プロジェクト研究B

～視線認識 with SVM, CNN～

研究テーマ

顔画像からの視線認識

What?

顔の画像から視線（カメラを見てるかみてないか）を検出する

Why?

視線を認識することによってよりインタラクティブなシステムが構築可能

e.g.) 対話システム、自動販売、オートロック解除システム

How?

1. 画素値をそのまま入力としてSVMで分類
2. 画像特徴量（SIFT）を抽出してSVMで分類
3. CNNを使って分類

データの準備～分類までの流れ

1. 学習データの用意

用意した画像

正例：約1300枚、負例：約600枚（研究室から頂きました）

それぞれにラベル（正例→0、負例→1）が付いている

正例（0）

負例（1）

データの準備～分類までの流れ

1. 学習データの用意

①OpenFaceというAPIを使って、顔の部分のみを切り出す
②ヒストグラム平坦化を行う

元画像

顔切り出し
（96×96）

ヒストグラム平坦化

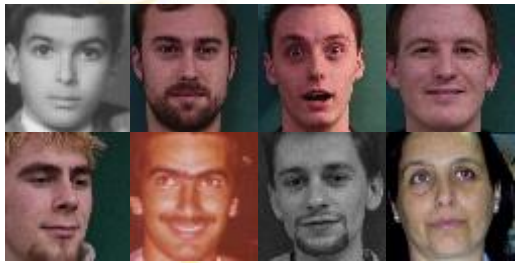
データの準備～分類までの流れ

2. 学習

用意したデータを用いて、モデルを学習させる。

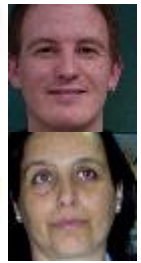
この時、訓練データの一部をテストデータとして分離し、その正答率から精度を評価する。

用意したデータ



訓練データ

テストデータ



9 : 1

データの準備～分類までの流れ

3. 分類

未知のデータに対して分類を行う

分類器

0→見てる
or
1→見てない

学習アルゴリズム

1. 画素値をそのままSVMに入力

ヒストグラム平坦化を行った後の画像の上1/3のみの画素値(0~255)を出力
横96*縦32=3072ピクセル

[1, 14, 18, 19, …, 200, 160
⋮ ⋮ ⋮
5, 10, 51, 12, …, 120, 100]

学習アルゴリズム

1. 画素値をそのままSVMに入力

グリッドサーチを行った結果、最適なパラメータは、

poly kernel（多項式カーネル）

C=1, degree=3, gamma=0.001

正答率は、0.81330であった。

試したパラメータ

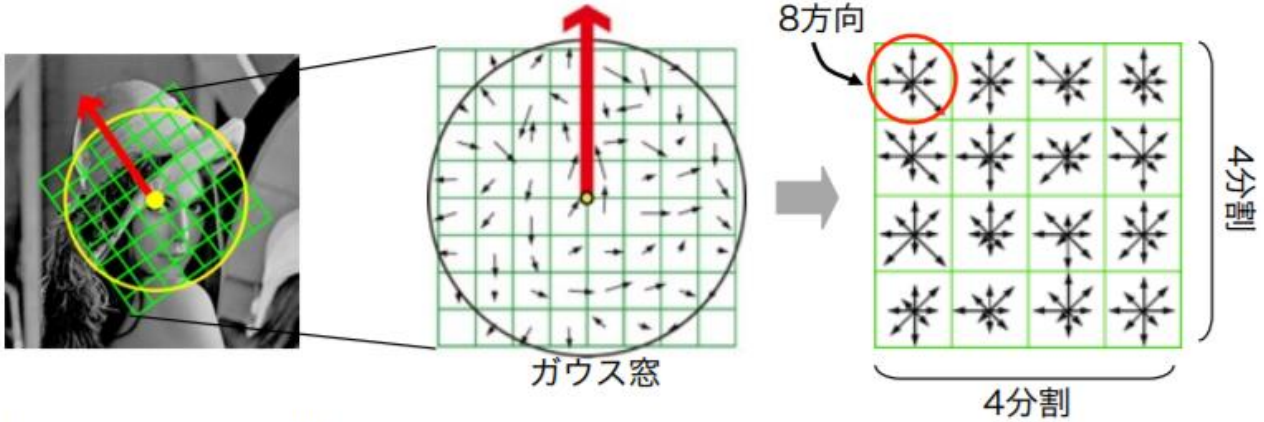
```
{'kernel': ['linear'], 'C': [1, 10, 100, 1000]},
 {'kernel': ['rbf'], 'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001]},
 {'kernel': ['poly'], 'C': [1, 10, 100, 1000], 'degree': [2, 3, 4], 'gamma': [0.001, 0.0001]},
 {'kernel': ['sigmoid'], 'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001]}
```

学習アルゴリズム

2. SIFT特徴量を抽出してSVMに入力

SIFT特徴量とは、

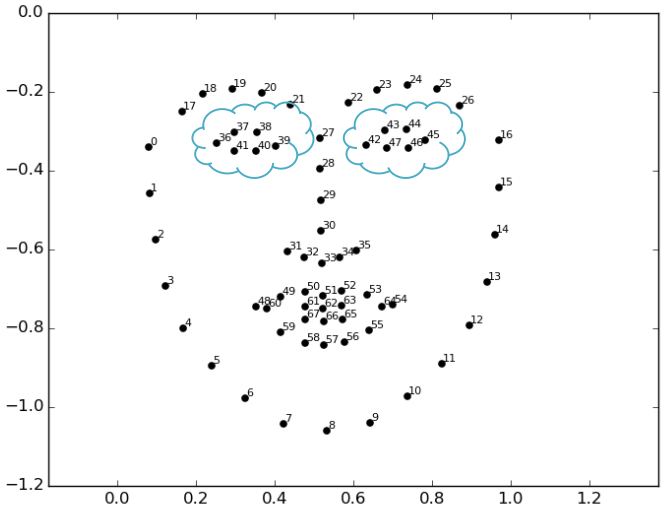
周辺の勾配強度を数値化したもの（通常は8×16=128次元）



学習アルゴリズム

2. SIFT特徴量を抽出してSVMに入力

目の周辺の12点において、SIFT特徴量を抽出→128×12=1536次元



学習アルゴリズム

2. SIFT特徴量を抽出してSVMに入力

グリッドサーチを行った結果、最適なパラメータは、

poly kernel（多項式カーネル）

C=1, degree=4, gamma=0.001

正答率は、0.82744であった。

試したパラメータ

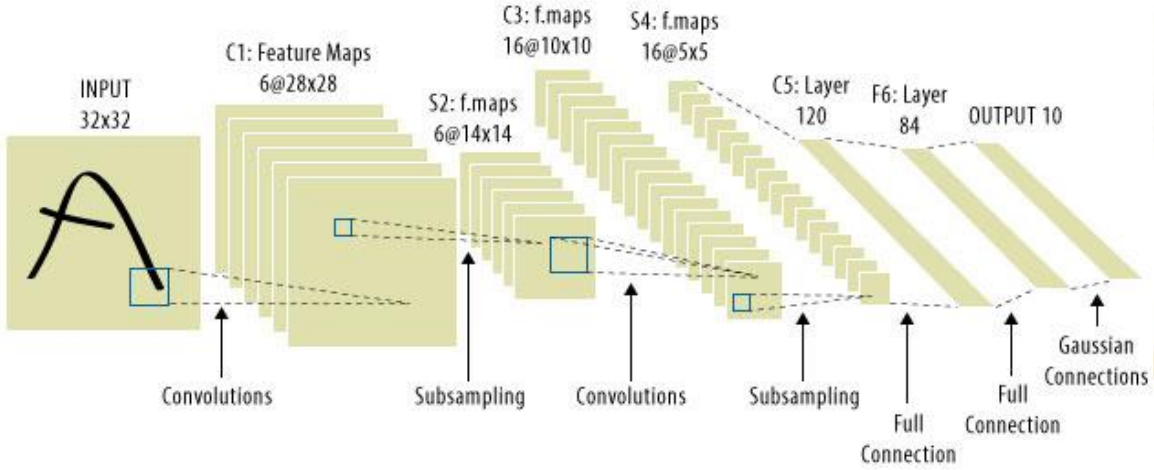
```
{'kernel': ['linear'], 'C': [1, 10, 100, 1000]},
{'kernel': ['rbf'], 'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001]},
{'kernel': ['poly'], 'C': [1, 10, 100, 1000], 'degree': [2, 3, 4], 'gamma': [0.001, 0.0001]},
{'kernel': ['sigmoid'], 'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001]}
```

学習アルゴリズム

3. CNNを使って分類

CNNとは？

画像に何種類ものフィルタを適用し、特徴を抽出していくネットワーク



学習アルゴリズム

3. CNNを使って分類

用意した画像

ヒストグラム平坦化を行った後の画像の上1/3のみ

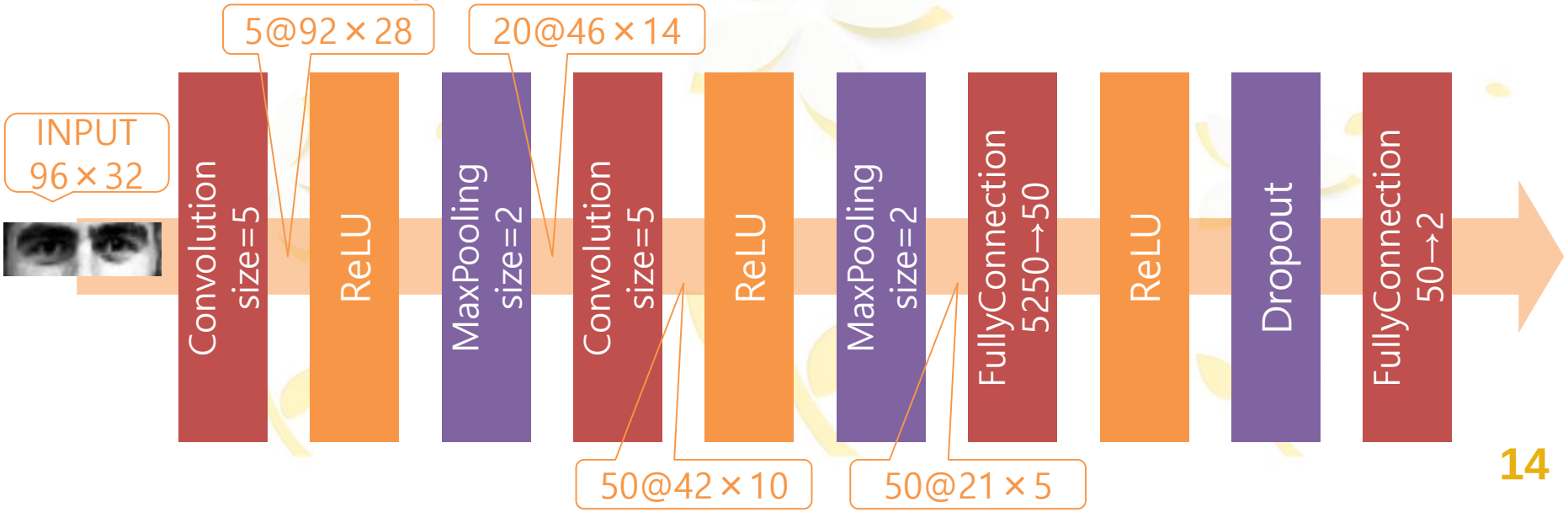
横96*縦32=3072ピクセル

[1, 14, 18, 19, …, 200, 160
⋮ ⋮ ⋮
5, 10, 51, 12, …, 120, 100]

学習アルゴリズム

3. CNNを使って分類

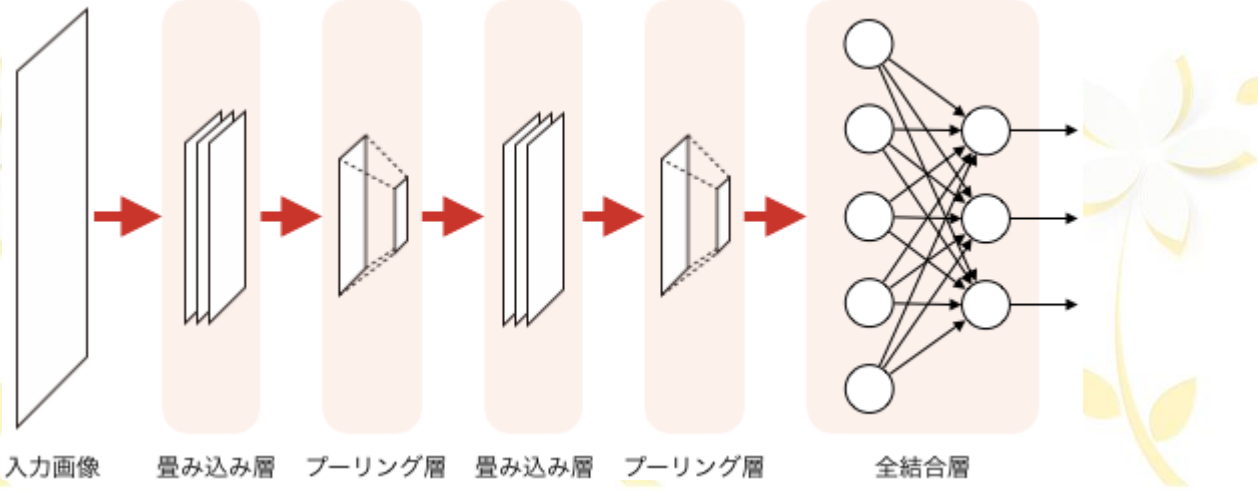
ネットワークの構造



学習アルゴリズム

3. CNNを使って分類

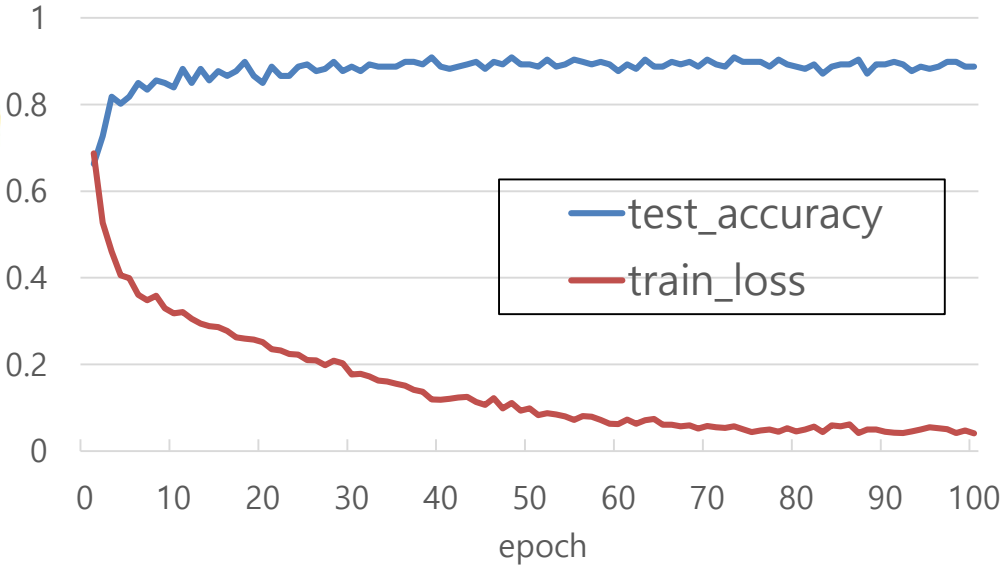
各層の役割



学習アルゴリズム

3. CNNを使って分類

一からネットワークを学習させた結果、最終的な正答率は0.8877であった



学習アルゴリズム

各手法の比較

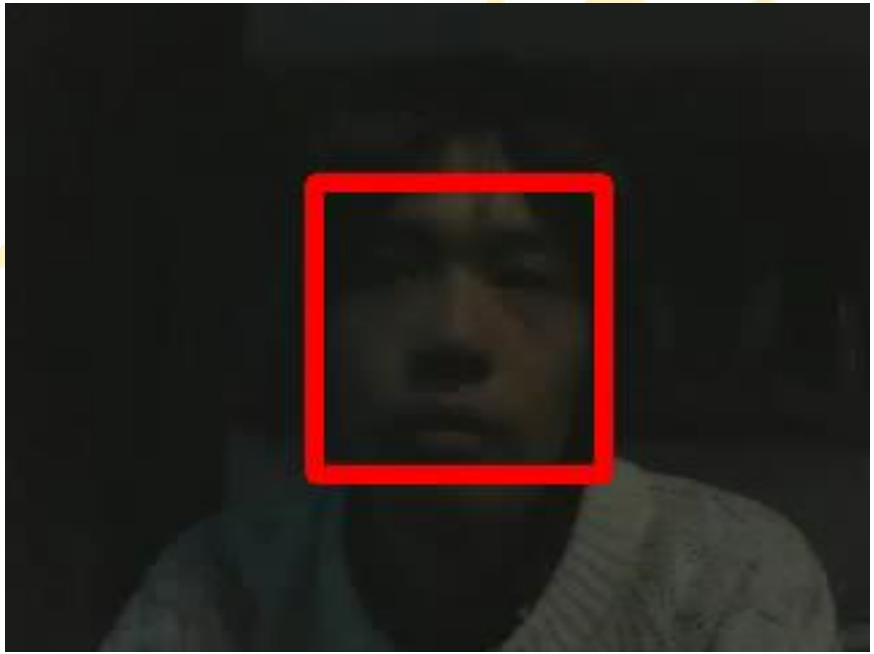
手法	SVM（画素値そのまま）	SVM（SIFT）	CNN
入力	96*32の画素値	SIFT特徴量128*12	32*96の画素値
正答率	81.3%	82.3%	**88.7%**

リアルタイムでの分類

画素をそのまま入力としたSVM

リアルタイムでの分類

CNNバージョン



考察

課題を通して学んだこと

- OpenFaceやscikit-learn、Chainerなどのツールの使い方
- グレースケール化やヒストグラム平坦化など、データの前処理テクニック
- ニューラルネットワークの偉大さ

考察

精度をより上げるために

リアルタイム（動画）の視線を検知したい場合、複数の画像を系列データとして与えると、より精度が上がるかもしれない。

→RNN, 3D-Convlution など

または、複数のフレームでの出力から判定する。

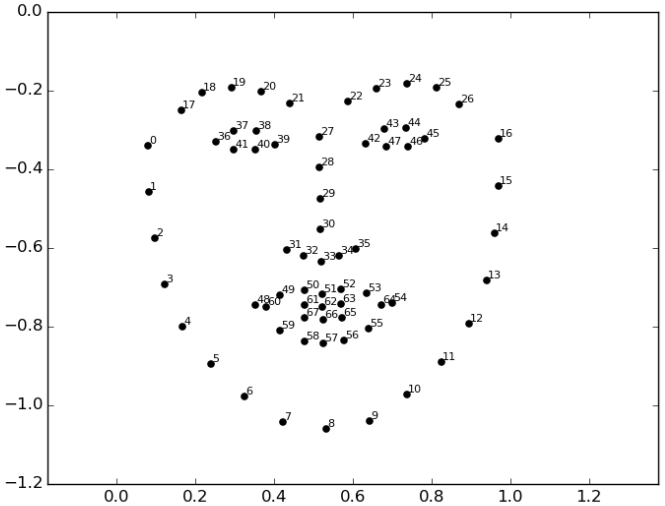
更に深い層のネットワークを用いることで、精度が上がる可能性がある。

もしくは、ImageNetを学習済みのモデル（AlexNetなど）を利用してみる。

Tips

OpenFaceでできること

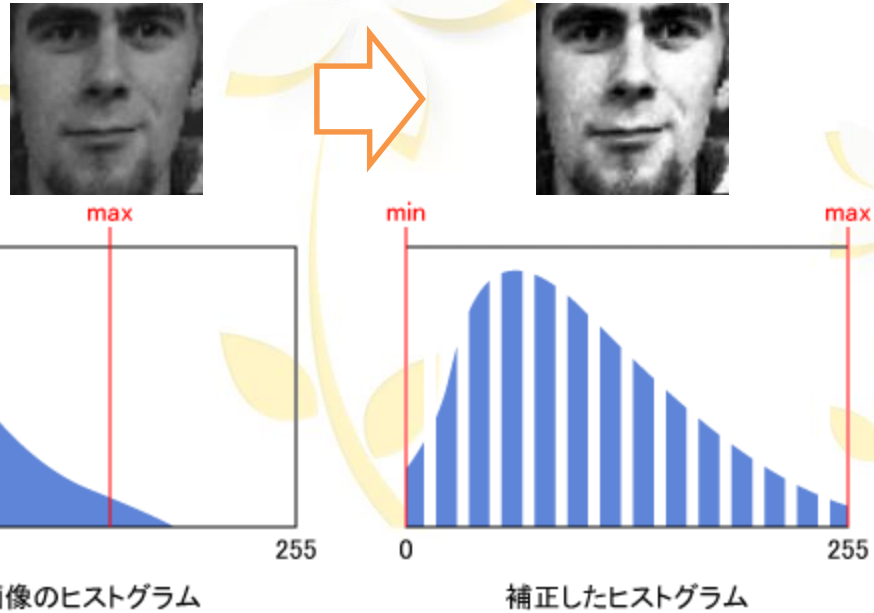
顔のパーツの座標を取得して、特定の部分を切り出すことができる。



Tips

ヒストグラム平坦化

輝度の分布を一定にすること。認識の精度が上がることがある



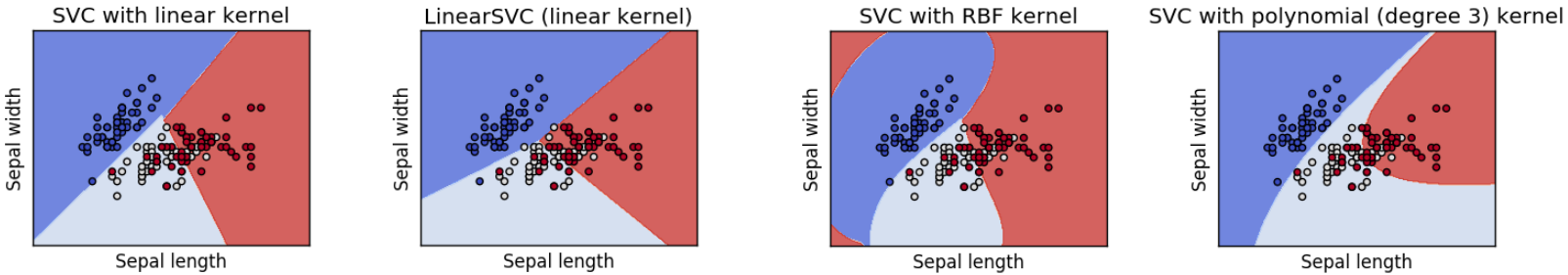
Tips

グリッドサーチ

SVMにおいて、どのカーネルを使うか、パラメータをどう調整するかは重要である。

線形カーネル、多項式カーネル、RBFカーネル、シグモイド

カーネルによって、パラメータの調整も必要（C, degree, gamma など）

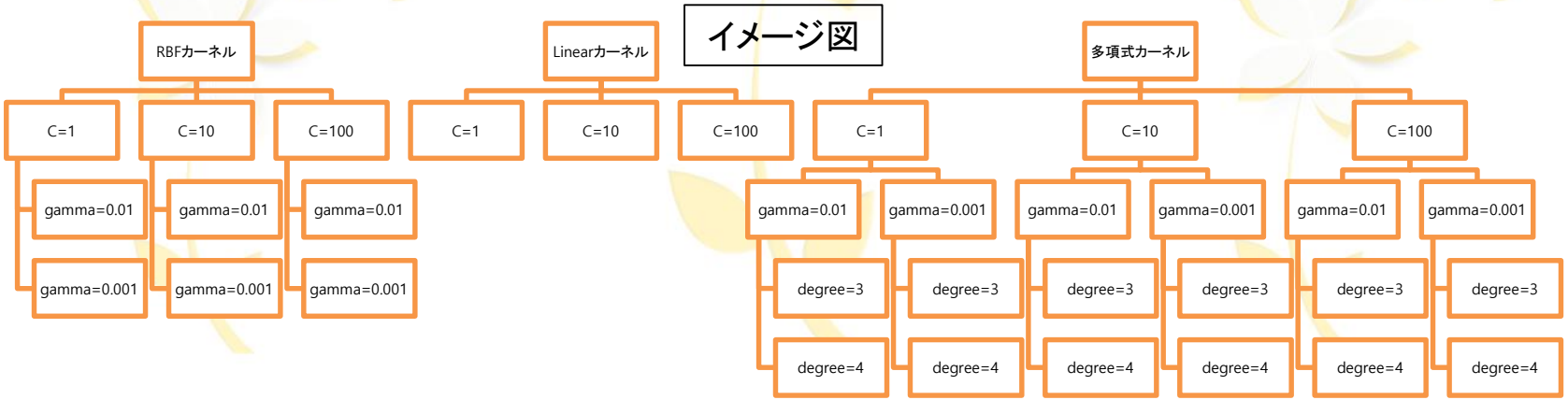


Tips

グリッドサーチ

グリッドサーチとは？

パラメータの組み合わせを全通り試した上で、最適なパラメータを選択する手法。



参考文献

- OpenFace(https://cmusatyalab.github.io/openface/)
- scikit-learn(http://scikit-learn.org/stable/)
- Convolutional Neural Networkとは何なのか(http://qiita.com/icoxfog417/items/5fd55fad152231d706c2)
- 畳み込みニューラルネットワークの仕組み(http://postd.cc/how-do-convolutional-neural-networks-work/)
- ヒストグラムの拡張・平坦化によるカラー画像の補正(https://codezine.jp/article/detail/214)
- 画像局所特徴量と特定物体認識(http://www.vision.cs.chubu.ac.jp/cvtutorial/PDF/02SIFTandMore.pdf)