知能システム学特論レポート

（DL2 班）Caffe on Ubuntu

2015 年 7 月 9 日

1　報告者

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

2　進行状況

- 理論研究
- 畳み込みネットワークについて

3　理論研究

3.1　畳込み層

実用的な畳込みネットでは, グレースケールの画像 1 枚に対してではなく, 多チャネルの画像に対し, 複数個のフィルタを並行して畳込む演算を行う. チャネルの画像とは各画素が複数の値を持つ画像であり, チャネル数が K の画像の各画素は K 個の値を持つ. 例えば, グレースケールの画像では $K = 1$,RGB の 3 色からなるカラー画像では $K = 3$ となる. 畳込みネットの中間層では, さらにそれ以上のチャネル数の画像を扱う. 以下では, 画像の縦横の画素数が $W \times W$ でチャネル数が K のとき, 画像のサイズを $W \times W \times K$ と書く.

濃淡地を各画素に格納したグレースケールの画像を考える. 画像サイズを $W \times W$ 画素とし, 画素をインデックス $(i, j)(i = 0, \cdots, W-1, j = 0, \cdots, W-1)$ で表す. これまでインデックスは 1 から開始していたが, 画像（及びそれに類するもの）を扱う場合はインデックスを 0 から始めることとする. 画素 (i, j) の画素値を x_{ij} と書き, 負の値を含む実数値をとるとする. そして, この画像に適用する $H \times H$ 画素のフィルタ（サイズの小さい画像）を考える. このフィルタのインデックス $(p, q)(p = 0, \cdots, H-1, q = 0, \cdots, H-1)$ で表し, 画素値を h_{pq} と書く.

図 1 を用いて畳込み層での計算を説明する. この畳込み層は直前の層から K チャネルの画像 $x_{ijk}(k = 0, ..., K-1)$ を受け取り, これに $M = 3$ 種類のフィルタ $h_{pqkm}(m = 0, ..., M-1)$ を適用している. 各フィルタ $(m = 0, 1, 2)$ は通常, 入力と同じチャネル数 K を持ち（サイズを $H \times H \times K$ とする）, 図 1 のように

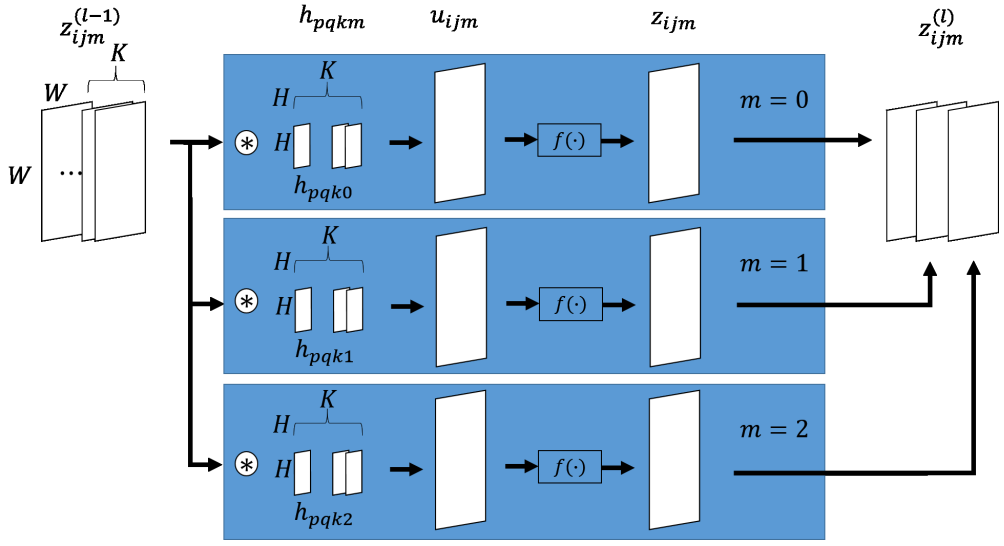

Fig.1　畳み込み層の概要

フィルタごとに計算は並行に実行される. 計算の中身は, そのフィルタの各チャネルごとに, これも並行に画像とフィルタの畳込みを行った後, 結果を画素ごとに全チャネルにわたって加算する.

$$u_{ijm} = \sum_{p=0}^{K-1} \sum_{q=0}^{H-1} \sum_{q=0}^{H-1} z_{i+p,j+q,k}^{(l-1)} h_{pqkm} + b_{ijm} \tag{3.1}$$

入力画像のチャネル数によらず,1 つのフィルタからの出力は常に 1 チャネルになる.

次にこうして得た u_{ijm} に活性化関数を適応する.

$$z_{ijm} = f(u_{ijm}) \tag{3.2}$$

図 1 のように，この値が畳み込み層の最終的な出力となり，その後の層へと伝わる．これらはフィルタ数 M と同数のチャネル数を持つ多チャネルの画像とみなせる．つまり入力サイズが $W \times W \times K$ のとき，出力サイズは $W \times W \times M$ になる．

式 (6.2),(6.3) は，層間に特別な構造の単層ネットワークとして表現できる．この層の入力と出力のユニット数はそれぞれ $W \times W \times K$ および $W \times W \times M$ であり，畳み込み計算の局所性を反映して，出力層のユニットとのみ結合する．その結合の重みがフィルタ係数 h_{pqkm} である．この重みは出力層の同一チャネルの全ユニットで共有される．これは重み共有 (weigh sharing, weight tying) と呼ばれ，このような結合の局所性と重みを共有することが畳み込み層の特徴である．

3.2　プーリング

プーリング層は通常，畳み込み層の直後に設置される．プーリング層は畳み込み層で抽出された特徴の位置感度を低下させる働きがあり，対象とする特徴量の画像内での位置が若干変化した場合においても，プーリング層の出力が不変になるようにする．

プーリング層での計算は次のようにして行う．サイズ $W \times W \times K$ の入力画像上で画素 (i, j) を中心とする $H \times H$ の正方領域とり，この中に含まれる画素の集合を P_{ij} で表す．P_{ij} 内の画素についてチャネル k ごと

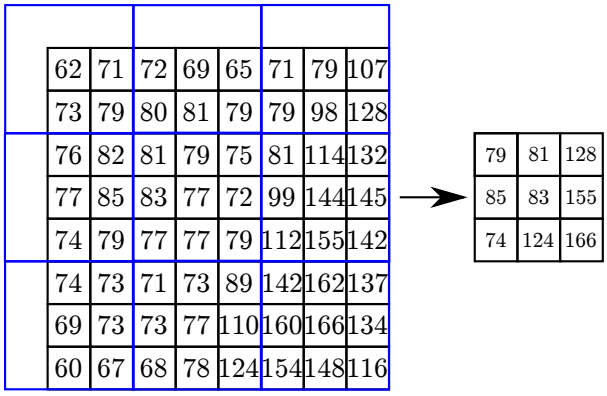

Fig.2　プーリングの例．プーリングサイズ 3×3，ストライド $s = 3$，ゼロパディングで最大プーリングを行った場合

に独立に，H^2 個ある画素値を使って 1 つの画素値 u_{ijk} を求める．これにはいくつかの方法があり，最大プーリング (max pooling) は，H^2 個の画素値の最大値を選ぶ．

$$u_{ijk} = \max_{(p,q) \in P_{ij}} z_{pqk} \tag{3.3}$$

また，平均プーリング (average pooling) はそれらの平均値を計算する．

$$u_{ijk} = \frac{1}{H^2} \sum_{(p,q) \in P_{ij}} z_{pq} \tag{3.4}$$

通常は，プーリング層の出力のチャネル数は入力画像のチャネル数と一致する．これらのプーリングを含む一般性を持った表記として，次の Lp プーリング (Lp pooling) があります．

$$u_{ijk} = \left(\frac{1}{H^2} \sum_{(p,q) \in P_{ij}} z_{pqk}^P \right)^{\frac{1}{P}} \tag{3.5}$$

$P = 1$ で平均プーリング，$P = \inf$ で最大プーリングが表現できる．なお，画像認識では最大プーリングが良く用いられている．

プーリング層では 2 以上のストライドを設定することが普通である．プーリング層では結合の重みは固定されており，学習によって変化するパラメータはない．

4　プログラミング

4.1　学習を実行する PC 環境の見直し

学習を行う PC 環境を caffe がサポートしている NVIDIA の「CUDA」及び，Deep Learning 用の CUDA ライブラリ「cuDNN」を使用する．後者のライブラリはデベロッパー登録申請（CUDA Registered Developer Program）が認可され，Ubuntu 14.04 のマシンにインストールすることができた．正確な計測は行っていないが，明らかに学習速度の向上が見られた．

4.2 データセット作成のためのプログラムの作成

独自のデータセットを作成する練習として，写真や動画から人の顔を切り出し，データセットを作成，そして学習という一連の操作を行う計画を立てた．まずデータセットを作成するためには大量の人物が一人ずつ写った写真データを用意する必要がある．したがって OpenCV に実装されている顔検出アルゴリズムを用いて，人物が写った部分を自動的に切り出すプログラムを作成した．プログラミングに用いた言語は C++，Python である．

4.2.1 画像データからの切り出し

実行は以下のようにソースとなる画像が含まれたディレクトリと，出力先のディレクトリを指定する．

```
$ python facedetect.py src output # 入力先と出力先ディレクトリ
```

Python で実装した機能を以下に示す．

- ディレクトリを指定したらそのディレクトリに入っているすべての画像ファイルを顔認識して切り取る．
- 指定したディレクトリの中に含まれているサブディレクトリの中もすべて探索してすべて取り込む．
- プログレスバーを設置して進捗を可視化

```
$ python facedetect.py src output
Input directoryname = src/
Output directoryname = output/
Exist "output/" folder.
File num =  1
66% (1 of 1) |####################          | Elapsed Time: 0:00:07 ETA:  0:04:26
```

- 壊れた画像ファイルを読み込まれた場合でも例外処理で次の画像へ．

4.2.2 動画データからの切り出し

画像とほぼ同様の機能を有しており，ソースは動画ファイルを直接指定する．

```
$ python facedetect_video.py src/test.mp4 output # 動画ファイルと出力先ディレクトリ
```

4.2.3 データセットの作成

このプログラムを用いて，アニメのキャラクターやアイドルグループのメンバーなどの写真から顔の部分のみを切り出すことが可能となる．しかし切り出したあとは自分で顔を識別してフォルダ分けする作業が必要である．

5 今後の課題

- 理論研究を進める．
- データセットの作成，学習の実行

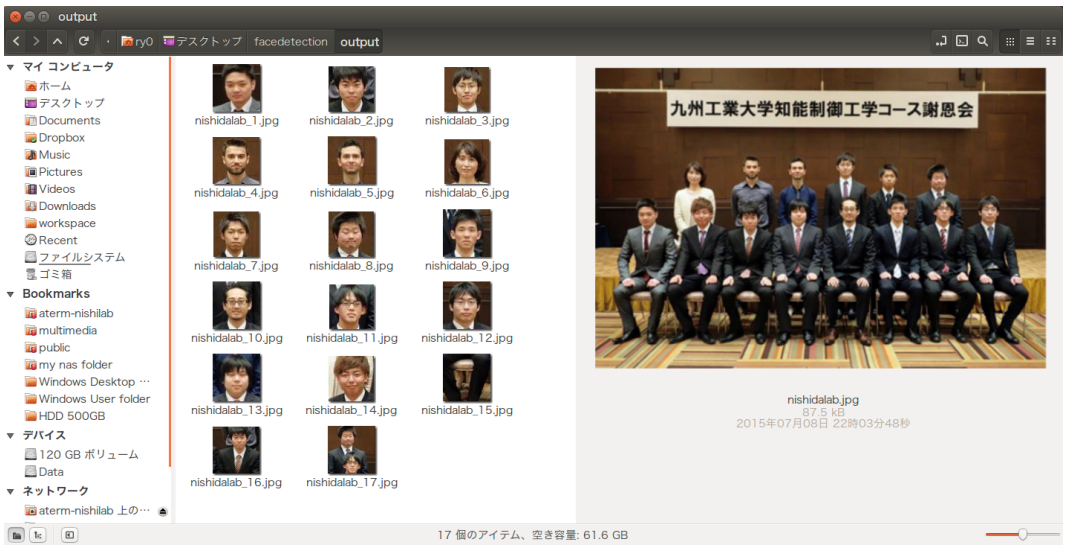

Fig.3　画像からの切り出し出力結果

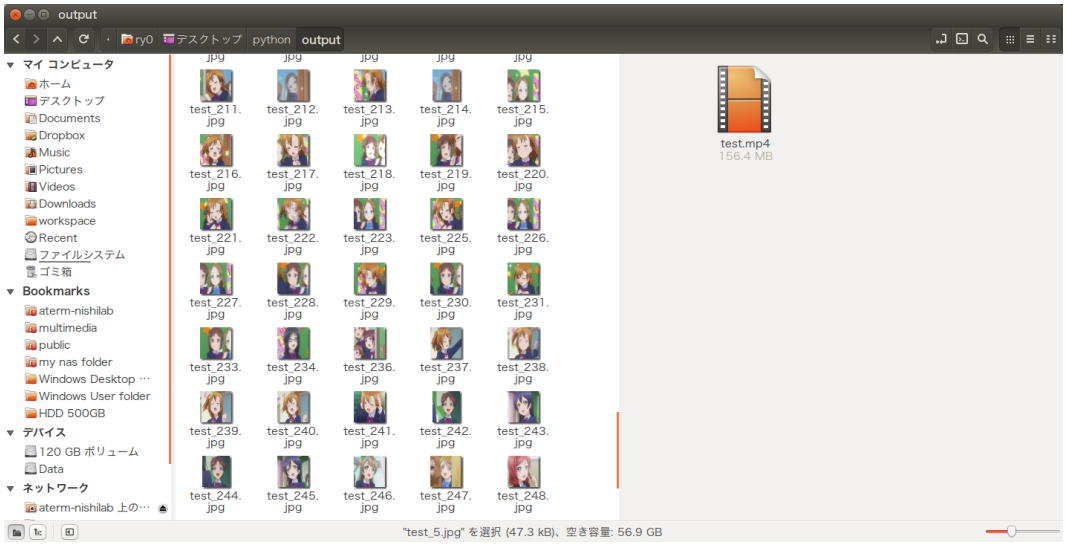

Fig.4　動画からの切り出し出力結果