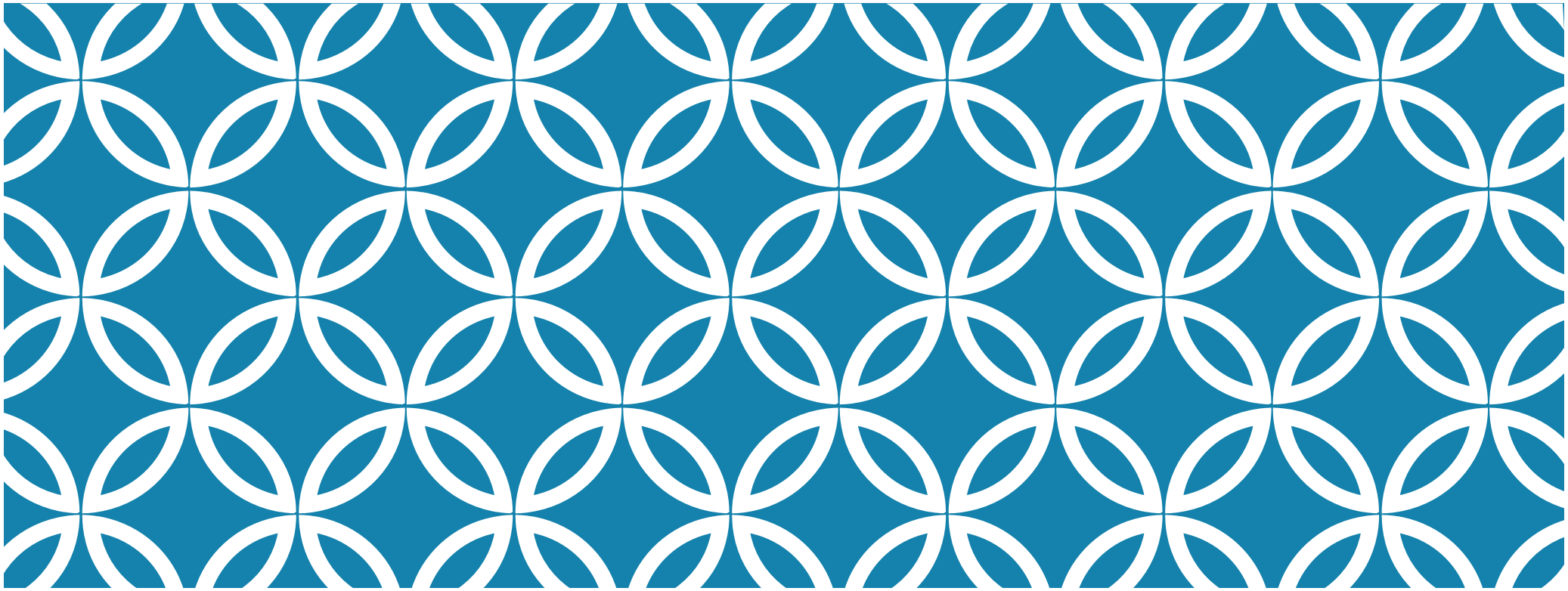
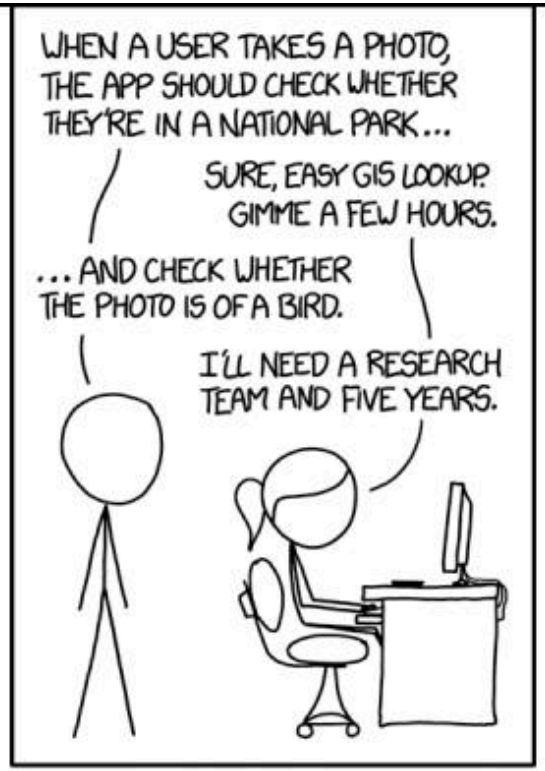


SPATIALLY AWARE COMPUTER VISION MODELS IN DEEP LEARNING

A lightening fast tour

2012:



2019:

```
from tensorflow.keras.applications
import PretrainedBirdDetector
```

Overview of tasks

Semantic segmentation

- FCN
- U-Net
- DeepLabv3+

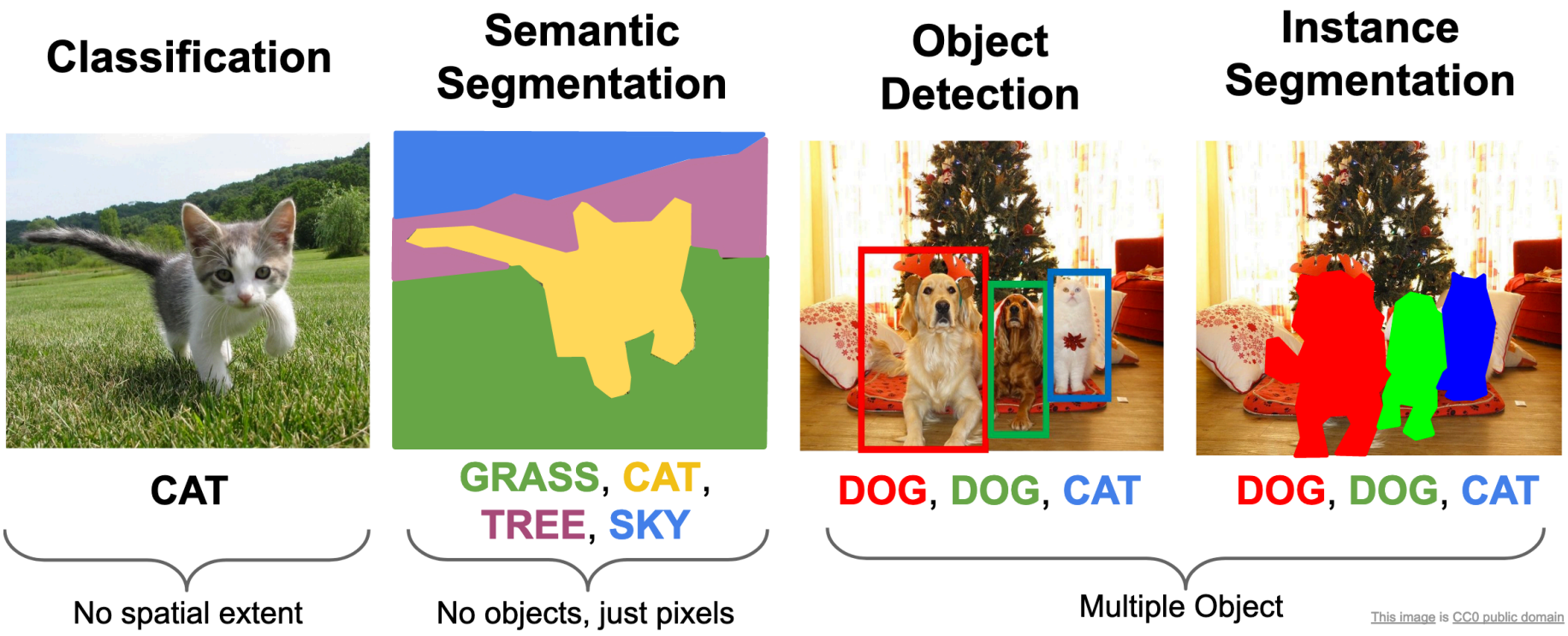
Object identification

- Faster R-CNN
- RetinaNet

Instance segmentation

- Mask R-CNN

CLASSIFICATION'S COMPLICATED COUSINS



Source: Slide 7 of http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture12.pdf

SEMANTIC SEGMENTATION

Simply think of as “pixel-level classification”

For a 512x512 picture – 260K+ predictions!

No concept of objects - only connected regions of similar classification

Applications:

Medical image labelling

Reservoir engineering

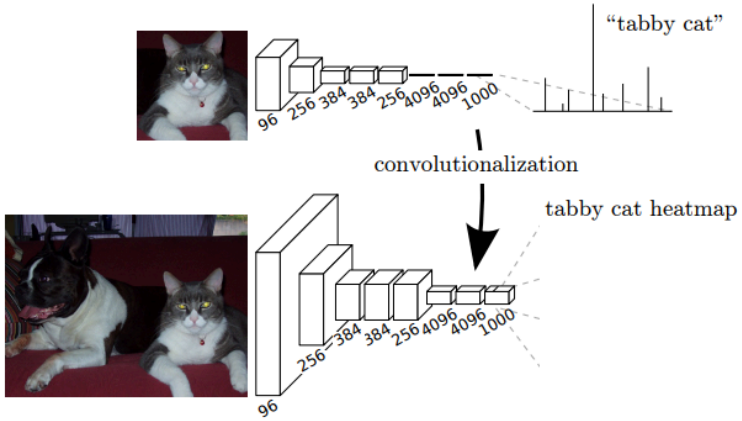
Aerial surveillance

Image: https://www.cityscapes-dataset.com/examples/

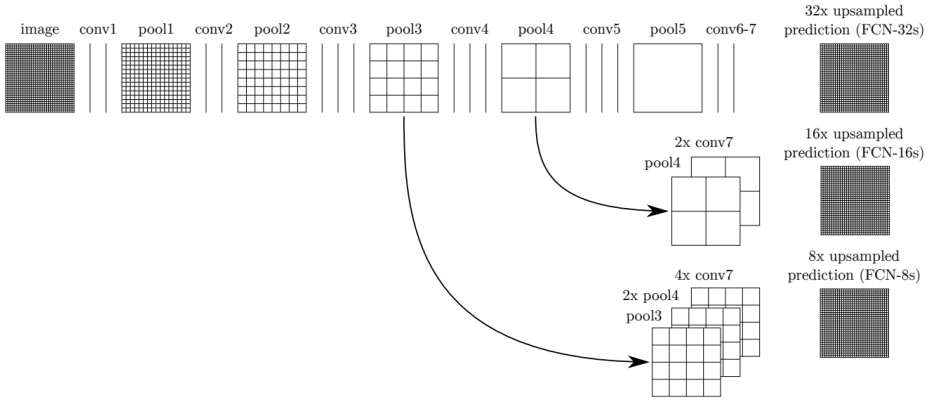
SEMANTIC SEGMENTATION: FCN

FULLY CONVOLUTIONAL NETWORKS FOR SEMANTIC SEGMENTATION

1st idea: Just use a CNN on each pixel…



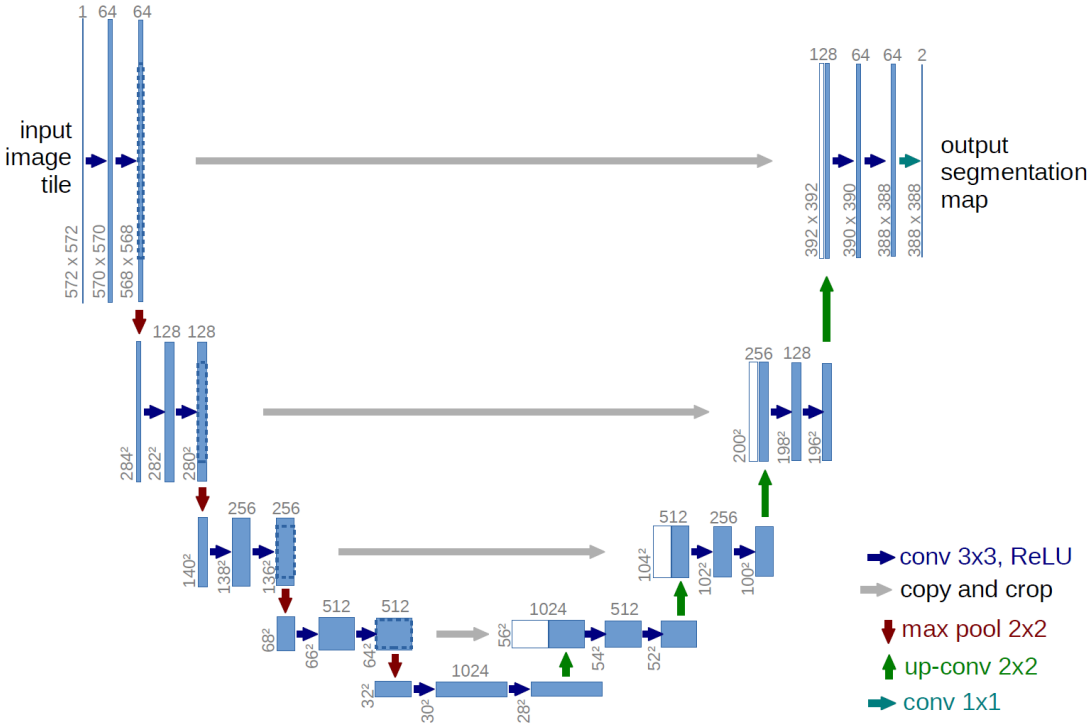
…but don't throw away all of the local feature maps calculated along the way!



Paper: https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

SEMANTIC SEGMENTATION: U-NET

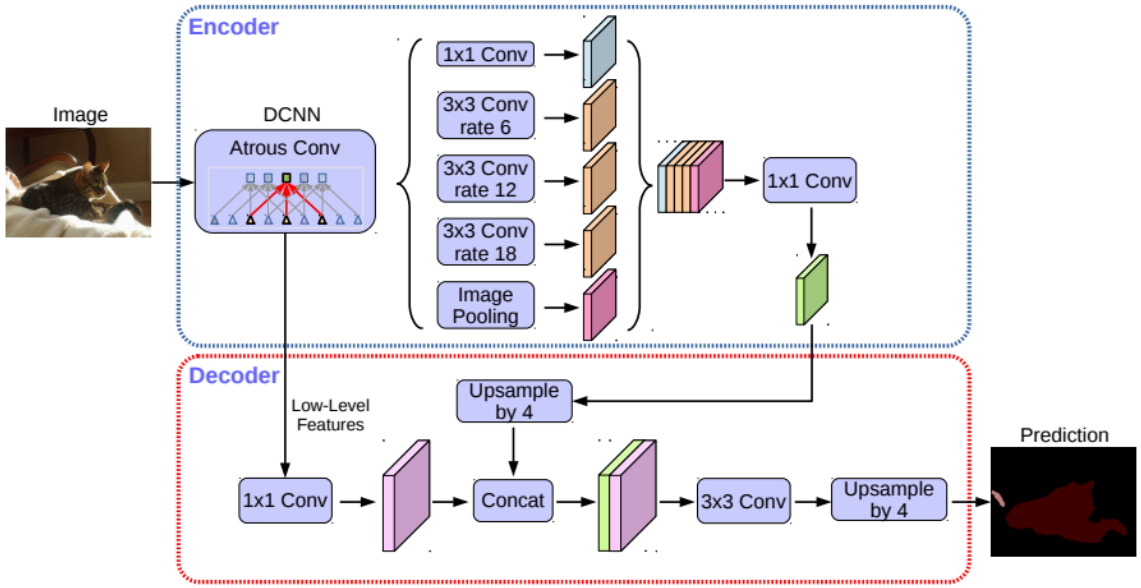
CONVOLUTIONAL NETWORKS FOR BIOMEDICAL IMAGE SEGMENTATION



Paper: https://arxiv.org/pdf/1505.04597.pdf

SEMANTIC SEGMENTATION: DEEPLABV3+

ENCODER-DECODER WITH ATROUS SEPARABLE CONVOLUTION FOR SEMANTIC IMAGE SEGMENTATION



For further info on evolution and implementations: https://github.com/tensorflow/models/tree/master/research/deeplab
Latest paper: https://arxiv.org/pdf/1802.02611v3.pdf

OBJECT IDENTIFICATION

Draw a box around every object in the image and classify the object in the box

Break it down into LOCALIZATION + CLASSIFICATION

..but treat localization as a classification problem

1. Cover the image with “boxes”
2. Classify each box as “has thing” or “no has thing”
3. If box “has thing”
    1. Classify the thing
    2. Use regression to nip/tuck the edges

OBJECT IDENTIFICATION: 2 STRATEGIES

One-stage detectors

i.e. YOLOv3, SSD, RetinaNet

1. Classifying “thingness”, regress box edges and classify boxes with things into their proper category

Two-stage detectors

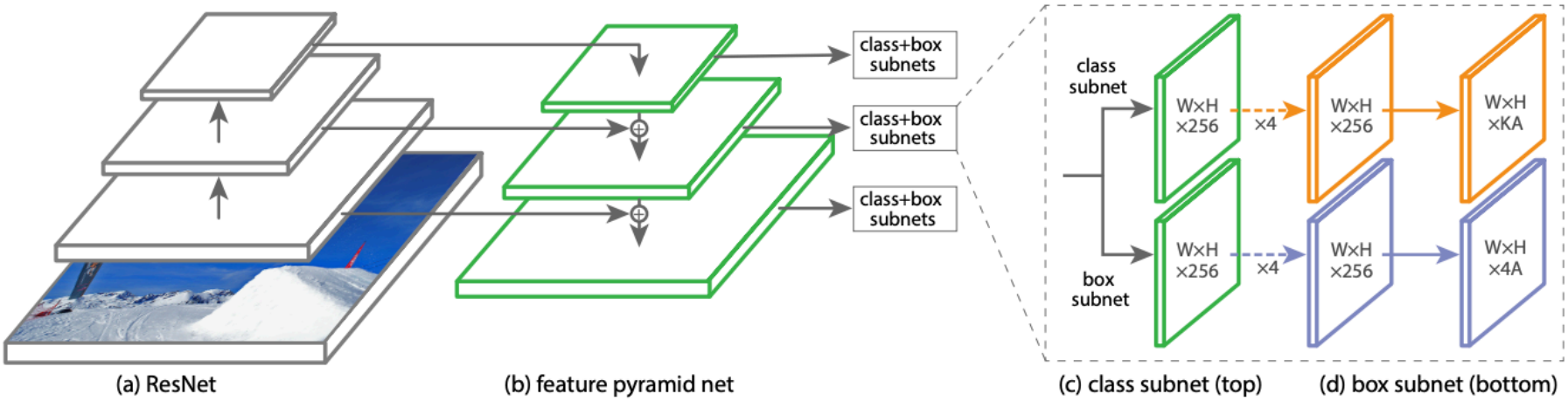
i.e. Faster R-CNN and family

1. Classifying “thingness” and regress box edges

2. Classify boxes with things into their proper category (and maybe more edge refinement)

OBJECT IDENTIFICATION: RETINA-NET

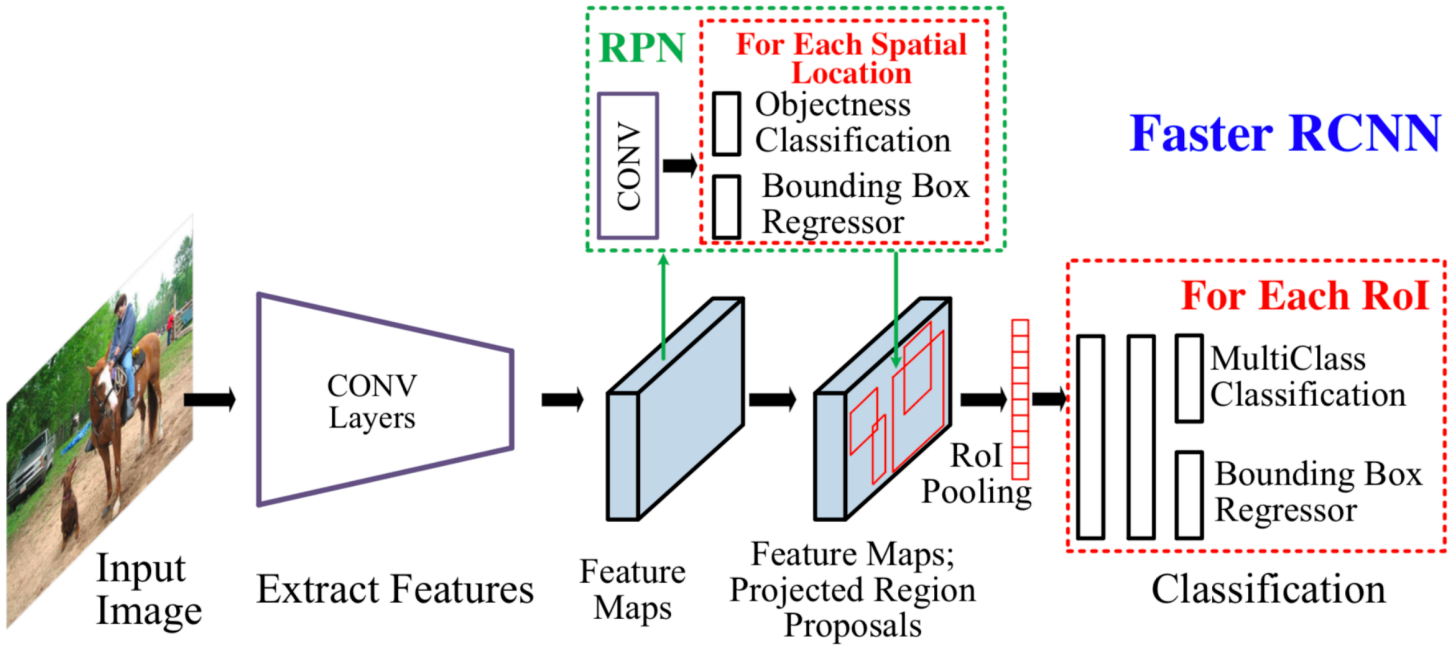
FOCAL LOSS FOR DENSE OBJECT DETECTION



Paper: https://arxiv.org/pdf/1708.02002.pdf

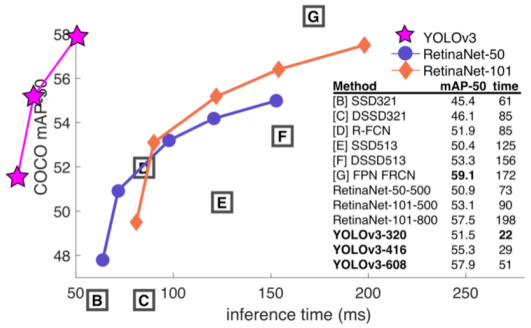
OBJECT IDENTIFICATION: FASTER R-CNN

TOWARDS REAL-TIME OBJECT DETECTION WITH REGION PROPOSAL NETWORKS

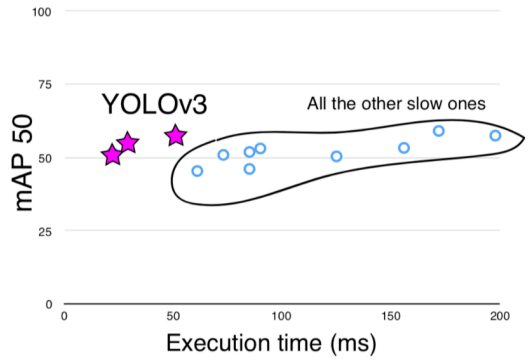


Paper: https://arxiv.org/pdf/1506.01497.pdf **Image from:** https://arxiv.org/abs/1809.02165

EPIC YOLOV3 PAPER



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51



1. Introduction

Sometimes you just kinda phone it in for a year, you know? I didn't do a whole lot of research this year. Spent a lot of time on Twitter. Played around with GANs a little. I had a little momentum left over from last year [12] [1]; I managed to make some improvements to YOLO. But, honestly, nothing like super interesting, just a bunch of small changes that make it better. I also helped out with other people's research a little.

Actually, that's what brings us here today. We have a camera-ready deadline [4] and we need to cite some of the random updates I made to YOLO but we don't have a source. So get ready for a TECH REPORT!

The great thing about tech reports is that they don't need intros, y'all know why we're here. So the end of this introduction will signpost for the rest of the paper. First we'll tell you what the deal is with YOLOv3. Then we'll tell you how we do. We'll also tell you about some things we tried that didn't work. Finally we'll contemplate what this all means.

2. The Deal

So here's the deal with YOLOv3: We mostly took good ideas from other people. We also trained a new classifier network that's better than the other ones. We'll just take you through the whole system from scratch so you can understand it all.

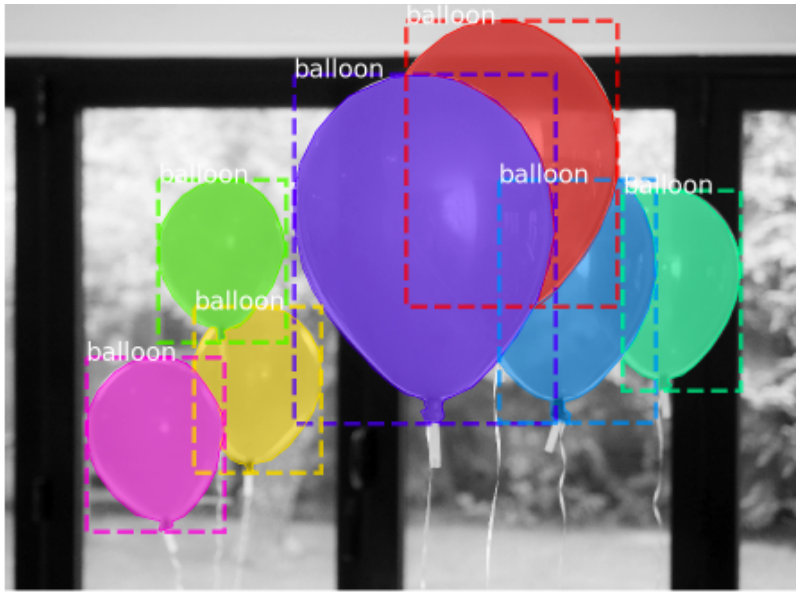
Text from https://arxiv.org/abs/1804.02767

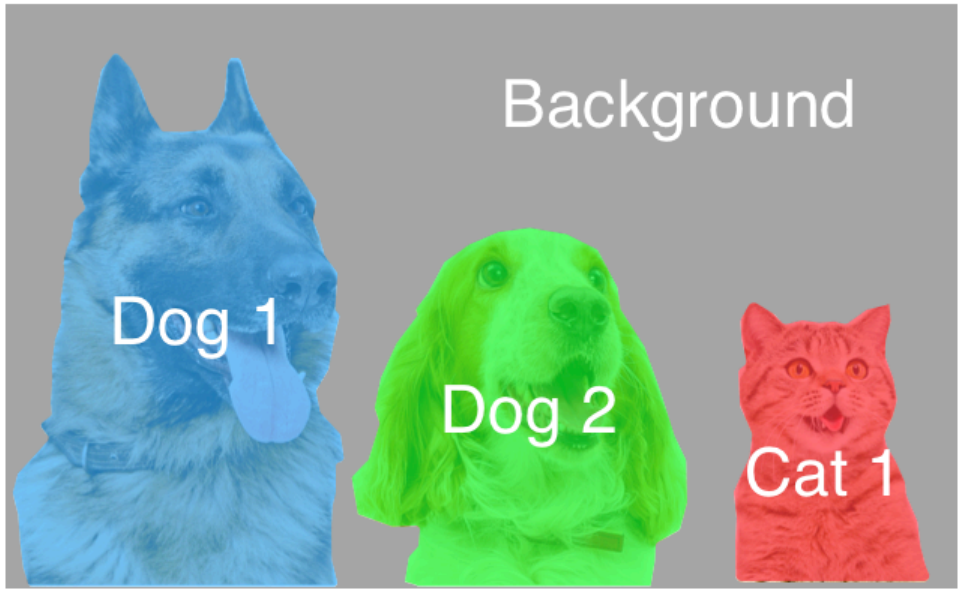
Source: https://github.com/arthurredfern/UT-Dallas-CS-6301-CNNs/blob/master/Lectures/xNNs_080_Vision.pdf

INSTANCE SEGMENTATION

Instance Segmentation = **Object Identification** + **Semantic Segmentation**

Classify each pixel as belonging to Object X of Class Y





Figures: https://towardsdatascience.com/instance-segmentation-using-mask-r-cnn-7f77bdd46abd
https://gluon-cv.mxnet.io/

INSTANCE SEGMENTATION: MASK R-CNN

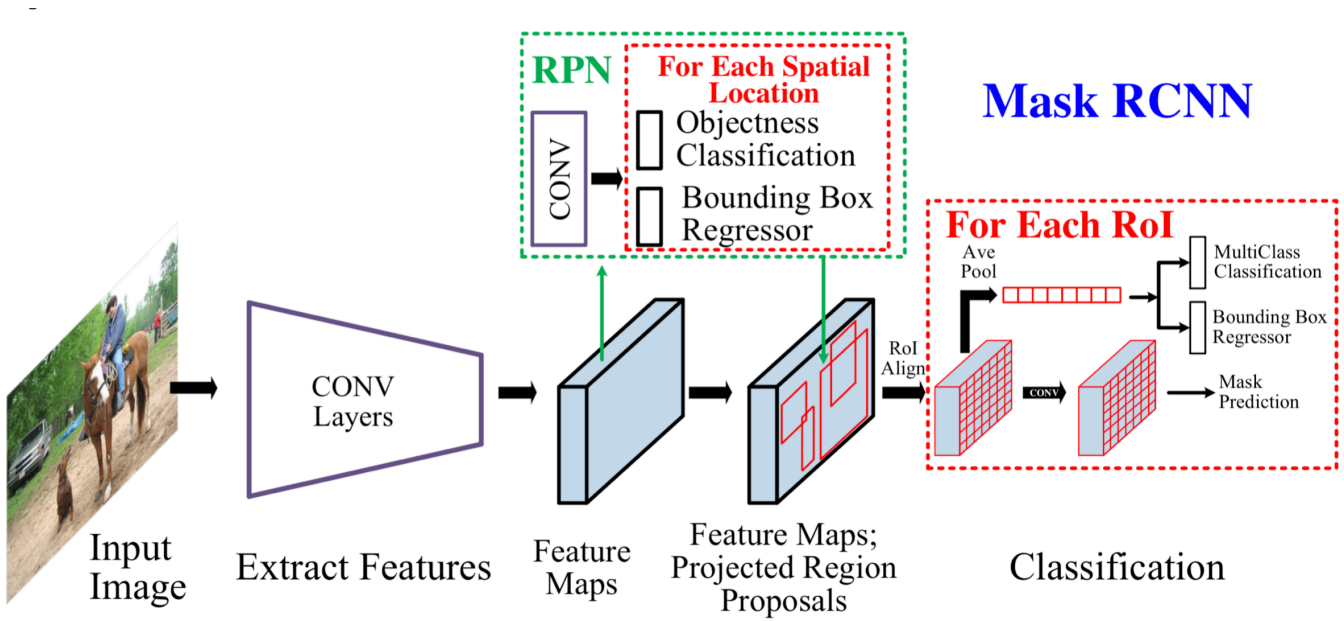


Figure: https://arxiv.org/pdf/1809.02165.pdf
Paper: https://arxiv.org/pdf/1703.06870.pdf

RECOMMENDED IMPLEMENTATIONS

Description	Implementation	Framework	Models
Google Research DeepLab	https://github.com/tensorflow/models/tree/master/research/deeplab	Tensorflow	DeepLabv3+
Matterport's Mask R-CNN	https://github.com/matterport/Mask_RCNN	Tensorflow/Keras	Mask R-CNN
Detectron2	https://github.com/facebookresearch/detectron2	PyTorch	RetinaNet Faster R-CNN Mask R-CNN
GluonCV	https://gluon-cv.mxnet.io/	MXNet + GluonCV	SSD YoloV3 Faster R-CNN Mask R-CNN
Open MMLab Detection Toolbox	https://github.com/open-mmlab/mmdetection	PyTorch	Too many to list…

MORE THEORY

Digital Photography:

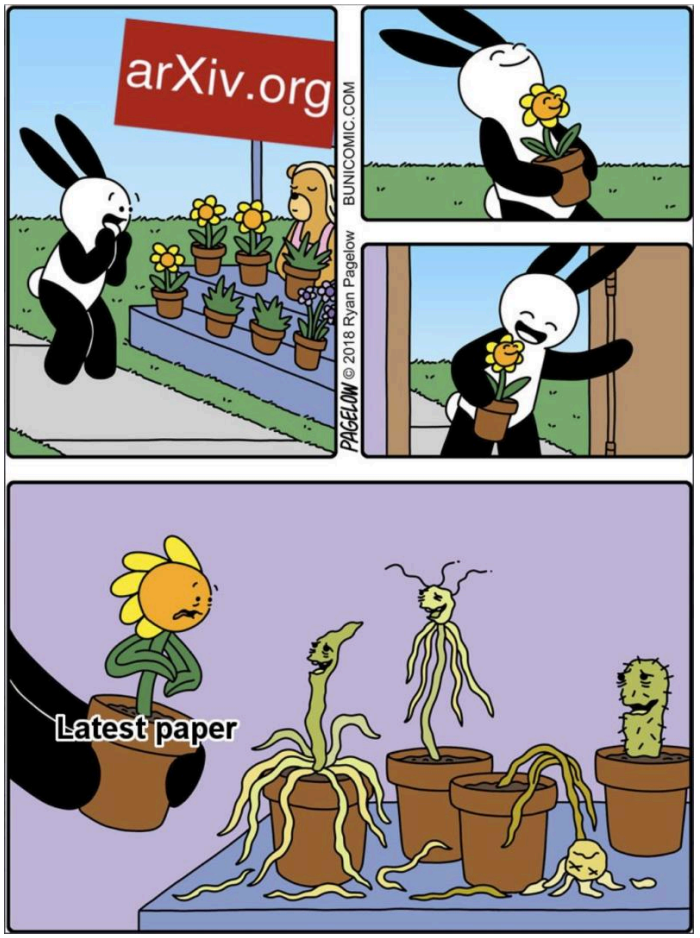
- **Brown's CSCI1290:** http://cs.brown.edu/courses/csci1290

Traditional Computer Vision:

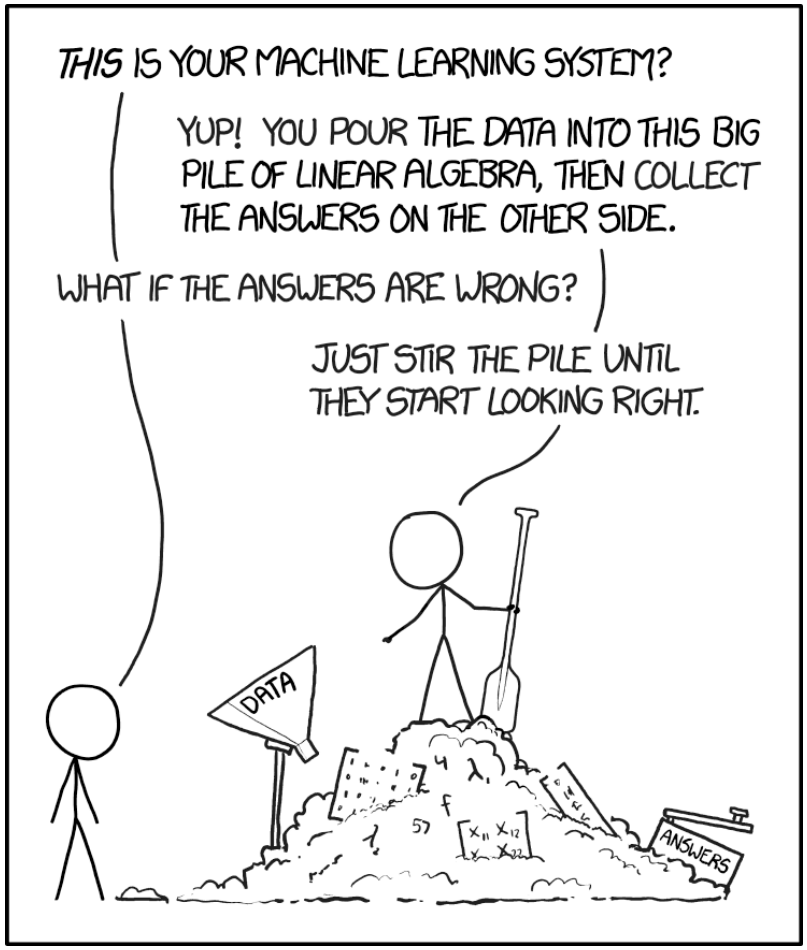
- **Brown's CSCI1430:** http://cs.brown.edu/courses/csci1430/
- **UT Dallas CS 6384**

CNN:

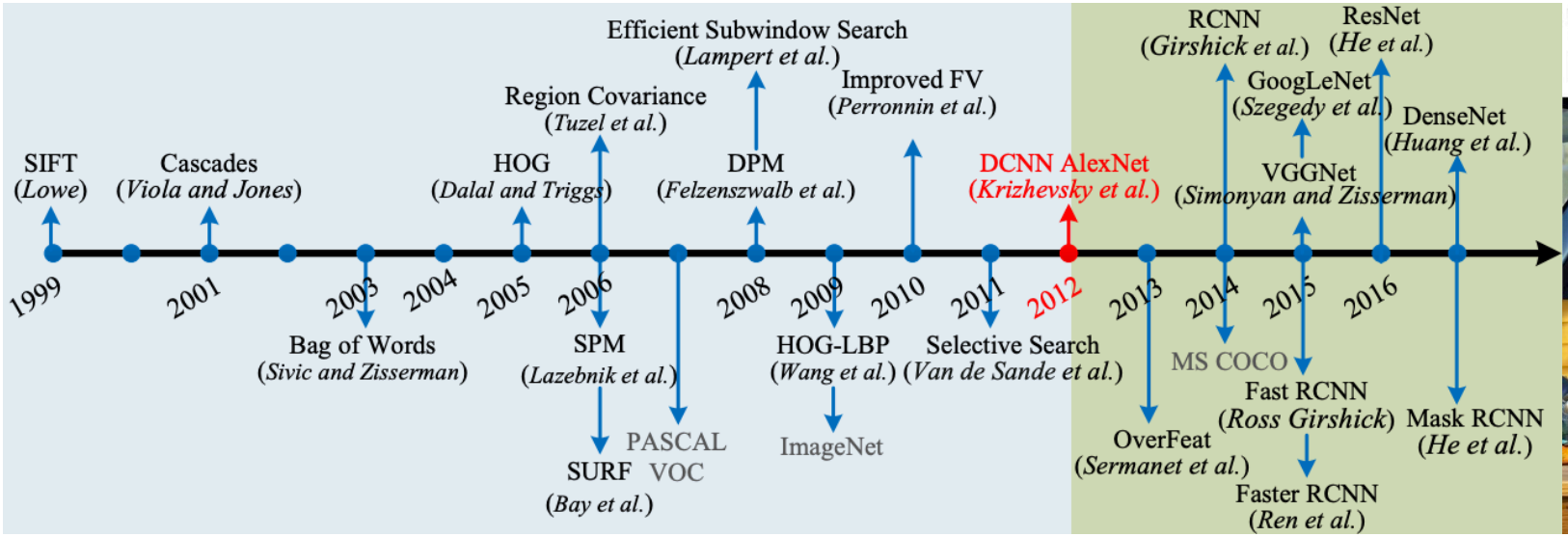
- **Excellent Survey Paper:** https://arxiv.org/pdf/1809.02165.pdf
- **Dr. Redfern's CS 6301:** https://github.com/arthurredfern/UT-Dallas-CS-6301-CNNs (especially Lecture 8)
- **Stanford's CS231n:** http://cs231n.stanford.edu/ (Specifically Lecture 12)



THANKS

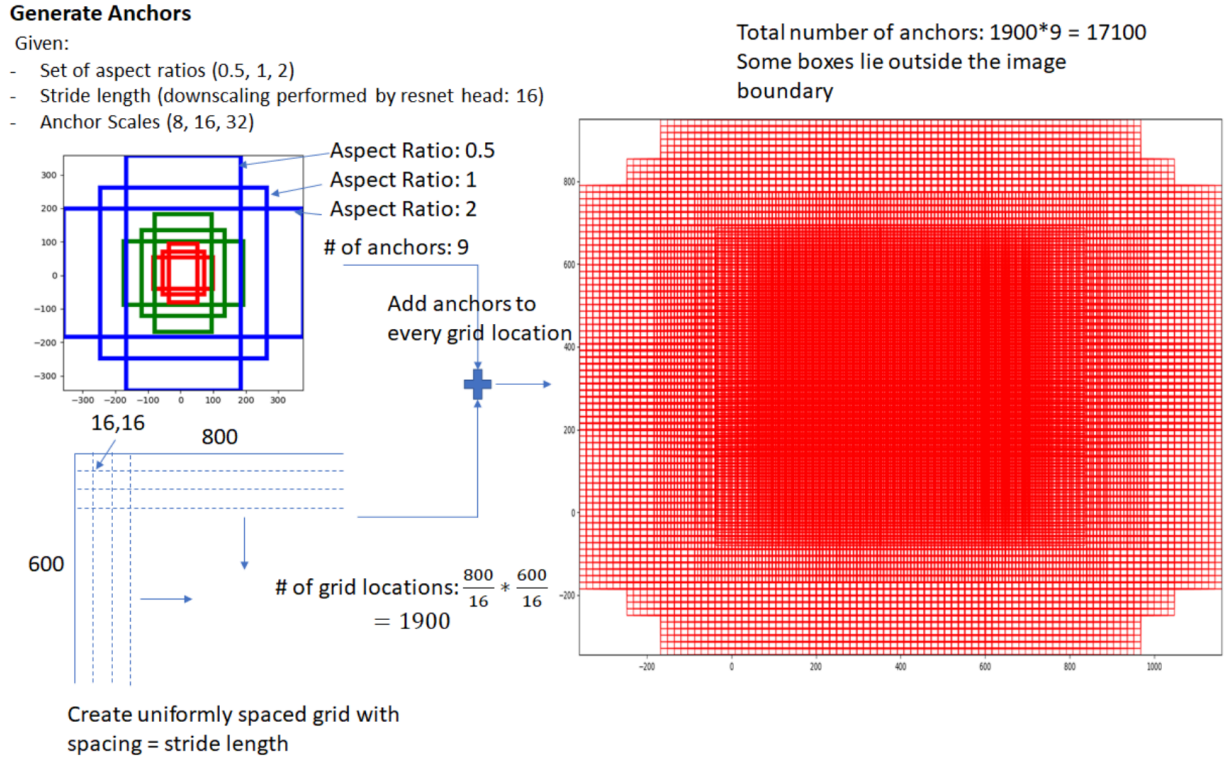

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

APPENDIX: A BRIEF HISTORY



Source: Fig 4 from https://arxiv.org/pdf/1809.02165.pdf

APPENDIX: ANCHOR BOXES



Source and further information: http://www.telesens.co/2018/03/11/object-detection-and-classification-using-r-cnns/

APPENDIX: NON-MAXIMUM SUPPRESSION

Algorithm 1 Non-Max Suppression

```
1:  procedure NMS(B,c)
2:      B_nms ← ∅
3:      for b_i ∈ B do
4:          discard ← False
5:          for b_j ∈ B do
6:              if same(b_i, b_j) > λ_nms then
7:                  if score(c, b_j) > score(c, b_i) then
8:                      discard ← True
9:          if not discard then
10:             B_nms ← B_nms ∪ b_i
11:     return B_nms
```

Greedy algorithm: https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c
Figure: https://arxiv.org/pdf/1705.02950.pdf