Deep Epitomic Nets and Scale/Position Search for Image Classification

TTIC_ECP team

George Papandreou
Toyota Technological Institute
at Chicago

Iasonas Kokkinos
Ecole Centrale Paris/INRIA
TTIC_ECP entry in a nutshell

Goal: Invariance in Deep CNNs

Part 1: Deep epitomic nets: local translation (deformation)
Part 2: Global scaling and translation

(0) Baseline: max-pooled net
13.0%

(1) epitomic DCNN
11.9%
~1% gain

(2) epitomic DCNN + search
10.56%
~1.5% gain

Fusion (1)+(2)
10.22%

Top-5 error. All DCNNs have 6 convolutional and 2 fully-connected layers.
Deep Convolutional Neural Networks (DCNNs)

Cascade of convolution + max-pooling blocks
(deformation-invariant template matching)

Our work: different blocks (P1) & different architecture (P2)

Krizhevsky et al.: ImageNet Classification with Deep CNNs, NIPS 2012
Part 1: Deep epitomic nets
Epitomes: translation-invariant patch models

Patch Templates

Separate modeling: more data & less power per parameter
Epitomes: a lot more for just a bit more

EM-based training

Jojic, Frey, Kannan: Epitomic analysis of appearance and shape, ICCV 2003
Benoit, Mairal, Bach, Ponce: Sparse image representation with epitomes, CVPR 2011
Grosse, Raina, Kwong, Ng: Shift-invariant sparse coding, UAI 2007
Mini-epitomes for image classification

Dictionary of mini-epitomes

Dictionary of patches (K-means)

Gains in (flat) BoW classification

Papandreou, Chen, Yuille: Modeling Image Patches with a Dictionary of Mini-Epitomes, CVPR14
From flat to deep: Epitomic convolution

Max-Pooling

\[ y_{i,k} = \max_{p \in N_{\text{image}}} x^T_{i+p} w_k \]

Max over image positions

\[ k = 1, 2, \ldots \]

Epitomic Convolution

\[ y_{i,k} = \max_{p \in N_{\text{epitome}}} x^T_i w_{k,p} \]

Max over epitome positions

\[ k = 1, 2, \ldots \]

Deep Epitomic Convolutional Nets

Supervised dictionary learning by back-propagation

Deep Epitomic Convolutional Nets

Parameter sharing: faster and more reliable model learning

Consistent improvements

(0) Baseline: max-pooled net

13.0%

(1) epitomic DCNN

11.9%

~1% gain
Part 2: Global scaling and translation
Scale Invariance challenge

Category-dependent (ear detector)

Scale-dependent (area)

Dogs
Scale Invariance challenge

- Category-dependent (ear detector)
- Scale-dependent

Dogs
Skyscrapers
Scale Invariance challenge

Training set

Category-dependent (ear detector)

Scale-dependent

Dogs

Skyscrapers
Scale Invariance challenge

Rule: Large skyscrapers have ears, large dogs don’t
Scale Invariant classification

Category-dependent

Scale-dependent

feature

\[ x \rightarrow \{ x_{s_1}, \ldots, x_{s_K} \} \]

‘bag’ of features

\[ F(x) \rightarrow \{ F(x_{s_1}), \ldots, F(x_{s_K}) \} \]

\[ F'(x) = \frac{1}{K} \sum_{k=1}^{K} F(x_{s_k}) \]

This work: \[ F'(x) = \max_k F(x_{s_k}) \]

Step 1: Efficient multi-scale convolutional features

- **$I(x,y)$**: Image pyramid
- **$I(x,y,s)$**: Stitched multi-scale features
- **Patchwork($x,y$)**
- **220x220x3**
- **$C(x,y)$**: Multi-scale convolutional features
- **$C(x,y,s)$**: Unstitched multi-scale features
- **5x5x512**

References:
- Dubout, C., Fleuret, F.: Exact acceleration of linear object detectors. ECCV 2012
Step 2: From fully connected to fully convolutional
Step 2: From fully connected to fully convolutional
Step 2: From fully connected to fully convolutional
TTIC_ECP: Deep Epitomic CNNs and Explicit Scale/Position Search

Step 3: Global max-pooling

Consistent, explicit position and scale search during training and testing
For free: argmax yields 48% localization error

\[ G_c = \max_{x,y} F_c(x, y) \]

learned class-specific bias

(0) Baseline: max-pooled net
13.0%

(1) epitomic DCNN
11.9%

~1% gain

(2) epitomic DCNN + search
10.56%

~1.5% gain

Fusion (1)+(2)
10.22%
Deep Epitomic Nets and Scale/Position Search for Image Classification

Goal: Invariance in Deep CNNs

(0) Baseline: max-pooled net
(1) Epitomic DCNN
(2) search

13.0% → 11.9% → 10.56% → 10.22%

~1% gain → ~1.5% gain

DCNN: 6 Convolutional + 2 Fully Connected layers

The Deeper the Better: stay tuned!
Epitomic implementation details

- Architecture of our deep epitomic net (11.94%)

<table>
<thead>
<tr>
<th>Layer</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>epit-conv</td>
<td>epit-conv</td>
<td>conv</td>
<td>conv</td>
<td>conv</td>
<td>epit-conv</td>
<td>full + dropout</td>
<td>full + dropout</td>
<td>full</td>
</tr>
<tr>
<td></td>
<td>+ lrn</td>
<td>+ lrn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output channels</td>
<td>128</td>
<td>384</td>
<td>768</td>
<td>768</td>
<td>768</td>
<td>768</td>
<td>4096</td>
<td>4096</td>
<td>1000</td>
</tr>
<tr>
<td>Epitome size</td>
<td>12x12</td>
<td>8x8</td>
<td></td>
<td></td>
<td></td>
<td>5x5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filter size</td>
<td>8x8</td>
<td>6x6</td>
<td>3x3</td>
<td>3x3</td>
<td>3x3</td>
<td>3x3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input stride</td>
<td>4x4</td>
<td>3x3</td>
<td>1x1</td>
<td>1x1</td>
<td>1x1</td>
<td>3x3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Epitome stride</td>
<td>2x2</td>
<td>1x1</td>
<td></td>
<td></td>
<td></td>
<td>1x1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Training took 3 weeks on a single Titan (60 epochs)
- Standard choices for learning rate, momentum, etc.
Pyramidal search implementation details

- Image warp to square image. Position in mosaic is fixed
- Scales: 400, 300, 220, 160, 120, 90 pixels → Mosaic: 720 pixels