LARGE URBAN ZONE CLASSIFICATION ON SPOT-5 IMAGERY WITH CONVOLUTIONAL NEURAL NETWORKS

by V. A. Krylov, M. de Martino, G. Moser and S. B. Serpico
University of Genoa, Italy
Outline

- Classification problem;
- Basics of deep learning with CNN;
- Proposed CNN architecture;
- Experimental analysis;
- Conclusions.
Applied problem

**Urban Land Recycling Information Services for Sustainable Cities (URBIS)** EU-project (2014-2017) targeting to identify and monitor the vacant and abandoned areas in European Large Urban Zones:

- 305 European cities with 100k+ populations;
- Mapping of the land-cover / land-use, imperviousness, vegetation extent;
- 2006, 2012 acquisition campaigns, target: future regular updates every 6 years.

[Images of SPOT-5 ©CNES 2012, Classification, Imperviousness map, Vegetation map]
Classification problem

- Extensive **preprocessing**: 2x10 m SPOT-5 **stereo** images
  >> 2x5 m **pan-sharpened** >> 1x2.5m **super-resolution**
- **3 bands**: Near Infrared, Red and Green.
- **3 Large Urban Areas**: 2006, 2012 (~5 full scenes), seasonality.
- **Ground truth** prepared via photo-interpretation.

![SPOT-5 ©CNES 2012](image1.png) ![GeoEye ©GoogleMaps 2013](image2.png)

<table>
<thead>
<tr>
<th>Thematic Classes</th>
<th>Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>building</td>
<td>pink</td>
</tr>
<tr>
<td>roads &amp; pavements</td>
<td>green</td>
</tr>
<tr>
<td>low vegetation</td>
<td>blue</td>
</tr>
<tr>
<td>tall vegetation</td>
<td>yellow</td>
</tr>
<tr>
<td>bare ground</td>
<td>orange</td>
</tr>
<tr>
<td>water</td>
<td>red</td>
</tr>
<tr>
<td>railroads</td>
<td>black</td>
</tr>
</tbody>
</table>
Classification problem

Disadvantages of the feature-based classification, such as Random Forest (RF) and Support Vector Machines (SVM):

- challenging and time consuming **feature-election** process;
- limited room for **data abstraction** (transfer learning);
- possibility of **accuracy** improvement with CNN.
Deep Learning

Shallow vs. deep learning:

- A priori knowledge and feature selection in shallow learning;
- Architecture engineering with high degree of generality in deep learning, i.e., Feature vs. Architecture engineering.
Deep Learning

Key ideas:

• Most learning (perception) in the human brain may be due to just one learning algorithm;

• Machine learning can imitate the human brain learning process to efficiently address machine vision problems;
Advantages of the deep architecture for the image analysis tasks:

- Automatic selection of the image feature sets;
- Use of tractable non-linear modeling
  - RF: piece-wise constancy, CNN: [mild] non-linearity via activation functions, SVM: full non-linearity with kernels/mixtures;
- Faster parallel computer implementation
  - GPUs, CPU clusters;
- Large (huge) databases, potential for weak labeling
  - ImageNet: 1.2 mln images, 1k classes;
- Recent additions to the deepnets architecture:
  - ReLU, Dropout, etc.
Convolutional Neural networks

- Use of 2d local filters as the network’s construction blocks to extract image information imitating the process performed at the visual cortex:

- This procedure is repeated densely to achieve translation invariance:

- The parameters (filters, weights) are estimated via an iterative cycle of forward- and back-propagations relying on the use of the (stochastic) gradient descent estimator
**Employed CNN architecture**

Why not employ (directly/finetune) one of the existing off-the-shelf CNNs:

- **Different image problem**: application, set of bands, properties;

- Dense labeling obtained via a **patch-based approach**: limited depth.
Employed CNN architecture

Classification is addressed as a **patch-based** (25-by-25) decision process.

**CNN architecture:**

- 2 convolution layers;
- 2 pooling (stoch.) layers;
- 2 fully-connected layers;
- ReLU activation functions;
- 1 dropout layer.
Implementation

- 25-by-25 patches provide an accuracy vs. quality compromise;
- 50% of training patches overlap inside ground truth (98k patches);
- Dense classification output;
- Caffe (C++-Python) deep learning framework.
- Manually prepared sets of non-overlapping training and testing sets

**CNN parameters:**

- 2k epochs;
- Drop-rate 0.5;
- Momentum for SGD set to 0.9.
Experimental analysis

- **Training** performed in 3h: 98k of 25x25 patches;
- **Dense classification** of a 10k-by-10k SPOT-5 image recovered in 15 minutes.
- **Random forest** (CPU-parallelized): feature selection (manual), spatial-spectral feature extraction (30min), training (3min), classification (7min)

**GPU:**
- GeForce GTX 960,
- 4GB RAM

**CPU:**
- Core-i7, (8x)3GHz,
- 16GB RAM
Experimental analysis

- **Accuracy** (numerical) vs. **quality** (over-smoothing);
- **Patch size** analysis: noise vs. interpretability, MMU.
Experimental analysis

- **Accuracy** (numerical) vs. **quality** (over-smoothing);
- **Patch size** analysis;
- Ground truth inside **homogeneous** areas!
- **Random-based** evaluation.

<table>
<thead>
<tr>
<th>Input SPOT-5 image</th>
<th>Classifier</th>
<th>Buildings</th>
<th>Roads</th>
<th>Low veg.</th>
<th>Bare soil</th>
<th>Tall veg.</th>
<th>Overall</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ostrava, Czech Republic</td>
<td>CNN</td>
<td>0.9732</td>
<td>0.8893</td>
<td>0.9978</td>
<td>0.95</td>
<td>0.9993</td>
<td>98.27%</td>
<td>0.9781</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>0.9805</td>
<td>0.7972</td>
<td>0.9257</td>
<td>0.9144</td>
<td>0.9983</td>
<td>95.64%</td>
<td>0.9449</td>
</tr>
<tr>
<td>Amiens, France</td>
<td>CNN</td>
<td>0.9971</td>
<td>0.7059</td>
<td>0.9698</td>
<td>0.9673</td>
<td>0.9346</td>
<td>94.14%</td>
<td>0.9228</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>0.9945</td>
<td>0.5972</td>
<td>0.8732</td>
<td>0.8687</td>
<td>0.9905</td>
<td>90.57%</td>
<td>0.8794</td>
</tr>
<tr>
<td>Osnabrueck, Germany</td>
<td>CNN</td>
<td>0.9998</td>
<td>0.7727</td>
<td>0.9305</td>
<td>0.9191</td>
<td>0.9911</td>
<td>95.14%</td>
<td>0.9340</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>0.9995</td>
<td>0.5729</td>
<td>0.9252</td>
<td>0.8912</td>
<td>0.9903</td>
<td>94.21%</td>
<td>0.9217</td>
</tr>
</tbody>
</table>
Conclusions

• CNN architecture **outperforms** the benchmark (RF) classifier with a set of carefully chosen features;

• CNN approach demonstrated a potential to **data abstraction**:
  • could be exploited as a *in-built* domain adaptation tool to mitigate inner-variability of the datasets (e.g., seasonality, light conditions);

• The CNN **complexity** might be reduced by designing a fully convolution framework.