Learning approach for multicontent analysis of compound images

Q. Besnehard, C. Marchessoux & T. Kimpe

June 2\textsuperscript{nd}, 2009
Barco’s research activities

- Barco has a specialized team (Technology Innovation Group) that
  - Looks for funding opportunities (ITEA, IBBT, IWT)
  - Writes and defends project applications
  - Follows up the funding projects

- Every year we are active in several projects.

- Barco’s typical way of working:
  - We never start projects purely for the funding.
  - As much as possible we try to collaborate with partners. This means that we jointly work around a specific topic, with every partner of course focusing on their domains of interest.
  - If relevant and useful, then we can also involve universities to do research for us.
Current projects

- **ITEA2:**
  - **CANTATA: Multi Content Analysis for Networked Medical Applications**
    - 2006-2009, 27 partners
    - Industries: Barco, Philips healthcare, VTT, Telefonica...
    - Research labs: Kingston Univ., ...

- **IBBT/IWT**
  - **MEVIC: MEdical Virtual Imaging Chain**
    - 2009-2011: 7 partners
    - Industries: Barco, Philips Healthcare, Hologic,
    - Research labs: Ghent, Brussels, UZLeuven

- **IBBT:**
  - **TIRPA: Tomographic Image Reconstruction, Processing, and Analysis**
    - 2008-2009, 9 partners
    - Industries: Barco, Skyscan...
    - Research labs: Ghent, Visionlab, ...
CANTATA – Multi Content Analysis for Networked Medical Applications

- Develop algorithms to automatically classify the content of a screenshot:
  - Without any a priori knowledge on the content
  - Color image, RGB 24 bits
  - The algorithm implemented in hardware devices should run in real time

Example
Multi Content Analysis

- Content classification:
  - Gui content
  - Text content
  - Uniform content (one color)
  - Medical image content
    - Digitized Screen Film Conventional Radiography (Chest, Mammography and General Radiography), Computed Radiography, Digital Radiography, Computed Tomography (CT), MRI, Angiography, ultrasonography, Positron Emission Tomography (PET), and multi modalities fusion.
  - Other complex content
**Potential application**

Tiles classification to optimize the compression strategy

- being able to adapt the encoding depending on the content of the thumbnail, saving network bandwidth without altering important content
Choice of MCA methodology: constraints

- Image decomposed in tiles / thumbnails:

- Fast classification:
  - Software implementation: 1 frame every 2 seconds
  - Hardware (FPGA) implementation: 5 frames / second

- Increased performance by re-learning:
  - Improve accuracy by identifying cases inducing classification errors
  - Learning approach
MCA: Thumbnail classifier

Classifier

- Medical image
- Other complex
- GUI
- Text
- Uniform
- Not Classified

Not Classified
Obtainment of a classifier

- Framework

Diagram:
- Dataset
  - Feature selection
  - Ground Truth / Annotation
  - MCA
  - Evaluation
  - Presentation
Dataset

- **Training dataset:**
  - Medical: 25,000 thumbnails
  - Text: 26,000 thumbnails
  - GUI: 18,000 thumbnails
  - Other complex: 18,000 thumbnails

- **Re-injection dataset:**
  - 75 screenshots
  - Resolution: 1600x1200

- **Testing dataset:**
  - Medical: 2,000 thumbnails
  - Text: 2,000 thumbnails
  - GUI: 2,000 thumbnails
  - Other complex: 400 thumbnails

- **Annotation**
  - **Viper toolkit:**
    http://viper-toolkit.sourceforge.net
## Feature list

- List of relevant features to compute from the thumbnails

<table>
<thead>
<tr>
<th>Feature</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of colors</td>
<td>1</td>
</tr>
<tr>
<td>Number of gray levels</td>
<td>1</td>
</tr>
<tr>
<td>Standard deviation of the hue component</td>
<td>1</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>4</td>
</tr>
<tr>
<td>Derivatives gradient counts</td>
<td>16</td>
</tr>
<tr>
<td>Sobel filters gradient counts</td>
<td>16</td>
</tr>
<tr>
<td>Laplace operator gradient counts</td>
<td>8</td>
</tr>
<tr>
<td>Kirsh operators gradient counts</td>
<td>32</td>
</tr>
<tr>
<td>Absolute DCT coefficient sums</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total number</strong></td>
<td><strong>81</strong></td>
</tr>
</tbody>
</table>
Feature analysis

- Elimination of redundant features thanks to a dimensionality reduction technique (Principal Feature Analysis, Cohen et al. 2002)
- Less features → faster calculation / classification

<table>
<thead>
<tr>
<th>Feature</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of colors</td>
<td>1</td>
</tr>
<tr>
<td>Number of gray levels</td>
<td>1</td>
</tr>
<tr>
<td>Standard deviation of the hue component</td>
<td>1</td>
</tr>
<tr>
<td>Standard deviation on color channels</td>
<td>3</td>
</tr>
<tr>
<td>X Derivative gradient counts on color channels</td>
<td>6</td>
</tr>
<tr>
<td>Sobel filters gradient counts on color channels</td>
<td>12</td>
</tr>
<tr>
<td>Laplace operator gradient counts on color channels</td>
<td>6</td>
</tr>
<tr>
<td>NW Kirsh operator gradient counts on color channels</td>
<td>6</td>
</tr>
<tr>
<td>Absolute DCT coefficient sums</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total number</strong></td>
<td><strong>36</strong></td>
</tr>
</tbody>
</table>
Classification: simple classifiers trained by the AdaBoost learning algorithm

- **Decision trees**
  - Combines features \((x)\) and thresholds \((t)\) to achieve classification
  - Do not need high accuracy to be used in AdaBoost → small trees, fast classification

- **AdaBoost**
  - Learning statistic tool
  - Trains and combines many simple classifiers (decision trees, classification error rate up to 50%)
  - Learning approach: iterative self improvement of the classification by learning from its mistakes
  - Results in an accurate and computationally fast classifier for one category of image
Training

Multiple different trees trained through Iterative process

Preventing overfitting:
- Split the training dataset in two parts
- Train on the first half
- Detect emergence of overfitting on the second half
- Determine best \{number of iterations / max number of nodes\} combination
- Text: 30 iterations / 63 nodes
- Other categories: 30 iterations / 15 nodes
Testing

\[
\text{Text (}\mathbf{r}_1 = 1) \quad \text{Text (}\mathbf{r}_2 = 1) \quad \text{Non-text (}\mathbf{r}_3 = -1) \quad \text{Text (}\mathbf{r}_4 = 1)
\]

\[
x_{1:t1} \quad x_{2:t2} \quad x_{3:t3} \quad x_{4:t4} \quad x_{5:t5} \quad x_{6:t6} \quad x_{7:t7} \quad x_{8:t8}
\]

\[
x \cdot w_1 \quad x \cdot w_2 \quad x \cdot w_3 \quad x \cdot w_4
\]

\[
\text{VOTE} \quad \text{sign} \left( \sum_{i=1}^{\mathbf{n}} r_i w_i \right)
\]

\[
\text{Text}
\]
Cascade of classifiers

- Essential for recognition of the most important categories
  - Uniform (one-color) image classification is first because it’s fast and 100% accurate.
  - Thumbnails are tested for Medical and Text classification before GUI and Other complex
  - Thumbnails pass through as few classifiers as possible
Re-injection

- Method for the detection of misclassified cases in the training dataset

- Classification of 75 screenshots (1600x1200)

- Extraction of misclassified thumbnails
  - Text: 2500 thumbnails
  - GUI: 1000 thumbnails
  - Other complex: 1500 thumbnails

- Addition of the misclassified thumbnails in the training dataset
  - Obtainment of an enhanced training dataset

- Important: at no moment the Testing dataset was used to enhance the Training dataset.
## Raw results

<table>
<thead>
<tr>
<th></th>
<th>True Positives</th>
<th>True Negatives</th>
<th>False Positives</th>
<th>False Negatives</th>
<th>Not Classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical</td>
<td>76.4%</td>
<td>99%</td>
<td>1%</td>
<td>23.6%</td>
<td>10.6%</td>
</tr>
<tr>
<td>Text</td>
<td>52.6%</td>
<td>95.6%</td>
<td>4.4%</td>
<td>48.4%</td>
<td>19.3%</td>
</tr>
<tr>
<td>GUI</td>
<td>52.8%</td>
<td>84.9%</td>
<td>15.1%</td>
<td>47.2%</td>
<td>26.1%</td>
</tr>
<tr>
<td>Other complex</td>
<td>81.6%</td>
<td>92.2%</td>
<td>7.8%</td>
<td>18.4%</td>
<td>11.2%</td>
</tr>
</tbody>
</table>
## Results after re-injection

<table>
<thead>
<tr>
<th></th>
<th>True Positives</th>
<th>True Negatives</th>
<th>False Positives</th>
<th>False Negatives</th>
<th>Not classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical</td>
<td>77.4%</td>
<td>99.3%</td>
<td>0.7%</td>
<td>22.6%</td>
<td>9.5%</td>
</tr>
<tr>
<td>Text</td>
<td>80.1%</td>
<td>94.4%</td>
<td>5.6%</td>
<td>19.9%</td>
<td>9.1%</td>
</tr>
<tr>
<td>GUI</td>
<td>55.6%</td>
<td>93.2%</td>
<td>6.8%</td>
<td>44.4%</td>
<td>30.1%</td>
</tr>
<tr>
<td>Other complex</td>
<td>78.5%</td>
<td>95.6%</td>
<td>4.4%</td>
<td>21.5%</td>
<td>15.4%</td>
</tr>
</tbody>
</table>

- Low FP rate

### Speed (software C++)
- 1.5 seconds per frame on a 2.5GHz computer
- Feature calculation is the most time-consuming step
- Classification itself takes 100ms
Results

used in our feature list, as we do not know which one will be better than Text versus Graphics. The calculation in [Kal 01] is 1 calculation on 32x32 thumbnails; they are decomposed into DCT coefficients on each block and then sum the 16 to the base pool.

Figure 2.1: a) Coefficients used in the first DCT coefficient on the second DCT coefficient sum in green.

Figure 2.1: c) Coefficients used in the first DCT coefficient on the second DCT coefficient sum in green.
Results

Figure 2.1: Coefficients used in the first DCT coefficient second DCT coefficient sum (in green).
Results

used in our feature list, as we do not know which one would
then Test versus Graphics. The calculation in [Kol 01] is a
calculation on 32x32 thumbnails, they are decomposed in 16
of DCT coefficients on each block and then sum the 16 to
the base point.

![Figure 2.1: Coefficients used in the first DCT coefficient
second DCT coefficient sum in green.](image)

```python
def open_image(path, image_filename):
    # Open the image located at the given path
    image_filename = Temp.openimage(image_filename)
    image_filename = image_filename

    # Creates a mapping of the coordinates of the annotation rectangles with the associated category
    self.original_boxes = {}
    self.original_boxes = {}  # Assuming the code snippet is incomplete and should be filled in
```

page 22
Conclusion

- Results not as good as expected but encouraging enough to further develop the framework

- Better classification accuracy could be achieved through:
  - Addition of new categories
  - Addition of new features
  - Other ways to combine decision trees
Aknowledgements

- CANTATA project
  - ITEA program

- IWT

- Dr Gert Van Hoey

- Dr Olivier Alata, SIC laboratory, University of Poitiers (France)

- University Institute GPhy, University of Poitiers (France):
  - Director: Pr Patrick Girard
  - Students: Guillaume Spalla, Arnaud Joubel, François Boudet
Thank you for your attention

- Questions ? Remarks ?

Quentin.Besnehard@barco.com