Literature Reviews of CNN Architectures

Kietikul Jearanaitanakij

Department of Computer Engineering, KMITL

https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html

LeNet-5 (1998)

- Yann LeCun's LeNet in 1998 was the real pioneering publication. ('*Gradient-Based Learning Applied to Document Recognition*')
- CNN was proposed to perform the character recognition from the input image.

MNIST (Modified National Institute of Standards and Technology) dataset.

It publishes Handprinted Sample Forms from 3600 writers, 810,000 character images.

LeNet-5

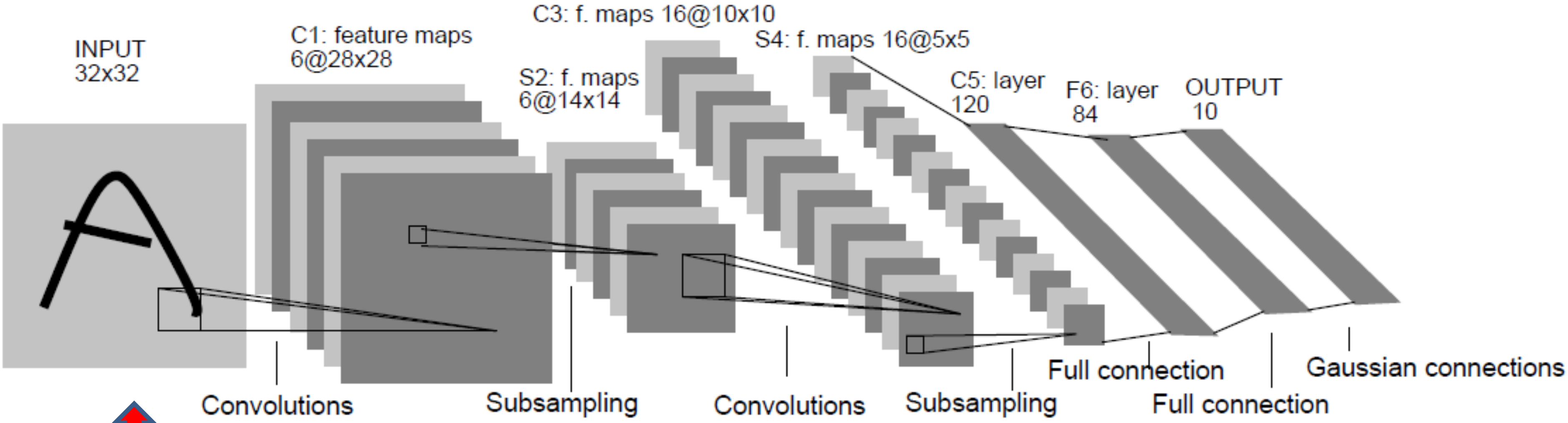


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Input for LeNet is large, i.e., 28x28, because all potential distinctive features such as stroke end-points or corner can appear in the center of the receptive field.

LeNet-5

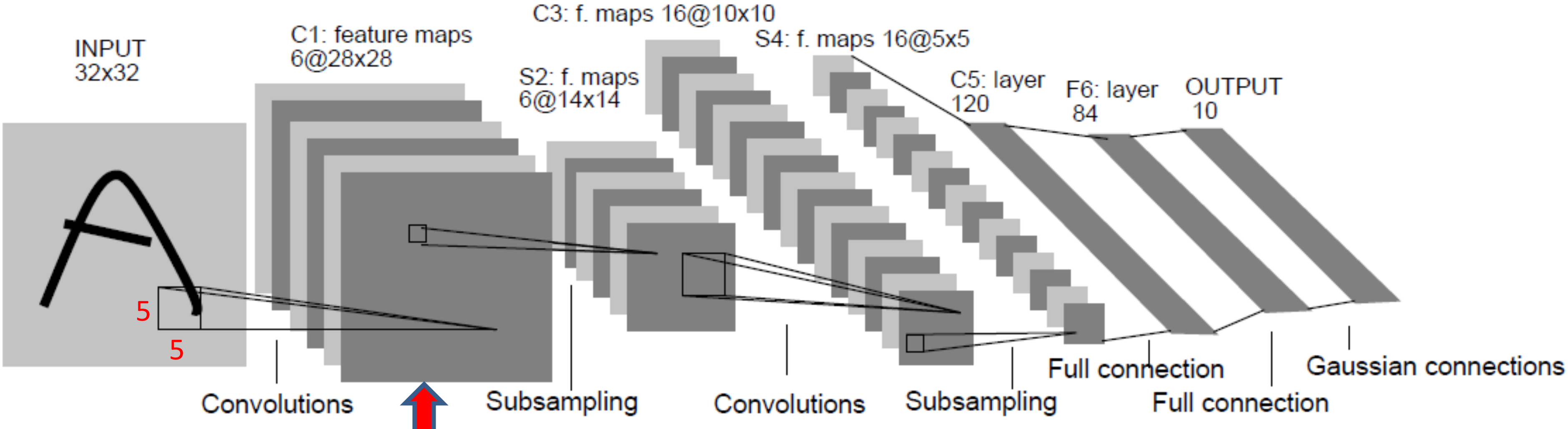


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- C1 is a convolutional layer with 6 feature maps (28x28).
- Each unit in each feature map is connected to $\underbrace{\text{5x5}}_{\text{Filter size}}$ neighborhood in the input.
- C1 contains 156 trainable parameters, and 122,304 connections.

(5x5 weights + 1 bias) x 6

[28x28x(25+1)x6]

LeNet-5

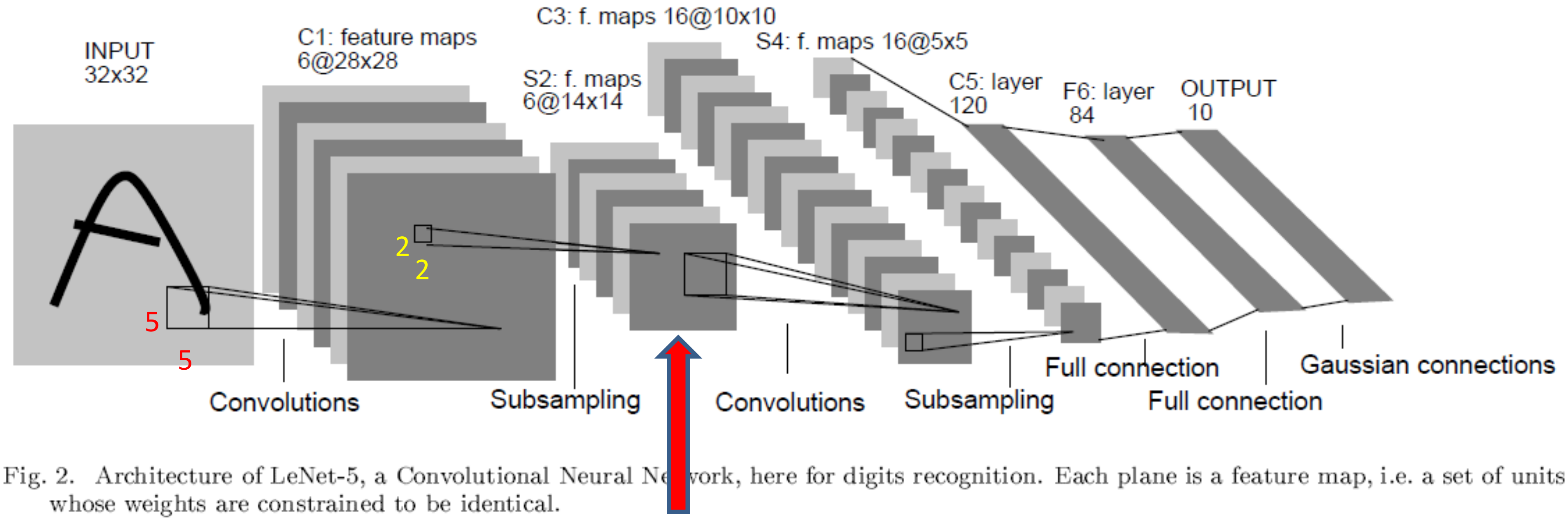


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- S2 is a sub-sampling layer with 6 features maps of size 14x14.
- Each unit in each feature map is connected to a 2x2 neighborhood in the corresponding feature map in C1.
- 2x2 inputs are added, then multiplied by a trainable coefficient, and added to a trainable bias.
- The result is passed through a sigmoidal function.
- There are 12 trainable parameters and 5,880 connections.

6 layers x (1 coef. + 1 bias)   (14x14) x (4+1) x 6 layers

LeNet-5

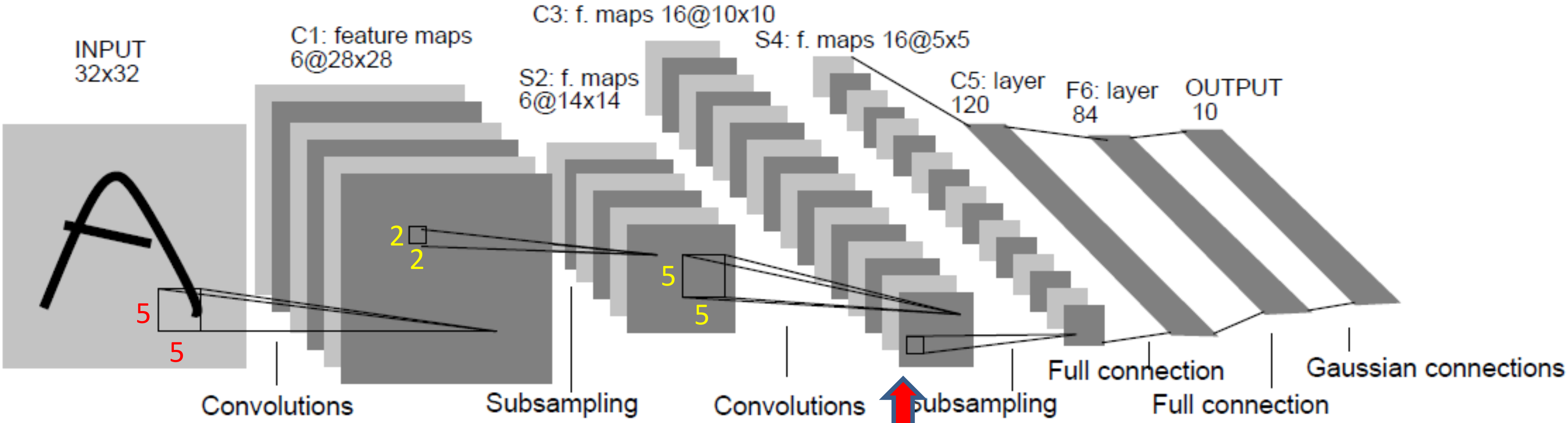


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digit recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- C3 is a convolutional layer with 16 feature maps.
- Each unit in each feature map is connected to several 5x5 neighborhoods at identical locations in a subset of S2’s feature maps.

LeNet-5

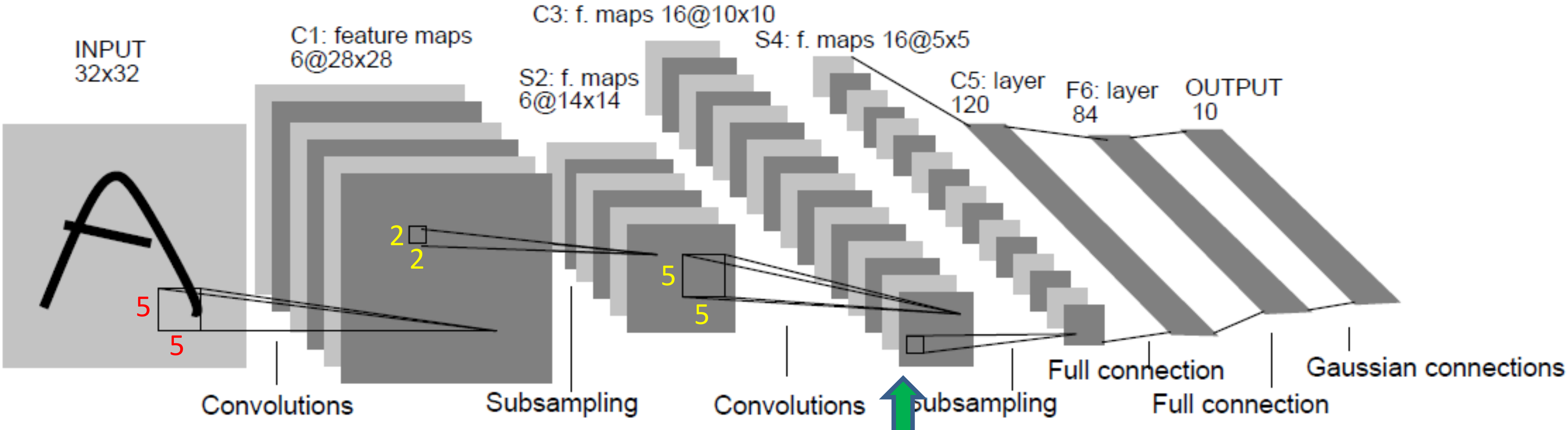
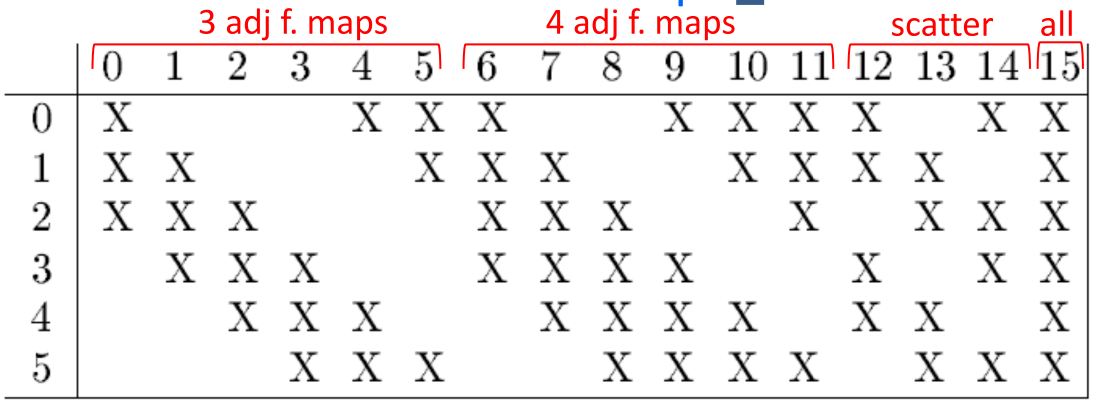


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digit recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

C3 feature maps

3 adj f. maps | 4 adj f. maps | scatter | all

S2 feature maps



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

Each column indicates which feature map in S2 are combined by the units in a particular feature map of C3.

LeNet-5

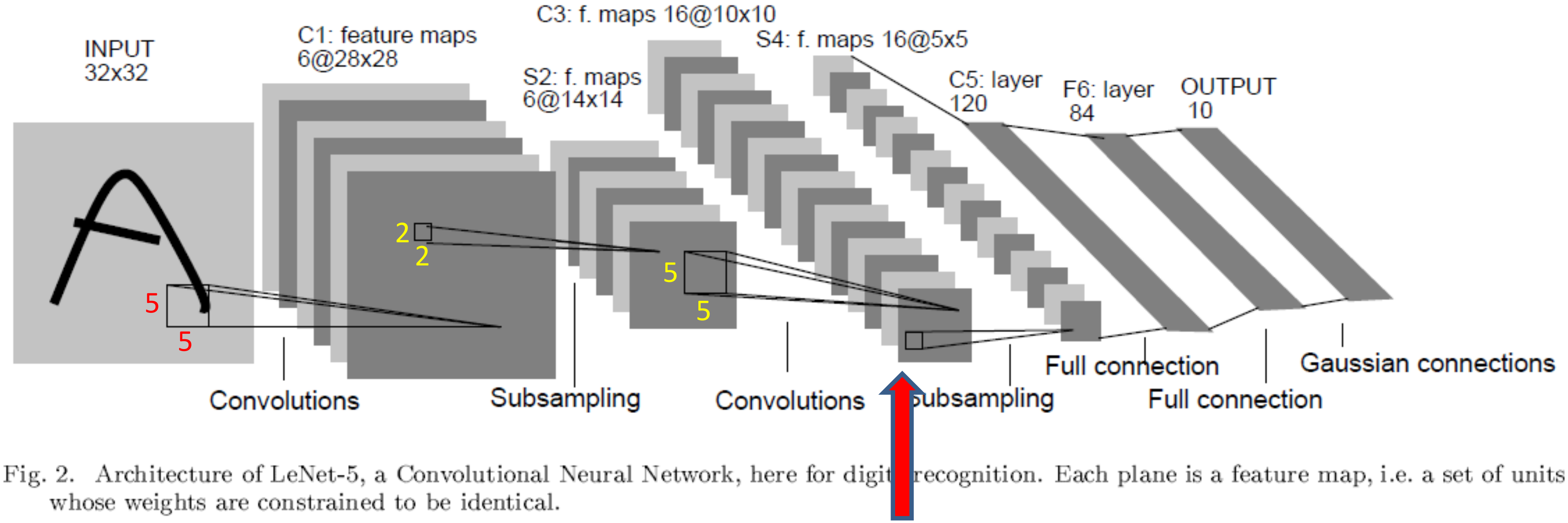


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- Why not connect every S2 feature map to every C3 feature map?
  - A non-complete connection scheme keeps the number of connections within reasonable bounds.
  - It forces a break of symmetry in the network. Different feature maps are forced to extract different features because they get different sets of inputs.
- Layer C3 has 1,516 trainable parameters and 151,600 connections.

LeNet-5

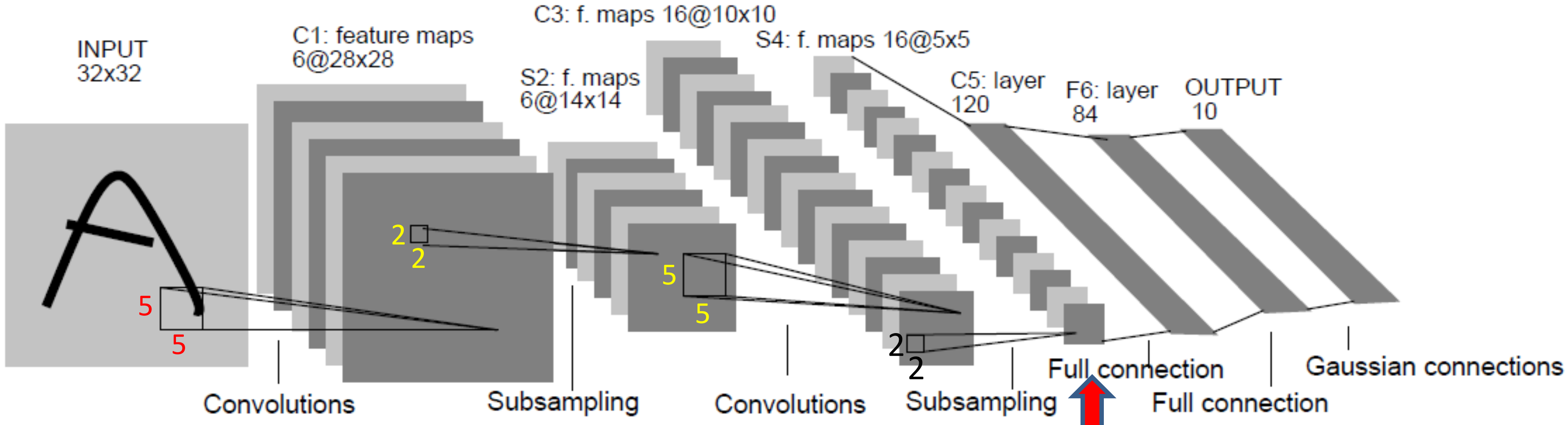


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- Each unit in each feature map is connected to subsampling with a 2x2 neighborhoods in the corresponding feature map in C3, in a similar way as C1-S2.
- Layer S4 has 32 trainable parameters and 2,000 connections.

LeNet-5

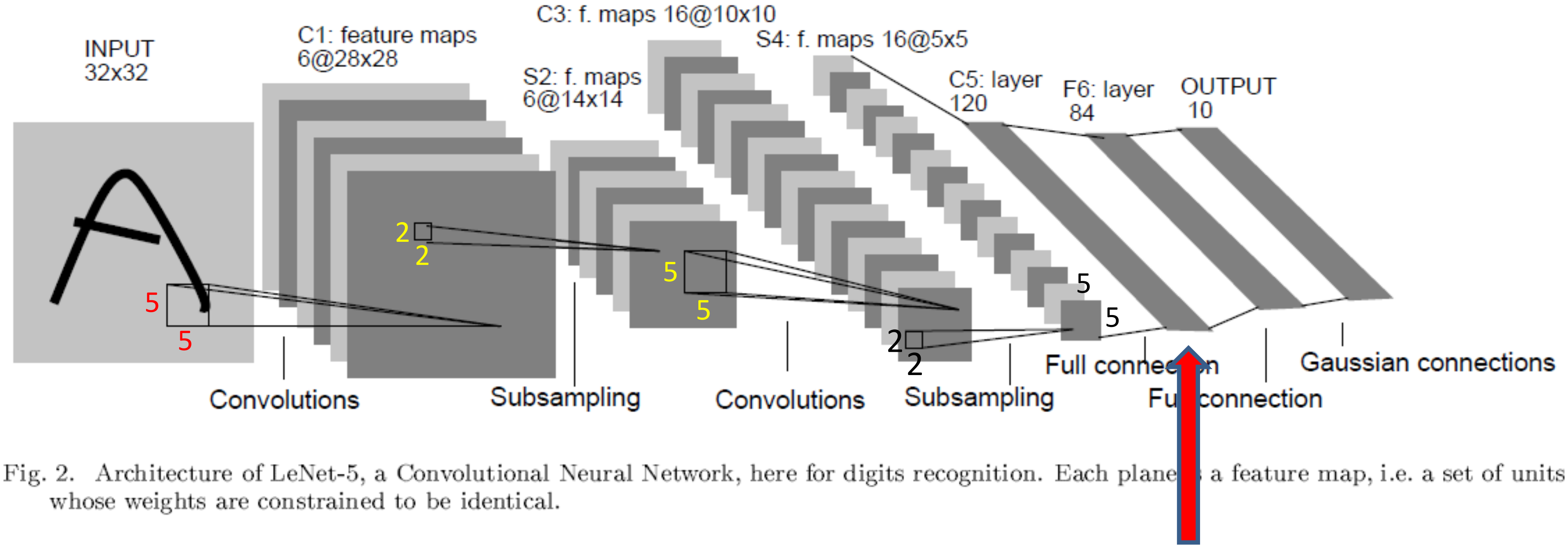


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- Layer C5 is a convolutional layer with 120 feature maps of size 1x1.
- Each unit in each feature map is connected to a 5x5 neighborhoods on all 16 of S4 feature maps.
- Layer C5 has 48,120 trainable connections.

LeNet-5

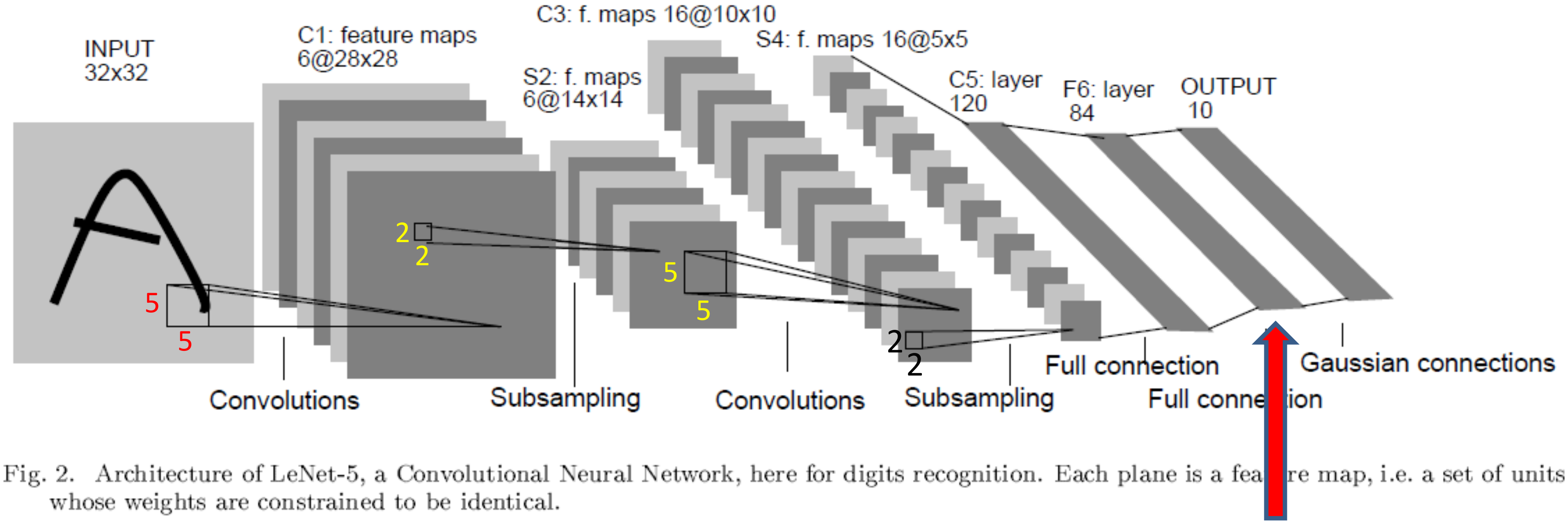


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- Layer F6 contains 84 units and is fully connected to the layer C5.
- Layer F6 has 10,164 trainable parameters.
- Each unit in layers up to F6 compute a dot product between their input vector and their weight vector. Then pass this weighted sum through a scaled hyperbolic tangent.

LeNet-5

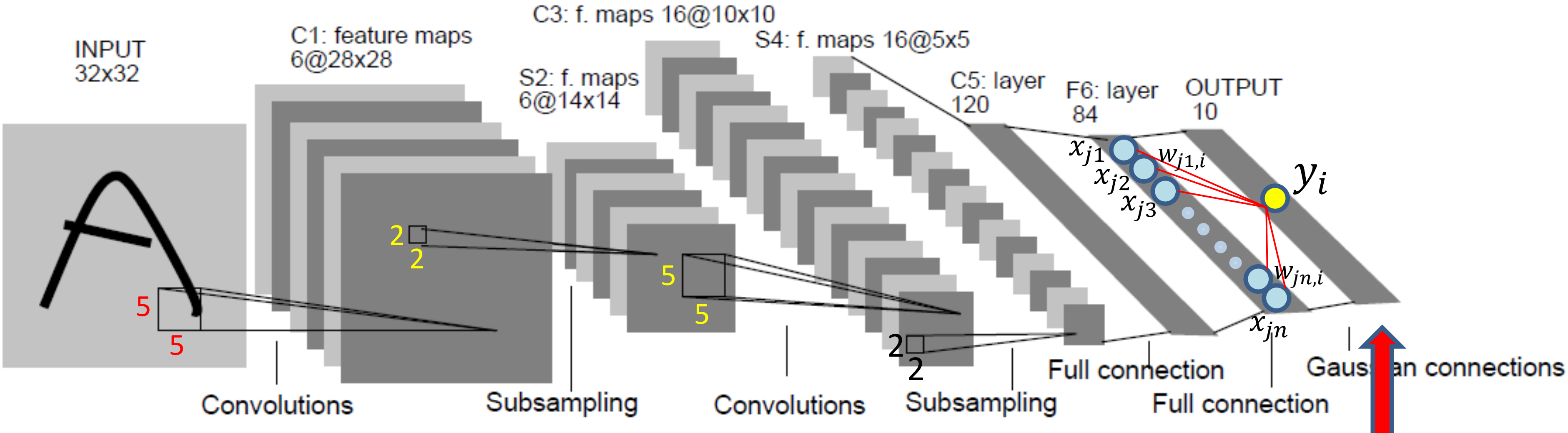


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

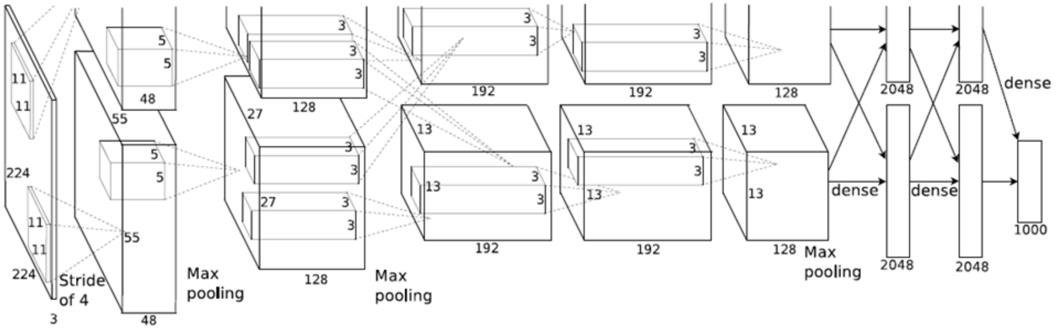
- Output layer is composed of 10 Euclidean Radial Basis Function units (RBF), one for each class, with 84 inputs each.

$$y_i = \sum_j (x_j - w_{ij})^2$$

- Each output RBF unit computes the Euclidean distance between its input vector and its parameter vector.
- LeNet produces test error rate about 1% on MNIST dataset.

AlexNet (2012)

- Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton presented AlexNet to win the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge).



- This competition can be thought of as the annual Olympics of computer vision, where teams from across the world compete to see who has the best computer vision model for tasks such as classification, localization, detection, and more.

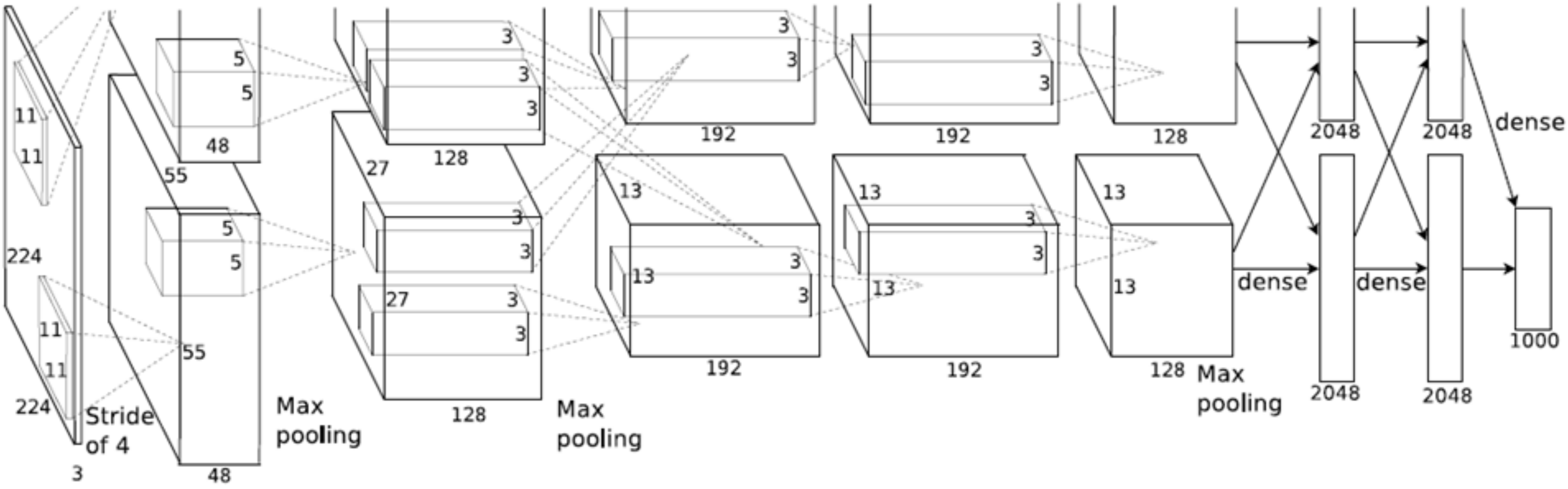
- AlexNet is widely regarded as one of the most influential publications in this field.
- Original paper: 'ImageNet Classification with Deep Convolutional Networks'

- 2012 marked the first year where a CNN was used to achieve a top 5 test error rate of 15.4% (Top 5 error is the rate at which, given an image, the model does not output the correct label with its top 5 predictions).

- The runner-up achieved an error of 26.2%, which was an astounding improvement that pretty much shocked the computer vision community.

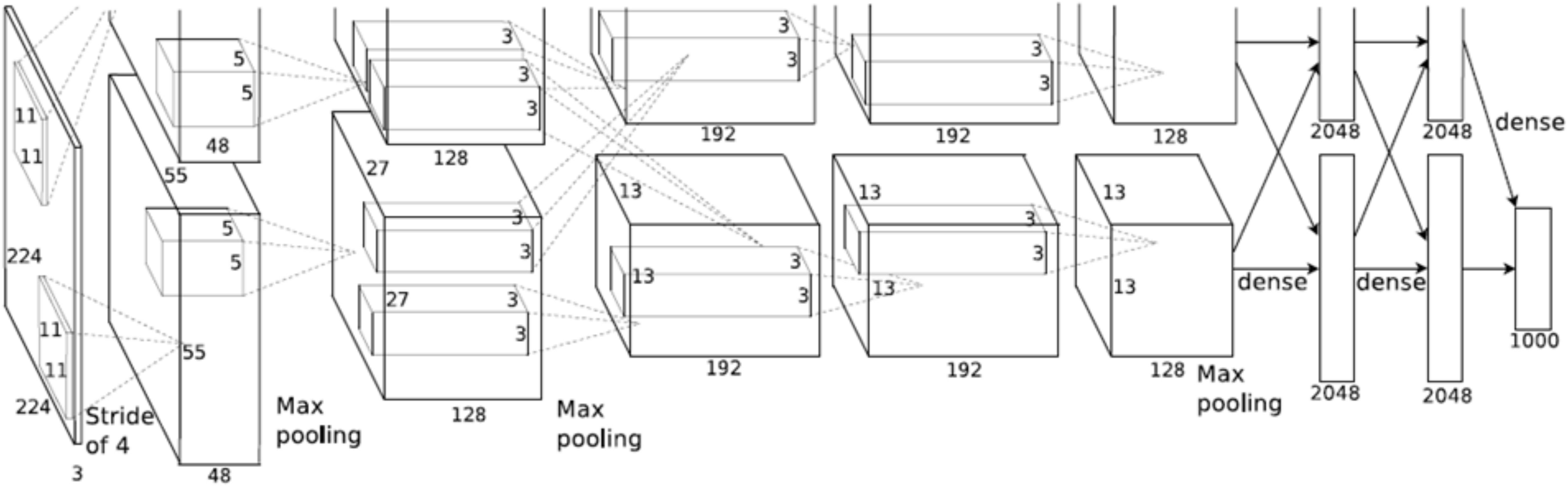
- AlexNet was designed to classify 1000 possible categories.

AlexNet



- AlexNet CNN is large & deep.
- The training process was so computationally expensive that they had to split the training onto 2 GPUs.

AlexNet



5 convolutional layers

3 fully-connected layers

- The network was made up of 5 conv layers, max-pooling layers, **dropout layers**, and 3 fully connected layers.

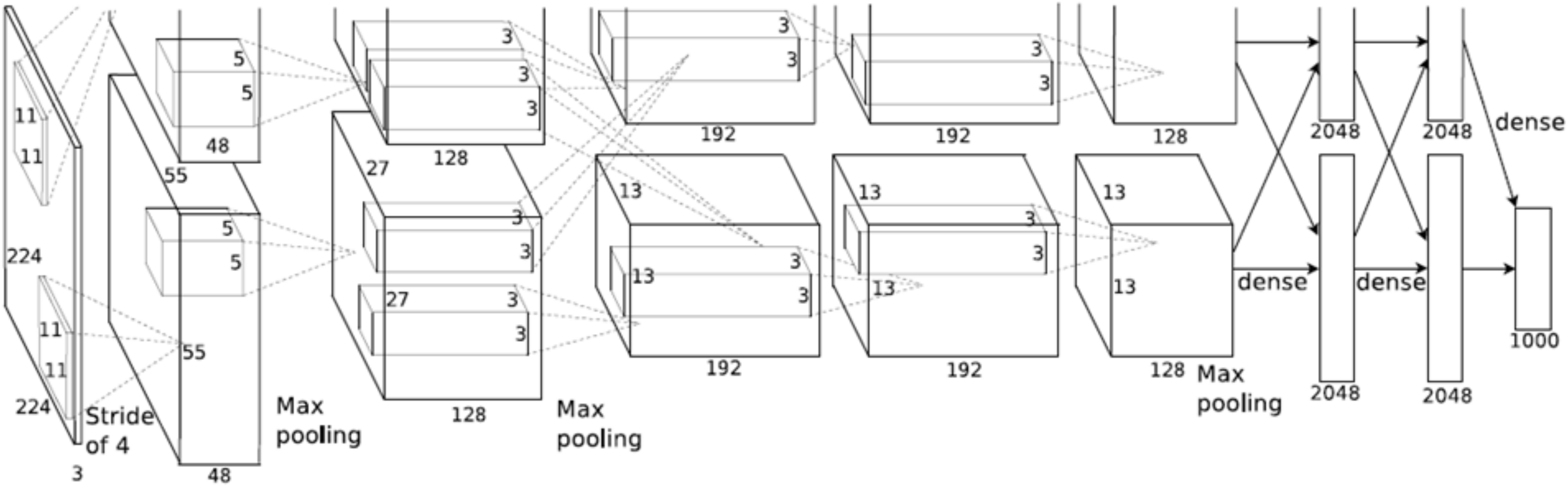
Dropout is a regularization technique for reducing overfitting.

AlexNet

Dropout

- Dropout sets the output of each hidden neuron to zero with probability 0.5.
- The neurons which are ‘dropped’ out in this way do not contribute to the forward pass and do not participate in backpropagation.
- AlexNet uses dropout in the first two FC layers.
- At test time, we use all the neurons but multiply their outputs by 0.5, which is the geometric mean of predictive distributions.
- Dropout cost only a factor of two during training.

AlexNet



1st 2nd 3rd 4th 5th

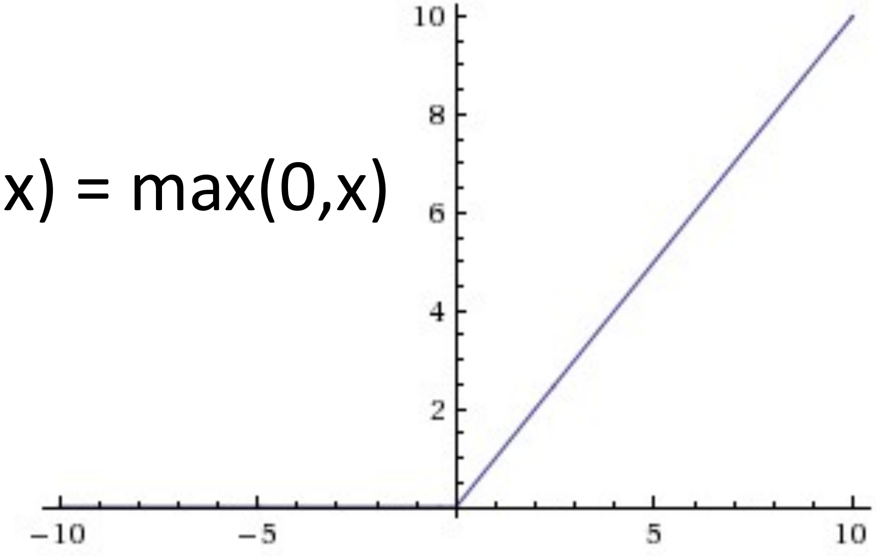
- Neurons of 2nd, 4th and 5th convolutional layers are connected only to those kernels in the previous layer which reside on **the same GPU**.
- Neurons of 3rd convolutional layer are connected to all kernels in 2nd convolutional layer on **both GPUs**.

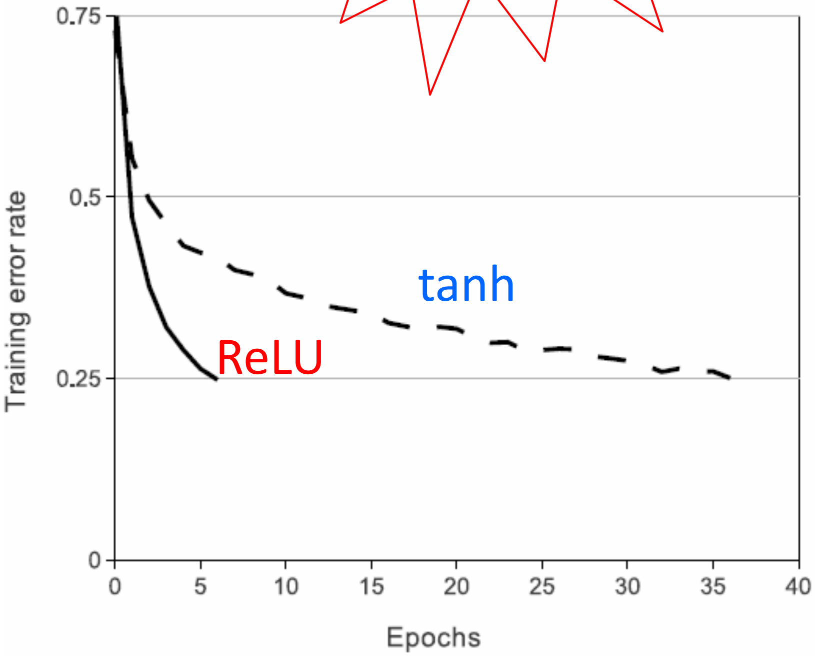
AlexNet

ReLU Nonlinearity

6 times faster !

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





- Use ReLUs to model a neuron's output f.
- The network using ReLUs is approx. six times, in training CIFAR-10, faster than an equivalent network with tanh neurons.
- ReLUs is applied to the output of every conv. and FC layers

AlexNet

Local Response Normalization (LRN)

- ReLUs have the desirable property that they do not require input normalization to prevent them from saturating.
- However, ReLUs neurons have unbounded activations and we need LRN to normalized them.
- LRN follows 1st and 2nd convolutional layers.

Total number of kernels

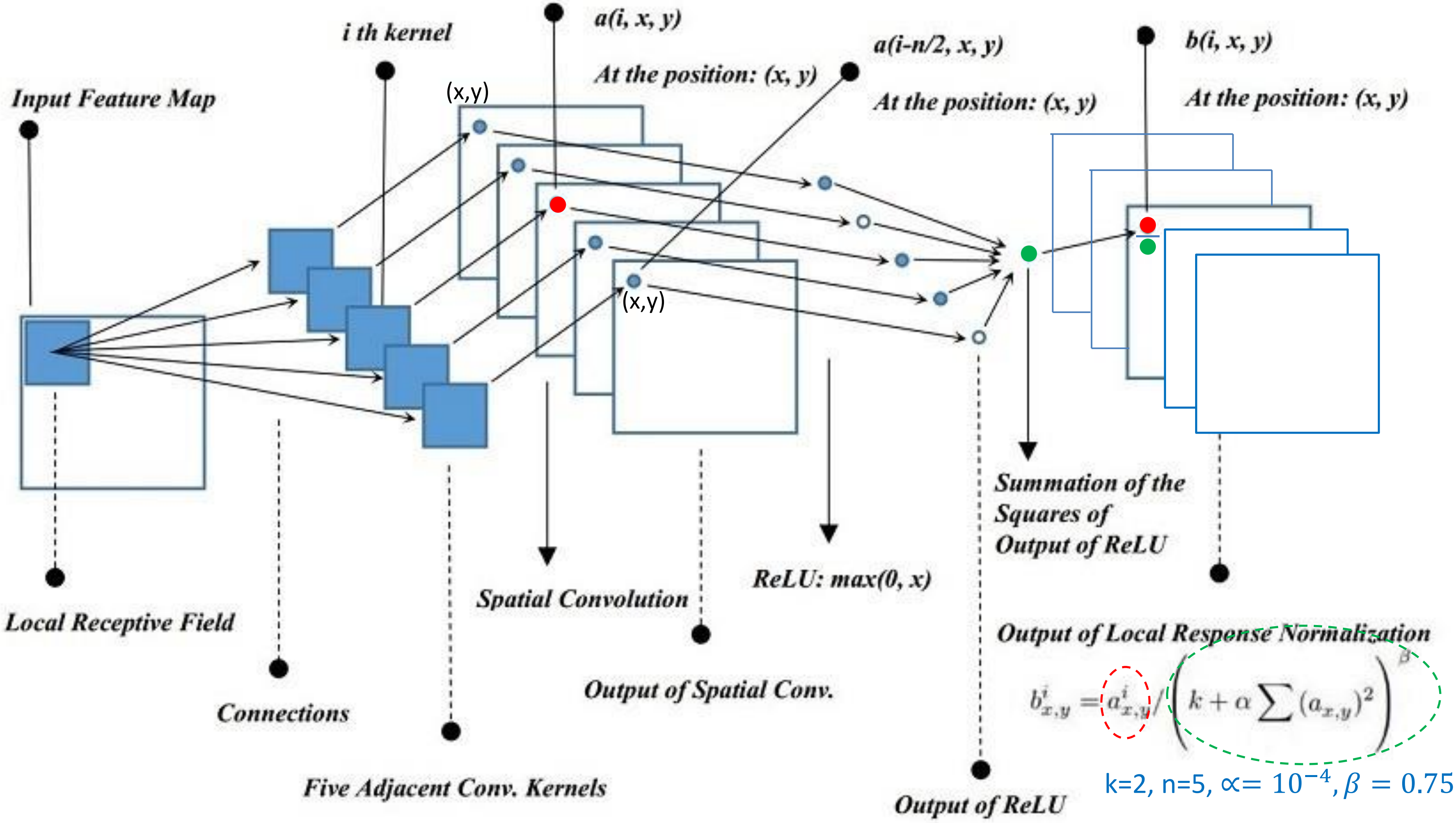
Adjacent kernels at the same spatial localtion (x,y)

$$b^i_{x,y} = a^i_{x,y} \Big/ \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

The activity of a neuron computed by applying kernel i at position (x,y)

AlexNet

Local Response Normalization (LRN)



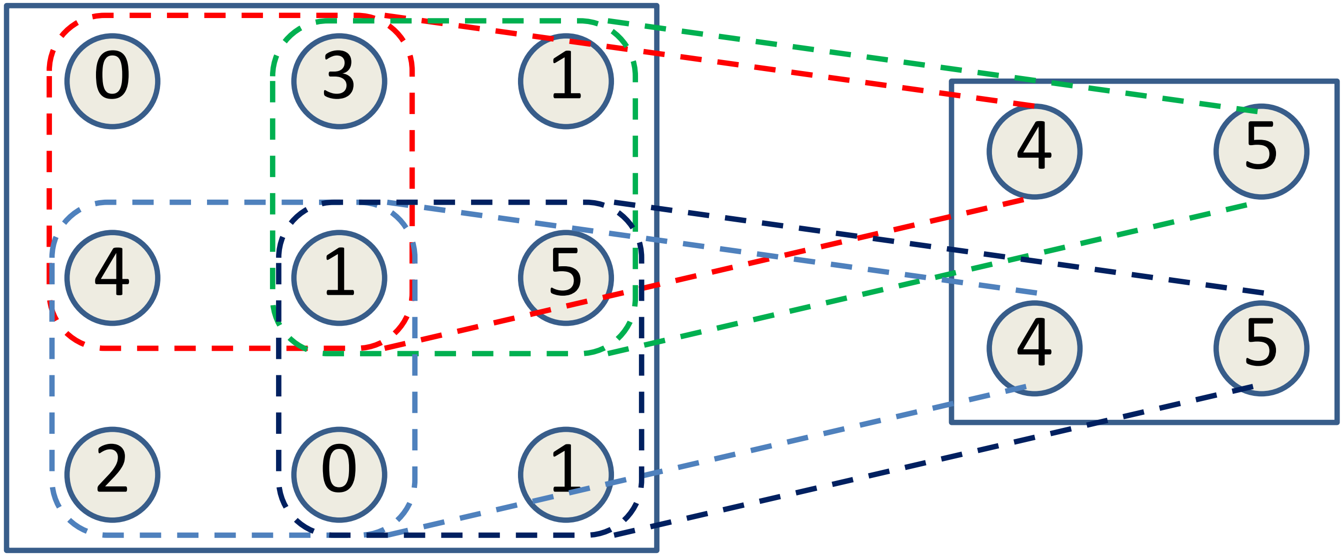
$$b^{i}_{x,y} = a^{i}_{x,y} / \left(k + \alpha \sum (a_{x,y})^{2} \right)^{\beta}$$

k=2, n=5, $\propto = 10^{-4}, \beta = 0.75$

AlexNet

Overlapping Pooling

- Pooling layers in AlexNet summarize the outputs of neighboring group of neurons in the same kernel map with overlapping (stride = 1).



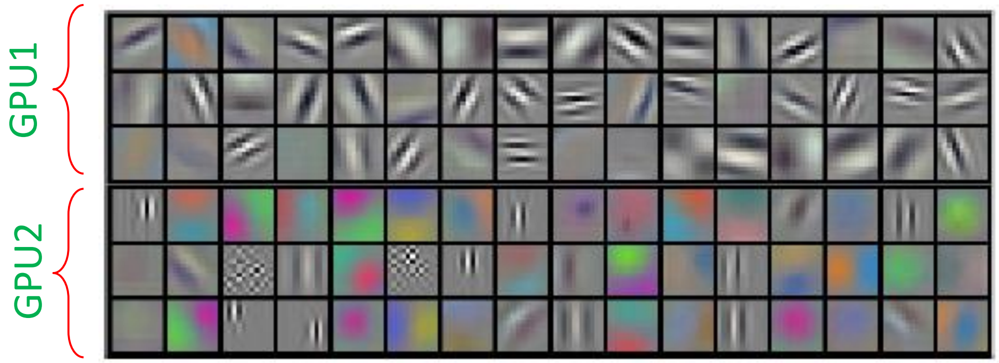
- Overlapping max-pooling layers follow both LRN layers and 5th convolutional layer.

AlexNet

Details of learning

- Stochastic gradient descent with a **batch size** of **128** examples, **momentum 0.9**, and weight decay of 0.0005.

96 trained Conv. filters



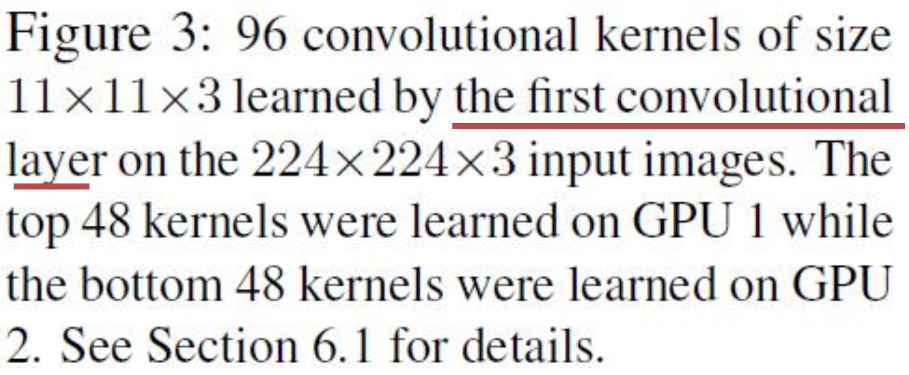
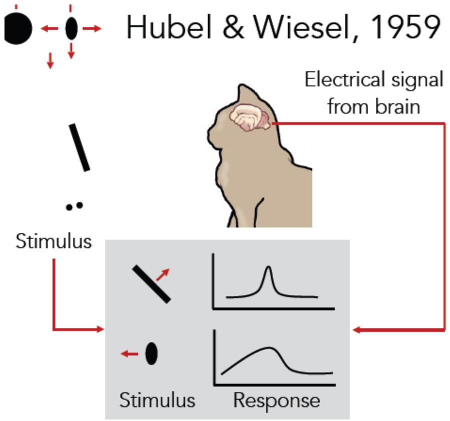


Figure 3: 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.


Hubel & Wiesel, 1959

Conclusions of AlexNet

- Trained the network on ImageNet data, which contained over 15 million annotated images from a total of over 22,000 categories.
- Used **ReLU** for the nonlinearity functions (found to decrease training time as ReLUs are several times faster than the conventional tanh function).
- Used **data augmentation** techniques that consisted of image translations, horizontal reflections, and patch extractions.
- Implemented **dropout** layers in order to combat the problem of overfitting to the training data.
- Trained the model using batch stochastic gradient descent, with specific values for momentum and weight decay.
- Trained on two **GTX 580 GPUs** for **five to six days**.

Why AlexNet is important ?

- AlexNet really illustrated the benefits of CNNs with record breaking performance in the competition.
- This was the first time a model performed so well on a historically difficult ImageNet dataset.
- Utilizing techniques that are still used today, such as data augmentation and dropout.

ZF Net (2013)

- The winner of the ILSVRC 2013 competition was a network built by Matthew Zeiler and Rob Fergus from New York University. This model achieved an 11.2% test error rate.
- This architecture was more of a **fine tuning to the AlexNet** structure.
- Authors also showed how to **visualize the filters and weights** correctly.
- Authors used only **one GPU** (GTX 580) to train the ZF Net for 12 days.

- Original paper: 'http://arxiv.org/pdf/1311.2901v3.pdf'

ZF Net

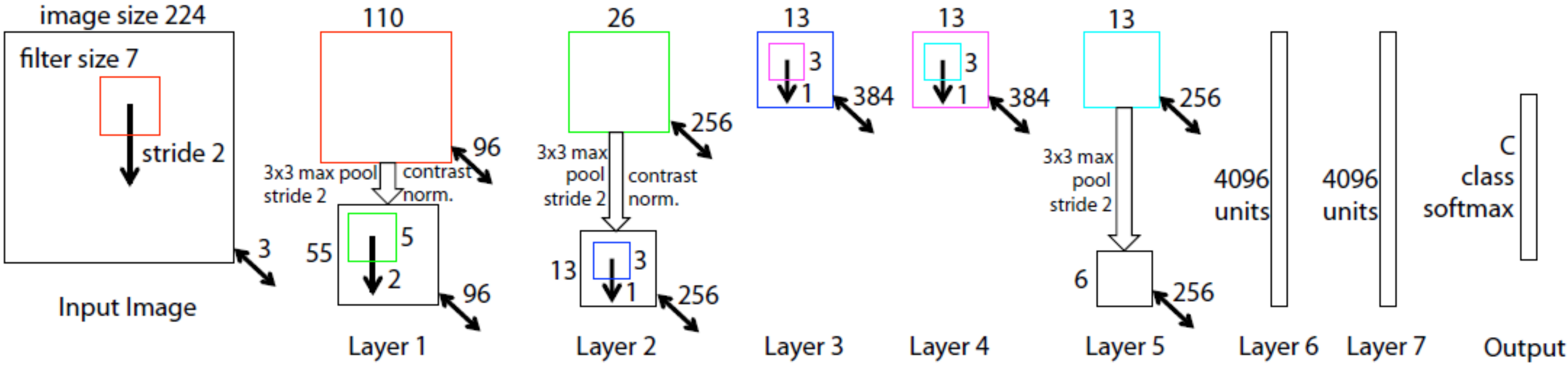


Figure 3. Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a C-way softmax function, C being the number of classes. All filters and feature maps are square in shape.

ZF Net

- Very **similar architecture to AlexNet**, except for a few minor modifications.
- AlexNet trained on 15 million images, while ZF Net trained on only 1.3 million images.
- Instead of 11x11 sized filters in the first layer, ZF Net used **smaller filters of size 7x7** and a **decreased stride** value from 4 to **2**. A smaller filter size in the first conv layer helps retain a lot of original pixel information.
- Developed a **visualization technique** named **Deconvolutional Network**, which helps to examine different feature activations and their relation to the input space.

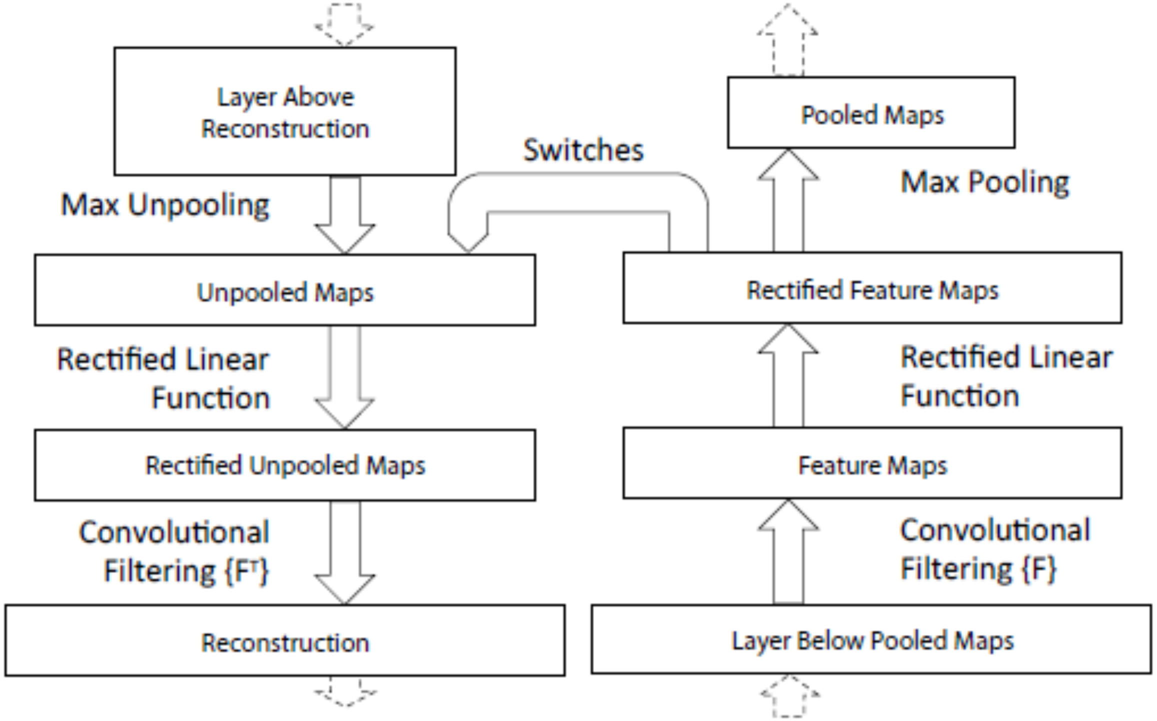
ZF Net

Deconvolutional Network (DeConvNet)

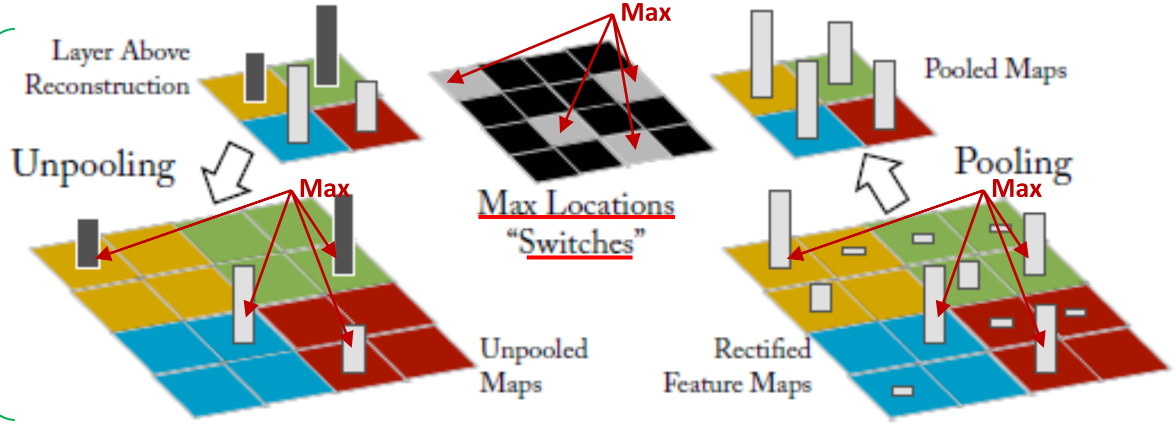
- Every layer of the trained CNN, we attach a “deconvnet”.
- We pass the feature map (activation map) as the input into the deconvnet.
- This input then reversely goes through a series of unpool (reverse maxpooling), rectify, and filter operations for each preceding layer until the input space is reached.

ZF Net

DeConvNet

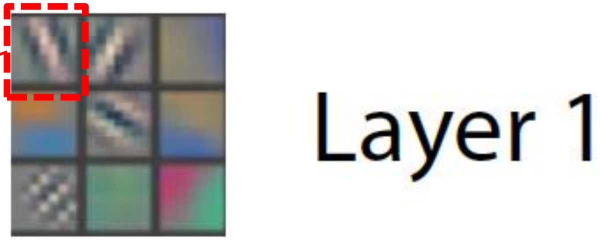


Unpooling

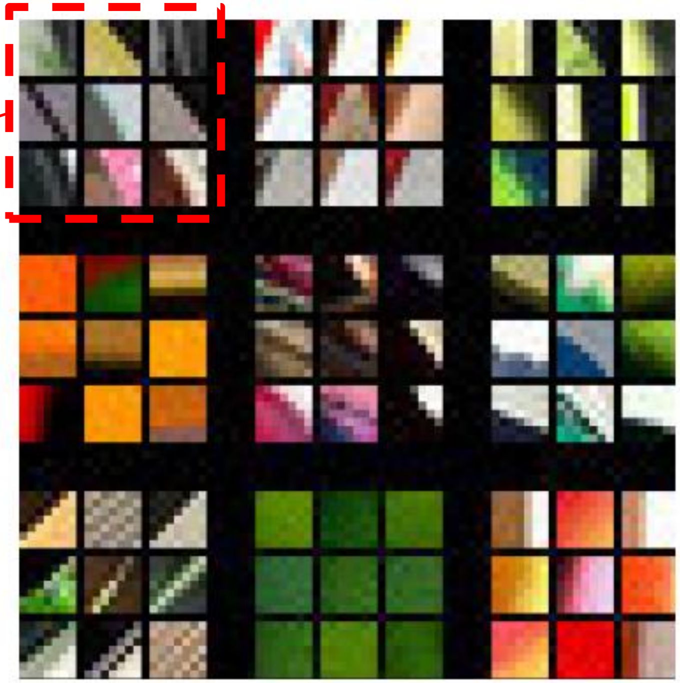


ZF Net

DeConvNet: projections from each layer show the hierarchical nature of the features in the network.



A visual images from DeConvNet in 1st layer. Layer 1 recognizes lines, color probs.

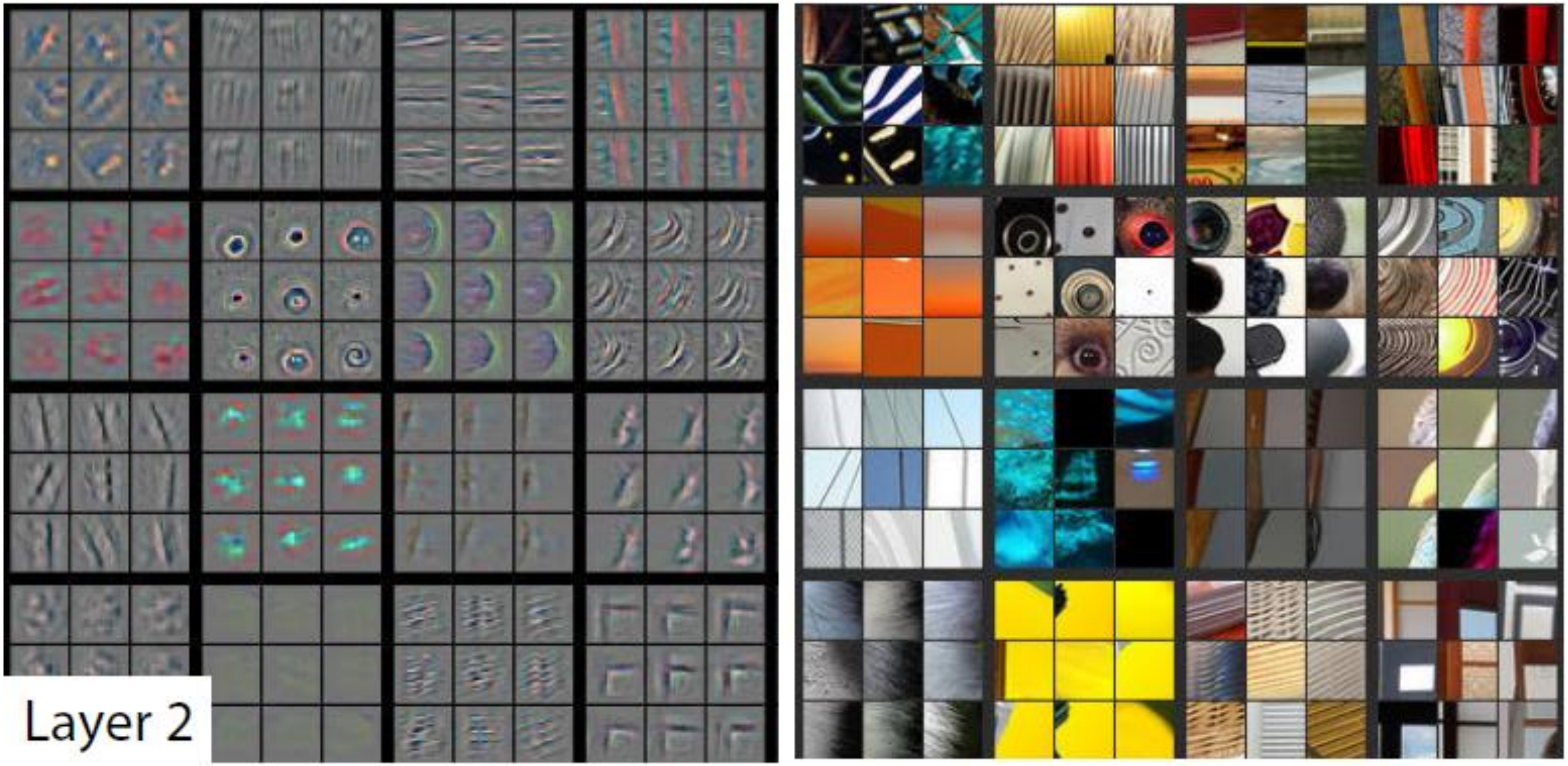


9 small portions from 9 original images, each portion is convolved with a filter.

Layer 1 recognizes lines, color probs.

ZF Net

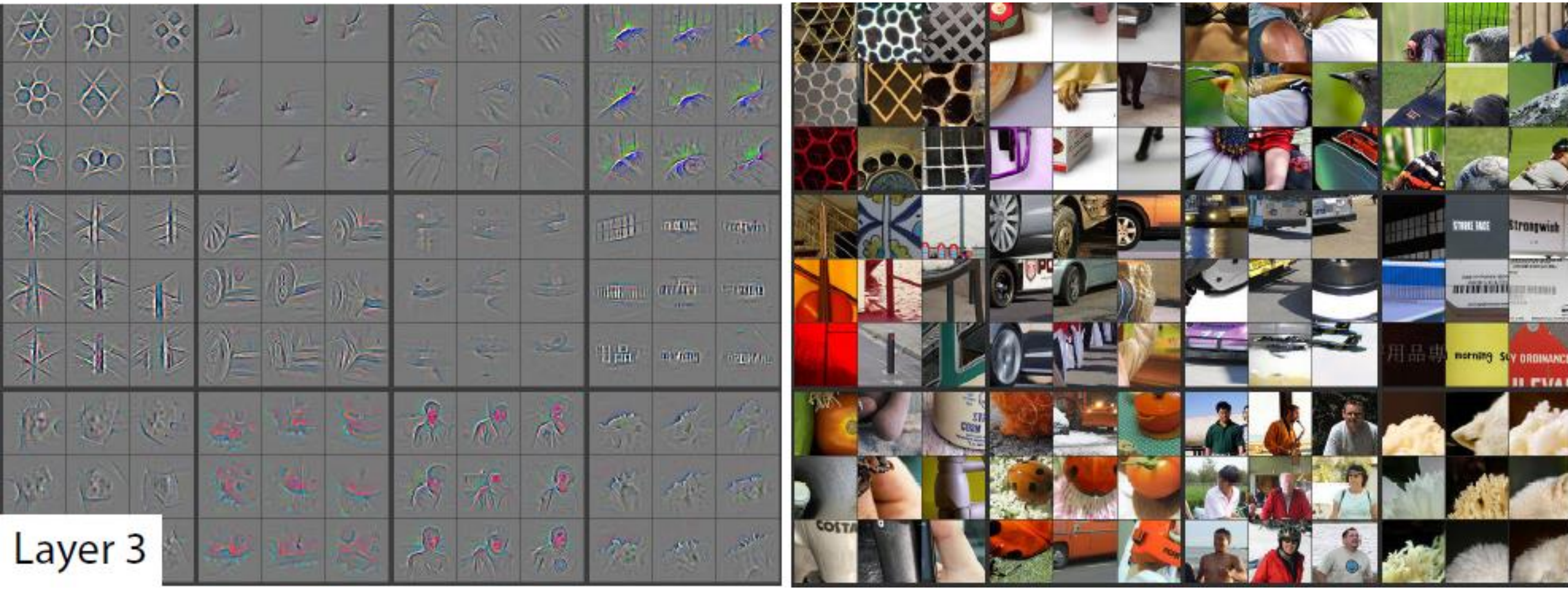
DeConvNet (Layer2)



Layer 2 responses to corners and other edge/color conjunctions.

ZF Net

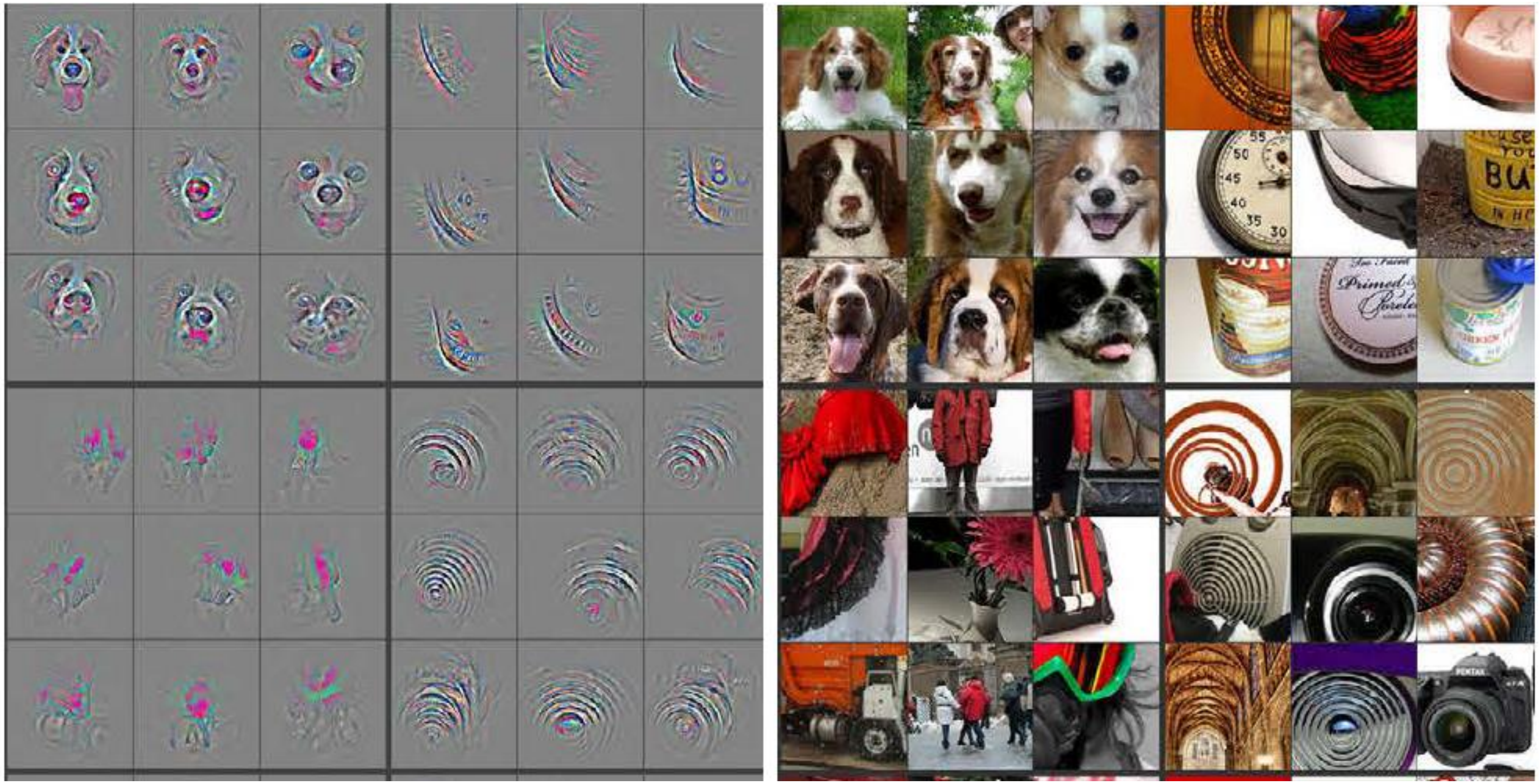
DeConvNet (Layer3)



Layer 3 responses to simple textures/patterns.

ZF Net

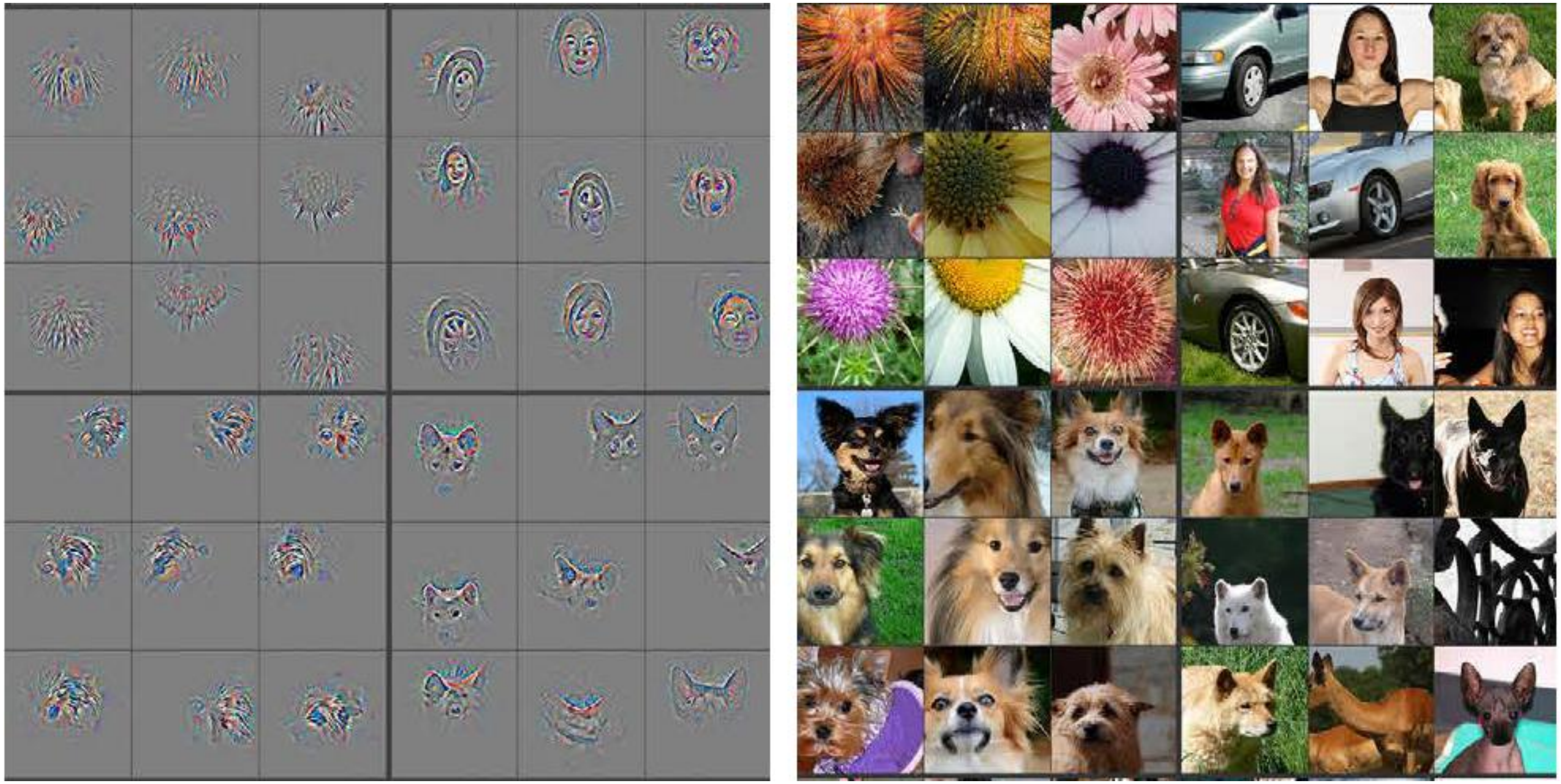
DeConvNet (Layer4)



Layer 4 shows significant variation which make a recognition among objects. It responses to various edges of object.

ZF Net

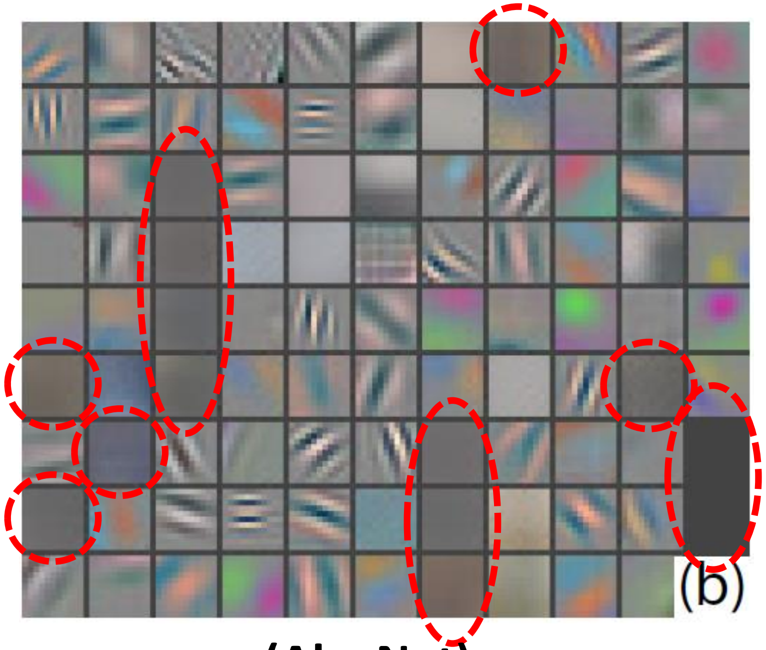
DeConvNet (Layer5)



Layer 5 **shows entire objects** with significant pose variation.

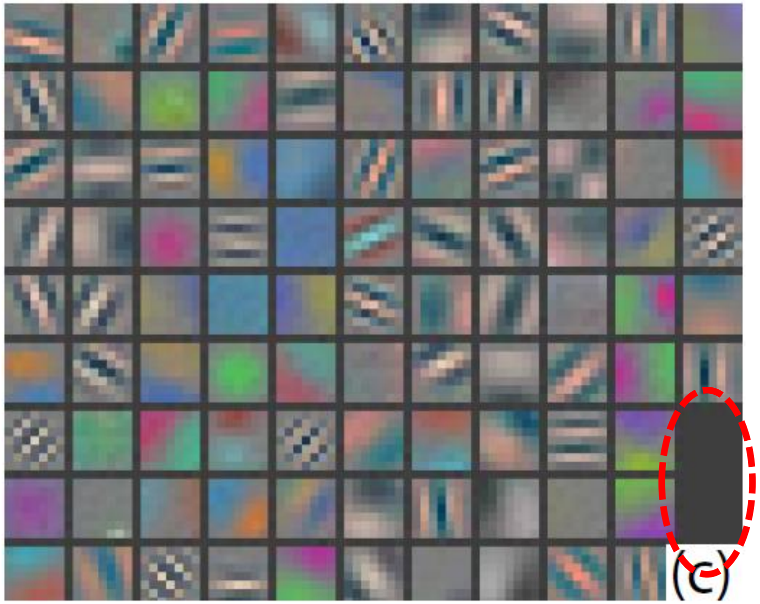
ZF Net

How do we use these visualizations for tuning the network?



(AlexNet)

- A lot of dead features.
- Alias artifacts: incorrect representation due to inadequate sampling rate.



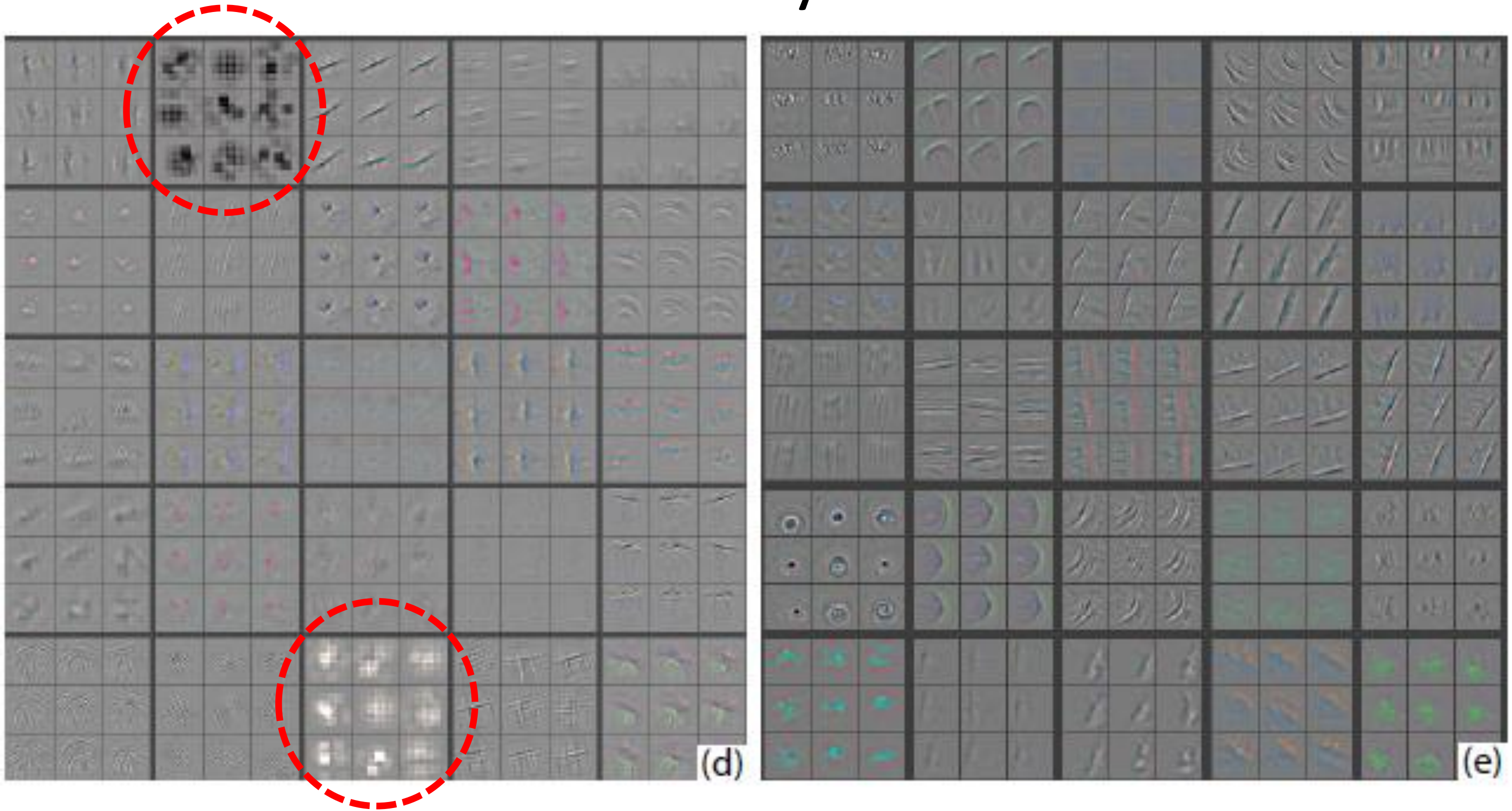
(ZFNet)

- Much better representation.
- Reduce 1st layer filter from 11x11 to 7x7.
- Reduce stride from 4 to 2.
- Better classification.

ZF Net

How do we use these visualizations for tuning the network?

2nd layer



(AlexNet) (ZFNet)

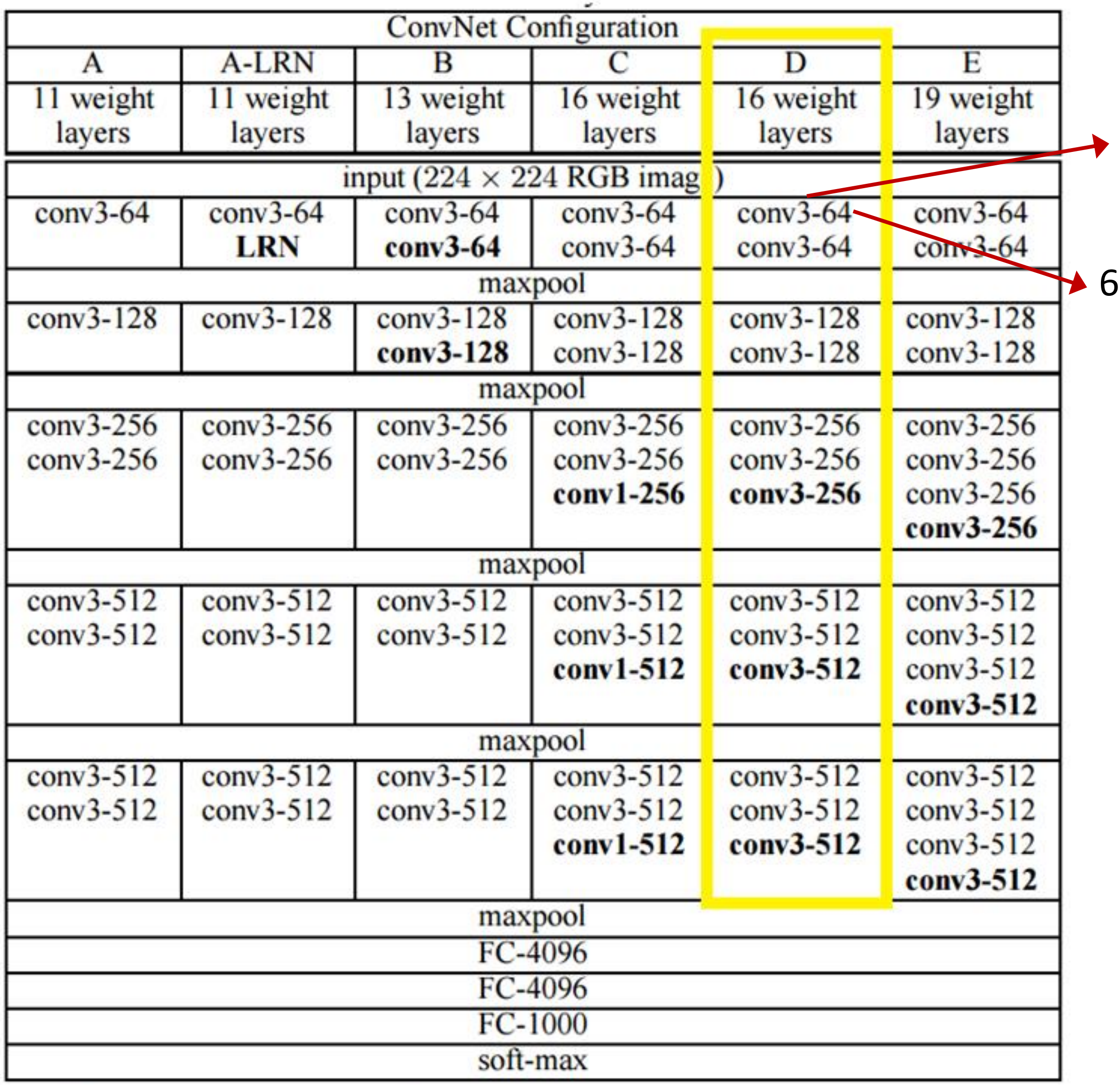
VGG Net (2014)

- Karen Simonyan and Andrew Zisserman of the University of Oxford proposed VGG Net to ILSVRC 2014 (with 7.3% error rate but weren't the winner) based on the concepts of **simplicity and more depth**.
- They used **19 layers** CNN that strictly used 3x3 filters with stride and pad of 1, along with 2x2 max-pooling layers with stride 2.
- VGG Net is important because it raised the idea that convolutional neural networks must have a **"deep" layers** in order for this hierarchical representation of visual data to work. Keep it deep and simple.
- Worked well on both image classification and localization tasks.
- Original paper: 'http://arxiv.org/pdf/1409.1556v6.pdf'

Classification
+ Localization

CAT

VGG Net



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Filter size 3x3

64 channels (filters)

The number of filters doubles after each max-pooling layer.

The 6 different architecures of VGG Net. Configuration D produced the best results

VGG Net

- The number of filters doubles after each max-pooling layer. This reinforces the idea of shrinking spatial dimensions, but growing depth.
- Built model with the Caffe toolbox. (C++)
- Use ReLU layers after each conv layer and train with batch gradient descent.
- Trained on 4 Nvidia Titan Black GPUs for **two to three weeks**.

GoogLeNet (2015)

- GoogLeNet is a 22-layer CNN and was the winner of ILSVRC 2014 with a top 5 error rate of 6.7%.
- It threw away the idea

of simplicity in the network architecture by introducing the **inception module**.

Believe it or not, Inception modules partially got their name from this meme (image source)

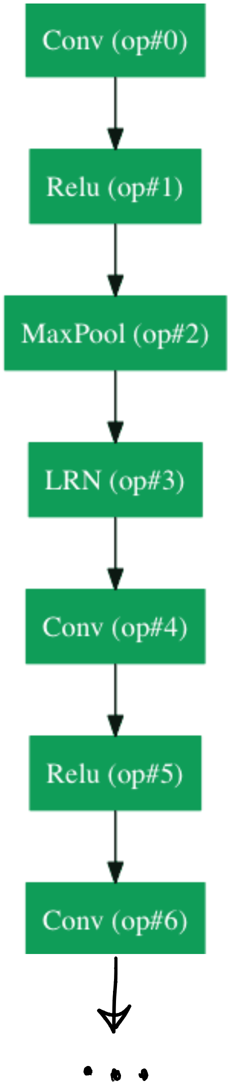
- This new model places

notable consideration on memory and power usage.

- Original paper:

www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf

GoogLeNet



Conv (op#0)

Relu (op#1)

MaxPool (op#2)

LRN (op#3)

Conv (op#4)

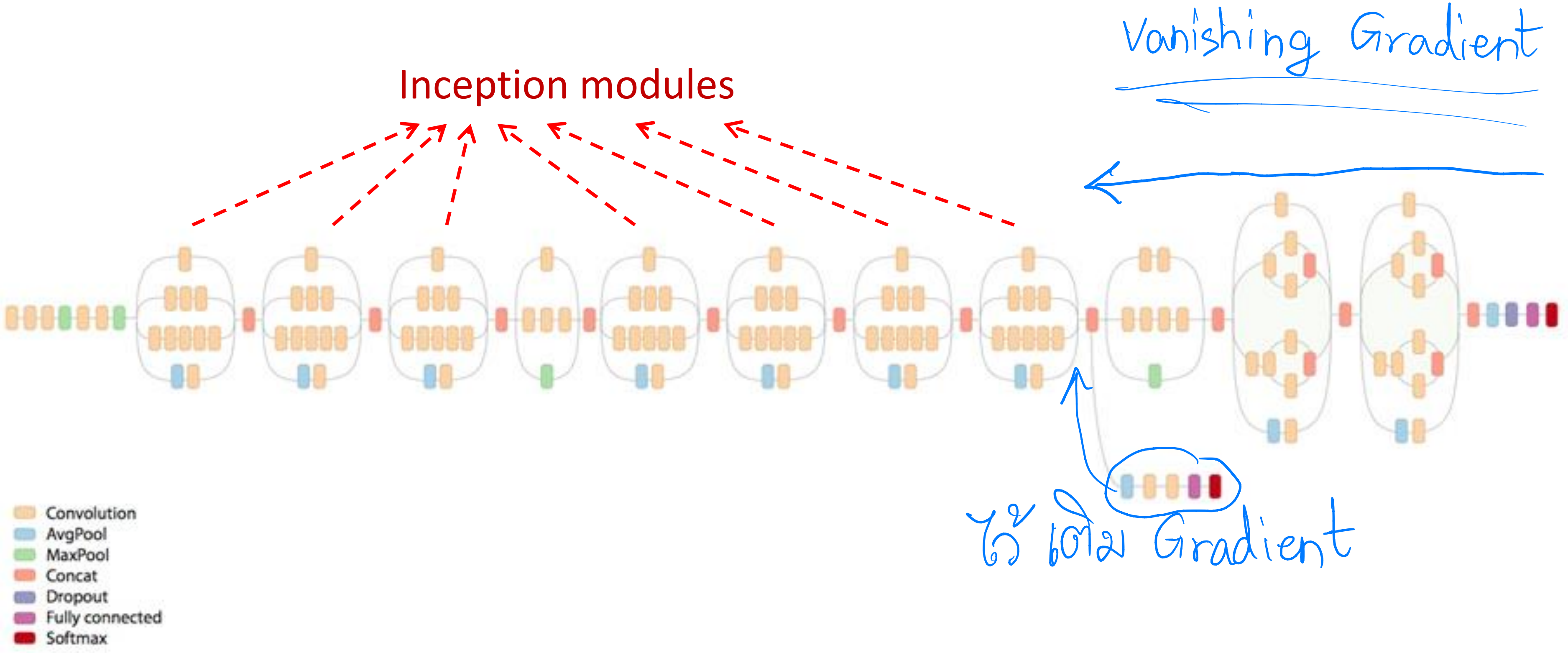
Relu (op#5)

Conv (op#6)

...

สไลด์โดนตัด

GoogLeNet



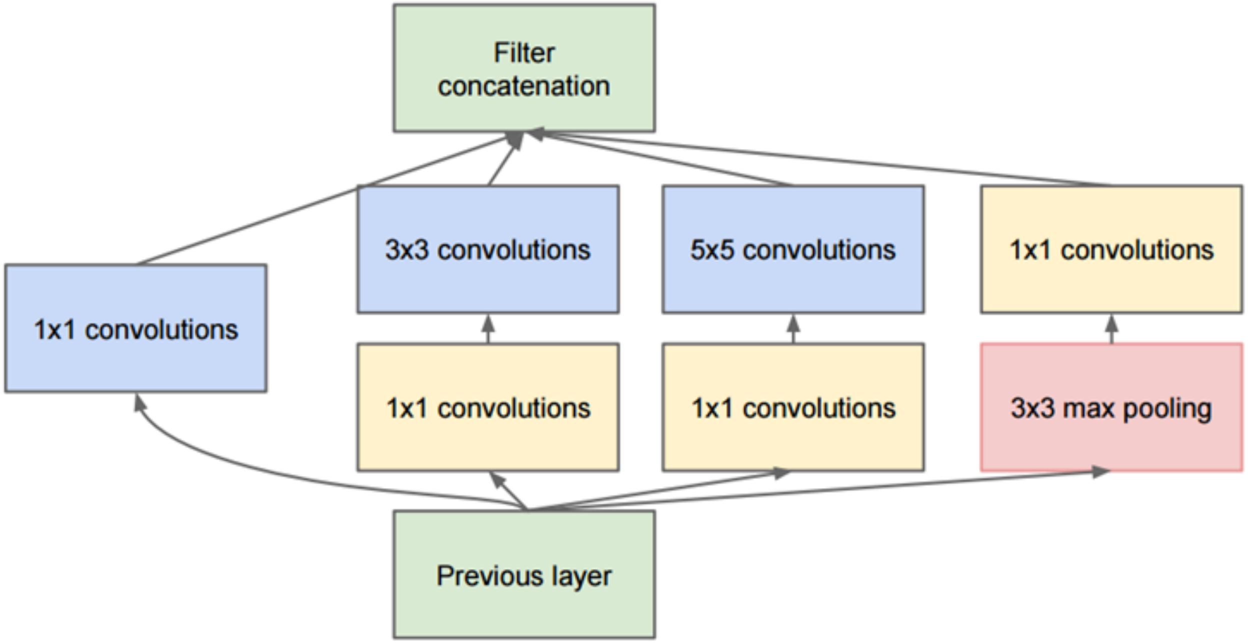
Another view of GoogLeNet's architecture.

- GoogLeNet consists of a series of inception modules.
- At each layer, you perform operations (i.e., a pooling operation / a conv operation) in parallel.

GoogLeNet

Inception module

- The network in network conv is able to extract information about the very fine grain details in the volume, while the 5x5 filter is able to cover a large receptive field of the input.

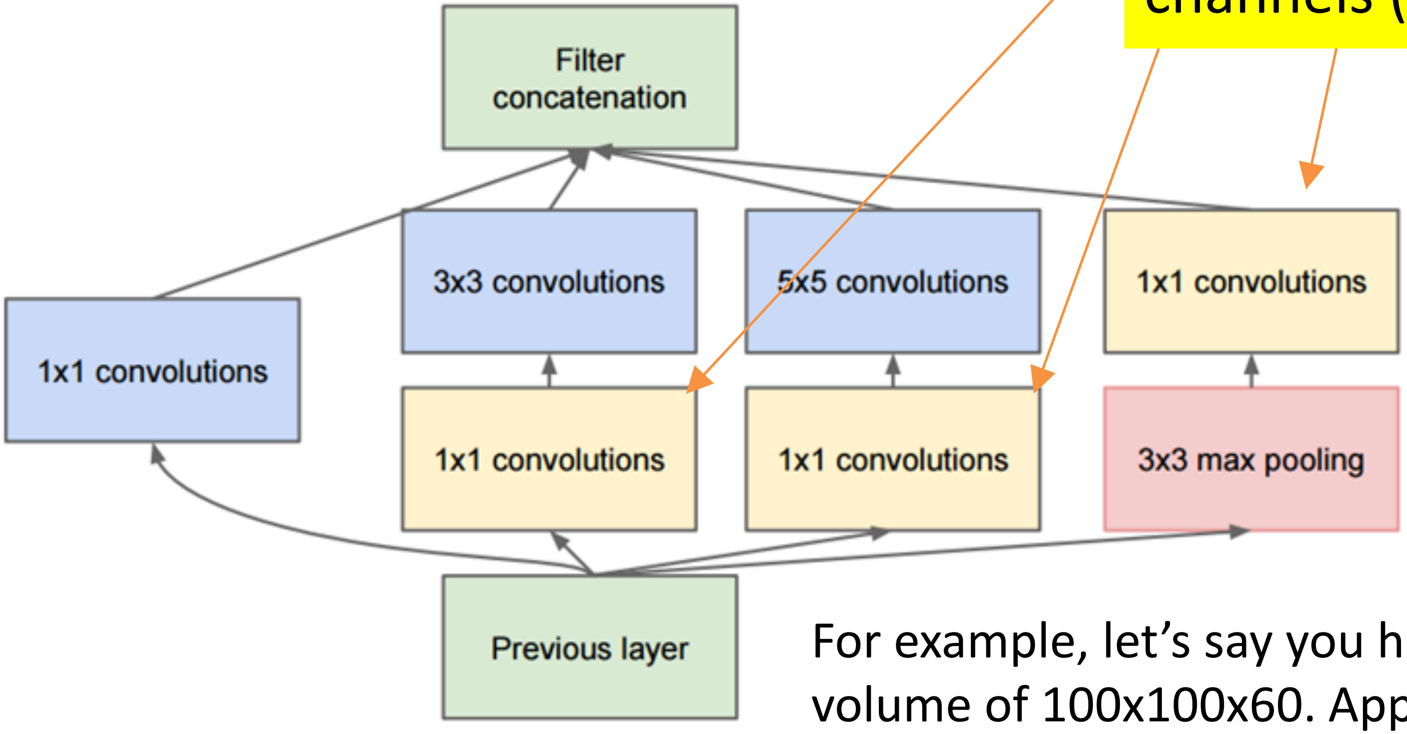


Full Inception module

GoogLeNet

Inception module

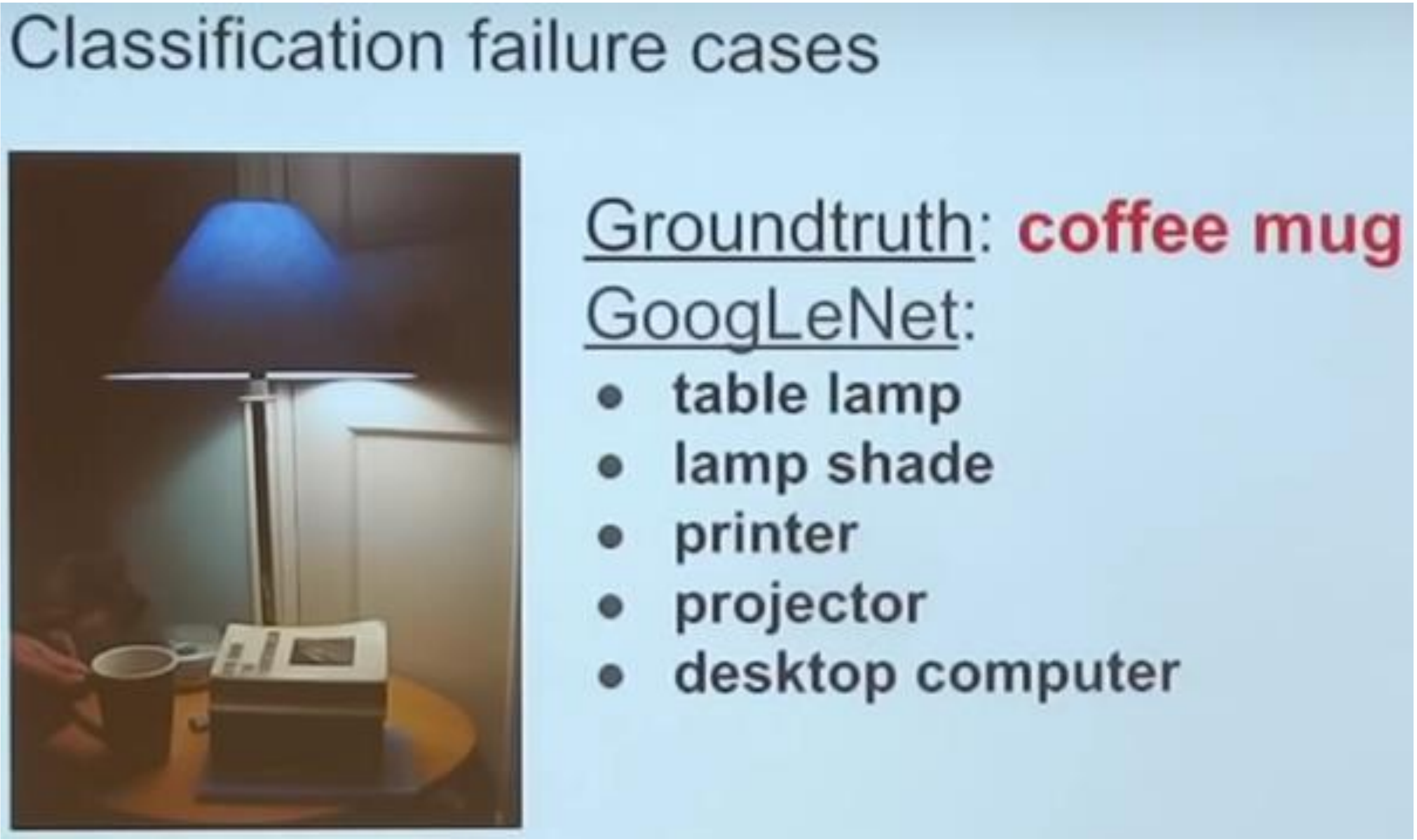
1x1 convolutions reduce the number of channels (dimensions)



Full Inception module

For example, let's say you had an input volume of 100x100x60. Applying 20 filters of 1x1 convolution would allow you to reduce the volume to 100x100x20.

GoogLeNet



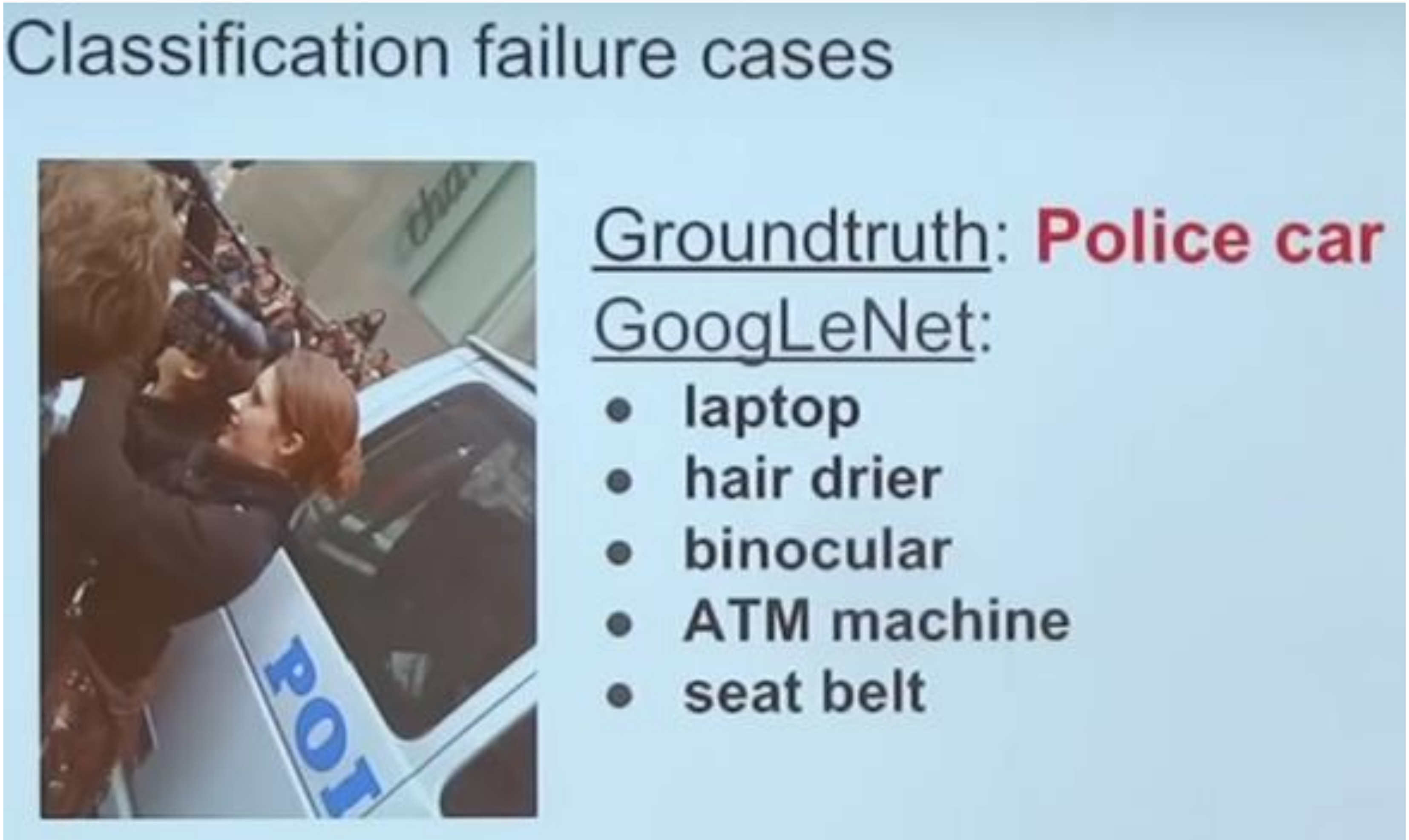
Classification failure cases

Groundtruth: **coffee mug**

GoogLeNet:

- **table lamp**
- **lamp shade**
- **printer**
- **projector**
- **desktop computer**

GoogLeNet



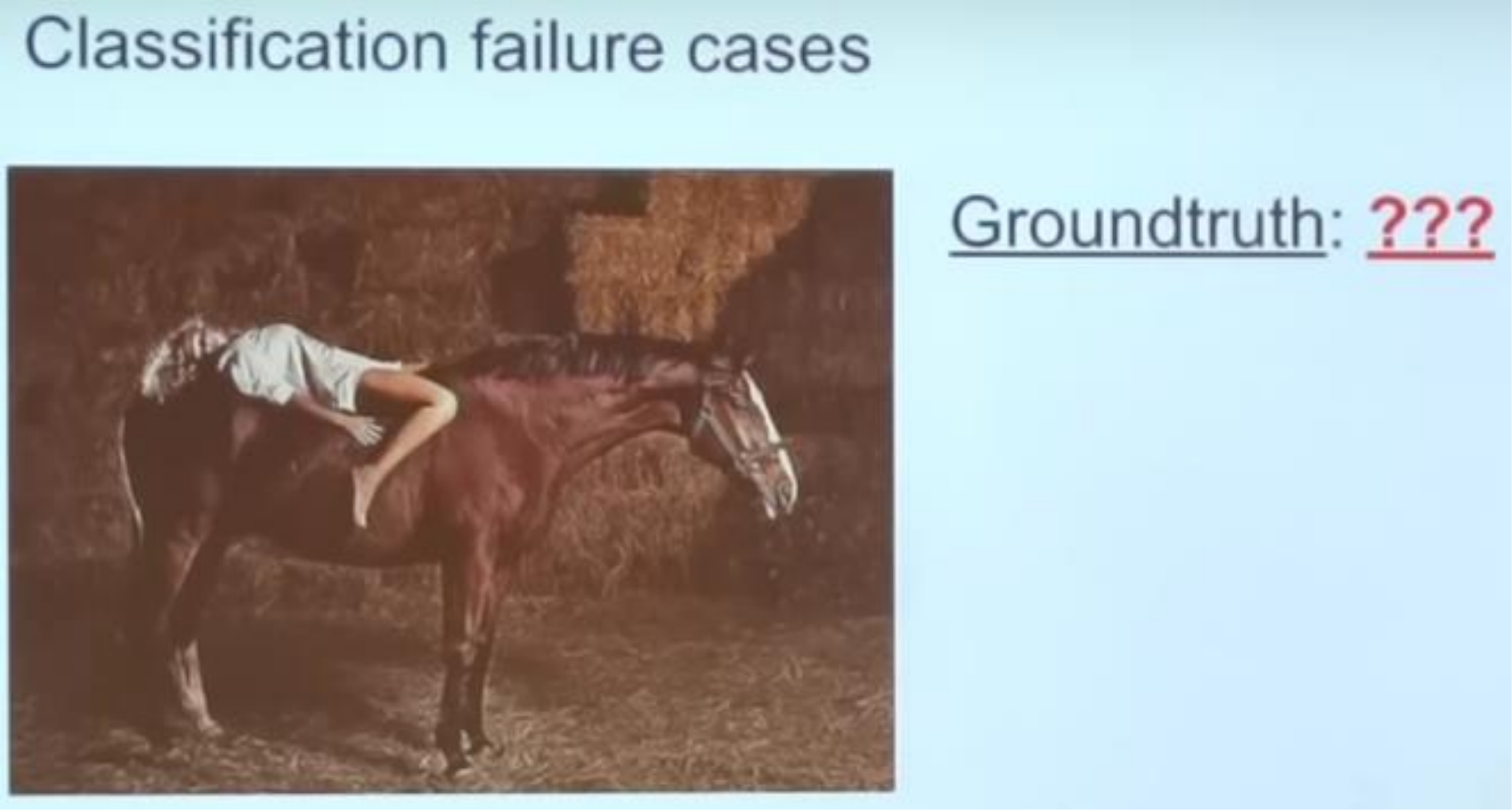
Classification failure cases

Groundtruth: **Police car**

GoogLeNet:

- **laptop**
- **hair drier**
- **binocular**
- **ATM machine**
- **seat belt**

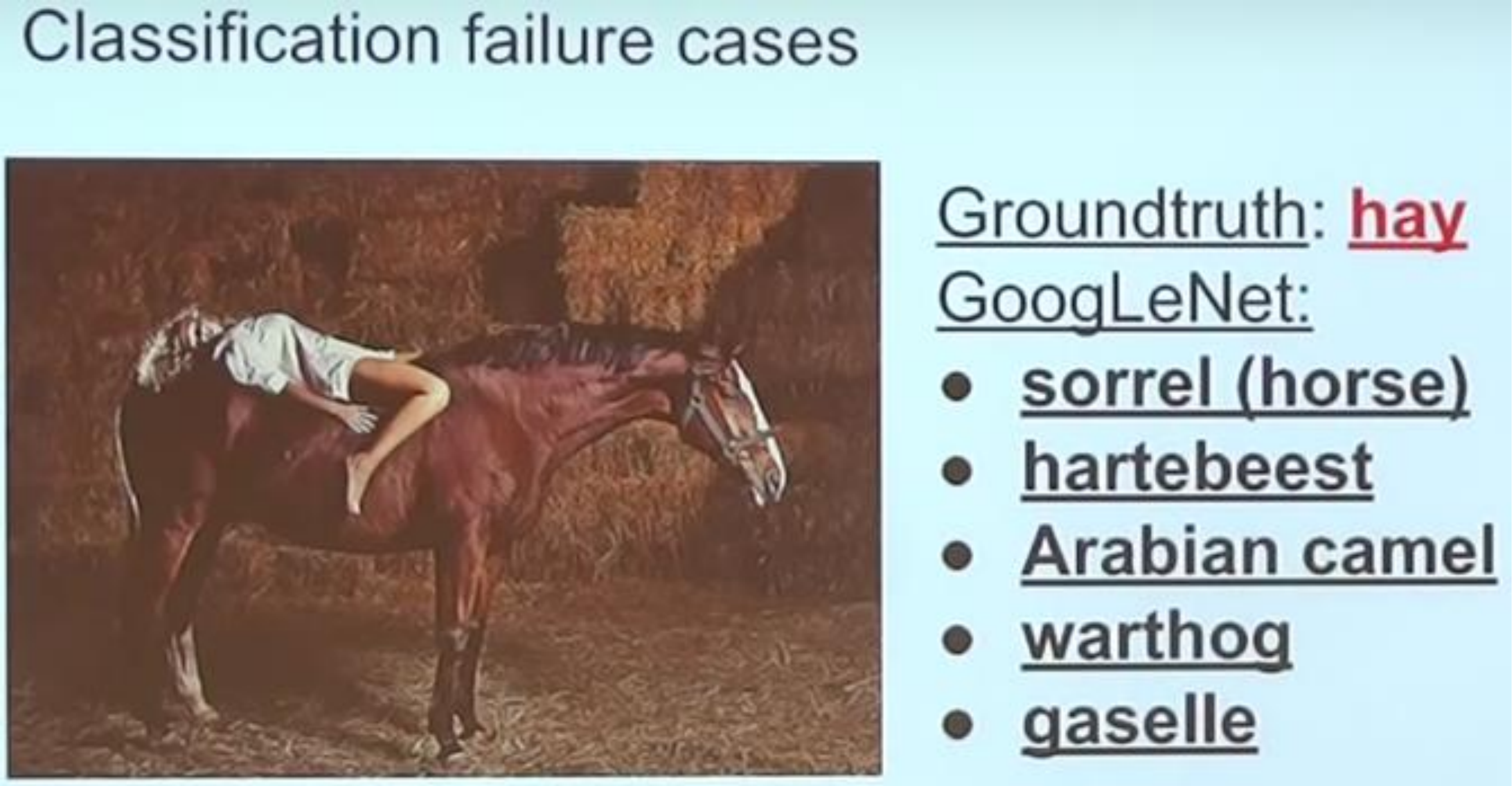
GoogLeNet



Classification failure cases

Groundtruth: ???

GoogLeNet



Classification failure cases

Groundtruth: hay

GoogLeNet:

- sorrel (horse)
- hartebeest
- Arabian camel
- warthog
- gaselle

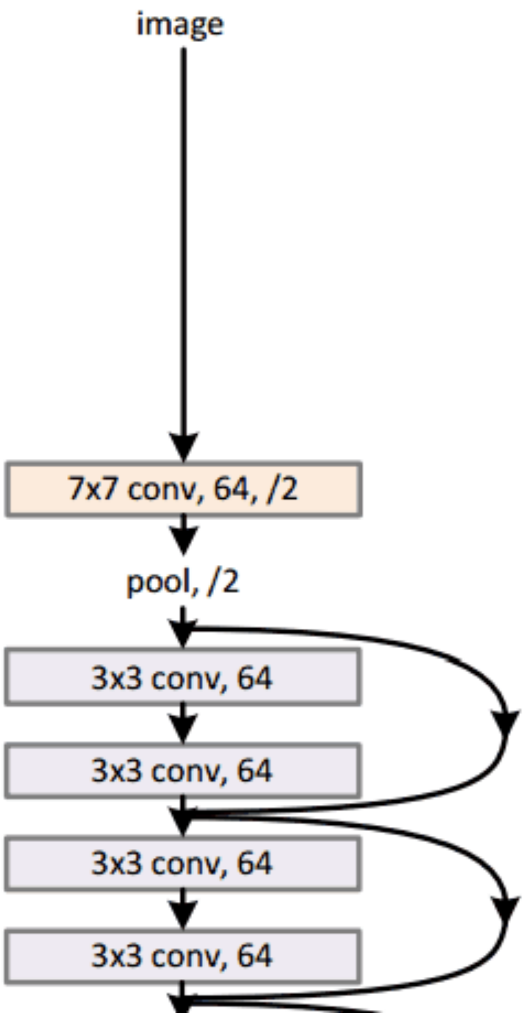
Microsoft ResNet (2015)

- **ResNet** is a new **152-layer** network architecture that set new records in classification and localization through one incredible architecture.
- ResNet is composed of a series of **Residual blocks**.
- **ResNet** won ILSVRC 2015 with an incredible **error rate of 3.6%**. Humans generally hover around a 5-10% error rate.
- The group tried a 1202-layer network, but got a lower test accuracy, presumably due to overfitting. (**Deeper is better, but beware of overfitting!**)
- Trained on an 8 GPU machine for **two to three weeks**.

- Original paper: https://arxiv.org/pdf/1512.03385v1.pdf

Microsoft ResNet

34-layer residual

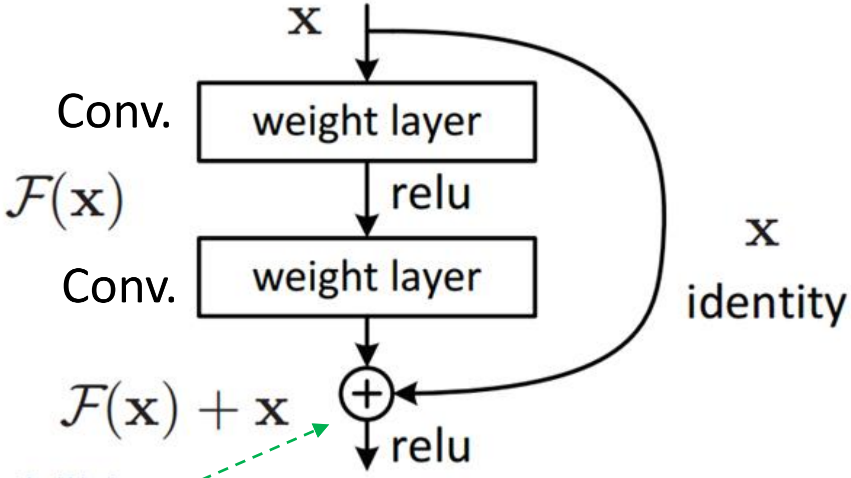


สไลด์โดนตัด

Microsoft ResNet

Residual Block

- Input x goes through **conv-relu-conv** series.
- Then **add** the result to the **original input x**.
- This architecture is effective since during the backward pass of backpropagation, **the gradient will flow easily through the graph** because we have addition operations, which distributes the gradient.



A residual block

- After the *first 2* layers, the **spatial size gets compressed** from an input volume of 224x224 to a 56x56 volume.