Convolutional Neural Networks

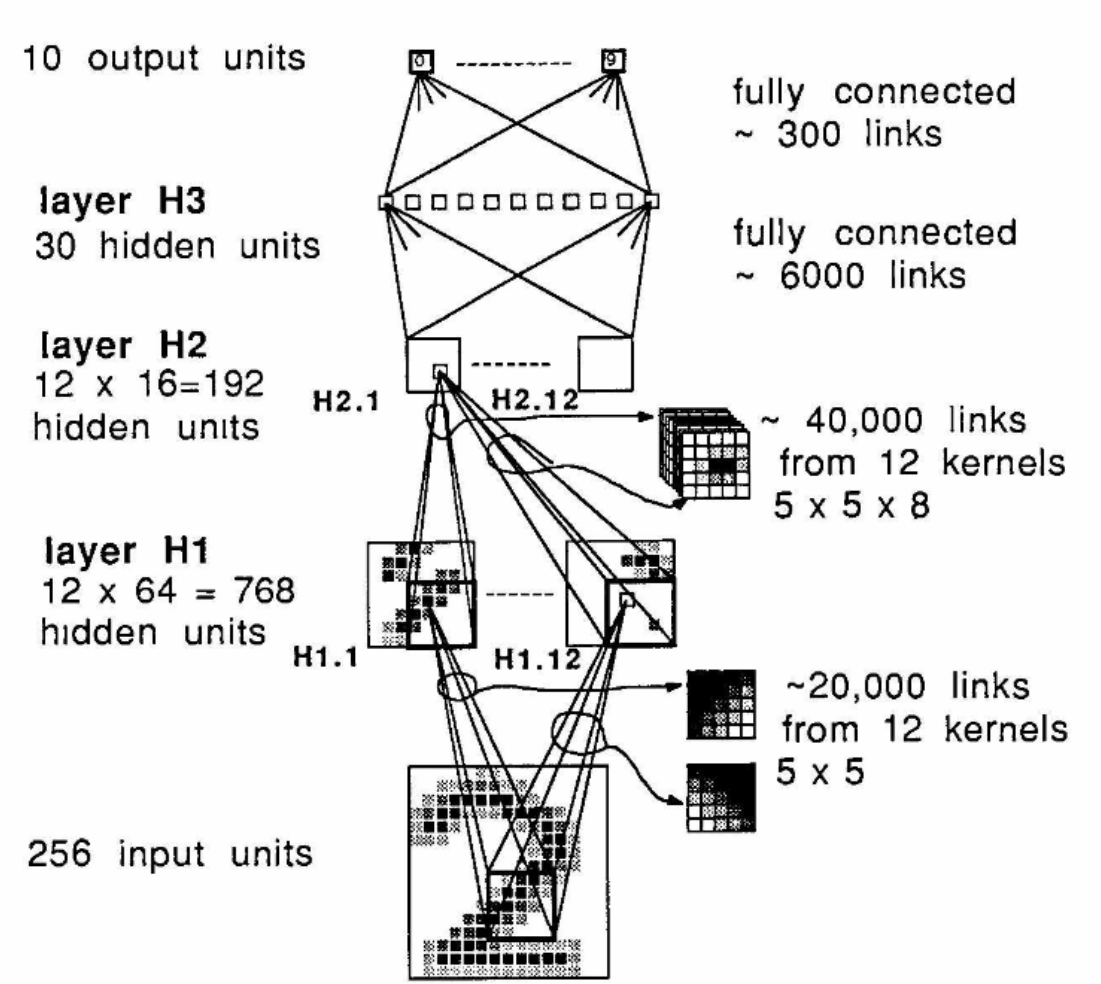
IE643 - Lectures 15, 16

Oct 1 & 4, 2024.

Convolutional Neural Networks (CNNs)

At New York Postal Exchange

The Birth of Convolutional Neural Nets



Backpropagation Applied to Handwritten Zip Code Recognition. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. Neural Computation, vol.1(4), 1989.

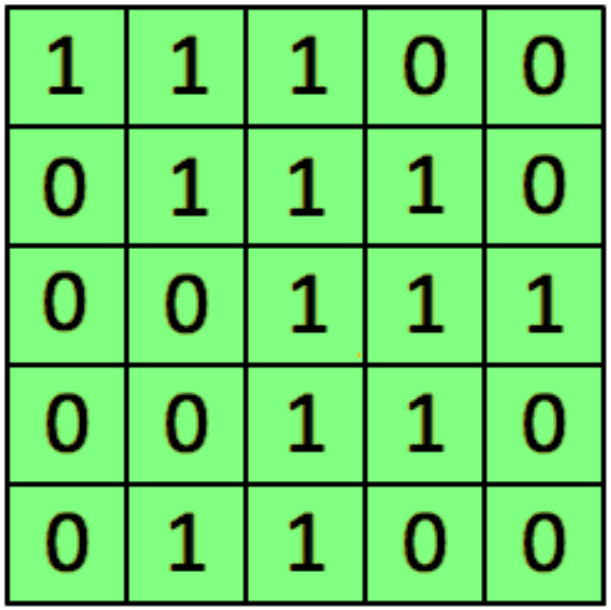
CNN

What We See

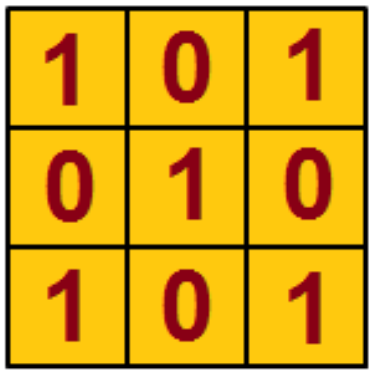
What Computers See

Image Source: `https://adeshpande3.github.io/assets/Corgi3.png`

CNN- Convolution



Image



Convolution Filter

CNN- Convolution

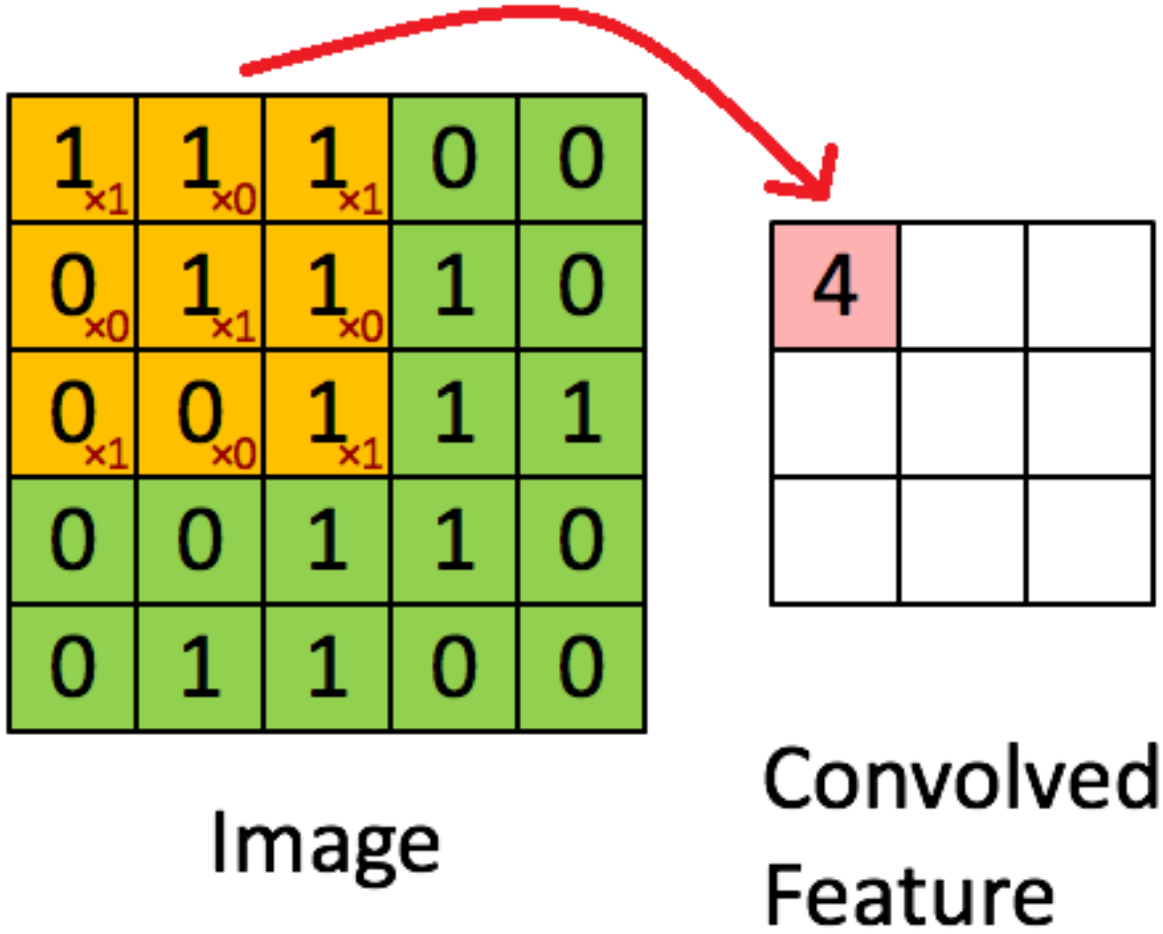
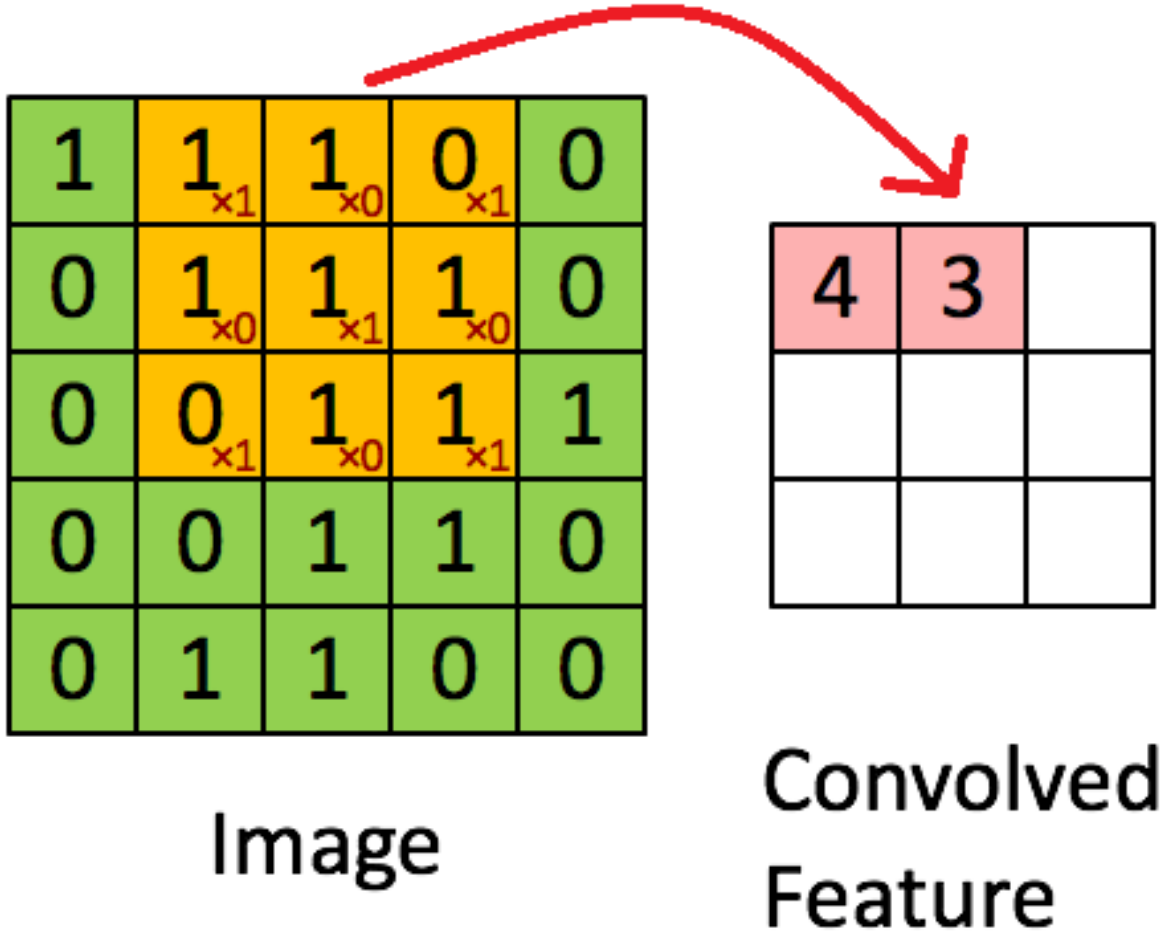
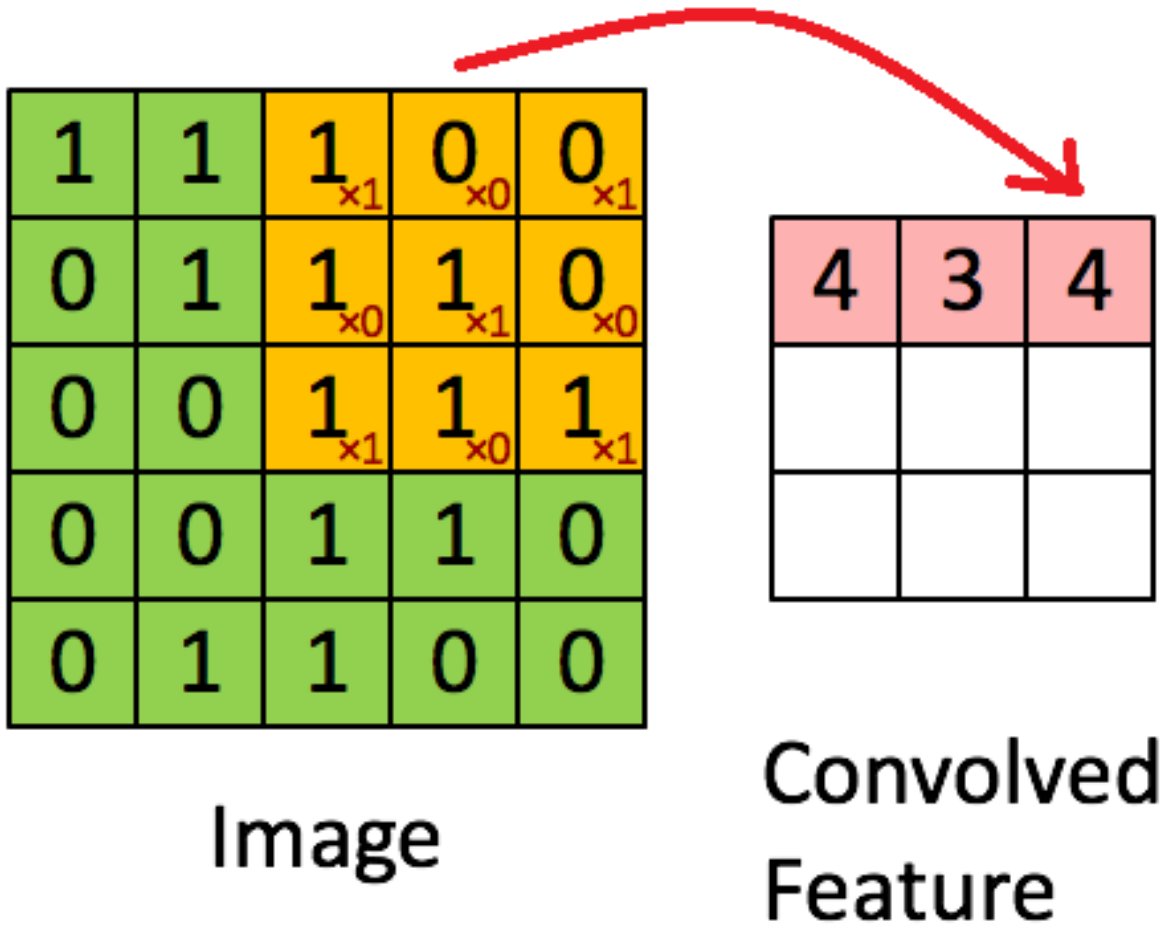


Image Source:
https://ujwlkarn.files.wordpress.com/2016/07/convolution_schematic.gif

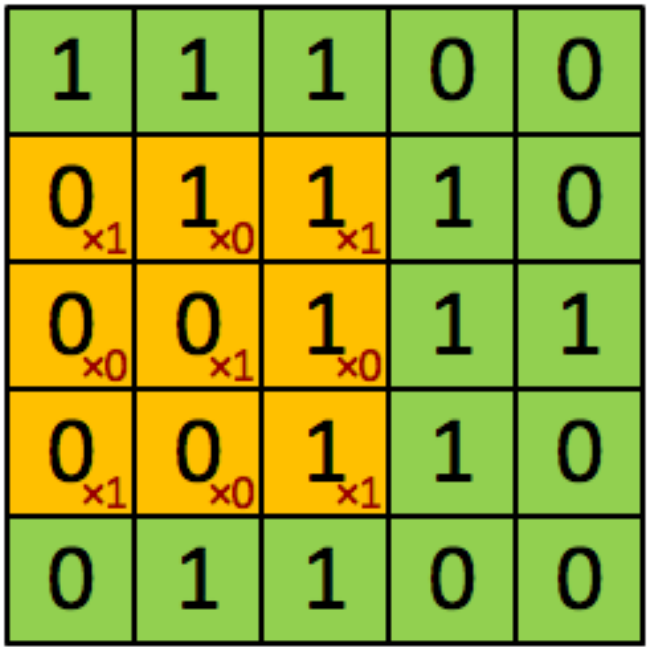
CNN- Convolution



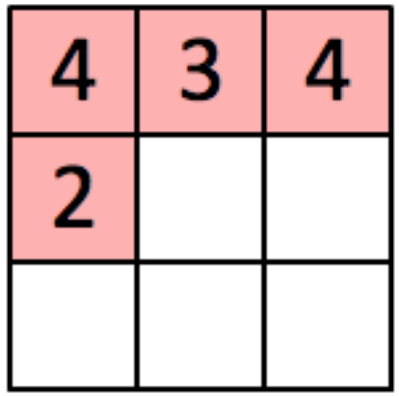
CNN - Convolution



CNN - Convolution

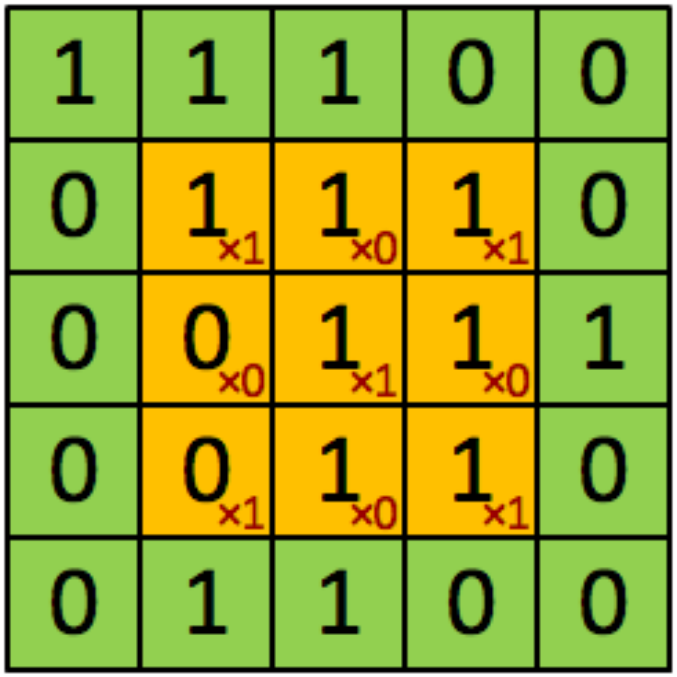


Image

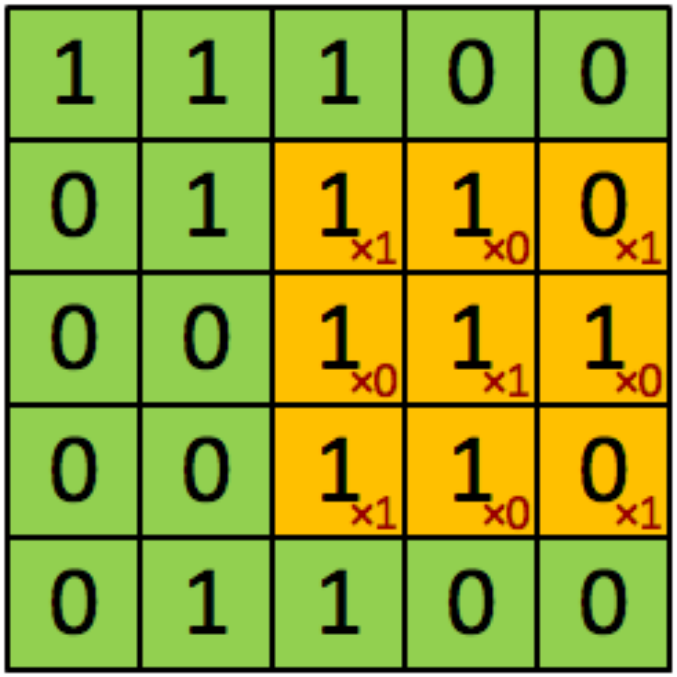


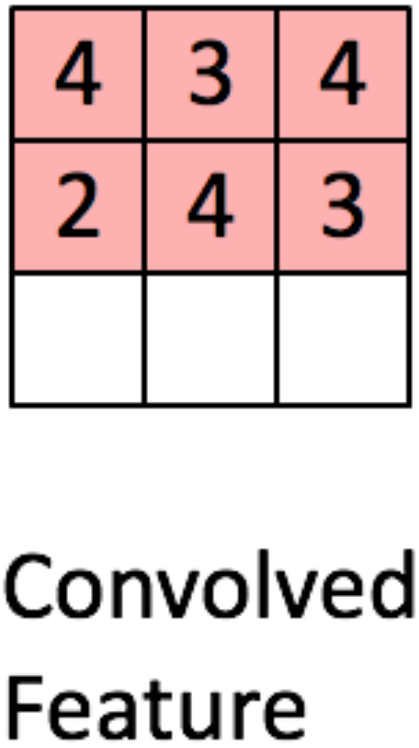
Convolved Feature

CNN - Convolution

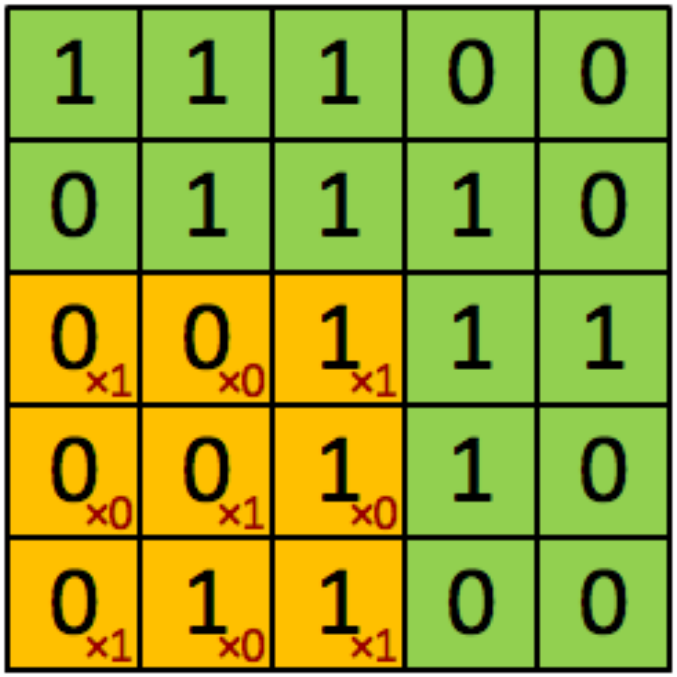


CNN - Convolution

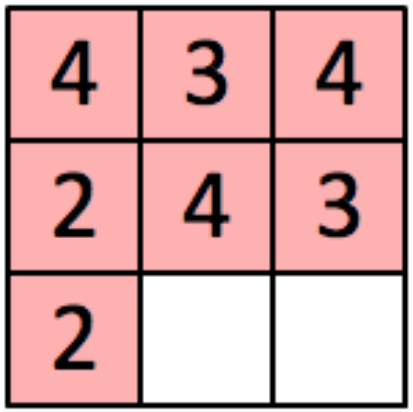




CNN - Convolution

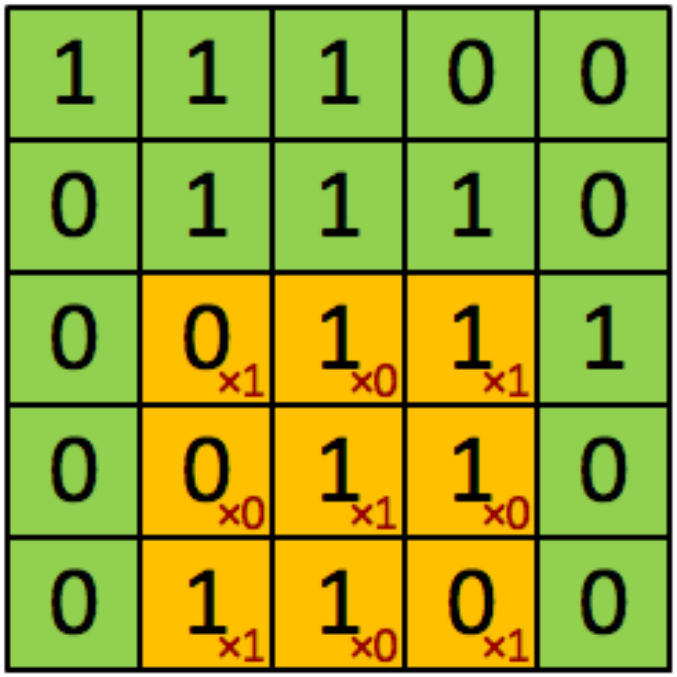


Image

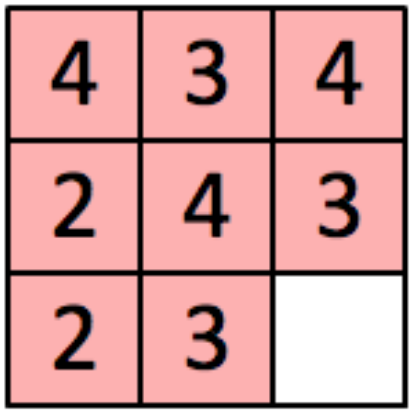


Convolved Feature

CNN - Convolution

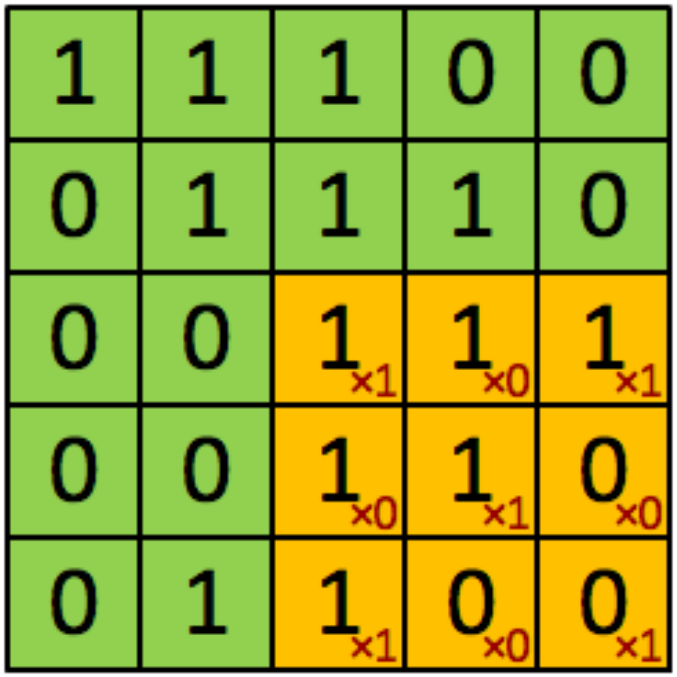


Image



Convolved Feature

CNN - Convolution

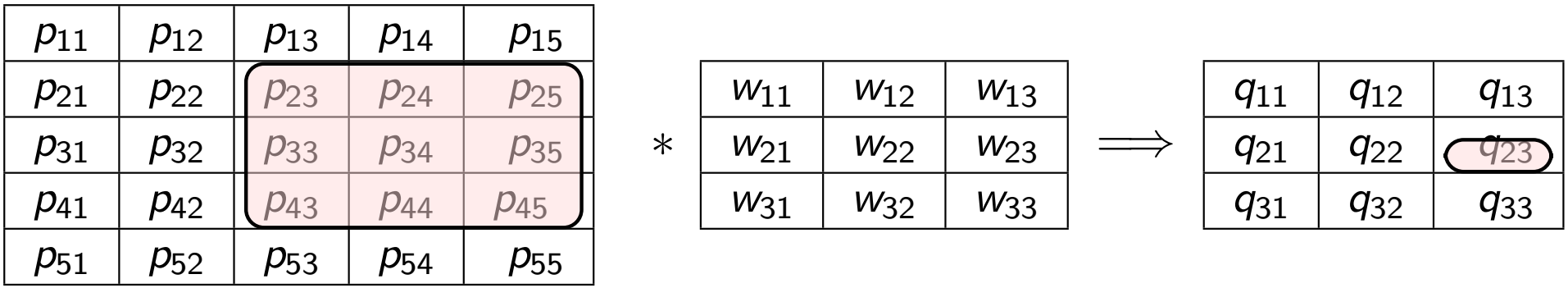


Image

4	3	4
2	4	3
2	3	4

Convolved Feature

CNN - Convolution with stride=1



Note: $kernelwidth = 3$, $stride = 1$

$$q_{ij} = \sum_{r=i}^{(i+kernelwidth-1)} \sum_{s=j}^{(j+kernelwidth-1)} p_{rs} w_{(r-i+1)(s-j+1)}$$

Exercise: Find the formula for $stride > 1$

CNN - Significance of Convolution

- Convolution operation is robust to ??
- Helps to extract local features which might be useful for the task.
- Significance of striding?

CNN - Significance of Convolution

- Changing the filter size (or kernel width) $\Longrightarrow$??
- Can we use a number of filters? If so, what sizes?
- Recall: In LeCun et al's work, **24** filters were used !
- What features do these convolutions capture?
  Answer:

CNN - Zero Padding

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Image Source: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=146227

CNN - Zero Padding

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

- Generally used to take care of the corner and border pixels
- Used to make the convolution operations unbiased towards these pixels

CNN - Convolution on single channel image

$$n_{out} = \left\lfloor \frac{n_{in} + 2 * \text{padsize} - kernelwidth}{stride} \right\rfloor + 1$$

- n_{out} = number of features in output image
- n_{in} = number of features in input image

CNN - RGB Convolution

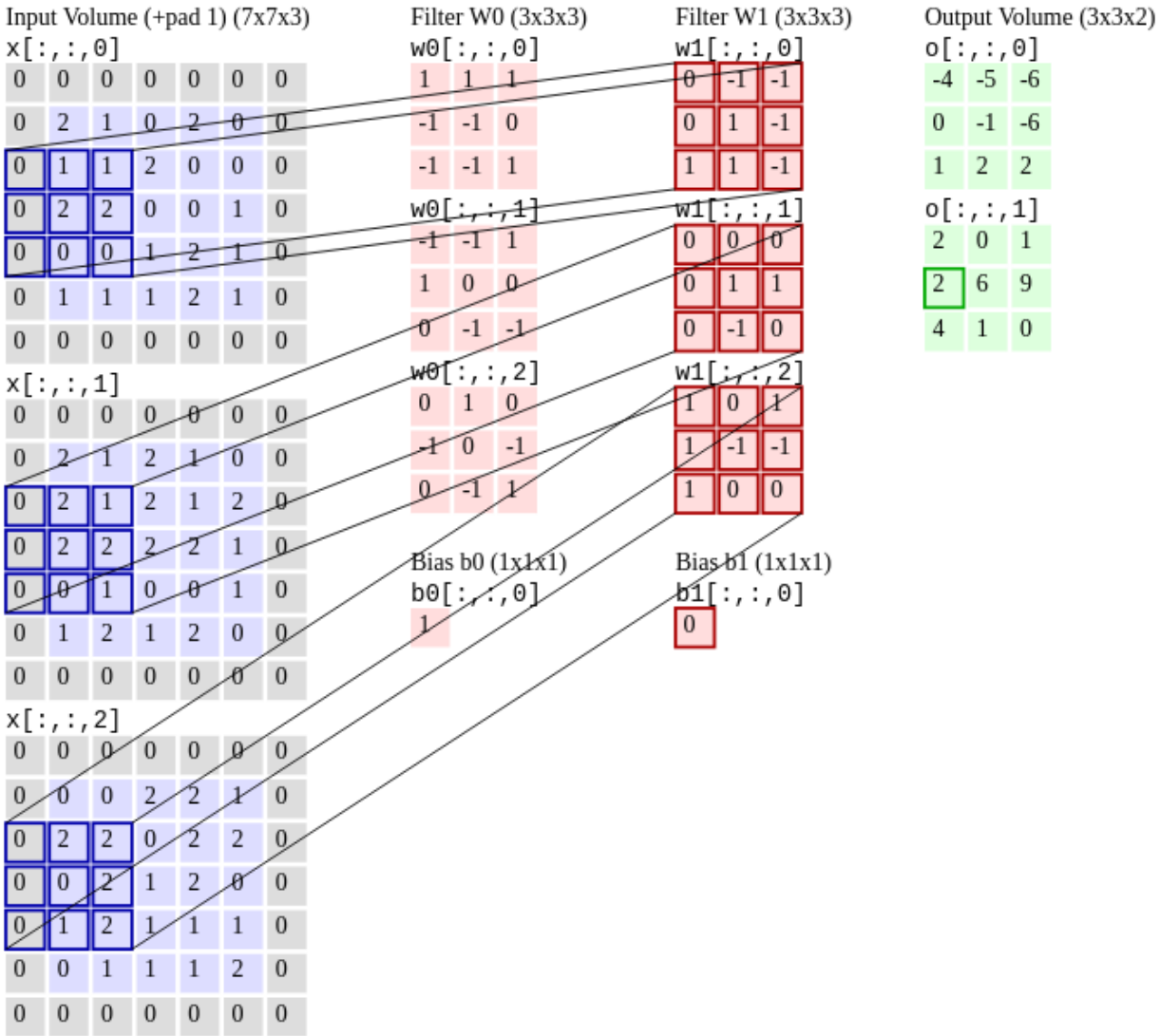


Image Source: http://cs231n.github.io/convolutional-networks/

CNN - Pooling

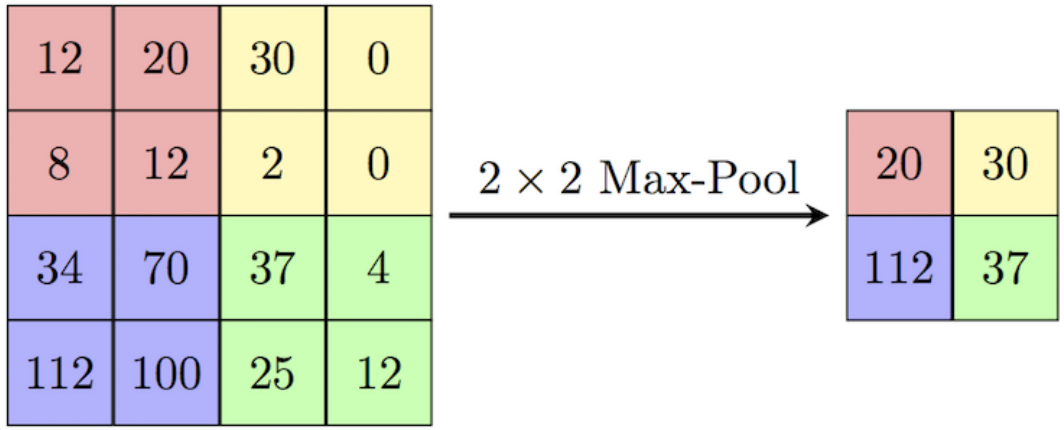


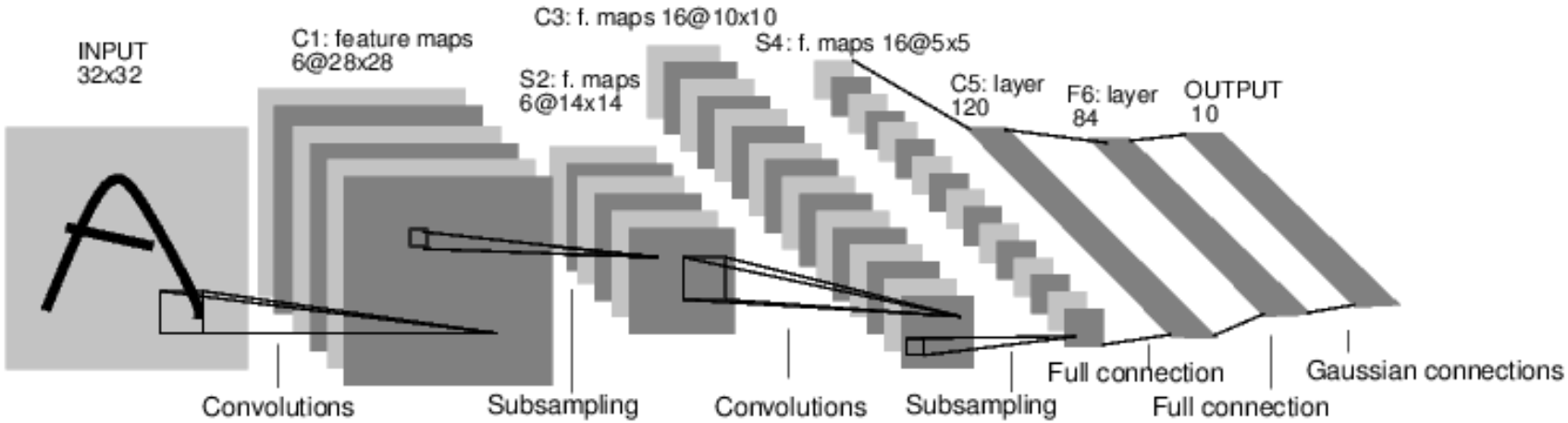
Image Source:
https://computersciencewiki.org/index.php/Max-pooling_/_Pooling

CNN - Pooling

- Max-pooling sometimes called sub-sampling. Why?
- Max-pooling operation preserves what?
- Effect of successive max-pooling operations with zero-padding ?

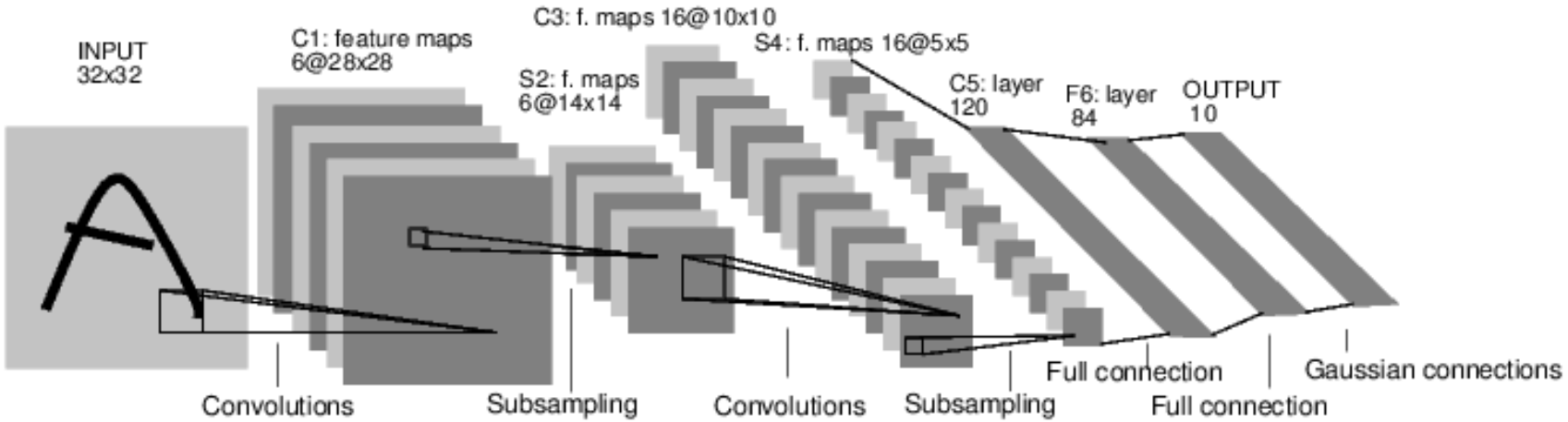
Popular CNN Architectures

CNN - LeNet Architecture



Gradient-based learning applied to document recognition.
Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. Proc. of the IEEE, 1998.

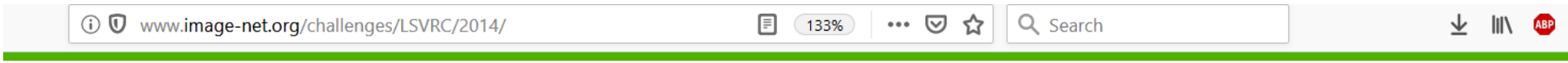
CNN - LeNet Architecture



Gradient-based learning applied to document recognition.
Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. Proc. of the IEEE, 1998.

- Achieved $\sim$ 92% accuracy on handwritten digit recognition

ILSVRC 2014 Challenge



IMAGENET Large Scale Visual Recognition Challenge 2014 (ILSVRC2014)

Introduction History Data Tasks FAQ Development kit Timetable Citation[new] Organizers Sponsors Contact

News

- June 2, 2015: Additional announcement regarding submission server policy is released.
- May 19, 2015: Announcement regarding submission server policy is released.
- December 17, 2014: ILSVRC 2015 is announced.
- September 2, 2014: A new paper which describes the collection of the ImageNet Large Scale Visual Recognition Challenge dataset, analyzes the results of the past five years of the challenge, and even compares current computer accuracy with human accuracy is now available. *Please cite it when reporting ILSVRC2014 results or using the dataset.*
- August 18, 2014: Check out the New York Times article about ILSVRC2014.
- August 18, 2014: Results are released.
- August 18, 2014: Test server is open.
- July 25, 2014: Submission server is now open.
- July 15, 2014: Computational resources available, courtesy of NVIDIA.
- July 3, 2014: **Please note that the August 15th deadline is firm this year and will not be extended.**
- June 25, 2014: You can now browse all annotated detection images.
- May 3, 2014: ILSVRC2014 development kit and data are available. Please register to obtain the download links.

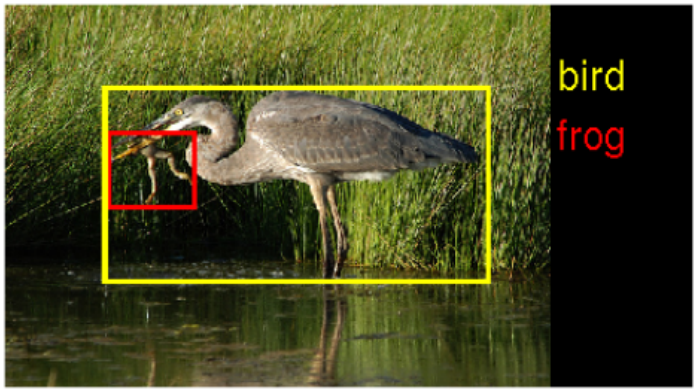
Image Net Data

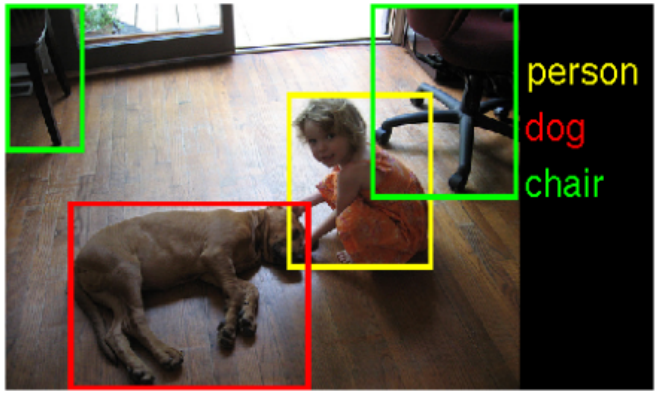
ImageNet: A Large-Scale Hierarchical Image Database

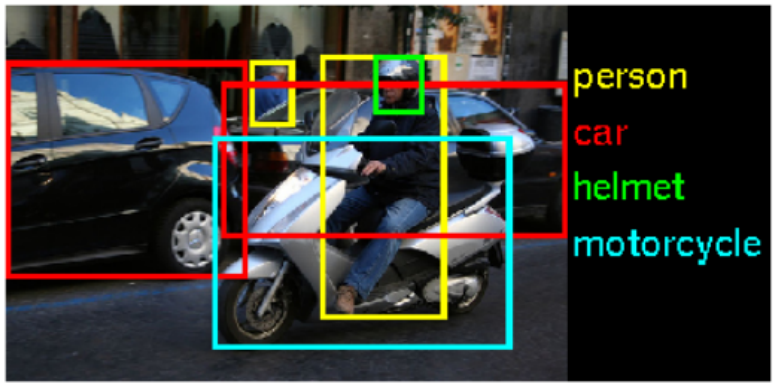
Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei
Dept. of Computer Science, Princeton University, USA
{jiadeng, wdong, rsocher, jial, li, feifeili}@cs.princeton.edu

ILSVRC 2014 Challenge

Example ILSVRC2014 images:







Two Tasks:
1. Object Detection
2. Classification

Image Net Data

- **Detection Task:** 200 classes, $\sim$ 457K images in training set, $\sim$ 60K images in test and validation set.
  - Aim: To predict the boundaries of bounding boxes.
- **Classification and Localization Task:** 1000 categories, validation and test data consist of 150,000 photographs collected from Flickr and other search engines.
  - Aim: To find the categories of objects and their localization (presence in the form of bounding boxes) in the images.

www.kaggle.com/c/imagenet-object-localization-challenge

Pascal VOC Data

- Goal of the challenge: To recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects)
- Person: person
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

---

Webpage: `http://host.robots.ox.ac.uk/pascal/VOC/voc2007/`
`http://host.robots.ox.ac.uk/pascal/VOC/voc2012/`

Pascal VOC Data

Two main objectives:

- Classification
- Detection

CNN - Alex Net

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

CNN - Alex Net Architecture

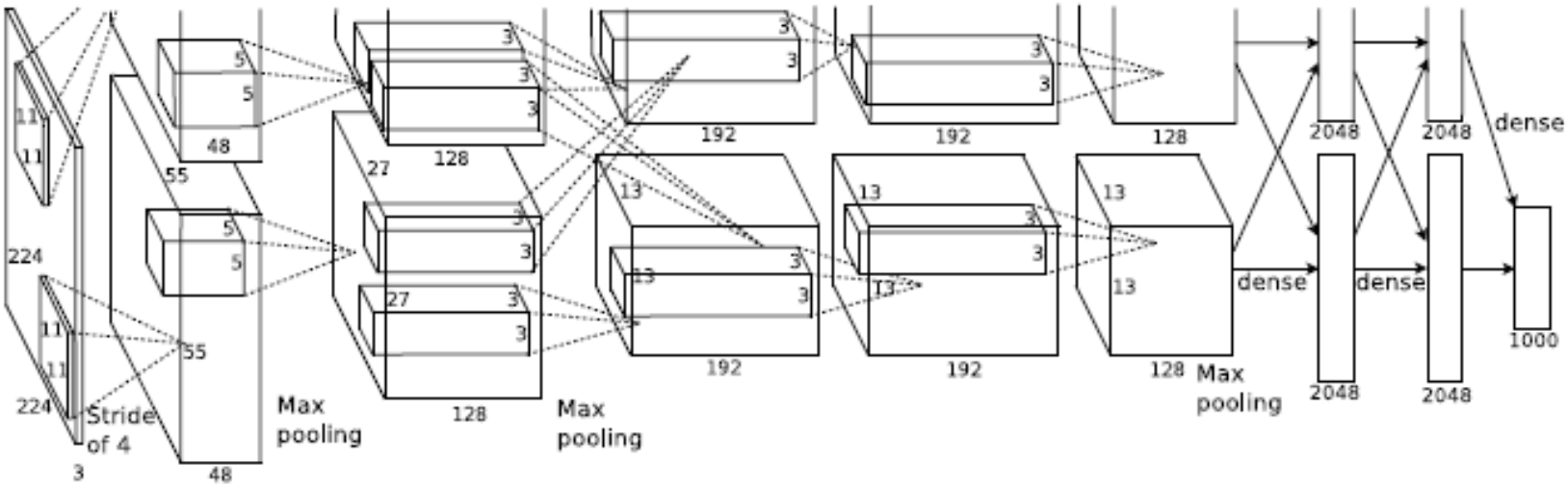


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

CNN - Alex Net

Peculiarities:

- First efficient GPU implementation of 2D convolution.
- ReLU non-linearity used
- Training done on multiple GPUs
- Local response normalization applied after ReLU

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

- $k = 2$, $\alpha = 10^{-4}$, $n = 5$ and $\beta = 0.75$ chosen using cross-validation
- Local response normalization decreased the error rate
- Overlapping pooling was employed instead of usual max pooling. (3×3 window with stride of length 2)

CNN - Alex Net

Peculiarities during Training:

- Data augmentation using
  - image translations
  - horizontal reflections
- Original image size: 256×256.
  - For larger images in the data set, the shorter side was first rescaled to 256 and then central 256×256 patch was cropped.
- Mean was subtracted from individual pixel values.
- Patches of size 224×224 considered by random patching and their reflections
- Dropout considered
- Training took 5 to 6 days to train on two GTX 580 3GB GPUs.

CNN - Alex Net

Peculiarities during Testing:

- M patches of size 224×224 considered for an input image
- Predictions were averaged over M patches
- All neurons' output was multiplied by 0.5 (to take dropout into account)

CNN - Alex Net Architecture

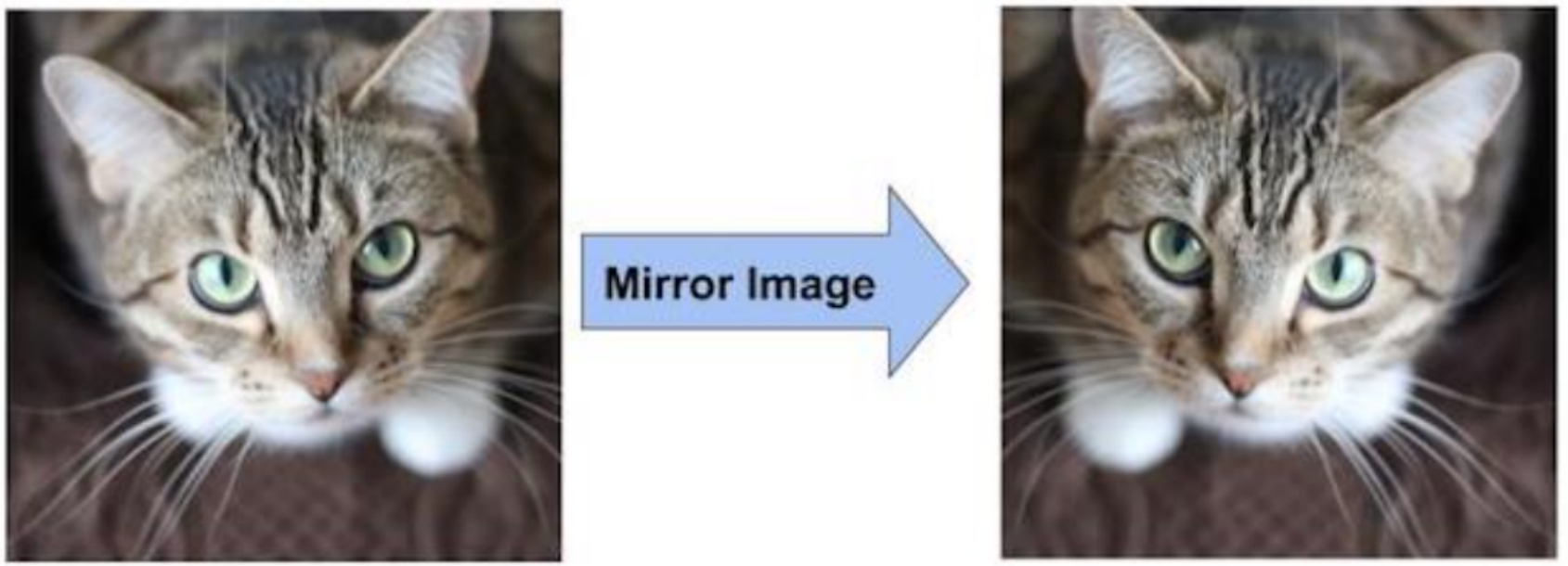


Figure: Data augmentation - mirroring*

* https://learnopencv.com/understanding-alexnet/

CNN - Alex Net Architecture

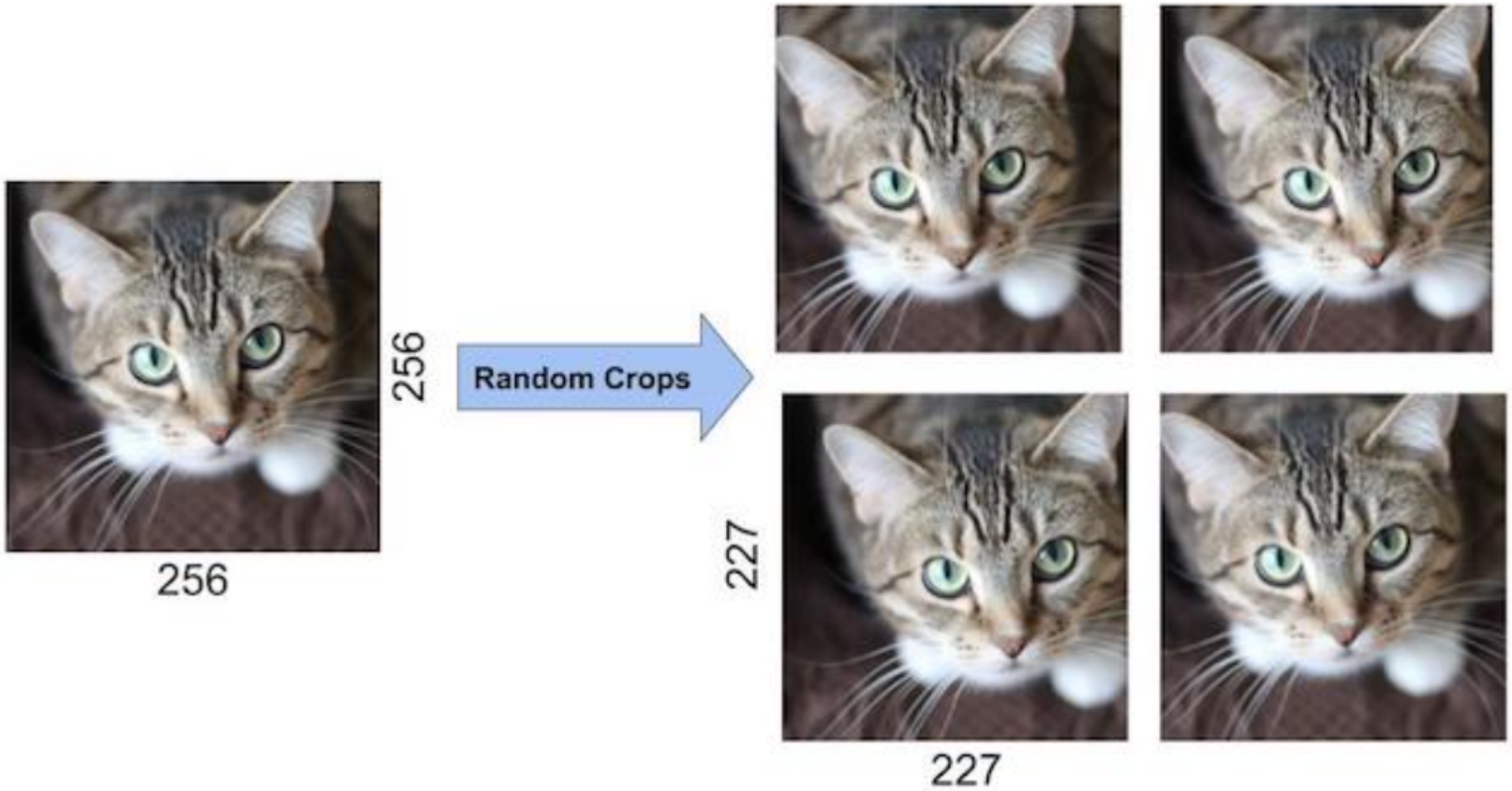


Figure: Data augmentation - random crop*

* https://learnopencv.com/understanding-alexnet/

CNN - Alex Net Architecture

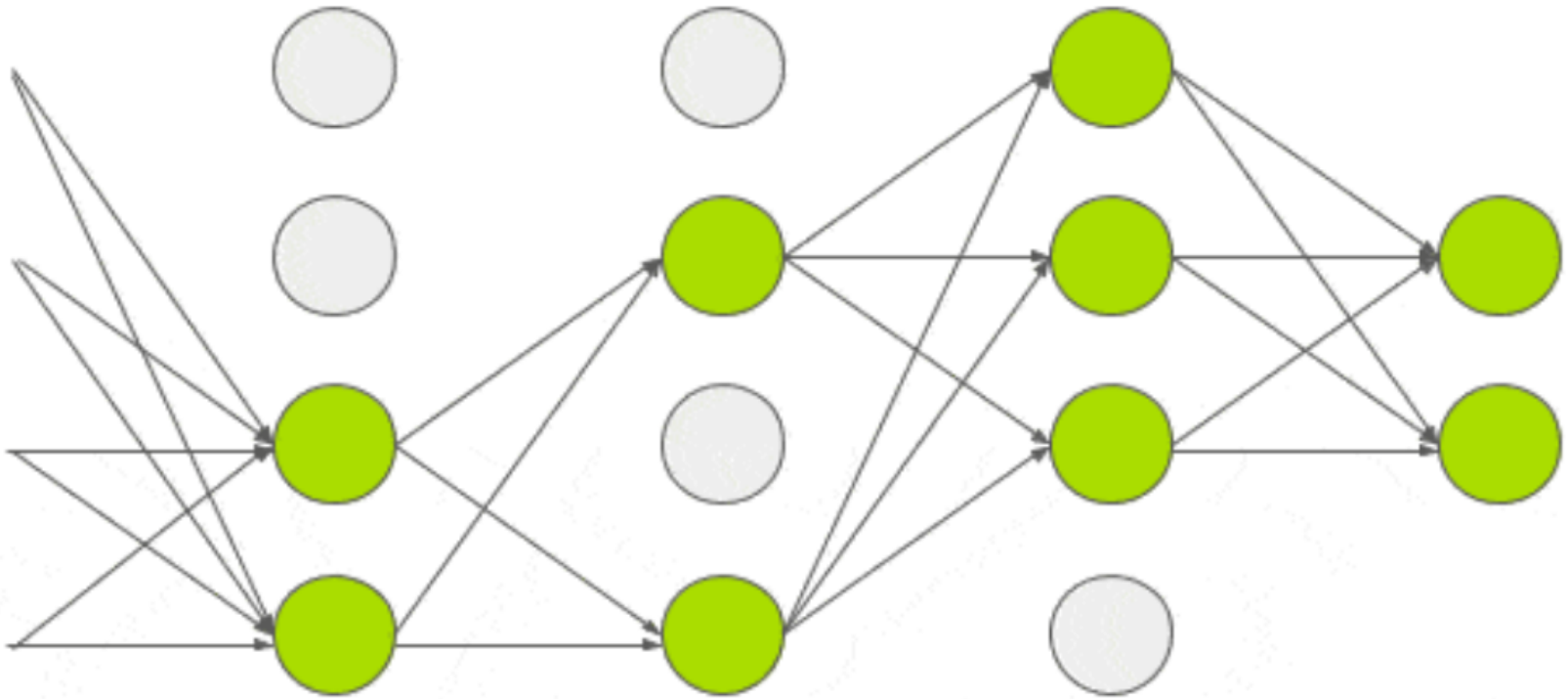


Figure: Dropout mechanism*

* https://learnopencv.com/understanding-alexnet/

CNN - Alex Net Architecture

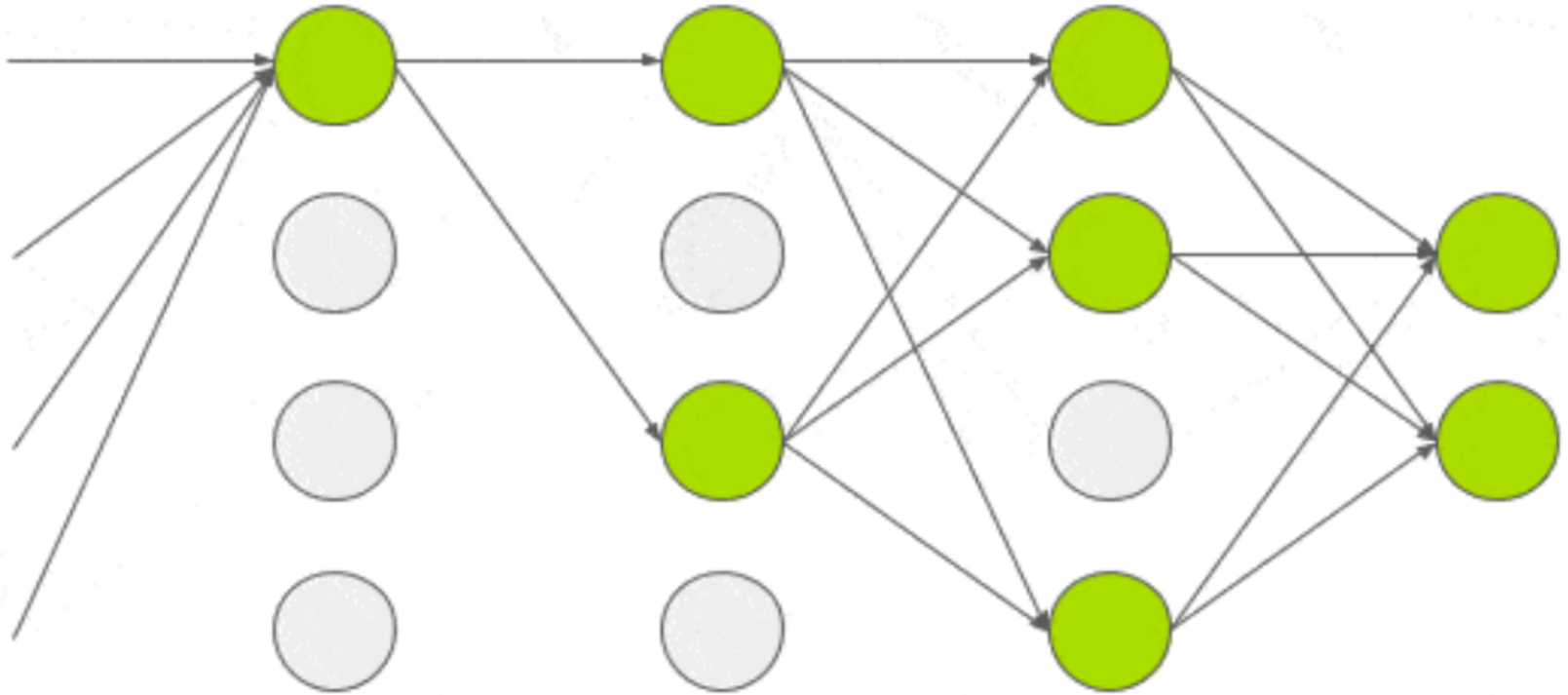


Figure: Dropout mechanism

CNN - Alex Net Architecture

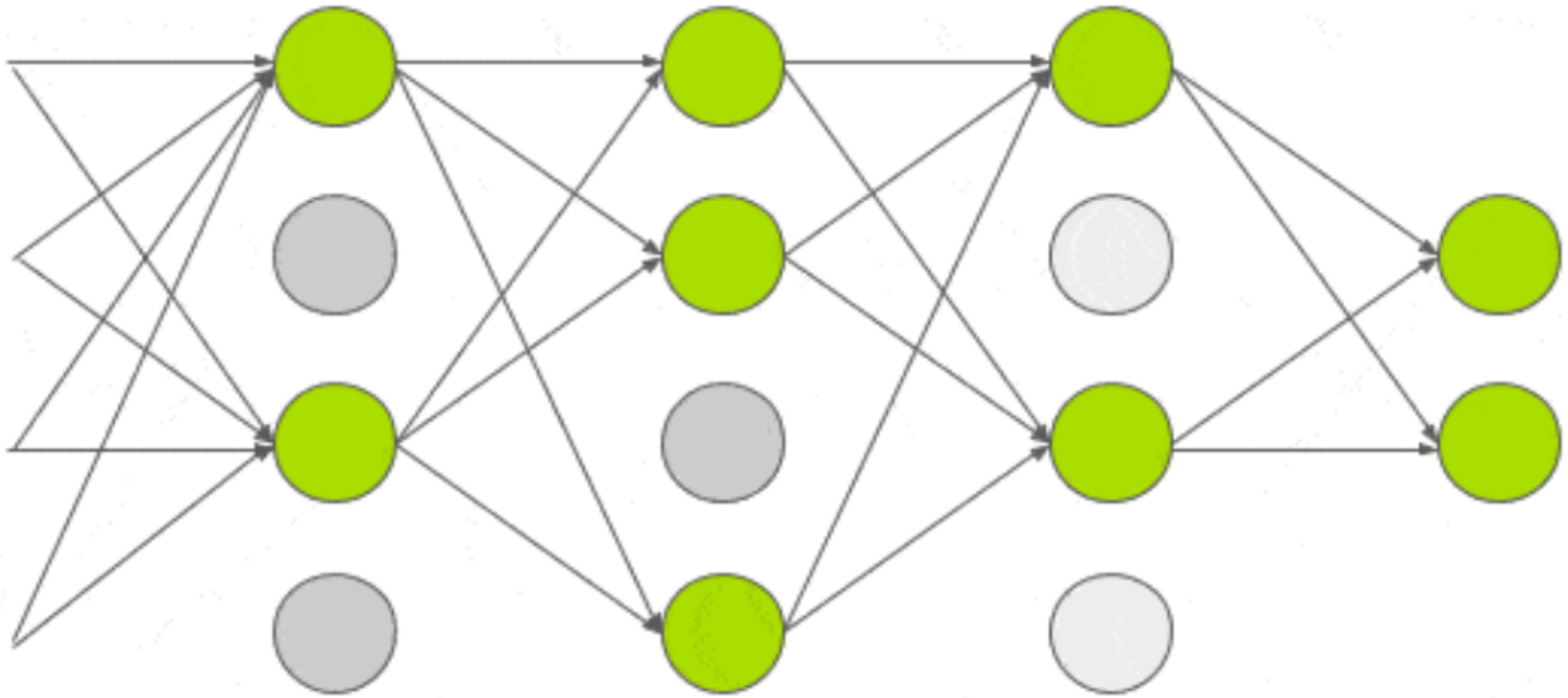


Figure: Dropout mechanism

CNN - Alex Net Architecture

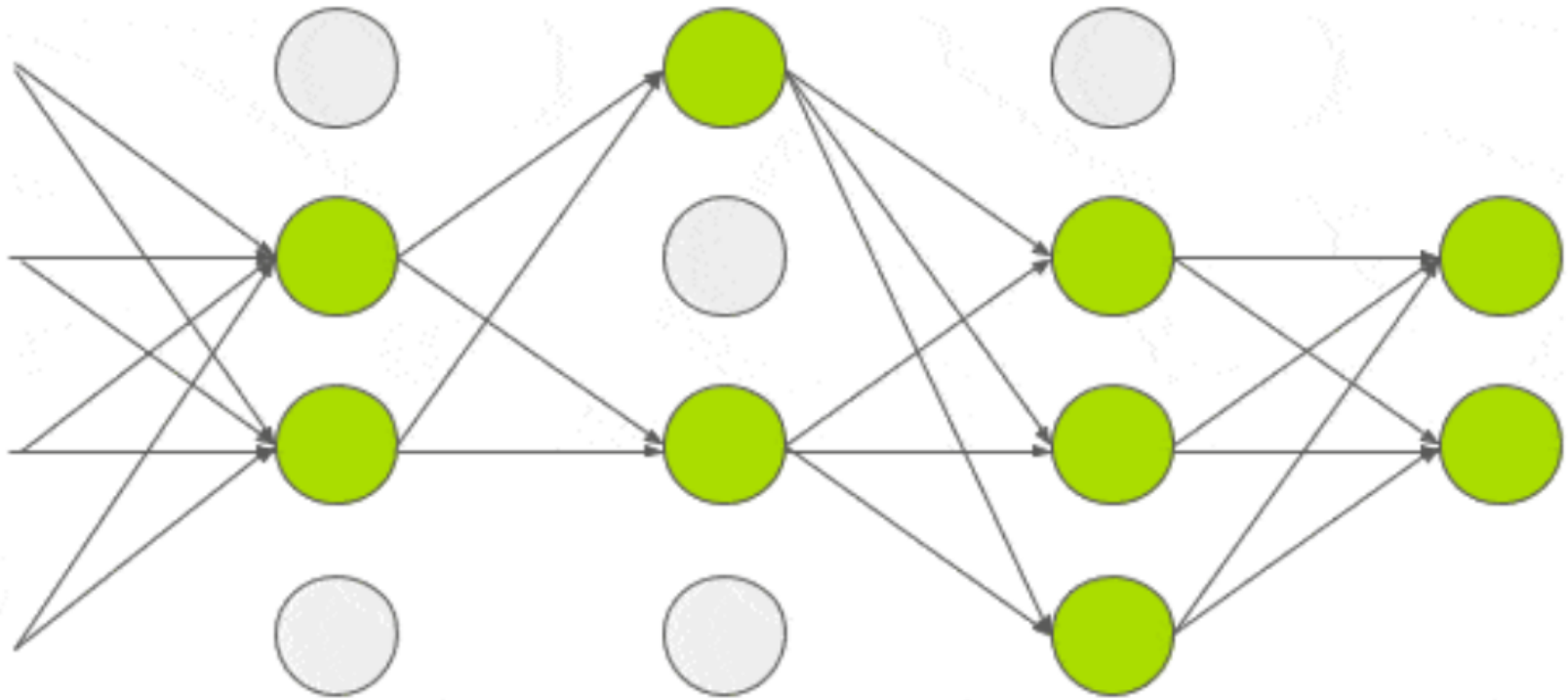


Figure: Dropout mechanism

CNN - Alex Net Architecture

Dropout

- Typically 50% of neurons are dropped in each pass.
- This leads to an increase in the number of training epochs (roughly twice).
- During testing, the whole network is used. Activations are scaled by a factor of 0.5 to account for the dropped neurons during training.
- Since all neurons are not involved in the forward pass, this makes the neurons somewhat robust to variations in the input and leads to less overfitting to training data and better generalization. (empirical observation.)

CNN - Alex Net Architecture

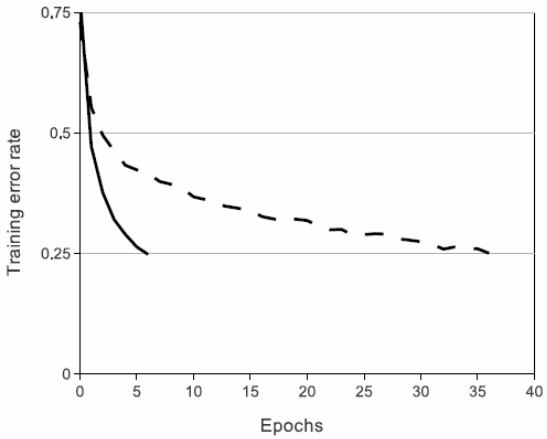


Figure 1: A four-layer convolutional neural network with ReLUs **(solid line)** reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons **(dashed line)**. The learning rates for each network were chosen independently to make training as fast as possible. No regularization of any kind was employed. The magnitude of the effect demonstrated here varies with network architecture, but networks with ReLUs consistently learn several times faster than equivalents with saturating neurons.

Figure: Impact of ReLU activation function

CNN - Alex Net Kernels After Training

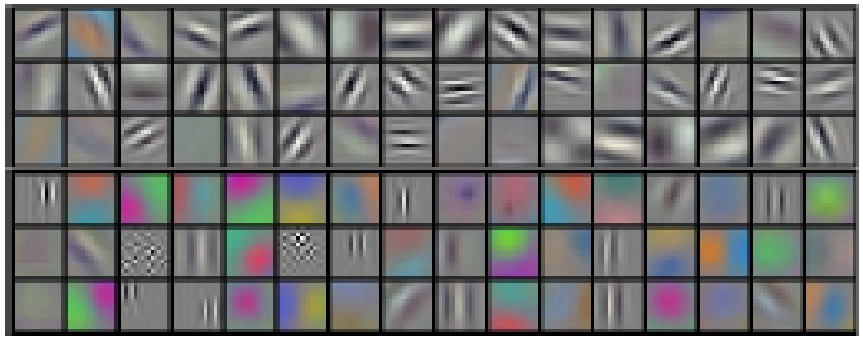


Figure 3: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

CNN - Alex Net Results

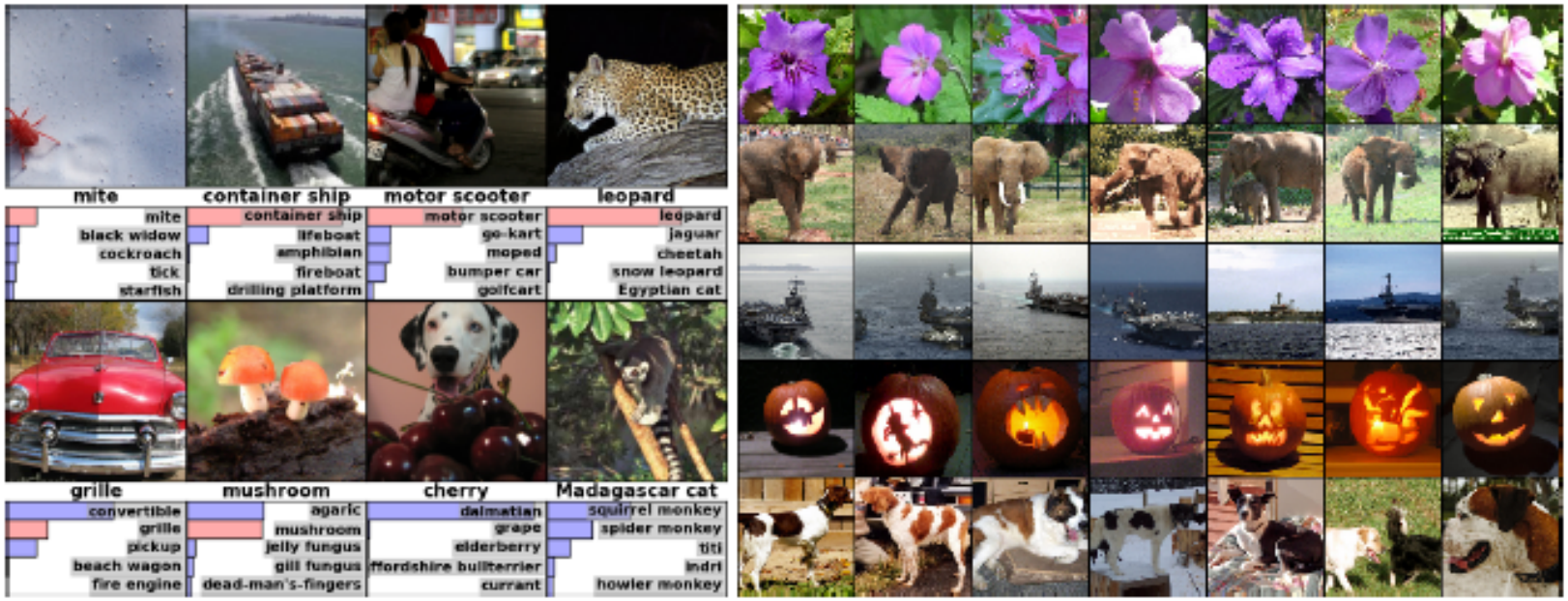


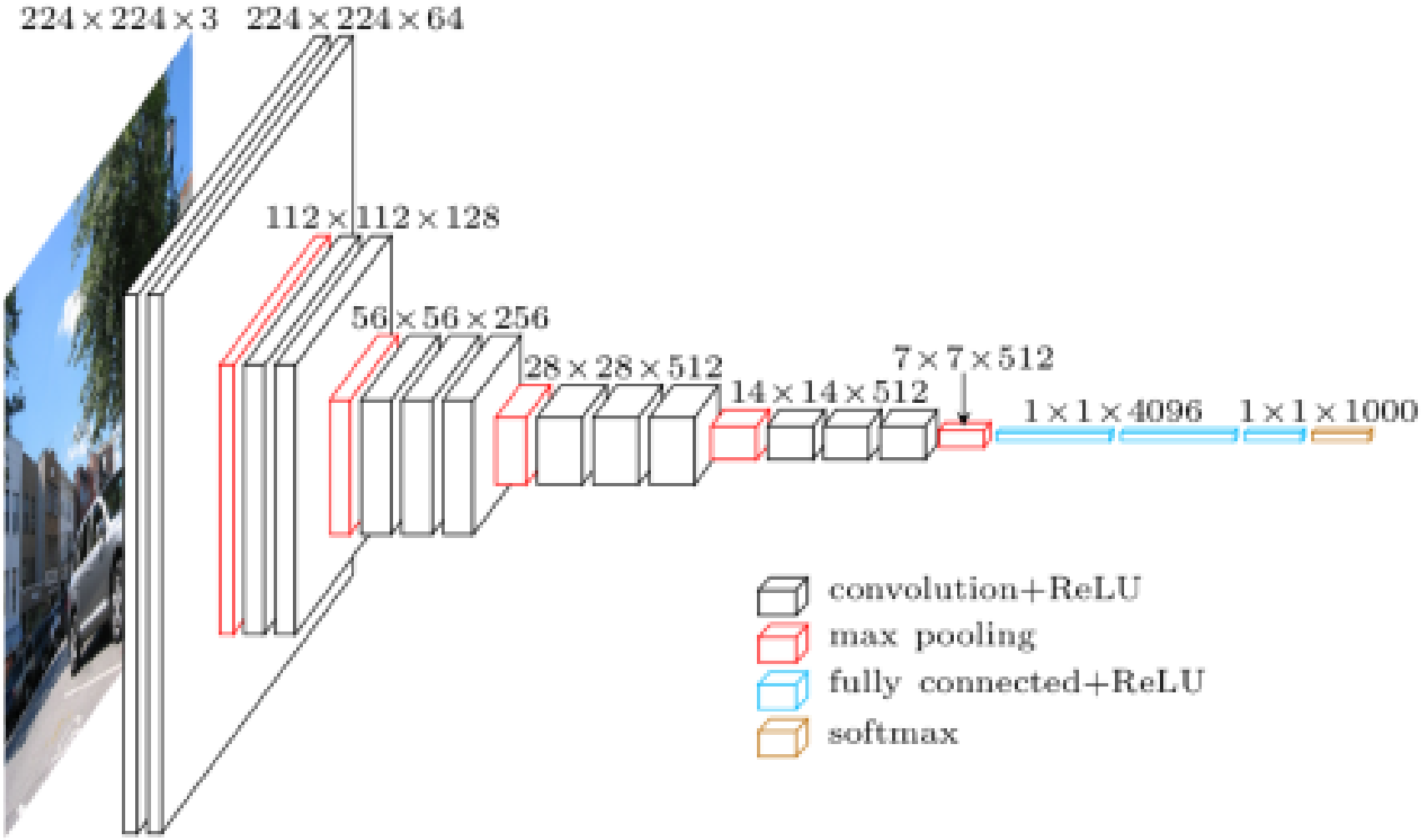
Figure 4: **(Left)** Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5). **(Right)** Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.

CNN - VGGNet

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* **& Andrew Zisserman+**
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

CNN - A VGG Net Architecture



CNN - VGG Net

Peculiarities:

- Minimal pre-processing (*e.g.* Subtraction of mean value from RGB pixel values)
- Small receptive field of size 3×3 with stride 1 (as opposed to 7×7 with stride 2 and 11×11 with stride 4 used in other previous approaches)
- Spatial resolution is preserved after convolution with 3×3 filter by using padding of appropriate size. (what padding size is suitable?)
- Max-pooling of size 2×2 with stride 2 was used after some convolution layers.
- Small receptive field leads to reduction in number of weights to be learned (check this claim!)

CNN - VGG Net

Peculiarities:

- Multiple networks with different depths (11 to 19 layers) were experimented with.
  - The depth was increased by increasing the number of convolution layers.
  - The three FC layers were same across the networks.
- The number of convolution filters used in the layers ranged from 64 to 512. This increase was done after each max-pooling operation.
- Using a stack of 2 convolution layers successively each with 3×3 filters effectively achieves a receptive field size of that of using 5×5 filter once. (check this claim!)

CNN - VGG Net

Peculiarities:

- Multinomial logistic regression objective
- Mini-batch SGD with momentum
- Weight decay employed for regularization
- Dropout used for some FC layers.
- Initial learning rate set to 10^{-2} and then decreased when validation error stalls
- Pre-initialization helped

CNN - VGG Net

Peculiarities in Training:

- Multiple scales considered ($S = 256$, $S = 384$, $S \in [256, 512]$)
- Training for larger S value performed by initializing with weights obtained from smaller S values
- Training was done on multiple GPUs where a single batch of images was further divided into GPU batches and passed to the GPUs for parallel processing.
- Titan Black GPUs of NVIDIA were used. Training took around 2 to 3 weeks.

CNN - VGG Net

Peculiarities in Testing:

- Multiple scales considered in testing too
- Scale values not exactly the same as training scales
- FC layers converted to convolution maps
- Dense vs multi-crop evaluation
  - Multi-crop evaluation was similar to that used in AlexNet.
  - In dense evaluation, arbitrary sized inputs were considered without any cropping, and conversion to an appropriate dimension was automatically done by using appropriate convolutions so that the final classification can be obtained on any arbitrary sized input.

CNN - VGG Net

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

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					

CNN - VGG Net

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

CNN - VGG Net

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

CNN - VGG Net

Table 4: **ConvNet performance at multiple test scales.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

CNN - VGG Net

Table 5: **ConvNet evaluation techniques comparison.** In all experiments the training scale S was sampled from $[256; 512]$, and three test scales Q were considered: $\{256, 384, 512\}$.

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
D	dense	24.8	7.5
	multi-crop	24.6	7.5
	multi-crop & dense	**24.4**	**7.2**
E	dense	24.8	7.5
	multi-crop	24.6	7.4
	multi-crop & dense	**24.4**	7.1

CNN - VGG Net

Table 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

CNN - VGG Net

Table 10: **Comparison with the state of the art in ILSVRC localisation**. Our method is denoted as "VGG".

Method	top-5 val. error (%)	top-5 test error (%)
VGG	**26.9**	**25.3**
GoogLeNet (Szegedy et al., 2014)	-	26.7
OverFeat (Sermanet et al., 2014)	30.0	29.9
Krizhevsky et al. (Krizhevsky et al., 2012)	-	34.2

CNN - VGG Net

Table 11: **Comparison with the state of the art in image classification on VOC-2007, VOC-2012, Caltech-101, and Caltech-256.** Our models are denoted as "VGG". Results marked with * were achieved using ConvNets pre-trained on the *extended* ILSVRC dataset (2000 classes).

Method	VOC-2007 (mean AP)	VOC-2012 (mean AP)	Caltech-101 (mean class recall)	Caltech-256 (mean class recall)
Zeiler & Fergus (Zeiler & Fergus, 2013)	-	79.0	86.5 ± 0.5	74.2 ± 0.3
Chatfield et al. (Chatfield et al., 2014)	82.4	83.2	88.4 ± 0.6	77.6 ± 0.1
He et al. (He et al., 2014)	82.4	-	$\mathbf{93.4 \pm 0.5}$	-
Wei et al. (Wei et al., 2014)	81.5 (85.2*)	81.7 (**90.3***)	-	-
VGG Net-D (16 layers)	89.3	89.0	91.8 ± 1.0	85.0 ± 0.2
VGG Net-E (19 layers)	89.3	89.0	92.3 ± 0.5	85.1 ± 0.3
VGG Net-D & Net-E	**89.7**	**89.3**	92.7 ± 0.5	$\mathbf{86.2 \pm 0.3}$

CNN - VGG Net

Table 12: **Comparison with the state of the art in single-image action classification on VOC-2012.** Our models are denoted as "VGG". Results marked with * were achieved using ConvNets pre-trained on the *extended* ILSVRC dataset (1512 classes).

Method	VOC-2012 (mean AP)
(Oquab et al., 2014)	70.2*
(Gkioxari et al., 2014)	73.6
(Hoai, 2014)	76.3
VGG Net-D & Net-E, image-only	**79.2**
VGG Net-D & Net-E, image and bounding box	**84.0**