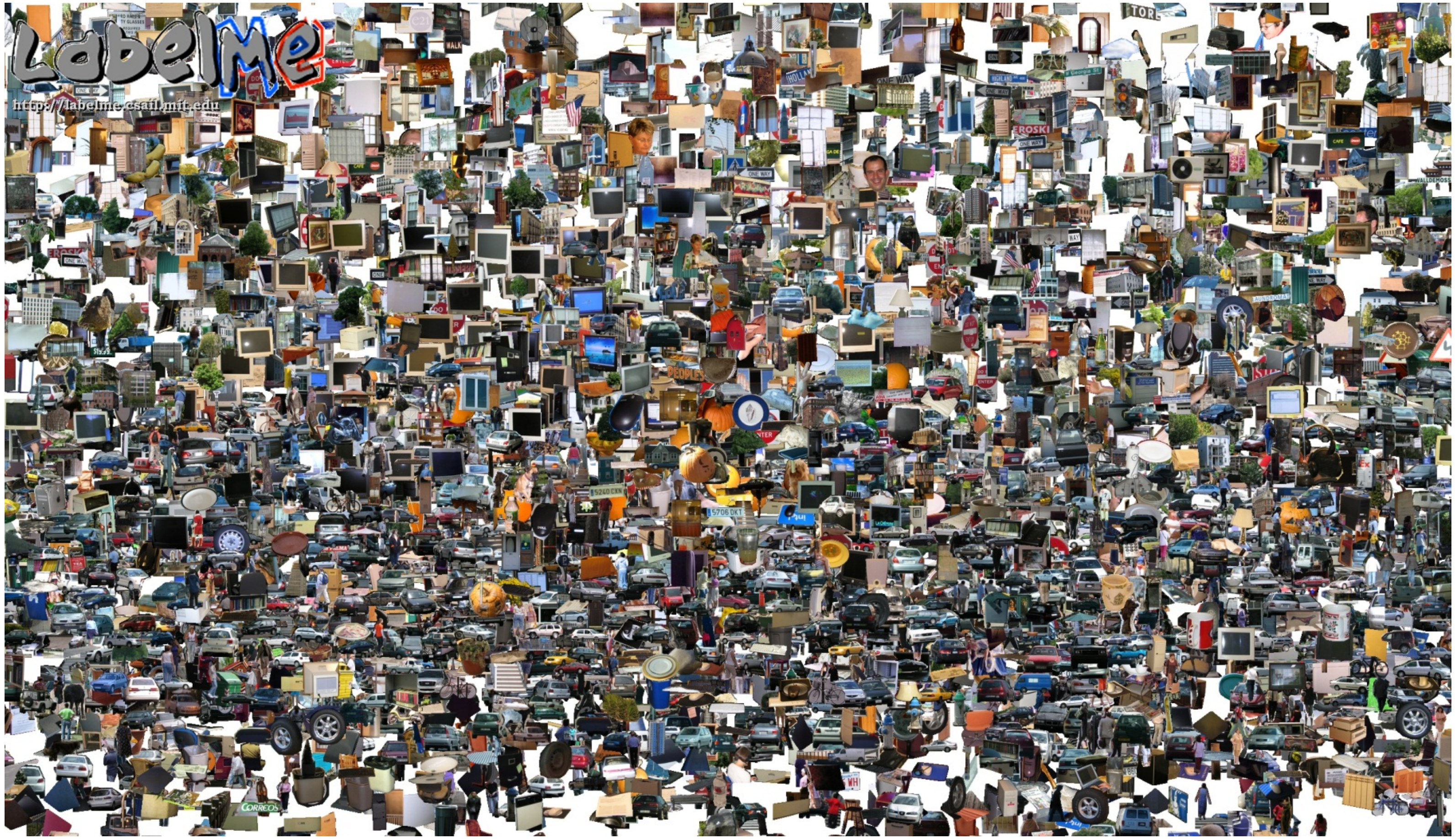
Opportunities of Scale



Computational Photography

Derek Hoiem, University of Illinois

Some slides from Alyosha Efros

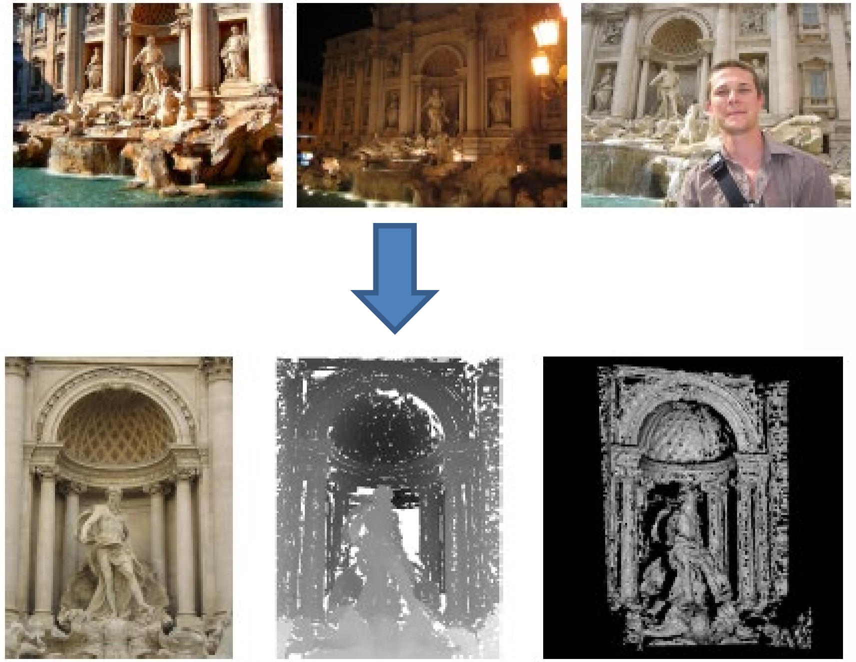
Graphic from Antonio Torralba

Today’s class: Opportunities of scale

- 3D Reconstruction
- Data-driven methods
  - 3D reconstruction
  - Scene completion
  - Im2gps
  - Colorizing
  - Recognition
  - and much more…
- Deep network representations

3D Reconstruction from Flickr

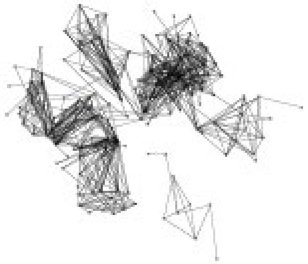
- Create detailed 3D scenes from thousands of consumer photographs
- Challenges include variations in season, lighting, occluding objects, etc.



“Building Rome in a Day”, Agarwal et al. 2009

3D Reconstruction from Flickr: How it works

1. Download ~10,000 images, convert to grayscale, compute SIFT keypoints
2. Match images
   1. Get similar images with vocabulary tree (like in recognition from last class)
   2. Match keypoints across similar images and perform geometric verification with RANSAC (similar to photo stitching)
3. Form a graph of matched images and features
4. 3D Reconstruction by triangulating points, bundle adjustment



Large-scale 3D Reconstruction

Useful references

- Snavely thesis: “Scene Reconstruction and Visualization from Internet Photo Collections”
- COLMAP: package for sparse and dense reconstruction (with two related papers) https://colmap.github.io/
- List of good papers and tutorials https://github.com/openMVG/awesome_3DReconstruction_list

Google and massive data-driven algorithms

A.I. for the postmodern world:

– all questions have already been answered…many times, in many ways

– Google is dumb, the “intelligence” is <u>in the data</u>

Google Translate

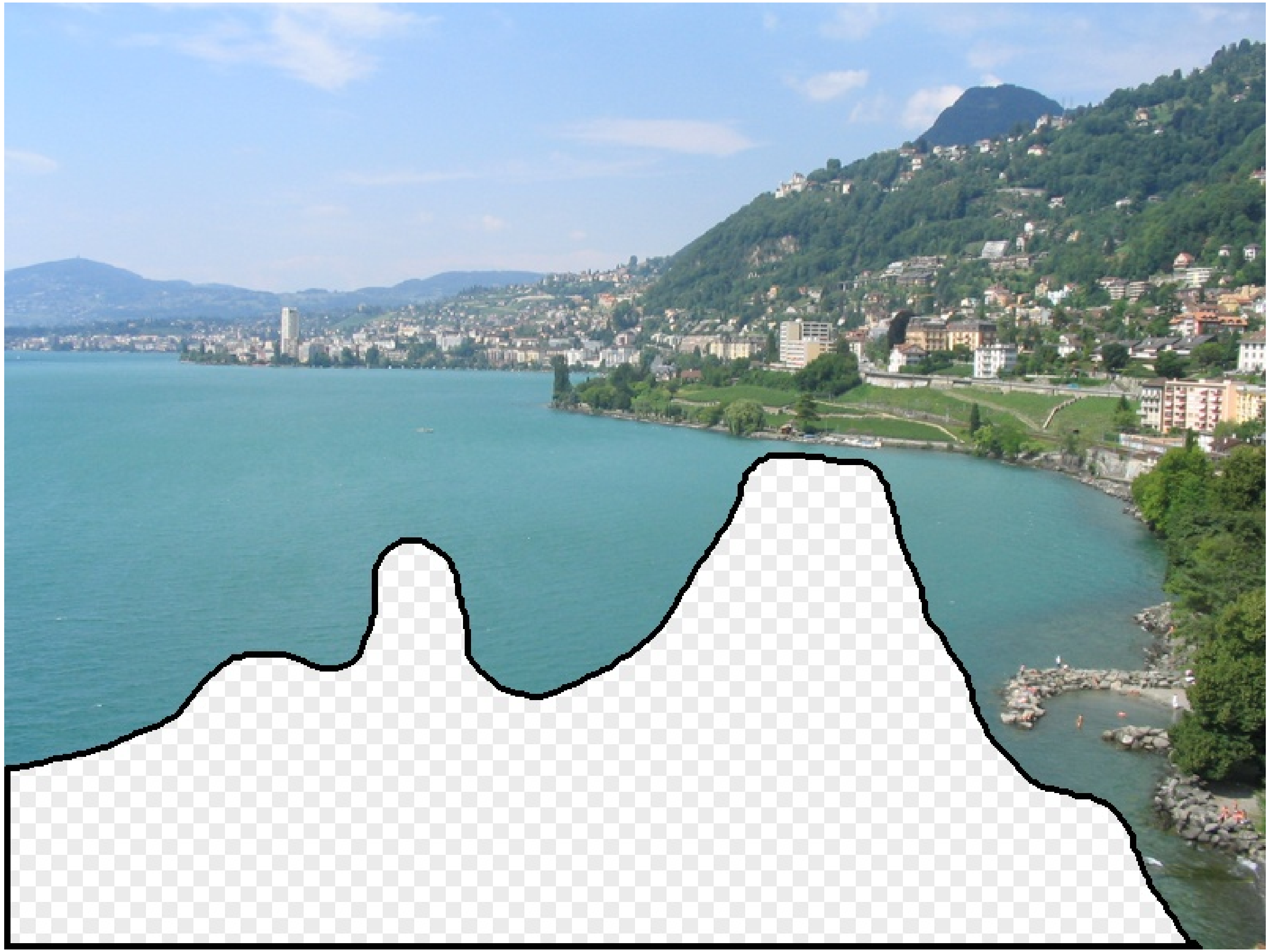
Chinese Room

- John Searle (1980)

Image Completion Example

[Hays and Efros. Scene Completion Using Millions of Photographs.
SIGGRAPH 2007 and CACM October 2008.]

What should the missing region contain?



Which is the original?

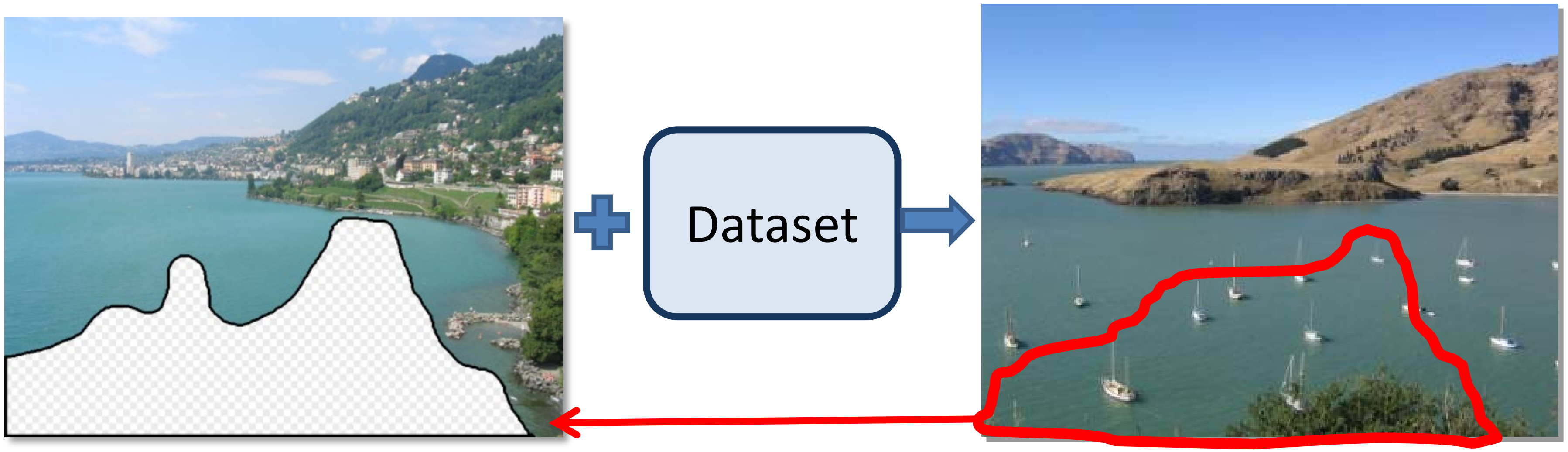
(a)

(b)

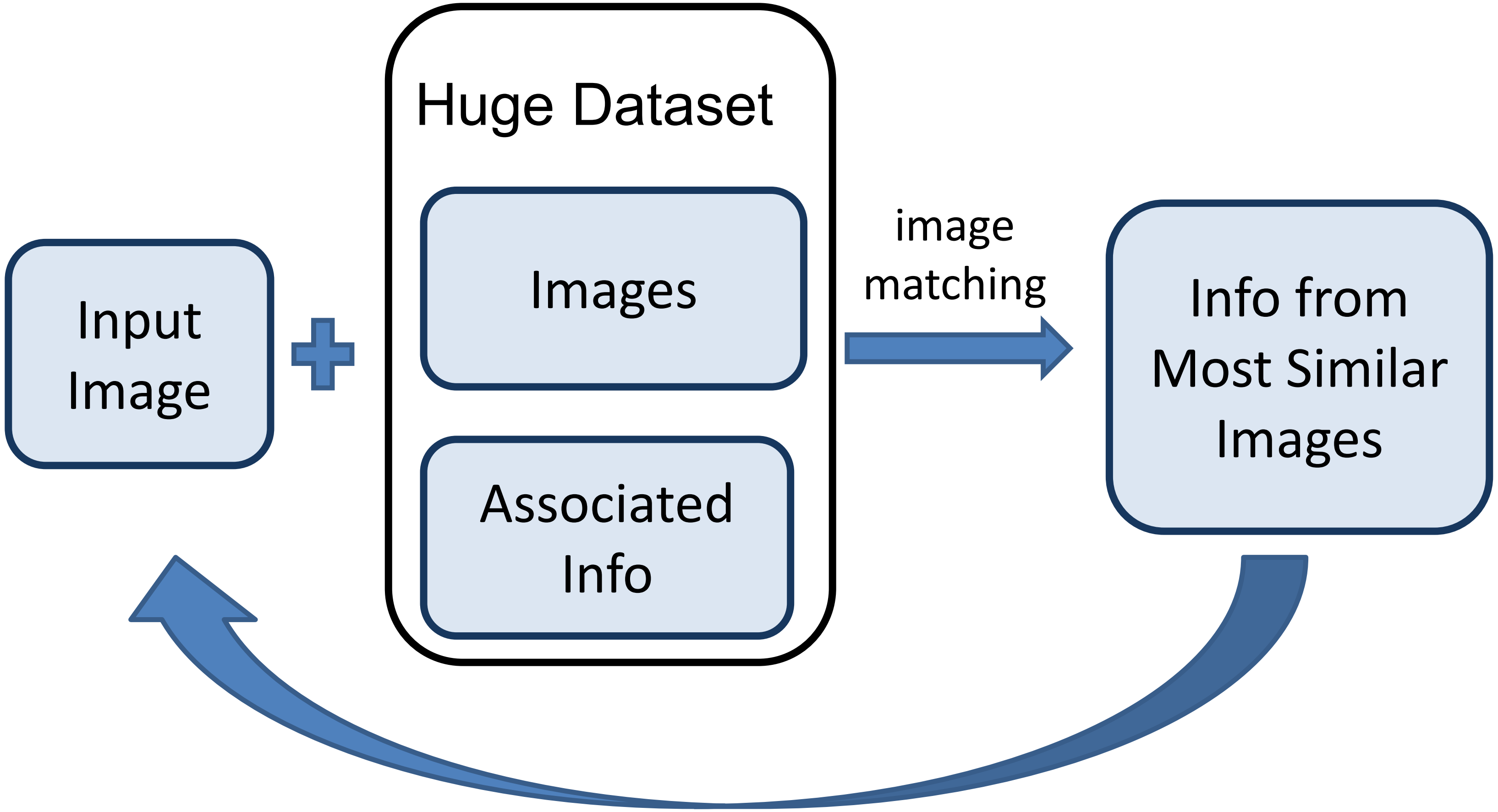
(c)

How it works

- Find a similar image from a large dataset
- Blend a region from that image into the hole

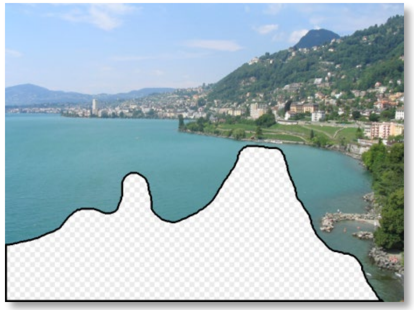


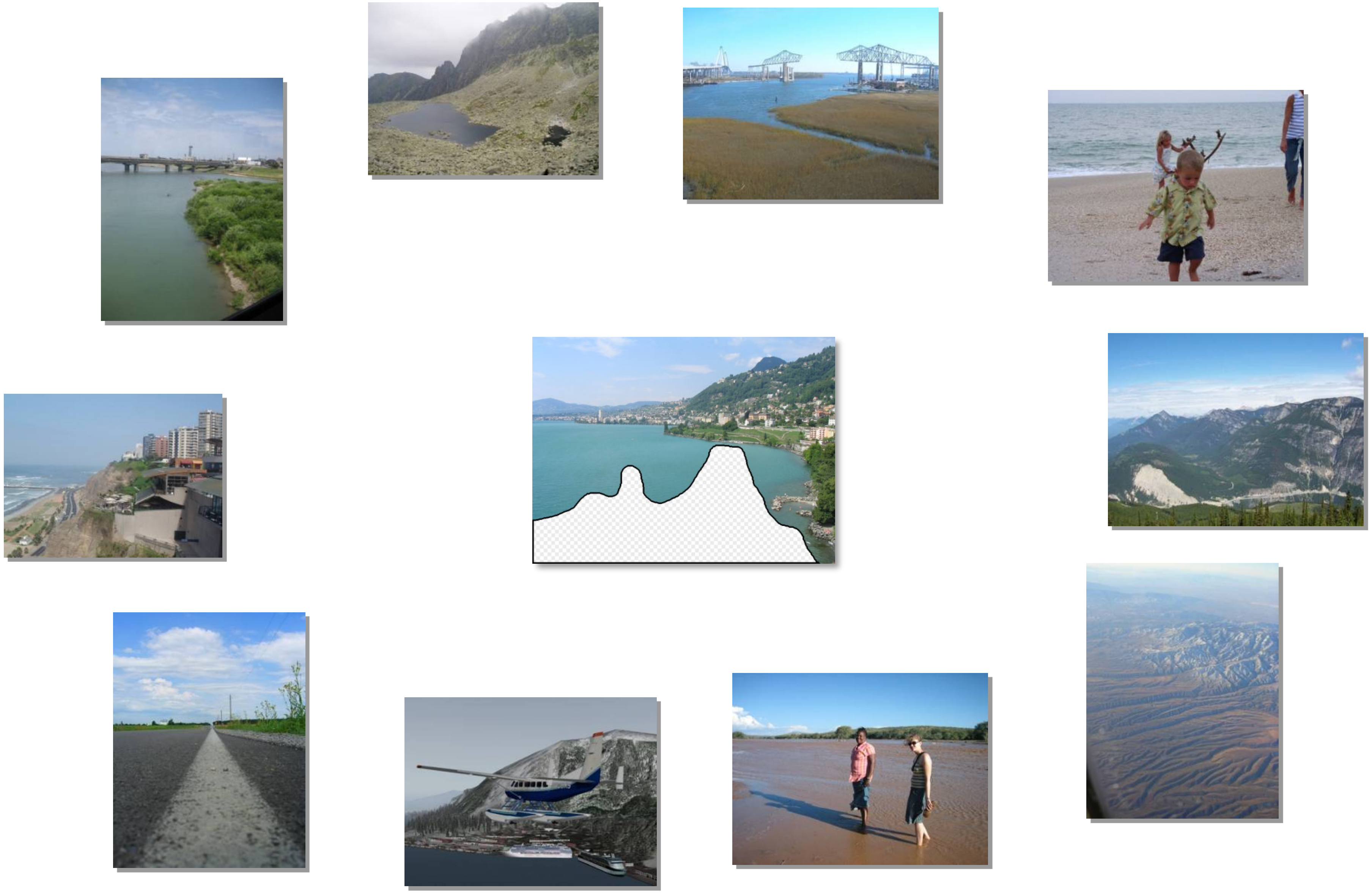
General Principal



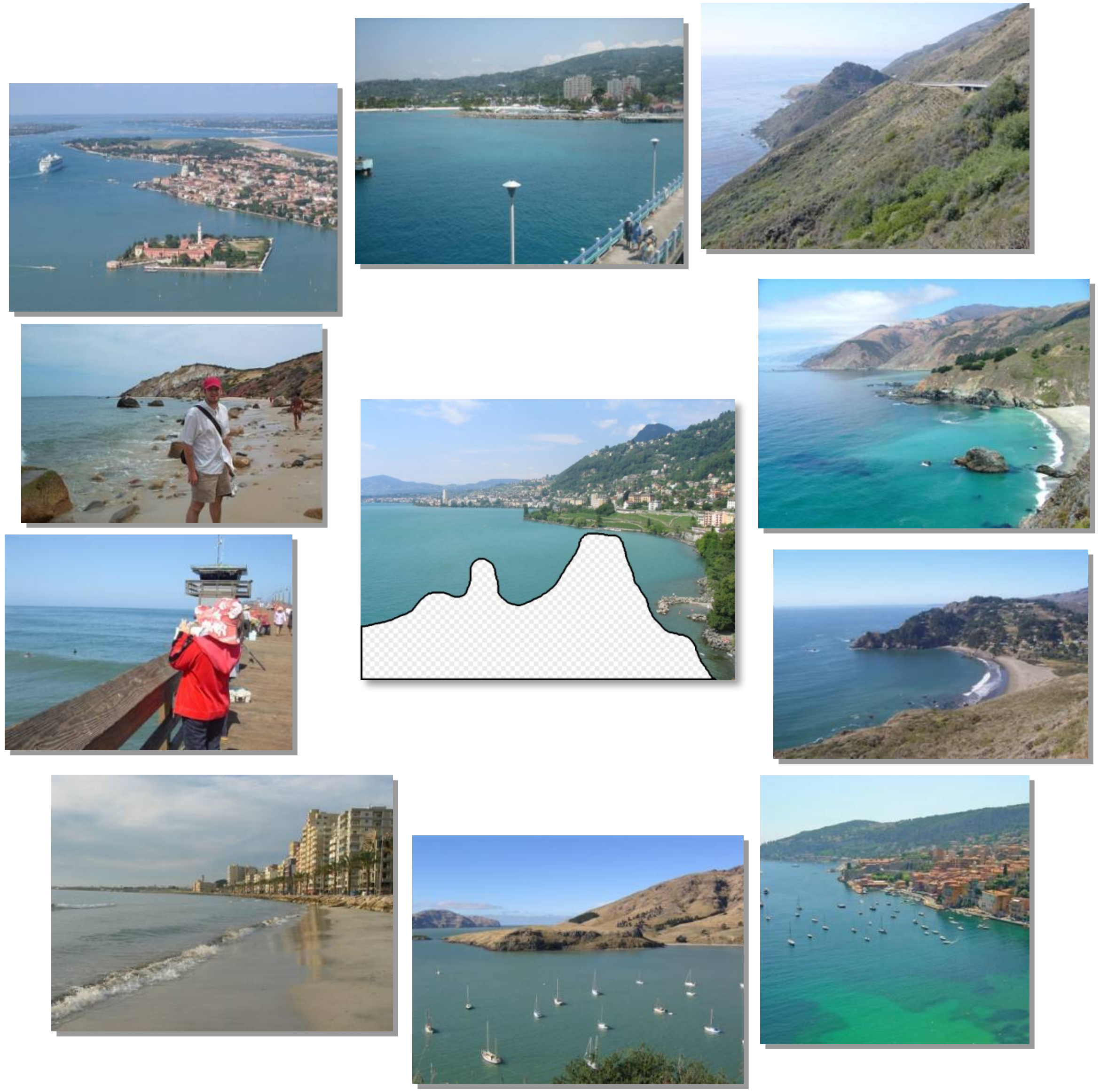
Trick: If you have enough images, the dataset will contain very similar images that you can find with simple matching methods.

How many images is enough?





Nearest neighbors from a collection of 20 thousand images



Nearest neighbors from a collection of 2 million images

Image Data on the Internet

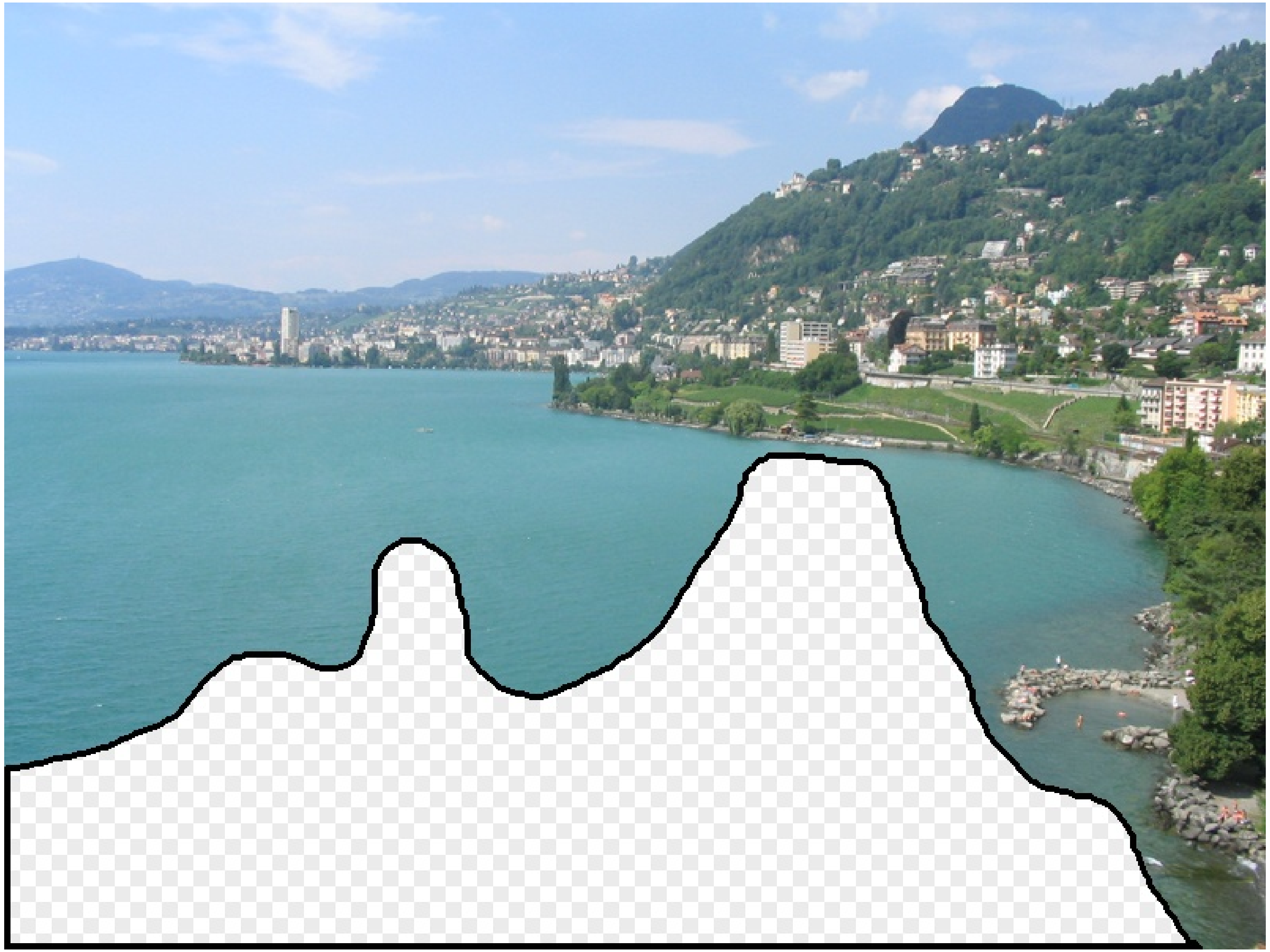
- Now: nobody counts anymore
- Facebook (2014)
  - 250 billion total, +350 million per day
- Facebook (2011)
  - 6 billion images per month
  - More than 100 petabytes of images/video
- Flickr (2010)
  - 5 billion photographs
  - 100+ million geotagged images
- Imageshack (as of 2009)
  - 20 billion
- Facebook (as of 2009)
  - 15 billion

Image completion: how it works

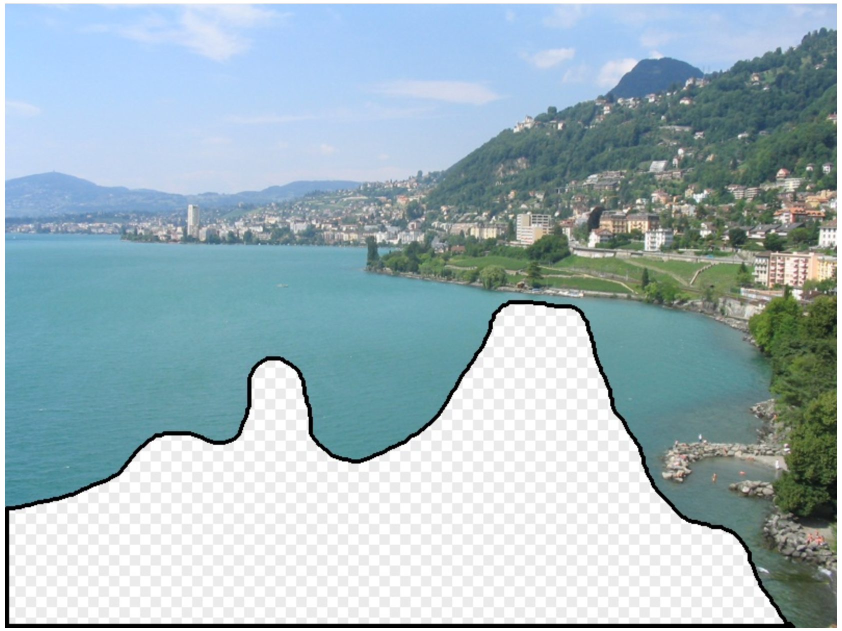
[Hays and Efros. Scene Completion Using Millions of Photographs. SIGGRAPH 2007 and CACM October 2008.]

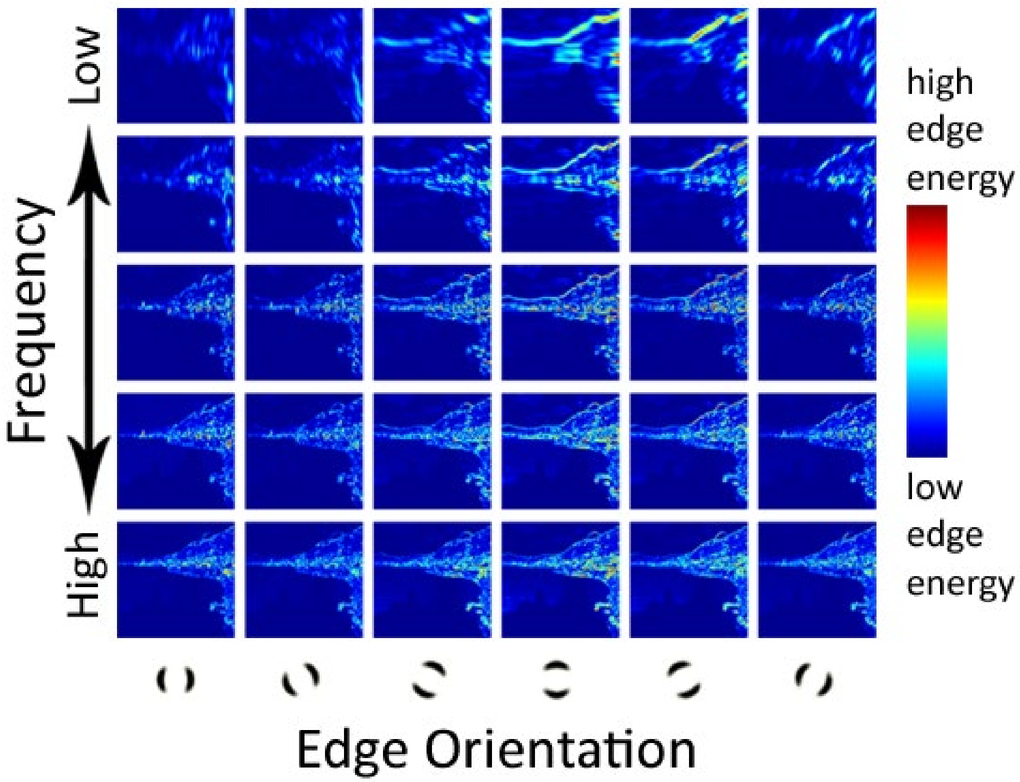
The Algorithm

Scene Matching

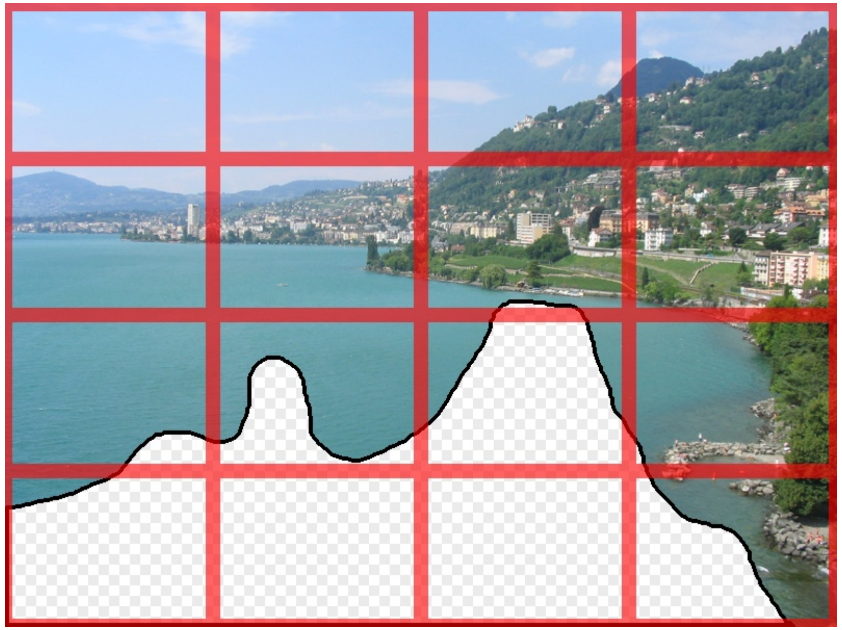


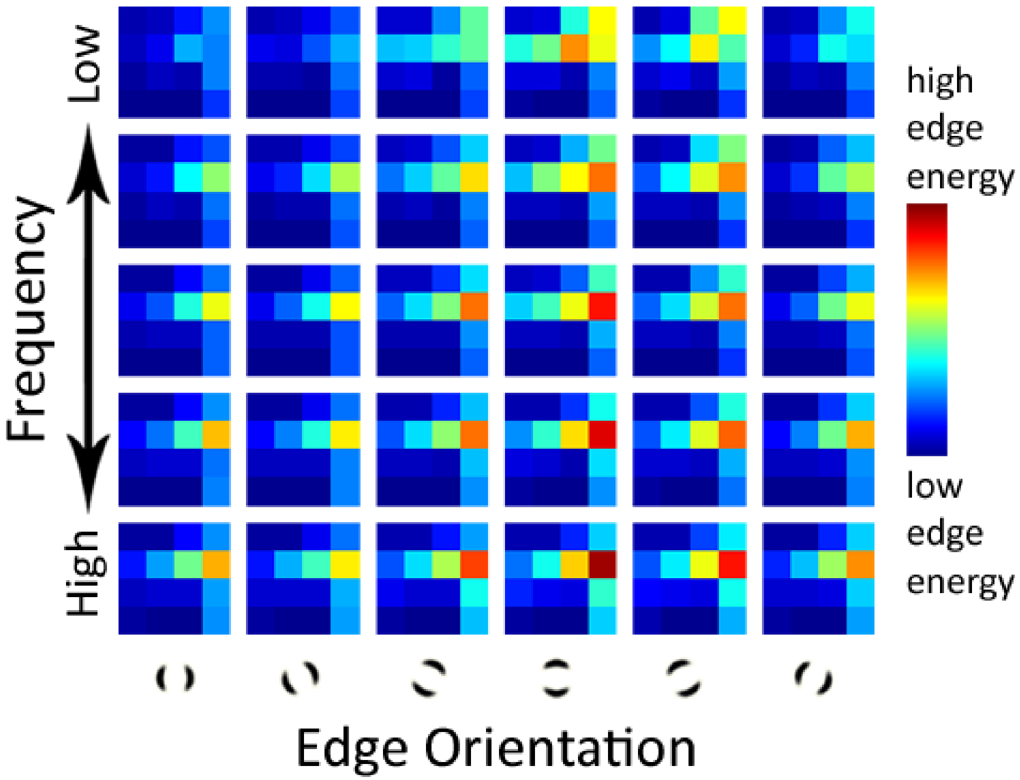
Scene Descriptor





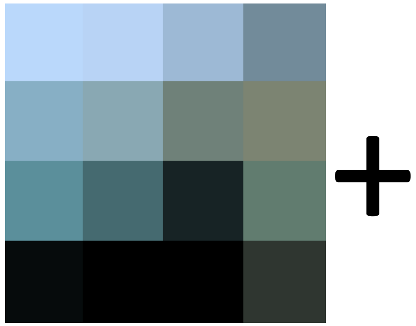
Scene Descriptor

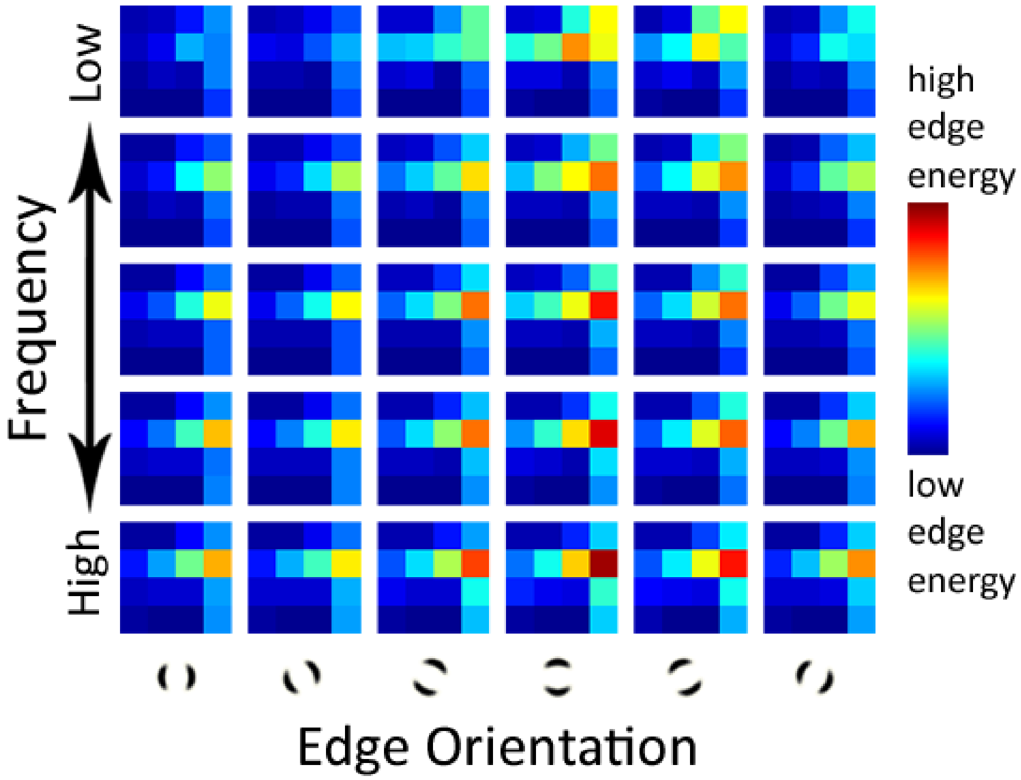




Scene Gist Descriptor
(Oliva and Torralba 2001)

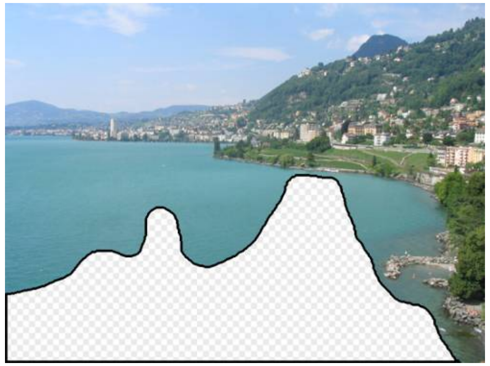
Scene Descriptor

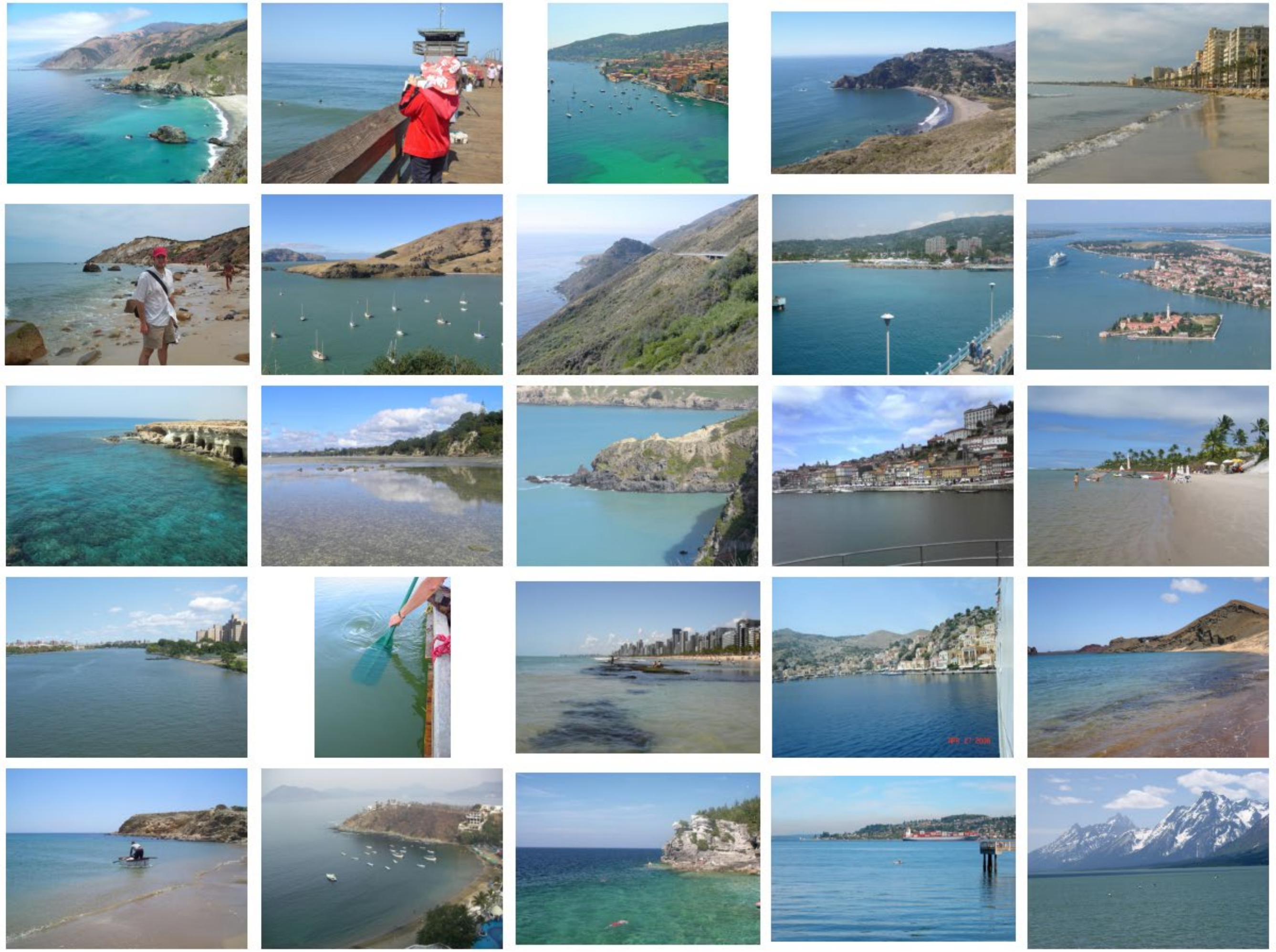




Scene Gist Descriptor
(Oliva and Torralba 2001)

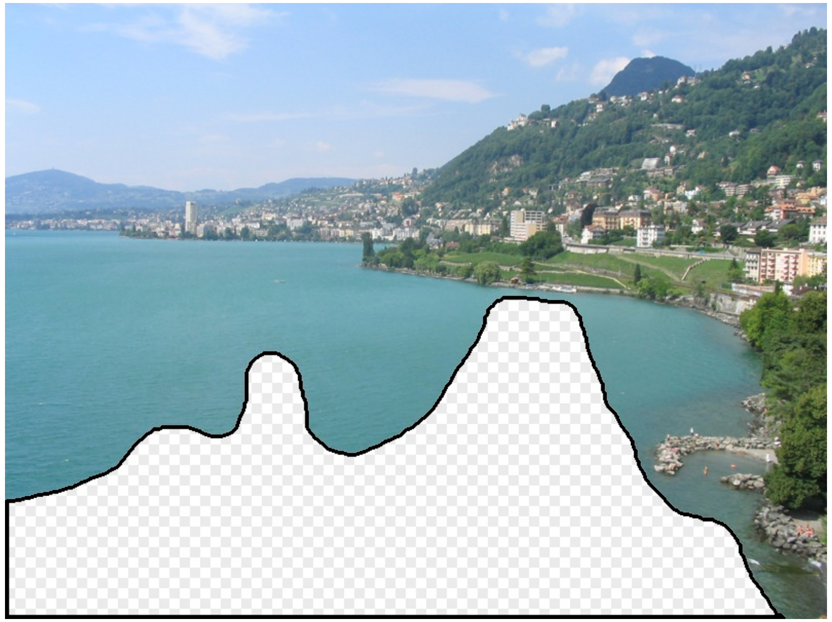

2 Million Flickr Images





… 200 total

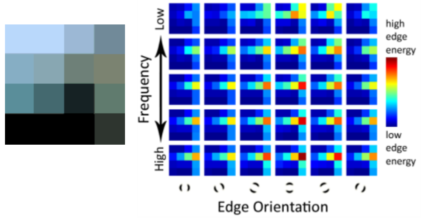
Context Matching



Graph cut + Poisson blending

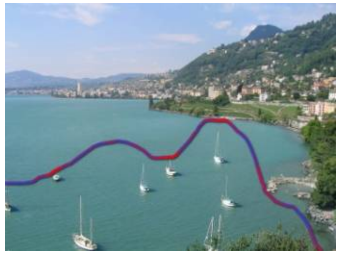
Result Ranking

We assign each of the 200 results a score which is the sum of:

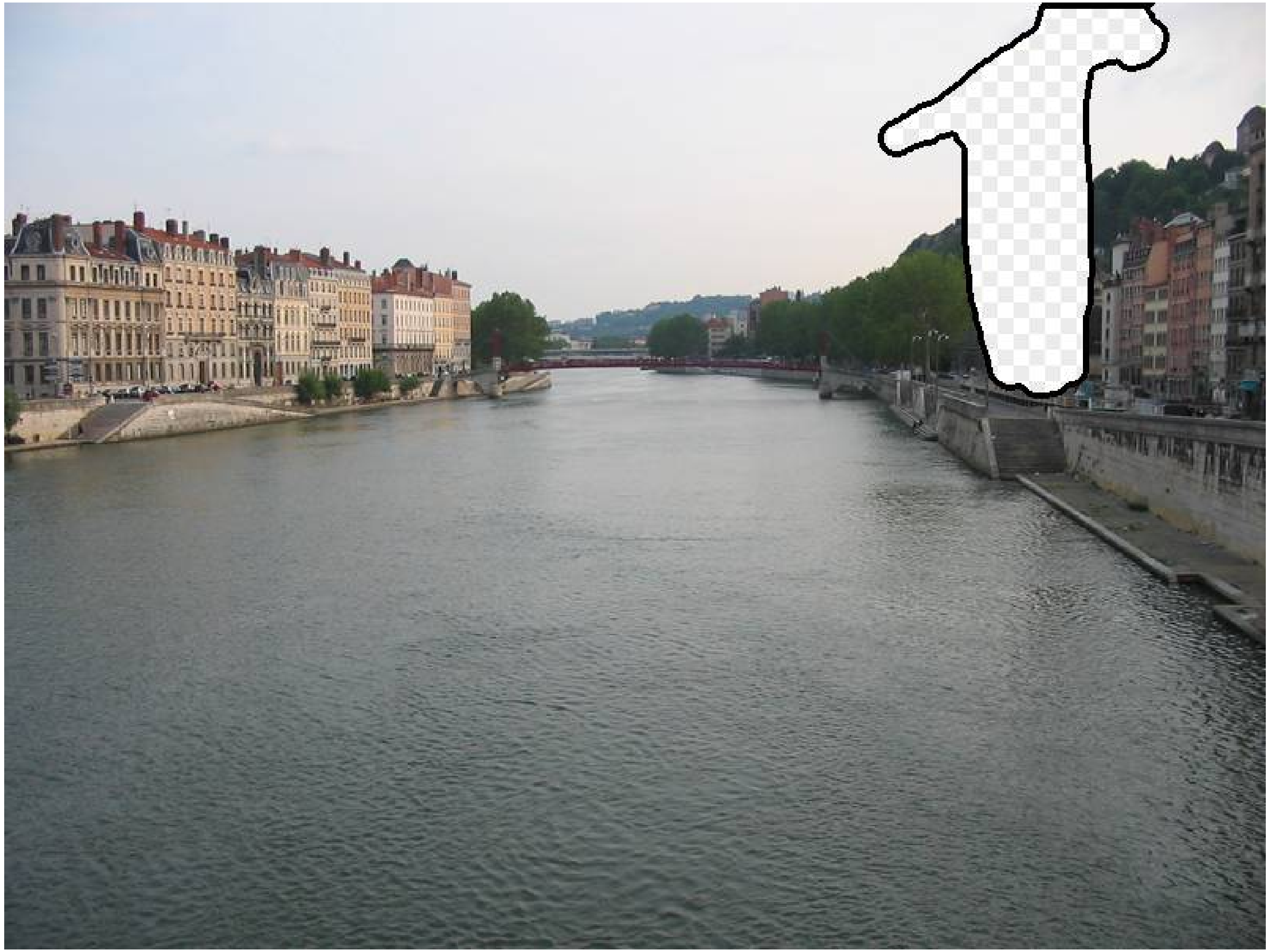


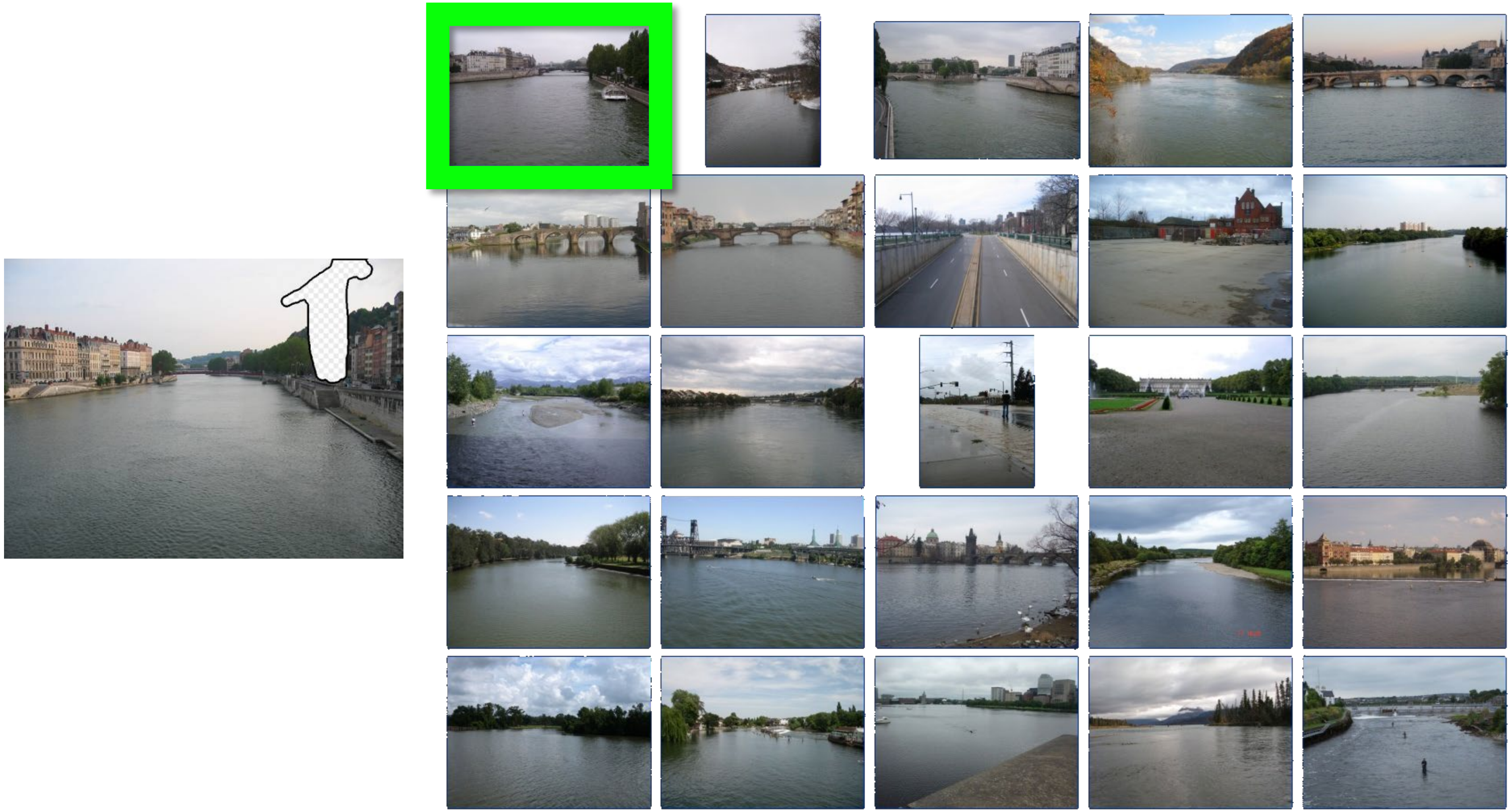
The scene matching distance

The context matching distance
(color + texture)

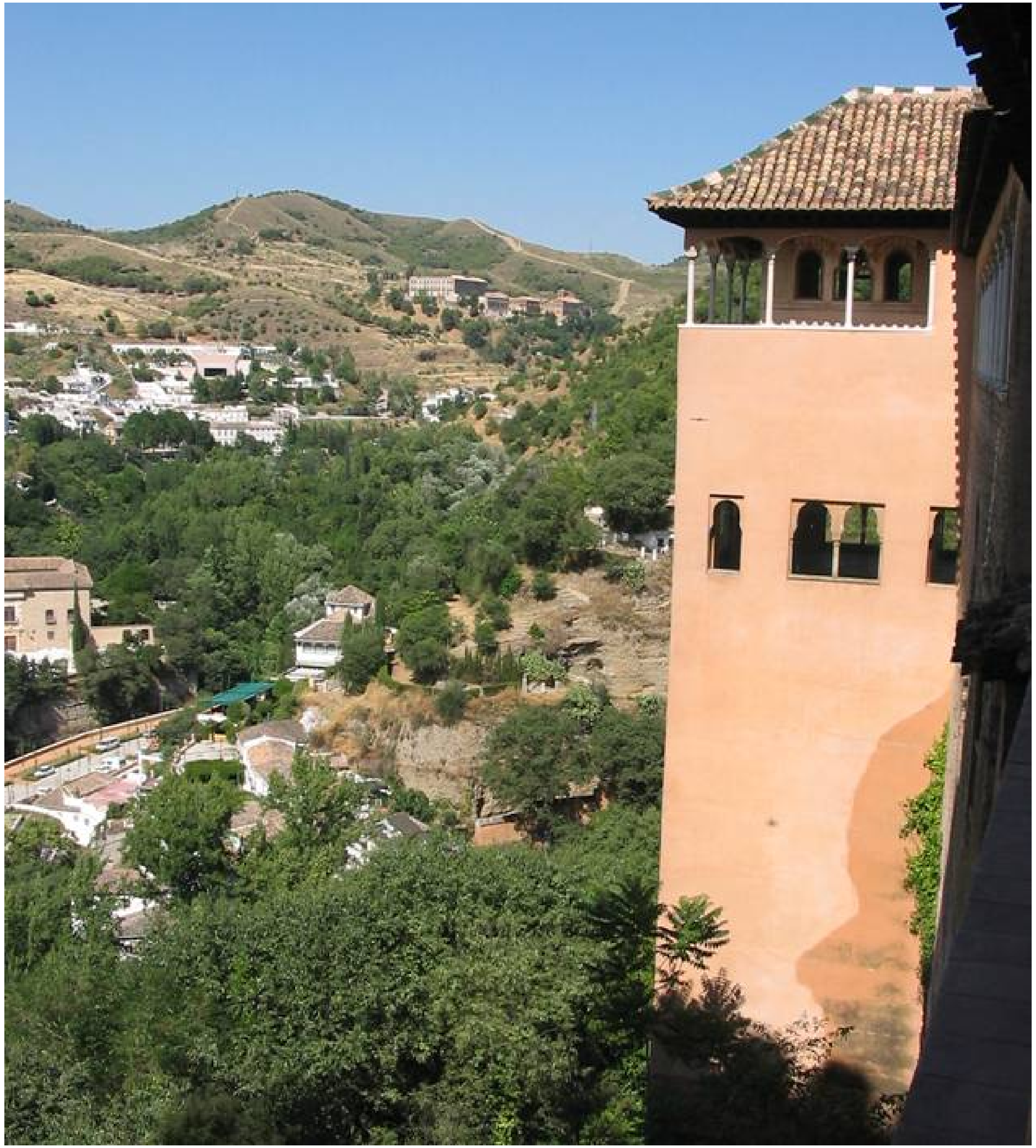


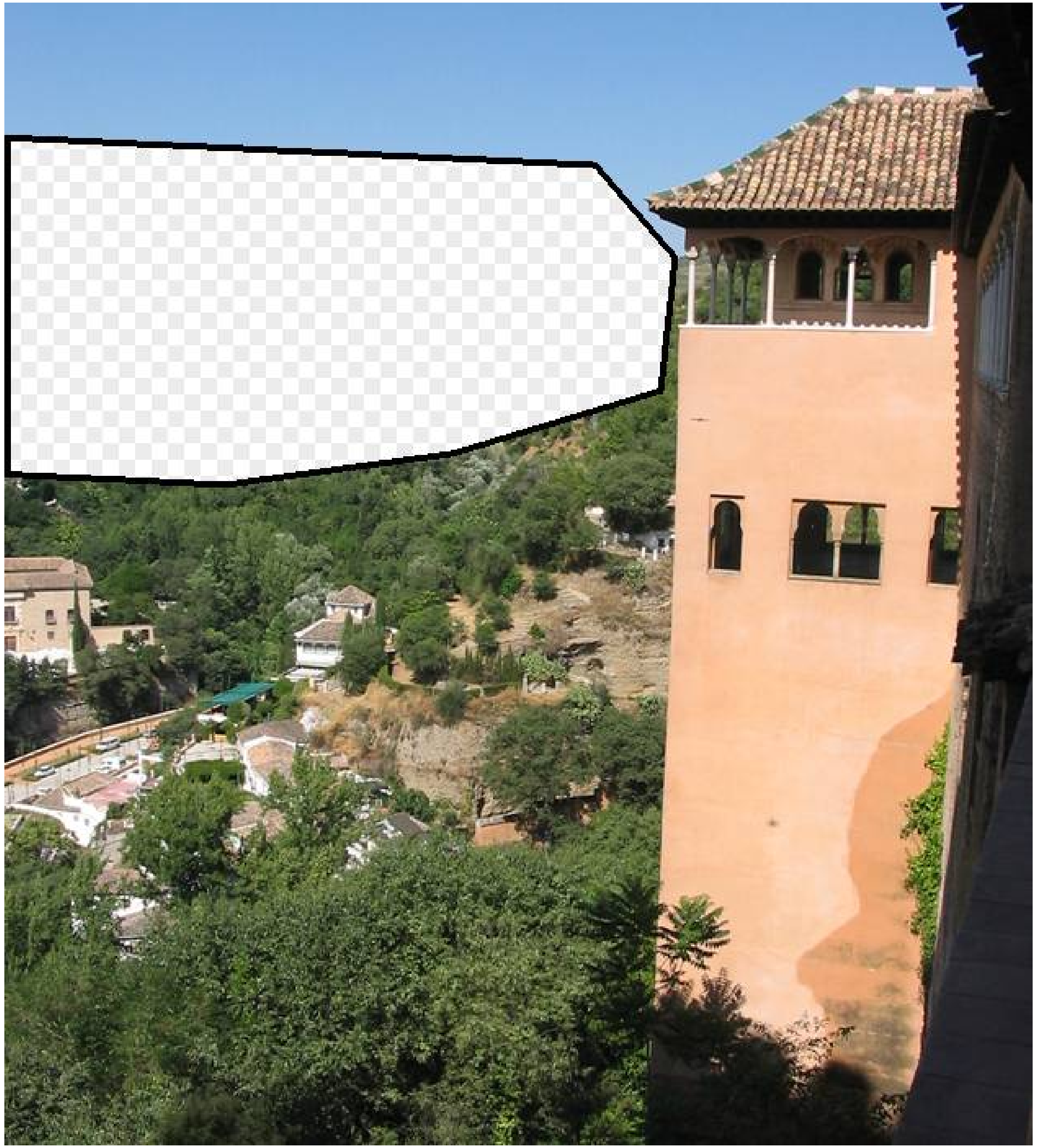
The graph cut cost

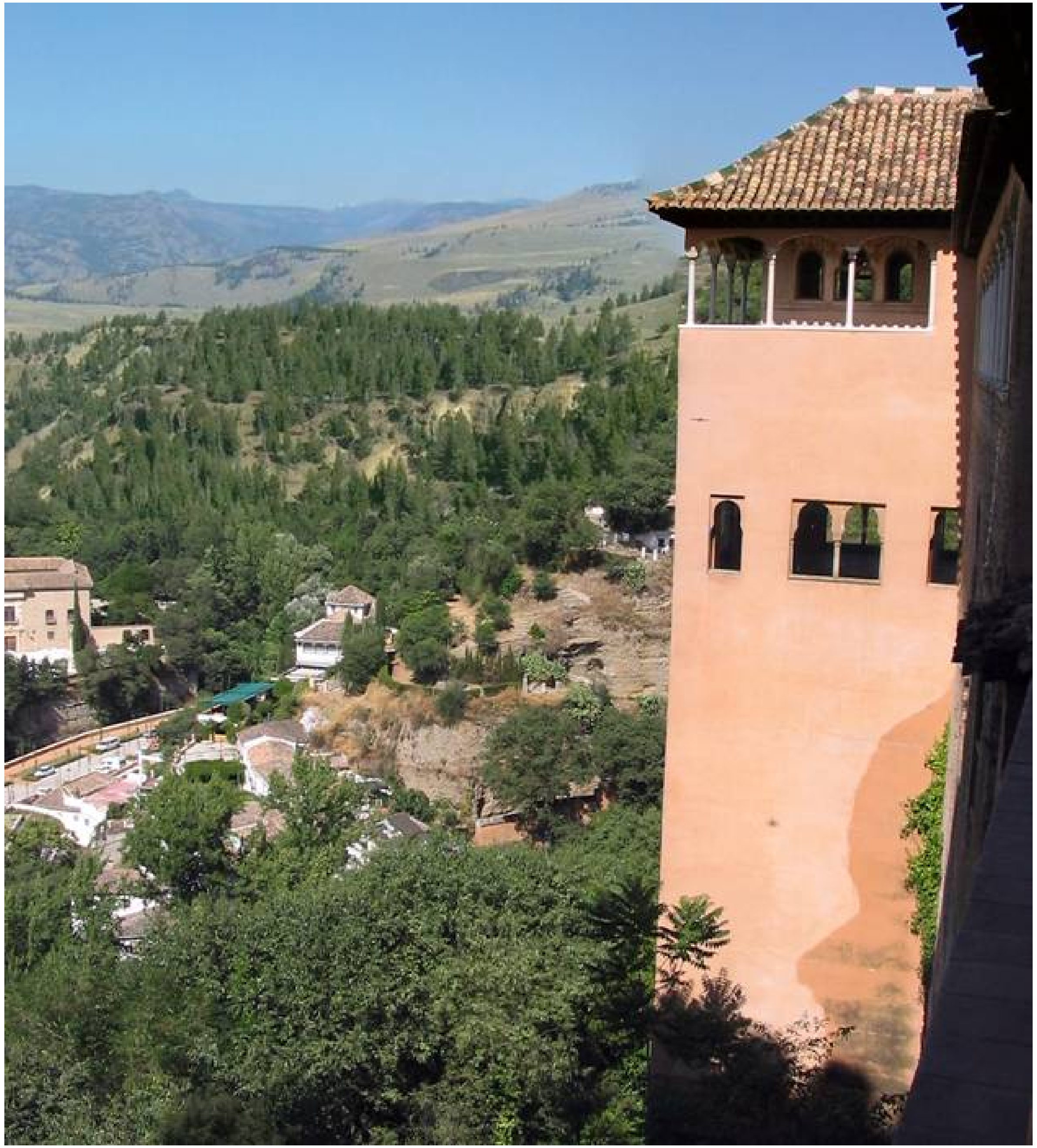




… 200 scene matches







Which is the original?


5407 BWL

Diffusion Result

Efros and Leung result

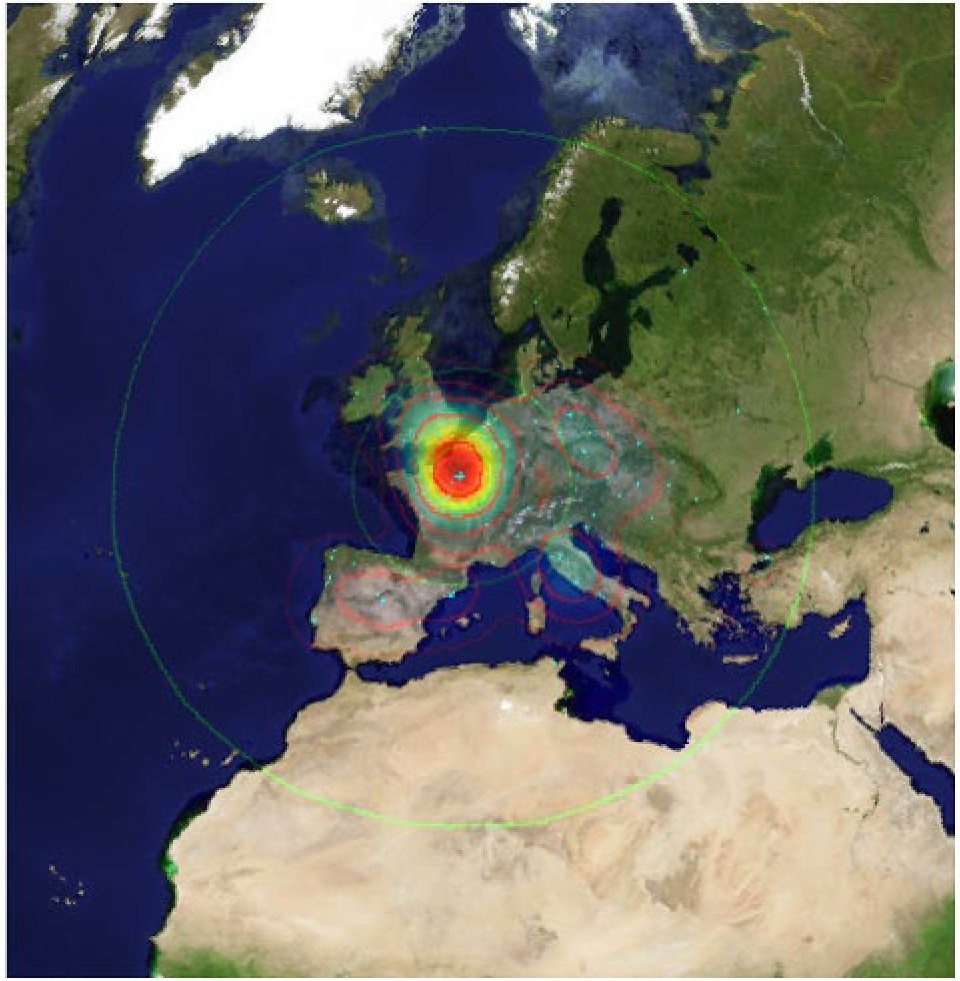
Scene Completion Result

im2gps (Hays & Efros, CVPR 2008)

6 million geo-tagged Flickr images

How much can an image tell about its geographic location?

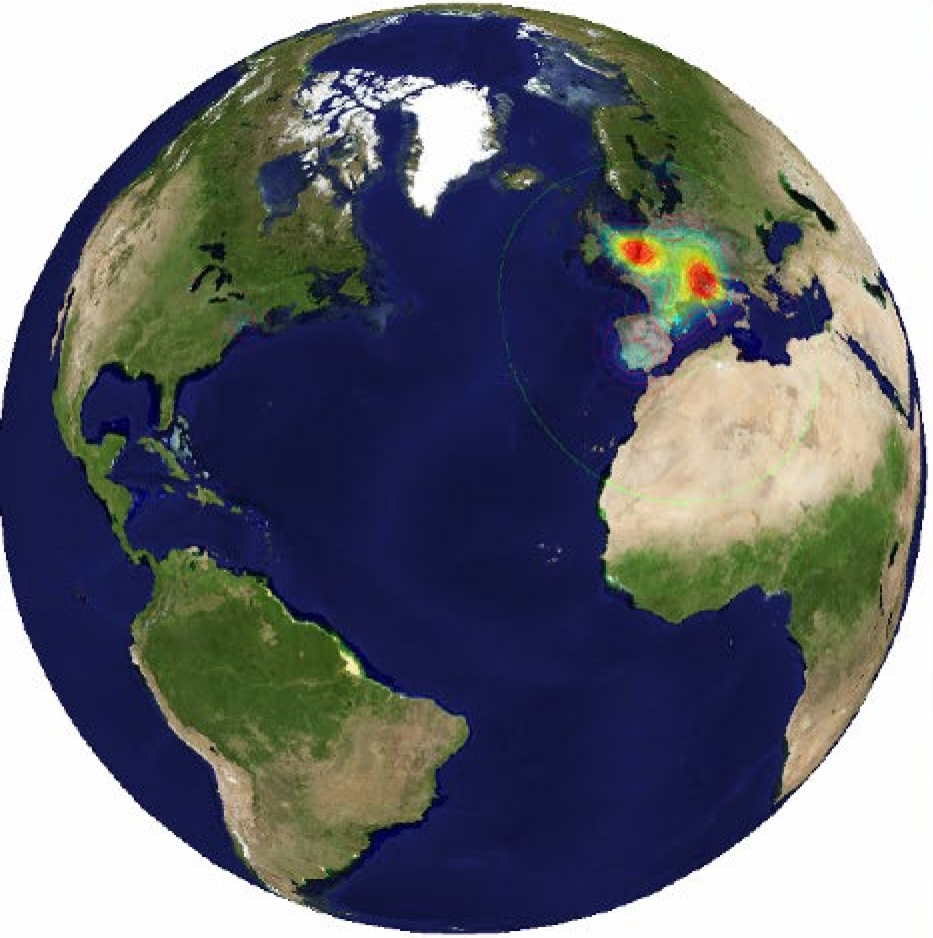

Paris
Paris
Paris
Paris
Paris
Paris
Paris
Madrid
Rome
Paris
Cuba
Paris
Paris
Poland
Paris
Paris

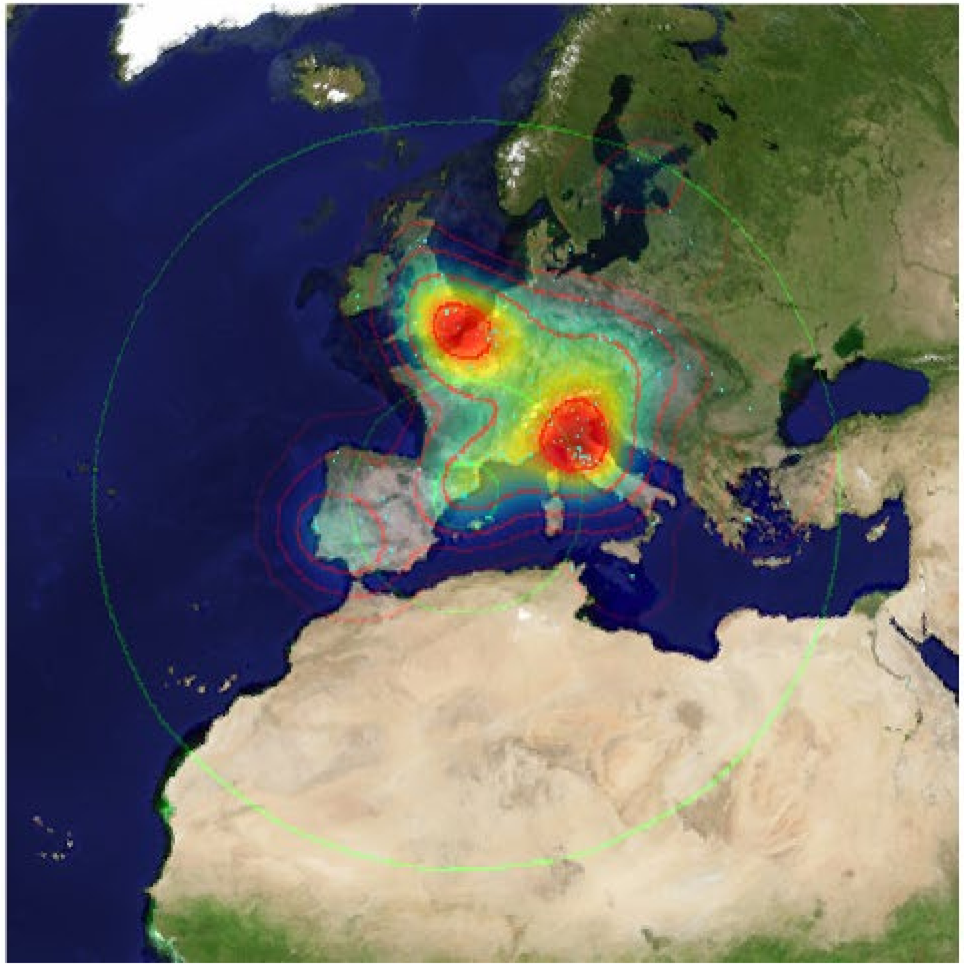


Im2gps

Example Scene Matches

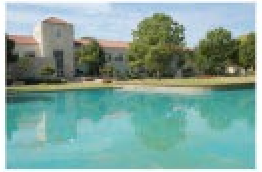
Voting Scheme





im2gps

Philippines



Houston

Thailand

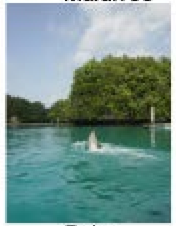
Houston

Maldives

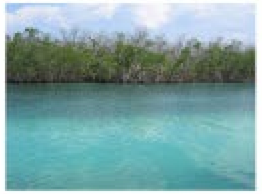
Philippines

NewZealand

Bermuda



Palau



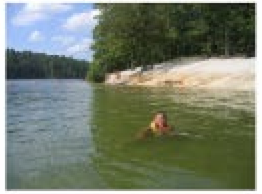
Mexico2

Brazil

Mendoza

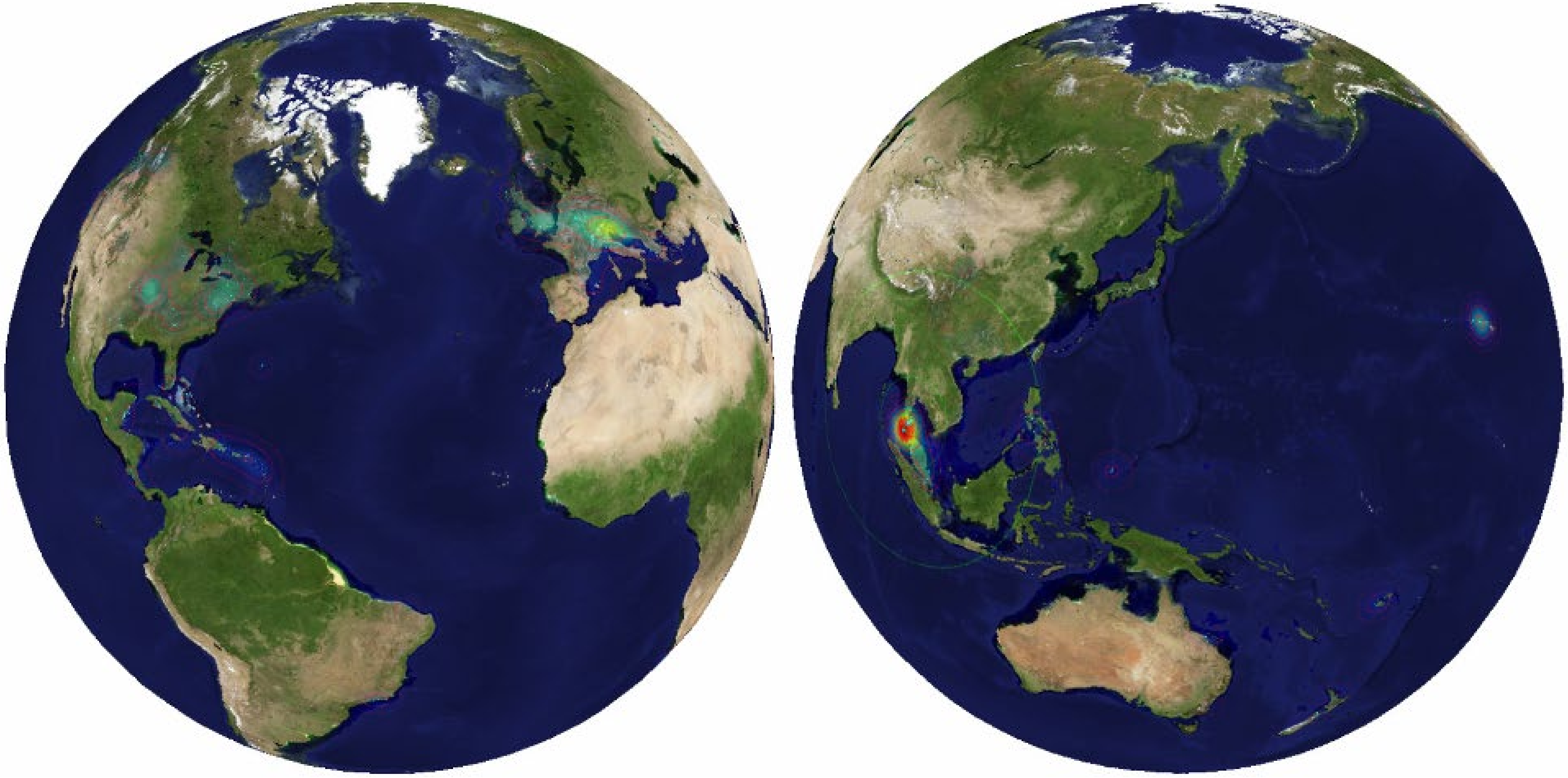
Brazil

Thailand

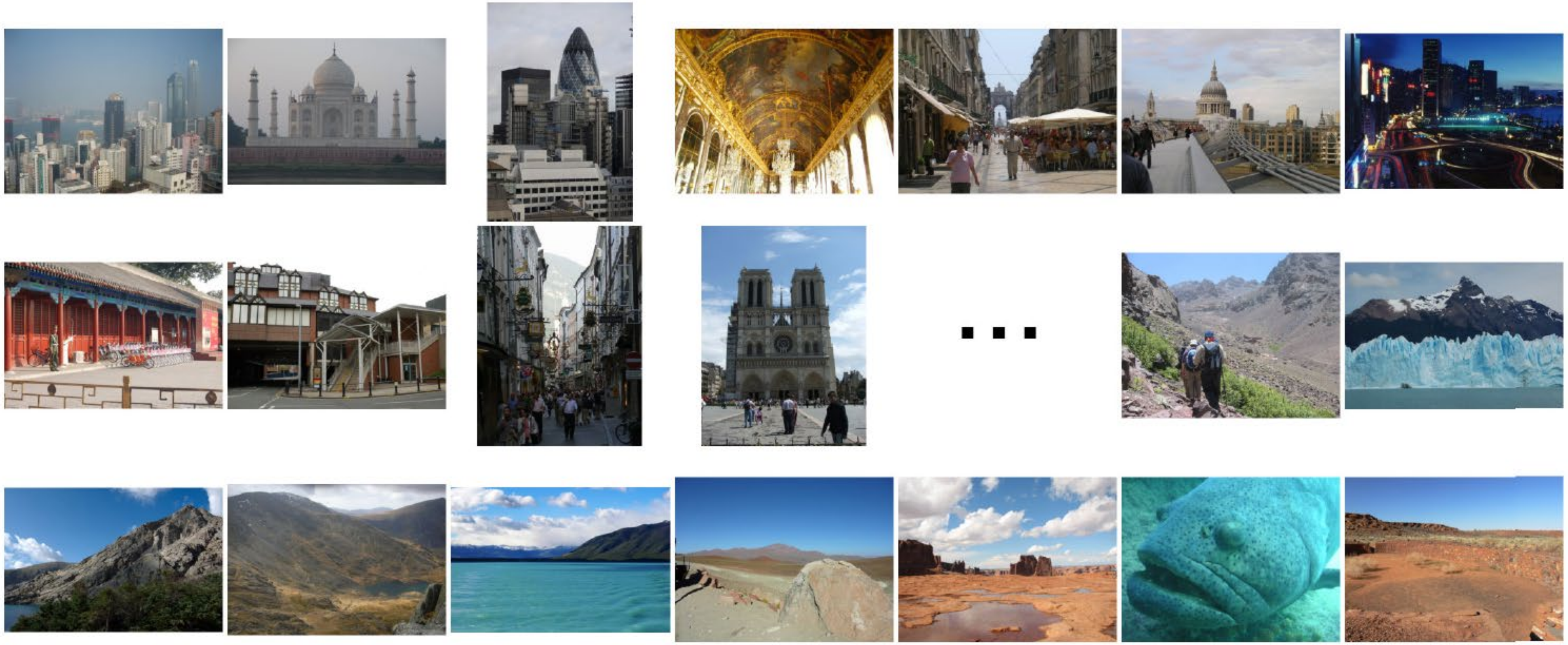


Arkansas

Hawaii



Population density ranking



Where is This?

[Olga Vesselova, Vangelis Kalogerakis, Aaron Hertzmann, James Hays, Alexei A. Efros. Image Sequence Geolocation. ICCV’09]

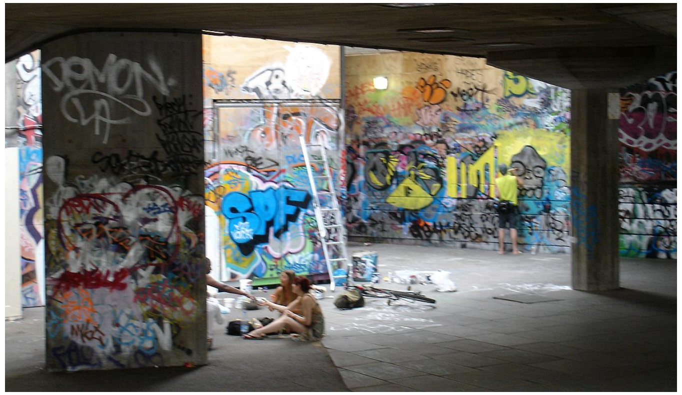
Where is This?

Where are These?

15:14,
June 18th, 2006

16:31,
June 18th, 2006

Where are These?



15:14,
June 18th, 2006

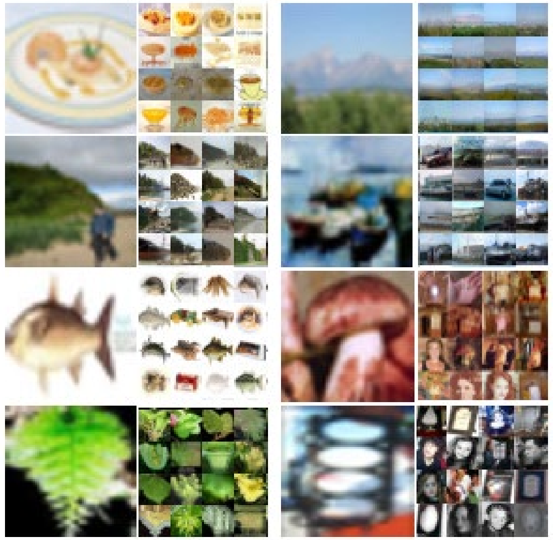
16:31,
June 18th, 2006

17:24,
June 19th, 2006

Results

- im2gps – 10% (geo-loc within 400 km)
- temporal im2gps – 56%

Tiny Images

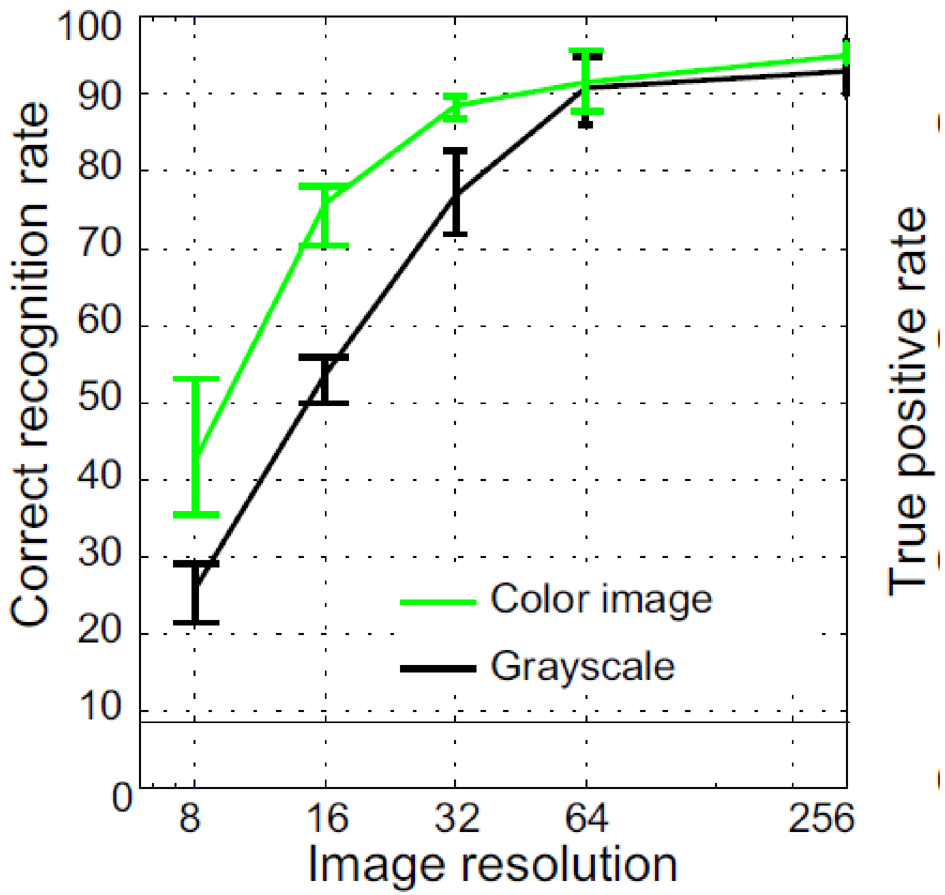


80 million tiny images: a large dataset for non-parametric object and scene recognition
Antonio Torralba, Rob Fergus and William T. Freeman. PAMI 2008.

http://groups.csail.mit.edu/vision/TinyImages/

c) Segmentation of 32x32 images

Human Scene Recognition



a) Scene recognition

Powers of 10

Number of images on my hard drive: 10^4

Number of images seen during my first 10 years: 10^8

(3 images/second * 60 * 60 * 16 * 365 * 10 = 630720000)

Number of images seen by all humanity: 10^{20}

106,456,367,669 humans[1] * 60 years * 3 images/second * 60 * 60 * 16 * 365 =

1 from http://www.prb.org/Articles/2002/HowManyPeopleHaveEverLivedonEarth.aspx

Number of photons in the universe: 10^{88}

Number of all 32x32 images: 10^{7373}

$256^{32*32*3} \sim 10^{7373}$

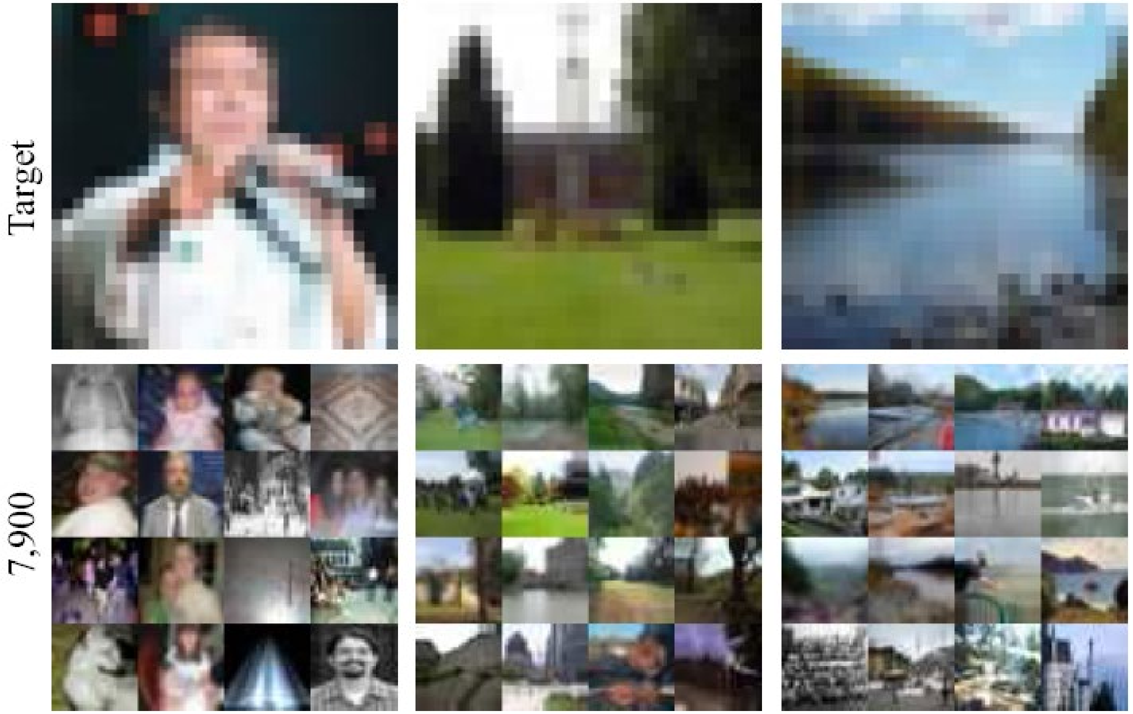
Scenes are unique

But not all scenes are so original

Lots

Of

Images

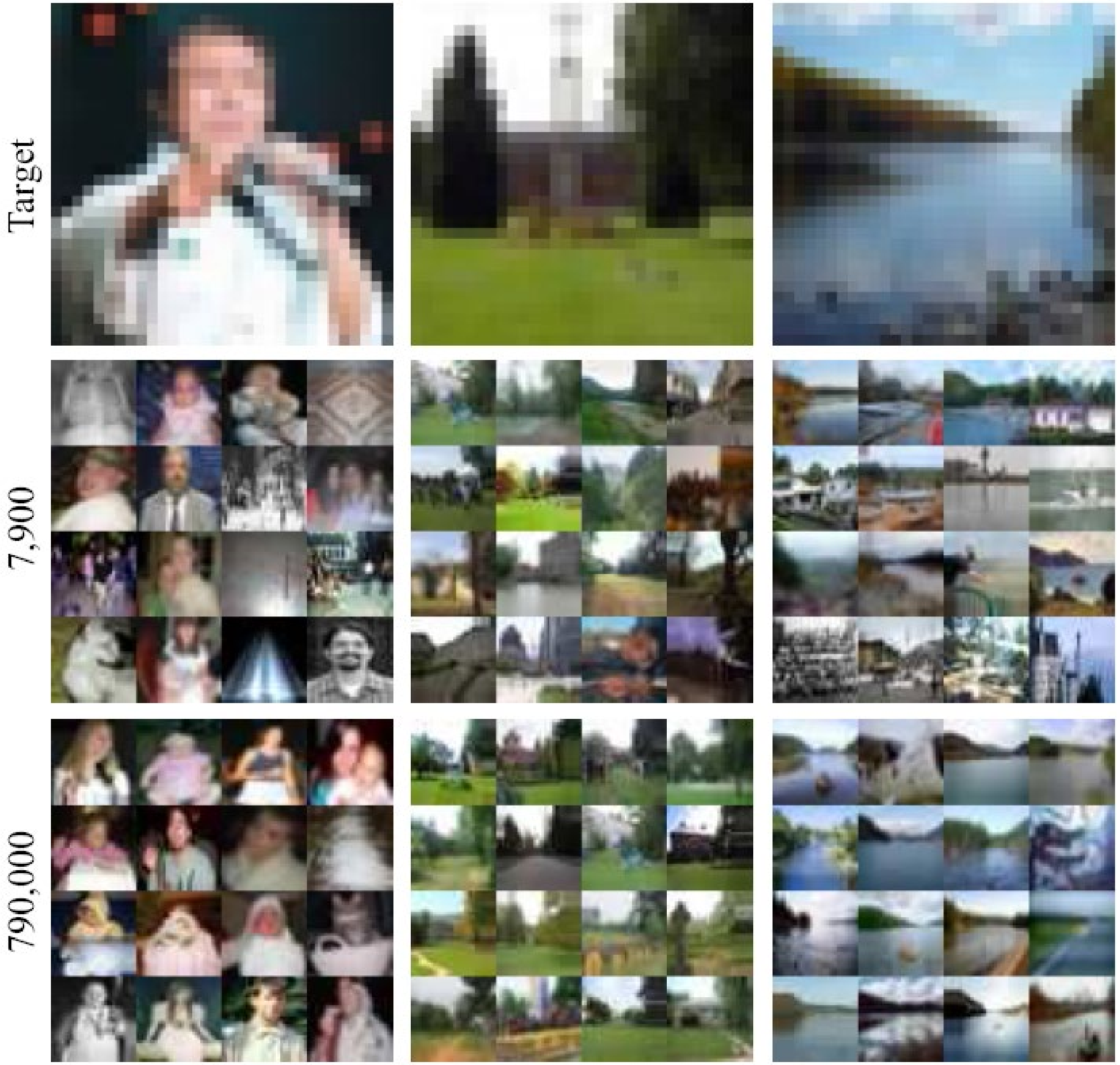


A. Torralba, R. Fergus, W.T.Freeman. PAMI 2008

Lots

Of

Images

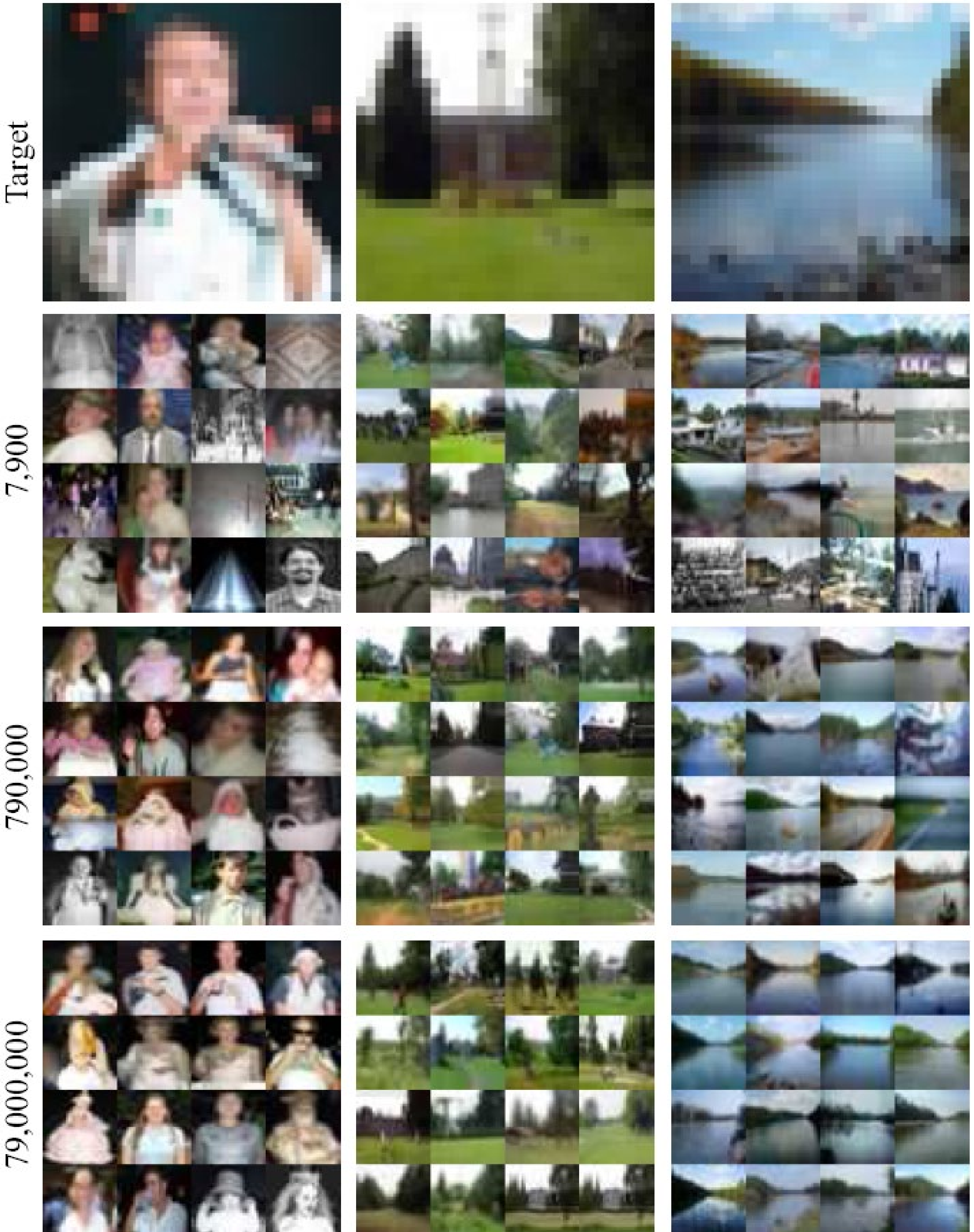


A. Torralba, R. Fergus, W.T.Freeman. PAMI 2008

Lots

Of

Images



Automatic Colorization

Input

Color Transfer

Color Transfer

Matches (gray)

Matches (w/ color)

Avg Color of Match

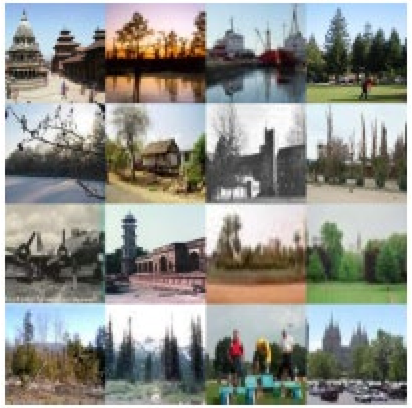
Automatic Colorization

Input

Color Transfer

Color Transfer

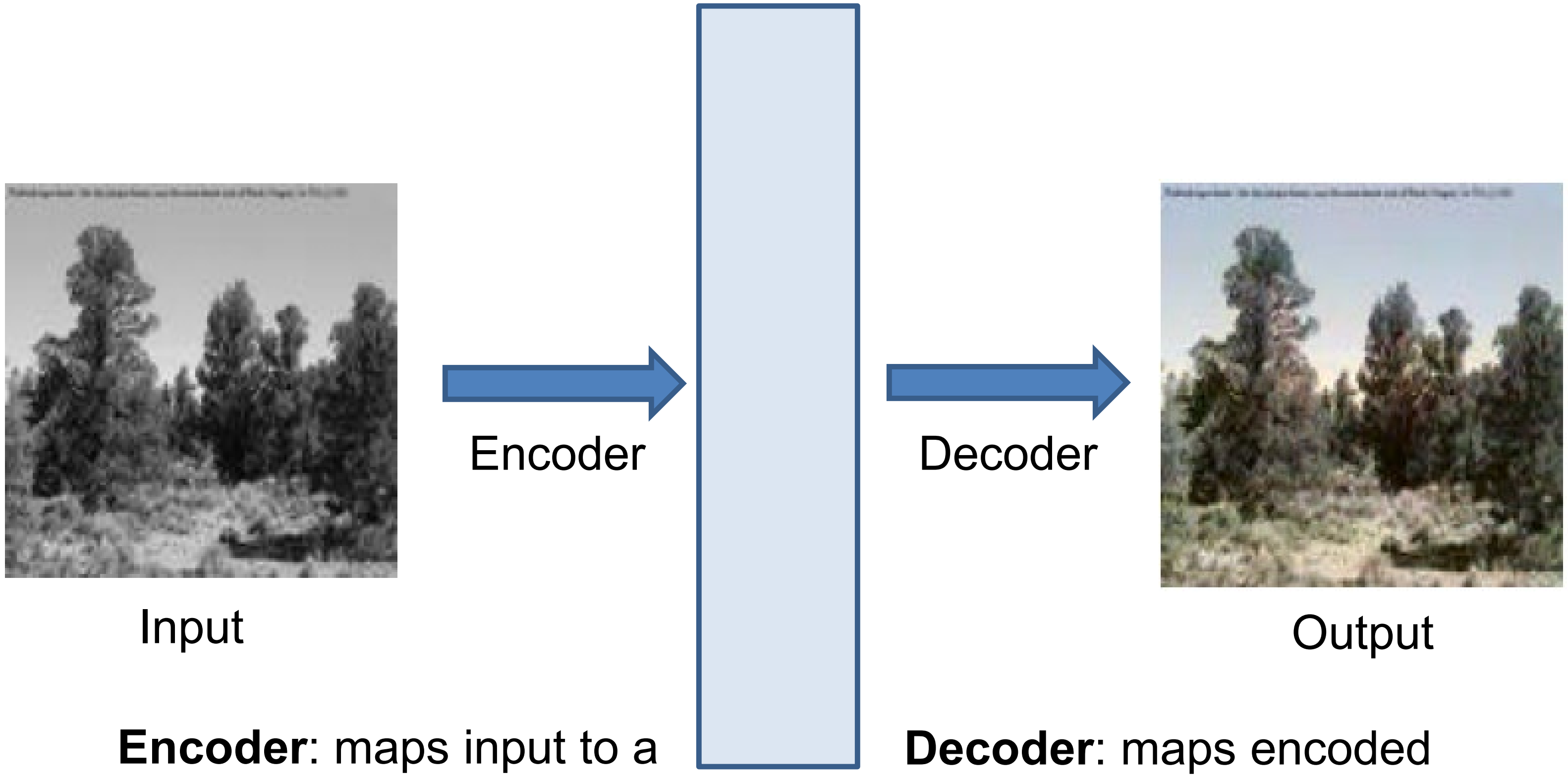
Matches (gray)



Matches (w/ color)

Avg Color of Match

Encoder – Decoder view



Encoder: maps input to a new representation

Images with similar encodings should have similar outputs

Decoder: maps encoded representation to output

Encoder – Decoder: simple example

Input

Encoder → [ ] → Decoder

Output

Encoder: shrink image to 32x32 → 1024x1 intensity vector

Decoder: retrieve nearest neighbor in database and copy color channels

Encoder – Decoder: deep network

Learn parameters of convolutional networks so that encoding / decoding satisfies some training objective for training samples

Input

Encoder

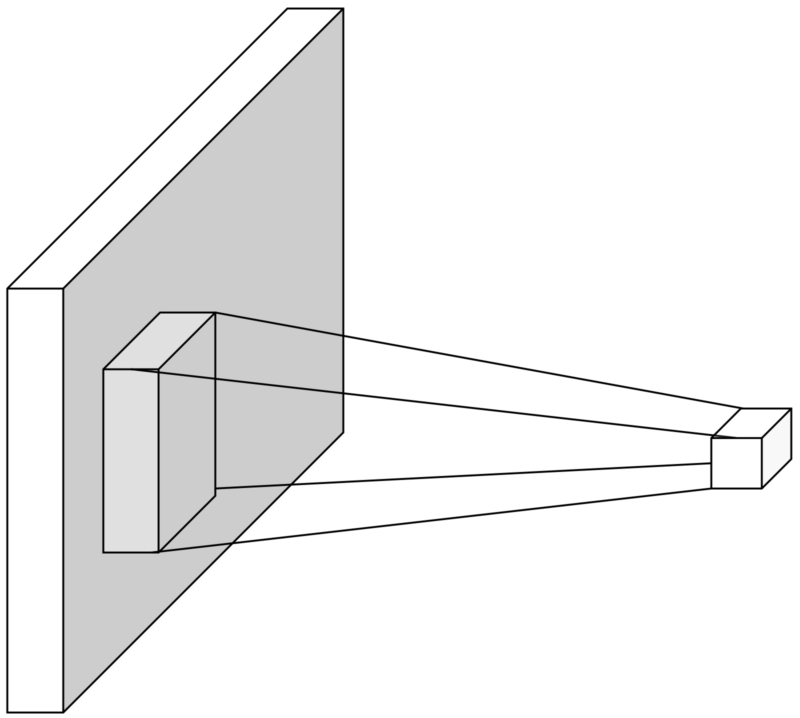
Decoder

Output

Encoder: learned convolutional network

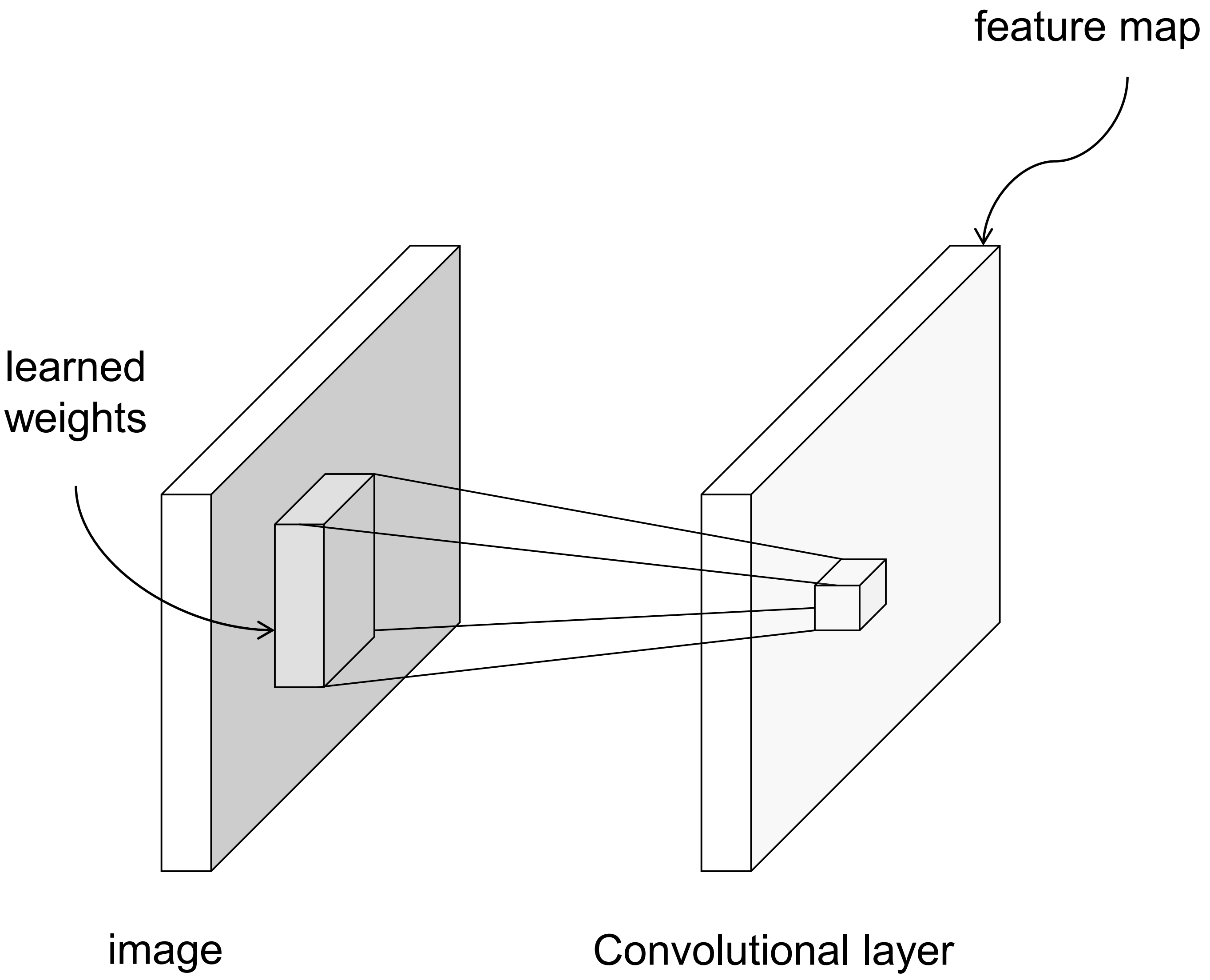
Decoder: learn convolutional network

Convolutional network



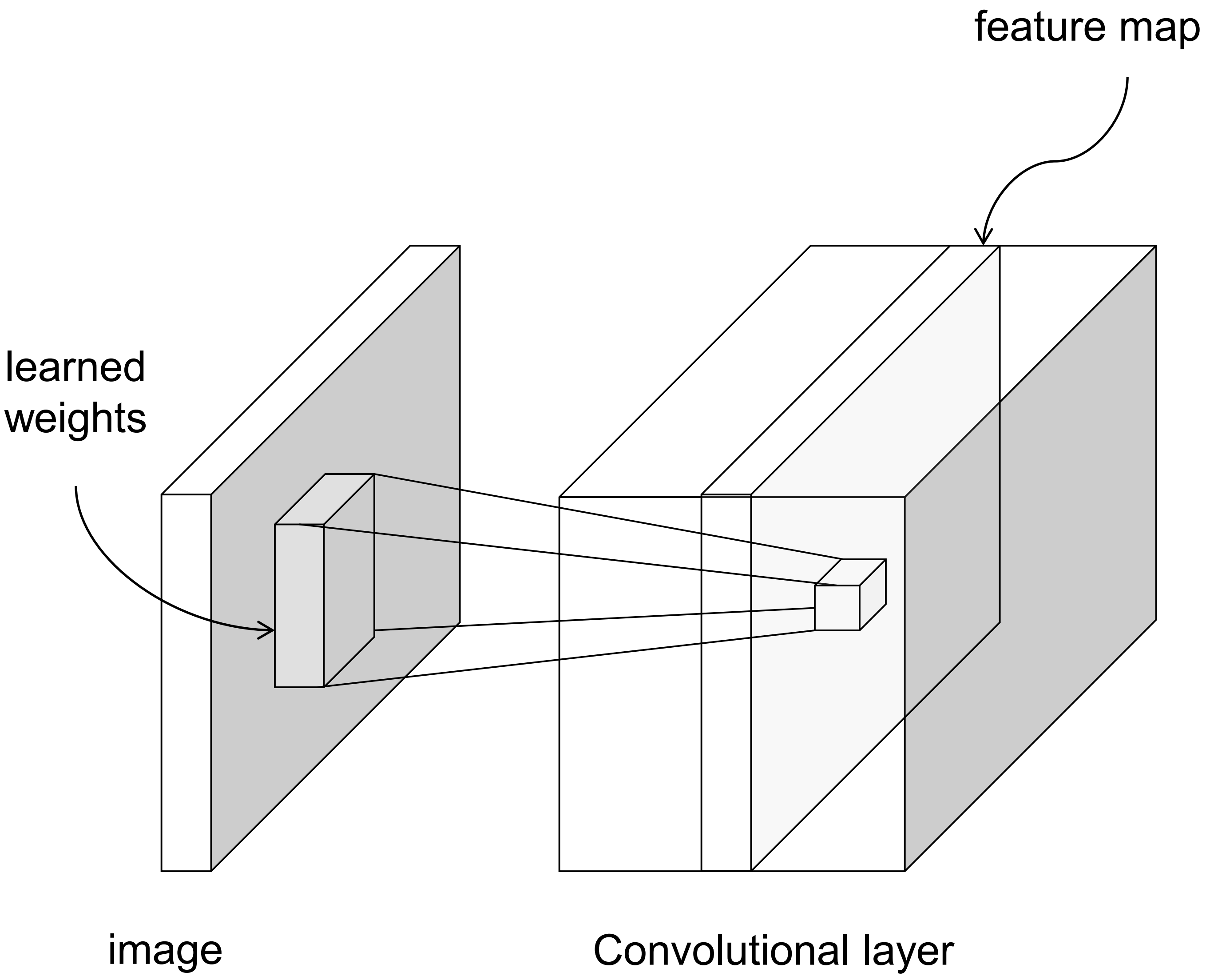
Slide: Lazebnik

Convolutional network



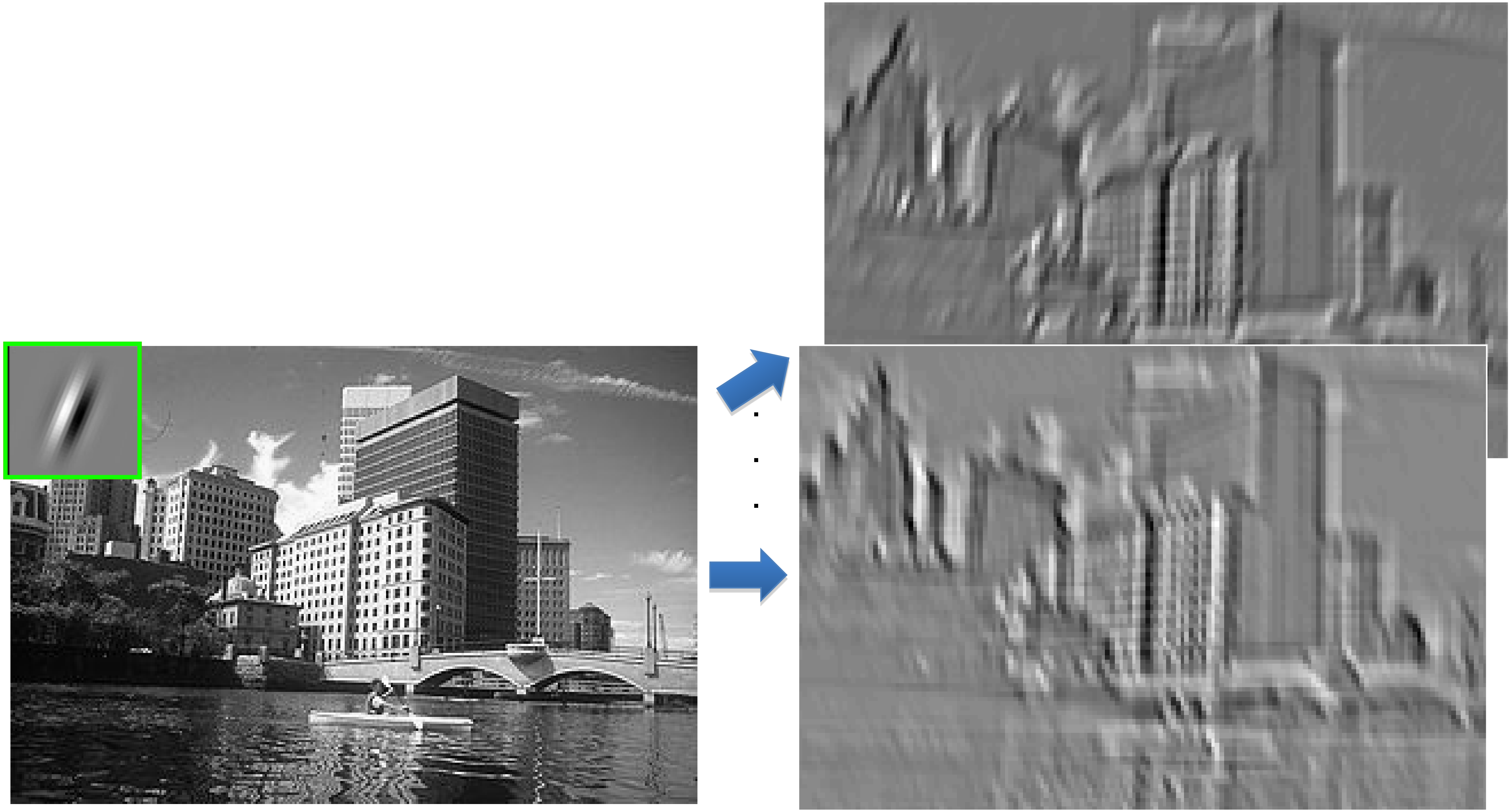
Slide: Lazebnik

Convolutional network



Slide: Lazebnik

Convolution as feature extraction

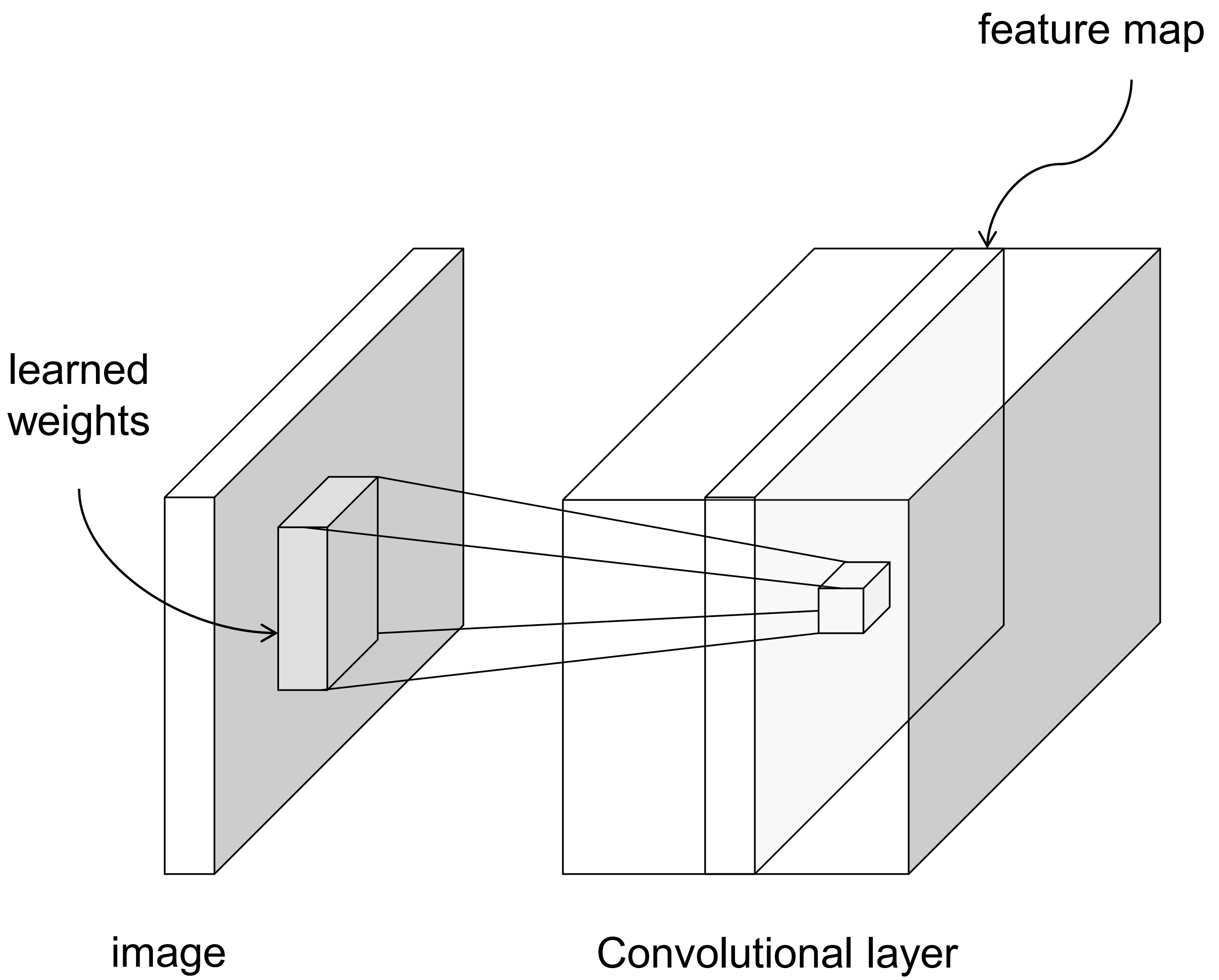


Input

Feature Map

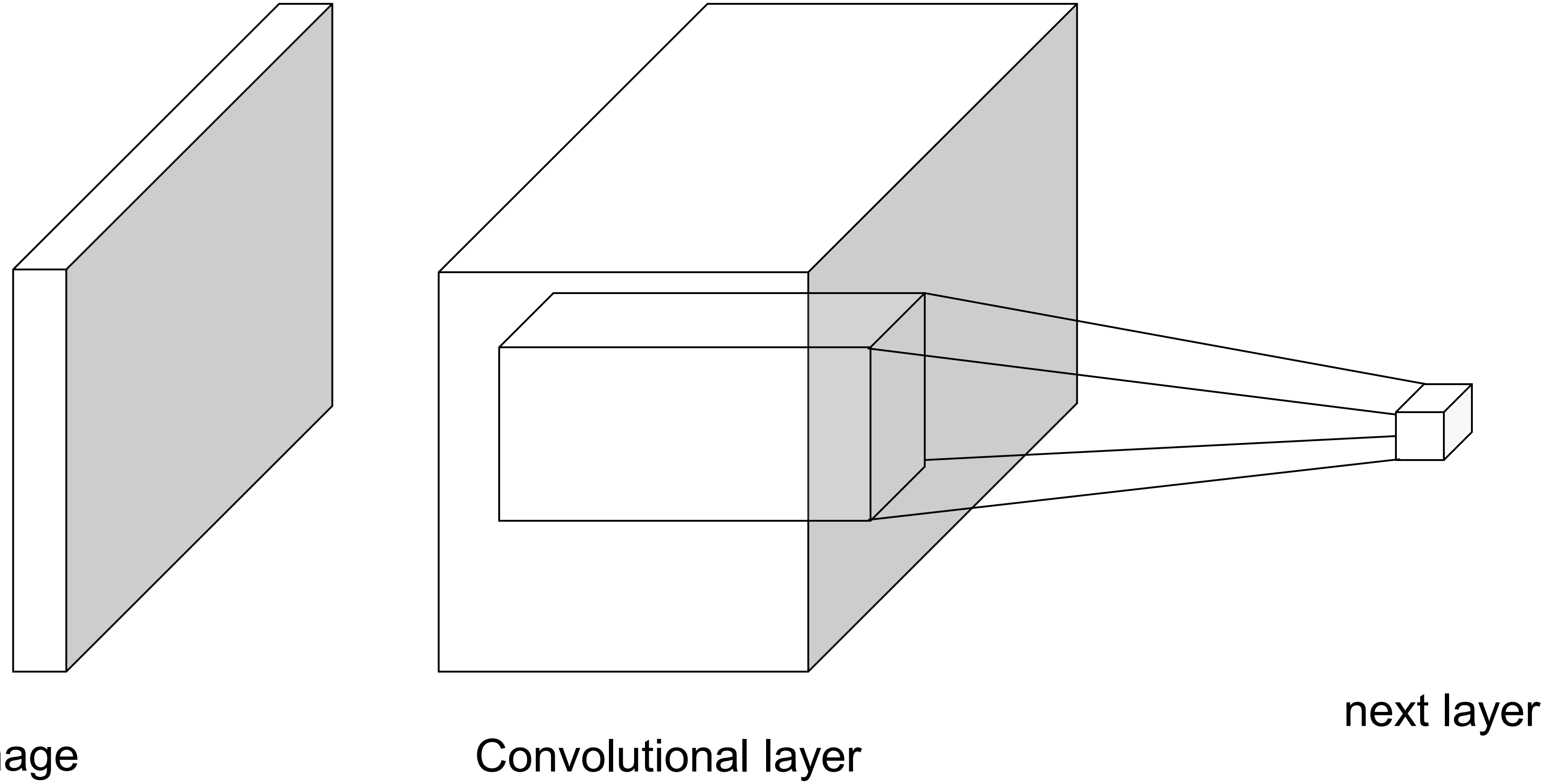
Slide: Lazebnik

From fully connected to convolutional networks



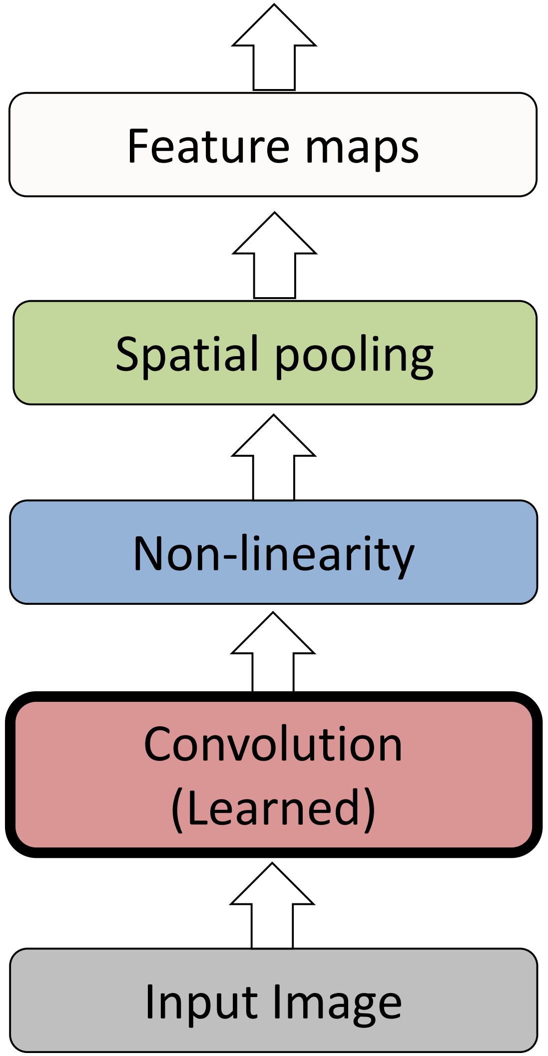
Slide: Lazebnik

From fully connected to convolutional networks

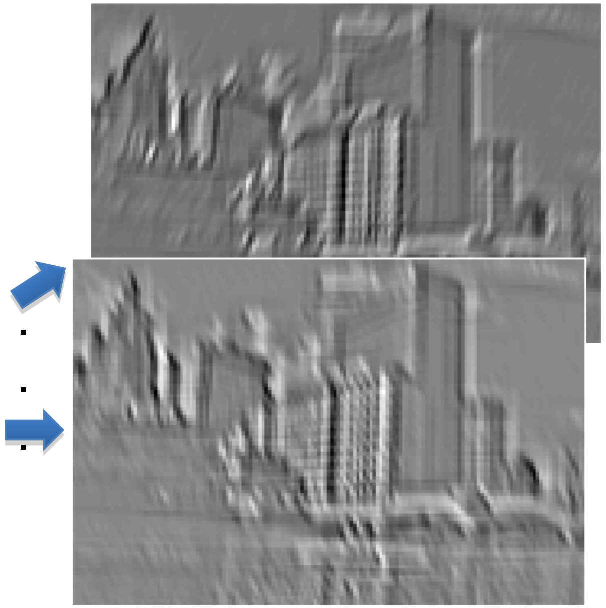


Slide: Lazebnik

Key operations in a CNN



Input

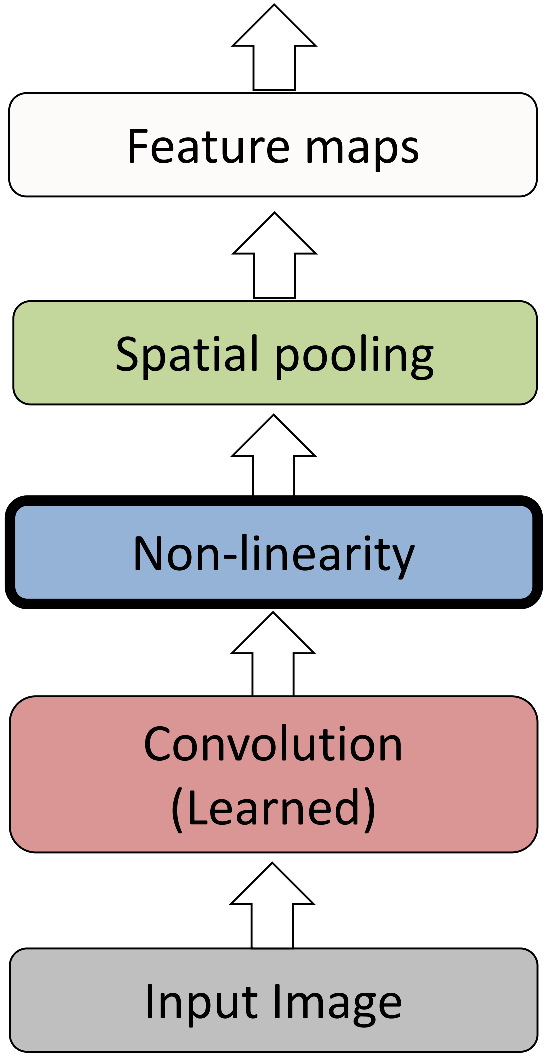


Feature Map

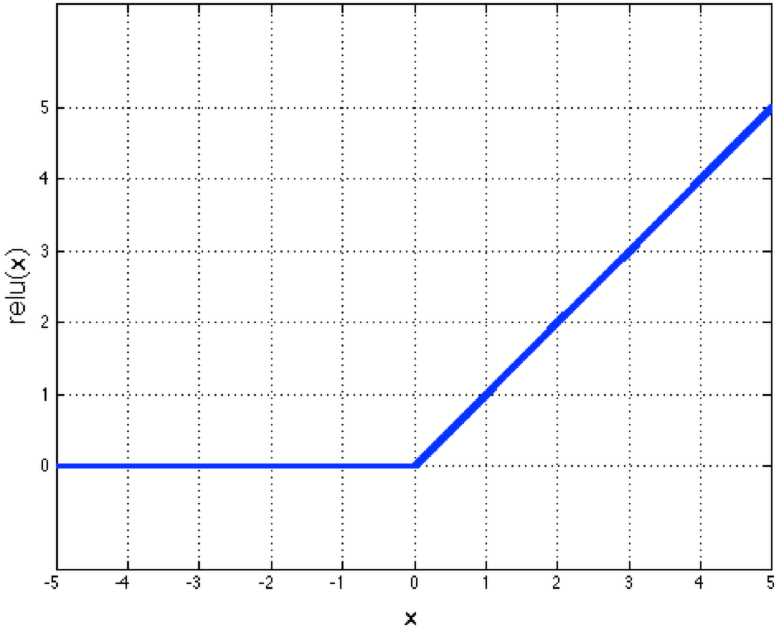
Source: R. Fergus, Y. LeCun

Slide: Lazebnik

Key operations

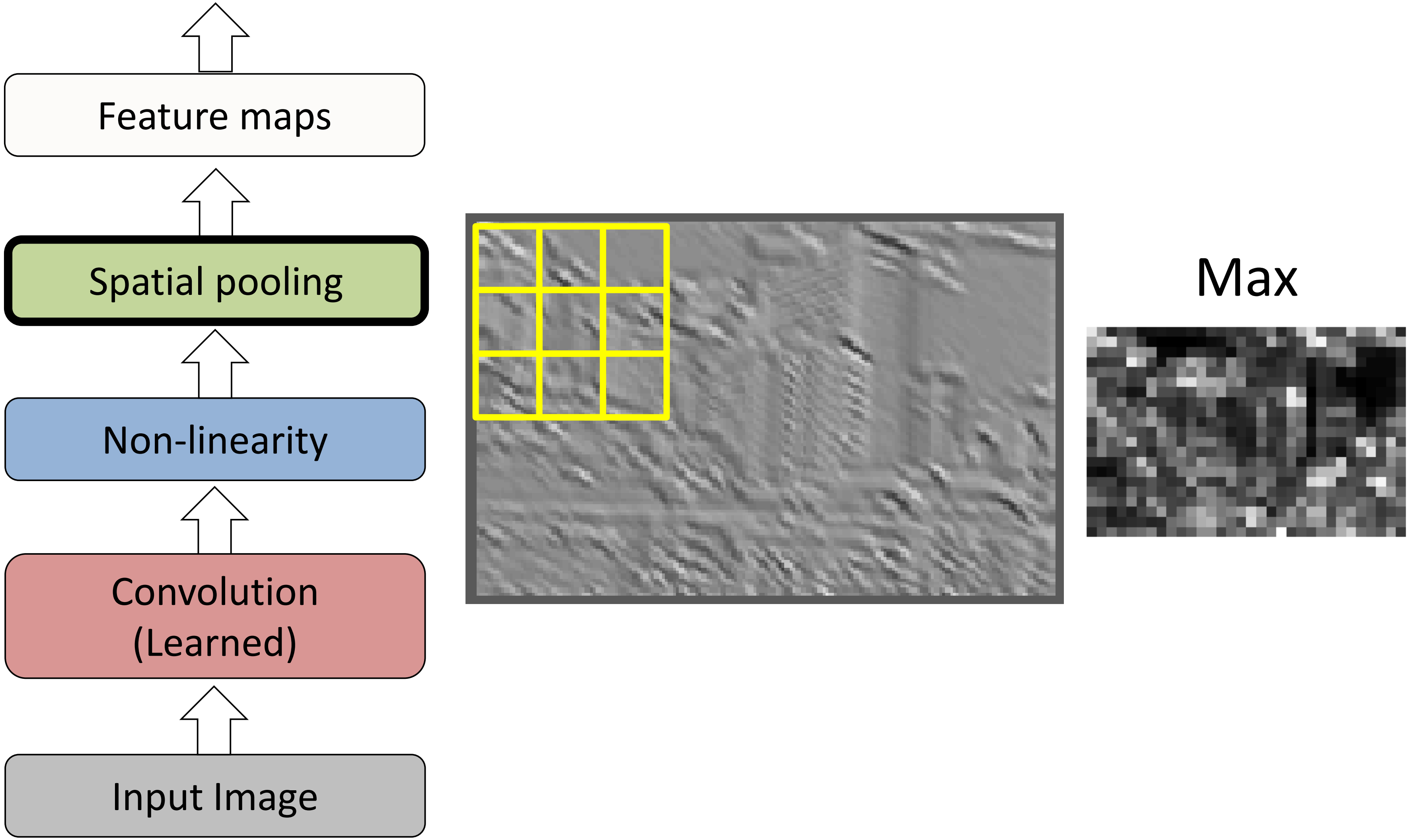


Rectified Linear Unit (ReLU)



Source: R. Fergus, Y. LeCun

Key operations

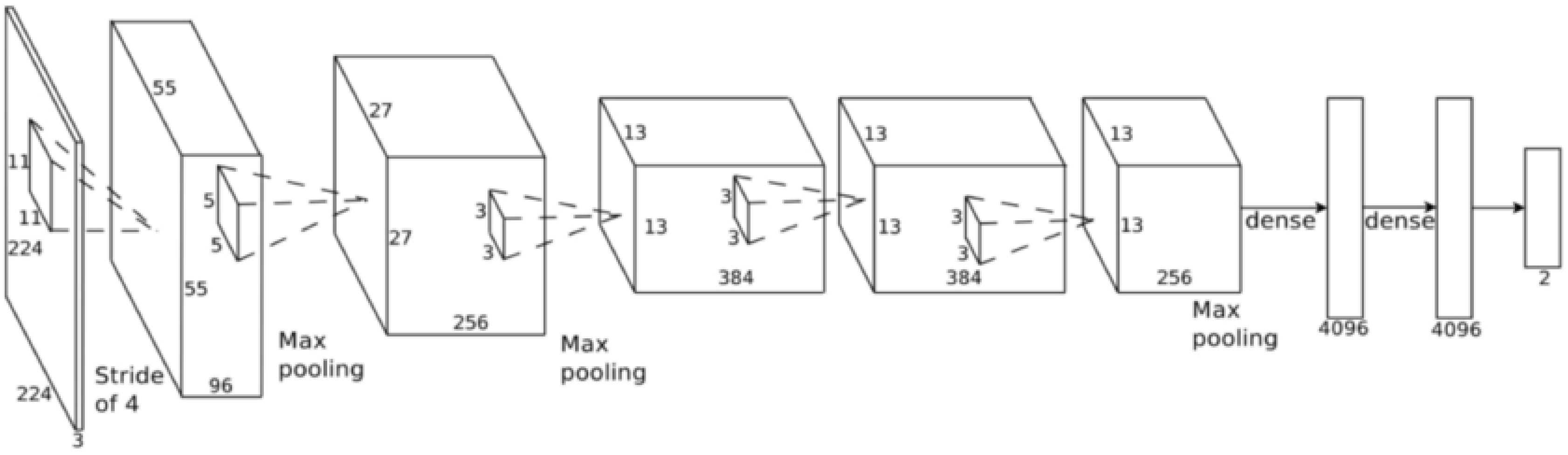


Source: R. Fergus, Y. LeCun

Quick summary of deep network encoders

Create encoding by passing image through a series of steps

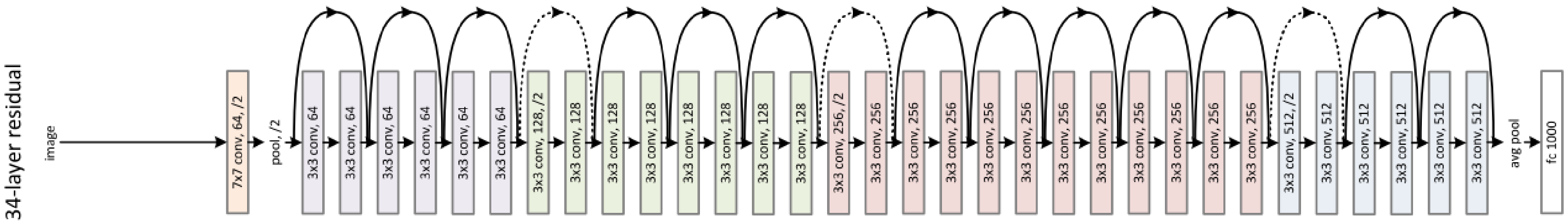
1. Feature generation
   a. Apply filters
   b. ReLU: Zero out negative values
   c. Downsample or “pool” by taking average or max response
2. Vectorize and add dense neural network layers



AlexNet: achieved good results on ImageNet in 2012 to convince computer vision researchers of potential

Most popular architecture is ResNet which adds "skip" connections

- Layers add their response to previous layer outputs so they don't need to re-encode it
- Makes network more compact and easier to train



ResNet Architecture

Key factors in network performance

- **Objective function**: defines what the network is trying to do

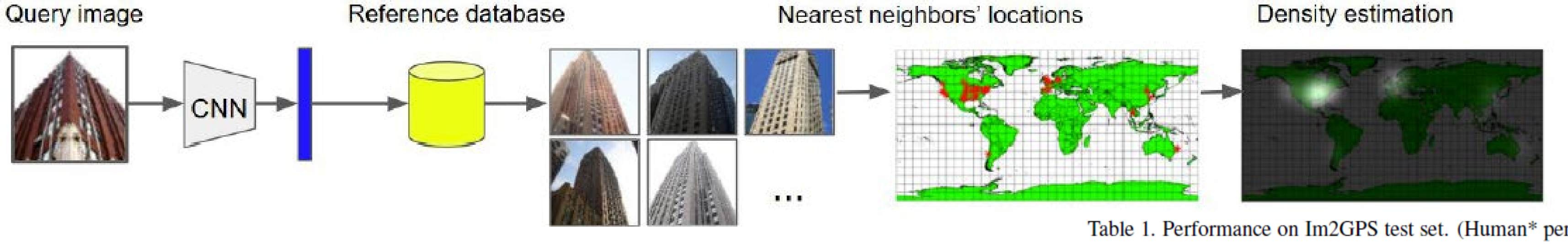
- **Architecture:** number of filters, width of "fully connected layers", connections between layers

- Amount of **training data**: more is better

- **Optimization**: normalization and gradient descent tools

Example: im2gps

- Encoder: deep network that trains to classify images into one of a large number of global regions (classification layers are discarded)
- Decoder: retrieve image(s) with similar encoded representations



"Revisiting Im2GPS in the Deep Learning Era", Vo, Jacobs, Hays 2017

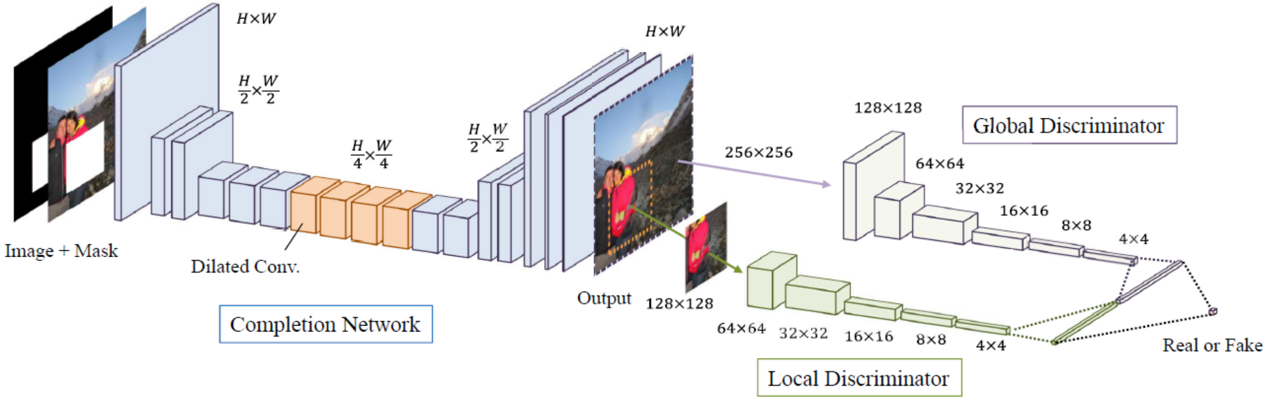
Table 1. Performance on Im2GPS test set. (Human* performance is average from 30 mturk workers over 940 trials, so it might not be directly comparable)

Threshold (km)	Street 1	City 25	Region 200	Country 750	Cont. 2500
Human*			3.8	13.9	39.3
Im2GPS [9]		12.0	15.0	23.0	47.0
Im2GPS [10]	02.5	21.9	32.1	35.4	51.9
PlaNet [36]	08.4	24.5	37.6	53.6	**71.3**
[L] 7011C	06.8	21.9	34.6	49.4	63.7
[L] kNN, σ=4	**12.2**	**33.3**	**44.3**	**57.4**	**71.3**
... 28m database	**14.4**	**33.3**	**47.7**	**61.6**	**73.4**

Globally and Locally Consistent Image Completion

SATOSHI IIZUKA, Waseda University
EDGAR SIMO-SERRA, Waseda University
HIROSHI ISHIKAWA, Waseda University

Fig. 1. Image completion results by our approach. The masked area is shown in white. Our approach can generate novel fragments that are not present elsewhere in the image, such as needed for completing faces; this is not possible with patch-based methods.



SIGGRAPH 2017

Why deep networks work

- "**End-to-end training**": feature learner (encoder) and regressor/classifier (decoder) guided by same objective

- **Flexible objective** design: can use any differentiable function to guide learning

- **Convolutional features** make sense for images because they are shift invariant and have relatively few parameters

- **High capacity** – can encode lots of data

Summary

- Many questions have been asked before, photos have been taken before

- Sometimes, we can shortcut hard problems by looking up the answer

- Deep networks learn features that make the lookup more effective

Next class

- Generating and detecting fakes