深度学习实战

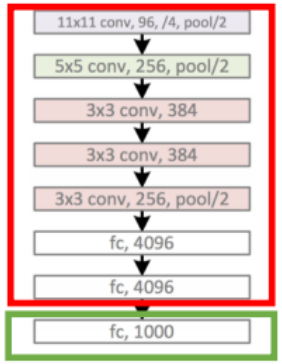
内容大纲

- 图像分类CNN网络
- 可视化工具和技术
- 迁移学习
- fastai
- 目标检测
- 数据增广
- 数据标注

图像分类CNN网络

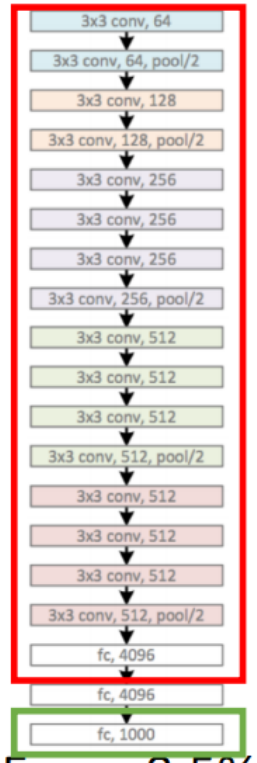
Deep ConvNet for Visual Recognition

2012: AlexNet
5 conv. layers



Error: 15.3%

2014: VGG
16 conv. layers

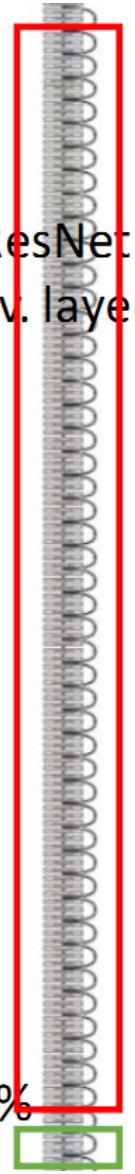


Error: 8.5%

2015: GoogLeNet
22 conv. layers

Error: 7.8%

2016: ResNet
>100 conv. layers

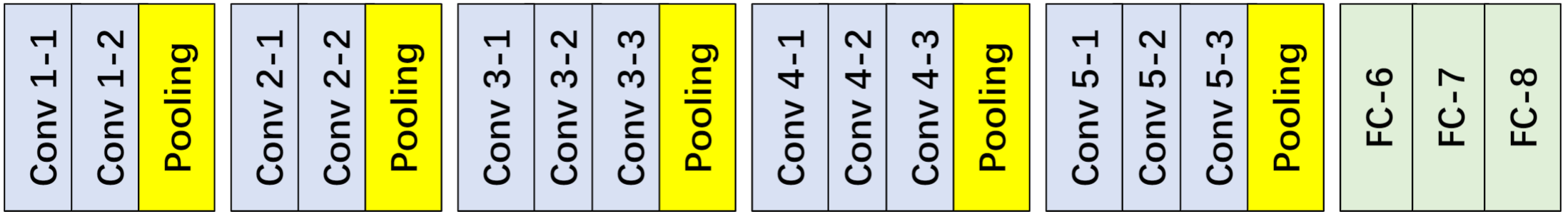


Error: 4.4%

图像分类CNN网络

- VGG Recap
  - 结构非常简洁，整个网络都使用了同样大小的卷积核尺寸（3x3）和最大池化尺寸（2x2）
  - 多个小滤波器（3x3）卷积层的组合比一个大滤波器（5x5或7x7）卷积层好
  - 验证了通过不断加深网络结构可以提升性能

VGG 16



图像分类CNN网络

多种不同的配置
结构类似

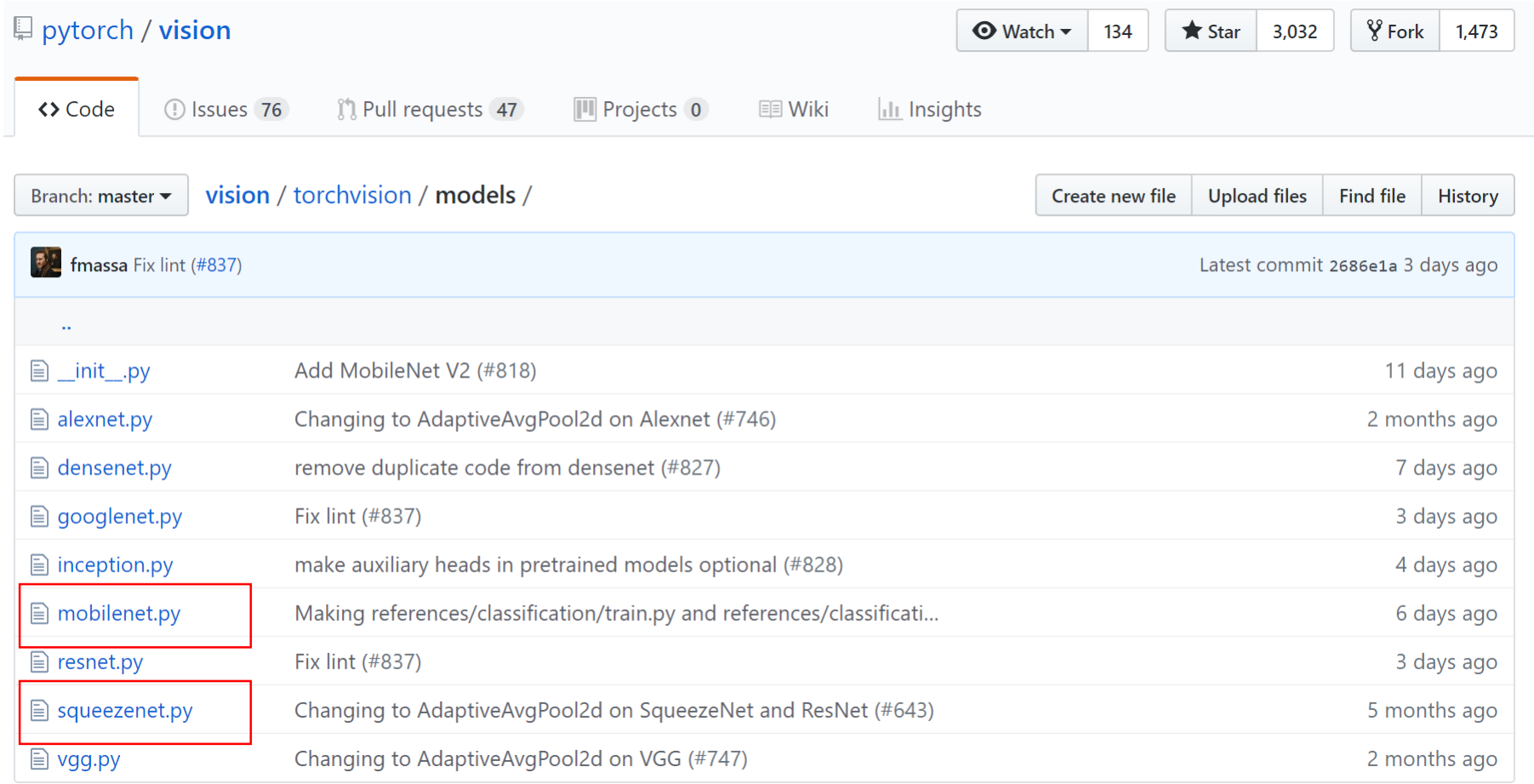
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

class VGG16(nn.Module)
class VGG19(nn.Module)
单独实现？

100多层的
ResNet怎么办？

图像分类CNN网络

- And more in torchvision.models



轻量化卷积网络

更多实现网络和预训练模型 https://github.com/Cadene/pretrained-models.pytorch

可视化工具

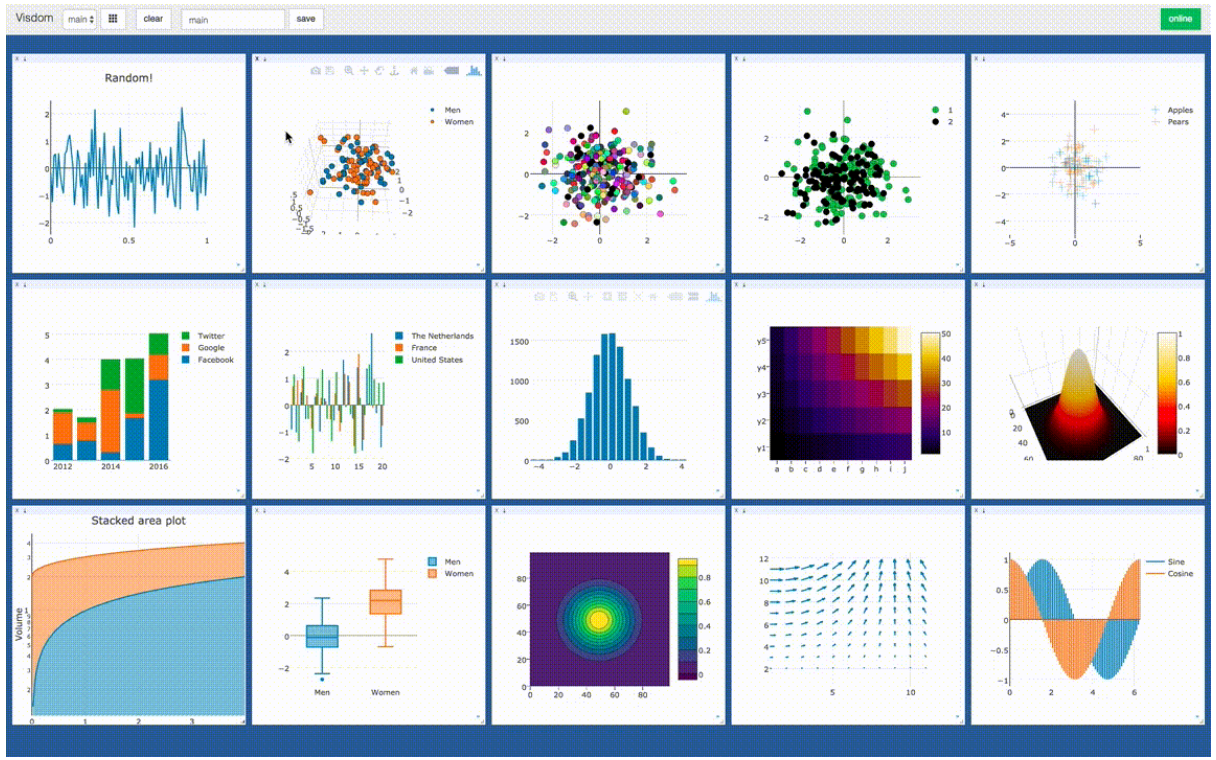
- torchsummaryX 简单可视化网络结构与参数量
- Matplotlib 本地可视化图表
- Visdom
- Tensorboard

远程查看

更便捷的可视化

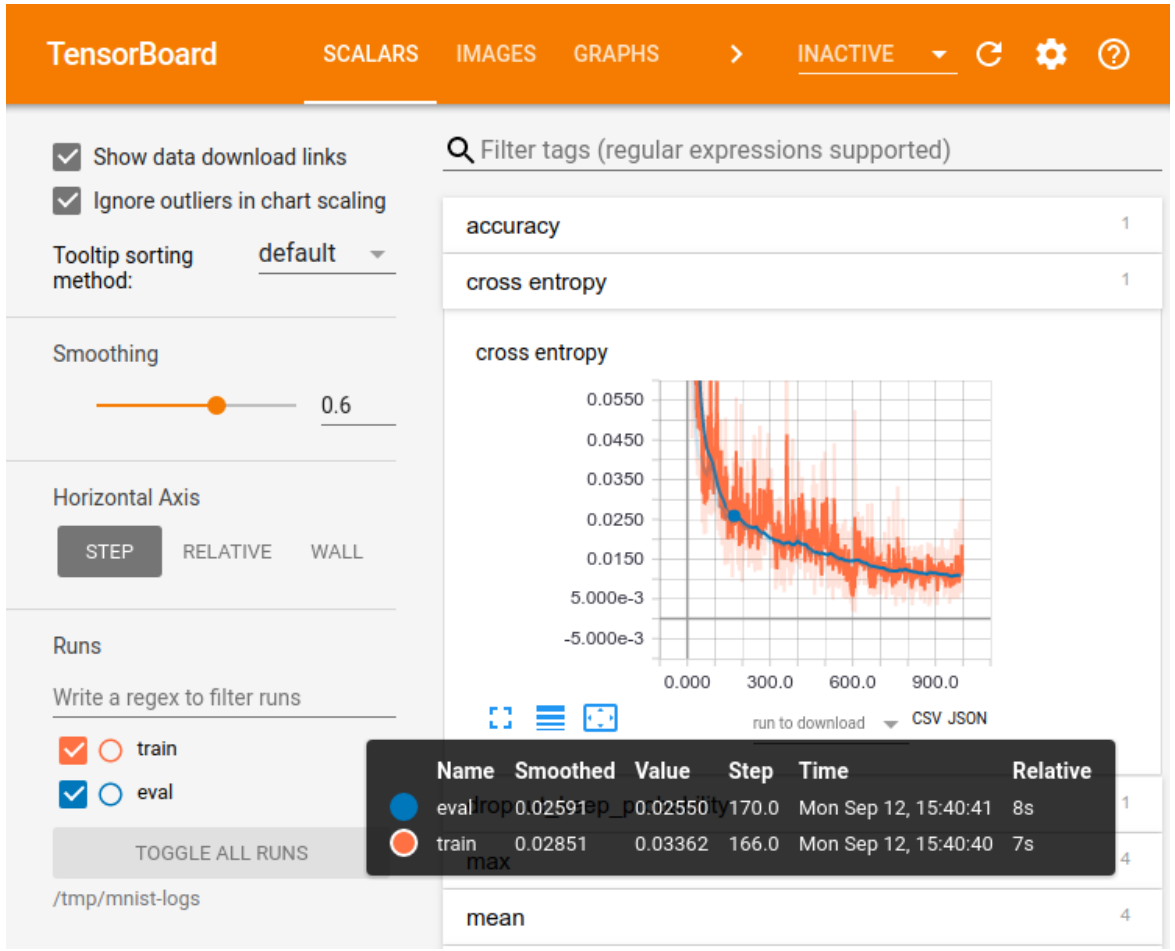
更好的交互

Visdom



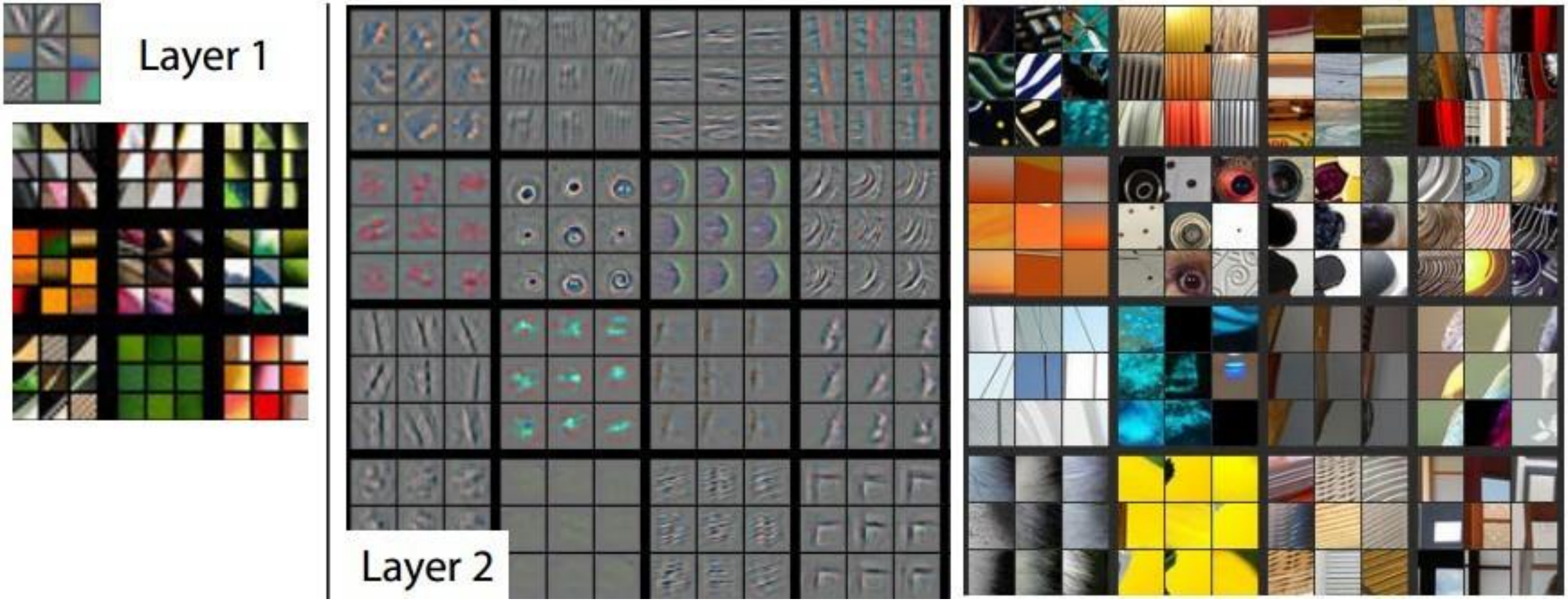
- Facebook开发的一个灵活的可视化工具，用于创建，组织和共享实时丰富数据的可视化。旨在促进（远程）数据的可视化，重点是支持科学实验
- 直接支持numpy和PyTorch
- 安装：
  - pip install visdom
  - conda install -c conda-forge visdom
- 使用：
  - python -m visdom.server

TensorBoard



- TensorBoard是Google集成在TensorFlow中的一个可视化工具，能够有效地展示Tensorflow在运行过程中的计算图、各种指标随着时间的变化趋势以及训练中使用到的数据信息
- PyTorch通过tensorboardX使用
- 安装：
  - pip install tenorflow
  - pip install tensorboardX
- 使用：
  - tensorboard –logdir=“./logs”

卷积可视化技术



HOW???

卷积可视化技术

utkuozbulak / **pytorch-cnn-visualizations**

Watch 49 | Unstar 1,973 | Fork 419

Convolutional Neural Network Visualizations

Note: I removed cv2 dependencies and moved the repository towards PIL. A few things might be broken (although I tested all methods), I would appreciate if you could create an issue if something does not work.

Implemented Techniques

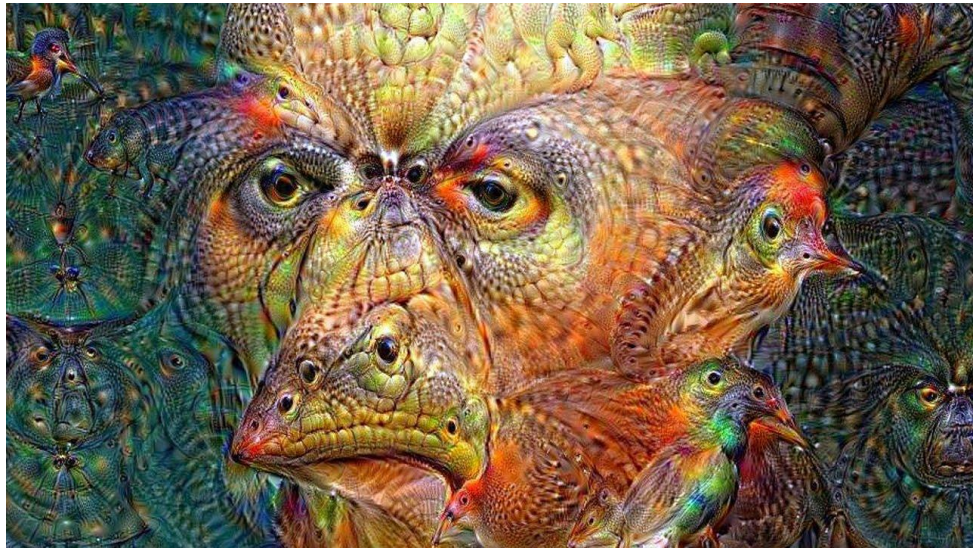
This repo contains following CNN visualization techniques implemented in Pytorch:

- Gradient visualization with vanilla backpropagation
- Gradient visualization with guided backpropagation [1]
- Gradient visualization with saliency maps [4]
- Gradient-weighted [3] class activation mapping [2]
- Guided, gradient-weighted class activation mapping [3]
- Smooth grad [8]
- CNN filter visualization [9]
- Inverted image representations [5]
- Deep dream [10]
- Class specific image generation [4]

https://distill.pub/2017/feature-visualization/

http://cs231n.github.io/understanding-cnn/

Deep Dream

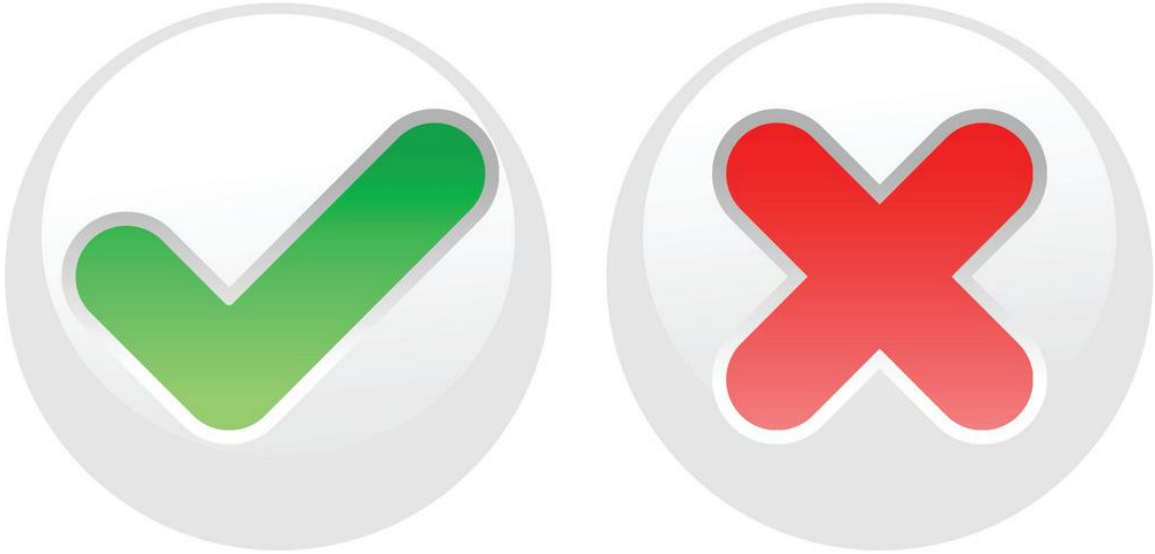


迁移学习

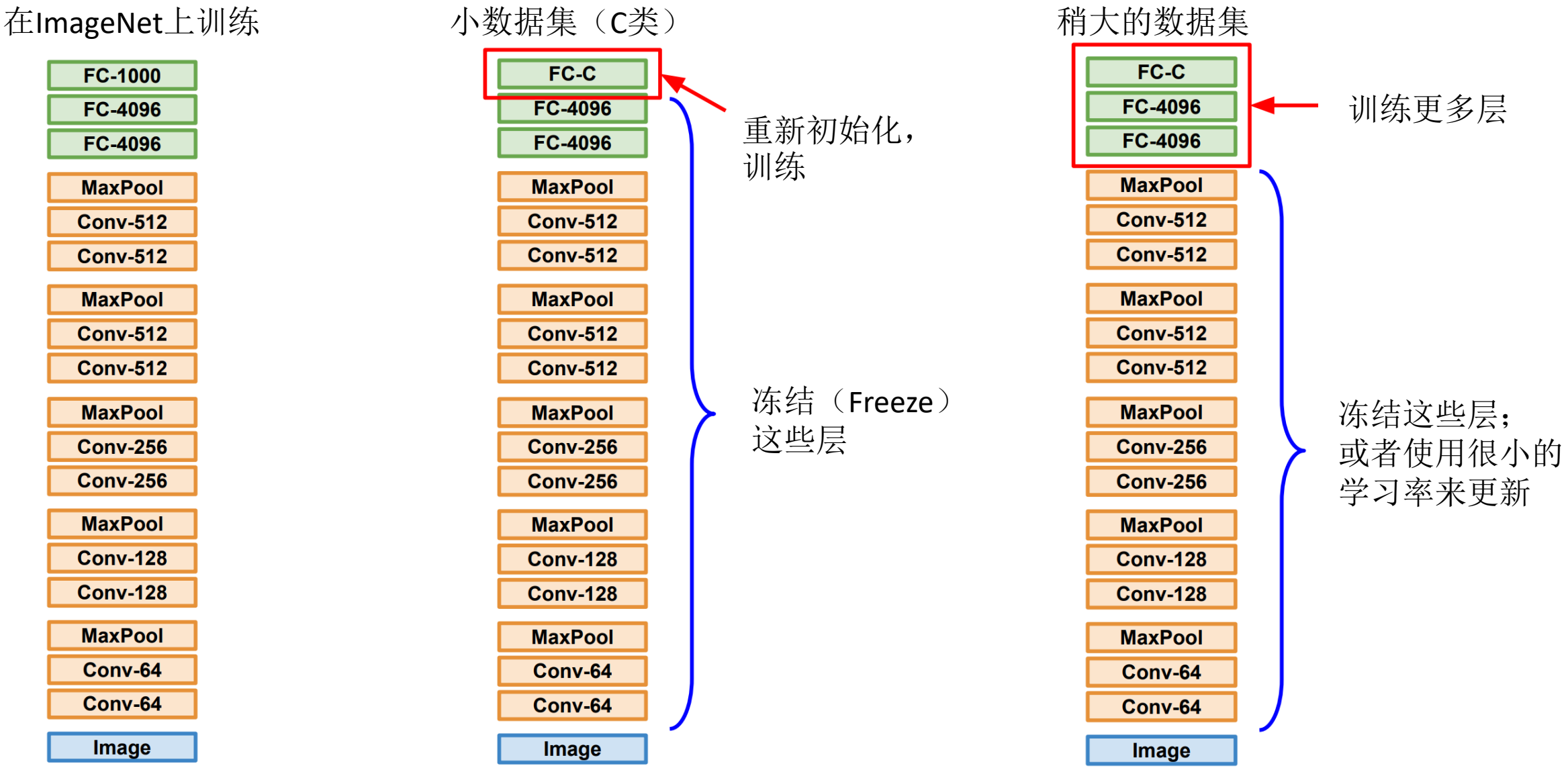

Our vision was that Big Data would change the way machine learning works. Data drives learning.
Fei-Fei Li
量子位
IMAGENET

迁移学习

- 深度学习需要非常大量的数据？



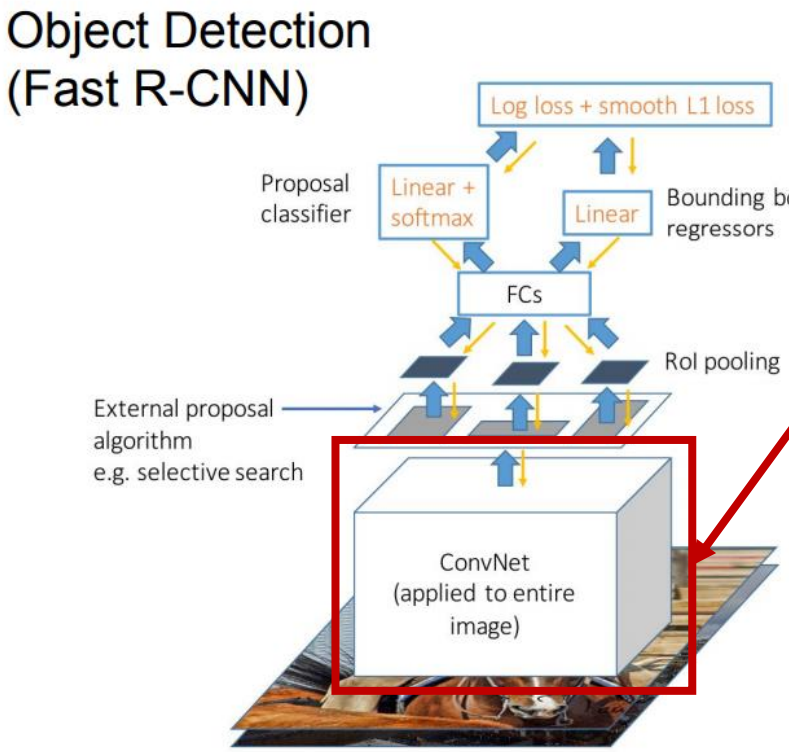
迁移学习

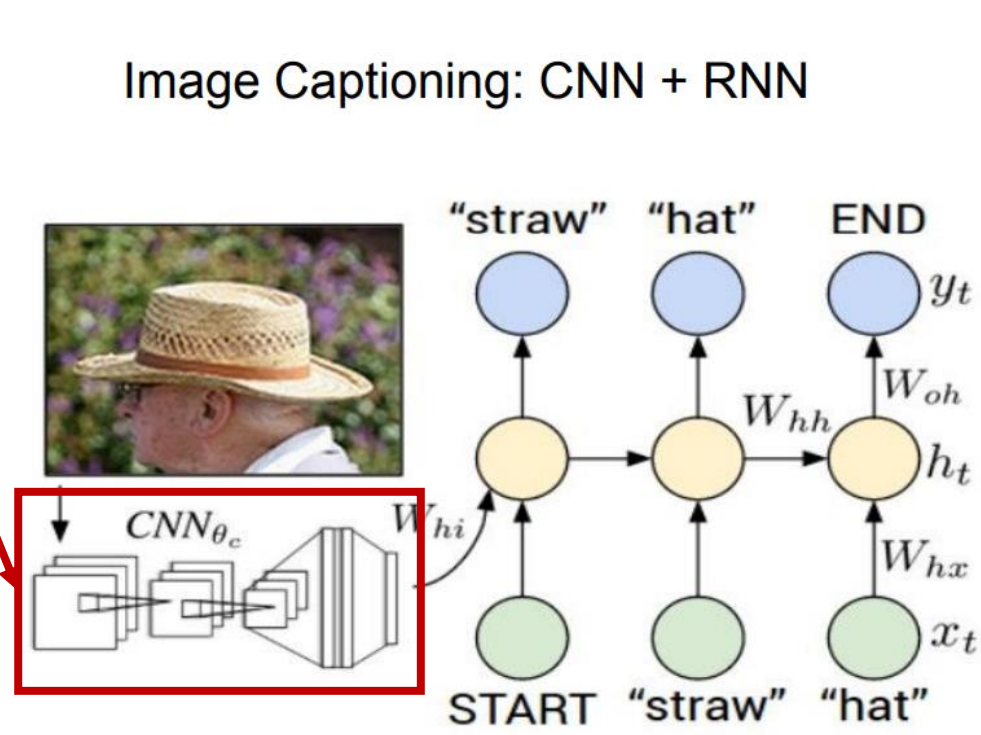

在ImageNet上训练
FC-1000
FC-4096
FC-4096
MaxPool
Conv-512
Conv-512
MaxPool
Conv-512
Conv-512
MaxPool
Conv-256
Conv-256
MaxPool
Conv-128
Conv-128
MaxPool
Conv-64
Conv-64
Image
小数据集（C类）
FC-C
FC-4096
FC-4096
MaxPool
Conv-512
Conv-512
MaxPool
Conv-512
Conv-512
MaxPool
Conv-256
Conv-256
MaxPool
Conv-128
Conv-128
MaxPool
Conv-64
Conv-64
Image
重新初始化，
训练
冻结（Freeze）
这些层
稍大的数据集
FC-C
FC-4096
FC-4096
MaxPool
Conv-512
Conv-512
MaxPool
Conv-512
Conv-512
MaxPool
Conv-256
Conv-256
MaxPool
Conv-128
Conv-128
MaxPool
Conv-64
Conv-64
Image
训练更多层
冻结这些层；
或者使用很小的
学习率来更新

迁移学习

- 迁移学习已成为“标准”

在ImageNet上预训练的CNN

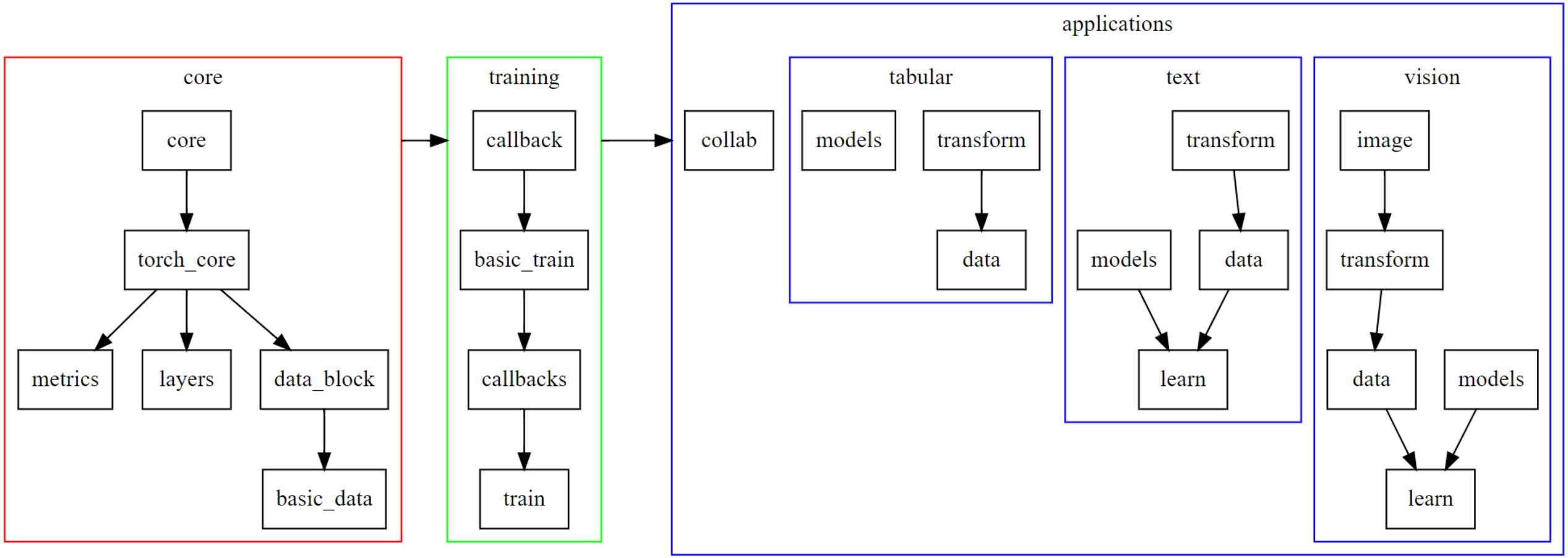




使用fastai

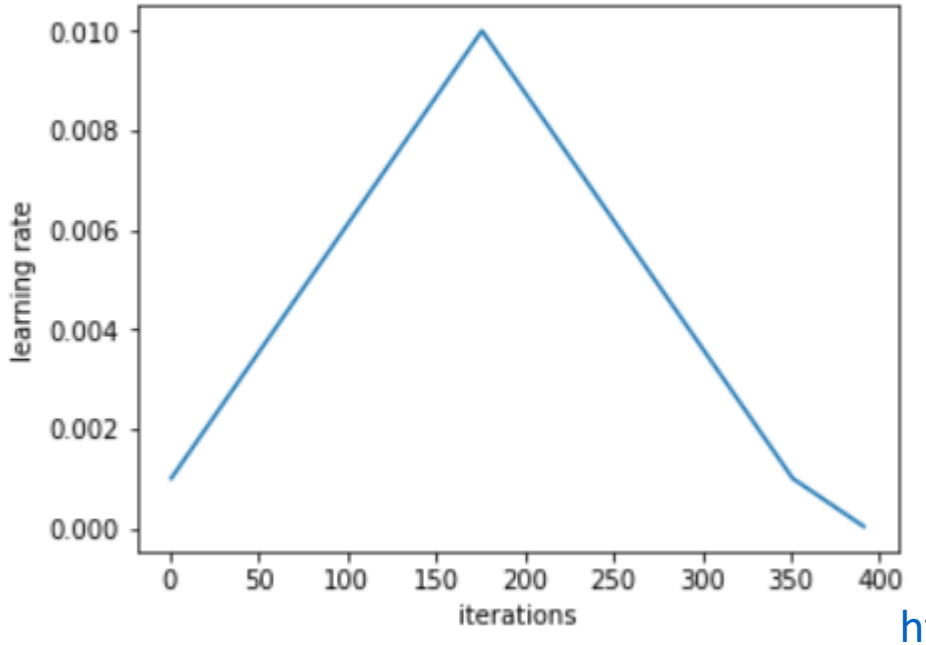
- fastai不是简单意义上的将Pytorch封装重构
- 同时集成了经过研究有效的多种训练技巧
- 可用与研究的快速原型开发，也能用于生产环境部署
- 是目前把易用性和功能都做到了极致的深度学习工具/框架。
- fastai使用了大量的python技巧并且高度紧凑、高度可扩还有良好的编码风格
- 更多：https://docs.fast.ai/

使用fastai


core
core
torch_core
metrics
layers
data_block
basic_data
training
callback
basic_train
callbacks
train
applications
collab
tabular
models
transform
data
text
transform
models
data
learn
vision
image
transform
data
models
learn

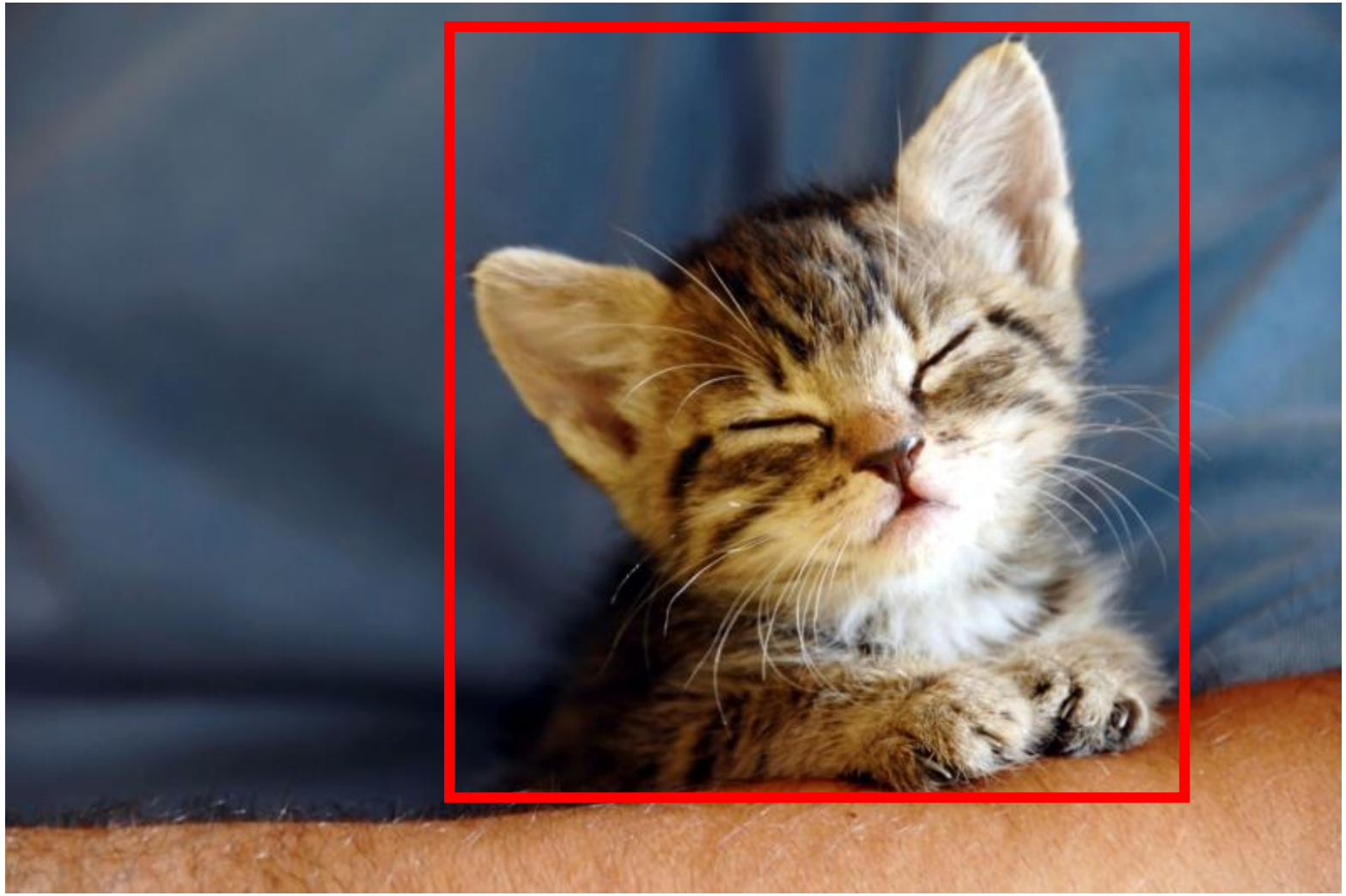
1cycle policy

- To cycle the learning rate between lower bound and upper bound during complete run
- Cycle is number of iterations where we go from lower bound learning rate to higher bound and back to lower bound

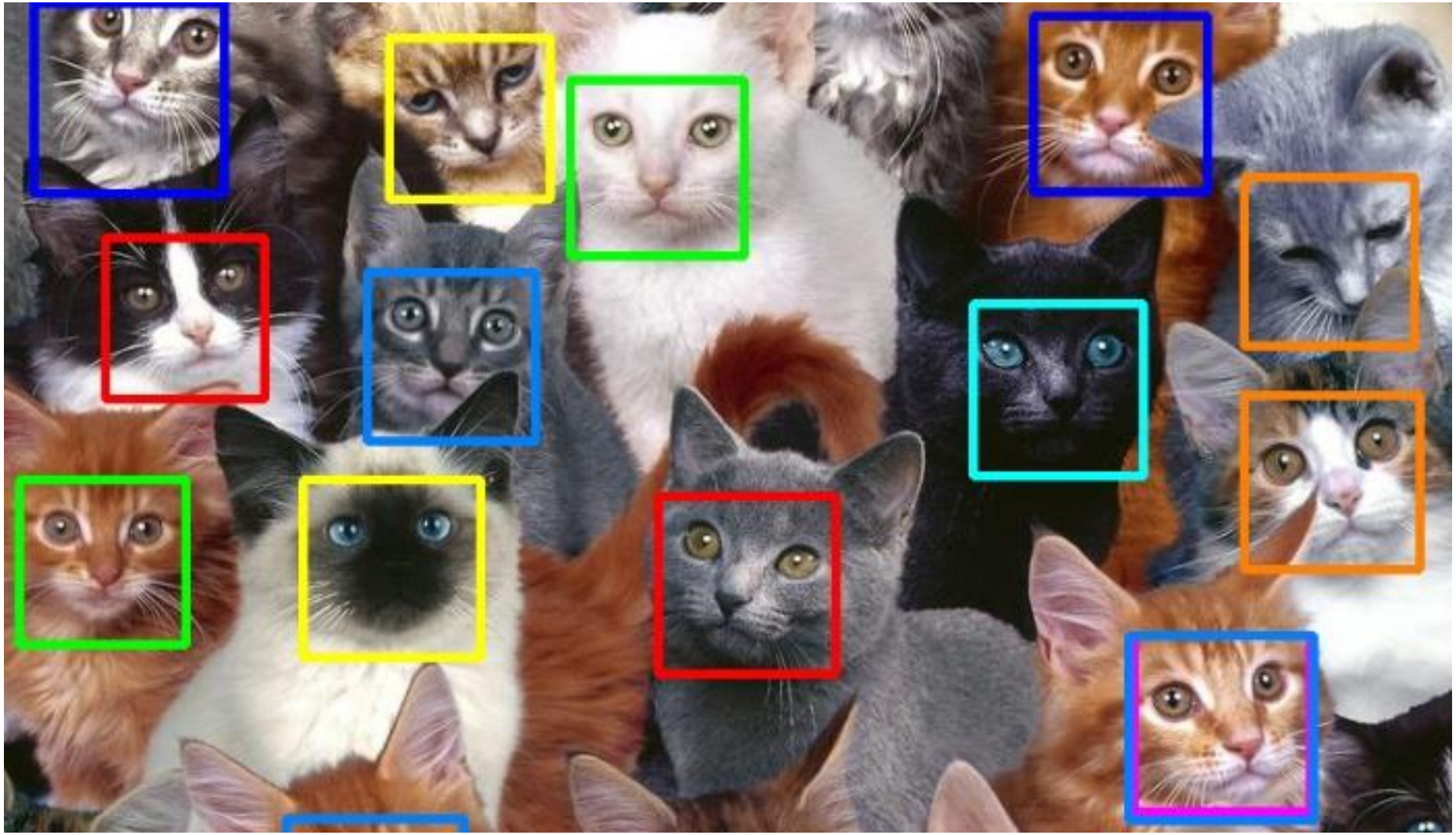


https://sgugger.github.io/the-1cycle-policy.html

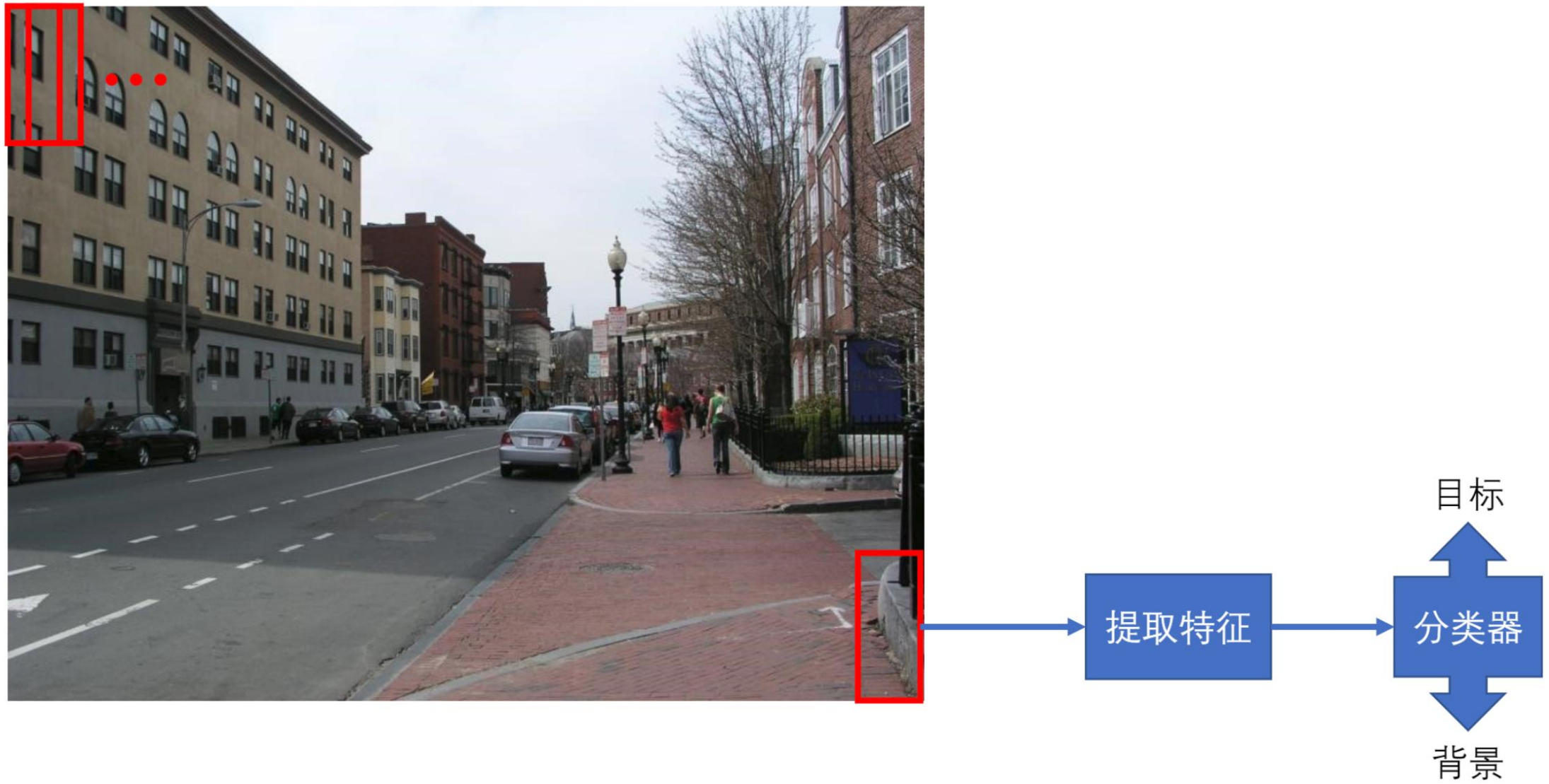
目标检测



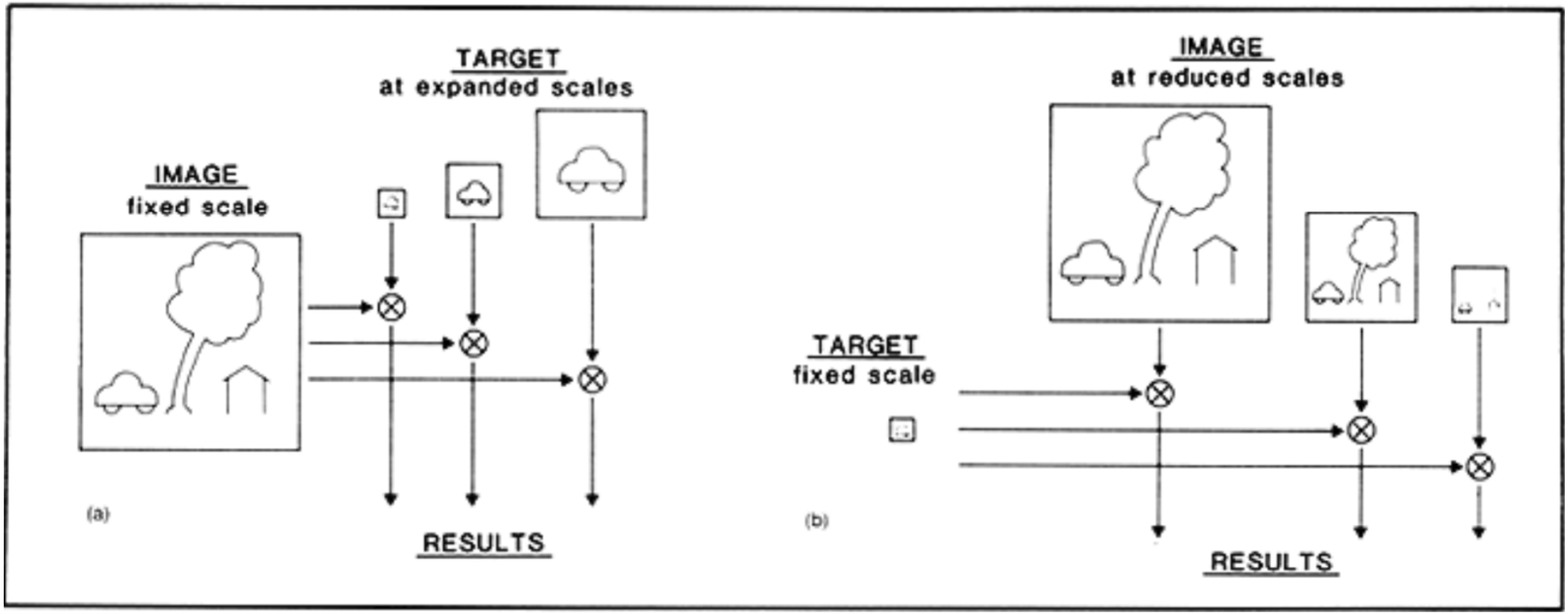
目标检测



滑动窗方法



如何检测不同尺度的目标



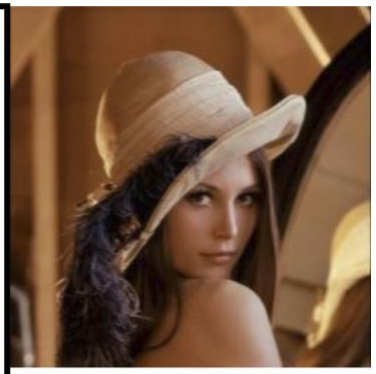
图像金字塔

$S_0 = 512$

$S_1 =?$

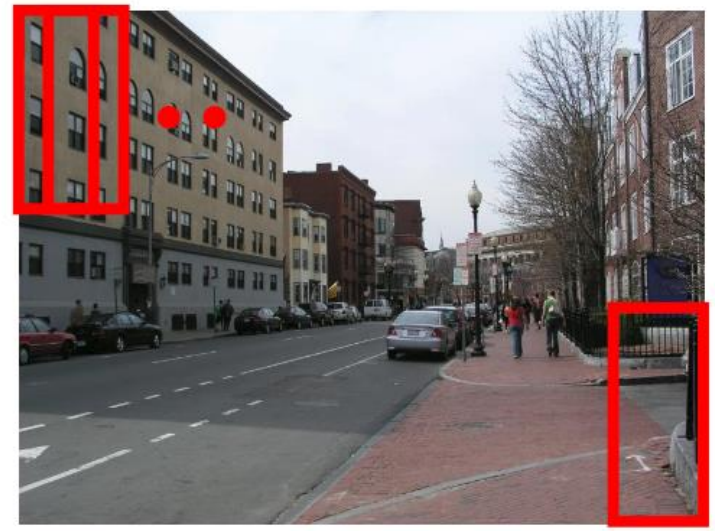
$S_2 =?$

$S_3 =?$



$S_4 = 256$

$$S_1 = \lambda S_0$$
$$S_2 = \lambda S_1$$
$$S_3 = \lambda S_2$$
$$S_4 = \lambda S_3$$

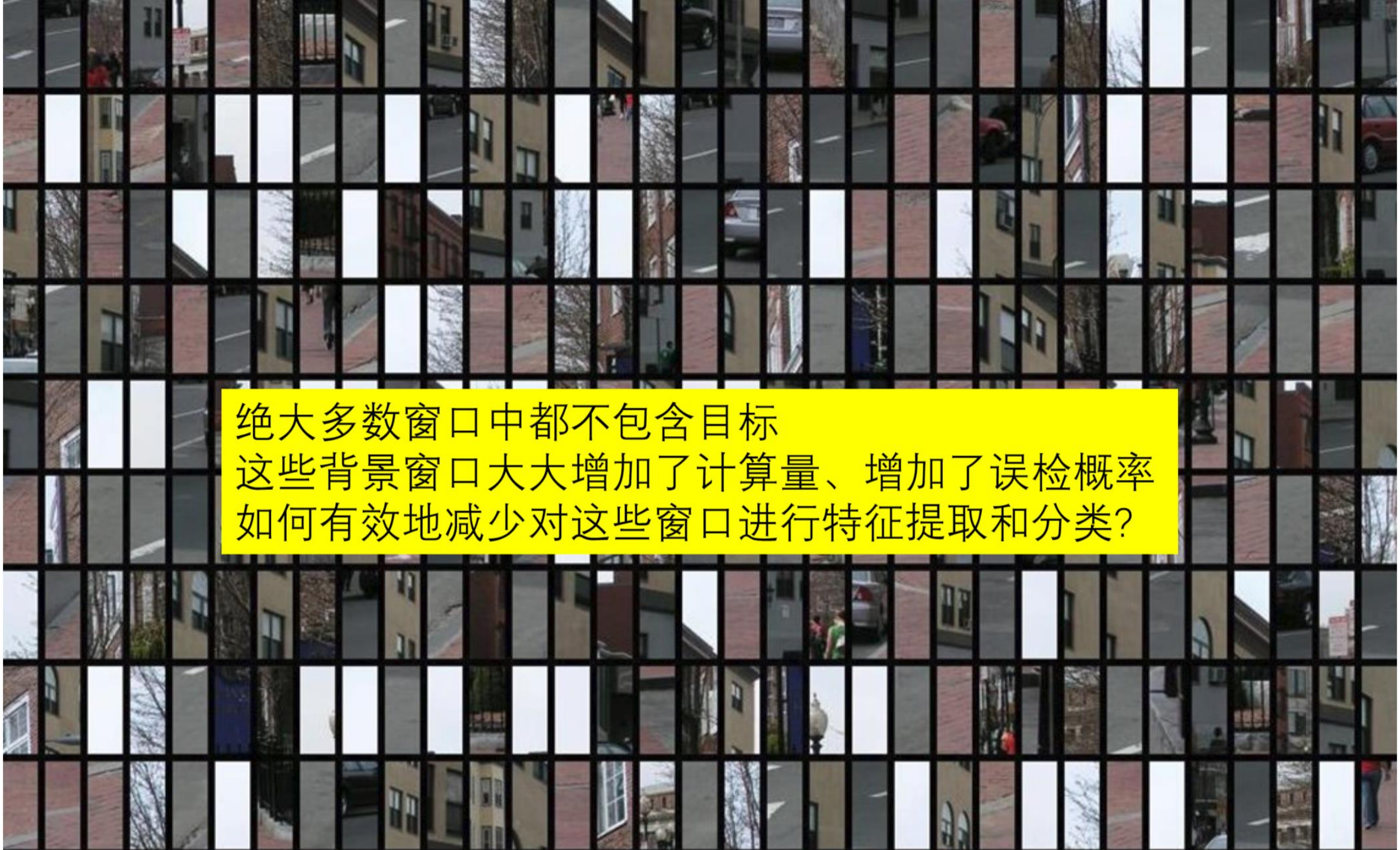


Problems?

图像大小: $W \times W$
检测窗口大小: $s \times s$
窗口数量: $(W - s + 1) \times (W - s + 1)$
目标数量: ~ 10

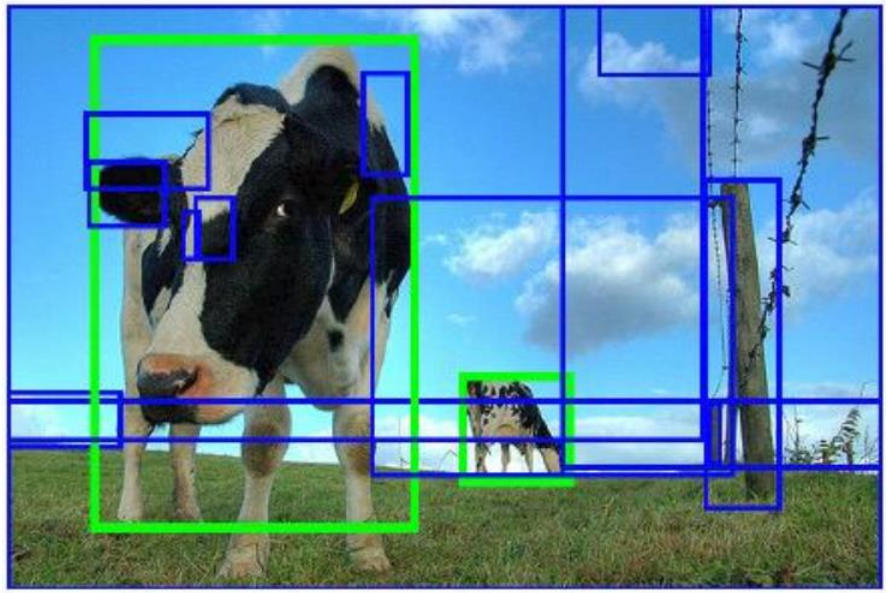
图像大小: 1000×1000
窗口大小: 100×100

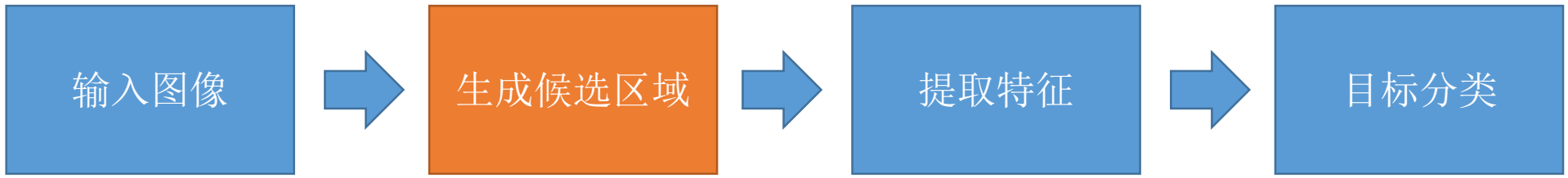
窗口数量: $901 \times 901 \approx 80$万


绝大多数窗口中都不包含目标
这些背景窗口大大增加了计算量、增加了误检概率
如何有效地减少对这些窗口进行特征提取和分类?

候选区域

- Region Proposals
- 筛选出可能是目标的区域
  - 快速
  - 尽可能少的候选区域
  - 不错过任何一个目标

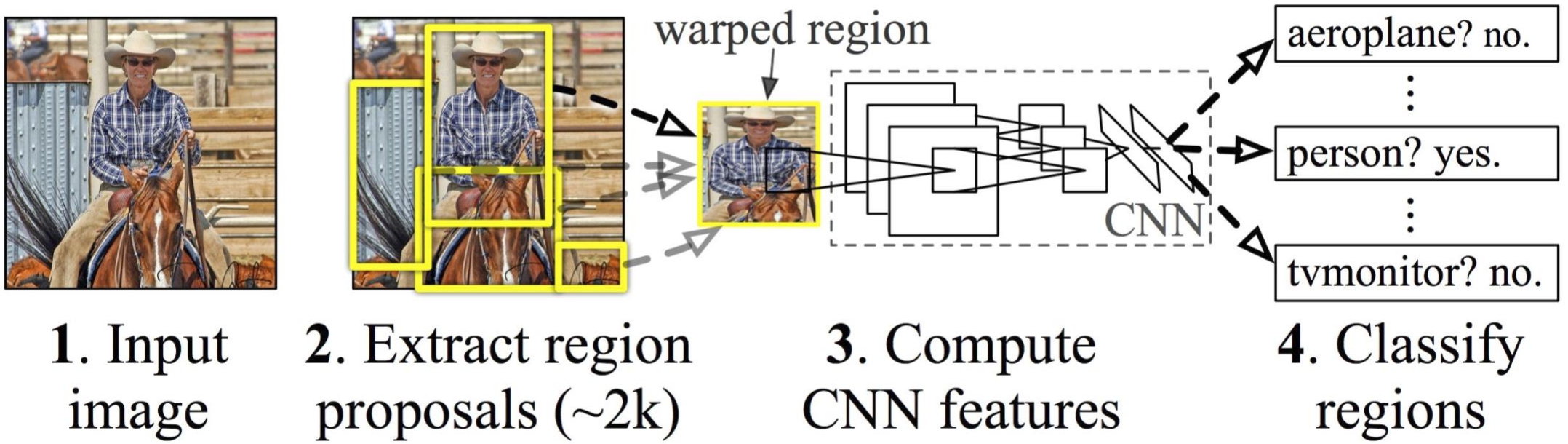




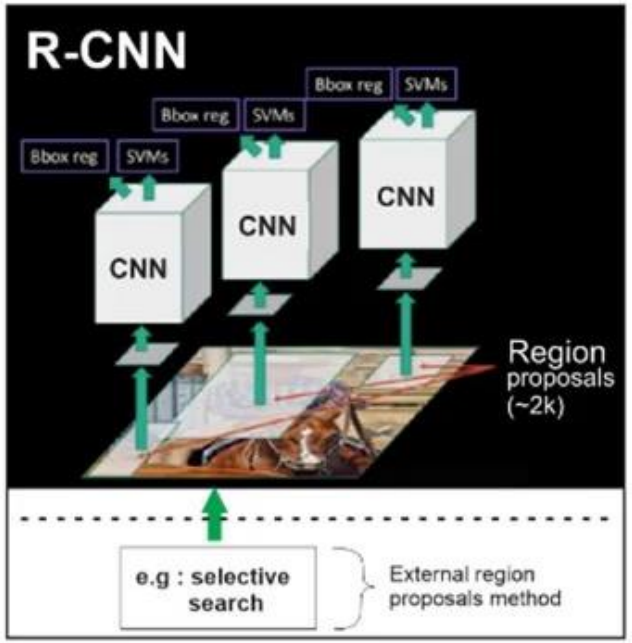
候选框算法

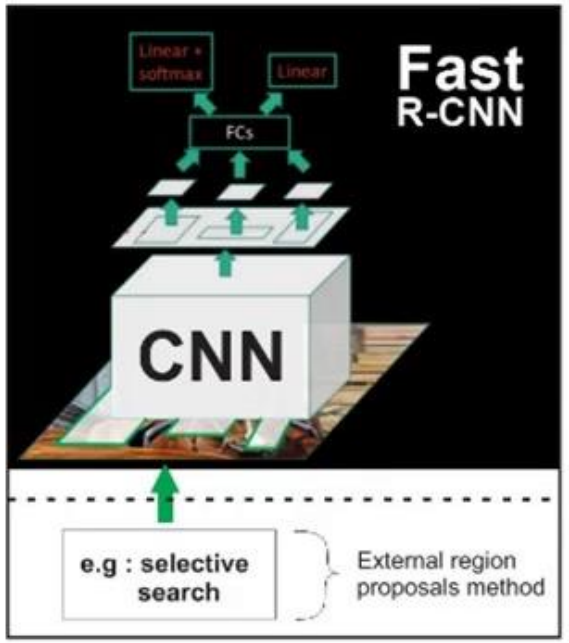
Method	Approach	Outputs Segments	Outputs Score	Control #proposals	Time (sec.)
Bing [18]	Window scoring		✓	✓	0.2
CPMC [19]	Grouping	✓	✓	✓	250
EdgeBoxes [20]	Window scoring		✓	✓	0.3
Endres [21]	Grouping	✓	✓	✓	100
Geodesic [22]	Grouping	✓		✓	1
MCG [23]	Grouping	✓	✓	✓	30
Objectness [24]	Window scoring		✓	✓	3
Rahtu [25]	Window scoring		✓	✓	3
RandomizedPrim's [26]	Grouping	✓		✓	1
Rantalankila [27]	Grouping	✓		✓	10
Rigor [28]	Grouping	✓		✓	10
SelectiveSearch [29]	Grouping	✓	✓	✓	10

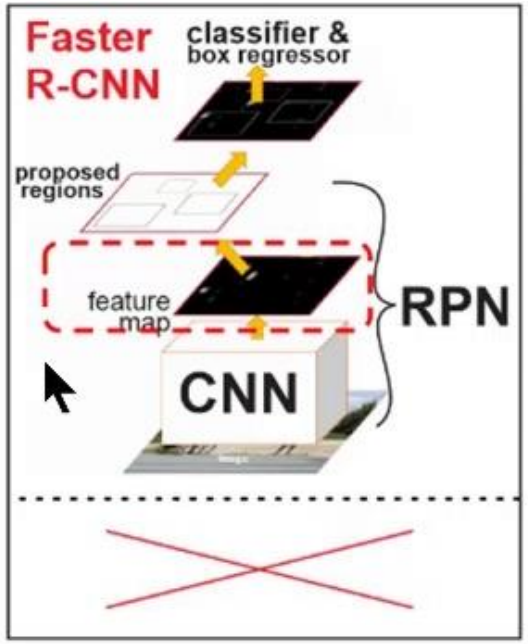
R-CNN
(Regions with Convolutional Neural Network Features)



Two Stage Detector



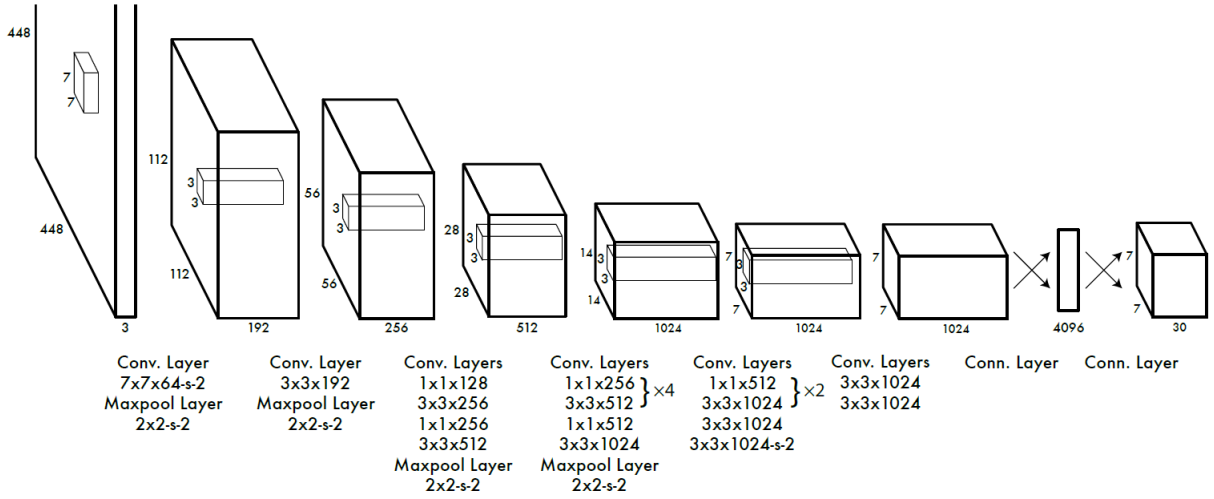


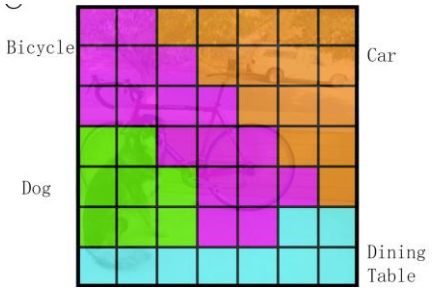


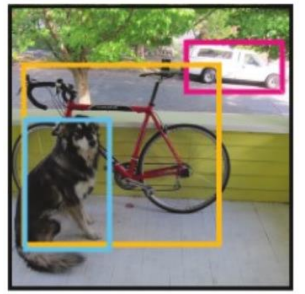
	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image	50 seconds	2 seconds	0.2 seconds
Speed-up	1x	25x	250x
mAP (VOC 2007)	66.0%	66.9%	66.9%

One Stage Detector

YOLO

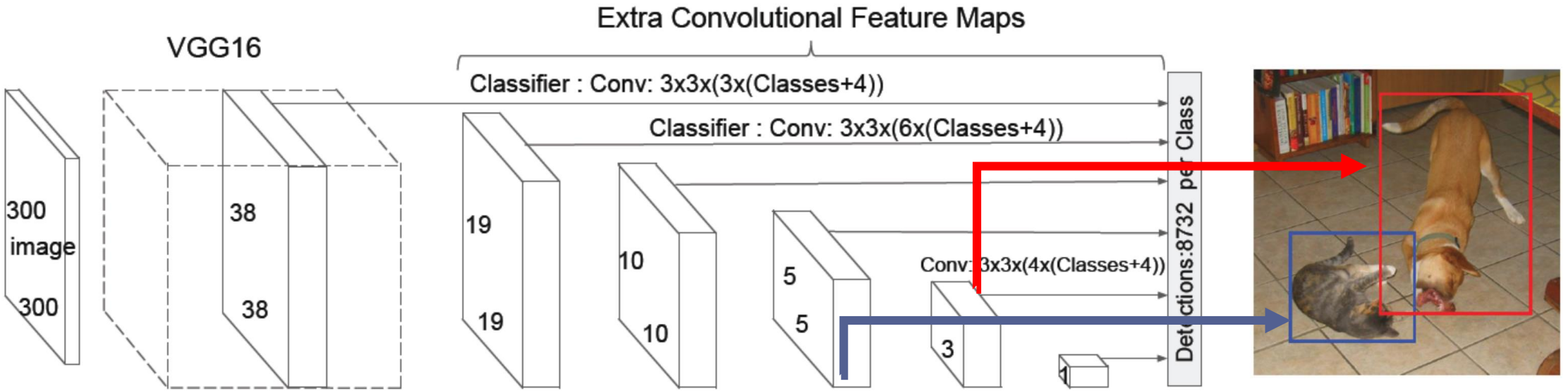




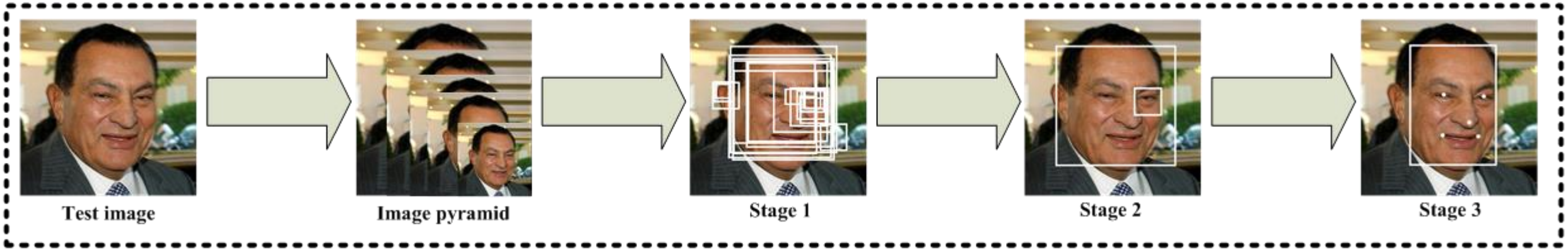


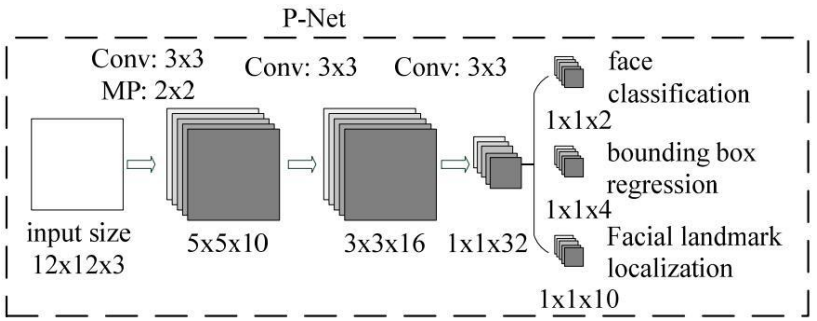
Final detections

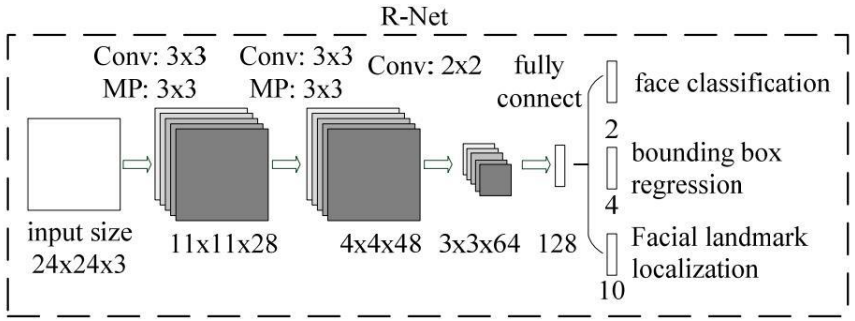
SSD

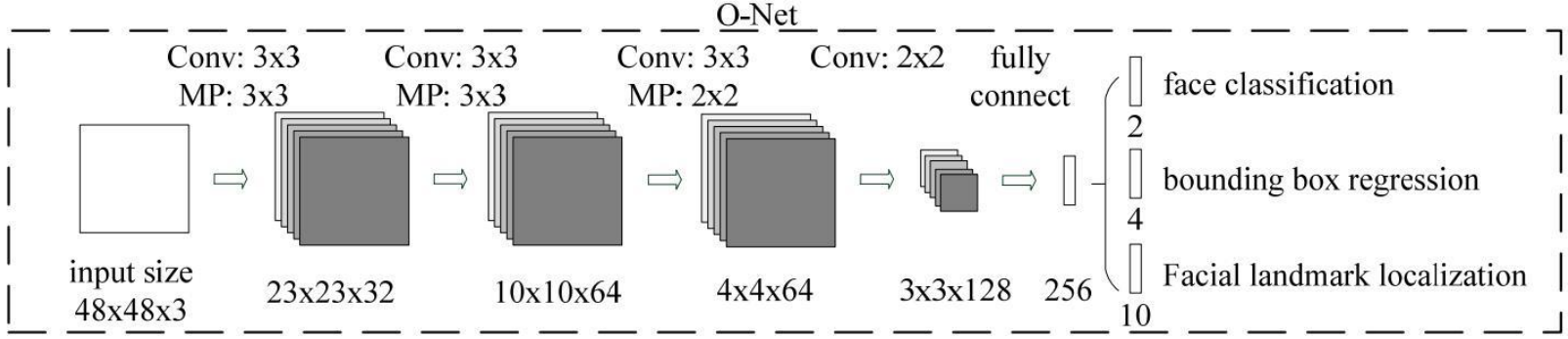


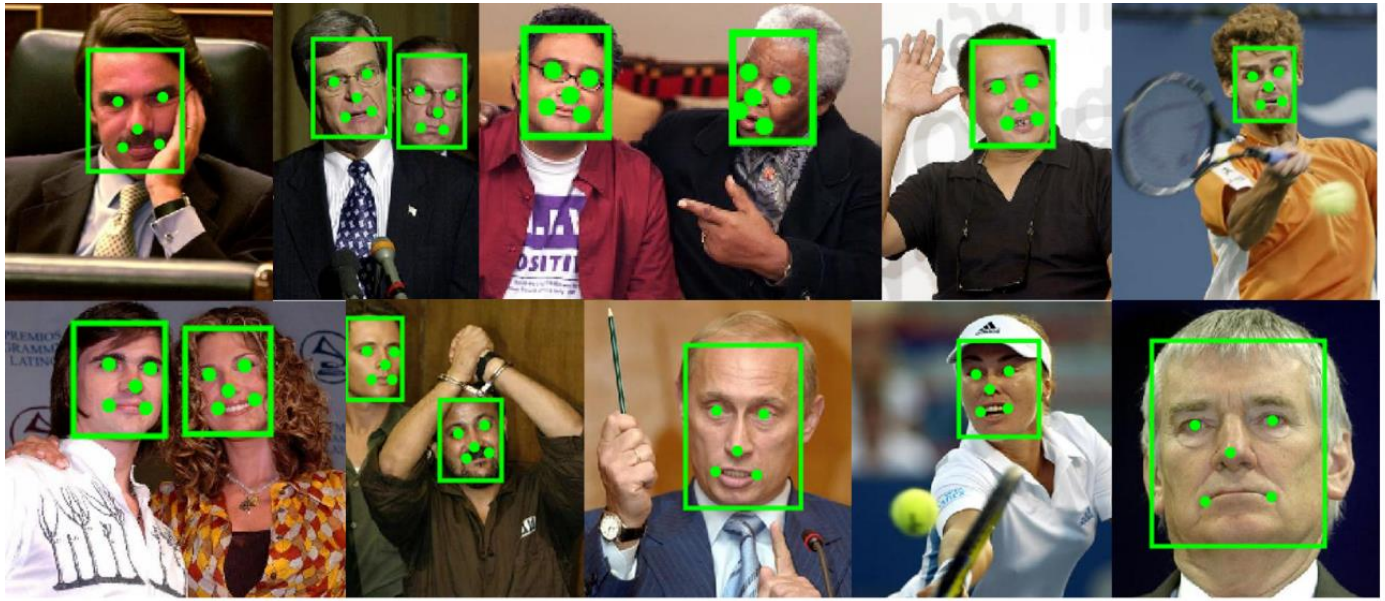
MTCNN


Test image
Image pyramid
Stage 1
Stage 2
Stage 3


P-Net
Conv: 3x3
MP: 2x2
Conv: 3x3
Conv: 3x3
face classification
1x1x2
bounding box regression
1x1x4
Facial landmark localization
1x1x10
input size
12x12x3
5x5x10
3x3x16
1x1x32


R-Net
Conv: 3x3
MP: 3x3
Conv: 3x3
MP: 3x3
Conv: 2x2
fully connect
face classification
2
bounding box regression
4
Facial landmark localization
10
input size
24x24x3
11x11x28
4x4x48
3x3x64
128

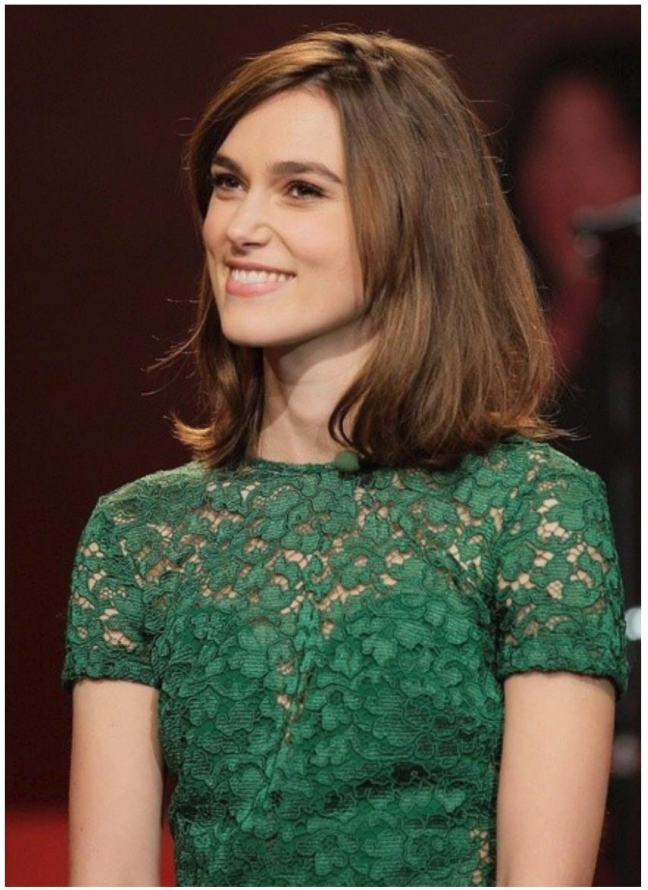

O-Net
Conv: 3x3
MP: 3x3
Conv: 3x3
MP: 3x3
Conv: 3x3
MP: 2x2
Conv: 2x2
fully connect
face classification
2
bounding box regression
4
Facial landmark localization
10
input size
48x48x3
23x23x32
10x10x64
4x4x64
3x3x128
256

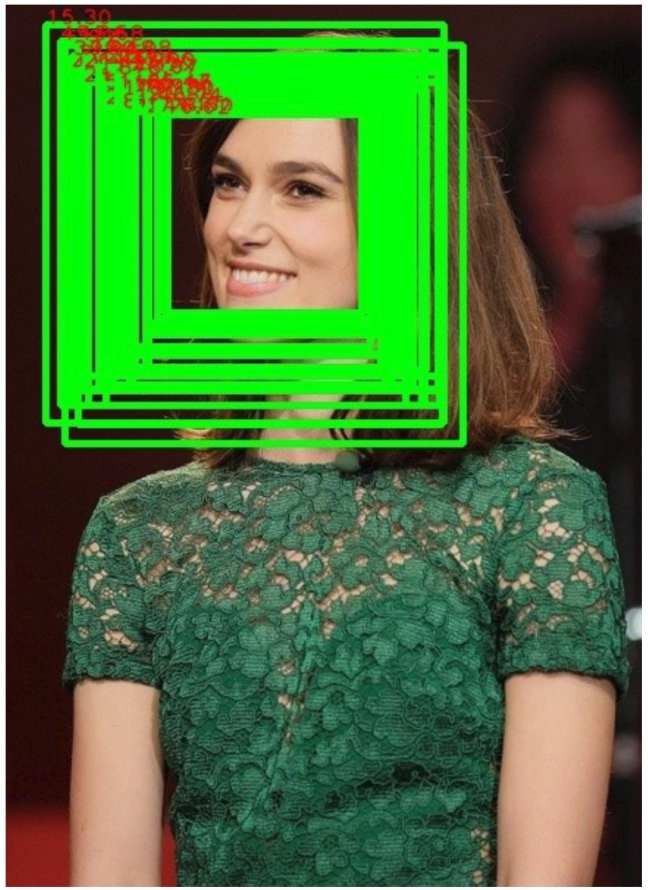


(a) Examples of results on FDDB

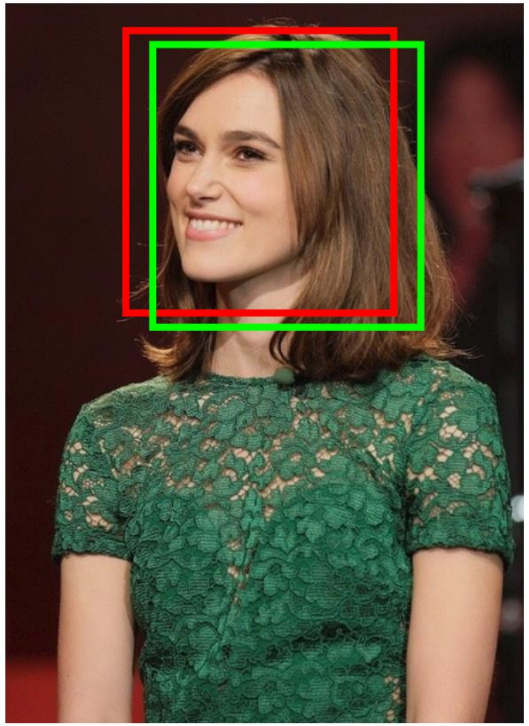
(b) Examples of results on WIDER FACE

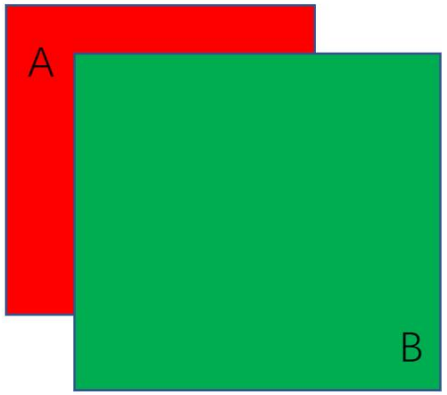
非极大值抑制(NMS, Non-Maximum Suppression)

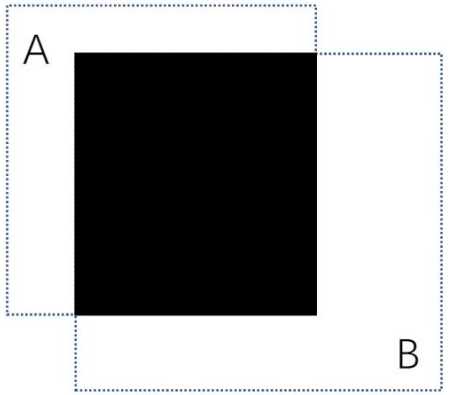




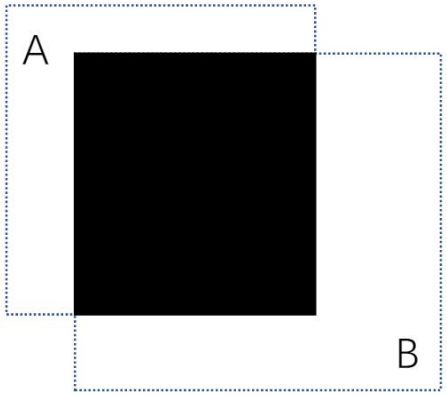
交并比(IOU, Intersection Over Union)

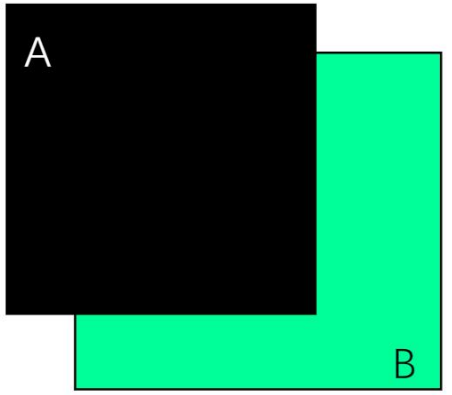






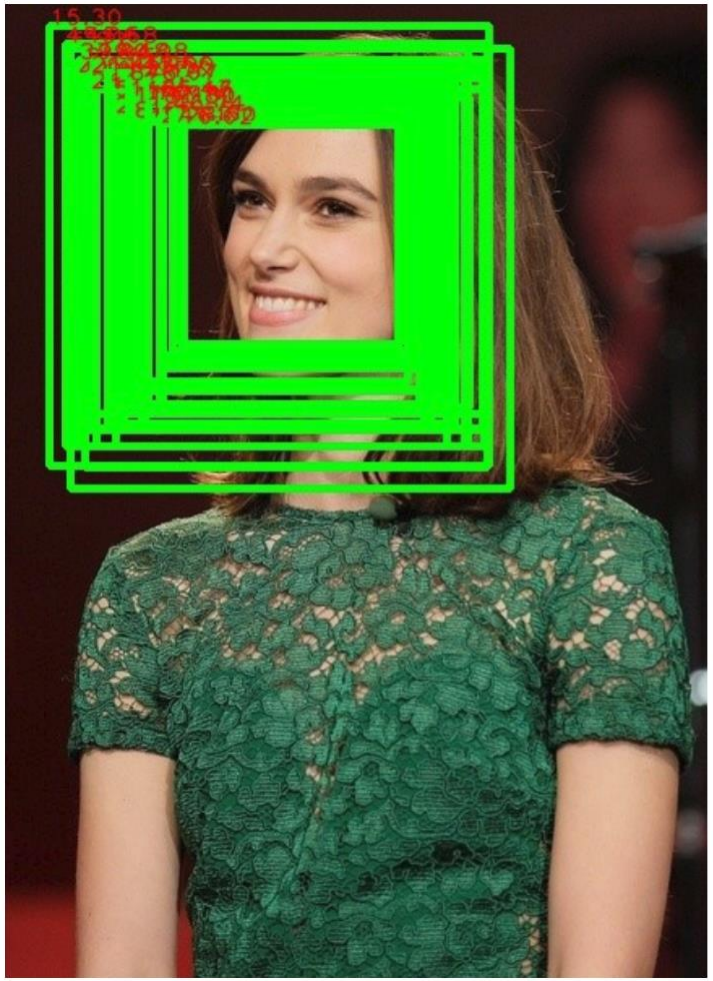
$$IOU(A,B) = \frac{|A \cap B|}{|A \cup B|}$$



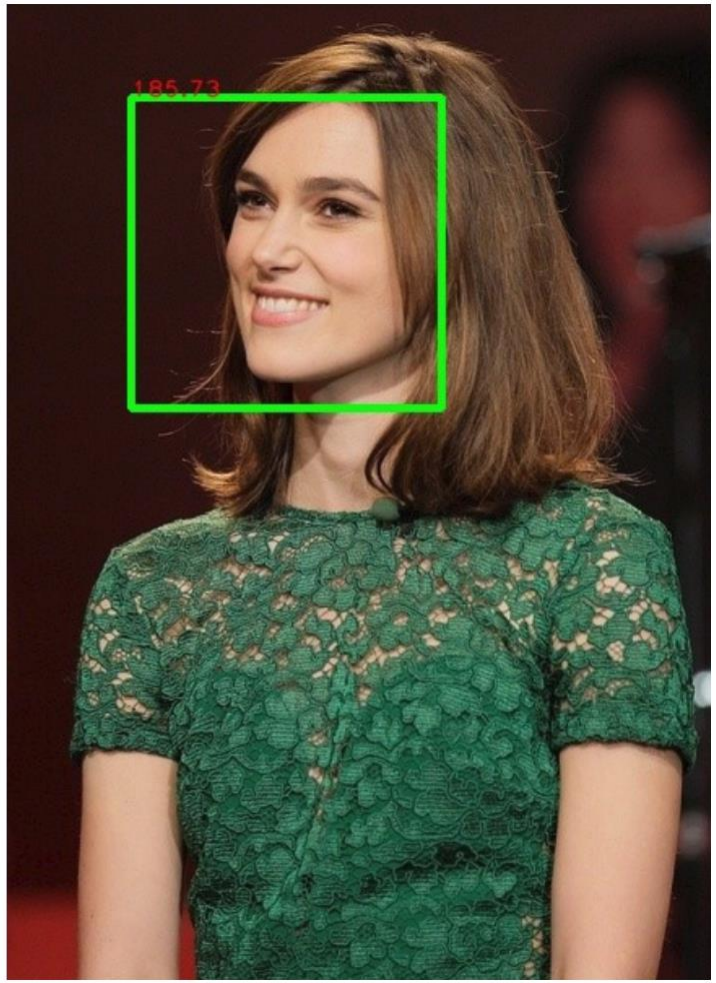


$$IOM(A,B) = \frac{|A \cap B|}{\min\{|A|,|B|\}}$$

NMS



NMS

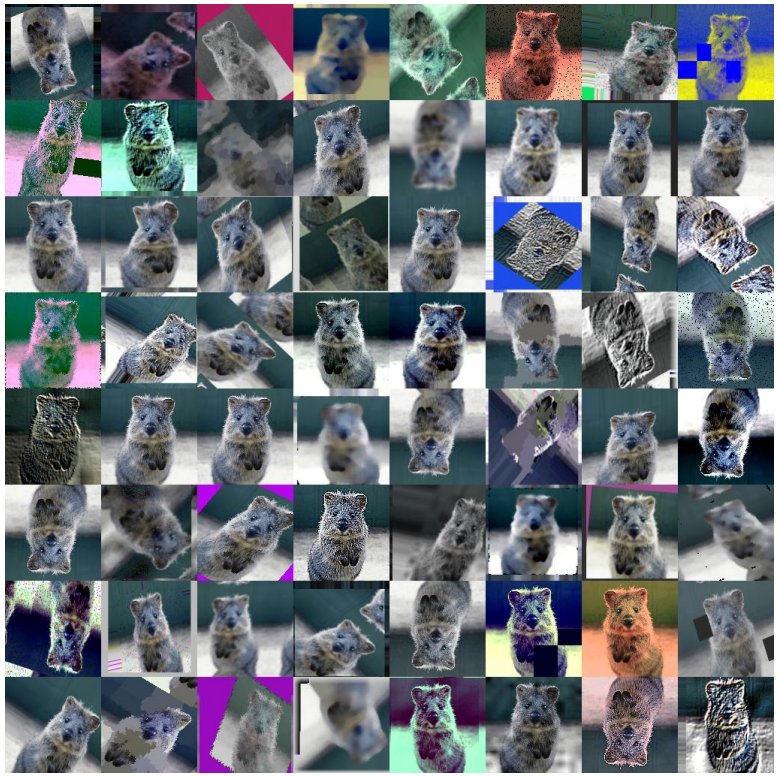


数据增广

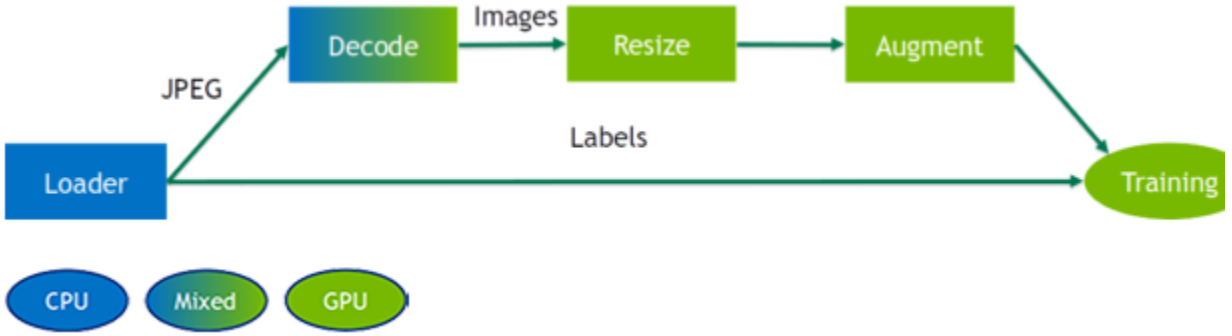
- 数据增广是深度学习中常用的技巧之一
- 增加训练数据集，让数据集尽可能的多样化，使得训练的模型具有更强的泛化能力
- 常用的数据增光：水平/垂直翻转，旋转，缩放，裁剪，剪切，平移，对比度，色彩抖动，噪声等
- 实际使用过程中，需要“因地制宜”
  - 人脸识别中，图像的垂直翻转就没有意义

libraries

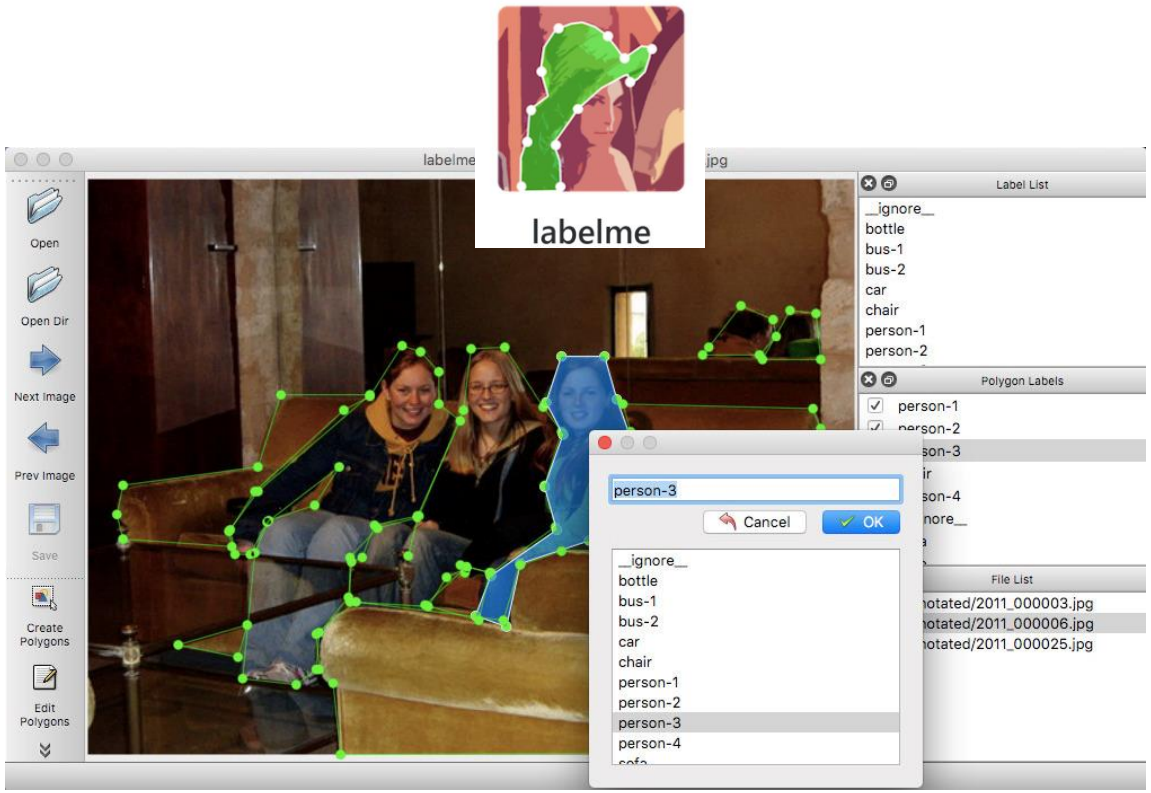
aleju / **imgaug**	Watch 185	Unstar 5,577	Fork 1,154
mdbloice / **Augmentor**	Watch 109	Unstar 3,038	Fork 567
albu / **albumentations**	Watch 66	Unstar 2,229	Fork 276
NVIDIA / **DALI**	Watch 64	Unstar 1,225	Fork 149



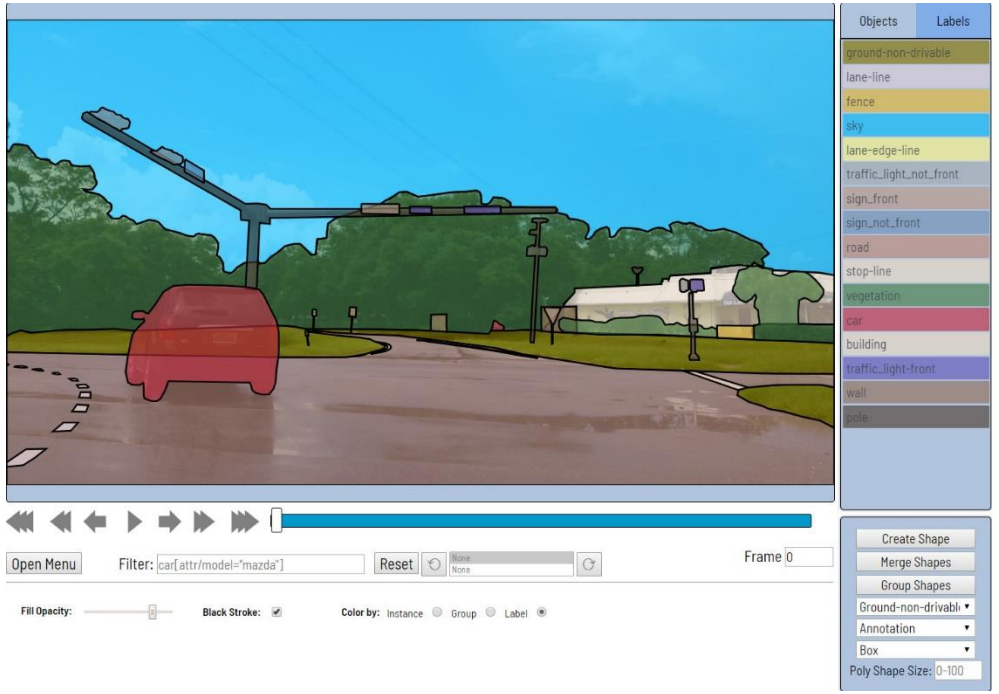
Augmentor



数据标注工具



CVAT



更多 https://github.com/mingx9527/Data_Label_Tools

Final Project

- 任何“有趣”的项目，例如：
  - 基于人脸识别的自动签到系统
  - 自动售货机的商品识别
  - 自然场景的车牌识别
  - 头像漫画风格化
  - 历史照片上色
  - 图像内容转换
- 其它启发：
  - 学术会议：CVPR、ICCV、ECCV、 NeurIPS 、ICLR、ICML
  - https://github.com/kjw0612/awesome-deep-vision

要求

- 团队项目，每个团队4-5人
- 项目流程包含：
  - 数据收集（拍摄、下载、清理、标注）
  - 模型选择和实现
  - 模型训练和调参
  - 演示系统搭建
- 提交项目PPT并做presentation，同时提交代码
- 最终分数由presentation（70%）和代码（30%）共同决定

Pick one project,
do it very well,
and make it **fantastic**.

下周预告

- 图像复原与增强 - Guest Lecture from 傅雪阳