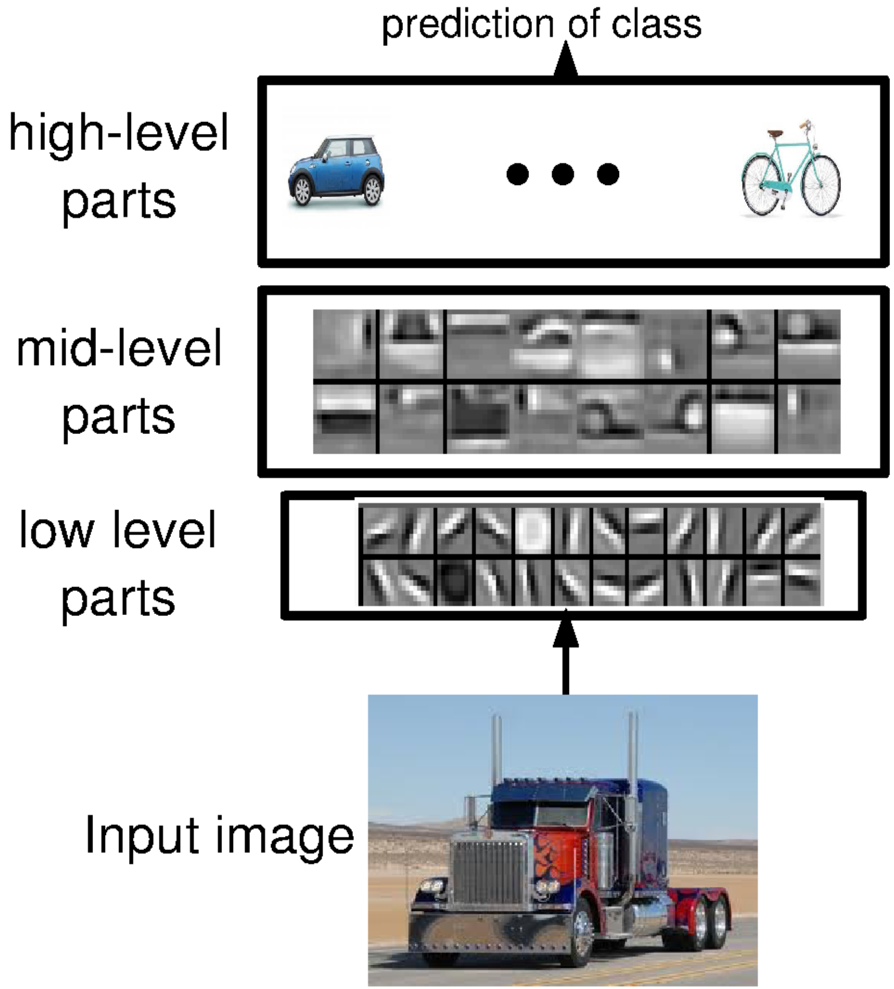
Convnet Applications

Samuel Cheng

(Slide credit: James Thompkins, Juan Carlos Niebles and Ranjay Krishna)

Interpretation



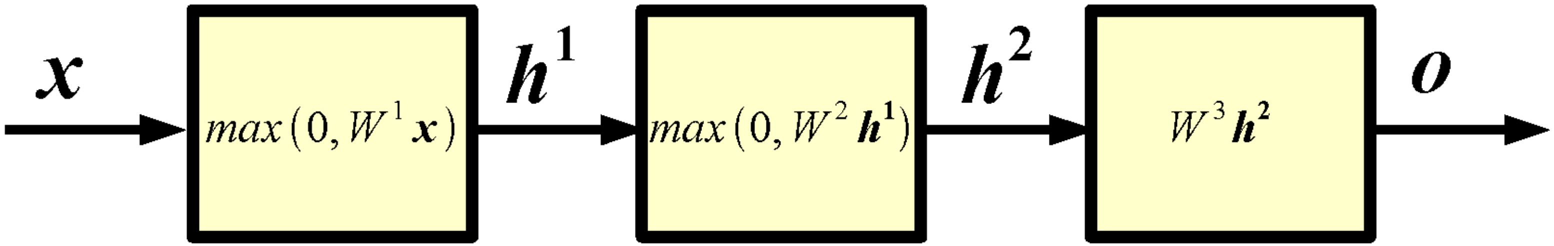
- distributed representations
- feature sharing
- compositionality

NOTE: Not actually the weights; a demonstrative visualization!

Lee et al. "Convolutional DBN's ..." ICML 2009

Ranzato

Neural Networks: example



x input

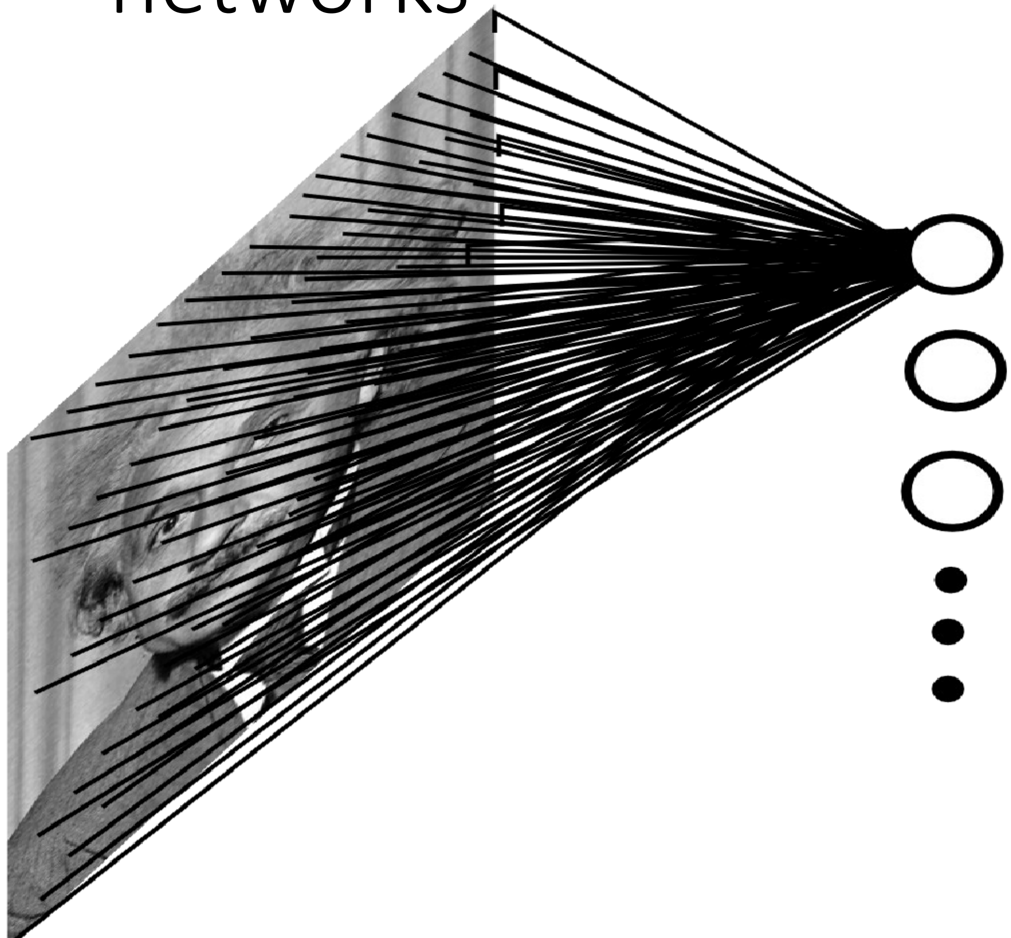
h^1 1-st layer hidden units

h^2 2-nd layer hidden units

o output

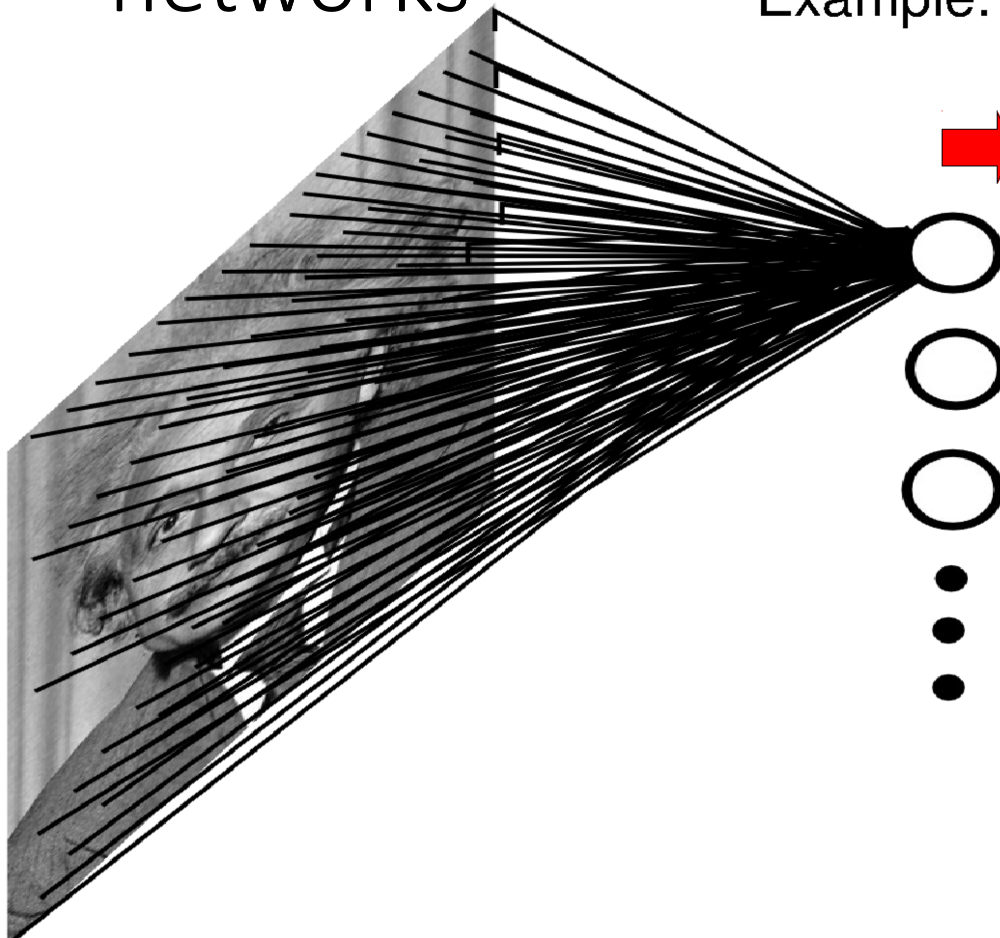
Example of a 2 hidden layer neural network (or 4 layer network, counting also input and output).

Images as input to neural networks



Ranzato

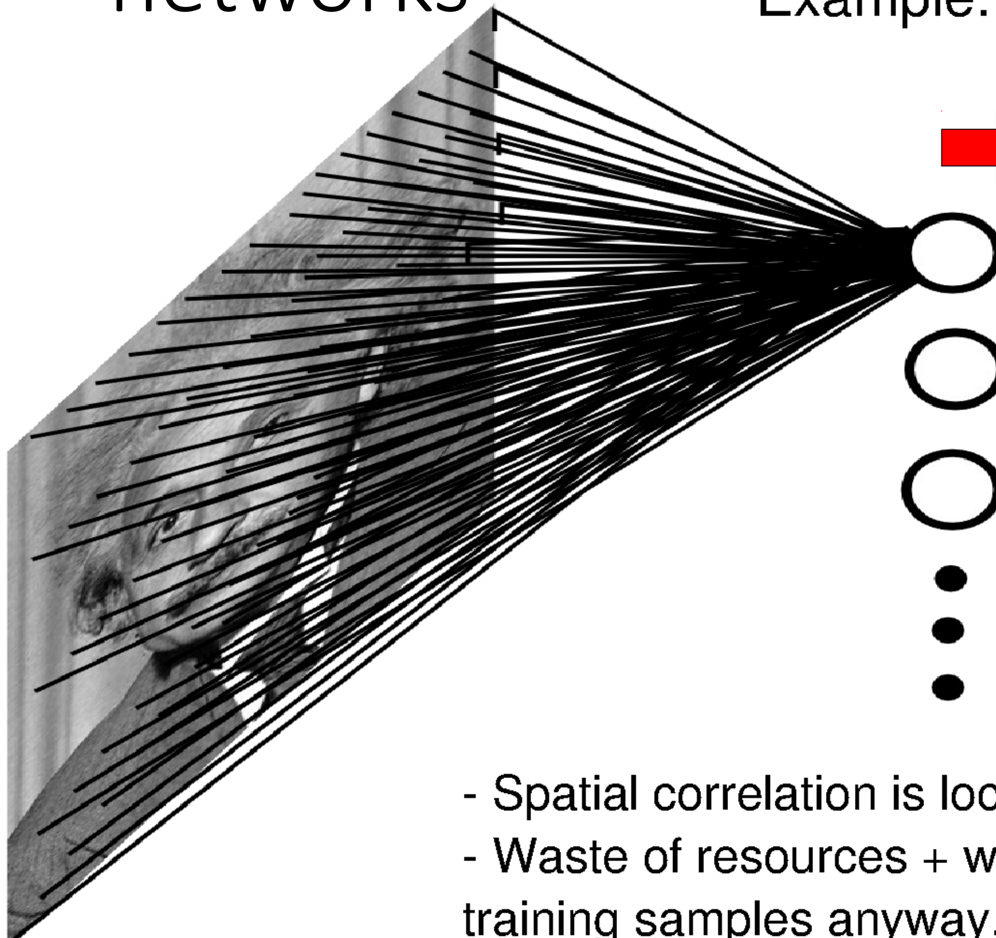
Images as input to neural networks



Example: 200x200 image
40K hidden units
~2B parameters!!!

Ranzato

Images as input to neural networks



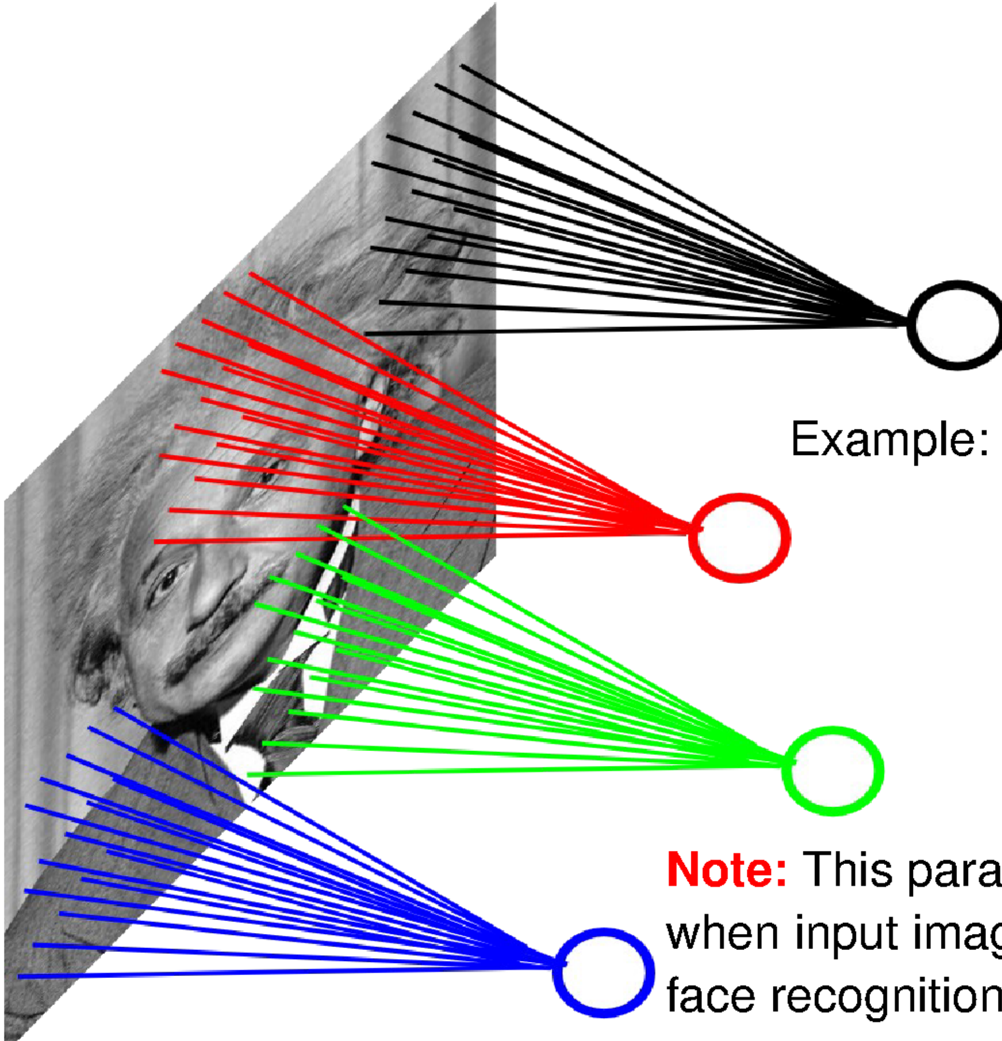
Example: 200x200 image
40K hidden units
~2B parameters!!!

- Spatial correlation is local
- Waste of resources + we have not enough training samples anyway..

Ranzato

Motivation

- Sparse interactions – *receptive fields*
  - Assume that in an image, we care about ‘local neighborhoods’ only for a given neural network layer.
  - Composition of layers will expand local -> global.



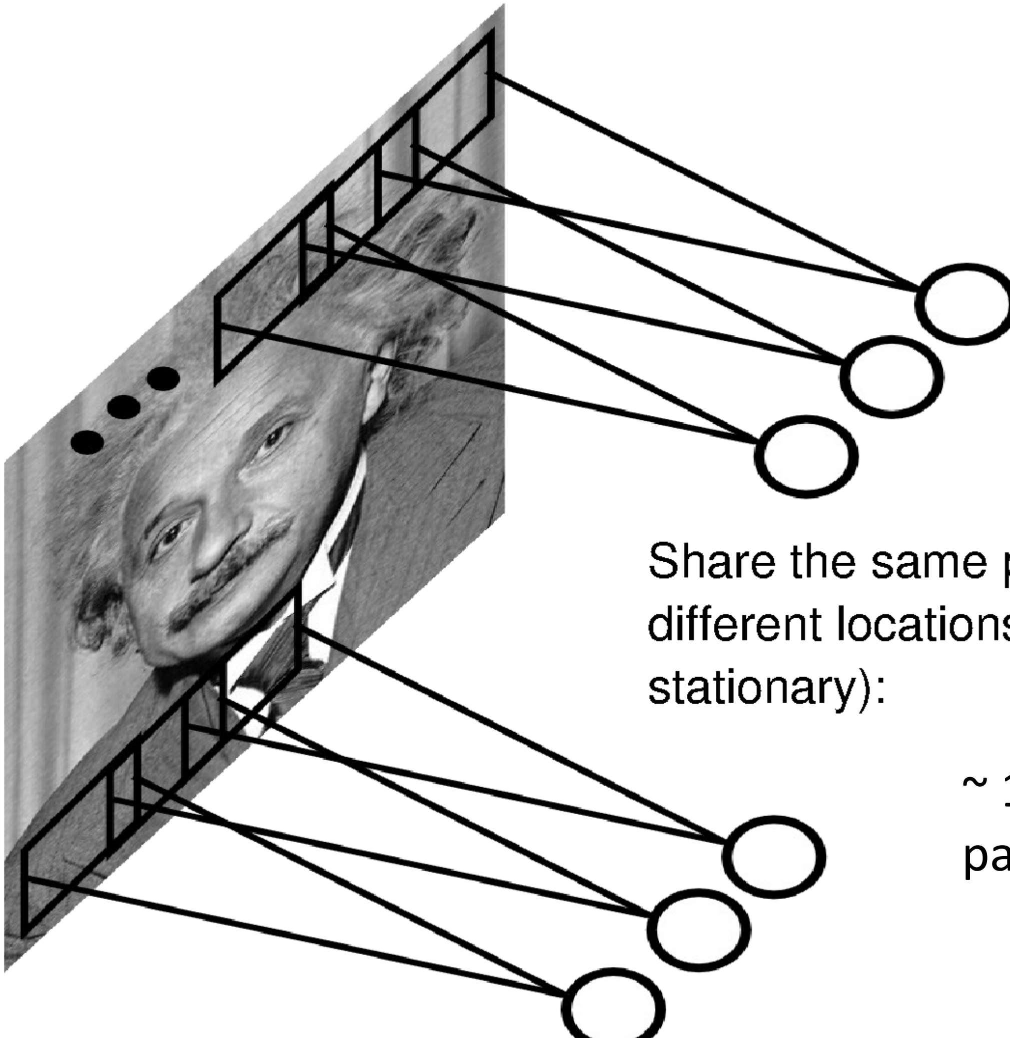
Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Note: This parameterization is good when input image is registered (e.g., face recognition).

Ranzato

Motivation

- Sparse interactions – *receptive fields*
  - Assume that in an image, we care about 'local neighborhoods' only for a given neural network layer.
  - Composition of layers will expand local -> global.
- Parameter sharing
  - 'Tied weights' – use same weights for more than one perceptron in the neural network.
  - Leads to *equivariant representation*
    - If input changes (e.g., translates), then output changes similarly



Share the same parameters across different locations (assuming input is stationary):

~ 10 x 10 = 100 parameters

Ranzato

Filtering reminder:

Correlation (rotated convolution)

$$f[\cdot\,,\cdot\,] \quad \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

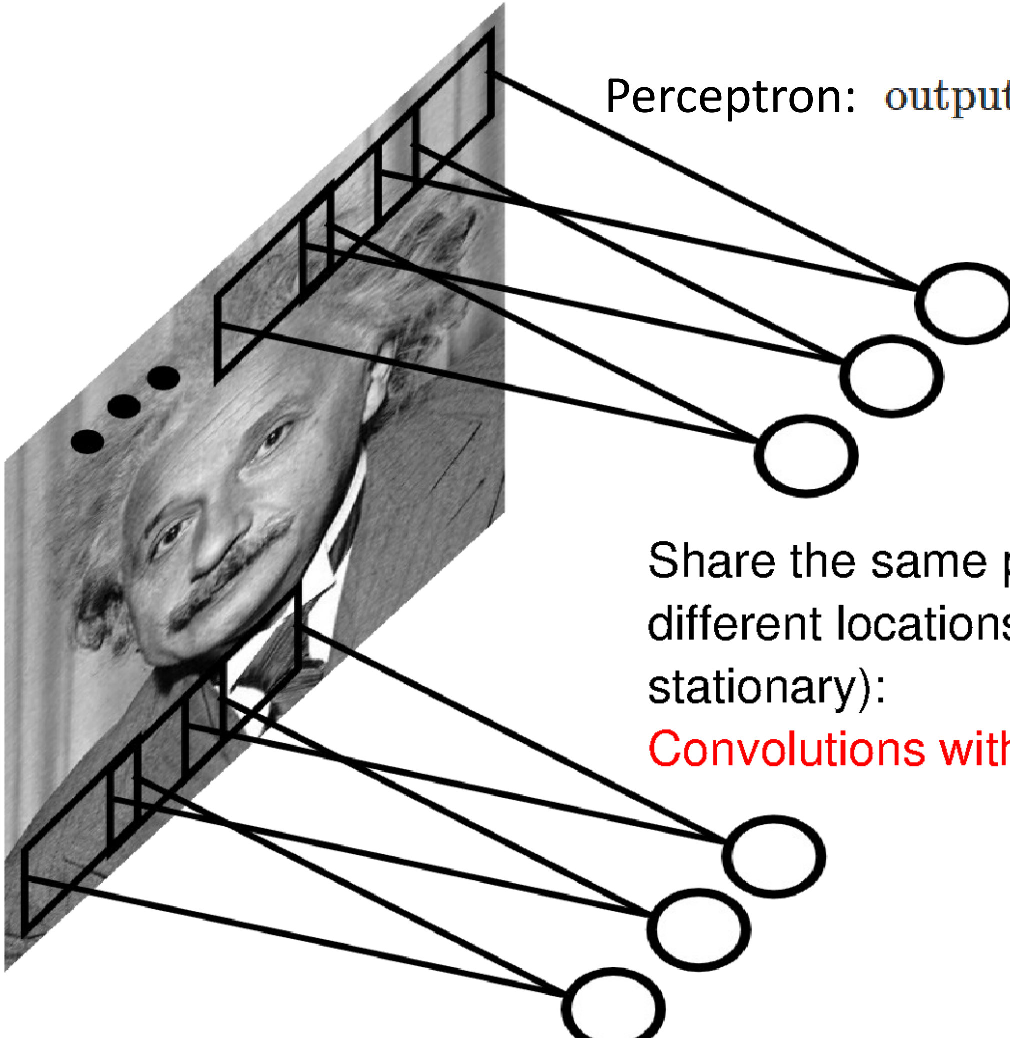
$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

Credit: S. Seitz

Convolutional Layer



Perceptron: $\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$

$$w \cdot x \equiv \sum_j w_j x_j,$$

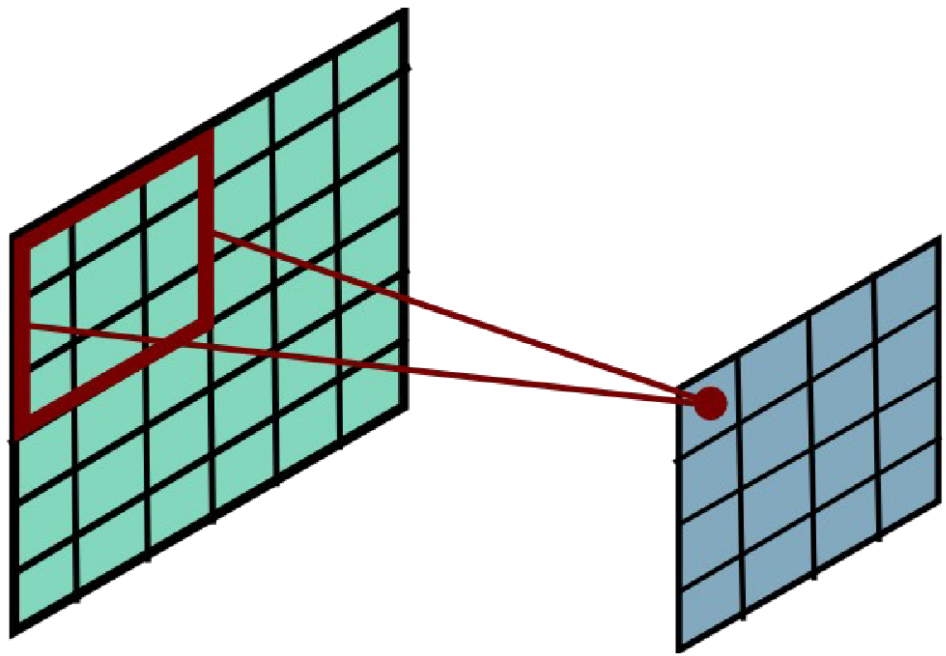
This is convolution!

Share the same parameters across different locations (assuming input is stationary):

Convolutions with learned kernels

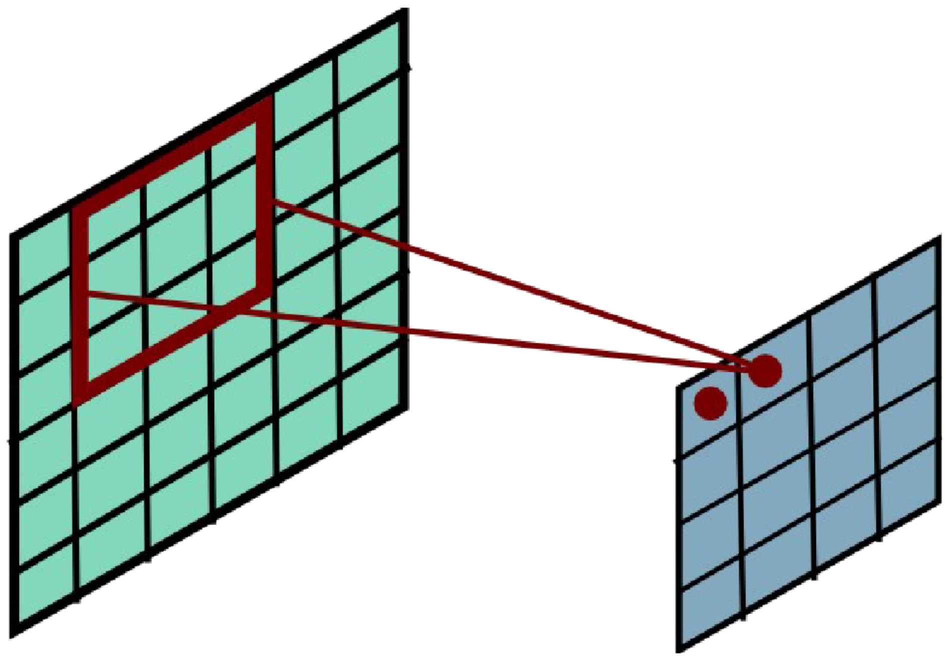
Ranzato

Convolutional Layer

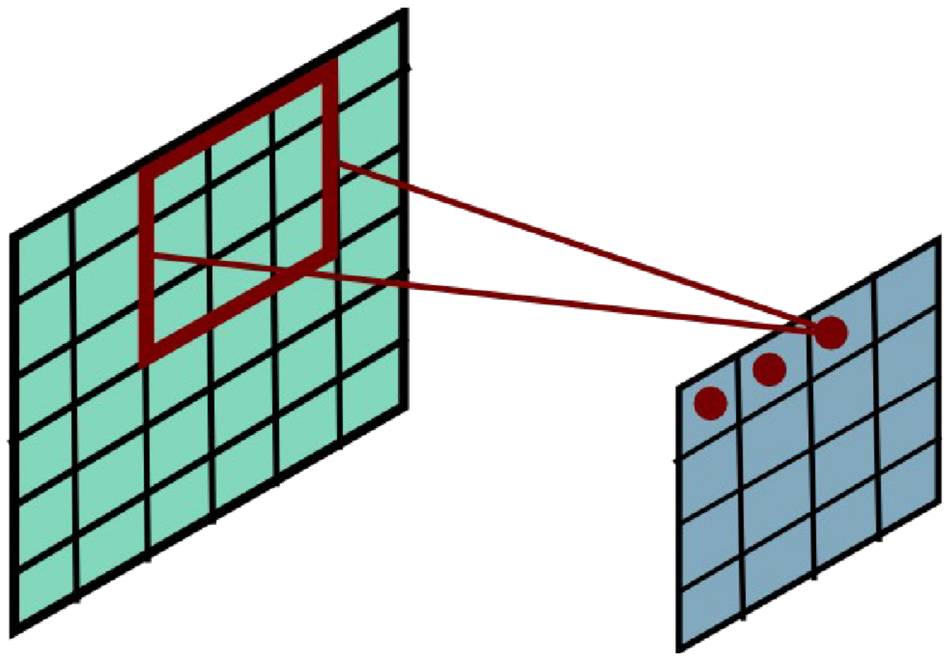


Ranzato

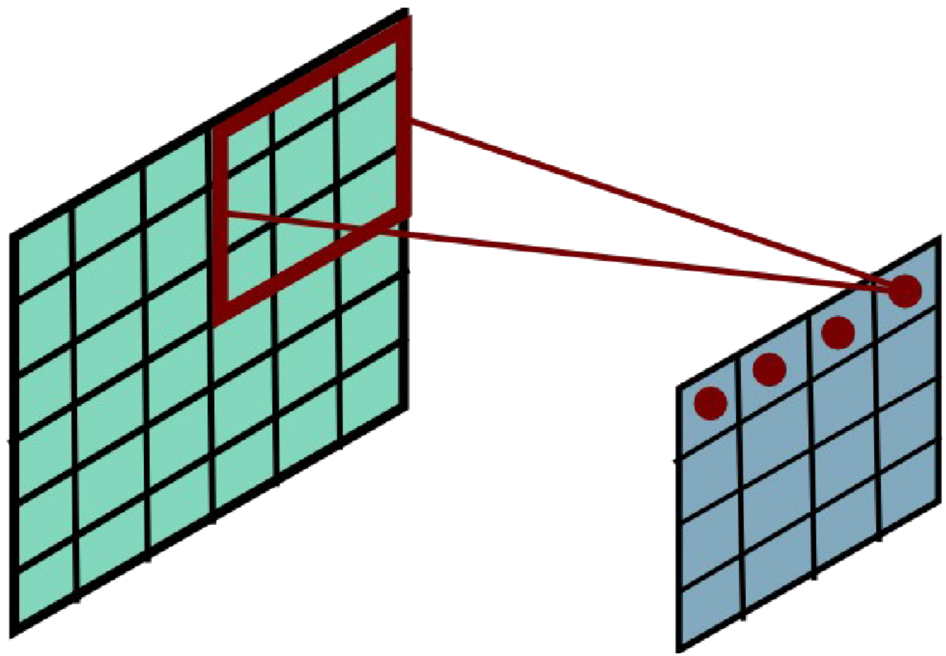
Convolutional Layer



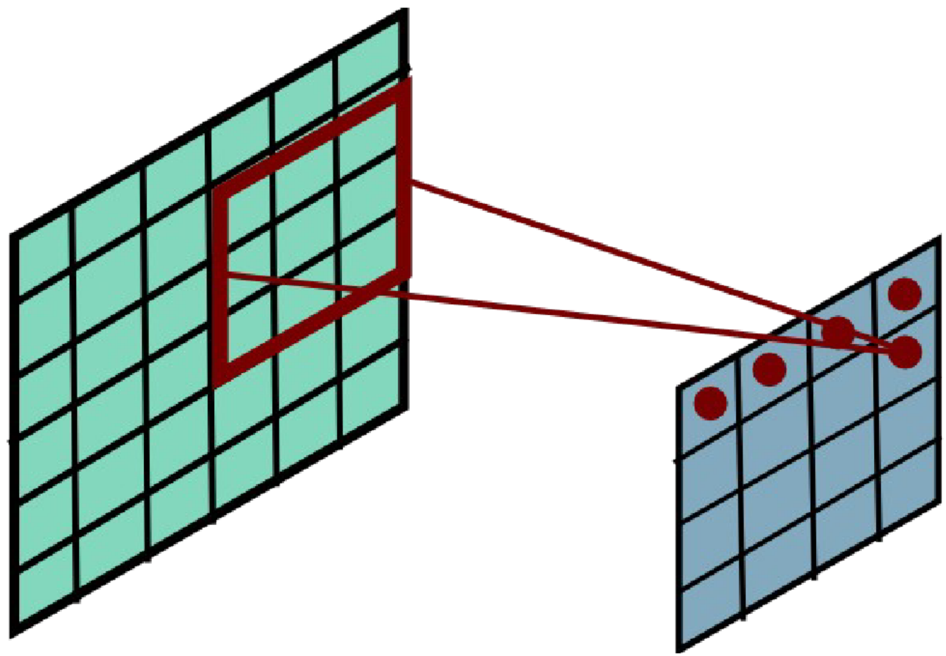
Convolutional Layer



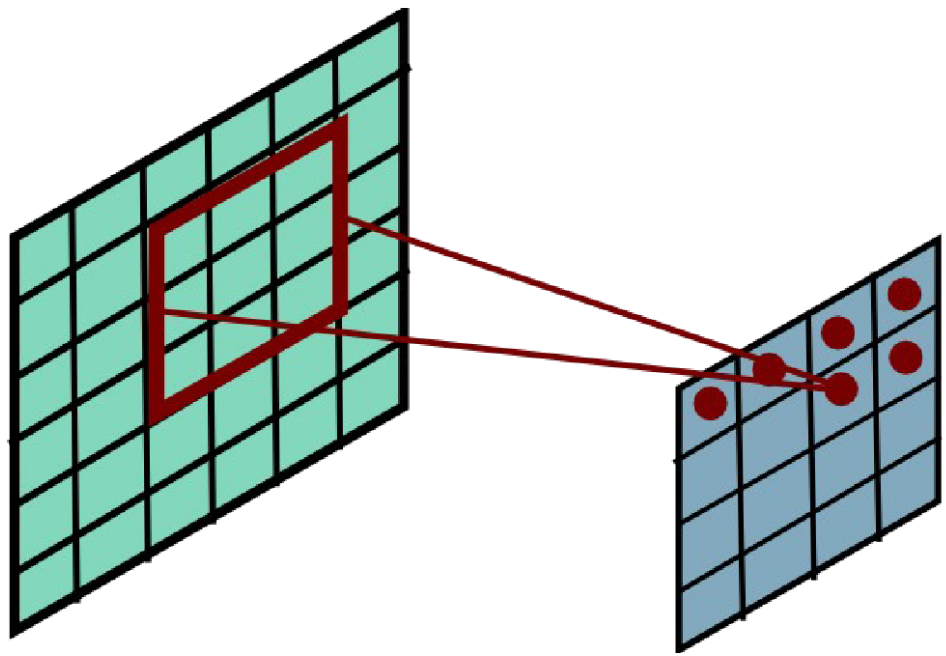
Convolutional Layer



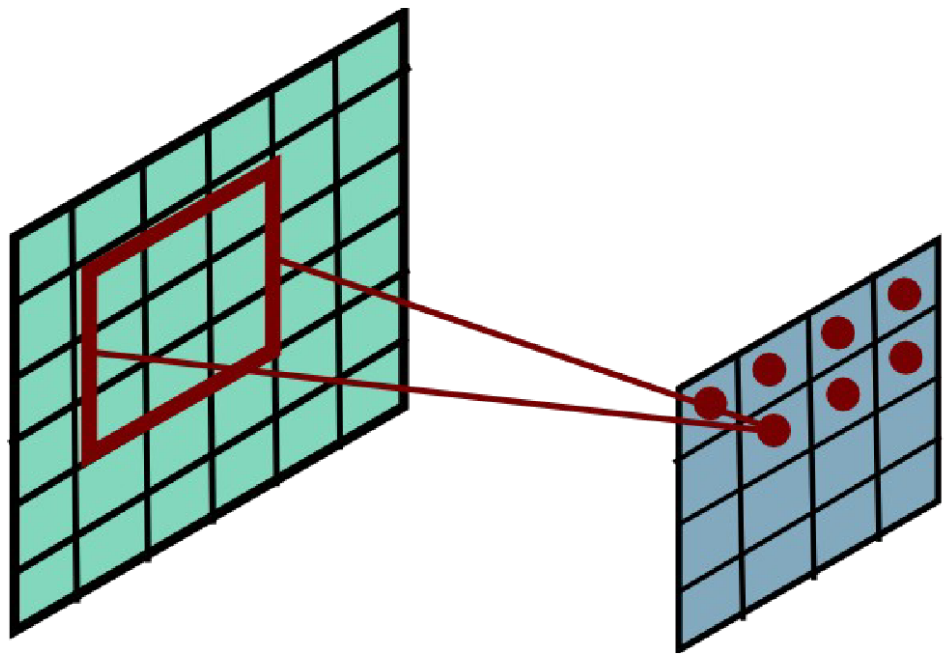
Convolutional Layer



Convolutional Layer

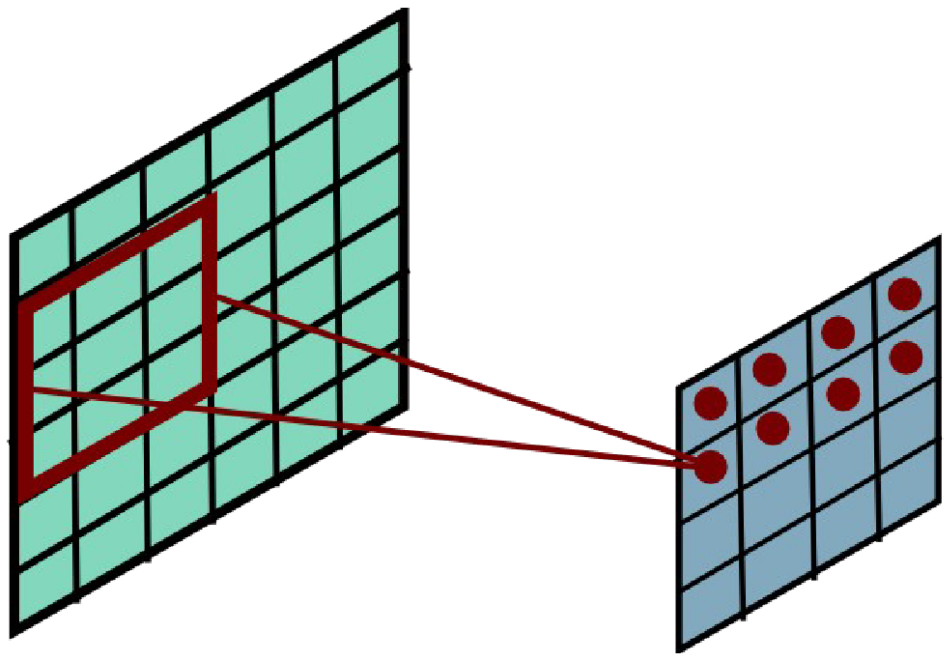


Convolutional Layer

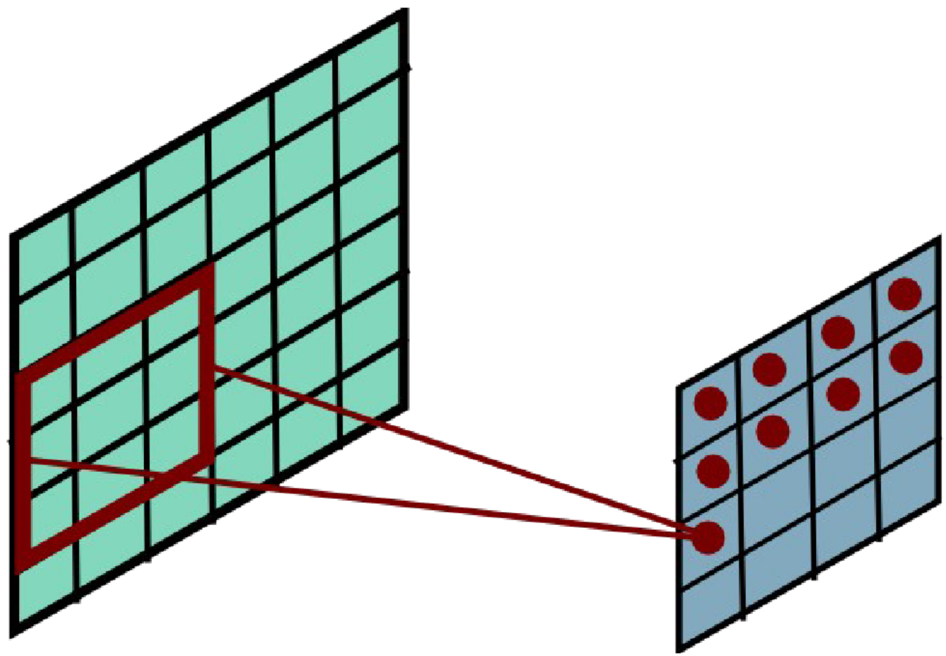


Ranzato

Convolutional Layer

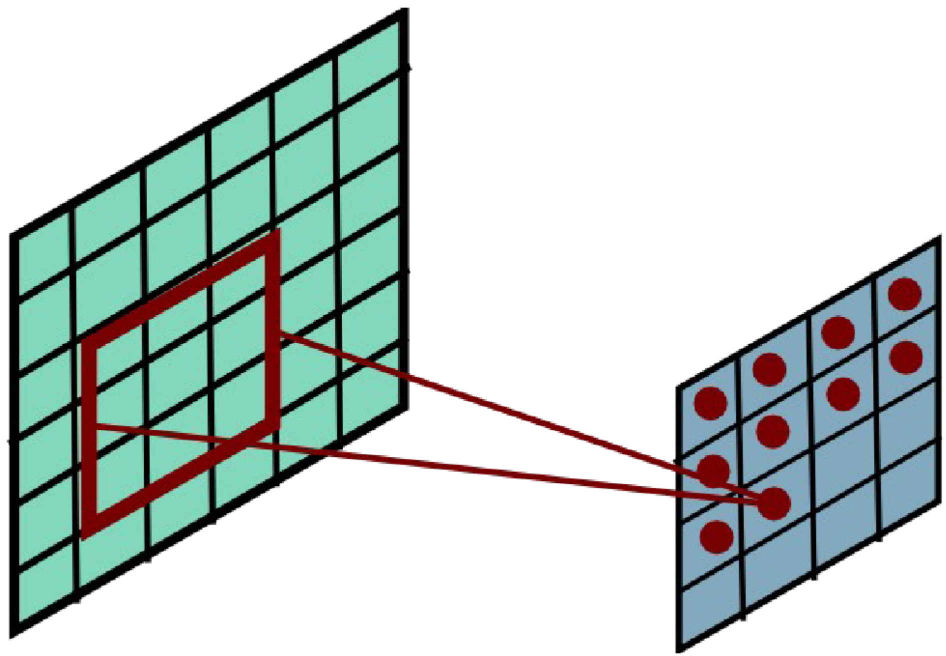


Convolutional Layer



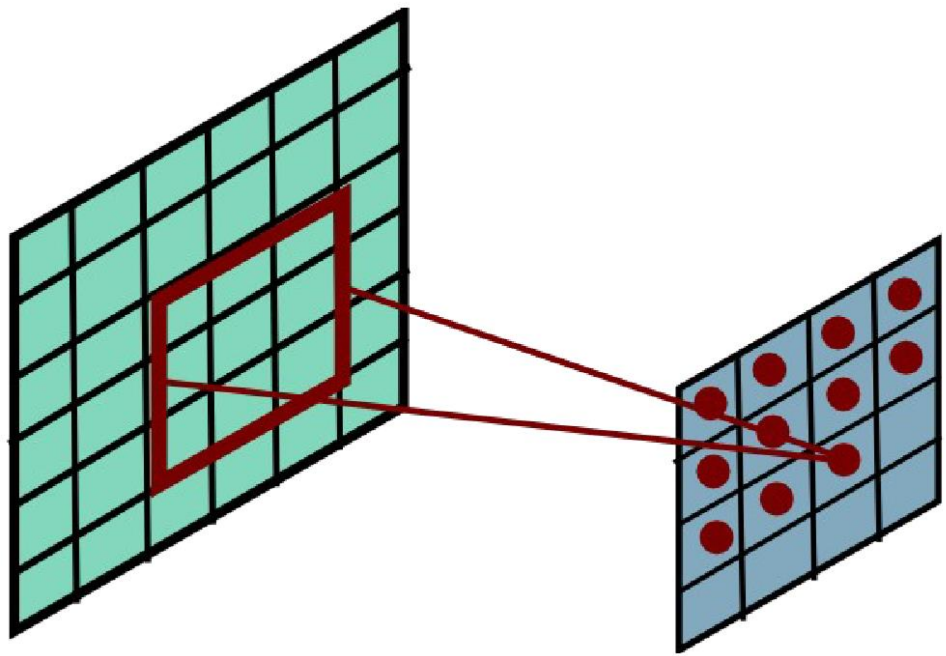
Ranzato

Convolutional Layer

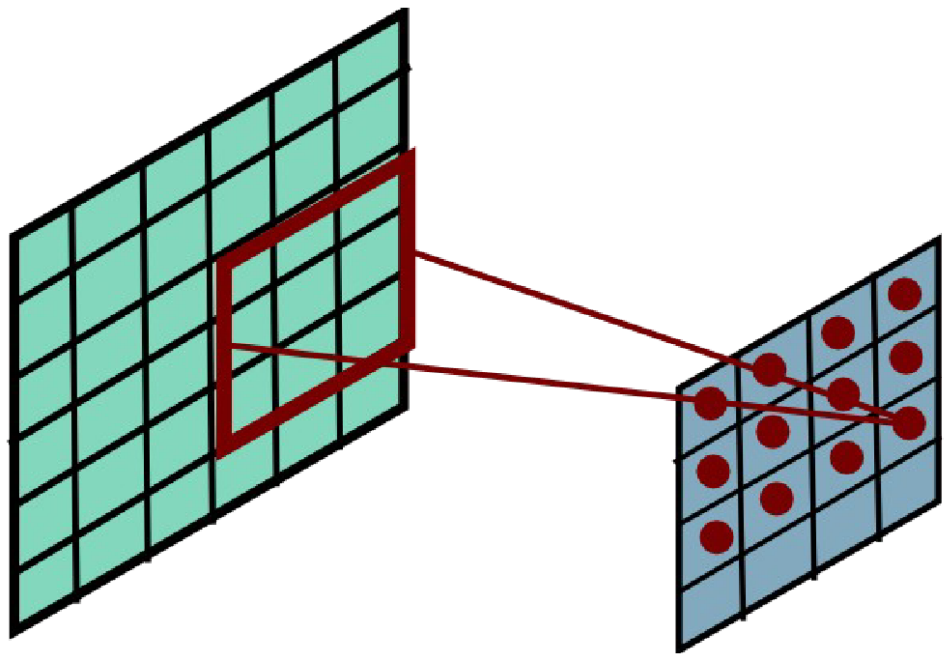


Ranzato

Convolutional Layer

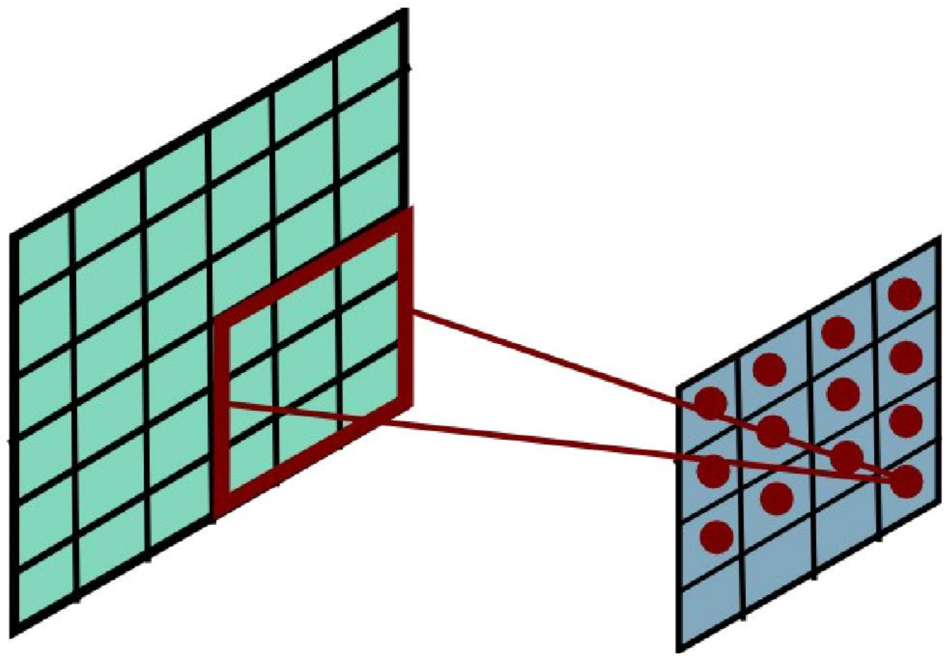


Convolutional Layer



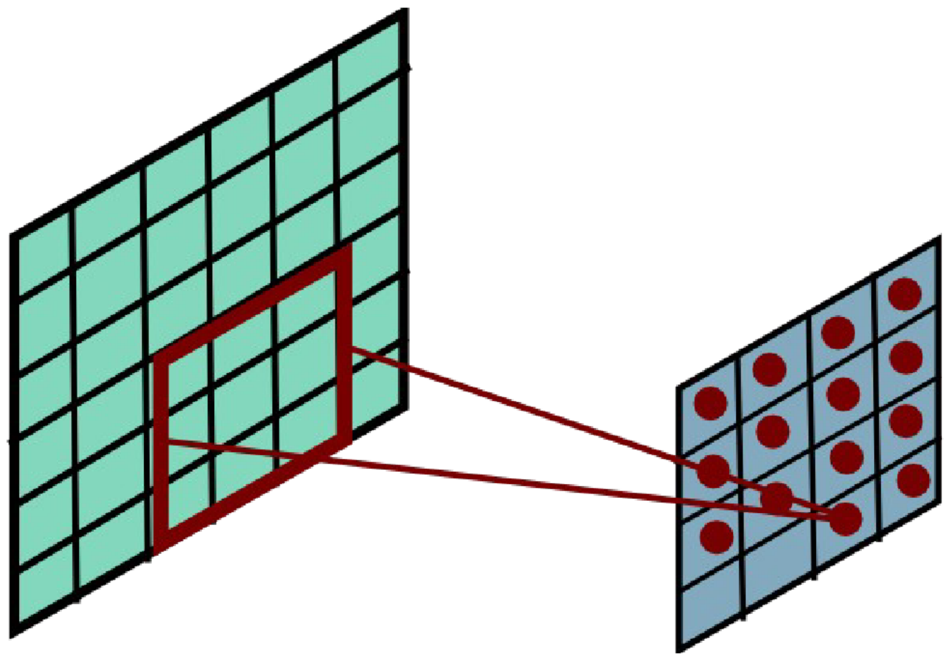
Ranzato

Convolutional Layer



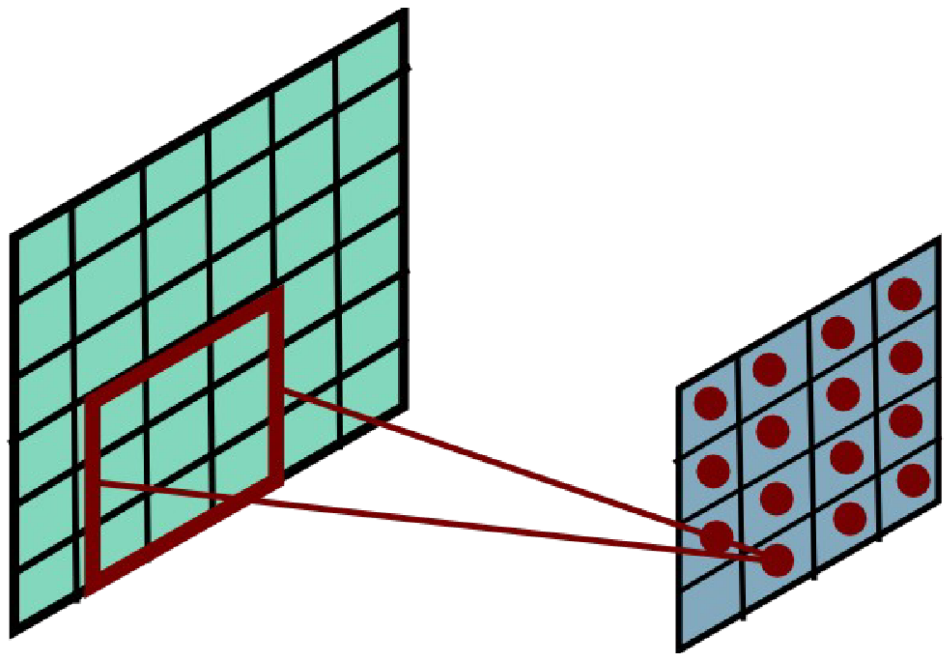
Ranzato

Convolutional Layer

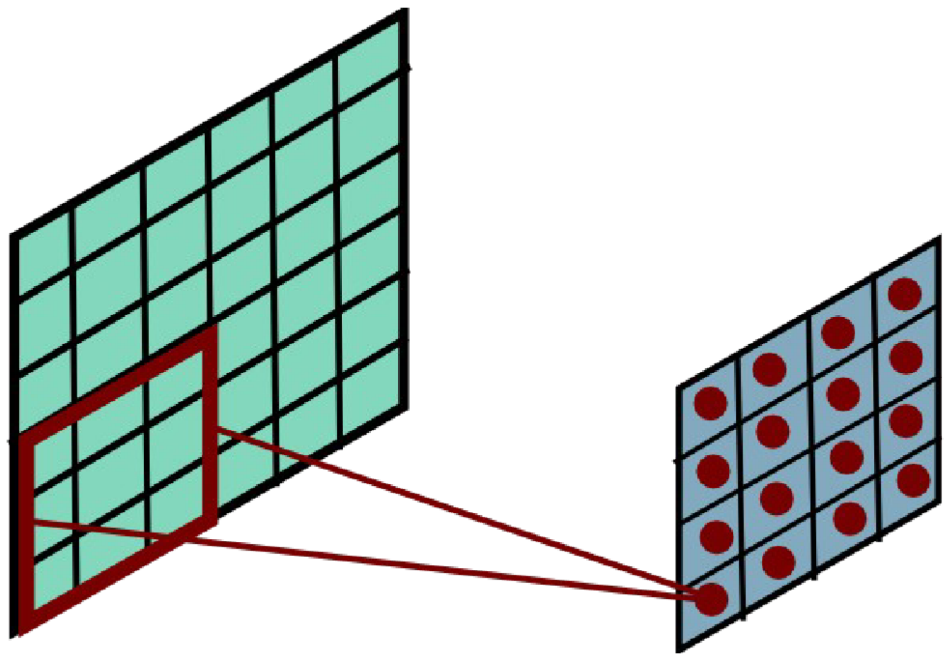


Ranzato

Convolutional Layer

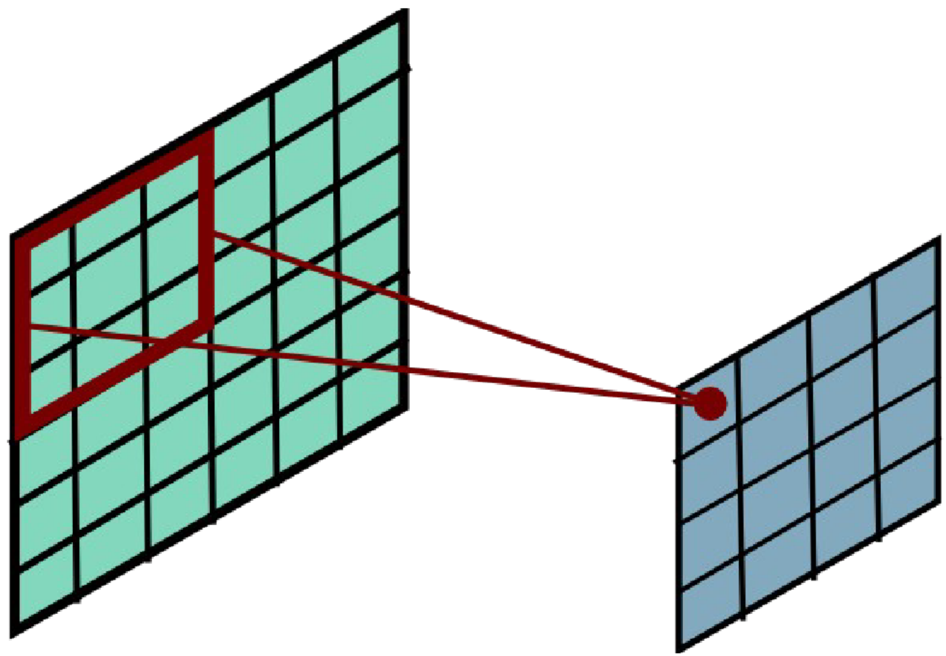


Convolutional Layer

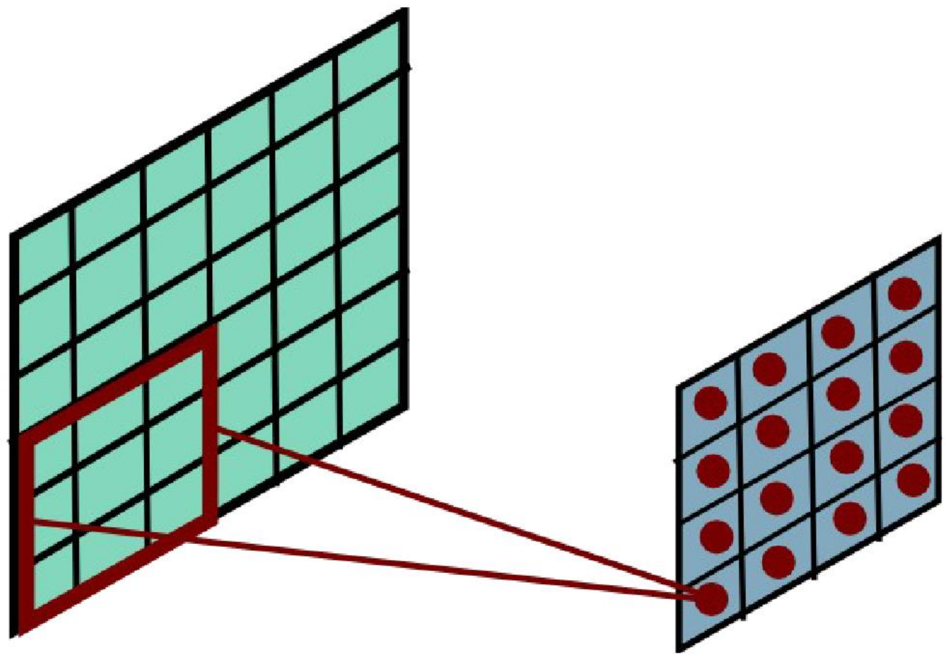


Ranzato

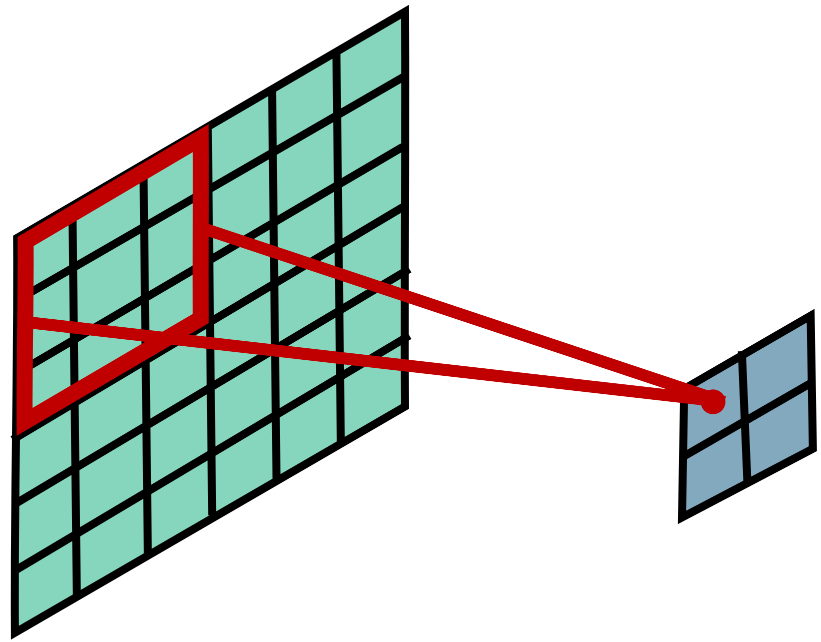
Stride = 1



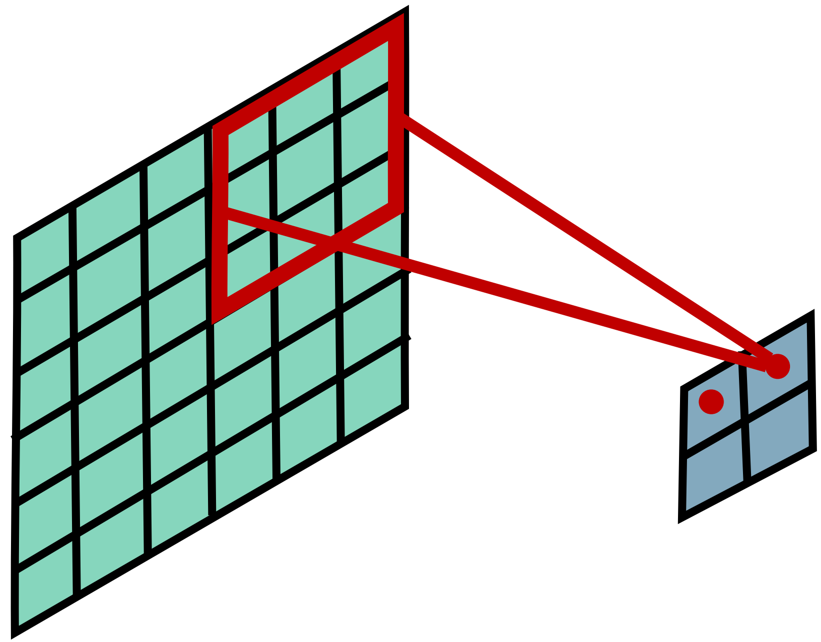
Stride = 1



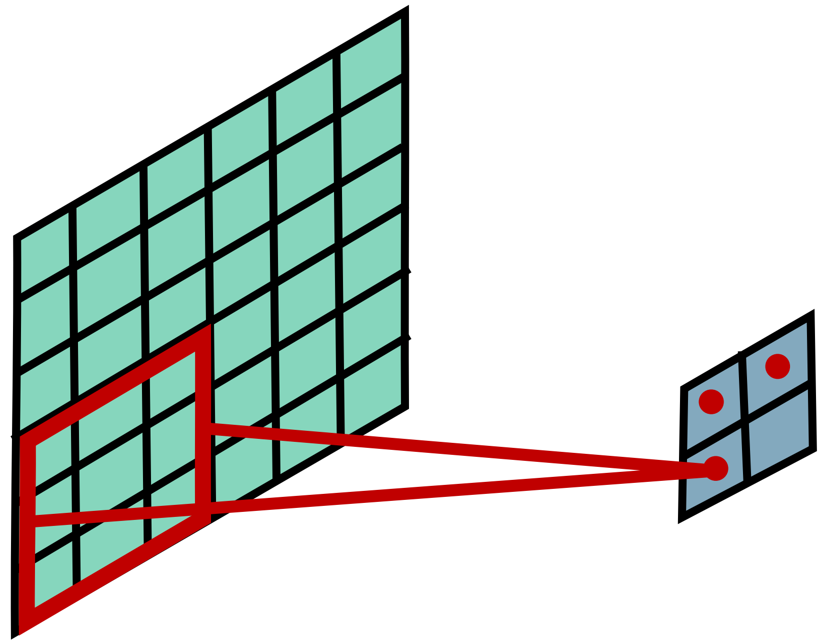
Stride = 3



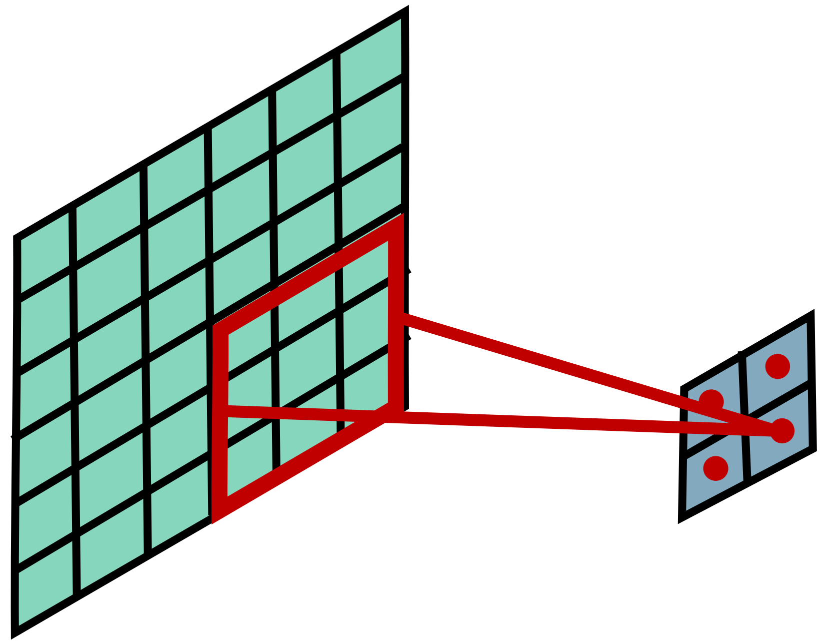
Stride = 3



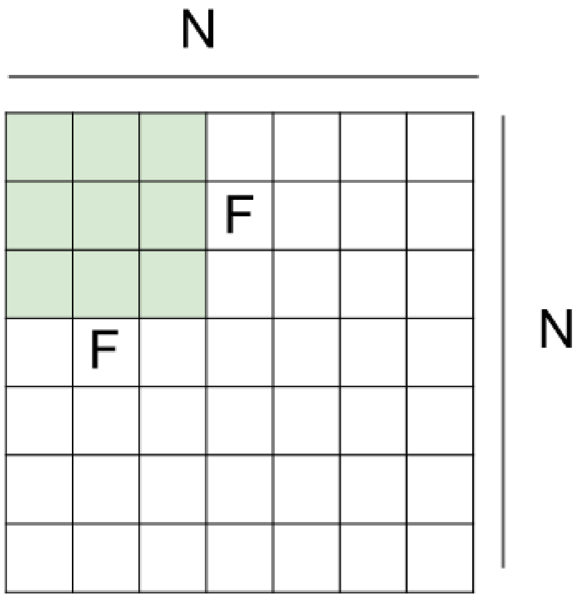
Stride = 3



Stride = 3



Some boundary consideration …

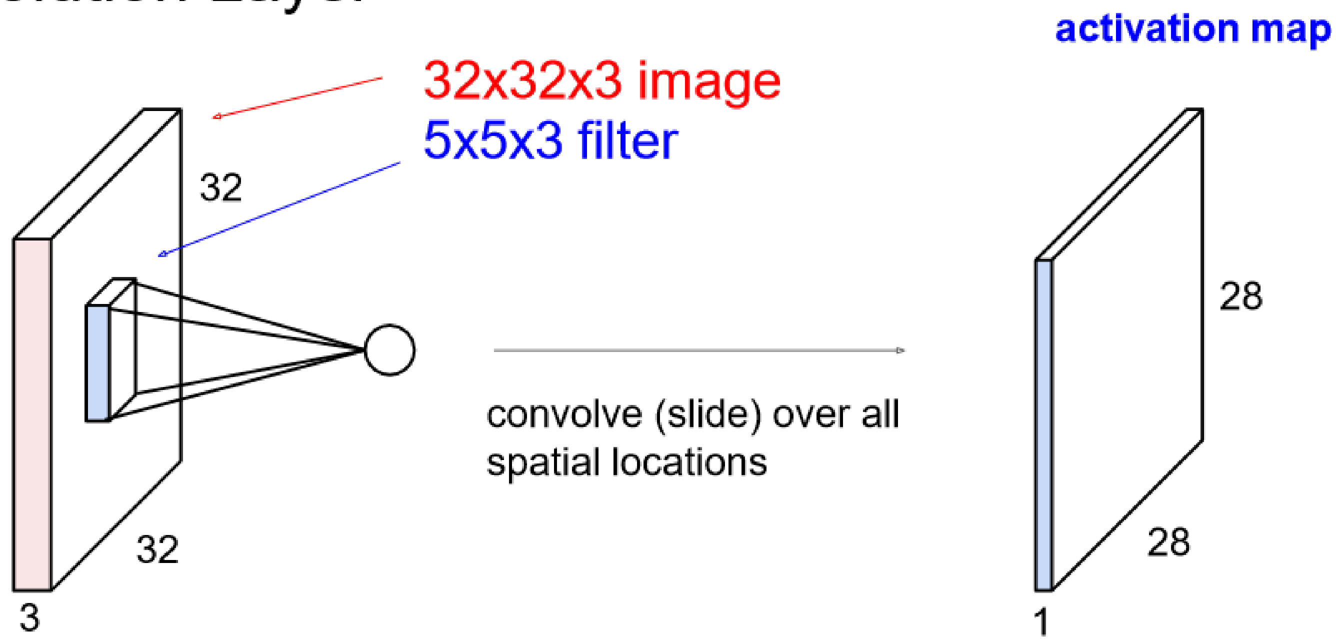


Output size:

(N - F) / stride + 1

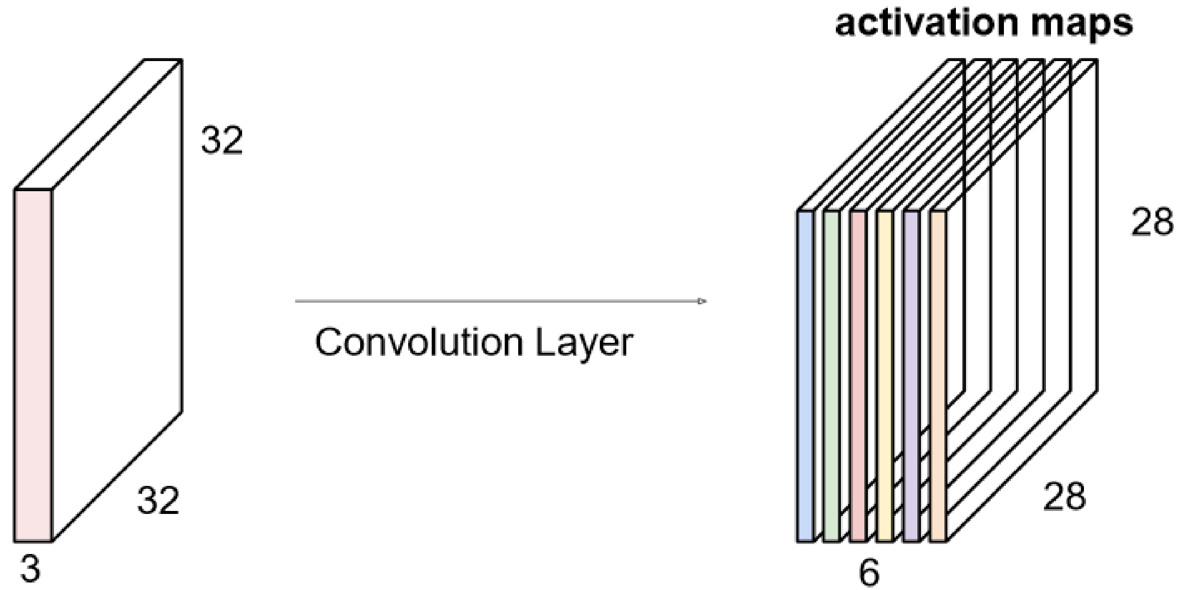
Single filter …

Convolution Layer



Multiple filters …

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

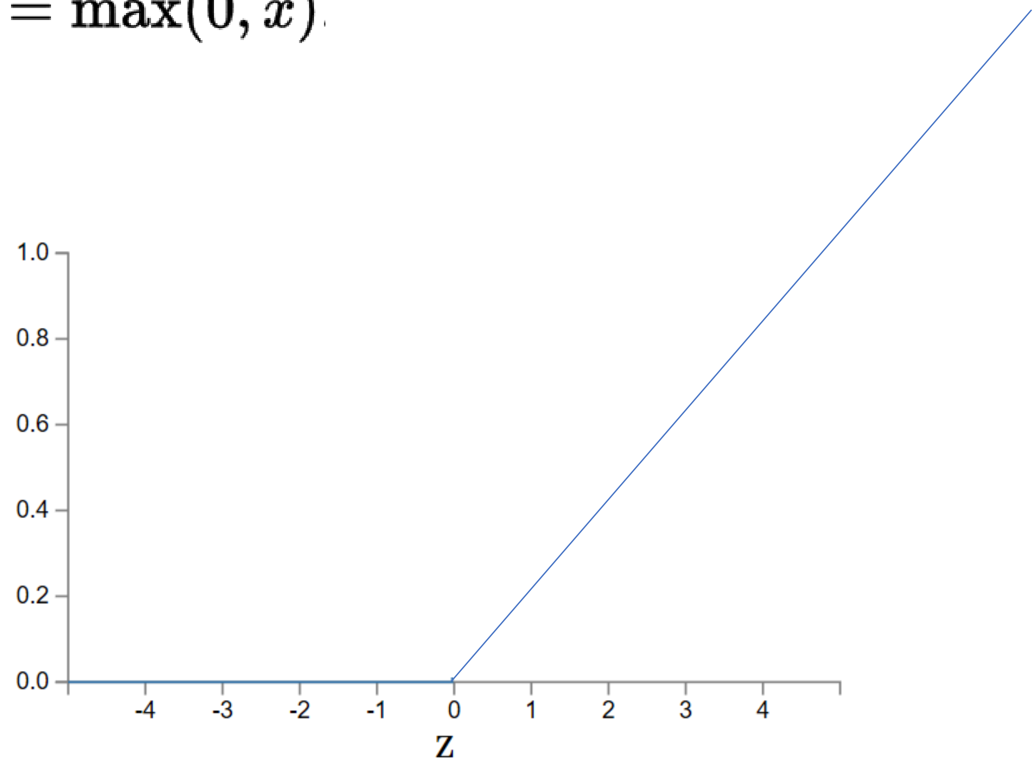


We stack these up to get a "new image" of size 28x28x6!

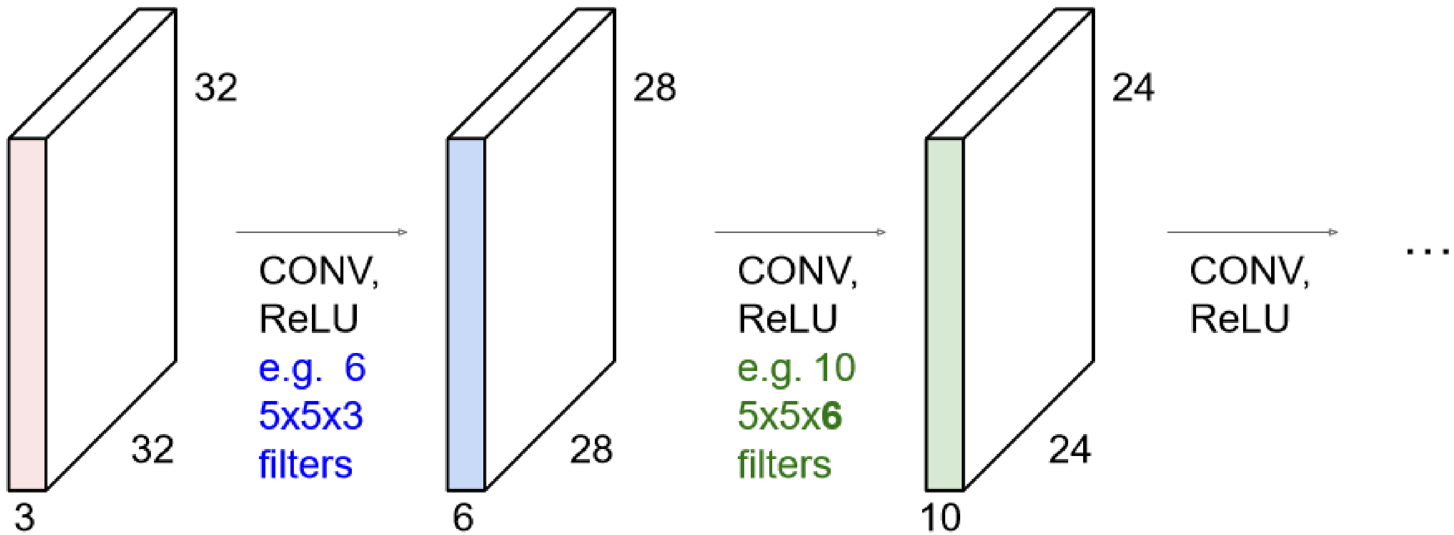
Andrej Karpathy

A common activation function: Rectified Linear Unit

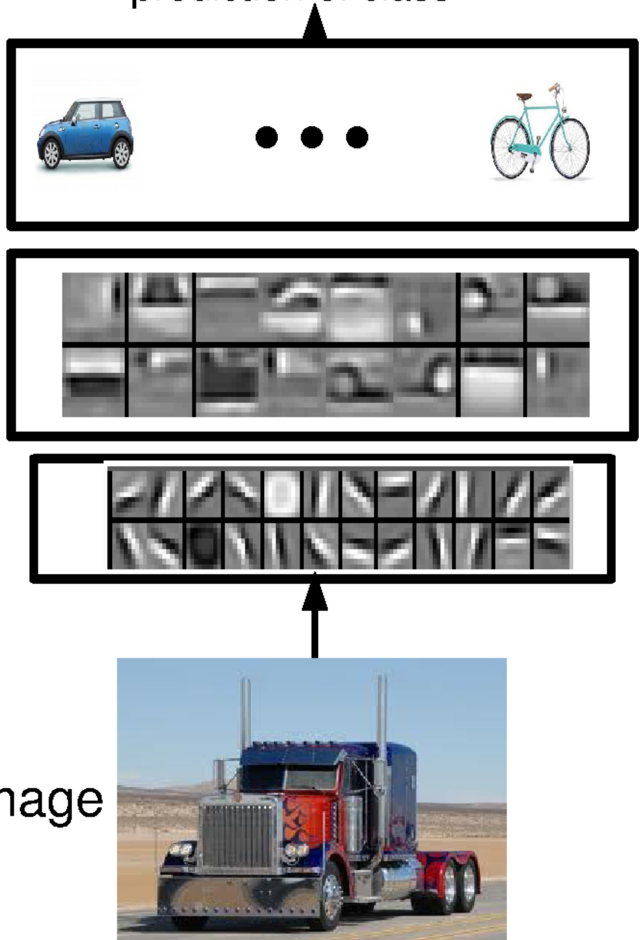
- ReLU $$f(x) = \max(0, x)$$



Stacking conv layers …



Interpretation

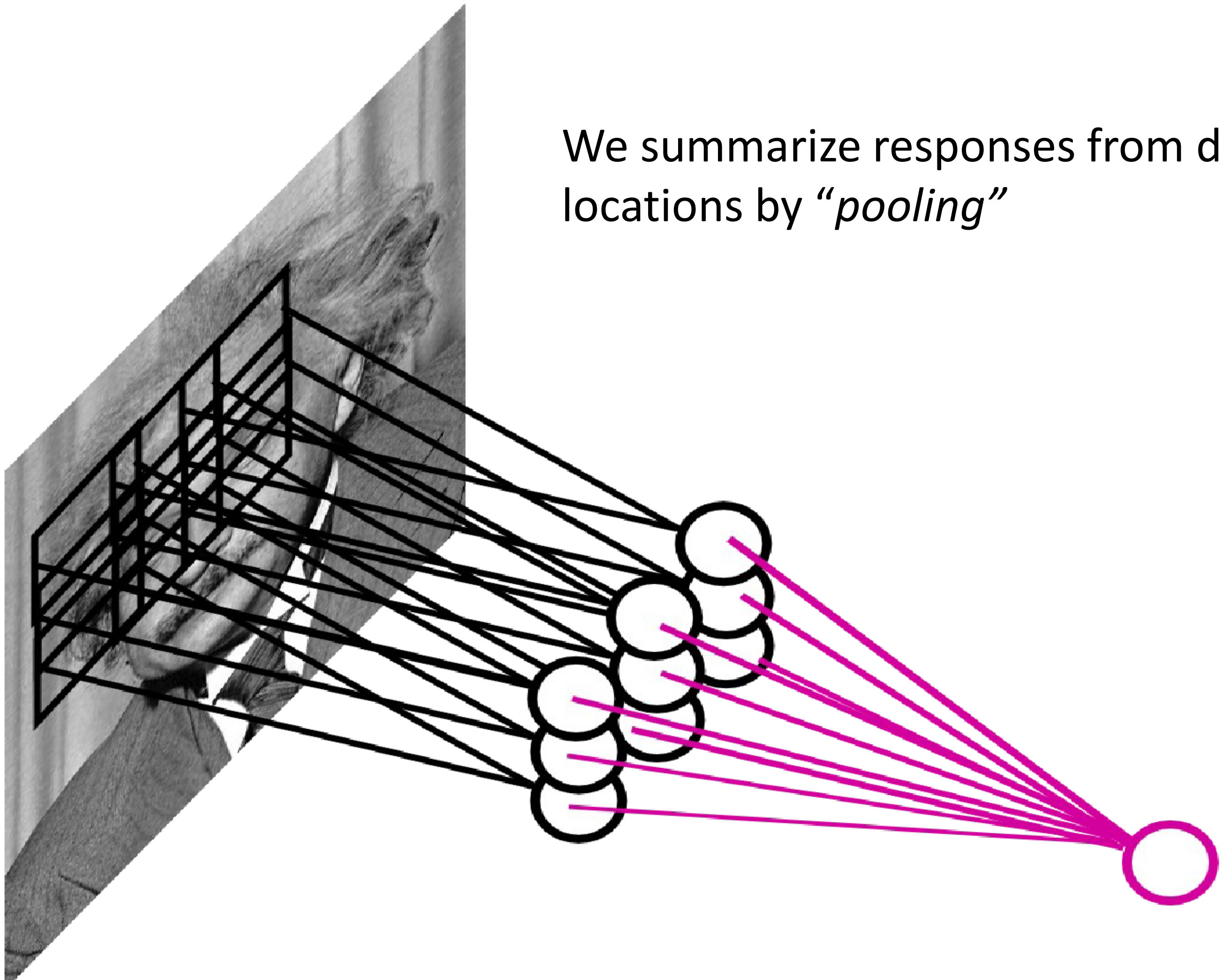


- distributed representations
- feature sharing
- compositionality

Lee et al. "Convolutional DBN's ..." ICML 2009

Ranzato

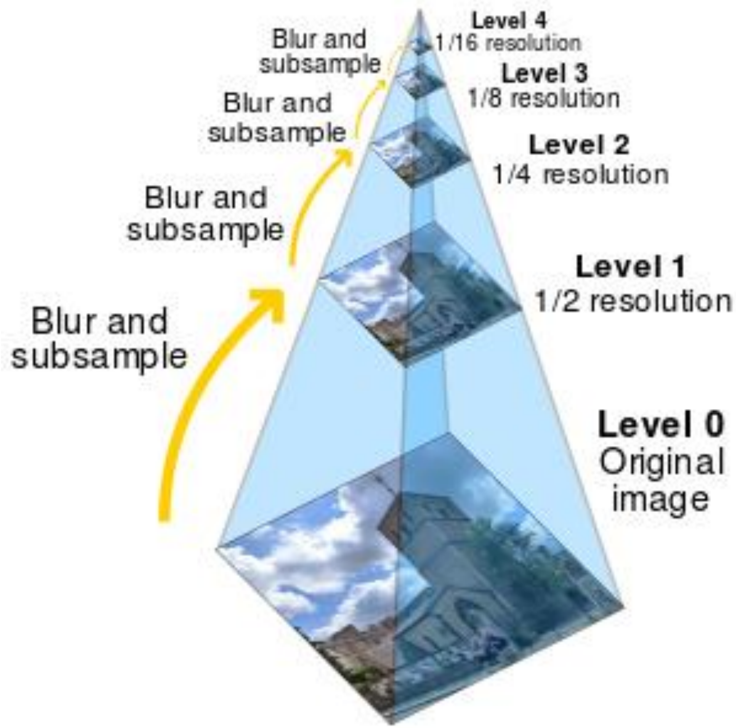
Pooling Layer



We summarize responses from different locations by “*pooling*”

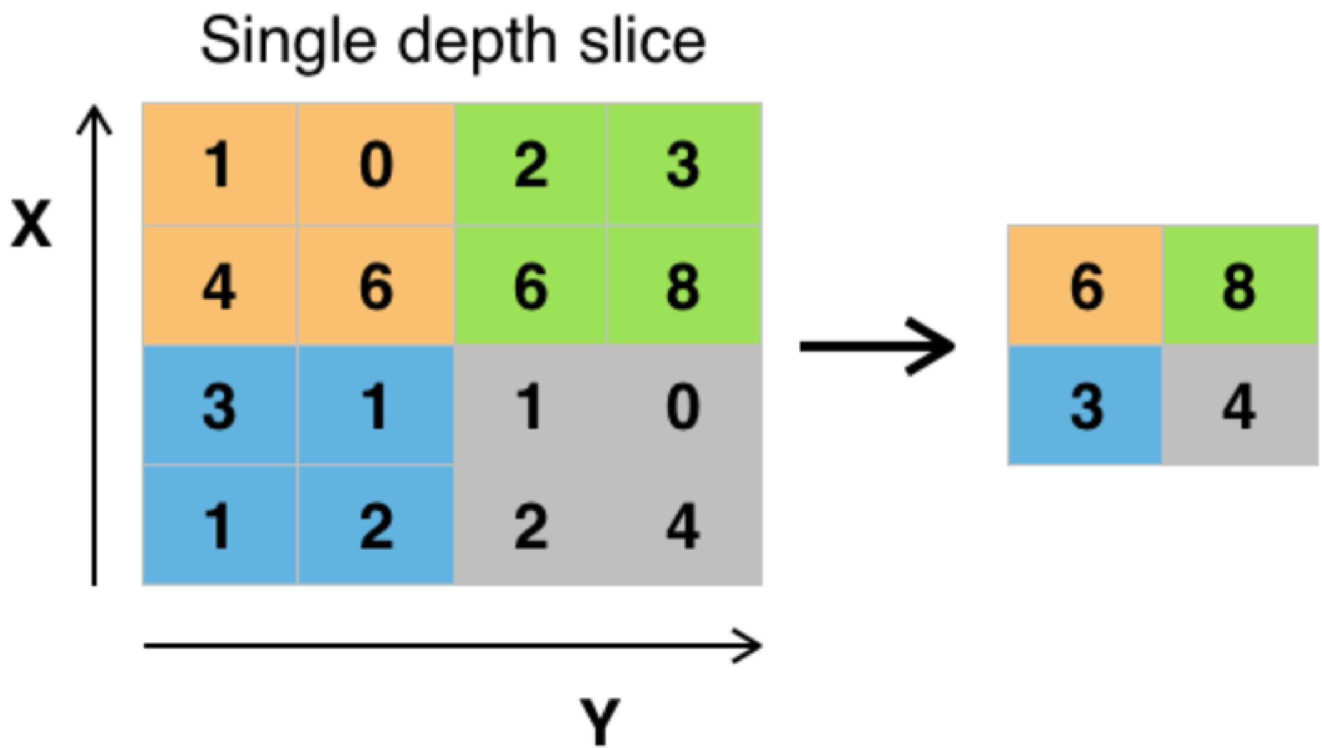
Ranzato

Pooling is similar to downsampling



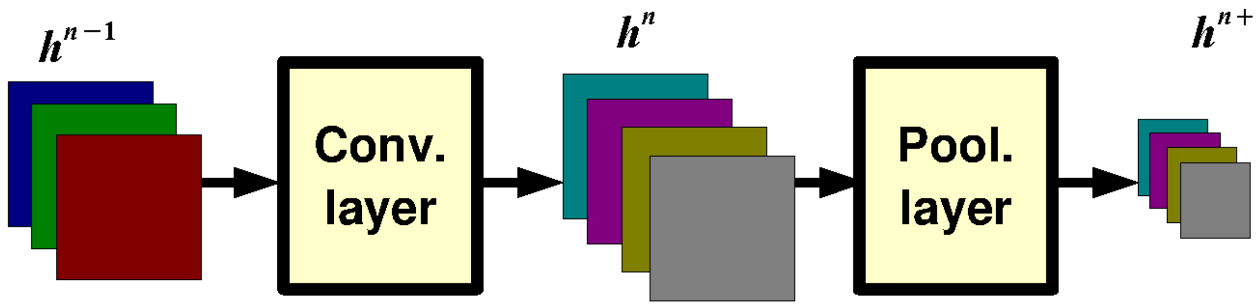
…except sometimes we don’t want to blur,
as other functions might be better for classification.

Max pooling

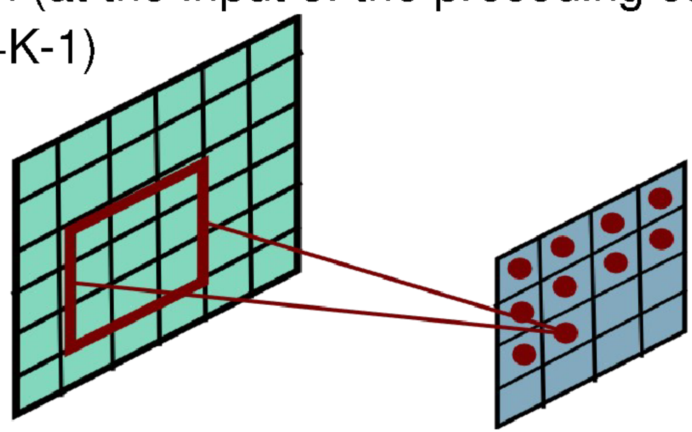


Wikipedia

Pooling Layer: Receptive Field Size

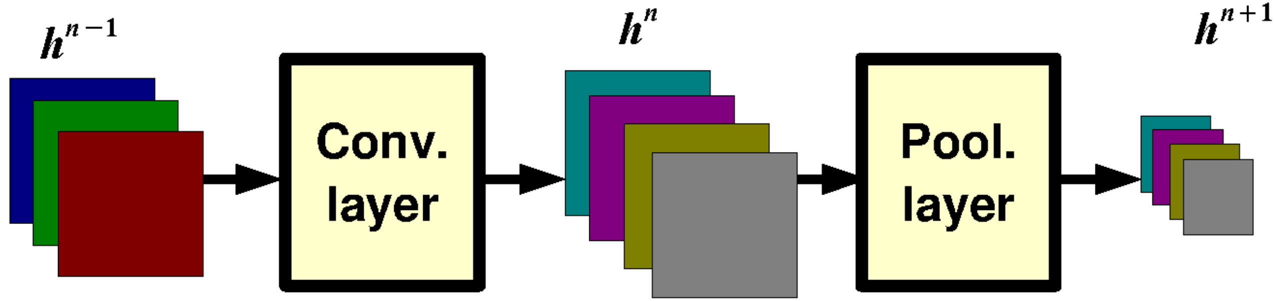


If convolutional filters have size KxK and stride 1, and pooling layer has pools of size PxP, then each unit in the pooling layer depends upon a patch (at the input of the preceding conv. layer) of size: (P+K-1)x(P+K-1)

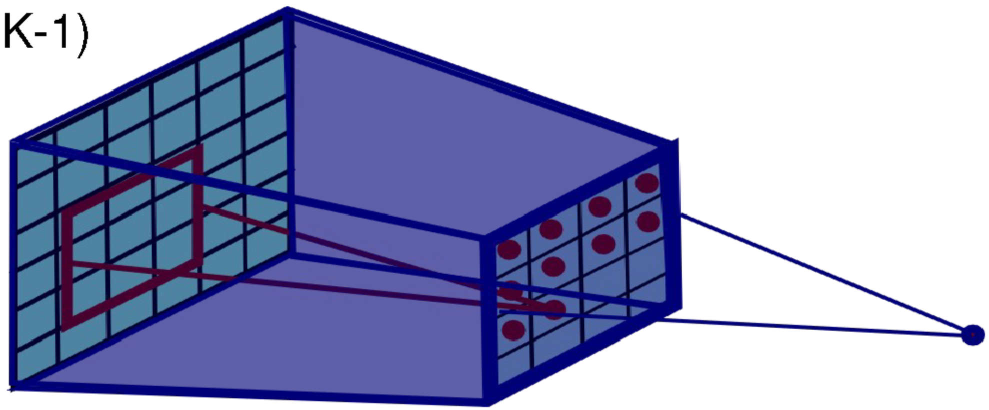


Ranzato

Pooling Layer: Receptive Field Size

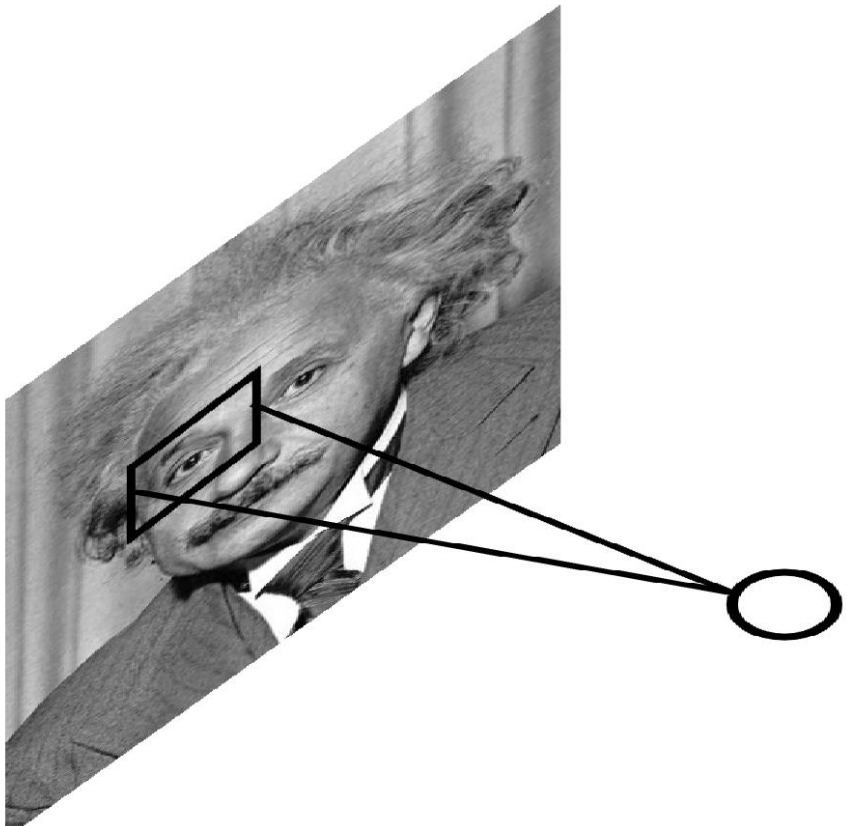


If convolutional filters have size KxK and stride 1, and pooling layer has pools of size PxP, then each unit in the pooling layer depends upon a patch (at the input of the preceding conv. layer) of size: (P+K-1)x(P+K-1)

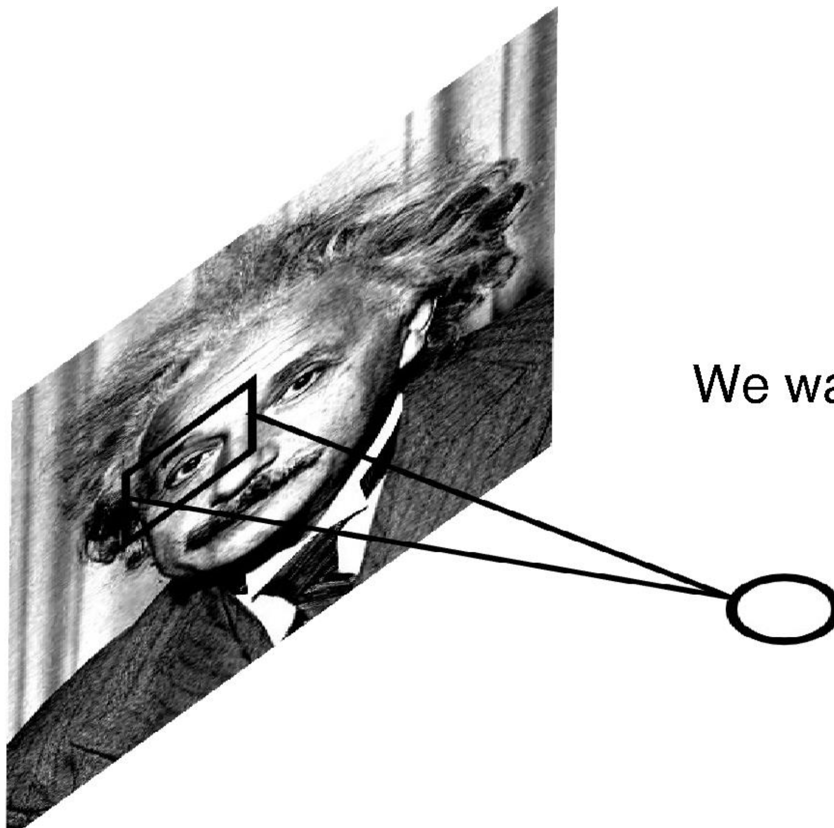


Ranzato

Local Contrast Normalization



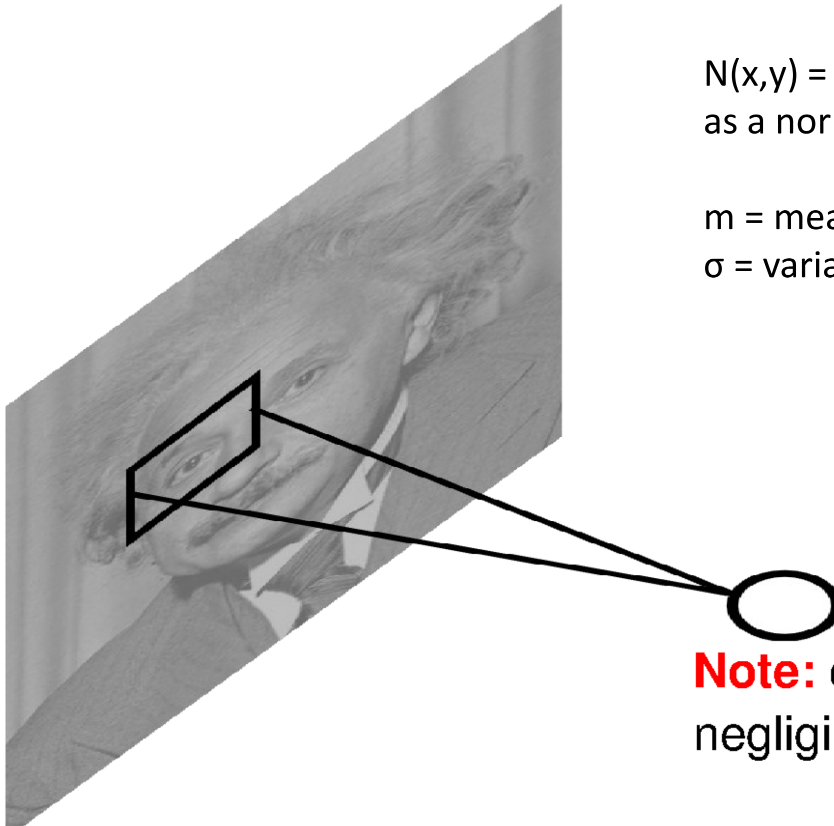
Local Contrast Normalization



We want the same response.

Local Contrast Normalization

$$h^{i+1}(x,y) = \frac{h^i(x,y) - m^i(N(x,y))}{\sigma^i(N(x,y))}$$



N(x,y) = model pixel values in window as a normal distribution

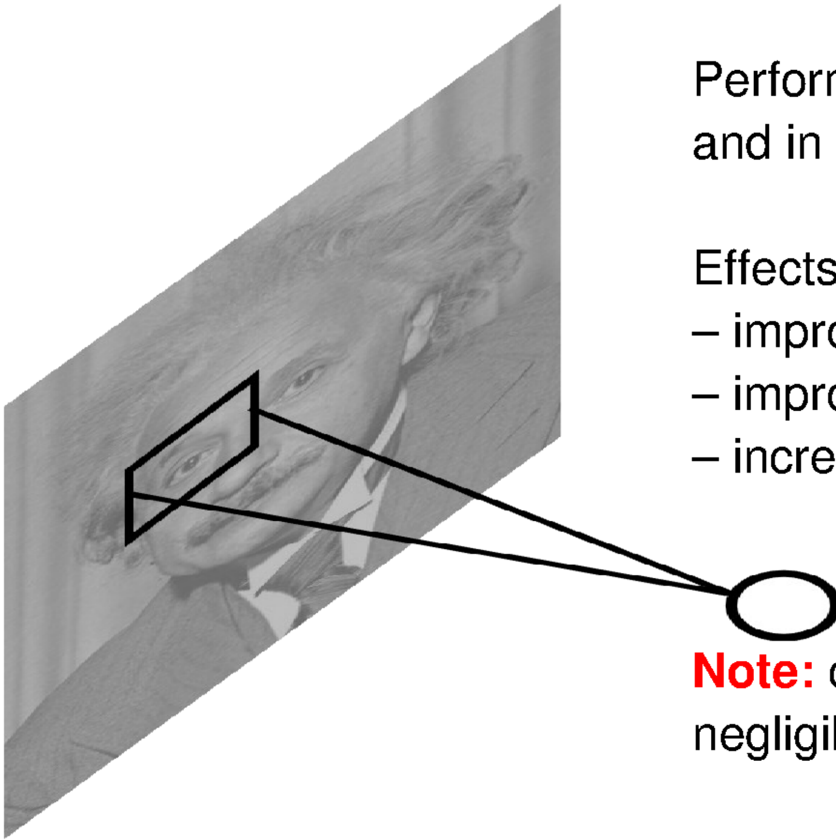
m = mean
σ = variance

Note: computational cost is negligible w.r.t. conv. layer.

Ranzato

Local Contrast Normalization

$$h^{i+1}(x,y) = \frac{h^i(x,y) - m^i(N(x,y))}{\sigma^i(N(x,y))}$$

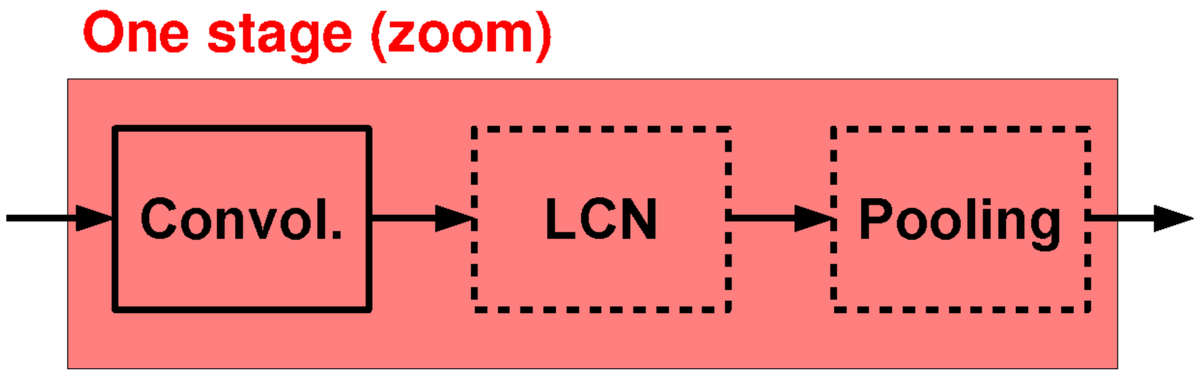


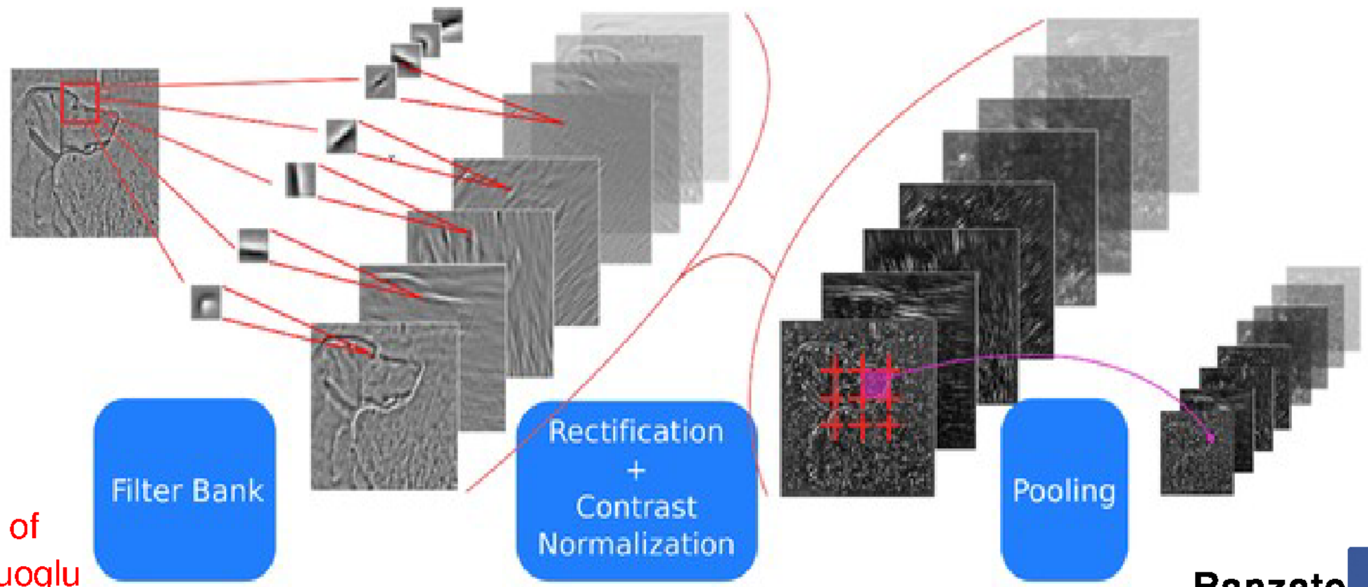
Performed also across features and in the higher layers..

Effects:
– improves invariance
– improves optimization
– increases sparsity

Note: computational cost is negligible w.r.t. conv. layer.

ConvNets: Typical Stage

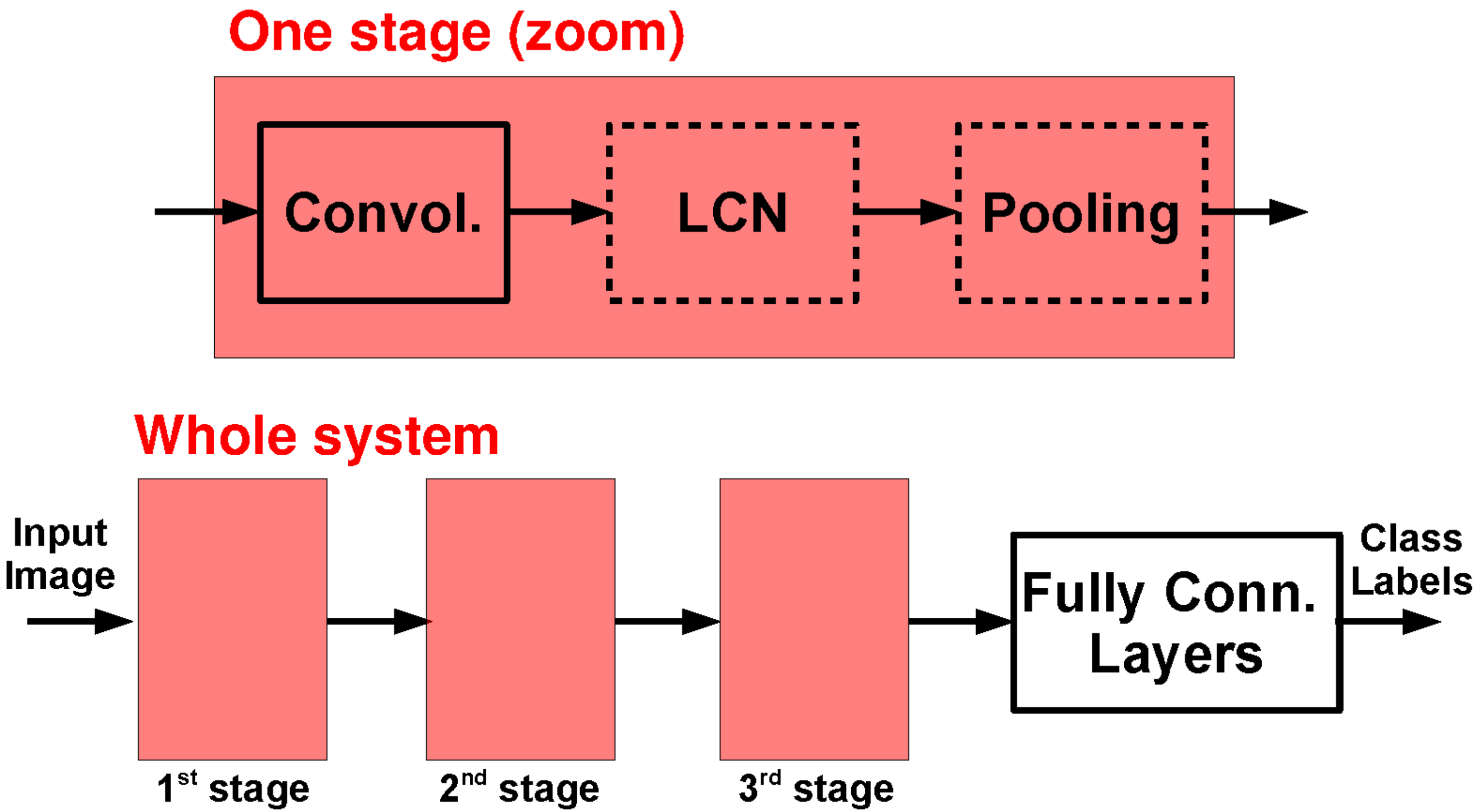




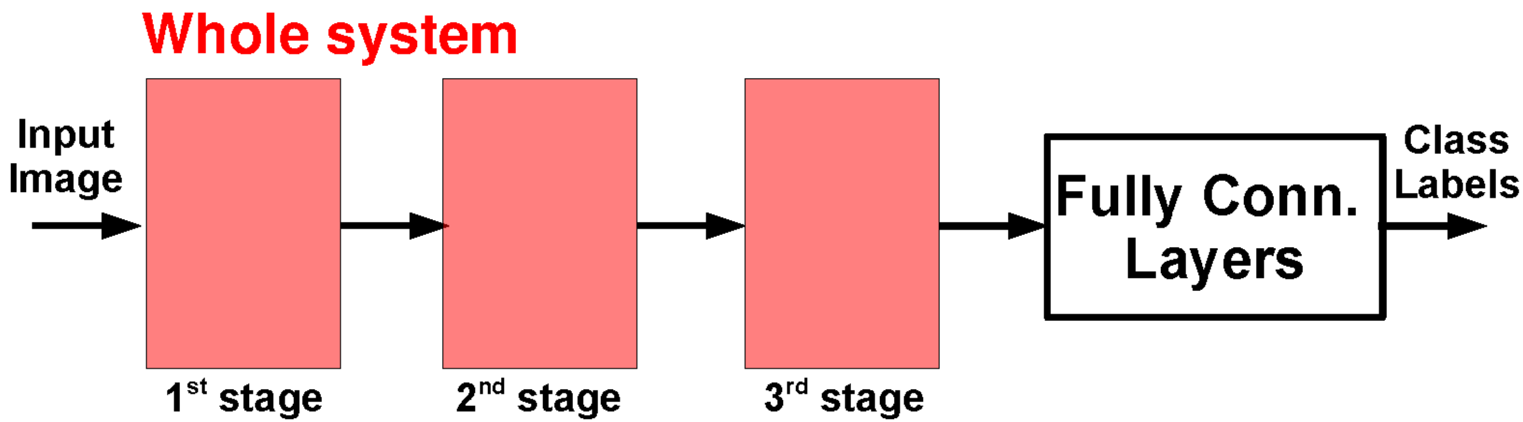
courtesy of
K. Kavukcuoglu

Ranzato

ConvNets: Typical Architecture



ConvNets: Typical Architecture



Conceptually similar to:

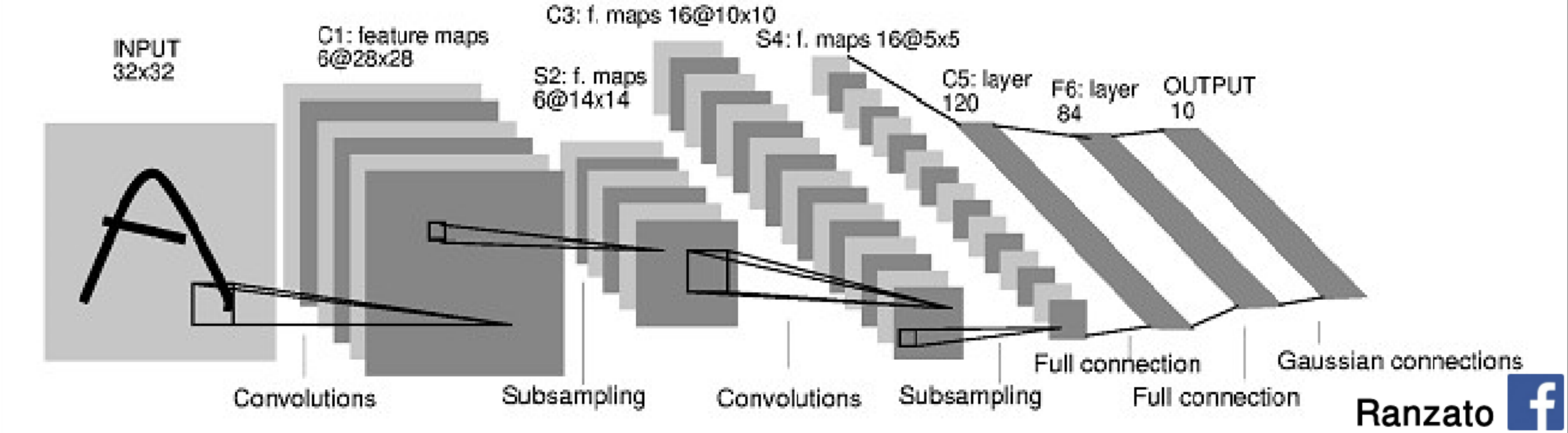
SIFT → K-Means → Pyramid Pooling → SVM

Lazebnik et al. "...Spatial Pyramid Matching..." CVPR 2006

SIFT → Fisher Vect. → Pooling → SVM

Sanchez et al. "Image classifcation with F.V.: Theory and practice" IJCV 2012

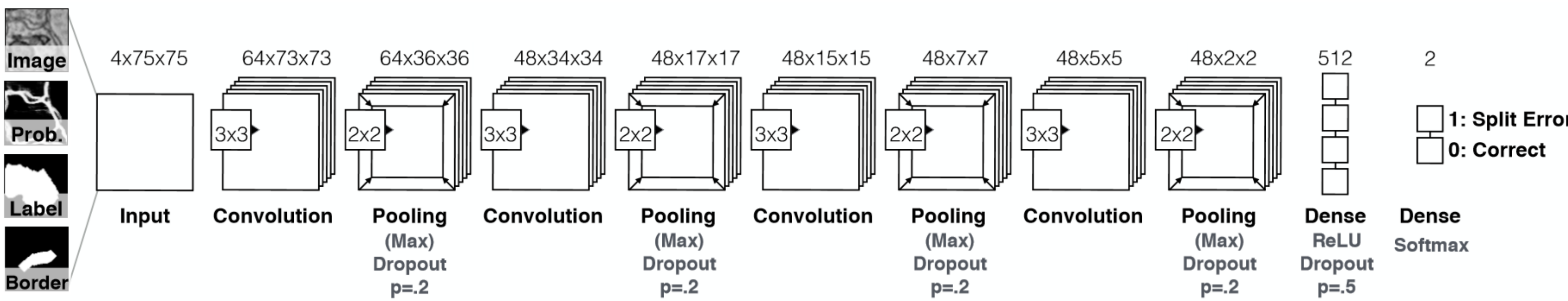
Yann LeCun’s MNIST CNN architecture



Our connectomics diagram

Auto-generated from network declaration by *nolearn* (for Lasagne / Theano)

Input
75x75x4



Conv 1
3x3x4
64 filters

Conv 2
3x3x64
48 filters

Conv 3
3x3x48
48 filters

Conv 4
3x3x48
48 filters

Max pooling
2x2 per
filter

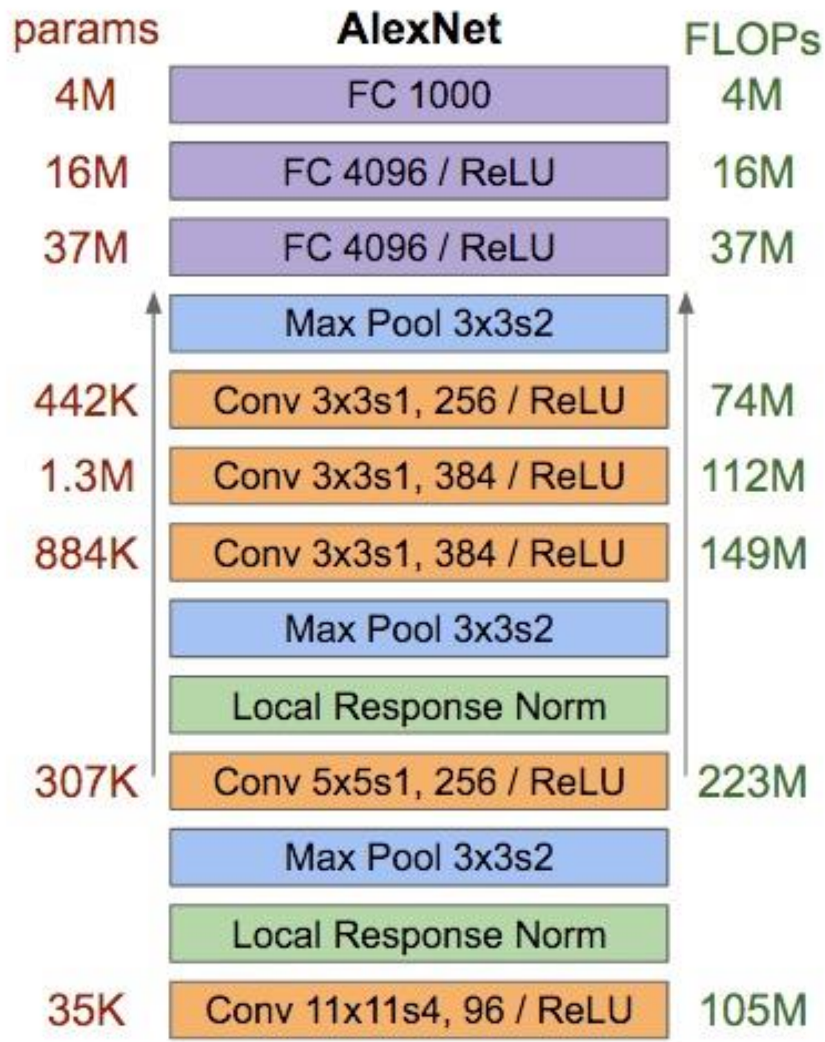
Max pooling
2x2 per
filter

Max pooling
2x2 per
filter

Max pooling
2x2 per
filter

Reading architecture diagrams

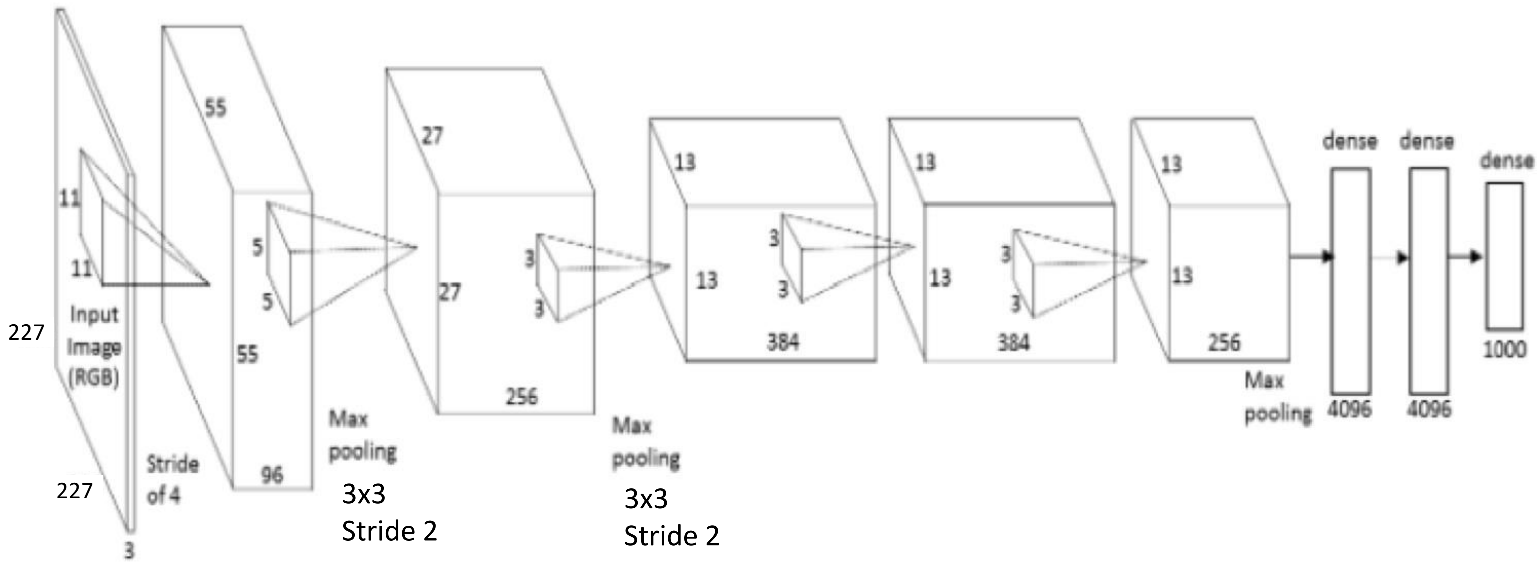
Layers
- Kernel sizes
- Strides
- # channels
- # kernels
- Max pooling



[Krizhevsky et al. 2012]

AlexNet diagram (simplified)

Input size
227 x 227 x 3



Conv 1
11 x 11 x 3
Stride 4
96 filters

Conv 2
5 x 5 x 48
Stride 1
256 filters

Conv 3
3 x 3 x 256
Stride 1
384 filters

Conv 4
3 x 3 x 192
Stride 1
384 filters

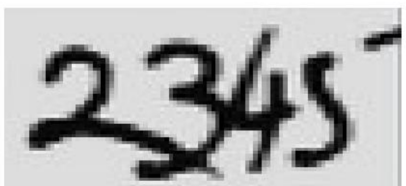
Conv 4
3 x 3 x 192
Stride 1
256 filters

Outline

- Supervised Neural Networks
- Convolutional Neural Networks
- Examples
- Tips

CONV NETS: EXAMPLES

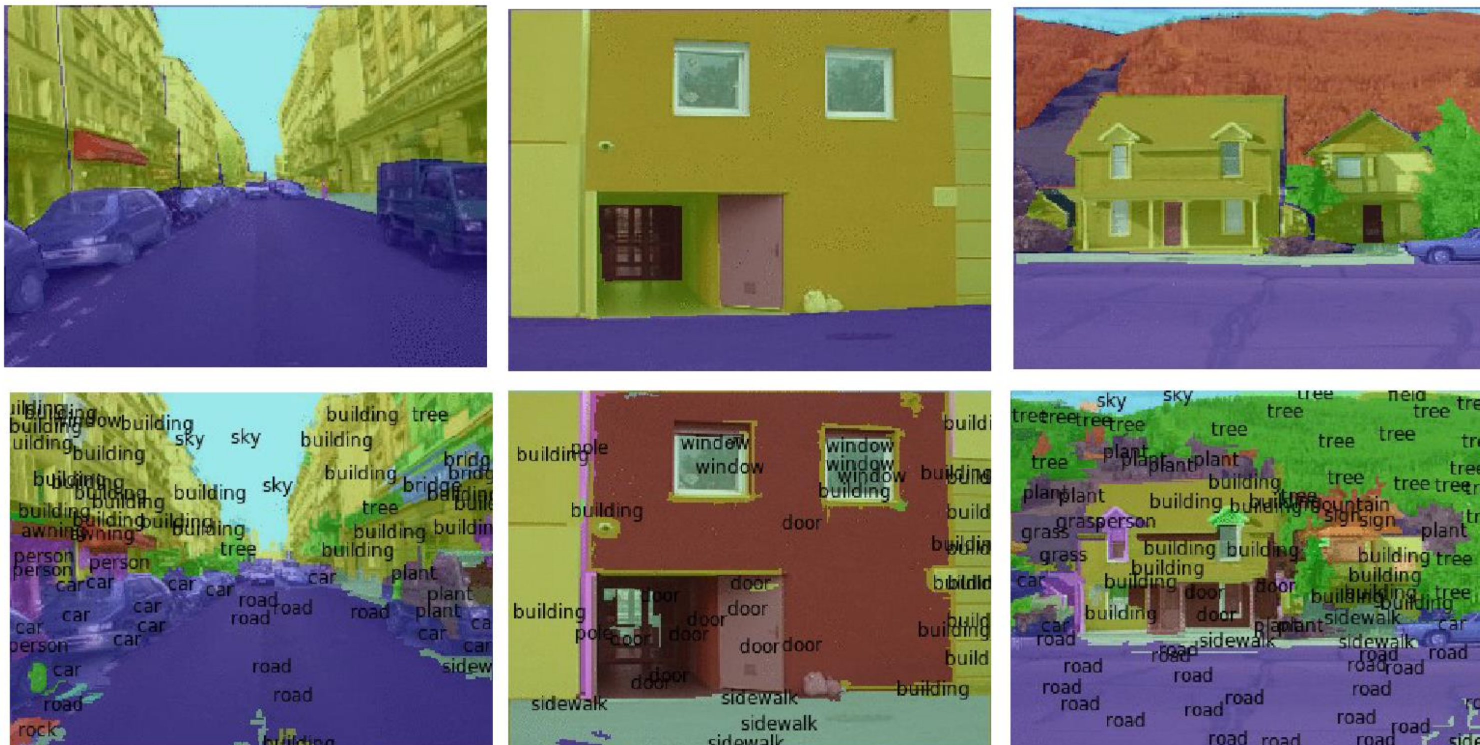
- OCR / House number & Traffic sign classification



Ciresan et al. “MCDNN for image classification” CVPR 2012
Wan et al. “Regularization of neural networks using dropconnect” ICML 2013
Jaderberg et al. “Synthetic data and ANN for natural scene text recognition” arXiv 2014

CONV NETS: EXAMPLES

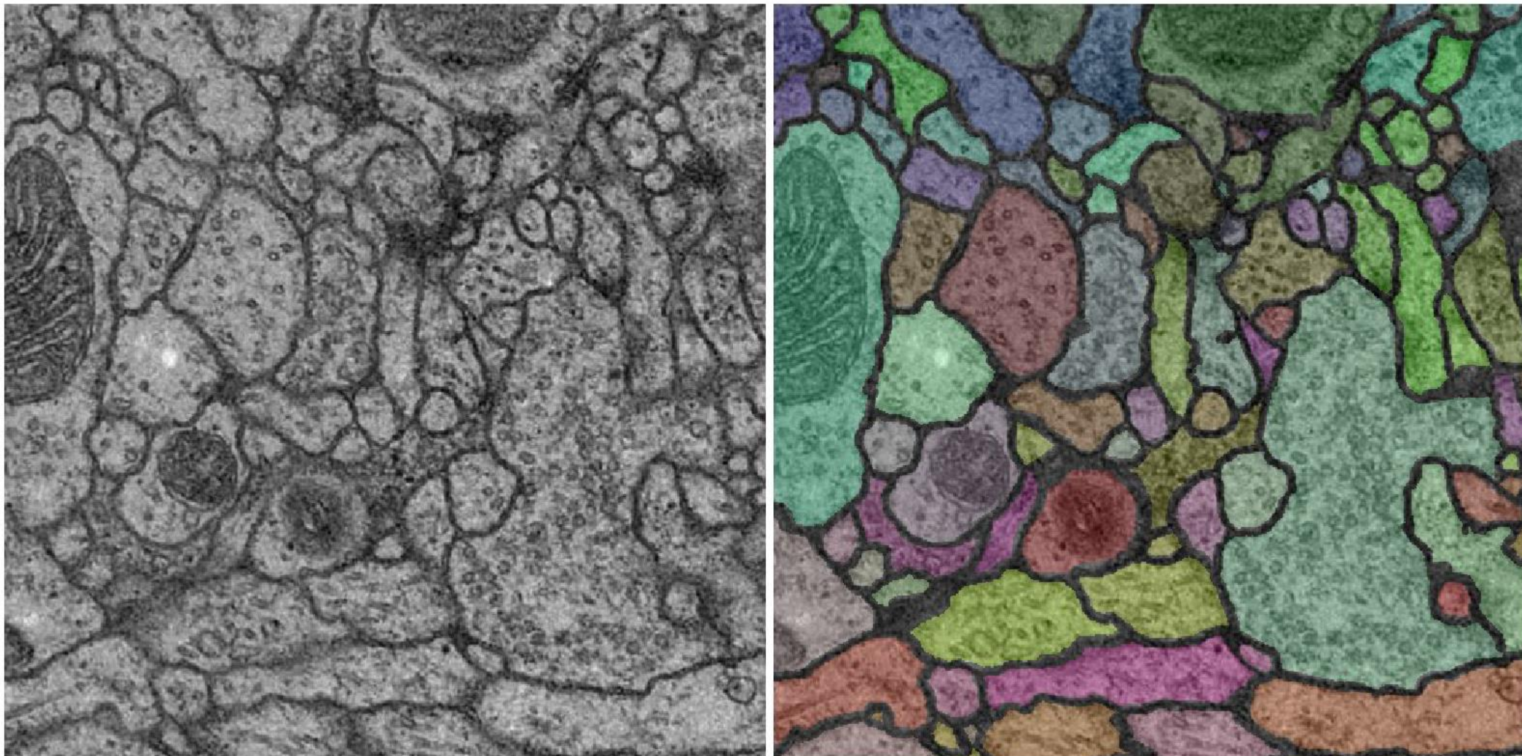
- **Scene Parsing**



Farabet et al. "Learning hierarchical features for scene labeling" PAMI 2013
Pinheiro et al. "Recurrent CNN for scene parsing" arxiv 2013

CONV NETS: EXAMPLES

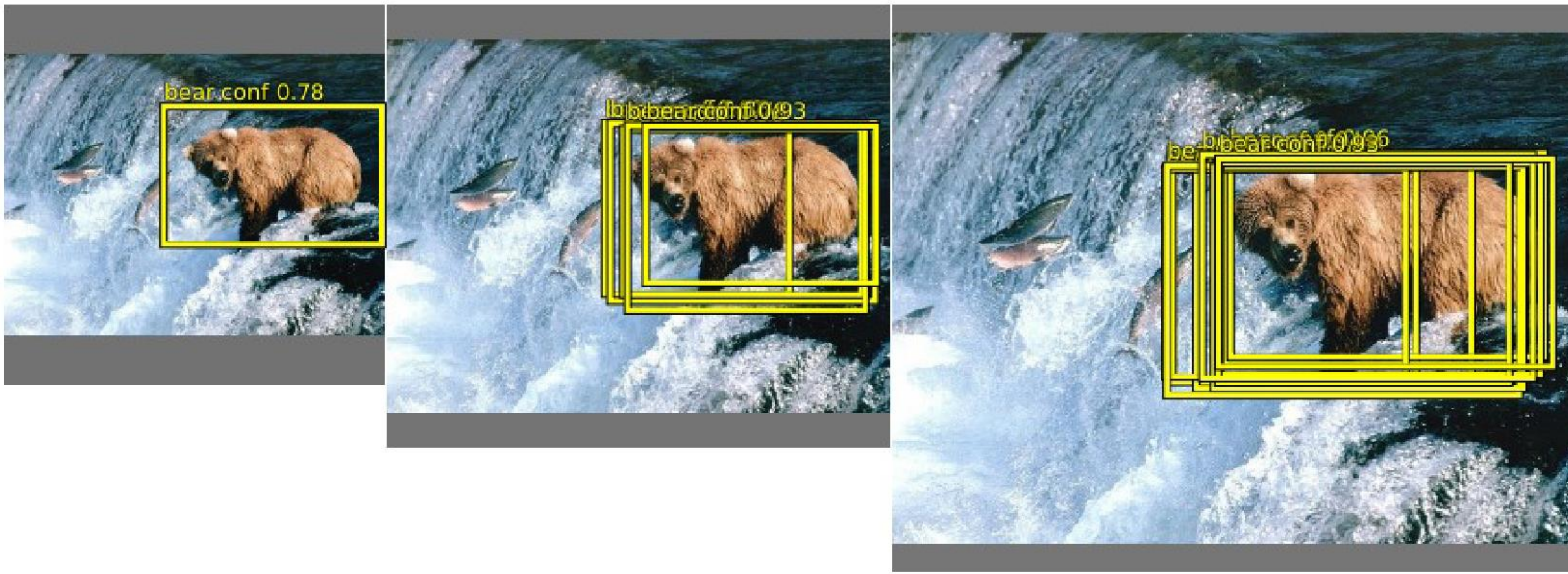
- **Segmentation 3D volumetric images**



Ciresan et al. “DNN segment neuronal membranes...” NIPS 2012
Turaga et al. “Maximin learning of image segmentation” NIPS 2009

CONV NETS: EXAMPLES

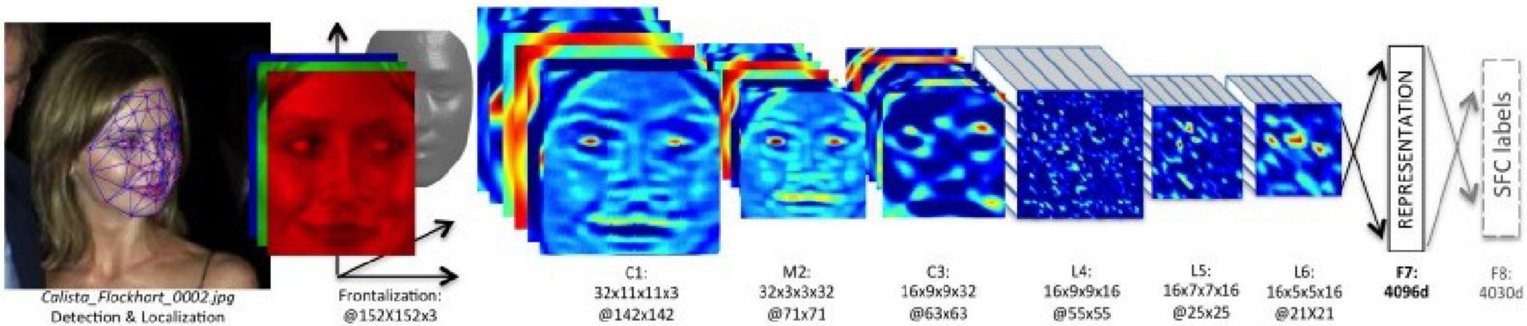
- **Object detection**



Sermanet et al. “OverFeat: Integrated recognition, localization, ...” arxiv 2013
Girshick et al. “Rich feature hierarchies for accurate object detection...” arxiv 2013
Szegedy et al. “DNN for object detection” NIPS 2013

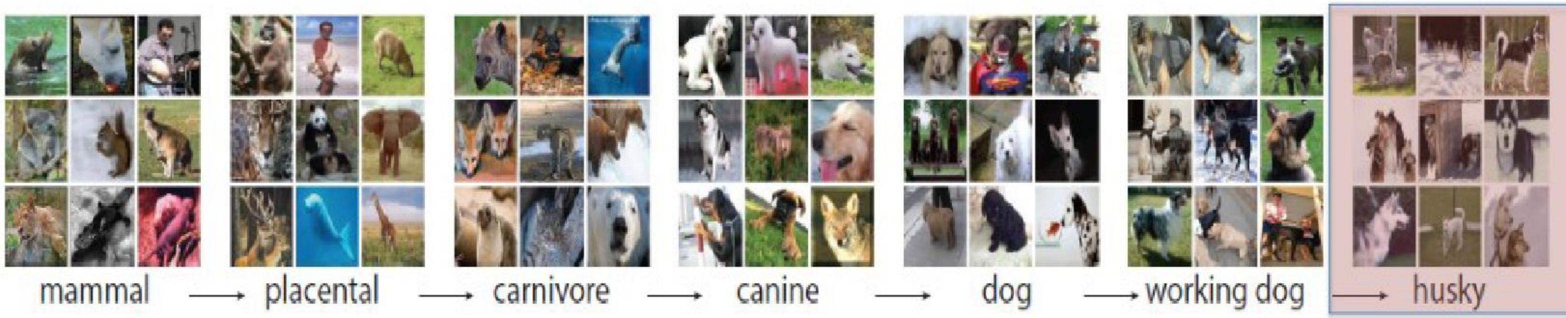
CONV NETS: EXAMPLES

- **Face Verification & Identification**



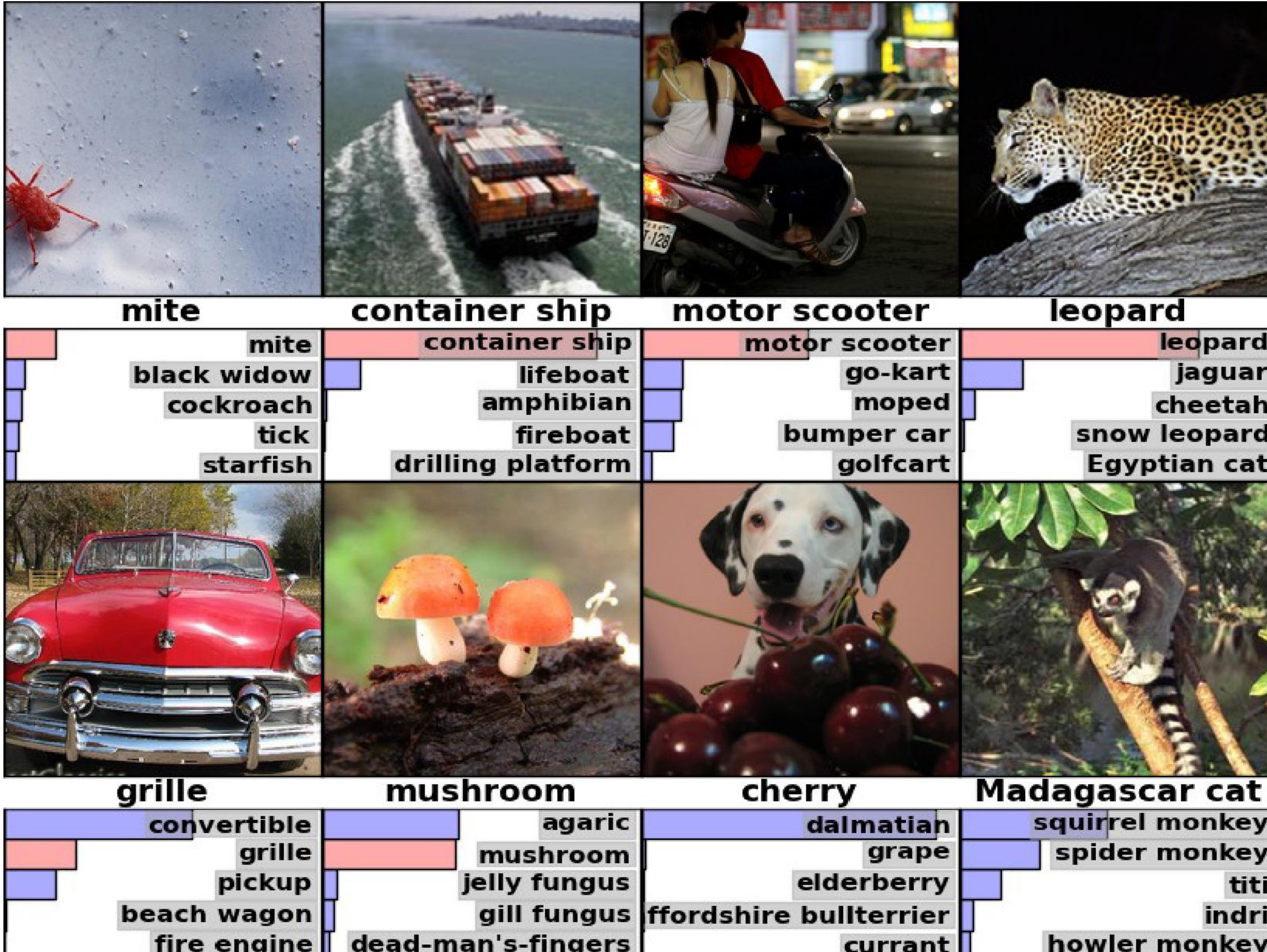
Taigman et al. "DeepFace..." CVPR 2014

Dataset: ImageNet 2012



- S: (n) Eskimo dog, **husky** (breed of heavy-coated Arctic sled dog)
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) working dog (any of several breeds of usually large powerful dogs bred to work as draft animals and guard and guide dogs)
      - S: (n) dog, domestic dog, Canis familiaris (a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) *"the dog barked all night"*
        - S: (n) canine, canid (any of various fissiped mammals with nonretractile claws and typically long muzzles)
          - S: (n) carnivore (a terrestrial or aquatic flesh-eating mammal) *"terrestrial carnivores have four or five clawed digits on each limb"*
            - S: (n) placental, placental mammal, eutherian, eutherian mammal (mammals having a placenta; all mammals except monotremes and marsupials)
              - S: (n) mammal, mammalian (any warm-blooded vertebrate having the skin more or less covered with hair; young are born alive except for the small subclass of monotremes and nourished with milk)
                - S: (n) vertebrate, craniate (animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium)
                  - S: (n) chordate (any animal of the phylum Chordata having a notochord or spinal column)
                    - S: (n) animal, animate being, beast, brute, creature, fauna (a living organism characterized by voluntary movement)
                      - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
                        - S: (n) living thing, animate thing (a living (or once living) entity)
                          - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                            - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                              - S: (n) physical entity (an entity that has physical existence)
                                - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

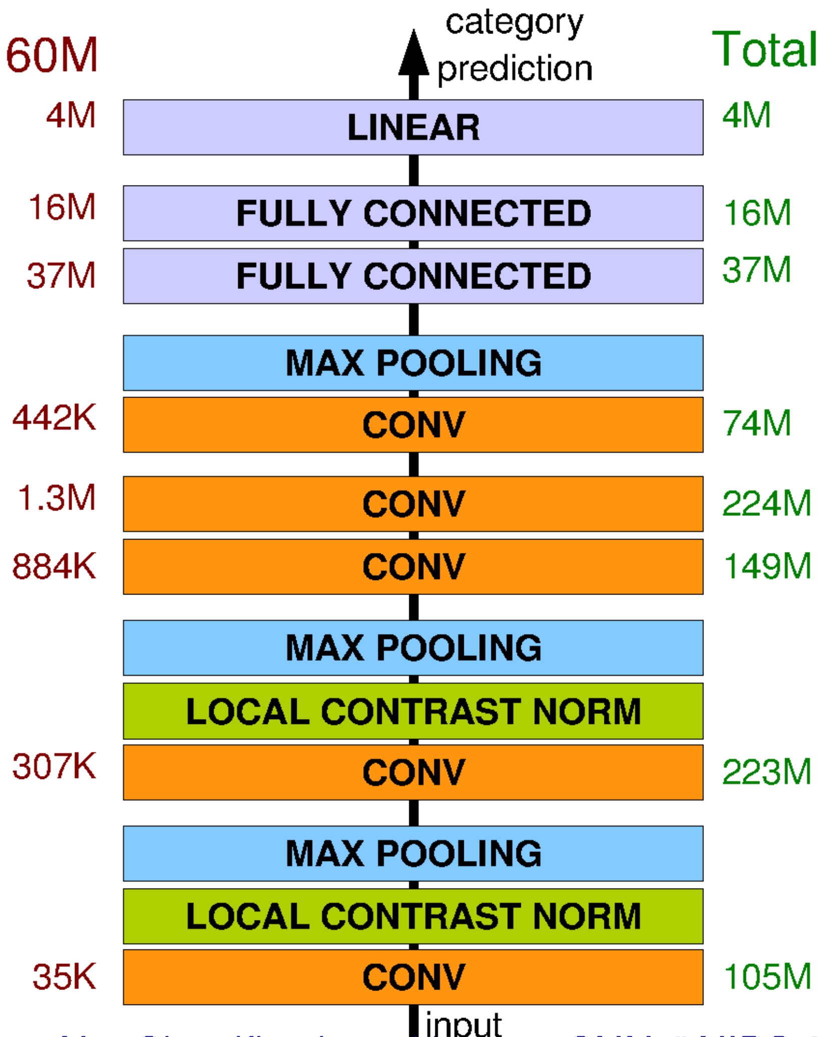
Deng et al. "Imagenet: a large scale hierarchical image database" CVPR 2009


mite
mite
black widow
cockroach
tick
starfish
container ship
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
motor scooter
go-kart
moped
bumper car
golfcart
leopard
leopard
jaguar
cheetah
snow leopard
Egyptian cat
grille
convertible
grille
pickup
beach wagon
fire engine
mushroom
agaric
mushroom
jelly fungus
gill fungus
dead-man's-fingers
cherry
dalmatian
grape
elderberry
ffordshire bullterrier
currant
Madagascar cat
squirrel monkey
spider monkey
titi
indri
howler monkey

Architecture for Classification

Total nr. params: 60M

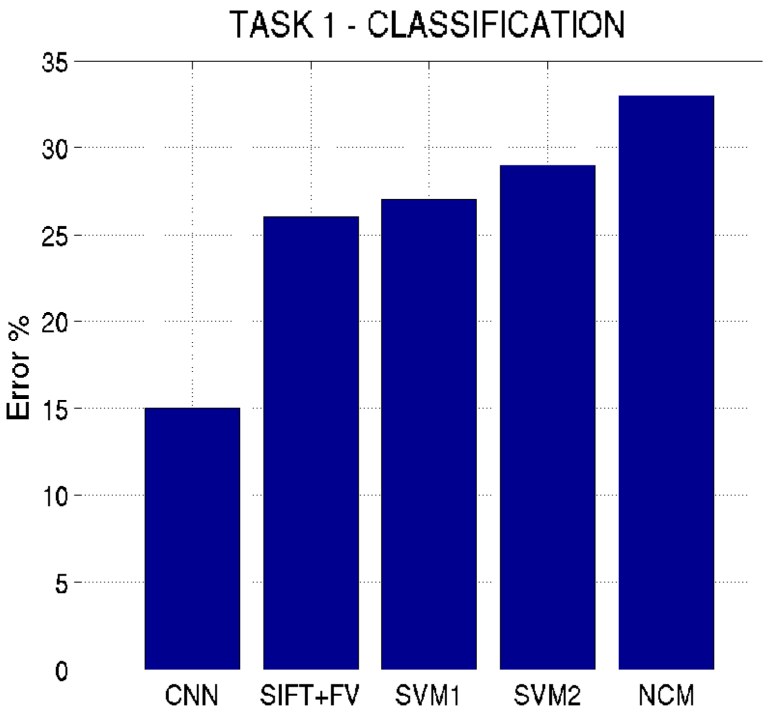
Total nr. flops: 832M

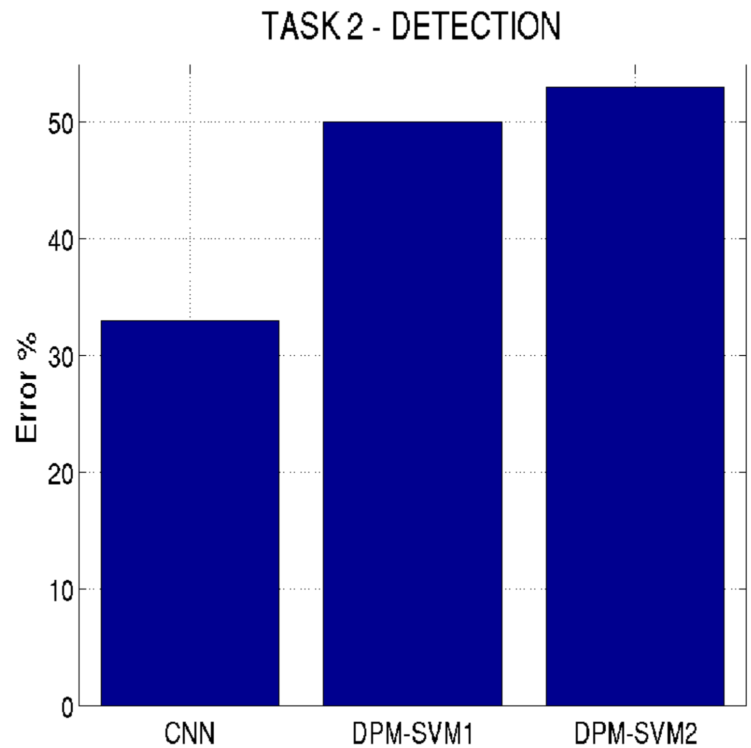


Krizhevsky et al. “ImageNet Classification with deep CNNs” NIPS 2012

Ranzato

Results: ILSVRC 2012



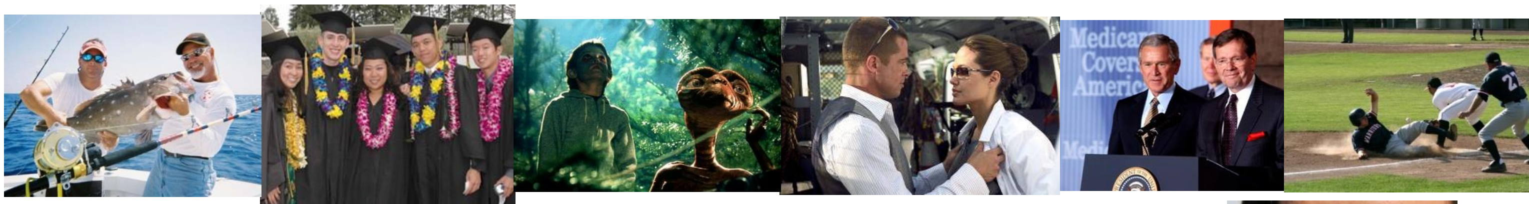


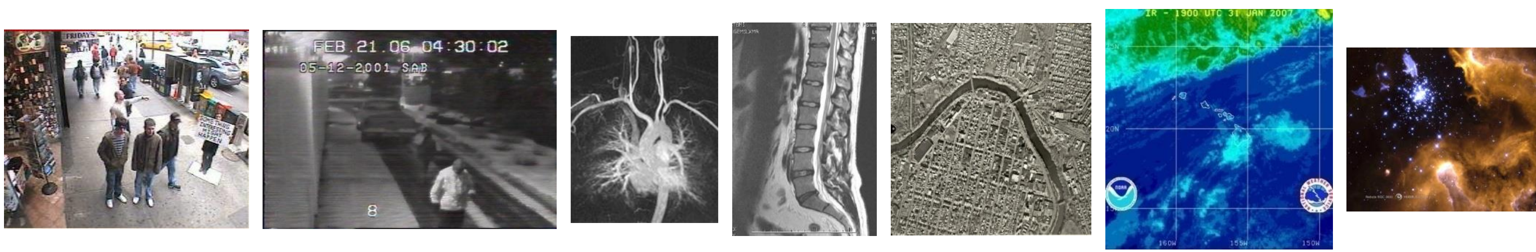
Krizhevsky et al. “ImageNet Classification with deep CNNs” NIPS 2012

Ranzato

Why study computer vision?

- Millions of images being captured all the time





- Loads of useful applications

Computer vision and AI

- The development of computer vision have benefit enormously from signal processing and "AI"
- Three pillars of AI
  - Symbolic model (expert systems)
  - Probabilistic (Bayesian) model
  - Neural networks
- We see a little bit of neural networks from the last couple weeks (will look deeper in my ANN class next spring)
- More on probabilistic models in ECE 5973: information theory and probabilistic programming

Information theory and Probabilistic programming (coming fall)

- Use probabilistic model for inference (prediction)
  - Organized unknown with graphical models
  - Infer unknown given observations
  - Learn variable models
- Applications like
  - Predict stock markets
  - Recommending products (movies, books, etc.)
  - Medical diagnosis and prognosis
  - Predict trend (e.g., COVID-19, when it is going to end?)

Hope to see you all in future classes!

Good luck with finals and have a fruitful summer break!