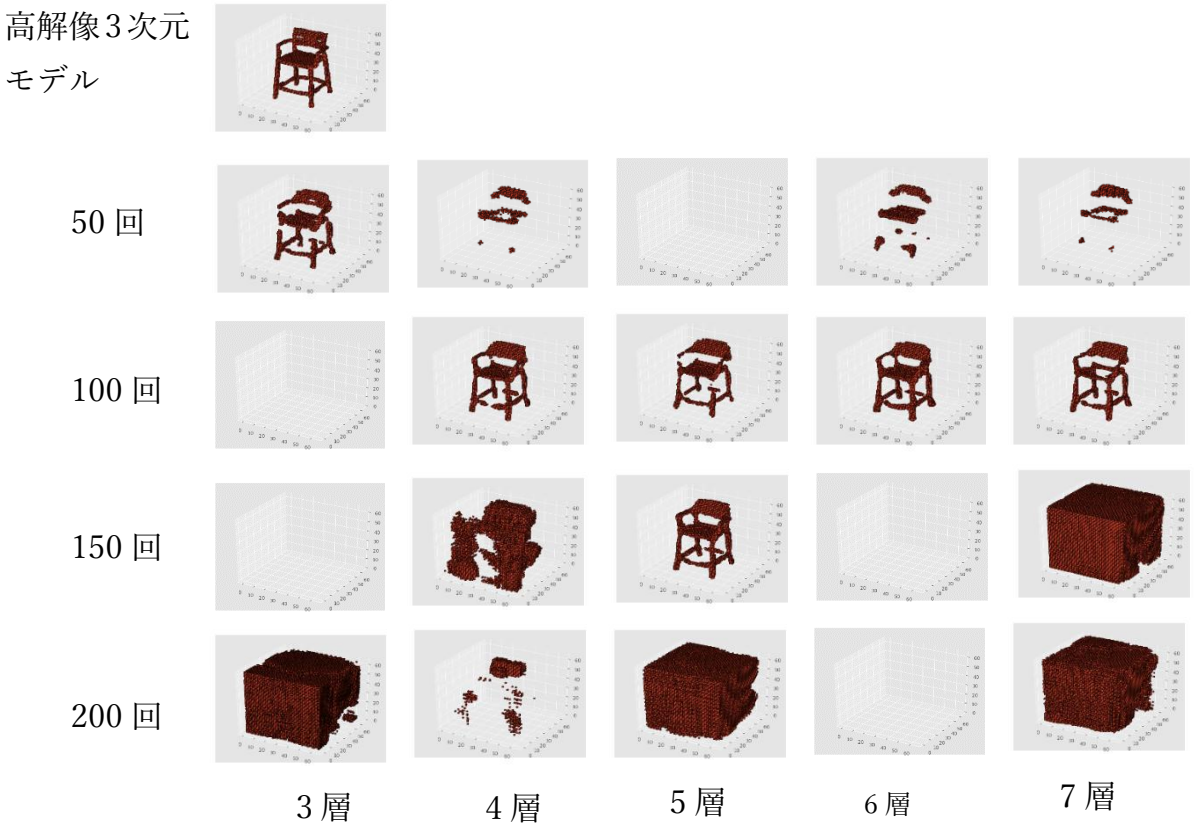


図 12. 各 ResNet 層数での 3 次元モデル超解像(2)

表 2. 各層数の超解像 3 次元モデルと高解像度 3 次元モデルの誤差の平均

	3 層	4 層	5 層	6 層	7 層
誤差の平均	8148	7776	7505	8331	7607
学習回数	50	100	150	100	100

4.2. 学習手法の変更による学習安定化の調査

4.2.1. 実験手法

この実験では、WGAN-GP を参考に学習手法の変更により学習を安定化できるか調査した。具体的には、学習のためのハイパーパラメータの変更と目的関数の変更を行った。3D-SRGAN に WGAN-GP で行われていたハイパーパラメータの変更を加えて学習した生成モデルを 3D-SRGAN_G1D5、3D-SRGAN に WGAN-GP で行われていたハイパーパラメータの変更と目的関数の変更の両方を加えて学習した生成モデルを 3D-SRWGAN とよぶ。これらの生成モデルと従来の 3D-SRGAN で学習した生成モデルで比較実験を行った。

ハイパーパラメータの変更内容は Generator の学習率を 2.5×10^{-3} から 2×10^{-4}、Discriminator の学習率を 1×10^{-5} から 2×10^{-4}、Generator と Discriminator の更新頻度を 1:1 から 1:5 にするものである。これは WGAN-GP で使われているパラメータに合わせている。

3D-SRWGAN の目的関数は式(8)を用いている。Generator の Loss 関数は 3D-SRGAN の Generator の Loss 関数の式(6)を式(9)に変更したものとなる。

$$l_{Gen}^{3DSRW} = \sum^{m} -D\big(G(I^{LR})\big) \qquad (9)$$

Discriminator の Loss 関数は 3D-SRGAN の Discriminator の Loss 関数の式(7)を式(10)に変更したものである。

$$l_{D}^{3DSRW} = \frac{1}{m}\sum^{m}\big[D(I^{HR}) - D\big(G(I^{LR})\big)\big] \qquad (10)$$

Gradient Penalty は式(11)で表現できる。

$$gp = \lambda\frac{1}{m}\sum^{m}\left[\left(\left\|\nabla D\big(\varepsilon I^{HR} + (1-\varepsilon)G(I^{LR})\big)\right\|_2 - 1\right)^2\right] \qquad (11)$$

式(11)のλは WGAN-GP で使われている 10 にしている。また、εは 0 以上 1 未満の一様乱数を表している。Discriminator 更新時にgpを加えて収束計算を行う。

4.2.2. 実験方法

3 D-SRGAN と 3D-SRGAN_G1D5、3D-SRWGAN で、学習の安定性と超解像を行った際の誤差を比較した。3 次元モデルとして、データセット ModelNet10 を使用した。3 つの手法それぞれで、chair の学習データの低解像度 3 次元モデルと高解像度 3 次元モデルのペアを学習させた。それぞれの手法で超解像を行い、3 次元モデルを生成した。

学習が安定化したかを確認するため、それぞれの Generator の Loss を確認した。また、主観的な評価として、それぞれの超解像 3 次元モデルを同じ角度から見た結果を画像として見比べた。さらに、chair のテストデータの高解像度 3 次元モデルと超解像 3 次元モデルの同じ座標間でのボクセルの有無の違いを誤差としてそれぞれ計算し、最も適切な超解像を行う生成モデルで比較した。

4.2.3. 結果・考察

図 13 に各層数での Generator の Loss の学習回数による変化を示した。縦軸は Generator の Loss を表しており、横軸は学習回数を表している。さらに、図 14 にそれぞれの手法で 50 回、100 回、150 回、200 回学習した生成モデルで chair のテストデータを超解像した結果を示した。図 15 に 3D-SRGAN_G1D5 で chair のテストデータを超解像した結果を学習回数 400 回まで 100 回毎に示した。図 16 に 3D-SRWGAN で chair のテストデータを超解像した結果を学習回数 1000 回まで 200 回毎に示した。さらに、表 3 に最も適切な生成モデルを使った超解像 3 次元モデルと高解像度 3 次元モデルの誤差の平均を示す。小数点以下を四捨五入して示している。図 17 にそれぞれの手法の学習回数 50 回毎の

誤差を示した。

図 13 を見ると 3D-SRGAN の Loss は 170 回程で大きくなっており、その後も前と比べると大きいままである。3D-SRGAN_G1D5 は学習を 400 回まで行っており、Loss は途中で極端に大きくなることはなく、収束していっているように見える。3D-SRWGAN は 1200 回学習を行っており、Loss は収束していっているように見える。

図 14 を見ると、3D-SRGAN は学習回数 50 回と 200 回では、適切な超解像を行えていない。3D-SRGAN_G1D5 は 50 回のときは一部欠けているが 100 回以降は高解像度モデルと似た 3 次元モデルを生成出来ており、適切な超解像を行えている。3D-SRWGAN はどの回数でも高解像度 3 次元モデルと似た 3 次元モデルとなっているので、適切な超解像を行えている。図 15 では、3D-SRGAN_G1D5 はところどころ欠けている部分があるが、大きく形が崩れているわけではなかった。図 16 を見ると、3D-SRWGAN は学習回数が多くなっても超解像した 3 次元モデルは形が大きく崩れることはなかった。

3D-SRGAN_G1D5 と 3D-SRWGAN は学習回数を増やしていっても Loss は収束していっており、適切な超解像も行うことができていた。また、表 3 を見ると超解像 3 次元モデルと高解像度 3 次元モデルの誤差の平均は 3D-SRGAN_G1D5 が最も小さくなっていた。図 17 では、3D-SRGAN_G1D5 は 350 回までは誤差は収束していっている。400 回で誤差が増えているが、この時たまたま誤差が大きくなっていただけの可能性がある。3D-SRWGAN は 1000 回まで誤差は収束していっており、学習を続けることでより誤差が小さくなる可能性もある。3D-SRGAN_G1D5 と 3D-SRWGAN の両方に加えたハイパーパラメータの変更が学習の安定化に有効であったと考えられる。

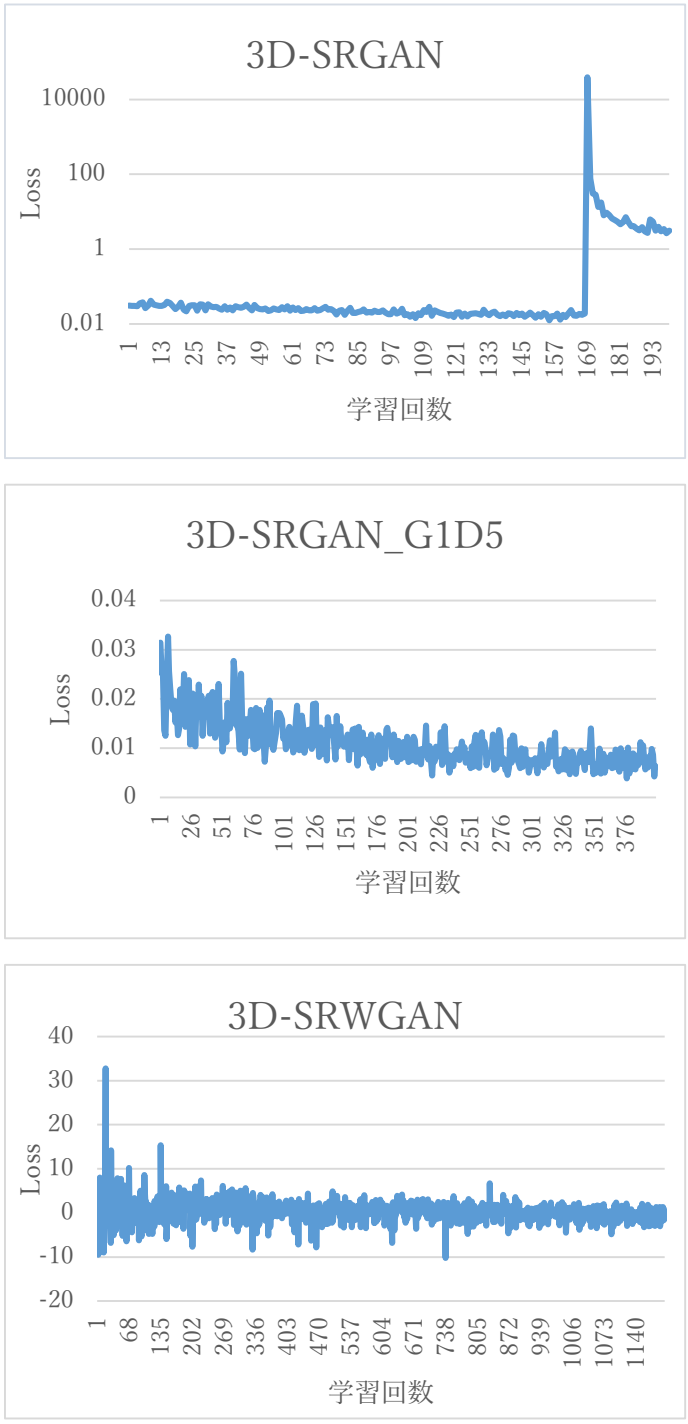


図 13. 3 つの手法の Generator の Loss

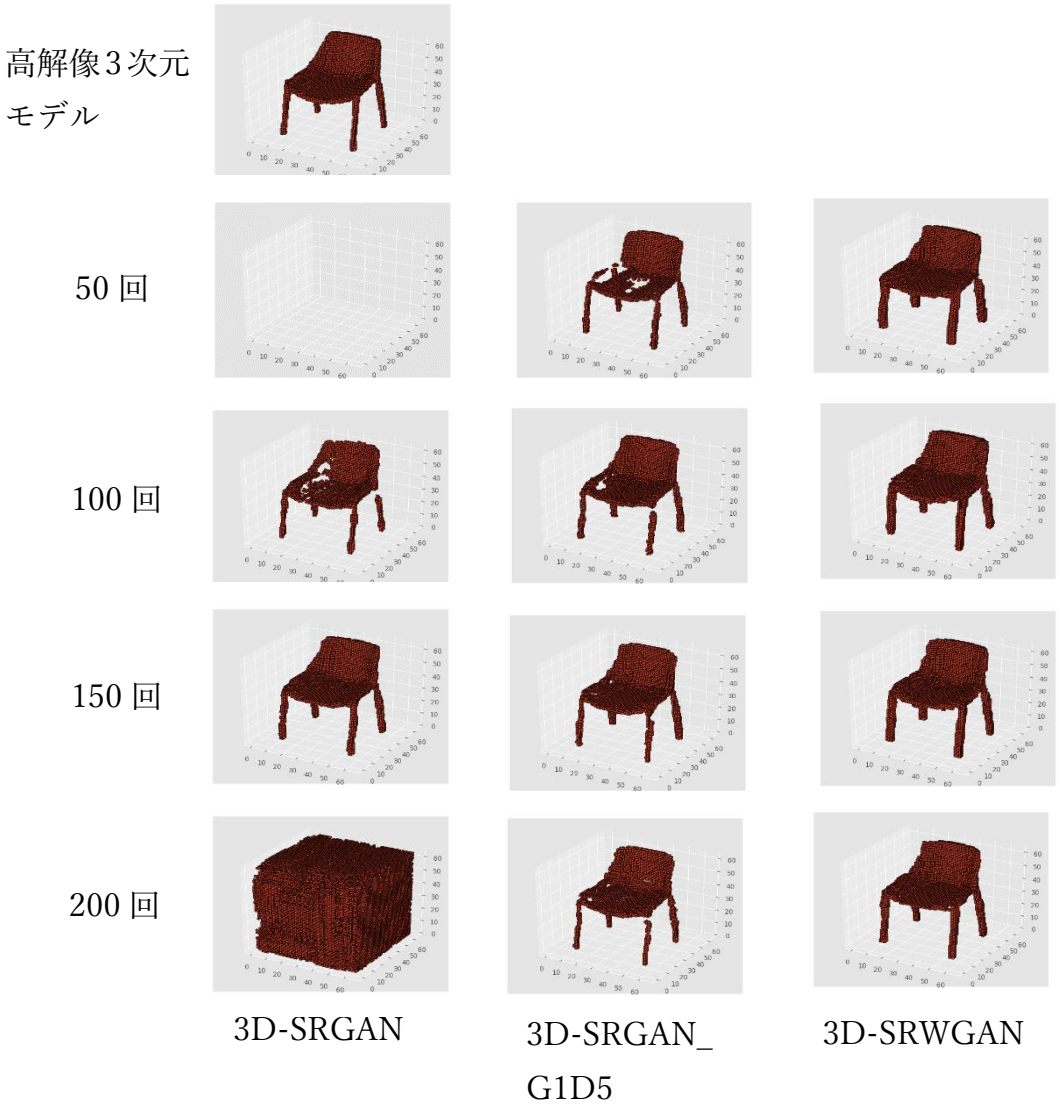


図 14. 3D-SRGAN、3D-SRGAN_G1D5、3D-SRWGAN による 3 次元モデル超解像

図 15. 3D-SRGAN_G1D5 による 3 次元モデル超解像

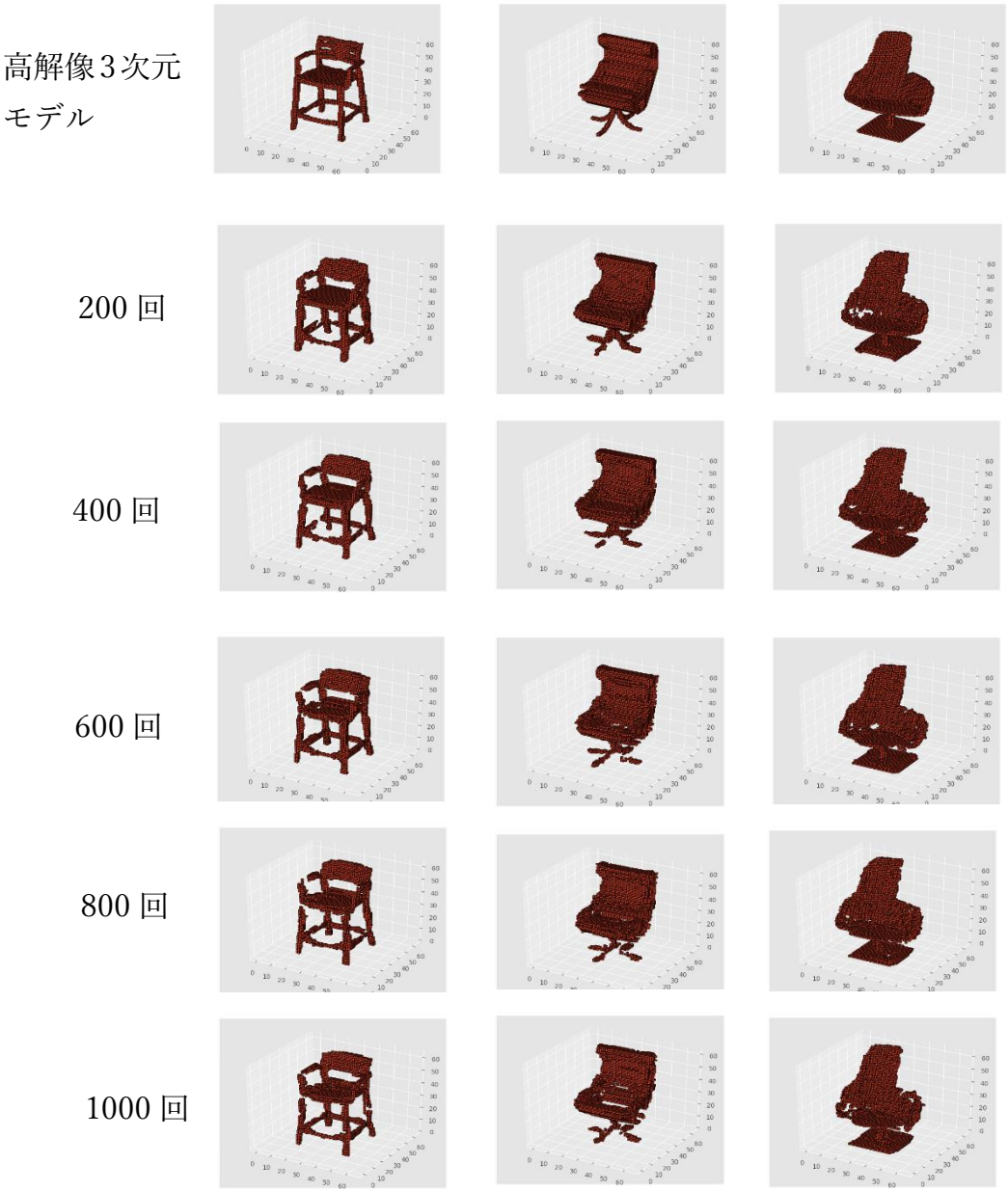


図 16. 3D-SRWGAN による 3 次元モデル超解像

表 3. 3 つの手法の超解像 3 次元モデルと高解像度 3 次元モデルの誤差の平均

	3D-SRGAN	3D-SRGAN_G1D5	3D-SRWGAN
誤差の平均	7505	7045	7302
学習回数	150	250	1000

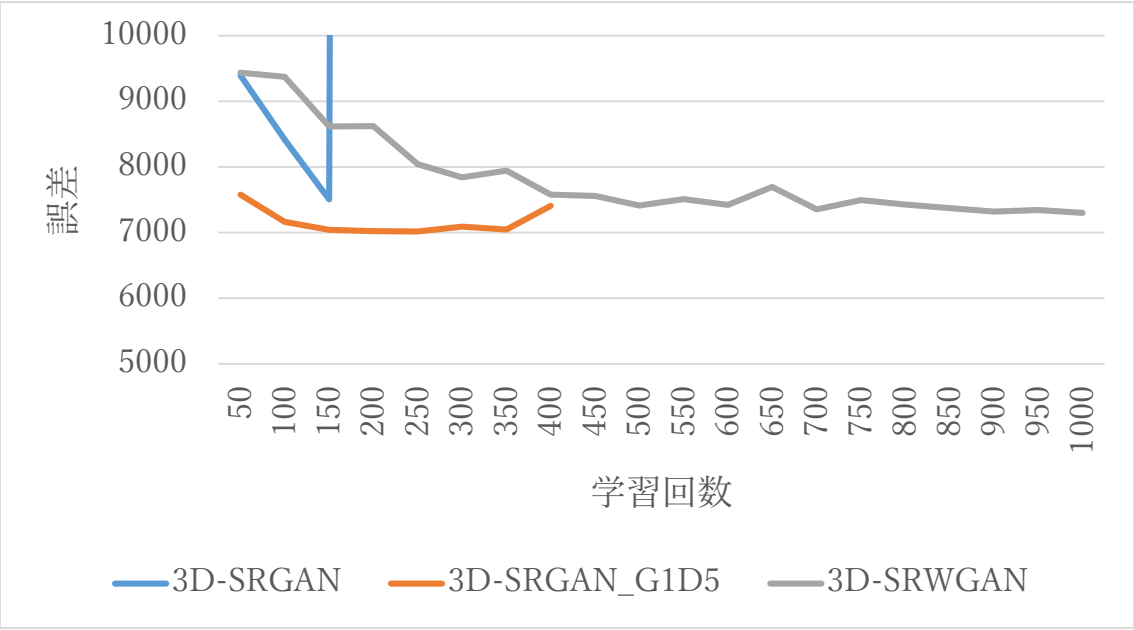


図 17. 3 つの手法の学習回数毎の誤差

4.3. 別クラスの超解像による精度評価

4.3.1. 実験方法

4.2 節の実験では、3D-SRGAN_G1D5 と 3D-SRWGAN では学習を安定化できたが、生成精度は 3D-SRGAN_G1D5 の方が高かった。一般にこの傾向が見られるのか確認するために、3D-SRGAN と 3D-SRGAN_G1D5、3D-SRWGAN で、学習データと異なるクラスの 3 次元モデルの超解像を行い、精度評価を行った。

3 D-SRGAN と 3D-SRGAN_G1D5、3 D-SRWGAN の生成モデルは chair の学習データを学習したものを用いた。超解像を行った 3 次元モデルは、データセット ModelNet10 の chair 以外の 9 クラスの低解像度 3 次元モデルである。生成モデルとしては表 3 と同じものを用いた。

4.3.2. 結果・考察

それぞれのクラスの生成結果を図 18、図 19 に示した。また、表 4、図 20 に超解像 3 次元モデルと高解像度 3 次元モデルの誤差の平均を示す。

図 18、19 を見てみると、dresser や sofa では 3D-SRGAN_G1D5 と 3D-SRWGAN が 3D-SRGAN と比べて適切な超解像を行っていた。しかし、table では、3D-SRGAN が最もよく超解像を行えていた。

表 4 の結果を見ても table の誤差は 3D-SRGAN が最も少なくなっている。全体としてはクラス毎に誤差の最も少ない手法は異なっていた。またそれぞれのクラスの誤差はいずれの手法でも近い値になっていた。

このことから、生成精度は 3D-SRGAN_G1D5 が最もよいわけではないことが分かった。

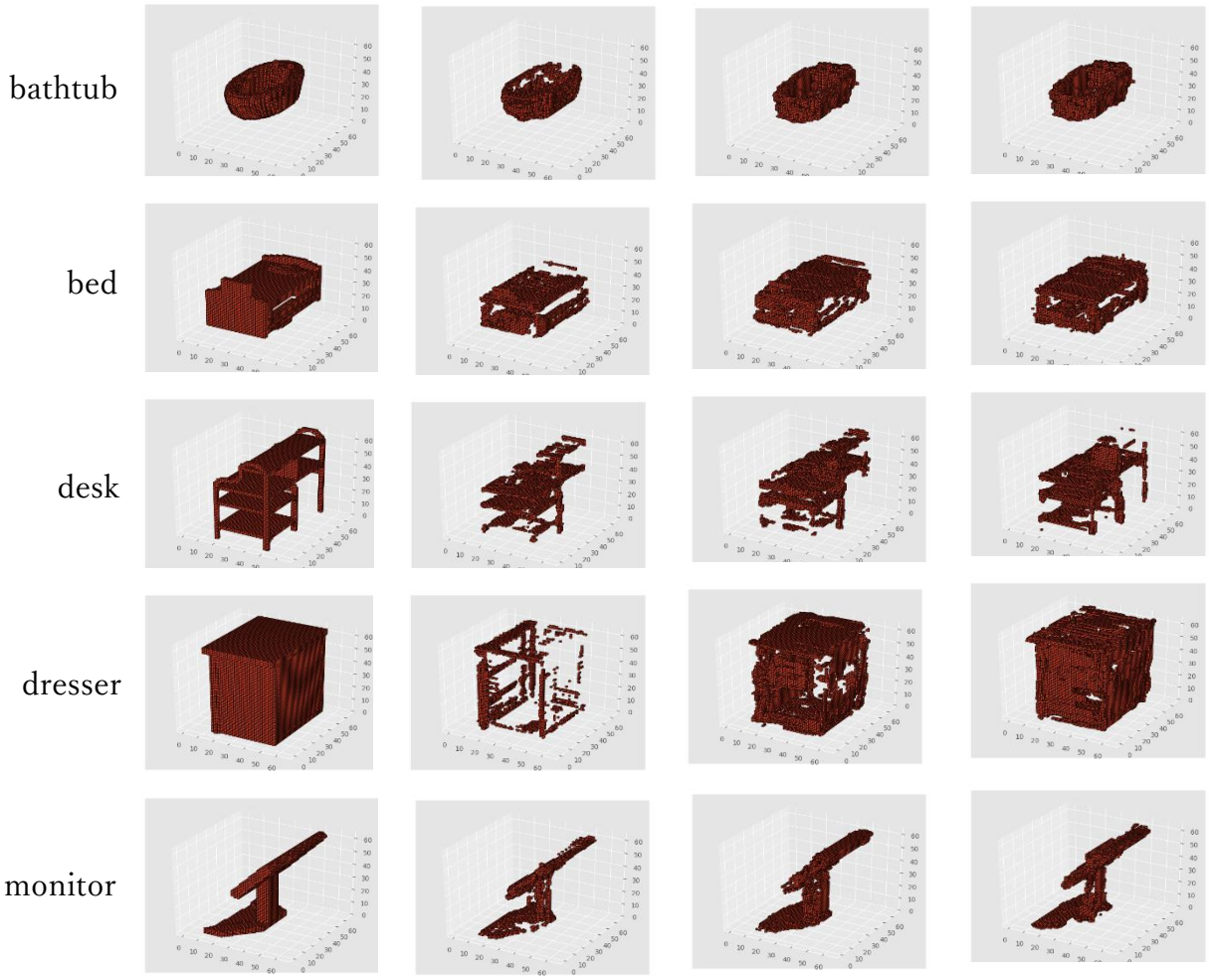


図 18. 別クラスの超解像(1)

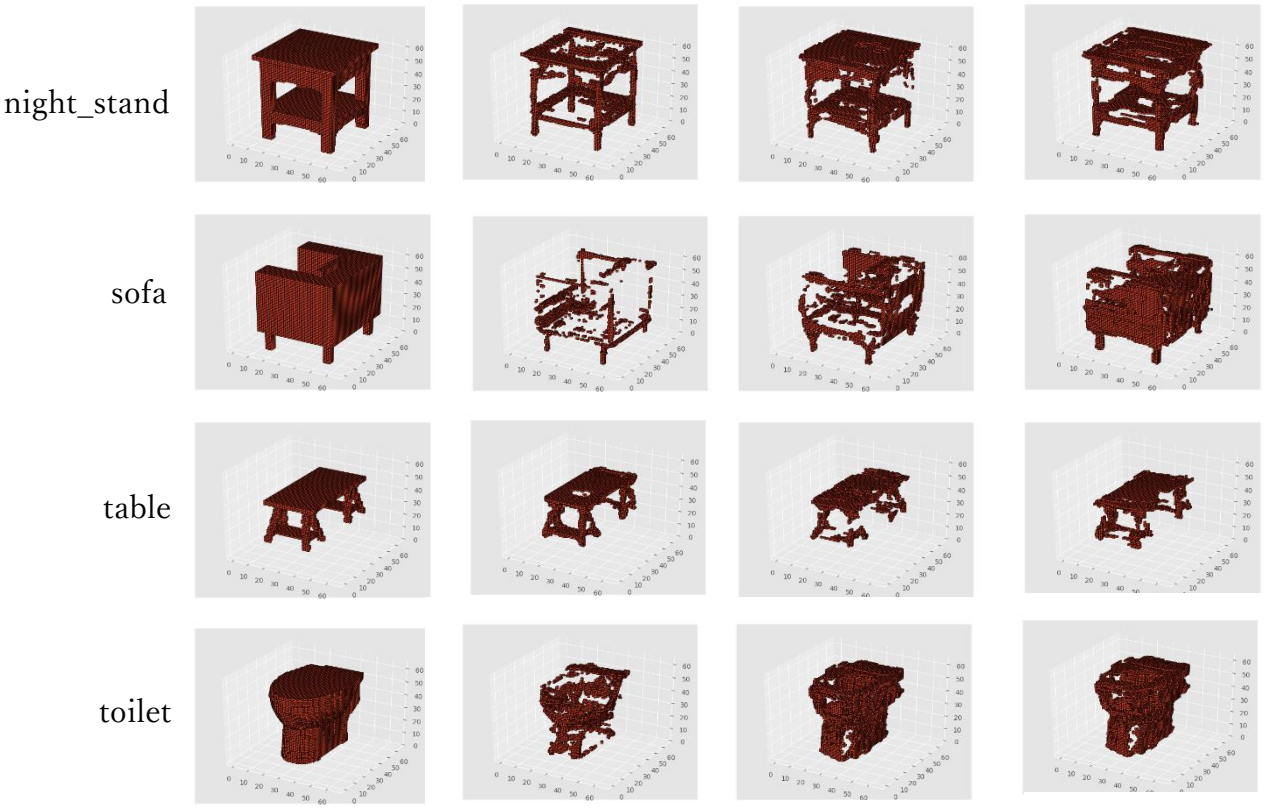


図 19. 別クラスの超解像(2)

表 4. 各クラスの超解像 3 次元モデルと高解像度 3 次元モデルの誤差の平均

クラス名	bathtub	bed	desk	dresser	monitor
3D-SRGAN	10323	15442	11972	22827	12313
3D-SRGAN_G1D5	10162	15291	12294	21377	12350
3D-SRWGAN	10630	15831	12715	23400	13276

クラス名	night_stand	sofa	table	toilet
3D-SRGAN	24907	12550	7092	16281
3D-SRGAN_G1D5	24403	12066	7494	15745
3D-SRWGAN	25285	12573	7237	16630

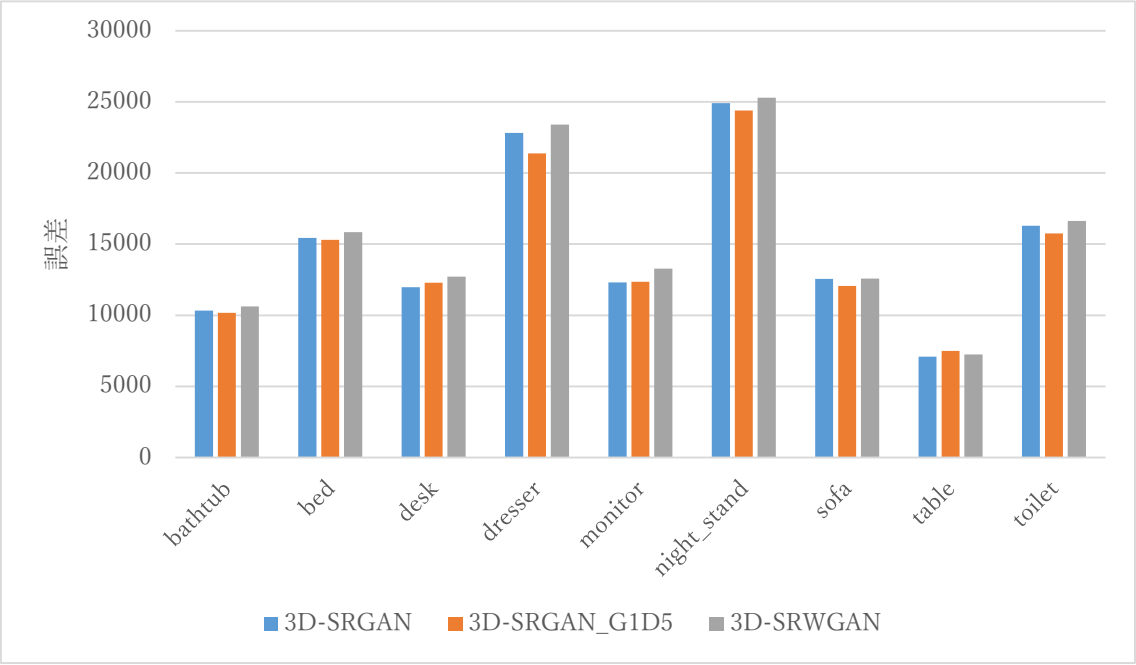


図 20. 3 つの手法の超解像 3 次元モデルと高解像度 3 次元モデルの誤差

5. おわりに

本論文では、3 次元モデルの超解像を行う 3D-SRGAN の学習を安定化させる方法の検討を行った。まず、3D-SRGAN の ResNet 層の Residual Block を変えて学習を行い、学習の安定性への影響を調査した。しかし、学習の不安定さは解消できず、生成精度もよくならなかった。この結果から、層数は 3D-SRGAN の学習安定化にそれほど影響しないことが示唆された。次に、3D-SRGAN に WGAN-GP で行われていたハイパーパラメータの変更を加えた 3D-SRGAN_G1D5、3D-SRGAN に WGAN-GP で行われていたハイパーパラメータの変更と目的関数の変更の両方を加えた 3D-SRWGAN、従来の 3D-SRGAN の 3 つの生成モデルで、学習の安定化の比較を行った。その結果、3D-SRGAN_G1D5 は学習回数 400 回の時点では Loss が大きくなることはなく収束しており、3 次元モデルの超解像も行えていた。3D-SRWGAN は 1200 回の時点で Loss が大きくなることなく収束しており、3 次元モデルの超解像を行えていた。3D-SRGAN_G1D5 と 3D-SRWGAN はともに学習を安定させることができた。chair を超解像した生成精度は、3D-SRGAN_G1D5 が最もよい結果となっていた。このことから学習の安定化にはハイパーパラメータの変更が有効であることが分かった。また、これらの 3 つの生成モデルで chair 以外のクラスの超解像を行った結果は、誤差はいずれの生成モデルでも同程度であった。3D-SRGAN が最も誤差の少ないクラスもあり、3D-SRGAN_G1D5 の生成精度が最もよいわけではなかった。

今後の課題として、ハイパーパラメータの細かい設定の見直しが挙げられる。本研究で行ったハイパーパラメータの変更では、WGAN-GP で用いられていた値をそのまま用いた。Generator や Discriminator の学習率や更新頻度を最適化することで安定性を維持したまま Loss が収束するまでの学習回数を縮められると考えられる。また、目的関数の変更による影響や安定化と精度の関係について

より詳しく検討する必要がある。

謝辞

本研究を行うにあたり、多くの方々に協力をしていただきました。ご指導いただいた椋木雅之教授に感謝いたします。指導教員である椋木雅之教授には研究や論文作成に関して様々なご指導を頂きました。本当にありがとうございました。

山森教授、伊達准教授には、副査を務めていただきました。お忙しい中、貴重な時間を割いて下さいましてありがとうございます。

椋木研究室の皆様には、研究を進めるにあたって、様々な助言を頂きました。皆さんのおかげで楽しく、充実した研究生活を過ごすことができました。大変感謝しています。

参考文献

[1] 安藤繁, 土井康弘, “超解像”, 計測と制御, vol. 22, no. 10, p. 828-836 (1983)

[2] 岡和寿, ”SRGAN の 3 次元モデル超解像への拡張”, 平成 30 年度 宮崎大学大学院 工学研究科 工学専攻 機械・情報系コース 情報システム工学分野修士論文(2019)

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets”, Advances in Neural Information Processing Systems (NIPS), pp. 2672–2680 (2014)

[4] Alec Radford, Luke Metz, Soumith Chintala, “UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS”, arXiv:1511.06434(2015)

[5] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, “Improved Training of Wasserstein GANs”, arXiv:1704.00028(2017)

[6] 池谷彰彦, 広明敏彦, “超解像ソリューション (組込みソフトウエア・ソリューションヨン特集)--(イメージ/音声処理コンポーネントソリューション)”, NEC 技報, vol. 60, no. 2, pp. 24-26 (2007)

[7] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 105-114 (2017)

[8] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, “Image Style Transfer Using Convolutional Neural Networks”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2414-2423 (2016)

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778 (2016)

[10] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, Zehan Wang, “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1874-1883 (2016)

[11] Kaiming He, Jian Sun, “Convolutional Neural Networks at Constrained Time Cost”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5353-5360 (2015)

[12] Rupesh Kumar Srivastava, Klaus Greff, Jürgen Schmidhuber, “Highway Networks”, arXiv:1505.00387 (2015)

[13] Odena, Augustus and Dumoulin, Vincent and Olah, Chris, “Deconvolution and Checkerboard Artifacts”, Distill (2016) http://doi.org/10.23915/distill.00003

[14] A. Radford, L. Metz, S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", International Conference on Learning Representations (ICLR) (2016)

[15] ModelNet10, http://modelnet.cs.princeton.edu/