Deep filter banks for texture recognition and segmentation

Mircea Cimpoi, University of Oxford
Subhransu Maji, UMASS Amherst
Andrea Vedaldi, University of Oxford
Texture understanding

Indicator of materials properties, e.g. brick vs wooden

Complementary to shape

Correlated with identity but not the same

Kickstarted orderless image representations (e.g. Bag of words)

[Bajcsy et al. 73, Julesz 81, Ojala et al. 96, 02, Dana et al. 99, Leung and Malik 99, Varma and Zisserman 03, 05, Caputo et al. 05, Lazebnik et al. 05, 06, Timofte and Van Gool 12 Sharma et al. 12, Sifre and Mallat 13, Sharan et. al 09, 13]
Is there a relation between texture representations and deep convolutional neural networks?
Texture representations

Filters + histogramming

image x

[Leung and Malik 99, 01, Schmid 01, Varma and Zisserman 02, 05]
Texture representations

Filters + histogramming

image $x$

[Leung and Malik 99, 01, Schmid 01, Varma and Zisserman 02, 05]
Texture representations

Filters + histogramming

image $x$  bank of filters  local descriptors  VQ + histogram

[Leung and Malik 99, 01, Schmid 01, Varma and Zisserman 02, 05]
Texture representations

Filters + histogramming

image $x$  bank of filters  local descriptors

[Leung and Malik 99, 01, Schmid 01, Varma and Zisserman 02, 05]
Texture representations

Filters + histogramming

image \( x \)  bank of filters  local descriptors  VQ + histogram

[Leung and Malik 99, 01, Schmid 01, Varma and Zisserman 02, 05]
Texture representations

Filters may be **non-linear**

\[ \phi(x) \]

Local descriptor

(SIFT, LBP, LTP, HOG, SURF, BRIEF, ORB, ...) non-linear filters

local descriptors

VQ + histogram

[Geusebroek et al 03, Lowe 99, Ojala et al. 02, Dalal and Triggs 05, Bay et al. 06, Tan and Triggs 10]
Texture representations

Replace histograms with an order-less pooling encoder

(Orderless pooling encoder: Bag-of-words, Fisher Vector, VLAD, sparse coding, …)

(SIFT, LBP, LTP, HOG, SURF, BRIEF, ORB, …)

Local descriptor

non-linear filters

local descriptors

encoder

[Sivic and Zisserman 03, Csurka et al. 04, Perronnin and Dance 07, Perronnin et al. 10, Jegou et al. 10]
Texture representations vs CNNs

image → Handcrafted features → feature field → Orderless pooling → \( \phi(x) \)
Texture representations vs CNNs

Handcrafted features

Orderless pooling

non-linear filters

feature field

encoder representation

\[ \phi(x) \]

\[ \begin{align*}
C_1 & \quad C_2 & \quad C_3 & \quad C_4 & \quad C_5 & \quad f_6 & \quad f_7 & \quad f_8 \\
\end{align*} \]

[Krizhevsky et al. 12]
Texture representations vs CNNs

Image feature field

Handcrafted features

Orderless pooling

non-linear filters

encoder representation

"convolutional" layers

"fully-connected" (FC) layers
Mix and match

image \rightarrow \text{non-linear filters} \rightarrow \text{Handcrafted local descriptors} \rightarrow \text{CNN local descriptors} \rightarrow \text{feature field} \rightarrow \text{encoder} \rightarrow \text{representation} \phi(x)
Mix and match

Standard texture representation

- Handcrafted local descriptors
- CNN local descriptors
- Orderless pooling
- CNN FC pooling

$x \rightarrow \phi(x)$

[Siivc and Zisserman 03, Csurka et al. 04, Perronnin and Dance 07, Perronnin et al. 10, Jegou et al. 10]
Mix and match

Standard application of CNN

FC-CNN

[Chatfield et al. 14, Girshick et al. 2014, Gong et al. 14, Razavin et al. 14]
Mix and match

Order-less pooling of CNN local descriptors

- Handcrafted local descriptors
- CNN local descriptors
- CNN FC pooling
- Feature field
- Encoder
- Representation
- non-linear filters
- $\phi(x)$
Mix and match

CNN descriptors pooled by Fisher Vector

image

non-linear filters

feature field

encoder

representation

Handcrafted local descriptors

Fisher Vector

CNN local descriptors

CNN FC pooling

$\phi(x)$

FV-CNN
Mix and match

- Handcrafted local descriptors
- CNN local descriptors
- Non-linear filters
- Feature field
- Encoder
- Representation

See [Perronnin and Larlus 15] Poster 2B-44
Tested modules

Baseline CNN models

- **Typical**
  - AlexNet [Krizhevsky et al. 12]
  - VGG-M [Chatfield et al. 14]

- **Deep**
  - VGG-VD [Simonyan Zisserman 14]
Tested modules

Baseline CNN models

- **Typical**
  - AlexNet [Krizhevsky et al.12]
  - VGG-M [Chatfield et al.14]

- **Deep**
  - VGG-VD [Simonyan Zisserman 14]

Local image descriptors

- **Handcrafted**: SIFT [Lowe 99]
- **Learned**: Convolutional layers of CNNs
Tested modules

Baseline CNN models

- **Typical**
  - AlexNet [Krizhevsky et al. 12]
  - VGG-M [Chatfield et al. 14]
- **Deep**
  - VGG-VD [Simonyan Zisserman 14]

Local image descriptors

- **Handcrafted**: SIFT [Lowe 99]
- **Learned**: Convolutional layers of CNNs

Pooling encoders

- **Classical**
  - Bag of Visual Words [Sivic and Zisserman 03, Csurka et al. 04]
  - Fisher Vector [Perronnin and Dance 07, Perronnin et al. 10]
- **CNN**
  - FC layers [Chatfield et al. 14, Girshick et al. 2014, Gong et al. 14, Razavin et al. 14]
How does FV-CNN perform compared to other descriptors?

How does FV-CNN handle region recognition?

What is the benefit of FV-CNN in domain-transfer?
Datasets and benchmarks

**Material recognition (FMD)**  
[Liu et al. 10, Sharan et al. 13]

**Fine-grained recognition (CUB)**  
[Wah et al. 11]

**Object recognition (VOC07)**  
[Everingham et al. 07]

**Texture attribute recognition (DTD)**  
[Cimpoi et al. 14]

**Scene recognition (MIT Indoors)**  
[Quattoni and Torralba 09]

**Things and stuff (MSRC)**  
[Criminisi 04, Shotton et al. 06]
Finding 1) BoVW < FV

Finding 2) SIFT < CNN
Finding 3) FV-pooling $\geq$ CNN-pooling

Finding 4) Deep $\geq$ shallow
CNN vs Fisher Vector pooling

Finding 3)
FV-pooling ≥ CNN-pooling

Finding 4)
Deep ≥ shallow
### Breadth of applicability

<table>
<thead>
<tr>
<th>Domain</th>
<th>Fully connected (VGG-VD)</th>
<th>Fisher vector (VGG-VD)</th>
<th>SoA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALOT (texture)</td>
<td>88.7</td>
<td>79.8</td>
<td>97.8</td>
</tr>
<tr>
<td>FMD (texture)</td>
<td>77.7</td>
<td>79.8</td>
<td>95.9</td>
</tr>
<tr>
<td>DTD (textures)</td>
<td>62.9</td>
<td>72.3</td>
<td>85.9</td>
</tr>
<tr>
<td>VOC07 (objects)</td>
<td>81.7</td>
<td>85.9</td>
<td>85.2</td>
</tr>
<tr>
<td>MIT (scenes)</td>
<td>67.6</td>
<td>81.0</td>
<td>81.0</td>
</tr>
<tr>
<td>CUB+R (fine-grained)</td>
<td>62.8</td>
<td>73.0</td>
<td>76.4</td>
</tr>
</tbody>
</table>

**Finding 5)** FV + CNN applies to many diverse domains

[Cimpoi et al. 14, Sulc and Matas 14, Sharan et al. 13, Wei and Levoy 14, Zhou et al. 14, Zhang et al. 14, Burghouts and Geusebroek 09, Sharan et al. 09, Everingham et al. 08, Quattoni and Torralba 09, Wah et al. 11]
How does FV-CNN perform compared to other descriptors?

How does FV-CNN handle region recognition?

What is the benefit of FV-CNN in domain-transfer?
Texture recognition in the “wild” and “clutter” (OS)

A new texture benchmark

- Based **OpenSurfaces dataset** [Bell et al. 13, 15]
- Textures in the wild (uncontrolled conditions)
- Textures in **clutter** (do not fill the image)

First extensive evaluation of texture material/attribute recognition of this kind
Regions: the crop & describe approach

E.g. R-CNN

Pros: straightforward & universal construction

[Chatfield et al. 14, Jia 13, Girshick et al. 2014, Gong et al. 14, Razavin et al. 14]
Crop & describe limitations

Expensive

May distort images

Can only do rectangles
Regions: the pooling encoder approach

Share the local descriptors

Cons: restricted to a convolutional representation

Pros: fast, flexible, multiscale, and often more accurate

[He et al. 2014, Cimpoi et al. 2015]
Finding 6) FV pooling ≫ CNN pooling for small, variable regions (and faster too!)
How does FV-CNN perform compared to other descriptors?

How does FV-CNN handle region recognition?

What is the benefit of FV-CNN in domain-transfer?
Late vs early transfer

Transfer either the fully connected or the convolutional layers

Deep feature encoder

Late transfer (Fully-connected CNN)
Late vs early transfer

Transfer either the fully connected or the convolutional layers

depth filter bank

Late transfer
(Fully-connected CNN)

Early transfer
(Fisher vector CNN)
Early vs late transfer (FV-CNN)

Pre-train CNN (AlexNet)
- ImageNet
  - 1.5M images
  - Generic objects, e.g., trilobite
- Transfer from dissimilar domain

Train-test SVM
- MIT Indoor
  - Indoor scenes, e.g., library
  - 6.7K images
- Transfer from similar domain

Transfer from similar domain
- MIT Places
  - Indoor/outdoor scenes, e.g., tennis court
  - 2.5M images

VGG-VD
- Early transfer (Fisher vector CNN)
  - 69.7%
- Late transfer (Fully-connected CNN)
  - 65.0%
- 81.0%

[Zhou et al. 14]
**Summary**

**Hybrid architectures**: Classical feature encoders can be used effectively as CNN building blocks, or inspire new ones.

**FV-CNN** has several benefits:

- Simple
- Excellent performance in diverse domain
- Works particularly well and efficiently with image regions
- Reduces the domain gap in transfer learning

A new **benchmark** for material and texture attribute recognition in clutter.

Many more experiments in the paper, IJCV version, and DPhil thesis.
Number of Gaussians

![Graph showing accuracy vs. number of Gaussians for different methods.](image)
Effect of Depth on CNN Features

Conv5 for VGG-VD – extra 4%
SIFT – same as Conv2 / Conv3
Dimensionality reduction and descriptor size

Effects of dimensionality reduction of FV-CNN descriptor on Pascal VOC07

mAP (%) vs Descriptor Size

- 32D
- 64D
- 80D
- 128D
- None
Visualizing top FV components

Locations of CNN descriptors that correspond to the FV-CNN components most strongly associated with the texture words (bubbly, studded, wrinkled ... )