Advanced Section 2: Semantic Segmentation and Object Detection

CS109B Advanced Topics in Data Science

Robbert Struyven

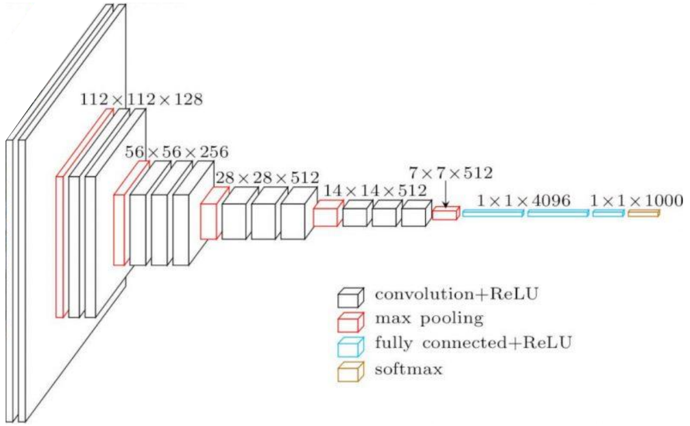
Pavlos Protopapas, Mark Glickman and Chris Tanner

Outline: Semantic Segmentation and Object Detection

- **Tasks**
  - Image Classification
  - Classification + Localization
  - Object Detection
  - Semantic Segmentation
- Object Detection: let's classify and locate
  - Sliding Window versus Region Proposals
  - Two stage detectors: the evolution of R-CNN , Fast R-CNN, Faster R-CNN
  - Single stage detectors: detection without Region Proposals: YOLO / SSD
- Semantic segmentation: classify every pixel
  - Fully-Convolutional Networks
  - SegNet & U-NET
  - Faster R-CNN linked to Semantic Segmentation: Mask R-CNN
- Code: Using Transfer-Learning to train a U-NET

Tasks: Image Classification: Fully-Connected CNN

- Fundamental to computer vision given a set of labels {dog, cat, human, ...}
- Predict the most likely class

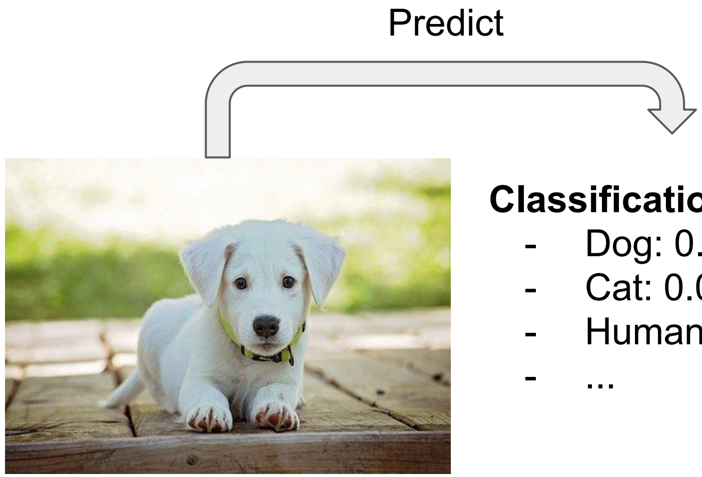


Classification output (C = 1000):

- Dog: 0.95
- Cat: 0.02
- Human: 0.01
- ...

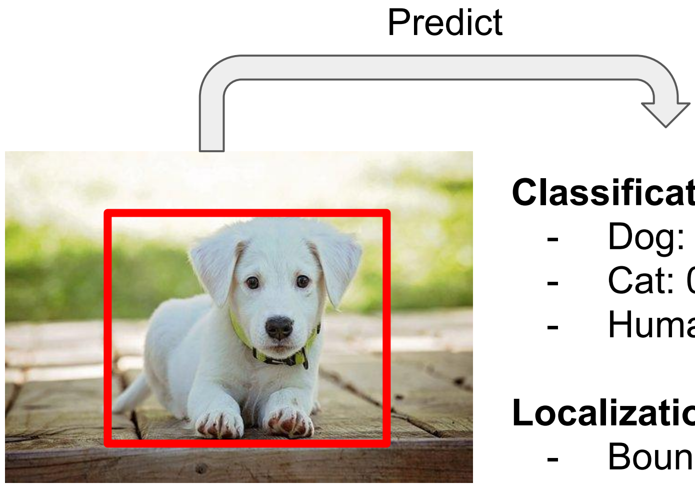
Tasks: From Image Classification to **Classification + Localization**

- Localization demands to compute **where 1 object is present in an image**
- Limitation: only 1 object (also non-overlapping)
- Typically implemented using a bounding box (x, y, w, h)



Classification output:
- Dog: 0.95
- Cat: 0.02
- Human: 0.01
- ...

Regular Image Classification

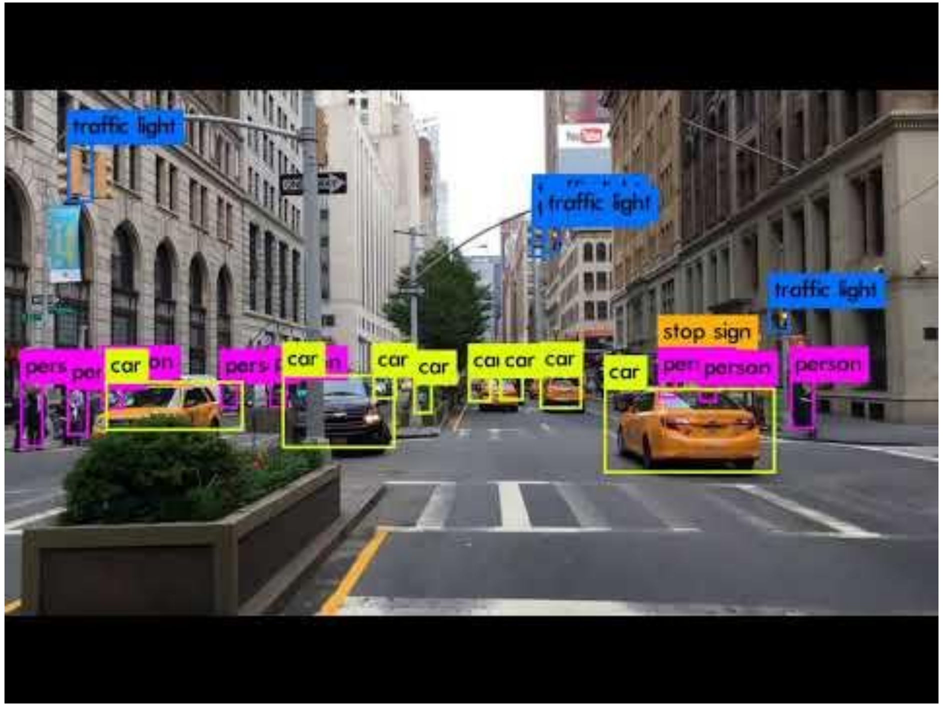


Classification output:
- Dog: 0.95
- Cat: 0.02
- Human: 0.01

Localization output:
- Bounding-Box: (x, y, w, h)

Tasks: From Classification + Localization to **Object Detection**

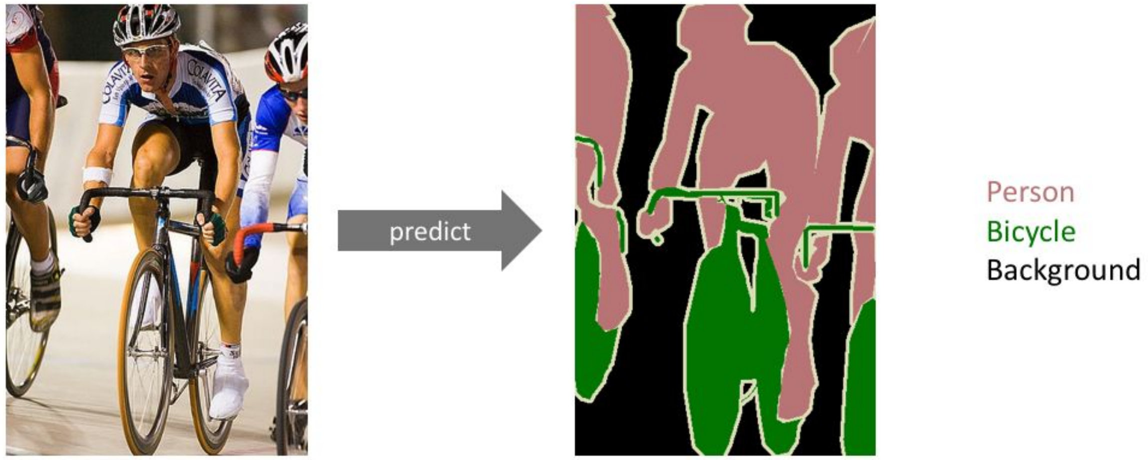
- Classification and Localization extended to multiple objects



Youtube 'YOLO in New York" by Joseph Redmon (creator of YOLO)

Tasks: From Classification to **Semantic Segmentation**

- **Image Classification:** assigning a single label to **the entire picture**
- **Semantic segmentation:** assigning a semantically meaningful label to **every pixel in the image**



Long, Shelhamer et al. "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015 : Cited by 14480

Why Object Detection and Semantic Segmentation

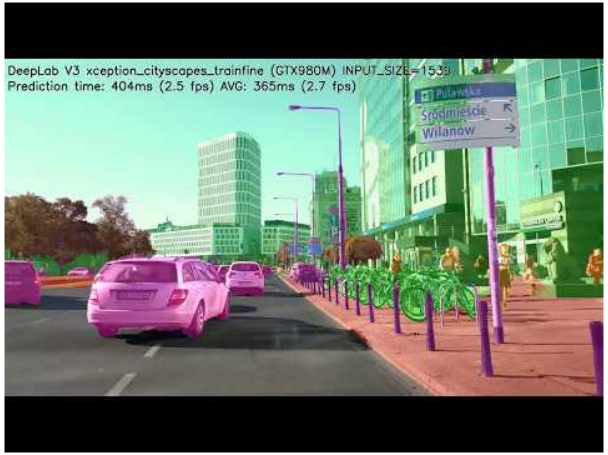
Computer vision:

- Autonomous vehicles
- Biomedical Imaging detecting cancer, diseases, ...
- Video surveillance:
  - Counting people
  - Tracking people
- Aerial surveillance
- Geo Sensing: tracking wildfire, glaciers, .. via satelite

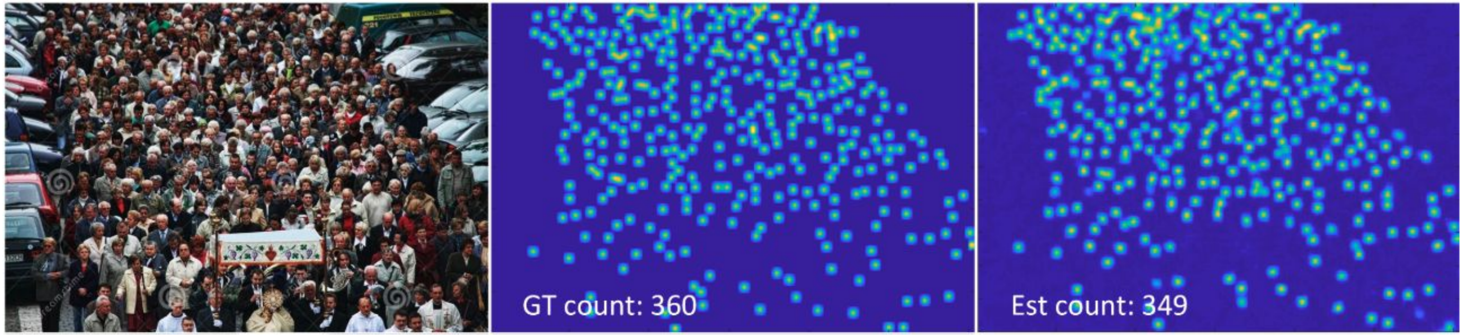
…

Note:

- Efficiency/inference-time is important!
- How many frames/sec can we predict?
- Must for real-time segmentation & detection.


DeepLab V3 xception_cityscapes_trainfine (GTX980M) INPUT_SIZE=1539
Prediction time: 404ms (2.5 fps) AVG: 365ms (2.7 fps)

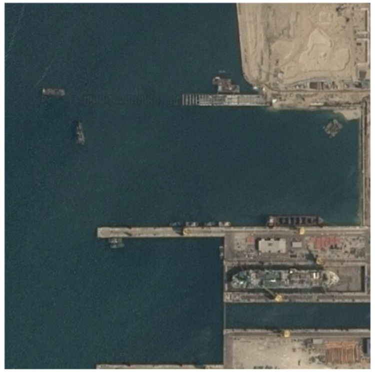
Youtube: "Tensorflow DeepLab v3 Xception Cityscapes"(link)

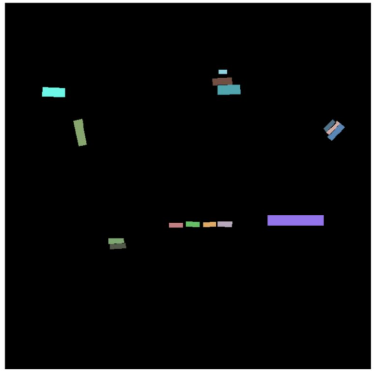


How to measure quality in detection and segmentation?

- **Pixel Accuracy:**
  - Percent of pixels in your image that are classified correctly
  - Our model has 95% accuracy! Great!

Input **Ground Truth** **Prediction**





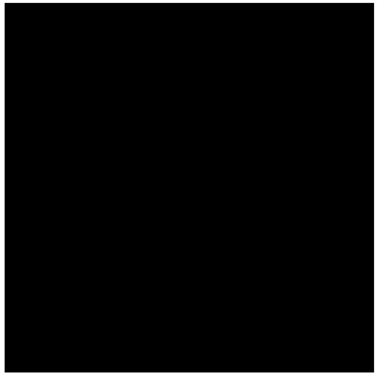
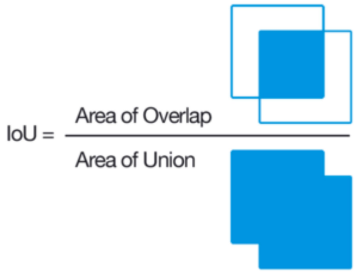


Image from Vlad Shmyhlo in article: Image Segmentation: Kaggle experience (Part 1 of 2) in TDS

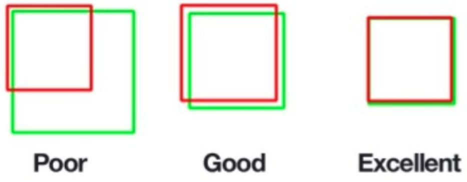
- Problem with accuracy: unbalanced data!

How do we measure accuracy?

- **Pixel Accuracy**: Percent of pixels in your image that are classified correctly
- **IOU:** Intersection-Over-Union (Jaccard Index): Overlap / Union

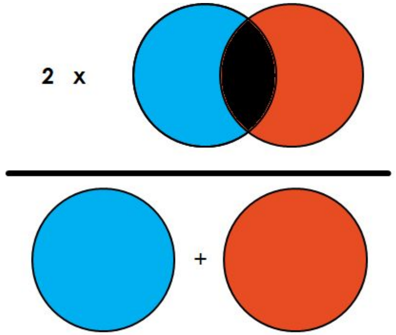


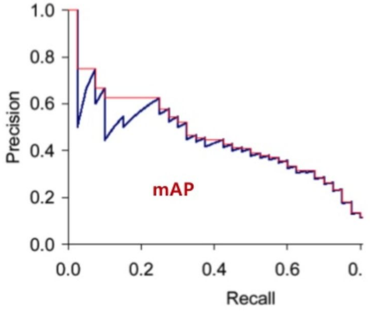
IoU calculation visualized. Source: Wikipedia



IoU: 0.40 IoU: 0.73 IoU: 0.92

- **DICE:** Coefficient (F1 Score): 2 x Overlap / Total number of pixels
- **mAP:** Mean Average Precision: AUC of Precision-Recall curve
- standard in literature, 0.5 is considerd high!)



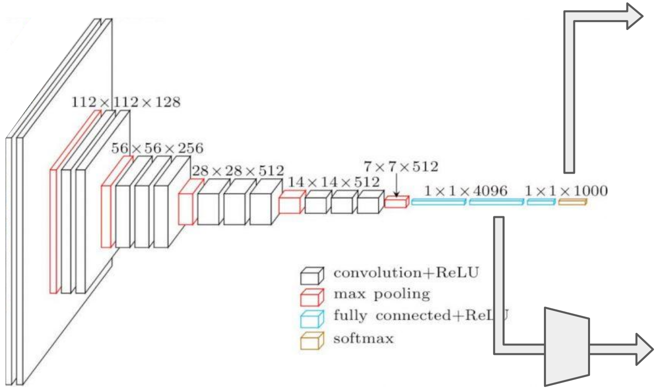


Outline: Semantic Segmentation and Object Detection

- Tasks
  - Image Classification
  - Classification + Localization
  - Object Detection
  - Semantic Segmentation
- **Object Detection: let’s classify and locate**
  - Sliding Window versus Region Proposals
  - Two stage detectors: the evolution of R-CNN , Fast R-CNN, Faster R-CNN
  - Single stage detectors: detection without Region Proposals: YOLO / SSD
- Semantic segmentation: classify every pixel
  - Fully-Convolutional Networks
  - SegNet & U-NET
  - Faster R-CNN linked to Semantic Segmentation: Mask R-CNN
- Code: Using Transfer-Learning to train a U-NET

Object detection: let’s classify and locate

- Object detection is just classification and localization combined
  - Classification: Using standard CNN
  - Localization: Regression problem for predicting box coordinates
  - **Combined loss:** Classification (Softmax) + Regression (L2)



Classification output:
Dog: 0.95
Cat: 0.02
Human: 0.01

Softmax Loss

Multi-Task Learning

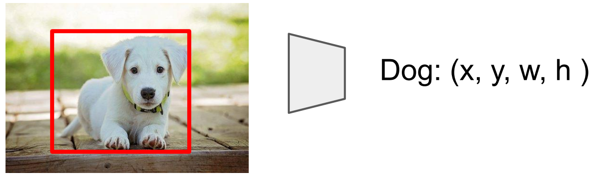
Localization output:
Bounding-Box:
(x, y, w, h)
(x, y, width, height)

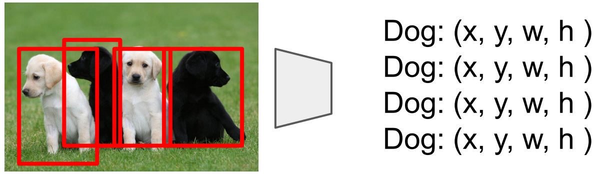
L2-norm Loss

Adopted from Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019 “Convolutional Neural Networks for Visual Recognition” Lecture 12 Slide 37

Object detection: From single to multiple objects: **Sliding Windows?**

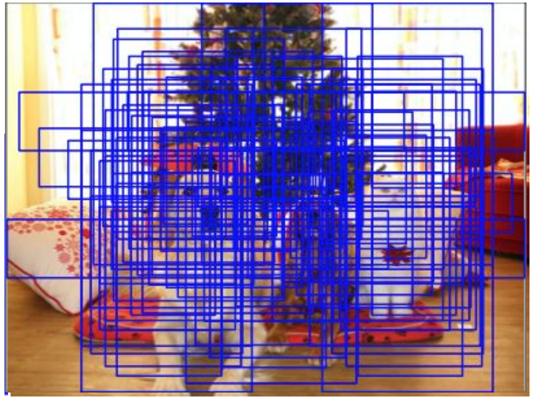
- Might work for single object, but not for multiple objects:
- Each image containing "*x*" objects :
  needs "*x*" number of classification and localization outputs.
- Solution for multiple objects:
  - Crop the image "in a smart way"
  - Apply the CNN to each crop





- Can we just **use sliding windows?**

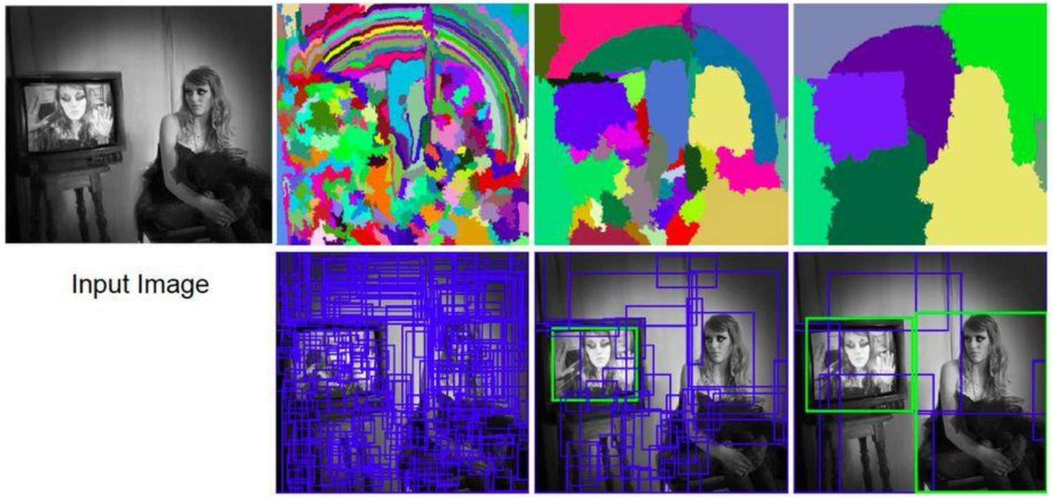
- **Problem:** Need for applying CNN to huge number of locations, scales, bbox aspect ratios: very computationally expensive!
- **Solution:** Region Proposals methods to find object-like regions.



Adopted from Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019
"Convolutional Neural Networks for Visual Recognition" Lecture 12 Slide 37
Uijlings et al, Selective Search for Object Recognition" IJCV 2013 link

Object detection: Multiple objects? Region Proposal Networks!

- **Problem:** Need for applying CNN to huge number of locations, scales, bbox aspect ratios, very computationally expensive!
- **Solution:** Region Proposals methods to find object-like regions:
- **"Selective Search Algorithm:** returns boxes that are likely to contain objects
  - Use hierarchical segmentation
  - Start with small superpixels
  - Merge based on similarity

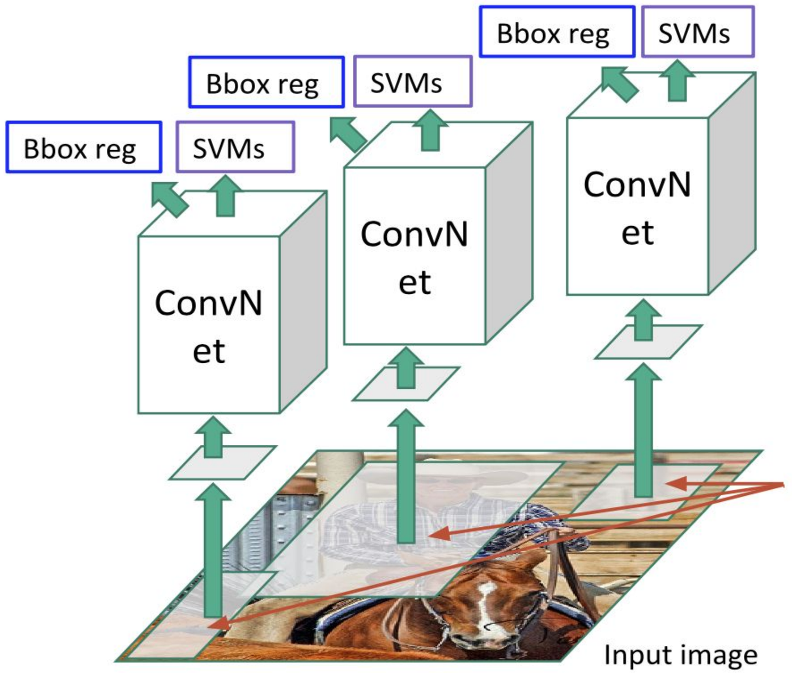


Output: Where are object like regions?

No classification yet.

Uijlings et al, Selective Search for Object Recognition" IJCV 2013 link

The evolution of R-CNN: **R-CNN** , Fast R-CNN, Faster R-CNN

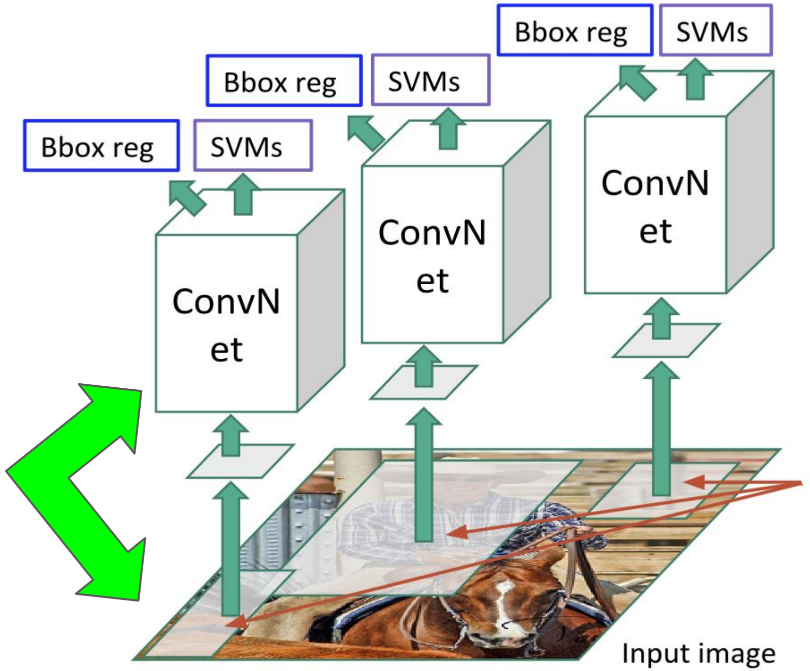


R-CNN = Region-based CNN

- Correct BBox by Bbox regressor (dx,dy,dw,dh)
- Classify regions with SVMs
- Forward each region through CNN
- Resize proposed RoI (224x224)
- Region of Interest (RoI) from selective search region proposal (approx 2k)
- **Problem:** need to do 2k independent forward passes for each image! **('slow' R-CNN)**

Adopted from Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019 "Convolutional Neural Networks for Visual Recognition" Lecture 12 Slide 37
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation" CVPR2014
Ross Girshick, "Fast R-CNN" Slides 2015

The evolution of R-CNN: R-CNN , **Fast R-CNN**, Faster R-CNN



R-CNN = Region-based CNN

- Correct BBox by Bbox regressor (dx,dy,dw,dh)
- Classify regions with SVMs
- Forward each region through CNN
- Resize proposed RoI (224x224)
- Region of Interest (RoI) from selective search region proposal (approx 2k)

Problem: need to do 2k independent forward passes for each image! **('slow' R-CNN)**

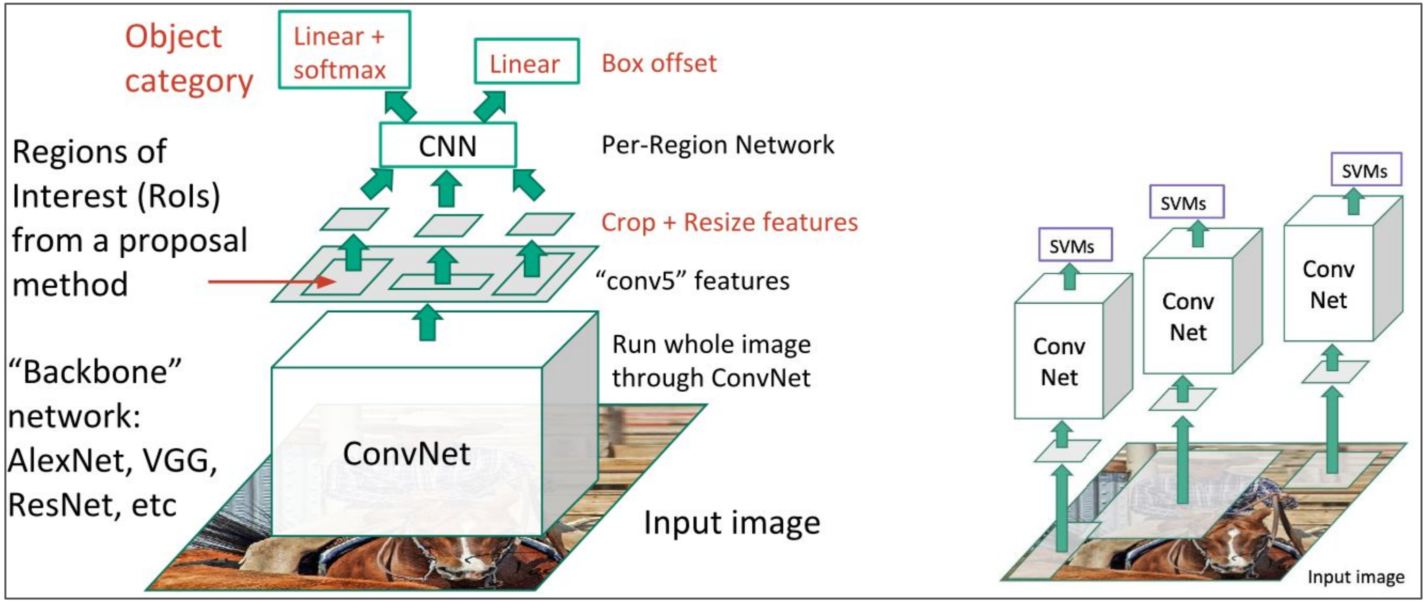
Solution: can we process the image before cropping?

Adopted from Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019 "Convolutional Neural Networks for Visual Recognition" Lecture 12 Slide 37
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation" CVPR2014
Ross Girshick, "Fast R-CNN" Slides 2015

The evolution of R-CNN: R-CNN, **Fast R-CNN**, Faster R-CNN

- **Problem:** need to do 2k independent forward passes for each image! **('slow' R-CNN)**
- Even inference is slow: 47s/image with VGG16 [Simonyan & Zisserman, ICLR 15]
- **Solution:** can we process (CNN forward pass) the image before cropping generates 2k regions?

Fast R-CNN

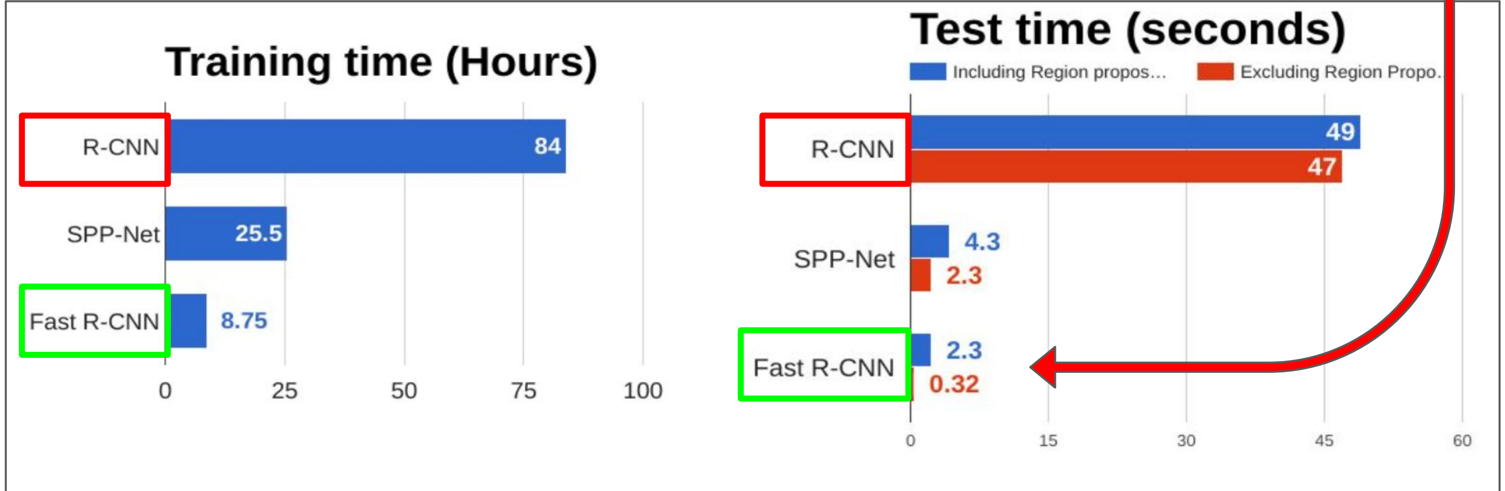


Slow R-CNN

Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019 "Convolutional Neural Networks for Visual Recognition" Lecture 12 Slide 62
Ross Girshick, "Fast R-CNN" Slides 2015≈

The evolution of R-CNN: R-CNN , Fast R-CNN, **Faster R-CNN**

- Fast R-CNN much faster than R-CNN
- Runtime dominated by region proposals; is a iterative method ('like selective search')
- **Solution:** can we Make CNN do proposals! **Fast R-CNN** $\Rightarrow$ **Faster R-CNN**

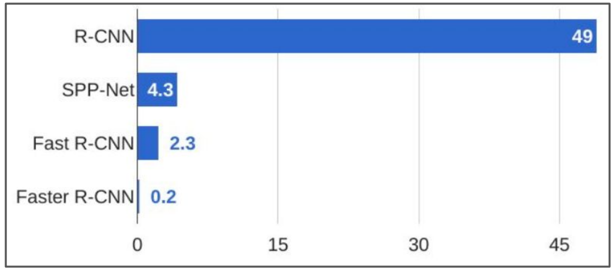


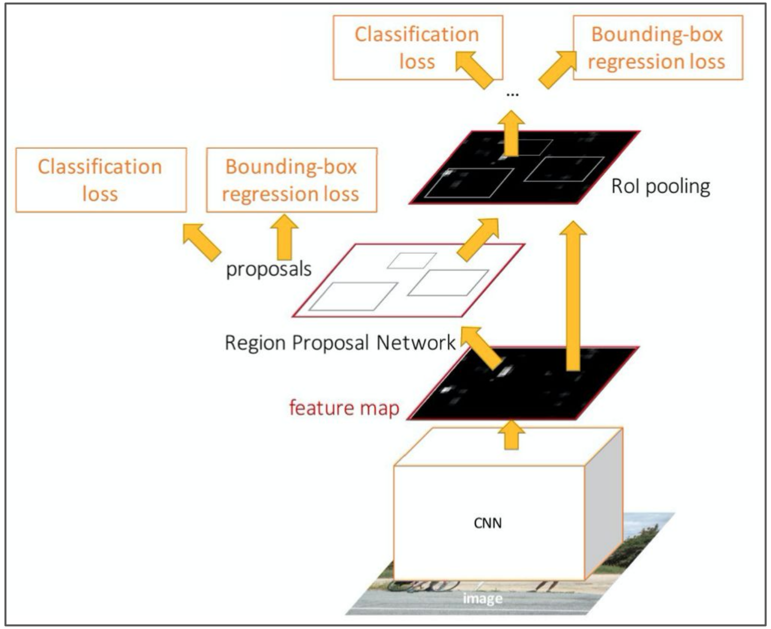
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019 "Convolutional Neural Networks for Visual Recognition" Lecture 12 Slide 76
Ross Girshick, "Fast R-CNN" Slides 2015

The evolution of R-CNN: R-CNN , Fast R-CNN, **Faster R-CNN**

- **Faster R-CNN**: Make CNN to do proposals! (single forward, not iterative selective search)
- **CNN Region Proposal Network (RPN)**: predicting region proposals from features
- Otherwise same as Fast R-CNN: crop and classify
- End-to-end quadruple loss:
  - RPN classify object / not object
  - RPN regress box coordinates
  - Final classification score (object classes)
  - Final box coordinates
- Test-time seconds per image:

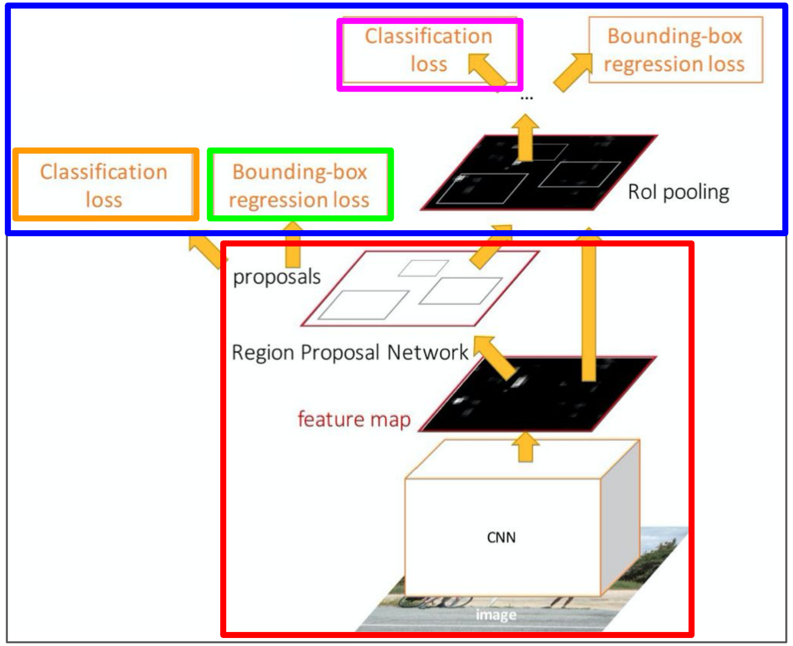




Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019 "Convolutional Neural Networks for Visual Recognition" Lecture 12 Slide 77, 84, and 85
Ross Girshick, "Fast R-CNN" Slides 2015

Two-Stage object detector

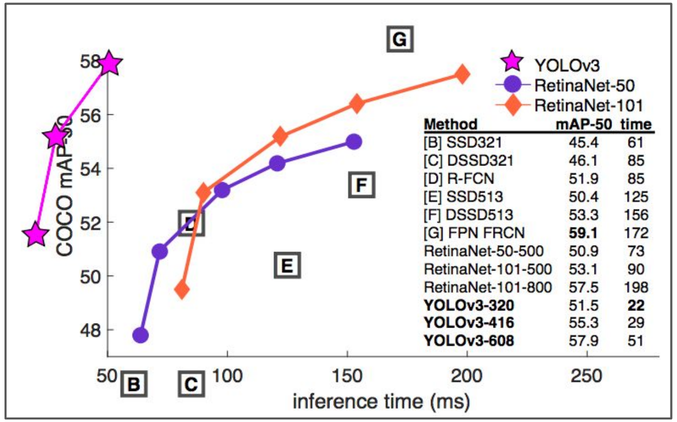
- Previously we said: “Multiple objects? Thus Need for Region Proposal Networks!”
- **Faster R-CNN is a two-stage object detector**
  a. Stage 1: Backbone network + RPN (once / image)
  b. Stage 2: crop - predict object & bbox (once / region)
- What is our RPN again?
- RPN runs prediction on many many anchor boxes
  - Loss 1: Tells is does the anchor bbox contain an object
  - Loss 2: For the top 300 boxes its adjusts the box
- What is the difference between our 2 classification losses?
  - one is detecting/classifying: object ‘yes/no’
  - one is classifying specific categories: dog ‘yes/no’ ,...
- Do we really need two stages?
- Can’t we do category-specific RPN directly?



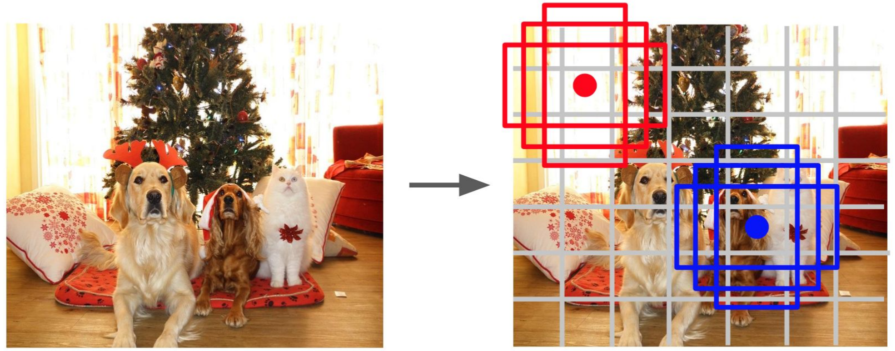
Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019 “Convolutional Neural Networks for Visual Recognition” Lecture 12 Slide 88

Single-Stage Detection without Region Proposals: **YOLO, SSD**

- Within each of the **NxN** grid regress over each **B** base boxes, predict: (dx,dy,dh,dw, confidence) = 5
- Predict **C** category specific class scores
  - **Output :** N x N x S (5 B + C)
- YOLOv3: YOU ONLY LOOK ONCE : Joseph Redmon
  - predicts at 3 scales, S = 3
  - Predicts 3 boxes at each scale, B=3
  - Darknet-53 as feature extractor (similar to ResNet 152, and 2x faster!)



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



Input image
3 x H x W

Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016
Lin et al, "Focal Loss for Dense Object Detection", ICCV 2017

Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019 "Convolutional Neural Networks for Visual Recognition" Lecture 12 Slide 89

(YOLO) Redmon, "You Only Look Once: Unified, Real-Time Object Detection" CVPR 2015: Cited by 8057 (link)

109a versus 109b

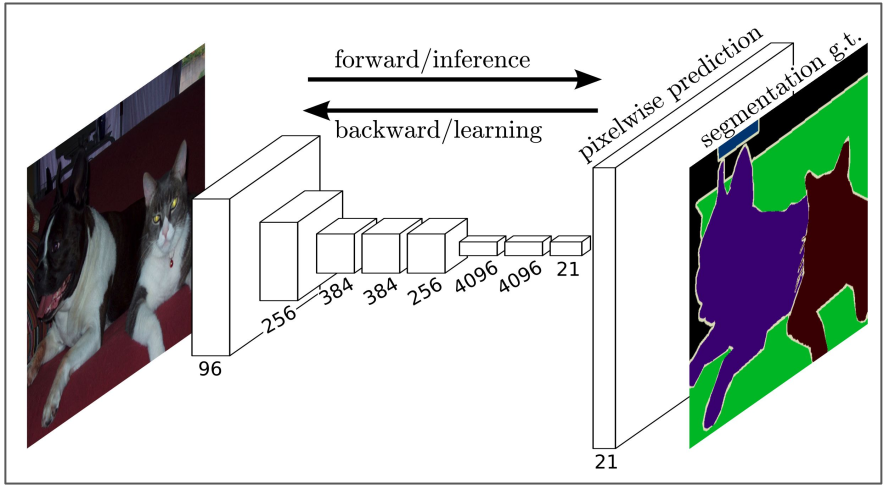
Outline: Semantic Segmentation and Object Detection

- Tasks
  - Image Classification
  - Classification + Localization
  - Object Detection
  - Semantic Segmentation
- Object Detection: let’s classify and locate
  - Sliding Window versus Region Proposals
  - Two stage detectors: the evolution of R-CNN , Fast R-CNN, Faster R-CNN
  - Single stage detectors: detection without Region Proposals: YOLO / SSD
- **Semantic segmentation: classify every pixel**
  - Fully-Convolutional Networks
  - SegNet & U-NET
  - Faster R-CNN linked to Semantic Segmentation: Mask R-CNN
- Code: Using Transfer-Learning to train a U-NET

Semantic segmentation: Classify every pixel

- **Image Classification:** assigning a single label to **the entire picture**
- **Semantic segmentation:** assigning a semantically meaningful **label to every pixel in the image**
  - So our output shouldn't be a class prediction (C numbers) but a picture (C x *w x h*)
    - Maybe we can have a network for each pixel location? Many (*w* times *h)* networks !
    - Sliding window inputs of patches predicting the class of the pixel in the center?

      Many forward passes! Not reusing overlapping patches and features.



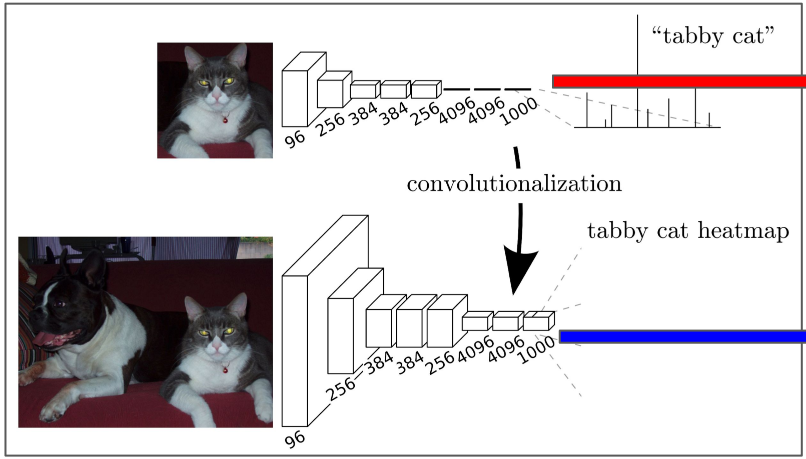
(FCN) Long, Shelhamer et al. "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015: Cited by 14480 (link)

Fully-Convolutional Networks

- **Semantic segmentation:** assigning a semantically meaningful label to **every pixel in the image**
  - So our output shouldn't be a classification prediction **(C numbers)** but a picture **(C x *w* x *h*)**
    - Maybe we can have a network for each pixel location? Many (*w* times *h)* networks !
    - Sliding window inputs of patches predicting the class of the pixel in the center? Many forward passes! Overlapping features not used.
- **Solution: FCN = Fully-Convolutional Networks! (not fully-connected)** (abbr. confusing!)
    - 1 network - 1 prediction would be a lot better
    - Why convolutions? every pixel is very much influenced by its neighbourhood

Image Classification

Image Segmentation



fully-connected layers

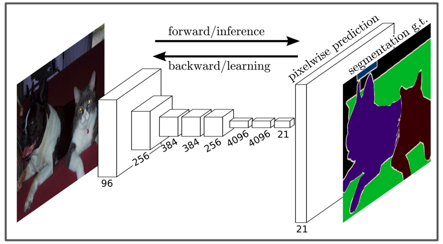
"convolutionalised"

fully convolutional layers

(FCN) Long, Shelhamer et al. "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015: Cited by 14480 (link)

Fully-Convolutional Networks

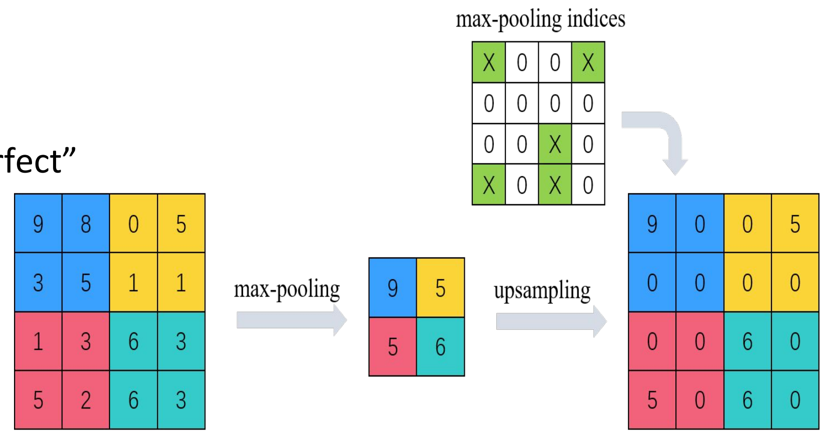
- **FCN:** design a network as a bunch of conv layers to make predictions for all pixels all at once.
    - **Encoder** (= Localization)**: downsample** through **convolutions**
        - reduces number of params (bottleneck), can make network deeper
    - **Decoder** (= Segmentation): **upsampled** through **transposed convolutions**
    - **Loss:** cross-entropy loss on every pixel
- **Contribution:**
    - Popularize the use of end-to-end CNNs for semantic segmentation
    - Re-purpose imagenet pretrained networks for segmentation = Transfer Learning
    - Upsample using transposed layers
- **Negative:** upsampling = loss of information during pooling
    - <u>224x224</u> image downsampled to <u>20x20</u> back upsampled to <u>224x224</u>



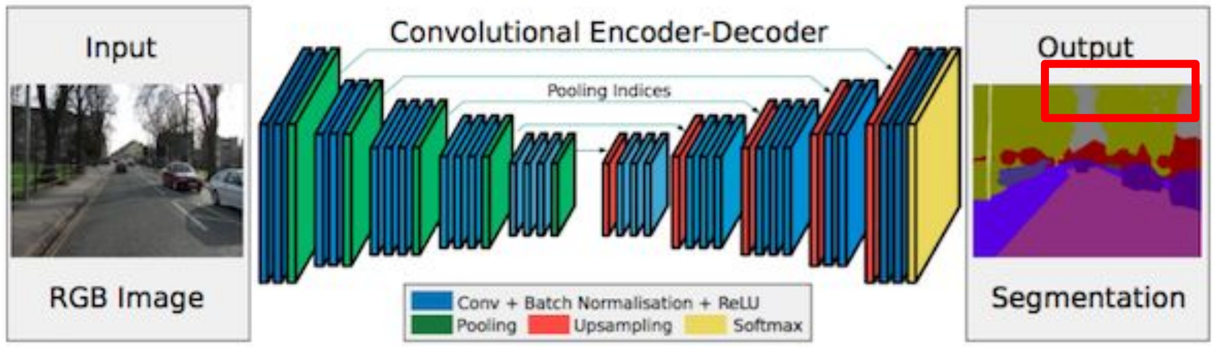
(FCN) Long, Shelhamer et al. "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015: Cited by 14480 (link)

Segnet: pooling indices

- The indices from max pooling downsampling are transferred to the decoder: **pooling indices**
- Improves fine segmentation resolution, we want “pixel-perfect”
- More efficient since no transposed convolutions to learn

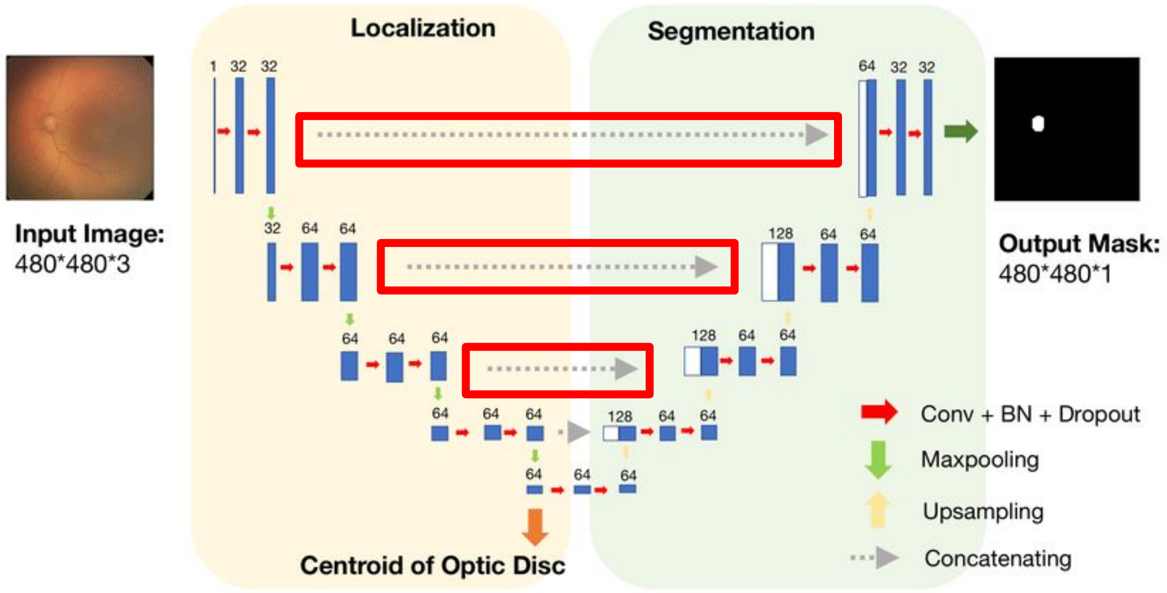


SegNet: A deep Convolutional Encoder-Decoder Architecture for Image Segmentation. (link)



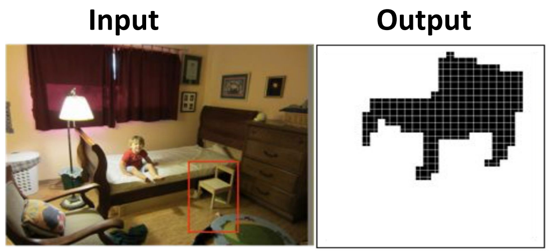
U-NET: long skip connections

- The U-Net is a encoder decoder using:
  - **location information** from the **downsampling** path of the **encoder**
  - **contextual information** in the **upsampling** path by the **“concatenating” long-skip connections**

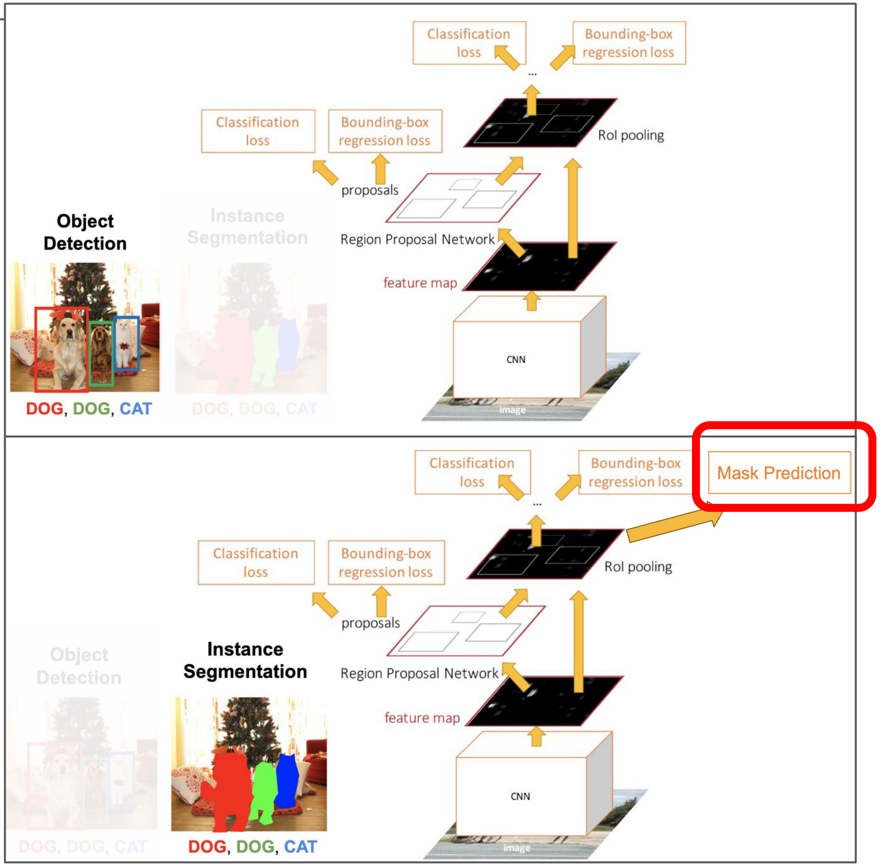


Faster R-CNN linked to Semantic Segmentation: Mask R-CNN

- **Object Detection:**
  - E.g. Faster R-CNN or YOLO
  - classification and Localization **of every object**
- **Semantic segmentation:**
  - assigning a semantically meaningful label to **every pixel in the image**
  - We can use the bbox prediction of R-CNN
  - Train a network to get a pixel mask on each RoI
  - Output binary 28x28 mask:
    - Faster R-CNN -> **Mask R-CNN**



Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019
“Convolutional Neural Networks for Visual Recognition” Lecture 12 Slide 99



Outline: Semantic Segmentation and Object Detection

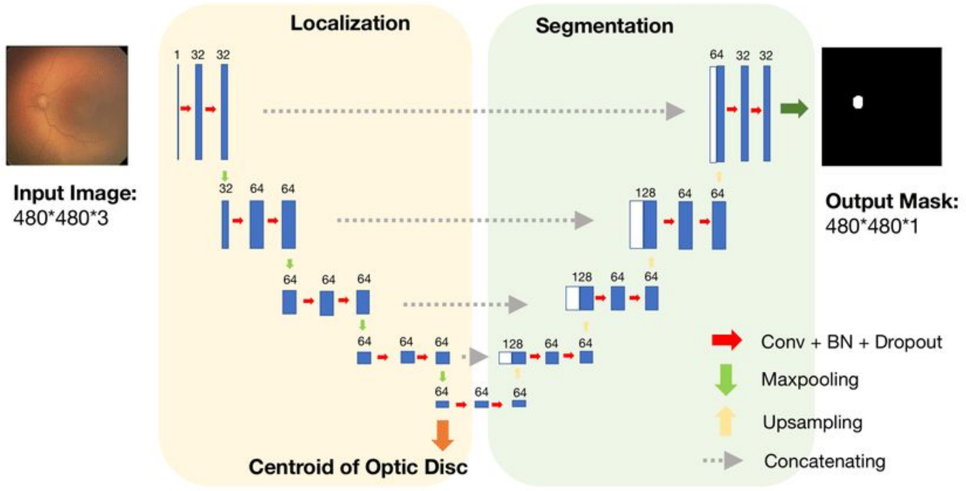
- Tasks
  - Image Classification
  - Classification + Localization
  - Object Detection
  - Semantic Segmentation
- Object Detection: let’s classify and locate
  - Sliding Window versus Region Proposals
  - Two stage detectors: the evolution of R-CNN , Fast R-CNN, Faster R-CNN
  - Single stage detectors: detection without Region Proposals: YOLO / SSD
- Semantic segmentation: classify every pixel
  - Fully-Convolutional Networks
  - SegNet & U-NET
  - Faster R-CNN linked to Semantic Segmentation: Mask R-CNN
- **Code: Using Transfer-Learning to train a U-NET**

Code: Using Transfer-Learning to train a U-NET

Can we train a segmentation U-Net on only 500 images in 4 minutes on colab!?

YES: Transfer Learning!

- Use a MobileNet classification network to help the segmentation network to learn!
- How can we use the weights of the MobileNet in our U-Net?



Code: Using Transfer-Learning to train a U-NET

How can we use the weights of the MobileNet in our U-Net?

- MobileNet goes from images by downsampling to a 1D number / class.
- The encoder of U-Net also downsamples
- We can construct the encoder of the U-Net to have exactly the layers as some low-, mid-, and high-level layers of the mobileNet and reuse their weights

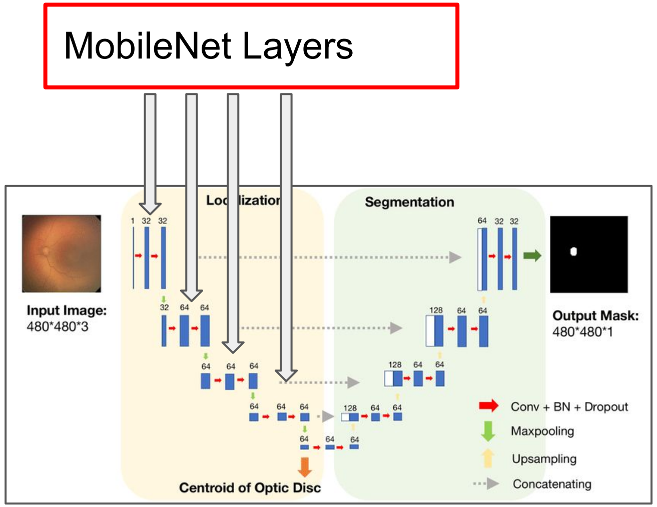
```
selected_encoder = tf.keras.applications.mobilenet_v2.MobileNetV2(
    input_shape=(INPUT_SPATIAL, INPUT_SPATIAL, 3),
    include_top=False,
    alpha=1.0,
    weights='imagenet' if pretrained else None)
```

```
conv0 = selected_encoder.get_layer("expanded_conv_project").output # 112 x 112
conv1 = selected_encoder.get_layer("block_2_project").output # 56 x 56
conv2 = selected_encoder.get_layer("block_5_project").output # 28 x 28
conv3 = selected_encoder.get_layer("block_12_project").output # 14 x 14

up6 = selected_encoder.output
conv7 = up6

up8 = concatenate([UpSampling2D()(conv7), conv3], axis=-1)
conv8 = conv_block_simple(up8, 128, "conv8_1")

up9 = concatenate([UpSampling2D()(conv8), conv2], axis=-1)
conv9 = conv_block_simple(up9, 64, "conv9_1")
```



Code: Using Transfer-Learning to train a U-NET

Can we train a segmentation U-Net on only 500 images in 4 minutes on colab!?

YES: Transfer Learning! Notebook HERE

Use a MobileNet classification network to get a segmentation network to learn!

Thanks to ComputeFest **Camilo Fosco** and **Vincent Casser**

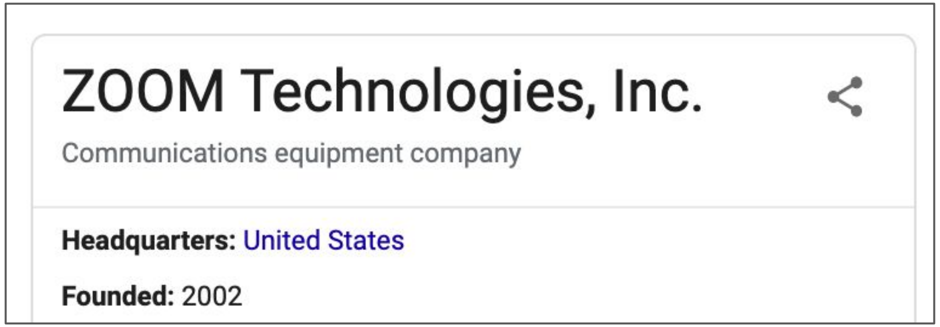
2. Train a U-Net from scratch:

- 2.3 The segmentation network architecture
- 2.4 Training the network from scratch
- 2.5 Visualize results

3. Transfer Learning to the rescue: again?

- 3.1 Re-run training with pretrained encoder weights
- 3.4 Running on webcam images

See you on Zoom!

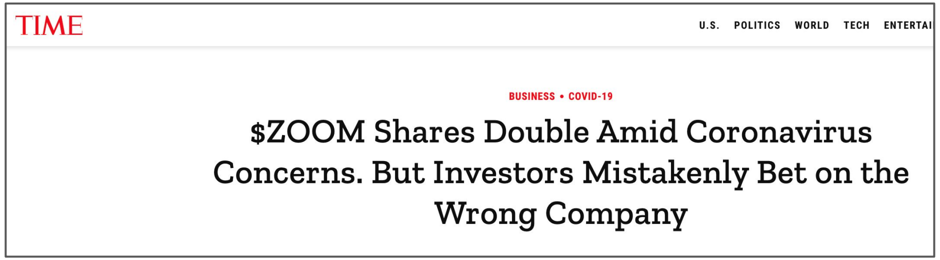


ZOOM Technologies, Inc.

Communications equipment company

Headquarters: United States

Founded: 2002



TIME

U.S. POLITICS WORLD TECH ENTERTAI

BUSINESS • COVID-19

$ZOOM Shares Double Amid Coronavirus Concerns. But Investors Mistakenly Bet on the Wrong Company

Enjoy Spring Break!

References

Presentations:

- Fei-Fei Li & Justin Johnson & Serena Yeung Stanford CS231n 2019/2018 “Conv. Neural Networks for Visual Recognition” Lecture 12 !
  - BTW: Great course / youtube series (youtube 2017)
- Ross Girshick, “Fast R-CNN” Slides 2015 (link)

Papers:

- **VGG** Simonyan, Zisserman. “Very Deep CNNs for Large-scale Image Recognition”, ILSVRC 2014: Cited by 34652 (link)
- **Select. Search** Uijlings et al, Selective Search for Object Recognition” IJCV 2013: Cited by 3944 (link)
- **R-CNN** Girshick et al, “Rich feature hierarchies for accurate object detect. & sem. segmentation” CVPR2014: Cited by 12000 (link)
- **Fast-R-CNN** Girshick, ‘Fast R-CNN“ ICCV 2015: Cited by 8791 (link)
- **Faster- R-CNN** Ren et al, “Faster R-CNN: Real-Time Object Det. with Region Proposal Networks” NEURIPS 2015 Cited by 16688 (link)
- **Mask-R-CNN** He et al, “Mask R-CNN” ICCV 2017: Cited by 5297 (link)
- **YOLO** Redmon, “You Only Look Once: Unified, Real-Time Object Detection” CVPR 2015: Cited by 8057 (link)
- **FCN** Long, Shelhamer et al. “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015: Cited by 14480 (link)
- **SegNet** Badrinarayanan et al. “SegNet: A deep Conv Encoder-Decoder Architecture for Image Segmentation”. Cited by 4258 (link)
- **U-Net** Ronneberger et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. Cited by 12238 (link)