令和２年度　特別研究報告書

styleGAN 技術を用いた
顔画像からの衛生マスクの除去

龍谷大学　理工学部　情報メディア学科

T170539　美濃部和也

指導教員　三好力　教授

内容概要

新型コロナウイルスの感染拡大の伴い、衛生マスクの着用する人が増加したことで街頭写真にはマスク姿が多く写り込むようになった。マスク姿からは表情が読み取りにくいことで不安感を抱いでしまい、写真全体の印象が下がってしまう。一方で、衛生マスクをプライバシー保護の目的でも使用されることがあり、安直に素顔を晒すことはできない。そこで本研究では、StyleGAN のエンコーダである pixel2Style2pixel を用いた超解像手法と Style Mixing を組み合わせて、低解像度の顔画像からマスクの除去と顔の特徴変換した高解像度の顔画像の生成を提案する。また、写真からマスクを着用した顔の検出精度を上げるため、畳み込みニューラルネットワーク(CNN)モデルの顔検出モデルを採用し、従来手法と検出精度を比較する。

目次

第1章 はじめに

新型コロナウイルスの感染拡大により、他人との接触の機会がある場所では衛生マスクの着用が当たり前となっている。そのため、街頭風景や講演会など人が多い場所で撮影された写真には、ほとんどの人がマスクを着用している姿が写っており、この光景が奇妙に感じてしまうことがある。これは、写真に写り込む人物のほとんどが衛生マスクを着用している姿が見慣れないということ以外に、マスク姿そのものに原因があると考えられる。田中裕二(2009)は、マスク姿は完全に表情を確認できていないことが不安感や緊張感などを生み出していると述べている[1]。実際、目から表情を読み取ることは不可能ではないが、口より変化が薄いためマスクの上から表情を読み取ることは困難である。この不気味さを払拭するには、表情が読み取れる顔に変換する必要がある。

しかし、衛生マスクを着用するのは、病気や感染対策以外に"隠す"目的で使用されることがある。口元を隠す目的の一つに、プライバシーを守ることが挙げられる。マスク以外にもサングラスや帽子が同じ用途で使われるが、日本では口元を隠すことに抵抗が少ないため、衛生マスクを着用する習慣がある[2]。そのため表情が読み取れるようにした結果、個人を特定できてしまえば肖像権の侵害にあたる可能性が出てくるため、安易に取り除くことができない。

本研究では、写真に写る人物に対して着用する衛生マスクを除去し表情を読み取れる状態にすると共に、マスクを除去した顔からその人物を特定できないように顔のパーツの特徴を変換することを目的とする。この課題に対し顔入れ替えツールや、別人の顔画像との組み合わせなど様々な手法が考えられる。しかしこれらは現実にある特定の顔画像と組み合わせる必要があり、自由に扱える画像を収集するのにコストがかかってしまう。一方で、機械学習の分野で Generative Adversarial Networks(GAN) [3]を用いた画像生成や変換の研究が進んでいる。2018 年には写真で撮ったかのように自然な高解像度画像を生成する StyleGAN[4]が発表された。StyleGAN の研究では、現実に存在しない顔画像を生成することや、別の顔の特徴を自然な形で組み合わせることを可能にしている。実際の研究結果では、サングラスの特徴をなくしペアの顔と特徴変換を行っている。そこで、StyleGAN 技術を用いることでマスクの除去や顔の特徴変換が期待できる。

本研究では、StyleGAN 技術を応用した超解像手法を用いて、マスクの除去と特徴変換を行った顔画像の生成を検討する。超解像手法を用いることで解像度の異なる顔画像からでもマスクの除去が期待できる。また、切り出した顔画像を低解像度に変換し曖昧な画像として処理することで、除去が容易になると考えた。

第2章　先行研究

2.1　GAN

GAN[3]は生成モデルの一種であり、データから特徴を学習することで実在しないデータの生成や、存在するデータの特徴に沿った変換が可能である。

GAN は、Generator(生成器)と Discriminator(判別器)の 2 つのネットワークから成り立っている。GAN のアーキテクチャを図 2.1 に示す。

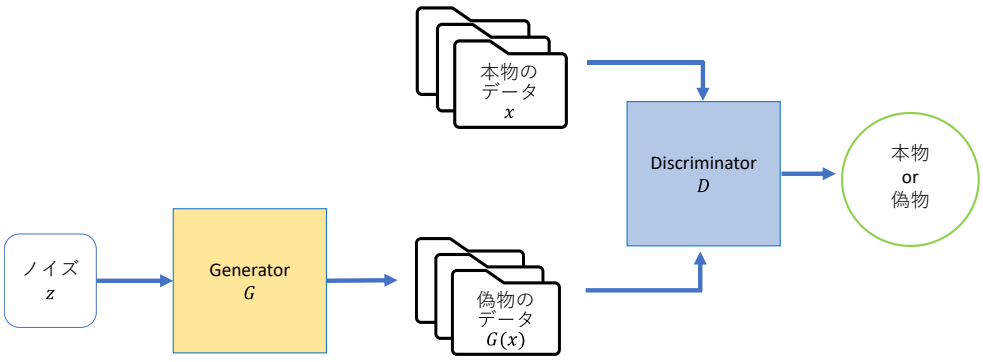

図 2.1 GAN の学習構成

Generator はノイズ(z)から偽の画像を生成し、Discriminator は Generator が生成した画像$G(z)$と本物の画像(x)を判別する役割を持つ。2 つのネットワークの競合関係は、損失関数を共有させることで表現される。Generator は損失関数の値を小さくすることを目的に、Discriminator は損失関数の値を大きくすることを目的に学習させる。

Generator の損失関数の式を 2.1 に示す。

$$L_G = \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D\left(G(z)\right)\right) \tag{2.1}$$

Discriminator の損失関数の式を 2.2 に示す。

$$L_D = \frac{1}{m}\sum_{i=1}^{m} \left[\log D(x) + \log\left(1 - D\left(G(z)\right)\right)\right] \tag{2.2}$$

また学習させた GAN から類似データを生成する場合、Generator にはランダムノイズを入力するが、ここにランダム性をもたせることで、生成されるデータにもランダム性が生まれる。すなわち、サンプルするたびに異なる類似データが生成されることになる。

GAN の欠点は学習の不安定さ、つまり生成されるデータに偏りが生じることである。これは GAN が備える構成の複雑さから生じるため、発生の予見は難しく回避するにはパラメータやネットワーク構成の見直しが必要となる[5][6]。

2.2 StyleGAN

StyleGAN[4]は GAN が派生した手法の一つで、Mapping Network と Synthesis network の 2 つのネットワークで構成されている[7][8]。また、Progressive Growing を用いた高解像画像生成、AdaIN を用いて各層に画像の Style を取り込む、という 2 つの特徴がある。StyleGAN のアーキテクチャを図 2.2 に示す。

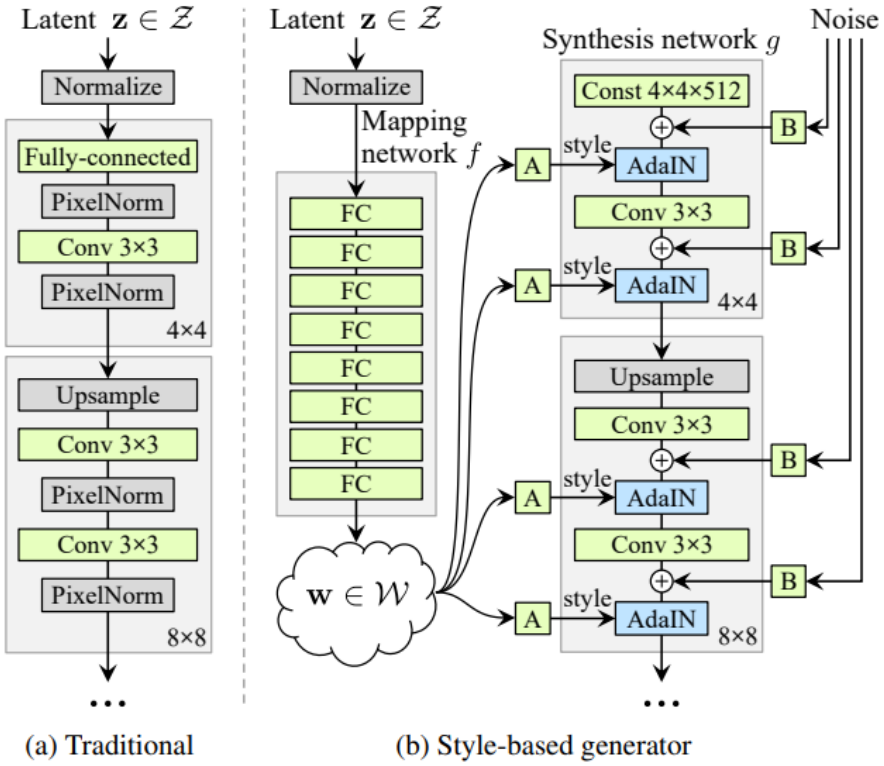

図 2.1 StyleGAN のネットワーク構造([4]より引用)

左の図がこれまでの GAN(PG-GAN)[9]、右の図が StyleGAN である。両者とも段階的に解像度を上げていく Progresive Growing を採用している。しかし、PG-GAN では潜在変数zから画像を生成しているのに対し、StyleGAN では固定の 4×4×512 のテンソルから画像を生成し、潜在変数は Style として取り込んでいるという部分が異なる。また潜在変数はそのまま使用されるのではなく、Mapping network と呼ばれる全結合ニューラルネットによって非線形変換されてから Style として取り込まれている。

Progressive Growing は GAN の学習過程において、低解像度の学習から初めて、モデルに徐々に高い解像度に対応した層を加えながら学習を進めることで高解像度画像の生成を可能にする。初めに 4×4 の学習から始め、次に 8×8 の層を追加というように学習を進め、最終的に 1024×1024 の画像を生成している。このとき、解像度を上げるネットワークを追加しても、Generator と Discriminator はパラメータを固定せずに学習させ続けることが特徴である。

Mapping Network では 8 層の全結合層によって潜在変数を潜在空間に非線形変換している。これは入力時には情報的な意味を持たないただの乱数の数列（潜在変数）を多次元的

なスタイル情報（年齢、性別、表情 etc）を表す空間 W に数値をマッピングし、スタイルの特徴を表すものに変換するという処理である。

Synthesis Network は 18 層で構成されており、これらを CNN で特徴マップをアップサンプリングすることで画像を生成する。各層に 4×4×512 の固定のテンソルと Mapping Network から得られた潜在変数を AdaIN で正規化されたスタイル情報とランダムなノイズ情報を入力する。アップサンプリングを何度にも分け、その都度スタイル情報を挿入する。これは画像のサイズによって持つ情報が異なるためである。高解像度画像では画像の特徴は非常に分散しており 1 ピクセルから得られる情報は極めて少なく、細かいものである。対して低解像度画像が表す 1 ピクセルの情報は画像最大の特徴（全体の色味）を表し、一番アバウトな情報である。つまり、画像サイズが大きくなるに従って、物体の色→物体の位置→物体の輪郭→物体の細かな模様と順番にアバウトな情報から細かい情報を表す物へと変化していくわけである。低解像度画像にスタイル情報を挿入すれば全体の色味等が操作でき、高解像度画像にスタイル情報を挿入すれば表情や服の模様などを操作することが可能である。

Synthesis Network で用いるノイズは、正しい法則に沿って生成すれば、ランダムに生成しても画像の見た目に大きな影響を与えない特徴である。具体的には細かな見た目の特徴である髪の流れやヒゲ、そばかすなどに確率的な変動を与える。逆にノイズを取り除けば、特徴のない絵画のような見た目になる。また、ノイズにおいても正規化する画像サイズによって影響の細かさをコントロールすることが出来る。

AdaIN は、スタイル変換の論文で提案された正規化手法である。正規化の計算式を意式 2.3 に示す。

$$AdaIN(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y) \tag{2.3}$$

コンテンツ画像xとスタイル画像yを受け取り、xのチャネルごとの平均と分散を調整して、yのそれらと一致させる。AdaIN には学習可能なアフィンパラメータがなく、代わりにスタイル入力から計算することで、正規化されたコンテンツ入力を$\sigma(y)$でシンプルにスケーリングし，それを$\mu(y)$でシフトしている。これをネットワークの中に導入することによりあらゆる画像によるスタイル変換を可能にした。

AdaIN を StyleGAN で用いた場合の数式を式 2.4 で示す。

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \tag{2.4}$$

x_iは特徴マップ、$y_{s,i}$ $y_{b,i}$は潜在変数wをアフィン変換したものであり、これは特徴マップを正規化した後にスタイルを適用していることを意味する。この AdaIN を Mapping Network から得られた潜在変数wをスタイル情報として空間データに適用する役割を持ち、姿勢、髪型、輪郭など大局的な特徴を変化させることができる役割を担っている。

2 つの異なる特徴をもつ潜在変数w_1、w_2を組み合わせることで 2 つの特徴をもった画像が表現される。このときの特徴は、解像度ごとで決まっており、低解像度のスタイルを適用すると、顔の輪郭や向きなどの大まかな特徴が適用され、高解像度のスタイルを適用すると髪や肌の色など細かい特徴が適用されている。このように、各レベルにおいて異なる潜在変数wを適用させることで生成画像のスタイルを制御する手法を、一般に「Style Mixing」または、「Mixing Regularization」と呼ぶ。

2.3 pixel2Style2pixel

pixel2Style2pixel [10](pSp)は画像から直接 StyleGAN の潜在変数を推定できるエンコーダである[11]。また、pSp の構造はそのままにセグメンテーションマップからの顔画像生成、顔の正面化、超解像など様々な画像変換タスクに応用可能になっている。pSp のアーキテクチャを図 2.3 に示す。

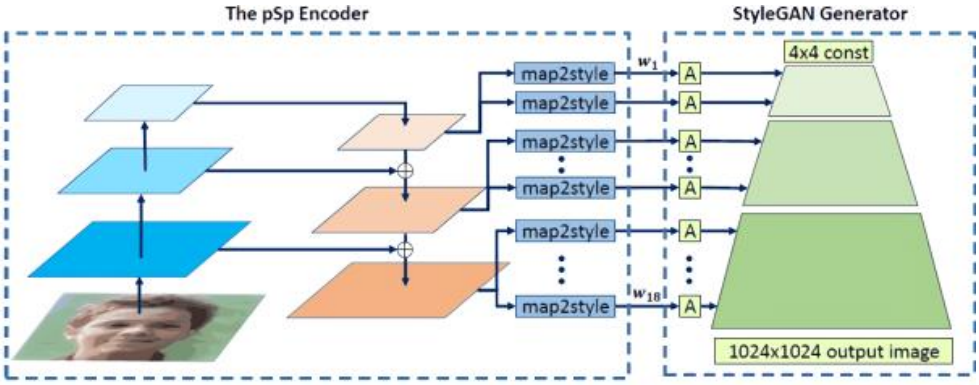

図 2.3　pSp のネットワーク構造([10]より引用)

エンコーダとは、StyleGAN でいうところの Mapping Network の部分を指しており、入力画像から潜在変数の得る手段を新たに提案している。

pSp フレームワークは、事前に訓練された StyleGAN ジェネレーターと潜在空間$W+$に基づいて構築されている。

まず入力画像を ResNet に通して特徴マップを得る。その後に特徴マップを 2 倍にリサイズし、対応する特徴マップとの和をとる。それぞれの特徴マップを map2style という 2-stride Convolution+LeakyReLU の構造をしたモジュールに入力し、18 個の 512 次元ベクトル(潜在変数w)の出力を得る。その出力をスタイルの情報として StyleGAN の Synthesis Network で用いる。

損失関数は L2 loss、LPIPS loss、Identity loss の 3 つの重み付き和で定義される。

L2 loss は入力画像と StyleGAN Generator の出力画像との差の平方和となる。LPIPS loss は Perceptual loss と同じように学習済みの VGG などに入力画像と出力画像を通して特徴マップの差の 2 乗を計算する。これに対してパラメータWをかけて正規化、平均したものを loss とする。Identity loss は 2 枚の画像の顔が同一人物かどうかを表す loss である。Arcface を通して最後の 1 つ前の層の 512 次元ベクトルをとり、コサイン類似度を

とったものを loss とする。和をとる時の係数はタスクによって異なる。

pSp の研究では、入力画像を忠実に再現することを目的として様々な手法を用いた実験を行っている。提案された超解像手法では、教師あり顔画像を 8 倍、32 倍にそれぞれダウンサンプリングした低解像度画像を入力したときの結果を pix2pixHD[12]と PULSE[13]と比較しており、再現度の高さと不自然さが少ないことから他よりも精度が高いことを証明している。さらに、低解像度画像から高解像度画像へのマッピングは 1 対多であるため、マルチモーダル手法を使用して生成した特定の低解像度画像に対して、いくつかの尤もらしい顔画像を出力する。ここでは、ランダムにサンプリングされた 512 次元のwベクトルを使用して、顔の特徴を制御する中レベルのスタイルに対しスタイルミキシングを実行している。

また、Inpainting というアプリケーションを提案しており、これは単純な対称三角形マスクを使用して、画像の欠落部分を再構築するフレームワークの機能である。元の画像との同一性を維持しながら、遮蔽された領域を正確に再構成すること可能であると述べられている。

一方で、pSp は補正された画像を学習に用いているため、入力する顔画像も同じように補正が必要な点と、顔が中央にない画像や、学習データにないような物体、細かな背景などは上手く再現することができない点を課題として述べている。

2.4 問題点

StyleGAN が提案する Style Mixing の研究では顔画像変換で高い精度を示しており、サングラスの除去やペアの特徴を組み合わせた顔変換を可能にしている。これらは学習データの特徴を示す潜在変数で表現された顔画像を用いて変換しているため、学習していない物体や特徴に対しては潜在変数で表現できない。そのため、Style Mixing を用いて顔画像からマスク除去を行うには、マスクを着用した顔画像を含むデータセットを用いて学習する必要があるが、衛生マスクには様々な色や形が存在しており、学習には膨大なコストが必要となる。

そこで、潜在変数で表現する際に学習していない特徴が再現できないという特徴を利用する。GAN の多くは学習していない物体の再現が難しく、不自然な顔画像を生成してしまう問題がある。それに対し正解データを学習することや周囲の情報から予測する方法で解決する研究がいくつか存在する。今回の場合、写真に写る不特定多数の人物を対象とするためマスクを着用していない正解データを用意できない。また、口や鼻などの繊細な部分を再現する場合周囲の情報から予測するが、完全に覆われていることで繊細な部分の表現が困難だとされる。さらに、マスクは顔の半分近くを占めるため完全な除去も困難である。

第3章 提案手法

これらの問題に対し、顔画像を低解像度に変換することでマスクの特徴量を減らすと同時に顔の部位の情報を曖昧にすることを検討する。低解像度の顔画像から高解像度の顔画像に変換する方法として超解像手法があるが、低解像度から学習することのできる StyleGAN と組み合わせることで、より精度の高い顔画像の生成ができると考えた。この場合、マスクを着用した顔画像の学習を必要としないため、顔画像の学習済みモデルを使用しコストを削減できる。また、教師なし画像に対しても同様の精度で生成が可能ならば正解データを必要としないため、不特定多数の人物に対して同様の結果を期待できる。さらに、低解像度画像の変換が可能ならば解像度の異なる様々な顔画像に対して衛生マスクを除去し表情を露わにすることが期待できる。

そこで本研究では、StyleGAN のエンコーダで低解像度画像から高解像度画像を生成する超解像手法を実装可能な pixel2style2pixel (pSp)を提案する。pSp の超解像手法を用いた画像生成全体のフローを図 3.1 に示す。学習データにない物体の生成がうまくできない特徴と超解像手法を用いることで低解像度画像のような曖昧な画像からでも自然な顔画像の生成ができる特徴あり、解像度の異なる顔画像からでも衛生マスクを除去すると同時に顔画像の特徴変換が期待できる。

pSp の研究では教師ありの顔画像を用いて再現を行っていたが、本研究では、すべて教師なしの顔画像を用いて入力画像とは異なる顔画像の生成を行う。学習データのシステム構成を図 3.2 に示す。この場合の教師なしとは、入力に用いるマスクを着用した顔に対しマスクを外したときの顔画像と低解像度に変換する前のマスクを着用した顔画像の 2 点を学習しないことである。これは、マスクを外した顔画像(正解データ)は入手が困難であることと、衛生マスクを学習させないことで潜在変数での表現を防ぐためである。さらに、学習していない物体(衛生マスク)が顔の下半分を隠している顔画像を用いる。これらの条件が重なっていても、違和感のない自然な顔画像を出力が可能か、またマスクを除去すると同時に顔画像の特徴変換が可能か検証する。

顔検出のネットワーク構成を図 3.3 に示す。pSp の実行には補正された顔画像のデータセットを用いた学習済みモデルを使用する。そのため、入力画像も補正した顔画像を扱う必要がある。そのためには顔検出が必要であるが、通常の顔検出と比べマスクを着用した顔や低解像度の顔の検出には困難を要する。そこで、dlib[14]が提供する従来の HOG 特徴量を使ったモデルよりも精度が高い結果を示している CNN ベースの機能を備えた最大マージンオブジェクト検出器 (MMOD) を使用することで解像度の異なるマスクを着目した顔の検出精度を高めることを検討する。dlib とは、画像処理、機械学習などの機能をもった C++ライブラリである。

顔検出には、dlib の他に OpenCV[15]の顔検出が有名ではあるが、今回使用する学習済みデータセットの顔補正が dlib を用いている点と、OpenCV の顔検出器は顔以外の物体を誤って検知することが考えられるため、今回は dlib の検出器を提案する。dlib が提供する顔検出器の最小の検出サイズは 80x80 pixel である。今回はより小さいサイズで検出す

るため、画像のサイズをアップサンプリングして行う。

またマスクを着用した画像を低解像度に変換したとき、マスクの色による影響を受けてしまう。それにより、マスクの色に近い肌をした顔画像を生成することや独自性を損なうことが考えられる。そこで、pSp が提案するマルチモーダル合成手法を用いてランダムにサンプリングされた顔画像の特徴を組み合わせる。これにより、マスクの色による影響を抑え、独自性を持たせることを可能にする。

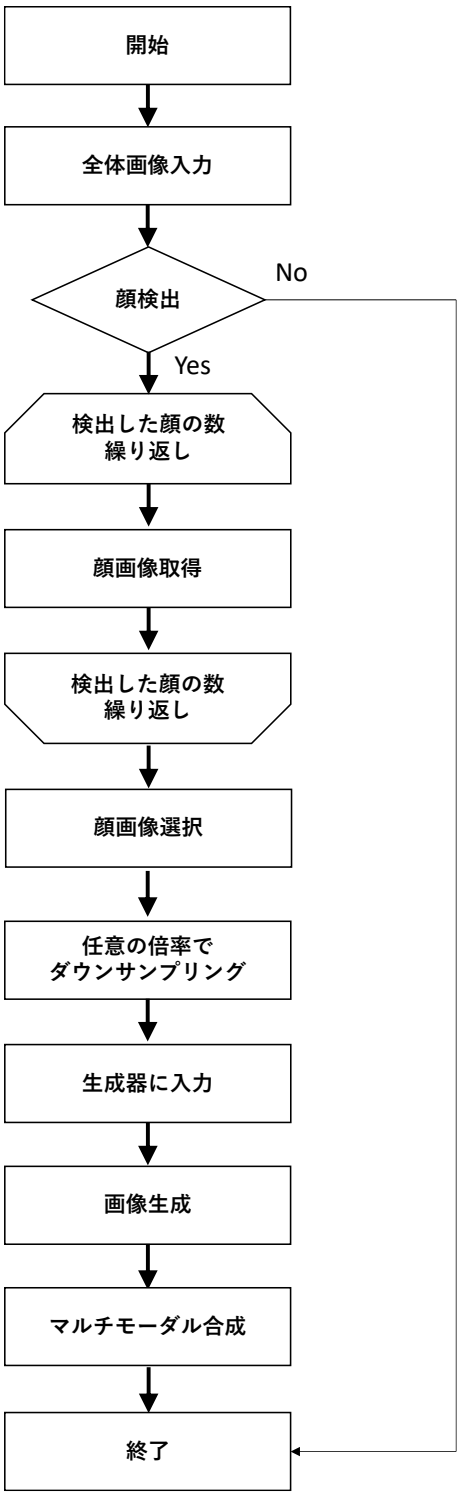

図 3.1　全体のフロー

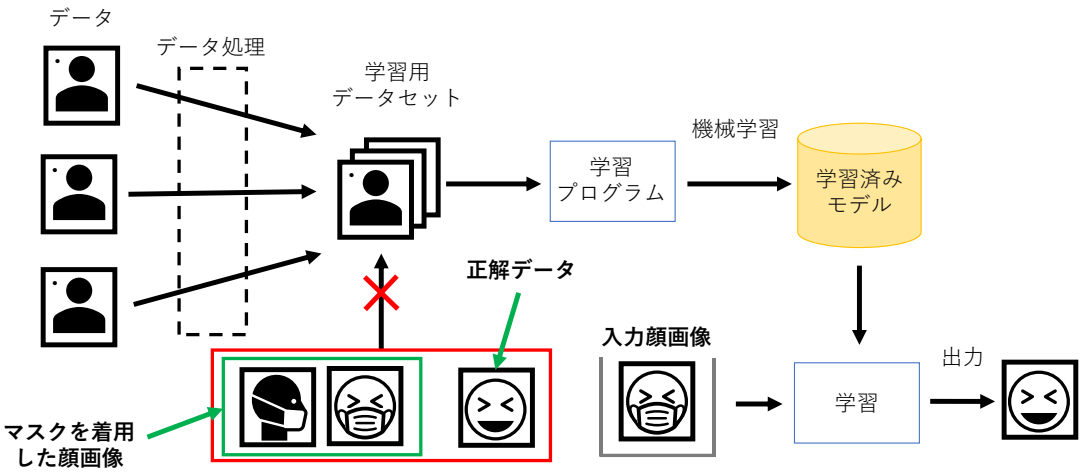

図 3.2 学習データのシステム構成

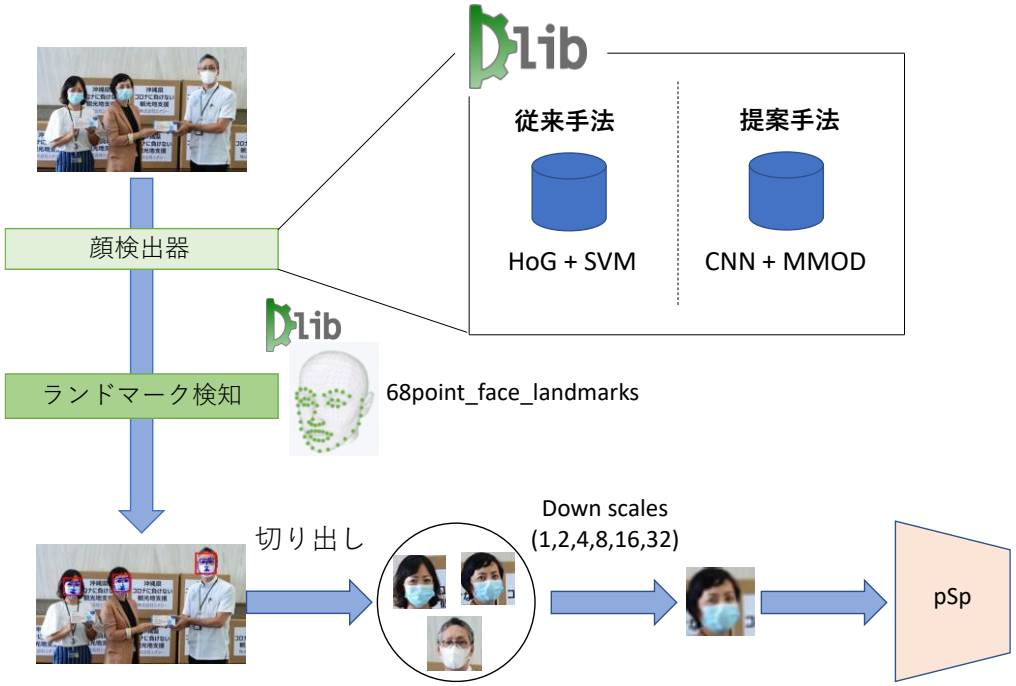

図 3.3　顔画像取得までのネットワーク構造

第4章 実験

4.1 実験環境

本実験では、ブラウザから Python を記述、実行できるサービスである Google Colaboratory[16]で機械学習の計算を行った。Google Colaboratory の計算環境を表 1 に示す。計算には GPU を用いたが、利用可能な GPU のタイプはその時々で変わり、通常 NVIDIA Tesla K80、T4、P4、P100 などがある。そのため、表 1 の GPU は計算時に使用した一例を挙げる。Google Colaboratory を使用するために用いた計算機環境を表 2 に示す。

表 1. Google Colaboratory の環境構成

CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
OS	Ubuntu 18.04.5 LTS
メモリ	12GB
GPU	NVIDIA Tesla T4
python	3.6.9

表 2. 計算機の環境構成

CPU	AMD Ryzen 5 2500U @ 2.00GHz
OS	Windows 10 (x64)
メモリ	8GB
Web ブラウザ	Google Chrome

4.2 実験 1

4.2.1 実験概要

従来の pSp の超解像手法で違和感のない自然な顔画像を出力が可能か、またマスクを除去すると同時に顔画像の特徴変換が可能かを検証することが目的である。マスクを着用している人物が単独で写る画像を対象に、顔の検出とランドマークを検知することで、顔画像を補正した。この顔画像を 8 倍と 32 倍にそれぞれダウンサンプリングした低解像度画像を入力に用いて、高解像度に変換した。

4.2.2 実験内容

マスクを着用している人物が単独で写る画像をランダムに用意する([17]より引用)。解像度は様々で 96ppi のものから 300ppi のものを使用している。対象の画像から顔を検出し、StyleGAN の学習済みモデルと同様に補正するため、ランドマークの情報から顔の位置が中心にくるように切り出す。顔検出には、C++の機械学習ライブラリである dlib の HOG 特徴量を使ったモデルを用いる。検出した顔画像に対し、学習済みモデルの shape_predictor_68_face_landmarks.dat を用いて 68 ポイントのランドマークを検知する。切り出した顔画像を 8 倍と 32 倍にダウンサンプリングした低解像度画像に変換する。pSp の超解像手法を実装するにあたって、CelebA-HQ[18]データセットでトレーニングされた pSp モデルの psp_celebs_super_resolution.pt を用いる[19]。この学習済みの pSp エンコーダに低解像度画像を入力することで拡張された潜在変数を獲得し、学習済みの StyleGAN の生成モデルに組み込むことで画像を生成する。生成した画像は 256×256 にリサイズして出力する。

4.2.3 実験結果

顔を検出した結果の例を図 4.1 に示す。図 4.1 からランドマークを検知した結果を図 4.2 に示す。8 倍でダウンサンプリングした低解像度画像を用いた結果を図 4.3 に、32 倍を用いた結果を図 4.4 に示す。

図 4.1 顔を検証した結果の例

図 4.2 ランドマークを検知した結果

図 4.3　8 倍にダウンサンプリングした結果

図 4.4　32 倍にダウンサンプリングした結果

4.3 実験 2

4.3.1 実験概要

従来の HOG 特徴量を使ったモデルと CNN ベースの物体検出モデルで、複数写る画像に対し解像度の異なるマスクを着目した顔の検出精度の比較を行った。提案手法で検出した顔画像に対し、実験 1 と同様に低解像度に変換し、高解像度画像の生成が可能か検証、比較を行った。

4.3.2 実験内容

マスク姿が複数写る写真([20]より引用)を用いる。dlib が提供する顔認証モデル CNN ベースの機能を備えた最大マージンオブジェクト検出器（MMOD）を使用して顔を検出する。これには dlib が提供する学習済みモデル mmod_human_face_detector .dat.bz2 を用いる。低解像度の顔検出を行うため、画像を 4 倍にアップサンプリングし、20×20 pixcel で検出を行う。さらに、提案手法で切り出した顔画像を 8 倍にダウンサンプリングした低解像度に変換し、実験 1 と同様に画像を生成する。写真からの顔検出、検出した顔画像とランドマーク、顔画像から生成した結果をそれぞれ従来の顔検出器と比較する。

図 5.1　複数のマスク姿が写る画像

4.3.3 実験結果

従来の手法で画像から顔検出を検出した結果を図 5.2 に、CNN で検出した結果を図 5.3 に示す。図 5.2 顔検出とランドマークの検知を行った結果を図 5.4、同一人物の顔に対し図 5.3 を用いた結果を図 5.5 に示す。図 5.2 で検出した顔から生成した結果と図 5.3 から生成した同一人物の結果を図 5.6 にまとめて示す。図 5.3 で検出したその他の顔に対して生成した結果を図 5.7 に示す。

図 5.2 従来の手法で顔検出した結果

図 5.3 DNN で顔検出をした結果

図 5.4 従来手法の顔検出

図 5.5 DNN の顔検出

図 5.6 従来手法と提案手法を用いた結果
(上：従来の手法、下：提案手法)

図 5.7 提案手法で検出した顔画像を入力に用いた結果

4.4 実験 3

4.4.1 実験概要

超解像手法で低解像度画像から高解像度に変換する場合、画像を生成するための情報が少なく、独自性を維持することが難しい。そこで、低解像度画像を学習した結果得られた潜在変数と別の顔の特徴をもつ潜在変数を組み合わせ、肌の色や口の形状を変換した画像を生成し、独自性が確保できることを検証した。

4.4.2 実験内容

図 5.3 で検出した 3 つの顔画像を低解像度画像に変換し、pSp が提案するマルチモーダル合成手法を用いてスタイル変換を行う。まず、肌の色に対しスタイル変換を行う。このときのマルチモーダル合成に関するパラメータを表 3 に示す。latent_mask はランダムにサンプリングされた潜在変数を用いるレイヤーの番号であり、 [8, 9, 10, 11, 12, 13]は主に肌の色の特徴を示すレイヤーである。mix_alpha は混合係数であり、一方の特徴に偏りすぎないように 0.5 にする。n_outputs_to_generate は生成する数であり、3 とする。合成に使用する潜在変数ベクトル(スタイル情報)は numpy.random.randn()関数でランダムに生成された値(512 次元特徴ベクトル)を 3 個(n_outputs_to_generate で指定した値)用いる。np.random.randn()は、平均 0、分散 1（標準偏差 1）の正規分布（標準正規分布）に従う乱数を返す関数である。

さらに肌の色の変換に加え、顔面の特徴に同様のスタイル情報の変換を行う。このときのマルチモーダル合成に関するパラメータを表 4 に示す。latent_mask は主に口や目など顔面の特徴を示すレイヤーである[4, 5, 6, 7]とする。mix_alpha は、0.3 にする。これは、latent_mask のレイヤー番号が小さいほど顔全体の特徴に影響するため、入力画像の特徴が強くなるように値を小さくする。n_outputs_to_generate は生成する数でありこれも同様に 3 とする。

表 3　肌の色に対しマルチモーダル合成に用いた変数と値

変数名	パラメータ
latent_mask	[8,9,10,11,12,13]
mix_alpha	0.5
n_outputs_to_generate	3

表 4　顔面の特徴に対しマルチモーダル合成に用いた変数と値

変数名	パラメータ
latent_mask	[4, 5, 6, 7]
mix_alpha	0.3
n_outputs_to_generate	3

4.4.3 **実験結果**

肌の色をスタイル変換した結果を図 6.1 に示す。肌の色に加え目や口などの顔面の特徴を変換した結果を図 6.2 に示す。左が入力に用いた低解像度画像、右の他 3 枚が生成した画像である。

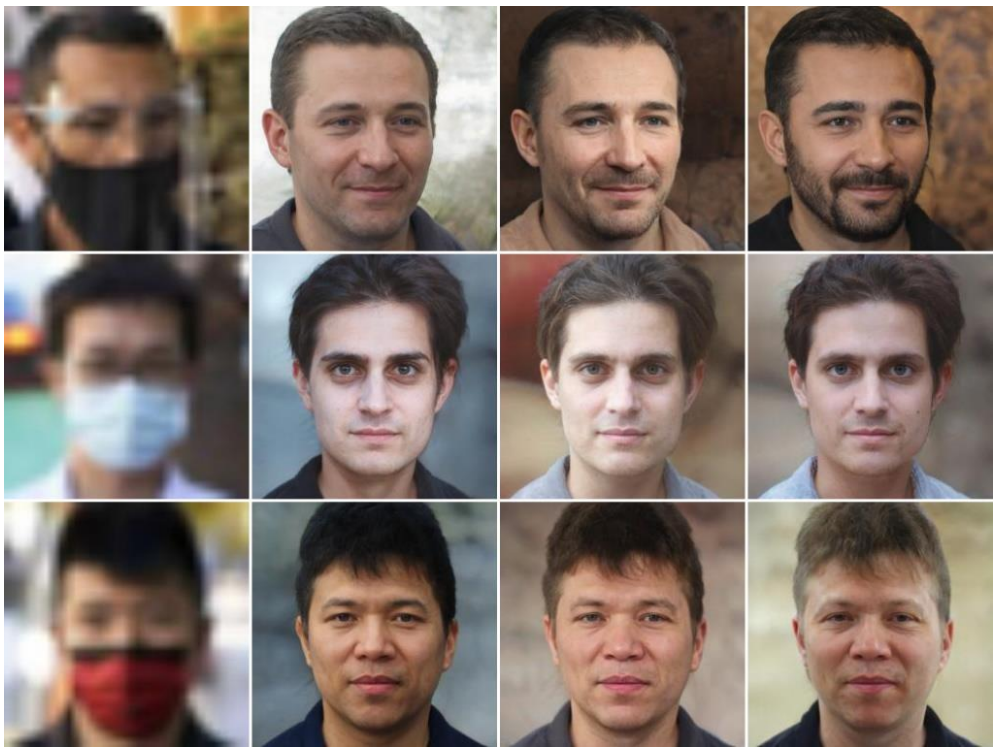

図 6.1 肌の色[8, 9, 10, 11, 12, 13]をメインにスタイル変換した結果
(左が低解像度画像「8 倍にダウンサンプリングしたもの」、
右 3 つがスタイル変換の結果)

図 6.2 図 6.1 にさらに顔面の特徴[4,5,6,7]をスタイル変換した結果
(左が低解像度画像「8 倍にダウンサンプリングしたもの」、
右 3 つがスタイル変換の結果)

第5章 考察

pSp の超解像手法を用いて画像を生成した。実験 1 では、pSp の超解像手法を用いた場合、マスクの除去と顔画像の変換は可能であった。また、解像度の違う画像を入力に用いた場合でも同等の精度が確認でき、どちらとも不自然さの残らない顔画像を生成した。入力した顔画像と生成画像を比較すると、顔の輪郭や、髪形の再現度は高かった。しかし、マスクの色影響を受けてしまい、肌の色がマスクの色の特徴を含んだ。これは、低解像度に変換したとき、マスクの占める顔の面積が大きいため、色の影響を強く受けてしまうことが原因だと考えられる。しかし、顔の上半分の特徴も考慮している点と、学習データにマスクの色と同じ肌の色をした人物が存在しない点がその影響を抑えているため、明確な不自然さを確認できるほどの影響は感じなかった。

実験 2 では、新たに提案した CNN ベースモデルの顔検出器はマスクを着用しており解像度の低い人物に対して従来の顔検出よりも高い精度を示した。しかし、マスクを着用した人物全員を検出することは不可能であった。検出精度を上げるには、マスクを着用した顔画像の追加学習あるいはラベル付けした目や輪郭が曖昧または顔の向きの異なる画像を学習させることが考えられる。CNN モデルのみ検出したマスク姿の顔に対してもランドマークの検知と画像の生成が可能であった。

従来の手法と CNN モデルを使用し顔検出した結果を比較すると、口や鼻は検知したランドマークと一致しなかったがマスクの上からでも口や鼻のランドマークを検知も可能であった。これはどちらの結果も目のランドマークの検知は正確だったため、目の情報から予想していると考えられる。顔検出の位置やランドマークに僅かな誤差が確認でき、目以外のランドマーク情報や顔検出に関しては従来の手法のほうが正確であった。そのためトリミングした顔画像や生成画像にも誤差が見られたが、髪形や輪郭などの大まかな特徴は維持しているため、顔の特徴変換を目的とする本研究では許容範囲内である。

実験 3 では、マルチモーダル合成を用いたスタイル変換を行った。実験 2 の結果と比較すると、肌の色の合成に関しては、顔の輪郭や顔の部位は生成画像の特徴を保ち、肌の色以外に髪の色も変換した。口や目などの顔の部位の合成では、顔の向きや輪郭には大きな変化はみられなかったが、口や目以外に髪形や性別なども変化した。これらから限定した部位の変換は難しいとされる。実験 1、実験 2 の結果では肌の色がマスクの色に寄っていたが、この問題をスタイル変換で解決することができた。以上から、マルチモーダル合成を用いたスタイル変換を行ったことで生成画像のマスクによる肌の色の問題解決や独自性の追求が実現できた。また、生成した顔画像も実在しない自然な画像で生成できることが分かった。

しかし、ペア画像(特徴)がランダムに生成され、特定の特徴を人間の主観的評価で決めているため、任意の画像を生成するには少し手間(コスト)がかかってしまう。そのため、正解データや隠れていない部分から特定の特徴を推測や、機械による客観的評価を取り入れることが必要だと考える。また、超解像手法、Style Mixing それぞれによって生成され

た画像に対して、背景や髪形、解像度も変換してしまうことから、元の画像に置き換えることが少し困難である。これには顔面のみで変換することが必要だと考える。

第6章 結論

写真に写る人物からマスクを除去し表情を読み取れるようにすると共に、個人を特定できないような顔画像に変換することを目的に、StyleGAN と pSp の超解像手法を用いることを提案した。pSp の超解像手法は異なる低解像度を用いた場合でも完全にマスクを除去し、隠れていた部分を再現することができた。入力画像との非類似性においては、顔の輪郭や髪形は類似するが、目などの細かな部分は類似性が低く、全体的に入力画像の再現度が低いため、別人の顔として認識することができた。入力画像の顔検出には HOG 特徴量を使ったモデルより CNN ベースの物体検出モデルの方が低解像度のマスク姿に対して高い精度を得られた。マルチモーダル合成を活用することでマスクによる色の影響を最小限抑え、肌の色や目や口など詳細な部分を変換することができ、独自性を持たせることができた。

以上のことから、CNN モデルで検出した顔に pSp の超解像手法とマルチモーダル合成を用いた変換を組み合わせることで、写真からマスクの除去と別人に変換した顔画像の生成に関しては有用性が高いことが確認できた。

今後の課題として、機械による客観的評価を取り入れることや、顔面のみで特徴変換することでコストを削減につながると考える。

謝辞

本論文を作成するにあたり、多くの指導、ご助言を頂きました三好力教授に厚くお礼申し上げます。また、議論・実験に協力してくださった三好研究室の皆様や学友の皆様に心から感謝いたします。

参考文献

[1] 田中 裕二. 非言語的コミュニケーションが身体に及ぼす影響について. 2016-04-21.https://kaken.nii.ac.jp/ja/grant/KAKENHI-PROJECT-20659331/,
(参照 2020-12-20).

[2] 株式会社プラネット. “マスクと伊達マスクに関する意識調査”. FROM プラネット. 2016-01-21. https://www.planet-van.co.jp/pdf/fromplanet/fromplanet_30.pdf, (参照 2020-12-20).

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Networks. 2014-06-10. arXiv:1406.2661v1. https://arxiv.org/abs/1406.2661, (参照 2021-01-12).

[4] Tero Karras, Samuli Laine, Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. 2019-3-29. arXiv:1812.04948v3.
https://arxiv.org/abs/1812.04948, (参照 2021-01-12).

[5] 平内 雅則. “GAN : 敵対的生成ネットワークとは何か ～「教師なし学習」による画像生成”. iMagazine. 2018-7.
https://www.imagazine.co.jp/gan%EF%BC%9A%E6%95%B5%E5%AF%BE%E7%9A%84%E7%94%9F%E6%88%90%E3%83%8D%E3%83%83%E3%83%88%E3%83%AF%E3%83%BC%E3%82%AF%E3%81%A8%E3%81%AF%E4%BD%95%E3%81%8B%E3%80%80%EF%BD%9E%E3%80%8C%E6%95%99%E5%B8%AB/, (参照 2021-01-12).

[6] NegativeMind. “GAN (Generative Adversarial Networks) : 敵対的生成ネットワーク”. NegativeMindException -Negative Simulation, Positive Planning--. 2021-01-02.
https://blog.negativemind.com/2019/06/22/generative-adversarial-networks/, (参照 2021-01-12).

[7] Akihiro FUJII. “GAN の基礎から StyleGAN2 まで”. 2019-12-22. https://akichan-f.medium.com/gan%E3%81%AE%E5%9F%BA%E7%A4%8E%E3%81%8B%E3%82%89stylegan2%E3%81%BE%E3%81%A7-dfd2608410b3, (参照 2021-01-12).

[8] YasutomoNakajima. “StyleGAN とはなにか”. Qiita. 2020-10-17.
https://qiita.com/YasutomoNakajima/items/1e0153cfb598641f5c9b, (参照 2021-01-12).

[9] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. 2018-2-26. arXiv : 1710.10196v3.
https://arxiv.org/abs/1710.10196, (参照 2021-01-18).

[10] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, Daniel Cohen-Or. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation, richardson2020encoding, 2020-08-03. arXiv:2008.00951v1.
https://arxiv.org/abs/2008.00951v1, (参照 2021-01-12).

[11] けやみぃ. “数多の画像変換が可能！StyleGAN の新たな Encoder！pixel2Style2pixel”. AI-SCHOLAR. 2020-09-14.

https://ai-scholar.tech/articles/gan/pixel2Style2pixel, (参照 2020-12-20).
[12] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. 2018. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. https://tcwang0509.github.io/pix2pixHD/, (参照 2021-01-12).
[13] achit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, Cynthia Rudin.
PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models. 2020-07-20. arXiv:2003.03808v3.
https://arxiv.org/abs/2003.03808, (参照 2021-01-12).
[14] Davis King. Dlib C ++ Library. 2020-08-08. http://dlib.net/. (参照 2021-01-12).
[15] OpenCV team. OpenCV: Home. OpenCV. https://opencv.org/, (参照 2021-01-12).
[16] Google. “Colaboratory へようこそ”. Colaboratory - Google Colab.
https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja#scrollTo=-Rh3-Vt9Nev9, (参照 2020-12-20).
[17] Freepik Company. Freepik: ベクター画像、写真、PSD ファイルの無料ダウンロード. freepik. https://jp.freepik.com/home, (参照 2021-01-12).
[18] Lee, Cheng-Han and Liu, Ziwei and Wu, Lingyun and Luo, Ping. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. 2020. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
https://github.com/switchablenorms/CelebAMask-HQ, (参照 2021-01-12).
[19] eladrich, yuval-alaluf. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation, github. https://github.com/eladrich/pixel2style2pixel, 参照 2021-01-12).
[20] MATT OTT. US services sector grows for seventh consecutive month. Seattlepi. 2021-1-7.https://www.seattlepi.com/news/article/US-services-sector-grows-for-seventh-straight-15852746.php, (参照 2021-01-16).
[21] link. “mmod_rectangles を Dlib 経由で矩形に変換するには?”, 366 SERVICE. 2020-02-06. https://www.366service.com/jp/qa/f34b3519fb2dfa6d0afa024a8fc81d8f, (参照 2021-01-16).
[22] chowagiken_kin. “OpenCV と dlib の顔検出機能の比較”, 調和技研 技術ブログ. 2019-06-28.https://blog.chowagiken.co.jp/entry/2019/06/28/OpenCV%E3%81%A8dlib%E3%81%AE%E9%A1%94%E6%A4%9C%E5%87%BA%E6%A9%9F%E8%83%BD%E3%81%AE%E6%AF%94%E8%BC%83, (参照 2021-01-16).

付録

実験１で使用した原画像