DEEP LEARNING

An introduction to deep learning

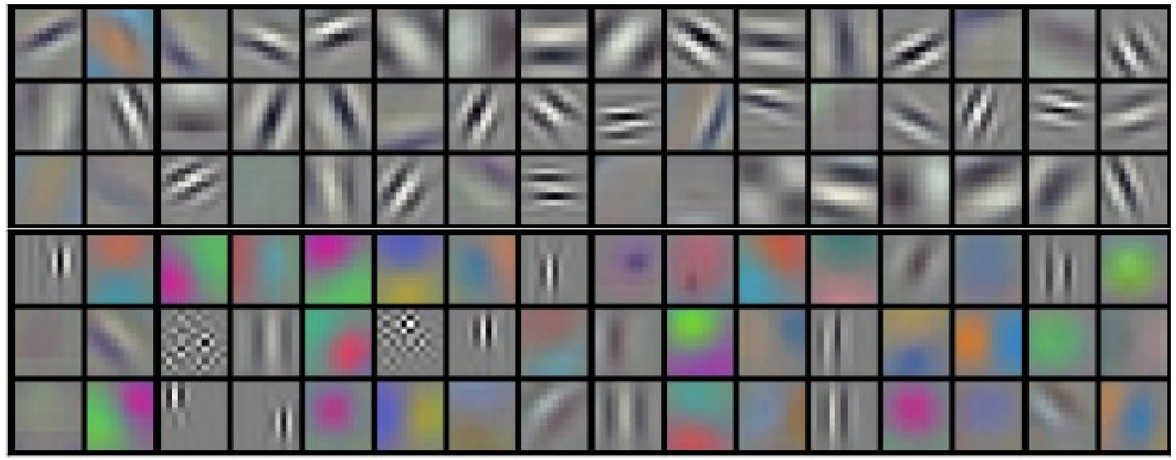
Jeremy Fix

December 18, 2020

OPENING THE BLACK-BOX : UNDERSTANDING NEURAL NETWORKS

CNN VISUALIZATION

You can interpret the kernels of the first layers, e.g. with Alexnet :



But what about the hidden/output layers ? Why is this network telling my tumor is malign ?

- Distill.pub: Feature visualization
- Distill.pub The building blocks of interpretability
- OpenAI Microscope
- Tensorflow lucid / Pytorch lucent

VISUALIZATION BY OPTIMIZATION

- We can optimize for the images of a dataset that maximally activate some units/channels/layers (Erhan, Bengio, Courville, & Vincent, 2009), (Olah, Mordvintsev, & Schubert, 2017)

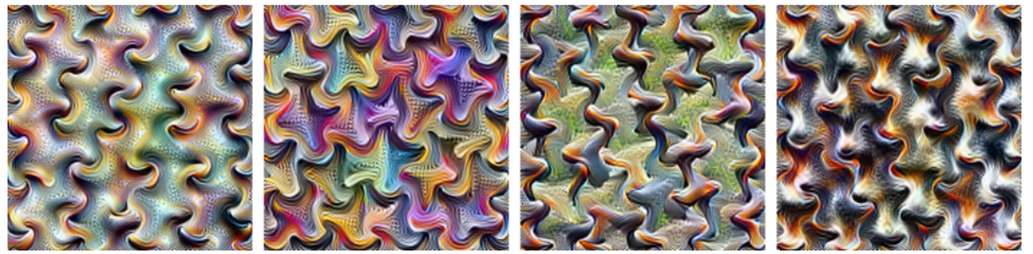
From a dataset :

$x = \text{argmax}_{x \in \mathcal{D}} h_{i,j}(\theta, x)$

Looking for images that maximally activate a unit

(Regularized) Gradient ascent :

$x = \text{argmax}_{x \in \mathbb{R}^n} h_{i,j}(\theta, x) - \lambda \mathcal{R}(x)$



Optimization with diversity reveals four different, curvy facets. *Layer mixed4a, Unit 97*

Optimizing images that maximally activate a unit

You need to use regularization or otherwise, the generated images contain artifacts (e.g. high frequency patterns) such as :

- total variation regularizer
- jitter/rotate/scale the input before the update

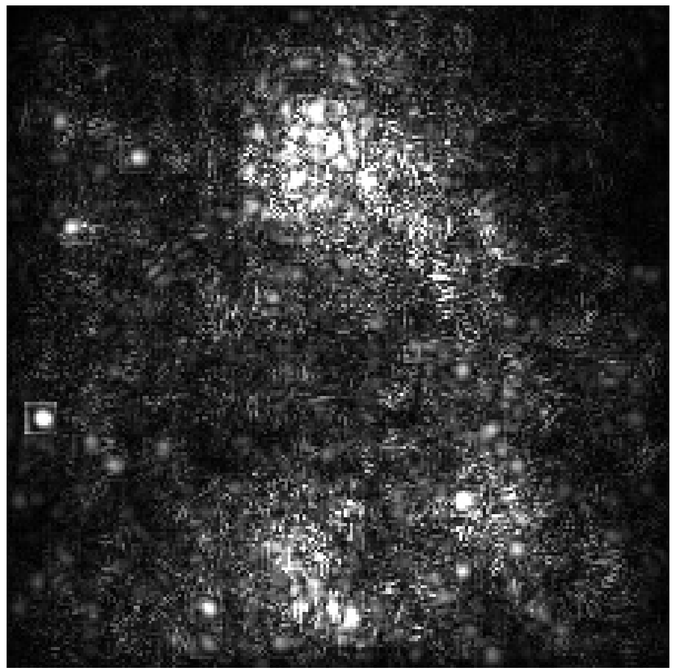
See Distill.pub: Feature visualization. The images come from (Olah et al., 2017). This opened the way toward adversial examples.

VISUALIZATION BY ATTRIBUTION

Given an input, we can compute how important are its pixels to the activation of a unit to produce **saliency maps** (Simonyan, Vedaldi, & Zisserman, 2014)

e.g. compute the gradient of the class **logits** (before the softmax)

Original image



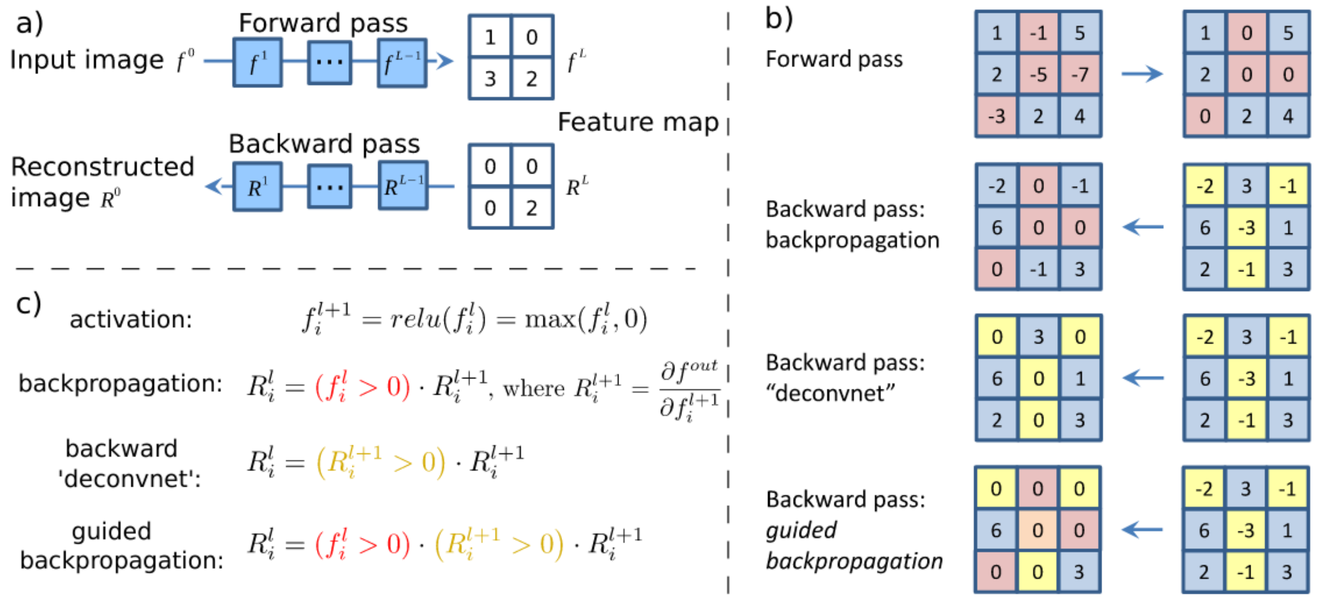
Gradient magnitude

Guided Backpropagation (Springenberg, Dosovitskiy, Brox, & Riedmiller, 2015)

Images produced for AlexNet with this pytorch implementation of CNN visualizations.

GUIDED BACKPROPAGATION

Introduced in (Springenberg et al., 2015)

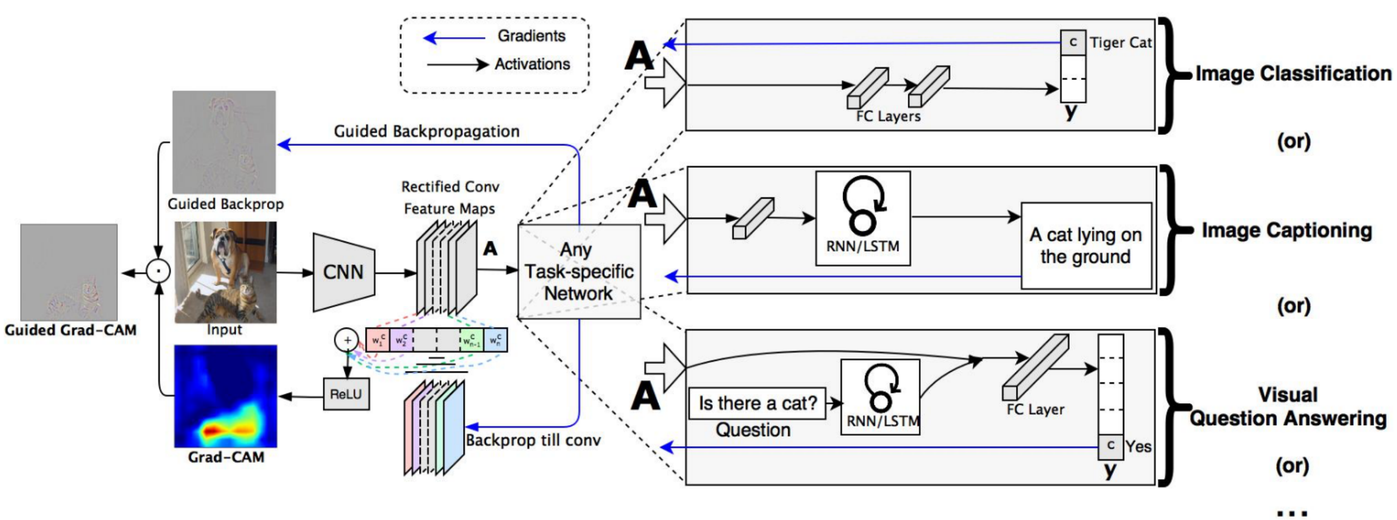


The functionning of backprop, deconv and guided backprop

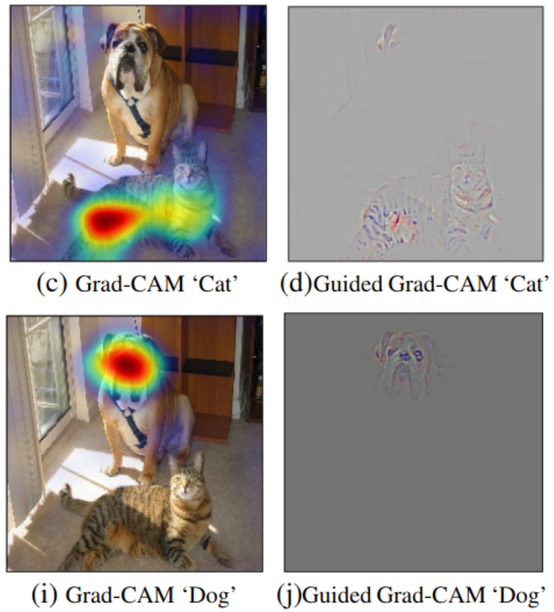
GRAD - CLASS ACTIVATION MAP

Introduced in (Selvaraju et al., 2020) :

- forward propagate the image
- compute the gradient of the class score (before the softmax) w.r.t. to last convolution layers
- spatially average these gradient images to obtain channel importances α_k
- linearly combine the feature maps with the importances (and relu it)
  $\longrightarrow$ coarse activation map



Grad-cam functionning (Selvaraju et al., 2020)



Guided Grad-cam for a higher resolution visualization

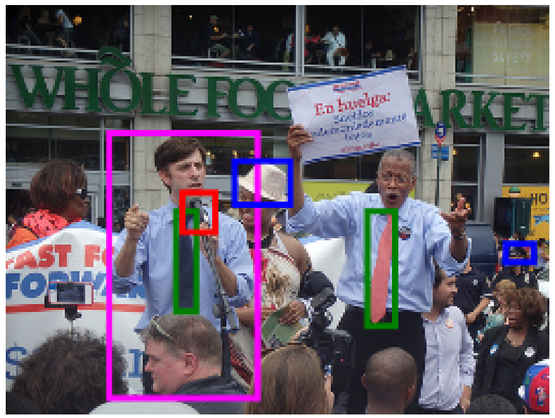
OBJECT DETECTION : INTRODUCTION

PROBLEM STATEMENT

Given :

- images x_i,
- targets y_i which contains objects bounding boxes and labels

Examples from ImageNet (see here)



ILSVRC2014_train_00005559 : few objects annotated.

ILSVRC2014_train_00029372 : 12 objets with occlusions

Bounding boxes given, **in the datasets** (the predictor parametrization may differ), by : $[x, y, w, h]$, $[x_{min}, y_{min}, x_{max}, y_{max}]$, ...

Datasets : Coco, ImageNet, Open Images Dataset

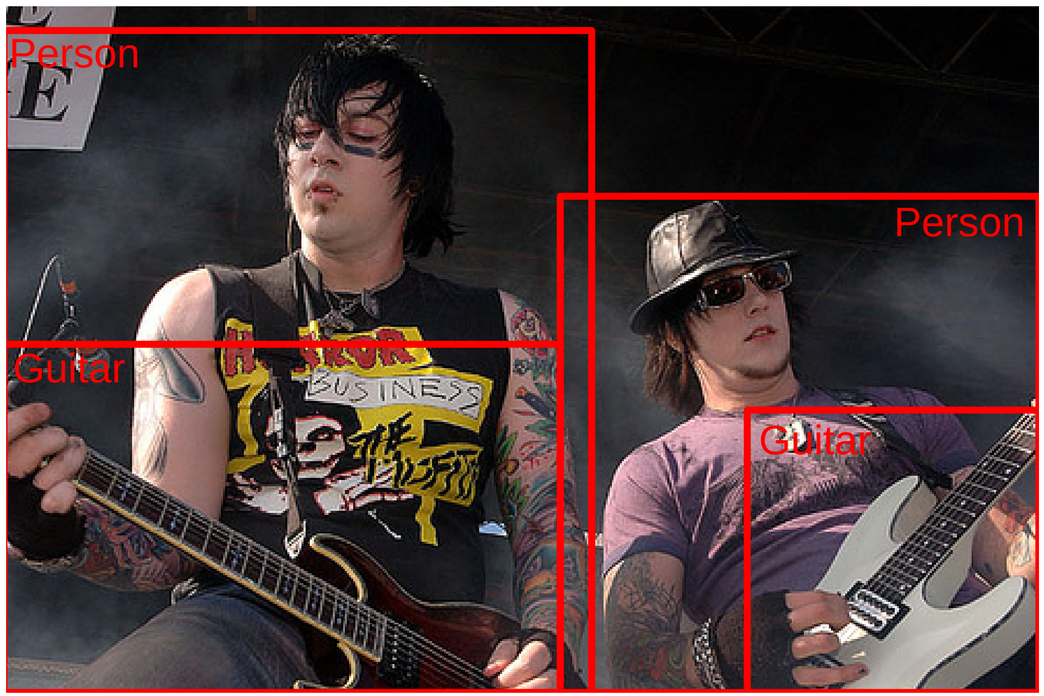
Recent survey : Object detection in 20 years: a survey

METRICS TO MEASURE THE PERFORMANCES

The metrics should ideally capture :

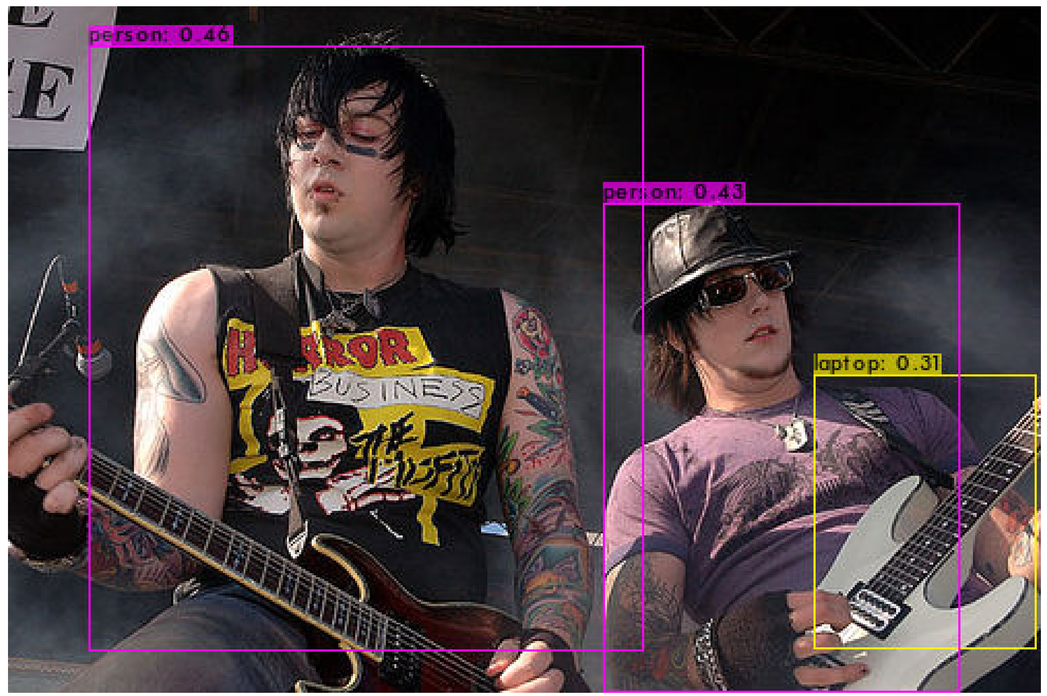
- the quality of the bounding boxes
- the quality of the label found in the bounding box

Given the “true” bounding boxes:



Pascal VOC 2012_000180.jpg

Quantify the quality of these predictions



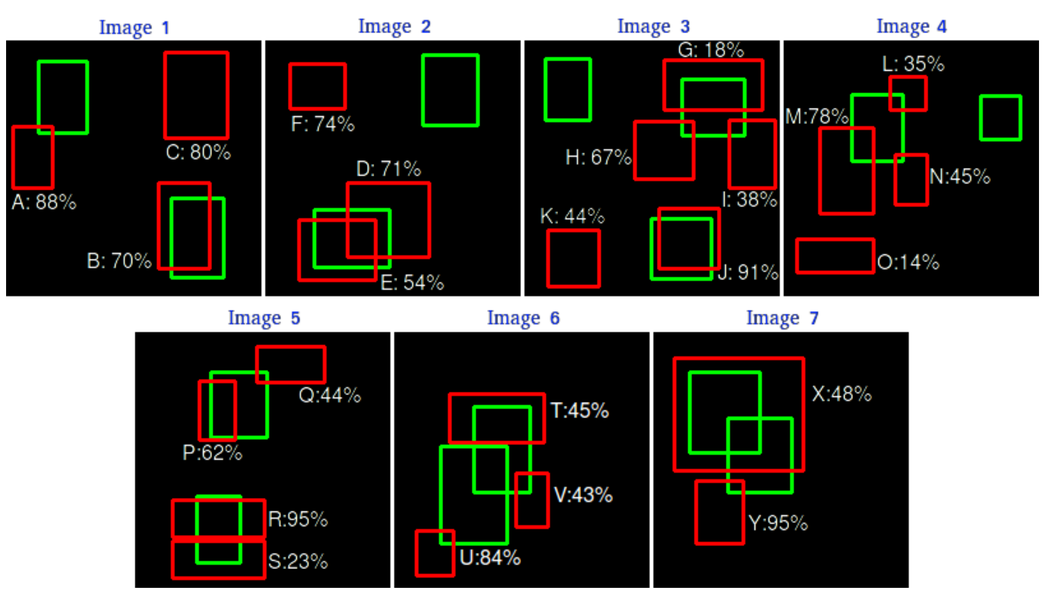
Yolo v4 tiny predictions using darknet

A predictor should output labeled bounding boxes with a confidence score.

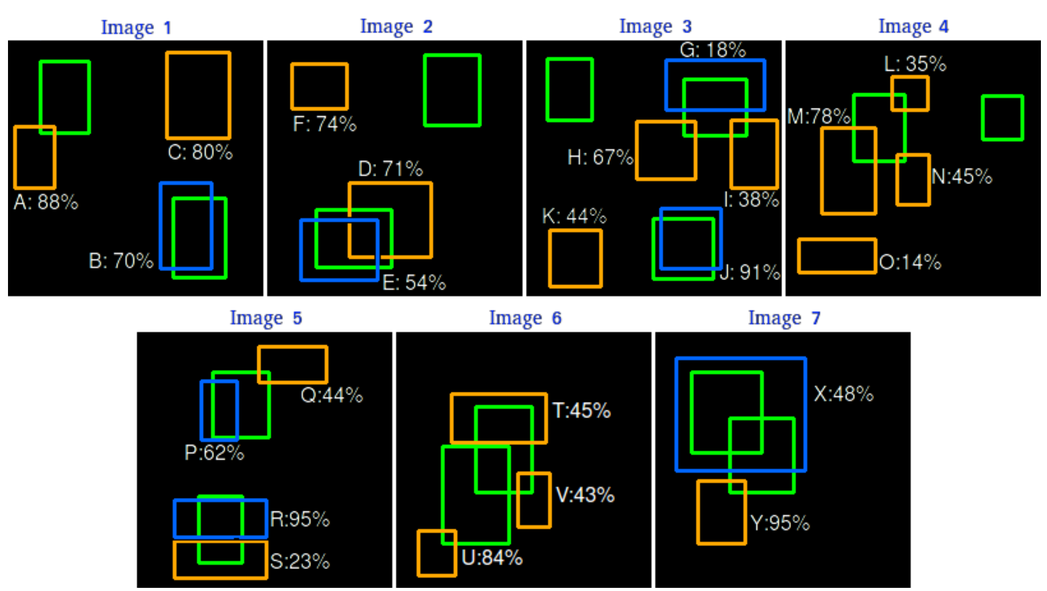
THE PASCAL MAP METRIC

For every class individually, every prediction of every images are considered :

- a true positive (TP) if IoU with ground truth bbox of the same class is higher than 0.5
- a false positive (FP) otherwise or if there is a TP for the same ground truth with a higher confidence (“5 detections (TP) of a single object is counted as 1 correct detection and 4 false detections”)



Example 24 predictions (red) with 15 ground truth for one arbitrary class



The 24 predictions are either TP (blue) or FP (orange) with IoU>0.3

Examples from the Object detection metrics repository.

THE PASCAL MAP METRIC

Rank the predictions by decreasing confidence and compute the average (of the interpolated corrected) precision :

$$\text{precision} = \frac{TP}{TP + FP}$$

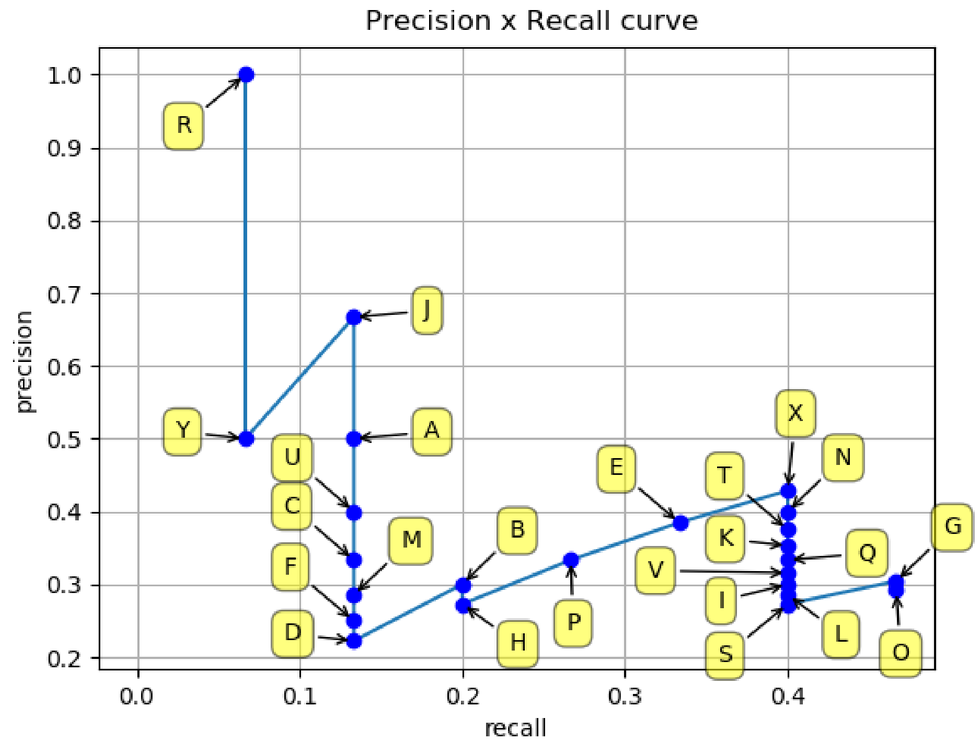
Which fraction of your detections are actually correct.

$$\text{recall} = \frac{TP}{TP + FN} = \frac{TP}{\#\text{gt bbox}}$$

Which fraction of labeled objects do you detect.

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Precision/recall for the ordered predictions



Precision recall curve

- Average over all the classes to get the mAP.

Examples from the Object detection metrics repository.

OTHER METRICS

MS-Coco detection challenge (see here):

- uses an average over multiple IoU $[0.5, 0.55, \dots 0.95]$

ImagNet (see (Russakovsky et al., 2015)) :

- uses the average precision for the different recalls, with a IoU adapted from the ground truth objects size

ImageNet now on Kaggle :

- the score is for object localization: if you predict only one object per image, that is correct, you get a perfect score

Open image evaluation:

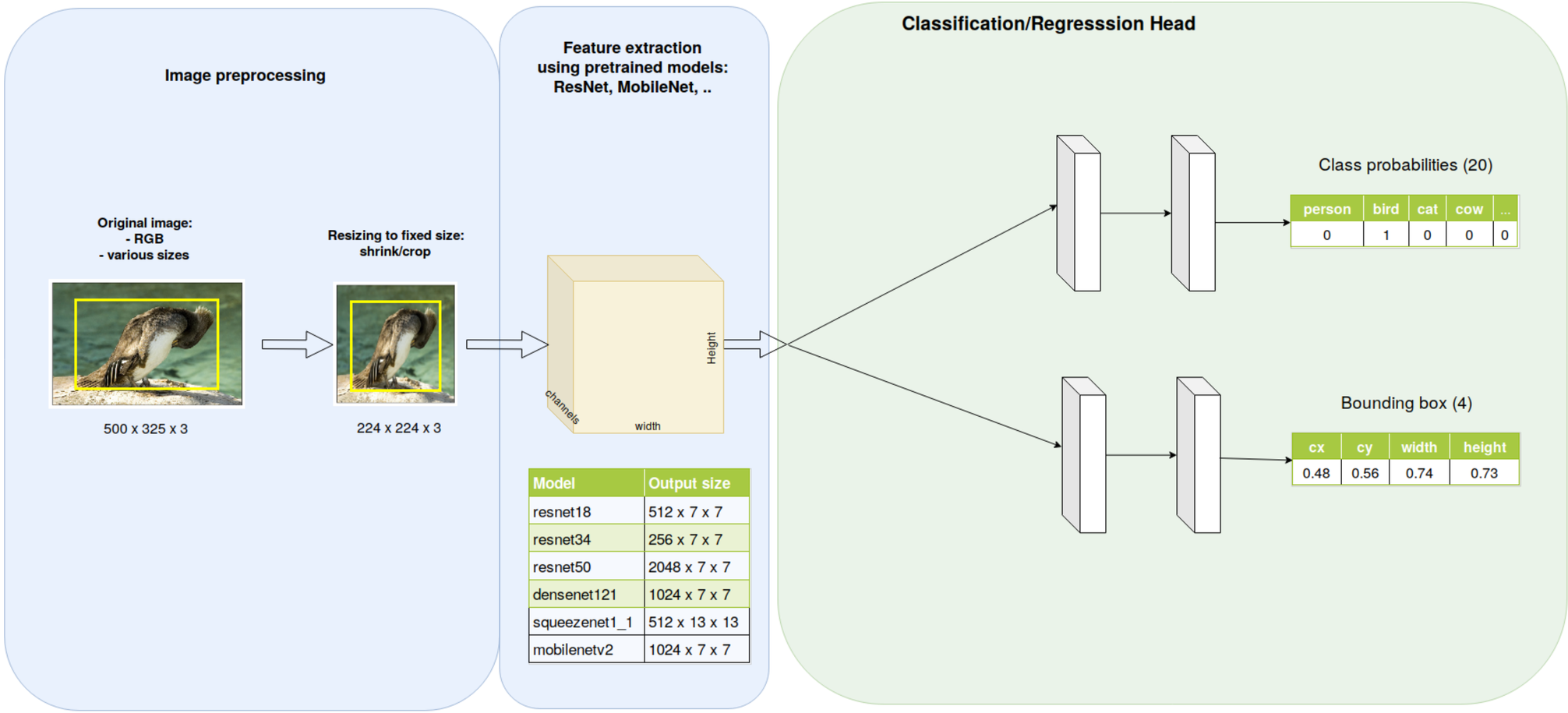
- uses a variant of VOC 2010. more details here

OBJECT LOCALIZATION

A FIRST STEP: OBJECT LOCALIZATION

Suppose you have a single object to detect, can you localize it into the image ?

Single object detection : Classification and Bounding box regression



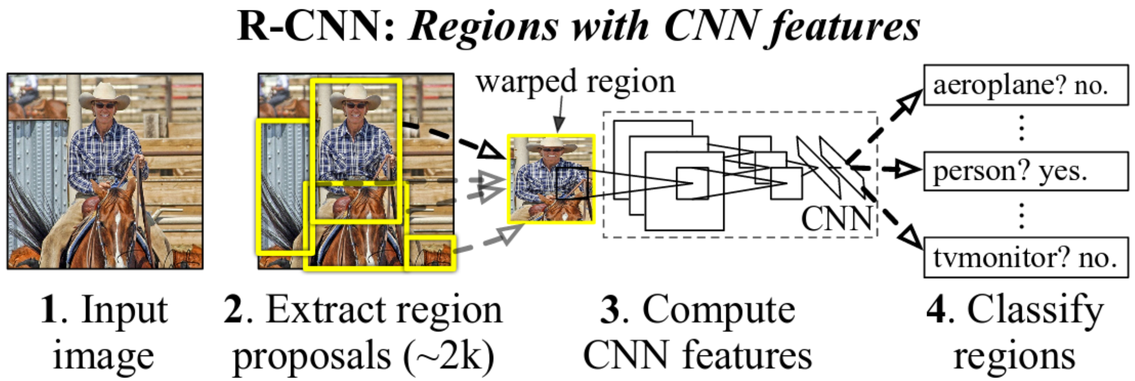
OBJECT DETECTION : STATE OF THE ART

RCNN

How can we proceed with multiple objects ? (Girshick, Donahue, Darrell, & Malik, 2014) proposed to :

- use **selective search** for proposing bounding boxes
- to classify with a SVM from the features extracted by a **pretrained** DNN.
- to optimize localization with linear bbox adaptors

Revolution in the object detection community (vs. “traditional” HOG like features).



Drawback :

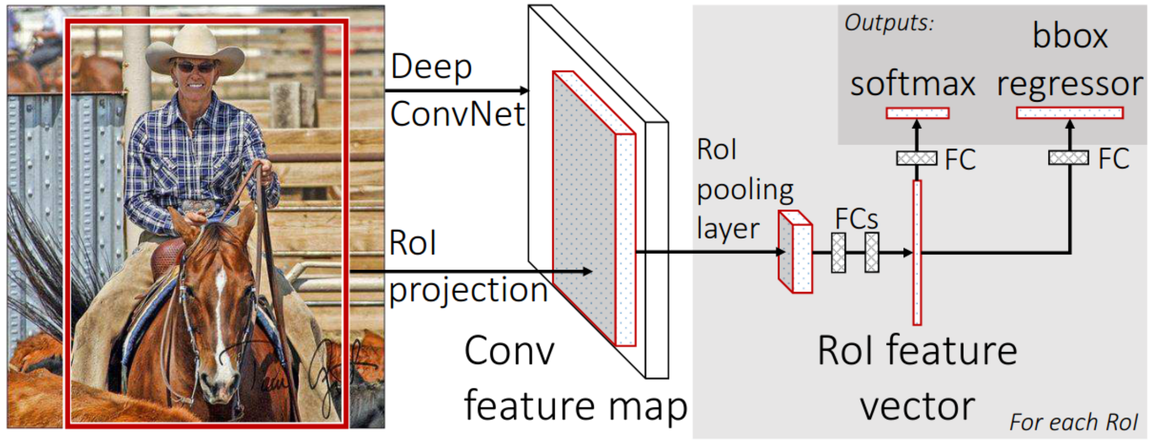
- external algorithm (not in the computational graph, not trainable)
- one forward pass per bounding box proposal ($\sim 2K$ or so) $\rightarrow$ training and test are slow ($47s.$ per image with VGG16)

Notes : pretained on ImageNet, finetuned on the considered classes with warped images. Hard negative mining (boosting).

FAST RCNN

Introduced in (Girshick, 2015). Idea:

- just one forward pass
- cropping the convolutional feature maps (e.g. $(1, 512, H/16, W/16)$ conv5 of VGG16)
- max-pool the **variable sized** crop to a **fixed-sized** (e.g. 7×7) vector before dense layers: **ROI pooling**



Drawbacks:

- external algorithm for ROI proposals: not trainable and slow
- ROIs are snapped to the grid (see here) $\longrightarrow$ ROI align

Github repository. CVPR'15 slides

Notes : pretrained VGG16 on ImageNet. Fast training with multiple ROIs per image to build the 128 mini batch from $N = 2$ images, using 64 proposals : 25% with IoU>0.5 and 75% with $IoU \in [0.1, 0.5[$. Data augmentation : horizontal flip. Per layer learning rate, SGD with momentum, etc..

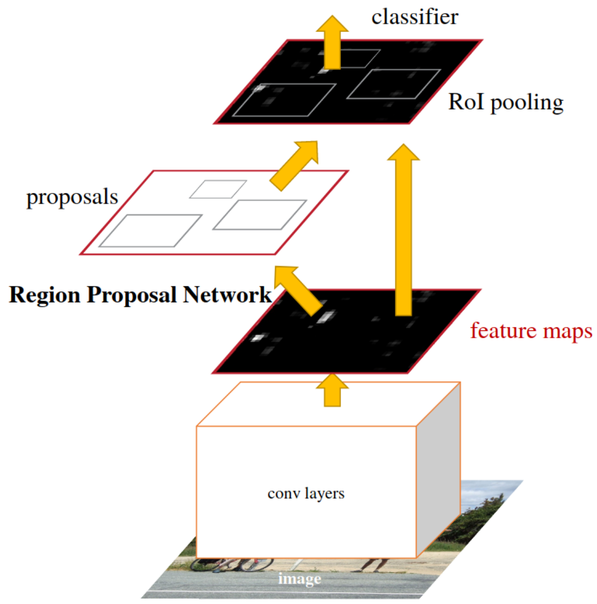
Multi task loss :

$$L(p, u, t, v) = -\log(p_u) + \lambda \text{smooth L1}(t, v)$$

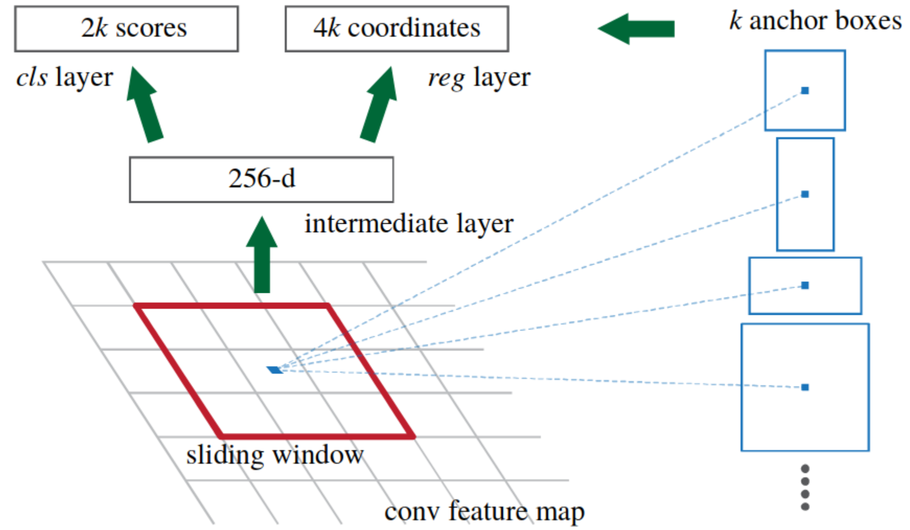
The bbox is parameterized as in (Girshick et al., 2014). Single scale is more efficient than multi-scale.

FASTER RCNN

Introduced in (Ren, He, Girshick, & Sun, 2016). The first **end-to-end trainable** network. Introducing the **Region Proposal Network** (RPN). A RPN is a sliding Conv(3×3) - Conv(1×1, k + 4k) network (see here). It also introduces anchor boxes of predefined aspect ratios learned by vector quantization.



Architecture



Region proposal network with anchors at 3 scales, 3 aspect ratios

Check the paper for a lot of quantitative results.

Github repository

Bbox parametrization identical to (Girshick et al., 2014), with smooth L1 loss. Multi-task loss for the RPN. Momentum(0.9), weight decay(0.0005), learning rate (0.001) for 60k minibatches, 0.0001 for 20k.

Multi-step training. Gradient is non-trivial due to the coordinate snapping of the boxes (see ROI align for a more continuous version)

With VGG-16, the conv5 layer is $H/16, W/16$. For an image 1000×600, there are $60 \times 40 = 2400$ anchor boxes centers.

FPN

Introduced in (Lin et al., 2017)

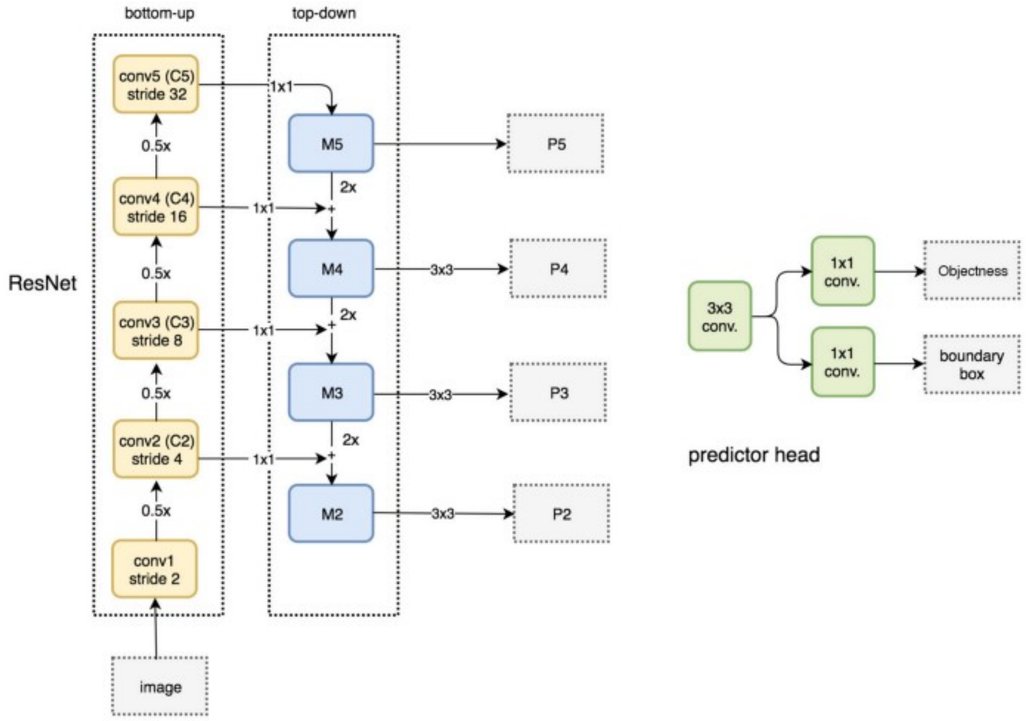


Illustration of a feature pyramid network. Credit to Jonathan Hui

Upsampling is performed by using nearest neighbors.

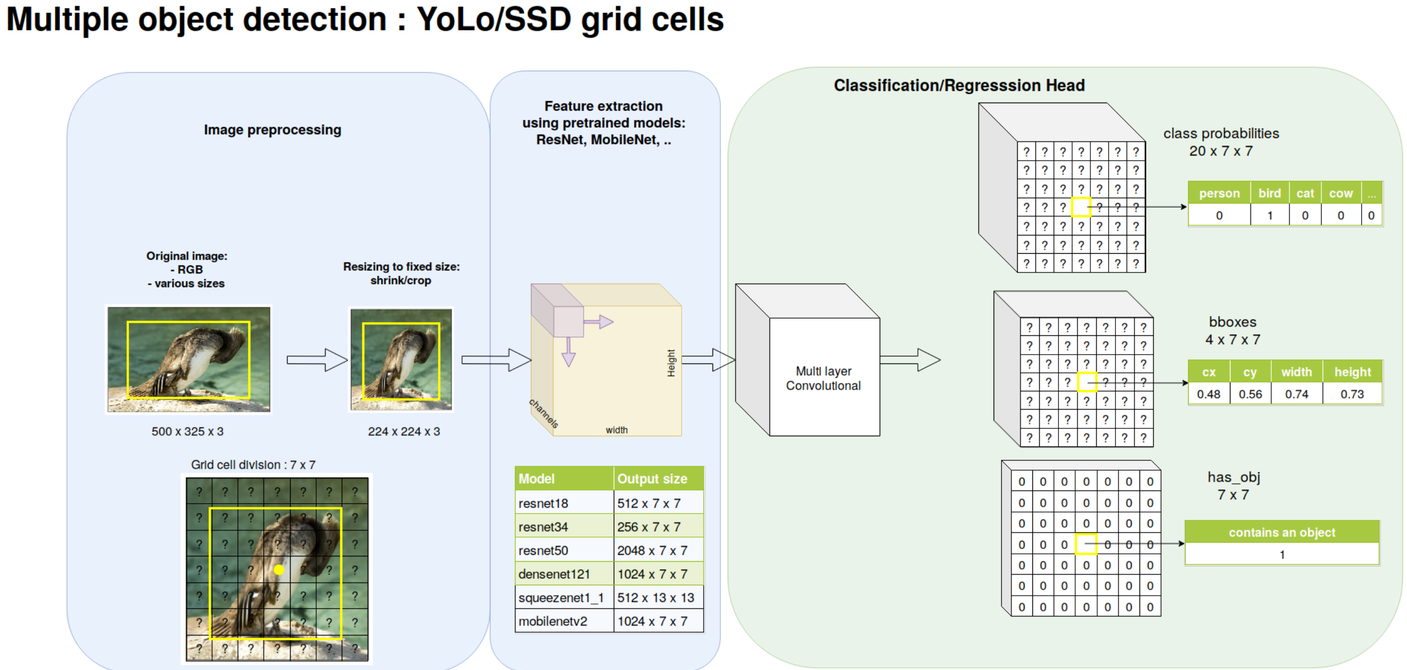
For object detection, a RPN is run on every scale of the pyramid P_2, P_3, P_4, P_5.

ROIPooling/Align is fed with the feature map at a scale depending on ROI size. Large ROI on small/coarse feature maps, Small ROI on large/fine feature maps

YOU ONLY LOOK ONCE (YOLO V1,V2,V3) (1/2)

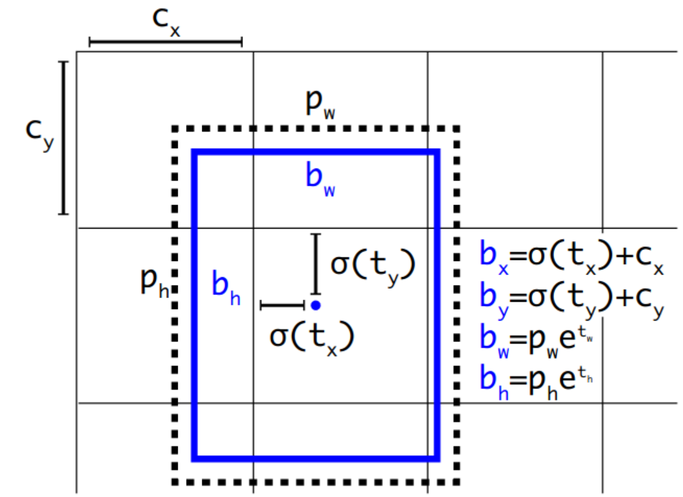
The first one-stage detector. Introduced in (Redmon, Divvala, Girshick, & Farhadi, 2016). It outputs:

- B bounding boxes $(x, y, w, h, conf)$ for each cell of a $S \times S$ grid
- the probabilities over the K classes
- the output volume is $(5 \times B + K) \times (S \times S)$ in YoloV1, then $(5 + K) \times B \times (S \times S)$ from v2



YoLo v0 with one bounding box per cell $B = 1$

Bounding box encoding:



From Yolo v2

In YoLo v3, the network is Feature Pyramid Network (FPN) like with a downsampling and an upsampling paths, with predictions at 3 stages.

YOU ONLY LOOK ONCE (YOLO V1,V2,V3) (2/2)

The loss is multi-task with :

- a term for the regression of the transformation of the anchor boxes
- a term for detecting the presence of an object
- a term for the (possibly multi) labelling of the boxes

$$\mathcal{L} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(t_x - t_x^*)^2 + (t_y - t_y^*)^2 + (t_w - t_w^*)^2 + (t_h - t_h^*)^2]$$

$$- \sum_{i=0}^{S^2} \sum_{j=0}^{B} BCE(\mathbb{1}_{ij}^{obj}, \text{has\_obj}_{ij})$$

$$- \sum_{i=0}^{S^2} \sum_{j=0}^{B} \sum_{k=0}^{K} BCE(\text{has\_class}_{ijk}, p_{ijk})$$

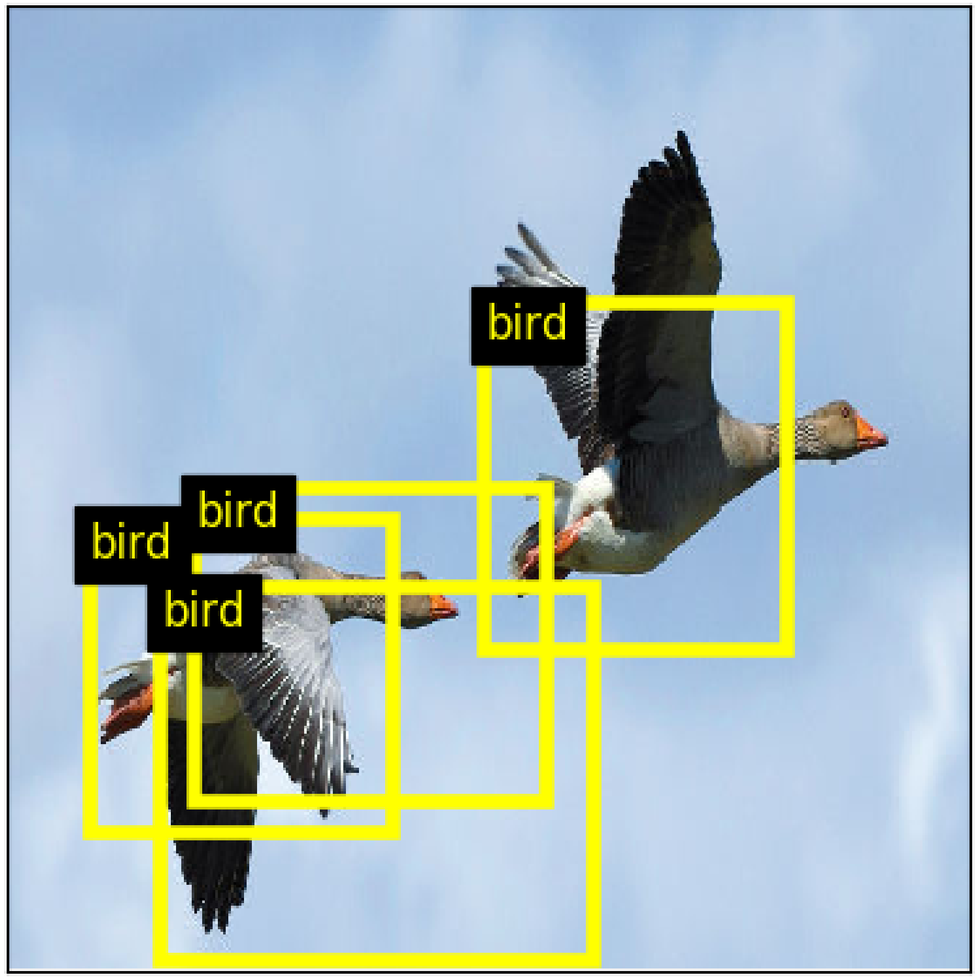
Darknet code

In v1 and v2, the prediction losses were L2 losses.

Multi labelling can occur in coco (e.g. women, person)

NON MAXIMUM SUPPRESSION (NMS)

The object detectors may output multiple overlapping bounding box for the same object



Multiple bounding boxes for the same object

NMS algorithm :

- order the bounding boxes by decreasing confidence
- for every rank, remove every bounding box, of lower rank, with IoU higher than a threshold

NMS may suppress one of two “overlapped” objects. It hard resets the scores of overlapping bboxes.

SoftNMS (Bodla, Singh, Chellappa, & Davis, 2017):

- order the bounding boxes by decreasing confidence
- for every rank, scale the confidence of the bounding boxes of lower rank bboxes (rather than setting to 0)

SEMANTIC/INSTANCE SEGMENTATION

PROBLEM STATEMENT

Given an image,

Semantic segmentation : predict the class of every single pixel. We also call dense prediction/dense labelling.

Example image from MS Coco

Image labeled with stuff classes

Instance segmentation : classify all the pixels belonging to the same countable objects

Example image from MS Coco

Image labeled with things classes

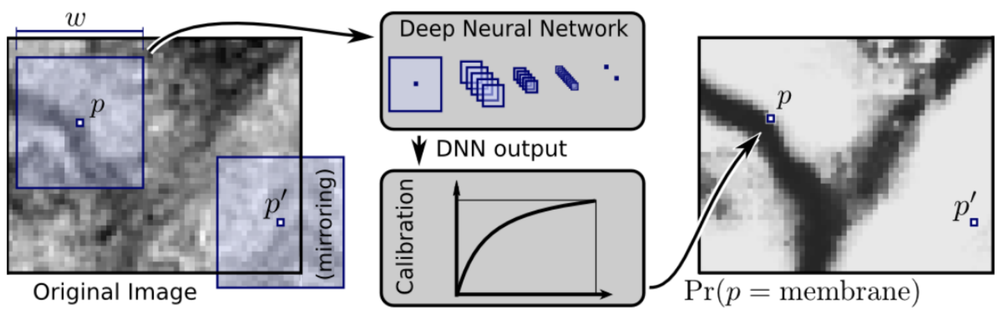
More recently, **panoptic segmentation** refers to instance segmentation for countable objects (e.g. people, animals, tools) and semantic segmentation for amorphous regions (grass, sky, road).

Metrics : see Coco panotpic evaluation

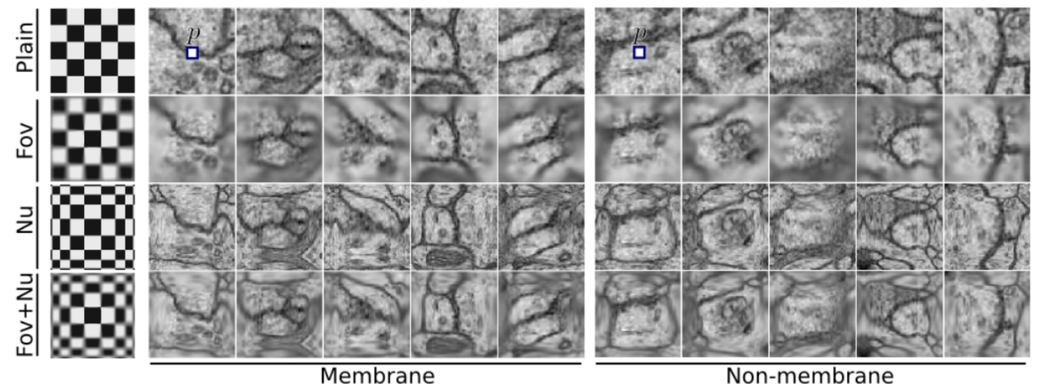
Some example networks : PSP-Net, U-Net, Dilated Net, ParseNet, DeepLab, Mask RCNN, …

SLIDING WINDOW

Introduced in (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012).



Segmentation with a sliding window pixel classifier



Class preserving data augmentation

- The predicted segmentation is post-processed with a median filter
- The output probability is calibrated to compensate for different priors on being a membrane / not being a membrane.

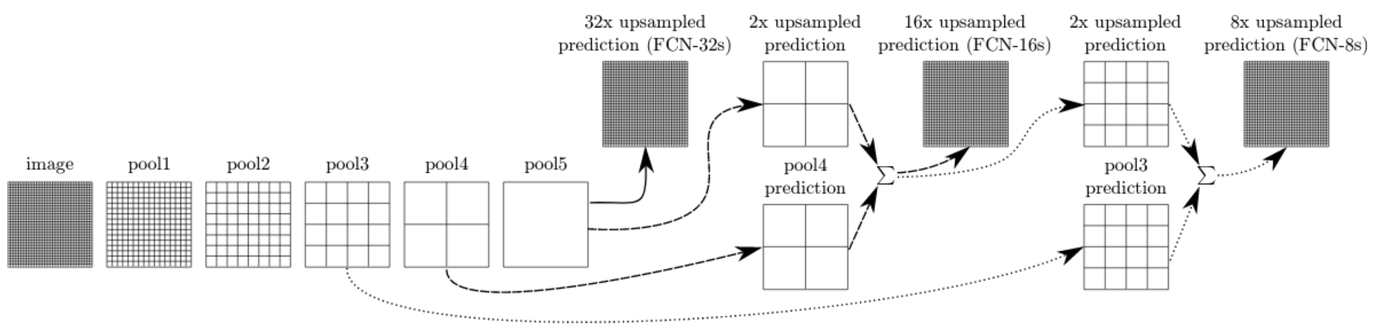
Drawbacks:

- one forward pass per patch
- overlapping patches do not share the computations

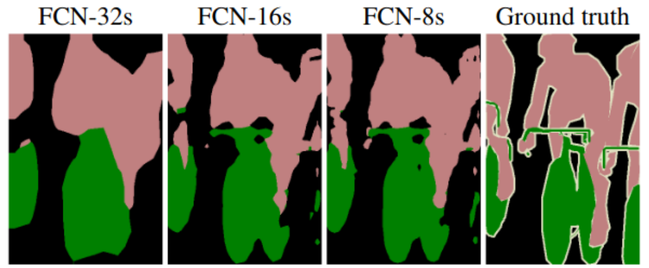
(on deep neural network calibration, see also (Guo, Pleiss, Sun, & Weinberger, 2017))

FCN

Introduced in (Long, Shelhamer, & Darrell, 2015). First end-to-end convolutional network for dense labeling with pretrained networks.



FCN network architecture with a backbone downsampling with pool5 size H/32,W/32 because the pool5 stride is 32. The segmentation prediction from each pool layer is computed with a Conv 1×1.



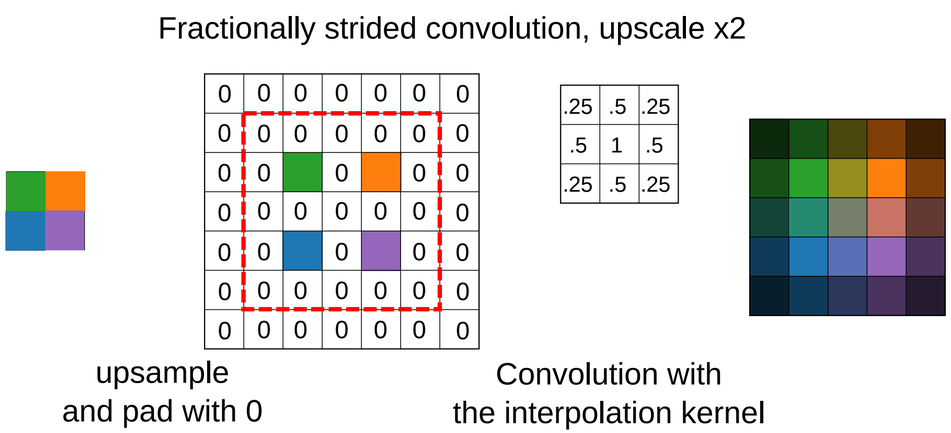
Refined segmentation by incoporating early feature maps

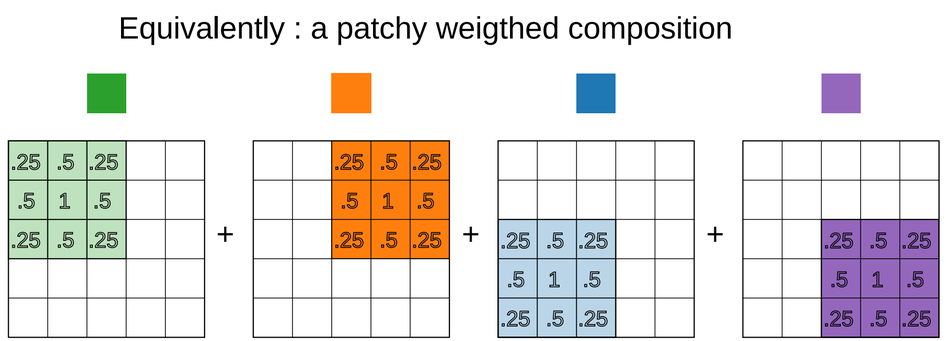
The upsampling is performed with a fractionally strided convolution (deconvolution), i.e. the upsampling is learned.

LEARNED UPSAMPLING : FRACTIONALLY STRIDED CONVOLUTION

Traditional approaches involves bilinear, bicubic, etc… interpolation.

For upsampling in a learnable way, we can use fractionally strided convolution. That's one ingredient behind Super-Resolution (Shi, Caballero, Huszár, et al., 2016).





Learned upsampling with fractionally strided convolution

You can initialize the upsampling kernels with a bilinear interpolation kernel. To have some other equivalences, see (Shi, Caballero, Theis, et al., 2016).

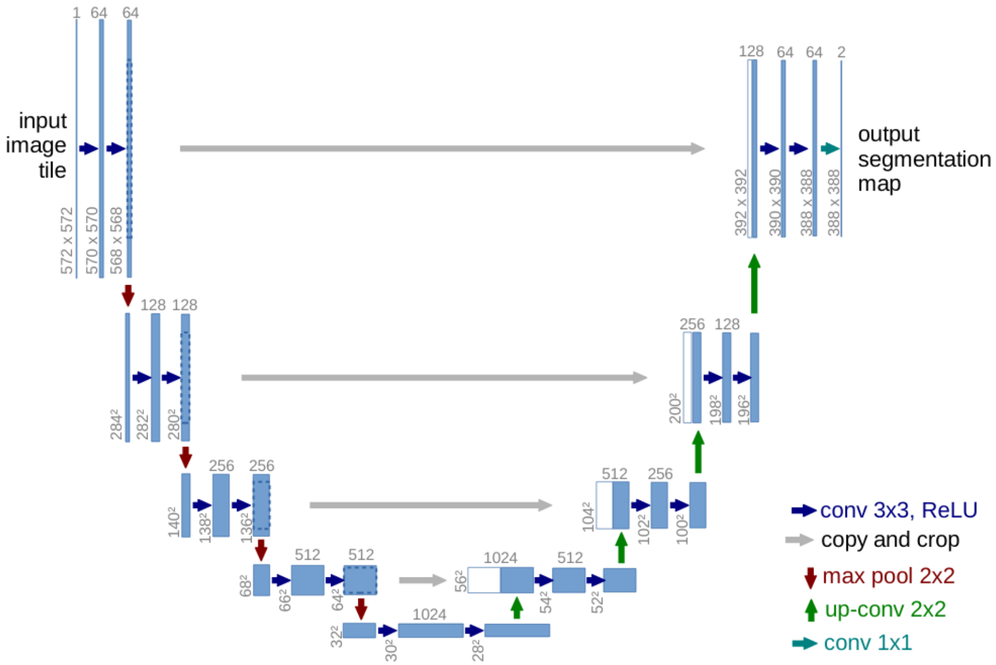
This can introduce artifacts, check (Odena, Dumoulin, & Olah, 2016)

ENCODER/DECODER ARCHITECTURES

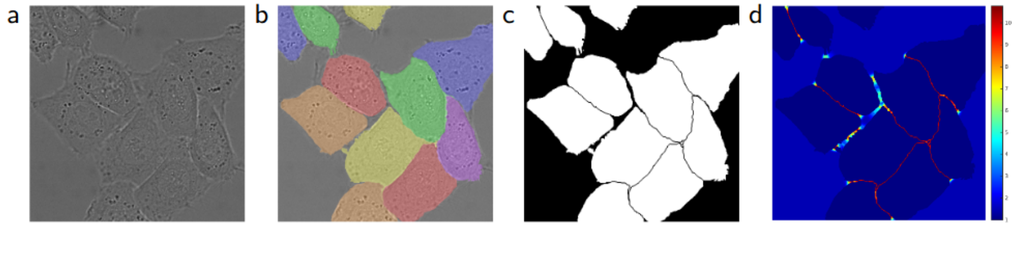
Several models along the same architectures : U-Net, SegNet. Encoder-Decoder architecture introduced in (Ronneberger, Fischer, & Brox, 2015)

There is :

- a contraction path from the image “down” to high level features at a coarse spatial resolution
- an expansion upsampling path which integrates low level features for pixel wise labelling



U-Net contraction/expansion convolution/deconvolution architecture



a) Input b) ground truth c) U-Net segmentation d) border weight cost

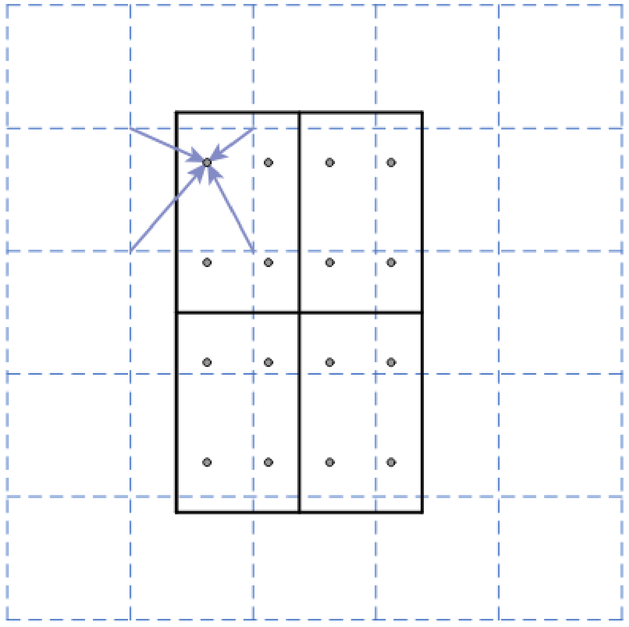
Data augmentation : rotation, “color” variations, elastic deformations.

MASK RCNN

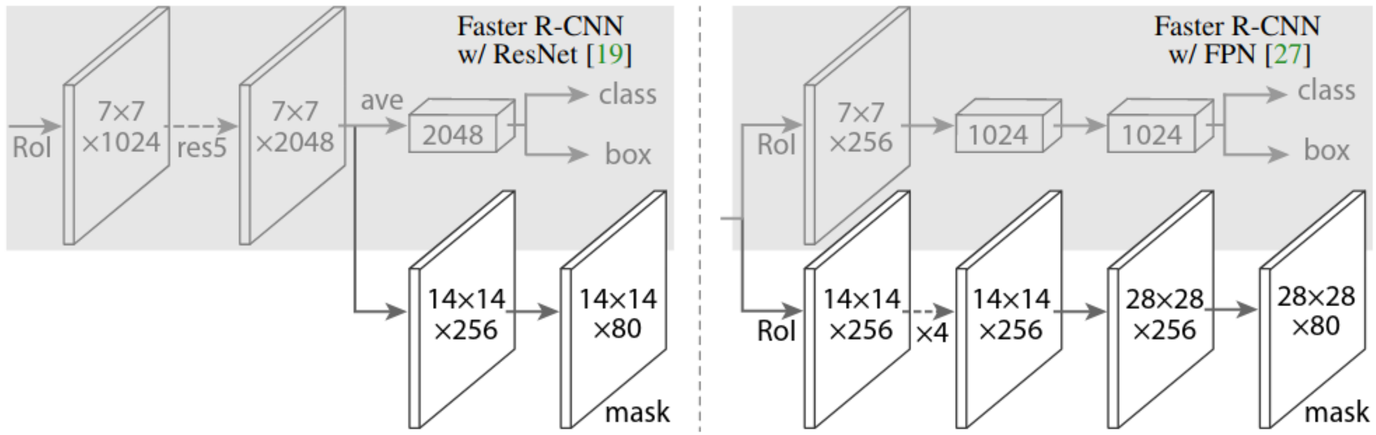
Introduced in (He, Gkioxari, Dollár, & Girshick, 2018) as an extension of Faster RCNN. It outputs a **binary mask** in addition the class labels + bbox regressions.

It addresses **instance segmentation** by predicting a mask for individualised object proposals.

Proposed to use ROI-Align (with bilinear interpolation) rather than ROI-Pool.



ROI Align prevents quantization of ROI Pooling



Mask for the 80 object categories of COCO

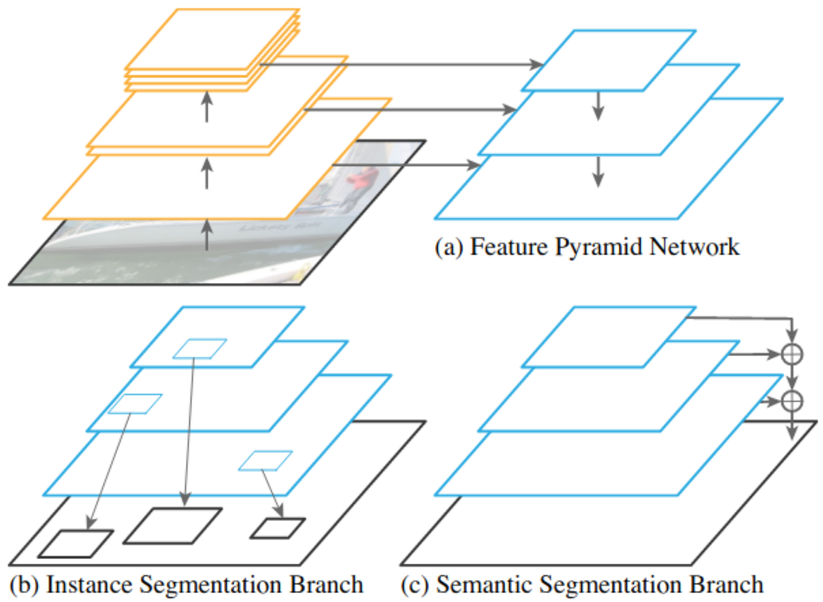
There is no competition between the classes in the masks. Different objects may use different kernels to compute their masks.

Can be extended to keypoint detection, outputting a K depth mask for predicting the K joints.

PANOPTIC FPN

Introduced in (Kirillov, Girshick, He, & Dollár, 2019), unifies instance segmentation (countable) and semantic segmentation (amorphous) in a single network with two heads on top of a FPN backbone :

- a region based Mask RCNN for instance segmentation
- an upscaled pixel-wise softmax layer for semantic segmentation



A shared FPN backbone with an instance segmentation head and a semantic segmentation branch

Careful design of :

- the multi-task loss : weighting the normalized instance losses (classification, bbox regression, mask) and the segmentation loss (pixel-wise cross-entropy)
- minibatch building, learning rate schedules
- data augmentation : color augmentation (Howard, 2013) and crop boostrapping

To give a try, check detectron2

CONCLUSION

OTHER PROBLEMS WE DID NOT DISCUSS

- Super resolution :
  - Learn to upscale an image
  - example networks : SRCNN, FSRCNN, VDSR, ESPCN, RED-Net
- Graph convolutions :
  - Learn to aggregate features on graph which are **structures of arbitrary sizes and arbitrary branching factor**
  - example networks : Weisfeilher lehman, see pytorch geometric, AlphaChem, ..
- Processing 3D point clouds :
  - unstructured sparse irregular 3D data
  - example networks : PointNet

REFERENCES

BIBLIOGRAPHY

Rather check the full online document references.pdf

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS – Improving Object Detection With One Line of Code. *arXiv:1704.04503 [Cs]*. Retrieved from http://arxiv.org/abs/1704.04503

Ciresan, D., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2012). Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images, 9.

Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). *Visualizing higher-layer features of a deep network* (No. 1341). University of Montreal.

Girshick, R. (2015). Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1440–1448). https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv:1311.2524 [Cs]*. Retrieved from http://arxiv.org/abs/1311.2524

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. *arXiv:1706.04599 [Cs]*. Retrieved from http://arxiv.org/abs/1706.04599

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2018). Mask R-CNN. *arXiv:1703.06870 [Cs]*. Retrieved from http://arxiv.org/abs/1703.06870

Howard, A. G. (2013). Some Improvements on Deep Convolutional Neural Network Based Image Classification, 6.

Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019). Panoptic Feature Pyramid Networks. *arXiv:1901.02446 [Cs]*. Retrieved from http://arxiv.org/abs/1901.02446

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 936–944). Honolulu, HI: IEEE. https://doi.org/10.1109/CVPR.2017.106

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. *arXiv:1411.4038 [Cs]*. Retrieved from http://arxiv.org/abs/1411.4038

Odena, A., Dumoulin, V., & Olah, C. (2016). Deconvolution and Checkerboard Artifacts. *Distill*, *1*(10), e3. https://doi.org/10.23915/distill.00003

Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*. https://doi.org/10.23915/distill.00007

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *arXiv:1506.02640 [Cs]*. Retrieved from http://arxiv.org/abs/1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [Cs]*. Retrieved from http://arxiv.org/abs/1506.01497

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [Cs]*. Retrieved from http://arxiv.org/abs/1505.04597

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., … Wang, Z. (2016). Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *arXiv:1609.05158 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1609.05158

Shi, W., Caballero, J., Theis, L., Huszar, F., Aitken, A., Tejani, A., … Wang, Z. (2016). Is the deconvolution layer the same as a convolutional layer?, 7.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034 [Cs]*. Retrieved from http://arxiv.org/abs/1312.6034

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2015). Striving for Simplicity: The All Convolutional Net. *arXiv:1412.6806 [Cs]*. Retrieved from http://arxiv.org/abs/1412.6806