

# SRGANの 3次元モデル超解像への拡張

---

工学研究科 情報システム工学分野

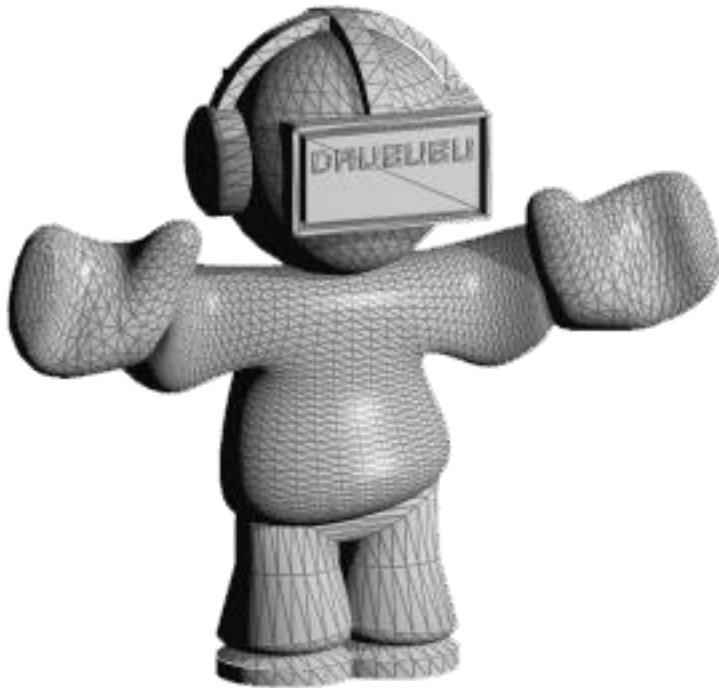
指導教員 椋木 雅之

T1703007 岡 和寿

2019/02/04

# 3次元モデリング

サーフェイス表現



ボクセル表現



引用:<https://raahii.github.io/2017/10/3dgan-chainer/>

# ボクセル表現の 3次元モデリング

## 3次元モデルを手軽に作成

積み木のように積み重ねて作成

例) Minecraft



引用:<https://chasgamers.com/minecraft-tofuarchitecture/>

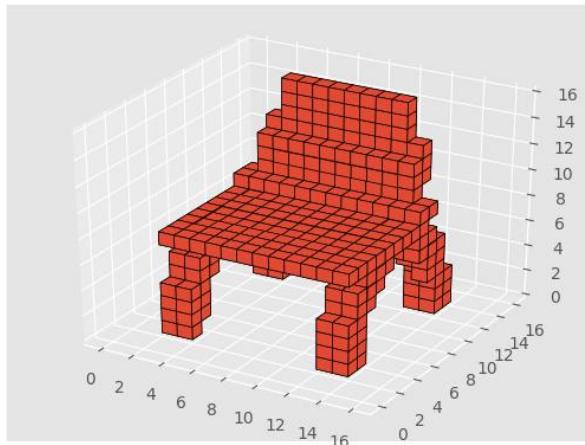
高精細な3次元モデルの  
作成はかなり手間がかかる



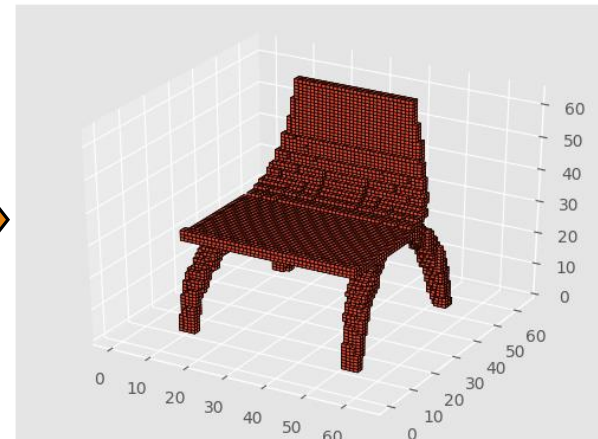
引用: <https://www.autodesk.co.jp/redshift/minecraft-architecture/>

# 研究の目的

## 3次元モデルの超解像



低解像度3次元モデル



高解像度3次元モデル

3次元モデル作成時の手間を削減！

# 超解像

---

低解像度データから高解像度データを生成する技術

画像分野の研究が盛ん

SRGAN

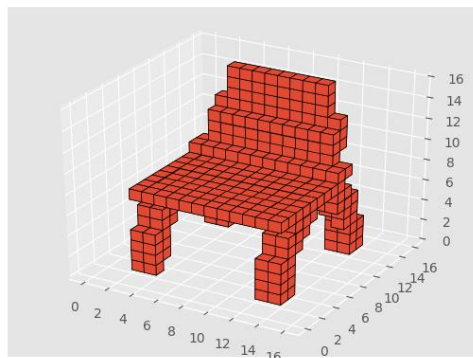
(Super Resolution Generative Adversarial Networks)[1]

3次元モデルの超解像は見当たらない

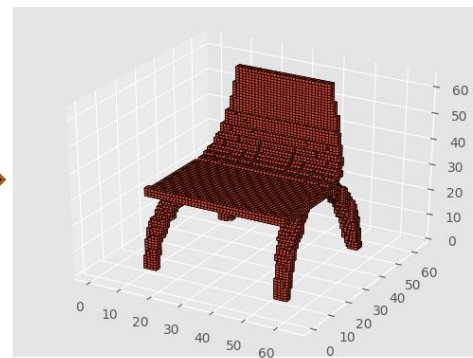
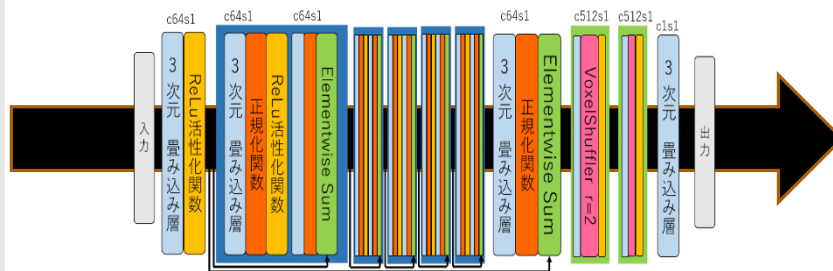
# 提案手法

## 3D-SRGAN

SRGANを3次元モデル超解像に拡張した手法



低解像度3次元モデル



高解像度3次元モデル

# SRGAN

---

- 深層学習を利用
- 低解像度画像から高解像度画像を生成する

## 予備実験

- カラー画像を学習
- 超解像 { カラー画像  
濃淡画像  
2値画像



# SRGANで超解像

- カラー画像の超解像



低解像度画像



SRGANで超解像した画像



# SRGANで超解像

- 濃淡画像の超解像



低解像度画像



SRGANで超解像した画像

# SRGANで超解像

---

- 2値画像の超解像



低解像度画像



SRGANで超解像した画像

# SRGANで超解像

---

- 2値画像の超解像



低解像度画像

SRGANで超解像した画像

# 2値画像を学習したSRGAN

- 2値画像の超解像



低解像度画像



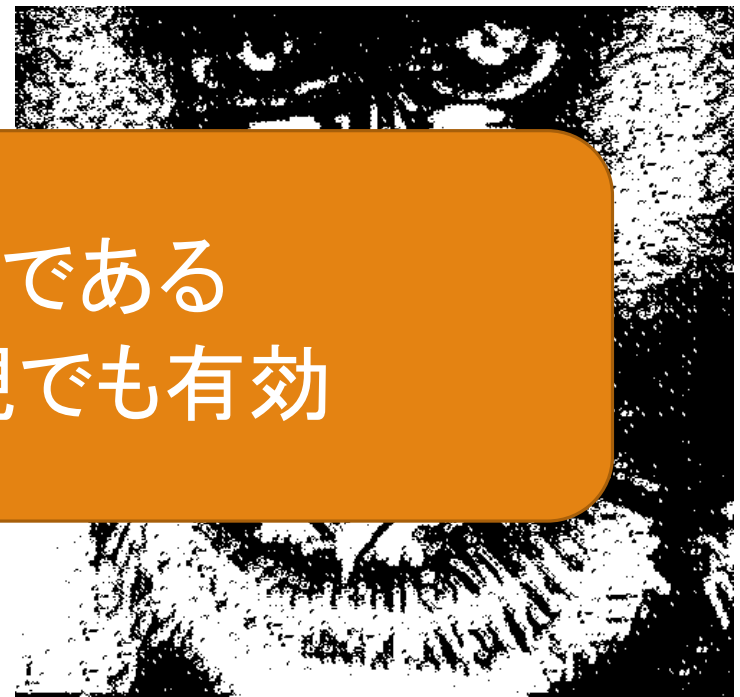
SRGANで超解像した画像

# 2値画像を学習したSRGAN

- 2値画像の超解像



低解像度画像

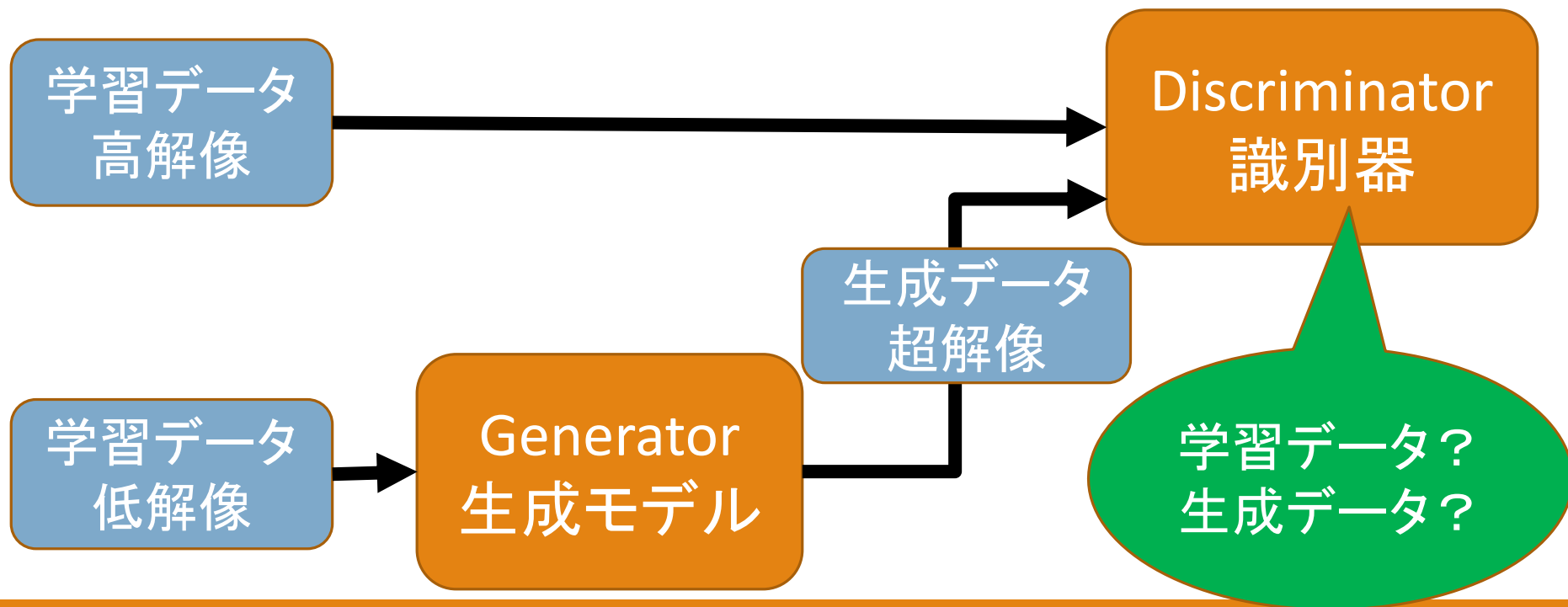


SRGANで超解像した画像

2値データである  
ボクセル表現でも有効

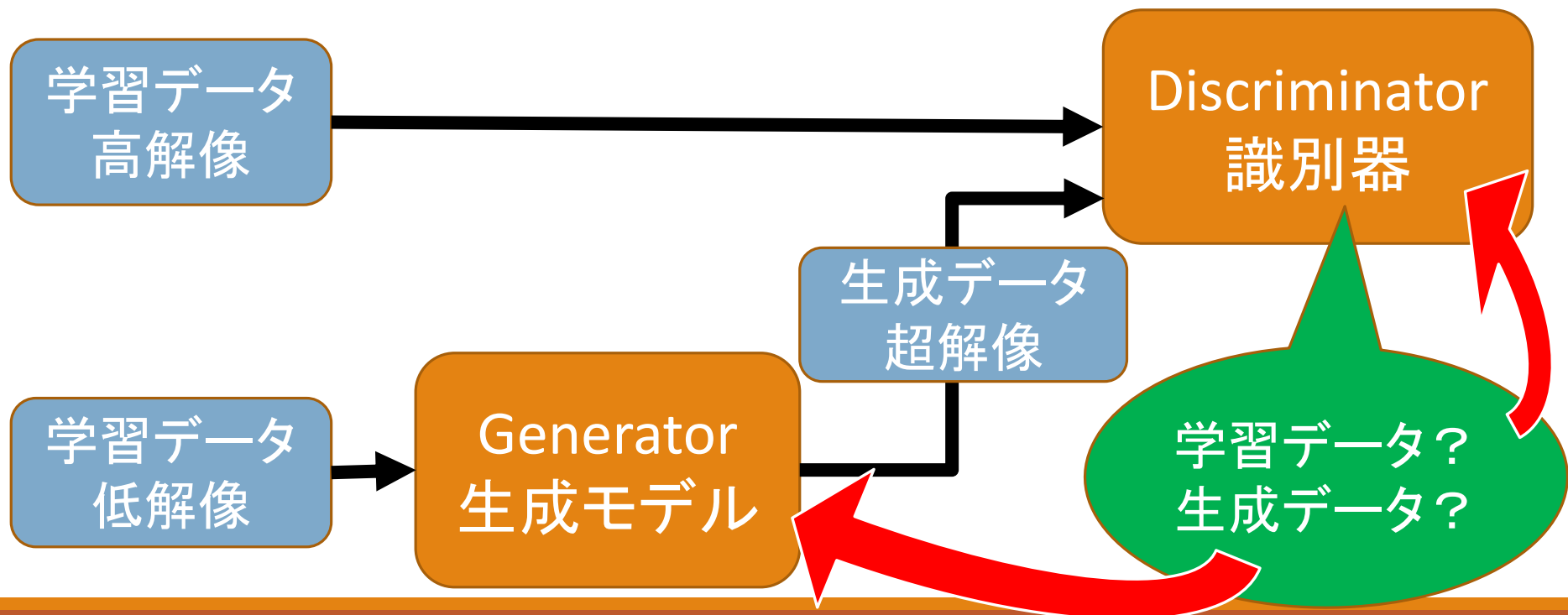
# 3D-SRGANの学習

Generatorが学習データに近い3次元モデルを生成できることが目的



# 3D-SRGANの学習

Generatorが学習データに近い3次元モデルを生成できることが目的





# 3D-SRGANの拡張点

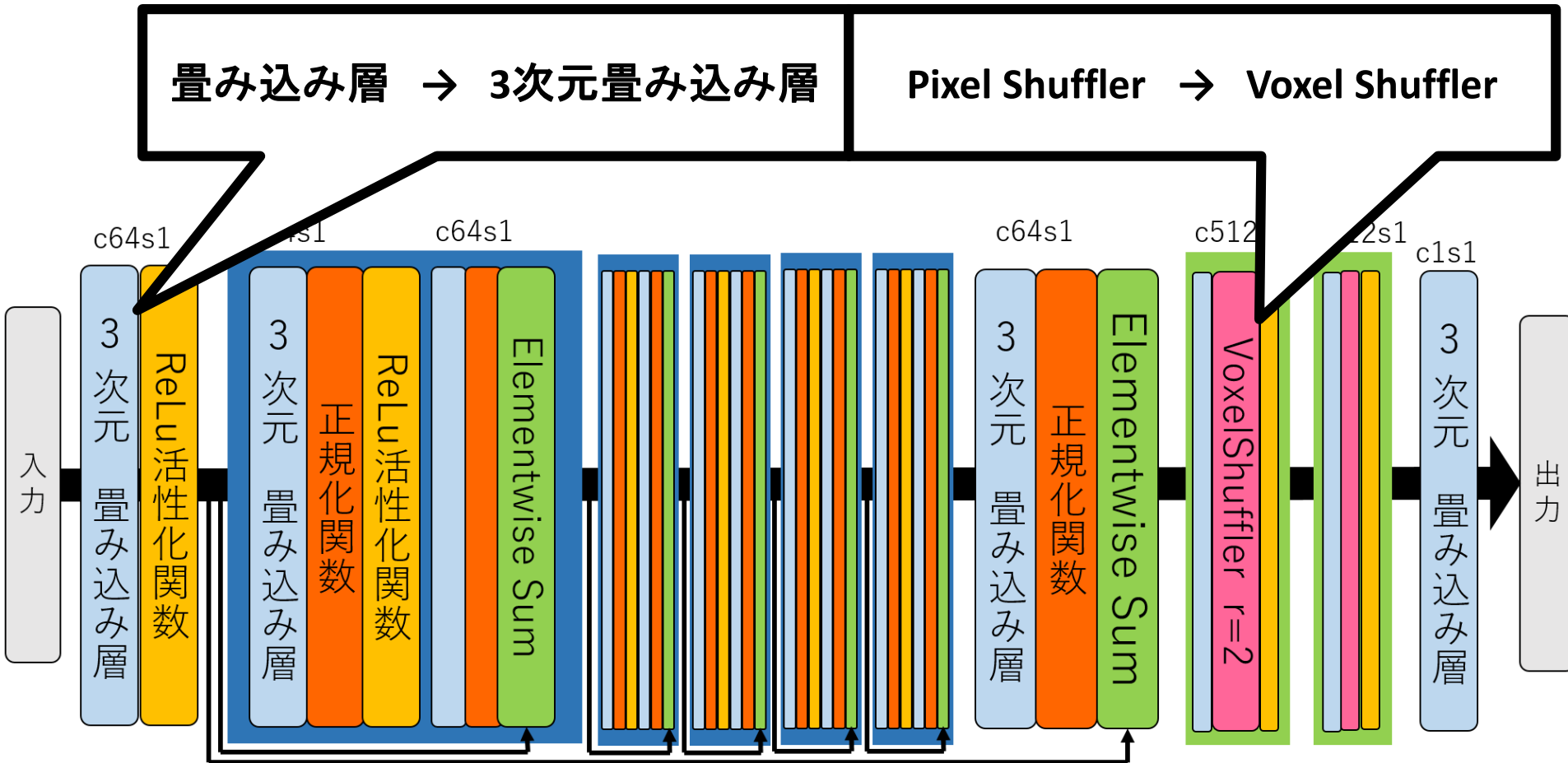
---

1. 畳み込み層 → 3次元畳み込み層
2. Pixel Shuffler → Voxel Shuffler

# Generator (生成モデル)

畳み込み層 → 3次元畳み込み層

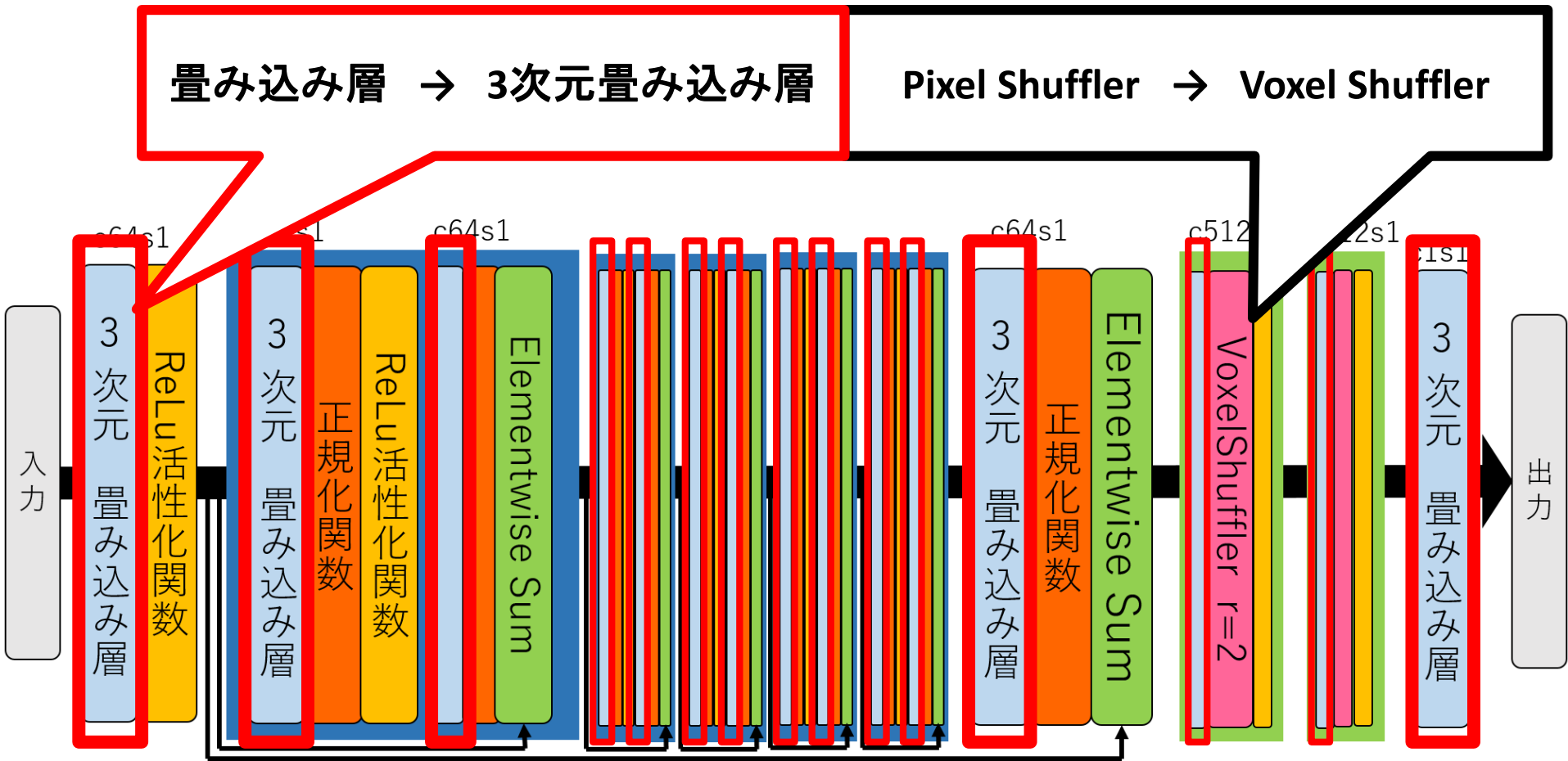
Pixel Shuffler → Voxel Shuffler



# Generator (生成モデル)

畳み込み層 → 3次元畳み込み層

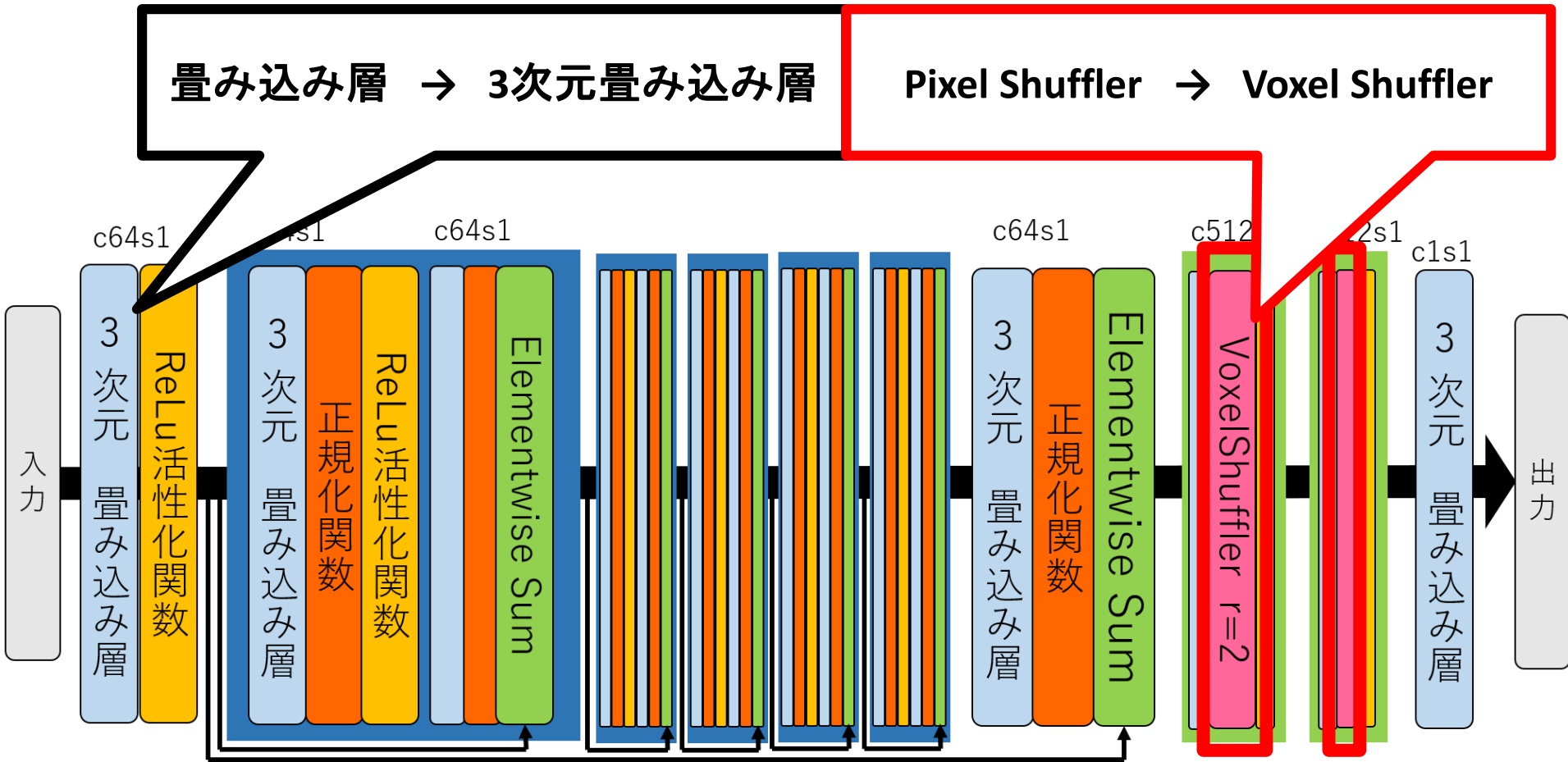
Pixel Shuffler → Voxel Shuffler



# Generator (生成モデル)

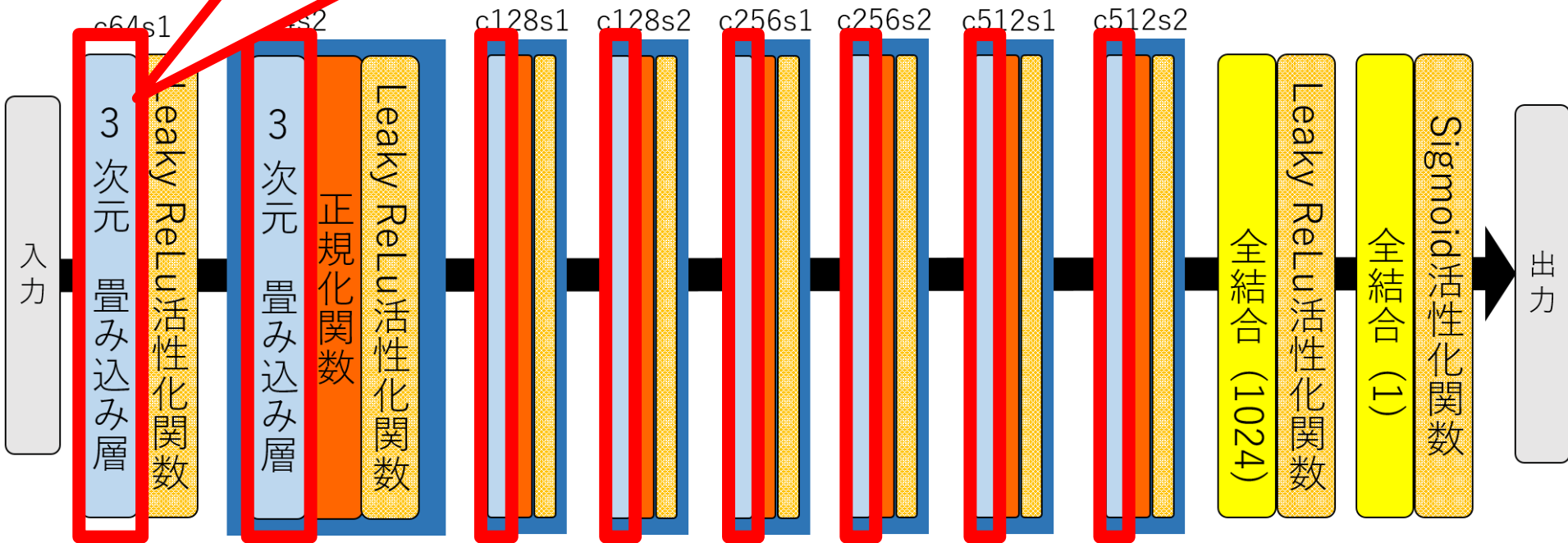
畳み込み層 → 3次元畳み込み層

Pixel Shuffler → Voxel Shuffler



# Discriminator (識別器)

畳み込み層 → 3次元畳み込み層



# 畳み込み層 → 3次元畳み込み層

畳み込み処理を行う

入力データとフィルターを積和演算

0	1	2	3
1	2	3	0
2	3	2	1
3	2	1	0

入力データ

×

0	1	2
1	2	0
2	0	1

フィルター

→

14	23
23	8

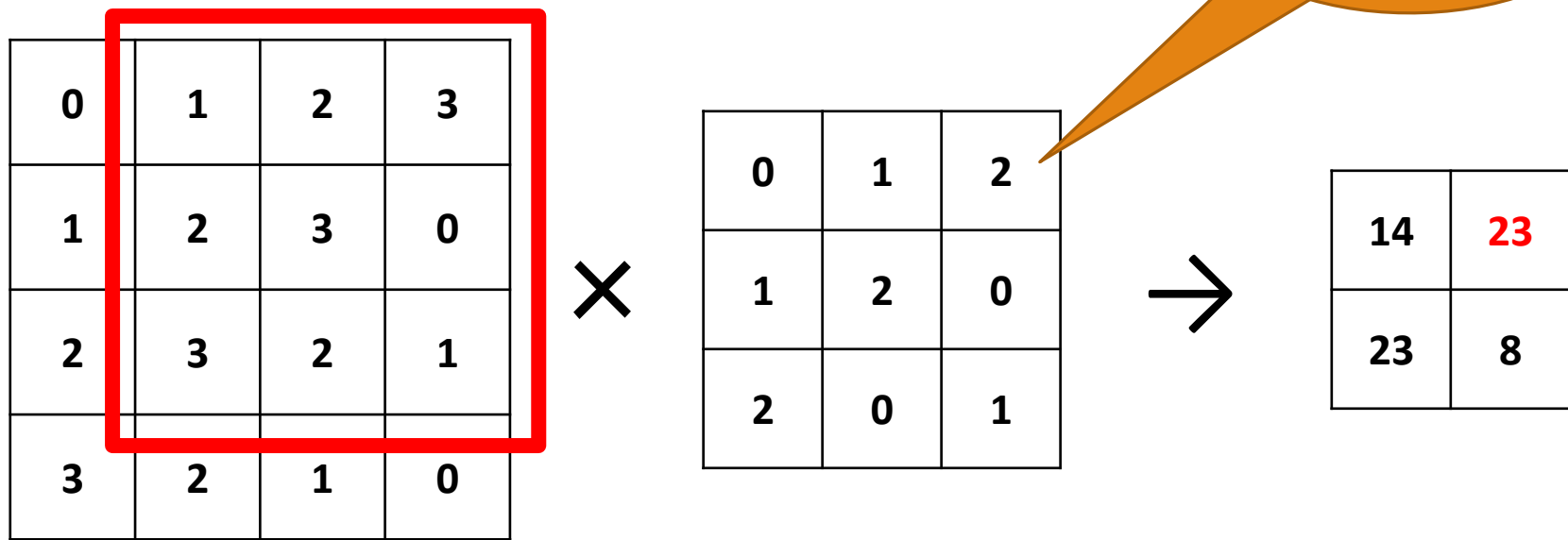
出力データ

フィルターの重みを学習

# 畳み込み層 → 3次元畳み込み層

畳み込み処理を行う

入力データとフィルターを積和演算



入力データ

フィルター

出力データ



# 畳み込み層 → 3次元畳み込み層

畳み込み処理を行う

入力データとフィルターを積和演算

フィルターの重みを学習

0	1	2	3
1	2	3	0
2	3	2	1
3	2	1	0

入力データ

×

0	1	2
1	2	0
2	0	1

フィルター

→

14	23
23	8

出力データ

# 畳み込み層 → 3次元畳み込み層

畳み込み処理を行う

入力データとフィルターを積和演算

フィルターの重みを学習

0	1	2	3
1	2	3	0
2	3	2	1
3	2	1	0

入力データ

×

0	1	2
1	2	0
2	0	1

フィルター

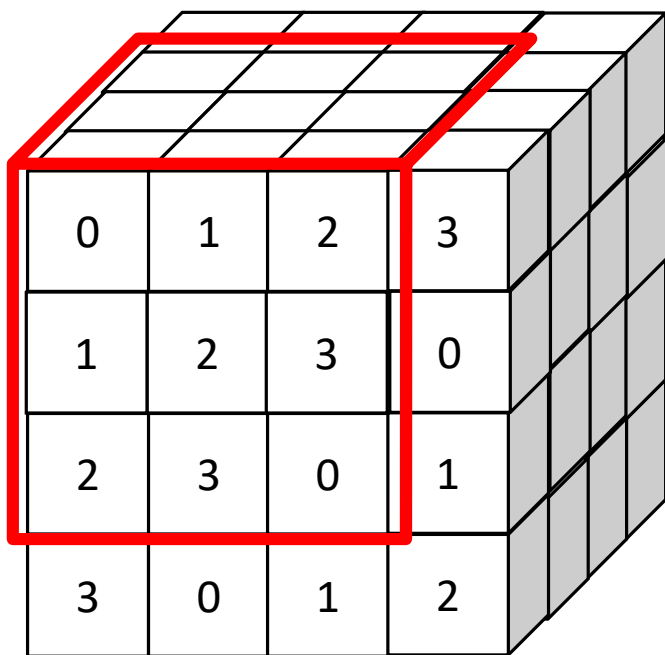
→

14	23
23	8

出力データ

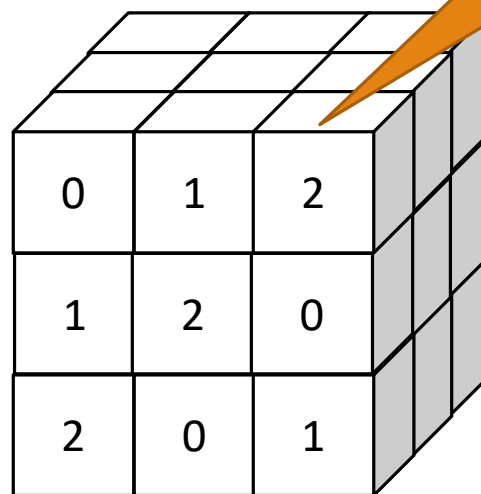
# 畳み込み層 → 3次元畳み込み層

フィルターのサイズに奥行を追加  
入力データとフィルターを積和演算



入力データ

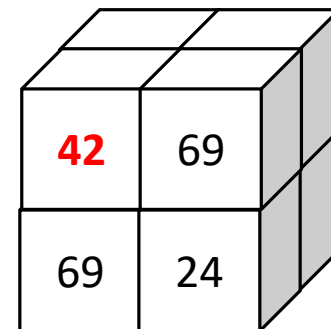
×



フィルター

フィルターの重みを学習

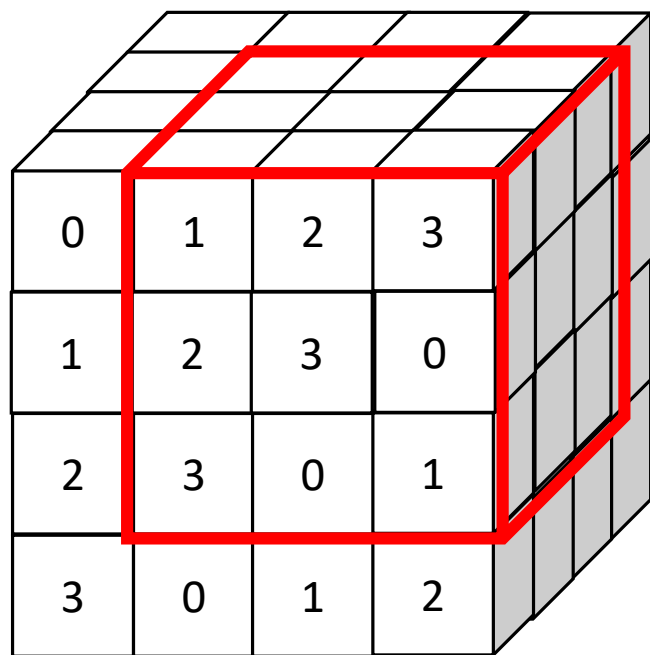
→



出力データ

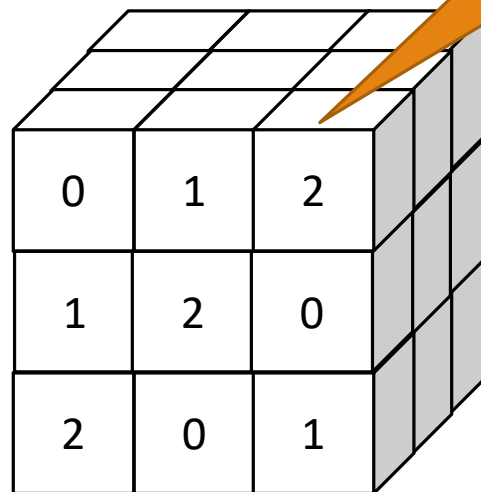
# 畳み込み層 → 3次元畳み込み層

フィルターのサイズに奥行を追加  
入力データとフィルターを積和演算



入力データ

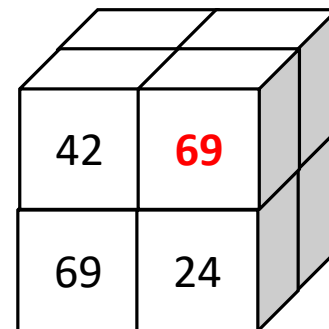
×



フィルター

フィルターの重みを学習

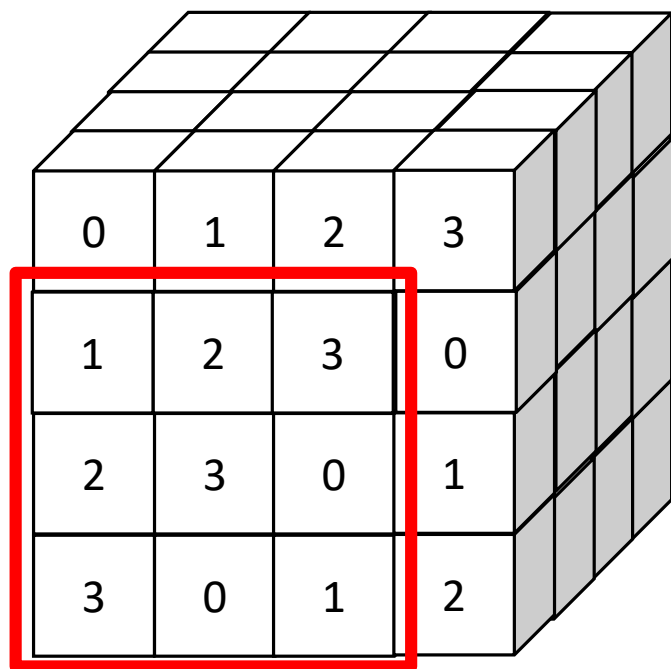
→



出力データ

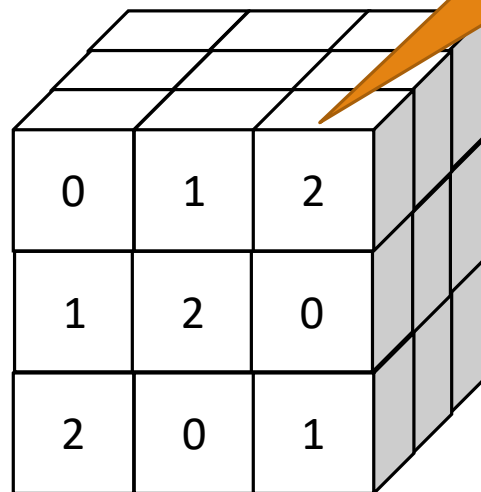
# 畳み込み層 → 3次元畳み込み層

フィルターのサイズに奥行を追加  
入力データとフィルターを積和演算



入力データ

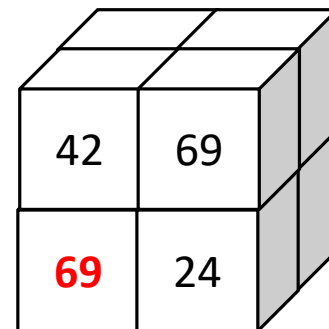
×



フィルター

フィルターの重みを学習

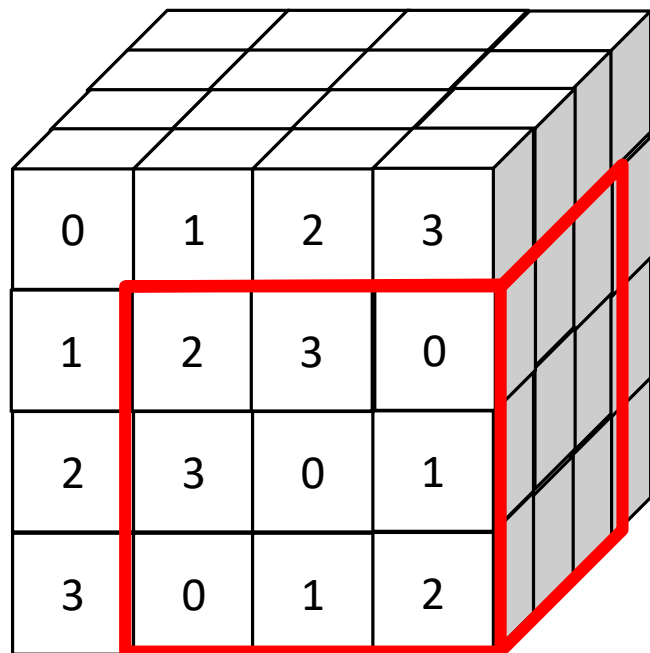
→



出力データ

# 畳み込み層 → 3次元畳み込み層

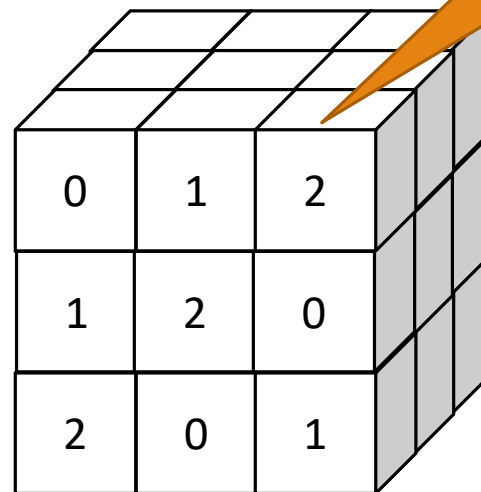
フィルターのサイズに奥行を追加  
入力データとフィルターを積和演算



0	1	2	3
1	2	3	0
2	3	0	1
3	0	1	2

入力データ

×

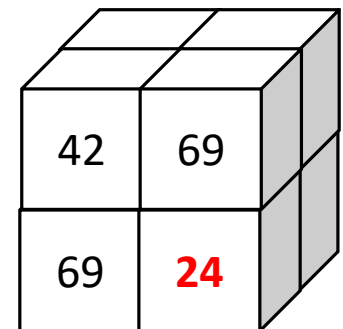


0	1	2
1	2	0
2	0	1

フィルター

フィルターの  
重みを学習

→

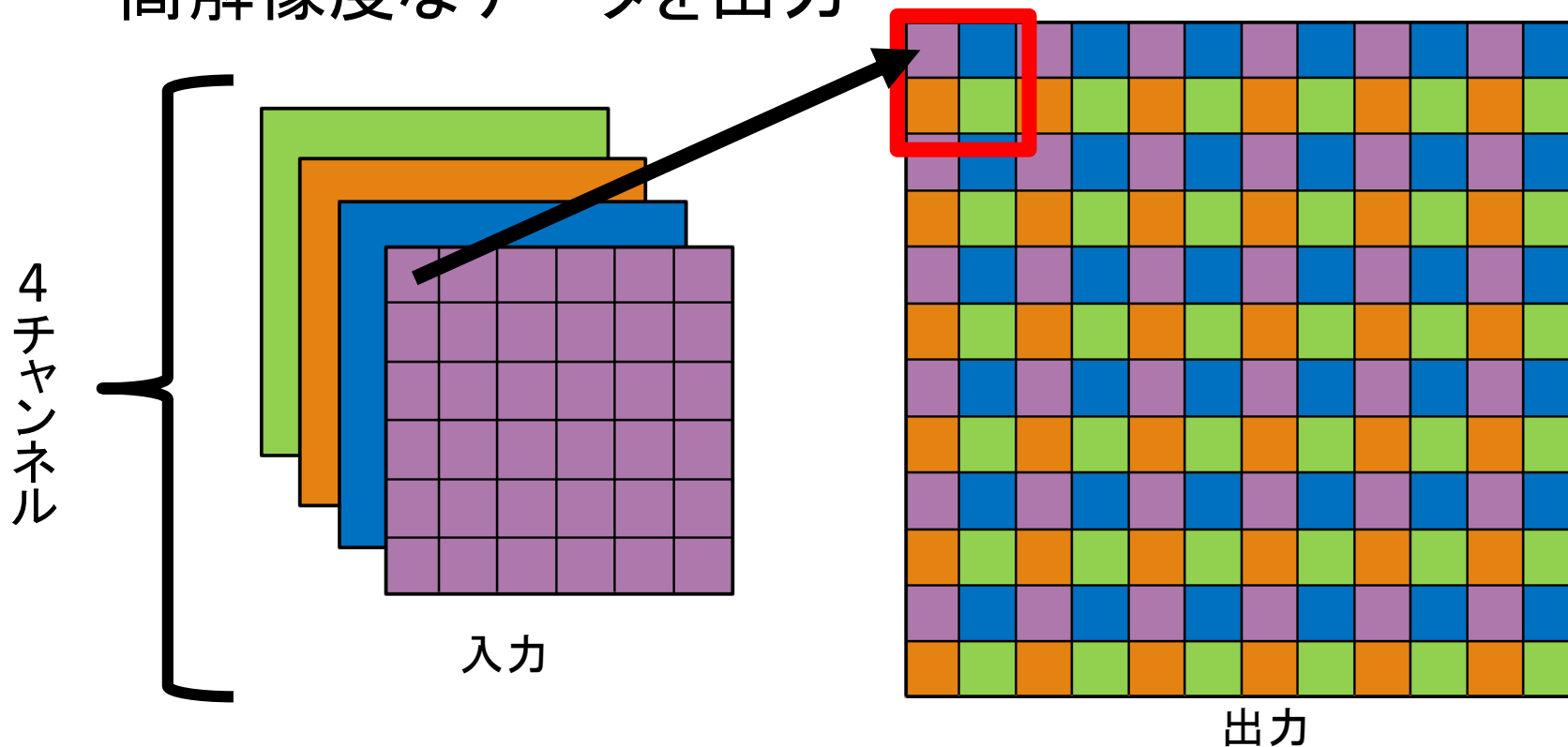


42	69
69	24

出力データ

# Pixel Shuffler[2]

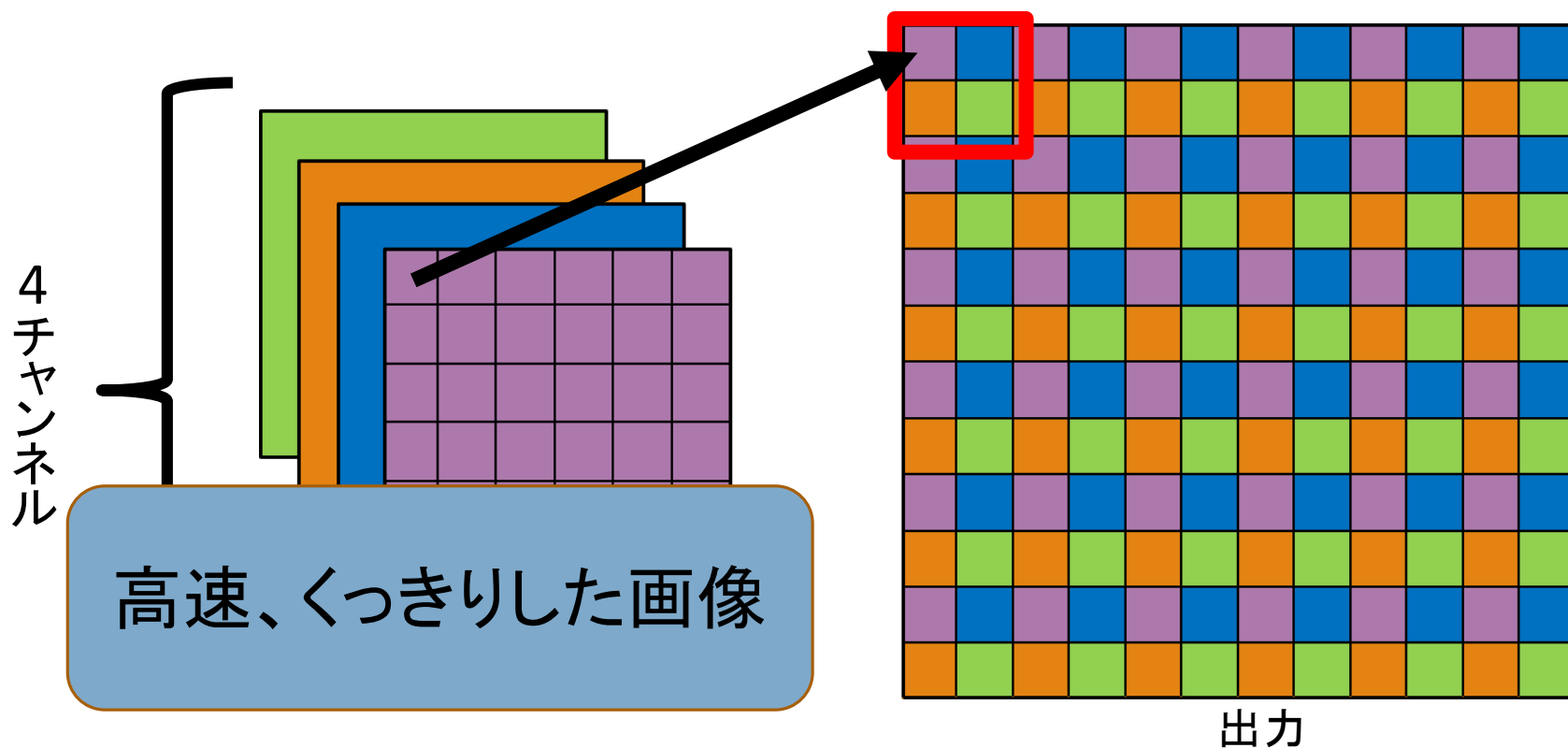
入力データの各ピクセルを並べ替えて  
高解像度なデータを出力





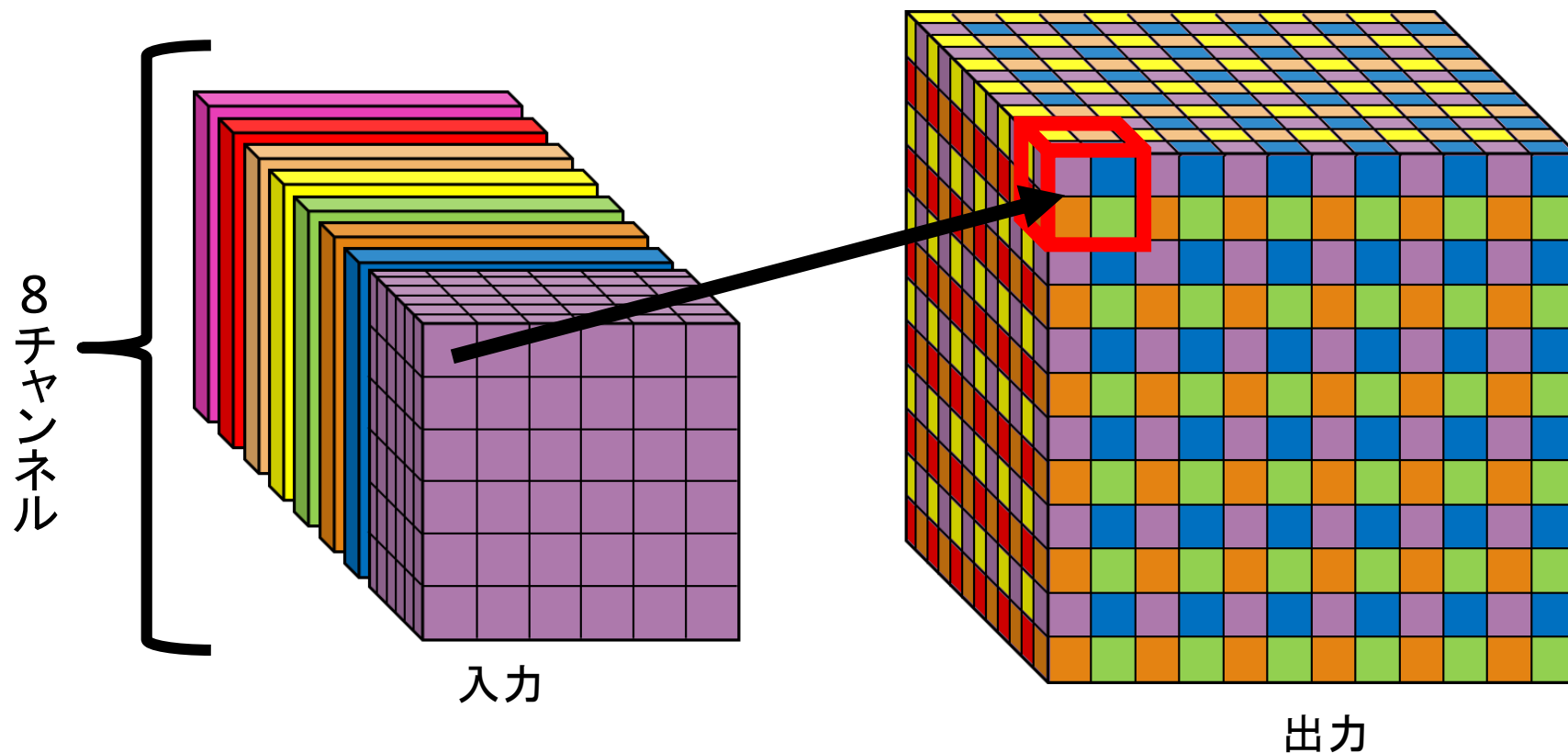
# Pixel Shuffler[2]

$$\mathcal{PS}(T)_{x,y,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, C_{out} \cdot r \cdot \text{mod}(y,r) + C_{out} \cdot \text{mod}(x,r) + c}$$



# Pixel Shuffler $\rightarrow$ Voxel Shuffler

$$\mathcal{VS}(T)_{x,y,z,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, \lfloor z/r \rfloor, C_{out} \cdot r^2 \cdot \text{mod}(z,r) + C_{out} \cdot r \cdot \text{mod}(y,r) + C_{out} \cdot \text{mod}(x,r) + c}$$



# 実験

---

# 実験内容

---

1. 3D-SRGANと他手法との比較
2. 学習データとは異なるクラスの超解像



# 実験1

## 他手法との比較

---

- 学習

chairの学習データ  $\left\{ \begin{array}{l} \text{高解像度3次元モデル} \\ \text{低解像度3次元モデル} \end{array} \right.$

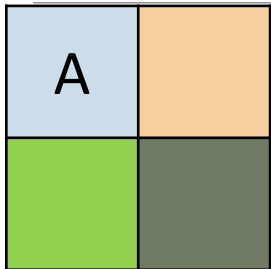
- 超解像する3次元モデル

chairのテストデータ 低解像度3次元モデル

- 比較手法: 最近傍補間、線形補間

- 正解データ: chairのテストデータ 高解像度3次元モデル

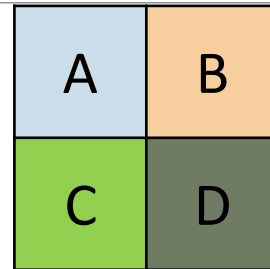
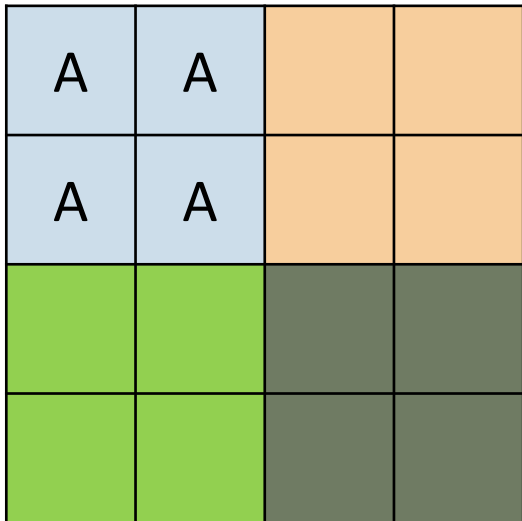
# 比較手法



画像をそのまま拡大



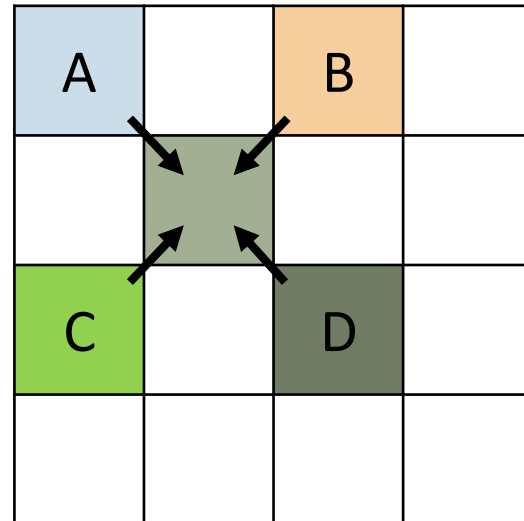
最近傍補間



周囲の画素の関係から内分比

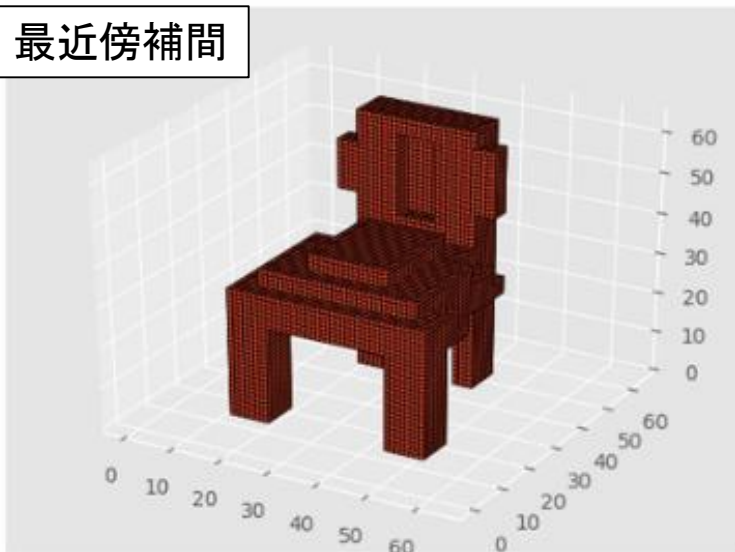


線形補間

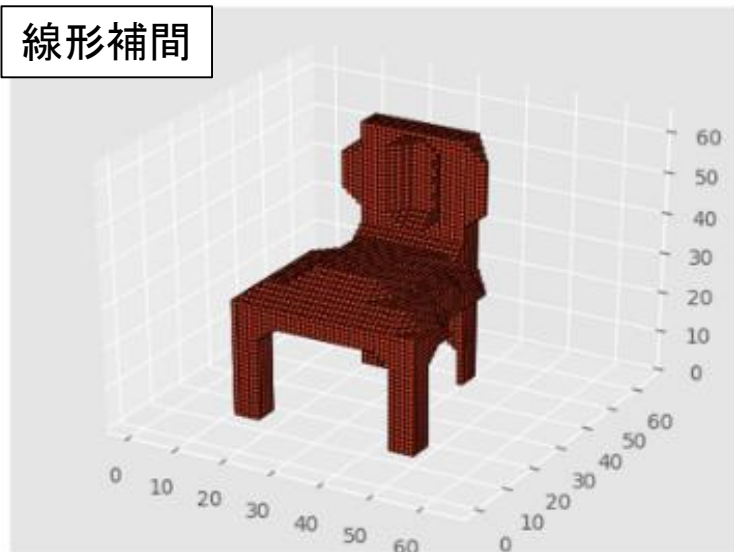


# 実験1の結果

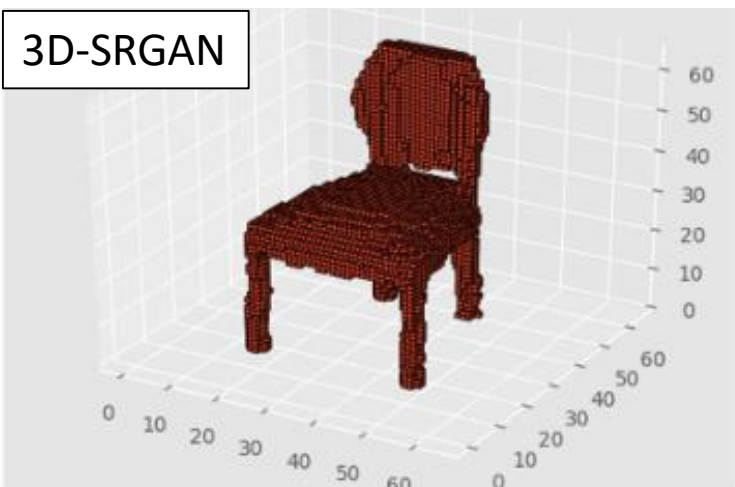
最近傍補間



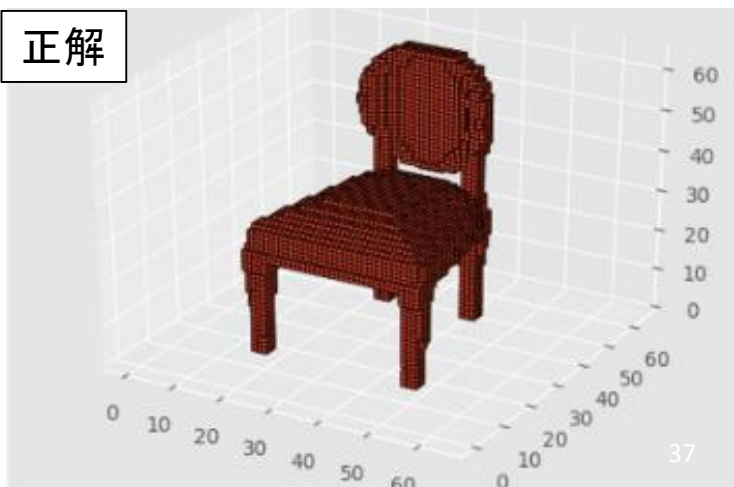
線形補間



3D-SRGAN



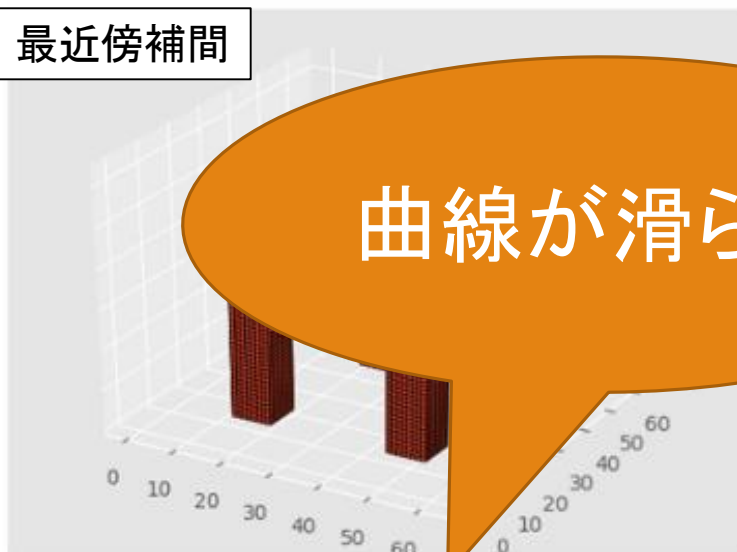
正解



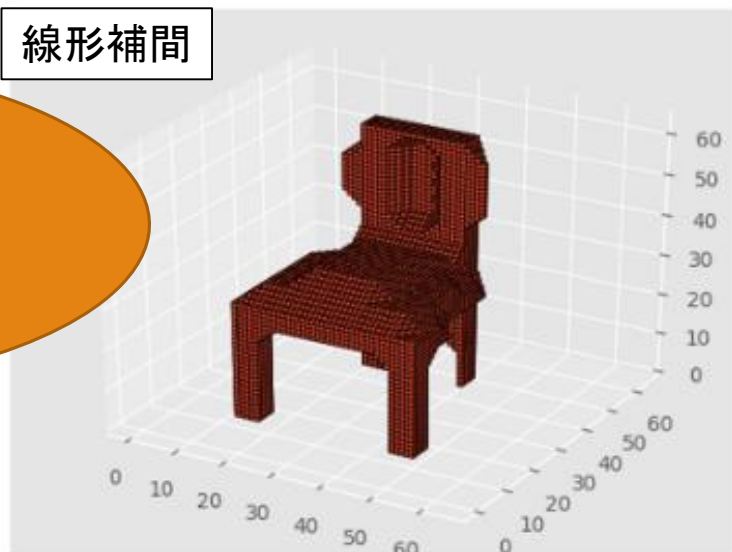


# 実験1の結果

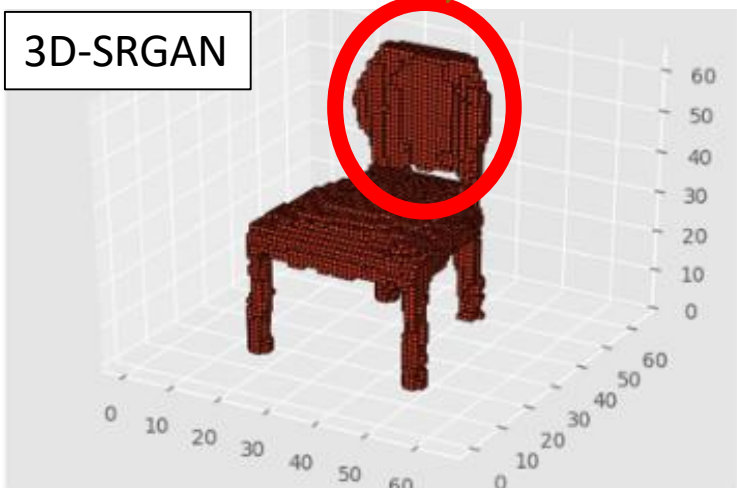
最近傍補間



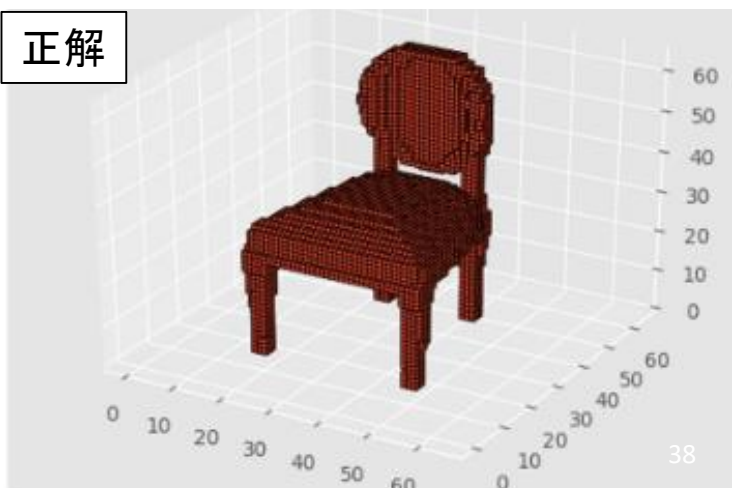
線形補間



3D-SRGAN

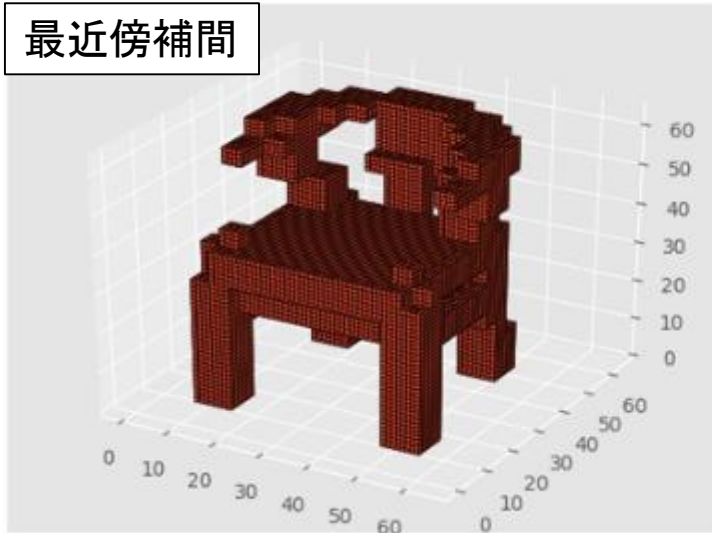


正解

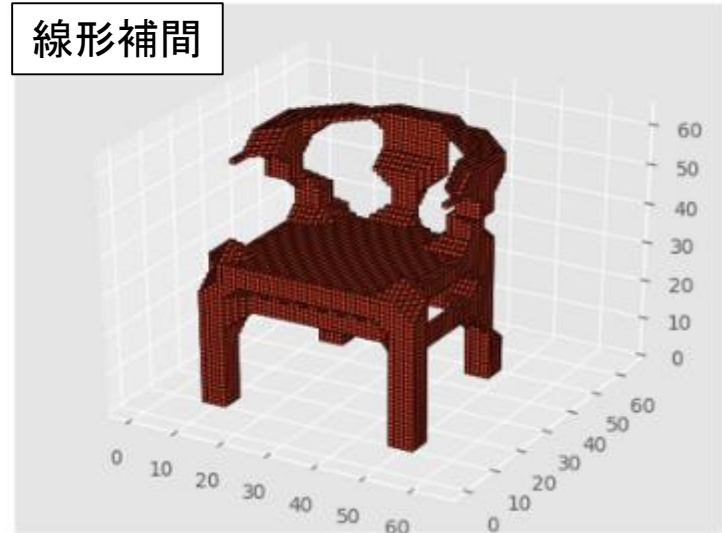


# 実験1の結果

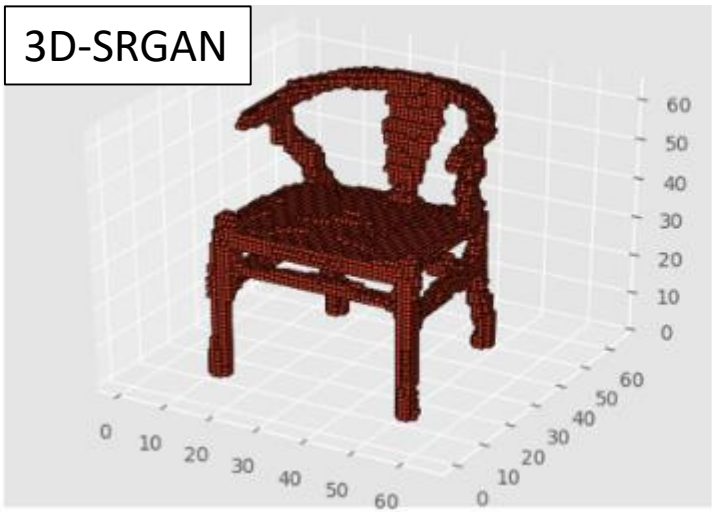
最近傍補間



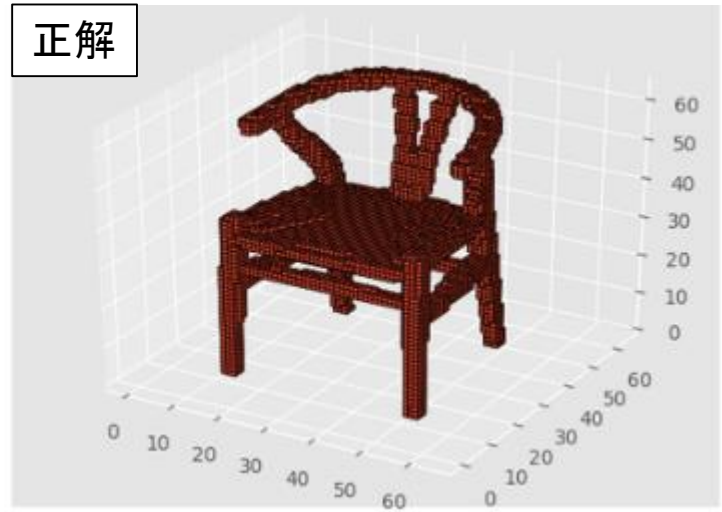
線形補間



3D-SRGAN

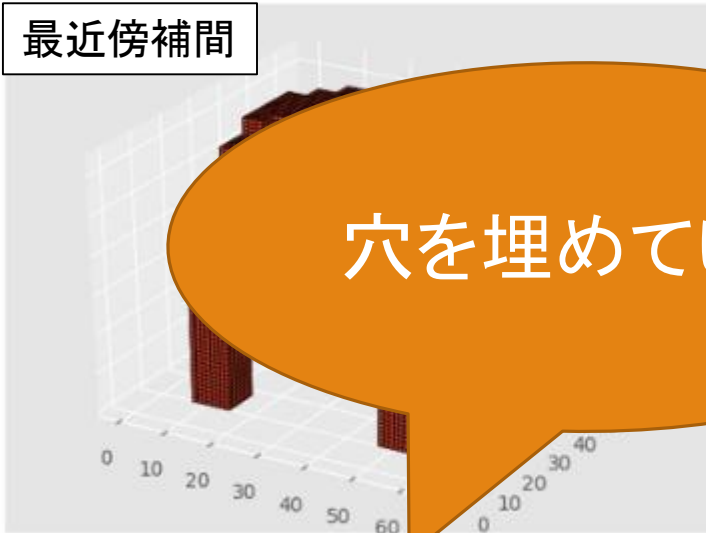


正解

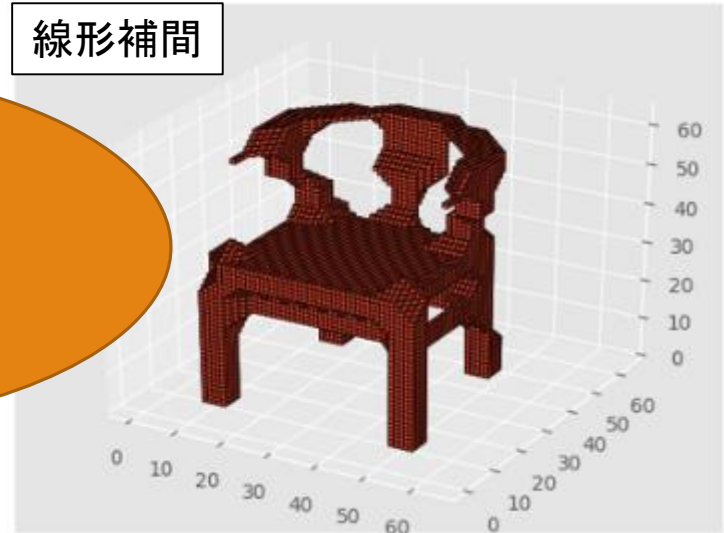


# 実験1の結果

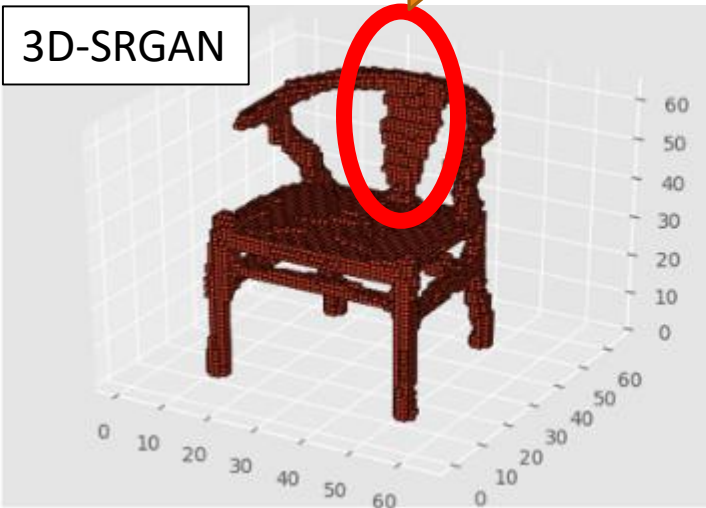
最近傍補間



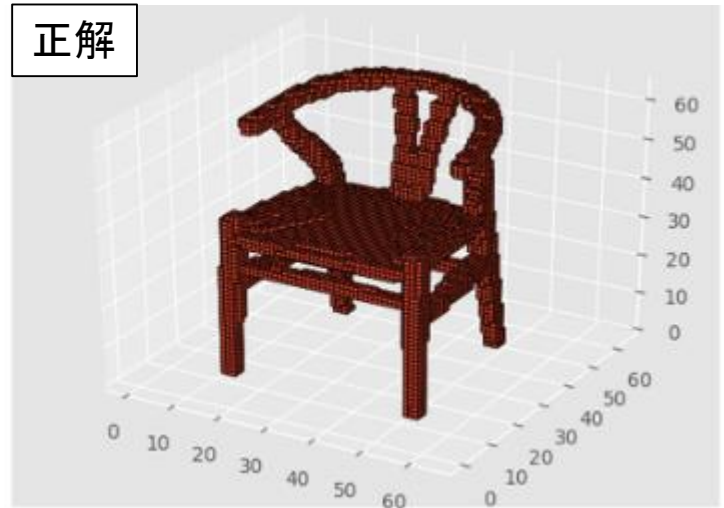
線形補間



3D-SRGAN



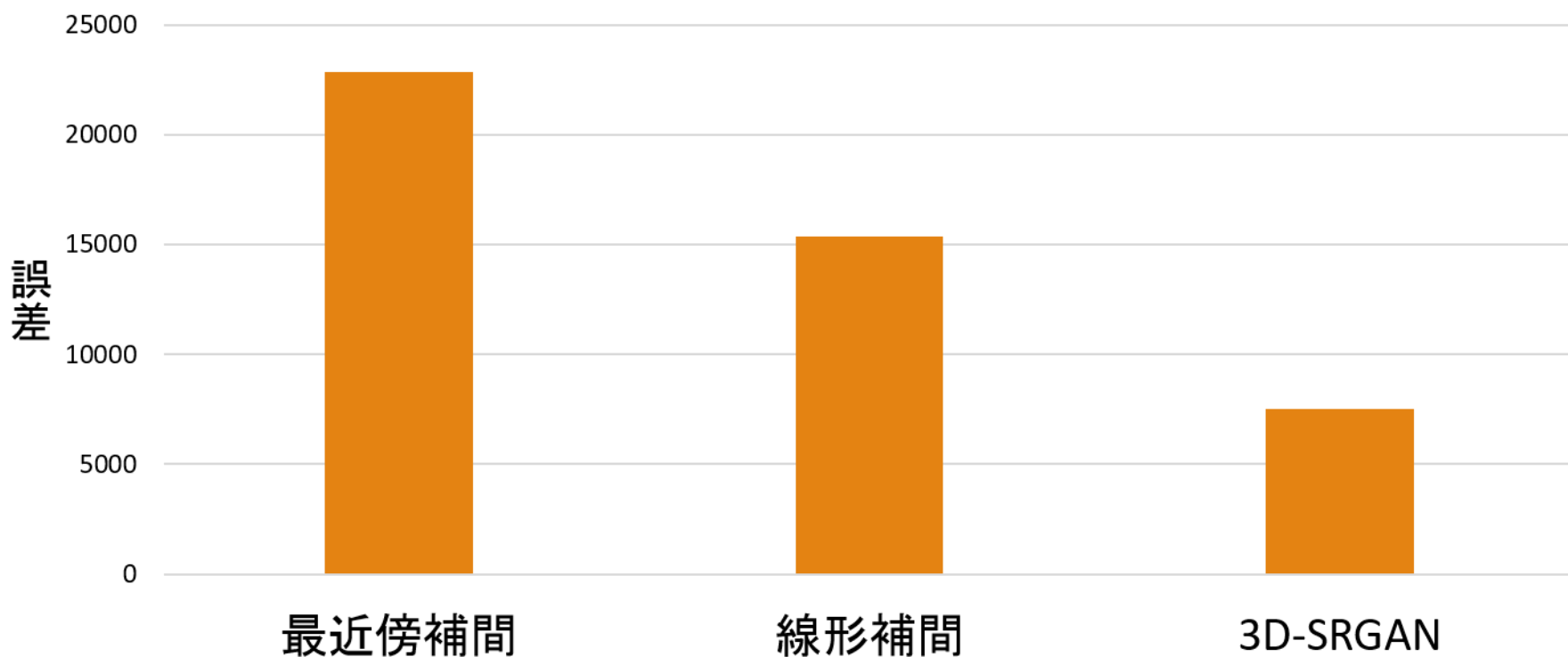
正解



# 実験1の結果（誤差）

生成結果と正解の同じ座標間でのボクセルの有無の違い

100個のデータの誤差の平均



# 実験2

## 学習データとは異なるクラスの超解像

---

- 学習

chairの学習データ  $\left\{ \begin{array}{l} \text{高解像度3次元モデル} \\ \text{低解像度3次元モデル} \end{array} \right.$

- 超解像する3次元モデル

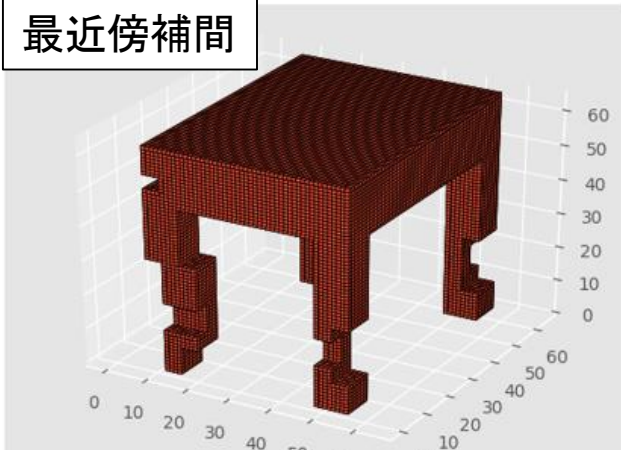
chair以外のテストデータ 低解像度3次元モデル

- 比較手法: 最近傍補間、線形補間

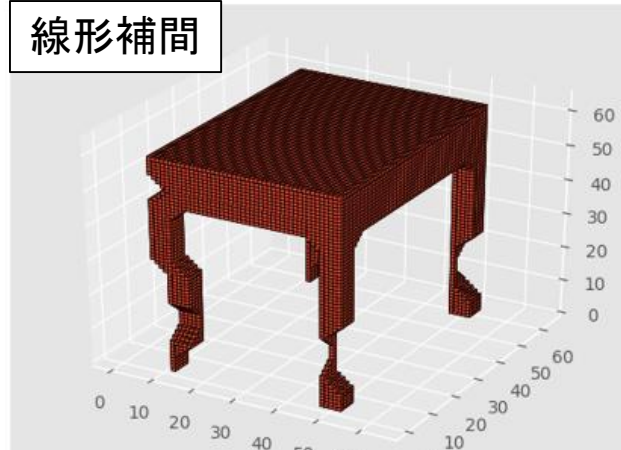
- 正解データ: それぞれのテストデータ 高解像度3次元モデル

# 実験2の結果 (table)

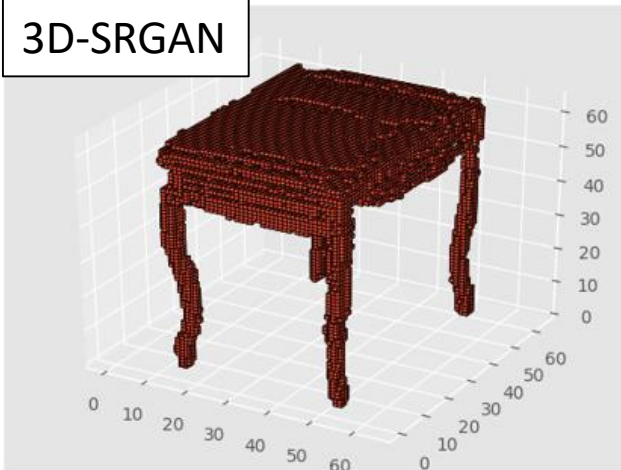
最近傍補間



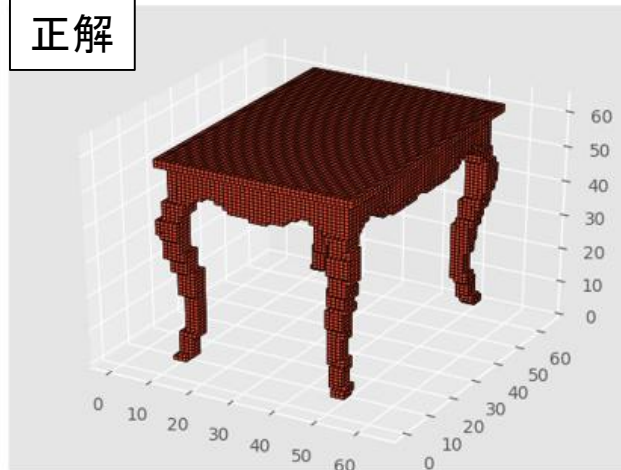
線形補間



3D-SRGAN



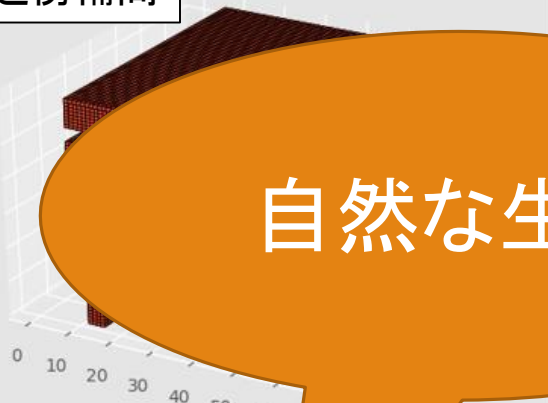
正解



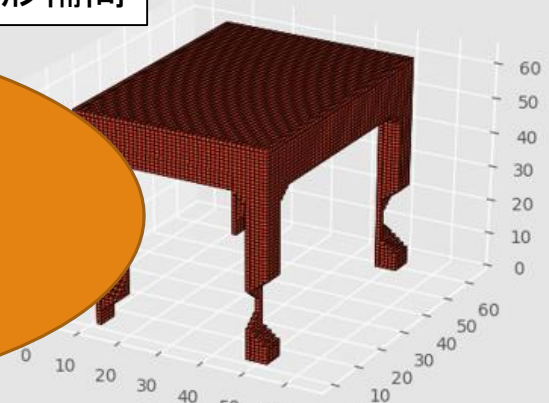


# 実験2の結果 (table)

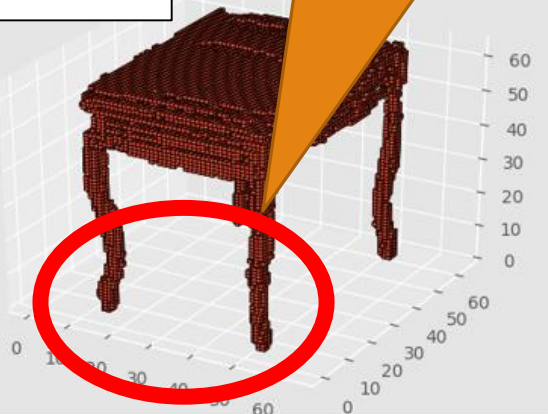
最近傍補間



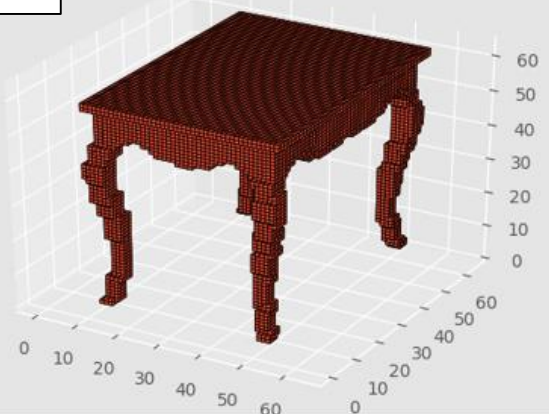
線形補間



3D-SRGAN



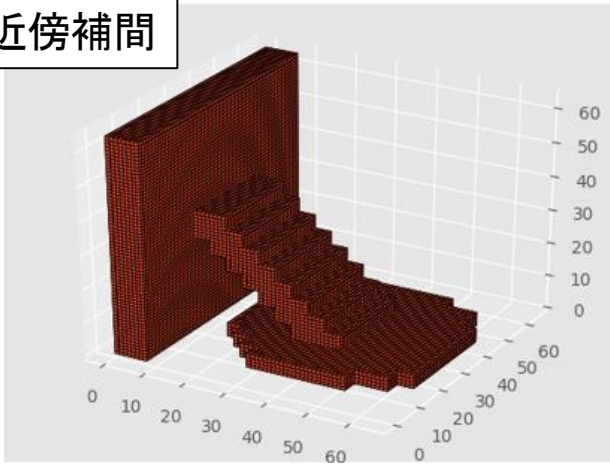
正解



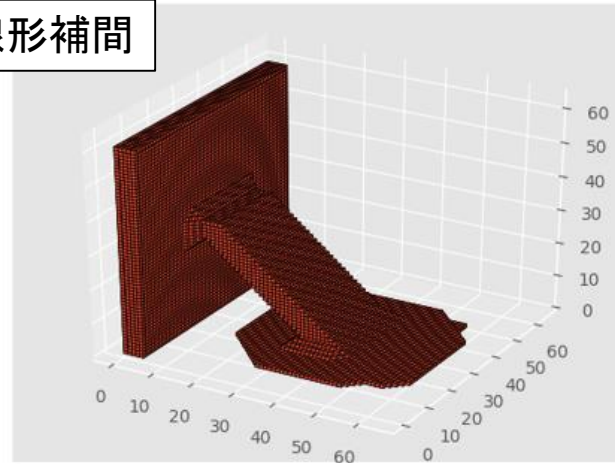
自然な生成

# 実験2の結果 (monitor)

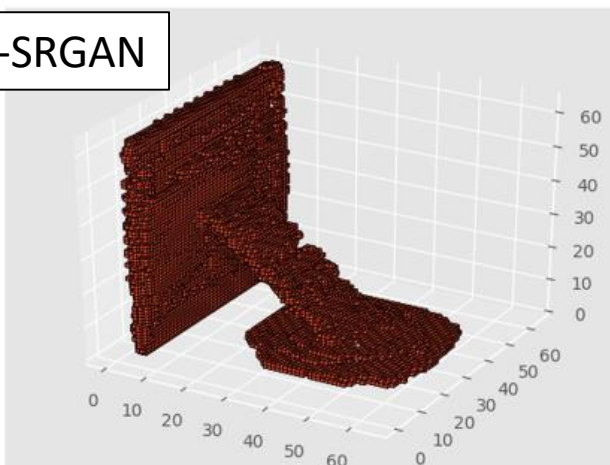
最近傍補間



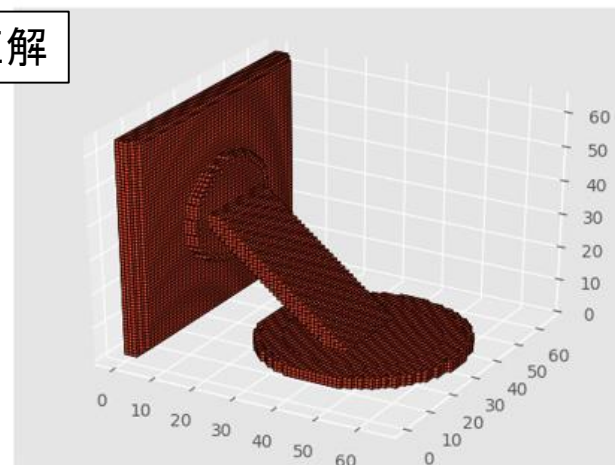
線形補間



3D-SRGAN



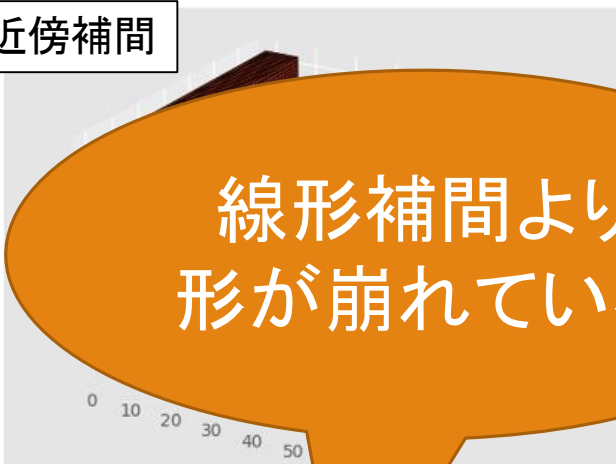
正解





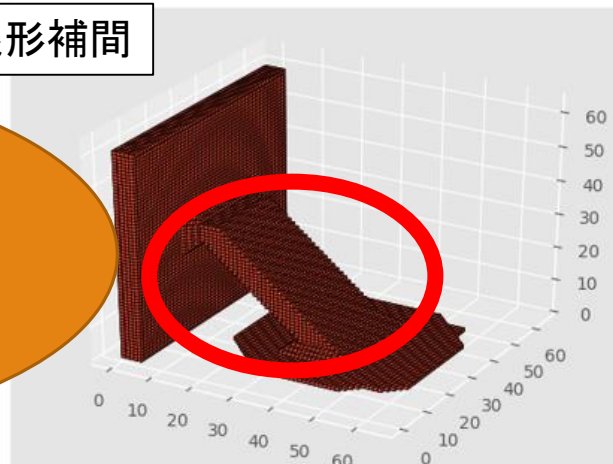
# 実験2の結果 (monitor)

最近傍補間

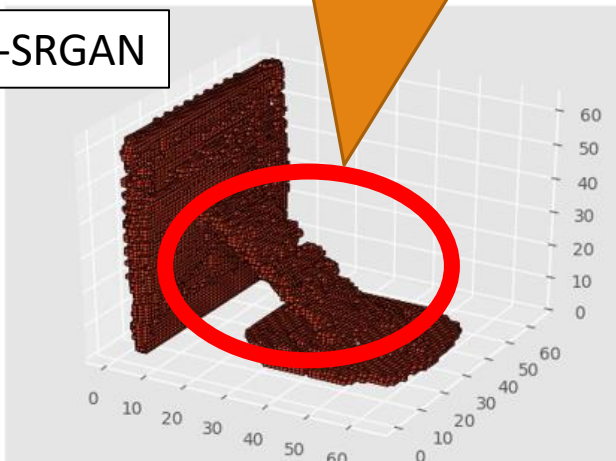


線形補間より  
形が崩れている

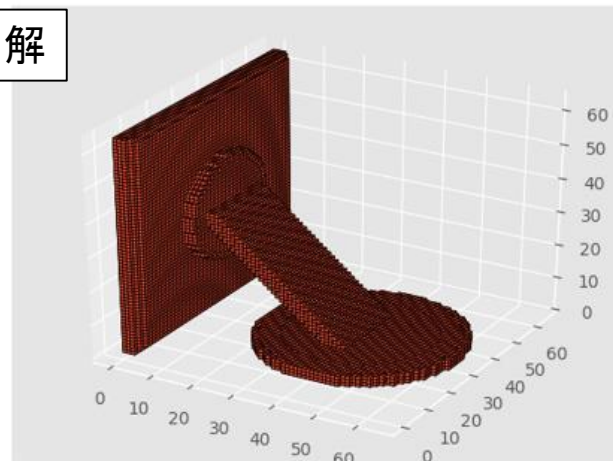
線形補間



3D-SRGAN

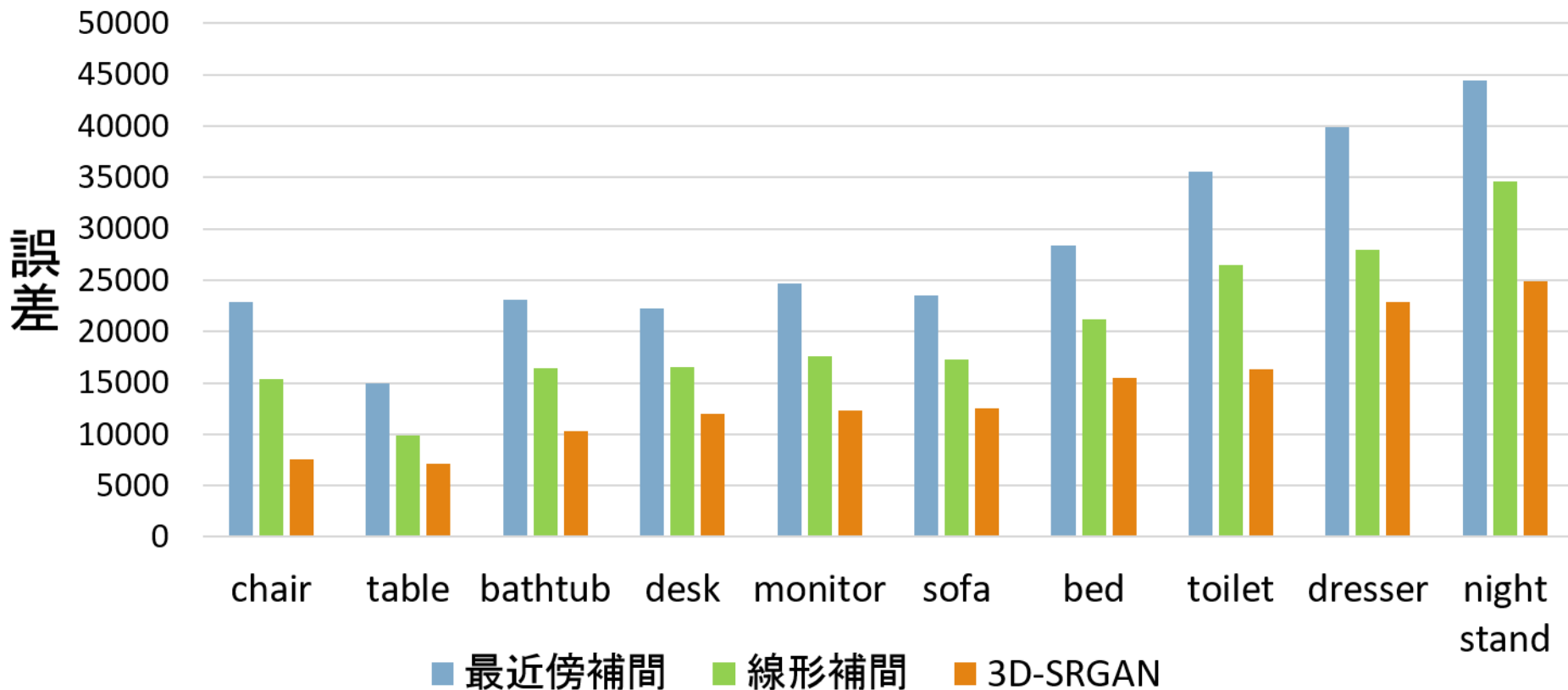


正解



# 実験2の結果

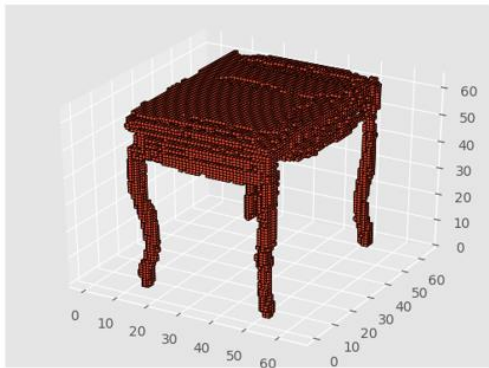
## クラス毎の誤差の平均



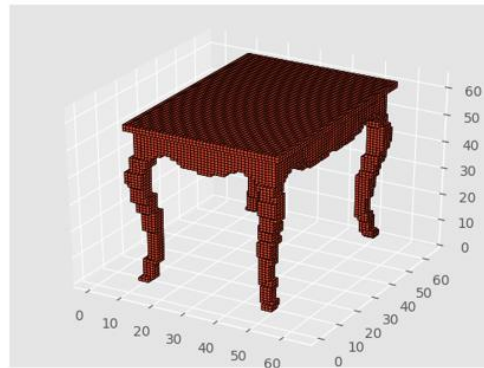
# 考察

---

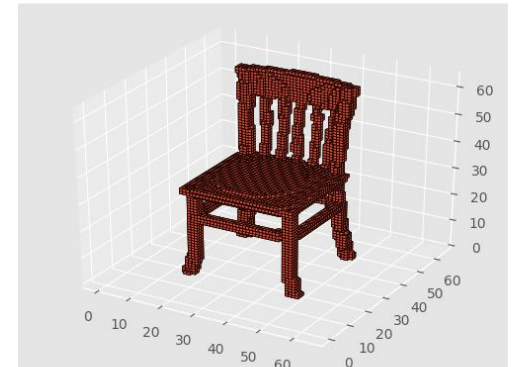
- tableの生成結果が自然な3次元モデル  
→ 学習したchairと形状が似ていた



3D-SRGAN



高解像度3次元モデル



chair

# まとめ

---

- 3D-SRGANを提案した
  - SRGANを3次元モデル超解像へ拡張した
- 3次元モデルで自然な超解像ができる
  - 比較手法より誤差が小さい
- 学習データと異なるクラスの超解像もできる
  - 多少の影響はある

# 今後の課題

---

- GeneratorとDiscriminatorの構造の最適化
  - 3D-SRGANの生成精度の向上
  
- 複数クラスの学習
  - 学習データの影響

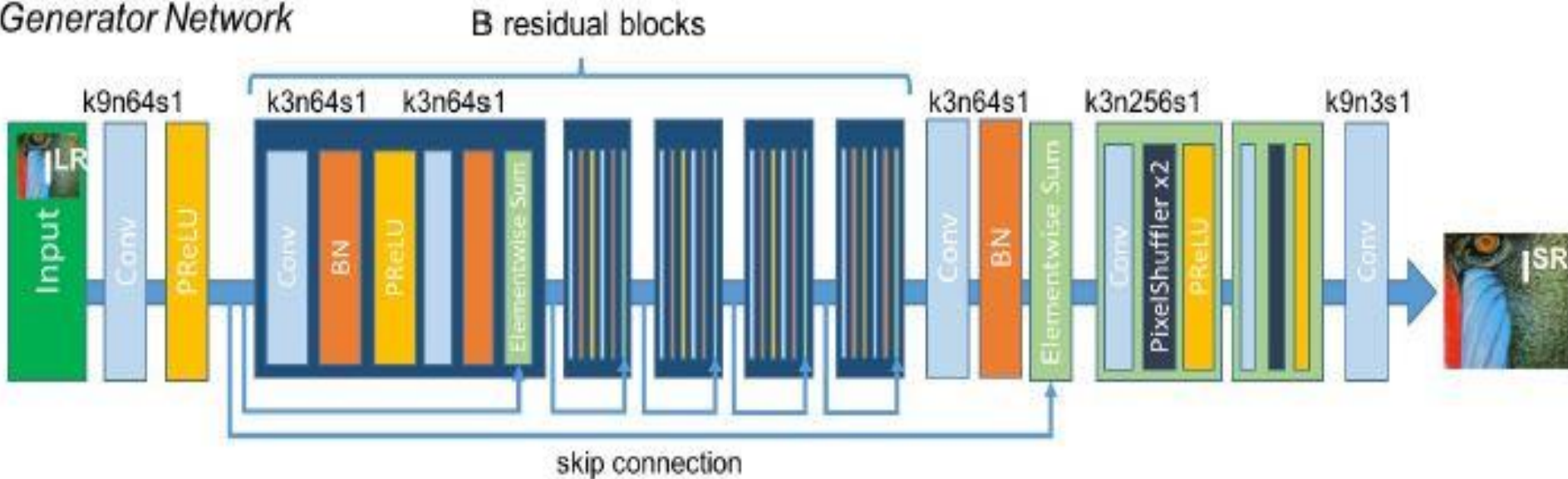
# 補足

---

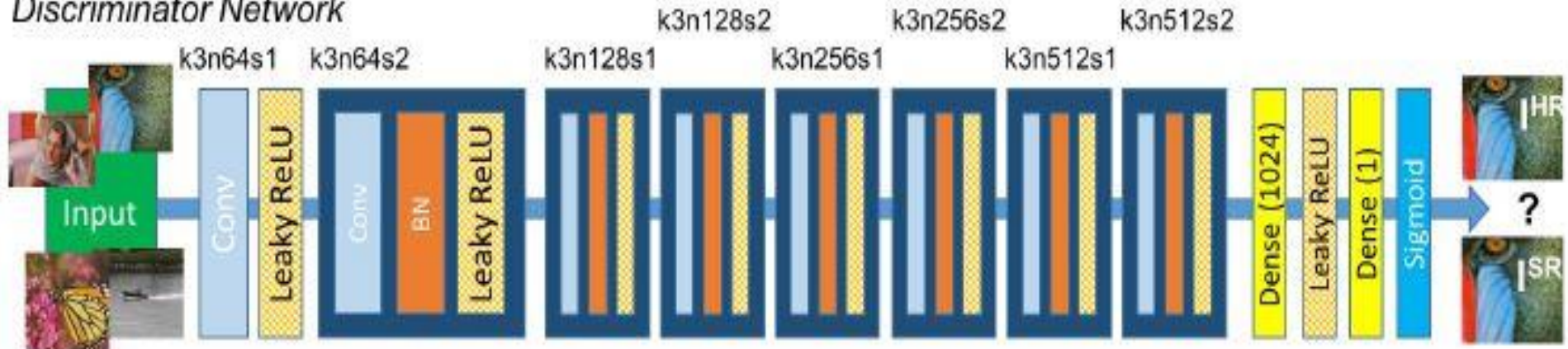
# SRGAN

## Generator & Discriminator

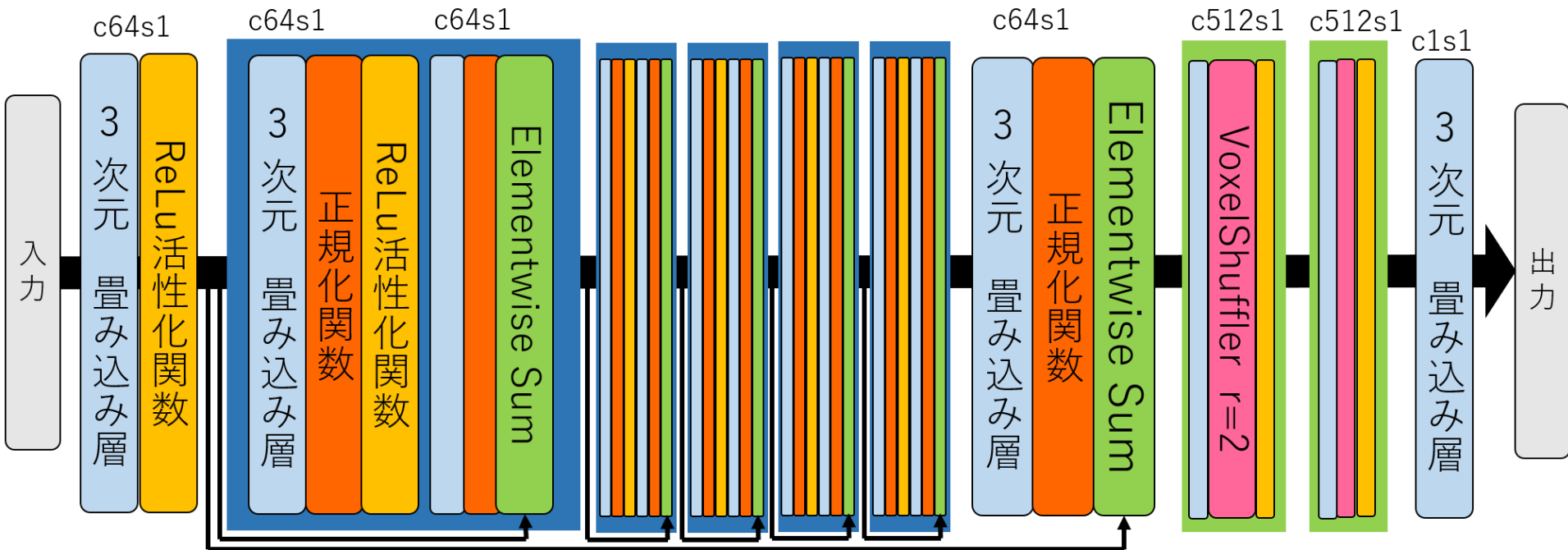
Generator Network



Discriminator Network

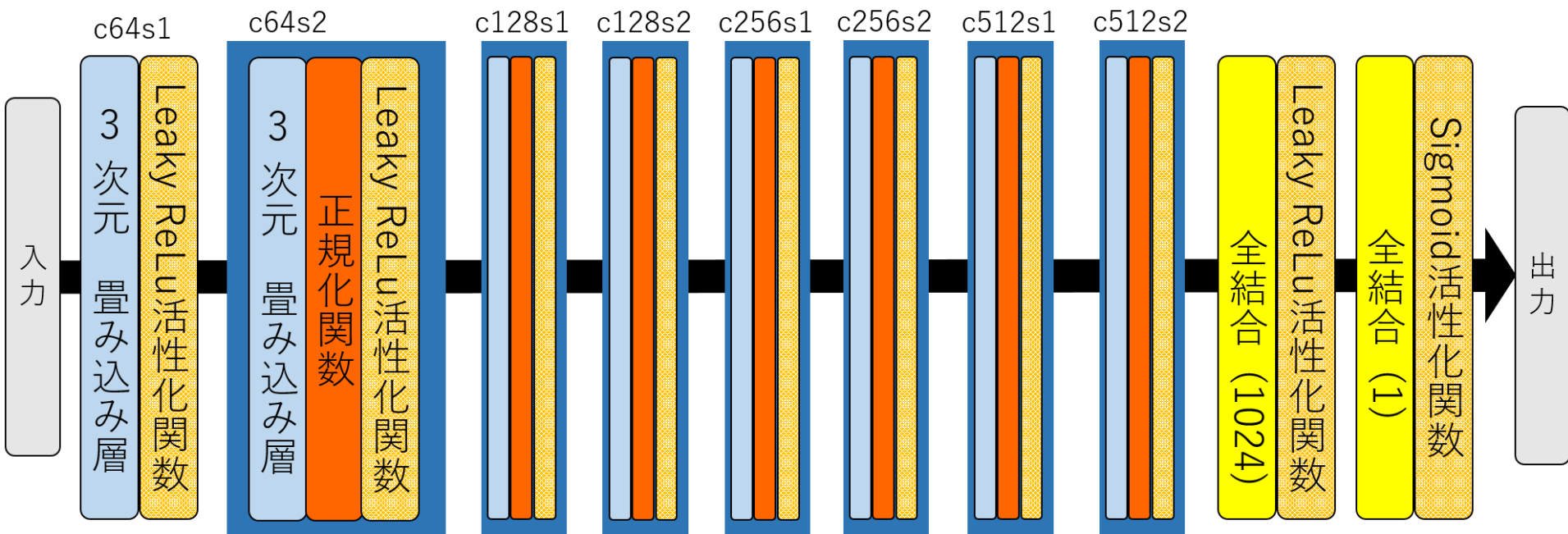


# Generator (生成モデル)





# Discriminator (識別器)



# 実験前準備

---

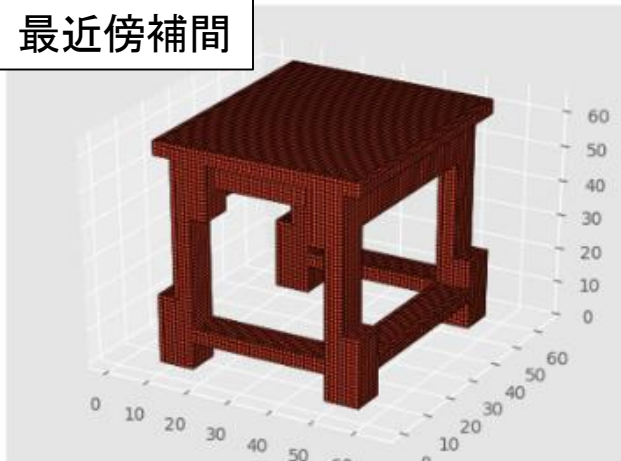
データセット: ModelNet10

10クラスの3次元モデル

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

# 実験2の結果 (㊦ desk)

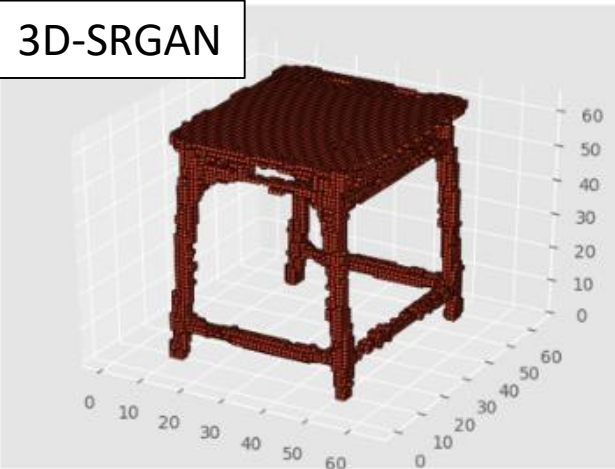
最近傍補間



線形補間



3D-SRGAN

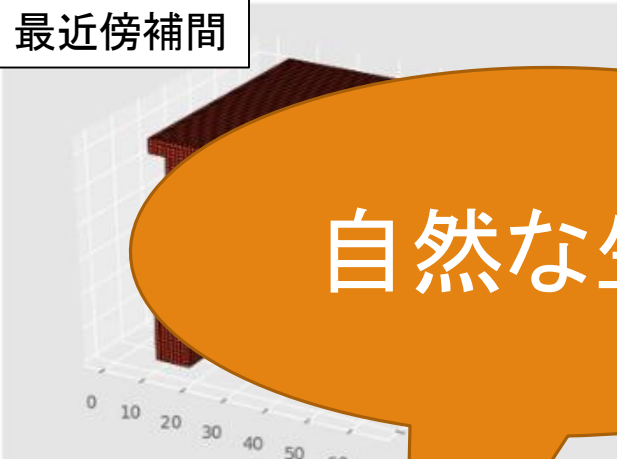


正解

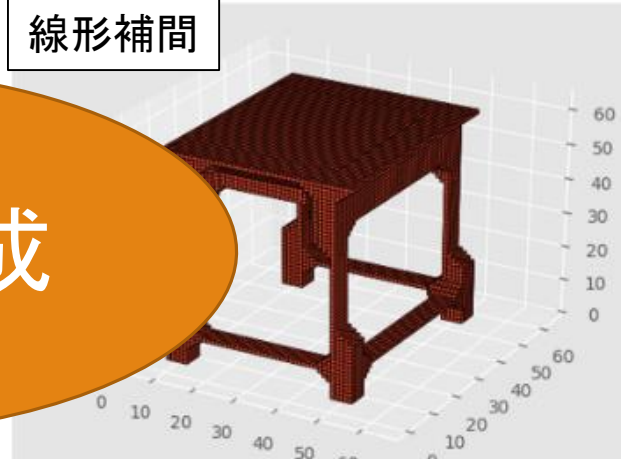


# 実験2の結果 (㊦ desk)

最近傍補間

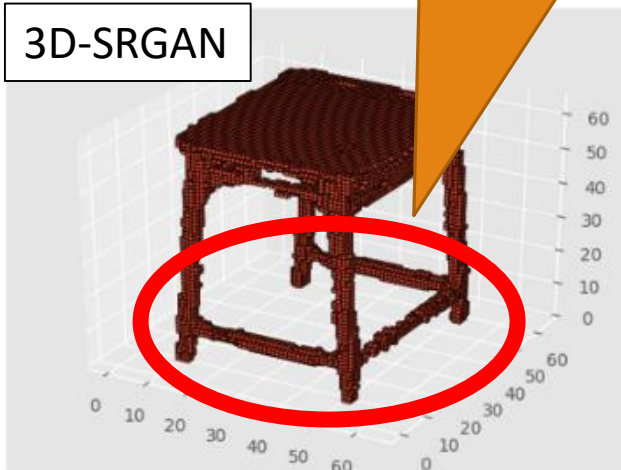


線形補間



自然な生成

3D-SRGAN



正解

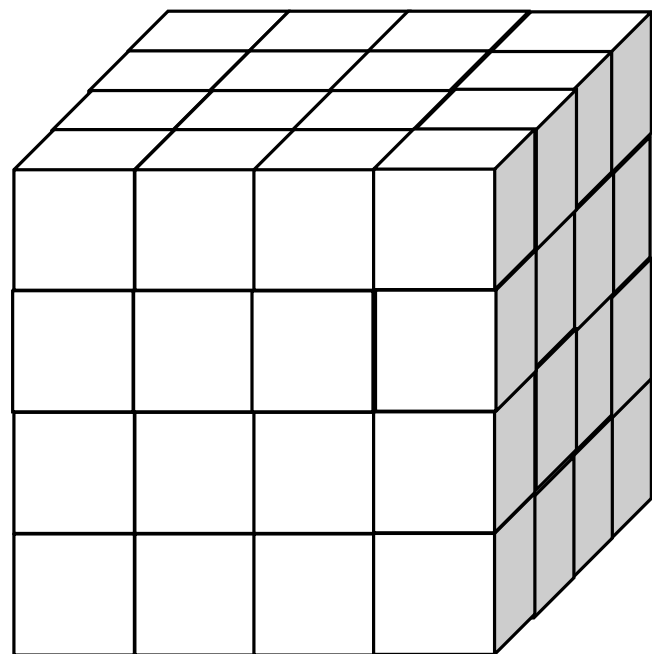


# 実験3

---

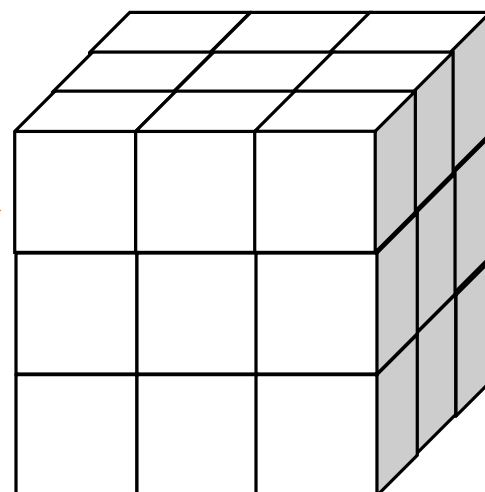
- 学習 : chair 学習データ
- 超解像する3次元モデル:  
chairのテストデータ
- 3次元モデルの局所的な情報を学習

# 局所的な情報を学習



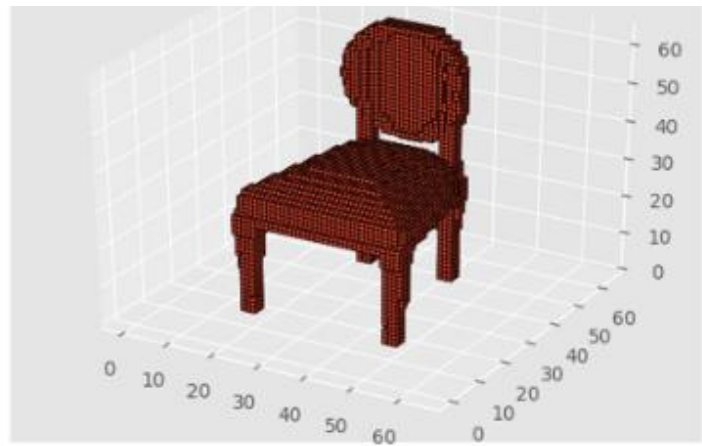
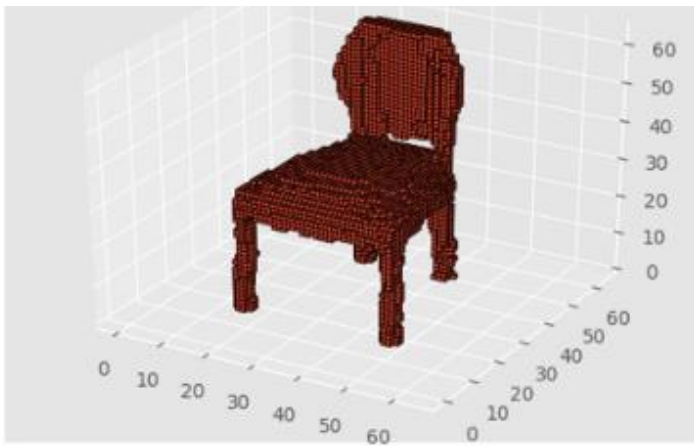
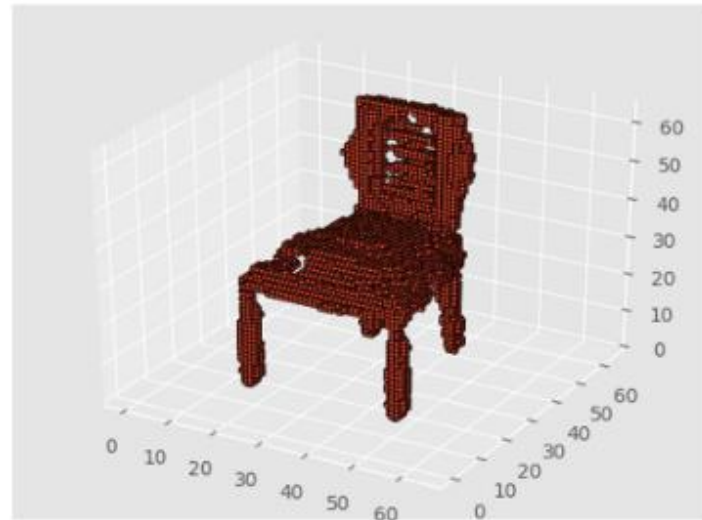
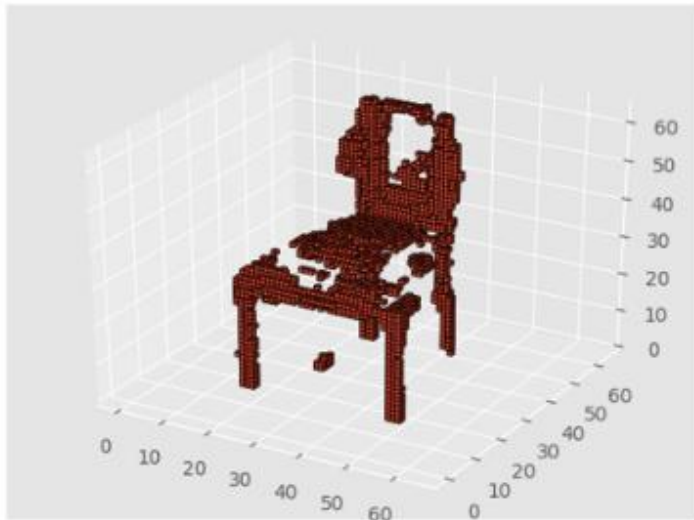
元の学習データ

$4 \times 4 \times 4$   
 $8 \times 8 \times 8$  で切り取る



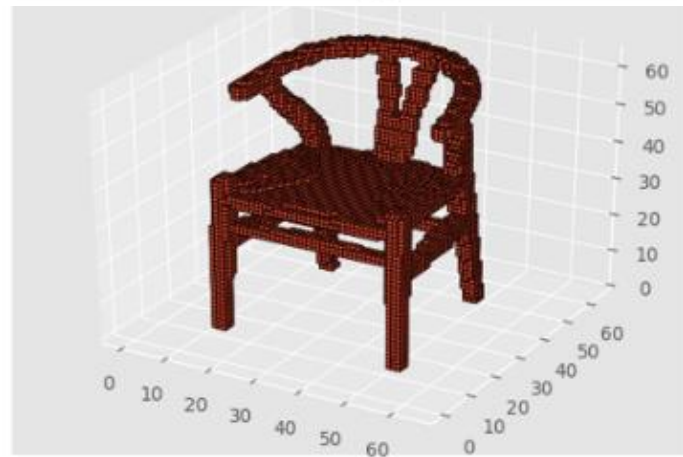
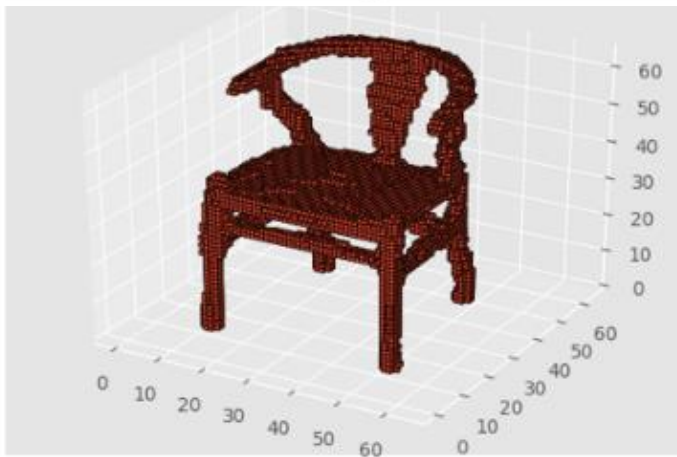
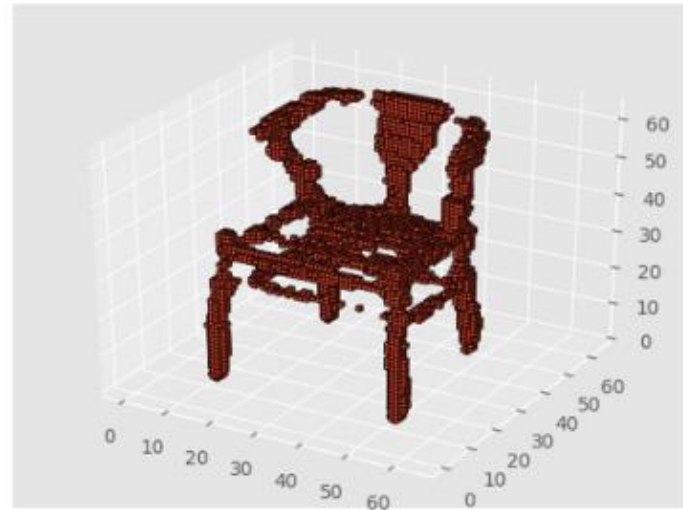
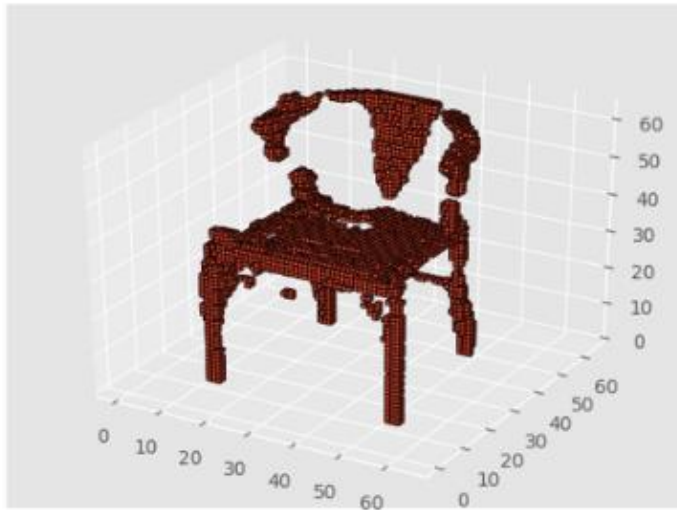
学習データ

# 実験3の結果(図)





# 実験3の結果(図)





# 実験3の結果(表)

	4-超解像3次元モデル 3D-SRGAN	8-超解像3次元モデル 3D-SRGAN	3D-SRGAN
誤差の平均	8707	8457	7512

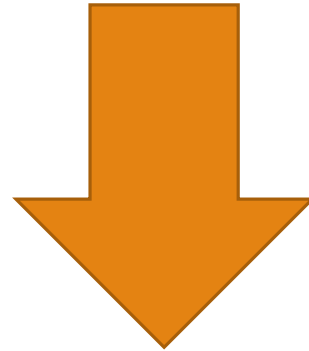
4-超解像3次元モデル: 学習データ  $4 \times 4 \times 4$

8-超解像3次元モデル: 学習データ  $8 \times 8 \times 8$

# 実験3の結果

---

学習データの局所的な情報を学習



- 誤差が大きい
- 形が崩れる