Deep Learning for Satellite/Aerial Image Analysis

Emmanuel Maggiori

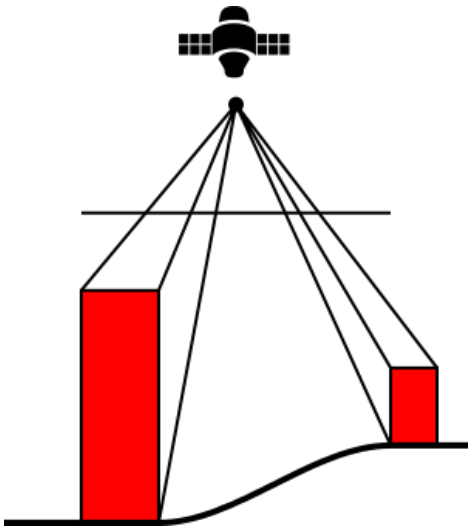
Data Science Meetup

Based on my recent work at
Inria & Université Côte d'Azur

Context

Remote sensing images

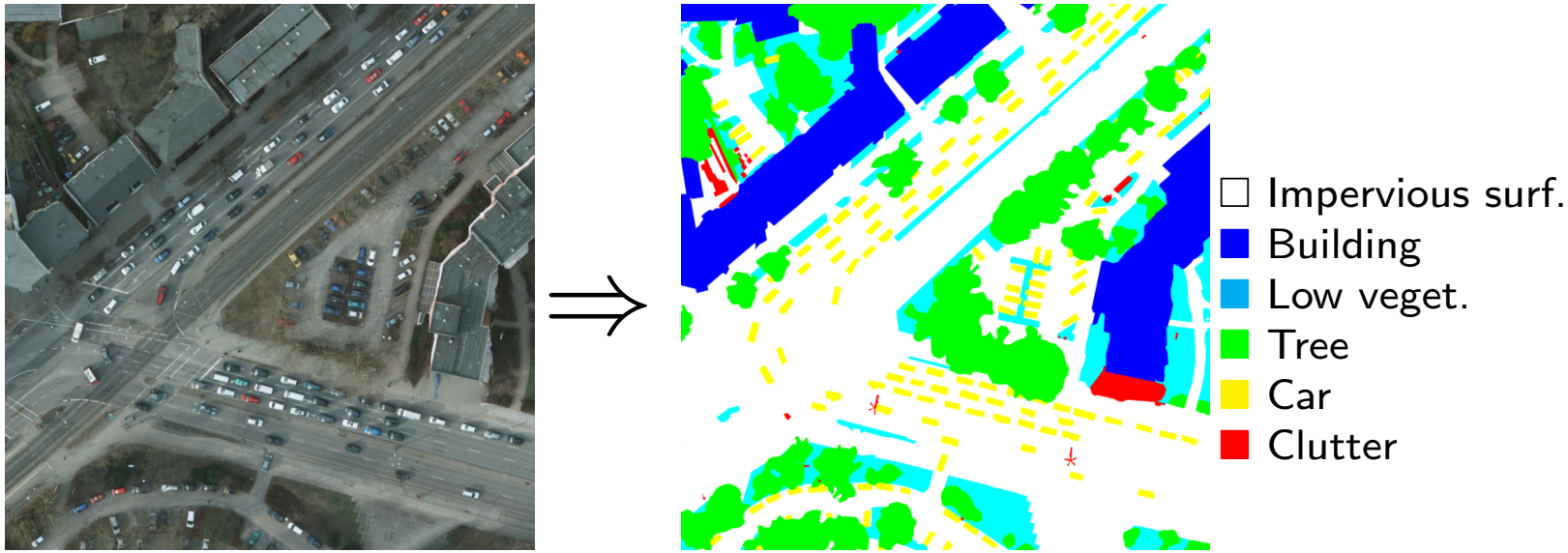
→ acquired from satellites/airplanes/drones



E.g., Pléiades optical satellite image:

Remote sensing image classification

Classification: assign a semantic class to every pixel



Input Output

Context: Large-scale data sources

- Increasing amount & openness of data.
  E.g., Pléiades:
  - Entire earth every day
  - 1-band ("grayscale") image at $\approx$ 0.5 m spatial resolution
  - 4-band image (R-G-B-Infrared) at $\approx$ 1 m spatial resolution
  - 2 bytes per pixel and band (values beyond [0..255])
- Intra-class variability:

Chicago

Vienna

Austin

$\Rightarrow$ Need for high-level contextual reasoning (shape, patterns,...)

$\Rightarrow$ Generalization to different locations

Outline

1. Introduction

2. Classification with CNNs

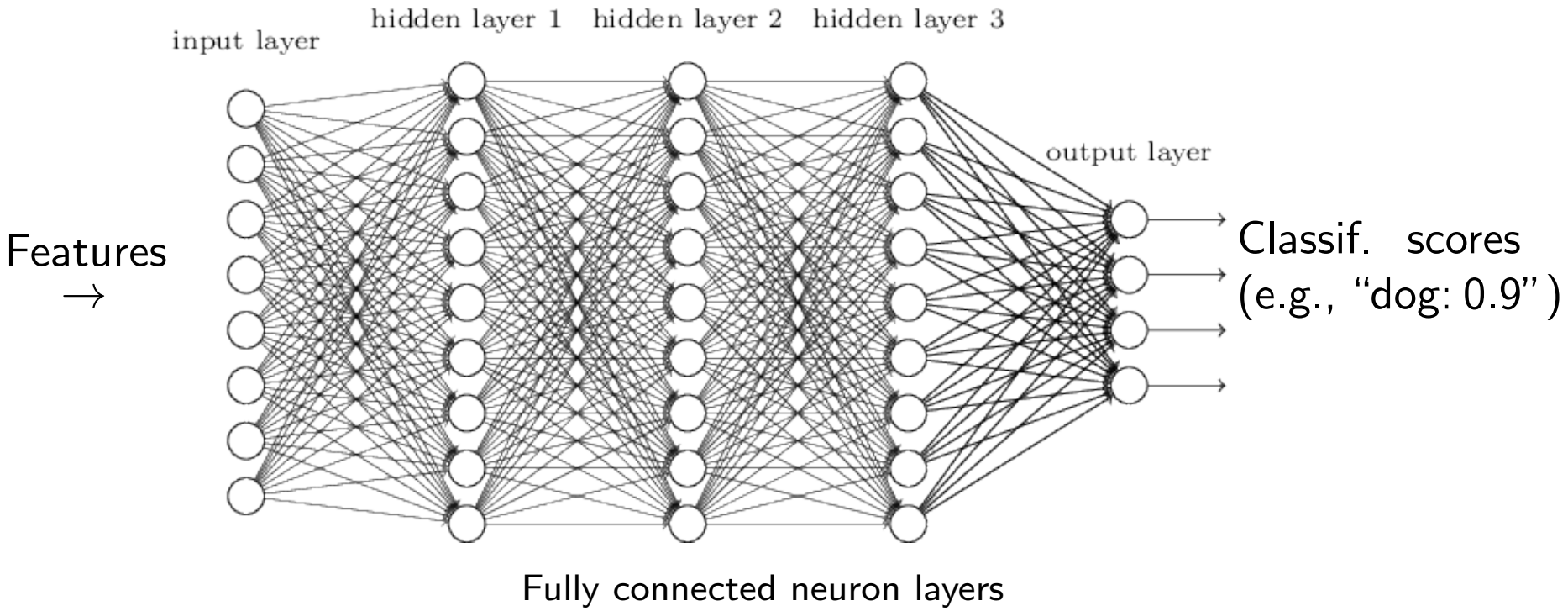
3. Challenge #1: High-resolution classification

4. Challenge #2: Imperfect training data

5. Concluding remarks

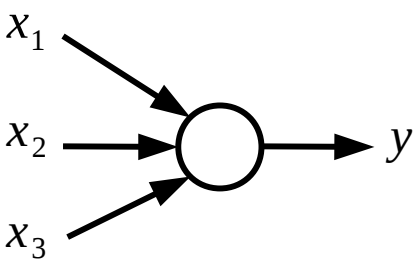
Artificial neural networks

Multilayer perceptron (MLP)



Fully connected neuron layers

Neuron



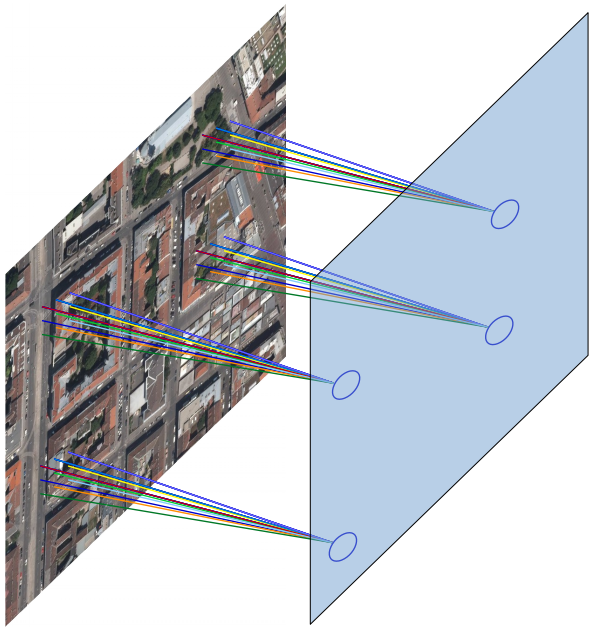
- $y = \sigma(\sum a_i x_i + b)$, σ nonlinear
- Parameters (a_i, b of all neurons) define the function
- Trained from samples by stoch. gradient descent

Convolutional neural networks (CNNs)

- Input: the image itself
- {*Convolutional* layers + *pooling* layers}* + MLP

Convolutional layer

Learned convolution filters $\rightarrow$ feature maps



Special case of fully connected layer:

- Only local spatial connections
- Location invariance

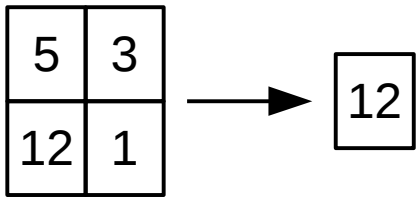
$\Rightarrow$ Makes sense in image domain (or text, time series,...)

Convolutional neural networks (CNNs)

Pooling layers

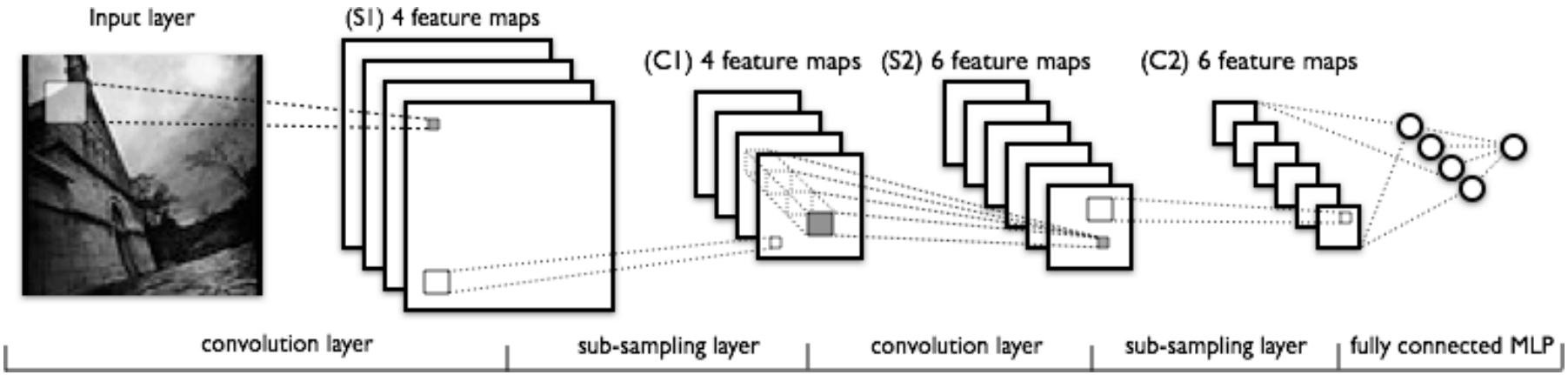
Subsample feature maps

- Increase *receptive field* ☺
- Downgrade resolution
  - Robustness to spatial variation ☺
  - Not good for *pixelwise* labeling ☹



Max pooling

Overall categorization CNN



Source: deeplearning.net

Outline

1. Introduction

2. Classification with CNNs

3. Challenge #1: High-resolution classification

4. Challenge #2: Imperfect training data

5. Concluding remarks

Challenge #1: Yielding high-resolution outputs

Recent work

Three families of architectures:

- *Dilation* (Chen et al., 2015; Dubrovina et al., 2016,...)
- *Unpooling/deconv.* (Noh et al., 2015; Volpi and Tuia, 2016,...)
- *Skip networks* (Long et al., 2015; Badrinarayanan et al., 2015,...)

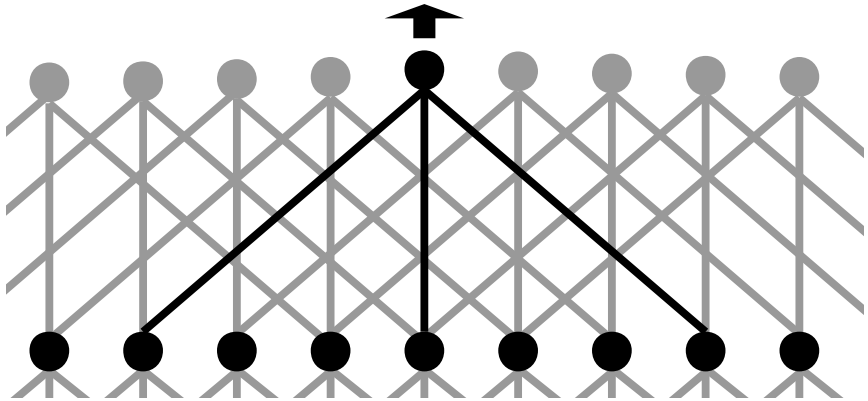
Goal: CNN architecture that addresses recognition/localization trade-off

---

Analysis of SoA: E. Maggiori, Y. Tarabalka, G. Charpiat, P. Alliez. "High-Resolution Aerial Image Labeling with Convolutional Neural Networks", TGRS 2017.

E.g., dilation networks

Convolutions on non-contiguous locations:



⇒ Larger context without introducing more parameters

- Not robust to spatial deformation
  (e.g., detect road located *exactly* 5px away)

Proposed method: MLP network

Premise

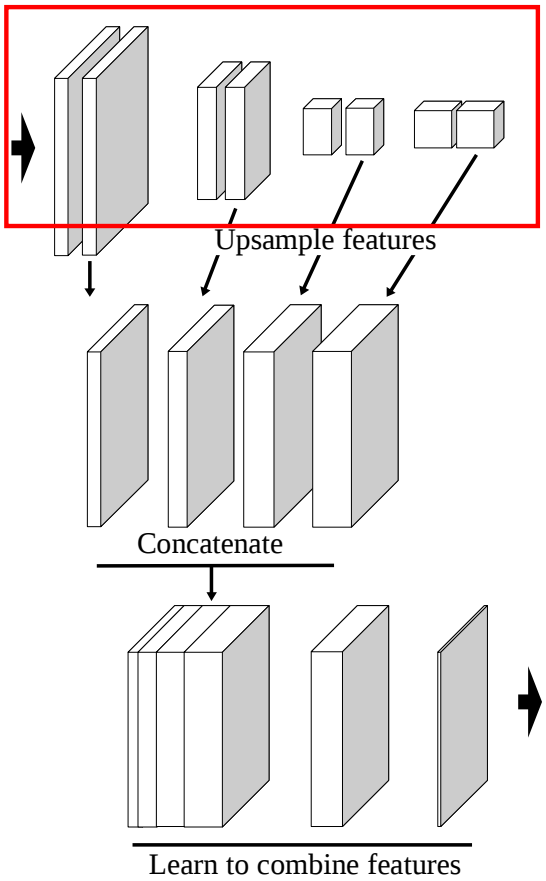
- CNNs do not need to "see" everywhere at the same resolution
- E.g., to classify central pixel:

Full resolution context

Full resolution only near center

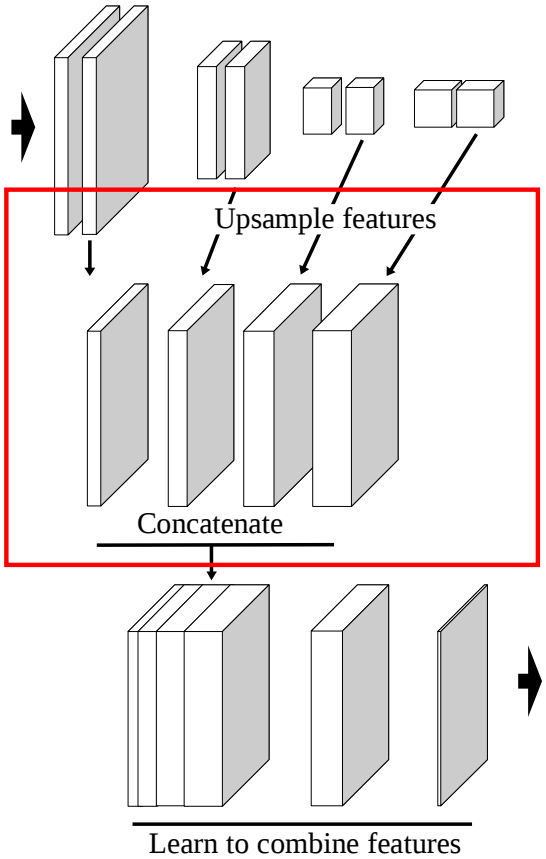
$\Rightarrow$ Combine resolutions to address trade-off, in a flexible way

MLP network



Base CNN

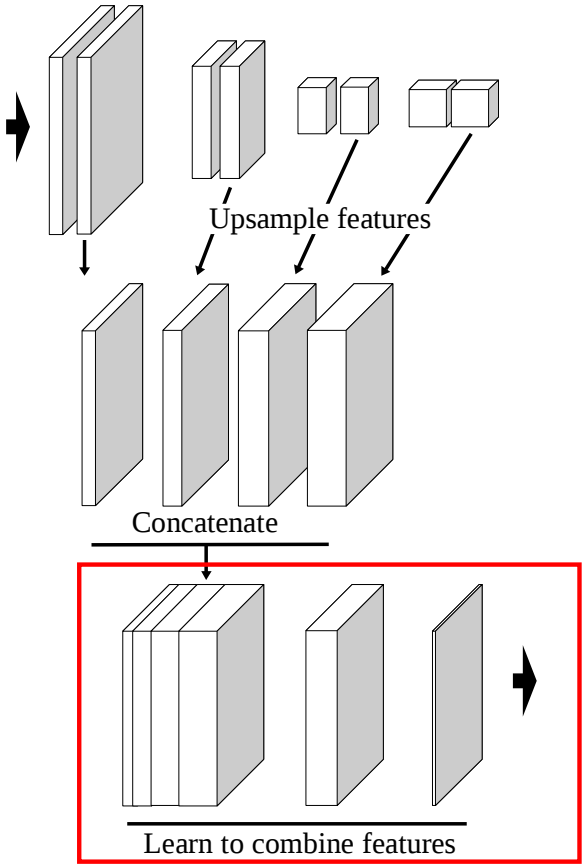
MLP network



- Extract intermediate features
- Upsample to the highest res.
- Concatenate

$\Rightarrow$ Pool of features (e.g., edge detectors, object detectors)

MLP network



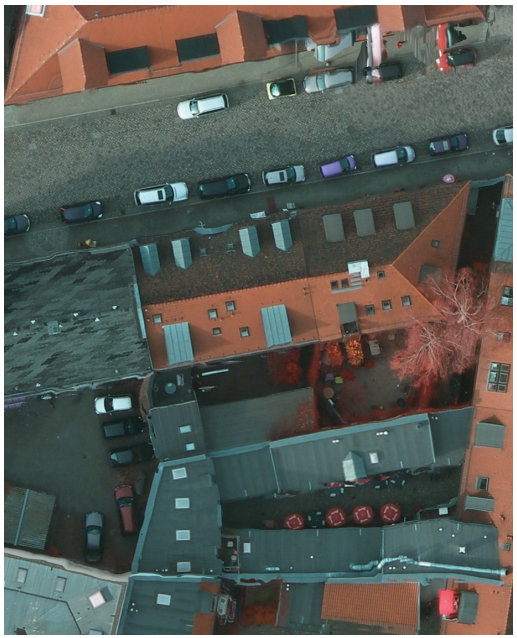
- Multi-layer perceptron (MLP) learns how to combine those features
  $\Rightarrow$ Output classif. map
- Pixel by pixel (series of 1×1 convolutional layers)

Experiments

Datasets

ISPRS 2D semantic labeling contest:

Vaihingen (9 cm)



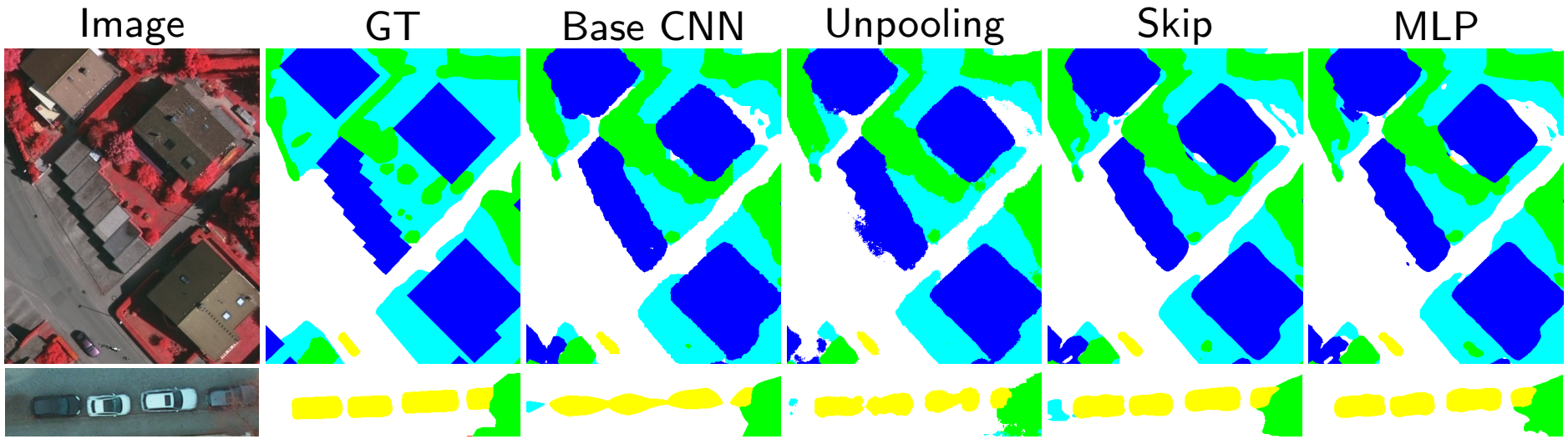
Potsdam (5 cm)

- CIR + Elevation model

Results: Base CNN vs derived architectures

Vaihingen	Imp. surf.	Building	Low veg.	Tree	Car	Mean F1	Acc.
Base CNN	91.46	94.88	79.19	87.89	72.25	85.14	88.61
Unpooling	91.17	95.16	79.06	87.78	69.49	84.54	88.55
Skip	91.66	95.02	79.13	88.11	77.96	86.38	88.80
MLP	**91.69**	**95.24**	**79.44**	**88.12**	**78.42**	**86.58**	**88.92**

Potsdam	Imp. surf.	Building	Low veg.	Tree	Car	Clutter	Mean F1	Acc.
Base CNN	88.33	93.97	84.11	80.30	86.13	75.35	84.70	86.20
Unpooling	87.00	92.86	82.93	78.04	84.85	72.47	83.03	84.67
Skip	89.27	94.21	84.73	**81.23**	93.47	75.18	86.35	86.89
MLP	**89.31**	**94.37**	**84.83**	81.10	**93.56**	**76.54**	**86.62**	**87.02**



Classes: Impervious surface (white), Building (blue), Low veget. (cyan), Tree (green), Car (yellow), Clutter (red).

Results: Comparison with other methods

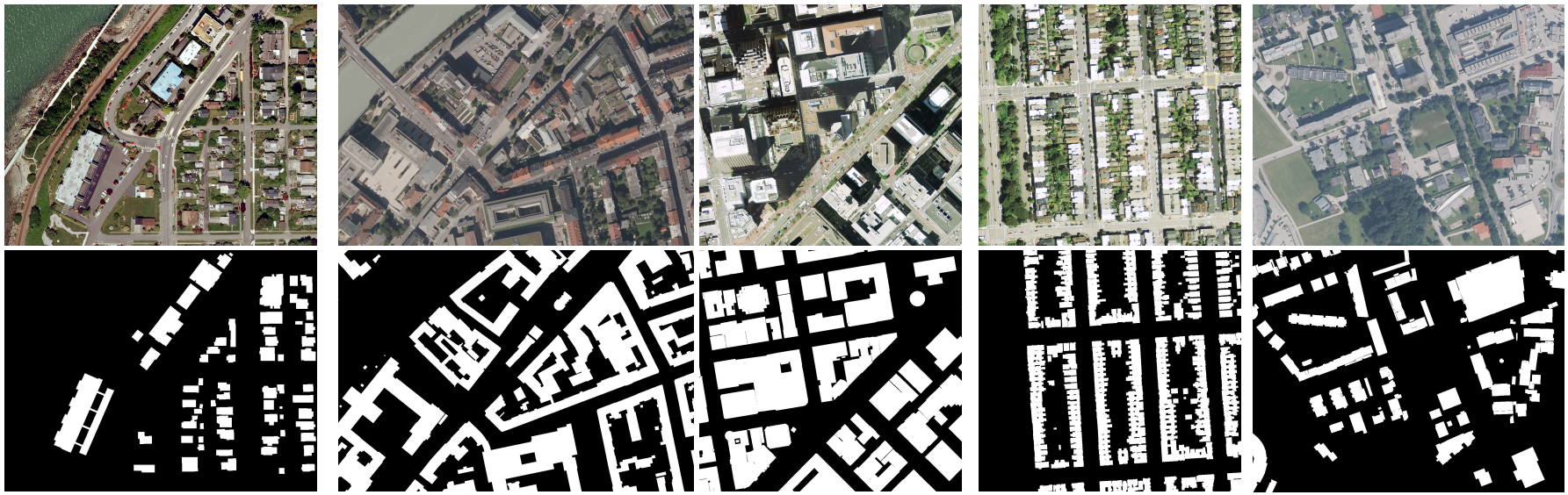
Vaihingen	Imp. surf.	Build.	Low veg.	Tree	Car	F1	Acc.
CNN+RF	88.58	94.23	76.58	86.29	67.58	82.65	86.52
CNN+RF+CRF	89.10	94.30	77.36	86.25	71.91	83.78	86.89
Deconvolution						83.58	87.83
Dilation	90.19	94.49	77.69	87.24	76.77	85.28	87.70
Dilation + CRF	90.41	94.73	78.25	87.25	75.57	85.24	87.90
MLP	**91.69**	**95.24**	**79.44**	**88.12**	**78.42**	**86.58**	**88.92**

Submission to ISPRS server

- Overall accuracy: 89.5%
- Second place (out of 29) at the time of submission
- Significantly simpler and faster than other methods

Classifying cities over the earth: can CNNs generalize?

Inria Aerial Image Labeling Dataset (810 km^2):



Bellingham Innsburck San Francisco Tyrol

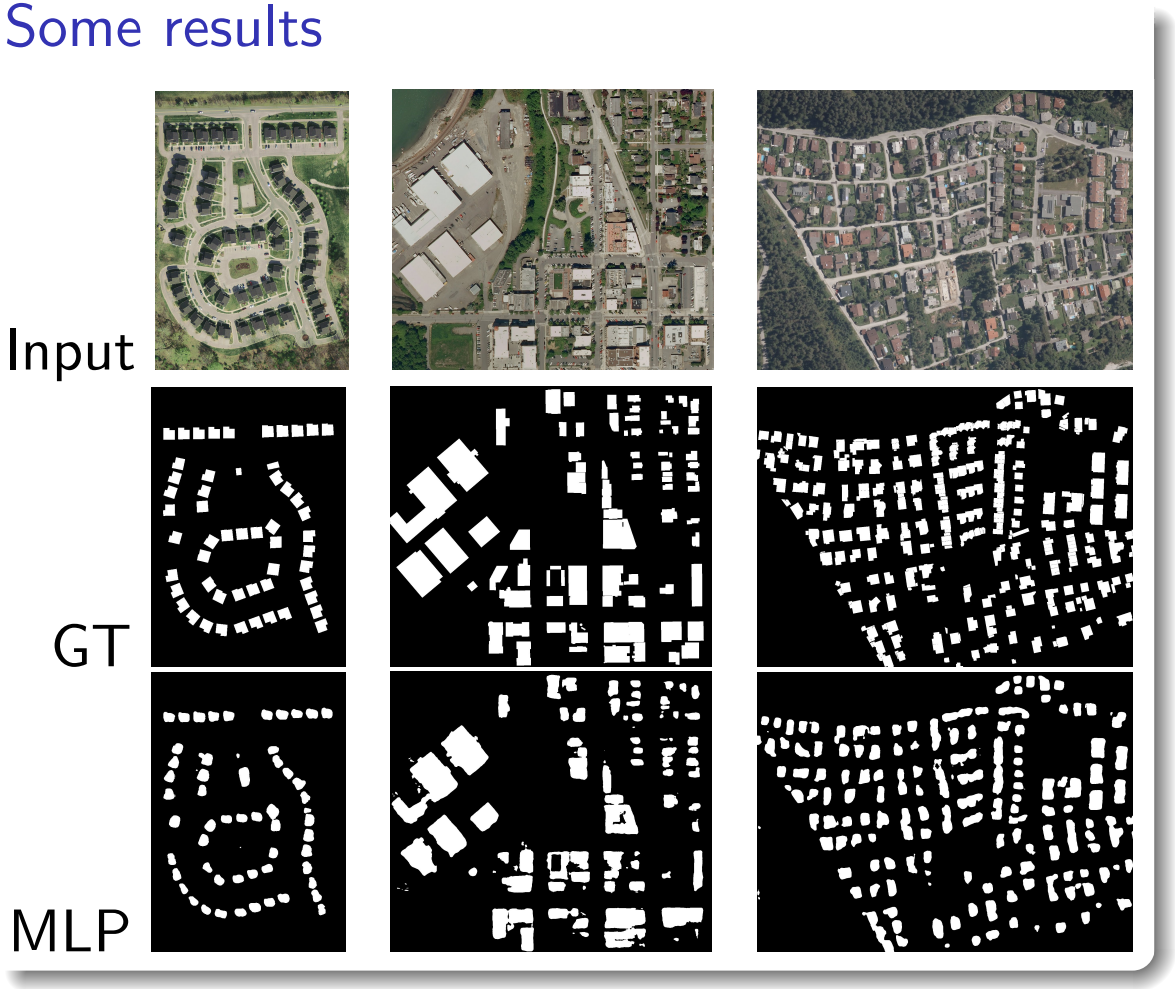
- Images over US and Austria with open images and building footprints
- Different cities in training and test sets

⇒ project.inria.fr/aerialimagelabeling

E. Maggiori, Y. Tarabalka, G. Charpiat, P. Alliez. "Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark". IGARSS 2017.

Classifying cities over the earth: can CNNs generalize?

Some results



⇒ project.inria.fr/aerialimagelabeling

Outline

1. Introduction

2. Classification with CNNs

3. Challenge #1: High-resolution classification

4. Challenge #2: Imperfect training data

5. Concluding remarks

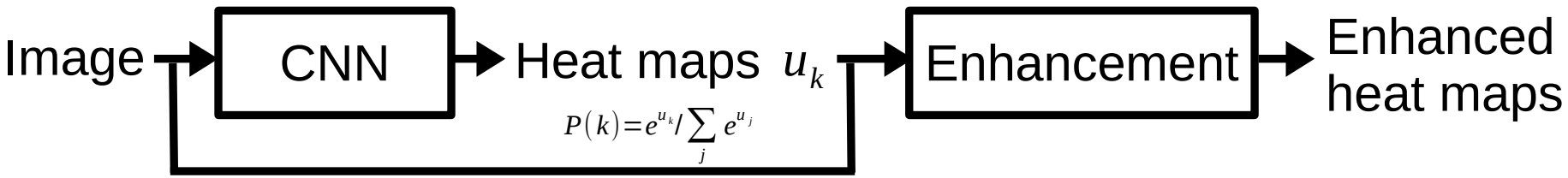
Challenge #2: Dealing with imperfect training data

Frequent misregistration/omission in large-scale data sources:

Pléiades image + OpenStreetMap (OSM) over Loire department

$\Rightarrow$ Results in fuzzy/blobby outputs

Proposed method



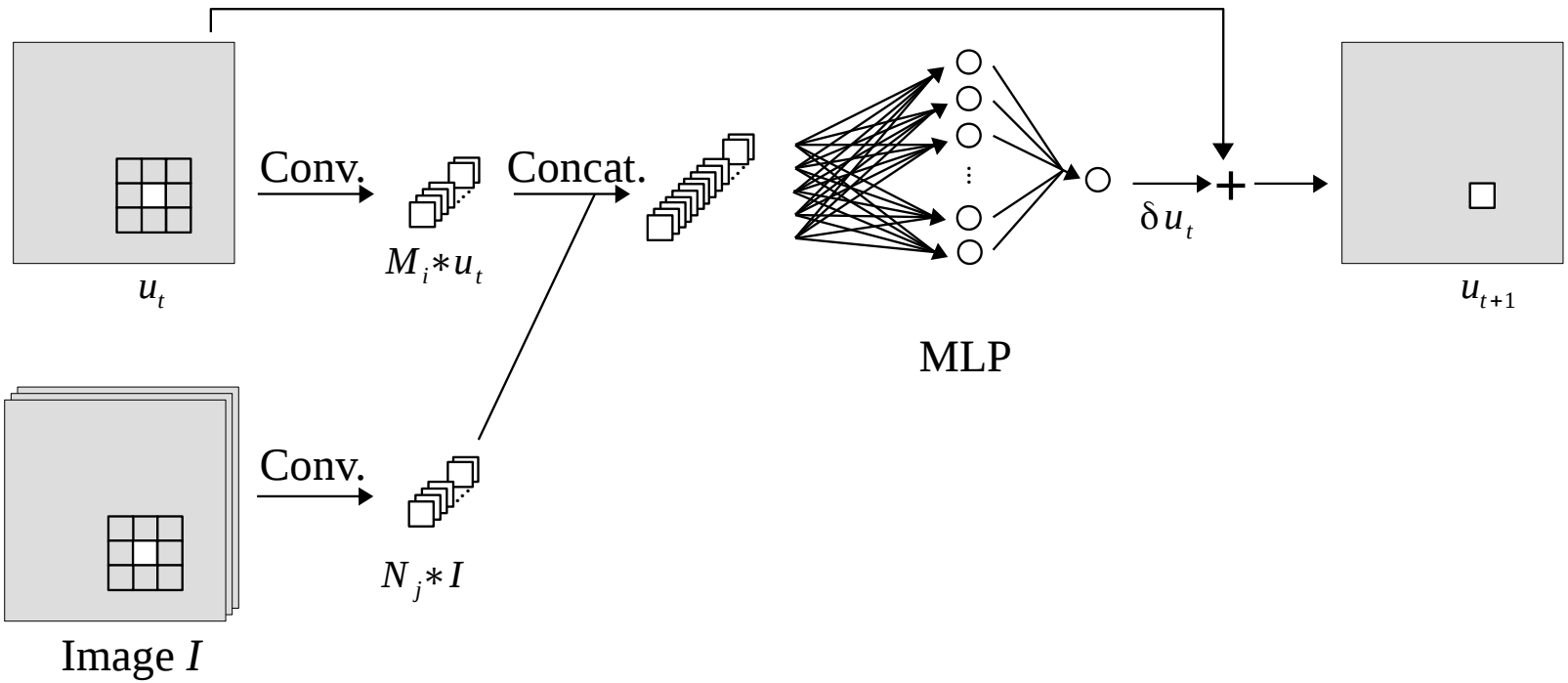
Proposed method

1. Train CNN on large amounts of imperfect data
   $\rightarrow$ Learn dataset generalities
2. Recurrent neural net to enhance outputs
   (trained on small manually labeled piece)

Analysis of SoA: E. Maggiori, G. Charpiat, Y. Tarabalka, P. Alliez. "Recurrent Neural Networks to Correct Satellite Image Classification Maps", TGRS 2017.

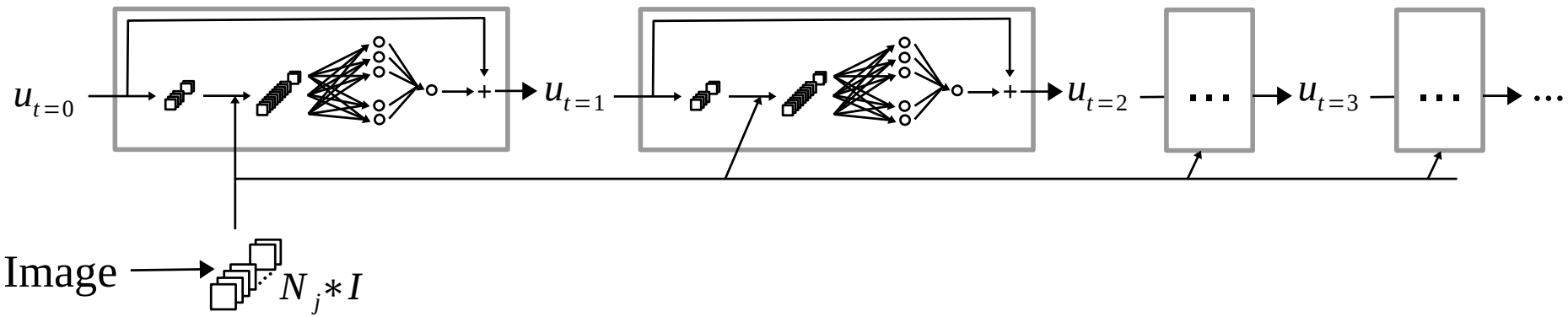
Learning an iterative enhancement process

- Generic process inspired by PDEs
- Input: classif. map + original image
- Output: enhanced map (1 iter.)
- Expressed as common CNN layers



Iterative processes as recurrent neural networks (RNNs)

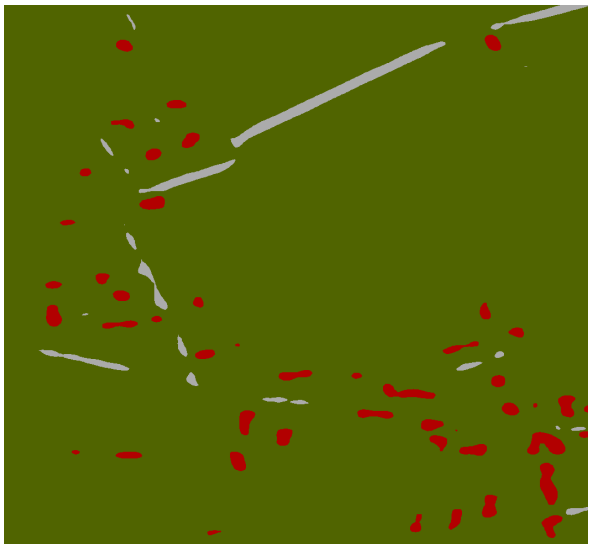
- "Unroll" iterations
- Enforce weight sharing along iterations
- Train by backpropagation as usual ("through time")
- Every iteration is meant to progressively refine the classification maps



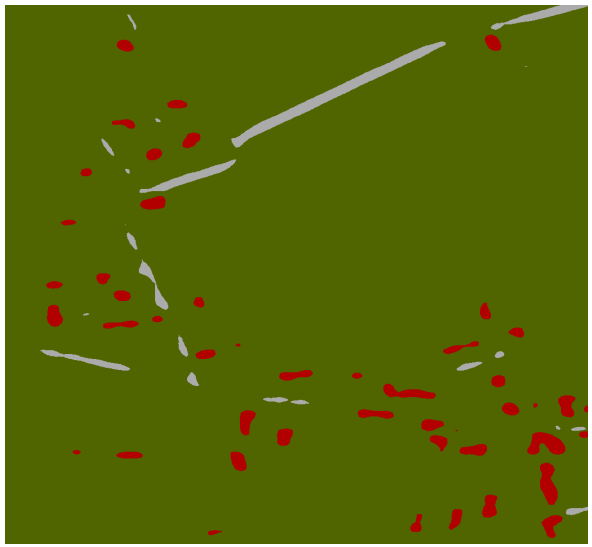
Experiments

Color input

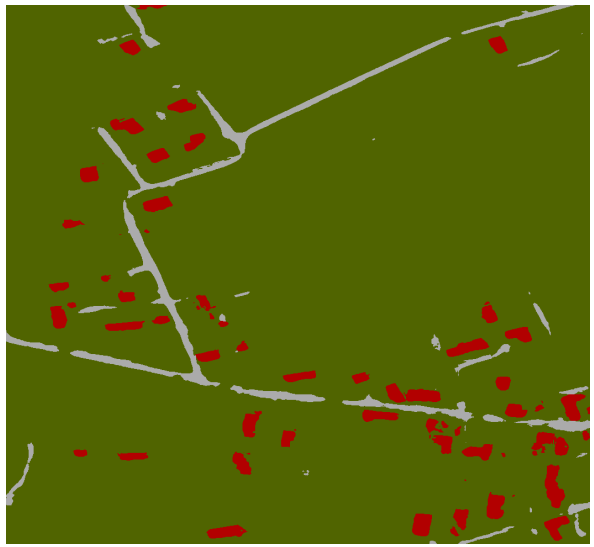
Reference



Coarse CNN

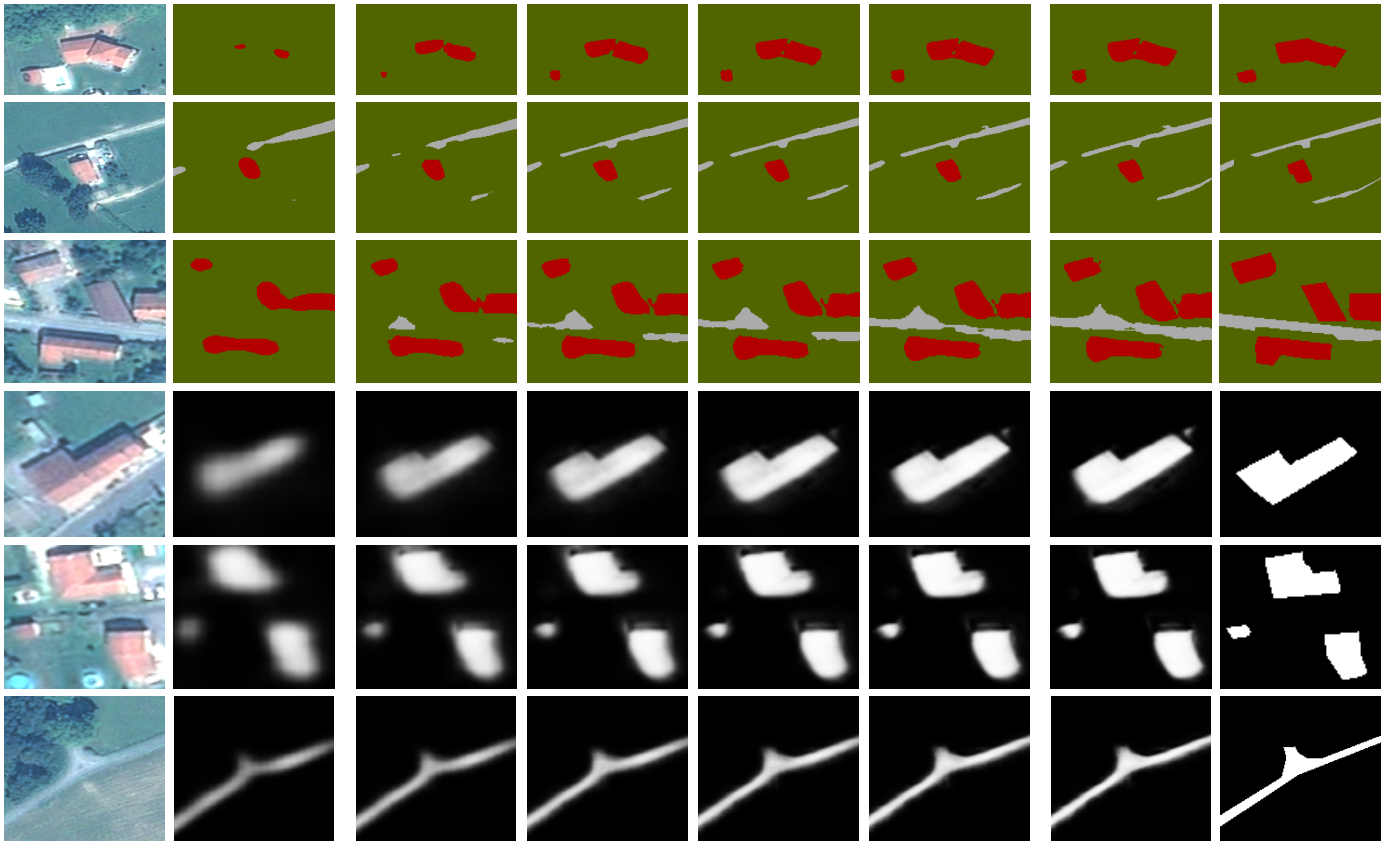


→ RNN enhancement →



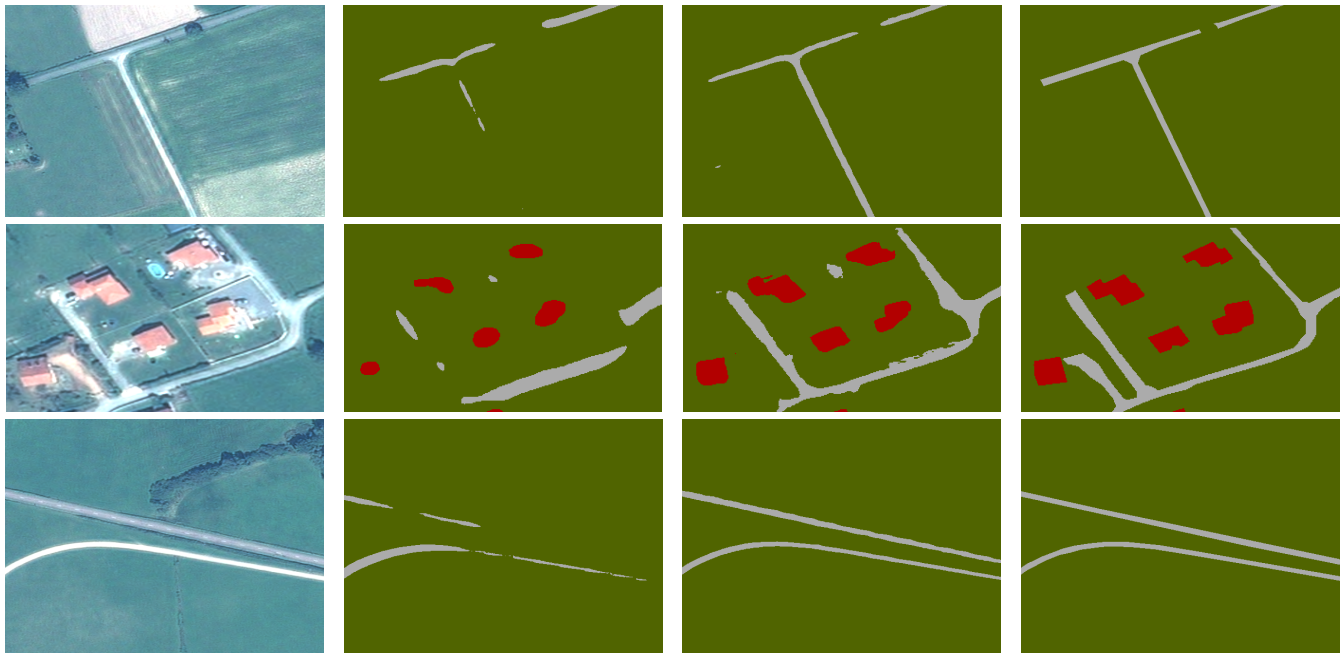
RNN output

Experiments



Experiments

More examples



- Removing recurrence constraint $\rightarrow$ Bad results

Outline

1. Introduction

2. Classification with CNNs

3. Challenge #1: High-resolution classification

4. Challenge #2: Imperfect training data

5. Concluding remarks

Concluding remarks

Key to CNNs' success

Imposing *sensible* restrictions to neuronal connections reduces optimization search space w.l.o.g:

- Better minima $\rightarrow$ better accuracy
- Computational efficiency

$\Rightarrow$ Win-win

A recurrent pattern in my reserach...

- MLP net $\rightarrow$ More accurate than more complicated models
- RNNs $\rightarrow$ Removing recurrence significantly degrades results
- ...

Concluding remarks

The "no free lunch" principle in machine learning (Wolper, 1996)

There is no such thing as a universally good classifier. A classifier is better than others under certain assumptions.

- CNNs exploit the properties of images particularly well
- Shifting efforts from feature engineering to network engineering
- Good *payoff* of the efforts,
  e.g., learning better features than handmade ones,
  convolutions $\rightarrow$ GPUs, borrowing pretrained network
- The CNNs assumptions may be their limiting factor in remote sensing classification
  $\rightarrow$ Rounded corners, unstructured outputs, etc.

Concluding remarks

- "Our method outperforms humans"
  - How's human performance measured?
  - Does your system make mistakes a human would never make? E.g., classifying a baseball bat as a toothbrush
- Beware of exaggerated results in scientific papers
  - Researching... the dataset that supports my hypothesis
- How do we obtain the training data?
- Will a 99%-accuracy method ever be integrated into a critical system? Or is anything below 100% too bad to be usable?

Thank you for your attention!

Questions?