

object detection and swift

brettkoonce.com/talks

february 23rd, 2019

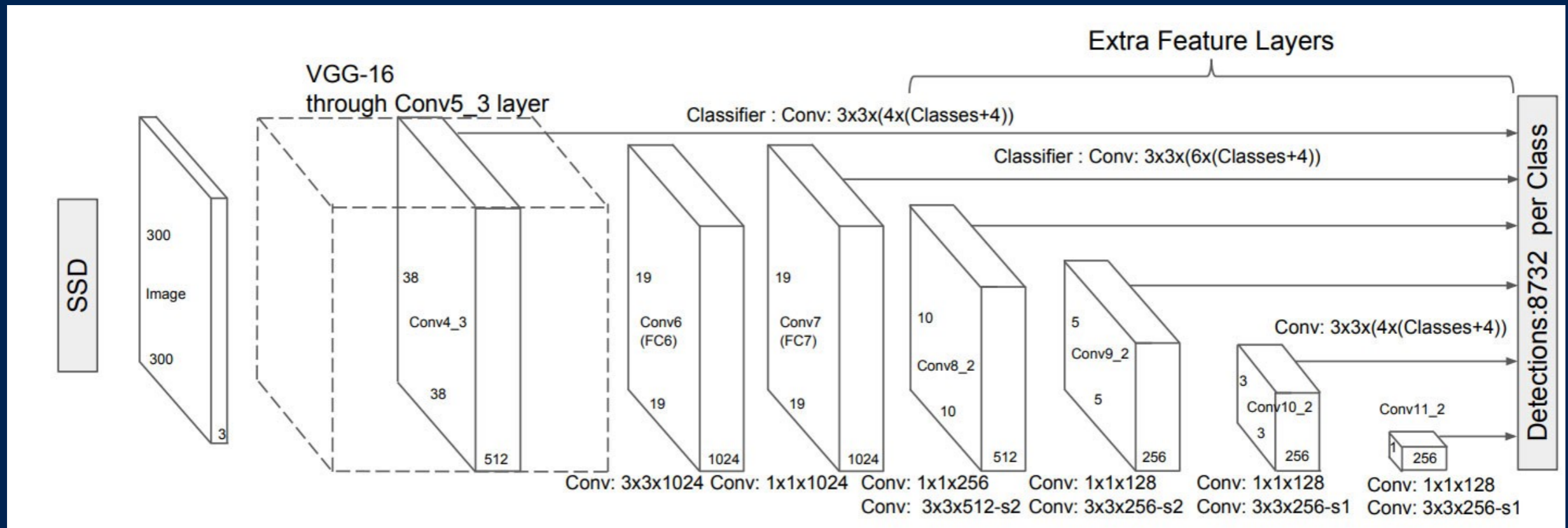
overview

- **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



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

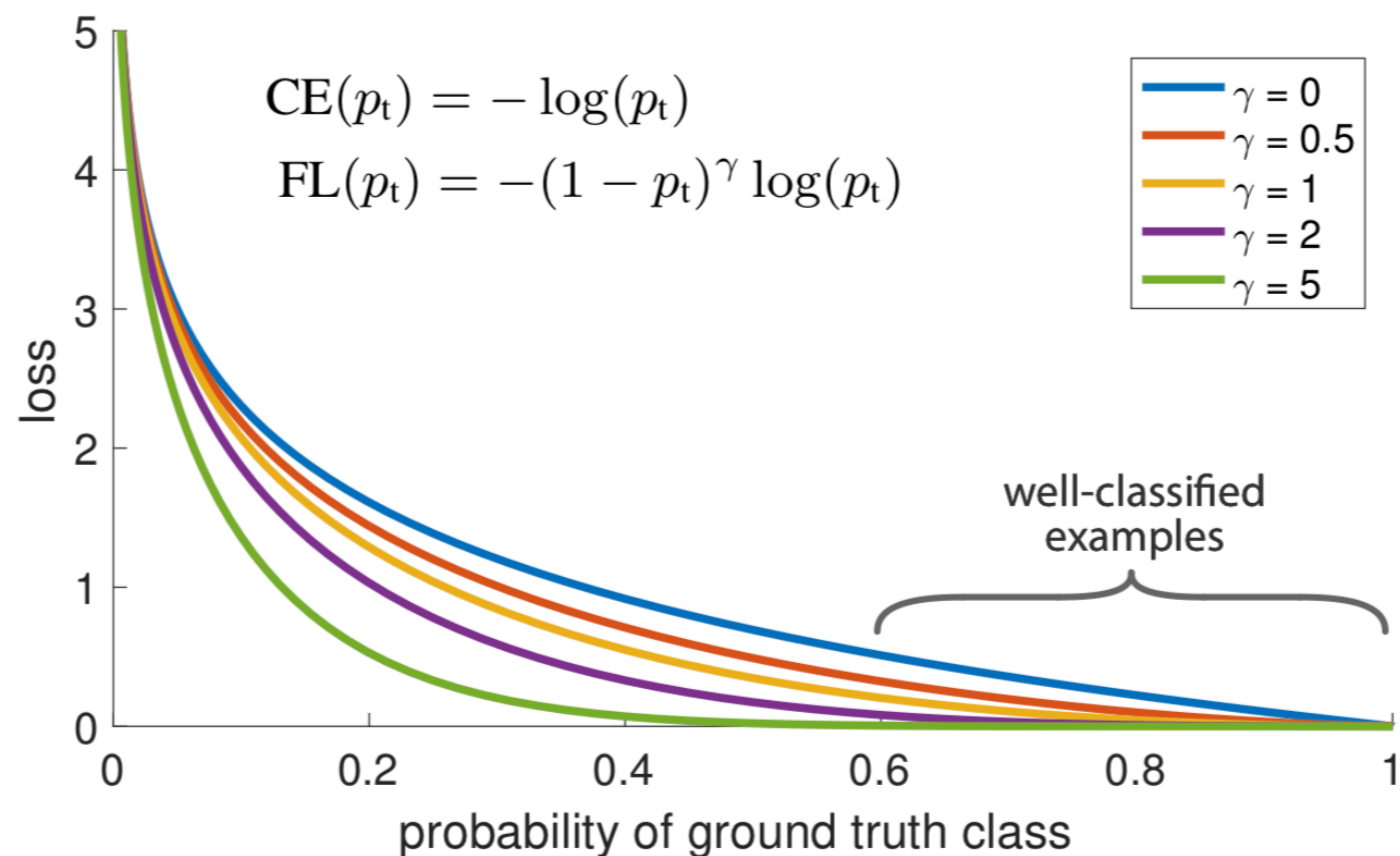


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will 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**

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**