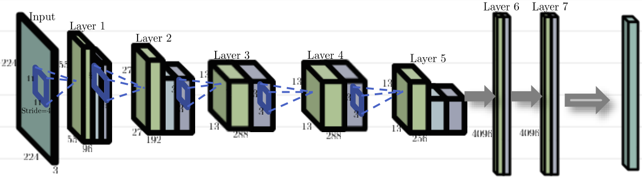
Lecture 5: Convolutional Networks in Computer Vision

Haven't it all been about computer vision?

(in fact, No)

What is this? Ostrich



What is this? Cat

The art of asking right questions

***What** is this?*

It is a car (and a road and a building)

A lot of applications need to answer ***Where?***

The art of asking right questions

What is this?

It is a human!

A lot of applications need to answer ***Who? / Is it the same person as X?***

Questions answered by computer vision

- ***What*** is this?
- ***Where*** are the things?
  - ..in the image
  - ..in the 3D world
- ***Who*** is this?
- ***How far*** is this thing?
- ***What is*** he/she/they ***doing***?
- *What is the shape*?
- ....

"Excuse me, is this the Society for Asking Stupid Questions?"

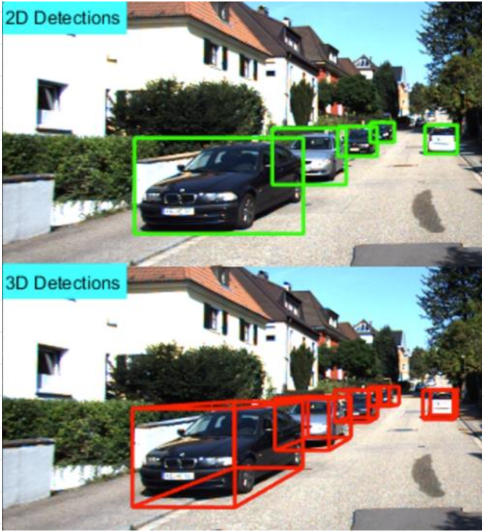
Answering the where question

Format 1: semantic segmentation

Answering the where question

Format 2: object detection

Today



Answering the where question

Format 3: instance segmentation

Images from Chaosmail Blog

Answering the where question

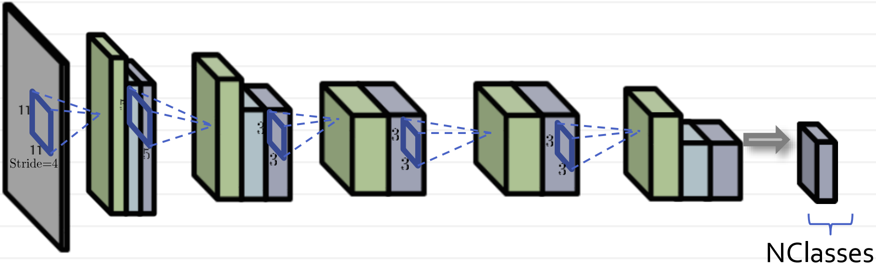
- Semantic segmentaion:
  - Relatively fast/easy
  - Allows “complete” explanation
  - Merges instances
- Object detection
  - Relatively fast/easy
  - Distinguishes instances
  - Inaccurate for some classes
  - Incomplete
- Instance segmentation
  - Complete
  - Distinguish instances
  - Accurate
  - Slow/hard

Semantic segmentation

Classification network

Semantic segmentation networks

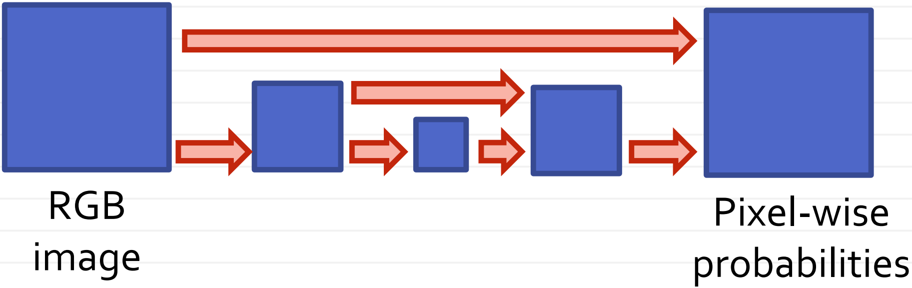
Semantic segmentation



- Problem 1: the answer is not full-size
- Problem 2: limited receptive fields

Downsampling-upsampling architectures

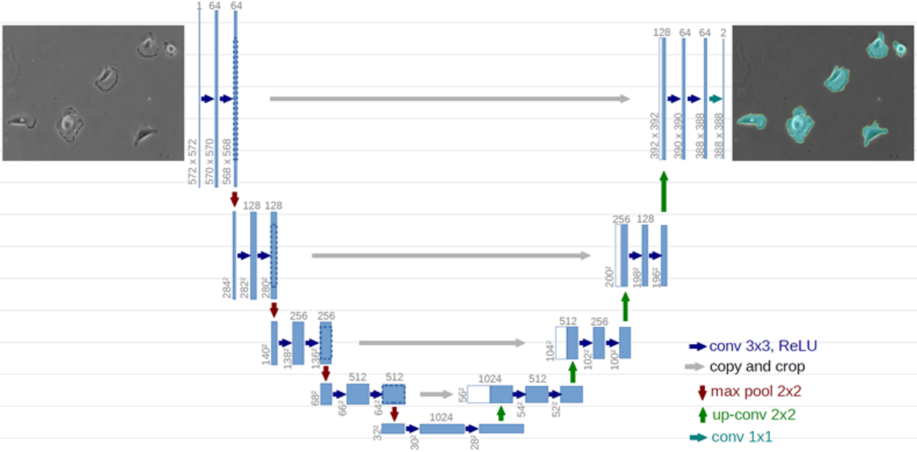
These architectures look approximately like:



- Bottom stream ensures large receptive fields
- Skip connections ensure fine spatial resolution

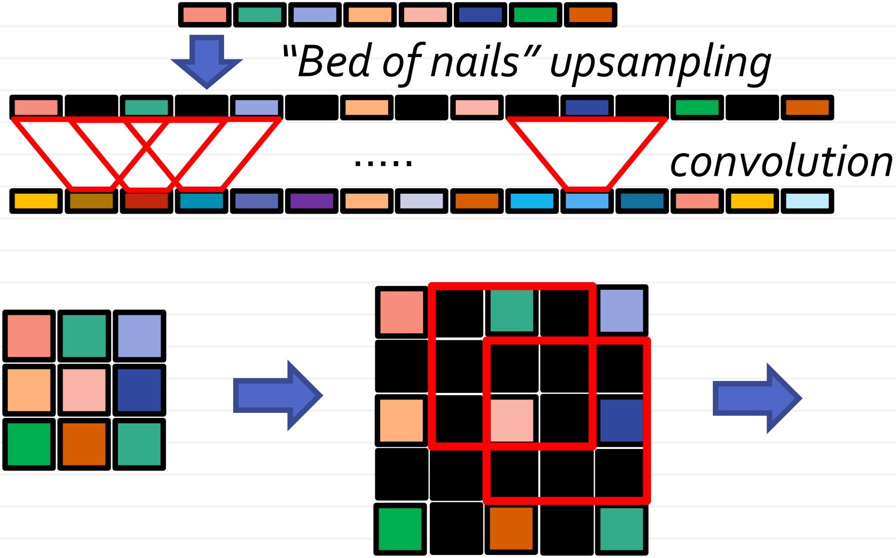
U-Net

An example of non-equivalent formulation

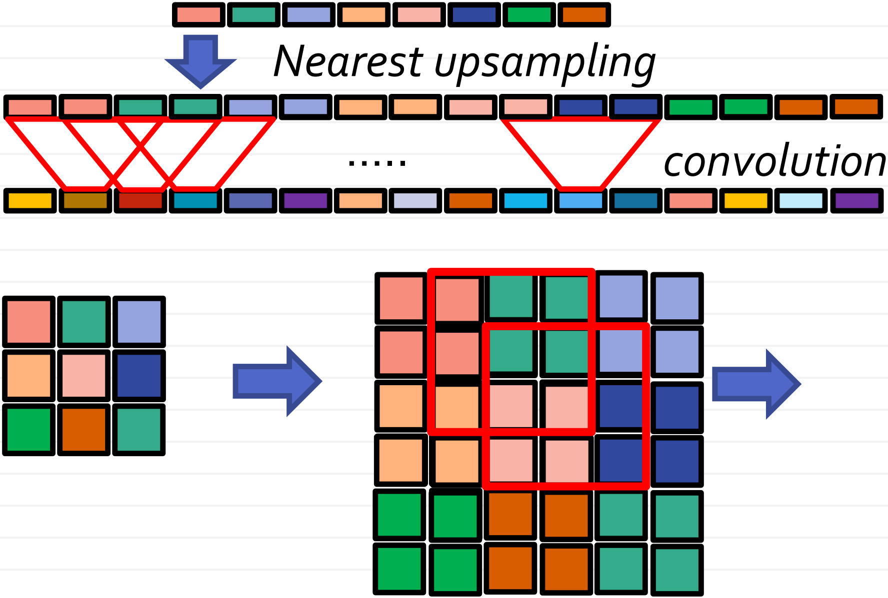


[Ronnerberger et al. MICCAI15]

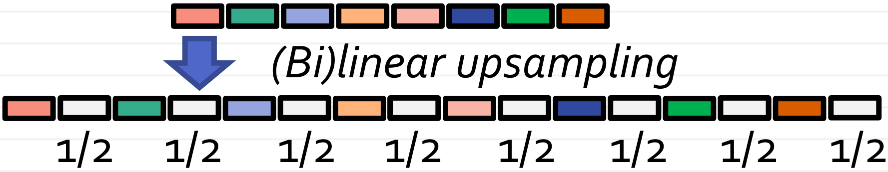
Bed-of-nails upsampling operation

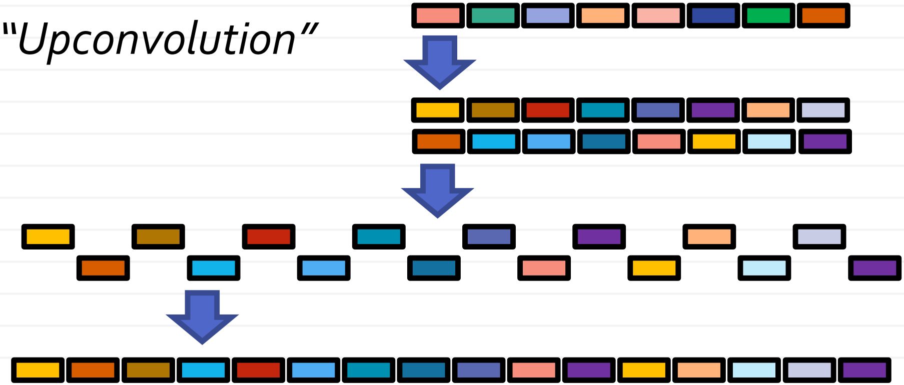


Nearest upsampling operation



Other upsampling variants

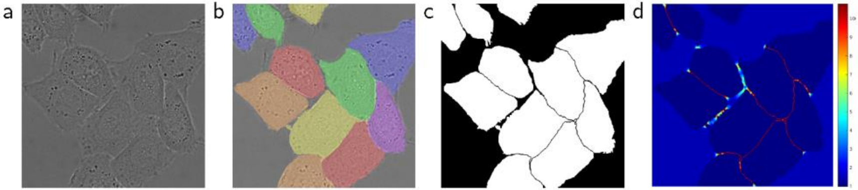




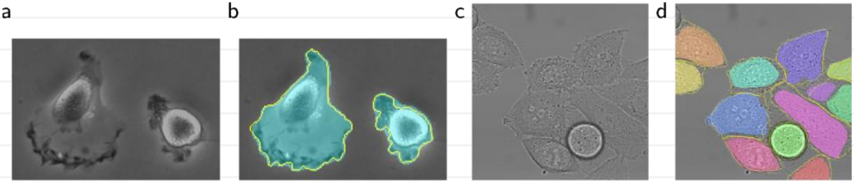
U-Net for instance segmentation

[Ronnerberger et al. MICCAI15]

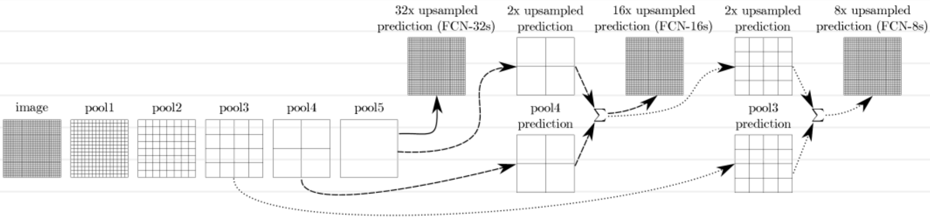
Upweighting thin borders:

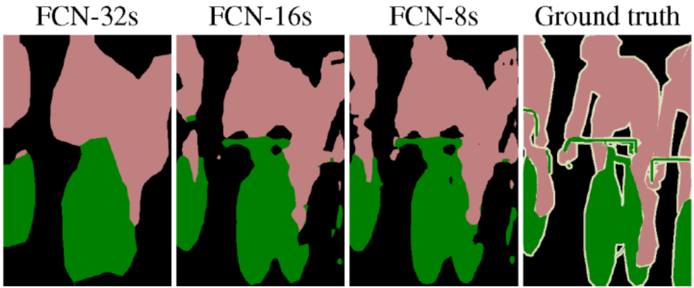


Result:



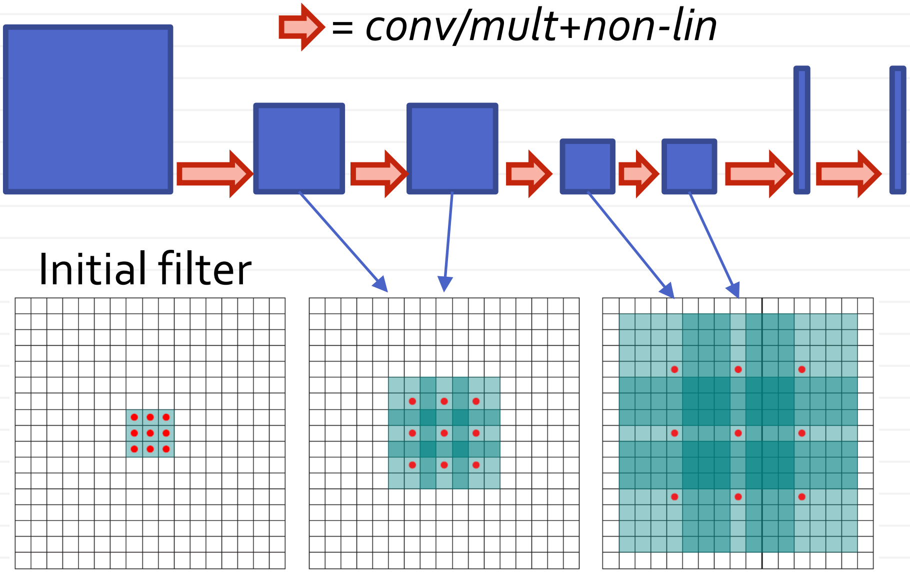
"Fully-convolutional networks"





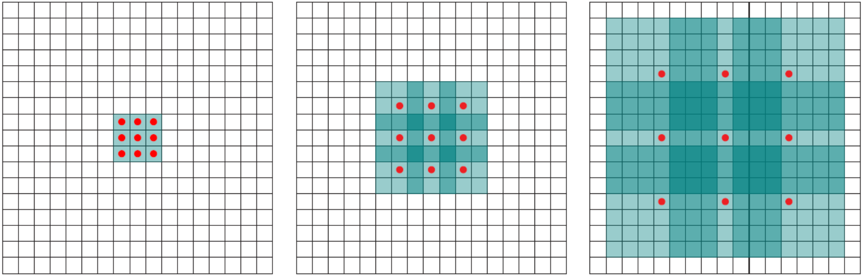
[Long et al. 2015]

Dilated convolutions



[Yu & Koltun ICLR16]

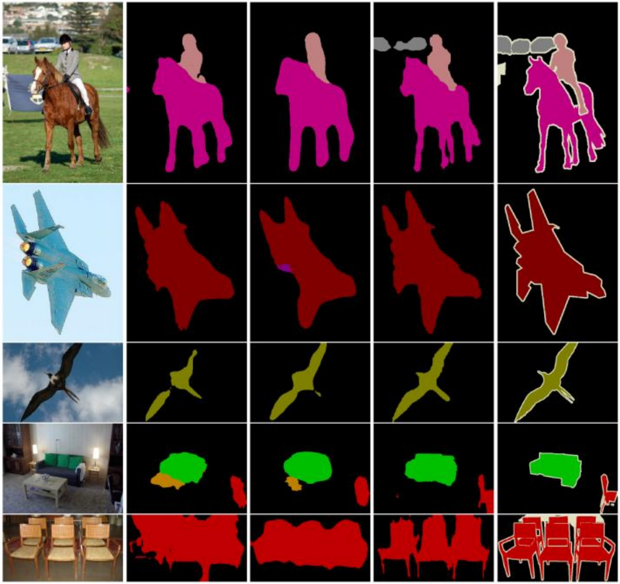
Dilated convolutions



$$V(x,y,t) = \sum_{i=x-\delta}^{x+\delta} \sum_{j=y-\delta}^{y+\delta} \sum_{s=1}^{S} K(i-x+\delta, j-y+\delta, s, t) \cdot U(x+(i-x)\,d,\; y+(j-y)\,d,\; s)$$

[Yu & Koltun ICLR16]

Dilated convolutions

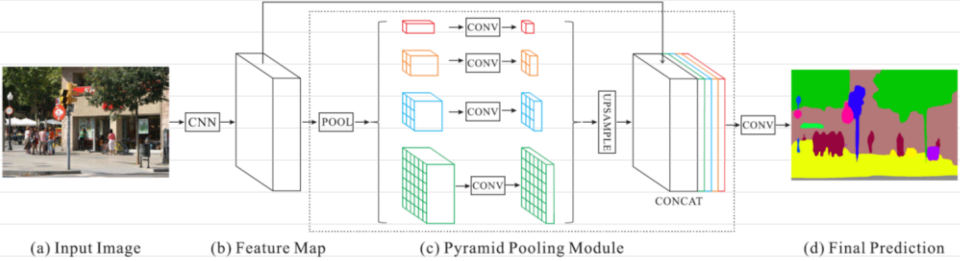


FCN Dilated [Yu & Koltun ICLR16]

Recap: ideas in semantic segmentation

- Dilated convolutions
- Upsampling layers/upconvolution layers (aka *transposed convolution/deconvolution*)
- Skip connections (to retain fine-details)
- We can mix and match all of the above

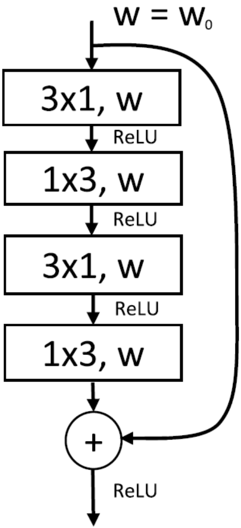
PSPNet



Method	Mean IoU(%)	Pixel Acc.(%)
ResNet50-Baseline	37.23	78.01
ResNet50+B1+MAX	39.94	79.46
ResNet50+B1+AVE	40.07	79.52
ResNet50+B1236+MAX	40.18	79.45
ResNet50+B1236+AVE	41.07	79.97
ResNet50+B1236+MAX+DR	40.87	79.61
ResNet50+B1236+AVE+DR	**41.68**	**80.04**

[Zhao et al. CVPR17]

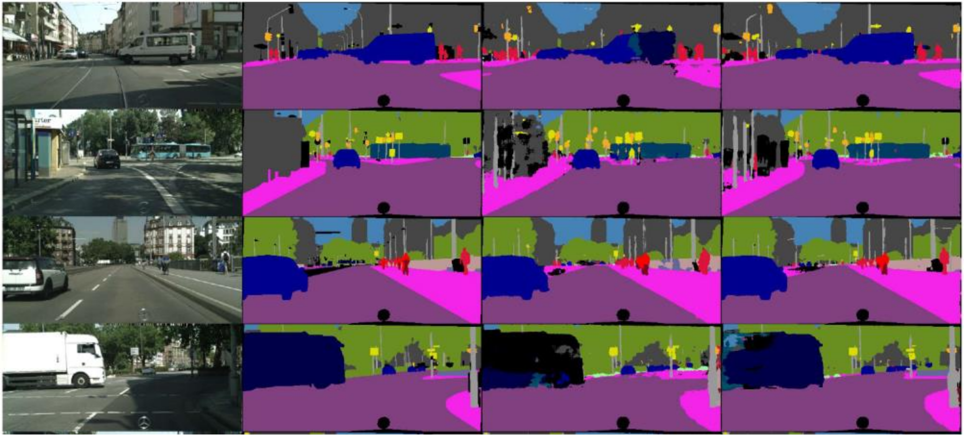
ERFNet



	Layer	Type	out-F	out-Res
ENCODER	1	**Downsampler block**	16	512x256
	2	**Downsampler block**	64	256x128
	3-7	5 x **Non-bt-1D**	64	256x128
	8	**Downsampler block**	128	128x64
	9	**Non-bt-1D** (dilated 2)	128	128x64
	10	**Non-bt-1D** (dilated 4)	128	128x64
	11	**Non-bt-1D** (dilated 8)	128	128x64
	12	**Non-bt-1D** (dilated 16)	128	128x64
	13	**Non-bt-1D** (dilated 2)	128	128x64
	14	**Non-bt-1D** (dilated 4)	128	128x64
	15	**Non-bt-1D** (dilated 8)	128	128x64
	16	**Non-bt-1D** (dilated 16)	128	128x64
DECODER	17	**Deconvolution** (upsampling)	64	256x128
	18-19	2 x **Non-bt-1D**	64	256x128
	20	**Deconvolution** (upsampling)	16	512x256
	21-22	2 x **Non-bt-1D**	16	512x256
	23	**Deconvolution** (upsampling)	C	1024x512

[Romera et al. TITS 2018]

ERFNet

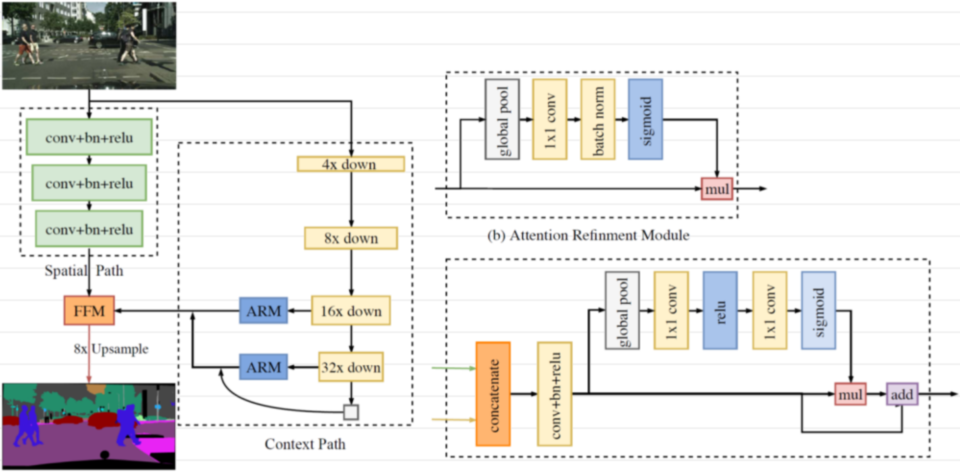


GT ENet ERFNet

- 24 ms on Titan X at 1024 x 512 resolution

[Romera et al. TITS 2018]

BiSeNet



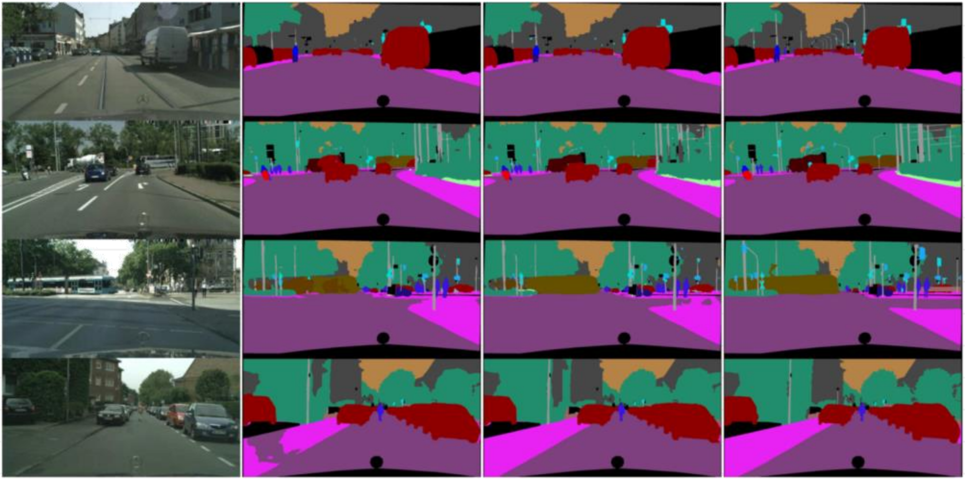
(a) Network Architecture

(b) Attention Refinment Module

(c) Feature Fusion Module

[Yu et al. ECCV18]

BiSeNet



(a) Image (b) U-Shape (c) BiSeNet (d) GT

[Yu et al. ECCV18]

Detection vs classification

Detection is harder than classification:

- Localization errors
- Huge class disbalance
- Variation in scale and aspect ratio's
- Tricky occlusions, including "intra"-occlusions

Intersection-over-Union measure

Common criterion for correct boxes:

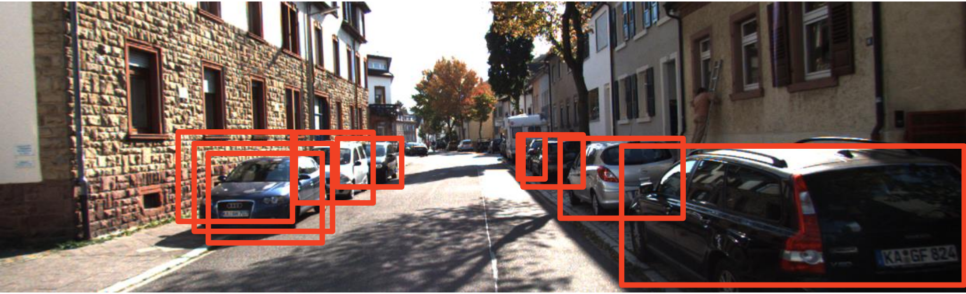
Intersection / Union > threshold (e.g. 0.5)

Double detection

Double detection of the same object is penalized as false positive

Non-maximum suppression

Almost invariably used in detection algorithms:

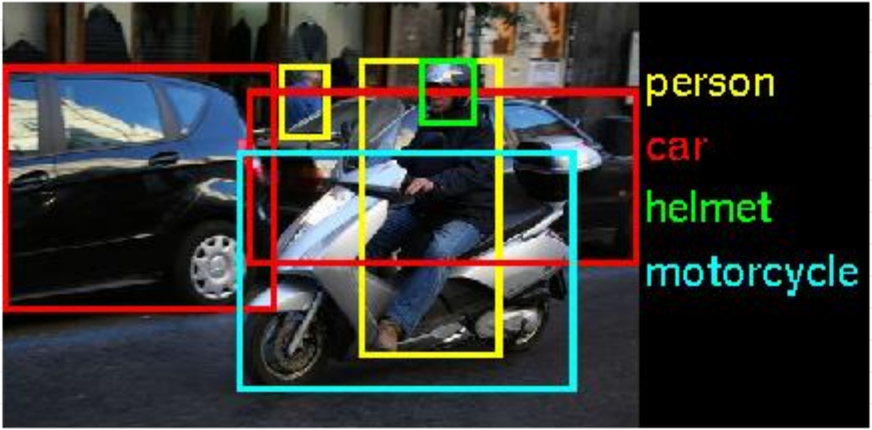


Input: set of detections ($\{B_i, s_i\}$)

- *Sort in the descending order of* s_i
- *For* $i = 1$ *to N*
- *Pick the bounding box i*
- *Suppress all subsequent boxes with IoU > 50%*

Multi-class detection

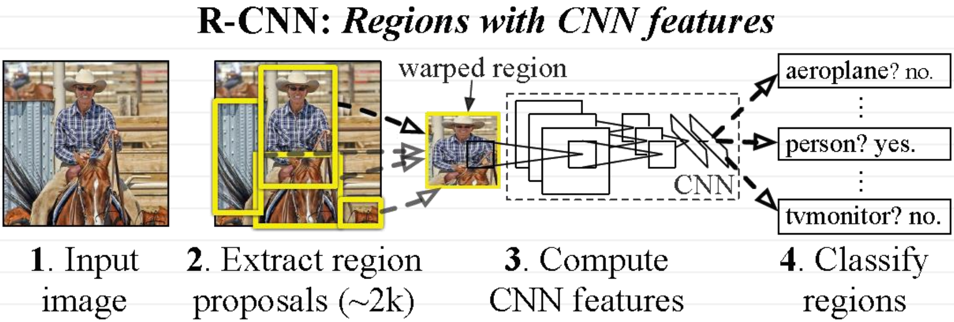
- Lots of research is going towards object detection for a large number of classes:



General ideas for object detection

- **Sliding-window:** use binary classification to classify every possible subwindow (infeasible with DL)
- **Region proposal:** pick a subset of prospective regions and score them with a binary classifier
- **Bounding box regression:** predict the coordinates of the boxes as real-valued variables

R-CNN framework

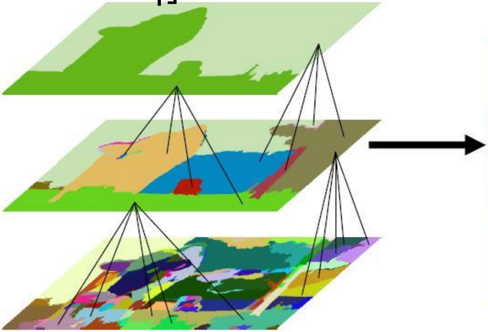


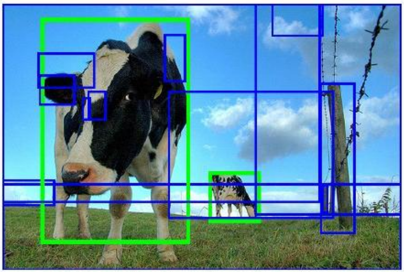
- Use an external box proposal method
- Fine-tune a CNN to score proposal

[Girshik et al. CVPR14]

Example source of external proposals

Graph-based hierarchical segmentation based on maximum-spanning trees [Felsenszwalb & Huttenlocher IJCV2004] :

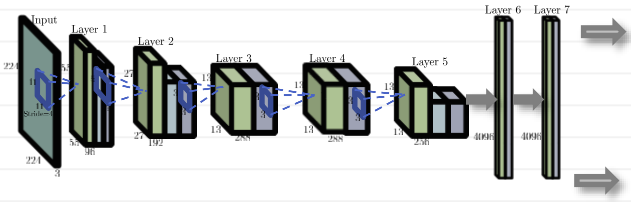


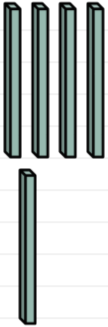


[Uijlings et al. ICCV11]

Bounding box regression

Goal: predicting 100 boxes that are likely to contain objects:





$$F_{\text{match}}(x, l) = \frac{1}{2} \sum_{i,j} x_{ij} ||l_i - g_j||_2^2$$

$$F_{\text{conf}}(x, c) = -\sum_{i,j} x_{ij} \log(c_i) - \sum_i (1 - \sum_j x_{ij}) \log(1 - c_i)$$

$$F(x, l, c) = \alpha F_{\text{match}}(x, l) + F_{\text{conf}}(x, c)$$

[Szegedy et al. 2013]

Optimization for bounding box regression

[Szegedy et al. 2014]

$$F_{\text{match}}(x,l) = \frac{1}{2}\sum_{i,j} x_{ij}||l_i - g_j||_2^2$$

$$F_{\text{conf}}(x,c) = -\sum_{i,j} x_{ij}\log(c_i) - \sum_i (1 - \sum_j x_{ij})\log(1-c_i)$$

$$F(x,l,c) = \alpha F_{\text{match}}(x,l) + F_{\text{conf}}(x,c)$$

Alternate:

- Optimize x (optimal matching)

$$x^* = \arg\min_x F(x,l,c)$$

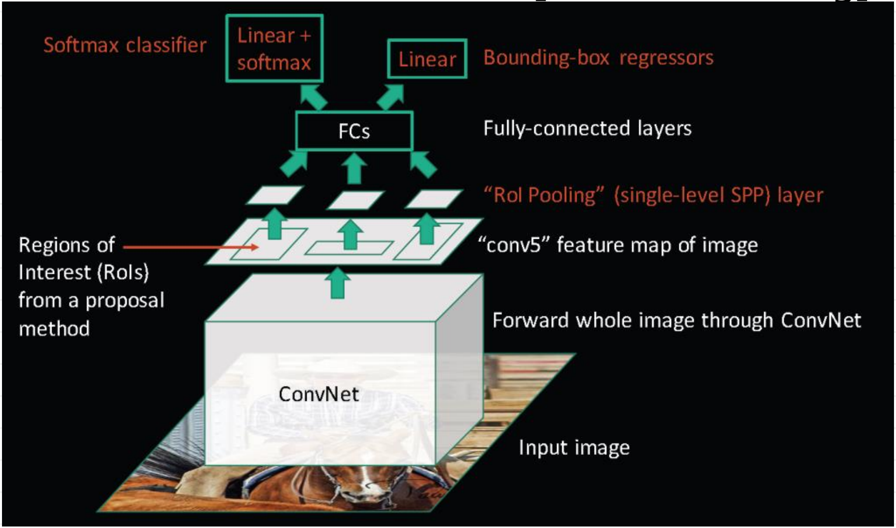
$$\text{subject to} \quad x_{ij} \in \{0,1\}, \sum_i x_{ij} = 1.$$

- Optimize network params (backprop)

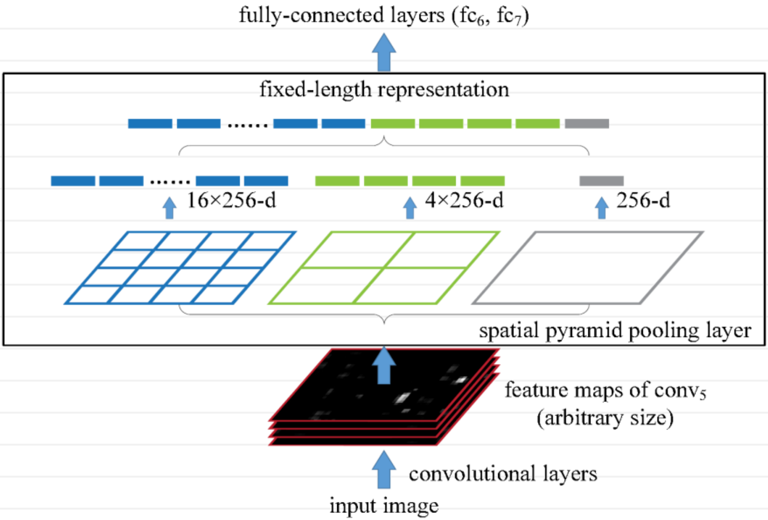
Fast R-CNN

- Processing lots of overlapping boxes is inefficient
- Alternative:

[Girshick ICCV15]



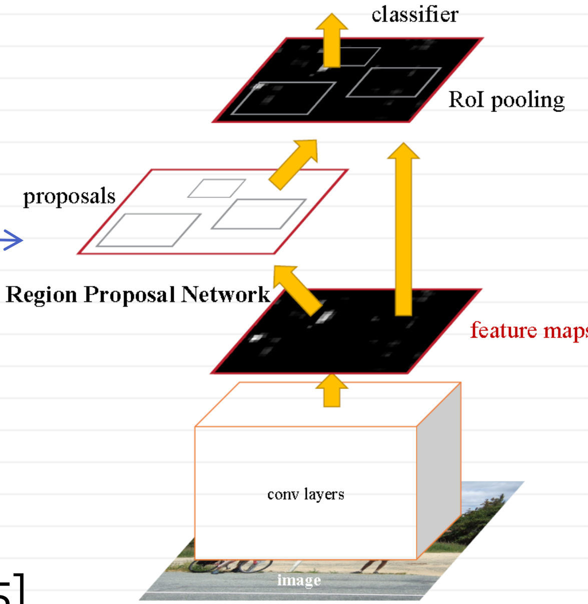
Spatial pyramid pooling



[He et al. ECCV14]

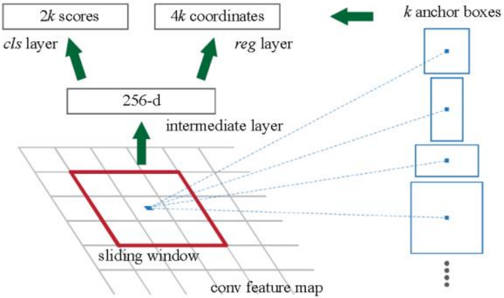
Faster CNN

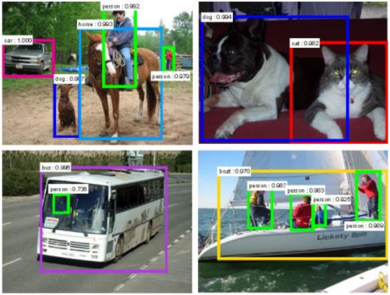
Key novelty: the proposals come from "sparse sliding window search"

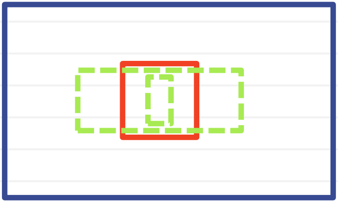


[Ren et al. NIPS15]

Faster CNN: Region-proposal network





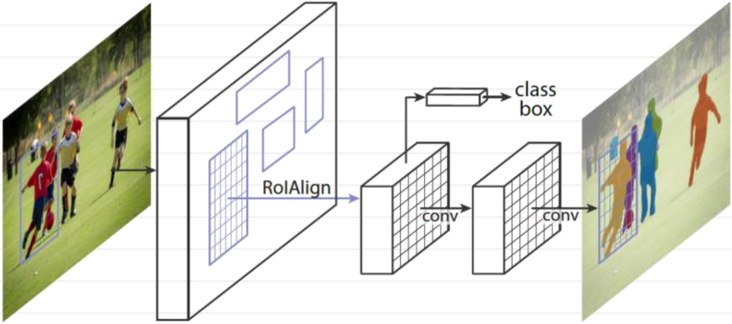


[Ren et al. NIPS15]

- Sparse set of positions
- At each positions, 9 centered "anchor" windows
- Each anchor is adjusted and scored for each class

Extension for Instance Segmentation

Mask R-CNN: adding mask prediction



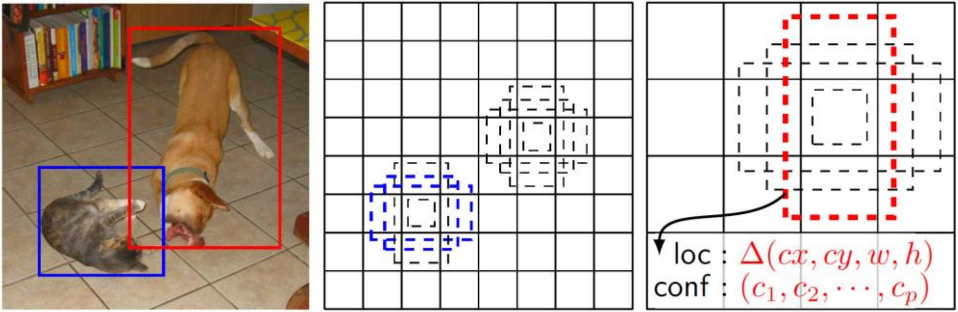
Masks for different classes are predicted and scored independently (decoupling classification and segmentation)

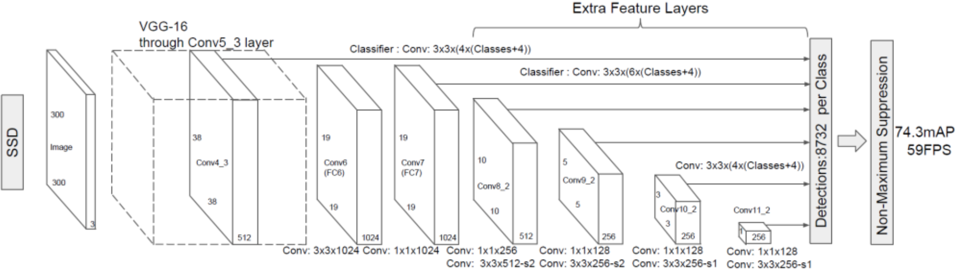
[He et al. 2017]

Mask R-CNN results

[He et al. 2017]

Single-shot detector





[Liu et al. ECCV16]

Examples: SSD detection

Recap: ideas for detection

- **ROI-pooling:** sharing convolutional features
- **Anchor+Regression:** "fast sliding window"
- **External proposals:** can be better if there is a good external source

Verification problems in vision

Key question: do two photos show the same object/subject? (*verification)*

Face recognition datasets (e.g. *MSRA-CF*):

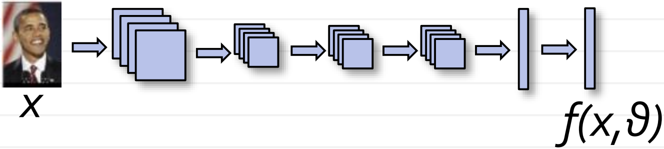
Re-identification datasets (e.g. *ViPER*):

Verification vs Classification

Key question: do two photos show the same object/subject? (*verification*)

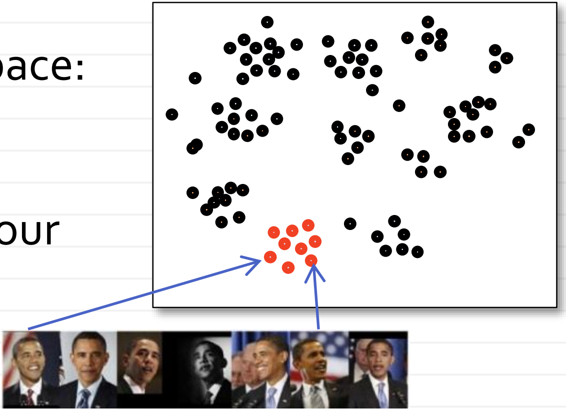
- System must be able to handle unseen "classes"
- During training classes can be numerous, small-sized, imbalanced, etc.
- Example from last lecture: retrieval

Verification as embedding learning

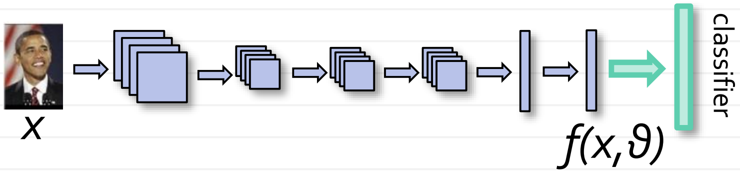


Semantic space:

NB: always normalize your descriptors!



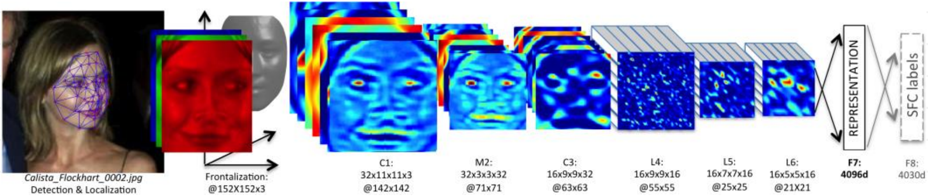
Approach 1: classification-based



- Same idea as "Train on ImageNet, use for retrieval"
- The bigger the classification dataset, the better is the performance
- Training-time classes can be seen as prototypes for test-time classes

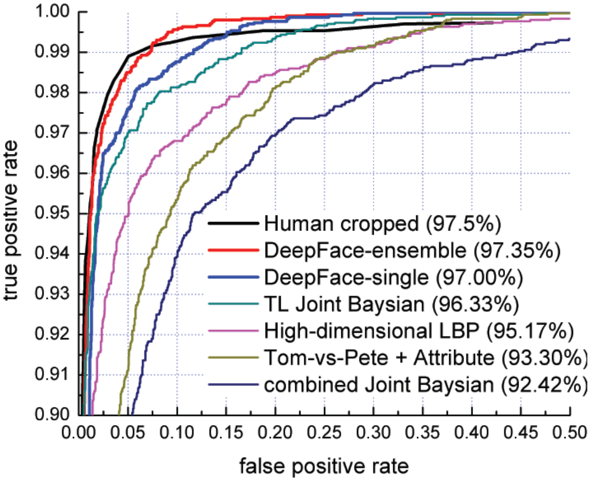
Face verification: “Deep face”

[Taigman et al. 2014]



- *Classification* network trained on 4030 people x ~1000 images.
- Target problem: *verification* (same vs different)

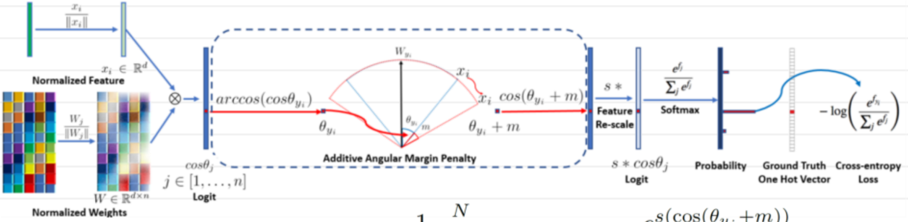
Face verification: “Deep face”



Different CNNs combined using SVM-learned weights on validation set

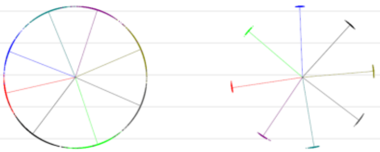
Adding normalization and margin

Angular soft-max with margin loss:



$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

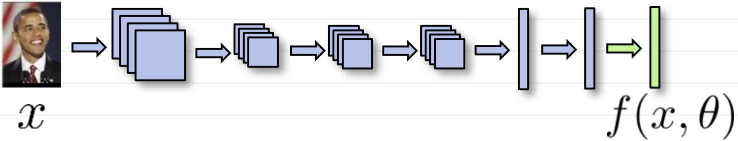
Loss Functions	LFW	CFP-FP	AgeDB-30
ArcFace (0.4)	99.53	95.41	94.98
ArcFace (0.45)	99.46	95.47	94.93
ArcFace (0.5)	**99.53**	**95.56**	**95.15**
ArcFace (0.55)	99.41	95.32	95.05
SphereFace [18]	99.42	-	-
SphereFace (1.35)	99.11	94.38	91.70
CosFace [37]	99.33	-	-
CosFace (0.35)	99.51	95.44	94.56
CM1 (1, 0.3, 0.2)	99.48	95.12	94.38
CM2 (0.9, 0.4, 0.15)	99.50	95.24	94.86
Softmax	99.08	94.39	92.33
Norm-Softmax (NS)	98.56	89.79	88.72



(a) Softmax (b) ArcFace

[Deng et al. CVPR 2019]

Pair-based learning (*aka Siamese*)



$$L^{+}\left((x_1, x_2); \theta\right) = \rho\big(f(x_1, \theta), f(x_2, \theta)\big)$$

$$L^{-}\left((x_1, x_2); \theta\right) = \max\left(0, M - \rho\big(f(x_1, \theta), f(x_2, \theta)\big)\right)$$

Example distances:

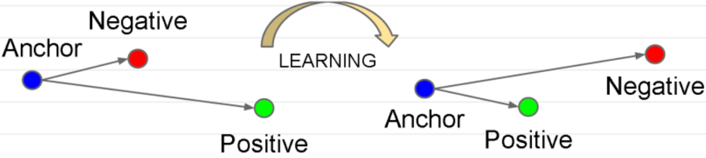
- 1 – cos
- L2 (equivalent if normalization is added)
- Separate network (verification network)

NB: all embedding-based systems work better with normalized descriptors

[Chopra et al. CVPR05]

Google "FaceNet"

[Schroff et al. CVPR15]



Simple **triplet loss**: $$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

- Use large mini-batches (1800, 40 images for several classes + lots of random)
- Take all positives from the batch
- Mine "*semi-hard*" negatives

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$$

Google "FaceNet" results

[Schroff et al. CVPR15]

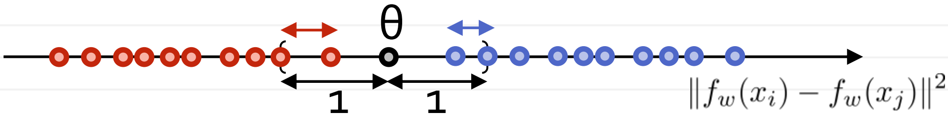
- Upto 99.63% on LFW (human is ~97%)

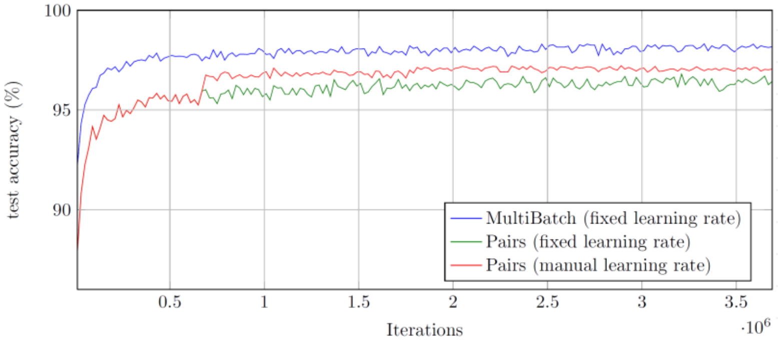
Performance vs training data:

#images	VAL
2.6M	76.3%
26M	85.1%
52M	85.1%
260M	86.2%

Quadruplet losses: multi-batch loss

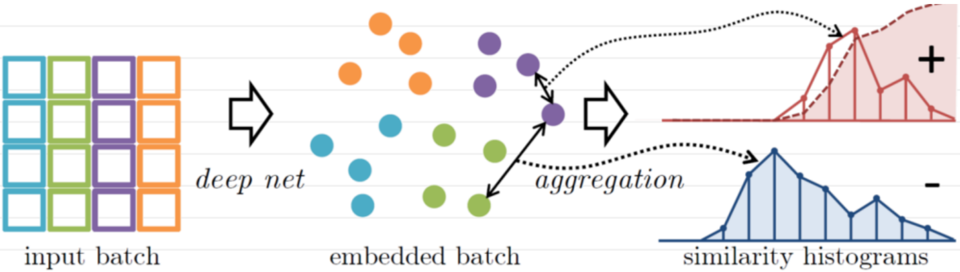
$$l(w,\theta;\ x_i,x_j,y_{ij}) = \left(1 - y_{ij}\left(\theta - \|f_w(x_i) - f_w(x_j)\|\|^2\right)\right)_+$$





[Tadmor et al, NIPS16]

Quadruplet losses: histogram loss

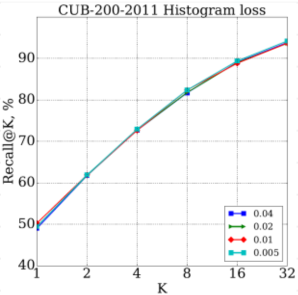


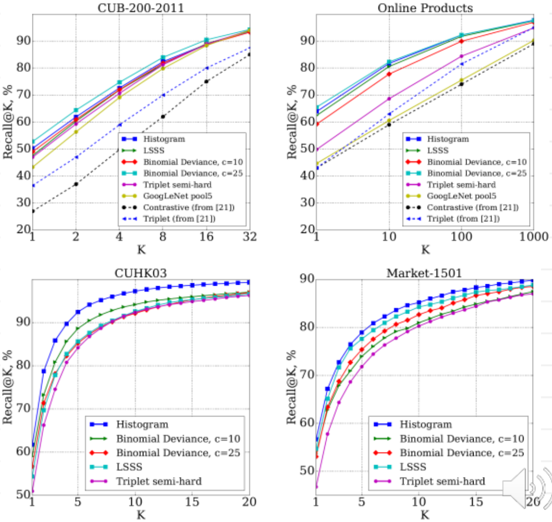
$$p_{\text{reverse}} = \int_{-1}^{1} p^{-}(x) \left[\int_{-1}^{x} p^{+}(y)\, dy \right] dx = \int_{-1}^{1} p^{-}(x)\, \Phi^{+}(x)\, dx$$

[Ustinova & Lempitsky NIPS16]

Quadruplet losses: histogram loss

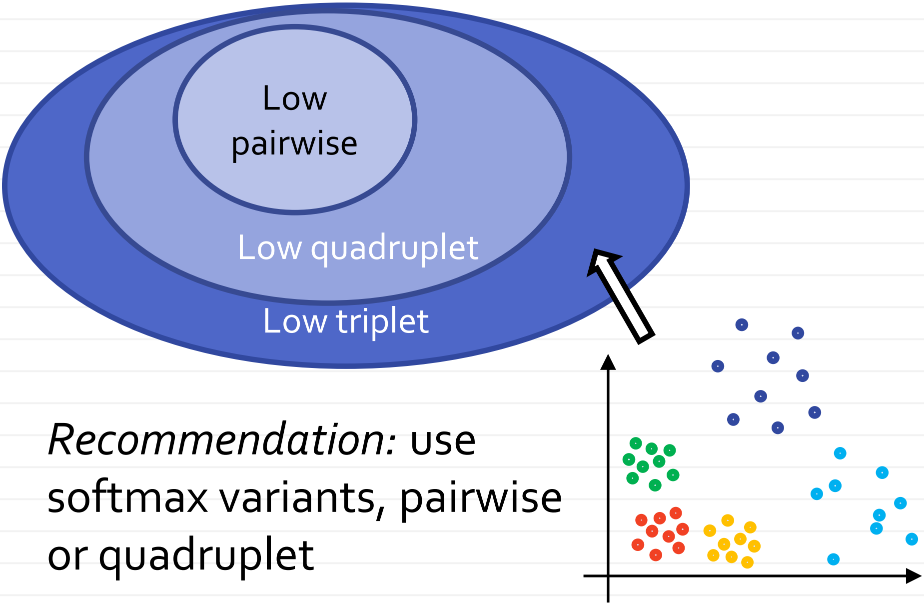
Sensitivity to the bin size:





[Ustinova & Lempitsky NIPS16]

Embedding learning: recap



Recommendation: use softmax variants, pairwise or quadruplet

References

Jonathan Long, Evan Shelhamer, Trevor Darrell:
Fully convolutional networks for semantic segmentation. CVPR 2015

Olaf Ronneberger, Philipp Fischer, Thomas Brox:
U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI (3) 2015

Fisher Yu, Vladlen Koltun:
Multi-Scale Context Aggregation by Dilated Convolutions. ICLR 2016

Ross B. Girshick:
Fast R-CNN. ICCV 2015

Ross B. Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik:
Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014

Shaoqing Ren, Kaiming He, Ross B. Girshick, Xiangyu Zhang, Jian Sun:
Object Detection Networks on Convolutional Feature Maps. NIPS 2015

References

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun:
Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV (3) 2014

Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, Arnold W. M. Smeulders:
Segmentation as selective search for object recognition. ICCV 2011

Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov:
Scalable Object Detection Using Deep Neural Networks. CVPR 2014

Spyros Gidaris, Nikos Komodakis:
LocNet: Improving Localization Accuracy for Object Detection. CoRR abs/1511.07763

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf:
DeepFace: Closing the Gap to Human-Level Performance in Face Verification. CVPR 2014

References

Sumit Chopra, Raia Hadsell, Yann LeCun:
Learning a Similarity Metric Discriminatively, with Application to Face Verification. CVPR (1) 2005

Yi Sun, Yuheng Chen, Xiaogang Wang, Xiaoou Tang:
Deep Learning Face Representation by Joint Identification-Verification. NIPS 2014

Florian Schroff, Dmitry Kalenichenko, James Philbin:
FaceNet: A unified embedding for face recognition and clustering. CVPR 2015

Omkar Parkhi, Andrea Vedaldi, Andrew Zisserman:
Deep Face Recognition. BMVC 2015

Andreas Geiger, Philip Lenz, Raquel Urtasun:
Are we ready for autonomous driving? The KITTI vision benchmark suite. CVPR 2012: 3354-3361

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg:
SSD: Single Shot MultiBox Detector. ECCV (1) 2016: 21-37

References

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg:
SSD: Single Shot MultiBox Detector. ECCV (1) 2016: 21-37

Oren Tadmor, Tal Rosenwein, Shai Shalev-Shwartz, Yonatan Wexler, Amnon Shashua: Learning a Metric Embedding for Face Recognition using the Multibatch Method. NIPS 2016: 1388-1389

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele:
The Cityscapes Dataset for Semantic Urban Scene Understanding. CVPR 2016: 3213-3223

Evgeniya Ustinova, Victor S. Lempitsky:
Learning Deep Embeddings with Histogram Loss. NIPS 2016: 4170-4178

Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick
Mask R-CNN, ArXiV 2017

References

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, Jiaya Jia:
Pyramid Scene Parsing Network. CVPR 2017: 6230-6239

Eduardo Romera, Jose M. Alvarez, Luis Miguel Bergasa, Roberto Arroyo:
ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. IEEE Trans. Intelligent Transportation Systems 19(1): 263-272 (2018)

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, Nong Sang:
BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. ECCV (13) 2018: 334-349

Jiankang Deng, Jia Guo, Stefanos Zafeiriou:
ArcFace: Additive Angular Margin Loss for Deep Face Recognition. CVPR 2019