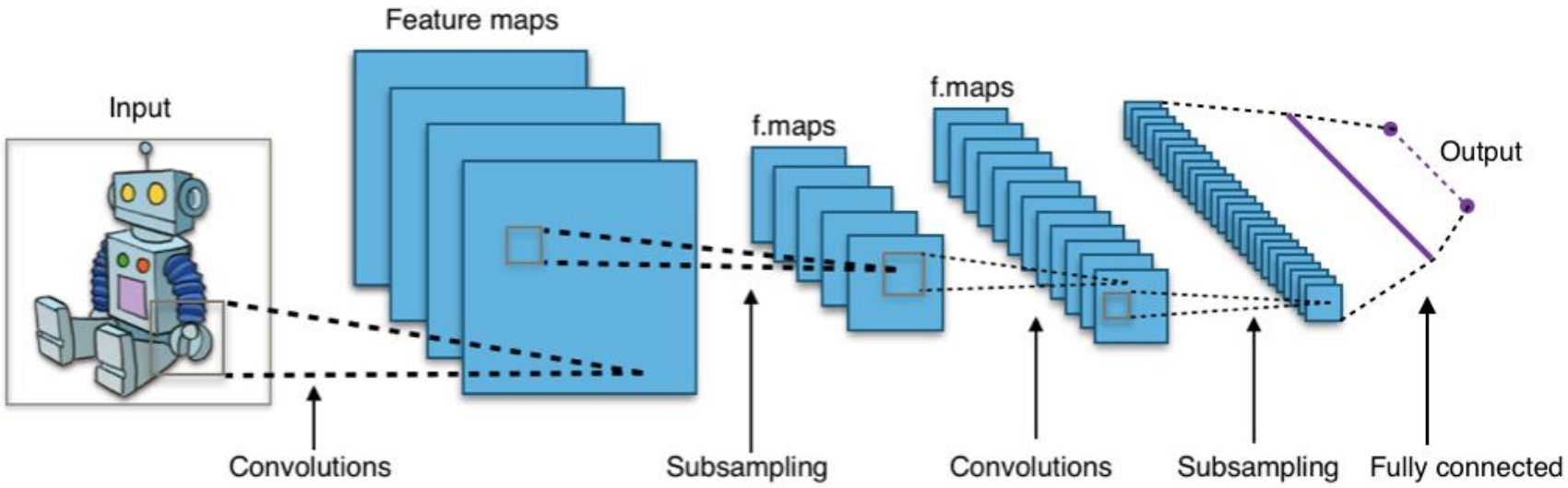
CNN
(Convolutional Neural Network)
と画像の水増し
(Image Augmentation)

Convolutional Neural Network (CNN) の基礎知識

- CNNの一般的な構造



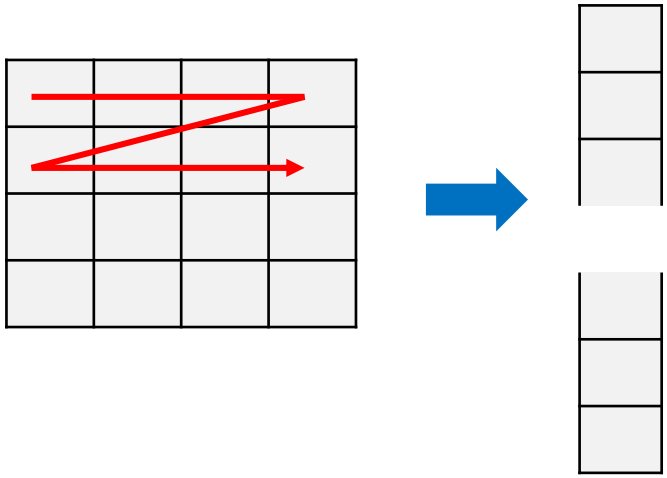
https://en.wikipedia.org/wiki/Convolutional_neural_network

- 画像データの処理
  - 畳み込み（Convolution）
  - プーリング（Pooling）（Subsamplingの一種）

画像情報の弱点

ニューラルネットワークへの入力は1次元の形にするため、2次元の画像情報が1次元になってしまう　→　画像の特徴が失われる

特徴を活かす形で入力できないといけない　→　解決策の1つがCNN

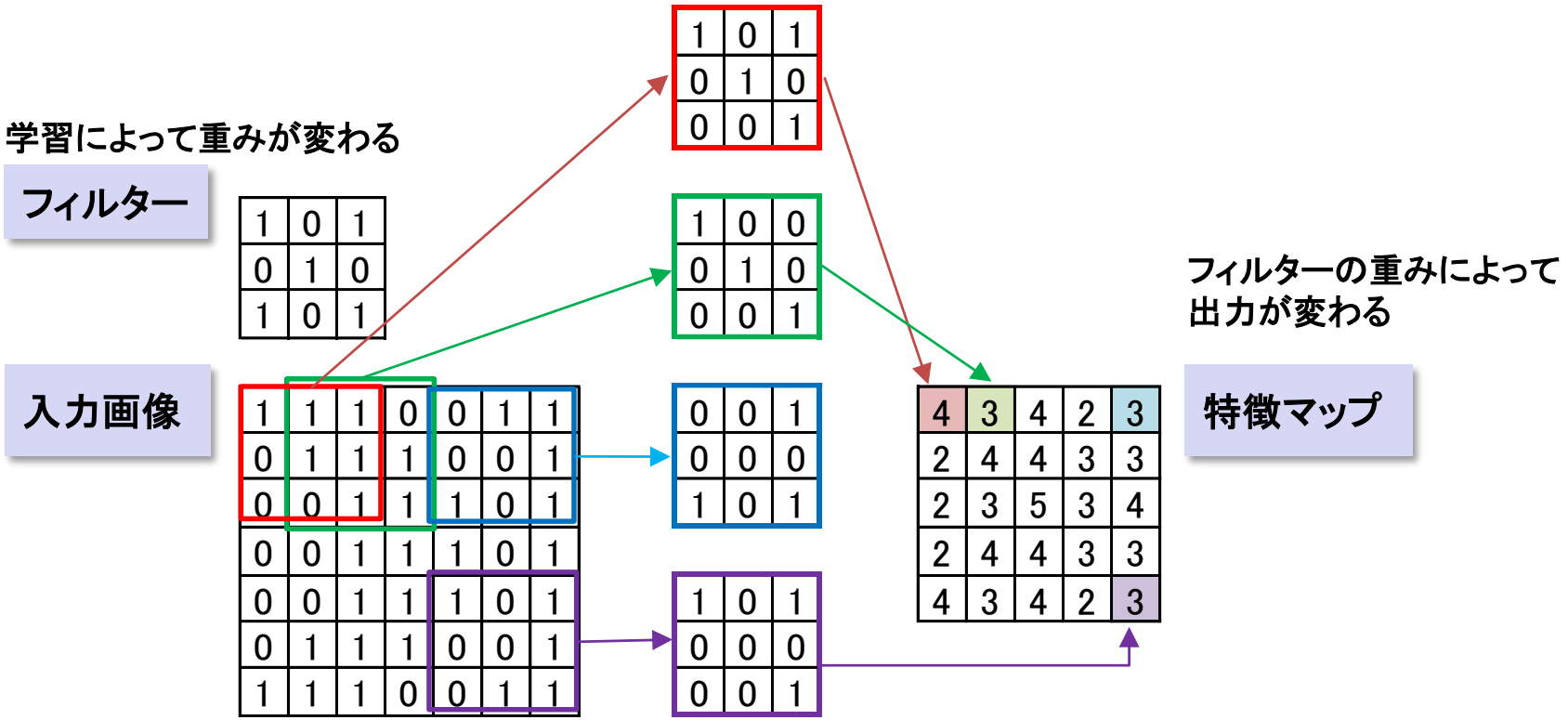


畳み込み（Convolution）

- 画像に小さなフィルターをかけ、その値をピクセル値として新たな画像を生成する
- 参考：https://ja.wikipedia.org/wiki/%E7%95%B3%E3%81%BF%E8%BE%BC%E3%81%BF (Wikipedia: 畳み込み)
- フィルターの重みを学習によって変化させることで、特徴を表す画像を生成する
  - 畳み込みで生成された特徴画像は特徴マップとも呼ばれる
- 後述するパディング、ストライドの大きさによって生成される画像のサイズが異なる

畳み込み（Convolution）

- 入力データに対してフィルターをかけることで、特徴マップを作成する
- 例：カーネルサイズ：3×3、ストライド：横1、縦1



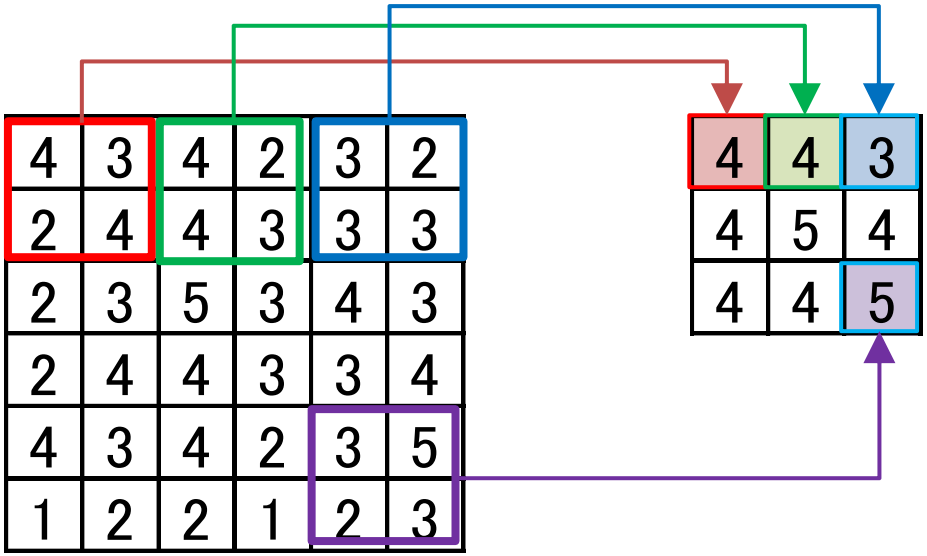
7_1_画像の畳み込みとプーリング.ipynbを参照

プーリング（Pooling）

- サブサンプリングの一種
- ある領域ごとに代表値を求め、その値をピクセル値として新たに画像を生成する
- 結果として画像サイズが小さくなる（解像度が低くなる）
  - Max pooling：最大値を代表値とする
  - Average pooling：平均値を代表値とする
  - Sum pooling：合計値を代表値とする
- 解像度が低くなることで、特徴の位置の多少のずれに頑健になる
- 領域の大きさによって、縮小率が変わる
  - 領域が2×2、ストライド（後述）が2の場合、サイズは縦1/2、横1/2の1/4になる。

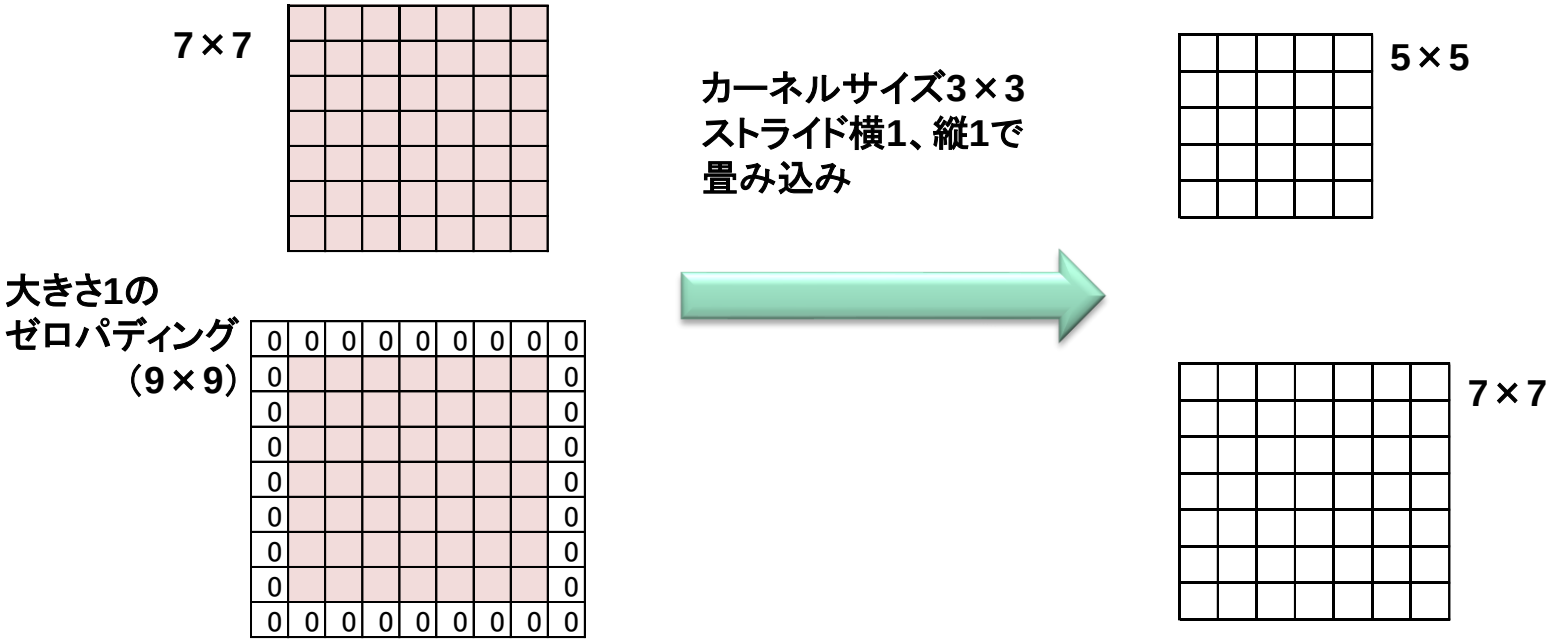
プーリング（Pooling）

- Max poolingの例
- ウィンドウサイズ（カーネルサイズ）：2×2
- ストライド：横2、縦2



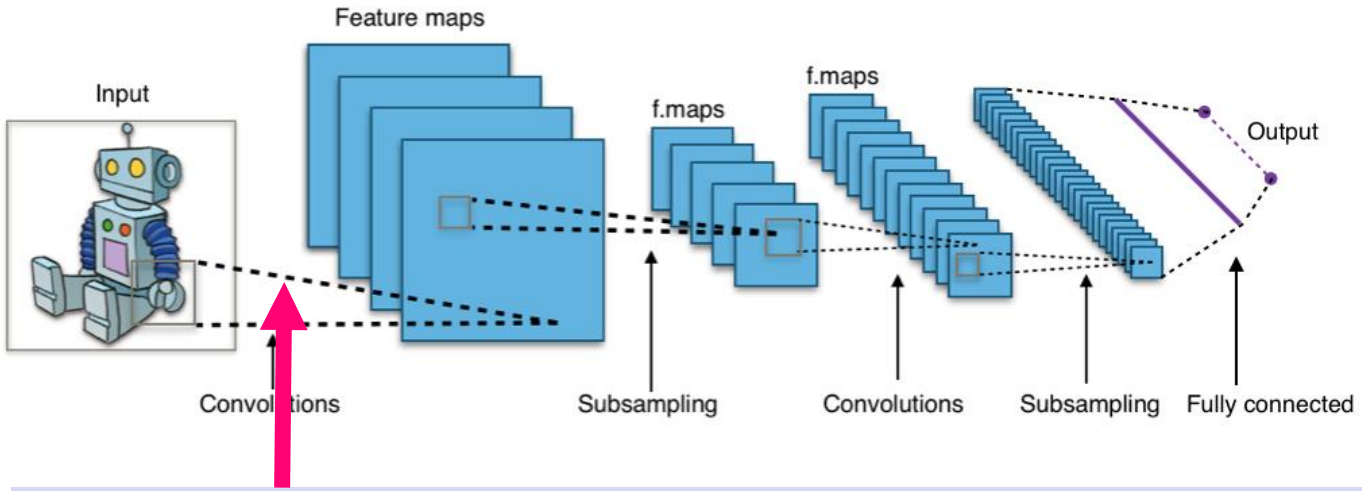
パディング（Padding）

- 畳み込みを行うと、生成される画像サイズは、元の画像よりも小さくなる
- 元の画像の外側を仮に何らかの値で埋め「ふち」をつくることで、生成される画像サイズを調整する
- 0で「ふち」を埋めるゼロパディングが良く使われる



畳み込みの意義

- 入力画像から目的にマッチした**特徴量を学習によって生成**する
  - 畳み込みフィルターの重みを学習によって決定する
- 中間層が全結合ではないので、計算量が少なくなる



もし全結合にすると、入力画素数と特徴マップの画素数×特徴マップ数の結合が必要

https://en.wikipedia.org/wiki/Convolutional_neural_network

CNNとDNNの比較演習

7_2_MNIST分類.ipynbを参照

演習のねらい

- KerasによるCNNの基本的な実装
- DNNとCNNによる分類の精度比較

精度の改善

CNNの表現力増加
- Convolution層やAffine層を深くする
- Convolution層の特徴マップ（OutMaps）数を増やす
- Affine層のOutput数を増やす
- KernelShape（フィルターサイズ）を小さくする

勾配消失の回避
- 層を深くすると勾配消失により誤差が小さくても精度が悪くなる
- 活性化関数をReLUにすると勾配損失を防ぐことがある

過剰適合の回避
- 層を増やしすぎない
- DropOut層を入れる（ランダムにノードを無効化し表現力を落とす）

BatchNormalization層を入れる
- 一部の入力のみの影響が大きくなりすぎることを防ぐ

自作データでの検証

7_3_自作手書き文字認識.ipynbを参照

演習のねらい

分類モデルが自作の手書き文字（準備されたデータ以外のデータ）でも分類できる汎用的な分類モデルであることを確認する

手書き数字データ

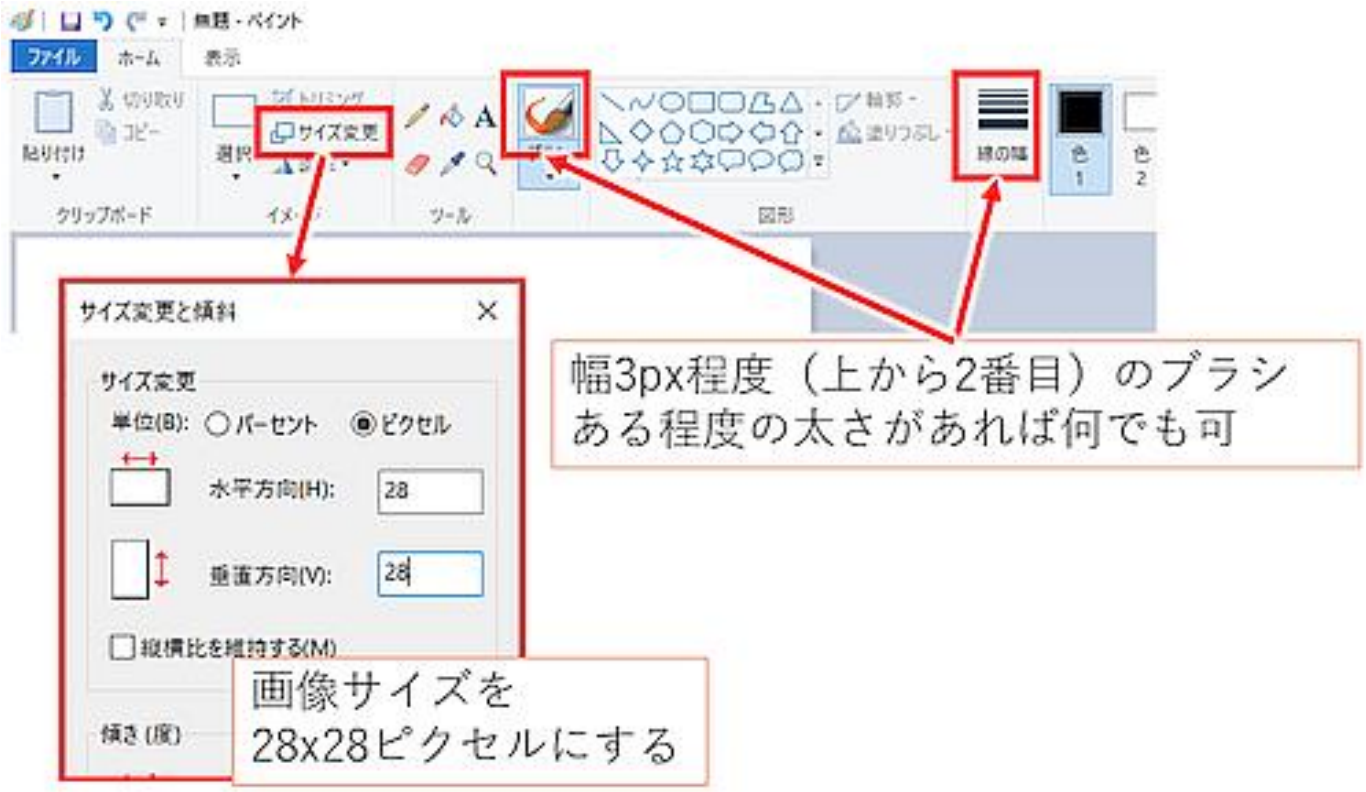
- 背景が黒で文字が白のグレイスケール画像
- サイズは28×28

画像データ作成

- ペイントソフトなどで文字を書く
  - ひとまず今回はペイントソフトで文字を書きます
- 紙に書いた文字をスキャナやカメラで画像化する

自作手書き画像の作成

ペイントを起動し、画像サイズを28x28に変更する
マウスで数字を書く（幅3px程度）
jpgとして保存したあと、GoogleDriveにアップロードする



画像データの水増し(Image augmentation)

学習には、画像内の対象の差異がある大量の画像データが必要
対象の差異の種類（手書き数字の場合）

- 文字の形
- 画像内での位置
- 画像内での文字が閉める割合（サイズ）
- アスペクト比
- 回転、反転
- その他変形
- 明るさ、コントラスト、彩度

傾きやサイズ、アスペクト比などが学習データに無い場合、精度が落ちる
画像処理で生成できるものも多い

- 一つの画像に様々な画像処理を組合せて適用し、学習データを増やす
- 予測の安定性につながる

Kerasによる画像の水増し

keras.preprocessing.image.ImageDataGeneratorの引数は次のとおり

- rotation_range:回転（degree）
- width_shift_range：横方向の移動（ピクセル）
- height_shift_range：縦方向の移動（ピクセル）
- shear_range：歪みの角度（degree）
- zoom_range：スケーリングの範囲（等倍=1.0）
- horizontal_flip：横方向反転（True or False）
- vertical_flip：縦方向反転（True or False）

画像データ水増し演習（1）

7_4_画像の水増し.ipynbを参照

演習のねらい

- 基本的な画像処理により水増しされる画像が、どのような画像になっているかを確認する
  - パラメータを一つずつ変更して、画像処理による加工の様子を確認する

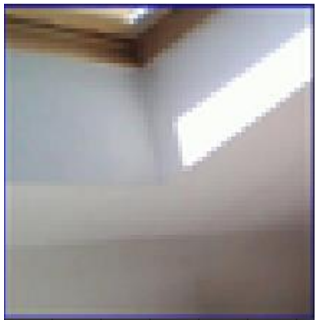
画像データ水増し演習（2）

7_5_ASL認識モデル.ipynbを参照

演習のねらい
データ水増しにより分類精度が上がるケースを体験する
CNNを改良して分類精度の向上を試みる

画像データ水増し演習（2）での学習データについて

- ASL画像データは下のような画像が、画像内の手の位置が少しずつ移動したり、画像内での手の大きさが少しずつ異なる（カメラからの距離が変わる）ような画像です。1種類につき3000枚ずつあります。
- 学習データはこれらの画像から10枚ずつ抜き出したものです。つまり、ある特定の手の位置や手の大きさの10枚の画像と言うことになります。
- CNNによる分類では、対象の位置や大きさが異なれば、それは異なる画像となります。
- 画像処理によって様々な位置や大きさの手の画像を作ることで、分類精度の向上が期待できます。



ブランク

ラベル:0

A

ラベル:1

B

ラベル:2

C

ラベル:3