Attributes in Class

CLASS
Cognitive-Level Annotation using Latent Statistical Structure

Vittorio Ferrari (ETH Zurich)
CogSys 2010
What are attributes?

(visual) properties of objects beside their category label

red

- berry
- car

furry

- cat
- polar bear

striped

- ball
- socks

has paws

- raccoon
- bird
What are attributes good for?

Transfer knowledge between categories (reduce training data)

More specific search queries (e.g. yellow car)
Part I

Learning color names

J. Van de Weijer, C. Schmid, J. Verbeek
CVPR 2007
learning color names

Goal: learn appearance model for basic color names

The English language has 11 basic color names.
learning color names

Use Google images to collect a set of weakly labeled images. E.g. query ‘blue+color’

black  blue  brown  green  orange  purple  white

false positives

Images retrieved by Google images
learning color names

Labeled input images:

yellow  yellow  red

LAB histogram for whole image

bins = words w

PLSA-bg

learn $P(w|z)$, and $P(z|d)$ by EM

Each d constrained to bg + 1 col

Color name distribution $P(w|z)$

yellow  ...  red

d = images
z = col names
Results flower data set

test color names for image classification on a flower data set of 1360 images over 17 classes (Nilsback and Zisserman CVPR 2008)

<table>
<thead>
<tr>
<th>dataset</th>
<th>flower</th>
<th>color</th>
<th>shape</th>
<th>color &amp; shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>color</td>
<td>shape</td>
<td>color &amp; shape</td>
<td></td>
</tr>
<tr>
<td>HSV-SIFT</td>
<td>-</td>
<td>-</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>hue</td>
<td>40</td>
<td>65</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>opponent</td>
<td>39</td>
<td>65</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>color names</td>
<td>57</td>
<td>65</td>
<td>81</td>
<td></td>
</tr>
</tbody>
</table>

Color names better than photometric invariant color representations:
+ distinguish grey/white/black
+ semantically group different shades of green (and red, …)
Part II

Learning patterns of segments

V. Ferrari and A. Zisserman
NIPS 2007
Which attributes?

Visual properties of objects:

- Red
- Black/white stripes
- Round
- Generic stripe patterns

One or two elements
Appearance, shape, layout
Specific appearance, generic patterns
Objective

- is attribute present?
- spatial extent (pixelwise segmentation)
From text to visual models

Weakly supervised learning

Positive training images (unsegmented)

Negative training images

Noisy labels!
Image units = segments

**segment** =
appearance + shape (= geometric properties)

**pair** =
appearances + shapes + relative properties
Binary attribute model

- **Model Parameter**:\(\gamma\), \(\lambda_i\), \(\lambda_c\)
- **Latent**: \(\alpha\), \(\delta\), \(\beta\)
- **Foreground Appearance for This Image**: \(\gamma\), \(\lambda_i\), \(\lambda_c\)
- **Allowed Appearances**: \(\alpha\), \(\delta\), \(\beta\)
- **Adjacency**: \(\alpha\), \(\delta\), \(\beta\)
- **Background Probability**: \(\alpha\), \(\delta\), \(\beta\)
- **A Segment**: \(=\) appearance + shape
- **A Pair of Segments**: \(=\) two segments + rel geom properties
- **Foreground Label**: \(\alpha\), \(\delta\), \(\beta\)

Definitions:
- \(D\): # images
- \(S\): # segments
- \(C\): # pairs
- \(G, R\): # properties

- **Relative Geometry**: for this image

- **Geometric Property**: \(=\) fg distribution + activation state
Binary attribute model

Model components $\mathcal{M}$:
- appearance ($\alpha$)
- shape ($\lambda^j_1$, $\lambda^j_2$)
- layout ($\gamma^k$, $\delta$)
- background probability ($\beta$)

Likelihood of a pair:

$$p(c|\mathcal{M}, a) = p(s_{1,a}, s_{2,a}|a) \cdot \prod_j \left( p(s_{1,g}^j|\Phi_1^j)^{v_1^j} \cdot p(s_{2,g}^j|\Phi_2^j)^{v_2^j} \right) \cdot \prod_k \left( p(c^r_k|\Psi^k)^{v^k} \right) \cdot p(c|\delta)$$

Likelihood of a segment:

$$p(s|\mathcal{M}; f, a) = \begin{cases} \max_{\{c|s \in c\}} p(c|\mathcal{M}, t) & \text{if } f = 1 \\ \beta & \text{if } f = 0 \end{cases}$$
An example binary model: stripes

- general appearance (\(\alpha = \) all pairs of app in codebook)
- adjacent segments (\(\delta = \) true)
From segments to images and pixels

Probability of a pixel is probability of its segment and is independent across segments

Image likelihood = product over segments, counted by their areas $N_s$:

$$p(I|M; F, a) = \prod_{s \in I} p(s|M; f, a)^{N_s}$$
Inferring the latent variables

Given model $\mathcal{M}$, compute approximate image likelihood (max vs sum):

$$p(I|\mathcal{M}) = \max_{a \in \alpha} \max_{F} p(I|\mathcal{M}; F, a)$$

determines:

- $F$ (foreground segmentation)
- $a$ (attribute appearance for the image)

+ only about 0.2 seconds per image
Learning the model parameters

Weakly supervised setting:

Positive training images not segmented, and unreliably labeled

Approach:

- use negative training images

- train model discriminatively, by maximizing likelihood-ratio

\[
\frac{p(I_+ | \mathcal{M})}{p(I_- | \mathcal{M})} = \frac{\prod_{I^i_+ \in I_+} p(I^i_+ | \mathcal{M})}{\prod_{I^i_- \in I_-} p(I^i_- | \mathcal{M})}
\]

- circular dependencies between $\mathbf{F}$, geometry, and $\beta$

  -> iterative algorithm (see NIPS 2007 paper)
Results for learning

- Unary: 4 colors
  - pos train set: top 14 hits google-images
  - noise: ±30%
  - neg train set: all pos images for the other colors

- Binary: stripes / dots / b/w checkerboard
  - pos train set: 22-74 google/yahoo
  - noise: ±30%
  - neg train set: pos images for colors
Color models

- red
- green
- blue
- yellow

- unary
- specific appearance
- no active geometry
Stripe and dot models

- binary
- general appearance ($\alpha = \text{all pairs of app in codebook}$)
- adjacent segments ($\delta = \text{true}$)
Results for stripes
Results for dots
Part III

Using attributes for recognizing unseen classes

C. Lampert, H. Nickisch, S. Harmeling
CVPR 2009
Attribute Based Classification

Which of these images shows an axolotl?
Attribute Based Classification

Description: **Axolotls**

- live in **water**,
- are **white**,
- have no **long fur**.

Which of these images shows an **axolotl**?

We can classify objects based on a **description**.
Even without any training image
Attribute Based Classification

Description: **Axolotls**
- have property $X$,
- have property $Y$,
- don’t have property $Z$.

Which of these images shows an **axolotl**?

To classify objects based on a **description**, if we need to **understand** the description.
## Attribute Based Classification

<table>
<thead>
<tr>
<th>class</th>
<th>description</th>
<th>training images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>water</td>
<td>white</td>
</tr>
<tr>
<td>Axolotls</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>Doves</td>
<td>¬X</td>
<td>Y</td>
</tr>
<tr>
<td>Persian cats</td>
<td>¬X</td>
<td>Y</td>
</tr>
<tr>
<td>Polar bears</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>Walruses</td>
<td>X</td>
<td>¬Y</td>
</tr>
</tbody>
</table>

- we can **learn the properties** from classes with examples
- we can detect properties also in images of other objects classes
- Using the descriptions, we can **transfer visual information** between object categories.
Flat Multiclass Classification

- images \( x \)
- class labels \( y_1, \ldots, y_K \in \mathcal{Y} \) (at training time)
- class labels \( z_1, \ldots, z_L \in \mathcal{Z} \) (at test time)
- classifiers \( \alpha \) (to be learned at training time)
Classification with Attribute Layer

- images $x$,
- class labels $y_1, \ldots, y_K \in \mathcal{Y}$ (at training time)
- class labels $z_1, \ldots, z_L \in \mathcal{Z}$ (at test time)
- attributes $a_1, \ldots, a_M \in \{0,1\}^M$ (encode description)
- attribute classifiers $\beta$ (to be learned)
Animals with Attributes Dataset

- 85 binary attributes for 50 animals categories
  - collected from human subjects
Animals with Attributes Dataset

- >30,000 images for the 50 animal categories
  - collected from the internet,
  - cleaned out and annotated manually.

- extract global feature vectors (Color Histos, SIFT, PHOG, ...)
- train probabilistic output SVMs $\beta$, each predicting the presence of an attribute $a_m$ in an image $x$: $p(A_m = a_m | X = x)$
Task: calculate \( \arg\max_z p(Z = z \mid X = x) \).

We don’t have training data to learn this directly.

Introduce attribute \( a = (a_1, \ldots, a_M) \) layer:

\[
p(Z = z \mid X = x) = \sum_{a \in \{0,1\}^M} p(Z = z \mid A = a)p(A = a \mid X = x)
\]

\[
= \frac{p(Z = z)}{\prod p(A_m = a^z_m \mid X = x)} \prod_{m=1}^M p(A_m = a^z_m \mid X = x)
\]

where \( a^z \) is the attribute description of class \( z \).
Direct Attribute Prediction

- Per-class attribute vectors
  - \( a^y = (a^y_1, \ldots, a^y_L) \) for \( y \in \mathcal{Y} \)
  - \( a^z = (a^z_1, \ldots, a^z_K) \) for \( z \in \mathcal{Z} \)
- Training examples: \((x_1, y_1), \ldots, (x_n, y_n)\)

\[
p(Z = z | X = x) = \frac{p(Z = z)}{\prod p(A_m = a^z_m)} \prod_{m=1}^{M} p(A_m = a^z_m | X = x)
\]

- \( p(Z = z) \) constant,
- \( p(A_m = a_m) \) estimate from training data, or constant,
- \( p(A_m = a_m | X = x) \) learn from \((x_1, a^y_1), \ldots, (x_n, a^y_n)\).
  - SVM with \( \chi^2 \)-kernel, Platt scaling
Results: *Animals with Attributes* dataset

- **training**: 40 animal categories with training images
- **evaluation**: predict 10 disjoint animals categories
- **each class** has a description by 85 binary attributes

**Conclusion:**
attributes allow to transfer information between classes and to recognize classes with *no training image* better than chance
Results: *Animals with Attributes* dataset
- training: 40 animal categories with training images
- evaluation: predict 10 disjoint animals categories
- each class has a description by 85 binary attributes

Conclusion:
attributes allow to transfer information between classes and to recognize classes with *no training image* better than chance
Human body pose as an attribute

All tracks

Most confident person

No temporal integration

www.robots.ox.ac.uk/~vgg  www.vision.ee.ethz.ch/~calvin

V. Ferrari, M. Eichner, M. Marin, A. Zisserman