

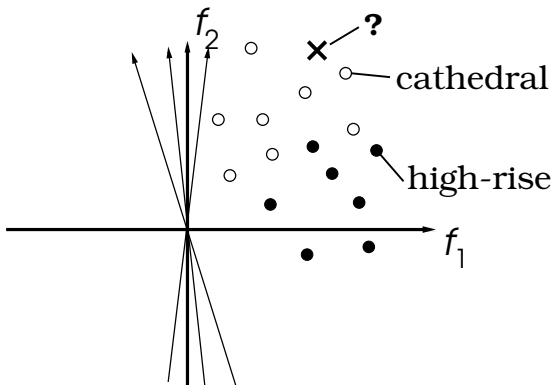
Deep Learning in Image Processing

Roland Memisevic

University of Montreal & TwentyBN

ICISP 2016

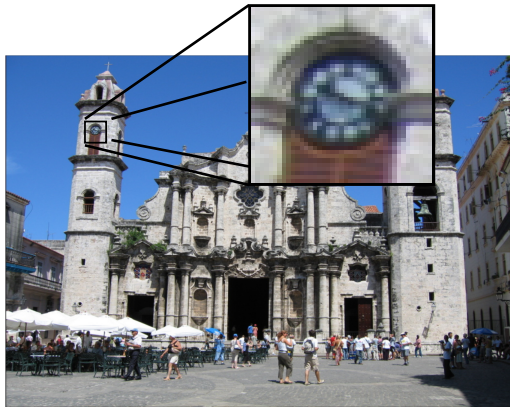




“It’s the features, stupid!”



“It’s the features, stupid!”



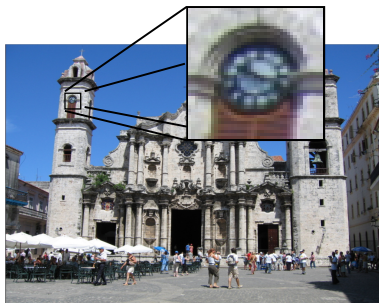
“It’s the features, stupid!”



A common computer vision pipeline before 2012

1. Find interest points.

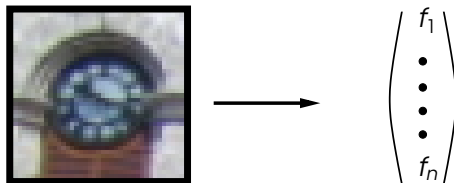
“It’s the features, stupid!”



A common computer vision pipeline before 2012

1. Find interest points.
2. Crop patches around them.

“It’s the features, stupid!”



A common computer vision pipeline before 2012

1. Find interest points.
2. Crop patches around them.
3. Represent each patch with a sparse local descriptor.

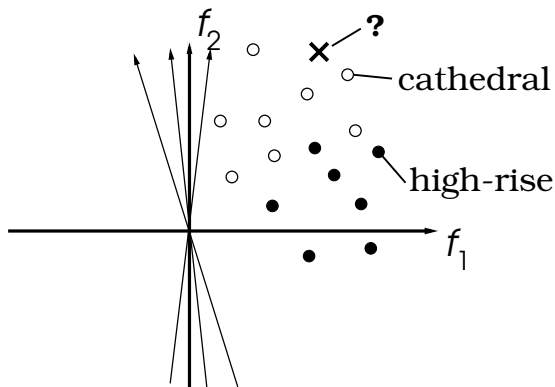
“It’s the features, stupid!”

$$\begin{pmatrix} f_1^1 \\ \vdots \\ f_n^1 \end{pmatrix} + \dots + \begin{pmatrix} f_1^M \\ \vdots \\ f_n^M \end{pmatrix}$$

A common computer vision pipeline before 2012

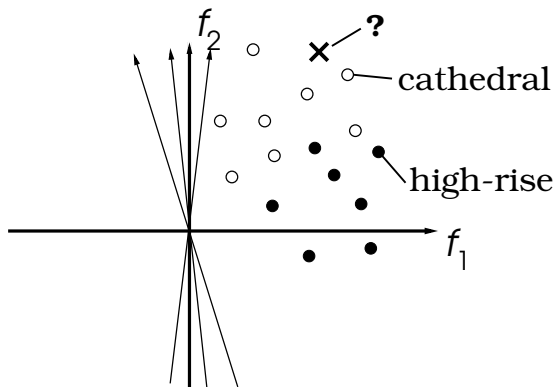
1. Find interest points.
2. Crop patches around them.
3. Represent each patch with a sparse local descriptor.
4. Combine the descriptors into a representation of the image.

“It’s the features, stupid!”



- ▶ This creates a representation that even a linear classifier can deal with.

“It’s the features, stupid!”



- ▶ This creates a representation that even a linear classifier can deal with.

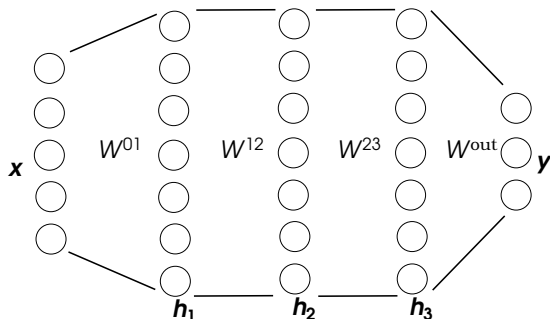
bottom line: **non-linear pipelines are useful**
(aka *“the representation matters”*)

What do good low-level features look like?



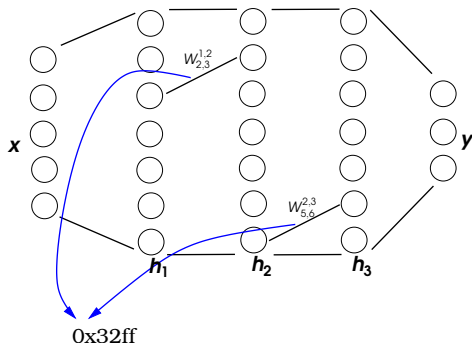
- ▶ Local features that are often found to work well are based on oriented structure (such as Gabor features)
- ▶ These were discovered again and again (also in other areas) and are closely related to the Short Time Fourier Transform.

Neural networks are *trainable* pipelines



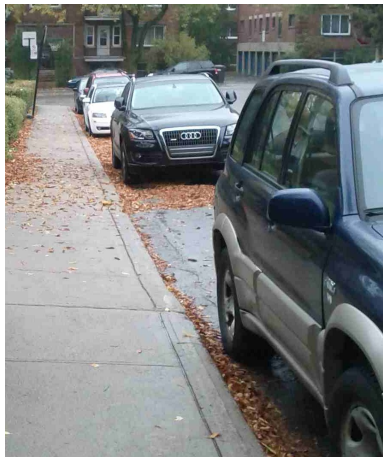
- ▶ Learning allows us to harness **training data**
 $(\mathbf{x}_n, \mathbf{t}_n)_{n=1 \dots N}$
- ▶ Learning allows us to harness **parallel hardware**

Weight sharing



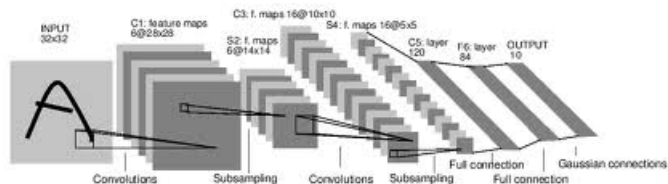
- ▶ Parameters can be shared by having them point to the same memory location.
- ▶ This is a very common way to reduce parameters and encode prior knowledge.
- ▶ The central ingredient in **conv-nets (CNNs)** and **recurrent nets (RNNs)**.
- ▶ *Caveat: It requires long-range communication.*

Translation invariance and locality



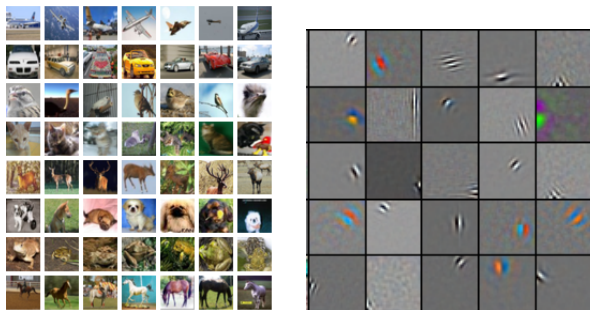
- ▶ Most structure in natural images is **local**
→ Most low-level operations may be based on patches.
- ▶ Most structure in natural images is **position-invariant**
→ We may perform the same set of operations everywhere across the image.

Convolutional neural networks (CNN)



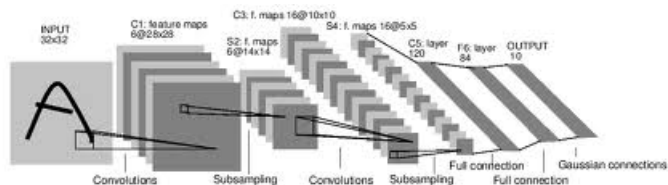
- ▶ Two main ideas:
 1. Use only **local** features.
 2. Apply **the same** features at many different positions.
- ▶ Add subsampling (max- or average-pooling) layers, so that higher layers can see (gradually) larger regions of the image.
- ▶ LeCun et al. 1998

Features *want* to be local



- ▶ (right) Features learned by a fully connected network on RGB images of size 32×32 pixels.
- ▶ A reasonable number of features for these images might be $32 \times 32 \times 3 = 3072$ (a complete basis)
- ▶ Better would be overcomplete.
- ▶ \rightarrow you should use ≥ 9 mio parameters in the first layer

Convolutional neural networks (CNN)



- ▶ The number of parameters in a convolutional layer is:
num vertical filtersize \times num horizontal filtersize \times num input channels \times num output channels
- ▶ Without weight sharing, this could be, say,
 $10 \times 10 \times 3 \times 3072 \approx 1$ mio (if complete)
- ▶ With weight sharing this could be
 $10 \times 10 \times 3 \times 100 \approx 30$ k (already overcomplete)
- ▶ The number of *hidden units* in this case:
num vertical pixels \times num horizontal pixels \times num channels (eg. $32 \times 32 \times 100 = 102400$)

Neural network activations are sparse

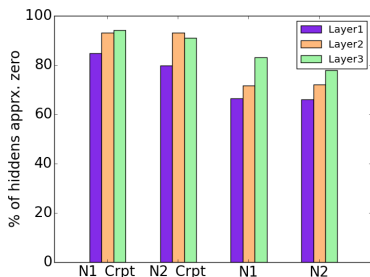


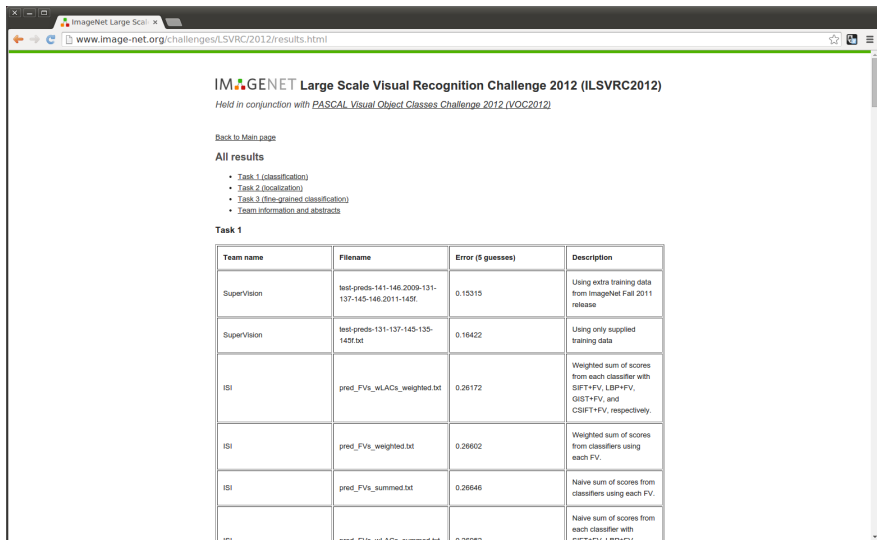
figure by Kishore Konda

Sparsity levels in two networks trained on CIFAR-10. N1=(1000-2000-3000), N2=(2000-2000-2000 units). (N1_Crpt, N2_Crpt trained with dropout).

- ▶ Sparsity is good, because it “disentangles”.
- ▶ However, sparsity can lead to update scarcity.
- ▶ Conv-nets maximize *the ratio*

$$\frac{\#neurons}{\#parameters}$$

ImageNet challenge 2012



ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

Held in conjunction with [PASCAL Visual Object Classes Challenge 2012 \(VOC2012\)](#)

[Back to Main page](#)

All results

- [Task 1 \(classification\)](#)
- [Task 2 \(localization\)](#)
- [Task 3 \(fine-grained classification\)](#)
- [Team information and abstracts](#)

Task 1

Team name	Filename	Error (5 guesses)	Description
SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
ISI	pred_FVs_wLACs_weighted.txt	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.
ISI	pred_FVs_summed.txt	0.26646	Naive sum of scores from classifiers using each FV.
ISI	pred_FVs_wLACs_summed.txt	0.26682	Naive sum of scores from each classifier with SIFT+FV, LBP+FV,

GoogLeNet (Szegedy et al. 2014)



- ▶ Won ImageNet 2014 with **6.66%** top-5 error rate
- ▶ Key insights: Scaling up, unconventional architecture (eg. cross-channel pooling), intermediate targets
- ▶ A variation of this network based on *Batch Normalization* (Ioffe, Szegedy, 2015) achieves **4.8%** top-5 error rate, surpassing the accuracy of human labelers
- ▶ Since then:
 - ▶ Face identification
 - ▶ Place recognition
 - ▶ Scene rendering
 - ▶ Denoising
 - ▶ Robotics, etc.

Research challenges

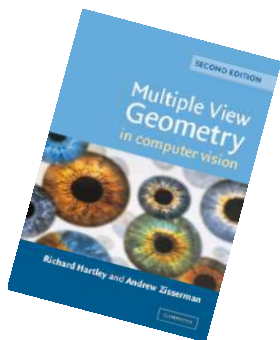
- ▶ Hardware, scaling up
- ▶ Network design (attention mechanisms, dealing with vanishing gradients, memory mechanisms, etc.)
- ▶ Multi-view learning
- ▶ Datasets
- ▶ Reinforcement learning
- ▶ Transfer learning
- ▶ Theory

Research challenges

- ▶ Hardware, scaling up
- ▶ Network design (attention mechanisms, dealing with vanishing gradients, memory mechanisms, etc.)
- ▶ **Multi-view learning**
- ▶ Datasets
- ▶ Reinforcement learning
- ▶ Transfer learning
- ▶ Theory

Vision is not object recognition

- ▶ Many vision (and other) tasks depend on encoding relations across multiple images:
- ▶ Geometry, stereo, structure-from-motion, motion understanding, activity analysis, tracking, optical flow, modeling object relations, articulation, odometry, analogy, ...

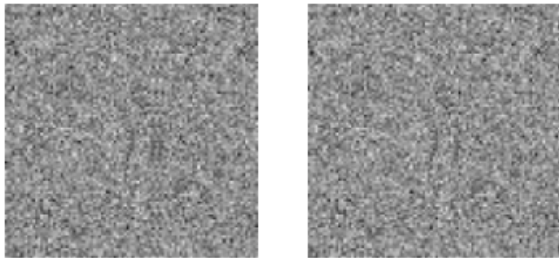


Some things are hard to infer from still images

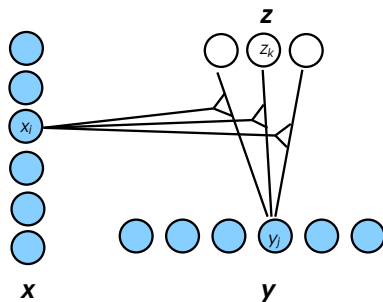


(Ayvaci, Soatto 2012)

Random dot stereograms



Learning transformations with bi-linear models

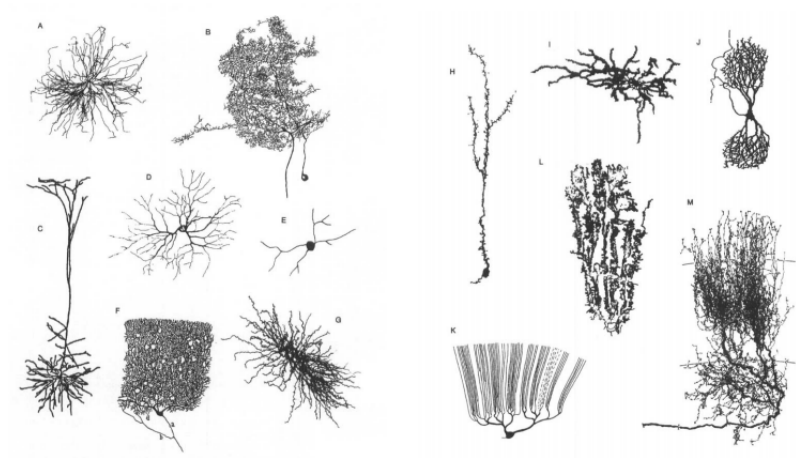


- ▶ $w_{jk}(\mathbf{x}) = \sum_i w_{ijk} x_i$, so

$$z_k = \sum_j w_{jk} y_j = \sum_j \left(\sum_i w_{ijk} x_i \right) y_j = \sum_{ij} w_{ijk} x_i y_j$$

(Tenenbaum, Freeman; 2000), (Grimes, Rao; 2005),
(Olshausen; 2007), (Memisevic, Hinton; 2007)

Real neurons: $w^T x$?



(Mel, 1994)

Some theory (Konda et al. ICLR 2014)

- ▶ Assume two images \mathbf{x} , \mathbf{y} are related through an orthogonal transformation T :

$$\mathbf{y} = T\mathbf{x}$$

- ▶ **Goal:** Detect the transformation, given the images
- ▶ **Synchrony condition:** Take a filter pair \mathbf{w}_x , \mathbf{w}_y with

$$\mathbf{w}_y = T\mathbf{w}_x$$

and check whether

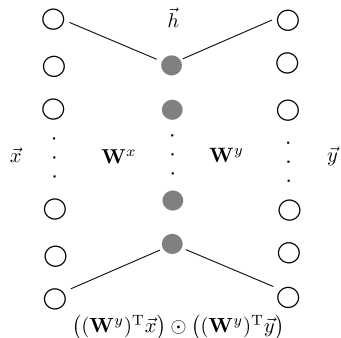
$$\mathbf{w}_y^T \mathbf{y} = \mathbf{w}_x^T \mathbf{x}$$

- ▶ **Why does this work?**

$$\mathbf{y} = T\mathbf{x} \Rightarrow \mathbf{w}_y^T \mathbf{y} = \mathbf{w}_y^T T\mathbf{x} = (T^T \mathbf{w}_y)^T \mathbf{x} = \mathbf{w}_x^T \mathbf{x}$$

A learning algorithm (“synchrony K-means”)

- ▶ Goal: learn the filters given a set of image pairs
- ▶ Use the activation function



$$s = \arg \max_k [(\mathbf{w}_x^k)^T \mathbf{x} (\mathbf{w}_y^k)^T \mathbf{y}]$$

- ▶ Define the reconstruction error:

$$L_y = \left(\mathbf{y} - \mathbf{w}_y^s ((\mathbf{w}_x^s)^T \mathbf{x}) \right)^2$$

- ▶ Differentiating yields the update rule:

$$\Delta \mathbf{w}_y^s = \eta \left(\mathbf{y} (\mathbf{w}_x^s)^T \mathbf{x} - \mathbf{w}_y^s \left(((\mathbf{w}_x^s)^T \mathbf{x})^2 \right) \right)$$

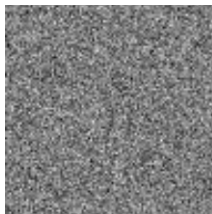
- ▶ This is a Hebbian term plus an active forgetting term.

Learning stereo vision

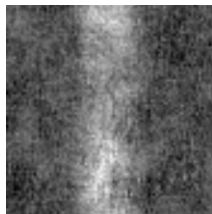
left image



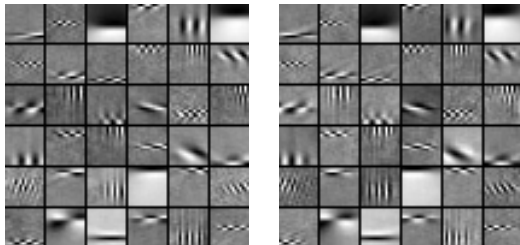
right image



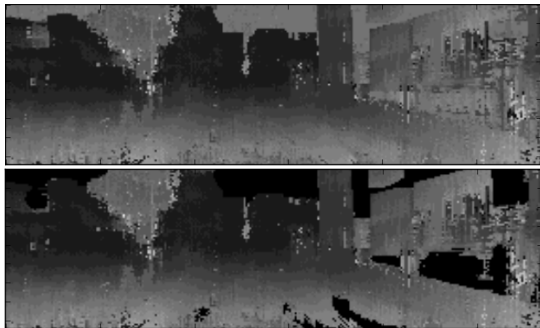
predicted disparity



Learning stereo vision (Konda et al. 2013)



Inferred depth map



- ▶ top: inferred depth map,
- ▶ bottom: thresholded to remove uncertain regions

Learning depth and motion: Hollywood 3D



x_{left} :



x_{right} :



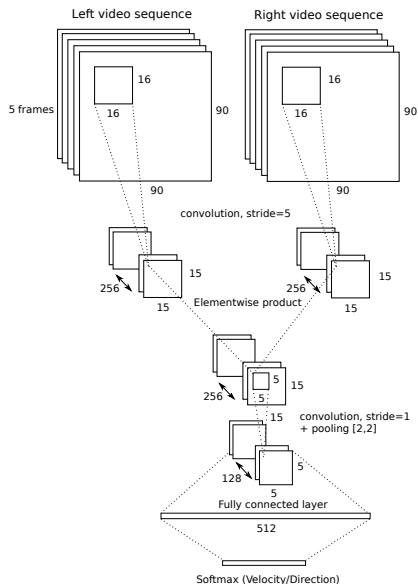
Action	SAE-MD	SAE-MD(Av)	SAE-MD(Ct)	SAE-M	SAE-D	ISA	3D-Ha	4D-Ha	3.5D-Ha
NoAction	12.10	12.77	13.10	15.73	12.15	12.27	12.1	12.9	13.7
Run	52.56	50.44	51.45	45.38	56.07	24.91	19.0	22.4	27.0
Punch	41.09	38.01	32.68	33.86	36.17	31.17	10.4	4.8	5.7
Kick	9.41	7.94	6.86	6.63	11.84	9.96	9.3	4.3	4.8
Shoot	30.26	35.51	30.49	30.52	40.72	32.48	27.9	17.2	16.6
Eat	5.85	7.03	6.78	7.29	9.03	6.89	5.0	5.3	5.6
Drive	52.65	59.62	51.35	61.61	45.19	54.47	24.8	69.3	69.6
UsePhone	22.79	23.92	19.01	23.60	23.36	17.67	6.8	8.0	7.6
Kiss	15.03	16.40	16.12	17.86	17.06	14.94	8.4	10.0	10.2
Hug	6.64	7.02	7.61	7.38	9.27	9.48	4.3	4.4	12.1
StandUp	37.35	34.23	37.01	29.16	15.01	26.71	10.1	7.6	9.0
SitDown	6.51	6.95	7.53	7.40	9.06	5.13	5.3	4.2	5.6
Swim	16.58	29.48	17.60	29.45	26.70	16.09	11.3	5.5	7.5
Dance	43.15	36.26	44.59	29.64	25.12	53.72	10.1	10.5	7.5
mean	25.14	26.11	24.45	24.61	24.05	22.55	12.6	13.3	14.1
AP									

Combining depth and motion for visual odometry

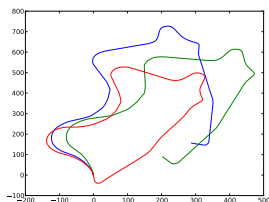
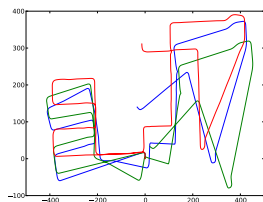


Images taken from KITTI odometry dataset and <http://www.cvlibs.net/datasets/kitti/index.php>

Learning visual odometry



Experiments



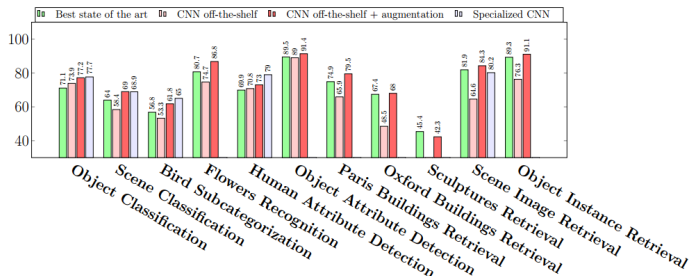
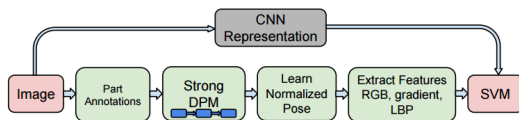
(d) Train sequence (Seq.8) (e) Test sequence (Seq.9)

Ground truth shown in **red**, discretised ground truth in **green**, prediction in **blue**.
(Need landmark detection/loop closure techniques to improve maps.)

Research challenges

- ▶ Hardware, scaling up
- ▶ Network design (attention, vanishing gradients, neural programs)
- ▶ Multi-view learning
- ▶ Datasets
- ▶ Reinforcement learning
- ▶ **Transfer learning**
- ▶ Theory

Conv-nets learn good *generic* features

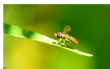


(Razavian, Azizpour, Sullivan, Carlsson; 2014), see also (Donahue et al, 2013)

Karayev et al 2014: Recognizing Image Style



HDR



Macro



Baroque



Rococo



Vintage



Noir



Northern Renaissance



Cubism



Minimal



Hazy



Impressionism



Post-Impressionism



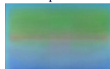
Long Exposure



Romantic



Abs. Expressionism

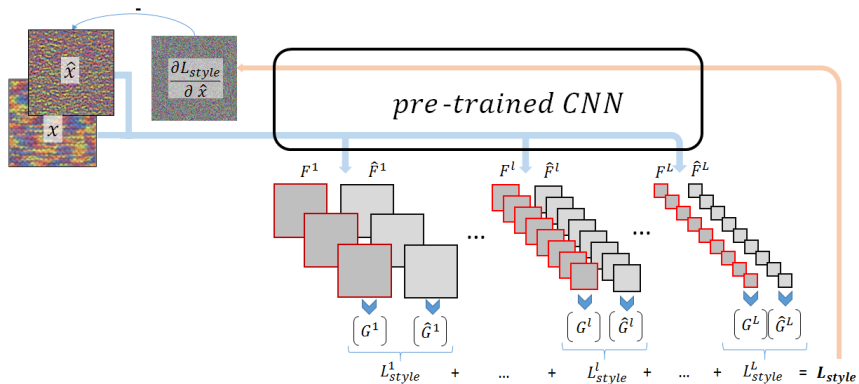


Color Field Painting

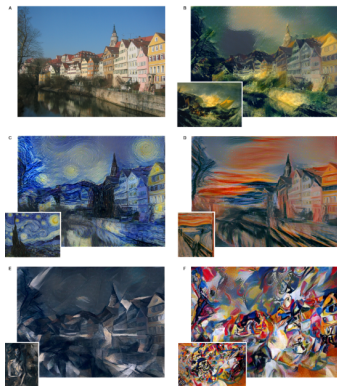
Flickr Style: 80K images covering 20 styles.

Wikipaintings: 85K images for 25 art genres.

A surprising application of imagenet-features: Gatys, Ecker, Bethge (2015)

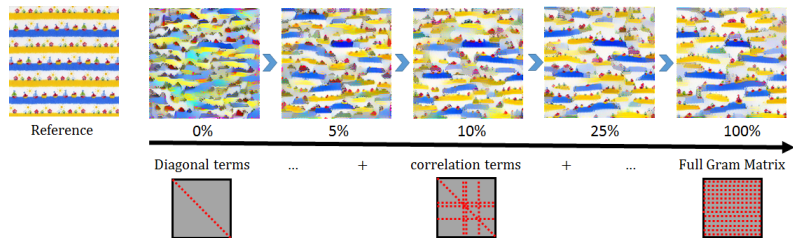


Adding a content-cost (Gatys, Ecker, Bethge; 2015)



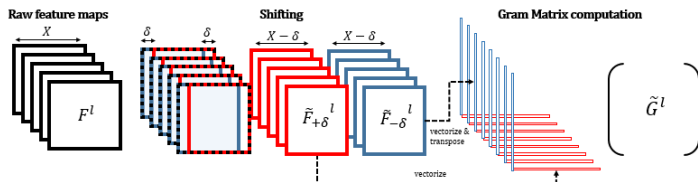
- ▶ Optimize pixels so as to (i) match the hidden layer activations of the content image (ii) match a non-linear function of the hidden layer activations of a “style image”

Why do Gramians work? (joint work with Guillaume Berger)

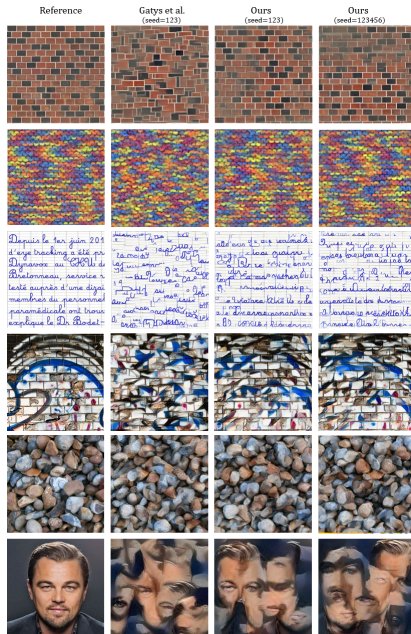


- ▶ Gatys et al's Gramian measures *coincidences across features*.
- ▶ On the lowest layer, this amounts to accounting for the relative "arrival phase" of individual Fourier components.
- ▶ Averaging is necessary, because textures are static by definition.

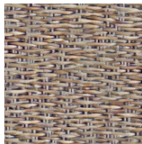
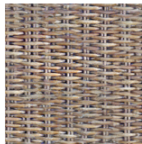
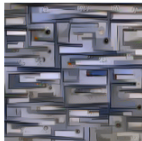
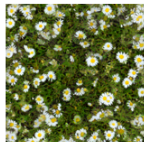
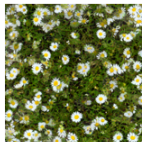
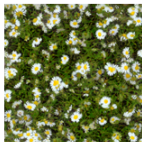
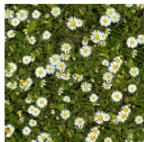
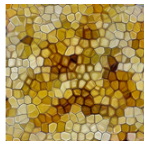
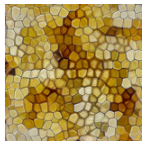
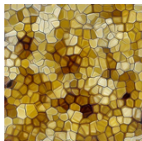
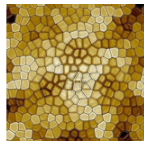
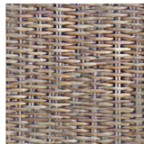
Adding long-range structure



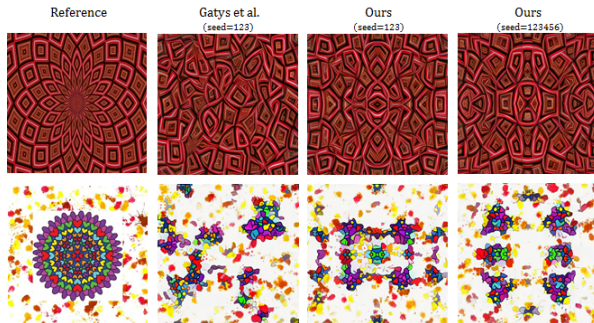
- ▶ To account for long-range correlations, we should measure coincidences across feature maps *and* across space.



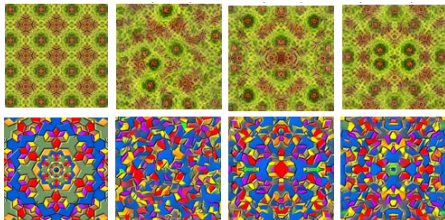
Reference

Gatys et al.
(seed=123)Ours
(seed=123)Ours
(seed=123456)

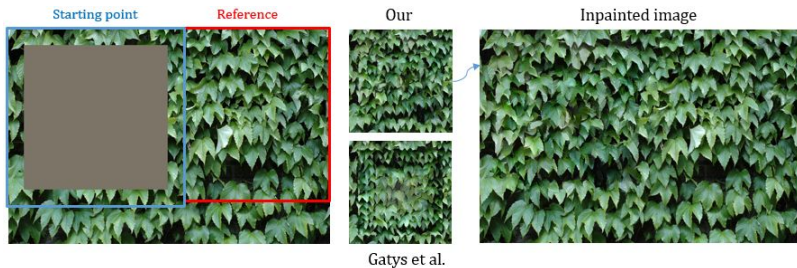
Symmetric textures



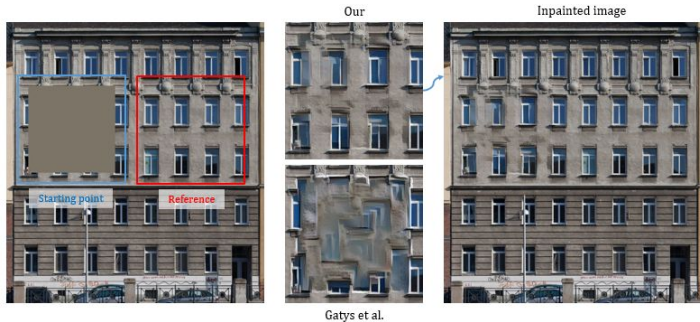
Symmetric textures



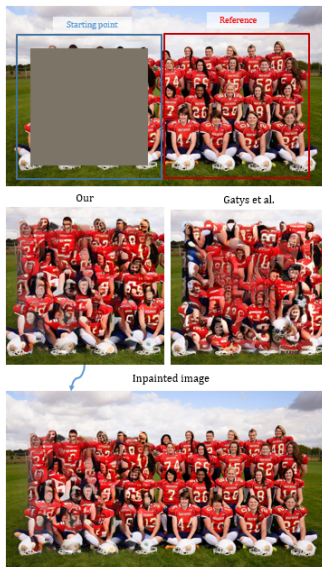
In-painting



In-painting



In-painting



What is wrong with unsupervised learning?

- ▶ The irony: We are using image classification to improve rendering, *not* the other way around.
- ▶ Transfer learning is becoming a huge practical success while unsupervised learning never took off.
- ▶ One reason why UL may be the wrong approach (but transfer learning should work):
There is more structure in natural images than that which is relevant for humans.
- ▶ Teasing out “the structure” in natural data, as attempted by UL, may be asking too much.
- ▶ Across tasks and modalities, humans get *a lot* of supervision signals:

Just how generic is human cognition?

$$3 + x = 7$$

- ▶ We solve equations like these using “dexterous manipulation” involving our motor cortex.
- ▶ See, eg., (Hofstadter, Sander 2013), (Lakoff, 1980), or the “embodied cognition” movement
- ▶ We now know that analogy making has a concrete practical benefit! They allow us to get more data!

**Transfer learning and analogical reasoning are useful,
because they eliminate the data dilemma**

Thank you
Questions?