Convolutional Neural Networks (CNN)

Algorithm and Some Applications in Computer Vision

Luo Hengliang

Institute of Automation

June 10, 2014

Table of Contents

Introduction to CNN[1]

PROC. OF THE IEEE, NOVEMBER 1998 1

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

Abstract—

Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful

I. Introduction

Over the last several years, machine learning techniques

[1] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

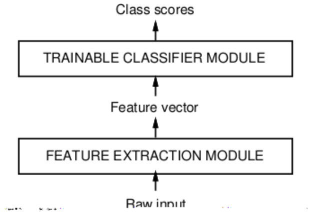
Traditional pattern recognition method for image

hand-crafted feature + general classifier

- The first module, called the feature extractor, transforms the input patterns so that they can be represented by low-dimensional vectors.
- The classifier, on the other hand, is often general purpose and trainable.

main problem

- The recognition accuracy is largely determined by the ability of the designer to come up with an appropriate set of features.

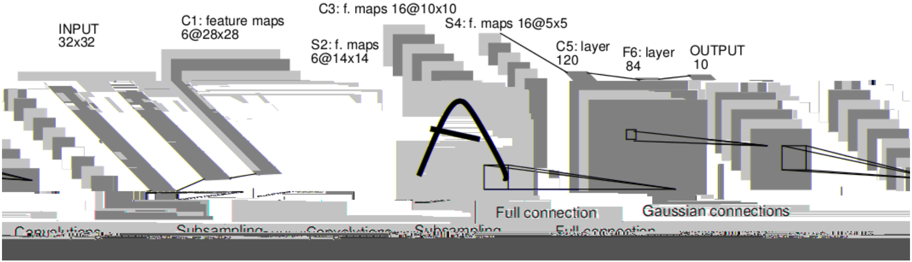


How to learn the feature extractor itself?

If we input the raw pixels to the multilayer networks and train it, then:

- Typical images are large, and the networks contain several tens of thousands of weights. Such a large number of parameters increases the capacity of the system and therefore requires a larger training set.
- The unstructured nets for image applications have no built-in invariance with respect to translations or local distortions of the inputs.
- The topology of the input is entirely ignored. The input variables can be presented in any (fixed) order without affecting the outcome of the training.

How to learn the feature extractor itself?(cont'd)



	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

Architecture of Convolutional Neural Networks

Rebirth of CNN[2]

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

[2] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks[C]//NIPS. 2012, 1(2): 4.

Dataset

ImageNet

- 15 million labeled high-resolution images belonging to roughly 22,000 categories.

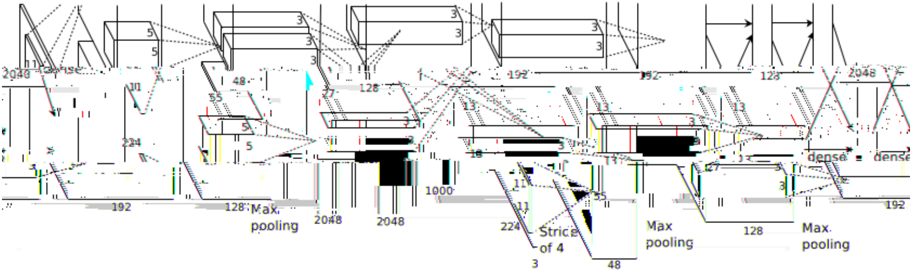
ILSVRC(ImageNet Large-Scale Visual Recognition Challenge)

- a subset of ImageNet with roughly 1000 images in each of 1000 categories.
- there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images.
- it is customary to report two error rates:top-1 and top-5, where the top-5 error rate is the fraction of test images for which the correct label is not among the five labels considered most probable by the model.

Data pre-process

- rescale the image such that the shorter side was of length 256, and then cropped out the central 256x256 patch from the resulting image.
- subtract the mean activity over the training set from each pixel.

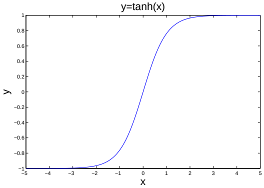
The Architecture



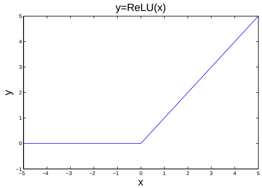
The Architecture

The Architecture(cont'd)

- ReLU(Rectified Linear Units) Nonlinearity: deep convolutional neural networks with ReLUs train several times faster than their equivalents with tanh units.
- Training on Multiple GPUs.
- Local Response Normalization:local normalization after ReLU Nonlinearity aids generalization.
- Overlapping Pooling: during training that models with overlapping pooling find it slightly more difficult to overfit.



(c) tanh



(d) ReLU

Reducing Overfitting

Data Augmentation

- generating image translations and horizontal reflections. We do this by extracting random 224x224 patches (and their horizontal reflections) from the 256x256 images and training our network on these extracted patches . This increases the size of our training set by a factor of 2048,
- The second form of data augmentation consists of altering the intensities of the RGB channels in training images. Specifically, we perform PCA on the set of RGB pixel values throughout the ImageNet training set.

Dropout

- We use dropout in the first two fully-connected layer. Without dropout, our network exhibits substantial overfitting.

Result on ILSVRC-2010

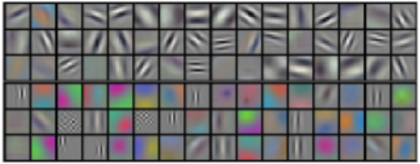


Figure 3: 96 convolutional kernels of size $11\times11\times3$ learned by the first convolutional layer on the $224\times224\times3$ input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

Result on ILSVRC-2010(cont'd)

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

CNN Features off-the-shelf[3]

CNN Features off-the-shelf: an Astounding Baseline for Recognition

Ali Sharif Razavian Hossein Azizpour Josephine Sullivan Stefan Carlsson

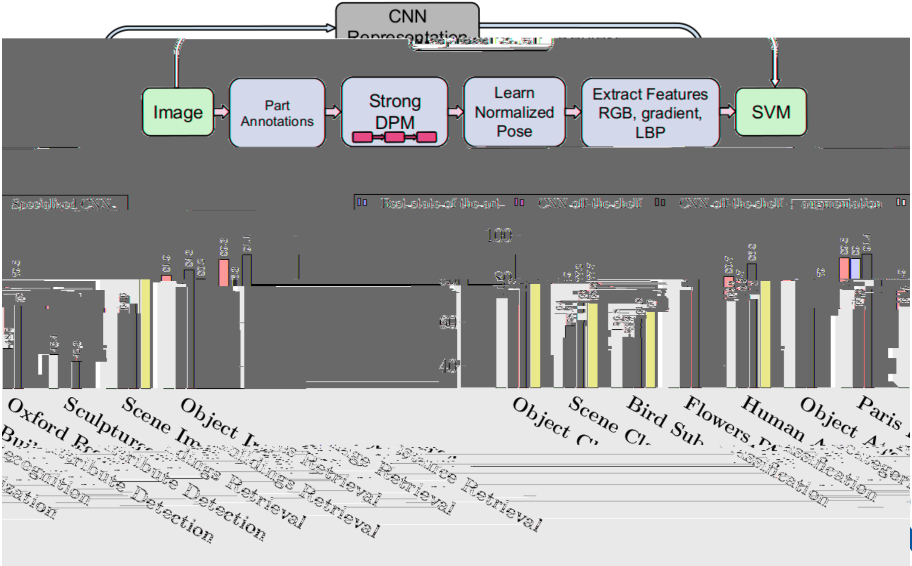
CVAP, KTH (Royal Institute of Technology)

Stockholm, Sweden

{razavian,azizpour,sullivan,stefanc}@csc.kth.se

[3] Razavian A S, Azizpour H, Sullivan J, et al. CNN Features off-the-shelf: an Astounding Baseline for Recognition[J]. arXiv preprint arXiv:1403.6382, 2014.

CNN result on many datasets



Visual Classification Method

For all the experiments we resize the whole image (or cropped sub-window) to 221x221 and input the image to `OverFeat`. This gives a vector of 4096 dimensions. We have two settings:

- The feature vector is further L2 normalized to unit length for all the experiments. We use the 4096 dimensional feature vector in combination with a Support Vector Machine (SVM) to solve different classification tasks (CNN-SVM).
- We further augment the training set by adding cropped and rotated samples and doing component-wise power transform and report separate results (CNNaug+SVM)

Pascal VOC 2007 Image Classification Results

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
GHM[8]	76.7	74.7	53.8	72.1	40.4	71.7	83.6	66.5	52.5	57.5	62.8	51.1	81.4	71.5	86.5	36.4	55.3	60.6	80.6	57.8	64.7
AGS[11]	82.2	83.0	58.4	76.1	**56.4**	**77.5**	**88.8**	69.1	**62.2**	61.8	64.2	51.3	**85.4**	**80.2**	91.1	48.1	61.7	**67.7**	86.3	70.9	71.1
NUS[39]	82.5	79.6	64.8	73.4	54.2	75.0	77.5	79.2	46.2	62.7	41.4	74.6	85.0	76.8	91.1	53.9	61.0	67.5	83.6	70.6	70.5
CNN-SVM	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.2	71.8	73.9
CNNaug-SVM	**90.1**	**84.4**	**86.5**	**84.1**	48.4	73.4	86.7	**85.4**	61.3	**67.6**	**69.6**	**84.0**	**85.4**	80.0	**92.0**	**56.9**	**76.7**	67.3	**89.1**	**74.9**	**77.2**

Table 1: **Pascal VOC 2007 Image Classification Results** compared to other methods which also use training data outside VOC. The CNN representation is not tuned for the Pascal VOC dataset. However, GHM [8] learns from VOC a joint representation of bag-of-visual-words and contextual information. AGS [11] learns a second layer of representation by clustering the VOC data into subcategories. NUS [39] trains a codebook for the SIFT, HOG and LBP descriptors from the VOC dataset. Oquab *et al.* [29] fixes all the layers trained on ImageNet then it adds and optimizes two fully connected layers on the VOC dataset and achieves better results (**77.7**) indicating the potential to boost the performance by further adaptation of the representation to the target task/dataset.

Table of Contents

GTSRB[4] Dataset and reletated works[5]

GTSRB

- 43 classes, 39,209 training images, 12,630 test images, images size vary from 15x15 to 250x250.

Result

CCR (%)	Team	Method	
ittee of CNNs	99.46	IDSIA	Comm
[illegible]	99.22	INI-RTCV	Huma
INI-RTCV	Human (average)	98.84	[illegible]
Sermanet	Multi-scale CNN	98.31	
CAOR	Random forests	96.14	
INI-RTCV	LDA (HOG 2)	95.68	
[illegible]	[illegible]	[illegible]	
92.34	INI-RTCV	LDA (HOG 3)	

Result overview for the final stage of the GTSRB.

4 http://benchmark.ini.rub.de/?section=gtsrb&subsection=news

5 Stallkamp J, Schlipsing M, Salmen J, et al. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition[J]. Neural networks, 2012, 32: 323-332.

Programming tool:Torch7[6]

About Torch7

Torch7 is a scientific computing framework with wide support for machine learning algorithms. It is easy to use and provides a very efficient implementation, thanks to an easy and fast scripting language, LuaJIT, and an underlying C implementation[a].

[a] torch.ch

Why Choose Torch7?

- It was recommended by Yann Lecun
- It is very fast
- See more from the website[a] below.

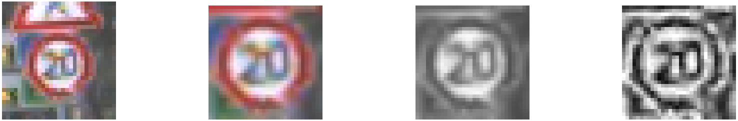
[a] http://www.kdnuggets.com/2014/02/exclusive-yann-lecun-deep-learning-facebook-ai-lab.html

[6] Collobert R, Farabet C, Kavukcuoglu K. Torch7: A matlab-like environment for machine learning[C]//BigLearn, NIPS Workshop. 2011 (EPFL-CONF-192376).

Image pre-processing

Crop the image → **Resize to 32x32** → **Color to gray** → **CLAHE**

Image preprocessing method from the paper[7]

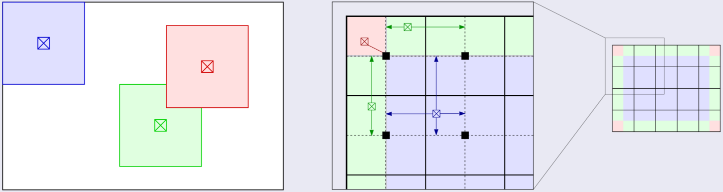


(g) Origin **(h)** Crop **(i)** Gray **(j)** CLAHE

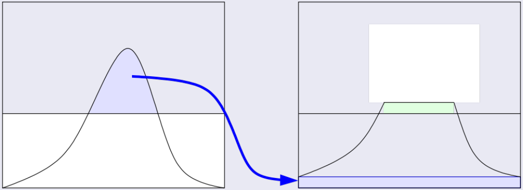
[7] Ciresan D, Meier U, Masci J, et al. A committee of neural networks for traffic sign classification[C]//Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 2011: 1918-1921.

Contrast Limited Adaptive Histogram Equalization[8]

Adaptive histogram equalization

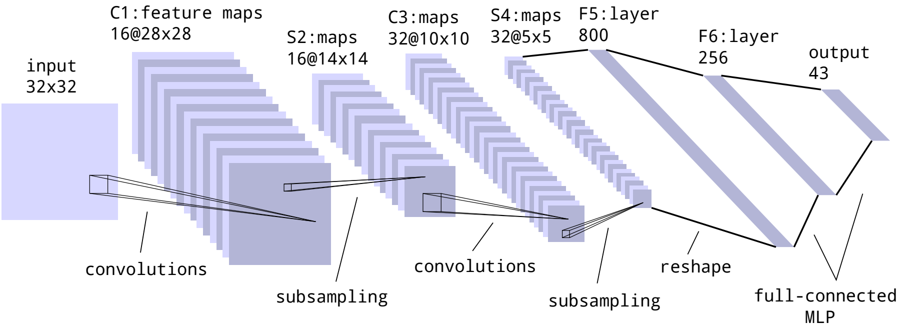


Contrast limited adaptive histogram equalization



[8] http://en.wikipedia.org/wiki/Adaptive_histogram_equalization

CNN Structure

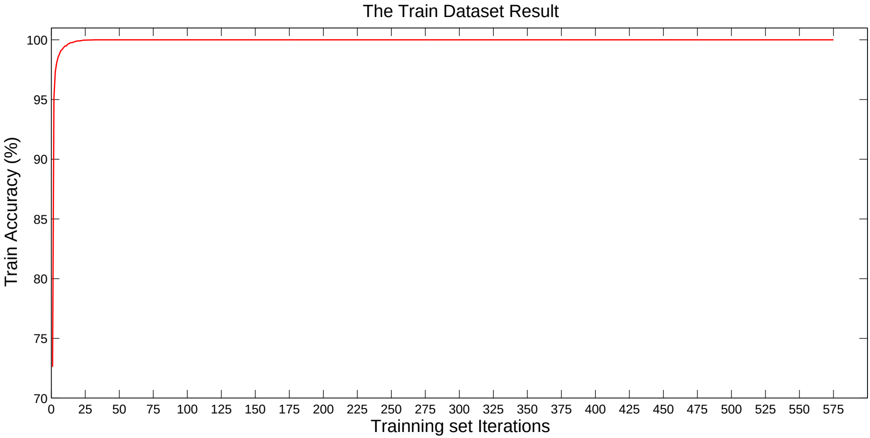


CNN Structure

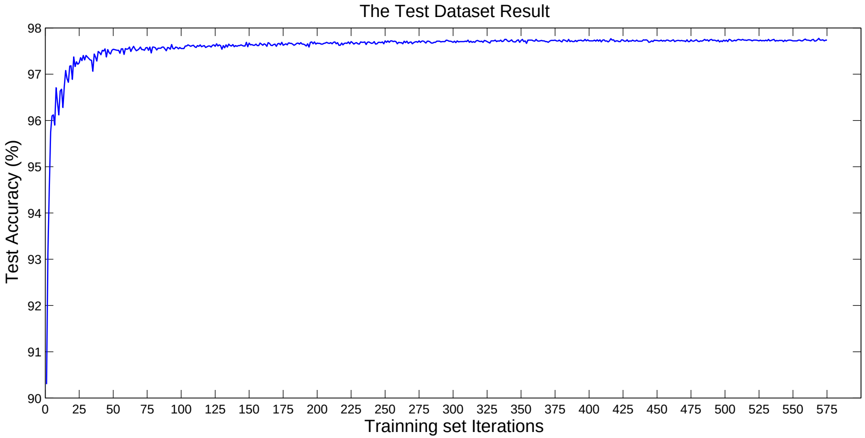
Define Model in Torch7

```
model:add(nn.SpatialConvolutionMM(1, 16, 5, 5))
model:add(nn.Tanh())
model:add(nn.SpatialLPPooling(16, 2, 2, 2, 2, 2))

model:add(nn.SpatialConvolutionMM(16, 32, 5, 5))
model:add(nn.Tanh())
model:add(nn.SpatialLPPooling(32, 2, 2, 2, 2, 2))

model:add(nn.Reshape(32*5*5))
model:add(nn.Linear(32*5*5, 256))
model:add(nn.Tanh())
model:add(nn.Linear(256, 43))
```

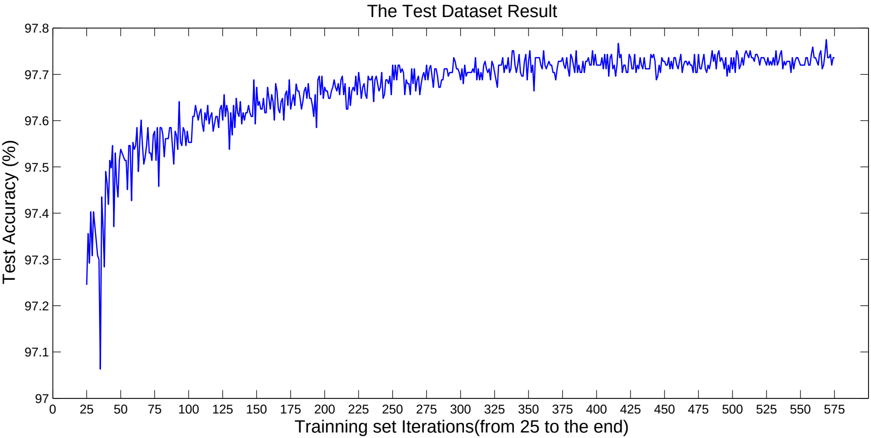
Train Result



Test Result

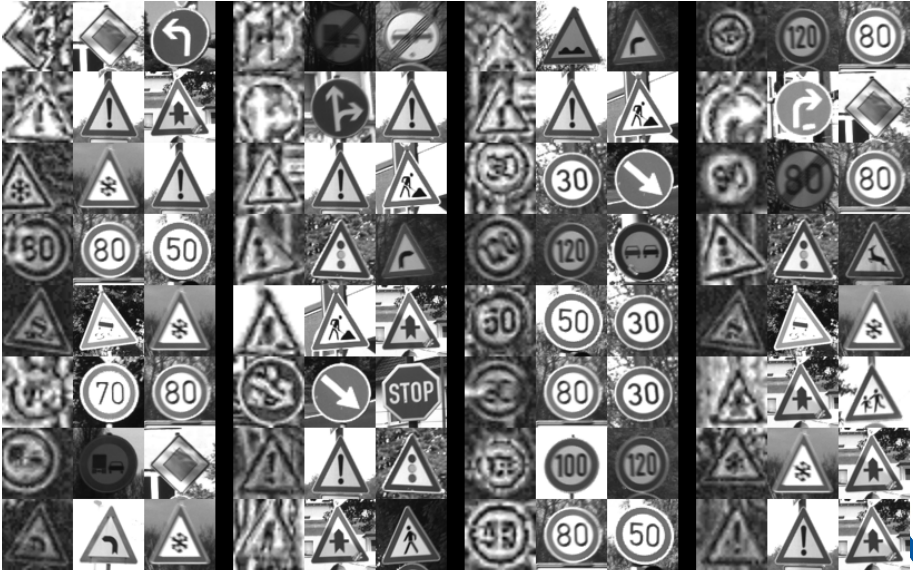

The Test Dataset Result
Test Accuracy (%)
98
97
96
95
94
93
92
91
90
0
25
50
75
100
125
150
175
200
225
250
275
300
325
350
375
400
425
450
475
500
525
550
575
Trainning set Iterations

Result(cont'd)



The best test accuracy is 97.75%.

Some misclassified examples in test set



Retrain:data augmentation

Data augmentation

- scaling:$[0.8, 1.2]$
- translation:`random`
- rotation:$[-15^\circ, 15^\circ]$

Recognize the detection results

Class	Total	Error Count	
		Without Aug.	With Aug.
Danger	62	12	0
Mandatory	56	23	1
Prohibitory	168	48	1

Some “hard” detection results

Integrated with the Application

Forward-cnn

- Torch7 doesn't support Windows now, but we need to create a gui demo application in Windows.
- So I write the forward cnn in C++ at `https://github.com/beenfrog/cnn-forward`

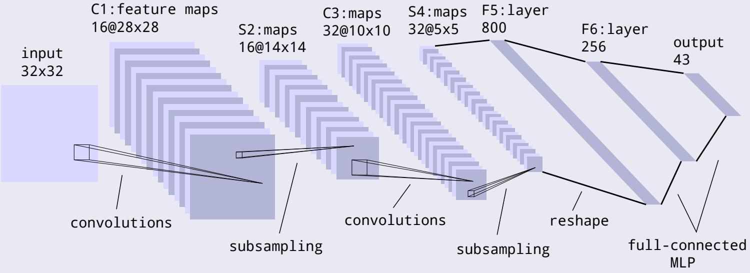


Table of Contents

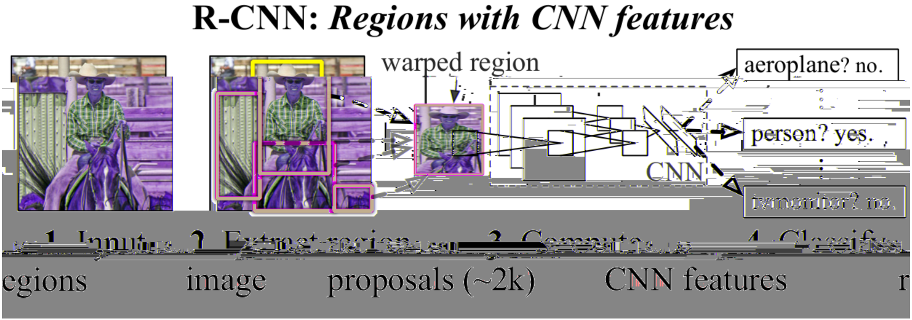
R-CNN: Regions with CNN features[9]

Rich feature hierarchies for accurate object detection and semantic segmentation

Ross Girshick[1] Jeff Donahue[1,2] Trevor Darrell[1,2] Jitendra Malik[1]
[1]UC Berkeley and [2]ICSI
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

[9] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[J]. arXiv preprint arXiv:1311.2524, 2013.

Object detection system overview

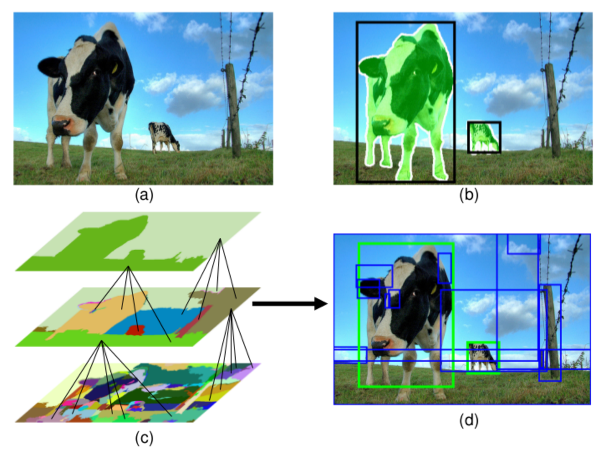


overview

- takes an input image
- extracts around 2000 bottom-up region proposals
- computes features for each proposal using a large convolutional neural network (CNN)
- classifies each region using class-specific linear SVMs

Region proposals:Selective Search[10]

Selective Search

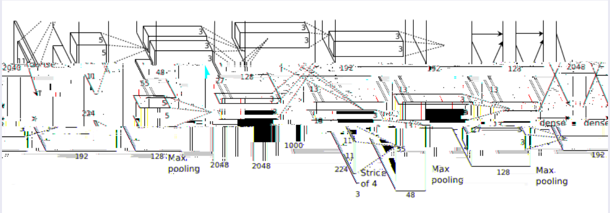


[10] van de Sande K E A, Uijlings J R R, Gevers T, et al. Segmentation as selective search for object recognition[C]//Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011: 1879-1886.

Feature extraction

Feature extraction

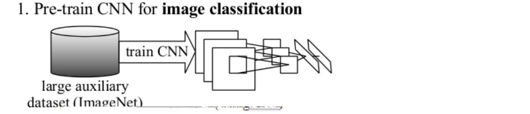
- We extract a 4096-dimensional feature vector from each region proposal using our own implementation of the CNN of [Hinton2012].



- In order to compute features for a region proposal, we must first convert the image data in that region into a fixed 224x224 pixel size.

Training[11]

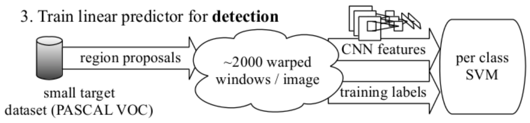
CNN pre-training



CNN fine-tuning

[redacted]

Object category classifiers



[11] http://www.image-net.org/challenges/LSVRC/2013/slides/r-cnn-ilsvrc2013-workshop.pdf

R-CNN: Result

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM HOG [19]	45.6	49.0	11.0	11.6	27.2	50.5	43.1	23.6	17.2	23.2	10.7	20.5	42.5	44.5	41.3	8.7	29.0	18.7	40.0	34.5	29.6
SegDPM [18]	56.4	48.0	24.3	21.8	**31.3**	**51.3**	47.3	48.2	16.1	29.4	19.0	37.5	44.1	51.5	44.4	12.6	32.1	28.8	**48.9**	39.1	36.6
UVA [36]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	**31.8**	47.0	44.8	35.1
ours (R-CNN FT fc$_7$)	**65.4**	**56.5**	**45.1**	**28.5**	24.0	50.1	**49.1**	**58.3**	**20.6**	**38.5**	**31.1**	**57.5**	**50.7**	**60.3**	**44.7**	**21.6**	**48.5**	24.9	48.0	**46.5**	**43.5**

Table 1: Detection average precision (%) on VOC 2010 test. Our method competes in the *comp4* track due to our use of outside data from ImageNet. Our system is most directly comparable to UVA (row 3) since both methods use the same selective search region proposal mechanism, but differ in features. We compare to methods before rescoring with inter-detector context and/or image classification.

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN pool$_5$	49.3	58.0	29.7	22.2	20.6	47.7	56.8	43.6	16.0	39.7	37.7	39.6	49.6	55.6	37.5	20.6	40.5	37.4	47.8	51.3	40.1
R-CNN fc$_6$	56.1	58.8	34.4	29.6	22.6	50.4	58.0	52.5	18.3	40.1	41.3	46.8	49.5	53.5	39.7	23.0	46.4	36.4	50.8	59.0	43.4
R-CNN fc$_7$	53.1	58.9	35.4	29.6	22.3	50.0	57.7	52.4	19.1	43.5	40.8	43.6	47.6	54.0	39.1	23.0	42.3	33.6	51.4	55.2	42.6
R-CNN FT pool$_5$	55.6	57.5	31.5	23.1	23.2	46.3	59.0	49.2	16.5	43.1	37.8	39.7	51.5	55.4	40.4	23.9	46.3	37.9	49.7	54.1	42.1
R-CNN FT fc$_6$	61.8	62.0	38.8	35.7	29.4	52.5	**61.9**	53.9	22.6	49.7	40.5	**48.8**	49.9	**57.3**	44.5	**28.5**	50.4	**40.2**	54.3	61.2	47.2
R-CNN FT fc$_7$	**60.3**	**62.5**	**41.4**	**37.9**	29.0	**52.6**	61.6	**56.3**	**24.9**	**52.3**	**41.9**	48.1	54.3	57.0	**45.0**	26.9	**51.8**	38.1	**56.6**	**62.2**	**48.0**
DPM HOG [19]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	**58.1**	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [29]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [32]	32.2	58.3	11.5	16.3	**30.6**	49.9	54.8	23.5	21.5	27.7	34.0	13.7	**58.1**	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3

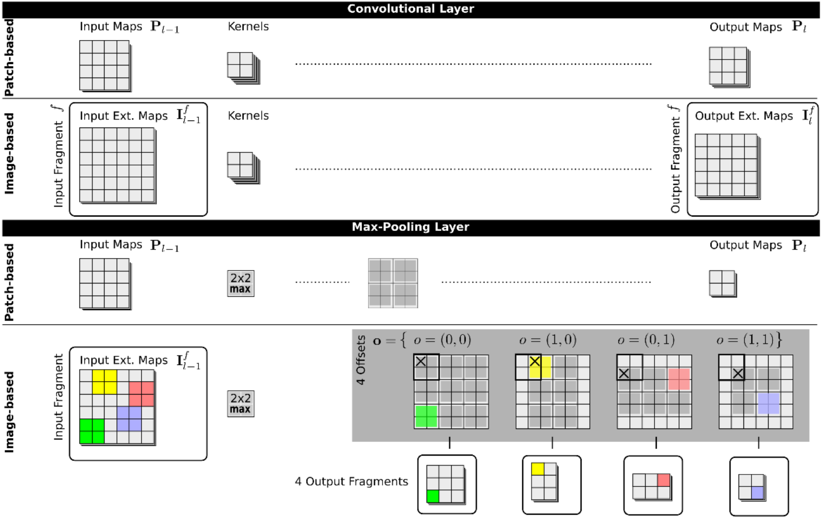
Table 2: Detection average precision (%) on VOC 2007 test. Rows 1-3 show results for our CNN pre-trained on ILSVRC 2012. Rows 4-6 show results for our CNN pre-trained on ILSVRC 2012 and then fine-tuned ("FT") on VOC 2007 trainval. Rows 7-9 present DPM methods as a strong baseline comparison. The first uses only HOG, while the next two use feature learning to augment or replace it.

Any other new approaches to detection with CNN?[12]

- **Question from me**:How to use the CNN effectively in object detection? The traditional sliding window method may be too slow. There are some works focused on generating region proposals first, such as http://arxiv.org/abs/1311.2524, any other new approaches? Thanks!
- **Answer by ylecun**:ConvNets are not too slow for detection. Look at our paper on OverFeat [Sermanet et al. ICLR 2014], on pedestrian detection [Sermanet et al. CVPR 2013], and on face detection [Osadchy et al. JMLR 2007] and [Vaillant et al. 1994]. The key insight is that you can apply a ConvNet.....convolutionally over a large image, without having to recompute the entire network at every location (because much of the computation would be redundant). We have known this since the early 90's.
- **Answer by osdf**:A recent paper that takes the idea of avoiding recomputations to CNNs with max-pooling operations: Fast image scanning with deep max-pooling convolutional neural networks.

[12] http://www.reddit.com/r/MachineLearning/comments/25lnbt/ama_yann_lecun

Fast Image Scanning [13]



13 Giusti A, Cirean D C, Masci J, et al. Fast image scanning with deep max-pooling convolutional neural networks[J]. arXiv preprint arXiv:1302.1700, 2013.

Speed up the forward computation of CNN[14]

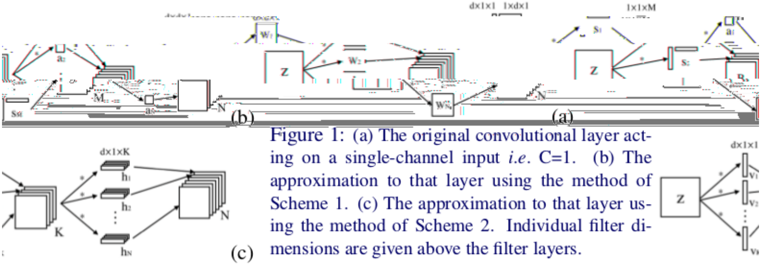


Figure 1: (a) The original convolutional layer acting on a single-channel input *i.e.* C=1. (b) The approximation to that layer using the method of Scheme 1. (c) The approximation to that layer using the method of Scheme 2. Individual filter dimensions are given above the filter layers.

Both schemes follow the same intuition: that CNN filter banks can be approximated using a low rank basis of filters that are separable in the spatial domain

14 Jaderberg M, Vedaldi A, Zisserman A. Speeding up Convolutional Neural Networks with Low Rank Expansions[J]. arXiv preprint arXiv:1405.3866, 2014.

Table of Contents

Descriptor Matching[15]

Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT

Philipp Fischer[*†]
Department of Computer Science
University of Freiburg
fischer@cs.uni-freiburg.de

Alexey Dosovitskiy[†]
Department of Computer Science
University of Freiburg
dosovits@cs.uni-freiburg.de

Thomas Brox
Department of Computer Science
University of Freiburg
brox@cs.uni-freiburg.de

[15] Fischer P, Dosovitskiy A, Brox T. Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT[J]. arXiv preprint arXiv:1405.5769, 2014.

Feature Learning with Convolutional Neural Nets

Supervised Training

- We used a pre-trained model form [Hinton2012].

Unsupervised Training

- We used random images from Flickr because we expect those to be better representatives of the distribution of natural images.
- Next N = 16000 "seed" patches of size 64x64 pixels were extracted randomly from different images at various locations and scales. Each of these "seed" patches was declared to represent a surrogate class of its own.
- These classes were augmented by applying K = 150 random transformations to each of the "seed" patches. Each transformation was a composition of random elementary transformations. These included translation, scale variation, rotation, color variation, contrast variation, and also blur, which is often relevant for matching problems.
- As a result we obtained a surrogate labeled dataset with N classes containing K samples each. We used these data to train a convolutional neural network.

The New Test DataSet

Figure 1: Some base images used for generating the dataset.

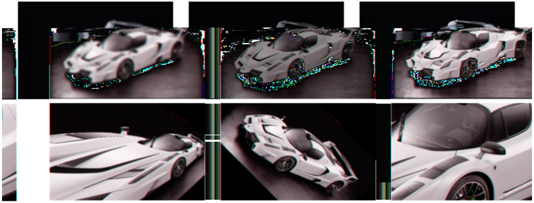
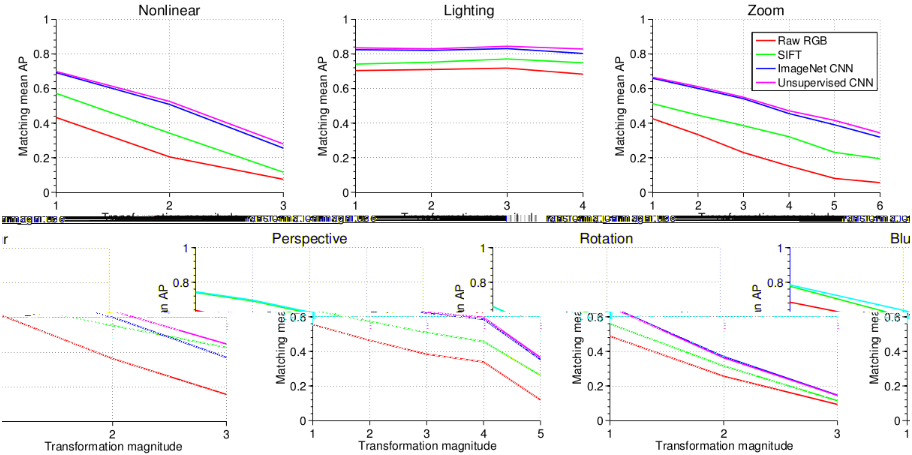


Figure 2: Most extreme versions of the transformations applied to the base images. From left to right: blur, lighting change, nonlinear deformation, perspective change, rotation, zoom.

Result



formations. Except for the
FT. The unsupervised net is

Figure 4: Mean average precision on the larger dataset for various trans
blur transformation, both neural nets perform consistently better than SIF
also better on blur.

Feature computation time

Method	SIFT	ImageNet CNN	Unsup. CNN
Time	$2.95\text{ms} \pm 0.04$	$11.1\text{ms} \pm 0.28$	$37.6\text{ms} \pm 0.6$

Table 1: Feature computation times for a patch of 91 by 91 pixels on a single CPU. On a GPU, the convolutional networks both need around 5.5ms per image.

Table of Contents

Deep ConvNets: astounding baseline for vision[16]

[Sermanet et al 2014]: OverFeat (fine-tuned features for each task)
(tasks are ordered by increasing difficulty)

competitive	13.6 % error	• image classification	ImageNet LSVRC 2013
state of the art	98.9%		Dogs vs Cats Kaggle challenge 2014
state of the art	29.9% error	• object localization	ImageNet LSVRC 2013
state of the art	24.3% mAP	• object detection	ImageNet LSVRC 2013

[Razavian et al, 2014]: public OverFeat library (no retraining) + SVM
(simplest approach possible on purpose, no attempt at more complex classifiers)
(tasks are ordered by "distance" from classification task on which OverFeat was trained)

73.9% mAP	• image classification	Pascal VOC 2007	competitive
58.4% mAP	• scene recognition	MIT-67	competitive
53.3% mAP	• fine grained recognition	Caltech-UCSD Birds 200-2011	competitive
74.70% mAP		Oxford 102 Flowers	competitive
89.0% mAUC	• attribute detection	UIUC 64 object attributes	state of the art
70.78% mAP		H3D Human Attributes	state of the art
0.52	• image retrieval	Oxford 5k buildings	?
0.676	(search by image similarity)	Paris 6k buildings	?
0.269		Sculp6k	?
0.646		Holidays	competitive
3.05		UKBench	relatively poor

---

16 http://cs.nyu.edu/~sermanet/papers/Deep_ConvNets_for_Vision-Results.pdf

Deep ConvNets: astounding baseline for vision (cont'd)

	Dataset	Performance	Score
[Zeiler et al 2013] • image classification	ImageNet LSVRC 2013 Caltech-101 (15, 30 samples per class) Caltech-256 (15, 60 samples per class) Pascal VOC 2012	**state of the art** competitive **state of the art** competitive	11.2% error 83.8%, 86.5% 65.7%, 74.2% 79% mAP
[Donahue et al, 2014]: DeCAF+SVM • image classification	Caltech-101 (30 classes)	**state of the art**	86.91%

=> Webcam	**state of the art**	82.1%, 94.8%	• domain adaptation	Amazon -> Webcam, DSLR
11	**state of the art**	65.0%	• fine grained recognition	Caltech-UCSD Birds 200-20
	competitive	40.9%	• scene recognition	SUN-397
			[Girshick et al, 2013]	
OC 2010 (comp4)	**state of the art**	43.5% mAP		Pascal V
(comp6)	**state of the art**	47.9% mAP	• image segmentation	Pascal VOC 2011
			[Oquab et al, 2013]	
	state of the art	77.7% mAP	• image classification	Pascal VOC 2007
	state of the art	82.8% mAP		Pascal VOC 2012
(action classification)	**state of the art**	70.2% mAP		Pascal VOC 2012

Deep ConvNets: astounding baseline for vision (cont'd)

	Dataset	Performance	Score
[Khan et al 2014] • shadow detection	UCF CMU UIUC	**state of the art** **state of the art** **state of the art**	90.56% 88.79% 93.16%
[Sander Dieleman, 2014] • image attributes	Kaggle Galaxy Zoo challenge	**state of the art**	0.07492

Future works

- Use the feature learned from CNN in other vision tasks.
- Unsupervised Learning.
- ...

Thank you!