人工知能による画像生成

AI演習

https://www.kkaneko.jp/ai/ae/index.html

金子邦彦

①画像生成の基本技術

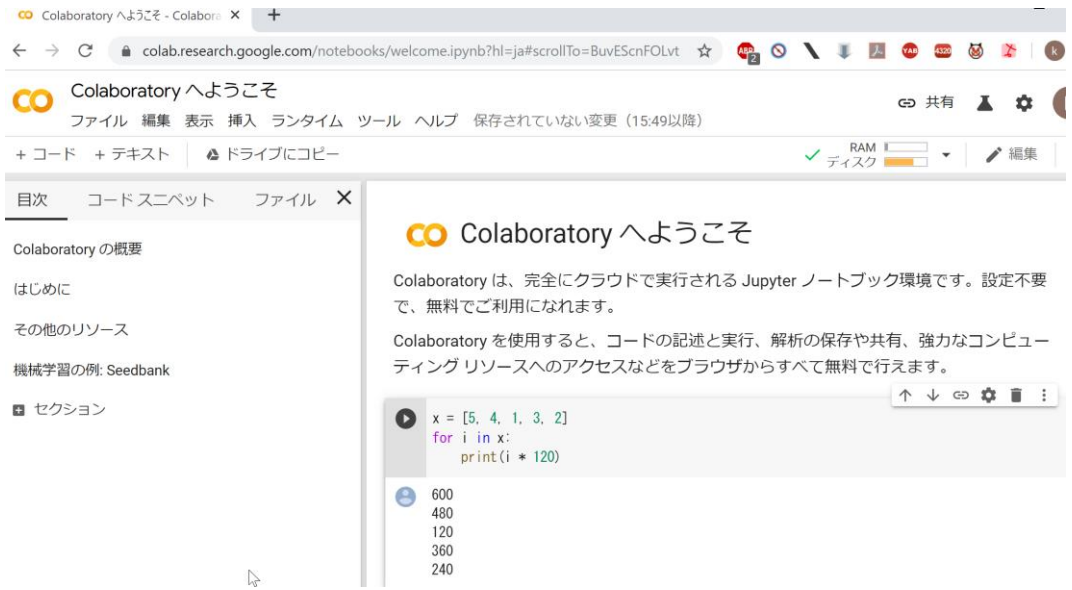
②テキストからの画像生成、実応用

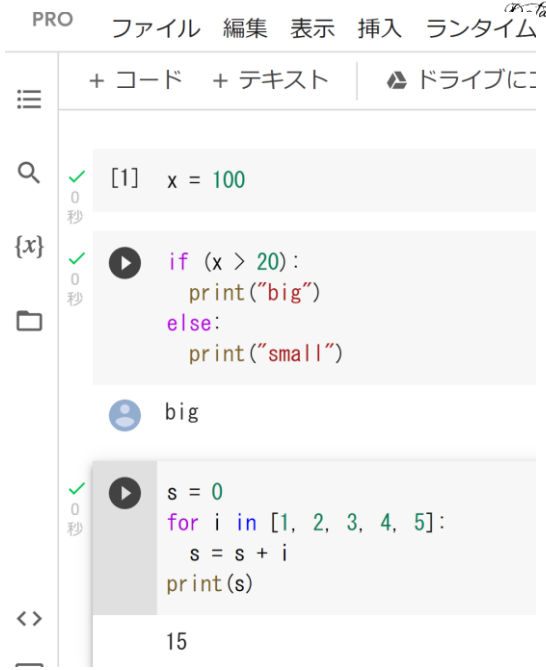
③実用スキルの向上

アウトライン

1. イントロダクション
2. 画像生成のバリエーション
3. GAN の概要
4. GAN の仕組み
5. 演習

Google Colaboratory

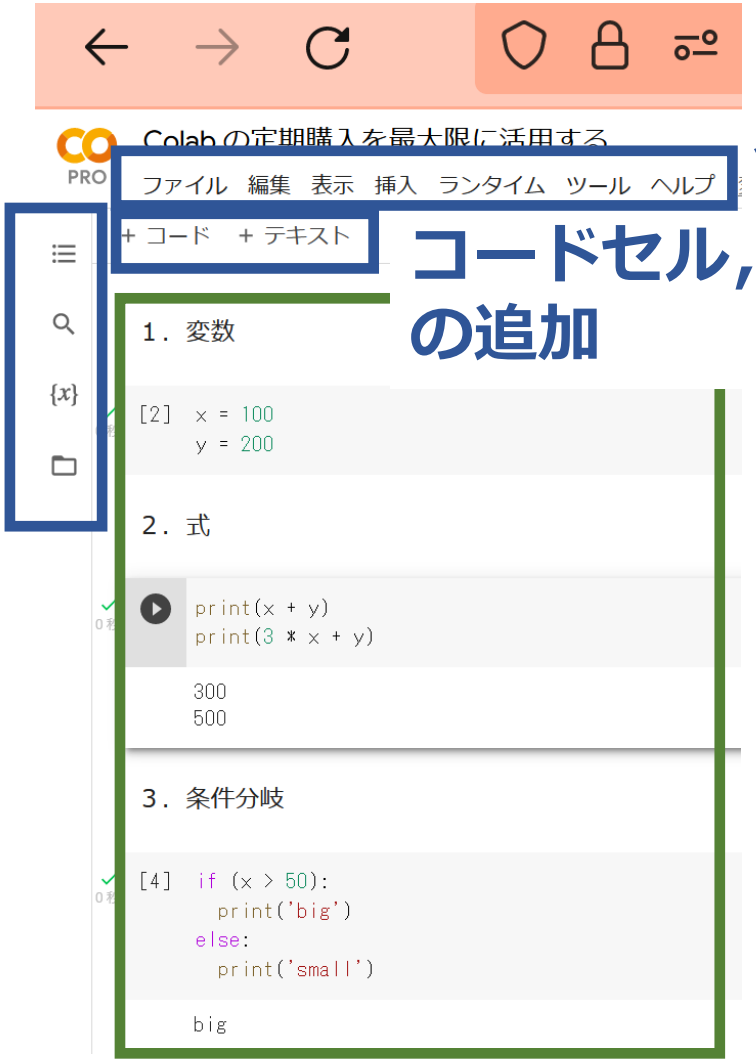




URL: https://colab.research.google.com/

- **オンラインで動く**
- **Python のノートブック**の機能を持つ
- Python や種々の機能が**インストール済み**
- **本格的な利用**には，**Google アカウントが必要**

Google Colaboratory の全体画面



メニュー

コードセル，テキストセルの追加

メニュー
（目次，検索と置換，変数，ファイル）

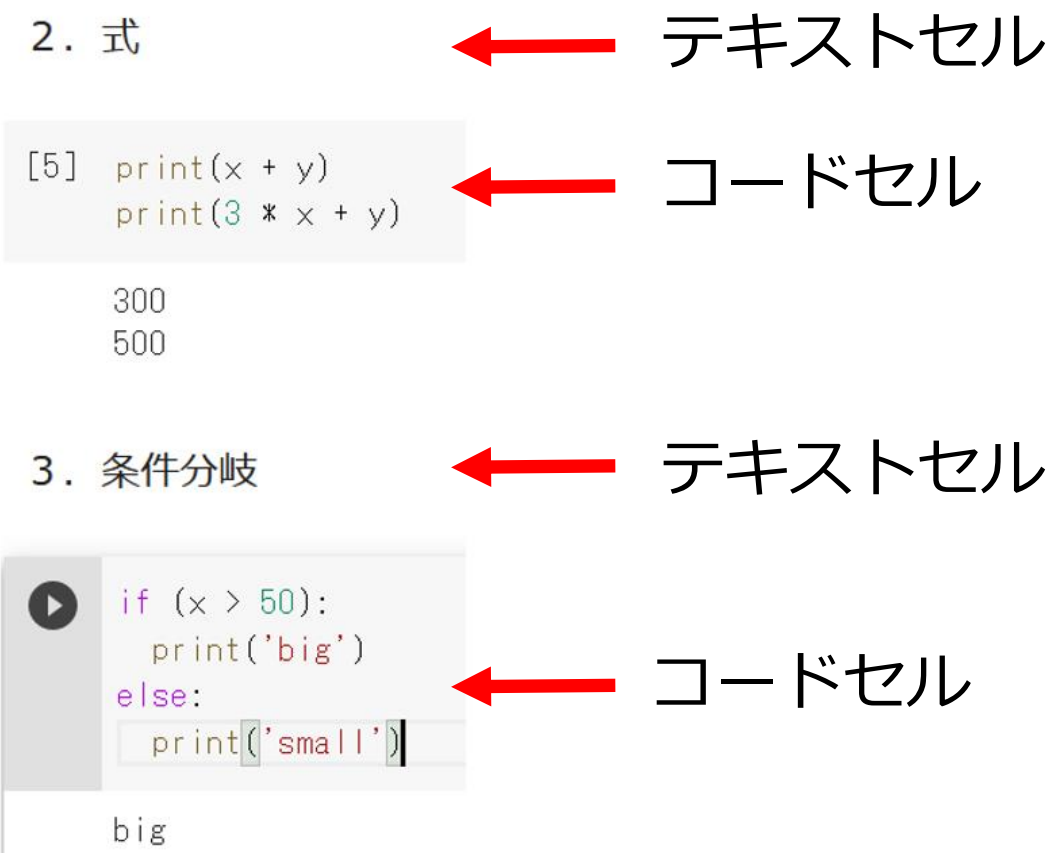
コードセル，テキストセルの並び

Web ブラウザの画面

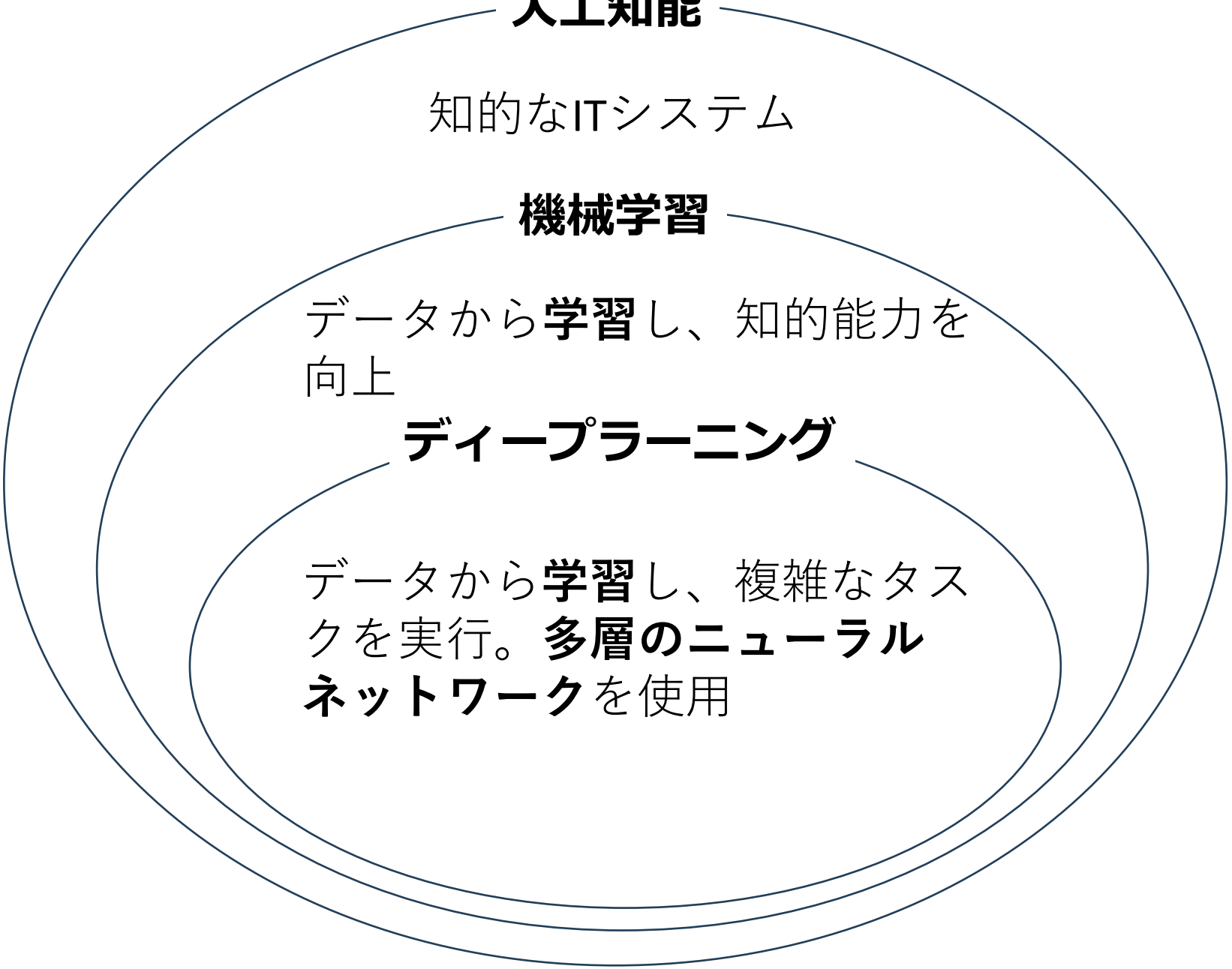
Google Colaboratory のノートブック

コードセル，**テキストセル**の２種類

- **コードセル**： Python プログラム，コマンド，実行結果
- **テキストセル**：説明文，図



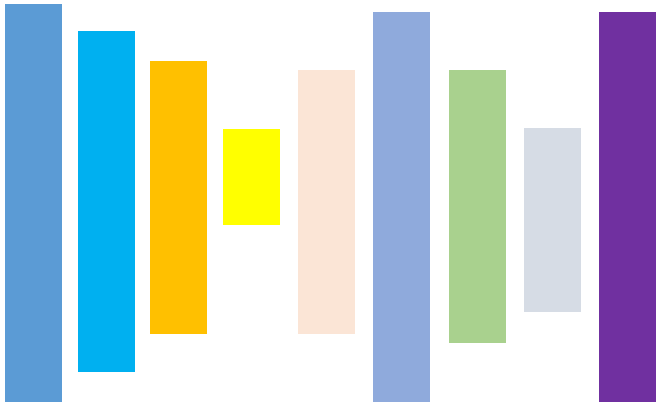
13-1. イントロダクション


人工知能
知的なITシステム
機械学習
データから**学習**し、知的能力を向上
ディープラーニング
データから**学習**し、複雑なタスクを実行。**多層のニューラルネットワーク**を使用

ディープラーニング

ディープラーニングに「**ディープ**」とついているのは、**多層のニューラルネットワーク**を使用するため

層の数が少ない



層の数が多い（ディープ）

ディープラーニングまとめ

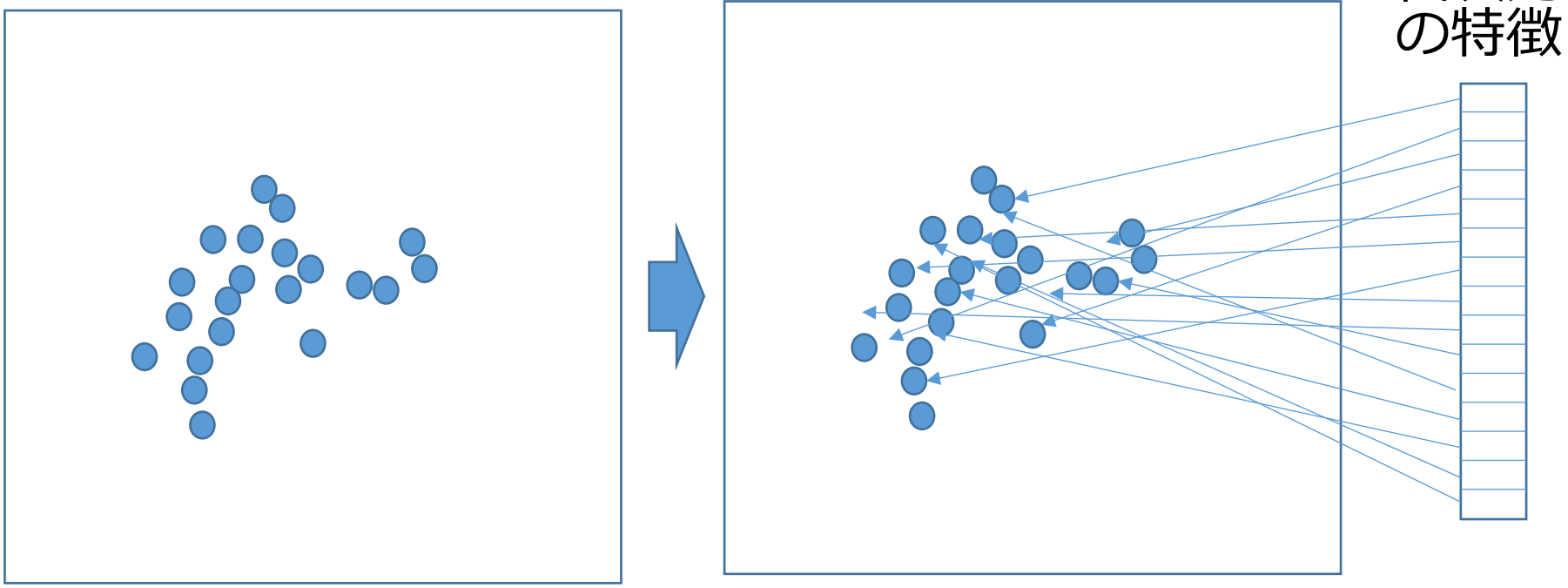
- **ディープラーニング**は**機械学習**の一種であり、人工**ニューラルネットワーク**を使用して**データから学習**し、複雑な**タスクを実行**する技術

- 「ディープ」の名前は、**多層のニューラルネットワーク**を使用することに由来

- ディープラーニングが広く利用される理由は、**多様なデータに適用**でき、**さまざまなタスク**で高性能を発揮するため。
  例：**画像認識**、**自然言語処理**、**音声認識**など。

自己符号化（オートエンコード）

訓練データ



入力データを**低次元の特徴に圧縮**し、その後、その低次元の特徴から**元のデータを再構築**

自然言語処理

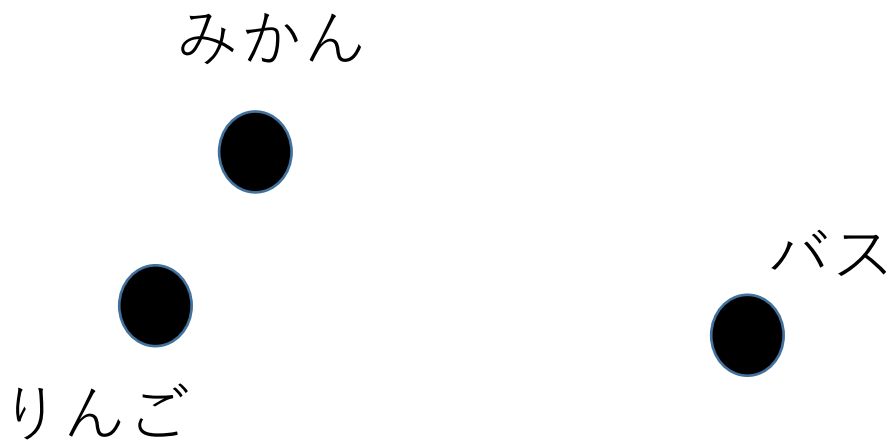
自然言語処理は、**人間が普段使う言語**（日本語、英語など）を**コンピュータが理解したり、生成する技術**

【自然言語処理のさまざまな応用】

情報検索、AIとの対話、AIへの指示、プログラミング支援、人間の指示による文書の作成や推敲、翻訳、要約

単語の特徴ベクトル

- **単語の特徴ベクトル**は，**単語を数値化**したもの
  このとき，複数の数値の組（**ベクトル**）を考える
- **意味の近い単語**は**近く**になるようにする。

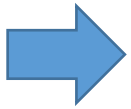


単語を**数値化**し，**単語の特徴ベクトル**を得ている

単語

iPhone

数値化



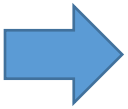
```
[-0.47851562 -0.23242188 -0.14746094  0.06787109 -0.23632812  0.11669922
  0.14550781 -0.09765625  0.15136719  0.11865234 -0.30859375 -0.14648438
 -0.19726562 -0.40039062  0.17578125  0.20507812  0.28710938  0.10449219
  0.08300781 -0.23925781  0.08349609 -0.03857422  0.265625    0.0534668
 -0.31445312  0.30664062 -0.25195312  0.34960938 -0.00582886 -0.25195312
 -0.02368164 -0.06591797 -0.13574219  0.26953125 -0.06298828  0.05493164
 -0.4609375  -0.06787109  0.15429688  0.0546875  -0.01312256  0.06982422
  0.08984375  0.21386719  0.07470703 -0.21972656  0.203125   -0.33007812
 -0.20410156 -0.30859375 -0.08349609  0.02246094  0.20703125  0.12597656
  0.07226562  0.1640625  -0.15039062 -0.0534668   -0.0534668   0.10791016
  0.05004883 -0.03100586 -0.17480469  0.13574219 -0.41601562  0.12597656
 -0.30273438 -0.04492188 -0.25976562  0.01599121  0.09277344  0.14941406
 -0.10058594 -0.0030365  -0.33203125 -0.20996094  0.04980469  0.11035156
 -0.10498047 -0.37890625 -0.31445312  0.328125   -0.24804688 -0.140625
 -0.08789062 -0.10791016  0.04174805  0.34375    -0.21679688  0.10693359
 -0.29882812  0.20507812 -0.25195312 -0.28125     0.22753906 -0.16894531
 -0.14160156 -0.07421875  0.6640625   0.14550781 -0.02331543 -0.19042969
 -0.01928711  0.19042969  0.13476562  0.20996094 -0.00512695 -0.140625
  0.27734375 -0.34765625 -0.2734375   0.13769531  0.08886719  0.09960938
 -0.00854492  0.13769531 -0.15917969 -0.75        0.12792969  0.03222656
  0.07763672 -0.33203125 -0.36914062 -0.04443359  0.5234375   0.19335938
 -0.20800781  0.14355469 -0.30859375  0.14746094 -0.11279297 -0.02392578
 -0.13769531  0.04345703 -0.07470703 -0.07617188  0.06225586  0.09277344
  0.09375    -0.12109375  0.23925781 -0.24902344 -0.31054688 -0.14453125
 -0.12695312  0.03320312  0.07080078 -0.02526855 -0.3046875  -0.38671875
  0.19140625 -0.53125     -0.03100586  0.05249023 -0.16308594 -0.20800781
 -0.11621094  0.1640625  -0.13769531  0.22070312 -0.3984375  -0.05981445
 -0.26367188 -0.29101562 -0.08984375 -0.25        0.16308594 -0.06884766
 -0.27539062  0.15039062  0.02197266  0.01696777 -0.36914062 -0.11962891
 -0.3984375  -0.0145874   0.515625   -0.55859375  0.12597656 -0.31054688
 -0.31835938  0.01104736 -0.07470703  0.06152344  0.24023438  0.10009766
 -0.24804688  0.13183594 -0.10205078 -0.19824219 -0.09912109 -0.0402832
 -0.65234375 -0.08203125  0.02368164  0.4765625  -0.37890625  0.03100586
 -0.11132812 -0.29882812 -0.41796875 -0.17089844 -0.34960938  0.05517578
  0.03417969  0.08300781 -0.09082031 -0.421875   -0.11425781  0.24707031
  0.08056641  0.28710938 -0.17578125 -0.2265625  -0.38671875  0.05981445
 -0.02502441 -0.1640625   0.12158203 -0.16210938 -0.26757812  0.16308594
  0.09667969  0.28320312  0.0534668  -0.140625   -0.20605469  0.16210938
 -0.18652344 -0.04248047 -0.08447266 -0.16796875  0.04467773 -0.29296875
  0.00671387 -0.13378906 -0.04003906 -0.28710938  0.18457031  0.21484375
  0.328125    0.15429688 -0.234375   -0.25585938 -0.19433594 -0.00405884
 -0.16992188  0.33007812 -0.21679688 -0.01403809 -0.14257812  0.265625
  0.20800781  0.18066406  0.25195312  0.00765991 -0.02685547 -0.046875
 -0.00811768 -0.03320312 -0.05737305 -0.20117188  0.20703125 -0.02026367
 -0.29882812  0.296875   -0.23632812 -0.05737305 -0.05712891 -0.26367188
  0.125       0.09423828 -0.16796875  0.2109375   0.09130859 -0.03881836
  0.34570312 -0.00436401 -0.28515625  0.13671875 -0.14257812  0.25976562
  0.03833008  0.22070312 -0.20800781 -0.02478027 -0.05932617 -0.03320312
  0.12353516  0.13476562  0.08642578  0.22460938 -0.25195312 -0.20898438
  0.22265625 -0.08691406 -0.359375   -0.07763672 -0.23242188  0.1875    ]
```

多数の数値の組
（ここでは 300個の数値）

単語

iPad

数値化



```
[-0.01696777 -0.11914062 -0.32226562 -0.02441406 -0.26171875  0.03515625
  0.11572266  0.28710938  0.28320312  0.01556396  0.00153351  0.15722656
  0.24707031 -0.23535156  0.13671875  0.16796875  0.2890625  -0.03735352
  0.19042969 -0.13867188  0.00442505 -0.171875    0.22851562  0.0625
 -0.11035156  0.26953125  0.16015625  0.31445312 -0.04443359 -0.32421875
 -0.17578125 -0.00479126 -0.30664062  0.17871094 -0.02783203 -0.08203125
 -0.3046875   0.01062012  0.046875    0.2734375   0.10986328  0.08007812
  0.09521484  0.01611328 -0.00872803 -0.03955078 -0.24023438  0.12451172
  0.09619141 -0.31445312 -0.01031494 -0.04443359 -0.18945312 -0.03710938
  0.07861328  0.19042969 -0.0703125  -0.01599121  0.04956055  0.11962891
  0.09570312  0.10351562 -0.33203125 -0.09716797 -0.38671875  0.16210938
 -0.20996094  0.11425781  0.00158691  0.15332031 -0.12695312 -0.12988281
 -0.00750732  0.02038574 -0.04663086 -0.20507812 -0.23828125 -0.04614258
  0.10595703 -0.26757812 -0.03613281  0.13867188 -0.2578125   0.15136719
 -0.19140625 -0.26367188  0.4140625   0.45507812 -0.14257812 -0.15136719
 -0.1953125  -0.14160156 -0.22851562 -0.16503906  0.02929688 -0.26953125
  0.16699219  0.18066406  0.30664062  0.07910156  0.00387573 -0.30859375
 -0.02111816 -0.03857422  0.31835938  0.12792969  0.00805664 -0.03222656
  0.38867188 -0.16113281 -0.05053711  0.02197266  0.1328125   0.23046875
  0.02941895 -0.02575684 -0.33984375 -0.4609375  -0.05395508  0.02624512
 -0.07568359 -0.375      -0.27539062  0.09814453  0.3203125  -0.0255127
 -0.11669922  0.07714844 -0.51953125  0.06689453 -0.00531006  0.04370117
 -0.23046875  0.08203125 -0.0378418   0.18457031  0.13867188  0.10302734
  0.2109375  -0.10839844  0.13476562  0.06933594 -0.06640625 -0.26367188
  0.06933594  0.05175781  0.3359375   0.01000977 -0.37695312 -0.3515625
  0.0625     -0.3359375  -0.10742188  0.12792969  0.10449219 -0.36914062
 -0.27734375 -0.31054688 -0.30078125  0.39648438 -0.3515625  -0.31640625
 -0.0534668  -0.16796875 -0.27539062  0.13574219  0.2890625   0.125
  0.00653076  0.04833984 -0.04052734 -0.13769531 -0.41796875  0.09912109
 -0.29492188  0.27148438  0.29492188 -0.3828125  -0.25        -0.12890625
 -0.3671875  -0.171875   -0.08154297 -0.26171875  0.1953125   0.02148438
 -0.18554688  0.04345703 -0.09716797  0.04223633  0.09228516  0.14550781
 -0.37109375 -0.0402832  -0.17480469  0.19824219 -0.16503906  0.03540039
 -0.06542969 -0.31640625 -0.14160156  0.02600098 -0.27148438  0.11669922
  0.06347656  0.22753906 -0.02502441 -0.16503906 -0.13378906  0.08837891
  0.05810547  0.32617188 -0.19726562 -0.10595703 -0.09228516 -0.01647949
  0.17480469 -0.08984375 -0.1328125  -0.27734375 -0.13671875  0.31445312
 -0.05957031  0.08105469 -0.02294922  0.06445312 -0.29882812  0.12207031
  0.08007812 -0.00300598 -0.28710938 -0.06689453  0.00604248 -0.2734375
  0.12695312 -0.06640625  0.01611328 -0.14160156  0.06640625  0.24121094
  0.12255859 -0.00909424 -0.12255859 -0.36523438 -0.203125   -0.22851562
 -0.2734375   0.10839844  0.00668335  0.00747681 -0.31835938  0.09033203
  0.11816406  0.09765625  0.125       0.03442383  0.02038574  0.12158203
 -0.23925781  0.06933594  0.13183594 -0.07373047  0.08251953  0.09667969
 -0.13769531  0.23535156 -0.20996094 -0.2890625  -0.04638672 -0.15820312
  0.21972656  0.29492188 -0.06347656  0.03222656  0.16992188 -0.09619141
  0.40234375 -0.01806641 -0.0612793   0.11962891 -0.00196838  0.25
 -0.15234375  0.18164062  0.02453613  0.04858398 -0.20507812  0.00245667
  0.13671875  0.5        -0.14746094 -0.13476562 -0.265625   -0.1953125
  0.4375     -0.00512695 -0.12890625 -0.04907227 -0.2421875   0.19335938]
```

多数の数値の組
（ここでは 300個の数値）

13-2. 画像生成のバリエーション

顔画像の生成①

実在しない人間の顔画像を**生成**

研究成果はオンラインで公開されている

tl-GAN のページ, https://docs.google.com/presentation/d/1OpcYLBVpUF1L-wwPHu_CyKjXqXD0oRwBoGP2peSCrSA/edit#slide=id.g4551faa5ed_0_208

顔画像の生成②

実在しない人間の顔画像を**生成**

実在

偽物

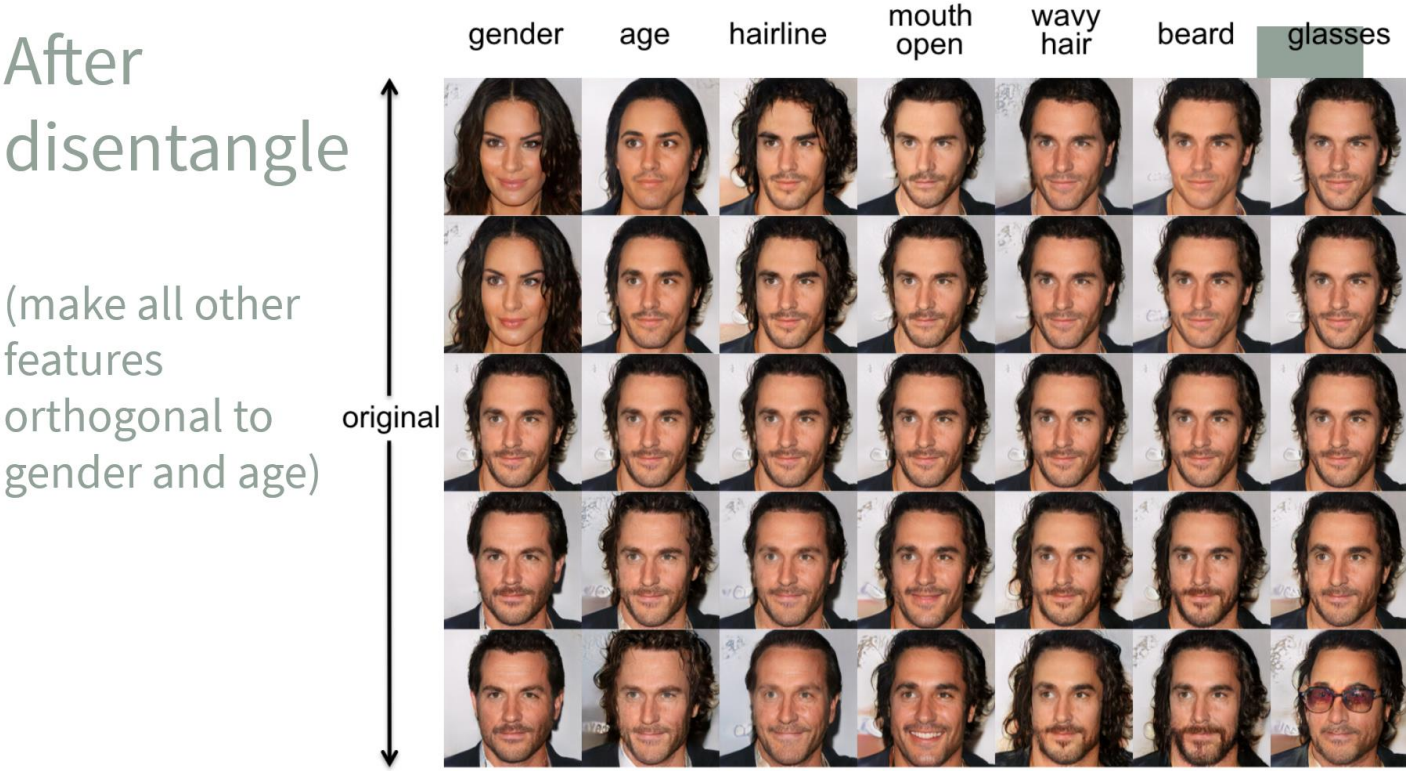
Web ブラウザで動く

https://www.whichfaceisreal.com/

様々な特徴の顔画像の生成

実在しない人間の顔画像を**生成**

年齢，髪量，口の開き具合，髪の波うち，眼鏡などさまざまな**特徴に応じた顔**を**生成**

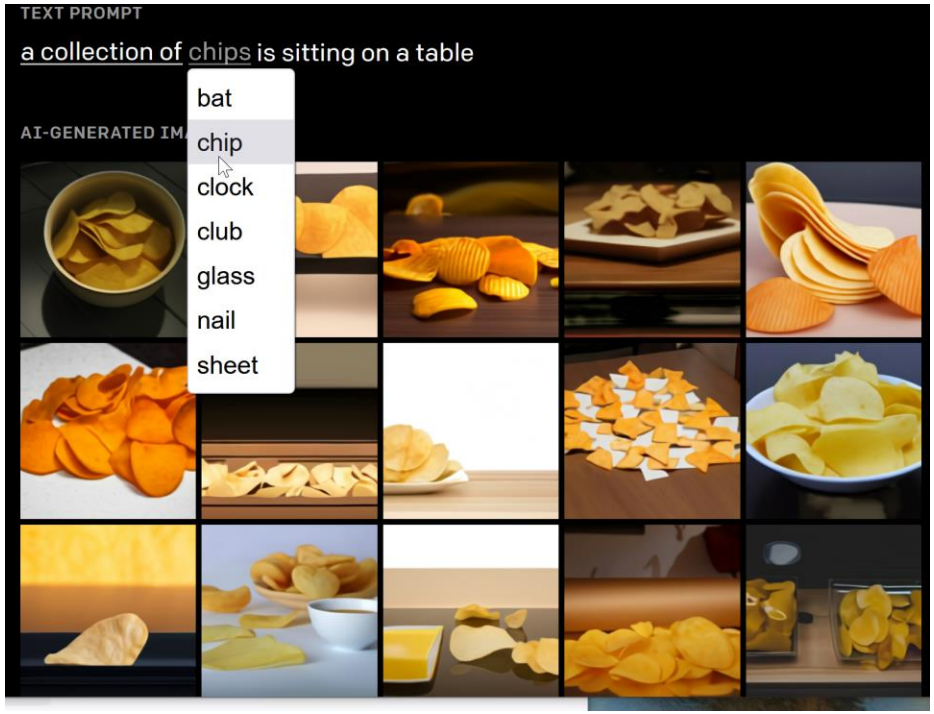


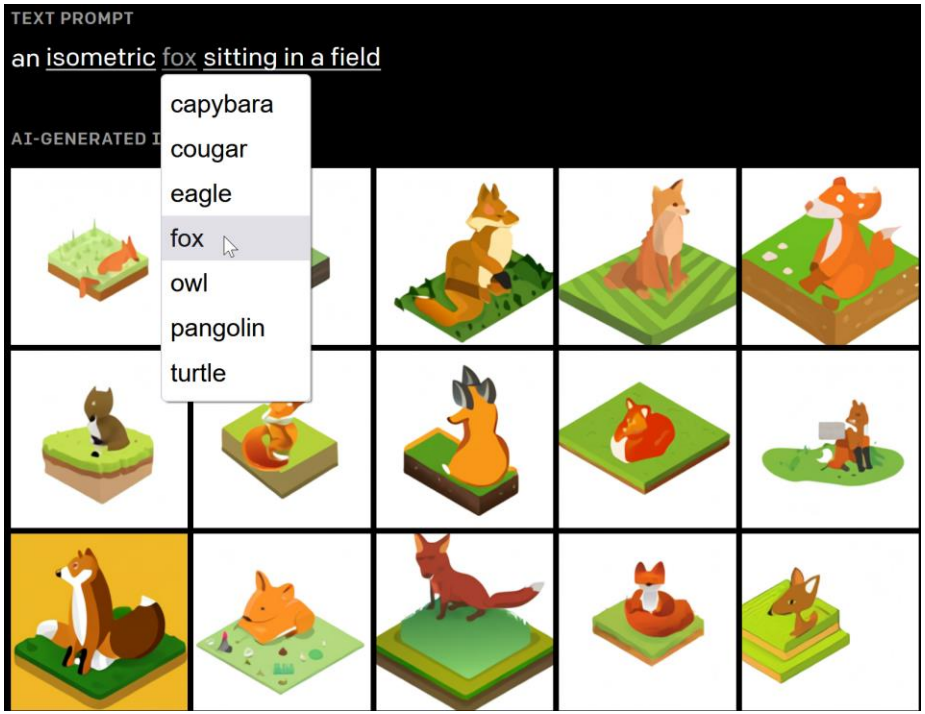
研究成果はオンラインで公開されている

tl-GAN のページ, https://docs.google.com/presentation/d/1OpcYLBVpUF1L-wwPHu_CyKjXqXD0oRwBoGP2peSCrSA/edit#slide=id.g4551faa5ed_0_208

単語からの画像生成　DALL E

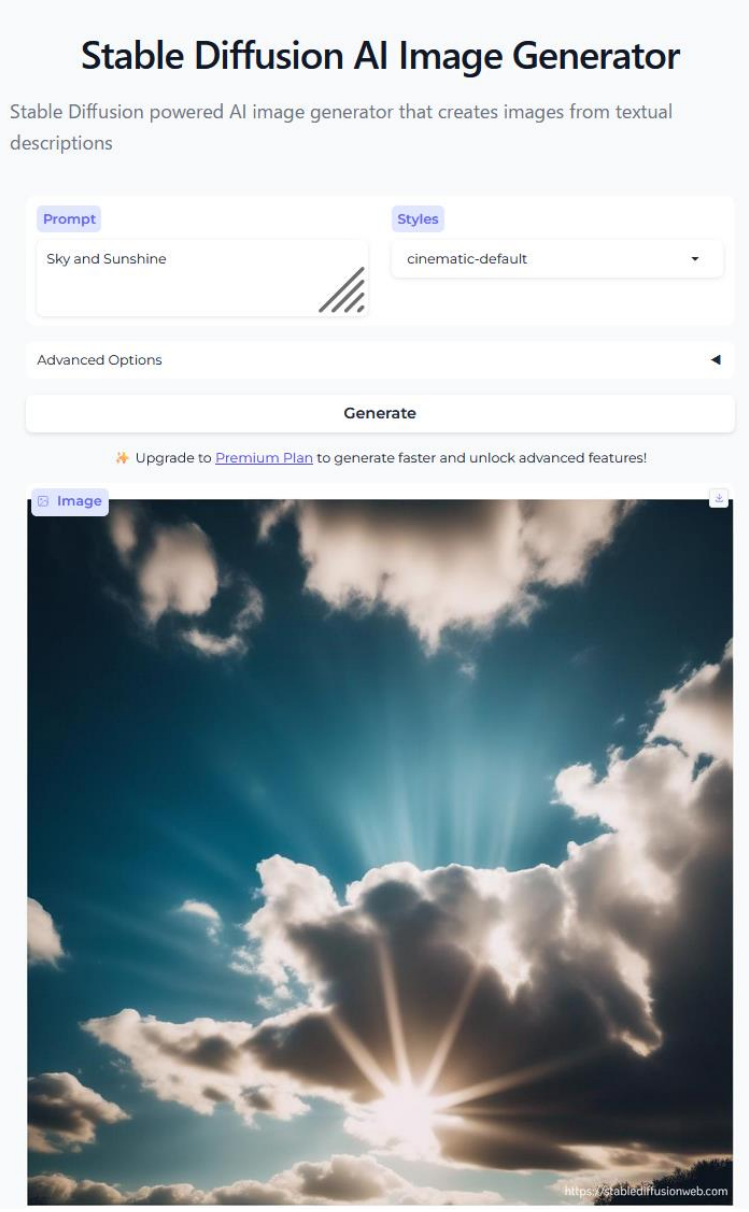
「chip」、「fox」などの**単語を指定**しての
写真の画像生成，イラストの画像生成 DALL E





研究成果，プログラムのソースコードはオンラインで公開されている
https://openai.com/blog/dall-e/

プロンプト（言葉による指示）からの画像生成
Stable Diffusion Online



画質改善

暗い画像をもとに，明るい画像を画像生成

研究成果，プログラムのソースコードはオンラインで公開されている
https://github.com/VITA-Group/EnlightenGAN

文献 EnlightenGAN: Deep Light Enhancement without Paired Supervision
Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, Zhangyang Wang

超解像

低解像度の画像をもとに、**高解像度**の画像を生成

処理前

処理後

線画からの画像生成

人間が描いた**領域図**や**線画**をもとに，画像を画像生成

研究成果はオンラインで公開されている

Video-to-Video Synthesis のページ

https://www.youtube.com/watch?v=S1OwOd-war8

ここまでのまとめ

人間の顔画像生成

- 実在しない人物の顔画像を生成
- 年齢や髪の特徴などを変更して多様な顔画像を生成

単語やプロンプトからの画像生成

- 与えられた単語（例：「chip」、「fox」）から写真やイラストの画像を生成

画質改善と超解像

- 画質改善：暗い画像を明るく
- 低解像度の画像を高解像度に変換

線画からの画像生成

- 人間が描いた線画をもとにした画像生成

GAN による画像生成の応用例

- 写真のようなリアルな画像の生成
- 画像の品質改善
- 既存の画像データの増量と、AIの学習での利用

「写真は信用できる」という**常識**が**変容する**

写真

＋

ビデオ

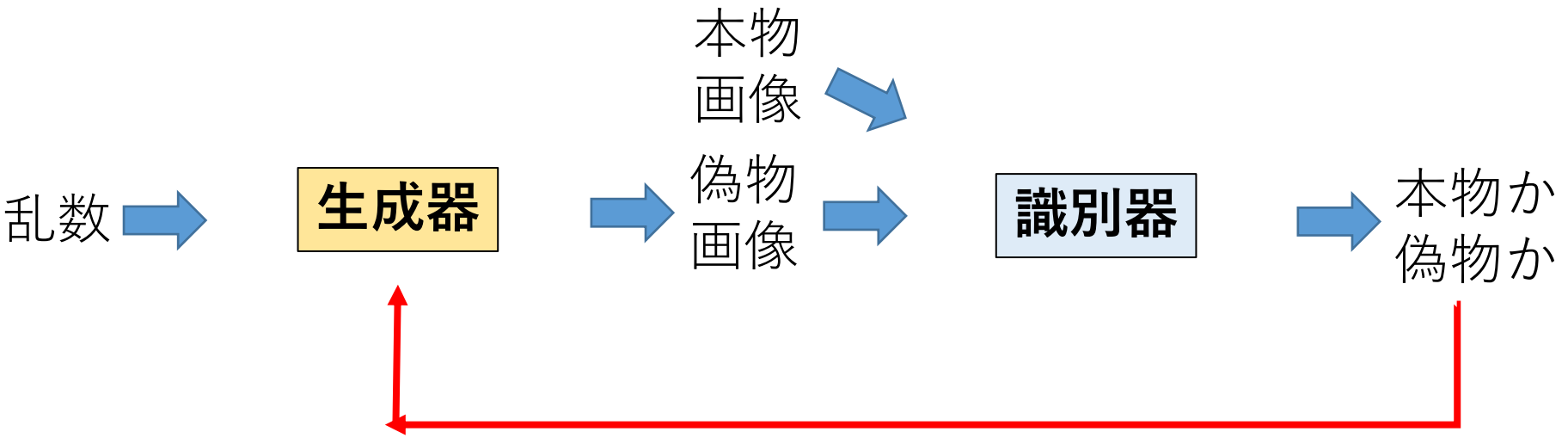
→

画像生成されたビデオ

13-3. GAN の概要

GAN の要点

- **GANは、生成器**と**識別器**という**２つのニューラルネットワーク**が**相互に競合しながら学習**する構造



GANによる画像生成の要点

- **GAN は、生成型敵対ネットワーク**（Generative Adversarial Networks）の略
- GANは、実際には存在しないが、**本物のように見えるデータを生成**
- **生成器**と**識別器**という**２つのニューラルネットワーク**が**相互に競合しながら学習**する構造
  - **生成器**: **リアルな画像を生成**
  - **識別器**: **生成された画像**が**本物**（実際のデータセットからの画像）か**偽物**（生成器によって生成された画像）かを**判別**
  - **学習**により、**生成器**は**リアルな画像を生成する能力**を向上。**識別器**は**本物と偽物の画像を識別する能力**を向上。

生成器

- **ランダムなノイズ（乱数）からの画像生成**：生成器は、乱数を入力とし、リアルな画像に変換
- **学習による改善**：生成器は、繰り返し学習を通じて、**識別機に偽物と見破られない**ような、よりリアルな画像を生成
- **ディープニューラルネットワークの使用**

識別器

- **本物と偽物の画像を判別**
- **学習による改善**：識別器は、繰り返し学習を通じて、**本物の画像と偽物の画像**をより**正確に判別**する能力を向上
- **ディープニューラルネットワークの使用**

演習１．学習済み PGAN を用いた顔画像の生成

【トピックス】

- **GAN**
- **生成器よる画像生成**

PGAN (Progressive Growing of GANs)

- **GAN の一種：** 2 つのネットワーク（生成器と識別器）が相互に競合しながら学習する構造
- **2016年発表**
- **高解像度の画像生成**ができる当時としては画期的なもの
- ソースコードと学習済みモデルが公開されており、簡単に試すことができる

演習１で行うこと

- **PGAN の使用。生成器を用いて画像生成。**
- celebAHQ-512 の**「本物画像」で学習済み**の学習済みの PGAN を使用
- 学習済みモデルが公開されており、簡単に試すことができる

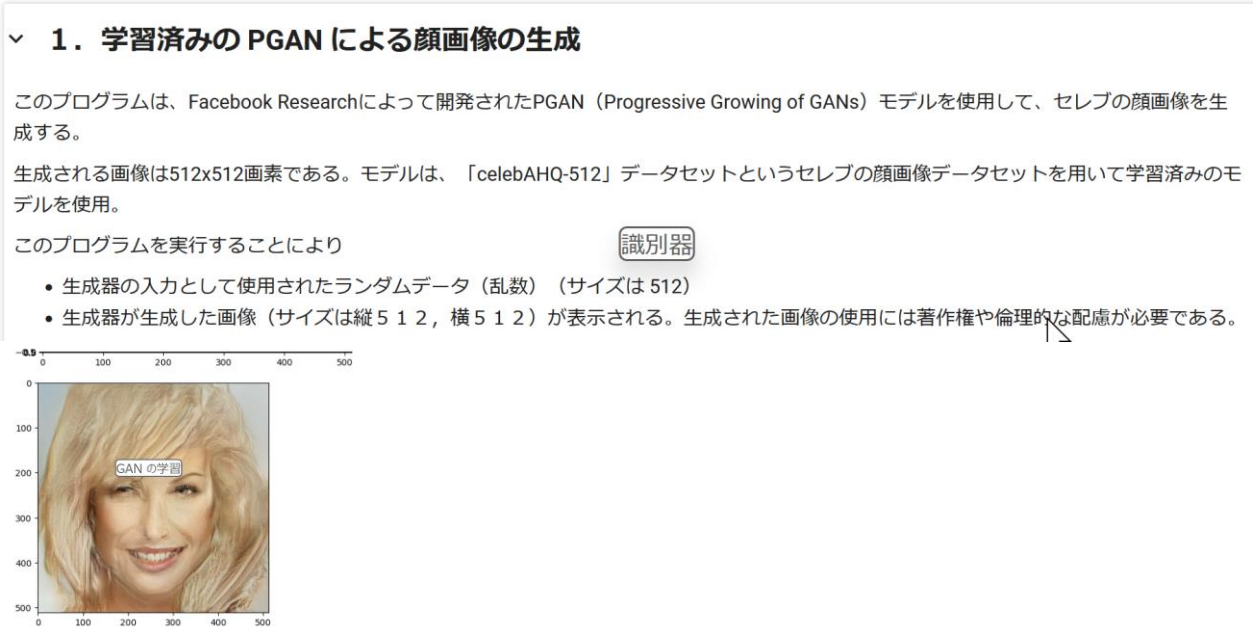
① Google Colaboratory のページを開く

https://colab.research.google.com/drive/1_UT_Vf2cCaGW8DRKpWpYAVSUGiMkQu5Q?usp=sharing

（このページは、演習１，２で使用する）

② **次**について、プログラムや説明や実行結果が掲載されていることを確認。各自でよく読む。

１．学習済みのPGANによる顔画像の生成



③ 次は自習とする

一番上のコードセルを実行するたびに**異なった顔が生成**される

```
import torch
import matplotlib.pyplot as plt
import torchvision

def load_model(use_gpu):
    """PGANの事前学習済みモデルをロードする関数"""
    model = torch.hub.load(
        "facebookresearch/pytorch_GAN_zoo:hub",
        "PGAN",
        model_name="celebAHQ-512",
        pretrained=True,
        useGPU=use_gpu,
    )
    return model

def generate_images(model, num_images=1):
    """指定された数の画像を生成する関数"""
    noise = torch.randn(num_images, 512)
    with torch.no_grad():
        return noise, model.test(noise)

def display_images(images):
    """画像を表示する関数"""
    grid = torchvision.utils.make_grid(images.clamp(min=-1, max=1), scale_each=True, normalize=True)
    plt.imshow(grid.permute(1, 2, 0).cpu().numpy())
    plt.show()
```

識別器

演習２．PGAN の構造

【トピックス】

- **GAN の構造**
- **生成器**
- **識別器**

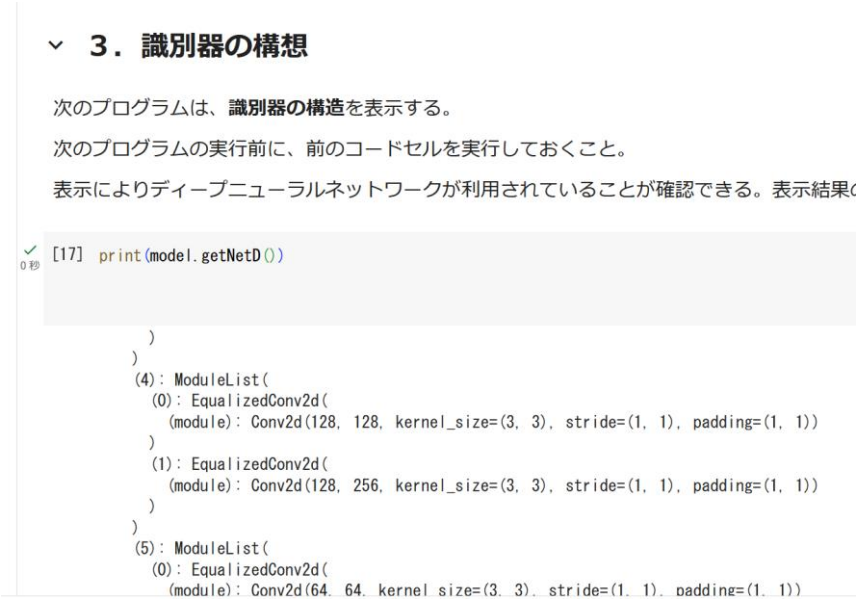
① Google Colaboratory のページを開く

https://colab.research.google.com/drive/1_UT_Vf2cCaGW8DRKpWpYAVSUGiMkQu5Q?usp=sharing

（このページは、演習１，２で使用する）

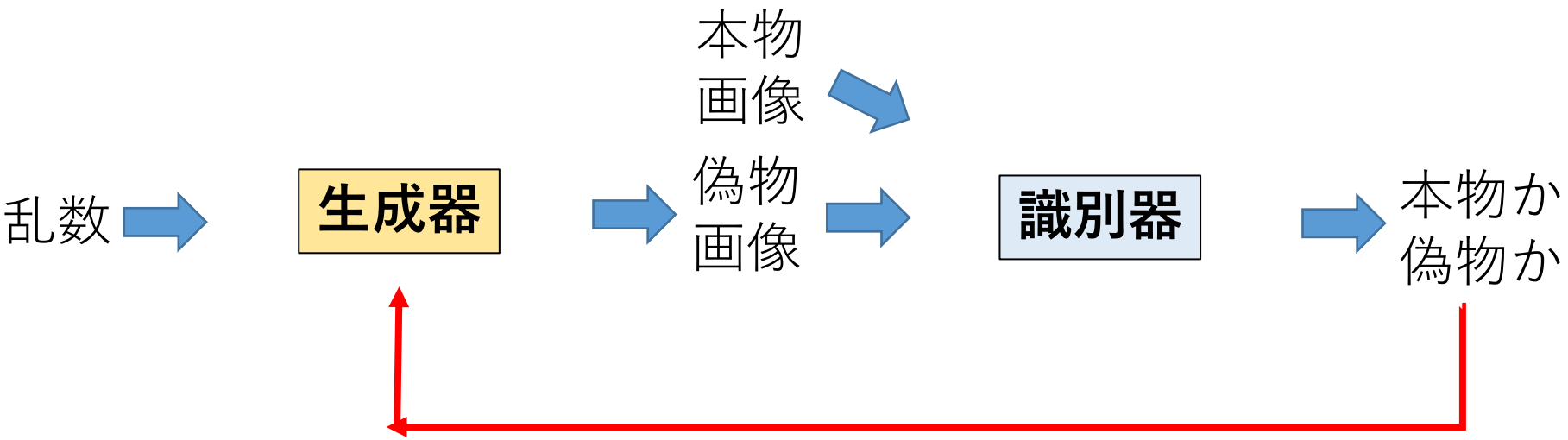
② **２と３**について、プログラムや説明や実行結果が掲載されていることを確認。各自でよく読む。

●次の２つのコードセルについて、説明と実行結果を各自確認



ここまでのまとめ

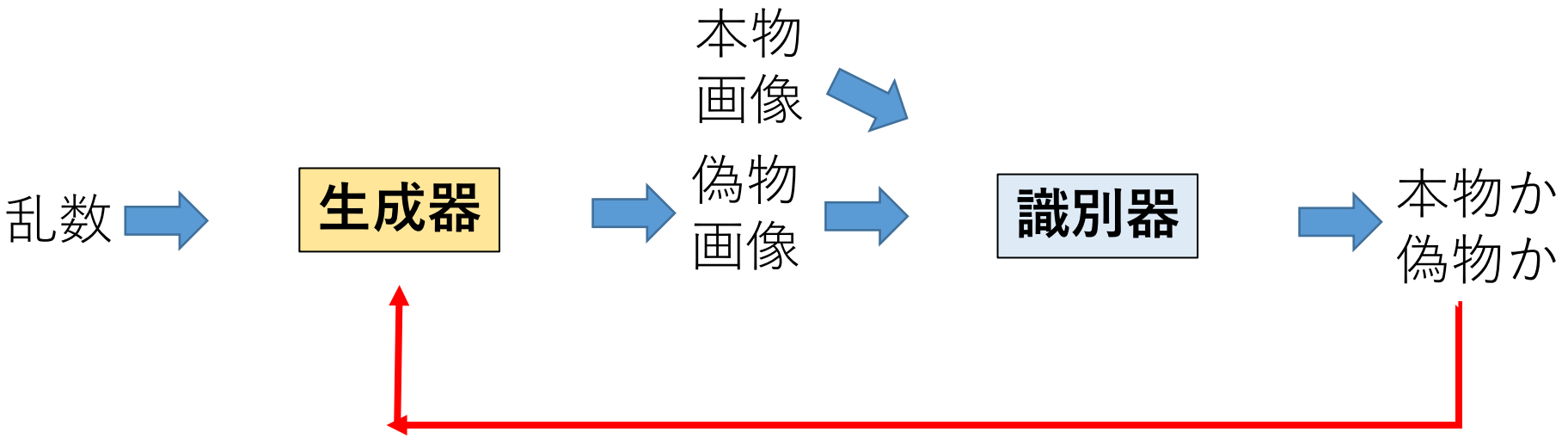
- **GANは、生成器**と**識別器**という**２つのニューラルネットワーク**が**相互に競合しながら学習**する構造
- **生成器**：ランダムなノイズ（乱数）から**リアルな画像**を**生成**
- **識別器：画像が本物か偽物化を判別**
- celebAHQ-512などの「本物画像」で学習済みのPGANを使用して、画像生成が可能



13-4. GAN の仕組み

GAN の学習

- 生成器の役割：リアルな画像を生成
- 識別器の役割：**生成器**が作った画像を**偽物**と判別し、**本物の画像**を**本物**と判別
- **相互作用による学習**：生成器は、識別器をだますような画像を生成する。識別器は、生成器によって作られた画像を偽物と見破る。そのことで、**互いの能力向上**を行う。

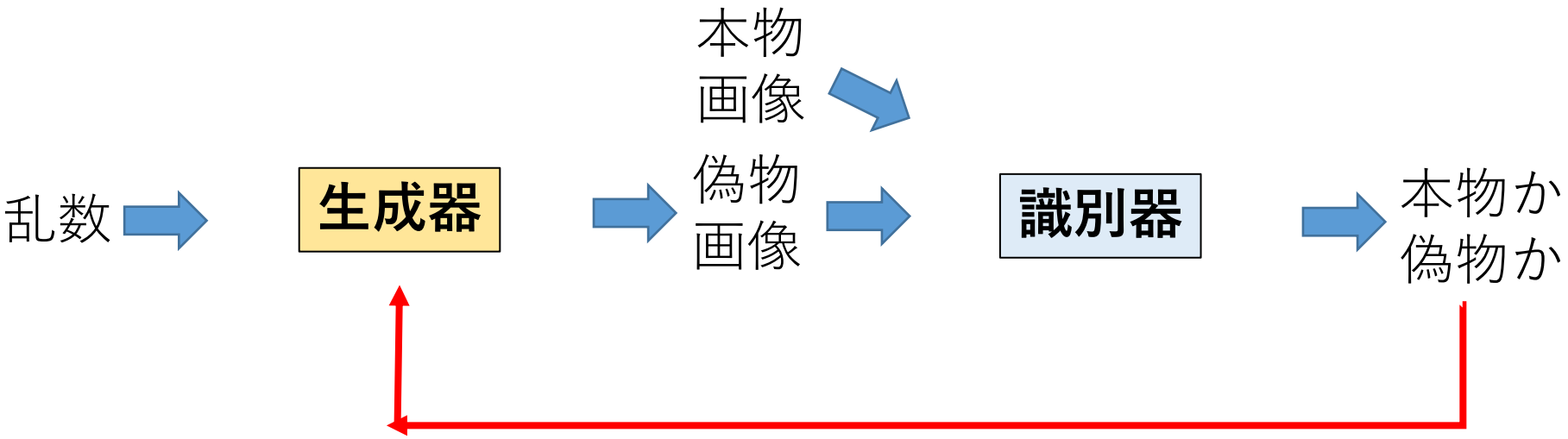


GANの歴史と発展

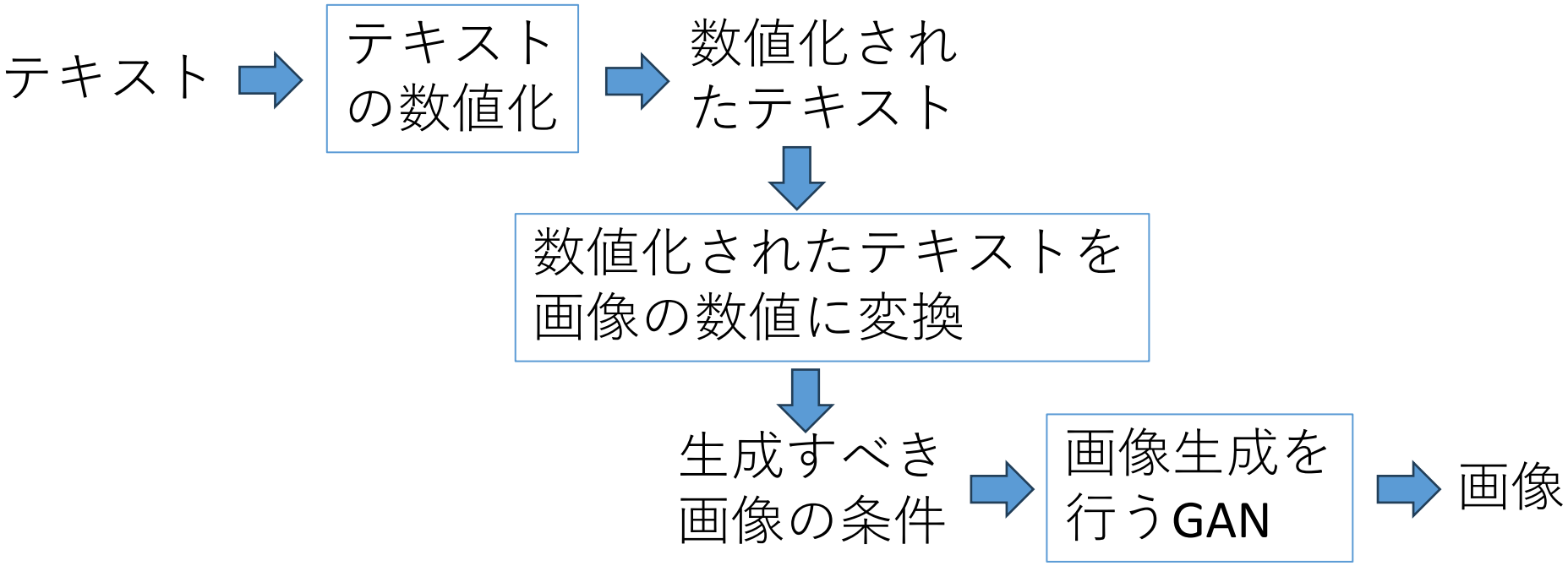
- **GANの提案**
  - イアン・グッドフェローによる提案（２０１４年）
  - 生成器と識別器による GAN を提案
- **GAN技術の進化**
  - 画像の生成
  - 音声の生成
  - 自然言語と画像の組み合わせなど、複雑なデータへの応用
  - 効率の向上
  - 精度の向上

GANの敵対的学習

- **GANでは、生成器と識別器が相互に競合しながら学習を進める。**
- このプロセスを「**敵対的学習**」という。
- 学習の繰り返しにより、生成器と識別器の**両方**の性能を向上



テキストからの画像生成



テキストからの画像生成

- 自然言語処理技術を用いてテキストを数値データに変換
- そのデータを基に画像を生成

ここまでのまとめ

GANの学習プロセス

- **生成器**と**識別器**という**２つのニューラルネットワーク**が**相互に競合しながら学習。**互いの能力を向上させる。
- このプロセスは「敵対的学習」と呼ばれる

GANの発展

- **画像生成**、**音声生成**、**自然言語と画像の組み合わせ**など、複雑なデータへの応用が進んでいる。
- **効率**と**精度の向上**が進んでいる。

テキストからの画像生成

- 自然言語処理技術を用いてテキストを数値データに変換
- そのデータを基に画像を生成

13-5. 自習

自習

- **顔画像で，どちらが実在で，どちらが偽物かのクイズを行う（オンラインデモ）**

https://www.whichfaceisreal.com/

- **テキストからの画像生成（オンラインのデモ） Craiyon (DALL E mini)**

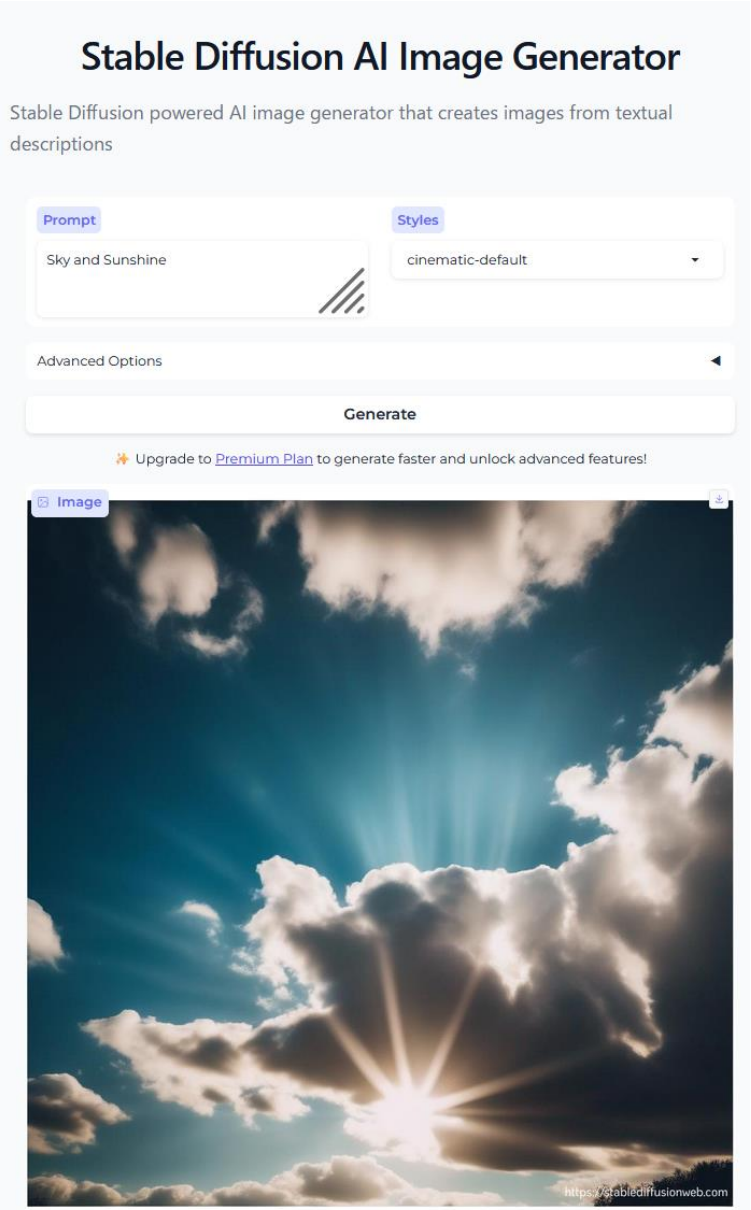
https://www.craiyon.com/

- **プロンプト（言葉による指示）からの画像生成**
  https://stablediffusionweb.com/

- **超解像に関する Google Colaboratory のページ**

https://colab.research.google.com/drive/1oT69ts_qzr1xrPYguSepcl24QaQv5AYq#scrollTo=mlVW1_628s0G

プロンプト（言葉による指示）からの画像生成
Stable Diffusion Online



① Stable Diffusion Online のサイト

https://stablediffusionweb.com/

② Get Started Now をクリック

③　プロンプトを英語で与える。創造力、発想力を発揮

「**Prompt**」の下に、プロンプトを**英語**で入れて、「Generate」をクリック

結果が出るまで**１分以上**待つ。

思い通りの結果を得るためにプロンプトを工夫する．プロンプトは具体的に。

超解像

Google Colaboratory のページ

https://colab.research.google.com/drive/1oT69ts_qzr1xrPYguSepcl24QaQv5AYq#scrollTo=mlVW1_628s0G

処理前　　処理後

全体まとめ

GANの概要

- **構造**：**生成器**と**識別器**という**2つのニューラルネットワーク**から構成
- 生成器：ランダムなノイズからリアルな画像を生成
- 識別器：生成された画像が本物か偽物かを判別

GANの学習プロセス

- **敵対的学習**：生成器と識別器が相互に競合しながら学習し、互いの性能を向上させる。
- **学習を繰り返す**ことで、**生成器**はより**リアルな画像**を、**識別器**は**本物と偽物の識別能力**を向上させる。

GANの応用

- **多様な画像生成**：実在しない人物の顔や、プロンプトからの画像生成など
- **画質改善**：暗い画像を明るくしたり、低解像度の画像を高解像度に変換