

Convolutional Neural Network を用いた顔の系統分類

日本大学文理学部情報科学科 谷聖一研究室 松平 礼史

目次

1. はじめに

1.1 背景

1.2 動機・目的

2. 演習内容・方法

2.1 系統の代表例選定

2.2 画像データの収集と 前処理・水増し

2.3 学習モデルの構築

3. 演習結果・考察

3.1 結果

3.2 考察

1. はじめに

1. はじめに

1.1 背景

Convolutional Neural Networkについて

Convolutional Neural Network（畳み込みニューラルネットワーク）

- 主に中間層が他のディープラーニングのモデルと違う
- 画像認識で特に優れた実績を収めている

1. はじめに

1.1 背景

Convolutional Neural Networkについて

Convolutional Neural Network（畳み込みニューラルネットワーク）

- 主に中間層が他のディープラーニングのモデルと違う
- 画像認識で特に優れた実績を収めている

CNNがなぜ画像認識に適しているのか

1. はじめに

1.1 背景

Convolutional Neural Networkについて

- 生物の脳の視覚神経系からヒントを得たニューラルネットワークの一種
- 視覚系における受容野をモデル化している

S層 (単純細胞層 – simple cell) . . . 図形の特徴を抽出

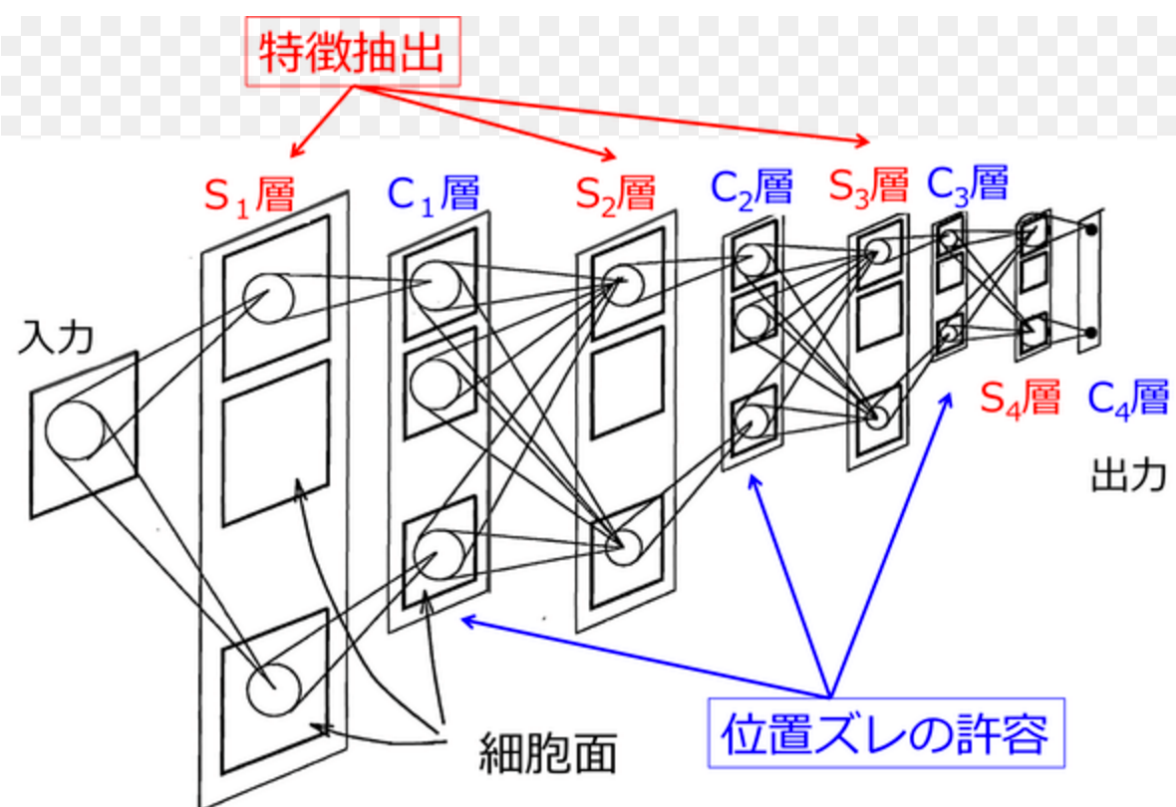
C層 (複雑細胞層 – complex cell) . . . 位置ズレを吸収する

1. はじめに

1.1 背景

Convolutional Neural Networkについて

視覚系における受容野モデル



S層とC層が交互に結合する構造により、入力された情報はS層による特徴抽出とC層による位置ずれの許容を繰り返しながら上位の各層に伝搬される

<https://lp-tech.net/articles/fbf5j>

1. はじめに

1.1 背景

Convolutional Neural Networkについて

視覚系における受容野

Convolutional Neural Network

S層 (単純細胞層 – simple cell)

→

畳み込み層

C層 (複雑細胞層 – complex cell)

→

プーリング層

1. はじめに

1.1 背景

Convolutional Neural Networkについて

畳み込み層の役割

CNNは入力のあらゆる位置で特徴の比較と一致点の検出を試み
画像全体にわたって、特徴の一致件数を計算し特徴マップを作成

この時に実行する演算が畳み込み



畳み込みニューラルネットワークの名称の元

1. はじめに

1.1 背景

Convolutional Neural Networkについて

畳み込み演算の計算方法

1. フィルターの各ピクセルを入力画像内の対応するピクセル値で掛ける
2. 答えを合計し, フィルターのピクセルの総数で割る

(例) 両方のピクセルが白(1)なら $1 \times 1 = 1$

片方が黒(-1), 片方が白(1)なら $1 \times (-1) = -1$

ピクセルが一致する場合の計算結果は全て1で, 一致しない場合は全て-1

畳み込み演算を $\otimes$ で表記

1. はじめに

1.1 背景

Convolutional Neural Networkについて

入力

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

フィルター

1	-1	-1
-1	1	-1
-1	-1	1

特徴マップ

<http://postd.cc/how-do-convolutional-neural-networks-work/>

1. はじめに

1.1 背景

Convolutional Neural Networkについて

入力

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

フィルター

1	-1	-1
-1	1	-1
-1	-1	1

特徴マップ

<http://postd.cc/how-do-convolutional-neural-networks-work/>

1. はじめに

1.1 背景

Convolutional Neural Networkについて

入力の一部

-1	-1	-1
-1	1	-1
-1	-1	1

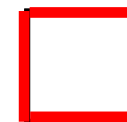


フィルター

1	-1	-1
-1	1	-1
-1	-1	1



特徴マップの1セル



<http://postd.cc/how-do-convolutional-neural-networks-work/>

畳み込み演算を $\otimes$ で表記

1. はじめに

1.1 背景

Convolutional Neural Networkについて

入力の一部

-1	-1	-1
-1	1	-1
-1	-1	1



フィルター

1	-1	-1
-1	1	-1
-1	-1	1

特徴マップの1セル

0.77

$$\frac{(-1)+1+1+1+1+1+1+1+1}{9}$$

=

$$\frac{7}{9}$$

=

$$\underline{\underline{0.77}}$$

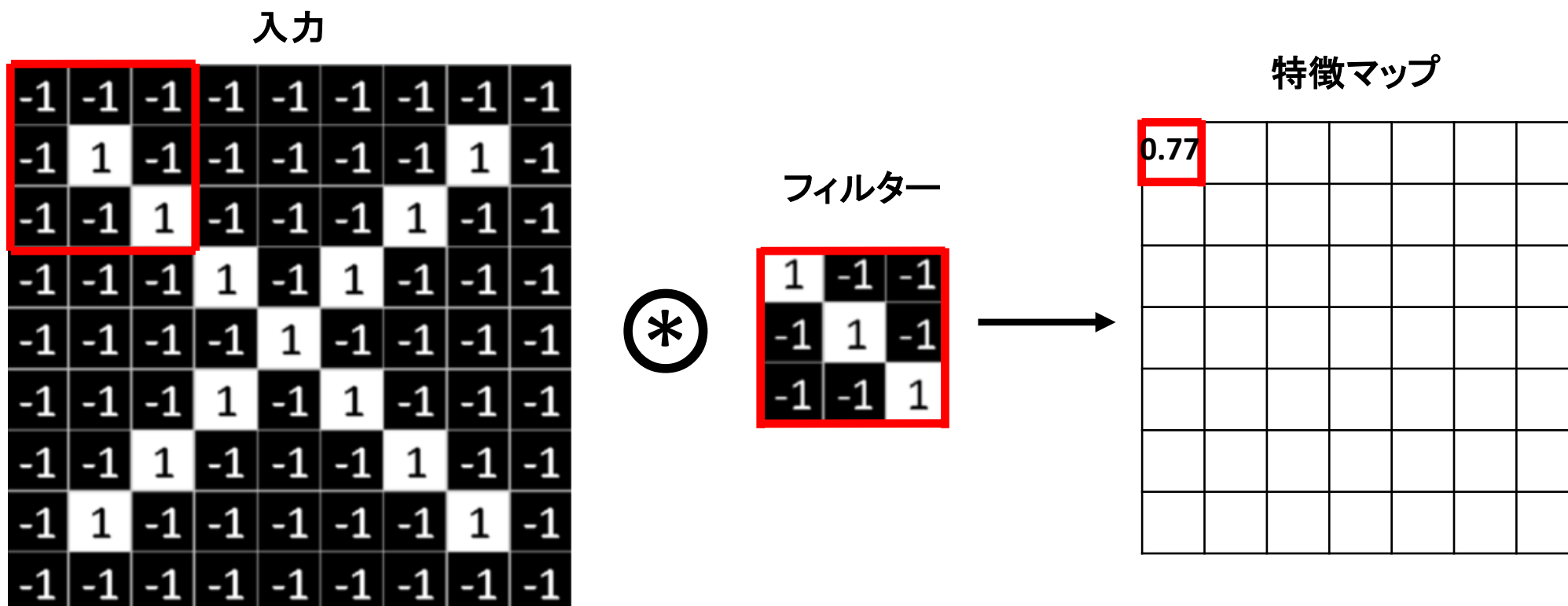
<http://postd.cc/how-do-convolutional-neural-networks-work/>

畳み込み演算を $\otimes$ で表記

1. はじめに

1.1 背景

Convolutional Neural Networkについて



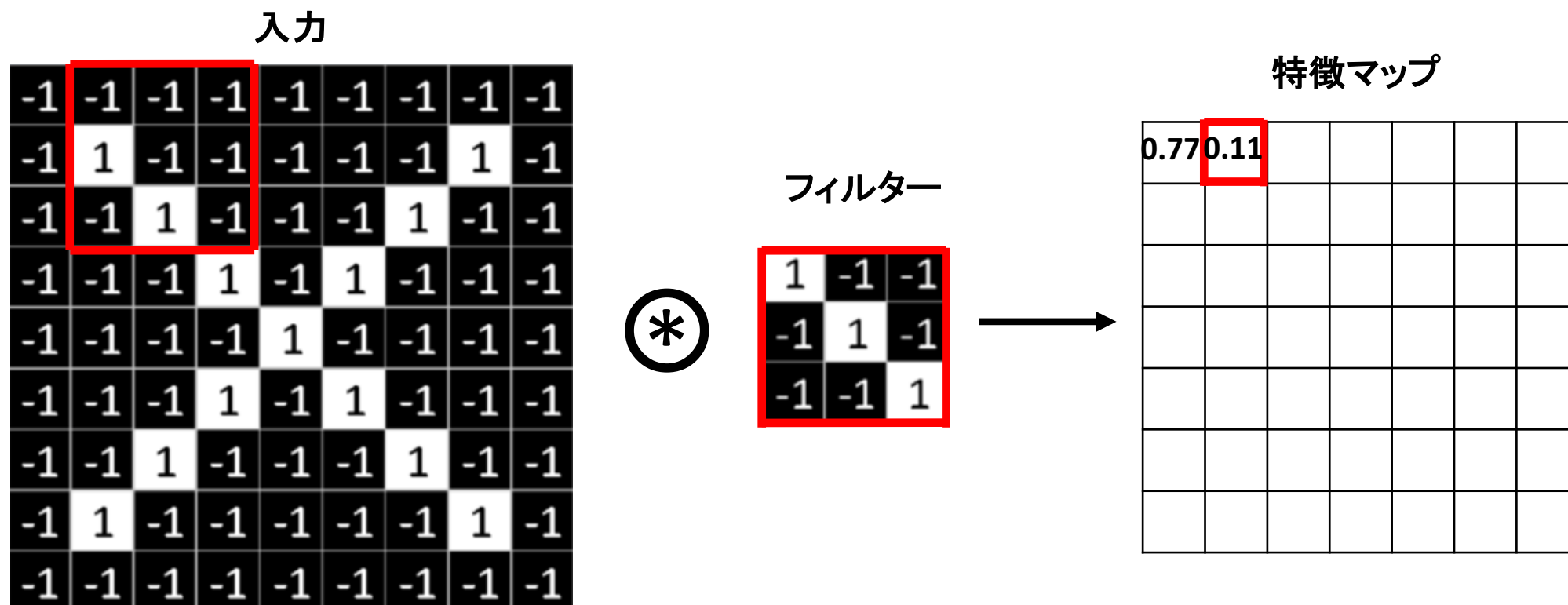
<http://postd.cc/how-do-convolutional-neural-networks-work/>

畳み込み演算を ⊗ で表記

1. はじめに

1.1 背景

Convolutional Neural Networkについて



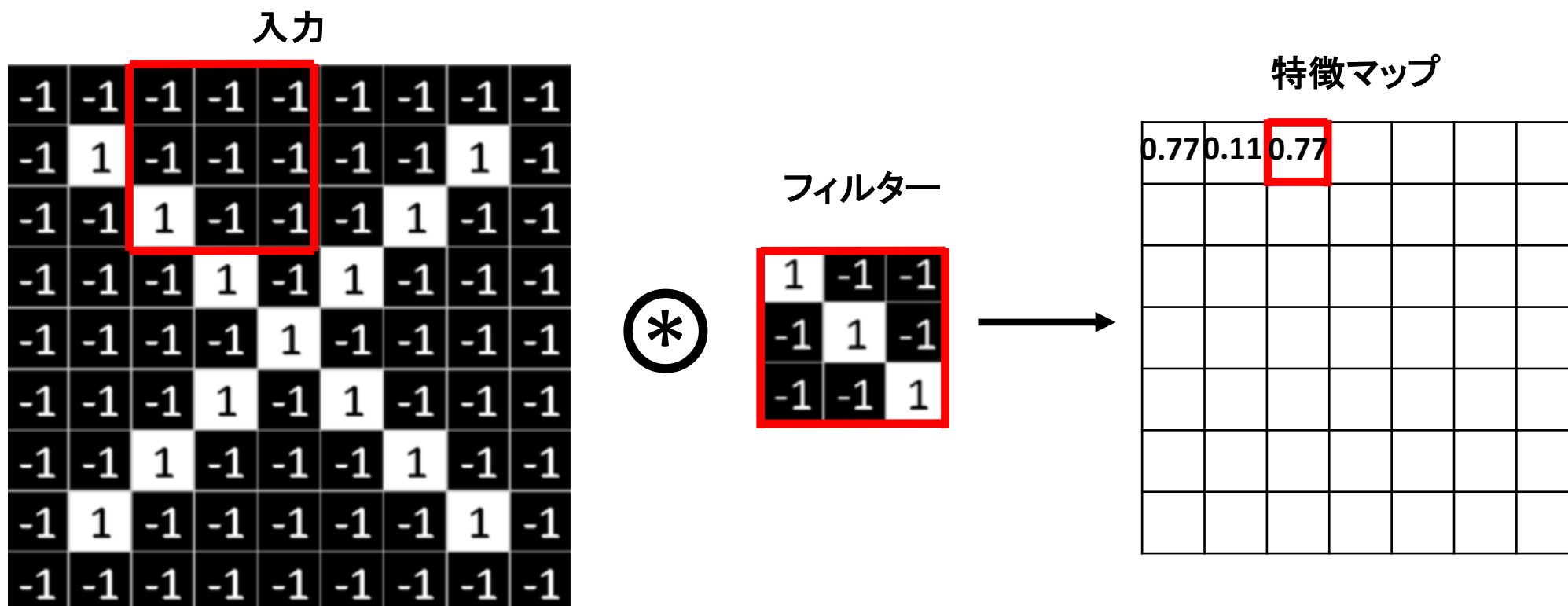
<http://postd.cc/how-do-convolutional-neural-networks-work/>

畳み込み演算を ⊗ で表記

1. はじめに

1.1 背景

Convolutional Neural Networkについて



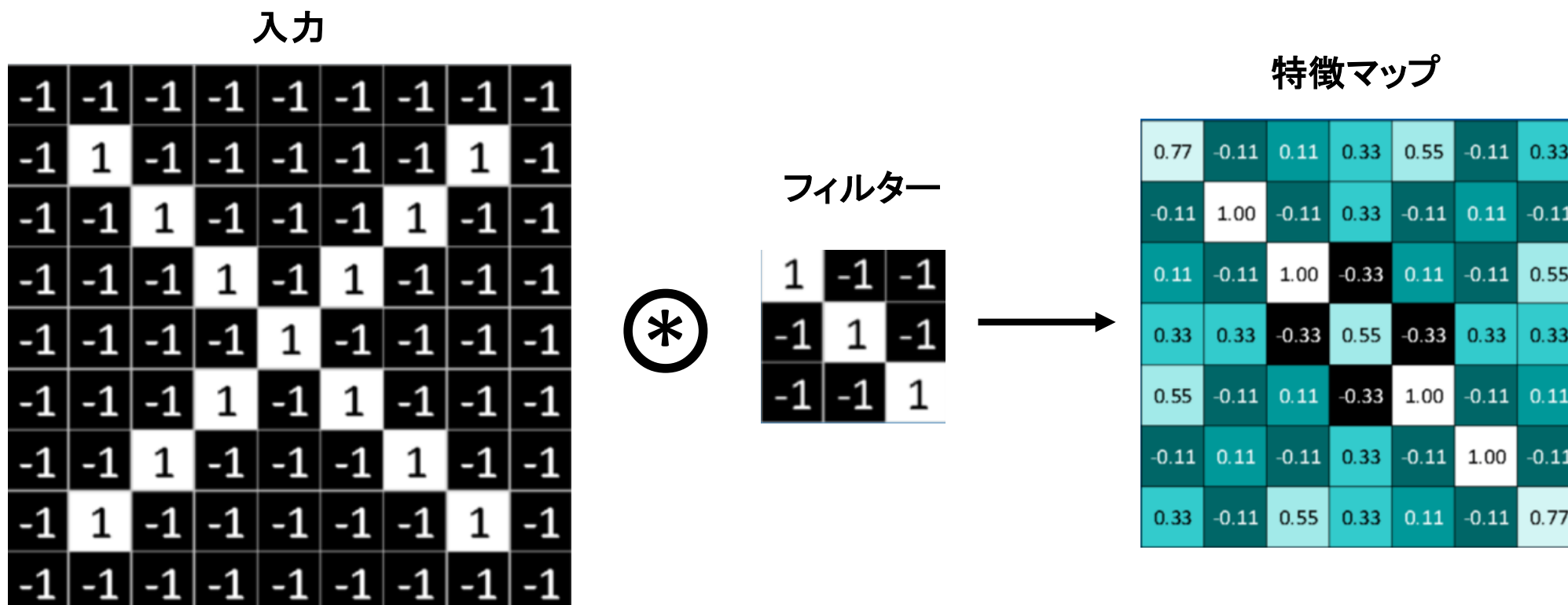
<http://postd.cc/how-do-convolutional-neural-networks-work/>

畳み込み演算を ⊗ で表記

1. はじめに

1.1 背景

Convolutional Neural Networkについて



<http://postd.cc/how-do-convolutional-neural-networks-work/>

畳み込み演算を ⊗ で表記

1. はじめに

1.1 背景

Convolutional Neural Networkについて

多くの加算や乗算，除算が繰り返されることによって
計算は画像内のピクセル数，各特徴のピクセル数に比例して増加

膨大な量の計算が必要



近年の計算性能の向上と計算機の安価化と共に
ディープラーニングは発達

1. はじめに

1.1 背景

Convolutional Neural Networkについて

プーリング層の役割

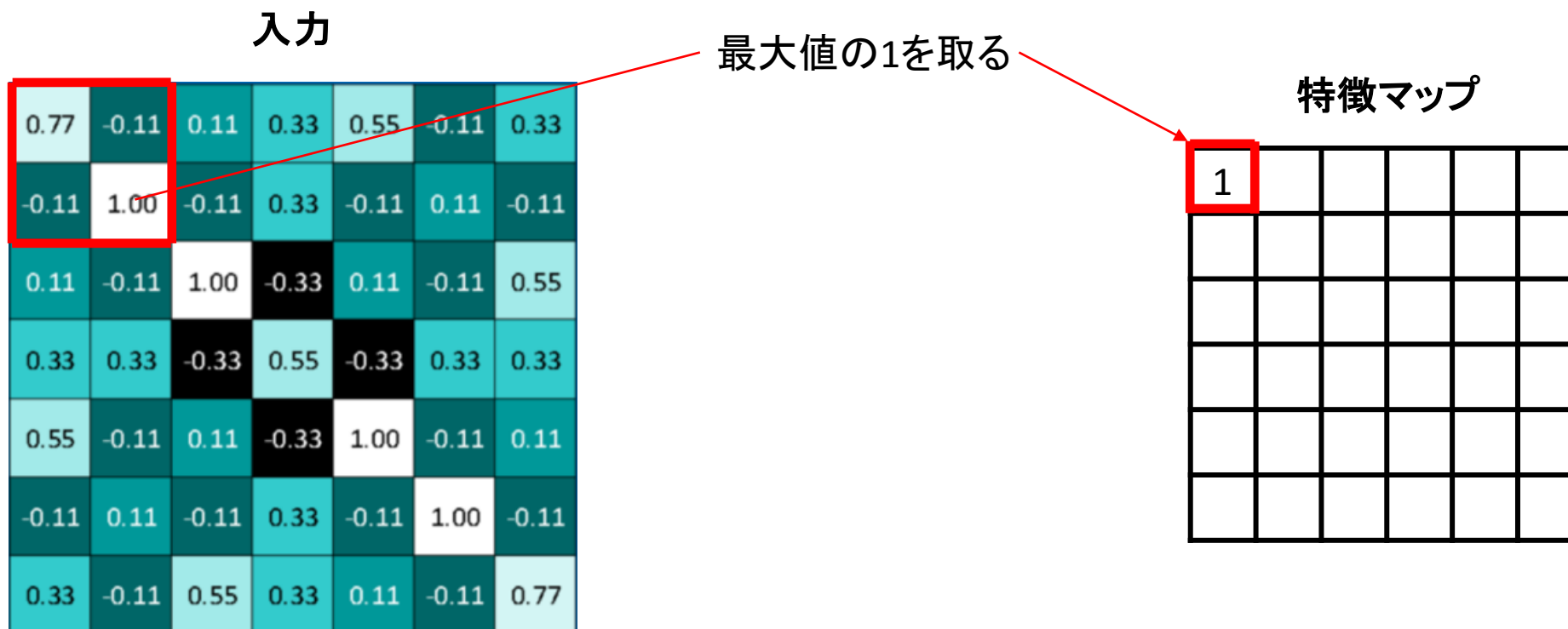
- ・ 畳み込み層から出力された特徴マップを縮小し新たな値を求める
- ・ 画像内を小さなウィンドウに区切り,
区切ったそれぞれのウィンドウから最大値を取る(max pooling)

大きな画像を重要な情報は残しつつ縮小する方法

1. はじめに

1.1 背景

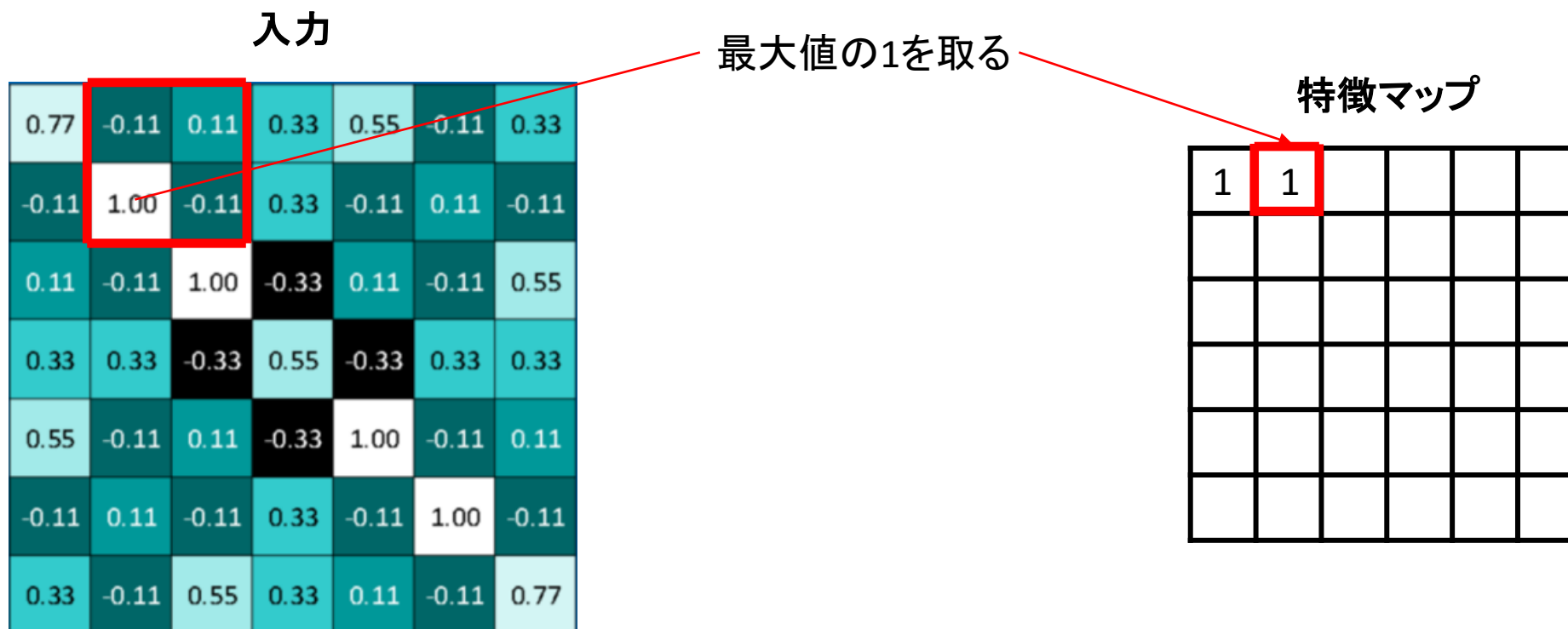
Convolutional Neural Networkについて



1. はじめに

1.1 背景

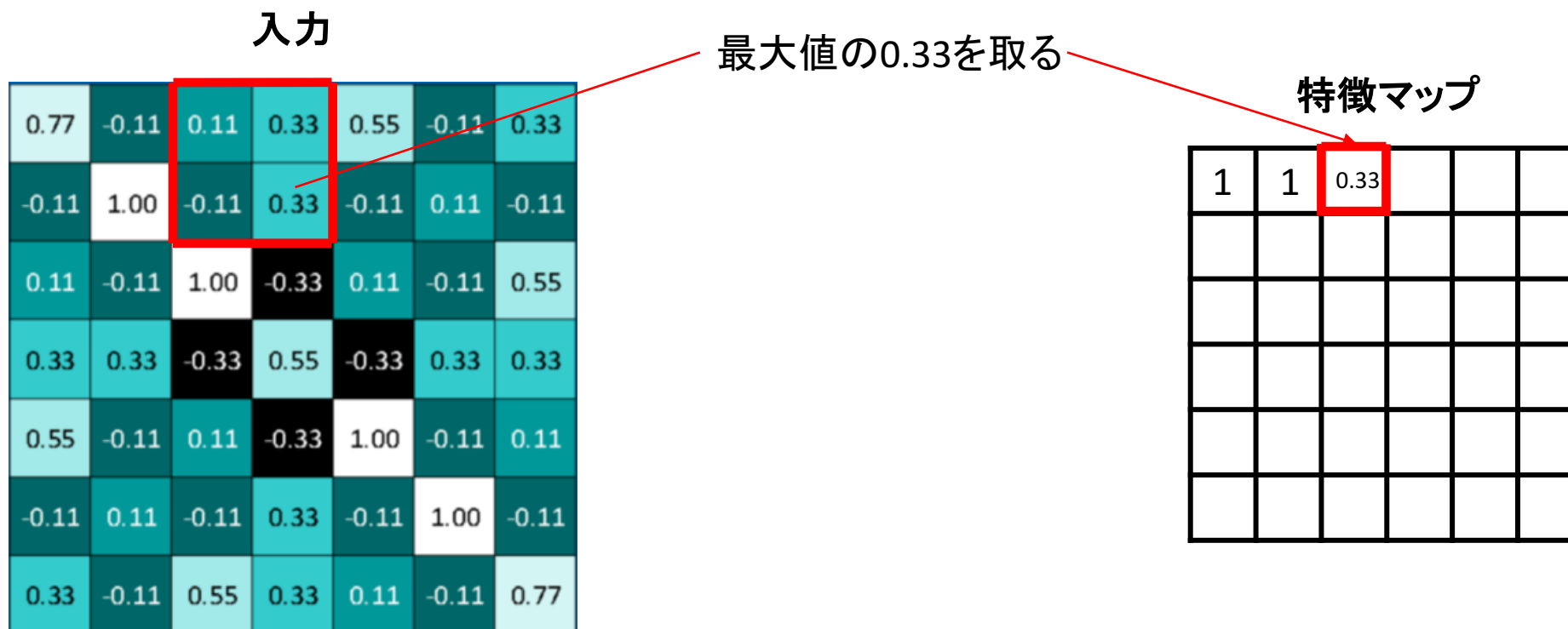
Convolutional Neural Networkについて



1. はじめに

1.1 背景

Convolutional Neural Networkについて



1. はじめに

1.1 背景

顔の系統について(1/2)

同じ日本人男性でも様々な顔の系統がある

代表的なのが 「**しょうゆ顔・ソース顔**」

しょうゆ顔・ソース顔

【年代】 1988年 【流行語大賞】 流行語部門・大衆賞 【種類】 若者言葉

男性の顔を分類する遊びが若い女性の間で流行。

ソース顔・しょうゆ顔以外にも『マヨネーズ顔』『ケチャップ顔』『みそ顔』など
様々なものがあったが、この二つだけが広く普及した。

当時の代表者は

<https://kw-note.com/marketing/1988-shingo-ryukogo-taisho/>

- ・ しょうゆ顔の代表 東山紀之
- ・ ソース顔の代表 錦織一清

1. はじめに

1.1 背景

顔の系統について(2/2)

代表的な顔の系統

しょうゆ顔

切れ長で一重・奥二重をした
和風の薄い顔立ち

例：東山紀之

ソース顔

彫りが深く目鼻がくっきりしている
洋風の濃い顔立ち

例：錦織一清

砂糖顔

いくつになっても少年っぽさが
残るベビーフェイスな顔つきの人

例：小池徹平

塩顔

一重か奥二重のシャープな
目元が特徴で中性的な人

例：竹内涼真

1. はじめに

1.2 目的・動機

動機

様々な系統の顔があり,
画像認識に適しているCNNを活用すれば分類できるのではないか

2. 演習内容・方法

2. 演習内容・方法

演習内容

CNNを活用して4系統の顔の分類

しょうゆ顔

切れ長で一重・奥二重をした和風の薄い顔立ち

例：東山紀之

砂糖顔

いくつになっても少年っぽさが残るベビーフェイスな顔つきの人

例：小池徹平

ソース顔

彫りが深く目鼻がくっきりしている洋風の濃い顔立ち

例：錦織一清

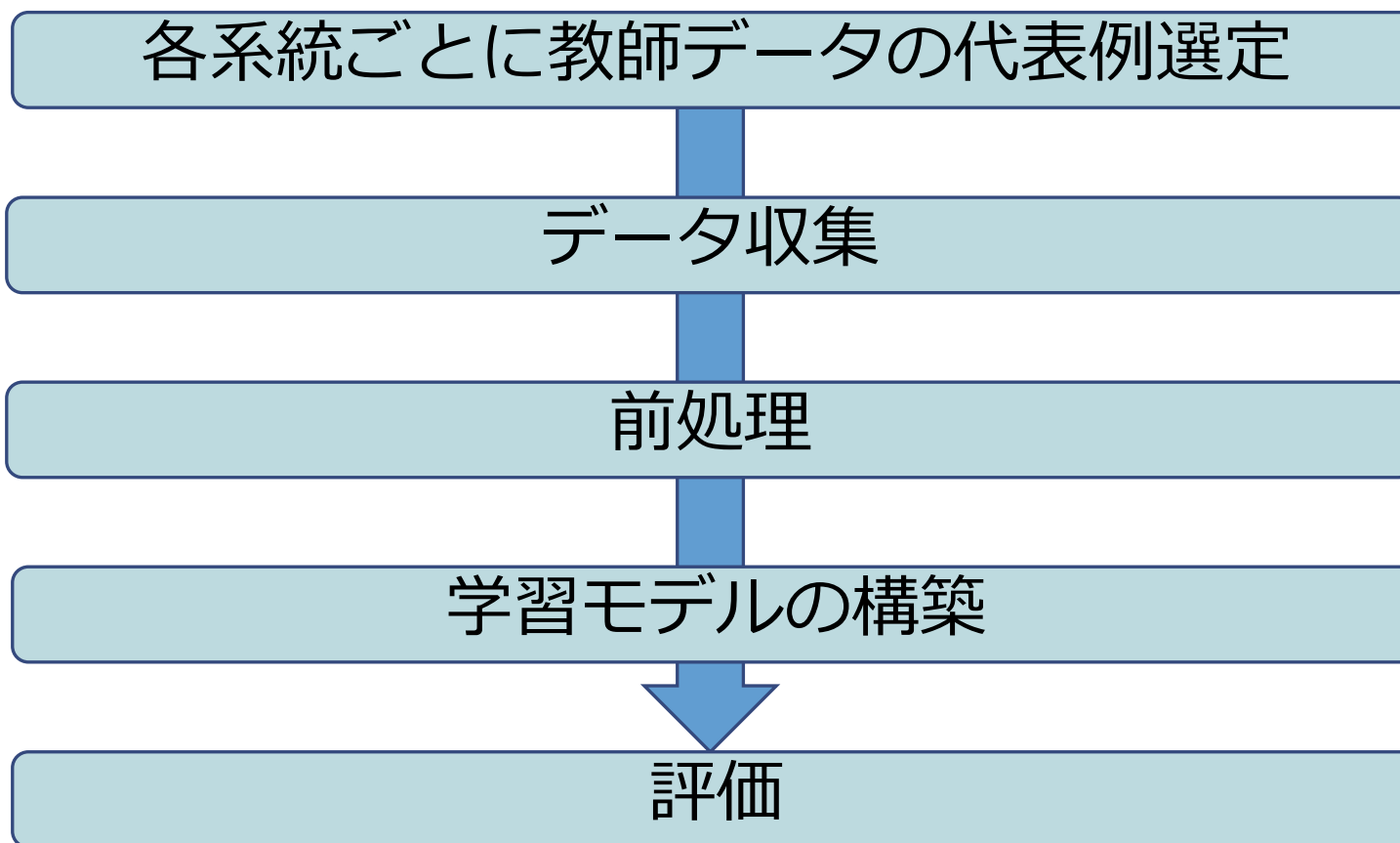
塩顔

一重か奥二重のシャープな目元が特徴で中性的な人

例：竹内涼真

2. 演習内容・方法

本演習の流れ



2. 演習内容・方法

2.1 系統の代表例選定

誰を各系統の教師データとするか決めることにする

主観でWebサイト等から4系統各7人ずつリストアップ

客観性を取り入れるため(アンケートにより)各系統を4人に絞る

系統の代表例とする

2. 演習内容・方法

2.1 系統の代表例選定

Webサイト等から4系統各7人ずつリストアップ

【しょうゆ顔】

小栗旬
木村拓也
相葉雅紀
佐々木蔵之介
二宮和也
福山雅治
大野智

【ソース顔】

高橋大輔
平井堅
反町隆史
長瀬智也
AKIRA
マイケル富岡
松本潤

【塩顔】

福士蒼汰
高橋一生
菅田将暉
星野源
坂口健太郎
加瀬亮
瑛太

【砂糖顔】

西島隆弘
瀬戸浩二
佐藤健
岩田 剛典
神木 隆之介
大倉忠義
山田涼介

2. 演習内容・方法

2.1 系統の代表例選定

Googleフォームでアンケートを友人や家族に対し配布・集計

顔の系統に関するアンケート

卒業演習「機械学習による顔の系統分類(仮)」のためのアンケート調査を行っています。

芸能人がどの系統の顔なのか教えてください。

顔の系統の候補は以下の4つの特徴の顔とします。

◆しょうゆ顔 . . . 一番日本人らしい顔
(例) 向井理

◆ソース顔 . . . 日本人離れした濃い顔
(例) 阿部寛

◆塩顔 . . . あっさりしたアジア人らしい顔
(例) 及川光博

◆砂糖顔 . . . 少年のような顔
(例) 小池徹平

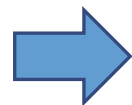
計77名が回答

2. 演習内容・方法

2.1 系統の代表例選定

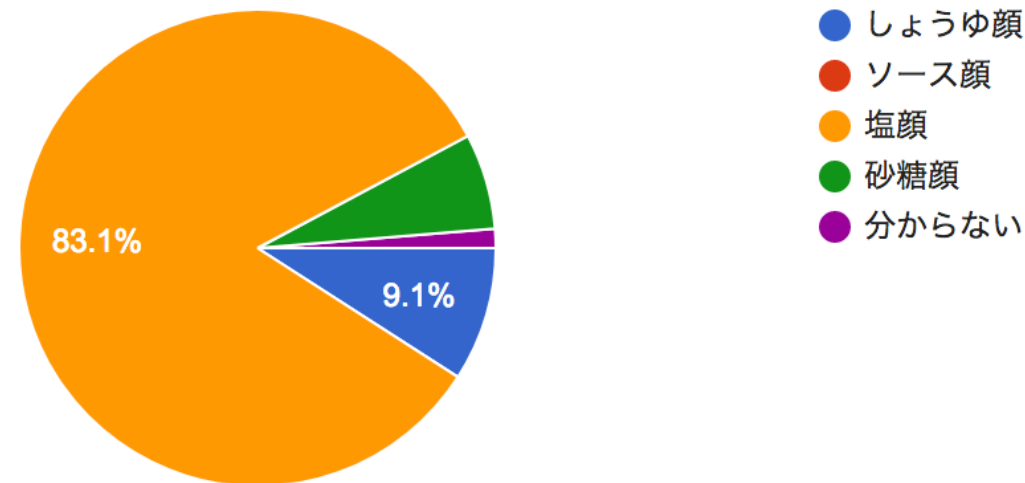
坂口健太郎 *

- ☐ しょうゆ顔
- ☐ ソース顔
- ☐ 塩顔
- ☐ 砂糖顔
- ☐ 分からない



坂口健太郎

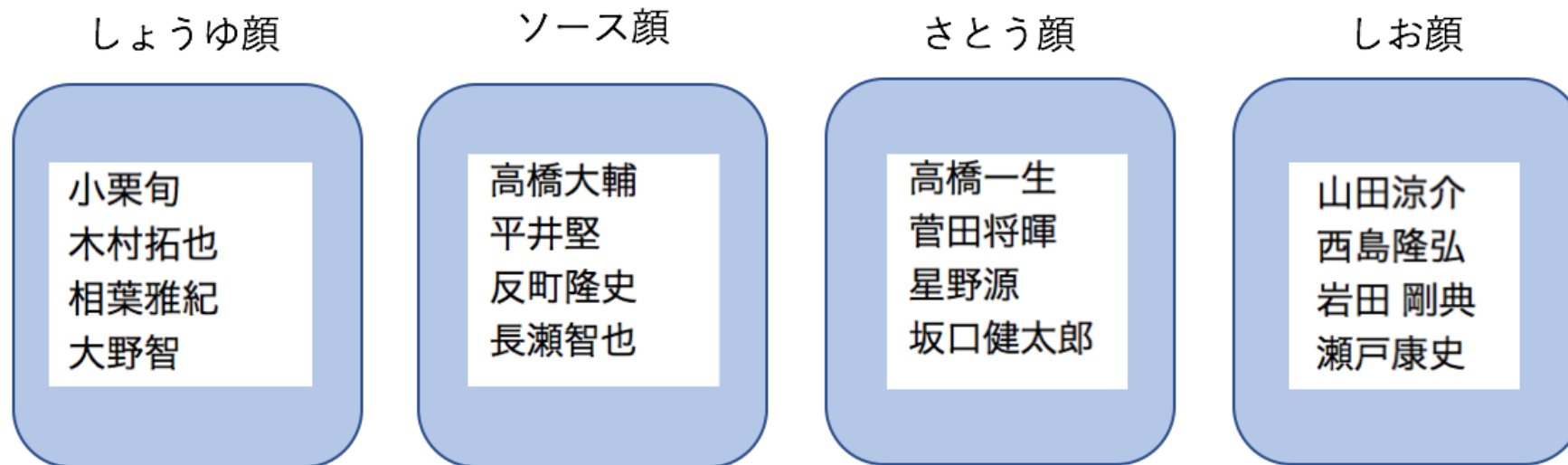
77 件の回答



2. 演習内容・方法

2.1 系統の代表例選定

アンケートを77人にした結果、
各系統で選ばれた割合の多い4人を各系統の教師データの
代表例として選定



2. 演習内容・方法

2.2 画像データの収集と前処理・水増し

画像データの収集

excite画像検索から収集

- ・ 4系統
- ・ 1系統につき4人
- ・ 1種につき160枚

$$4 \times 4 \times 160 = 2560$$

計2560枚

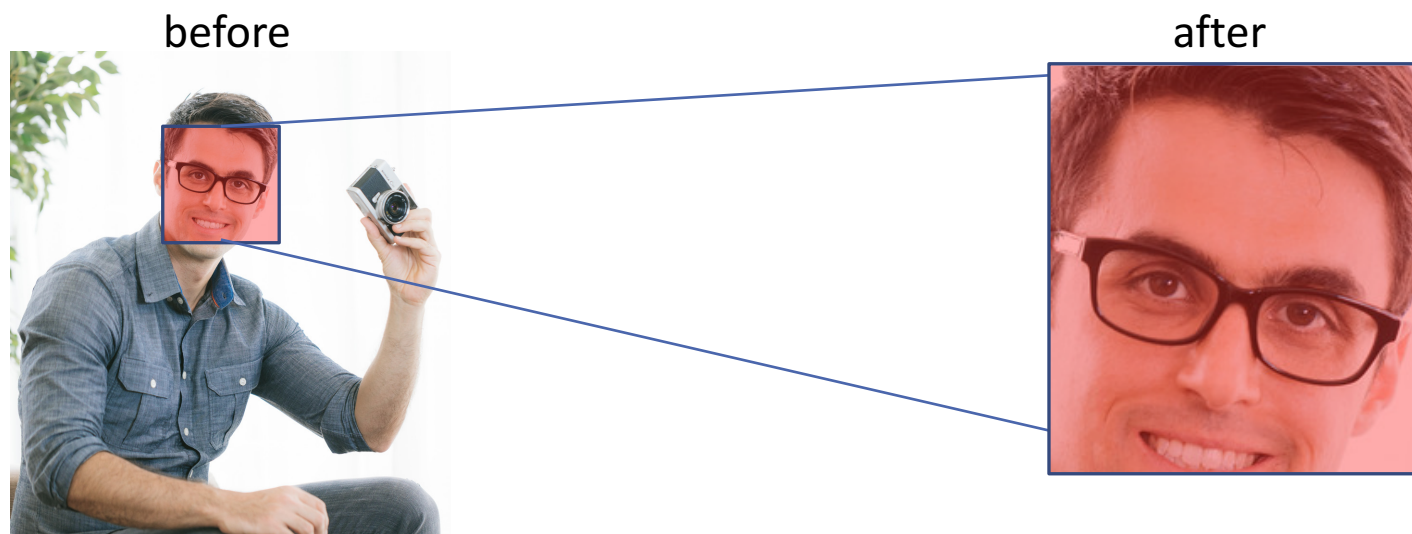
2. 演習内容・方法

2.2 画像データの収集と前処理・水増し

画像データの前処理

収集してきた各画像データに対し、

OpenCVを用いたプログラムにより画像データから顔部分をトリミング



ぱくたそ <https://www.pakutaso.com/20171032297post-13773.html>

2. 演習内容・方法

2.2 画像データの収集と前処理・水増し

画像データの前処理

収集してきた画像データの中には学習に不適切なものもあるため、手作業で以下に該当するものを削除する

- ・ 本人ではないもの
- ・ 顔が鮮明に写っていないもの
- ・ 画像サイズが極端に小さいもの
- ・ 顔の部分をトリミングできていないもの

2. 演習内容・方法

2.2 画像データの収集と前処理・水増し

画像データの前処理

収集してきた画像データの中には学習に不適切なものもあるため、手作業で以下に該当するものを削除する

- ・ 本人ではないもの
- ・ 顔が鮮明に写っていないもの
- ・ 画像サイズが極端に小さいもの
- ・ 顔の部分をトリミングできていないもの

1人につき**160枚** → **70枚**

2. 演習内容・方法

2.2 画像データの収集と前処理・水増し

画像データの水増し

データの水増しとはデータに様々な変換を加えることによってデータの数を増やすこと

<http://blog.aidemy.net/entry/2017/10/04/033336>

- コントラスト変更 4 種 (コントラスト変更無し, コントラスト(低, 中, 高))
- ノイズ付加 2 種 (ノイズ無し, ノイズ有り)
- 画像の反転 2 種 (左右反転無し, 左右反転有り)

コントラスト変更 × ノイズの付加 × 画像の反転 = 16種

2. 演習内容・方法

2.2 画像データの収集と前処理・水増し

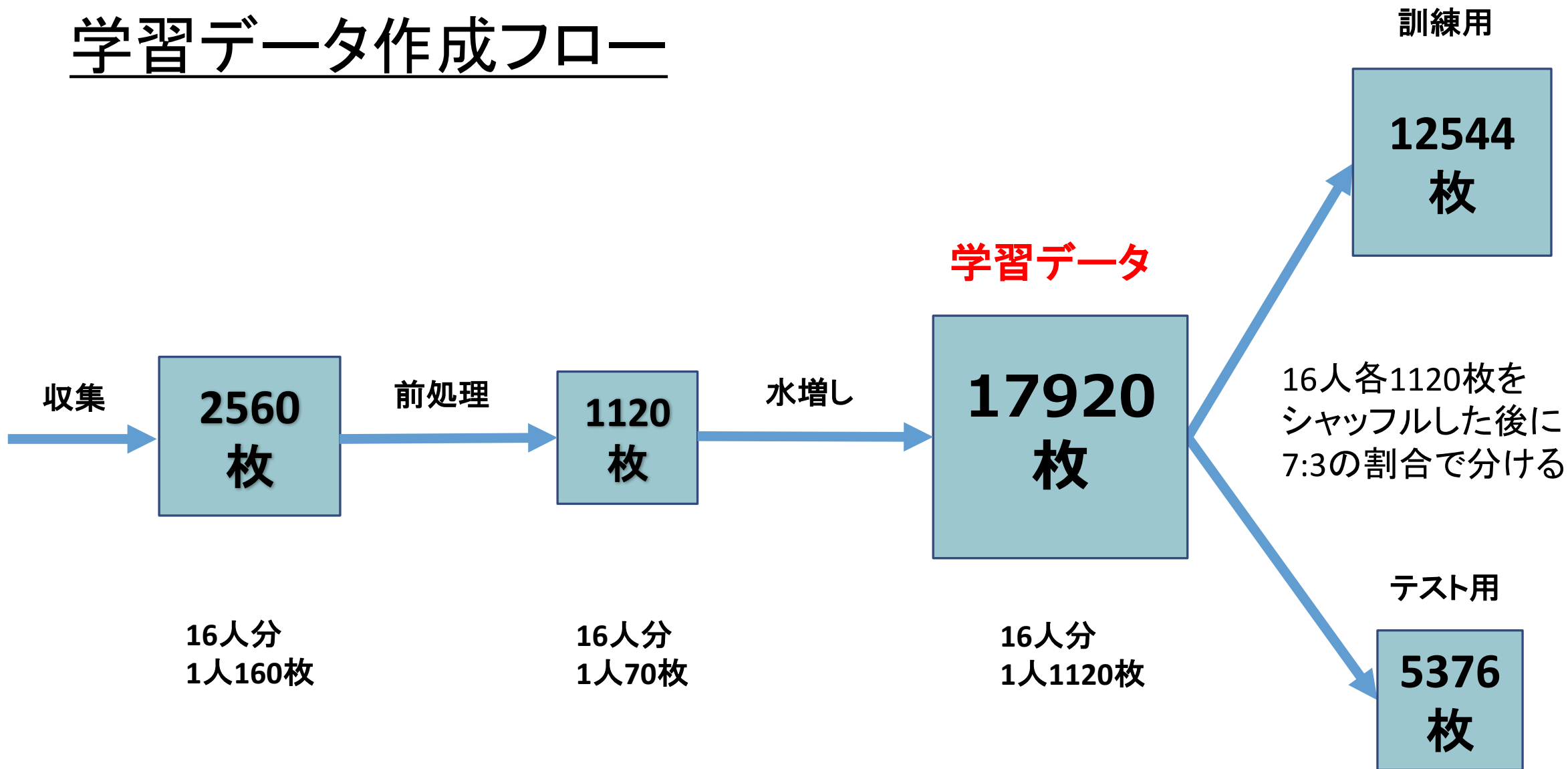
学習に用いる画像の枚数

- ・ 4系統
- ・ 1系統につき4人
- ・ 1種につき70枚
- ・ 1枚を16枚に水増し

$$4 \times 4 \times 70 \times 16 = 17920$$

以上 計**17920**枚 を学習に用いる

学習データ作成フロー



2. 演習方法

2.3 学習モデルの構築(1/2)

実行環境

MacBook	OS	CPU	モデル	クロック	メモリ
	Sierra	Intel Core m5	Early 2016	1.2 GHz	8GB

開発環境

言語

Python
3.5.1

ツール

	TensorFlow	Keras	OpenCV
バージョン	1.2.1	2.0	3.2.0

2. 演習方法

2.3 学習モデルの構築(2/2)

構築モデル

TensorFlow Deep MNIST for Experts
チュートリアルを参考に構成

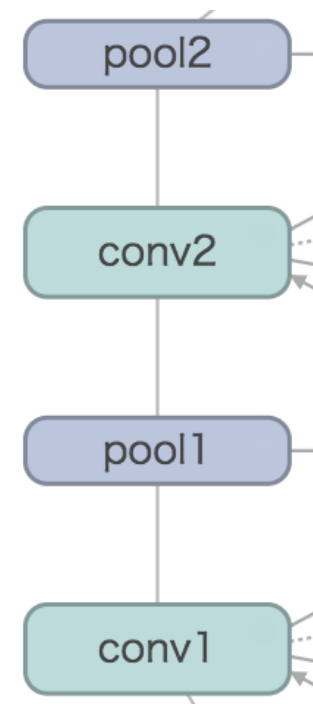
使用ツール

- TensorFlow
- Keras

中間層

- 畳み込み層 2層
- プーリング層 2層

中間層の構成



3. 演習結果・考察

3. 演習結果・考察

3.1 結果(1/2)

TensorFlowとKeras, 2つのツールを用いて同じ構成のモデルを構築し収集してきた画像を学習させた結果

使用ライブラリ	accuracy(正答率)
TensorFlow	0.796788
Keras	0.787482

学習時間はTensorFlowが3時間半, Kerasが 1 時間

3. 演習結果・考察

3.1 結果(2/2)

checker(評価器)を作成

コマンドライン上で画像を与えると
学習データに基づき予測ラベルを返す

実行結果

```
*-----*
      分類結果
*-----*
さとう 顔  : 67.0310556889 %
しお  顔  : 32.0989668369 %
しょうゆ 顔 : 0.799790769815 %
ソース 顔  : 0.0701905810274 %

『 reiji.jpg 』は『さとう 顔』と予測されます
```

3. 演習結果・考察

3.2 考察

考察

画像データのラベル付けや前処理・水増し方法により精度が変わるため、より良い学習データが用意できるのではないか

今後の課題

- ・ 本演習の結果を評価
- ・ 学習データを利用しWebやアプリ上で画像を与えることによって分類ができるものを作成