object detection and swift

brettkoonce.com/talks

february 23rd, 2019

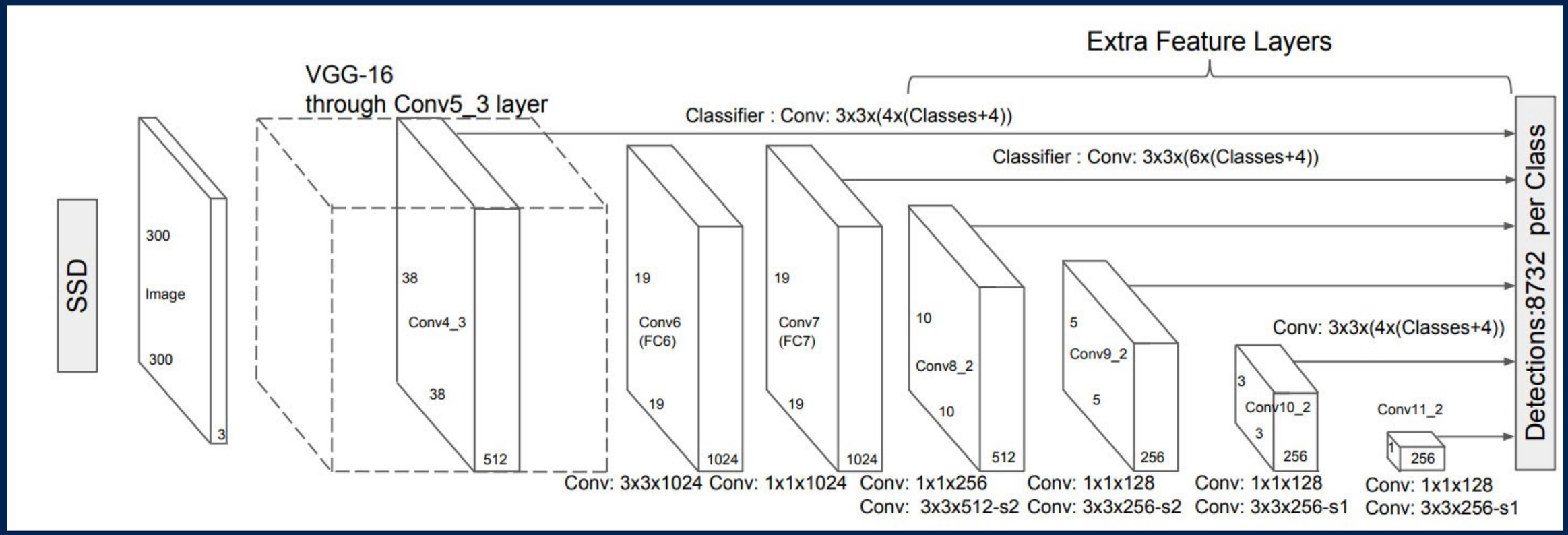
overview

- object detection
- vgg + ssd
- resnet + ssd + retinanet (fast.ai)
- mobilenets + ssdlite (tensorflow lite)
- yolo v1, v2, demo (turicreate), v3

vgg + ssd

- image recognition network backbone
- ssd head to combine bounding boxes
- loss function to train against

ssd


Extra Feature Layers
VGG-16
through Conv5_3 layer
Classifier : Conv: 3x3x(4x(Classes+4))
Classifier : Conv: 3x3x(6x(Classes+4))
Conv: 3x3x(4x(Classes+4))
SSD
300
Image
300
3
38
Conv4_3
38
512
19
Conv6
(FC6)
19
1024
19
Conv7
(FC7)
19
1024
10
Conv8_2
10
512
5
Conv9_2
5
256
3
Conv10_2
3
256
Conv11_2
1
256
Detections:8732 per Class
Conv: 3x3x1024
Conv: 1x1x1024
Conv: 1x1x256
Conv: 3x3x512-s2
Conv: 1x1x128
Conv: 3x3x256-s2
Conv: 1x1x128
Conv: 3x3x256-s1
Conv: 1x1x128
Conv: 3x3x256-s1

resnet + ssd + retinanet

- use resnet backbone —> slightly lower quality, but much much faster
- keep ssd head
- add focal loss function
- fast.ai demo, sota 2018

retinanet

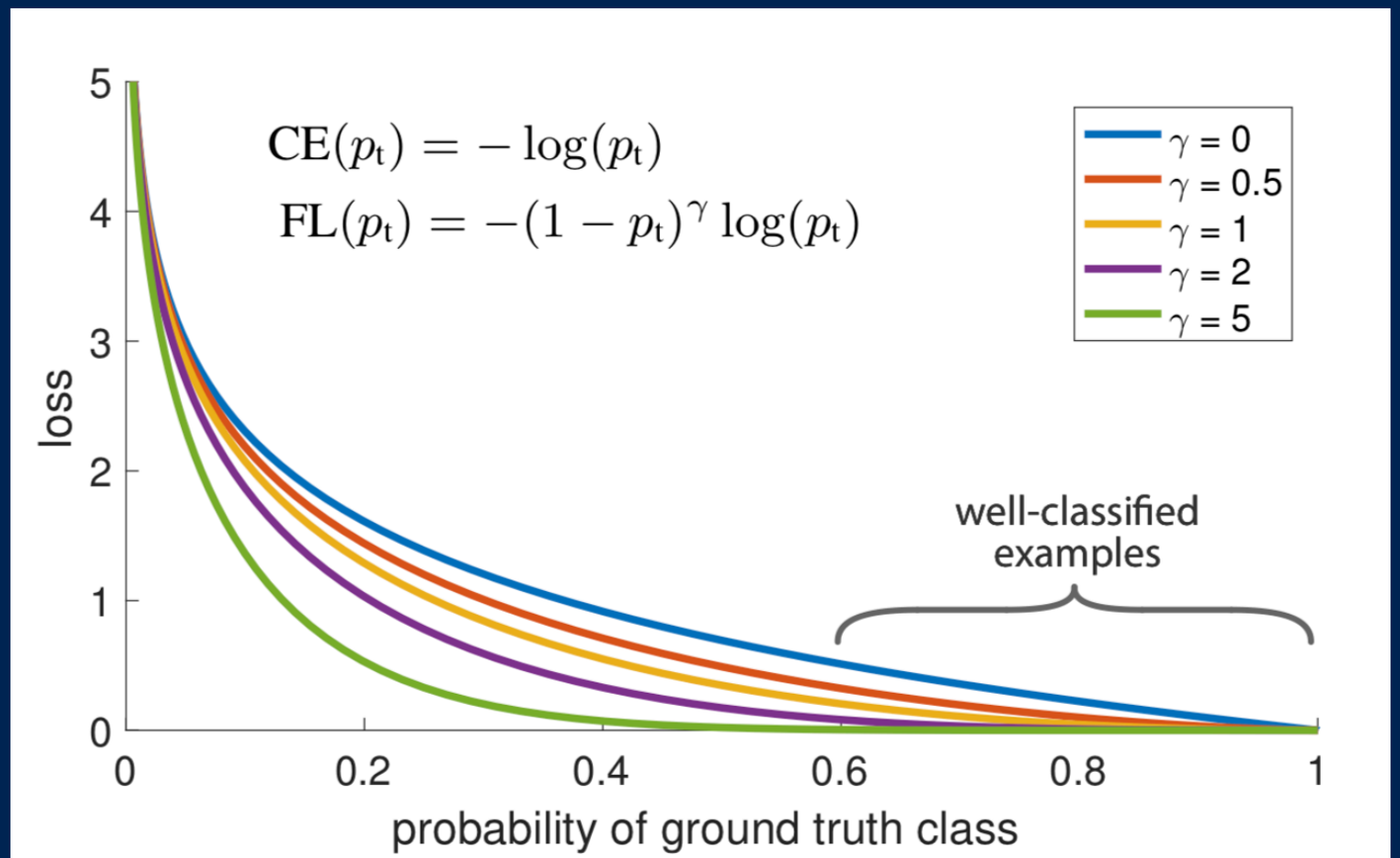


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^{\gamma}$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

mobilenets + ssdlite

- use mobilenets backbone, even faster network
- use simpler ssd head (1x1 conv, faster)
- drop focal loss function
- tensorflow lite demo

ssdlite

	Params	MAdds
SSD[34]	14.8M	1.25B
SSDLite	**2.1M**	**0.35B**

Table 5: Comparison of the size and the computational cost between SSD and SSDLite configured with MobileNetV2 and making predictions for 80 classes.

Network	mAP	Params	MAdd	CPU
SSD300[34]	23.2	36.1M	35.2B	-
SSD512[34]	26.8	36.1M	99.5B	-
YOLOv2[35]	21.6	50.7M	17.5B	-
MNet V1 + SSDLite	22.2	5.1M	1.3B	270ms
MNet V2 + SSDLite	22.1	**4.3M**	**0.8B**	200ms

yolo

- key idea: combine head + image recognition network to do everything (one-shot detection)
- lose some high end quality, gain a lot of speed
- turicreate demo

yolo

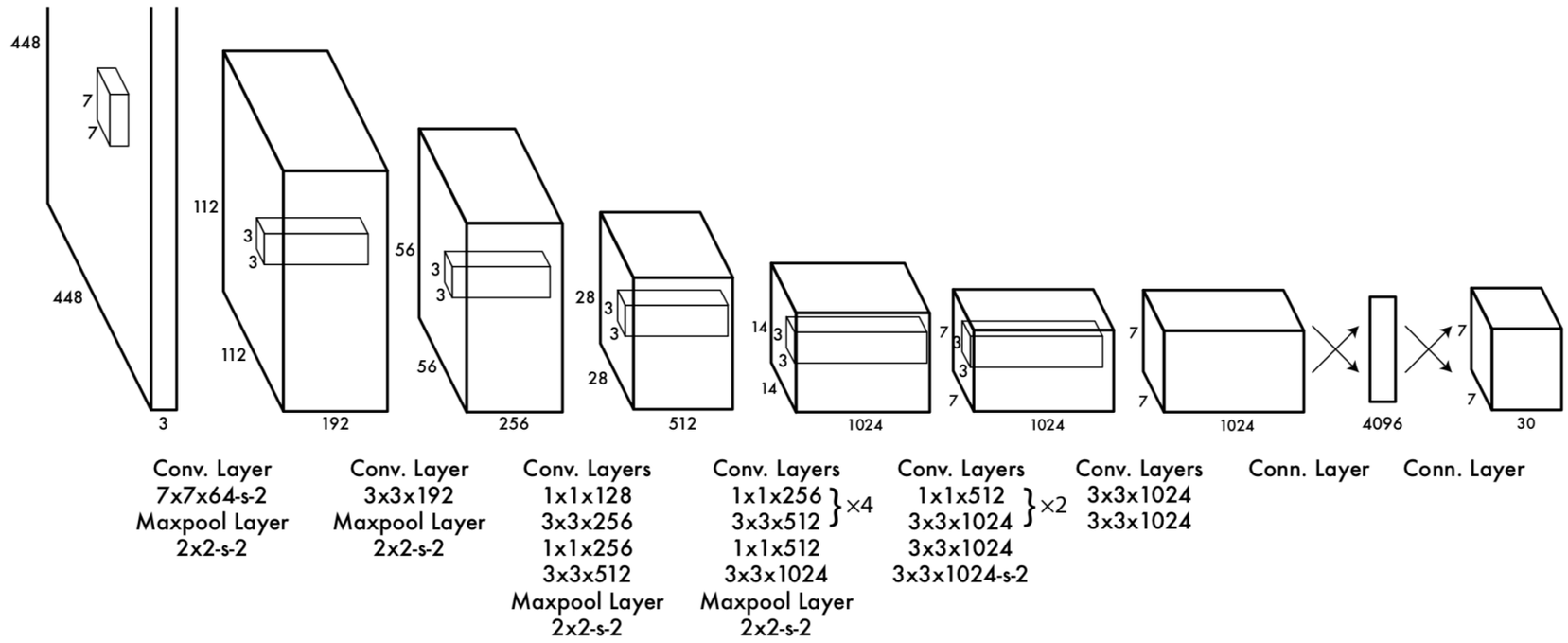


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

turicreate demo

- add custom data, bounding boxes
- run script, transfer learning on yolo network
- coreml model, deploy on device

next steps

- r-cnn, fast-rcnn, faster-rcnn
- feature pyramid networks
- yolo 3 (+mobilenets)
- cornernet