INTRODUCTION

STATIONARY FEATURES AND FOLDED HIERARCHIES FOR EFFICIENT CAT DETECTION

DONALD GEMAN

JOINT WORK WITH FRANCOIS FLEURET

INTRODUCTION (CONT.)

HIERARCHY OF CLASSIFIERS

Advantages
- Highly efficient scene parsing;
- Organized focusing on hard examples.

Disadvantages
- Many classifiers to learn;
- Inefficient learning (unless training examples can be synthesized).
**INTRODUCTION**

**RELATED WORK**

- **Part-based Models**
  - Constellation (M. Weber, M. Welling, P. Perona),
  - Composite (D. J. Crandall, D. P. Huttenlocher),
  - Implicit Shape (E. Seemann, M. Fritz, B. Schiele),
  - Patchwork of Parts (Y. Amit, A. Trouvè),
  - Compositional (S. Geman).

- **Wholistic**
  - Convolutional Networks (Y. LeCun),
  - Boosted Cascades (P. Viola, M. Jones),
  - Bag-of-Features (A. Zisserman, F. F. Li).

**POSE SPACE**

Let $\mathcal{Y}$ be the space of poses and $\mathcal{Y}_1, \ldots, \mathcal{Y}_K$ a partition.

Let $I$ be an image and $\mathbf{Y} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_K)$, where, for each $k$, $\mathcal{Y}_k$ is a Boolean variable stating if there is an object in $I$ with pose in $\mathcal{Y}_k$.

**TRAINING A DETECTOR**

**FRAGMENTATION**

Given a training set

$$\left\{ (I^{(i)}, Y^{(i)}) \right\}_t,$$

we are interested in building, for each $k$, a classifier

$$f_k : I \rightarrow \{0, 1\}$$

for predicting $\mathcal{Y}_k$.

Without additional knowledge about the relationship between $k$, $\mathcal{Y}_k$ and $I$, we would train $f_k$ with

$$\left\{ (I^{(i)}, Y_k^{(i)}) \right\}_t.$$

Hence, each $f_k$ is trained with a fragment ($\approx 1/K$) of the positive population.

For faces, typically $\mathbf{y} = (u_c, v_c, \theta, s)$. Cats are more complex.

A coarse description is $\mathbf{y} = (u_h, v_h, s_h, u_b, v_b)$.
To avoid fragmentation, samples are often normalized in pose. Let
\[ \xi : I \to \mathbb{R}^N \]
denote an \( N \)-dimensional vector of features. Let \( \psi : \{1, \ldots, K\} \times I \to I \) be a transformation such that the conditional distribution of \( \xi(\psi(k, I)) \) given \( Y_k = 1 \) does not depend on \( k \).

Then train a single classifier \( g \) with the training set
\[ \{(\psi(k, I(t)), Y_k^{(t)})\}_{t,k} \]
and define
\[ f_k(l) = g(\psi(k, I)) \]
Here, samples are aggregated for training: all positive samples are used to build each \( f_k \).

However:
- Evaluating \( \psi \) is computationally intensive for any non-trivial transformation.
- The mapping \( \psi \) does not exist for a complex pose.

Hence, in practice, fragmentation is still the norm to deal with many deformations. But how does one deal with complex poses?
Hence, we define a family of pose-indexed features as a mapping

\[ X : \{1, \ldots, K\} \times I \rightarrow \mathbb{R}^N \]

such that they are stationary in the following sense: for every \( k \in \{1, \ldots, K\} \), the probability distribution

\[ P(X(k, I) = x \mid Y_k = 1), \ x \in \mathbb{R}^N \]

does not depend on \( k \).

**Toy example, 1D signal**

\[ Y = \left\{ (\theta_1, \theta_2) \in \{1, \ldots, N\}^2, \ 1 < \theta_1 < \theta_2 < N \right\} \]

\[ P(I \mid Y = (\theta_1, \theta_2)) = \prod_{n<\theta_1} \phi_0(I_n) \prod_{\theta_1 \leq n \leq \theta_2} \phi_1(I_n) \prod_{\theta_2 < n} \phi_0(I_n) \]

\[ X((\theta_1, \theta_2), I) = (I(\theta_1 - 1), I(\theta_1), I(\theta_2), I(\theta_2 + 1)) \]

\[ P(X = (x_1, x_2, x_3, x_4) \mid Y = (\theta_1, \theta_2)) = \phi_0(x_1)\phi_1(x_2)\phi_1(x_3)\phi_0(x_4) \]

We then define

\[ f_k(I) = g(X(k, I)), \ k \in \{1, \ldots, K\}, \]

where one single classifier \( g : \mathbb{R}^N \rightarrow \{0, 1\} \) is trained with

\[ \left\{ (X(k, I^{(t)}), Y_k^{(t)}) \right\}_{t,k} \]

Notice that stationarity is the main condition required to ensure that the training samples are identically distributed.
Let $e_{\phi,\sigma}(I, u, v)$ be the presence on an edge in image $I$ at location $(u, v)$ with orientation $\phi \in \{0, \ldots, 7\}$ at scale $\sigma \in \{1, 2, 4\}$.
CAT DETECTION

BASE FEATURES

We use three types of stationary features:

- Proportion of an edge \((\phi, \sigma)\) in a window \(W\).

- \(L^1\)-distance between the histograms of orientations at scale \(\sigma\)
in two windows \(W\) and \(W'\).

- \(L^1\)-distance between the histograms of gray-levels in two windows \(W\) and \(W'\).

CAT DETECTION

CLASSIFIER

We build classifiers with Adaboost and an asymmetric weightingby sampling. If the weighted error rate is

\[
L(h) = \sum_{t,k} \omega_{t,k} \mathbb{I}\{h(k, t) \neq z^{(t)}_k\},
\]

we use all the positive samples, and sub-sample negative ones with

\[
\mu(k, t) \propto \omega_{t,k} \mathbb{I}\{y^{(t)}_k = 0\}.
\]

Total of 2327 scenes containing 1683 cats, 85% for training.

CAT DETECTION

STATIONARY FEATURES

The windows \(W\) and \(W'\) are indexed either with respect to the head or with respect to the head-belly.

Head registration

CAT DETECTION

FOLDED HIERARCHIES

SUMMARY

Experiments (not shown) demonstrate:

- Fragmentation applied to complex poses is disastrous in practice;

- Naive, brute-force exploration of the pose space is computationally intractable.

Alternatively:

- Stationary features largely avoid fragmentation;

- Hierarchical search concentrates computation on ambiguous regions.

We call this alternative a *folded hierarchy of classifiers*. 
**Folded Hierarchies**

Summary (cont.)

**Scene Parsing**

Strategy

![Diagram of scene parsing strategy with g1 and g2](image)

**Scene Parsing**

Error Criterion

![Diagram of scene parsing error criterion](image)

**Scene Parsing**

Error Rates

<table>
<thead>
<tr>
<th>TP</th>
<th>H</th>
<th>B</th>
<th>HB</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td></td>
<td></td>
<td>11.29 (3.61)</td>
</tr>
<tr>
<td>80%</td>
<td>17.23 (3.87)</td>
<td>5.07 (1.08)</td>
<td></td>
</tr>
<tr>
<td>70%</td>
<td>11.40 (2.89)</td>
<td>1.88 (0.32)</td>
<td></td>
</tr>
<tr>
<td>60%</td>
<td>7.80 (1.98)</td>
<td>0.95 (0.22)</td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>5.41 (1.62)</td>
<td>0.53 (0.13)</td>
<td></td>
</tr>
</tbody>
</table>

Average number of FAs per $640 \times 480$
A folded hierarchy of classifiers is highly efficient:
- For (offline) learning;
- For (online) scene processing.

It combines the strengths of:
- Template matching;
- Powerful, wholistic machine learning;
- Hierarchical search.

However, this is achieved at the expense of
- Rich annotation of the training samples.
- Designing stationary features.