Dictionary Methods for Micrograph Analysis

Students: Haoran Shu, Hang Cheung (CUHK)
Mentors: Dr. R. Archibald, Dr. A. Börisevich, Dr. E. D’Azevedo, Dr. K. Wong, Dr. J. Yin
Agenda

Problem Setting
Workflow
Algorithm
Current Outcome
Improvements
Future Work
Problem Setting

1. Every atom associated with a specific ‘mode’
2. The dictionary **exhausts** all possible modes
3. Goal: **identify** each atom in the micrograph with a specific mode in the dictionary
Problem Setting

Dictionary

Micrograph Image

- Colobus Guerezas
- Golden Lion Tamarin
- Spectacled Langur
- Whatever-kind-of monkey
- Chimp
Problem Setting

1. **Less** features to depend on
2. Extremely **similar** dictionary items
Problem Setting

Ideal Output
Workflow - Scale, Locate and Slice
Workflow - Scale, Locate and Slice

Bingo!!
Workflow

Scale

Locate
Workflow - Normalization

dark Chimp  normal Chimp
Workflow - Normalization

dark atom  normal atom

dark  bright
Workflow

Normalization
Algorithm

Scale + Locate

'Eating from Outside'

'Brute Force Search'
Algorithm

Step 1 : Eating from outside

When we do this to both the dictionary item and microscopy image, we get an approximated ‘time’
Algorithm

Step 2: Brute force search
Algorithm

Step 2: Brute force search
Normalization

In grayscale images, ‘colors’ are represented by ‘intensity’, from 0 (white) to 255 (black), thus each grayscale image has an intensity distribution.
The differences in contrast and brightness are actually differences in the intensity distribution!
Since we are only concerned about the subtle differences in the circled area, we use a mask to shade the cores and will NOT take them into consideration for final comparison.
Algorithm

Comparison

How to **summarize** the concerned images with effective **features**?

**HOG** (Histogram of Oriented Gradients) comes in.
HOG:

1. Divide an image into smaller patches
2. Calculate the gradients at each pixel
3. Generate a feature vector of gradients distribution for each small patch
Algorithm

Comparison

Use the cross correlation of HOG feature vectors to represent the similarity between two images.

Cross Correlation

\[
 r_{xy} = \frac{\sum_{i=1}^{n} (x(i) - \bar{x})(y(i) - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x(i) - \bar{x})^2 \sum_{i=1}^{n} (y(i) - \bar{y})^2}}
\]

PROBLEM: ‘gap’ too small between ‘similar’ and ‘dissimilar’!
Items are so similar that the cross correlations nearly all above 0.75. Items from different modes can even have cross correlation as high as 0.88.
Items are so similar that the cross correlation nearly all above 0.75. Items from different modes can even have cross correlation as high as 0.88.

**cc:**
- Same mode: 0.92
- Different modes: 0.88
Items are **so similar** that the cross correlation nearly all **above 0.75**. Items from different modes can even have cross correlation as high as **0.88**.

<table>
<thead>
<tr>
<th>Dictionary</th>
<th>mode 1</th>
<th>mode 2</th>
<th>mode 3</th>
<th>mode 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

**distance:**
- Same mode: 0.08
- Different modes: 0.12
Algorithm

Comparison

Items are so similar that the cross correlation nearly all above 0.75. Items from different modes can even have cross correlation as high as 0.88.

\[
1 - 0.92 \times 0.92 = 0.154 \\
1 - 0.88 \times 0.88 = 0.226
\]
Algorithm

Comparison

Dictionary

<table>
<thead>
<tr>
<th></th>
<th>mode 1</th>
<th>mode 2</th>
<th>mode 3</th>
<th>mode 4</th>
</tr>
</thead>
</table>

Items are **so similar** that the cross correlation nearly all **above 0.75**. Items from **different** modes can even have cross correlation as high as **0.88**.

\[(1 - 0.92)^2 = 0.0064\]

\[(1 - 0.88)^2 = 0.0144\]
Algorithm

Comparison

\[ 10000 \times (1 - cc(A, A'))(1 - cc(B, B'))(1 - cc(C, C'))(1 - cc(D, D'))(1 - cc(A, A')cc(B, B')cc(C, C')cc(D, D')) \]
Algorithm

Assume $cc = 0.92$ and $0.88$ for all parts.

Using $cc$ only once we get distances: $0.08$ and $0.12$
Using our formula we get distances: $0.12$ and $0.83$
Current Outcome

The numbers at the corner of each unit stands for the dictionary item it is matched to.

On average our algorithm takes 110s to run on a MacBook Air (1.8 GHz Intel Core i5, 4 GB 1600 MHz DDR3) for a dictionary of size 25 and a microscopy of size 30 (items). A more detailed complexity analysis will be included in our final
Divide the dictionary into subgroups using k-means, and label each atom with a signature of the group; then just pick the best group match first.
Improvements

Original Dictionary

Categorized Dictionary
(Partial)

Group 1

Group 2

Group 3

k-means + sampling
Future Work

- Accuracy: Machine Learning

- Compatibility: C/C++ platform

- Performance: LSH (Locality-Sensitive Hashing)
Q & A