Object Detection and Semantic Segmentation

Deep Learning
CS 435/635

Course Instructor: Chandresh
AI Lab Coordinator @IIT Indore

Course Content

Module I: History of Deep Learning, Sigmoid Neurons, Perceptrons, and learning algorithms. Multilayer Perceptrons (MLPs), Representation Power of MLPs,.

Module II: Feedforward Neural Networks. Backpropagation. first and second-order training methods. NN Training tricks, Regularization

Module III: Introduction to Autoencoders and their characteristics, relation to PCA, Regularization in autoencoders, and Types of autoencoders, Variational Autoencoder

Module IV: Architecture of Convolutional Neural Networks (CNN), types of CNNs. Image classification, Pre-training vs fine-tuning.- representation learning, **Object Detection and Semantic Segmentation**

Module V: Architecture of Recurrent Neural Networks (RNN), Backpropagation through time.
Encoder-Decoder Models, Attention Mechanism. Advanced Topics: Transformers and BERT.
Nodule VI: Gen AI- Deep generative models: VAE, GAN,

Acknowledgement

- Russ Salakhutdinov and Hugo Larochelle's class on Neural Networks
- Neural Networks and Deep Learning course by Danna Gurari University of Colorado Boulder

Today’s Topics

- Problems
- Applications
- PASCAL VOC detection challenge: R-CNNs
- PASCAL VOC semantic segmentation challenge: fully convolutional networks

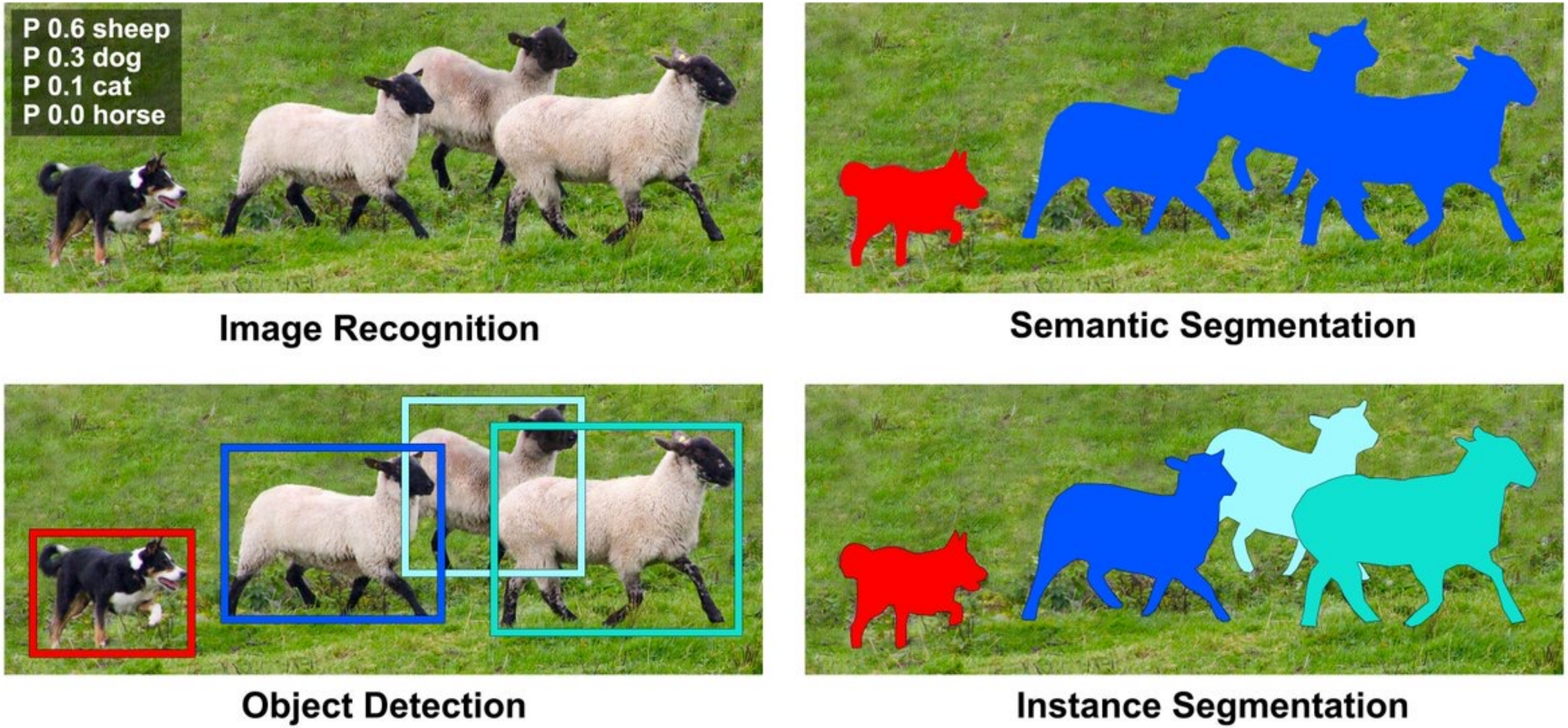
Today’s Topics

- Problems
- Applications
- PASCAL VOC detection challenge: R-CNNs
- PASCAL VOC semantic segmentation challenge: fully convolutional networks

Recall: Image Classification Task

https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

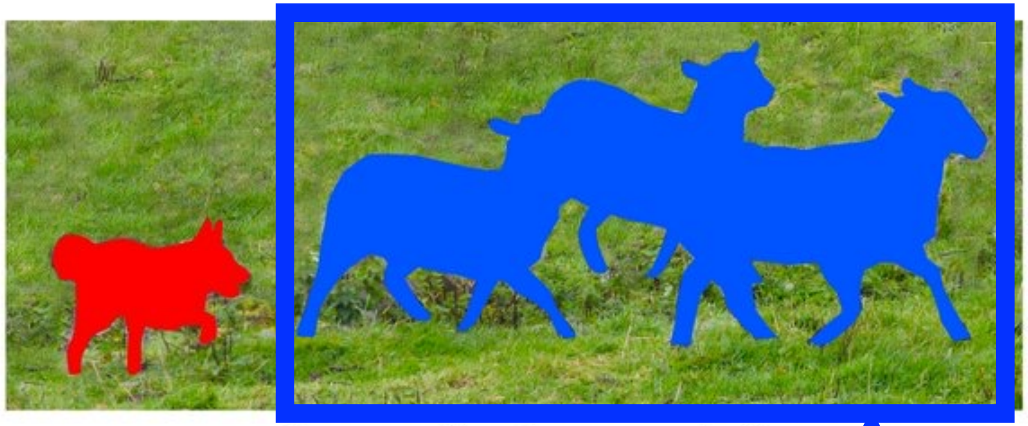
Today's Scope: Localize Content of Interest (Segmentation and Detection)



https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

Today's Scope: Localize Content of Interest (Segmentation and Detection)

Locate all pixels that belong to pre-specified categories

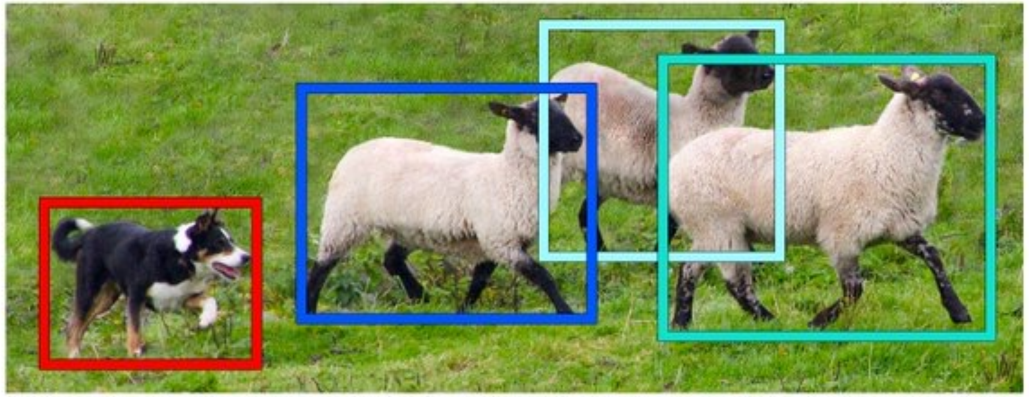


Semantic Segmentation

Note: instances of the same class are NOT separated

https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

Today's Scope: Localize Content of Interest (Segmentation and Detection)



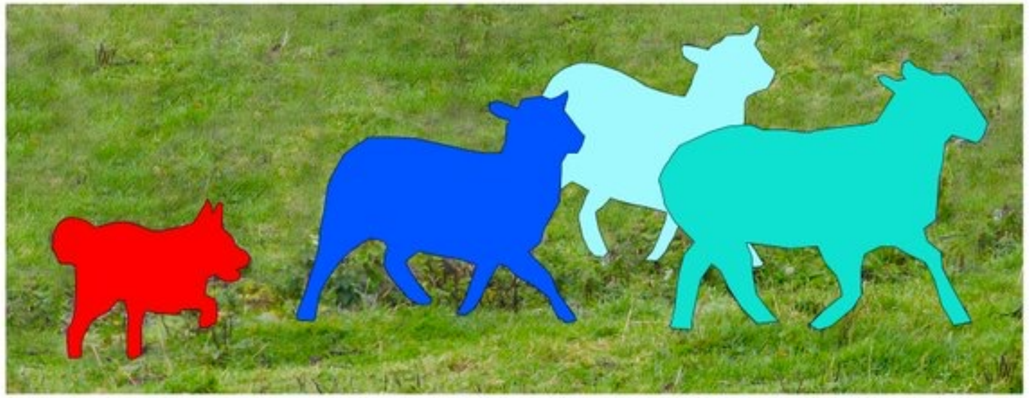
Object Detection

Use bounding boxes to locate every instance of an object from pre-specified categories

https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

Today's Scope: Localize Content of Interest (Segmentation and Detection)

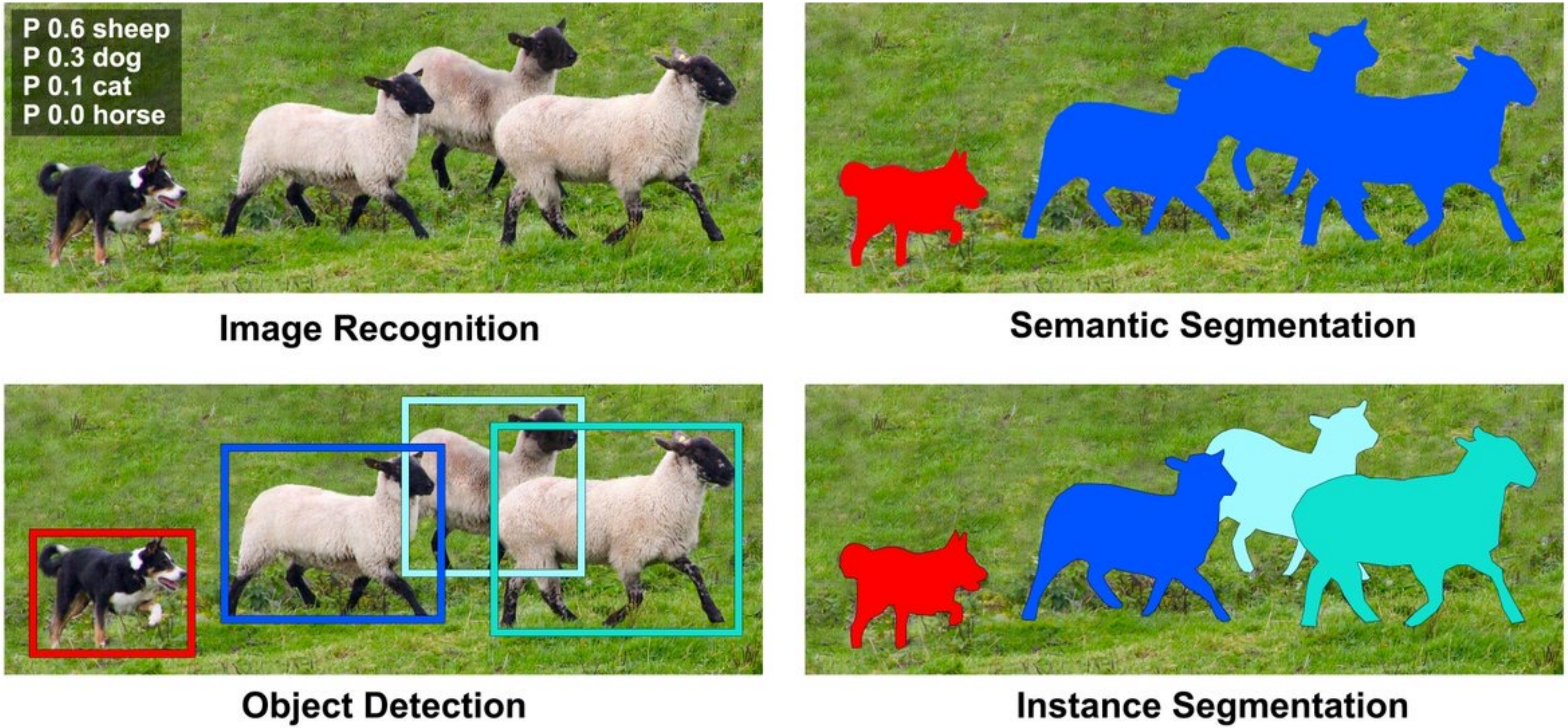
Segment every instance of objects from pre-specified categories



Instance Segmentation

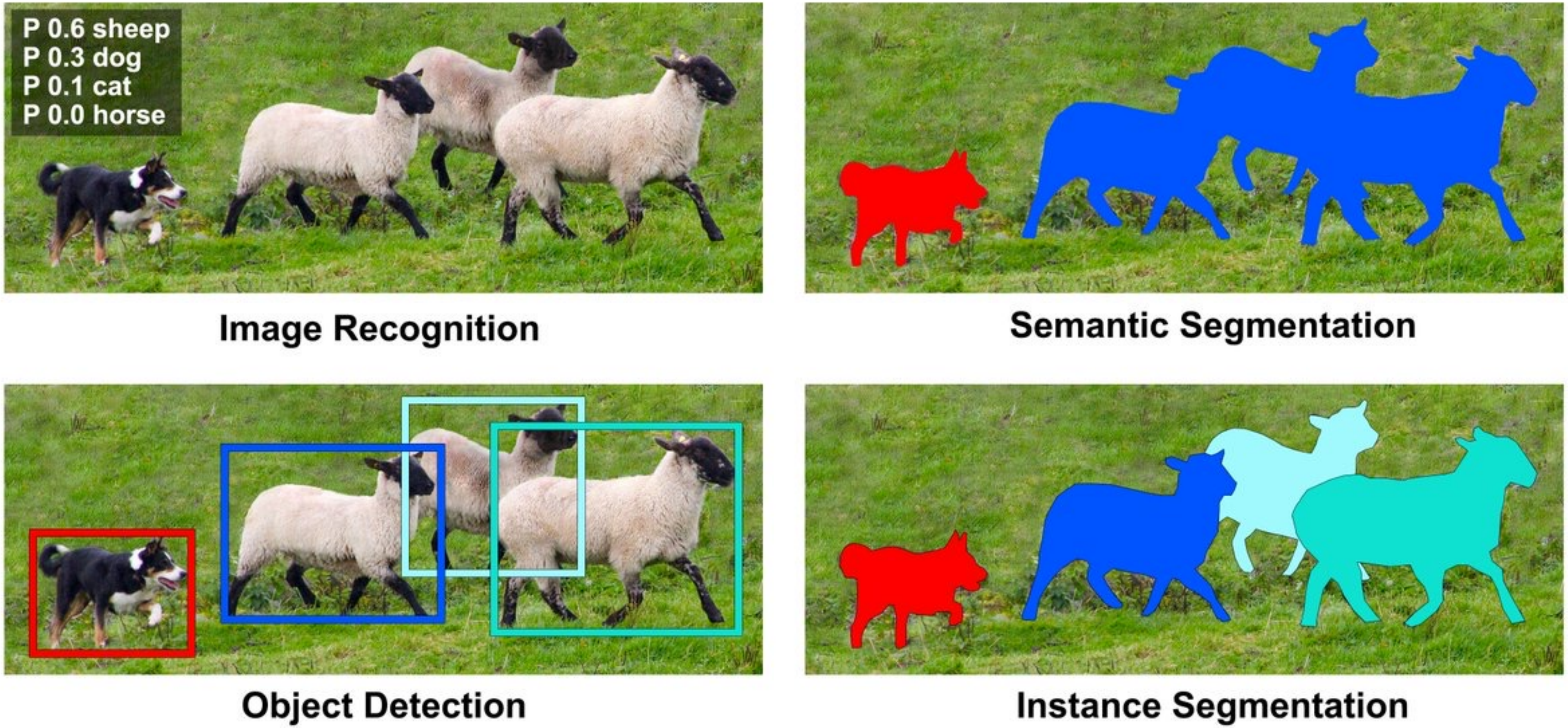
https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

Today’s Scope: Localize Content of Interest (Segmentation and Detection)



https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

Challenge: When to Choose Which Task?

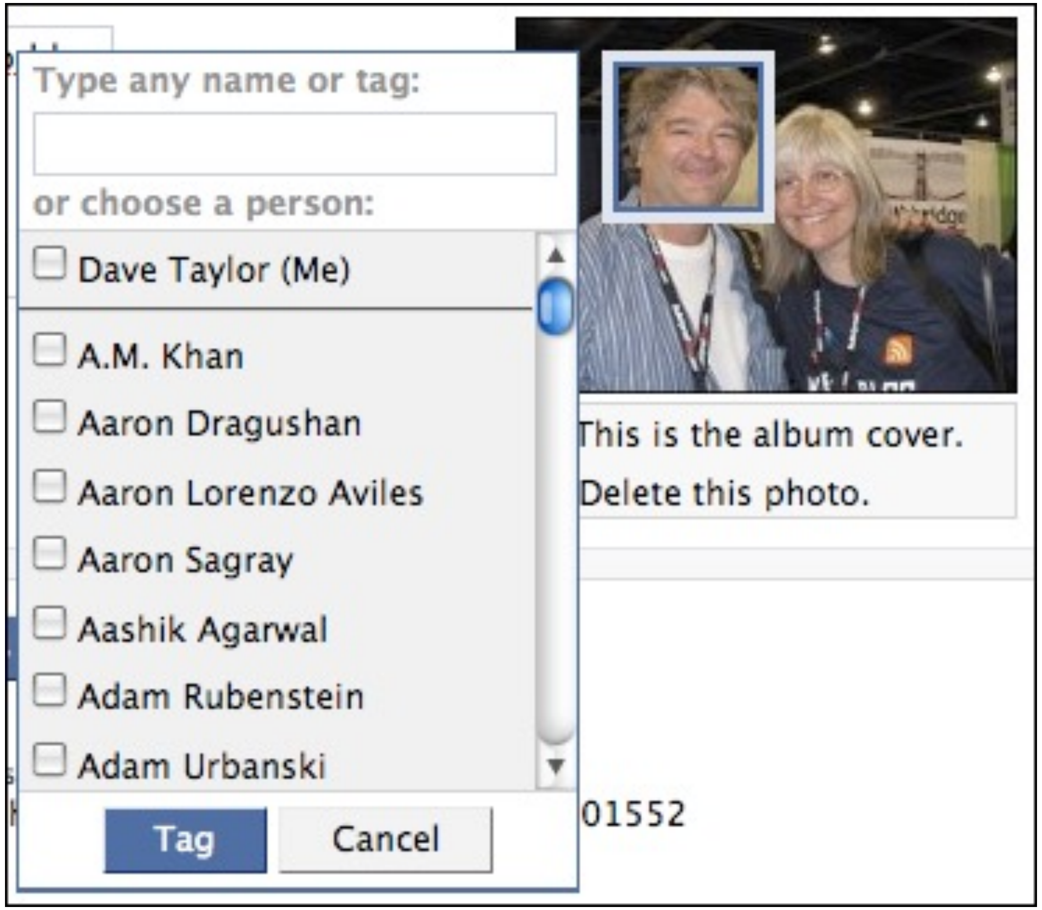


https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

Today’s Topics

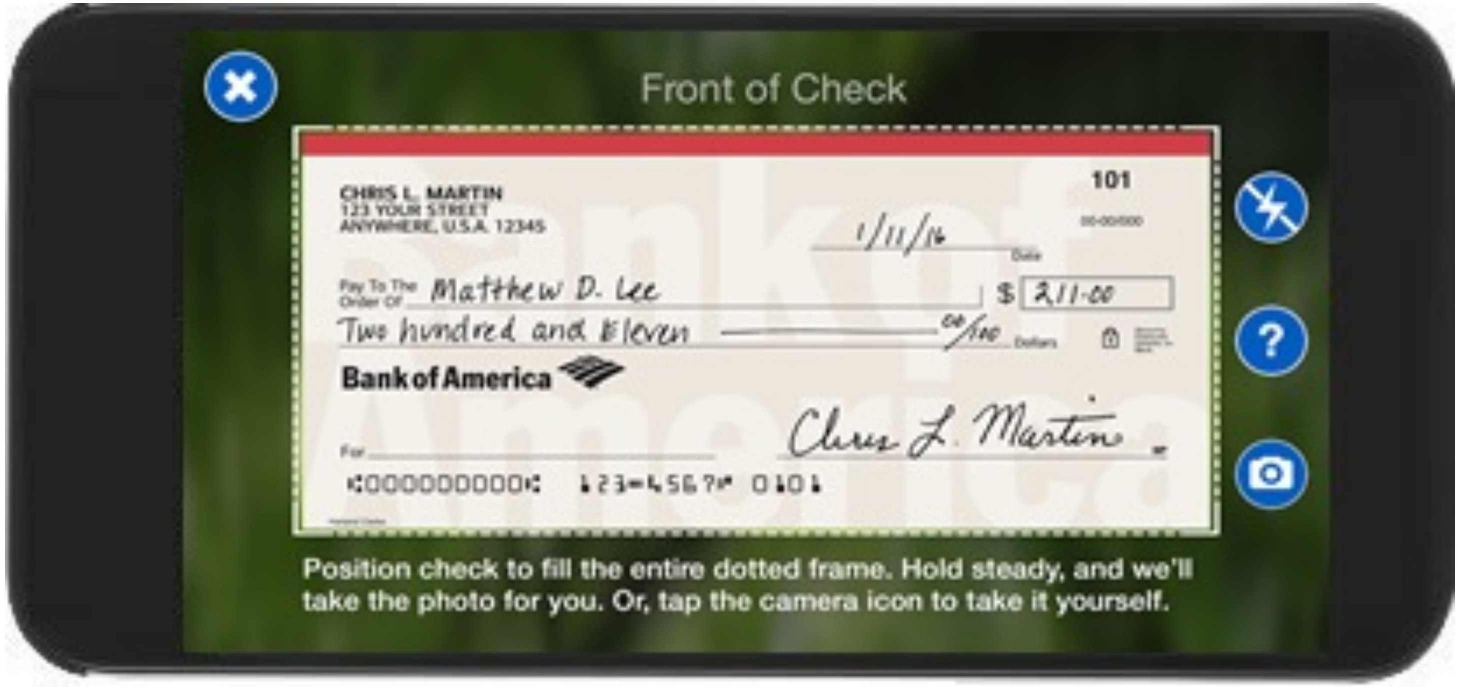
- Problems
- Applications
- PASCAL VOC detection challenge: R-CNNs
- PASCAL VOC semantic segmentation challenge: fully convolutional networks

Social Media



Face detection
(e.g., Facebook)

Banking

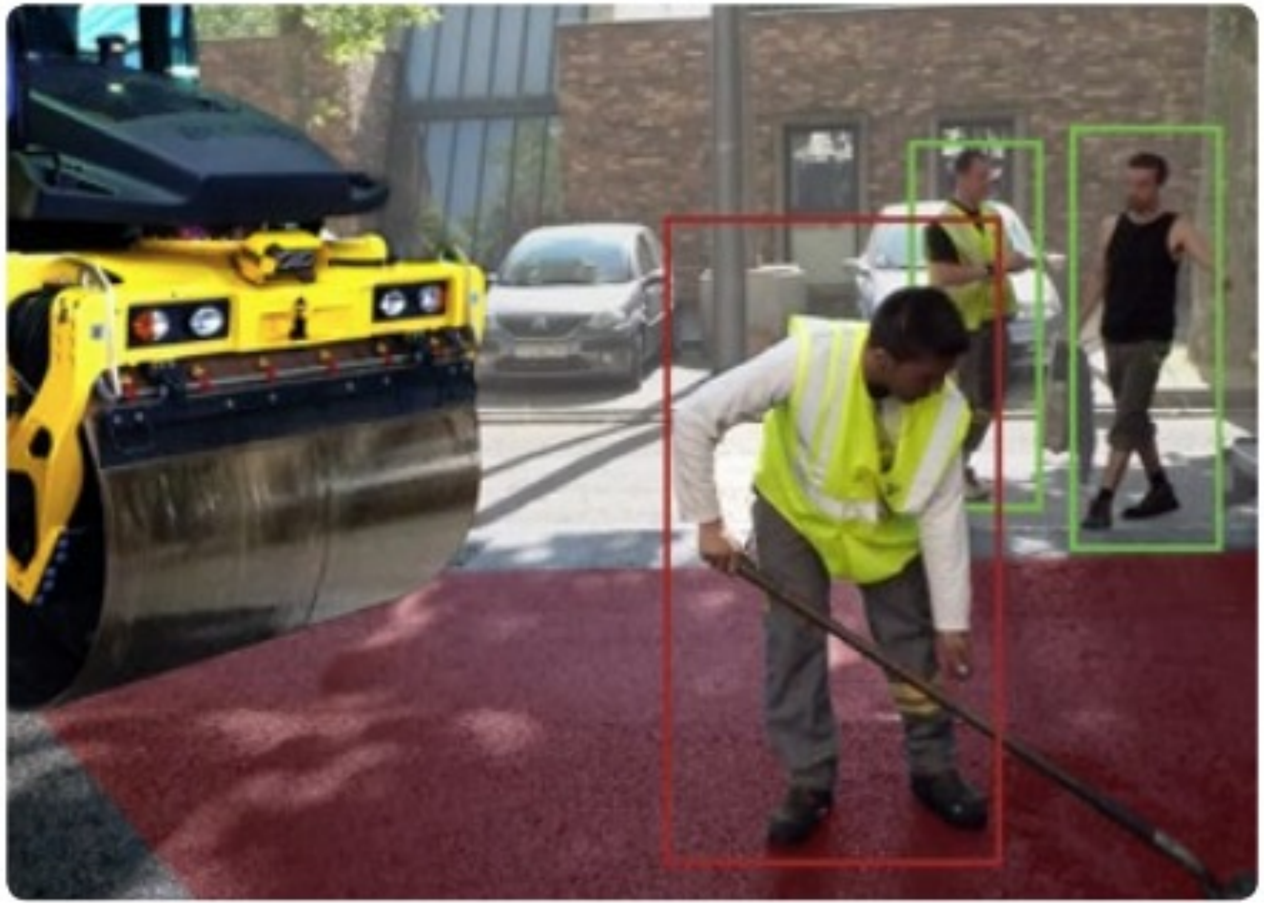


Mobile check deposit
(e.g., Bank of America)

Transportation

License Plate Detection (e.g., AllGoVision)

Construction Safety



Pedestrian Detection
(e.g., Blaxtair)
http://media.brintex.com/Occurrence/121/Brochure/3435/brochure.pdf

Counting

Counting Fish (e.g., SalmonSoft)
http://www.wecountfish.com/?page_id=143

Business Traffic Analytics

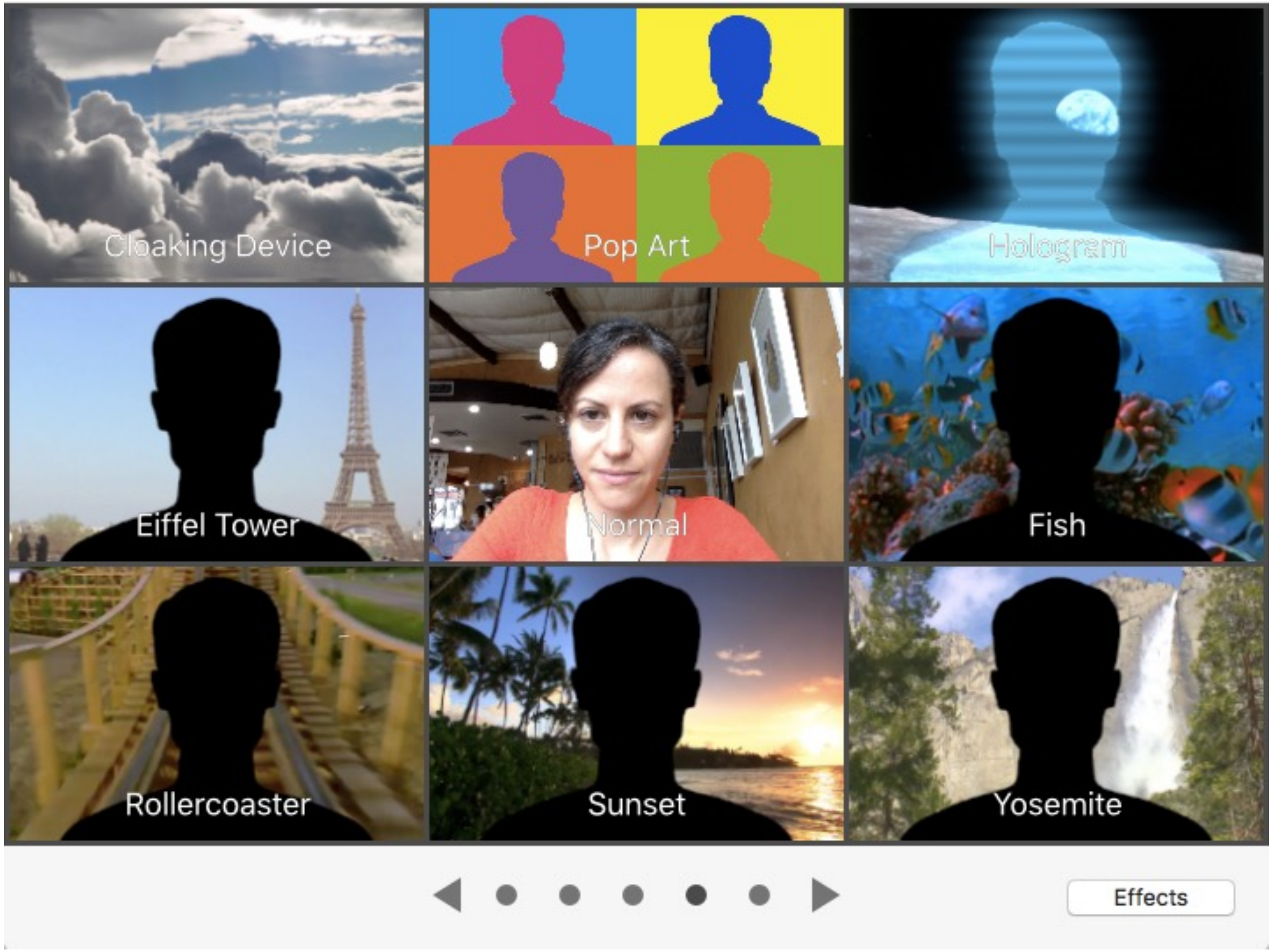
Remodeling Inspiration

(a) Target photo

(b) Retextured

Bell et al; SIGGRAPH; 2013

Rotoscoping (many examples on Wikipedia)



https://www.starnow.co.uk/ahmedmohammed1/photos/4650871/before-and-after-rotoscopinggreen-screening

Disease Diagnosis; e.g.,

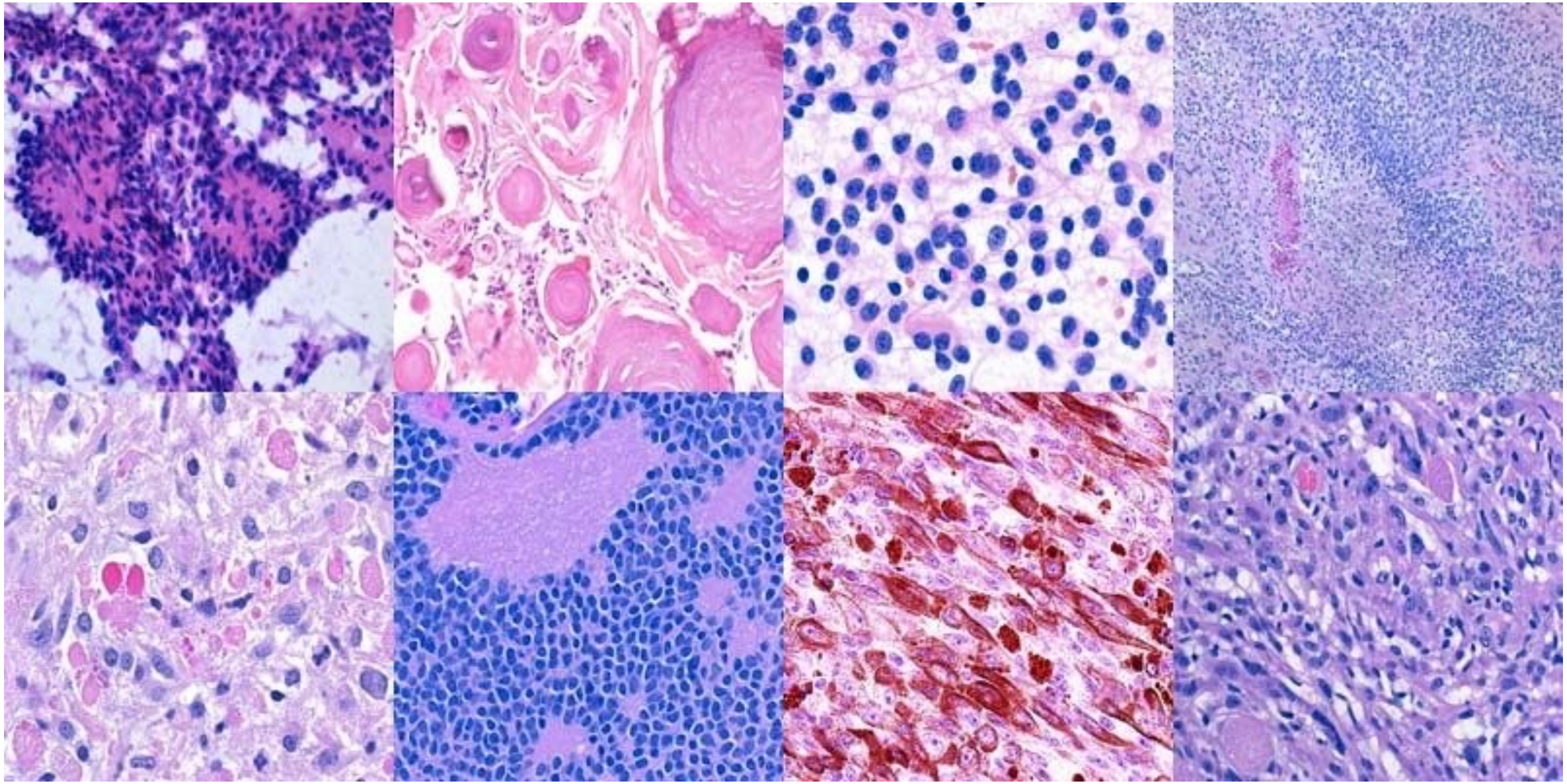
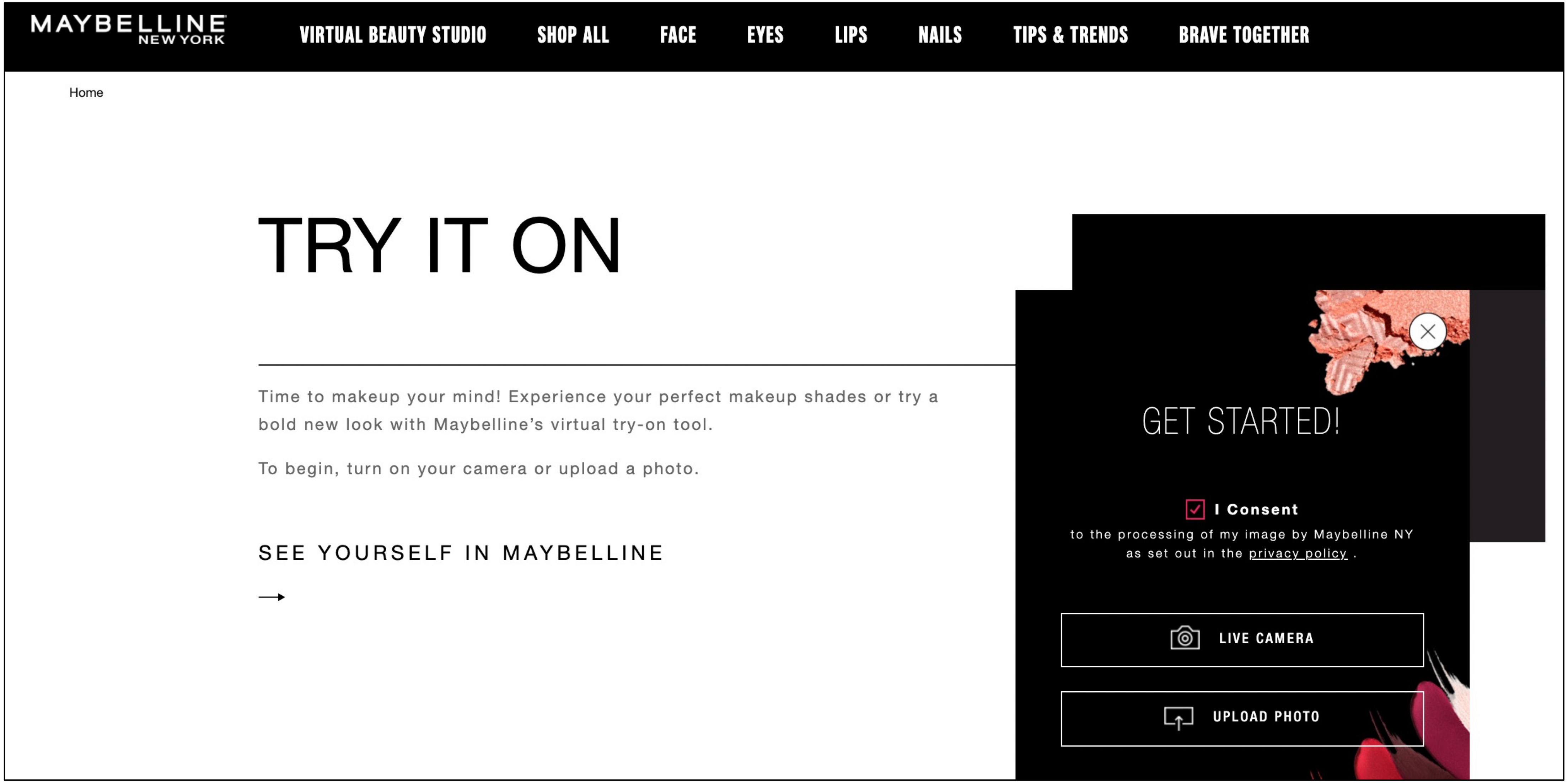


Figure Source: https://pathology.jhu.edu/brain-tumor/grading-classification

Face Makeover



Demo: https://www.maybelline.com/virtual-try-on-makeup-tools

Self-Driving Vehicles

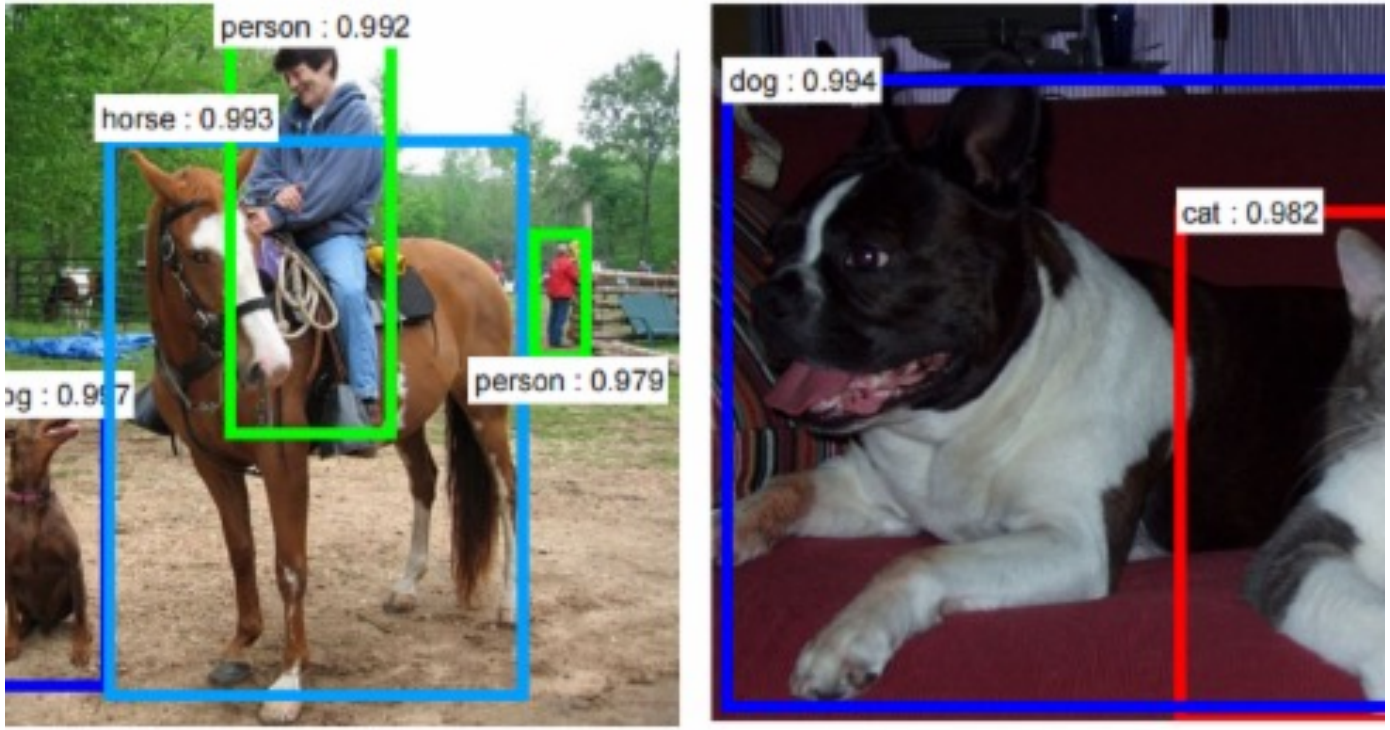
Figure Source: https://www.inc.com/kevin-j-ryan/self-driving-cars-powered-by-people-playing-games-mighty-ai.html

Today's Topics

- Problems
- Applications
- PASCAL VOC detection challenge: R-CNNs
- PASCAL VOC semantic segmentation challenge: fully convolutional networks

VOC Challenge

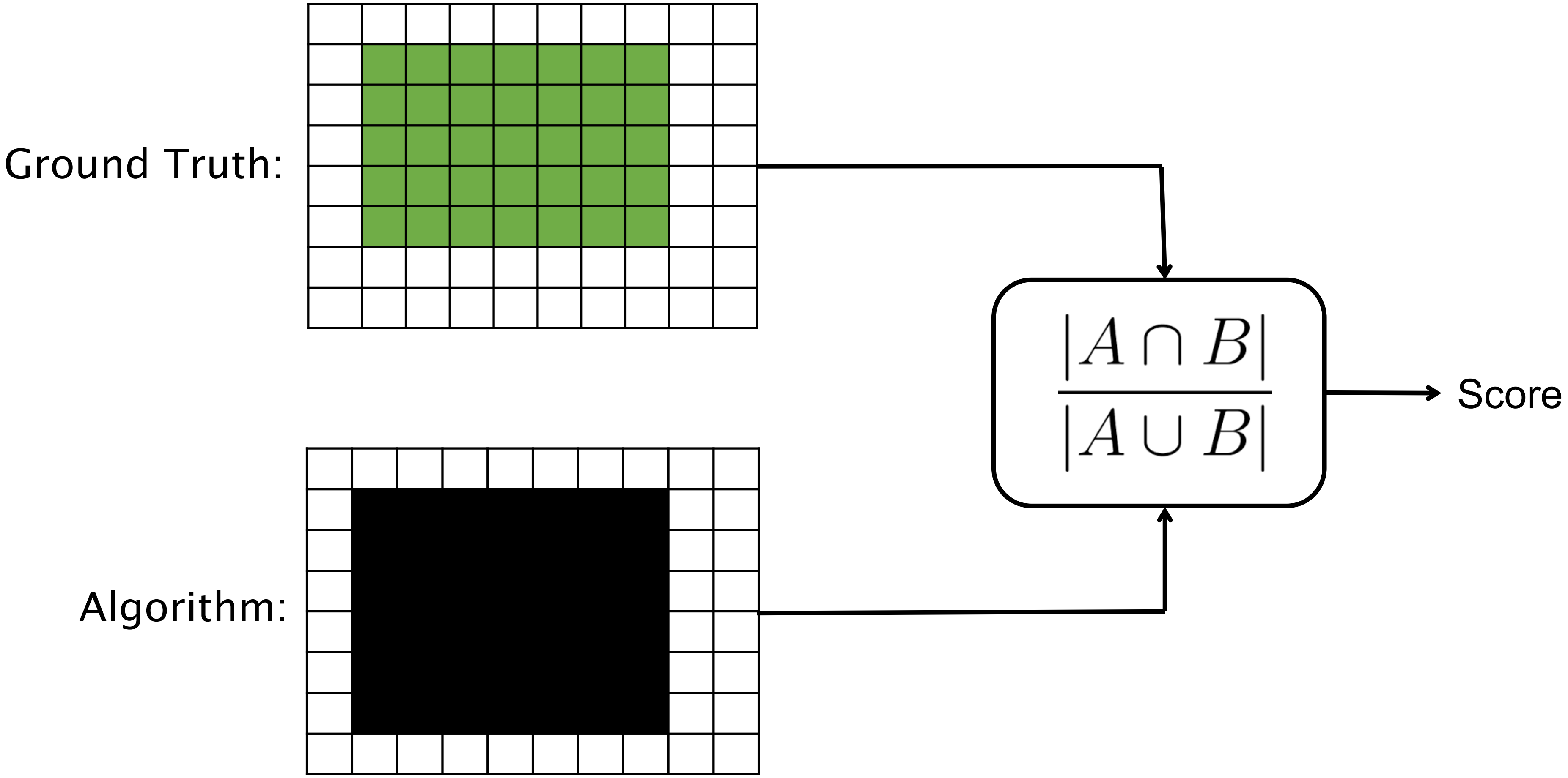
- Goal: locate all instances of 20 object categories with BBs
- Dataset: 11,530 images collected from Flickr and annotated by annotators at University of Leeds



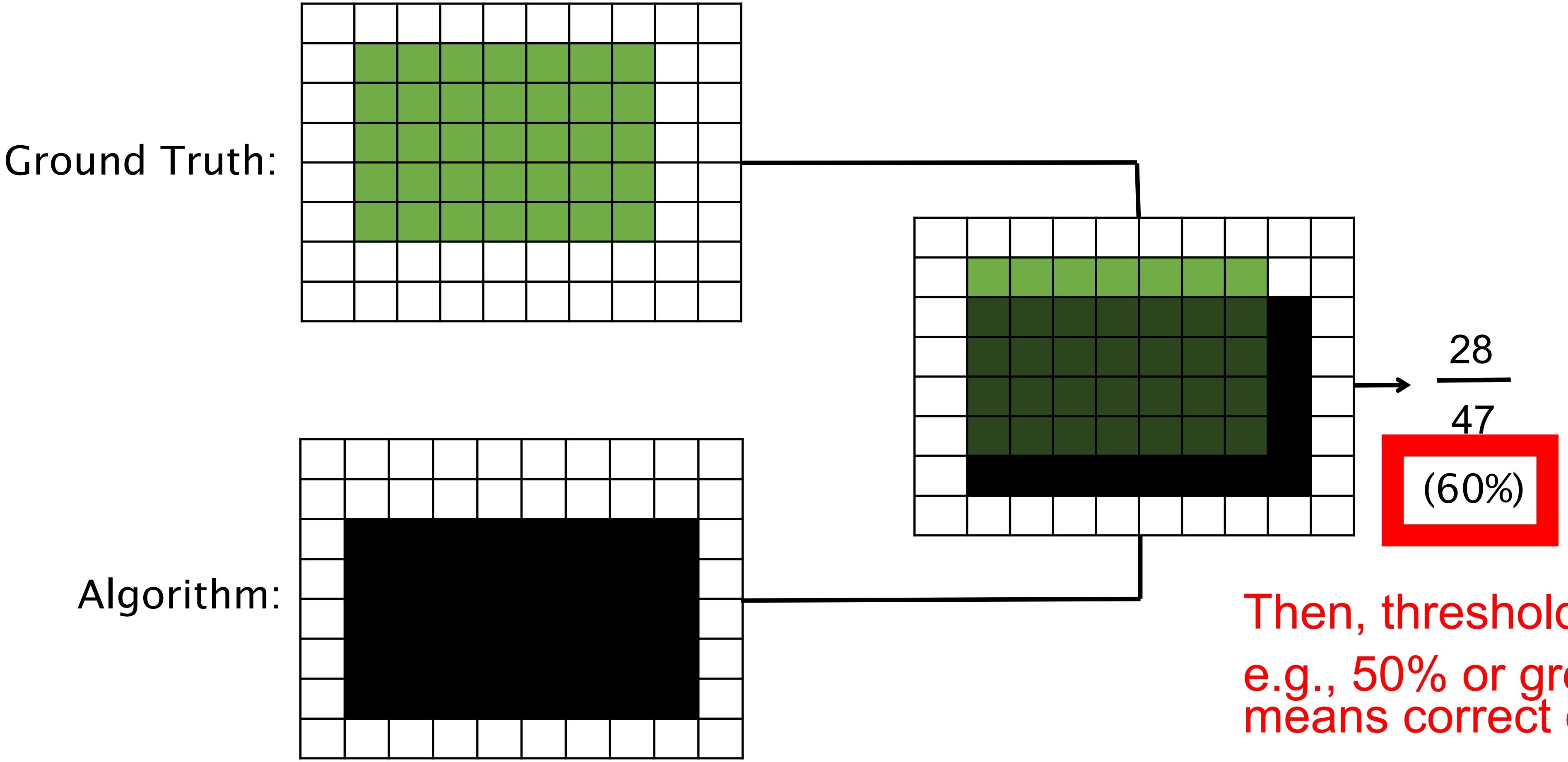
https://cv-tricks.com/artificial-intelligence/object-detection-using-deep-learning-for-advanced-users-part-1/

Dataset location: http://host.robots.ox.ac.uk/pascal/VOC/index.html
Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. IJCV 2010.

VOC Challenge: Evaluation Metric (IoU)



VOC Challenge: Evaluation Metric (IoU)

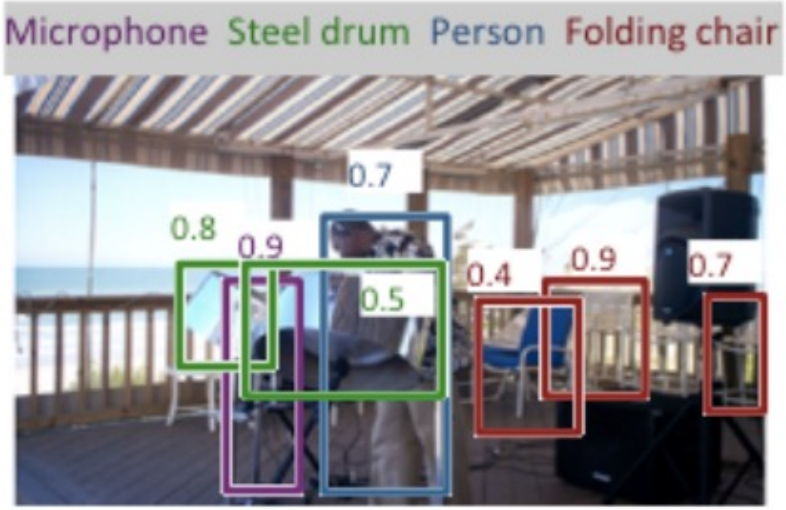


Then, threshold:
e.g., 50% or greater means correct detection!

VOC Challenge: Evaluation Metric (mAP)

- For each object class (e.g., cat, dog, …), compute:
  - Precision: fraction of correct detections from all detections using 0.5 IoU threshold

Ground truth



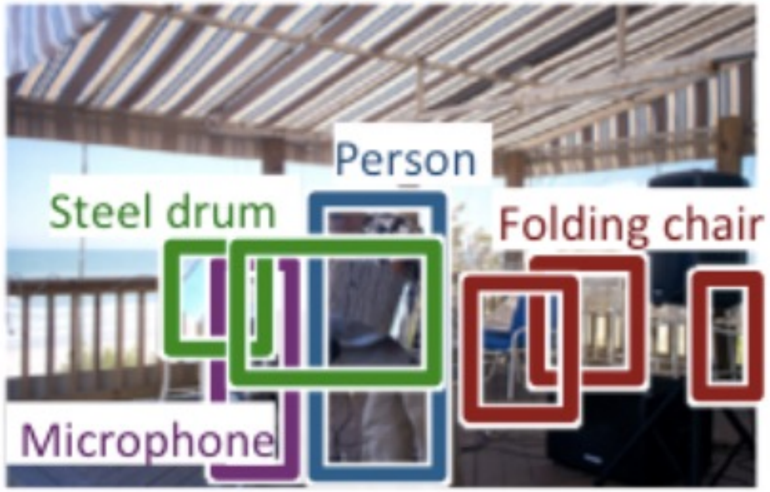
Algorithm BB + its Confidence

[Russakovsky et al; IJCV 2015]

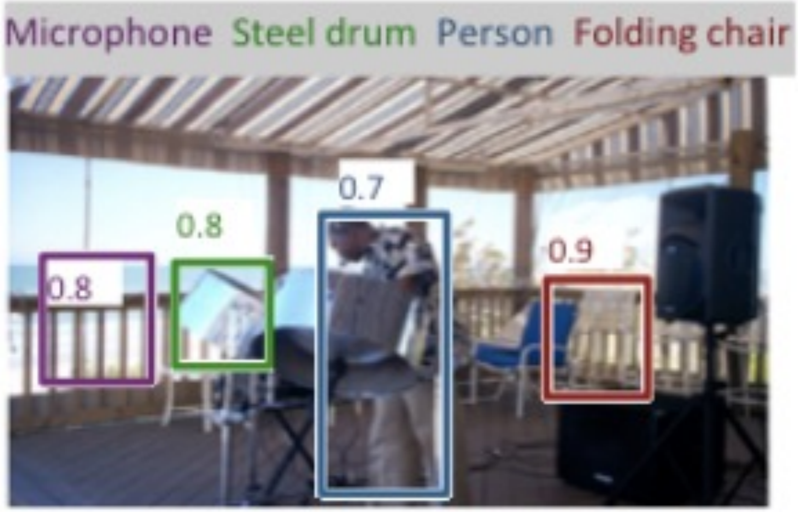
https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

VOC Challenge: Evaluation Metric (mAP)

- For each object class (e.g., cat, dog, …), compute:
  - Precision: fraction of correct detections from all detections using 0.5 IoU threshold



Ground truth



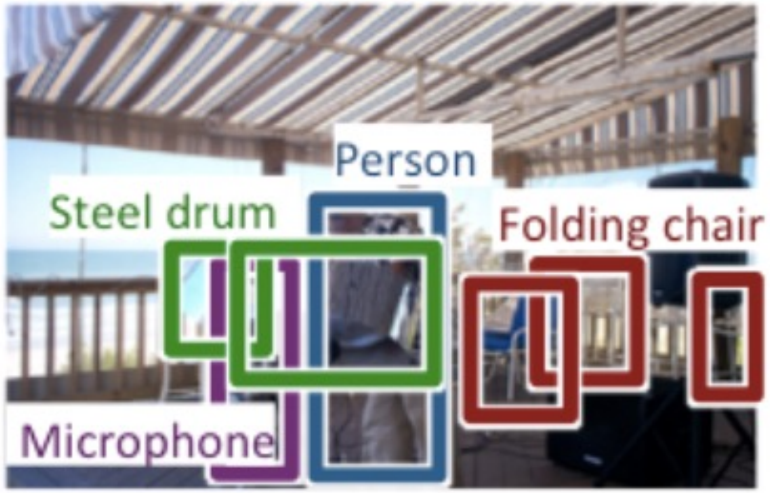
AP: ? ? ? ?

[Russakovsky et al; IJCV 2015]

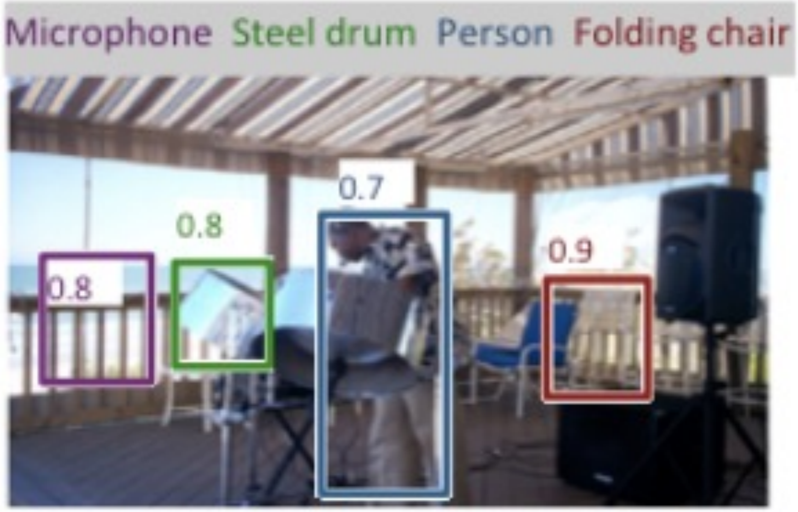
https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

VOC Challenge: Evaluation Metric (mAP)

- For each object class (e.g., cat, dog, …), compute:
  - Precision: fraction of correct detections from all detections using 0.5 IoU threshold
- Then, compute mean precision across all classes



Ground truth



AP: 0.0 0.5 1.0 0.3

[Russakovsky et al; IJCV 2015]

https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

Evaluation Metric (mAP): Why "Mean" and Why "Average"

- More generally, for each object class (e.g., cat, dog, …) :
  - AP: compute area under a precision-recall curve, created by varying IoU threshold

- Then, compute mean AP across all classes

[Russakovsky et al; IJCV 2015]
https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

Naïve Solution: Sliding Window Approach

Person?
Person?
Person?
Person?
Person?
Person?
Person?
Person?
Person?

Image Source: https://yourboulder.com/boulder-neighborhood-downtown/

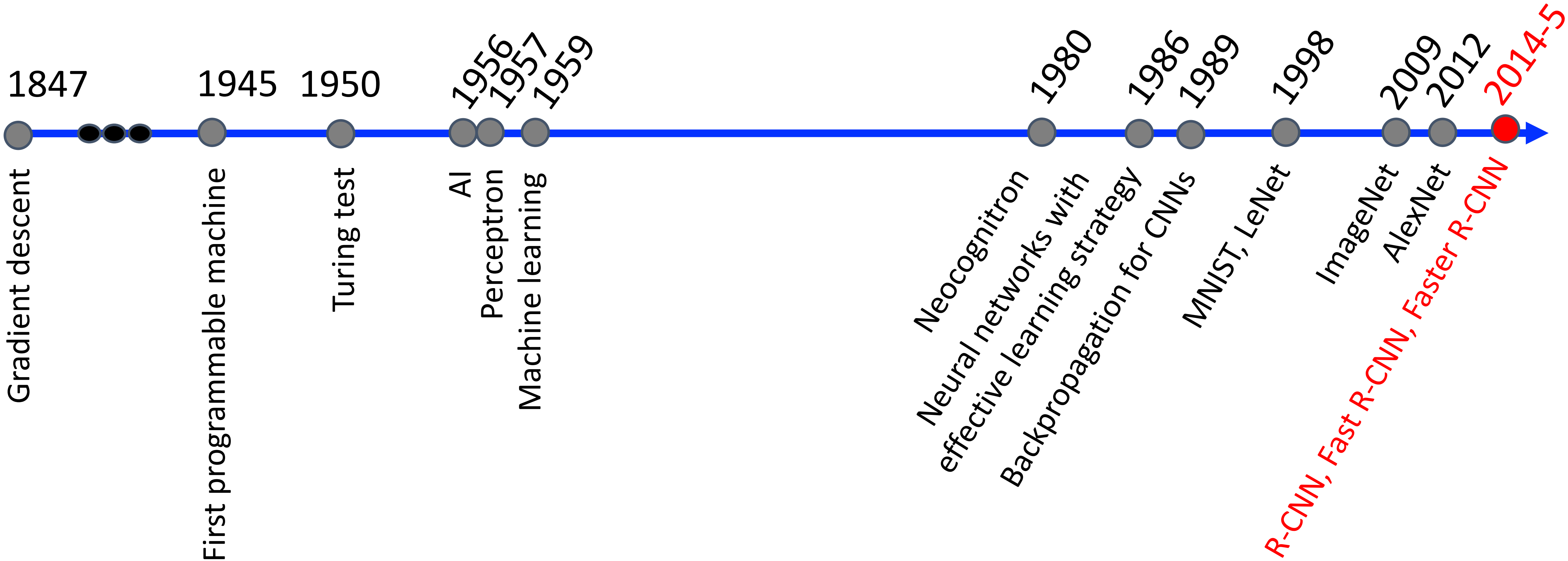
Naïve Solution: Sliding Window Approach

Image Source: https://yourboulder.com/boulder-neighborhood-downtown/

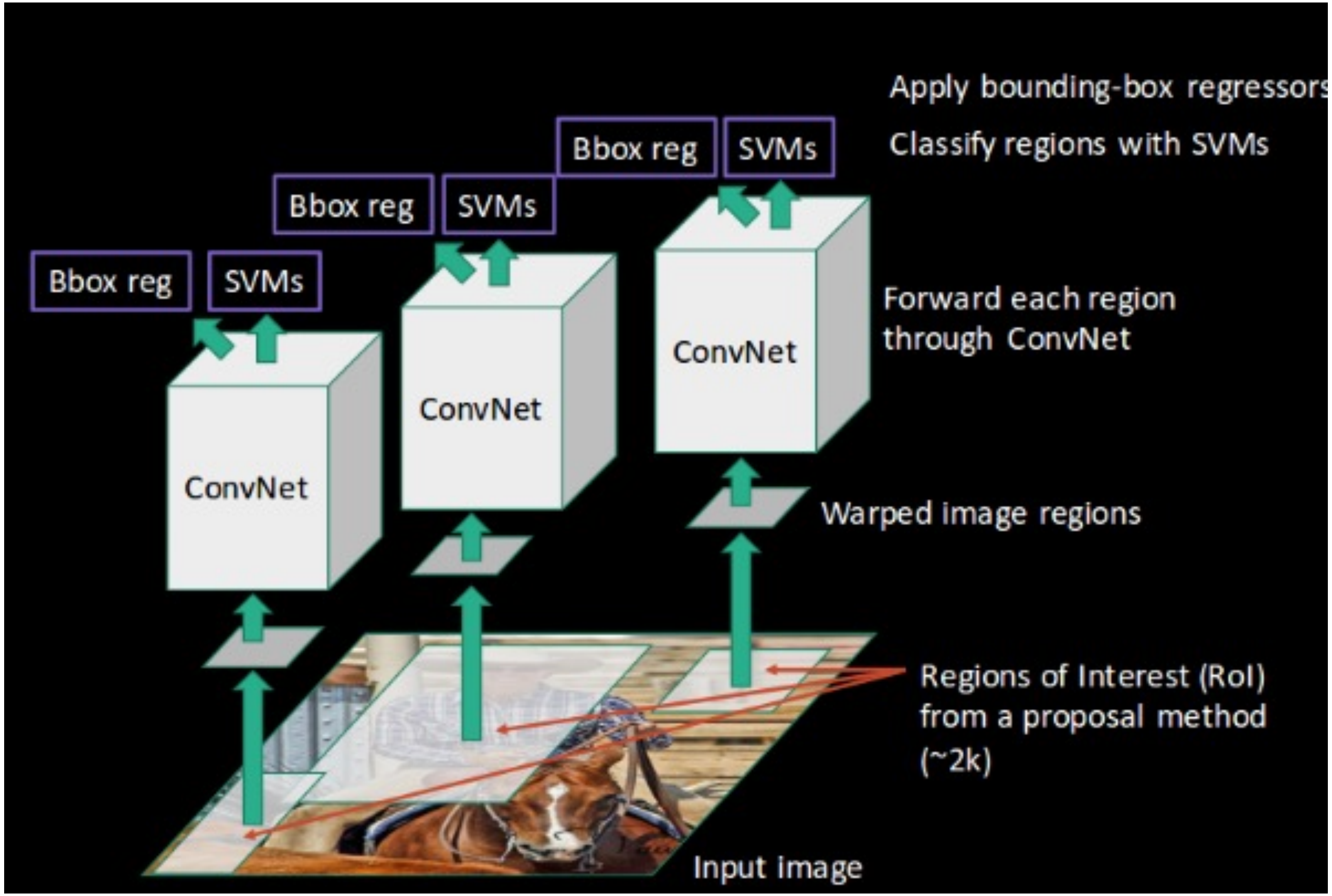
Naïve Solution: Sliding Window Approach

- Sliding window approach: must test different locations at…
  - Different scales
  - Different aspect ratios (e.g., for person vs car or car viewed at different angles)
- Number of regions to test? (e.g., 1920 x 1080 image)
  - Easily can explode to hundreds of thousands or millions of windows
- Key limitation
  - Very slow!

Historical Context: R-CNN Methods



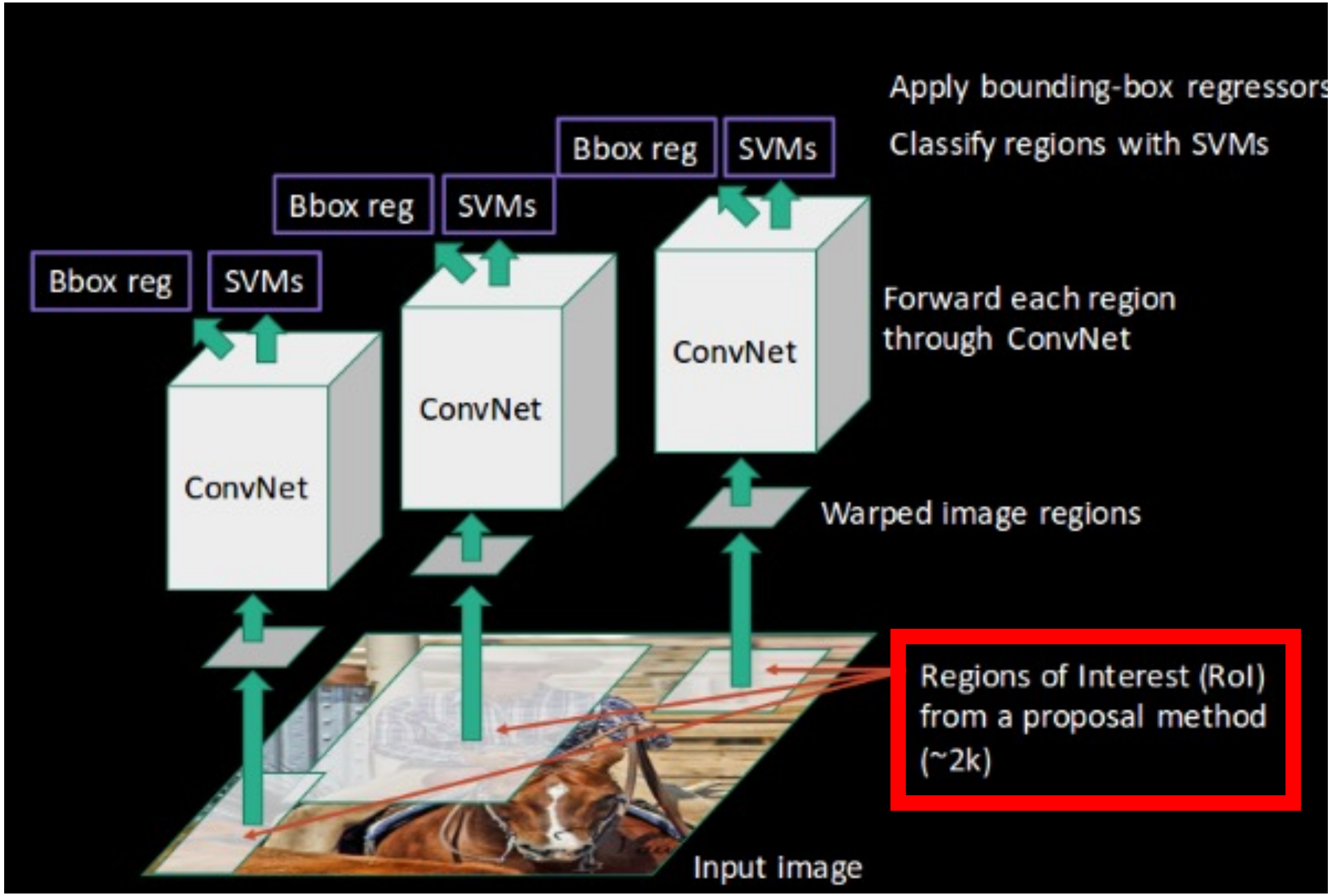
R-CNN



- First CNN to outperform hand-crafted features on detection challenges
- Named after technique: **R**egion proposals with **CNN** features

Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

R-CNN



Locate "object"-like regions using objectness methods

- Considerably fewer regions than sliding window approach
- Regions likely contain objects of interest (i.e., high recall)

Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

R-CNN

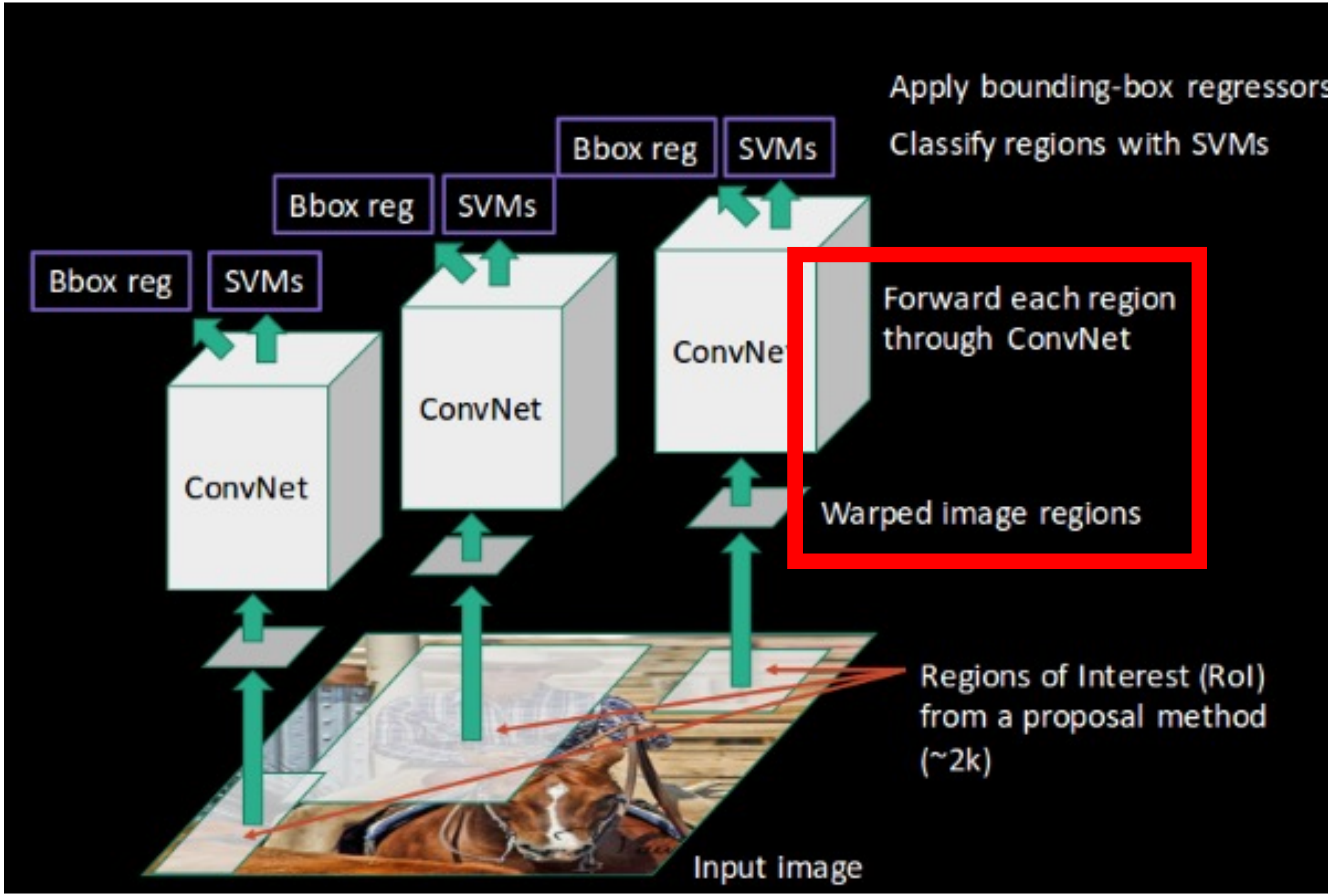


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Describe Each Region with Fixed-length Vector

Given relatively little amount of training data, devise good feature by fine-tuning pre-trained model

1) Replace final layer of AlexNet (trained on ImageNet) with # of categories in detection dataset

2) Train for image classification (use max IoU class, if IoU >= 0.5)

How many classes should be predicted?

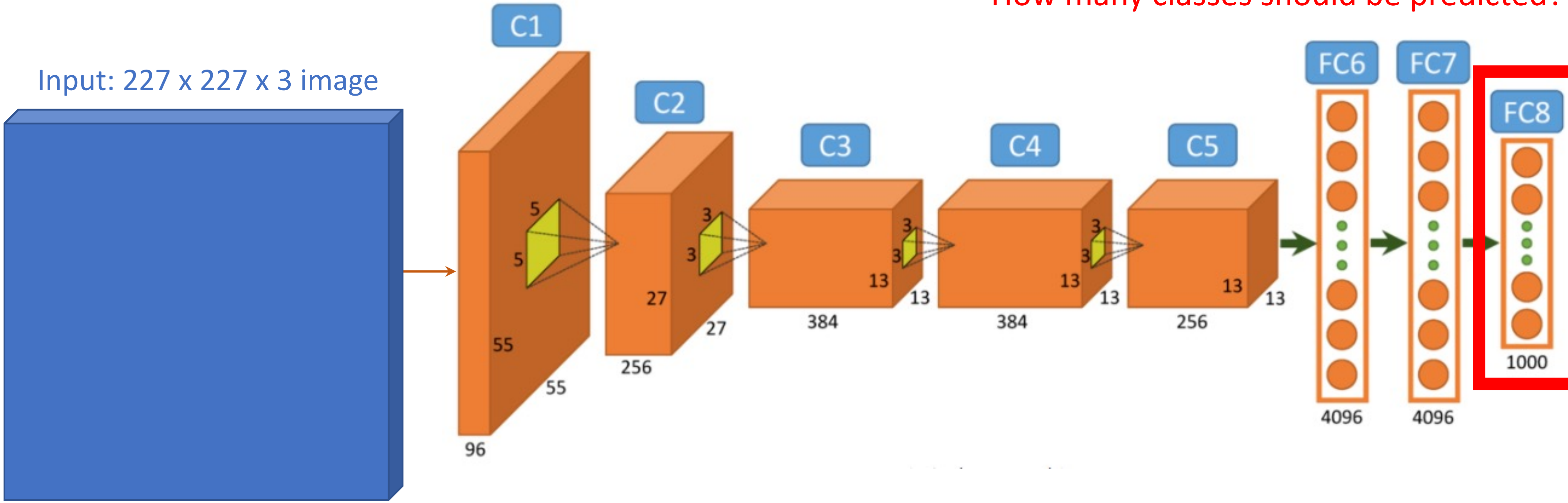


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Describe Each Region with Fixed-length Vector

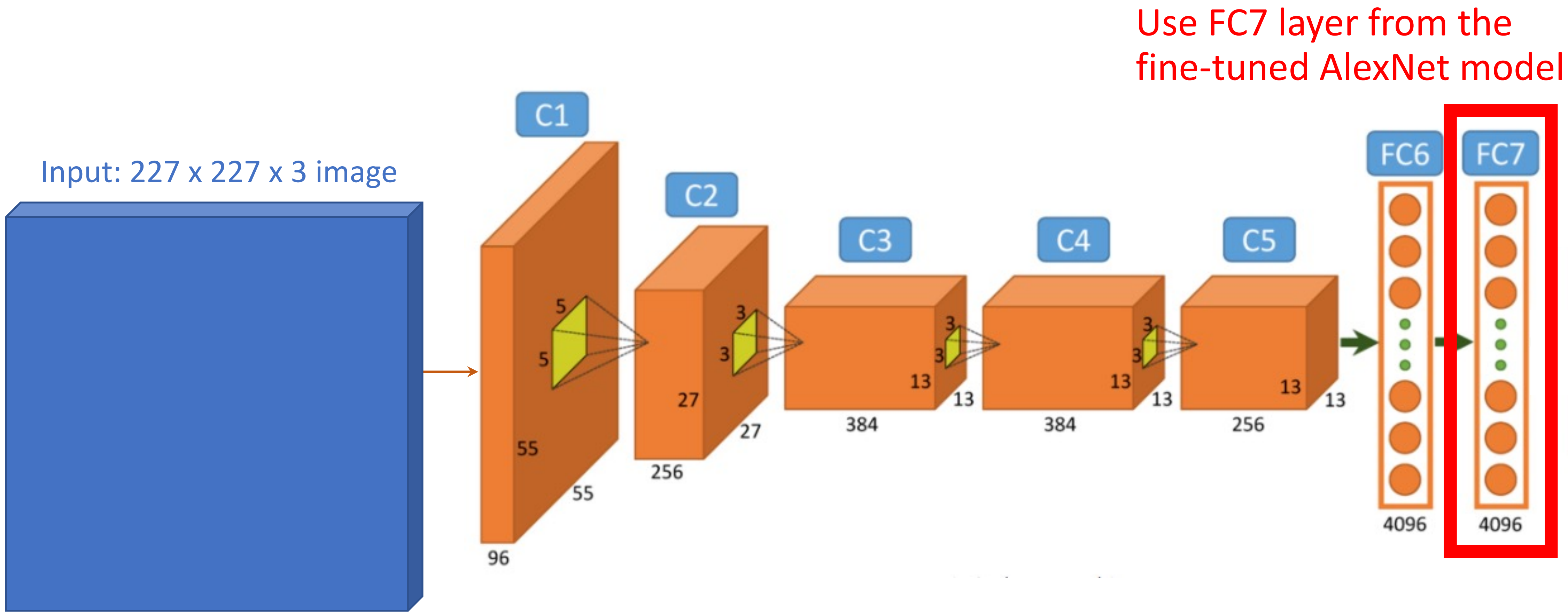


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Describe Each Region with Fixed-length Vector

Challenge: how to resize a proposed region to the required size?

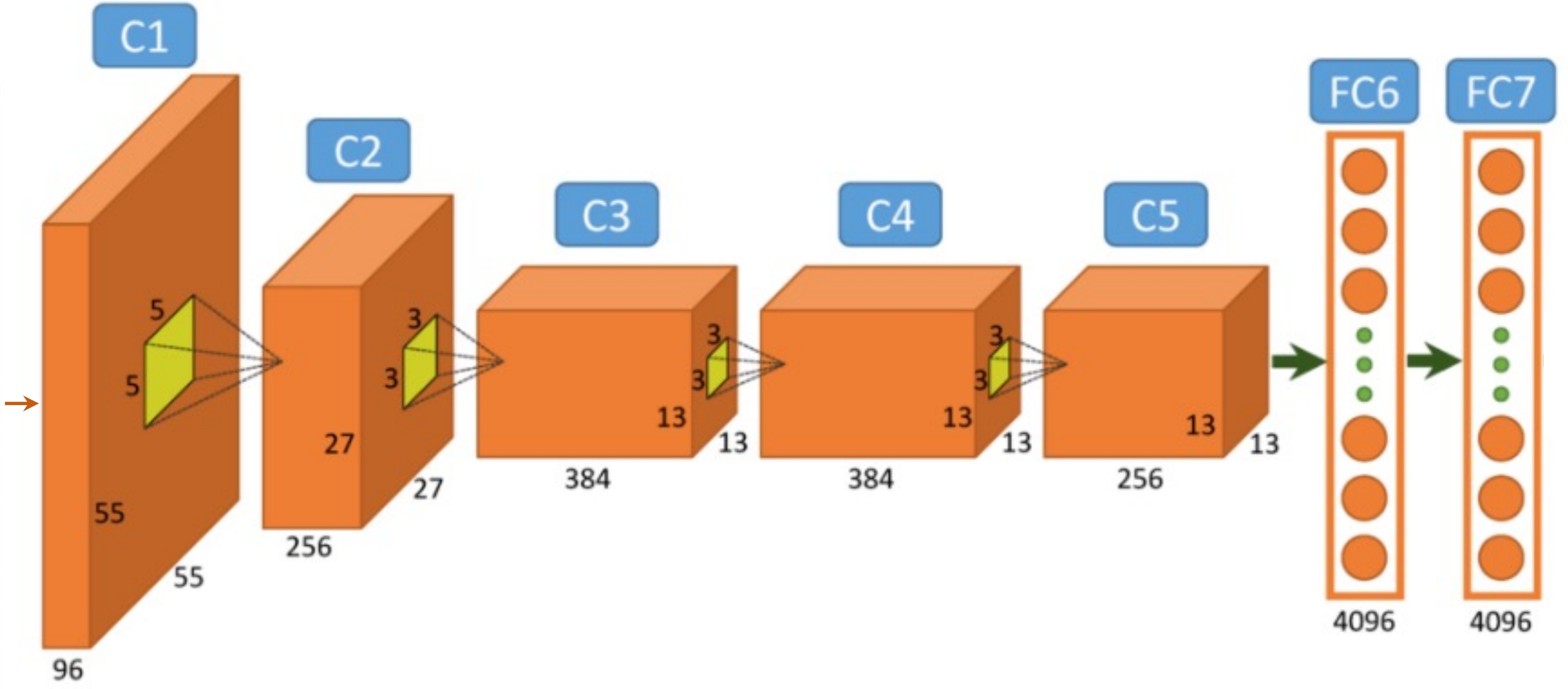


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Describe Each Region with Fixed-length Vector

Region anisotropically scaled to fit the required resolution

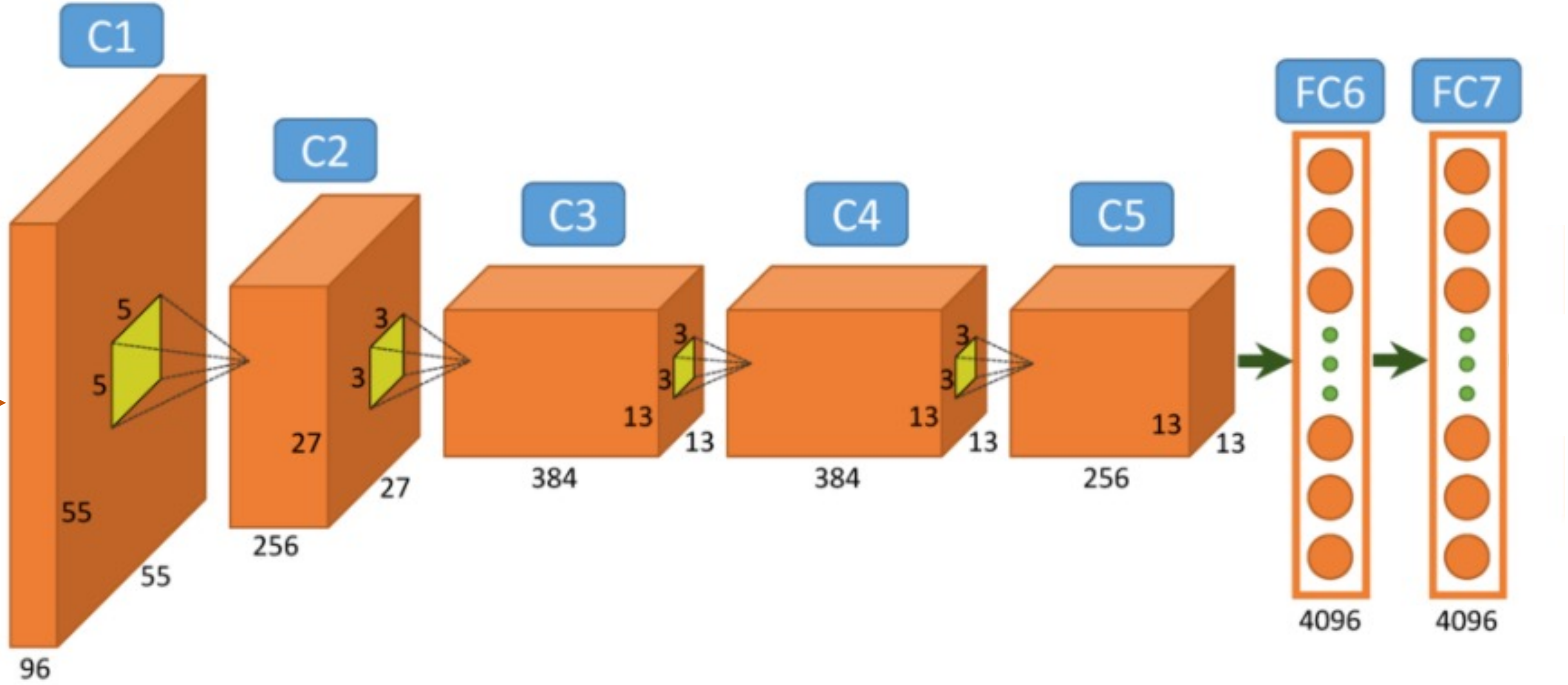


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Describe Each Region with Fixed-length Vector

Input: 227 x 227 x 3 image

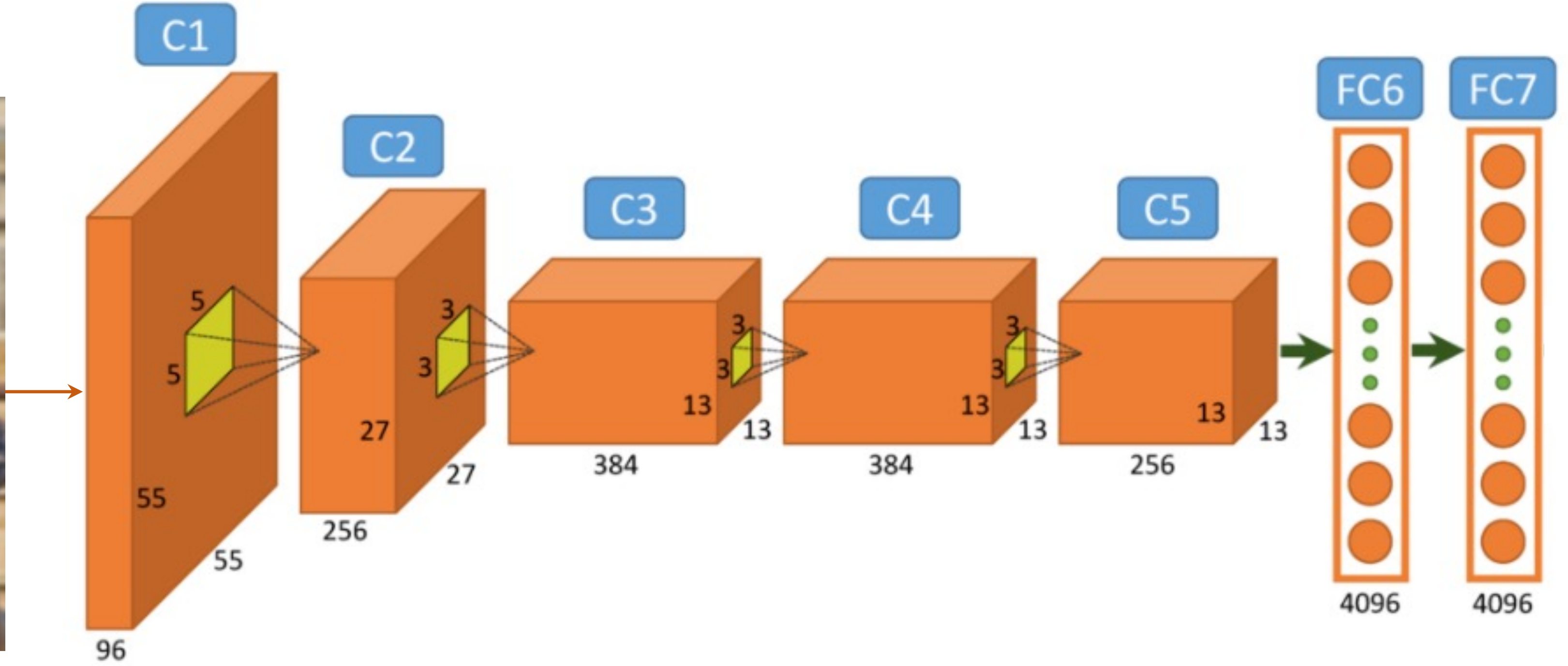
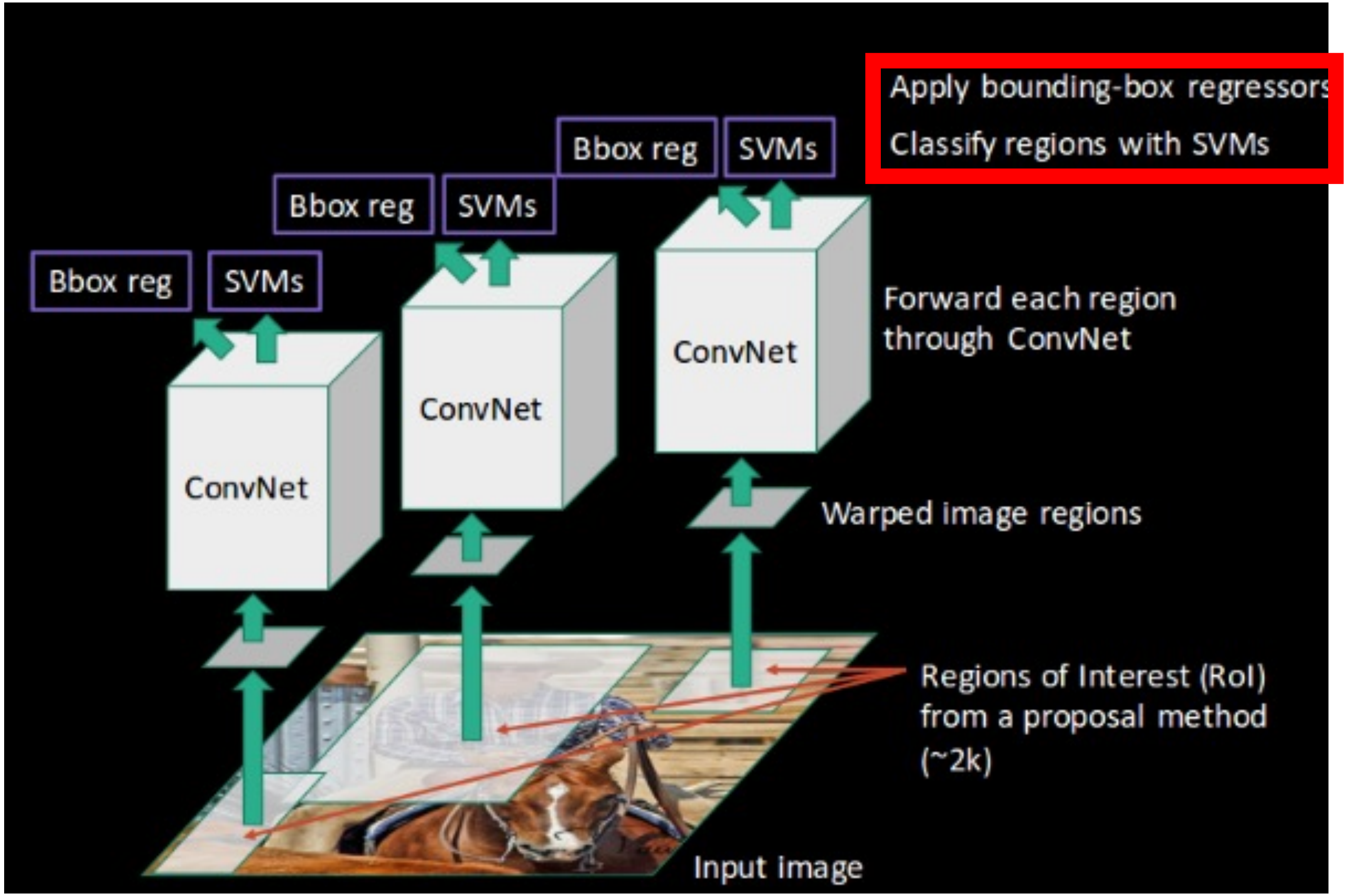


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

R-CNN



1. SVM classifier trained to use a region's CNN feature to assign a category from pre-defined set

2. Regressor trained to refine each region's position, width, and height

Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

R-CNN: Region Refinement

Original region proposal with center (p_x, p_y), width (p_w), and height (p_h) is refined using model parameters (d_x, d_y, d_w, d_h)

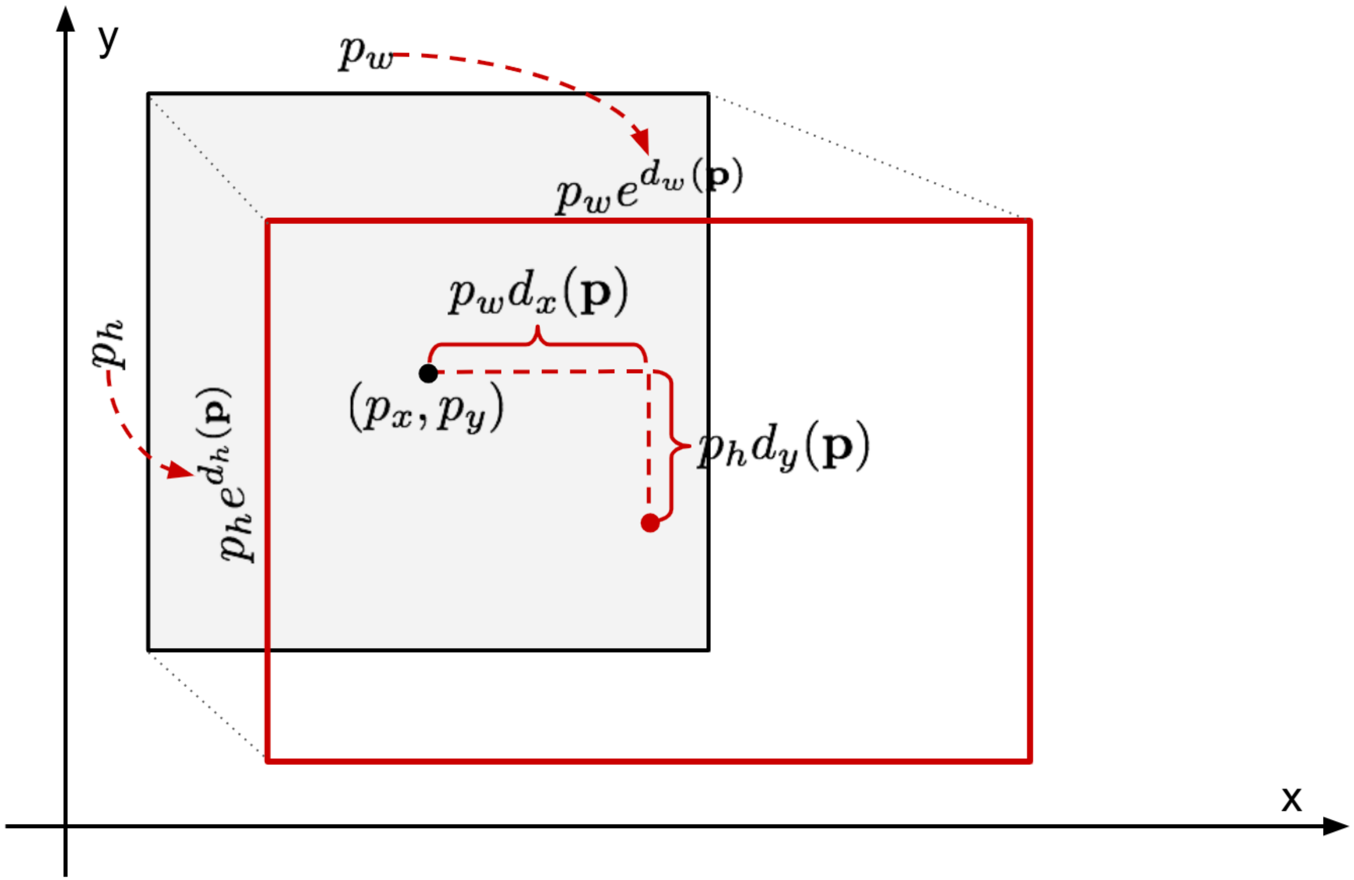


Image Source: https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html#bounding-box-regression

Algorithm Training: Linear Regression Model

- **Aim**: learn transformation from region proposal to ground truth
- **Input**: original region location; BB described by a center (p_x, p_y), width (p_w), and height (p_h)
- **Output**: learns four refinement functions: d_x, d_y, d_w, d_y
- Loss function for learning: SSE

$$\sum_{i\in\{x,y,w,h\}} (t_i - d_i(\mathbf{p}))^2$$

True location

Predicted location

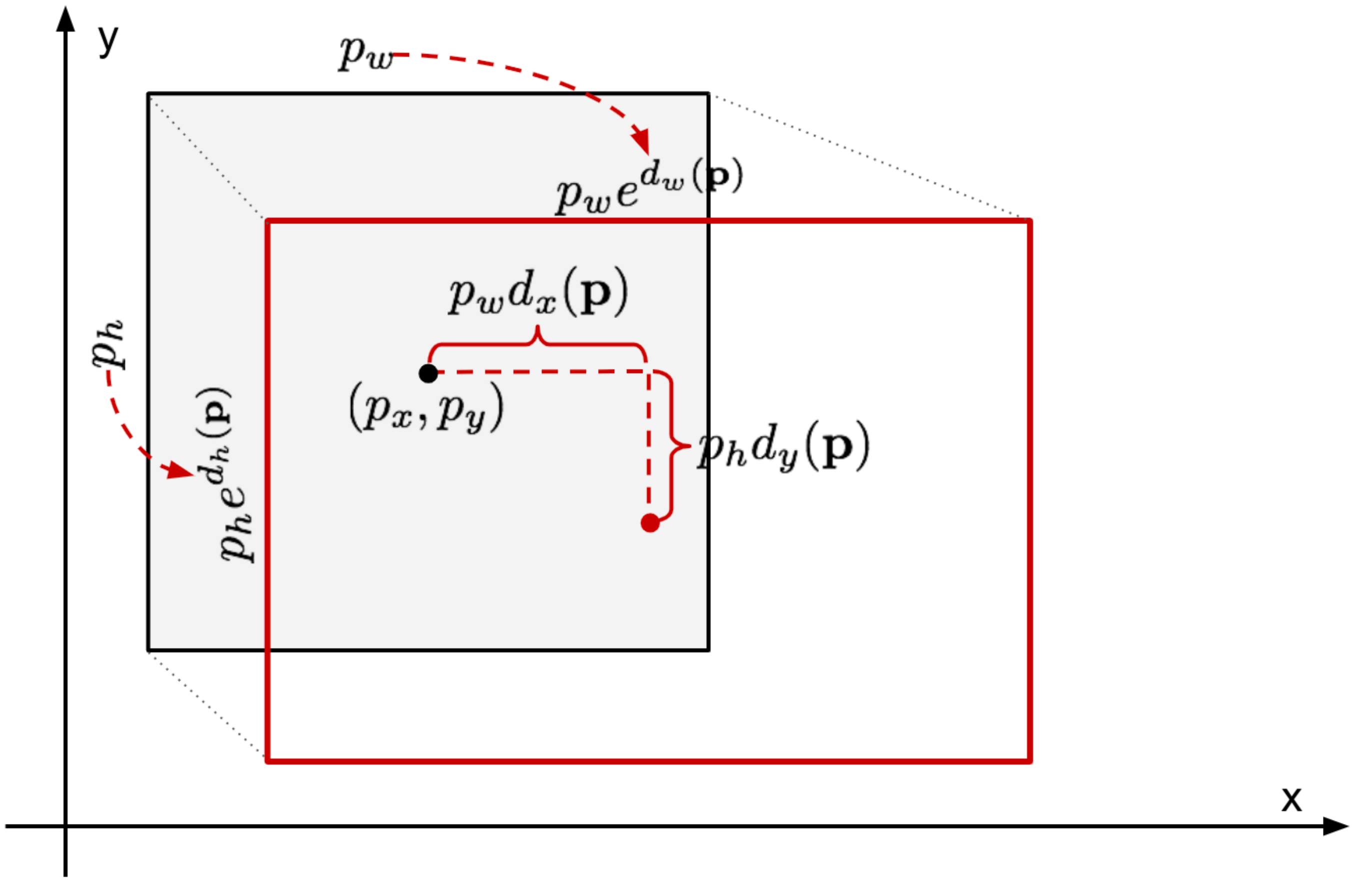
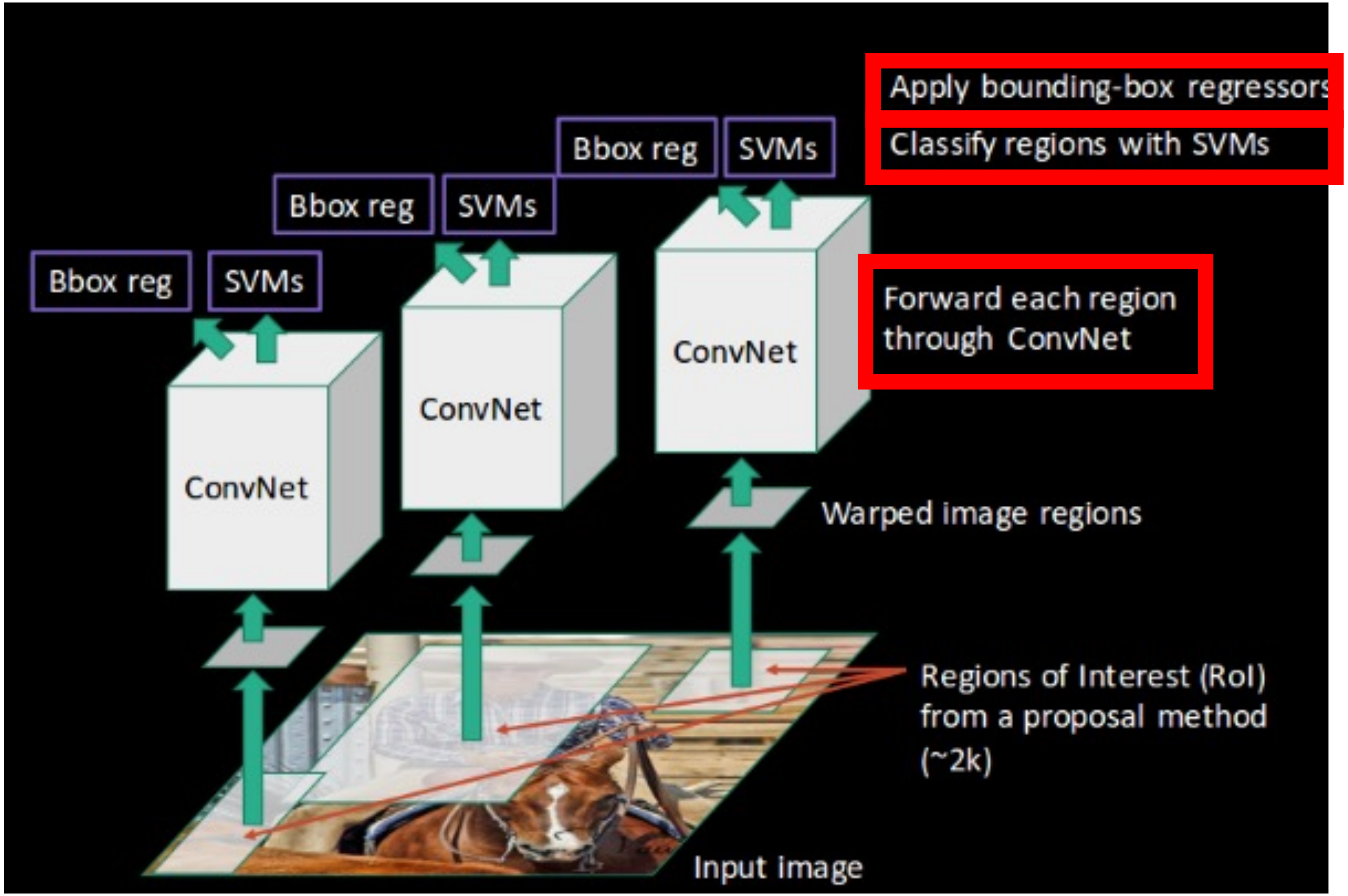


Image Source: https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html#bounding-box-regression

R-CNN Limitations

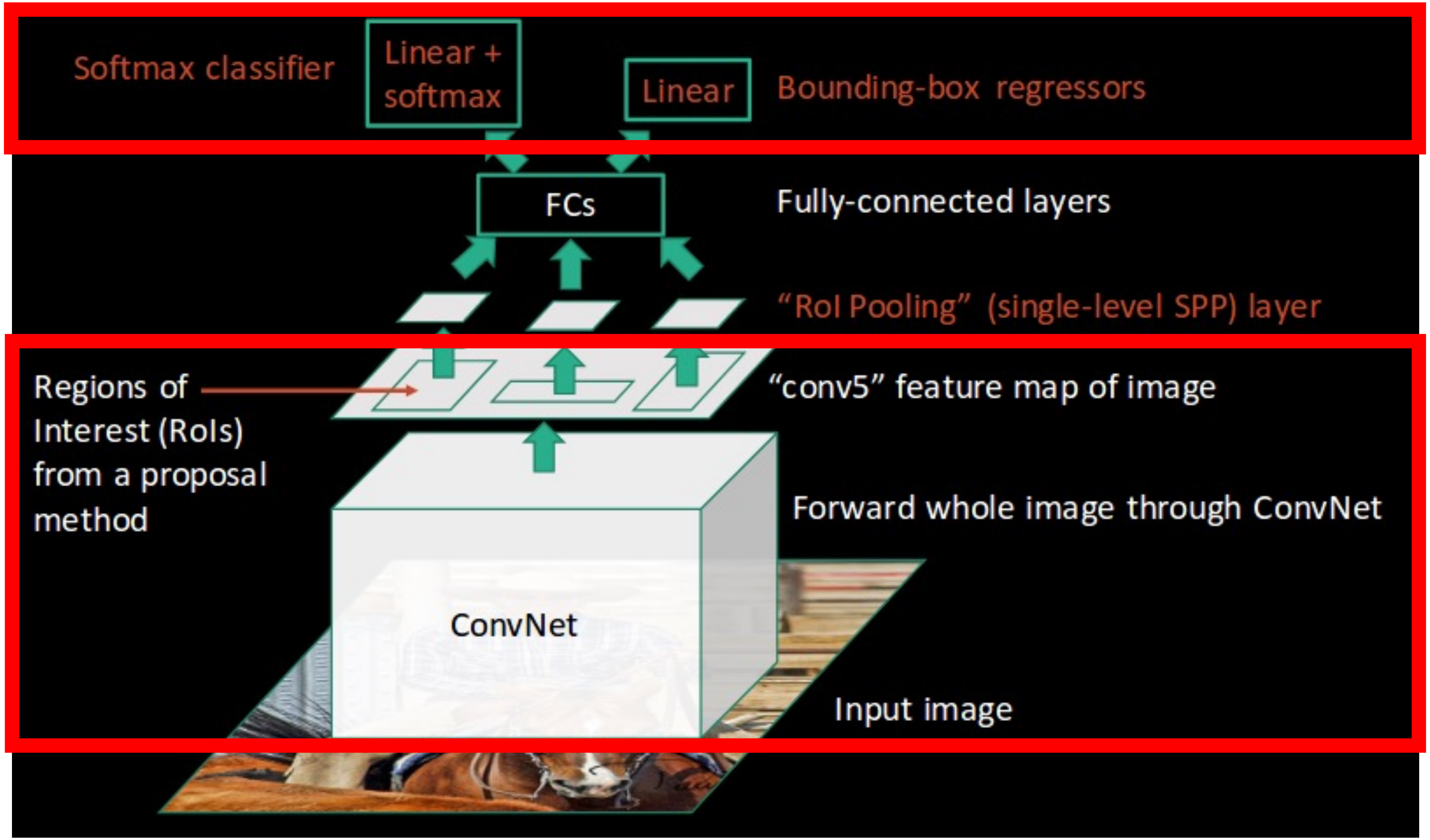


- Slow training procedure
  - Must train three models
- Slow at test time

  (~1 minute per image)

Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Fast R-CNN: Single Stage Training (rather than 3)



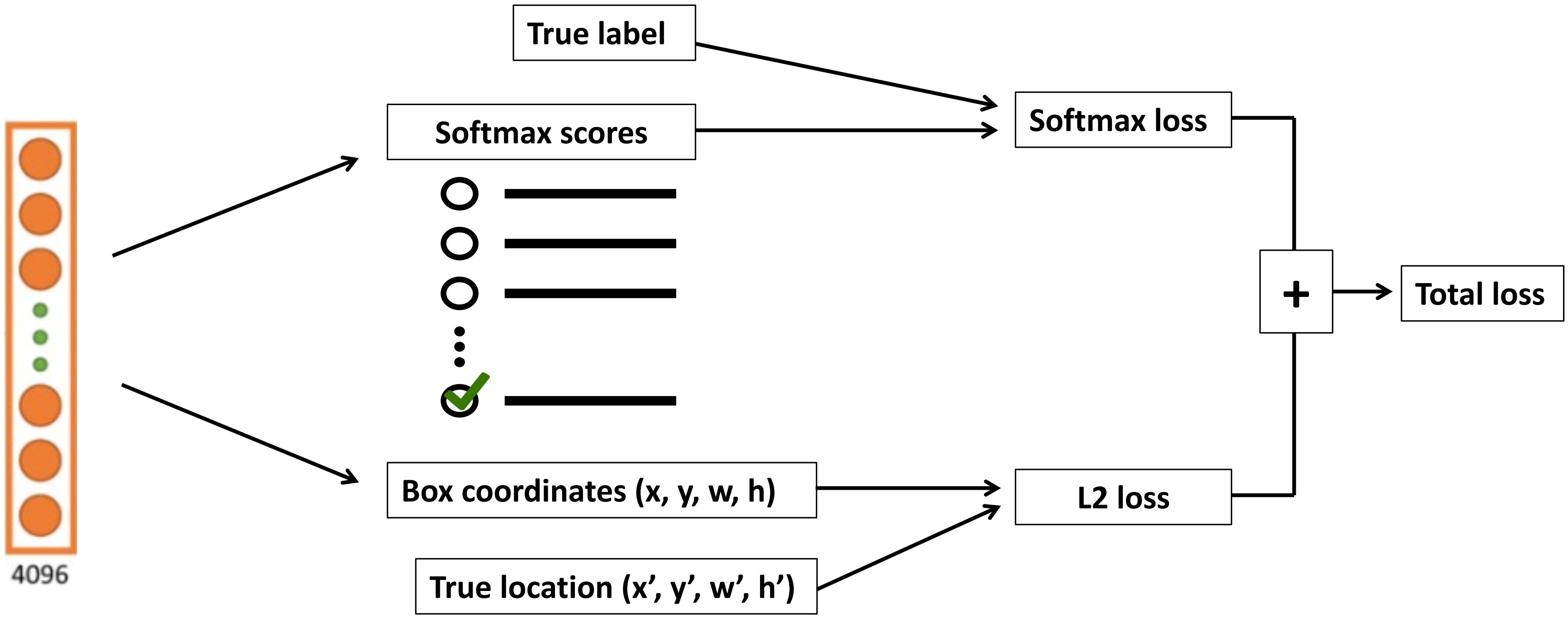
For each region, assign it to a class and refine it

Extract feature description per proposed region with section of feature map corresponding to region

Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

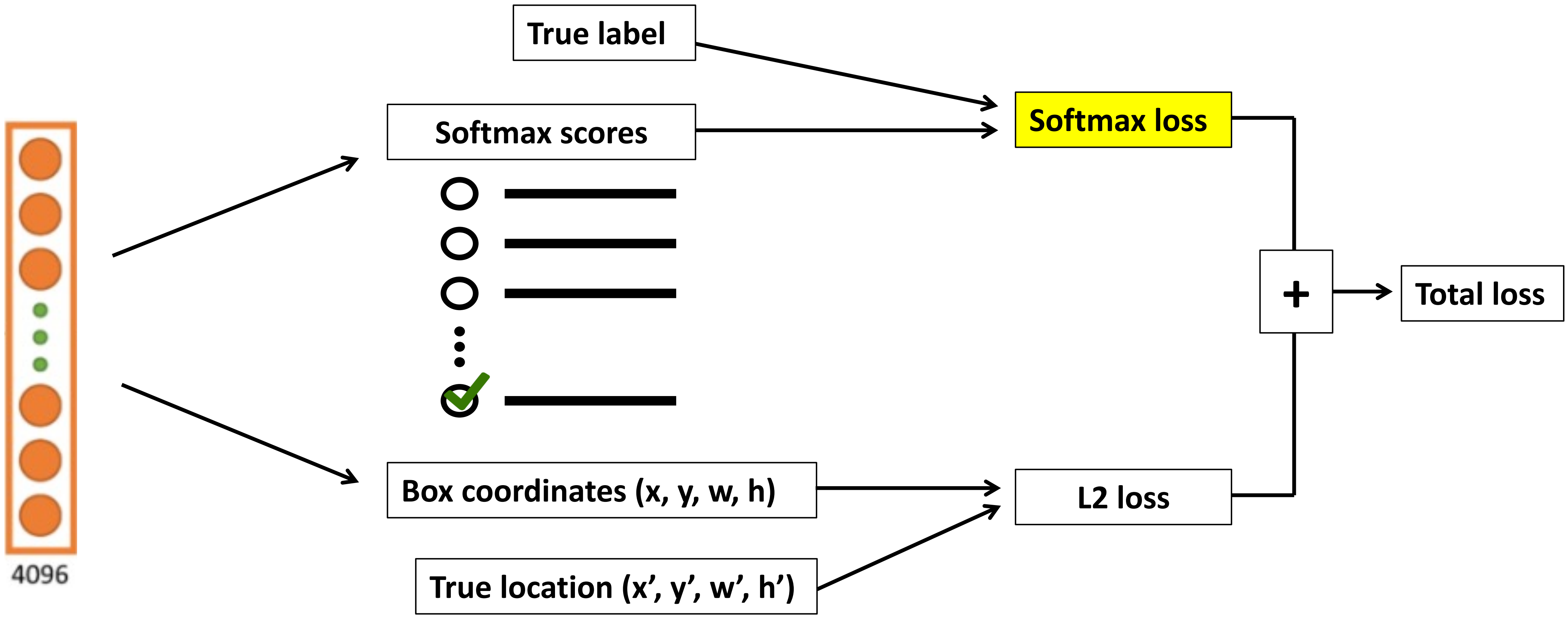
Fast R-CNN Training: Multi-task Loss

Objective function sums classification and localization losses for each region proposal



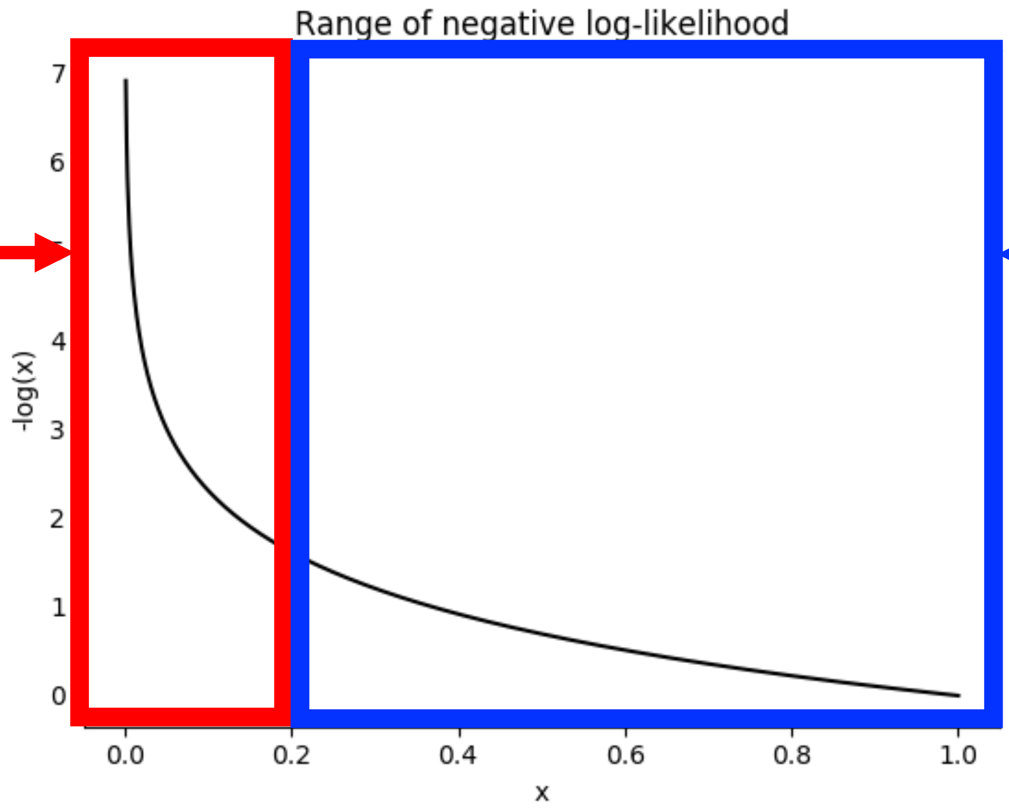
Fast R-CNN Training: Multi-task Loss

Objective function sums classification and localization losses for each region proposal



Fast R-CNN Training: Classification Loss (Recall Cross Entropy Loss, aka Log Loss)

Greater penalty when predicted probability of true class is confidently wrong



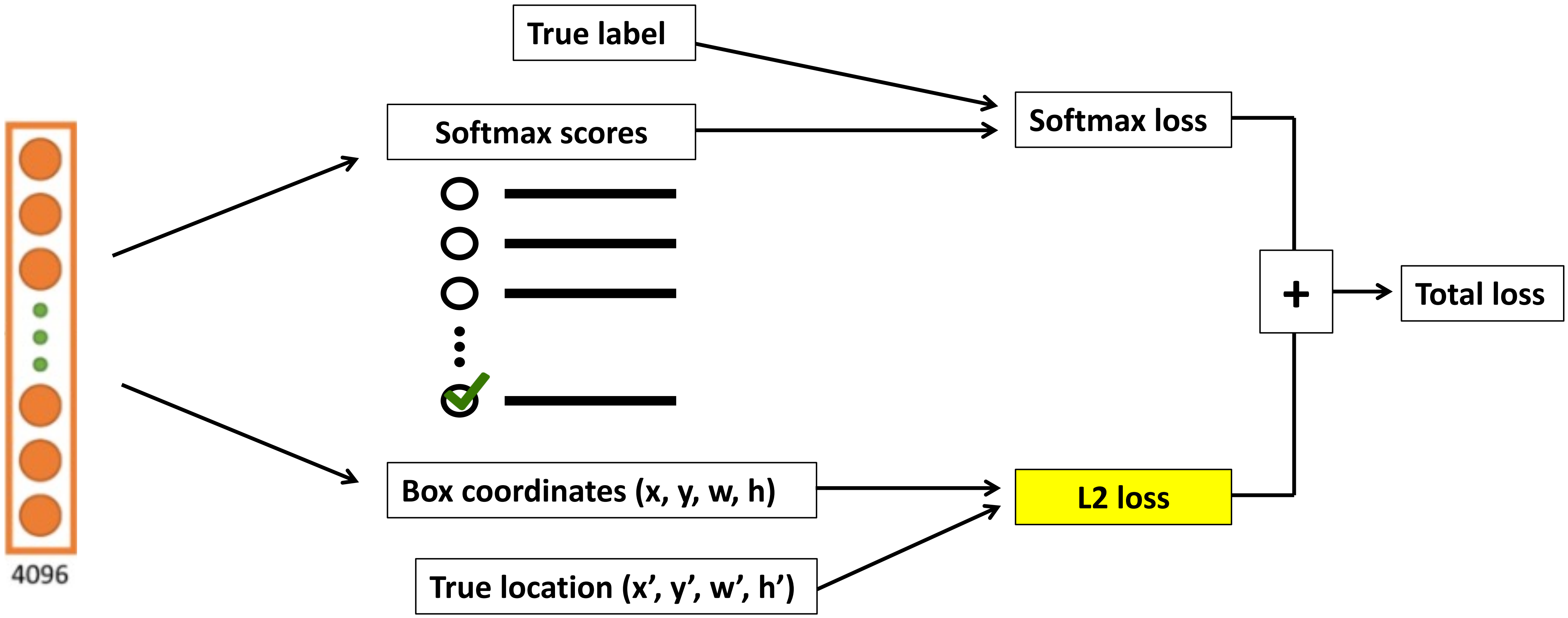
Lesser penalty otherwise

$$-\log \frac{\exp\left(w_k \cdot x + b_k\right)}{\sum_{j=1}^{K} \exp\left(w_j \cdot x + b_j\right)}$$

Figure source: https://ljvmiranda921.github.io/notebook/2017/08/13/softmax-and-the-negative-log-likelihood/

Fast R-CNN Training: Multi-task Loss

Objective function sums classification and localization losses for each region proposal



Fast R-CNN Training: Measure Localization Loss

$$\mathcal{L}_{\text{box}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} L_1^{\text{smooth}}(t_i^u - v_i) \rightarrow L_1^{\text{smooth}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

True location for true class “u”

Predicted location for class u

Less sensitive to outliers than SSE

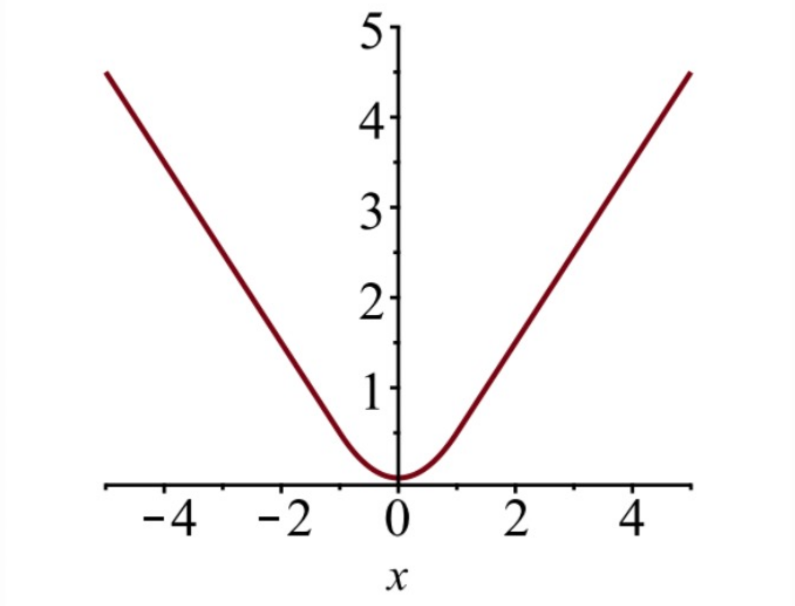
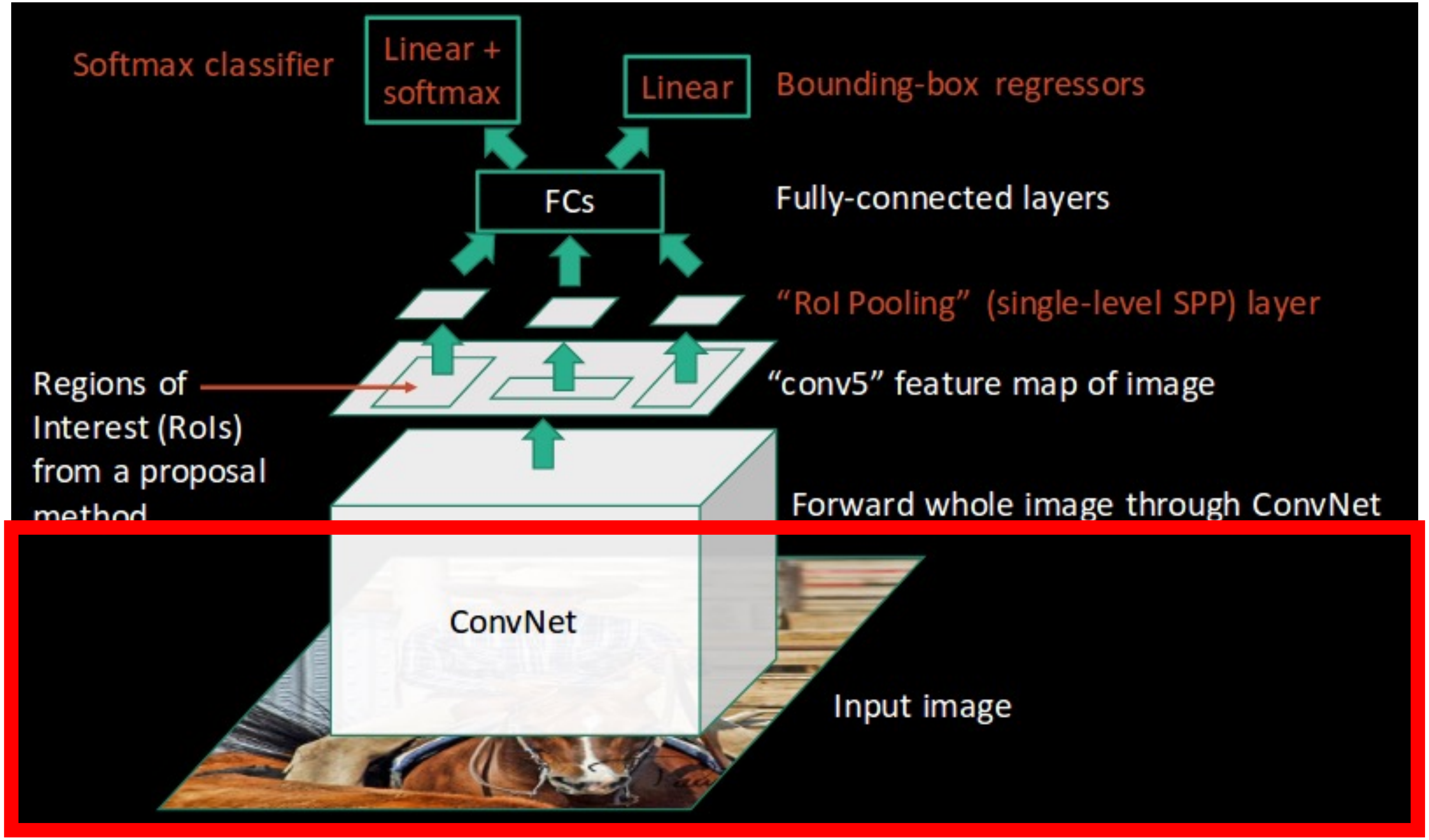


Image Source: https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html#bounding-box-regression

Fast R-CNN: Limitation



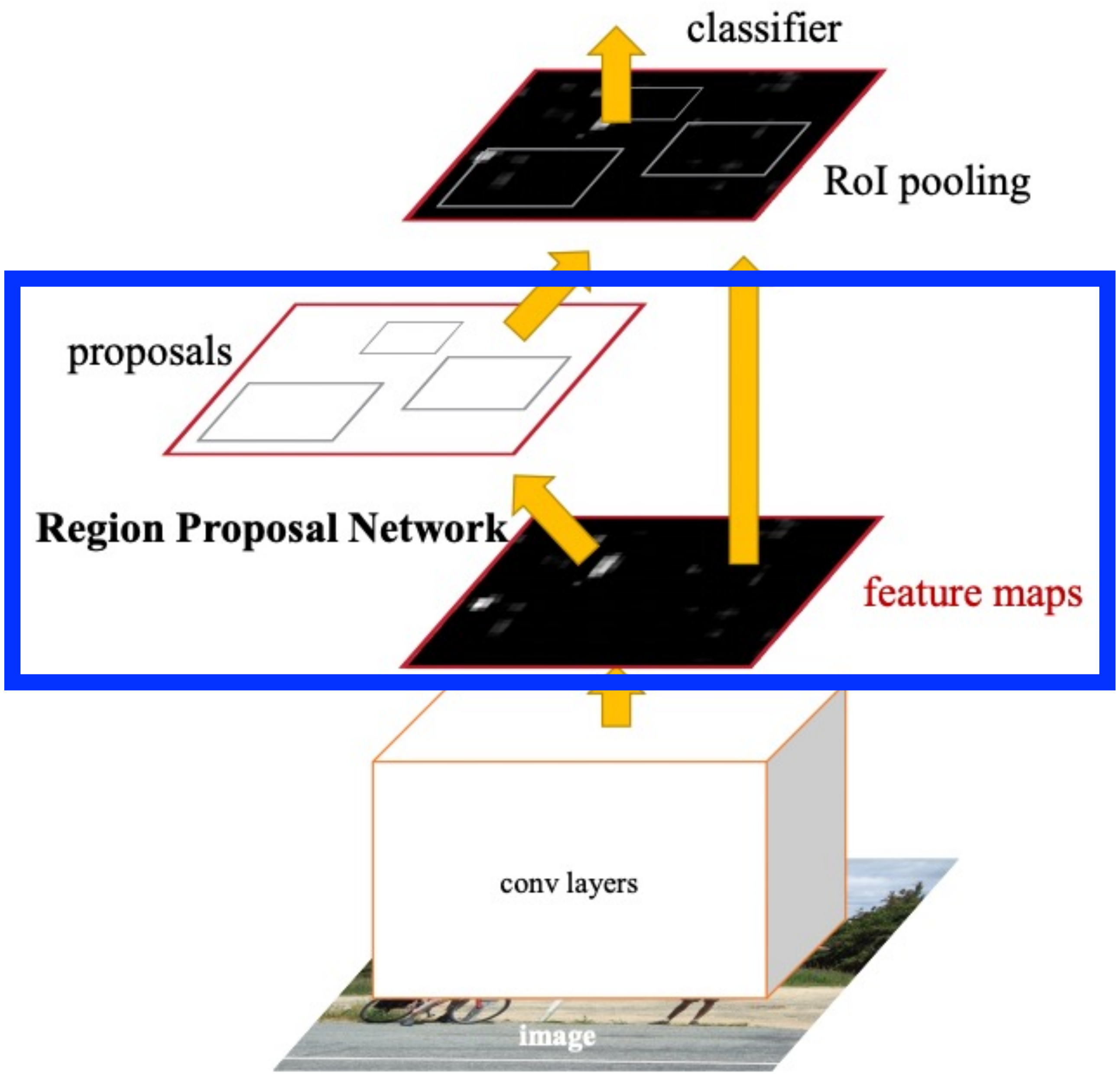
Still requires slow, initial step of generating region proposals

Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Faster R-CNN

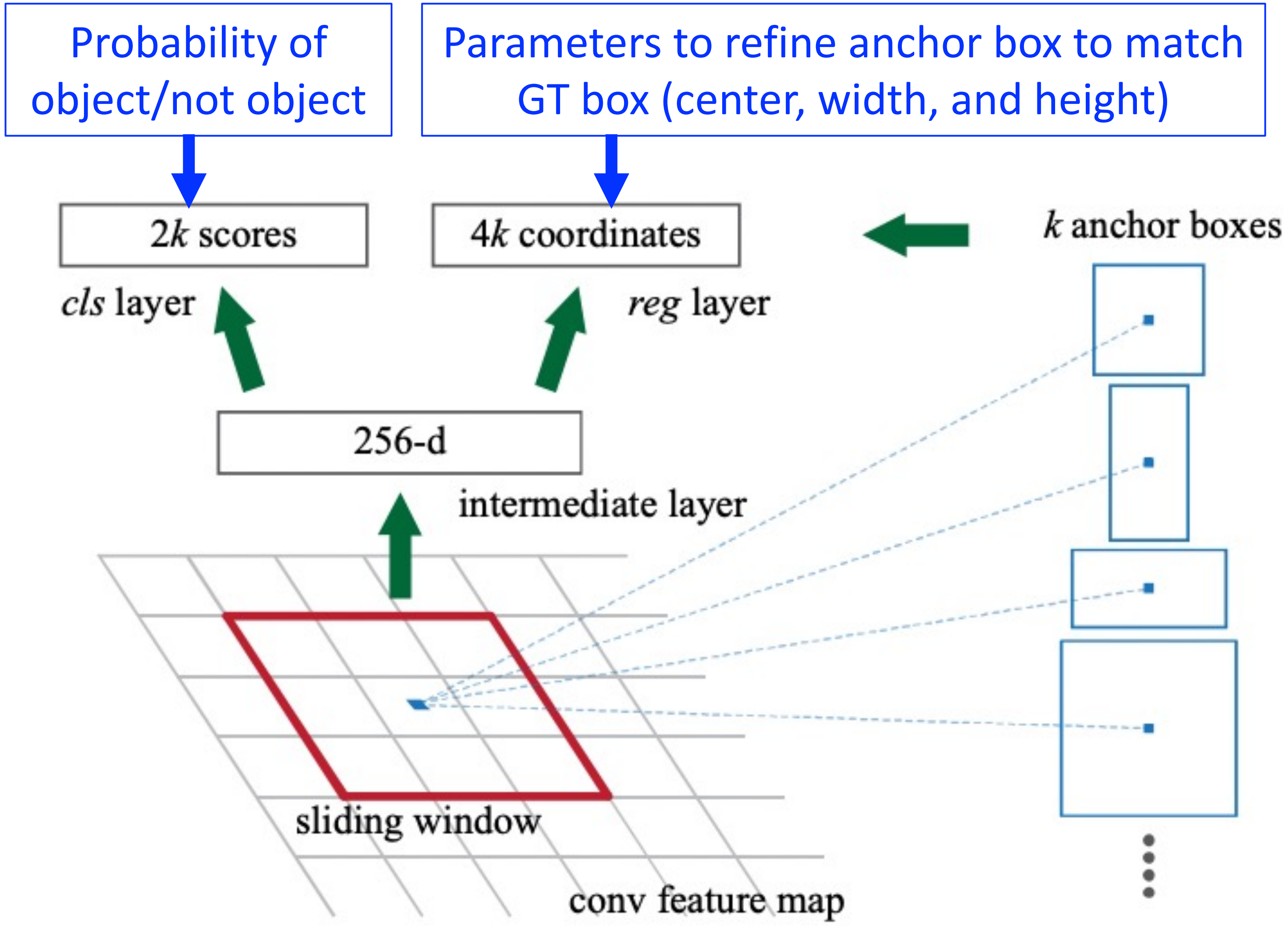
Adds finding region proposals to network so that all parts of model are learned in end-to-end fashion

Convolutional layers are shared for region proposal and detection



Ren Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” Neurips 2015.

Faster R-CNN: Region Proposal Network



Based on convolution, so uses sliding window

- At each sliding window position, region proposals are predicted with respect to an anchor point (i.e., center of sliding window position)
- At each anchor point, k = 9 anchors are used to represent 3 scales and 3 aspect ratios

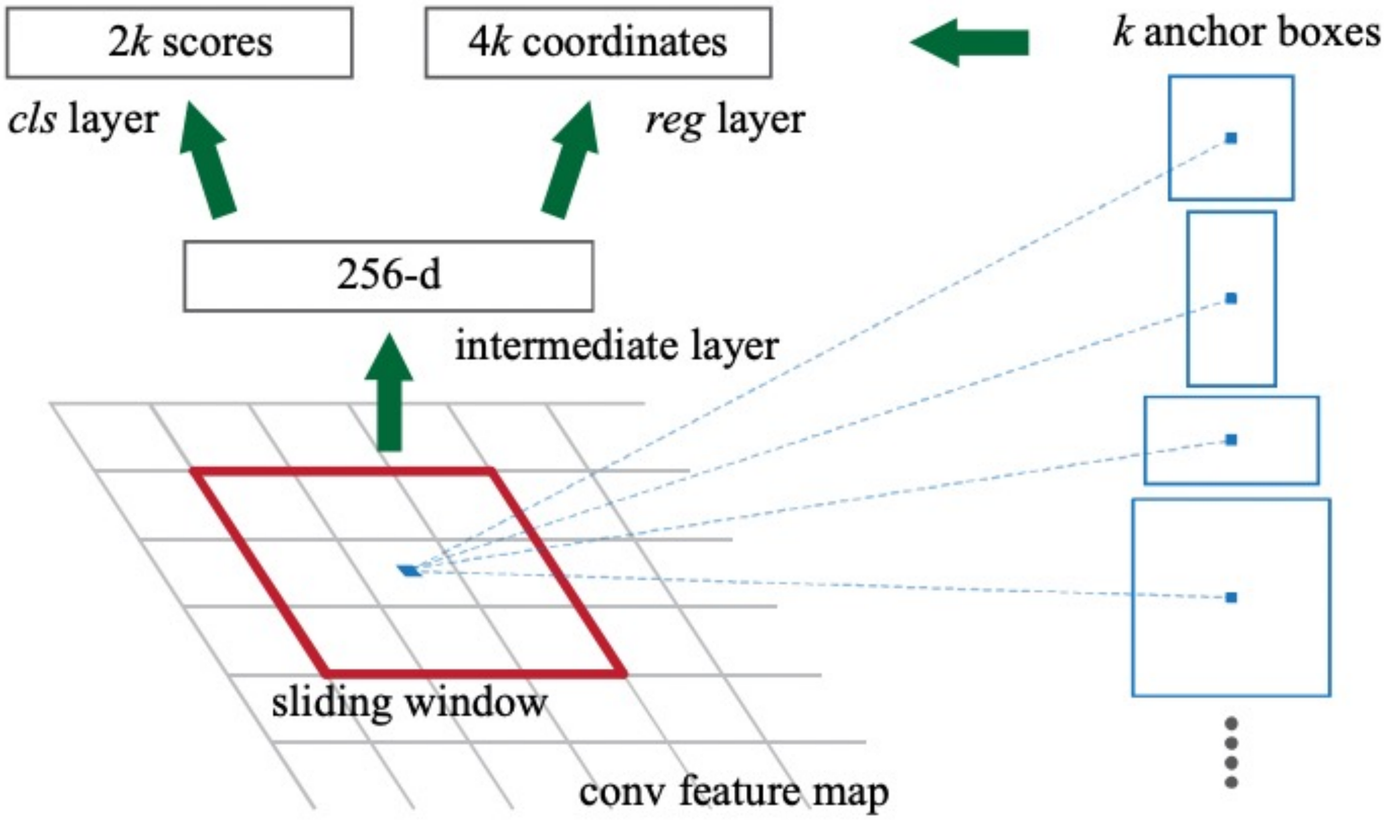
Ren Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Neurips 2015.

Faster R-CNN: Region Proposal Network

At training, loss for each region proposal is sum of classification and localization losses

Based on convolution, so uses sliding window

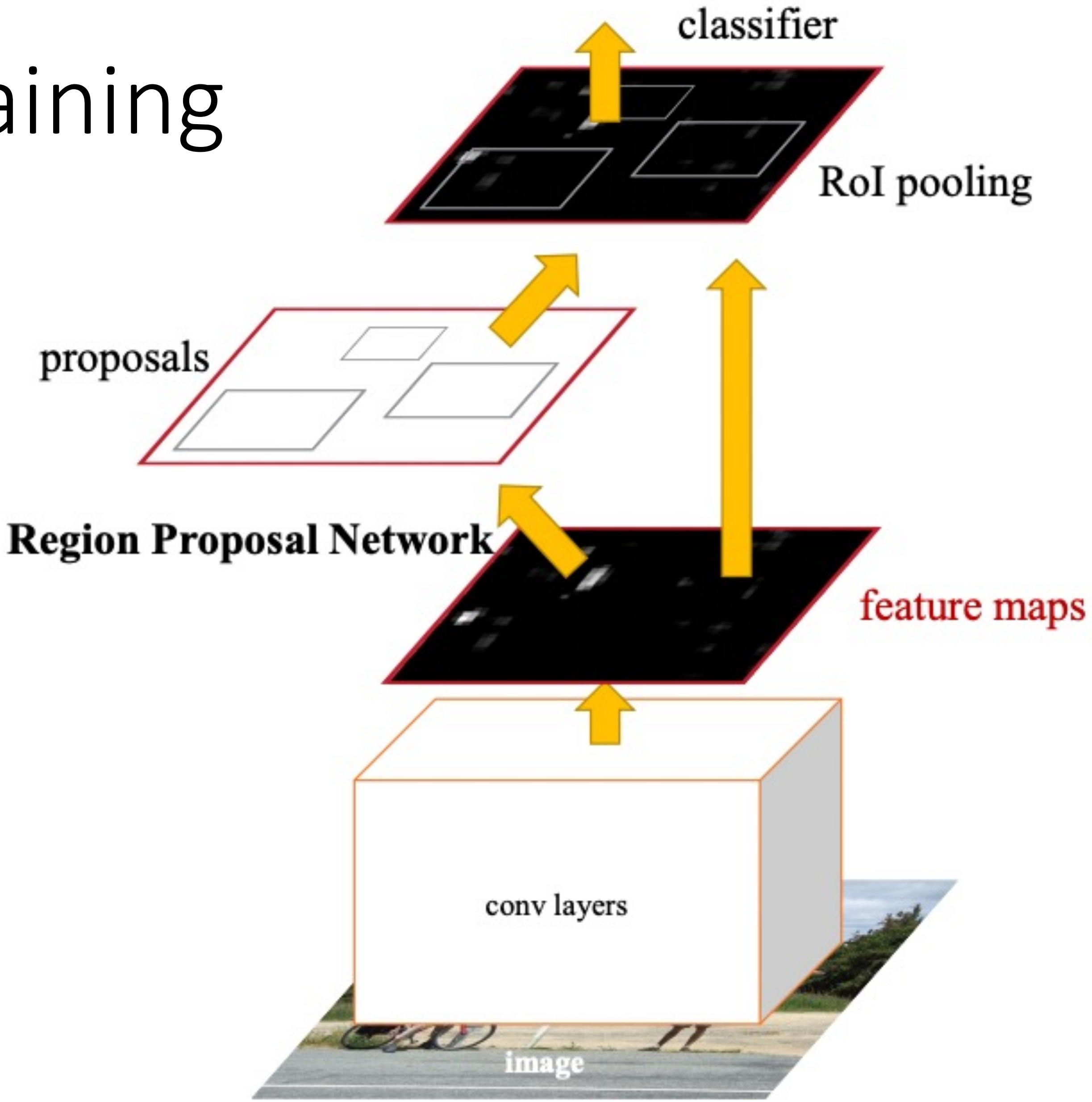
- At each sliding window position, region proposals are predicted with respect to an anchor point (i.e., center of sliding window position)
- At each anchor point, k = 9 anchors are used to represent 3 scales and 3 aspect ratios



Ren Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” Neurips 2015.

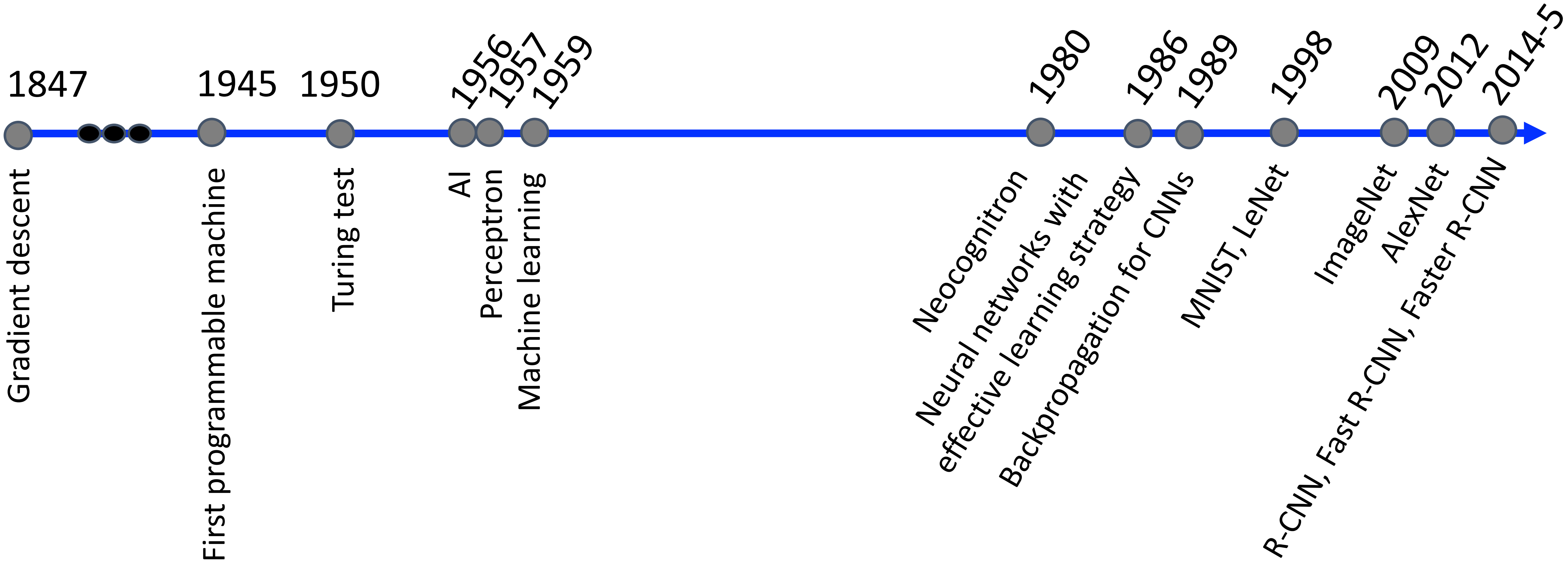
Faster R-CNN Training

1. Train RPN
2. Train Fast R-CNN using proposals from pretrained RPN
3. Fine-tune layers unique to RPN
4. Fine-tune the fully connected layers of Fast R-CNN



Ren Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” Neurips 2015.

Historical Context: In 2017, Mask R-CNN Introduced for Instance Segmentation

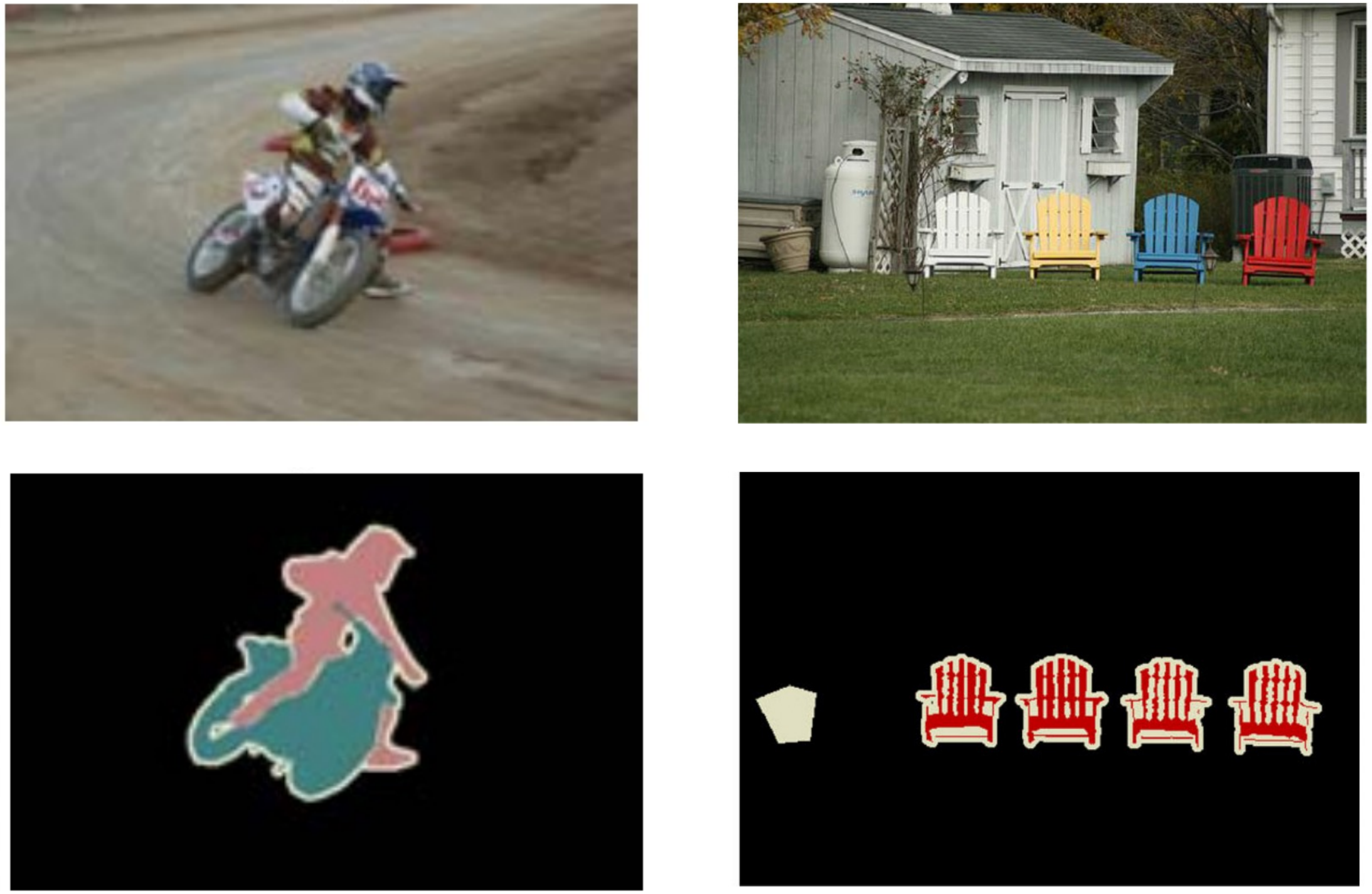


Today’s Topics

- Problems
- Applications
- PASCAL VOC detection challenge: R-CNNs
- PASCAL VOC semantic segmentation challenge: fully convolutional networks

VOC Challenge

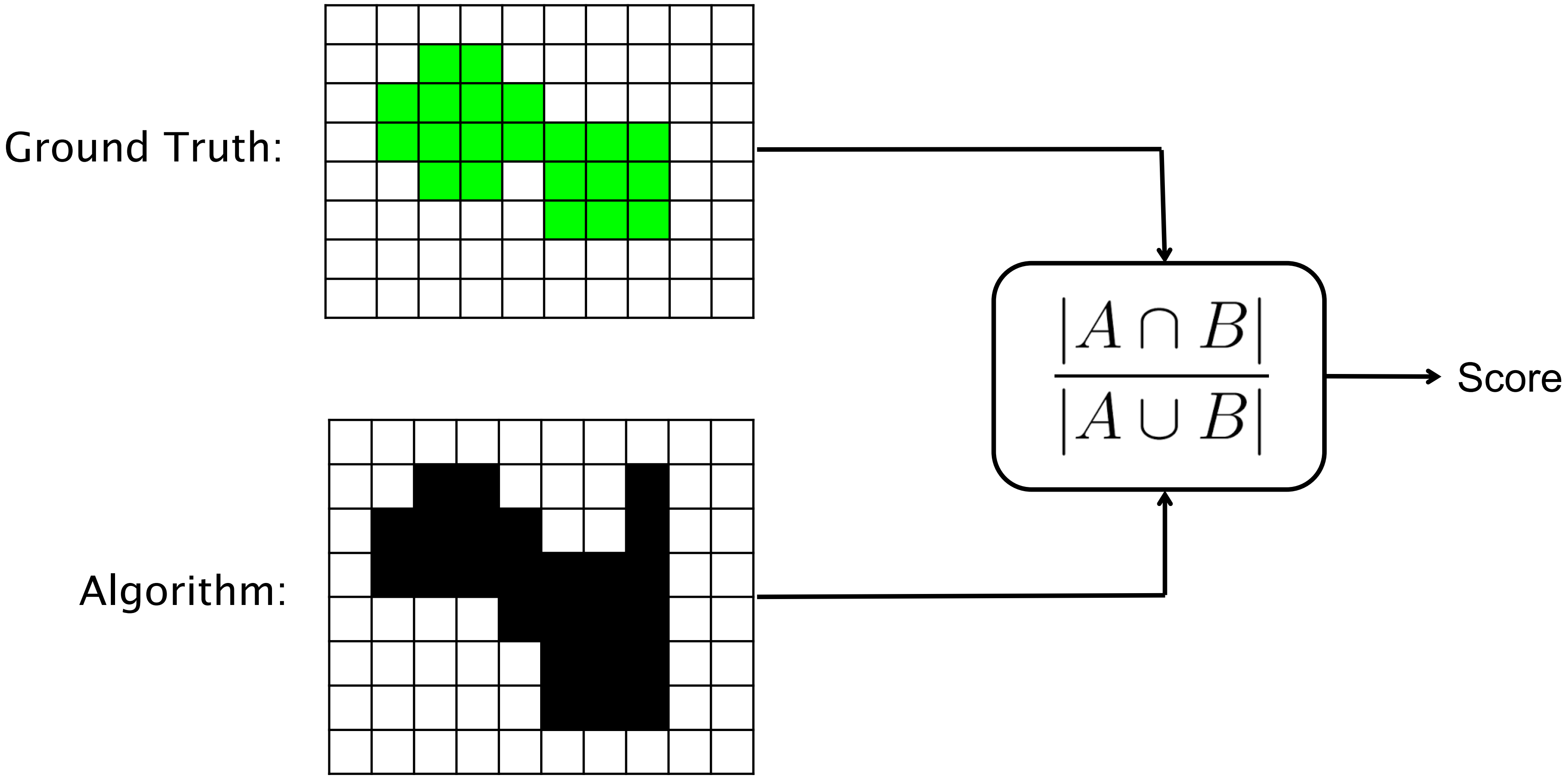
- Goal: locate all pixels belonging to 20 categories (e.g., person, cat, bus, mortorbike, potted plant, bottle) plus background
- Dataset: 11,530 images collected from Flickr and annotated by annotators at University of Leeds



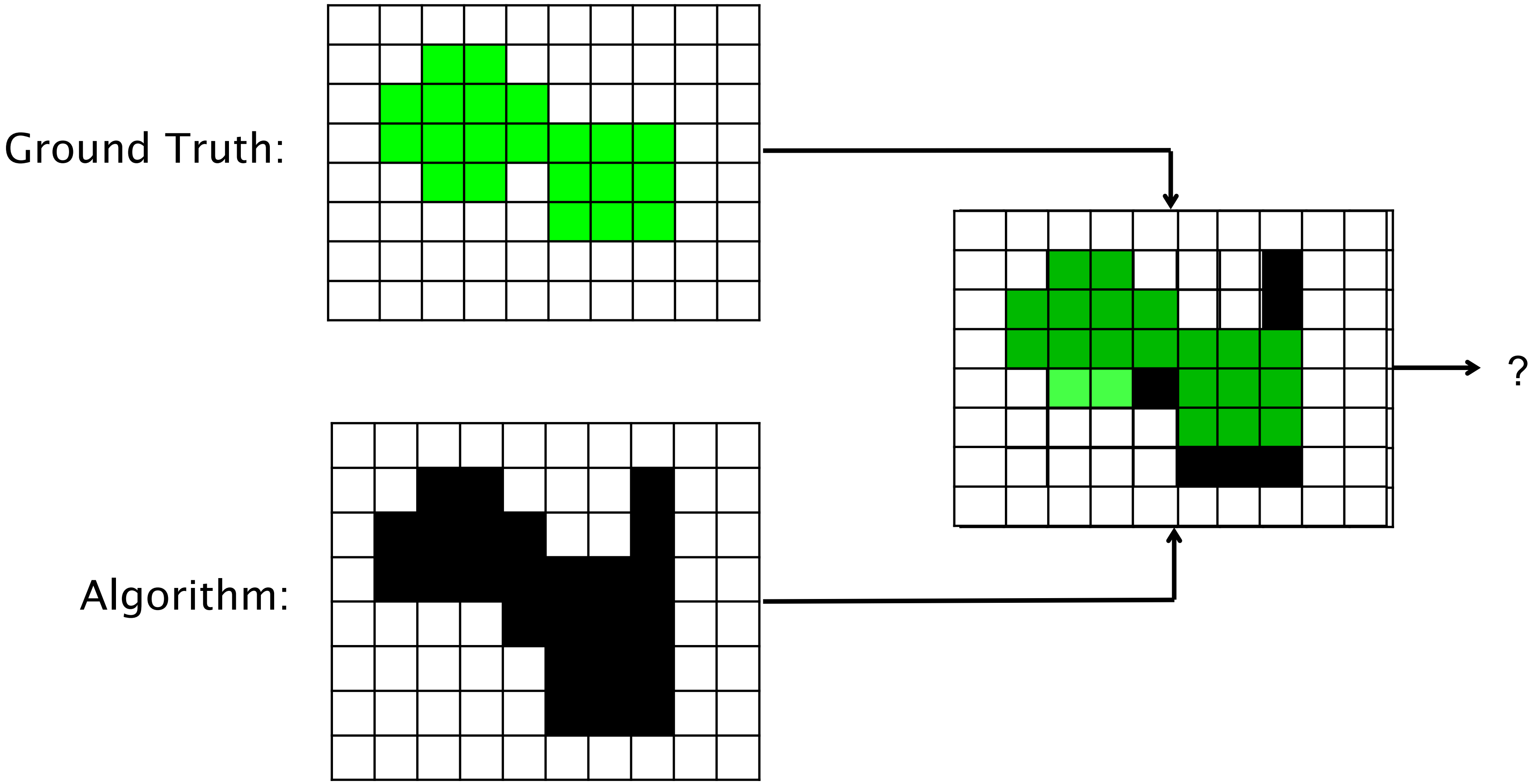
Dataset location: http://host.robots.ox.ac.uk/pascal/VOC/index.html

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. IJCV 2010.

VOC Challenge: Evaluation Metric (IoU)

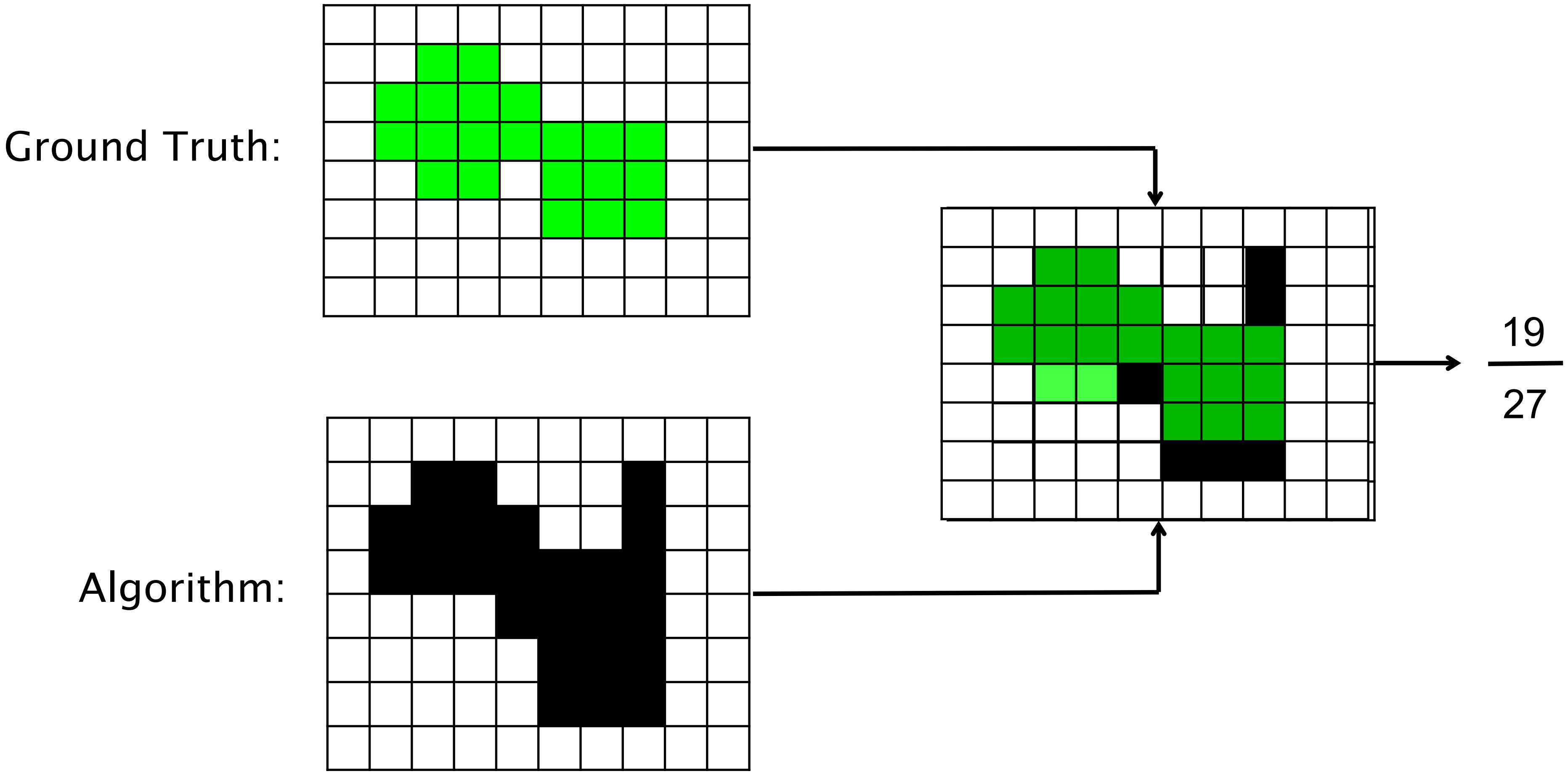


VOC Challenge: Evaluation Metric (IoU)



VOC Challenge: Evaluation Metric (IoU)

Mean IoU: IoU between predicted and ground-truth pixels, averaged over all 21 categories

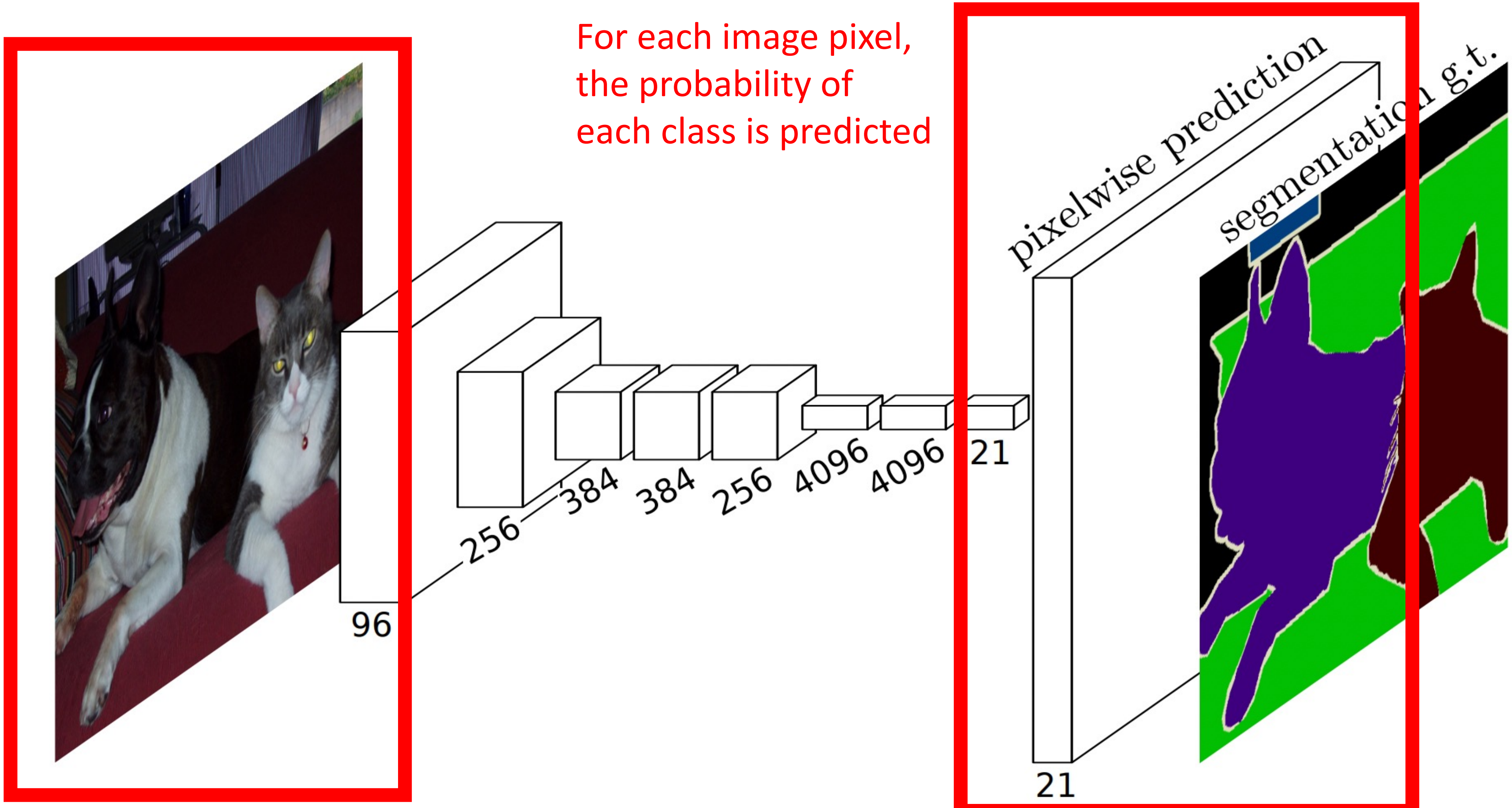


Architecture

Input: RGB image of ANY size

Output: Image of same size as input

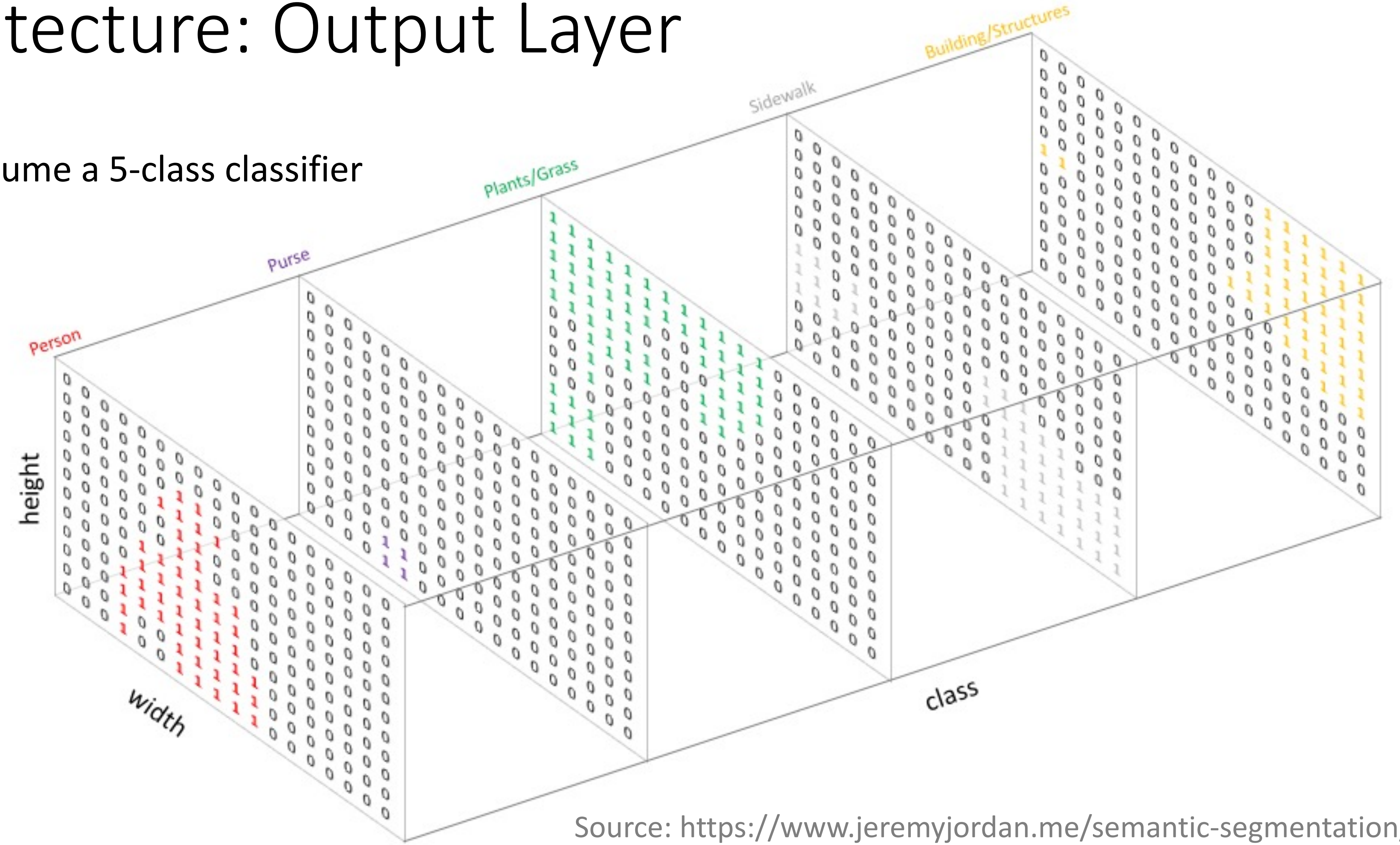
For each image pixel, the probability of each class is predicted



Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture: Output Layer

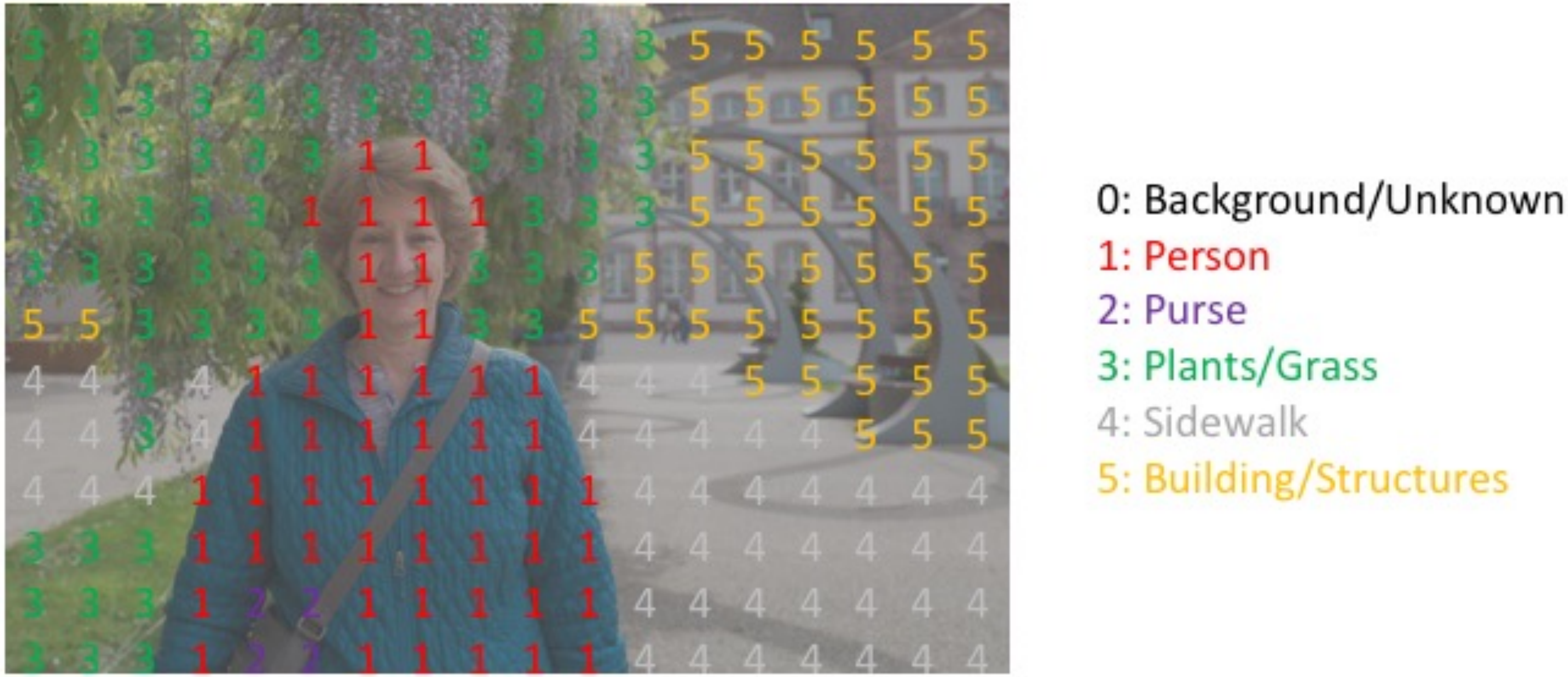
- e.g., assume a 5-class classifier



Source: https://www.jeremyjordan.me/semantic-segmentation/

Architecture: Output Layer

- e.g., assume a 5-class classifier; output 1-hot encoding collapsed into single mask image



Source: https://www.jeremyjordan.me/semantic-segmentation/

Input: RGB image of ANY size

Output: Image of same size as input

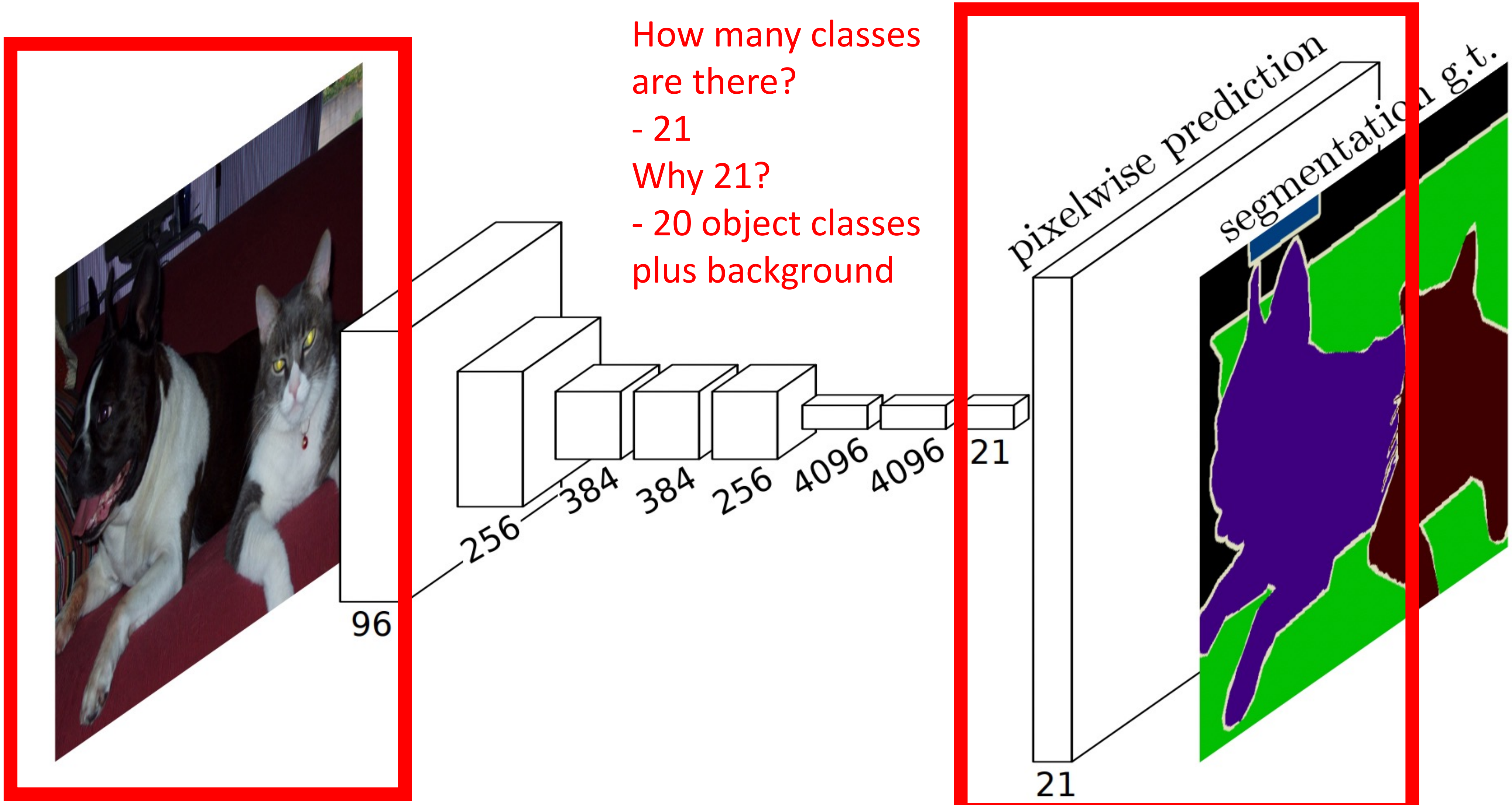
Architecture

How many classes are there?

- 21

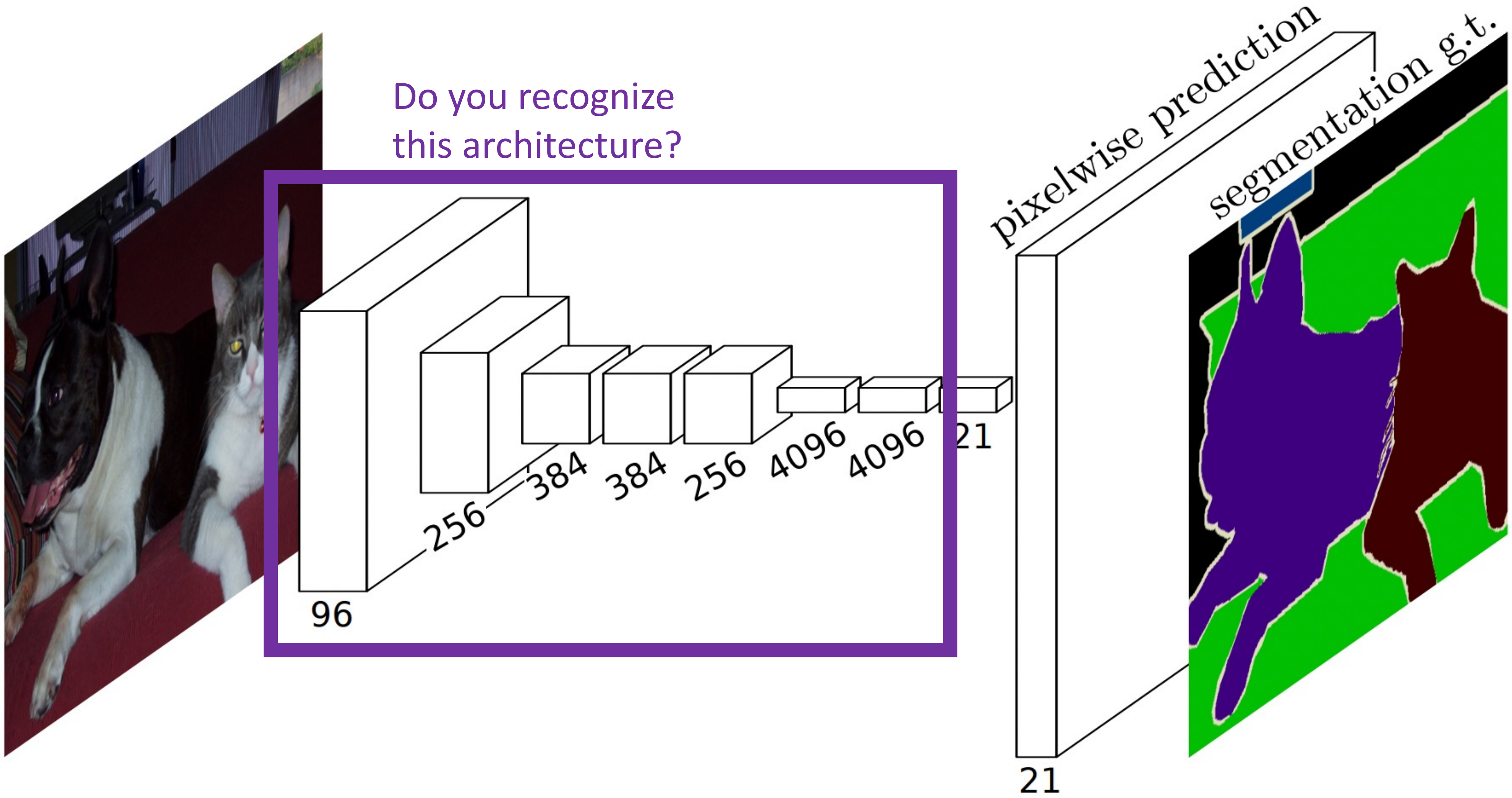
Why 21?

- 20 object classes plus background



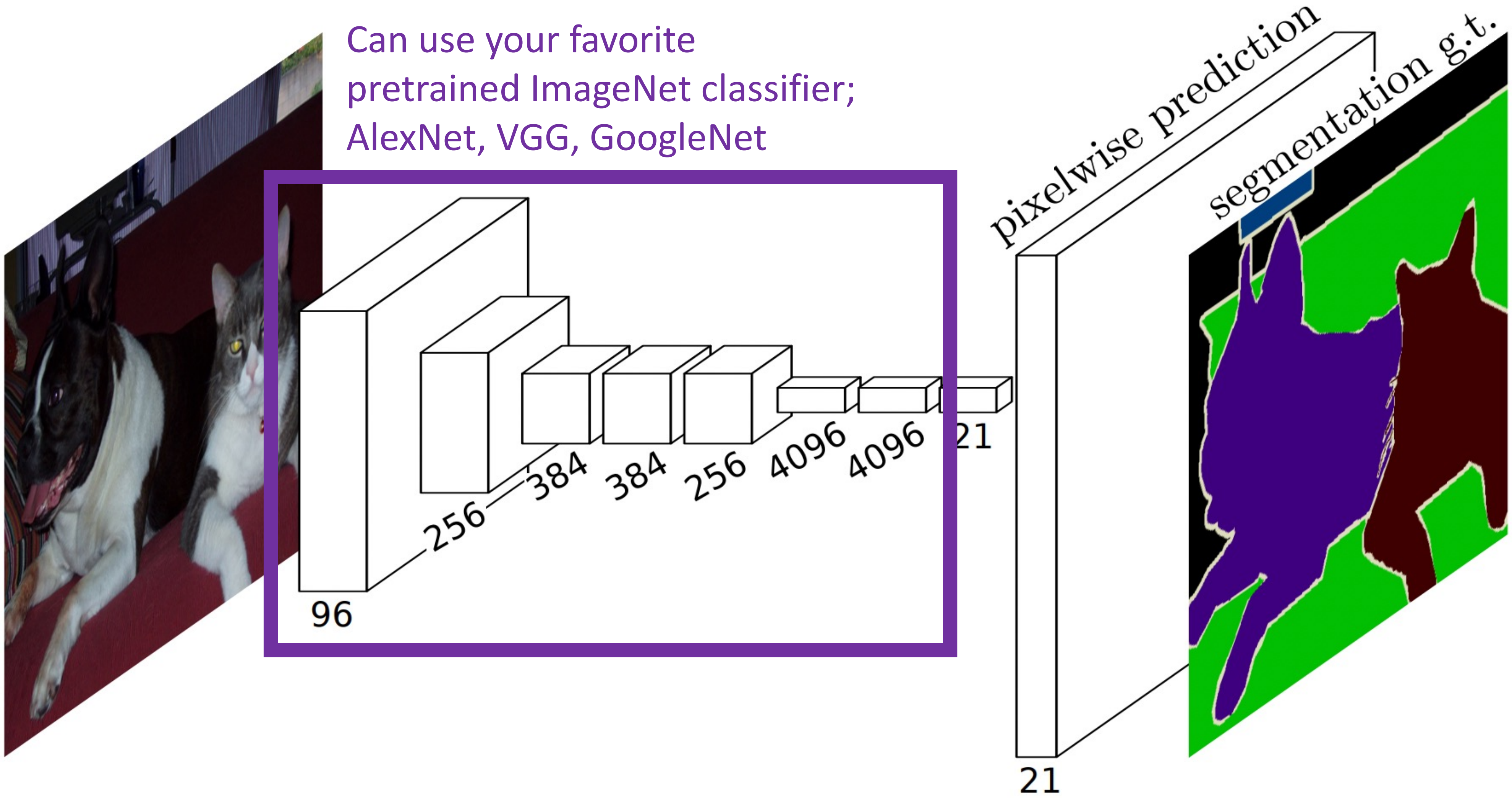
Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture



Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture

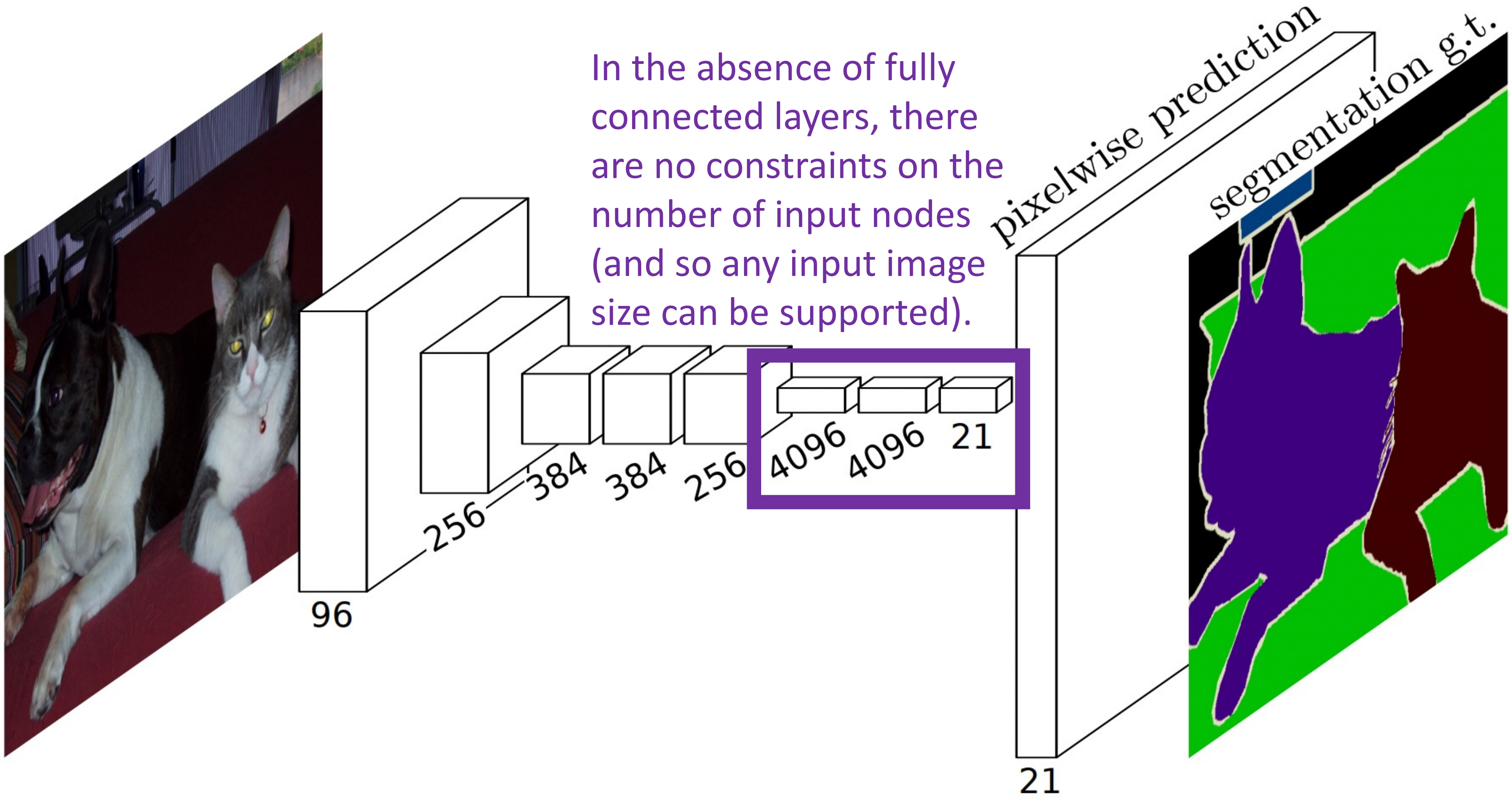


Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture

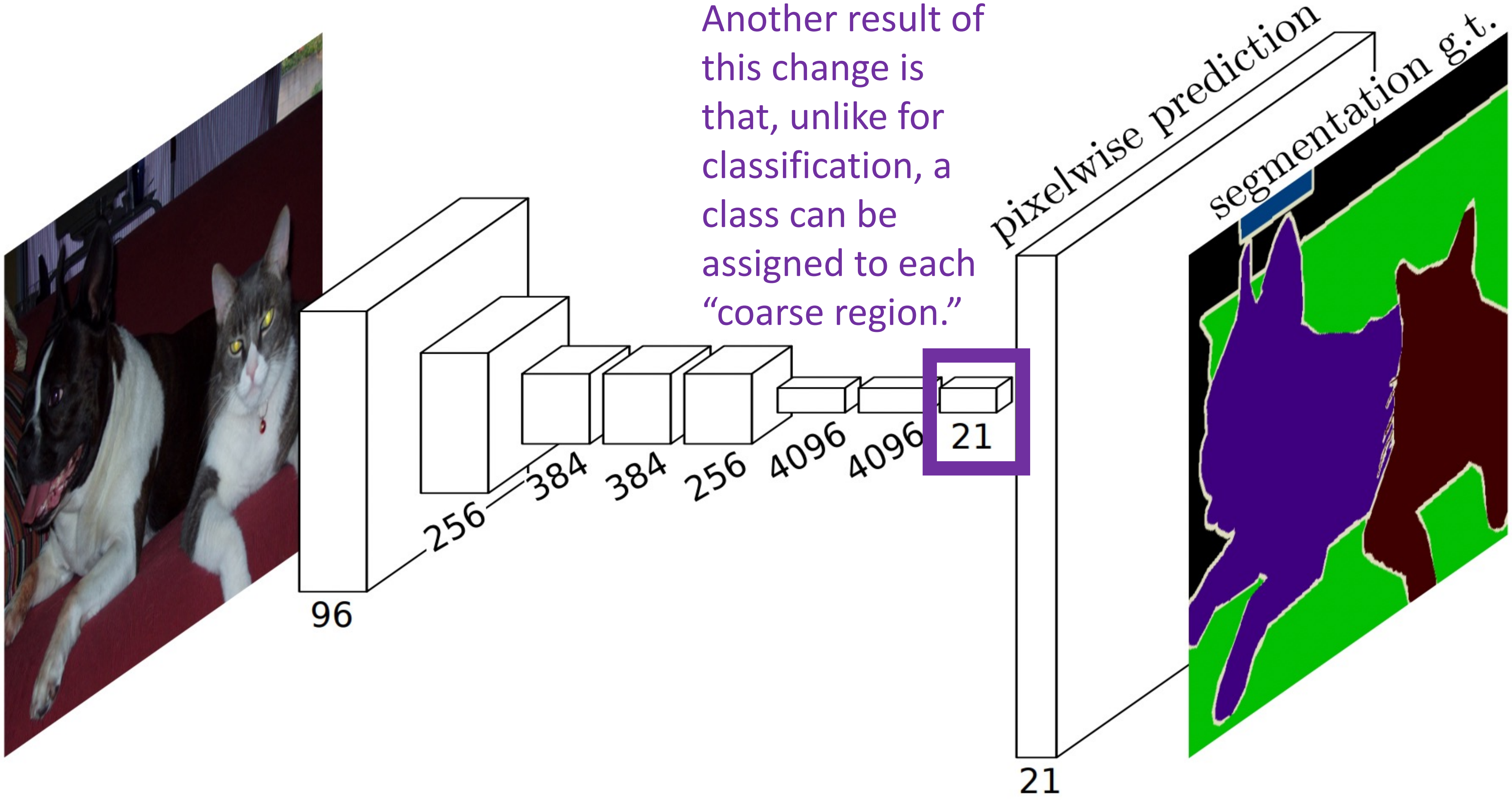
To make the architecture fully convolutional, fully connected layers are converted to convolutional layers.

In the absence of fully connected layers, there are no constraints on the number of input nodes (and so any input image size can be supported).



Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture

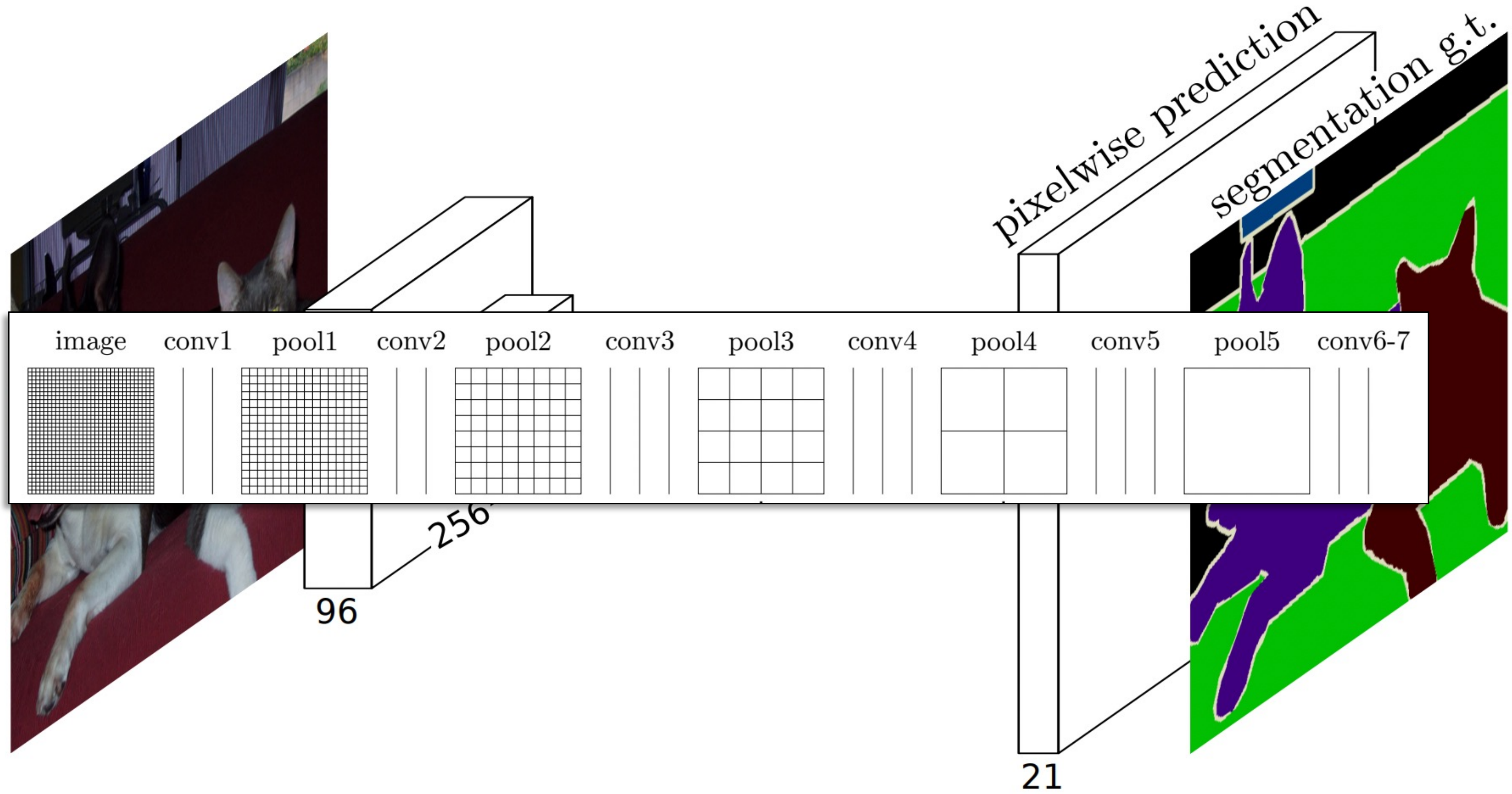


Another result of this change is that, unlike for classification, a class can be assigned to each "coarse region."

Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture: Coarse Region Classification (Recall Intuition)

Using VGG16 instead:

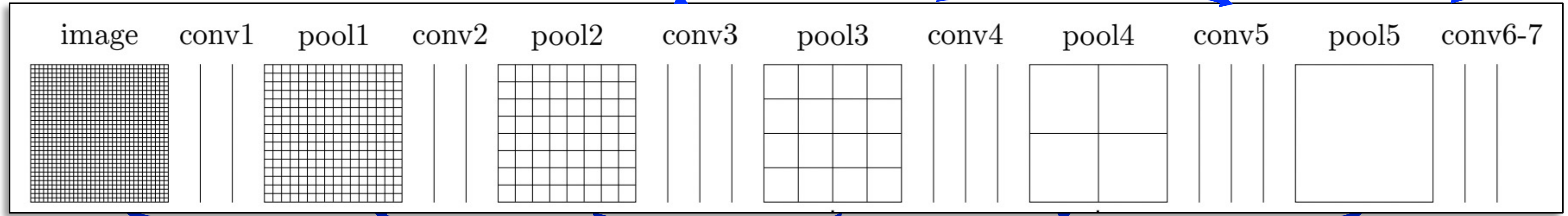


Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture: Coarse Region Classification (Recall Intuition)

Each line represents a convolutional layer

Using VGG16 instead:

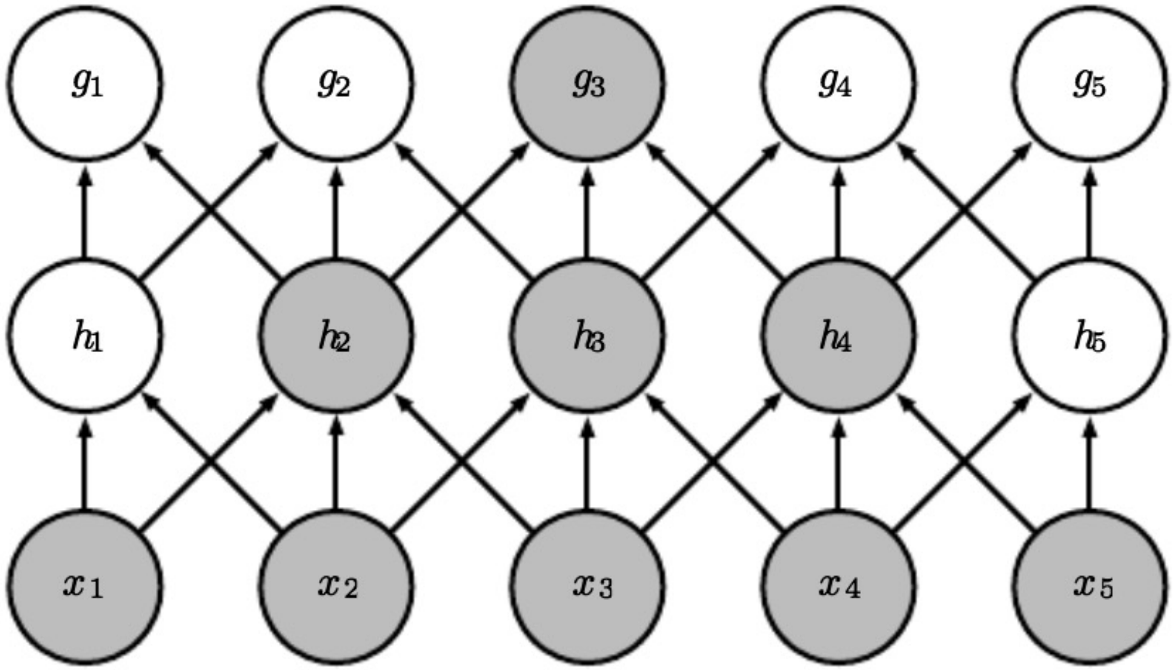


Grids reflect relative spatial coarseness at each layer

Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

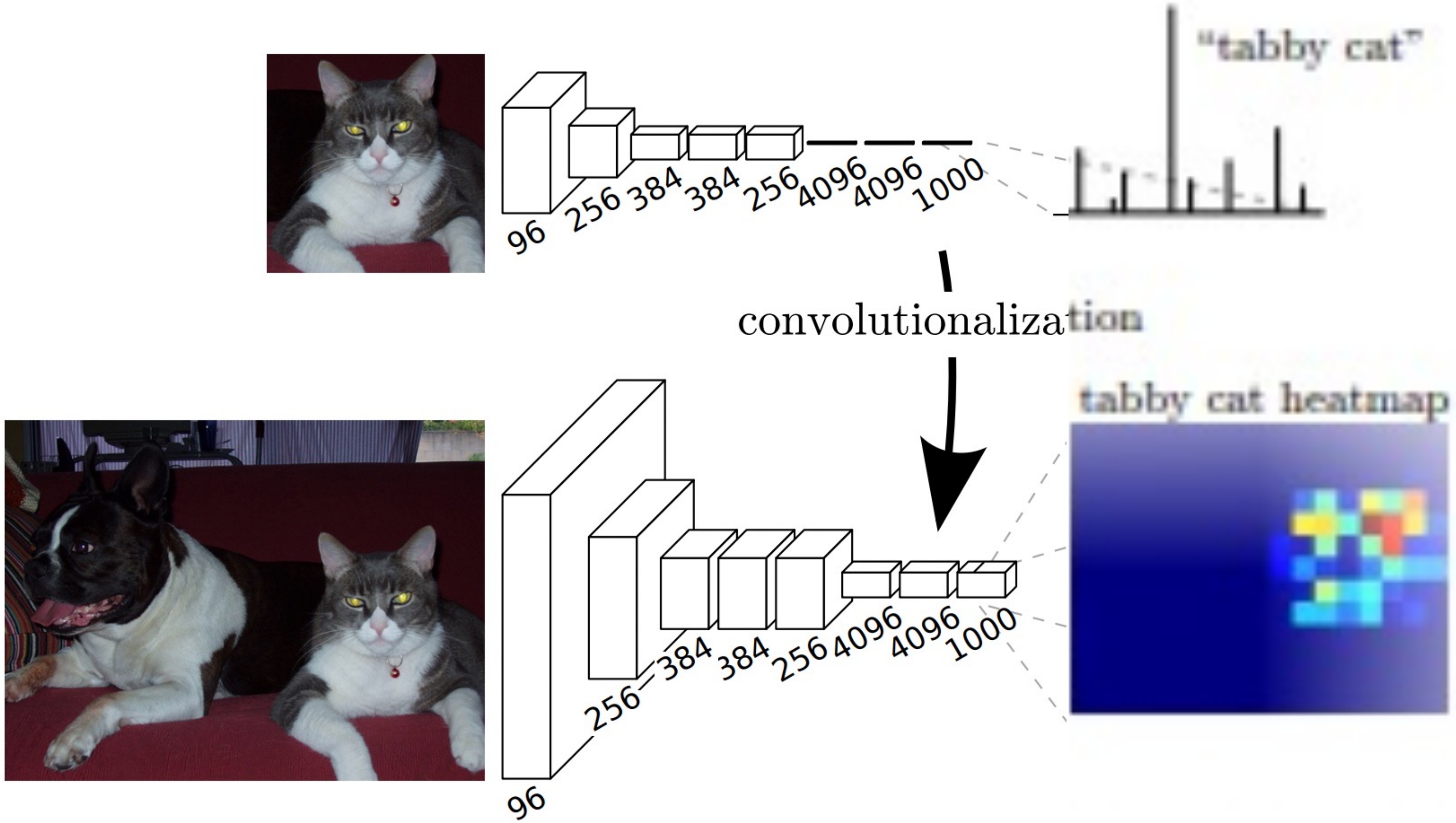
Architecture: Coarse Region Classification (Recall Intuition)

Stacking many convolutional layers leads to learning patterns in increasingly **larger regions of the input (e.g., pixel) space.**



https://www.deeplearningbook.org/contents/convnets.html

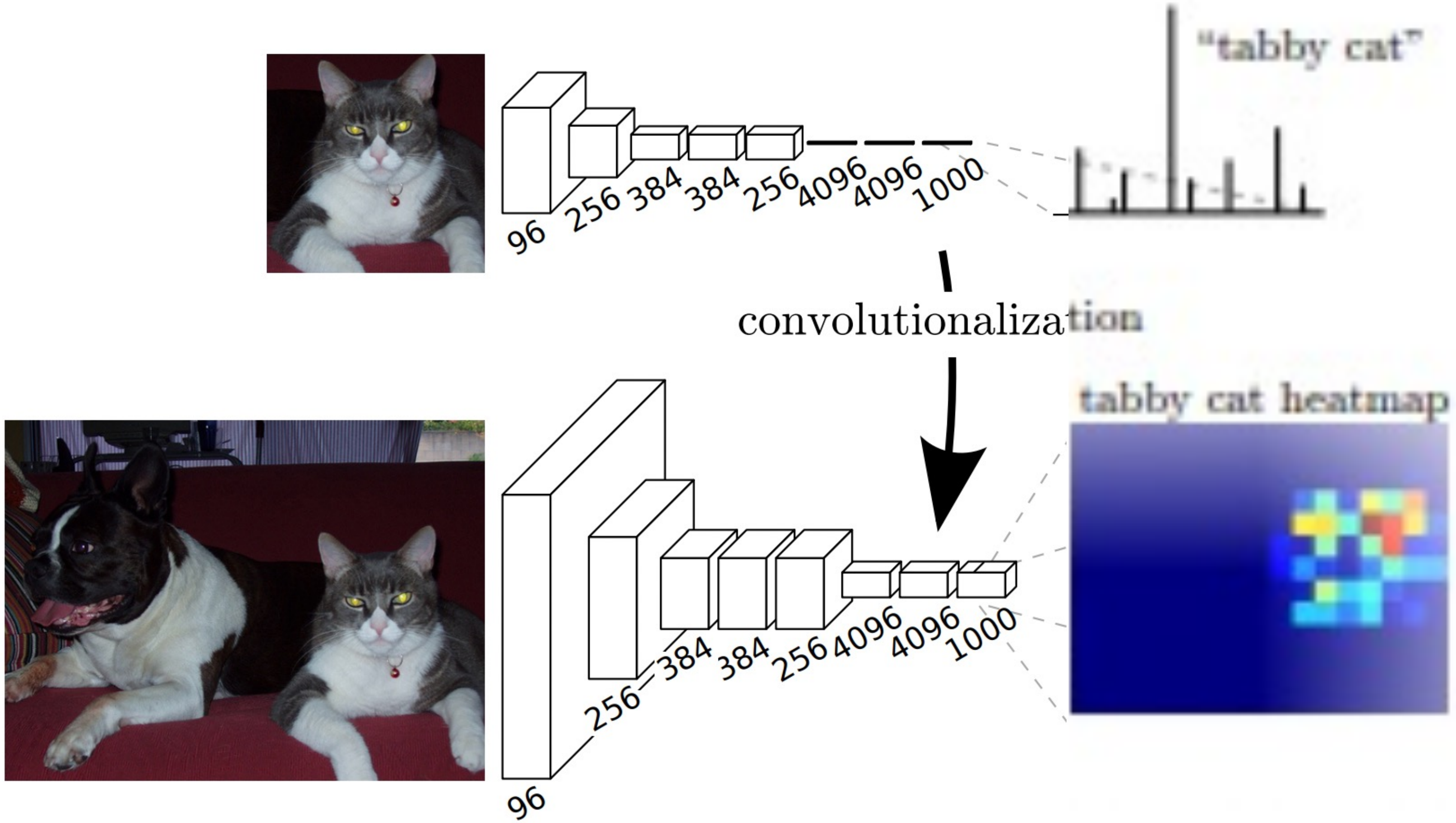
Architecture: Fully vs Convolution Layers



Each slice indicates the likelihood each pixel in the coarse region belongs to the class identified by the filter

Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

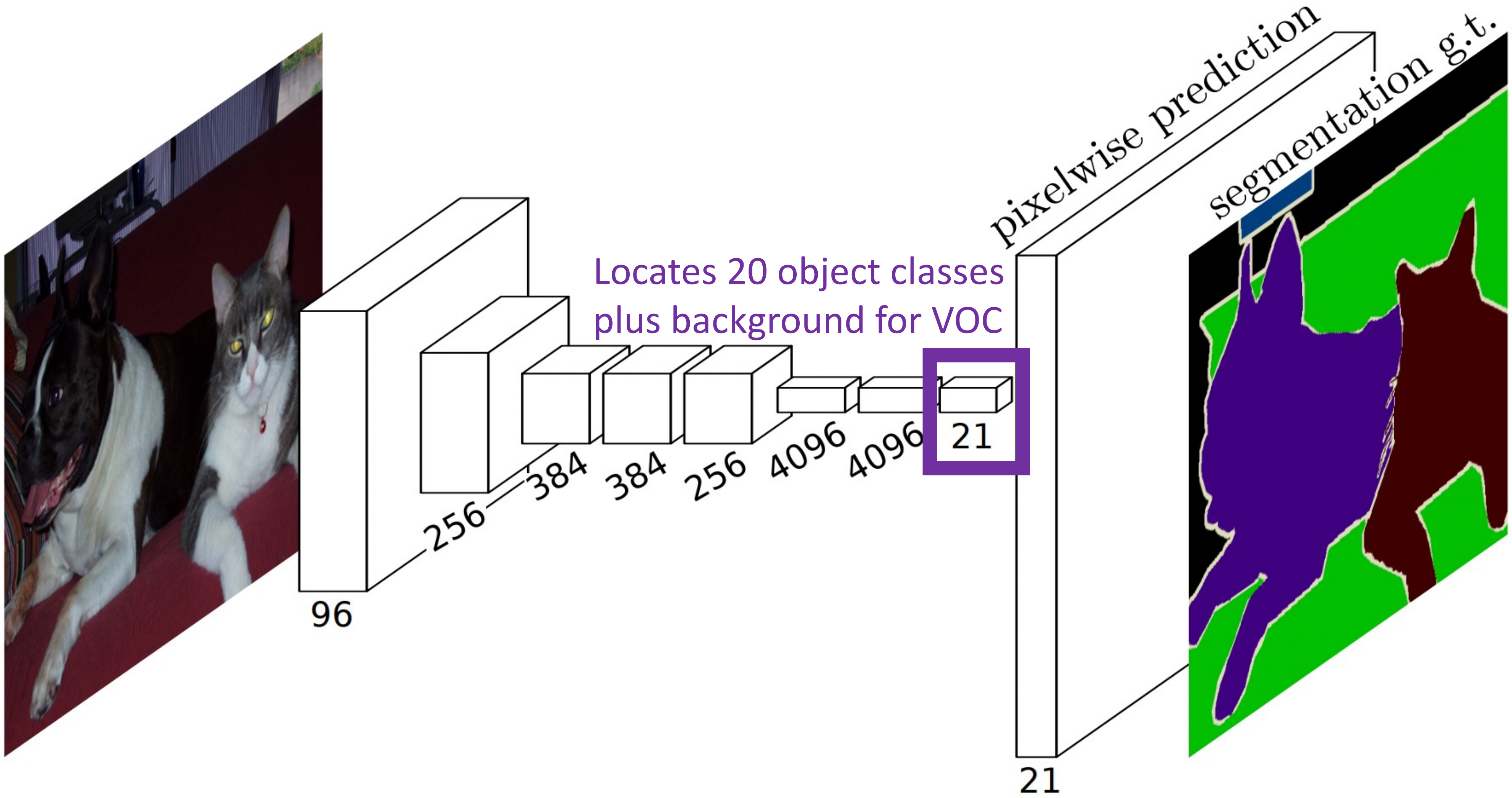
Architecture: Fully vs Convolution Layers



If convolutionizing ImageNet trained classifiers, how many classes would be predicted for each coarse region?

Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

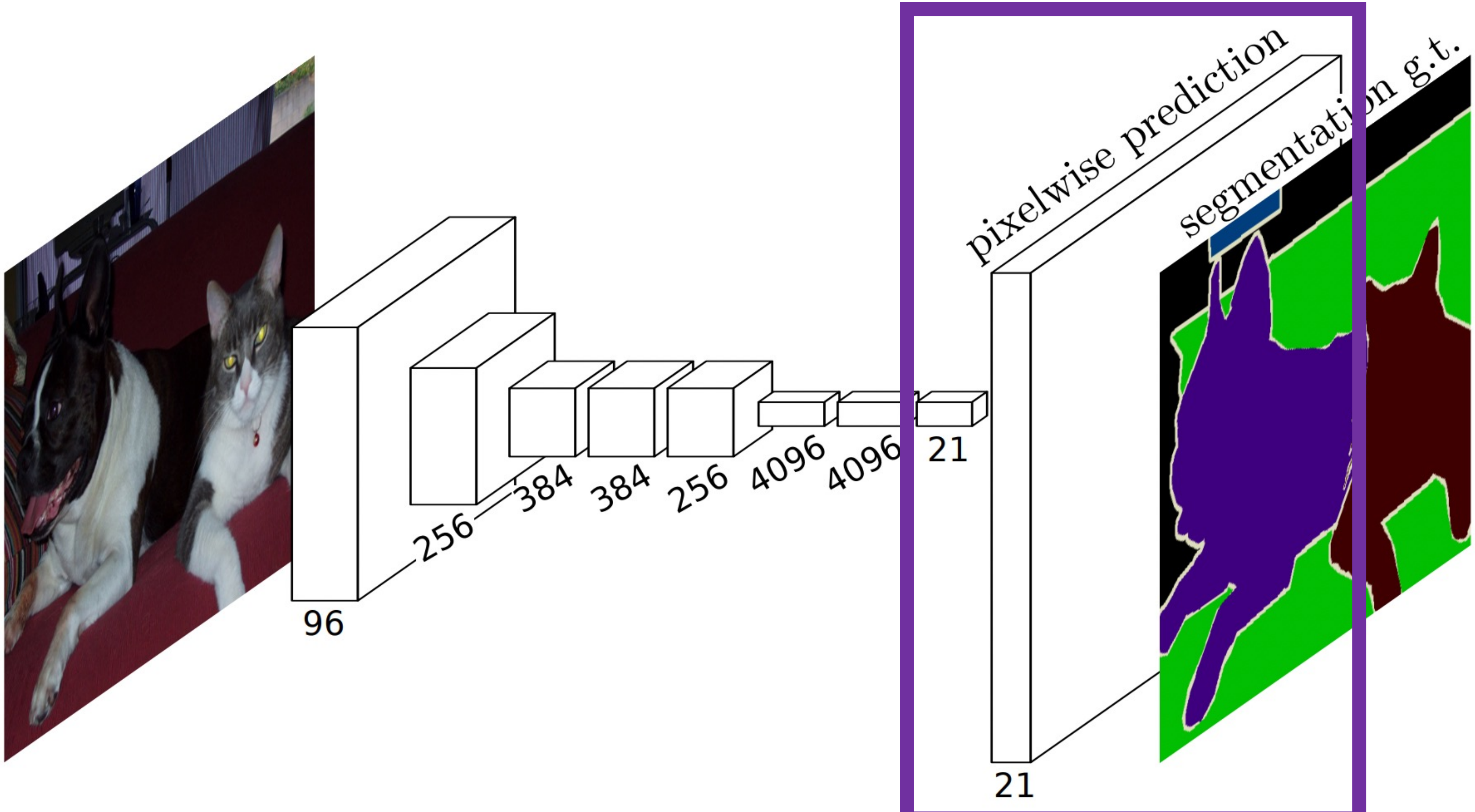
Architecture: Coarse Region Classification



Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

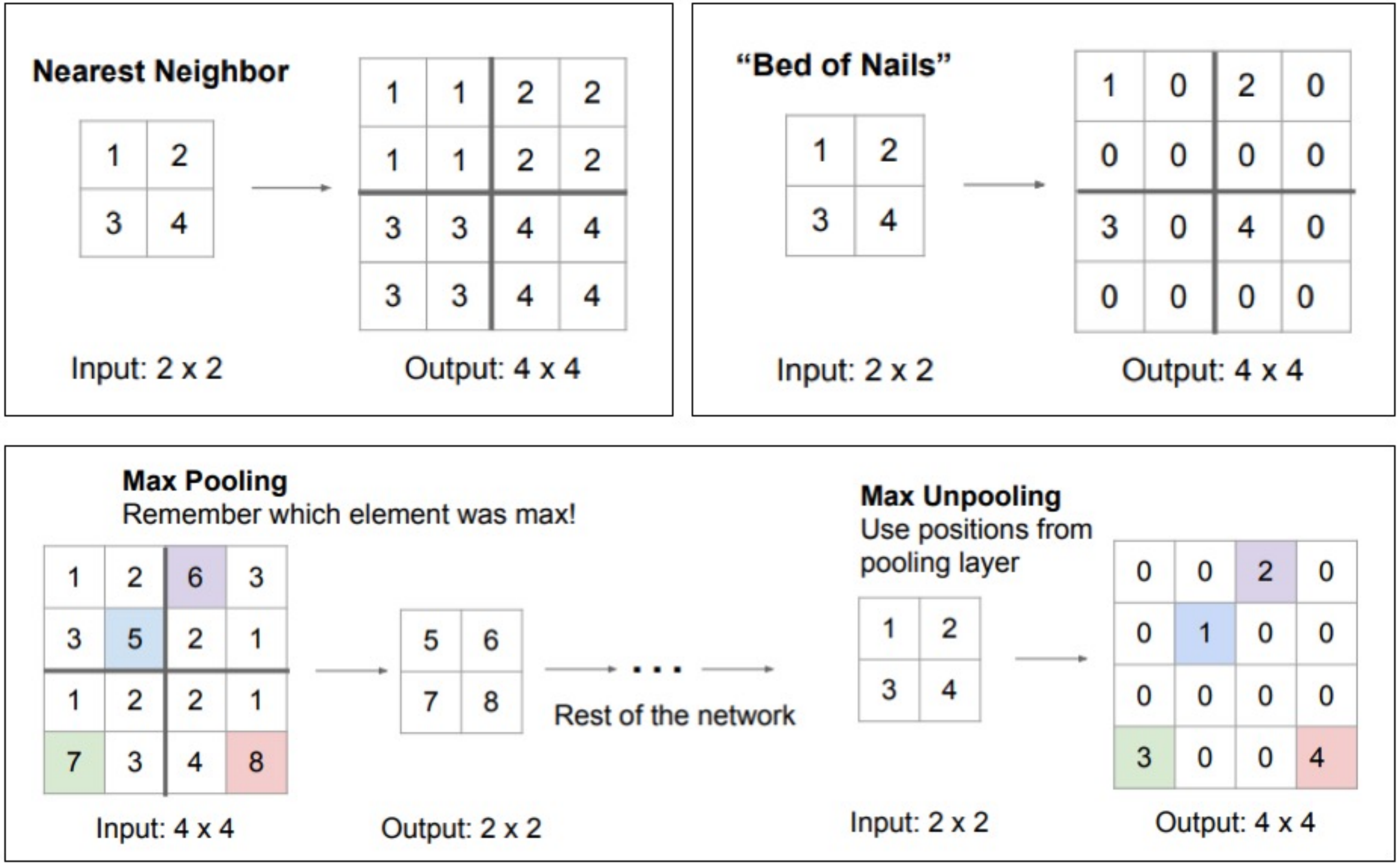
Architecture

Challenge: how to decode from coarse region classifications to per pixel classification?



Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture: Upsampling (Many Approaches)



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

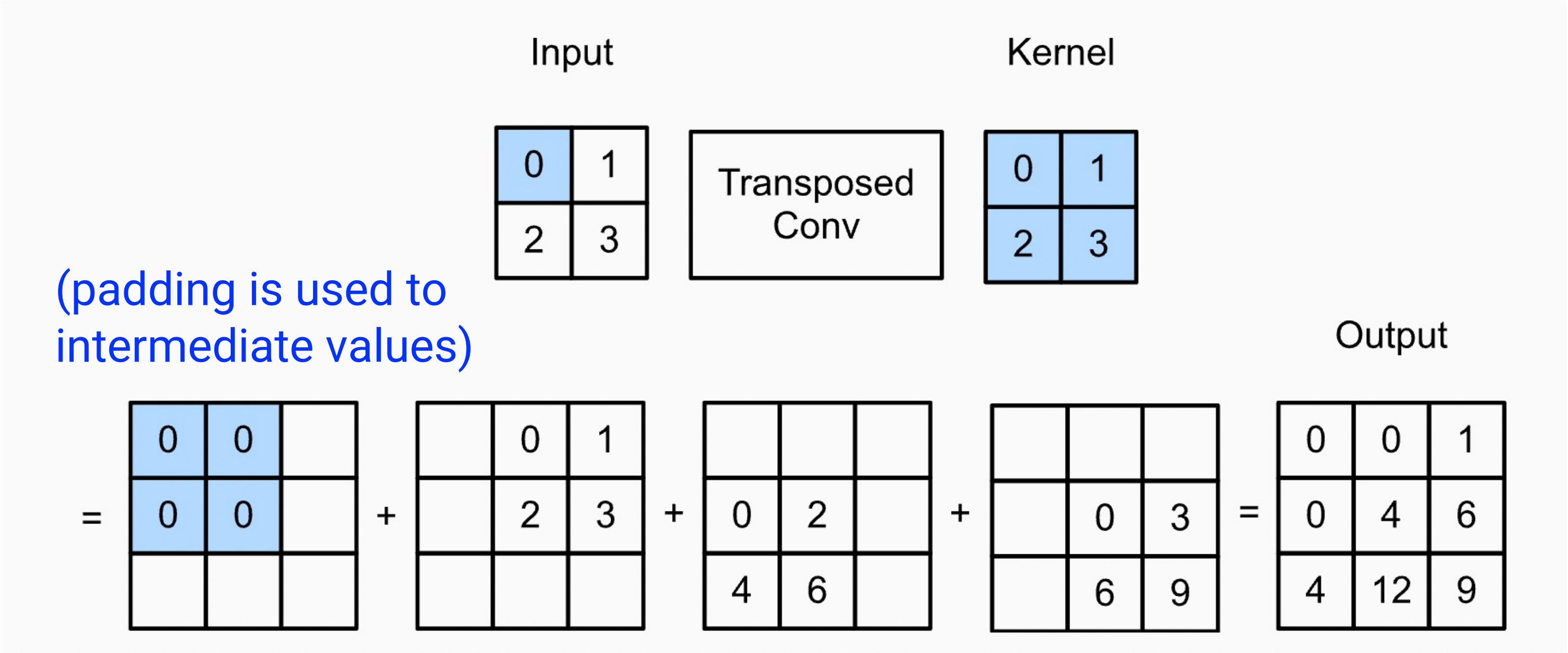
Architecture: Upsampling (Transposed Convolutional Layer)

- Also called “fractional convolutional layer”, “backward convolution”, and, incorrectly, ”deconvolution layer”
- Idea: learn filters with a fractional sized stride to upsample the coarse image while refining it based on the filter values; e.g.,

https://www.machinecurve.com/index.php/2019/09/29/understanding-transposed-convolutions/#the-goal-reconstructing-the-original-input

Architecture: Upsampling (Transposed Convolutional Layer)

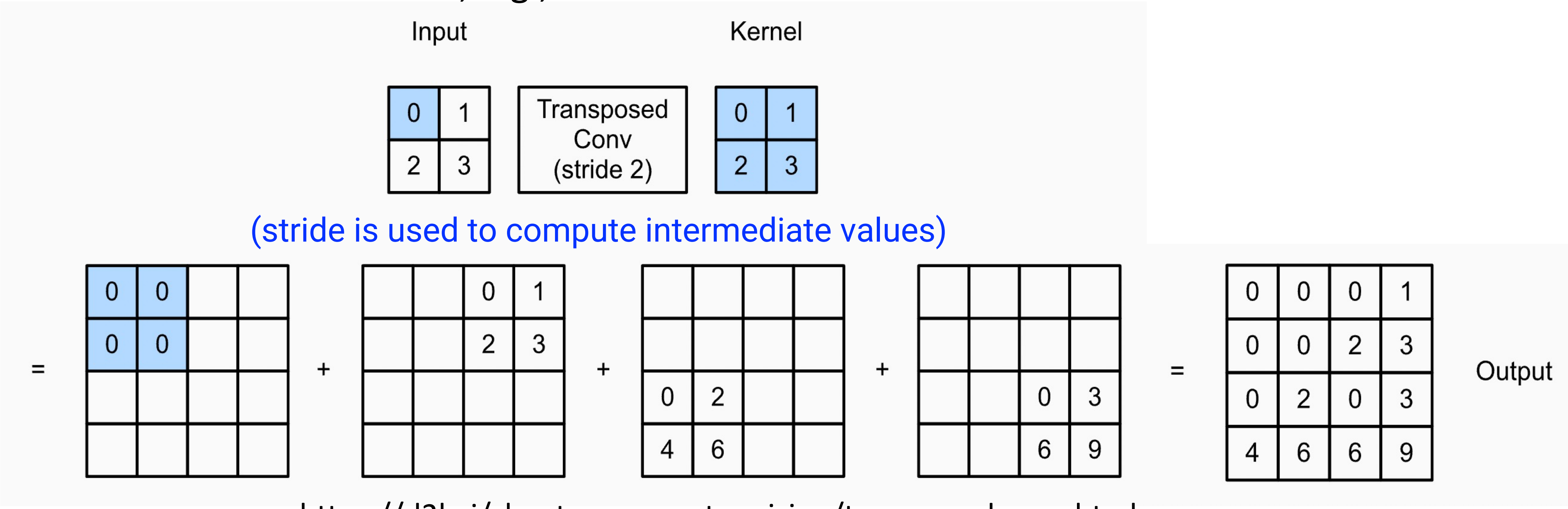
- Also called “fractional convolutional layer”, “backward convolution”, and, incorrectly, ”deconvolution layer”
- Idea: learn filters with a fractional sized stride to upsample the coarse image while refining it based on the filter values; e.g.,



https://d2l.ai/chapter_computer-vision/transposed-conv.html

Architecture: Upsampling (Transposed Convolutional Layer)

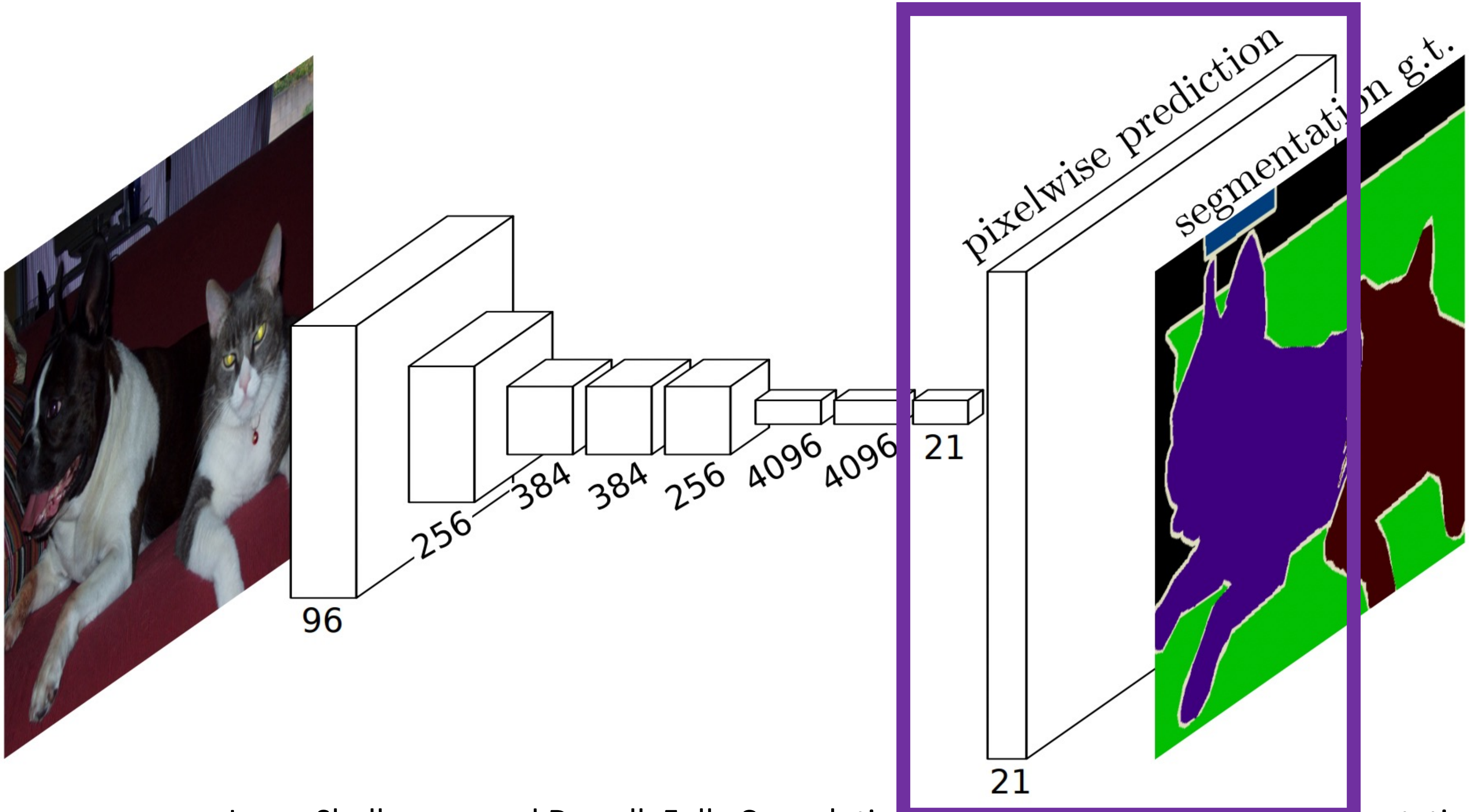
- Also called “fractional convolutional layer”, “backward convolution”, and, incorrectly, ”deconvolution layer”
- Idea: learn filters with a fractional sized stride to upsample the coarse image while refining it based on the filter values; e.g.,



https://d2l.ai/chapter_computer-vision/transposed-conv.html

Architecture

Next challenge: how to decode a **highly detailed** per pixel classification from the coarse region classifications?



Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture: Results

Next challenge: how to decode a **highly detailed** per pixel classification from the coarse region classifications?

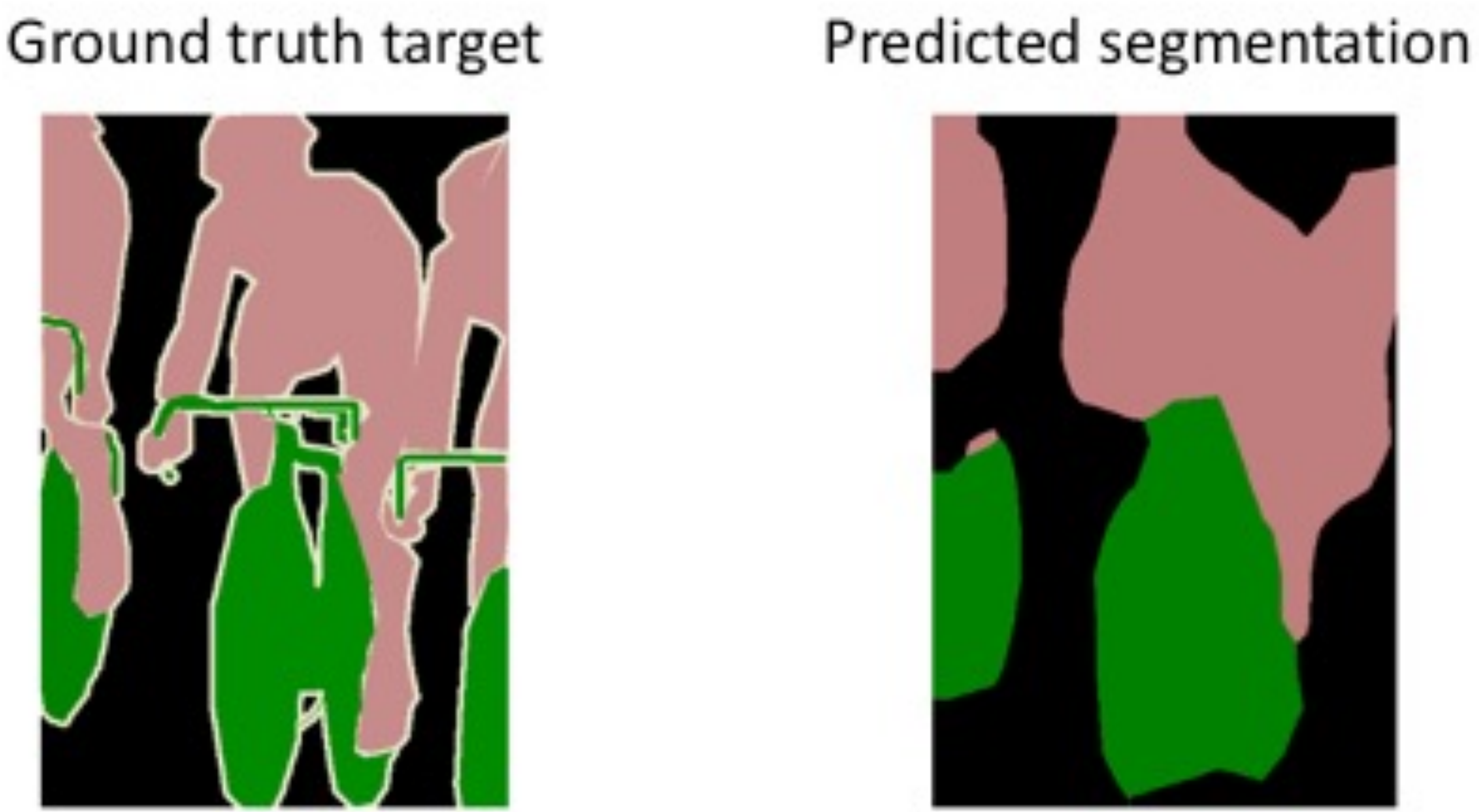
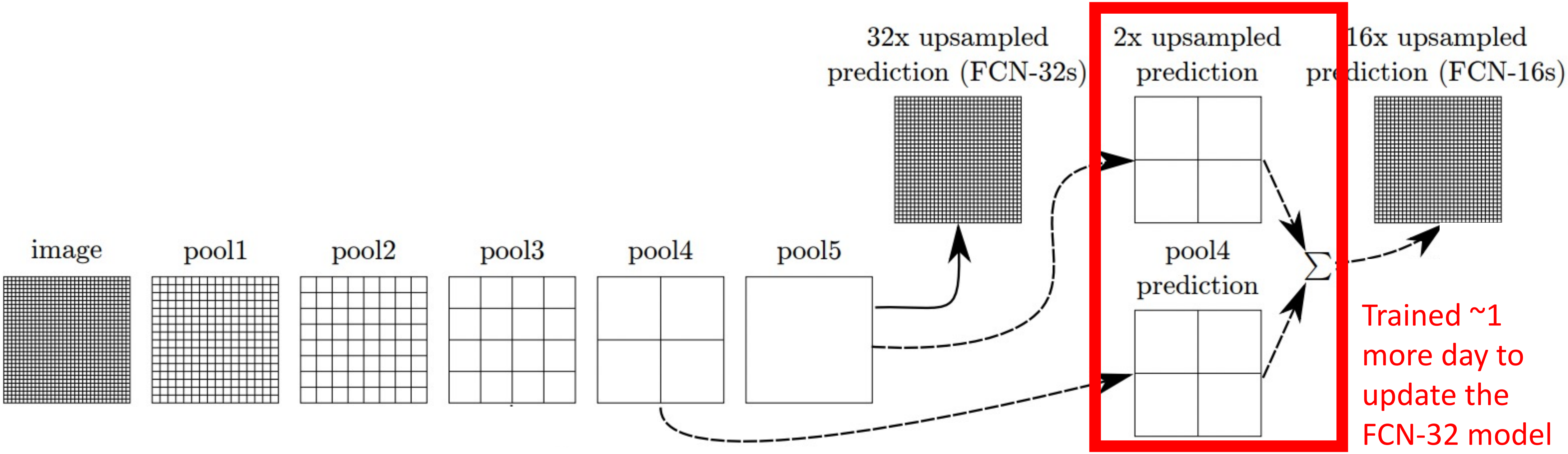


Figure source: https://www.jeremyjordan.me/semantic-segmentation/

Architecture: Update to Use Skip Connections

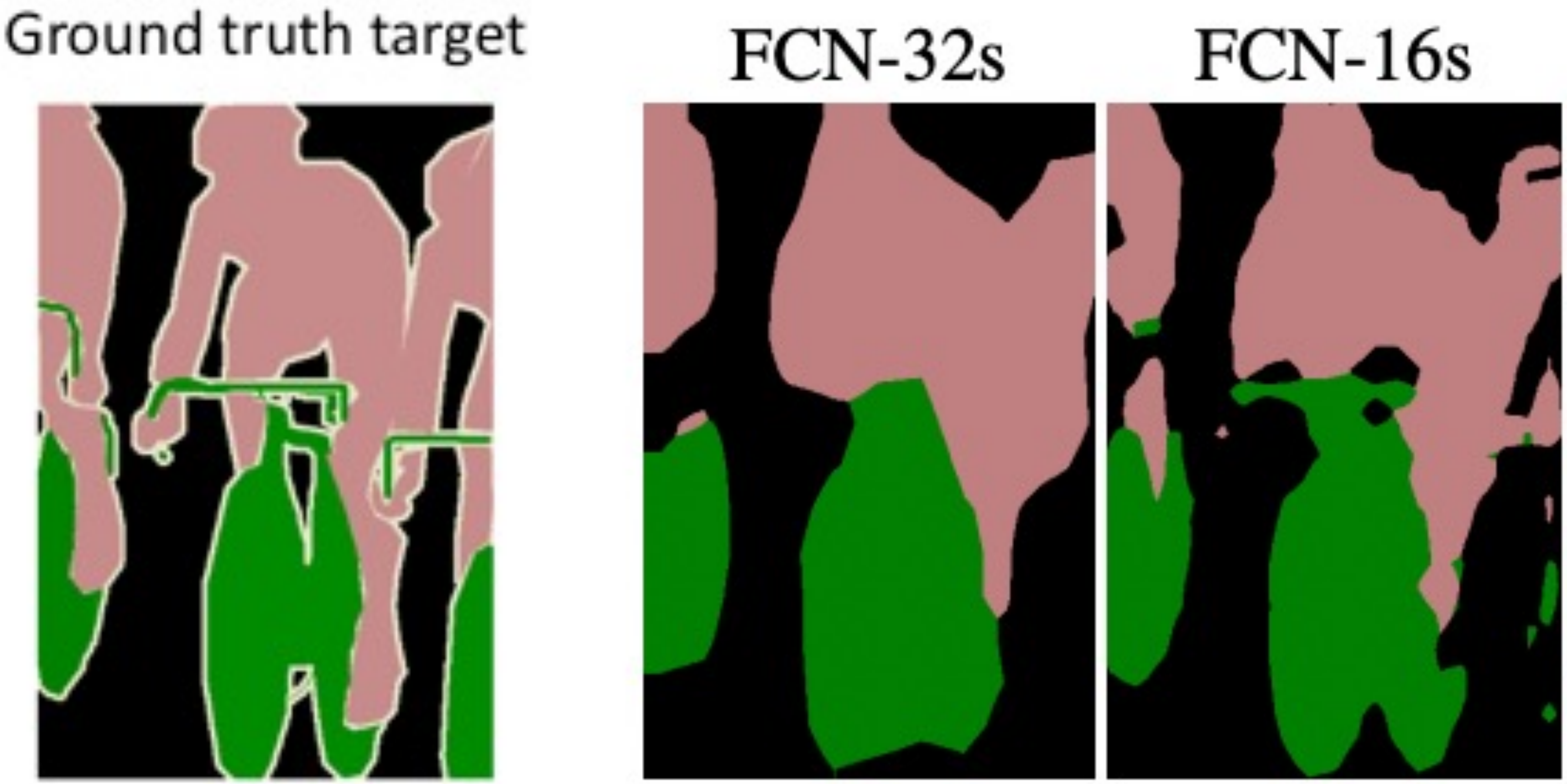


Trained ~1 more day to update the FCN-32 model

FCN16: Sums predictions of lower-level, more fine-grained features (pool4) with the predictions at the coarser features

Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture: Results

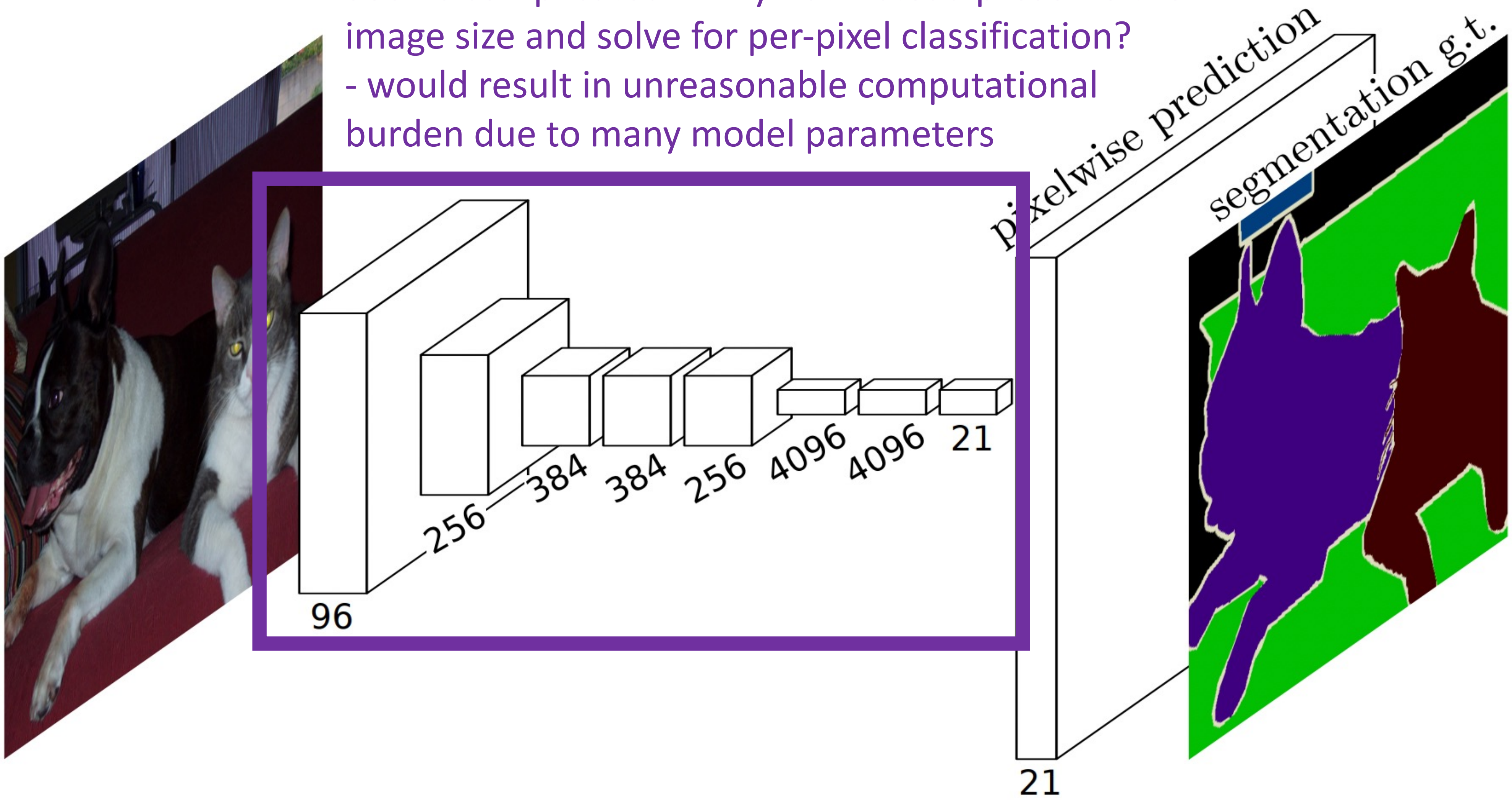


Skip connections support capturing finer-grained details while retaining the correct semantic information!

Figure source: https://www.jeremyjordan.me/semantic-segmentation/

Architecture: Upsampling + Skip Connections

Seems complicated… why not instead preserve the image size and solve for per-pixel classification?
- would result in unreasonable computational burden due to many model parameters

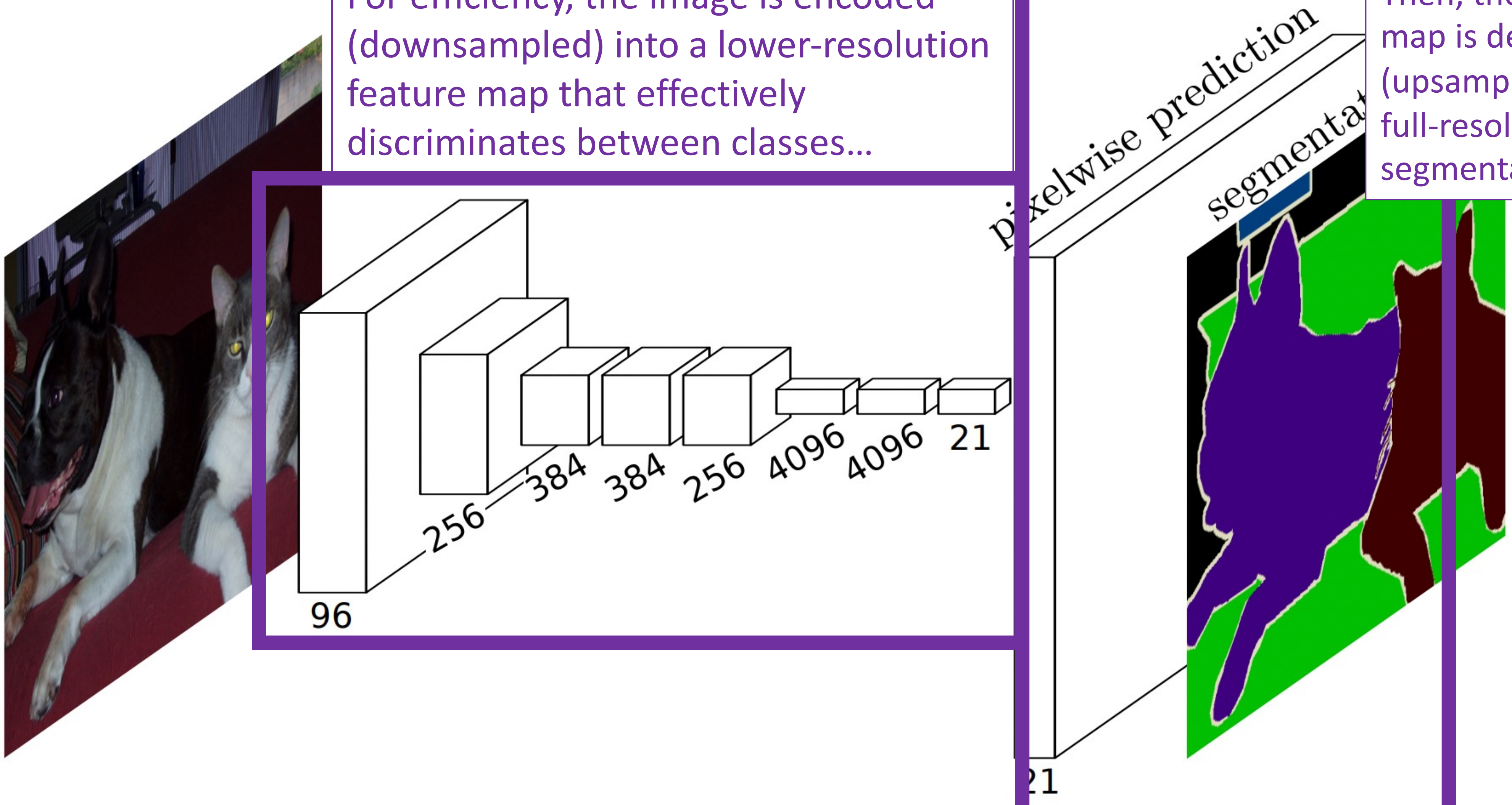


Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Architecture: Encoder Decoder Architecture

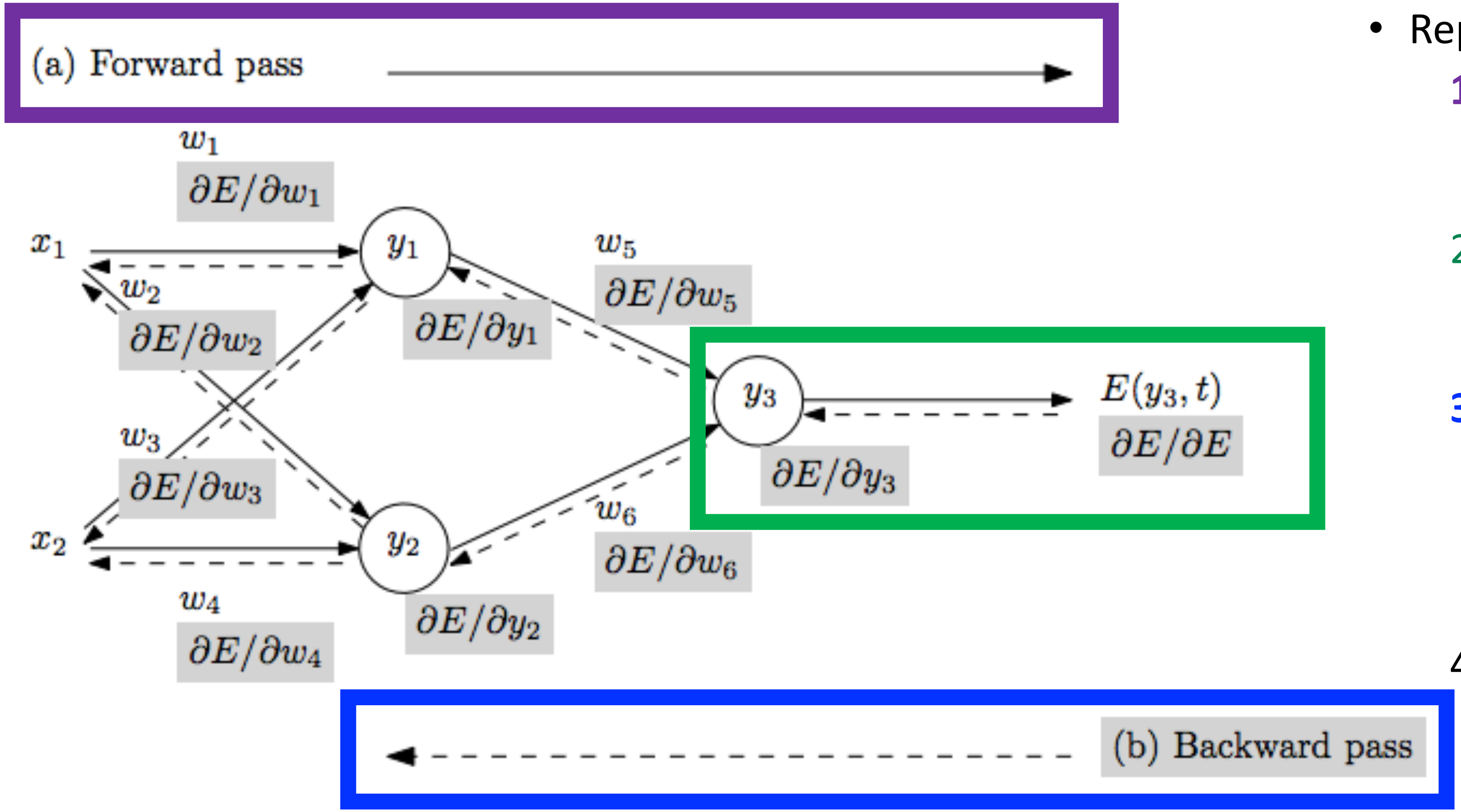
For efficiency, the image is encoded (downsampled) into a lower-resolution feature map that effectively discriminates between classes…

Then, the feature map is decoded (upsampled) into a full-resolution segmentation map.



Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Training: Took 3 days on 1 GPU



- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make prediction
  2. Quantify the dissatisfaction with a model's results on the training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

Figure from: Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, Jeffrey Mark Siskind; Automatic Differentiation in Machine Learning: a Survey; 2018

Training: How Neural Networks Learn

Sum across all pixels the distance between predicted and true distributions using cross entropy loss

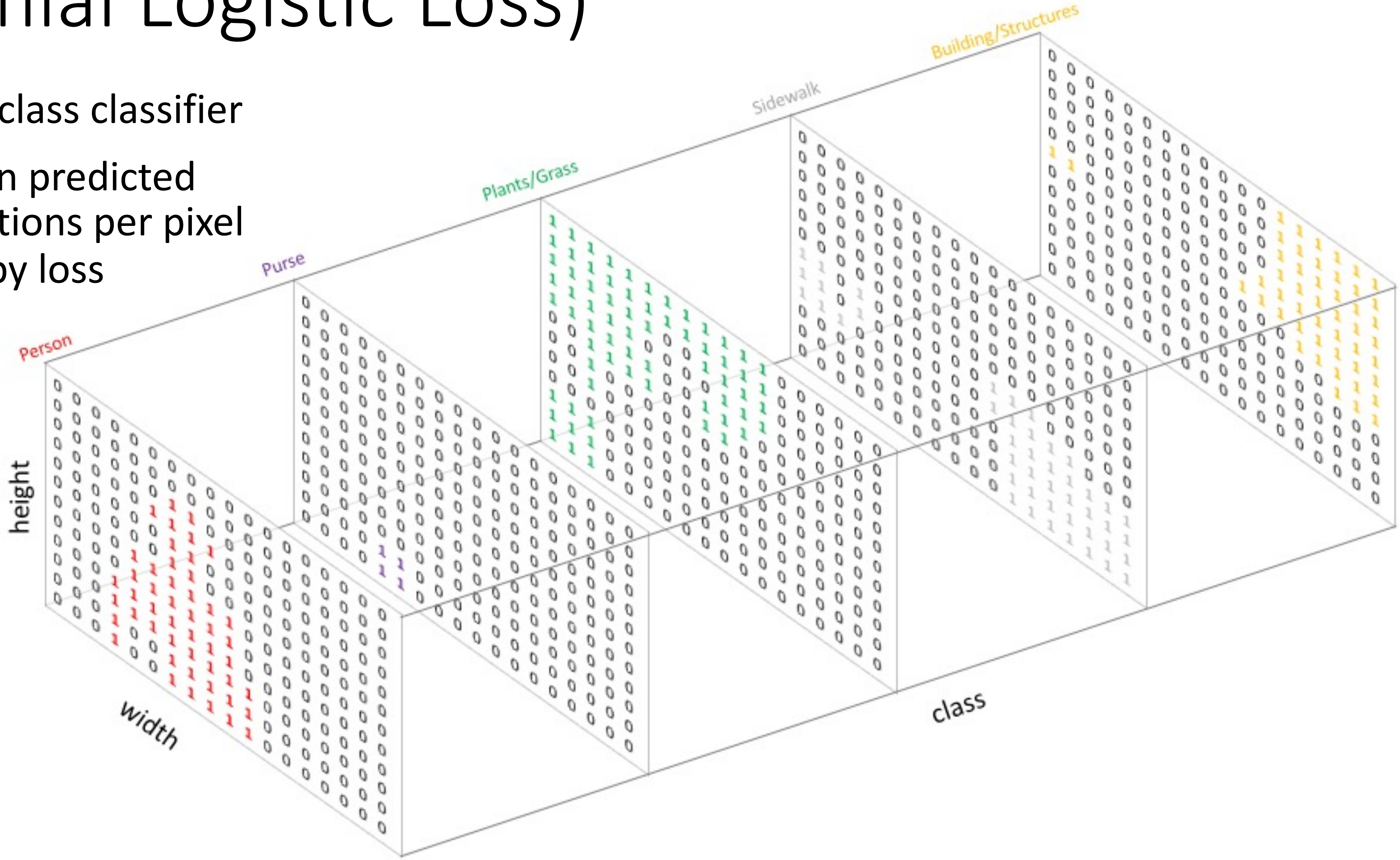
Sum of gradients for all pixels (acts like a minibatch)

- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make prediction
  2. Quantify the dissatisfaction with a model's results on the training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

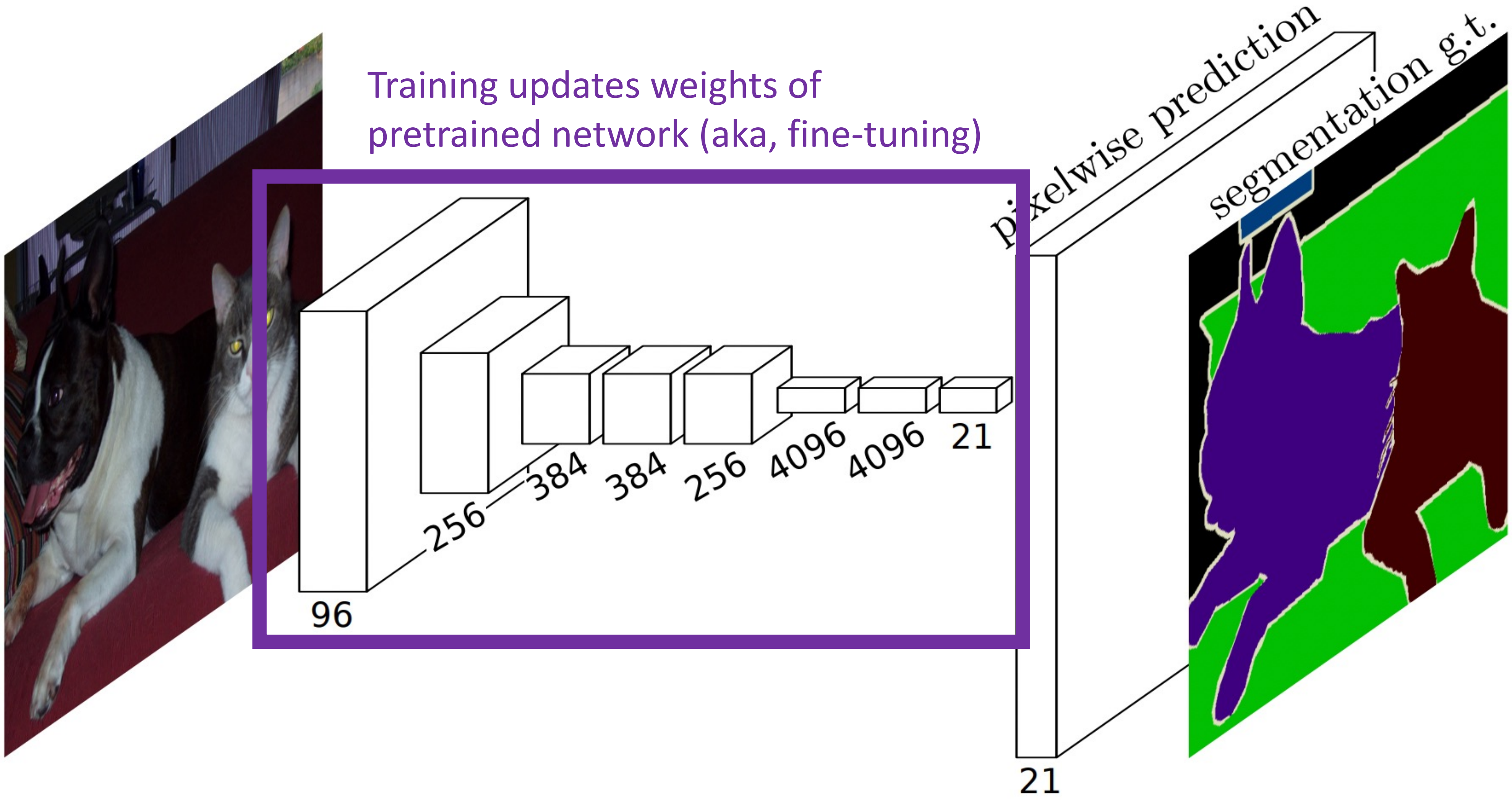
Figure from: Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, Jeffrey Mark Siskind; Automatic Differentiation in Machine Learning: a Survey; 2018

Training: Cross Entropy Loss (Multinomial Logistic Loss)

- e.g., assume a 5-class classifier
- Distance between predicted and true distributions per pixel with cross entropy loss



Architecture: Algorithm Training



Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Results

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [12]	47.9	-	-
SDS [16]	52.6	51.6	$\sim$ 50 s
FCN-8s	**62.7**	**62.2**	**$\sim$ 175 ms**

Compared to existing methods, produces better results at a faster speed!

Long, Shelhamer, and Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015.

Today’s Topics

- Problems
- Applications
- PASCAL VOC detection challenge: R-CNNs
- PASCAL VOC semantic segmentation challenge: fully convolutional networks


The End