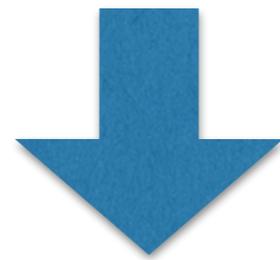


畳み込みニューラルネットワークを用いた モノクロ写真の自動部分カラーリングの試み

情報メディア学科 斎藤 一ゼミナール
1423082 安藤 聖人

近年、人工知能の技術が発展^[1]



モノクロ写真を自動でカラー化

^[1]注目すべき最新のITトレンド6選

先行事例（畳み込みニューラルネットワーク）³

TensorFlowを用いた写真を自動でフルカラーにする方法^[2]



モノクロ



自動フルカラー

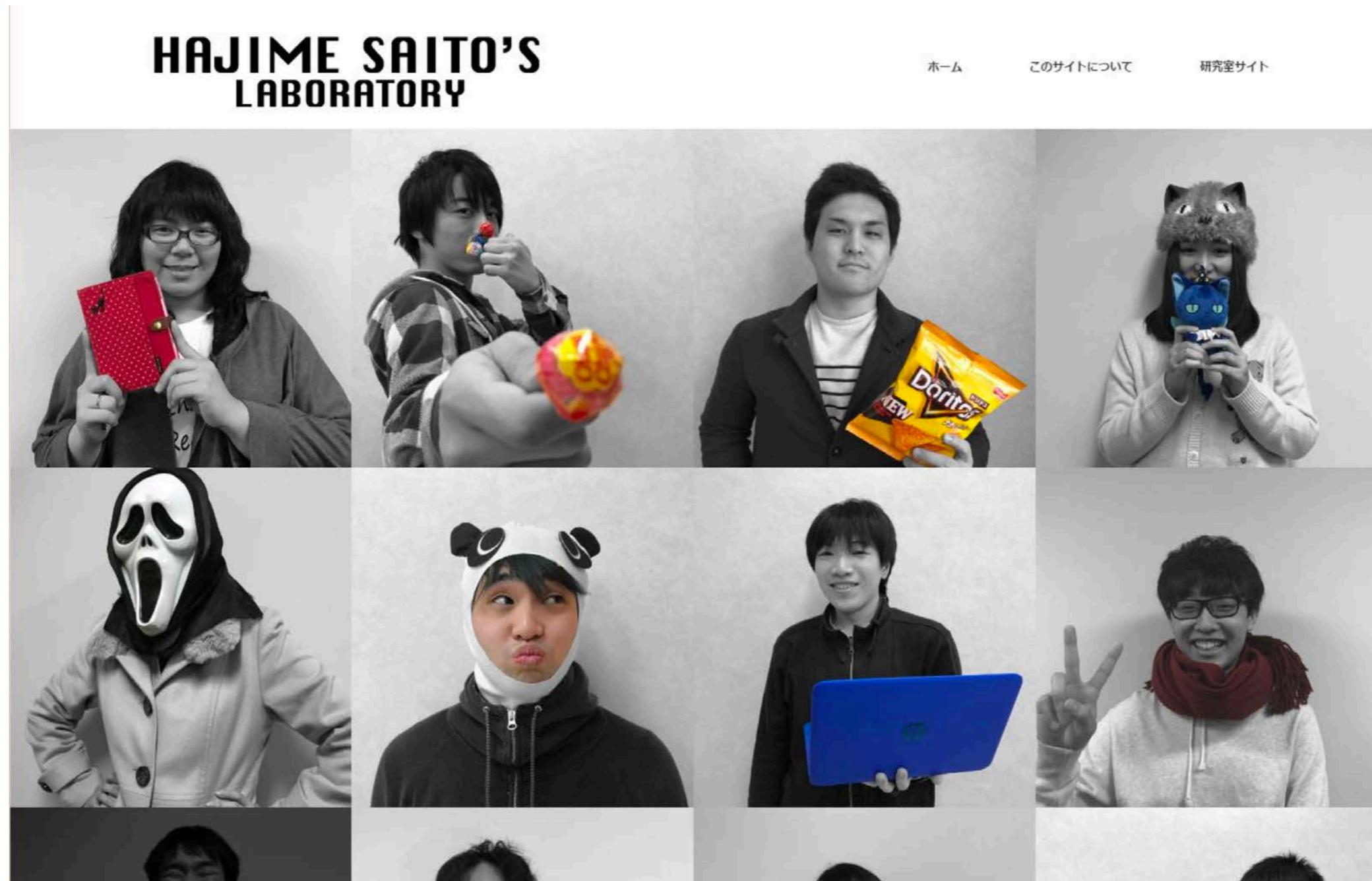
^[2]モノクロ写真をカラーにする方法

<http://www.shun.bz/20160127/1050814061.html> 参照 Jun.18, 2017

その人の特徴的なモノをカラー化する例

→メンバーの活動状況を動的に表現する
ゼミ生サイトの制作

メンバーの特徴を表す部分のカラー化^[3]
加工にはPhotoshopを使用している

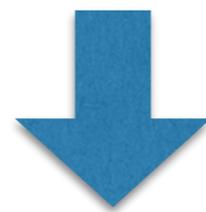


[3]HAJIME SAITO'S LABORATORY
<http://sherry.do-johodai.ac.jp/owned/> 参照 Jun.18, 2017

1. ゼミ生サイトの例で写真を加工した本人しか
全く同じように加工をするのは難しい

+

2. Photoshopなどのソフトを使って画像の加工を
手動で行う必要がある



写真を学習させることにより、
自動で写真の一部をカラーにできないか？

畳み込みニューラルネットワークを用いて モノクロ写真の部分的な自動着色

目的：人工知能を用い、写真の加工の効率化

- ゼミ生サイトの運用の手間を省く

⋮

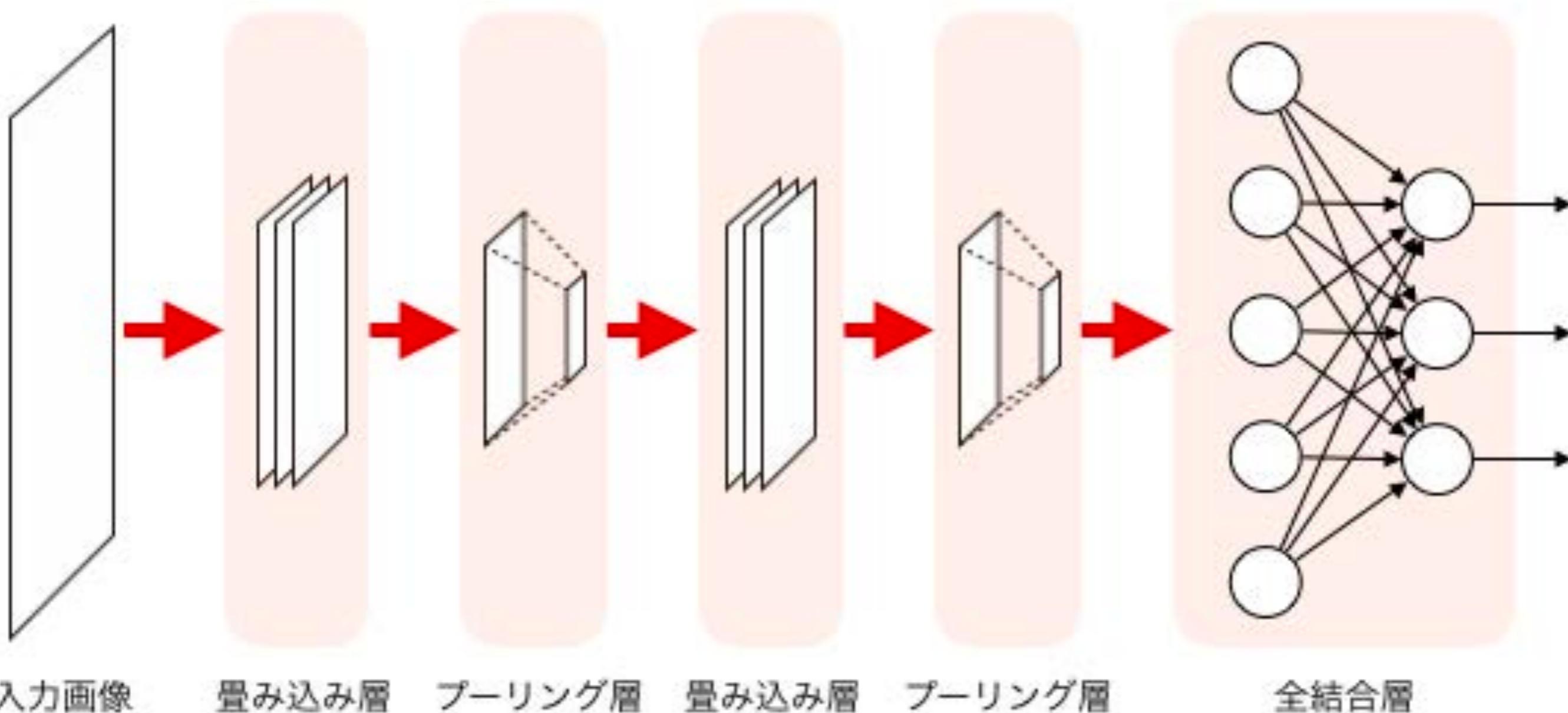
今後の展望として

- SNSの画像投稿の加工フィルタとして提案

ターゲット

ゼミ生サイトの運用者

SNS利用者



幾つかの層で構成され、画像認識の分野で使用される

[4]畳み込みニューラルネットワーク

<https://jp.mathworks.com/discovery/convolutional-neural-network.html> 参照 Dec.26, 2017

0 _{x1}	0 _{x0}	0 _{x1}	0	0
0 _{x0}	0 _{x1}	1 _{x0}	1	0
0 _{x1}	1 _{x0}	0 _{x1}	1	0
0	1	0	1	0
0	0	1	0	0

入力画像

0		

特徴マップ

- ① 入力画像を読み込む
- ② 階層ごとに局所的に輪郭などを数値化し計算
- ③ それを繰り返していき、特徴を抽出
- ④ 各数値を学習 → **トレーニングデータ**

[4]畳み込みニューラルネットワーク

線画の着色やモノクロ写真の着色に用いられるCNN

特徴

- 実験1 1枚の画像を学習 → 画像生成
- 実験2 ◎**2枚の画像を学習** → **画像生成**

2枚の画像とは、**入力画像とターゲット画像**

入力画像 →
加工前の写真



← ターゲット画像
手動で加工後の写真

[5]Image-to-Image Translation Using Conditional Adversarial Networks,
<https://arxiv.org/pdf/1611.07004v1.pdf>, 参照 Nov.29, 2017

着色にはトレーニングデータが必要

①



写真

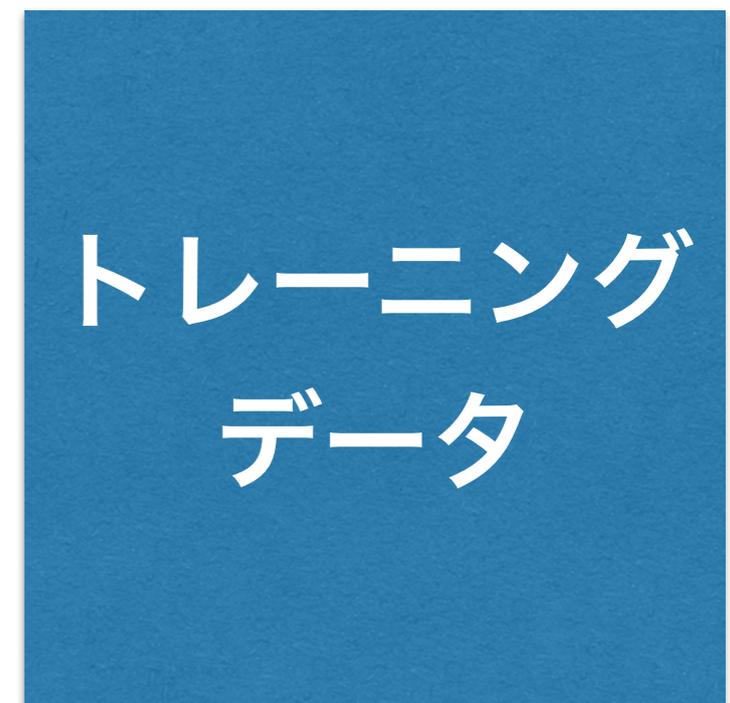
CNN



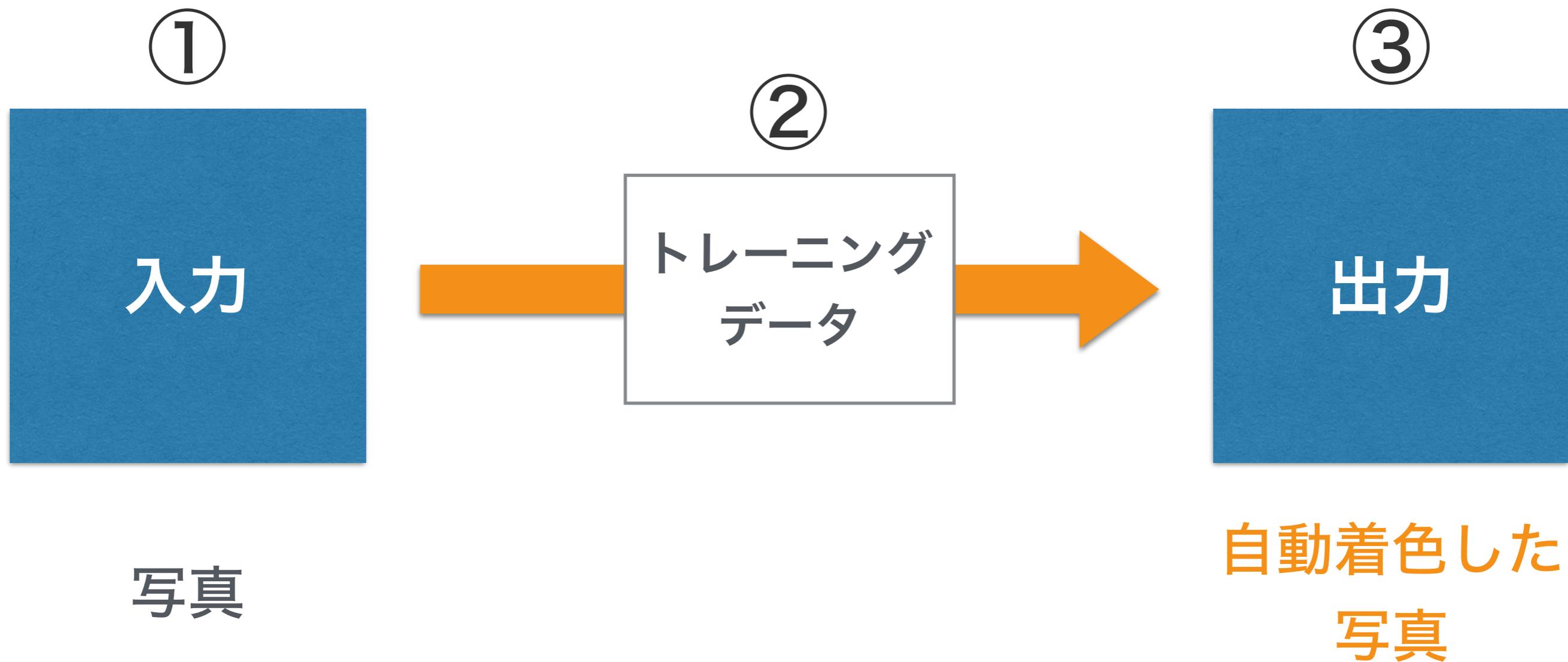
②

学習

③



トレーニング
データ



OS	MacOS Sierra
CPU	2.9 GHz Intel Core i5
メモリ	16 GB
言語	Python2.5.2
使用技術	TensorFlow1.4.1 OpenCV Pix2Pix

実験1：ゼミ生サイト写真，一枚のみを学習

①入力



モノクロ写真

②

トレーニング
データ



ゼミ生サイト写真

③出力



自動着色

トレーニングデータ

ゼミ生写真

計**35**枚



- 画像サイズ
256×256ピクセル
- 学習回数 (epoch数)
100回
- 学習時間→308分

入力画像：モノクロ写真

結果

入力
Input



手動加工
Target



自動着色
Output



結果

入力
Input



手動加工
Target



自動着色
Output



- 実験1では意図した着色ができなかった
- 35枚の画像の出力するのに1分もかからなかった

実験2：カラー写真とゼミ生サイト写真の2枚1セットで学習

①入力



カラー写真

②

トレーニング
データ



カラー写真
+
ゼミ生サイト写真

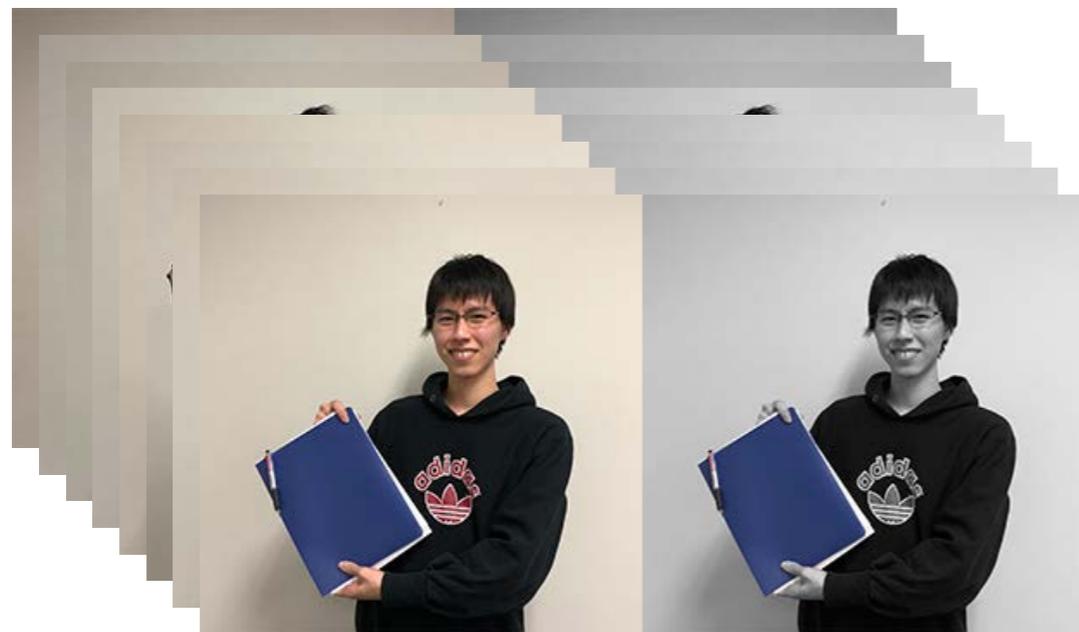
③出力



自動着色

トレーニングデータ

カラー写真+ゼミ生サイト写真
計**35**枚



- 画像サイズ
256×256ピクセル
- カラー画像+ゼミ生画像の
セットで**1枚**とする
- 学習回数 (epoch数)
100回
- 学習時間→357分

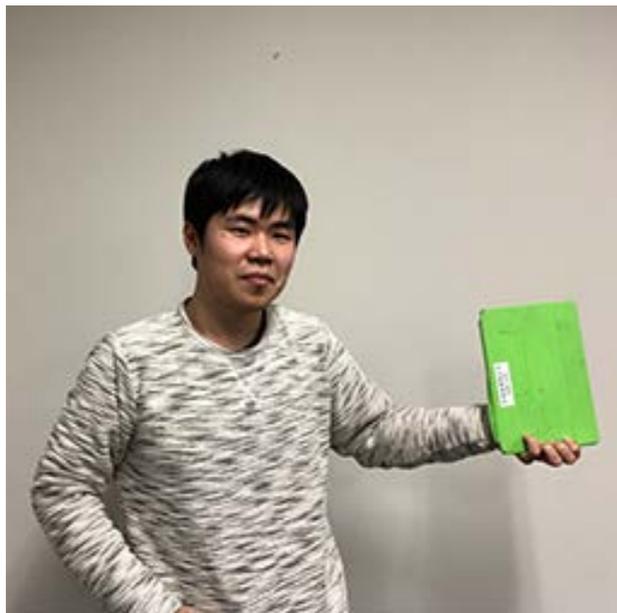
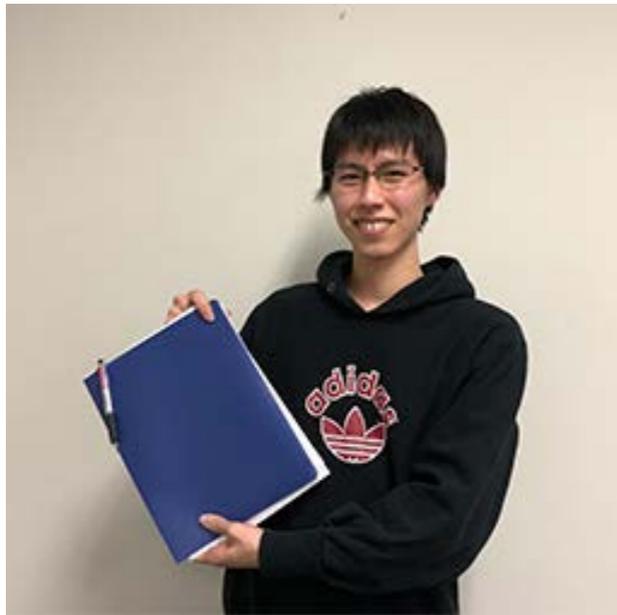
入力画像：カラー写真

結果

入力
Input

手動加工
Target

自動着色
Output



入力画像：カラー写真

結果

入力
Input



手動加工
Target



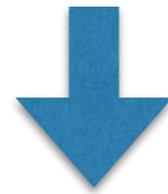
>

自動着色
Output



- 実験2では**部分的な着色が行われた**
- 35枚の画像の出力するのに1分もかからなかった

トレーニングデータには無い写真の着色



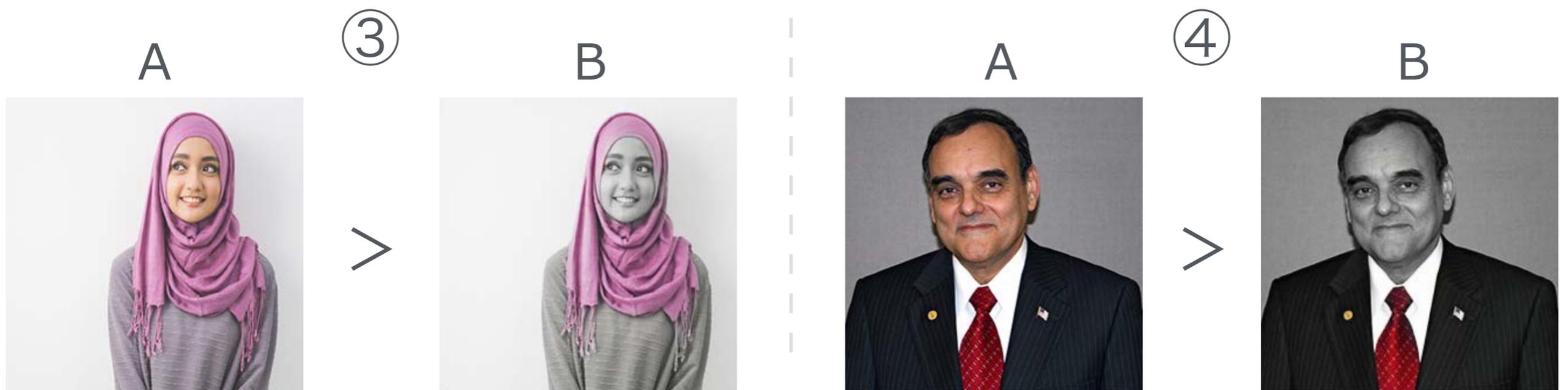
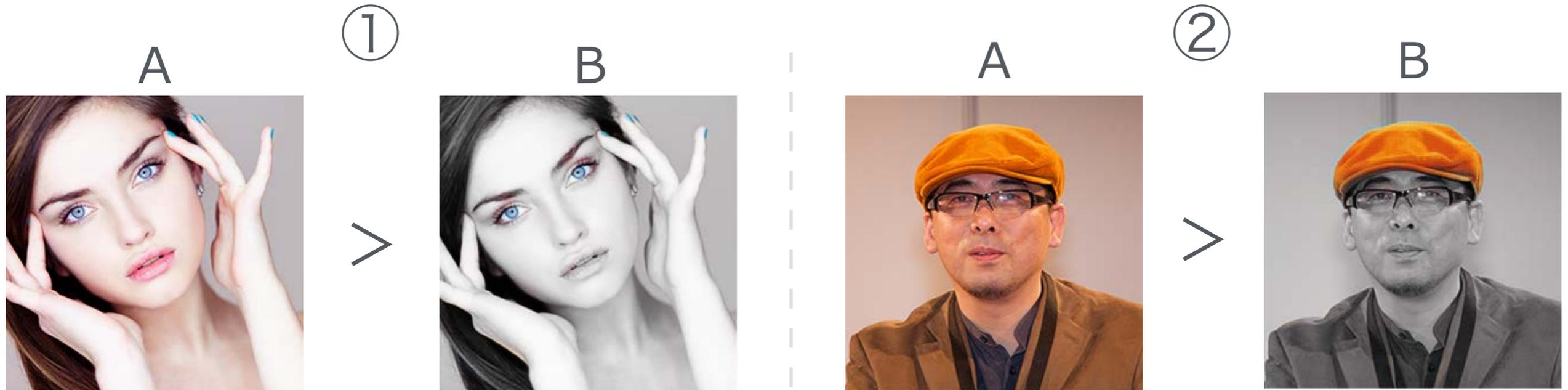
実験 2 の設定で
学習回数・トレーニングデータを増加し比較

学習回数 : 100回 → **1000**回
トレーニングデータ : 35枚 → **200**枚

トレーニングデータ外でテスト

➔ **部分的な着色**が概ねできていた

A:入力画像
B:自動着色



※著作権フリーの画像を使用しています。

OpenCVを用い, 0~1.0になるような類似度を算出 [7]



[a]手動で加工した写真



[b]自動で着色した写真

[7] Python + OpenCVで類似度を求める

https://qiita.com/best_not_best/items/c9497ffb5240622ede01, 参照 Jan.5, 2017.



[a] 手動で加工した写真



[b] 自動で着色した写真

- ① a,bの画像をグレースケールに変換
- ② a,bの画像の色合いを計算
- ③ a,bの計算した値を比較
- ④ 比較した結果（類似度）を表示

[7] Python + OpenCVで類似度を求める

https://qiita.com/best_not_best/items/c9497ffb5240622ede01, 参照 Jan.5, 2017.

手動で加工

自動着色

主観で着色された
と思う写真の類似度
0.9632



主観で着色されていない
と思う写真の類似度 [8]
0.1616



[8] 「伊藤美来」インタビュー

<https://ddnavi.com/interview/304229/a/>, 参照 Jan.10, 2017.

トレーニングデータとそれ以外の類似度

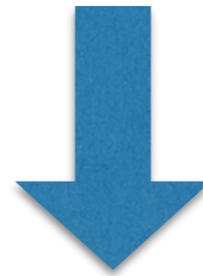
類似度 [0~1の実数]

	平均値	標準偏差
トレーニングデータ	0.852	0.212
トレーニングデータ外	0.862	0.115

トレーニングデータ以外の画像を用いているにも関わらず
平均値が**トレーニングデータよりも高**くなった

楽器、花束等

輪郭線が多い部分の着色は曖昧



トレーニングデータを増やす必要性

問題点 : 輪郭線が多い画像では着色されにくい

【目的】

人工知能を用い、写真の加工の効率化



自動で部分的な着色が行えた
手作業で着色した結果に近い着色を行えた

今後の課題

- ・ トレーニングデータを増やす