

敵対的生成 ネットワークを利用した 高解像画像生成の研究

神戸市立工業高等専門学校 電子工学科

藤本研究室 5年6番 入江一帆

目次

1. 研究背景
2. 研究目的
3. 敵対的生成の原理
4. DCGANの実験
5. 多段階GANの実験
6. DCGAN+SRCNNの実験
7. アンケート調査
8. まとめ

背景

深層学習の発展

画像認識

音声認識

自然言語処理

成果を上げている



生成モデルの登場

自己符号化器

敵対的生成ネットワーク

背景

高解像度の画像を生成

学習コスト

学習時間**増加**

使用メモリ**増加**

出力精度

ぼやける

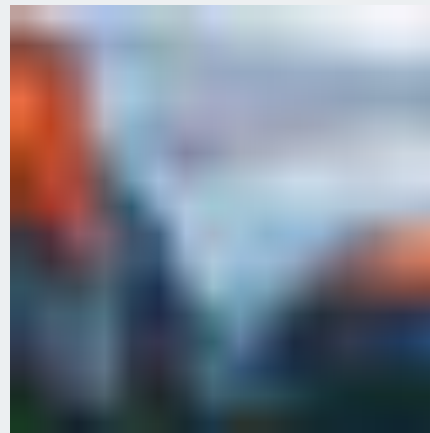
歪む



生成が難しい

背景

例 Web用画像素材を生成したい



32 x 32 [pixels]

認識できない

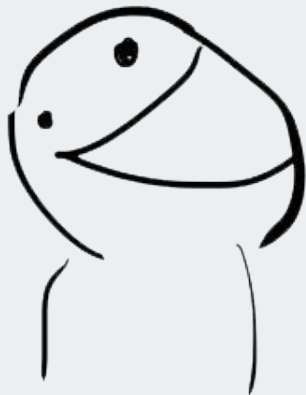


実用的ではない

背景

例 Web用画像素材を生成したい

やまだ



256 x 256 [pixels]

違和感を感じない



最低限必要

目的

解決策を比較検討

学習コスト **削減**

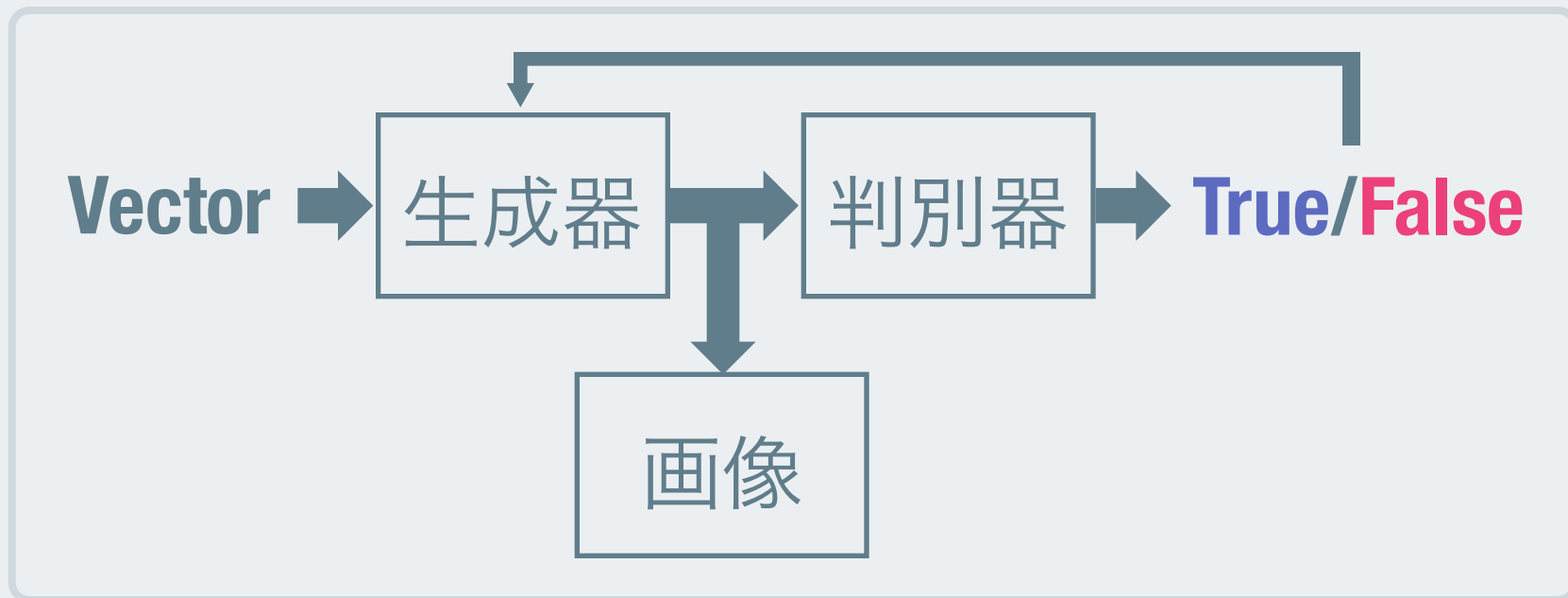
出力精度 **向上**



高解像度の画像を生成

原理

敵対的生成ネットワーク (GAN)



生成器と判別器を**組み合わせ**て構成
互いに**敵対**しながら学習

原理

生成器 (Generator)



多次元Vectorから画像を**生成**

原理

判別器 (Discriminator)



生成画像



生成画像を入力すると**偽物と判定**



訓練画像



訓練画像を入力すると**本物と判定**

原理

判別器 (Discriminator)



生成画像



生成画像を入力すると偽物と判定



訓練画像



訓練画像を入力すると本物と判定

原理

判別器 (Discriminator)



生成画像



生成画像を入力すると偽物と判定



訓練画像



訓練画像を入力すると本物と判定

原理

敵対的生成 (Generative Adversarial)

Vector



生成器



良い画像

生成器は**良い画像**の生成を目指す



良い画像



判別器



True

判別器が本物と間違い**誤判定**

原理

敵対的生成 (Generative Adversarial)

Vector



生成器



良い画像

生成器は**良い画像**の生成を目指す



良い画像



判別器



True

判別器が本物と間違い**誤判定**

実験 ①

DCGAN (Deep Convolutional GAN)

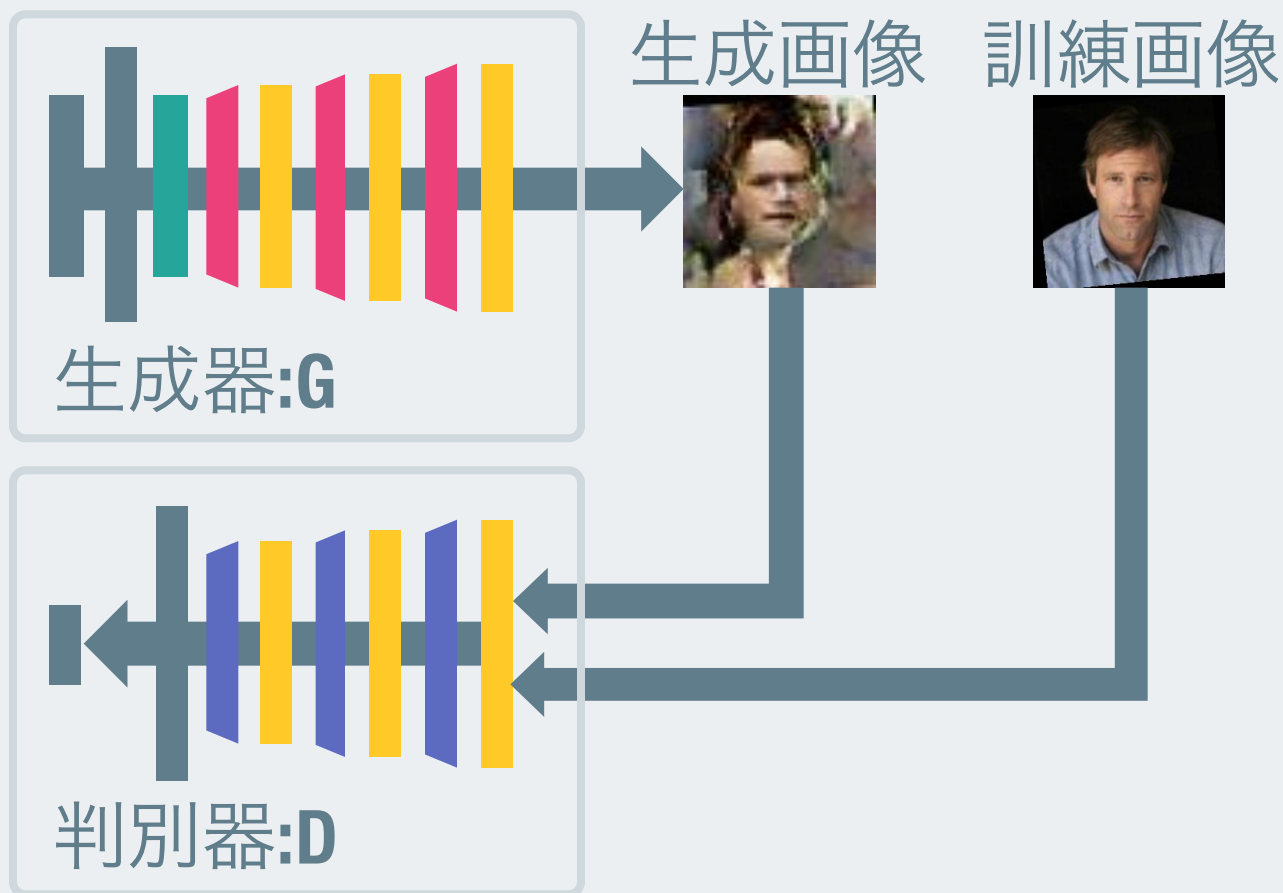
Fully Connected

Reshape

Convolution

Up Sampling

Max Pooling



実験 ①

DCGAN (Deep Convolutional GAN)

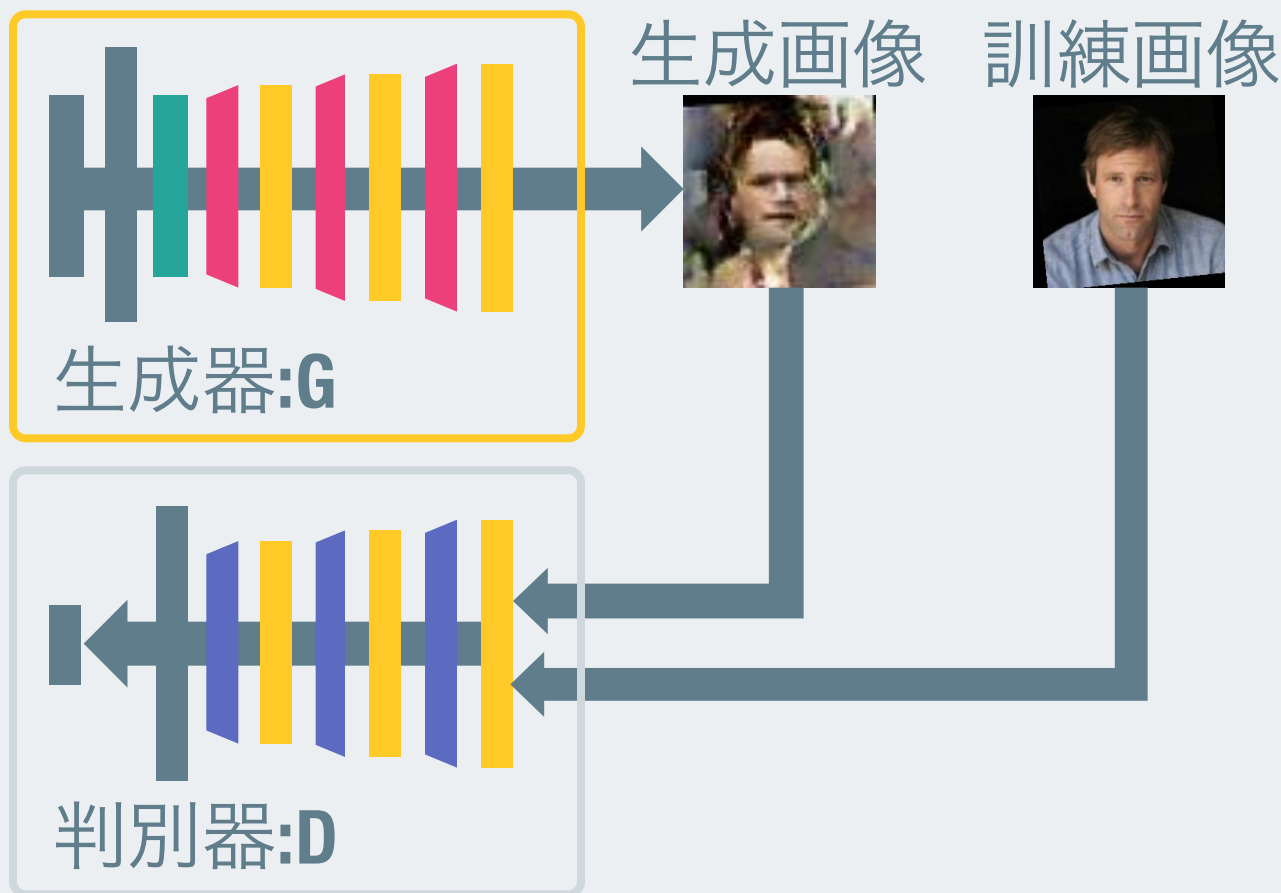
Fully Connected

Reshape

Convolution

Up Sampling

Max Pooling



実験 ①

DCGAN (Deep Convolutional GAN)

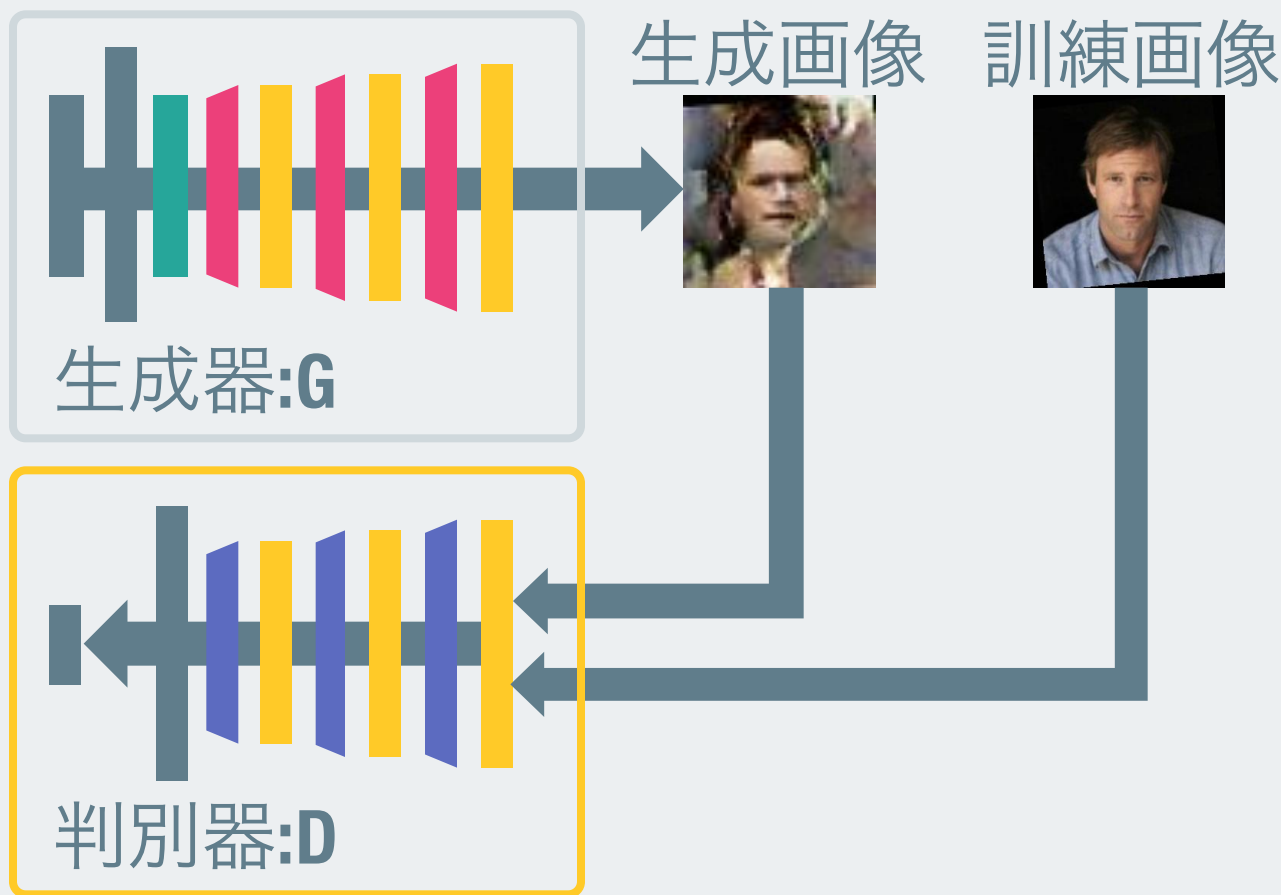
Fully Connected

Reshape

Convolution

Up Sampling

Max Pooling



結果 ①

解像度による変化 (3000[epochs])



32 x 32 [pixels]



64 x 64 [pixels]

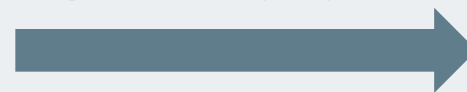


128 x 128 [pixels]

精度向上



ボケ・歪み



高解像度化に伴い学習コスト急増

結果 ①

解像度による変化 (3000[epochs])



32 x 32 [pixels]



64 x 64 [pixels]

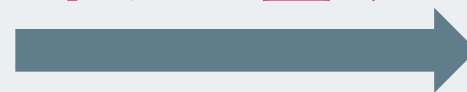


128 x 128 [pixels]

精度向上



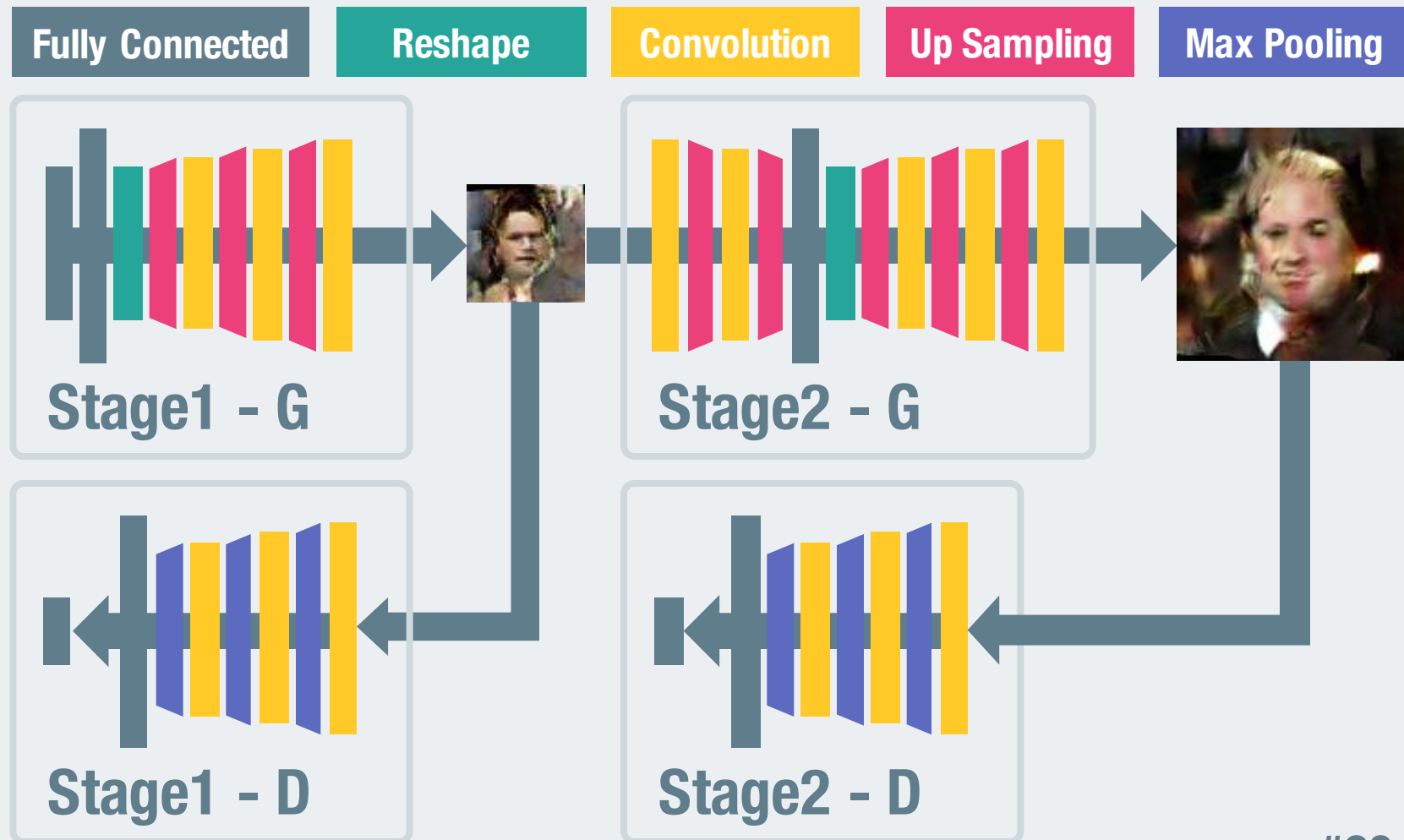
ボケ・歪み



高解像度化に伴い**学習コスト急増**

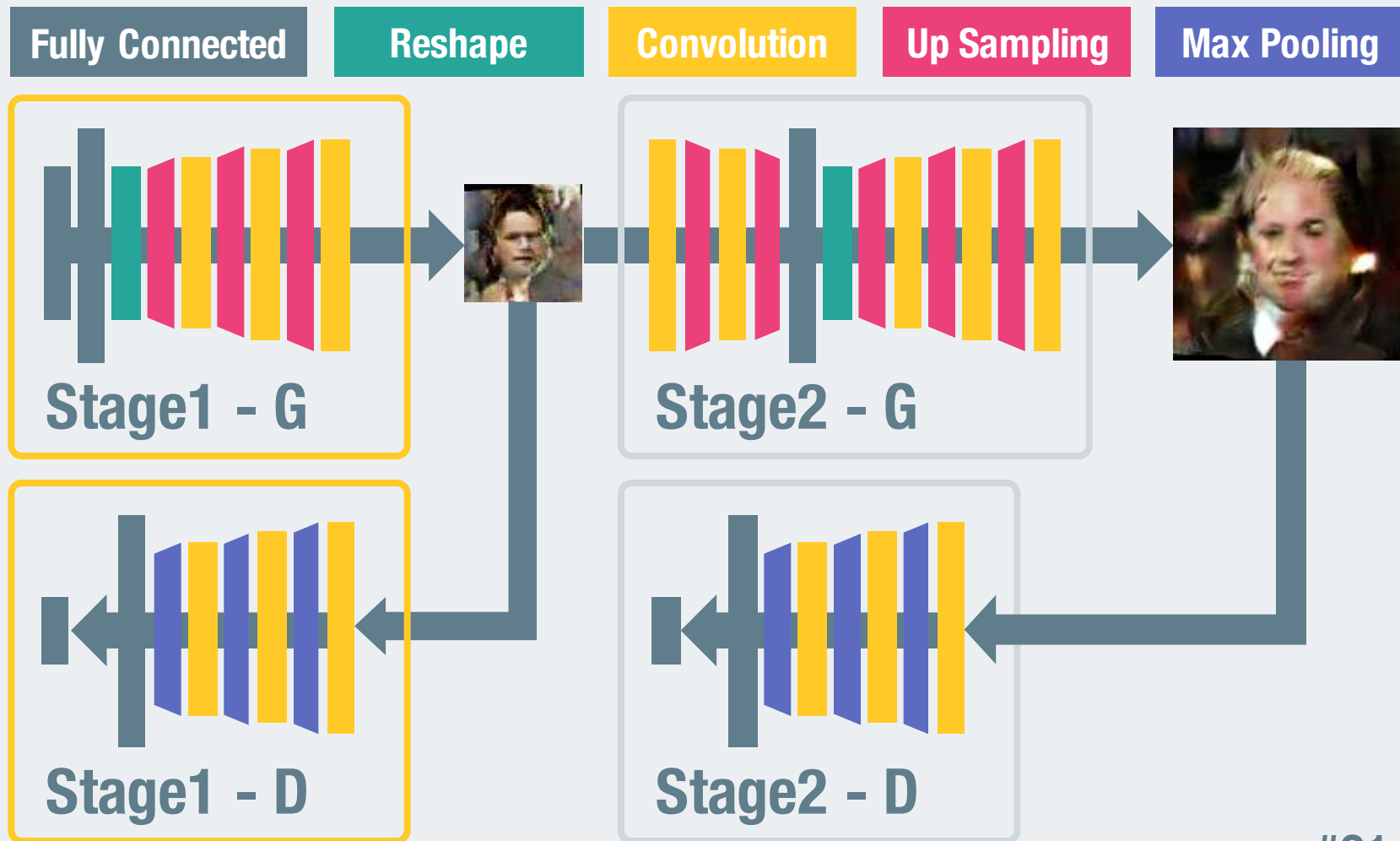
実験 ②

多段階GAN



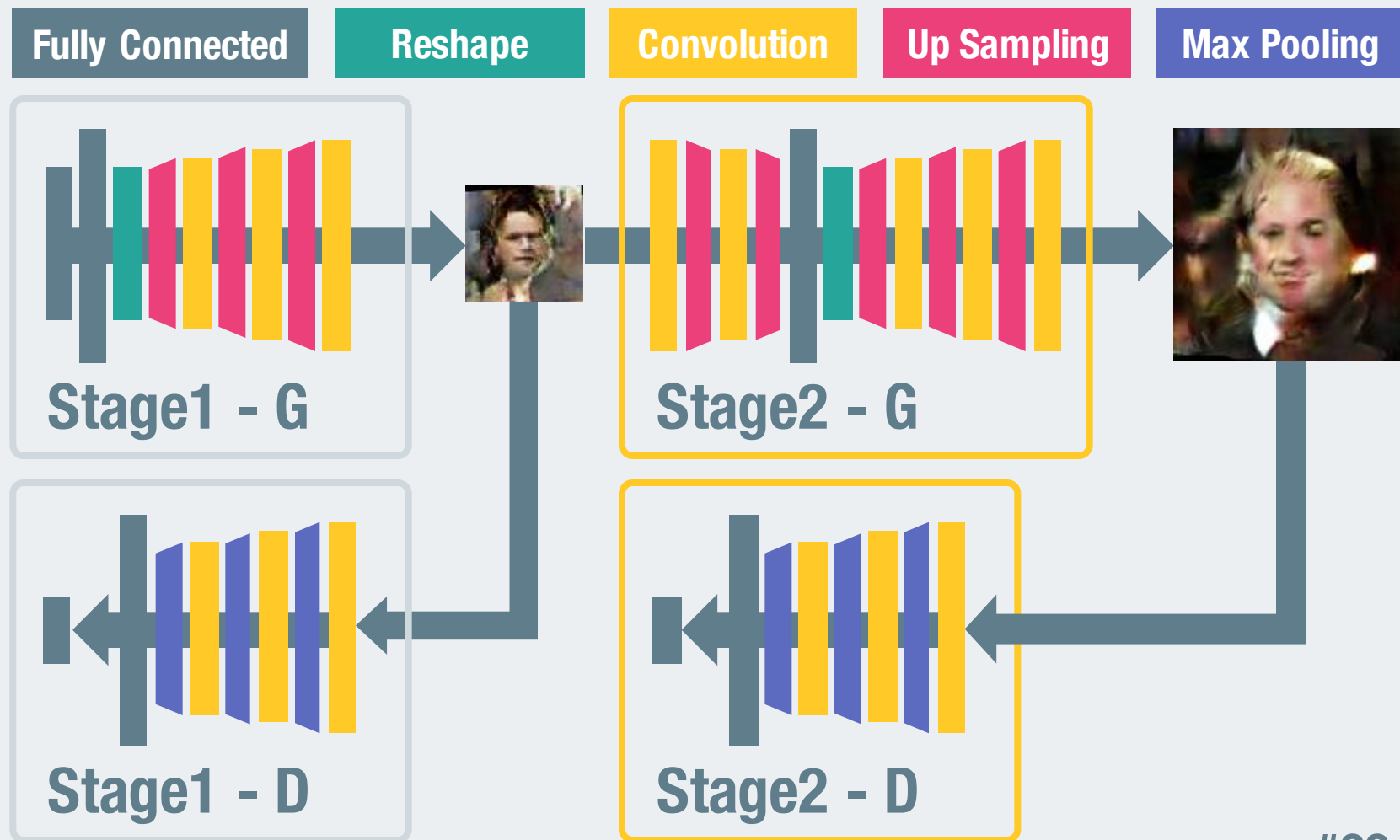
実験 ②

多段階GAN



実験 ②

多段階GAN



結果 ②

多段階GAN (3000[epochs])



多段階GAN

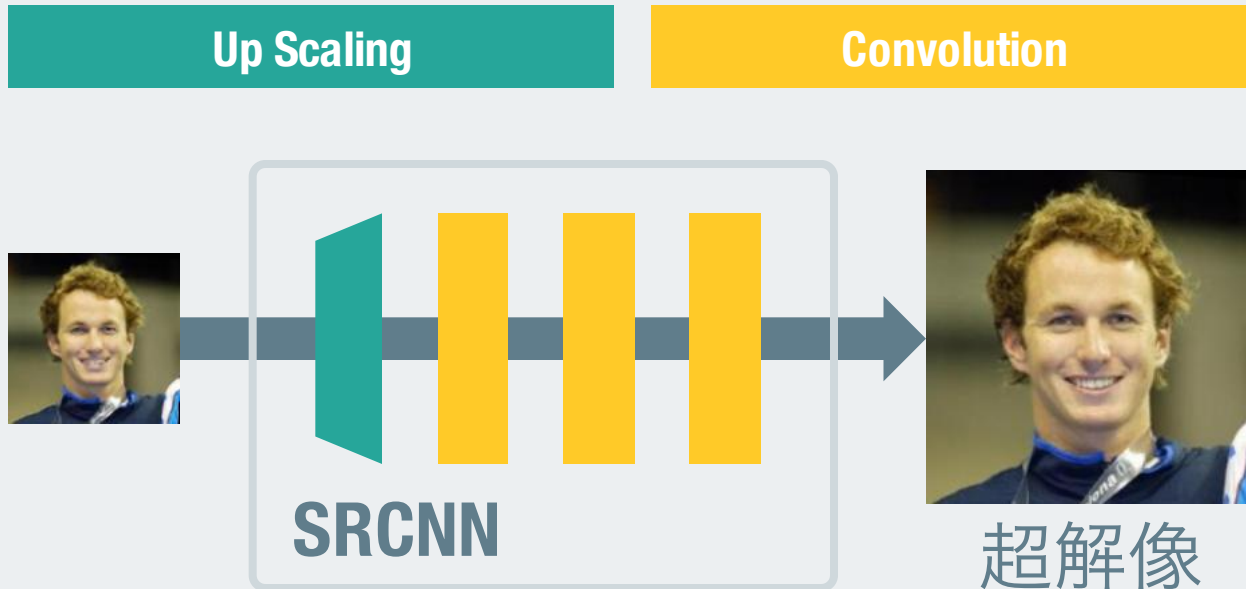


DCGAN (一段)

DCGANと比べて**良い結果**を得られた
学習コストが急増する問題は**解決せず**

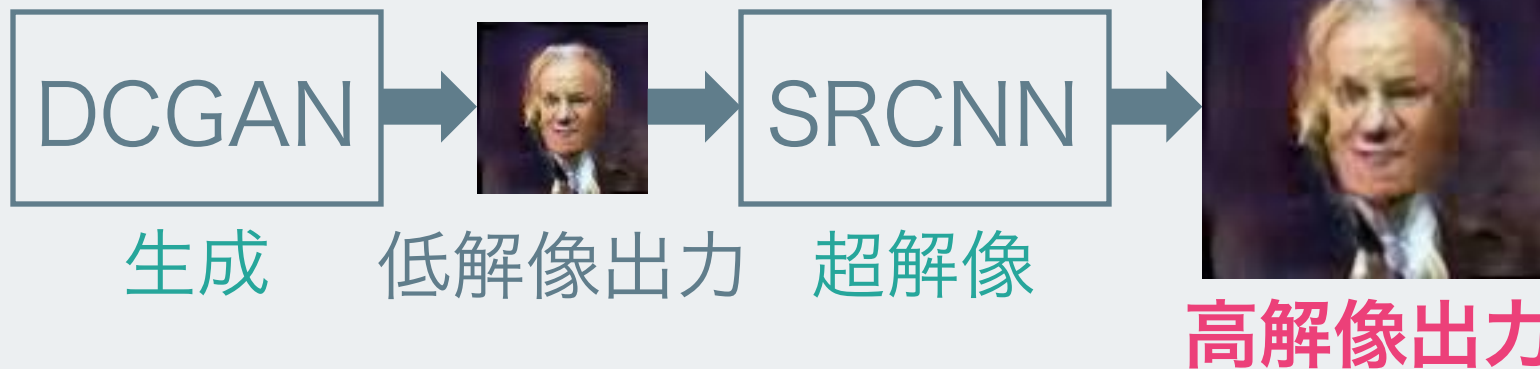
実験 ③

SRCNN (Super Resolution Neural Network)



実験 ③

DCGAN+SRCNN



結果 ③

DCGAN+SRCNN (6000, 2000_[epochs])



DCGAN出力



超解像結果

一段のDCGANと比べて**良い結果**を得られた
学習コストの**削減に成功**

実験 ④

アンケート調査

多段階GANとDCGAN+SRCNNを比較

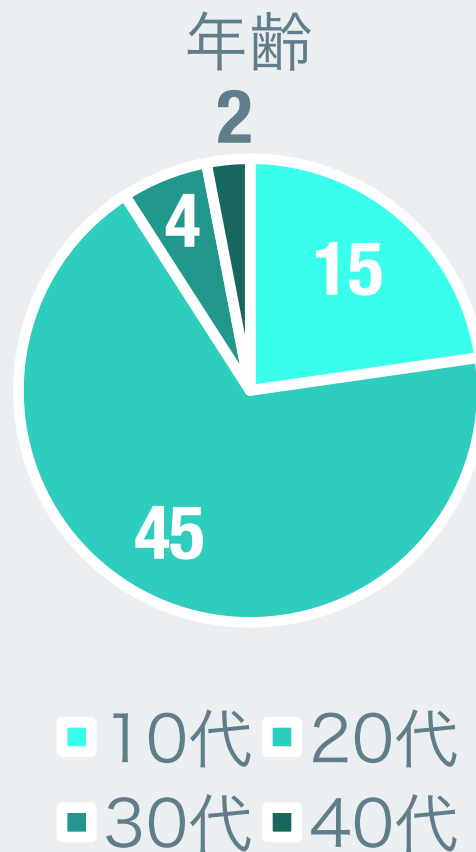
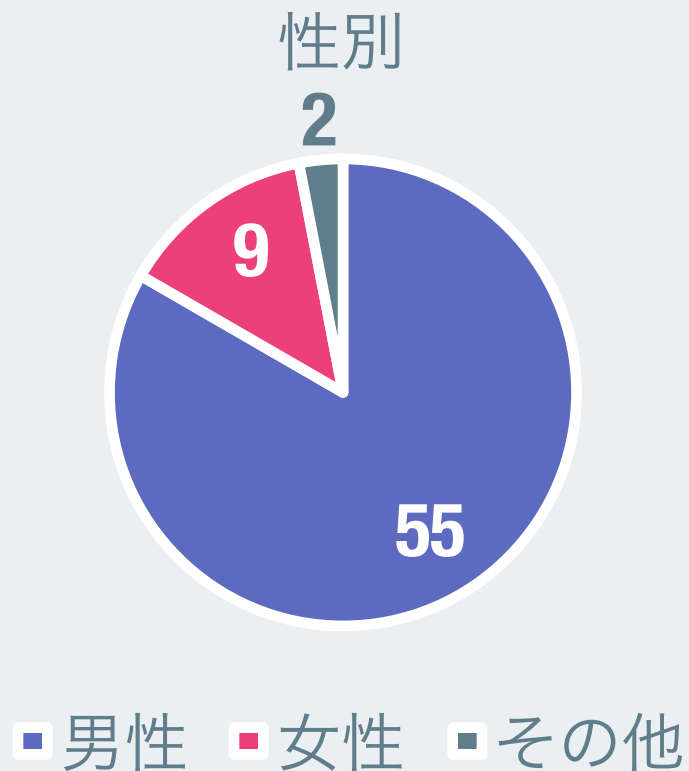
目視で良いと感じた画像を選択

(3/8) 最もよく生成できていると思われる画像をクリックまたはタップしてください。



実験 ④

母集団構成 (全66[人])



結果 ④

アンケート結果

学習反復数共通としたとき

方法	選択率 [%]	反復数 [epochs]	学習時間 [hrs]
多段階GAN	49.3	6000	113.8
DCGAN+SRCNN	50.8	6000, 2000	75.2

結果 ④

アンケート結果

学習反復数共通としたとき

方法	選択率 [%]	反復数 [epochs]	学習時間 [hrs]
多段階GAN	49.3	6000	113.8
DCGAN+SRCNN	50.8	6000, 2000	75.2

結果 ④

アンケート結果

学習反復数共通としたとき

方法	選択率 [%]	反復数 [epochs]	学習時間 [hrs]
多段階GAN	49.3	6000	113.8
DCGAN+SRCNN	50.8	6000, 2000	75.2

結果 ④

アンケート結果

学習時間共通としたとき

方法	選択率 [%]	反復数 [epochs]	学習時間 [hrs]
多段階GAN	23.1	4000	76.8
DCGAN+SRCNN	76.9	6000, 2000	75.2

結果 ④

アンケート結果

学習時間共通としたとき

方法	選択率 [%]	反復数 [epochs]	学習時間 [hrs]
多段階GAN	23.1	4000	76.8
DCGAN+SRCNN	76.9	6000, 2000	75.2

まとめ

一段のDCGAN

高解像度化



出力精度の限界

多段階GAN

高解像度化



学習コストの問題

DCGAN+SRCNN

学習コスト削減



学習時間短縮

概要

1. 研究背景 … 生成モデルの登場 高解像度出力の難しさ
2. 研究目的 … 学習コスト 出力精度 高解像度化
3. 敵対的生成の原理 … 生成器 判別器 敵対的生成
4. DCGANの実験 … 解像度による変化
5. 多段階GANの実験 … 学習コストの問題
6. DCGAN+SRCNNの実験 … 超解像 コスト削減
7. アンケート調査 … 反復数共通 学習時間共通
8. まとめ … DCGAN 多段階GAN DCGAN+SRCNN

学生からの質問大歓迎

質疑応答用資料

畳み込み

例 5x5画像に3x3フィルタをかける

1	2	3	1	2
3	1	2	3	1
2	3	1	2	3
1	2	3	1	2
3	1	2	3	1

画像

$$\begin{aligned} &1 \times 0 + 2 \times 1 + 3 \times 0 \\ &+ 3 \times 1 + 1 \times 2 + 2 \times 1 \\ &+ 2 \times 0 + 3 \times 1 + 1 \times 0 = 12 \end{aligned}$$



0	1	0
1	2	1
0	1	0

フィルタ



$$12 / 9 = 1.33$$

1	1	1
1	1	1
1	1	1

結果

畳み込み

例 5x5画像に3x3フィルタをかける

1	2	3	1	2
3	1	2	3	1
2	3	1	2	3
1	2	3	1	2
3	1	2	3	1

画像

$$\begin{aligned} &2 \times 0 + 3 \times 1 + 1 \times 0 \\ &+ 1 \times 1 + 2 \times 2 + 3 \times 1 \\ &+ 3 \times 0 + 1 \times 1 + 1 \times 0 = 12 \end{aligned}$$



0	1	0
1	2	1
0	1	0

フィルタ



$$12 / 9 = 1.33$$

1	1	1
1	1	1
1	1	1

結果

Up Sampling

例 3x3画像を2x2でUp Samplingする

1	2	3
1	2	3
1	2	3

画像

×

1	0
0	1

フィルタ



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

結果

Up Sampling

例 3x3画像を2x2でUp Samplingする

1	2	3
1	2	3
1	2	3

画像

×

1	0
0	1

フィルタ



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

結果

Max Pooling

例 6x6画像を2x2でMax Poolingする

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

画像

最大値



1	3	2
3	3	3
3	3	3

結果

Max Pooling

例 6x6画像を2x2でMax Poolingする

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

画像

最大値



1	3	2
3	3	3
3	3	3

結果

高解像度とは

大辞林による定義

一般的なシステムより解像度が高いこと



写真での定義

1200万画像 = 4000x3000 [pixels]

GANでは生成困難

GANでの定義

128x128 ~ 256x256 [pixels]

人間が違和感なく認識

ネットワーク構造 (Generator)

Layer (type)	Output Shape	Parameters
Input	100	
Dense	512	
Reshape	(h/8, h/8, 256)	
Up Sampling	(h/4, h/4, 256)	Size: (2, 2)
Convolution	(h/4, h/4, 128)	Kernel: (3, 3)
Up Sampling	(h/2, h/2, 128)	Size: (2, 2)
Convolution	(h/2, h/2, 64)	Kernel: (3, 3)
Up Sampling	(h, h, 64)	Size: (2, 2)
Convolution	(h, h, 3)	Kernel: (3, 3)

ネットワーク構造 (Discriminator)

Layer (type)	Output Shape	Parameters
Input	(h, h, 3)	
Convolution	(h, h, 64)	Kernel: (3, 3)
Max Pooling	(h/2, h/2, 64)	Size: (2, 2)
Convolution	(h/2, h/2, 128)	Kernel: (3, 3)
Max Pooling	(h/4, h/4, 128)	Size: (2, 2)
Convolution	(h/4, h/4, 256)	Kernel: (3, 3)
Max Pooling	(h/8, h/8, 256)	Size: (2, 2)
Flatten	256 x h/8 x h/8	
Dense	512	
Dense	1	

ネットワーク構造 (Up Sampler)

Layer (type)	Output Shape	Parameters
Input	(h, h, 3)	
Convolution	(h, h, 64)	Kernel: (3, 3)
Max Pooling	(h/2, h/2, 64)	Size: (2, 2)
Convolution	(h/2, h/2, 128)	Kernel: (3, 3)
Max Pooling	(h/4, h/4, 128)	Size: (2, 2)
Flatten	128 x h/4 x h/4	
Dense	256	
Reshape	(h/4, h/4, 256)	
Up Sampling	(h/2, h/2, 256)	Size: (2, 2)
Convolution	(h, h, 128)	Kernel: (3, 3)
Up Sampling	(h, h, 128)	Size: (2, 2)
Convolution	(h, h, 64)	Kernel: (3, 3)
Up Sampling	(2h, 2h, 64)	Size: (2, 2)
Convolution	(2h, 2h, 3)	Kernel: (3, 3)

ネットワーク構造 (SRCNN)

Layer (type)	Output Shape	Parameters
Input	(h, h, 3)	
Up Scaling	(2h, 2h, 3)	Nearest Neighbor
Convolution	(2h, 2h, 256)	Kernel: (9, 9)
Convolution	(2h, 2h, 128)	Kernel: (3, 3)
Convolution	(2h, 2h, 3)	Kernel: (5, 5)

実験環境

種別	種類
CPU	Intel Core i5 (2.8GHz)
GPU	GTX 1050 Ti
OS	Ubuntu 16
Python	3.6
Tensorflow	1.2
Keras	2.0

付録

ノートPCによる画像生成

富士通 LIFEBOOK で生成



一家に一台のGAN