Learning Visual Distance Function for Identification from one Example.

Eric Nowak and Frederic Jurie
Bertin Technologies / CNRS
LEAR Group – INRIA - France
This is an object you've never seen before …
… can you recognize it in the following images?
This is an object you've never seen before … … can you recognize it in the following images?

**Identification from One Example.**

- “Obviously” different
- Same pose and shape, but different object
- Different pose and light, but same object
This is an object you've never seen before … … can you recognize it in the following images?

Class A

Class B

Not Possible!
This is an object you've **never seen** before … … can you recognize it in the following images?

\[ S(), S(), S() \]
This is an object you've *never seen* before … … can you recognize it in the following images?

Knowledge about categories

Different

Same
Our goal: Learning from one Example with Equivalence Constraints.

- We want to learn a similarity measure on a generic category (e.g. cars)
- Given a training set of image pairs labelled «same» or «different»: equivalence constraints
- we can predict how similar two never seen images are
- despite occlusions, clutter and modifications in pose, light, ...
How to compare images?

Not adapted to visual classes
How to learn the distance?

$S = X^t A X$

Not robust to occlusions, background
How to be robust to occlusion, view point changes?

Robust combination” of local distances:

\[ S = f(d_1, d_2, \ldots, d_n) \]
Computation of corresponding patches

- P0 in I0: sampled **randomly** (quadratic in size, uniform in position)

- P1 in I1: the **best ZNCC match** of P0 around P0. Search region: extension of P0 in all directions.

- A pair of images is **simplified** into the np patch pairs sampled from it.
From multiple local similarities to one global similarity

Likelihood->Similarity
[Ferencz et al. Iccv 05]
Patch independence: a bad assumption

Space of patch pairs differences

=>$\rightarrow$ Vector quantization
Vector quantization of pair difference

\[
x = [1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1]
\]
Computation of the trees

Tree creation (EXTRA-Trees [Geurts et al. ML06, Moosman et al. NIPS06]):

- create a root node with positive and negative patch pairs.
- recursively split the nodes until they contain only pos or neg pairs:
  - create $n_{condtrial}$ random split conditions:
    - simple parametric tests on pixel intensity, gradient, geometry, etc.
      random $\leftrightarrow$ parameters drawned randomly
    - select the one with the highest information gain
  - split the node into two sub-nodes

<table>
<thead>
<tr>
<th>Node</th>
<th>#pos pairs</th>
<th>#neg pairs</th>
<th>Entropy</th>
<th>Splitcondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>10000+</td>
<td>10000-</td>
<td>$H = 0.693147$</td>
<td>$\text{Abs} \left( \Delta(P(2,3)) \right) \leq 0.72144$</td>
</tr>
<tr>
<td>n2</td>
<td>174+</td>
<td>317-</td>
<td>$H = 0.650115$</td>
<td>$\text{Abs} \left( \Delta(P(9,8)) \right) \leq 0.48665$</td>
</tr>
<tr>
<td>n4</td>
<td>19+</td>
<td>60-</td>
<td>$H = 0.551663$</td>
<td>Leaf</td>
</tr>
<tr>
<td>n5</td>
<td>155+</td>
<td>257-</td>
<td>$H = 0.66218$</td>
<td>Leaf</td>
</tr>
<tr>
<td>n3</td>
<td>9826+</td>
<td>9683-</td>
<td>$H = 0.69312$</td>
<td>$\text{Abs} \left( \Delta(P(2,2)) \right) \leq 0.31706$</td>
</tr>
<tr>
<td>n6</td>
<td>1845+</td>
<td>3040-</td>
<td>$H = 0.66292$</td>
<td>Leaf</td>
</tr>
<tr>
<td>n7</td>
<td>7981+</td>
<td>6643-</td>
<td>$H = 0.688956$</td>
<td>Leaf</td>
</tr>
</tbody>
</table>

Very Fast!
Computation of the trees

The positive patches of three different nodes during tree construction

("faces in the news" dataset)
From clusters to Similarity

The similarity measure is a linear combination of the cluster membership

\[ \text{Sim}(\cdot, \cdot) = \omega^T x \]

- and we want: the larger the more similar

- We define the weight vector as the normal of the linear SVM hyperplane separating the descriptors of positive and negative learn set image pairs.
Similarity measure

- Given 2 images ...
- Detect corresponding patch pairs.
- Affect them to clusters with extremely randomized trees.
- The similarity measure is a linear combination of the cluster membership.

\[ x = [1, 0, 1, 1, 0, 0, 1, 1] \]

\[ \text{Sim} (\text{image}_1, \text{image}_2) = \omega^T x \]
Conclusions

• Similarity of **never seen objects**, given a set of similar and different training object pairs of the same category.
• Original method consisting in
  – (a) finding similar patches
  – (b) clustering the set of patch pair differences with an ensemble of extremely randomized trees
  – (c) combining the cluster memberships of the pairs of local regions to make a global decision about the two images.
• Can learn complex visual concepts.
• Image polysemy->of pairs of “same” and “different” defines visual concepts
• Can automatically selects and combines most appropriate feature types
• Future works: recognize similar **object categories from a training set of equivalence constraints.**