第 8 回　知能システム学特論レポート

15344203　有田 裕太
15344206　緒形 裕太
15344209　株丹 亮
12104125　宮本 和

西田研究室，計算力学研究室

2015 年 7 月 13 日

進捗状況

理論研究の進捗

畳込みニューラルネットワークの理論について

プログラミングの進捗

プログラム実行環境の見直し
データセット作成のための顔検出と切り出しプログラムの作成

局所コントラスト正規化

局所コントラスト正規化

画像の中から対象物を捉えるための、画像の濃淡を正規化する方法の一種

- 1 つの層だけでこの処理が可能
- 畳み込みネットに組み込むことが可能
- 層の重みは固定され, 学習の対象となるパラメータはない
- 減算正規化と除算正規化の 2 種類がある

減算正規化

減算正規化

入力画像の各画素濃淡から平均を差し引く

$$z_{ij} = x_{ij} - \bar{x}_{ij} \tag{1}$$

濃淡値の平均

$$\bar{x}_{ij} = \sum_{(p,q)\in P_{ij}} x_{i+p,j+q} \tag{2}$$

- i, j:入力画像の画素
- p, q:フィルタの画素

除算正規化

除算正規化

同じ局所領域内で画素値の分散を抑える

$$z_{ij} = \frac{x_{ij} - \bar{x}_{ij}}{\sigma_{ij}} \tag{3}$$

画素値の分散

$$\sigma_{ij}^2 = \sum_{(p,q)\in P_{ij}} w_{pq}(x_{i+p,j+q} - \bar{x}_{ij})^2 \tag{4}$$

- w_{pq}:重み

除算正規化

以上の計算をそのまま行うと，濃淡変化が少ない局所領域ほど濃淡変化が増幅され，ノイズが強調される．そこで，入力画像のコントラストが大きい部分にのみ適用するため，ある定数 c を設定し濃淡の標準偏差がこれを下回る $(\sigma_{ij} < c)$ で除算する

$$z_{ij} = \frac{x_{ij} - \bar{x}_{ij}}{max(\sigma_{ij} < c)} \tag{5}$$

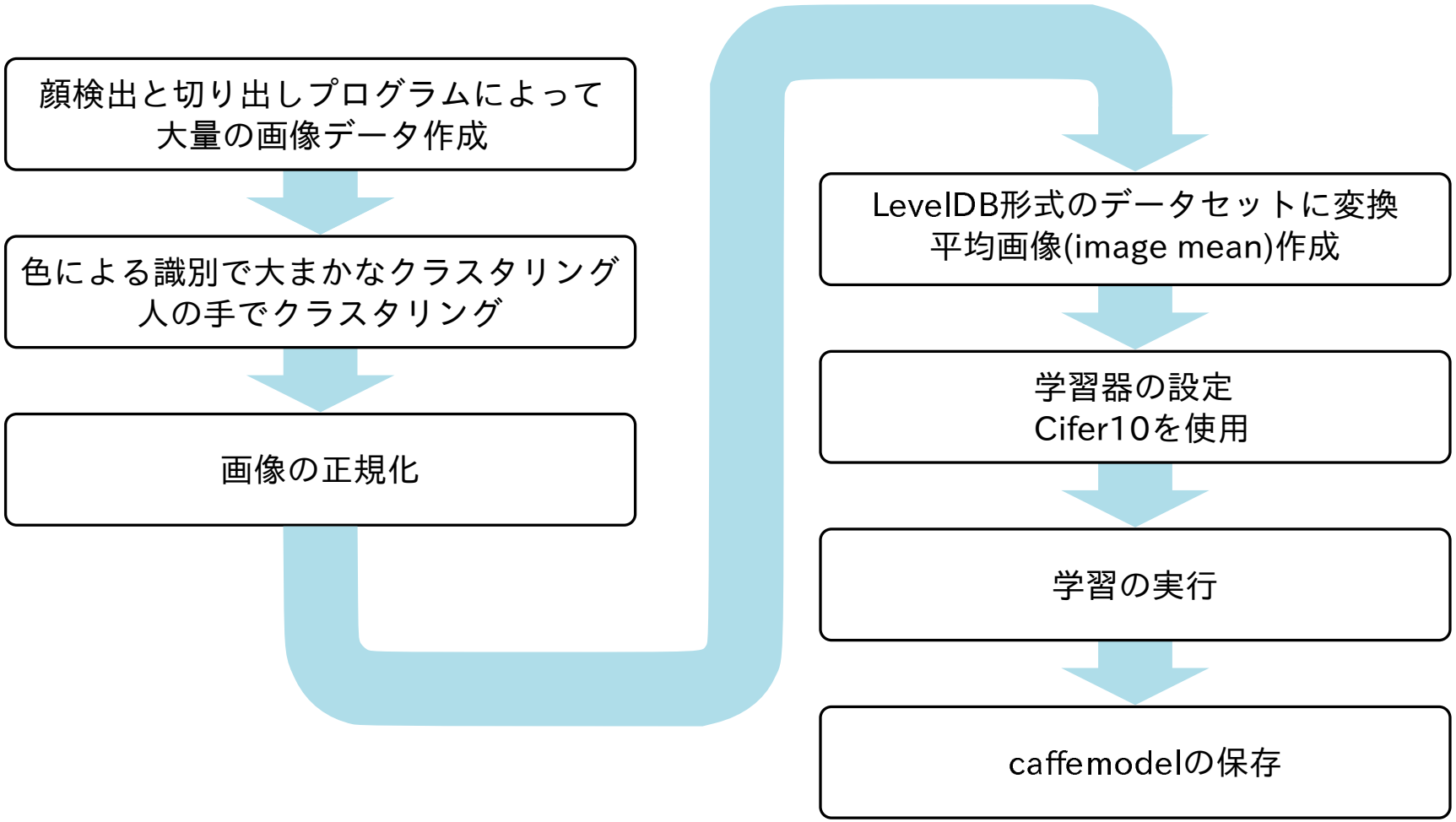
や，同様の効果が σ_{ij} に応じて連続的に変化する

$$z_{ij} = \frac{x_{ij} - \bar{x}_{ij}}{\sqrt{\sigma_{ij} + c}} \tag{6}$$

を用いる.

Caffe で学習を行うまでの流れ

独自のデータセットを用いて，Caffe で学習を行うまでの流れを下の図に示す．



顔検出と大量の画像データ

前回説明した通りなので今回は省略

色による大まかなクラスタリング

OpenCV に実装されているニューラルネットワーク（多層パーセプトロン）を用いて色特徴から大まかにクラスタリング．少量の画像を手動でラベリングして、それを学習．

```
$ ./cpp/build/train train-labeled.csv train-labeled.csv
Using training dataset
Training iterations: 908
Results on the testing dataset
 Correct classification: 496 (100%)
 Wrong classifications: 0 (0%)
    0     1    2    3    4    5    6    7    8    9
0   13    0    0    0    0    0    0    0    0    0
1   0     12   0    0    0    0    0    0    0    0
2   0     0    137  0    0    0    0    0    0    0
3   0     0    0    54   0    0    0    0    0    0
4   0     0    0    0    11   0    0    0    0    0
5   0     0    0    0    0    8    0    0    0    0
6   0     0    0    0    0    0    11   0    0    0
7   0     0    0    0    0    0    0    19   0    0
8   0     0    0    0    0    0    0    0    164  0
9   0     0    0    0    0    0    0    0    0    67
```

最後は人の目

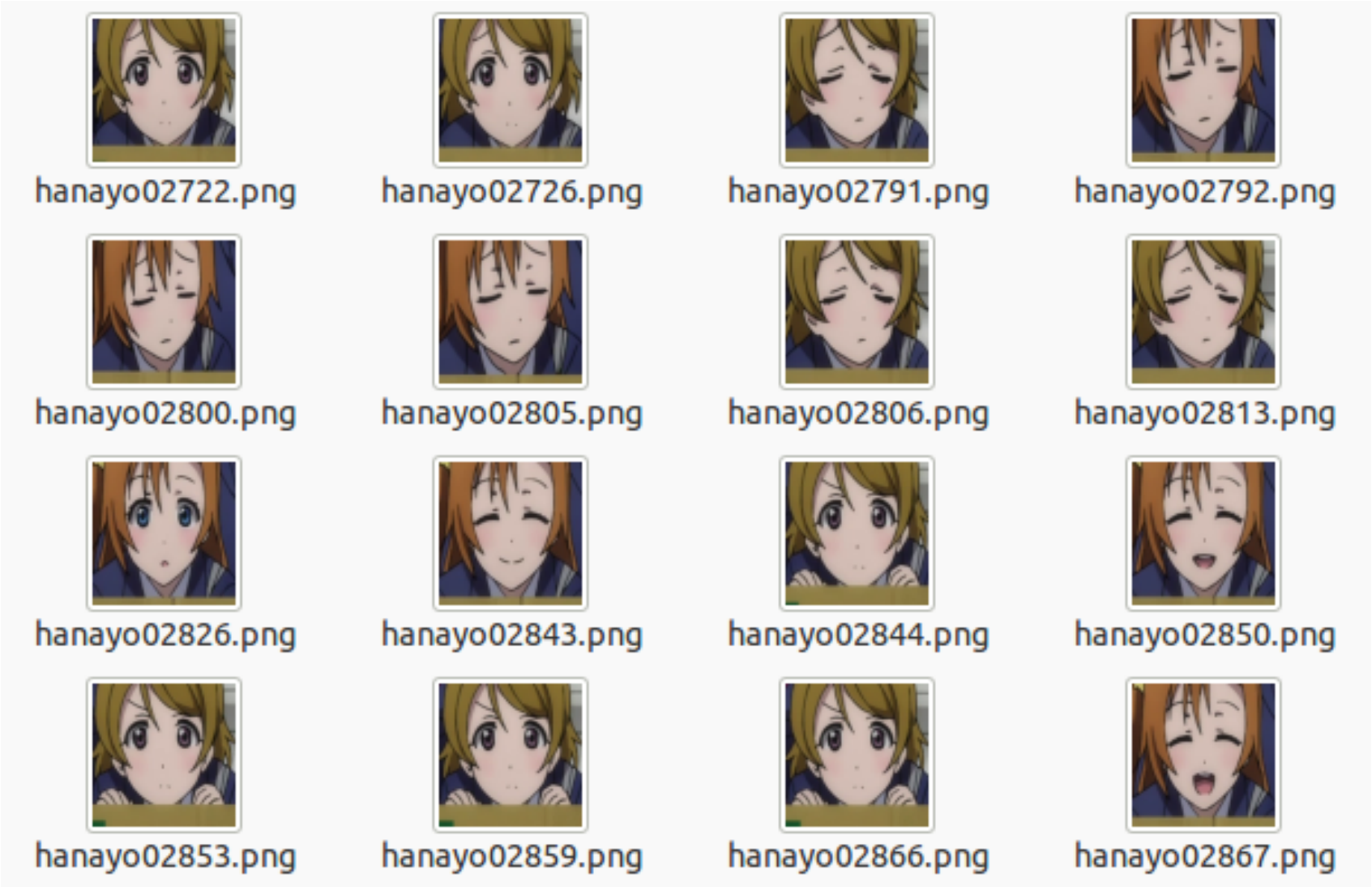


Figure: 分別結果（正解は「花陽」、「ほのか」が間違い）

LevelDB 形式のデータセットに変換

- 独自のデータセットを用意して Caffe を使って学習させるためには，LevelDB 形式と呼ばれる形式のデータセットに変換する必要がある．
- LevelDB 形式はキーバリュー型データストアであり，key と value の紐付けを高速に読み書きできる Google 製のライブラリである．

クラス分けした画像ファイルから LevelDB に

- この変換は既存のプログラム [1] を用いて変換を行った．
- これによって各画像データとラベル番号の紐付けを行う．
- 全入力画像から 9 割を訓練データ，1 割をテストデータに割り振ることができる．


SIG2D, “SIG2D’ 14 Proceedings of the 3rd Interdimensional Conference on 2D Information Processing ”, http://sig2d.org/publications/, 2014.

学習器の設定

- 今回は Cifer10 の学習モデルをそのまま使用した.
- ただし最終的な出力の数は学習にかけるラベル数に変更する必要がある.
- 変更するファイルは cifar10_quick_train_test.prototxt と cifar10_quick.prototxt である.

```
layers {
  name: "ip2"
  type: INNER_PRODUCT
  bottom: "ip1"
  top: "ip2"
  blobs_lr: 1
  blobs_lr: 2
  inner_product_param {
    num_output: 10        ここを変更
    weight_filler {
```

学習の実行

- 以下のコマンドによって学習を実行する.
- 学習は前回発表したように，GPU を用いて行った.
- GPU を用いる場合，標準設定が GPU で行うように設定してあるので，設定ファイルの記述を変更する必要はない.
- CPU のみを用いる場合，cifar10_full_solver.prototxt を最後の行にある solver_mode: GPU を CPU に変更する.

```
1 caffe train --solver examples/cifar10/cifar10_quick_solver.
    prototxt
```

学習が完了すると，cifar10_quick_iter_4000.caffemodel というファイルが生成される.

今後の課題

理論研究

CNN の詳細な調査

プログラミング

データセットの作成，学習実行結果の評価と過程の可視化