Object Detection

Danna Gurari
University of Colorado Boulder
Fall 2021

Review

- Last lecture:
  - Object Detection Problem
  - Object Detection Applications
  - Object Detection Datasets
  - Object Detection Evaluation Metric
- Assignments (Canvas)
  - Reading assignment due earlier today
  - Two new reading assignments due next week
- Questions?

Object Detection: Today’s Topics

- Overview of object detection algorithms
- Baseline Model: R-CNN
- Fast R-CNN
- Faster R-CNN
- YOLO
- Discussion

Object Detection: Today's Topics

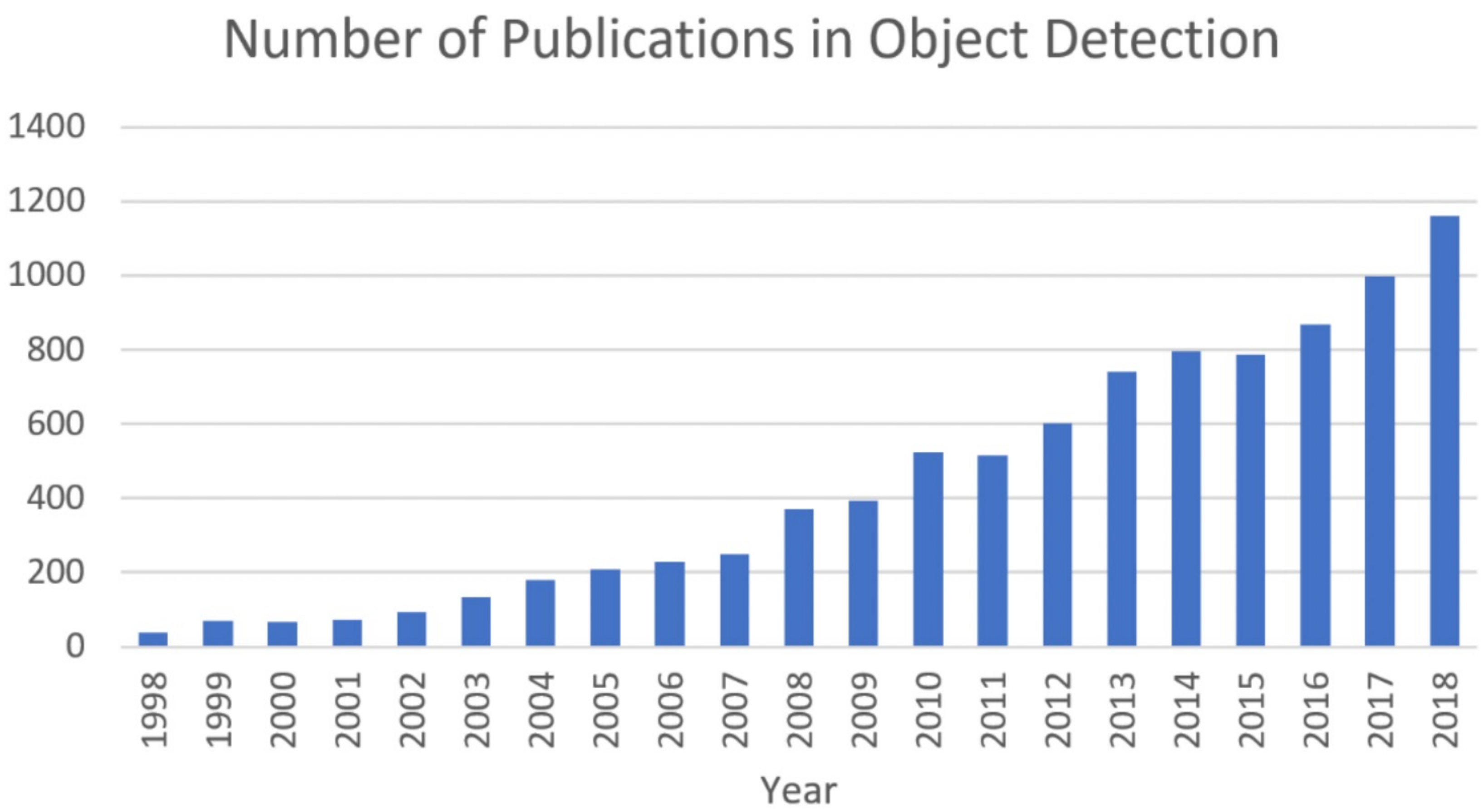
- Overview of object detection algorithms
- Baseline Model: R-CNN
- Fast R-CNN
- Faster R-CNN
- YOLO
- Discussion

Recall Motivation: Go **Faster** While Getting Good Accuracy

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

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

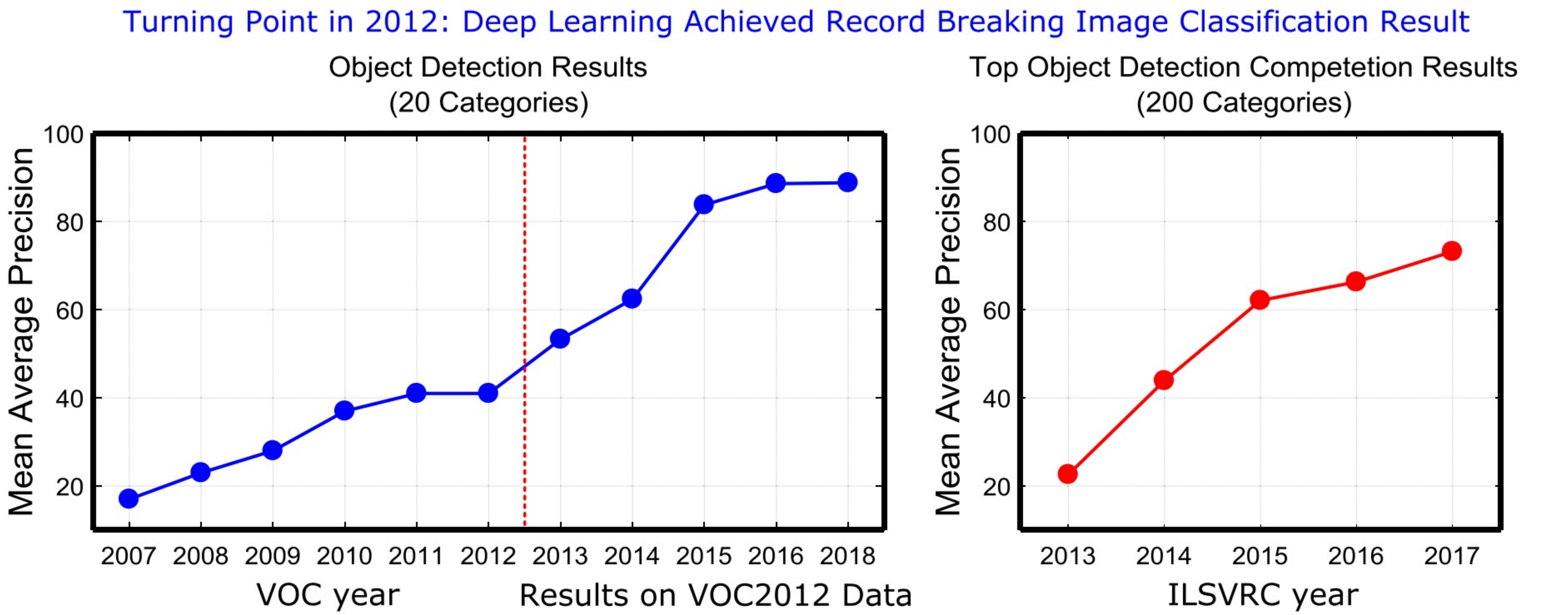
Community Research Engagement



"Data from Google scholar advanced search: allintitle: 'object detection' AND 'detecting objects'"

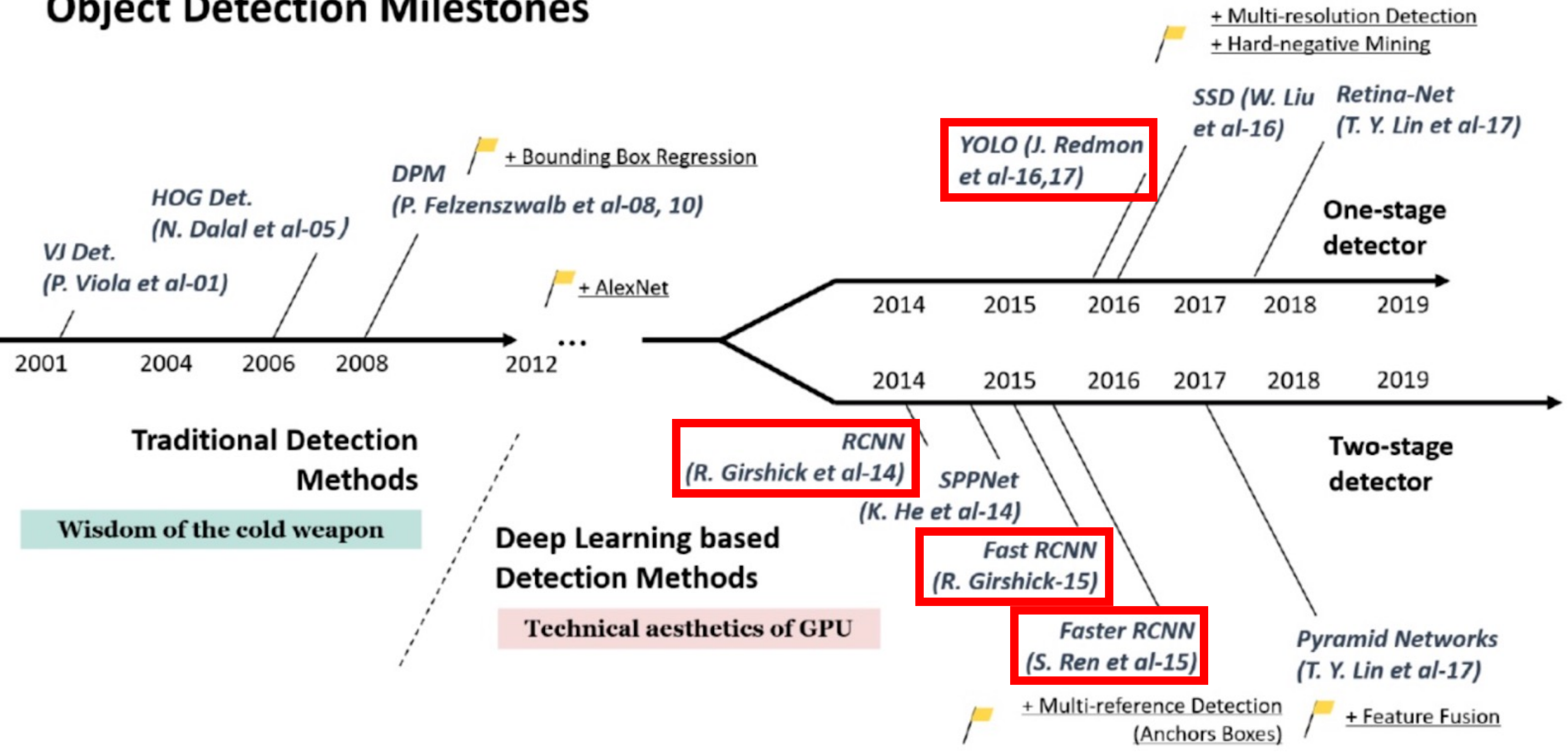
Zhou et al. Object Detection in 20 Years: A Survey. arXiv 2019.

Performance Trends on Two Datasets



Li Liu et al. “Deep Learning for Generic Object Detection: A Survey.” IJCV 2019

Object Detection Milestones

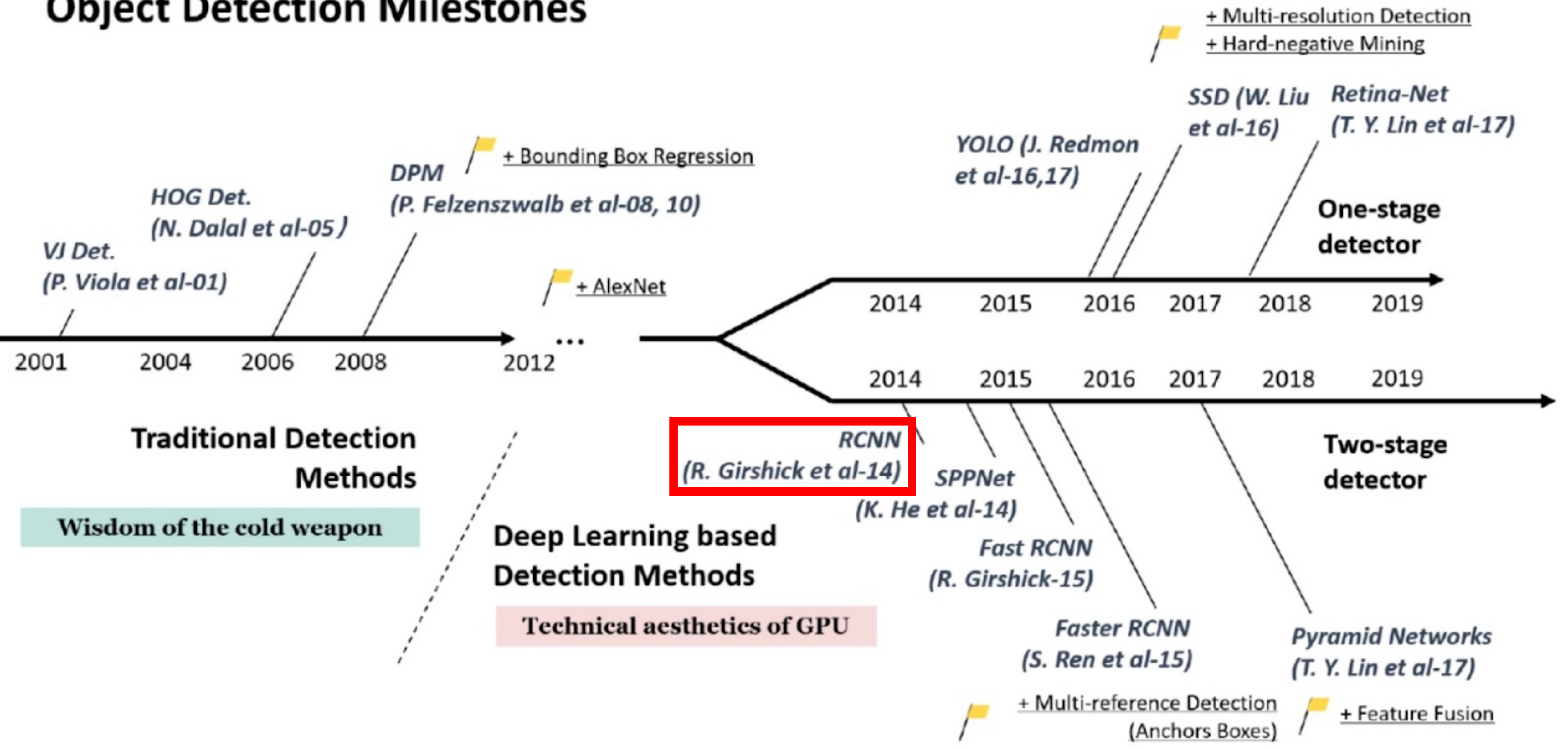


Zhou et al. Object Detection in 20 Years: A Survey. arXiv 2019.

Object Detection: Today's Topics

- Overview of object detection algorithms
- Baseline Model: R-CNN
- Fast R-CNN
- Faster R-CNN
- YOLO
- Discussion

Object Detection Milestones



Zhou et al. Object Detection in 20 Years: A Survey. arXiv 2019.

Why R-CNN?

Named after the proposed technique: use **R**egion proposals with **CNN** features

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation.” CVPR 2014.

Key Contributions of R-CNN

1. Demonstrate how to accurately localize objects with a neural network (NN)
    - First time a CNN outperformed hand-crafted features on VOC, achieving mAP of 54% compared to 33% for previous HOG based model (VOC 2010)

2. Demonstrate how train an accurate (high-capacity) NN with a scarce amount of annotated detection data

Key Contributions of R-CNN

1. Demonstrate how to accurately localize objects with a neural network (NN)
   - First time a CNN outperformed hand-crafted features on VOC, achieving mAP of 54% compared to 33% for previous HOG based model (VOC 2010)

2. Demonstrate how train an accurate (high-capacity) NN with a scarce amount of annotated detection data

Architecture

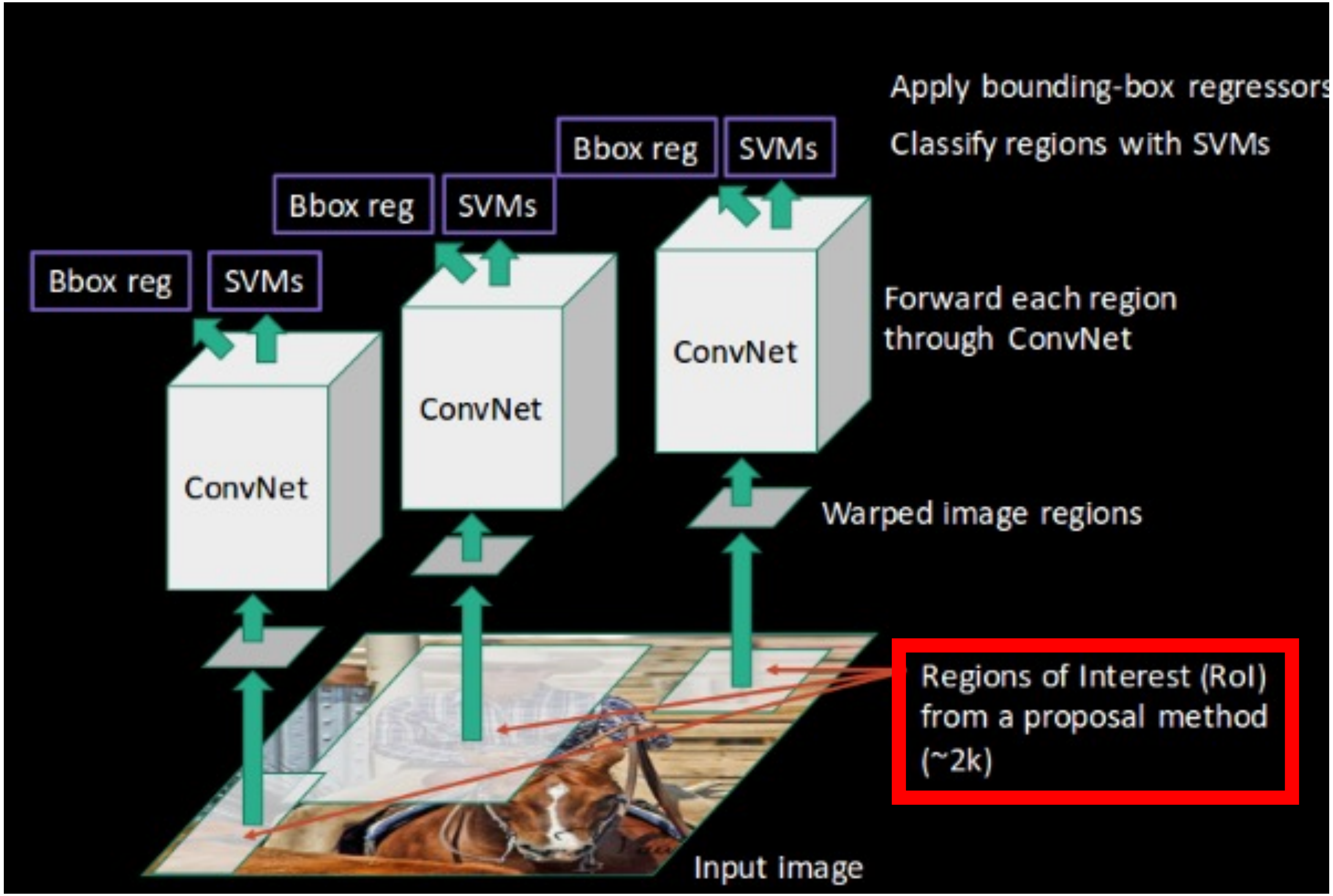
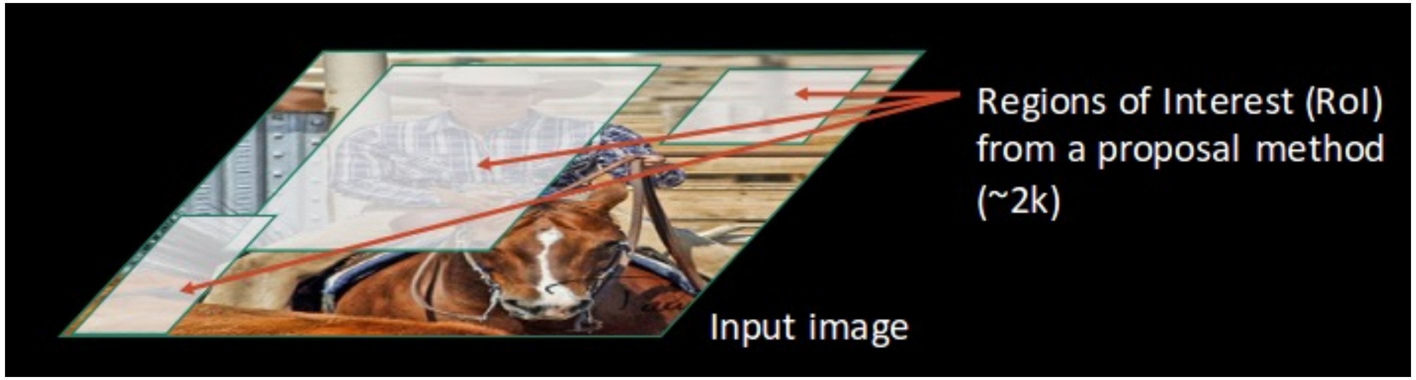


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture: Define Candidate Detection Regions



- Region proposal methods: given an image, produce bounding boxes around "object"-like regions
- Why use these methods?
  - The number of regions will be considerably fewer than needed in a naïve sliding window approach
  - Belief is that they will include regions that contain the objects of interest (i.e., high recall)
- Many options:
  - Objectness
  - Constrained Parametric Min-Cuts for Automatic Object Segmentation (CPMC)
  - Category Independent Object Proposals
  - Randomized Prim
  - Selective Search: used to enable comparison with prior work; creates ~2000 regions based on color, texture, size and shape

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

Architecture

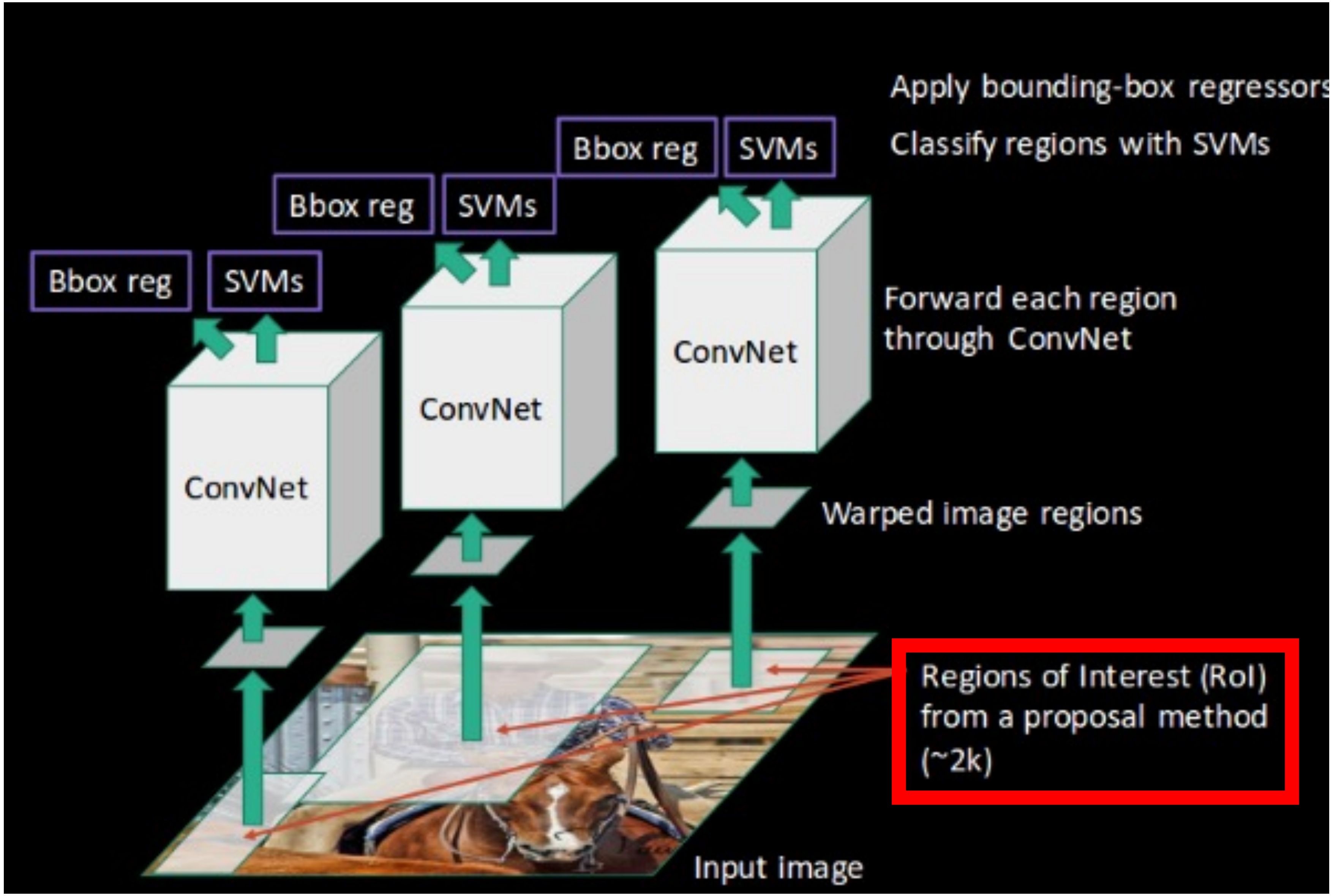


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture

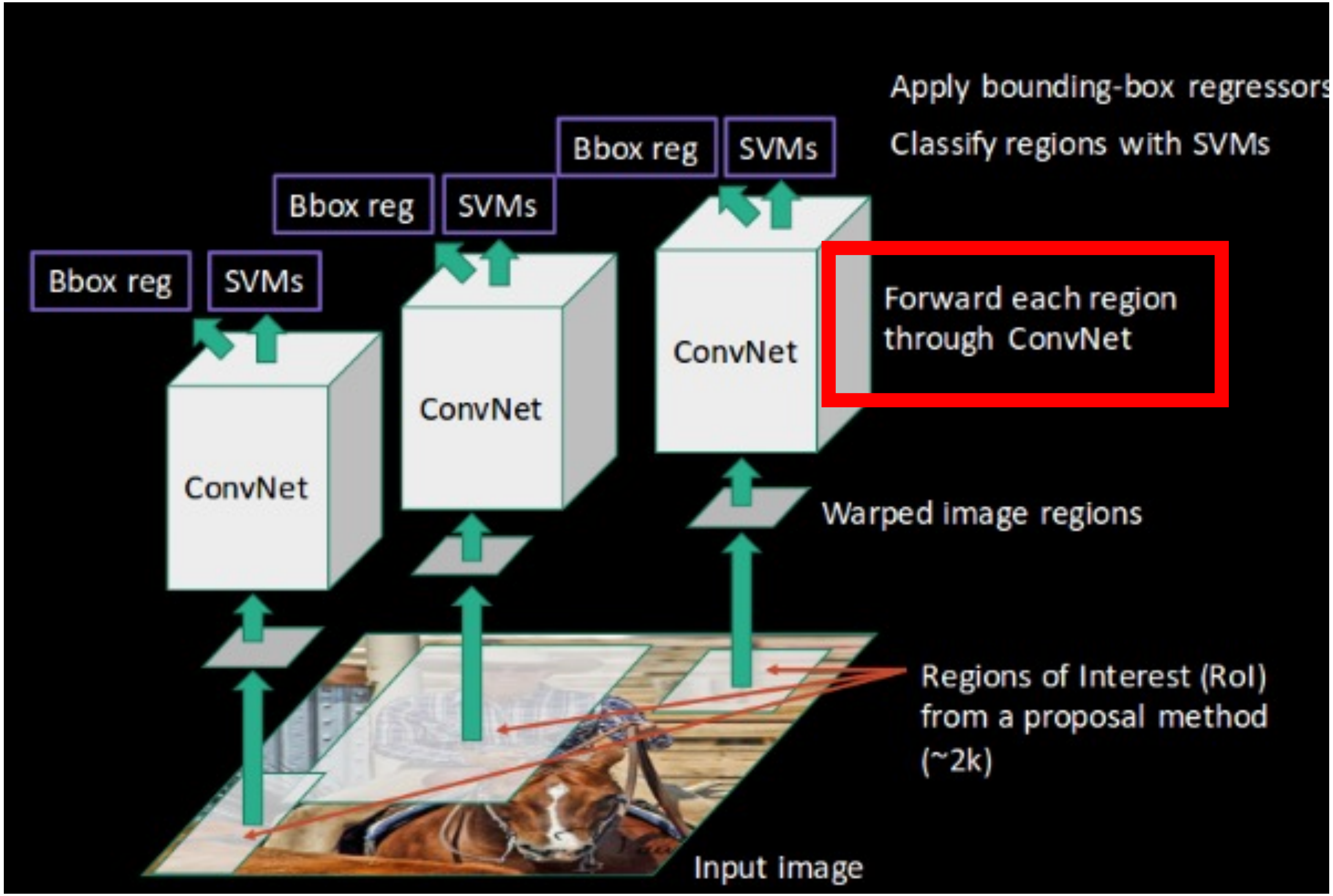


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture: Describe Each Region with a Fixed-length Feature Vector

- Use the FC7 layer from a pretrained AlexNet model

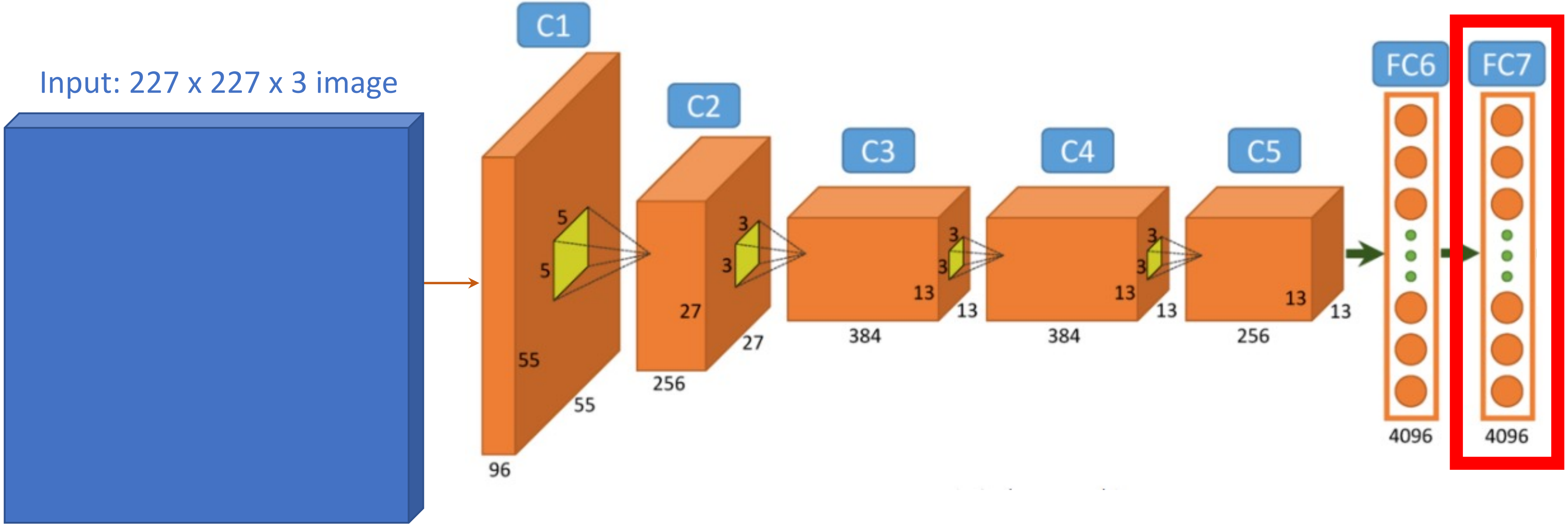


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Architecture: Describe Each Region with a Fixed-length Feature Vector

- Benefit: features can be learned for a dataset instead of handcrafted (e.g., HOG, SIFT)

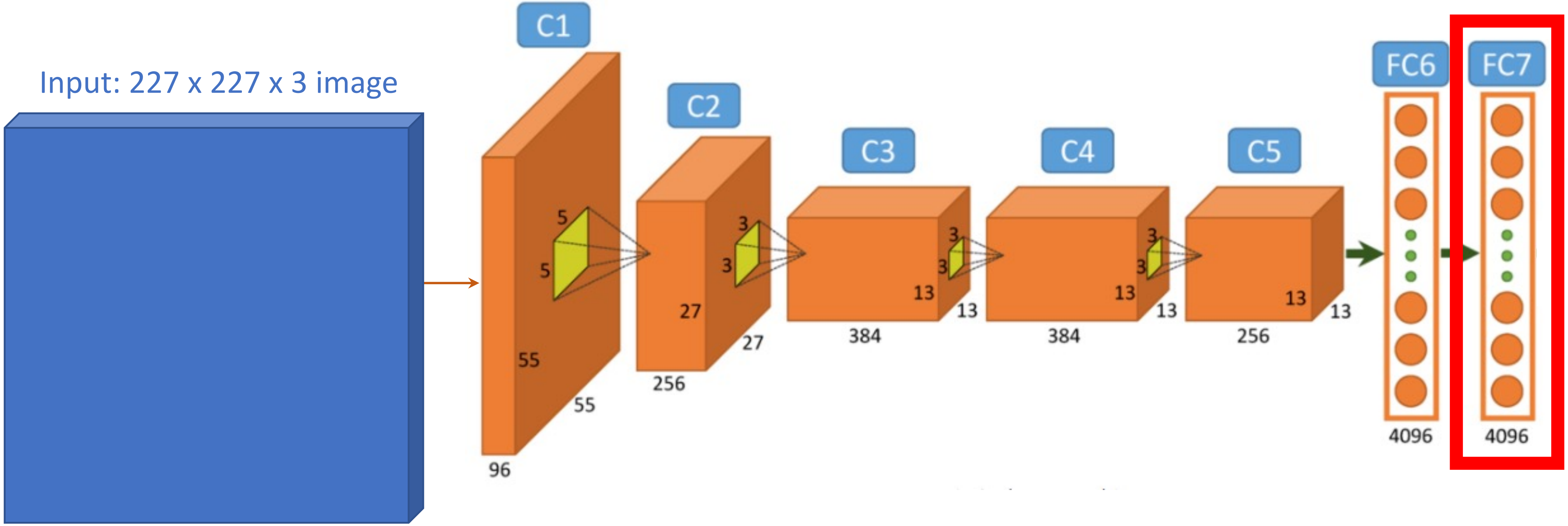


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Architecture: Describe Each Region with a Fixed-length Feature Vector

- Benefit: features are ~2 orders of magnitude smaller than traditional features (e.g., HOG, SIFT)

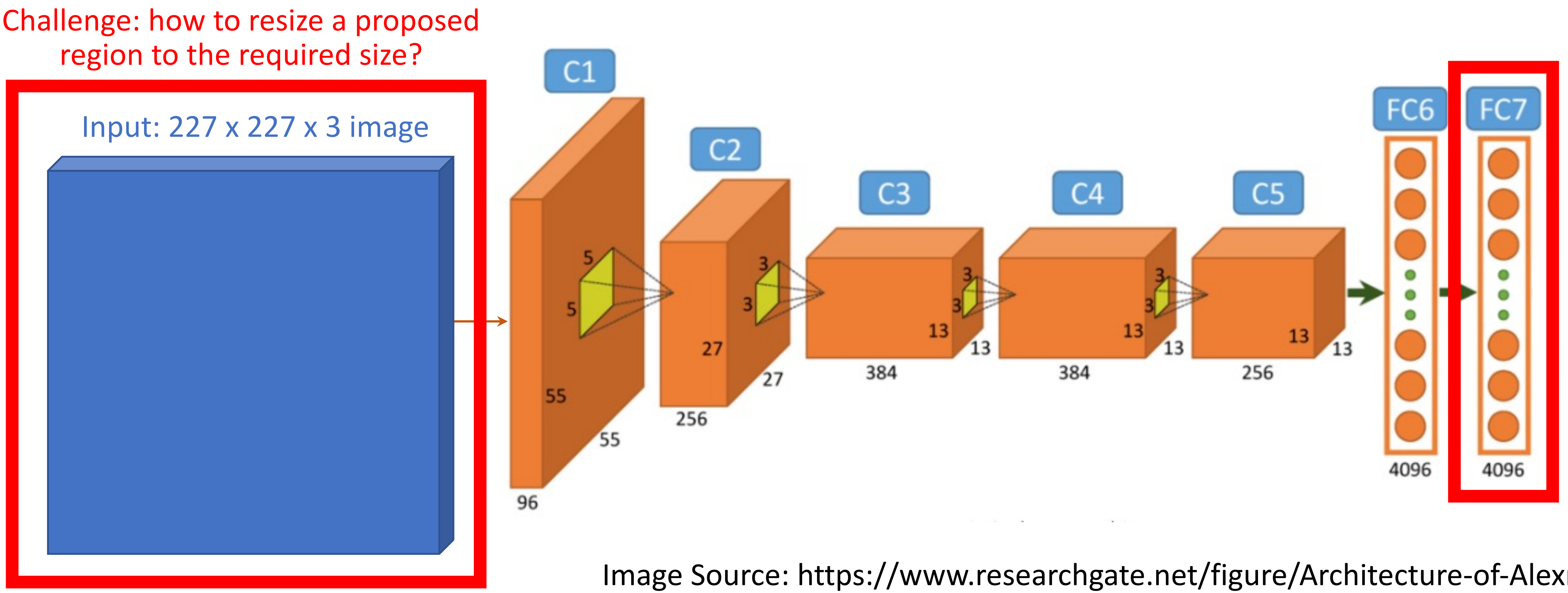


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Architecture: Describe Each Region with a Fixed-length Feature Vector

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

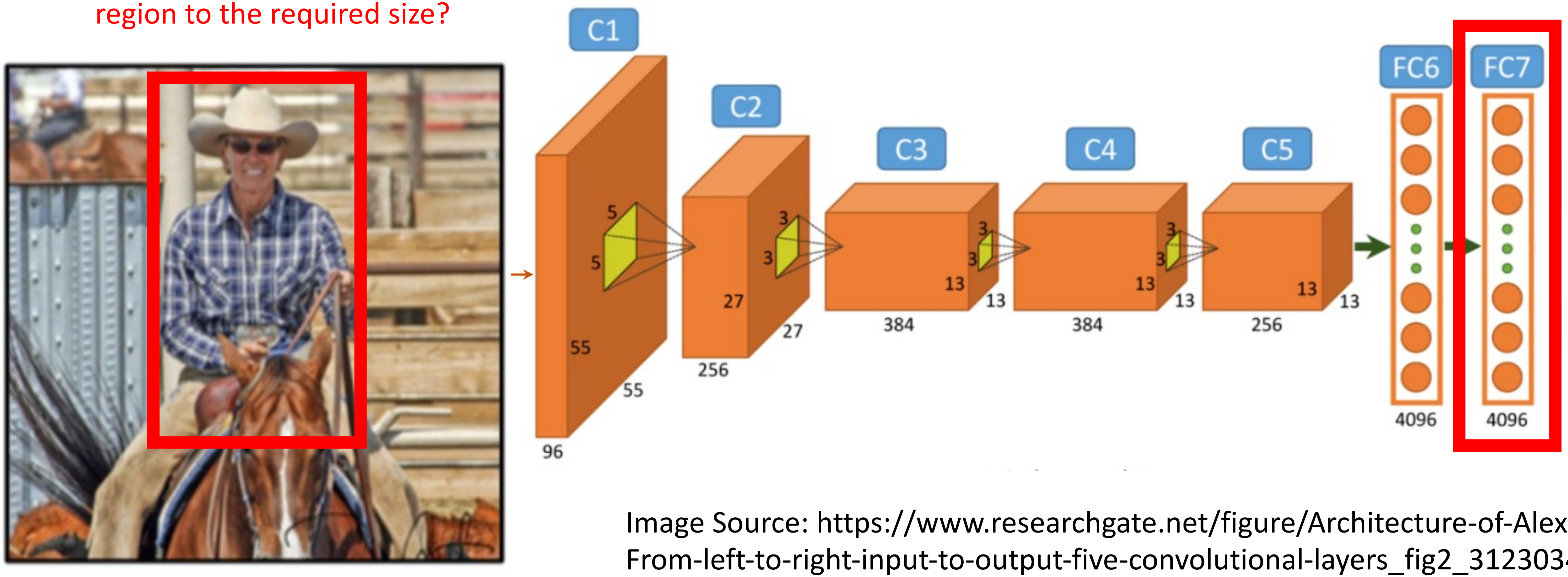


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Architecture: Describe Each Region with a Fixed-length Feature Vector

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

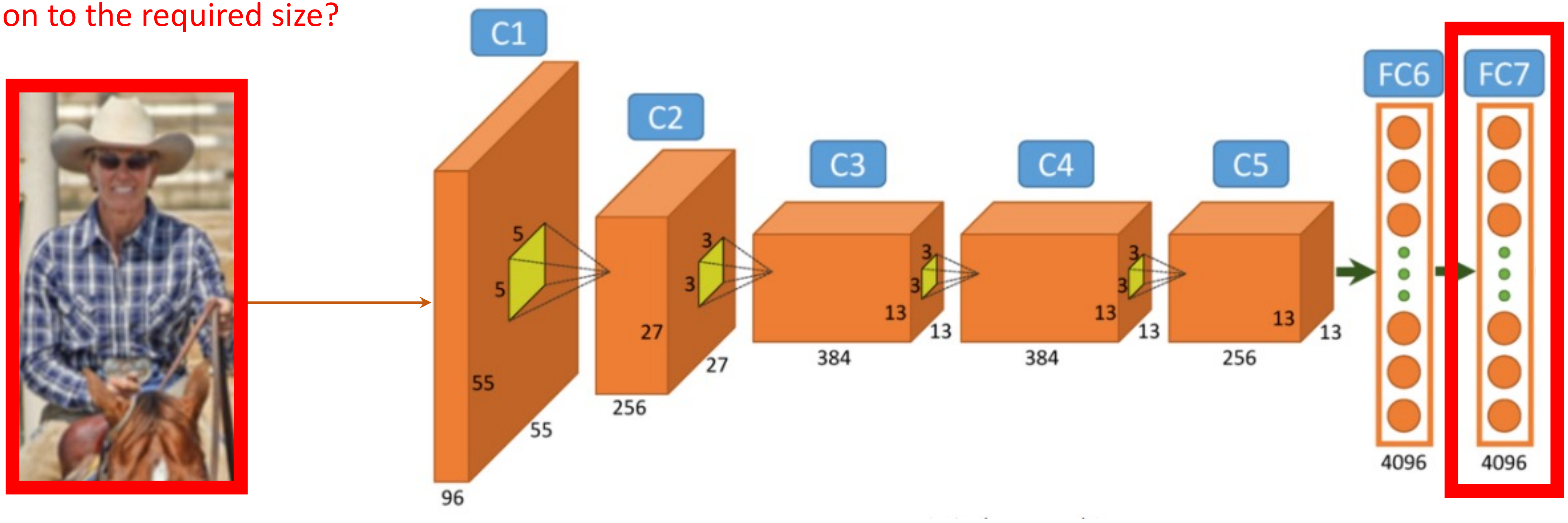


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Architecture: Region Resizing

As exemplified, region proposals come in different sizes and aspect ratios

Architecture: Region Resizing

(1) (2) (3)

Scale region and then pad with original image context or constant

Anisotropic scaling

Chosen because experimentally shown to lead to the best results

Many ways to convert a region into a fixed input size of 227 x 227 x 3

Architecture: Describe Each Region with a Fixed-length Feature Vector

Input: 227 x 227 x 3 image

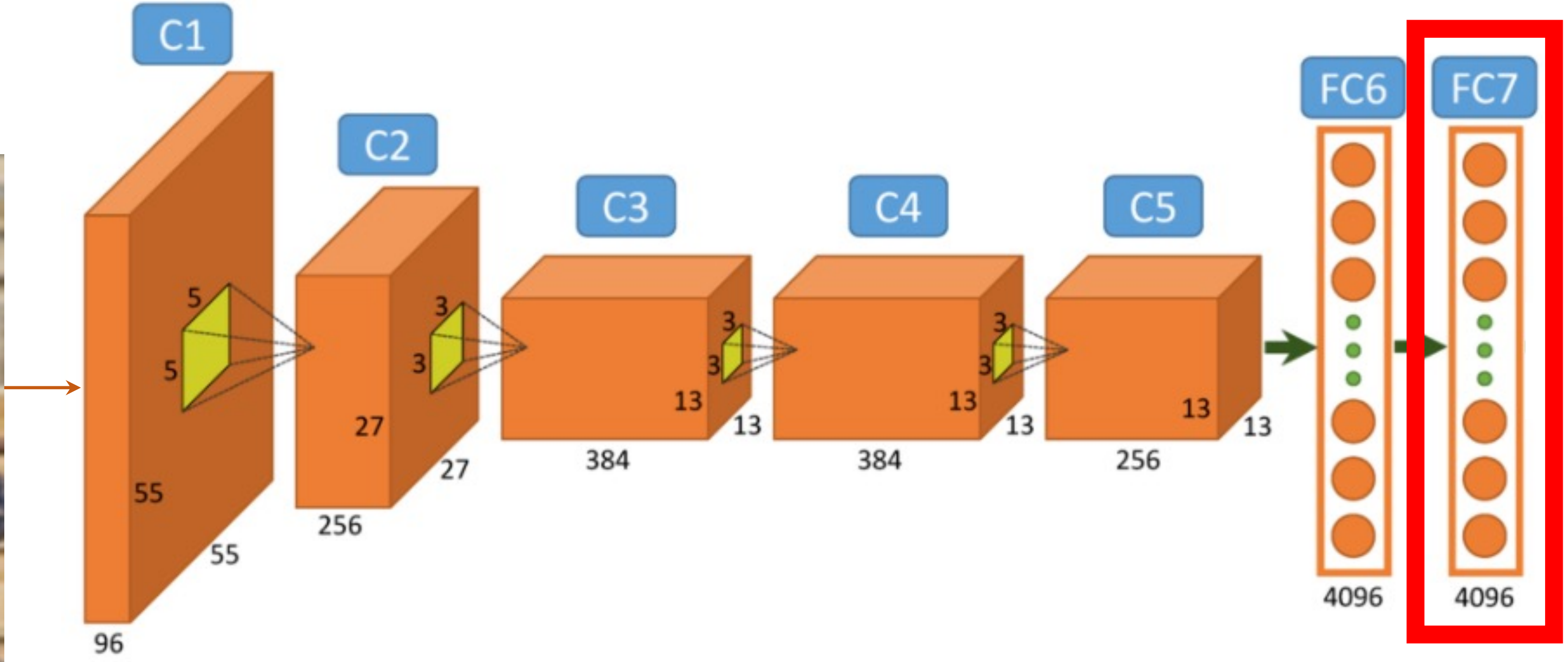


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Architecture

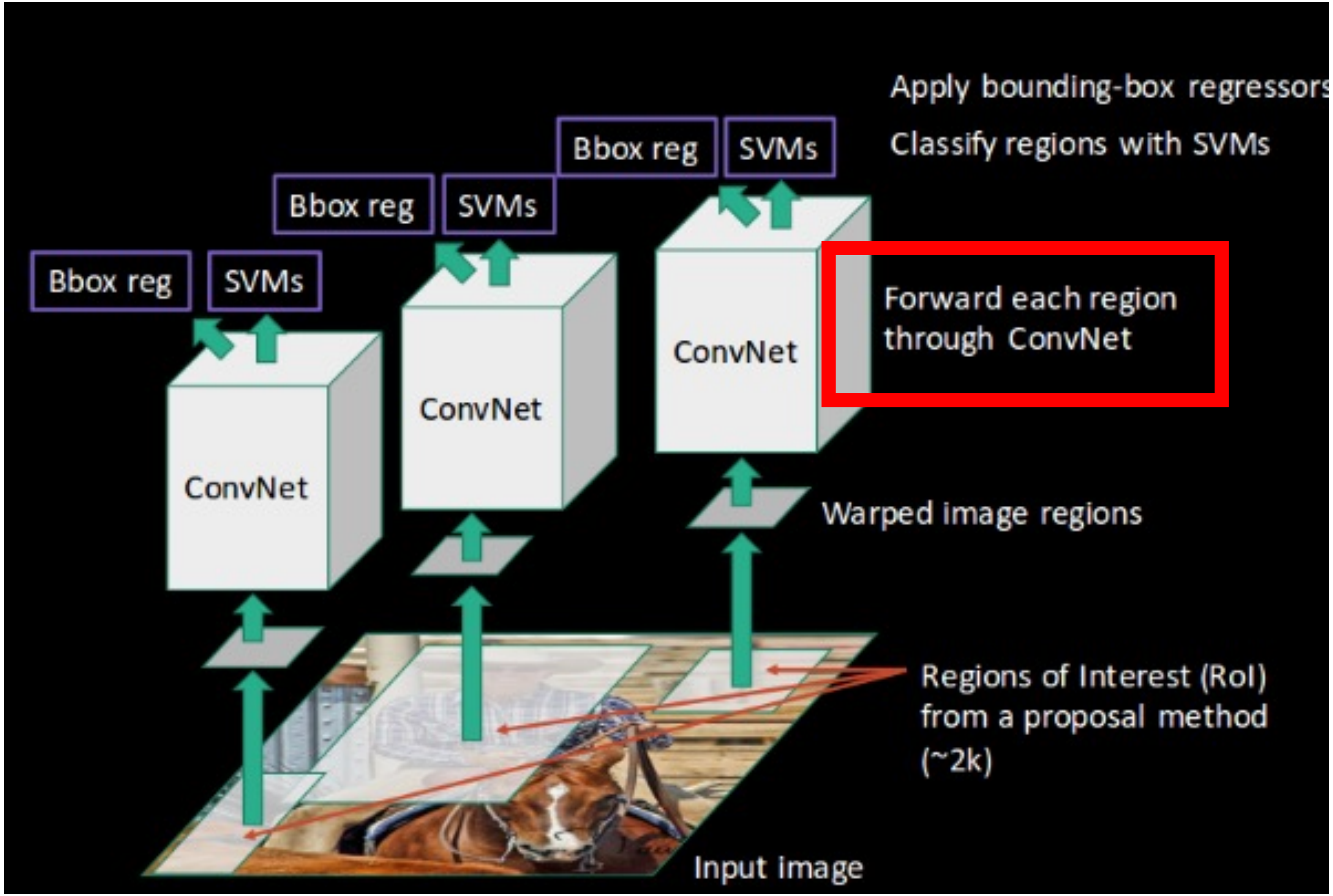


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture

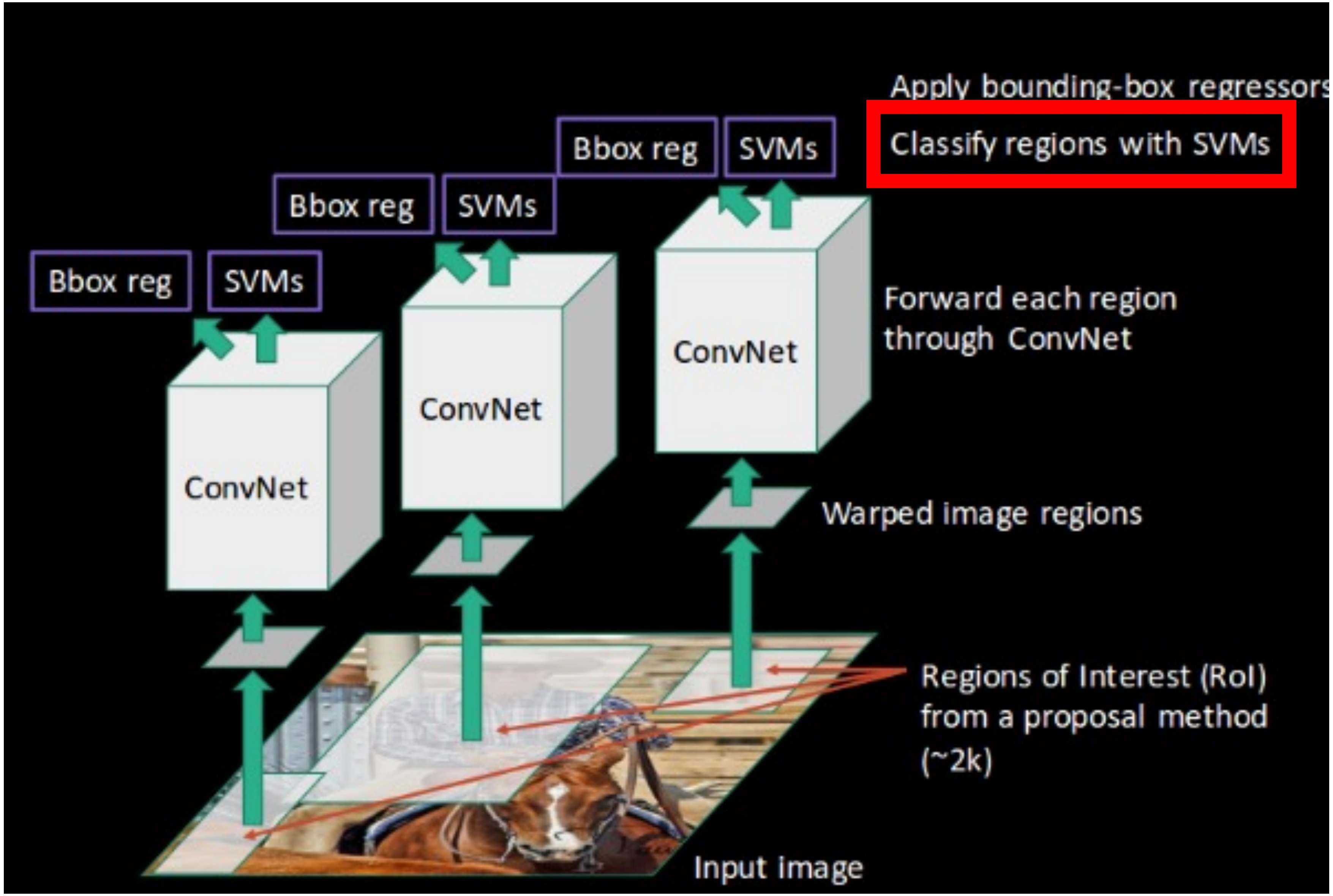
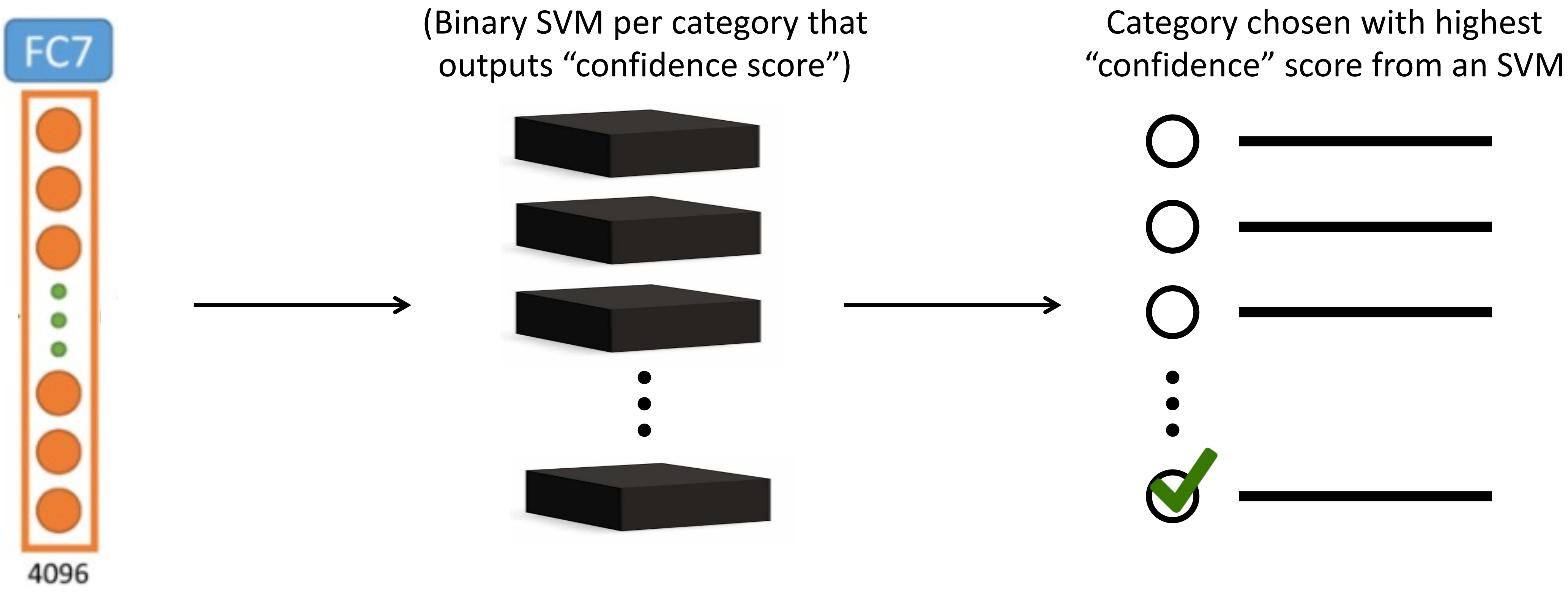


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture: Region Classification

- Assign each feature descriptor that characterizes a region a label from a pre-defined set of categories (i.e., multiple choice)



Architecture

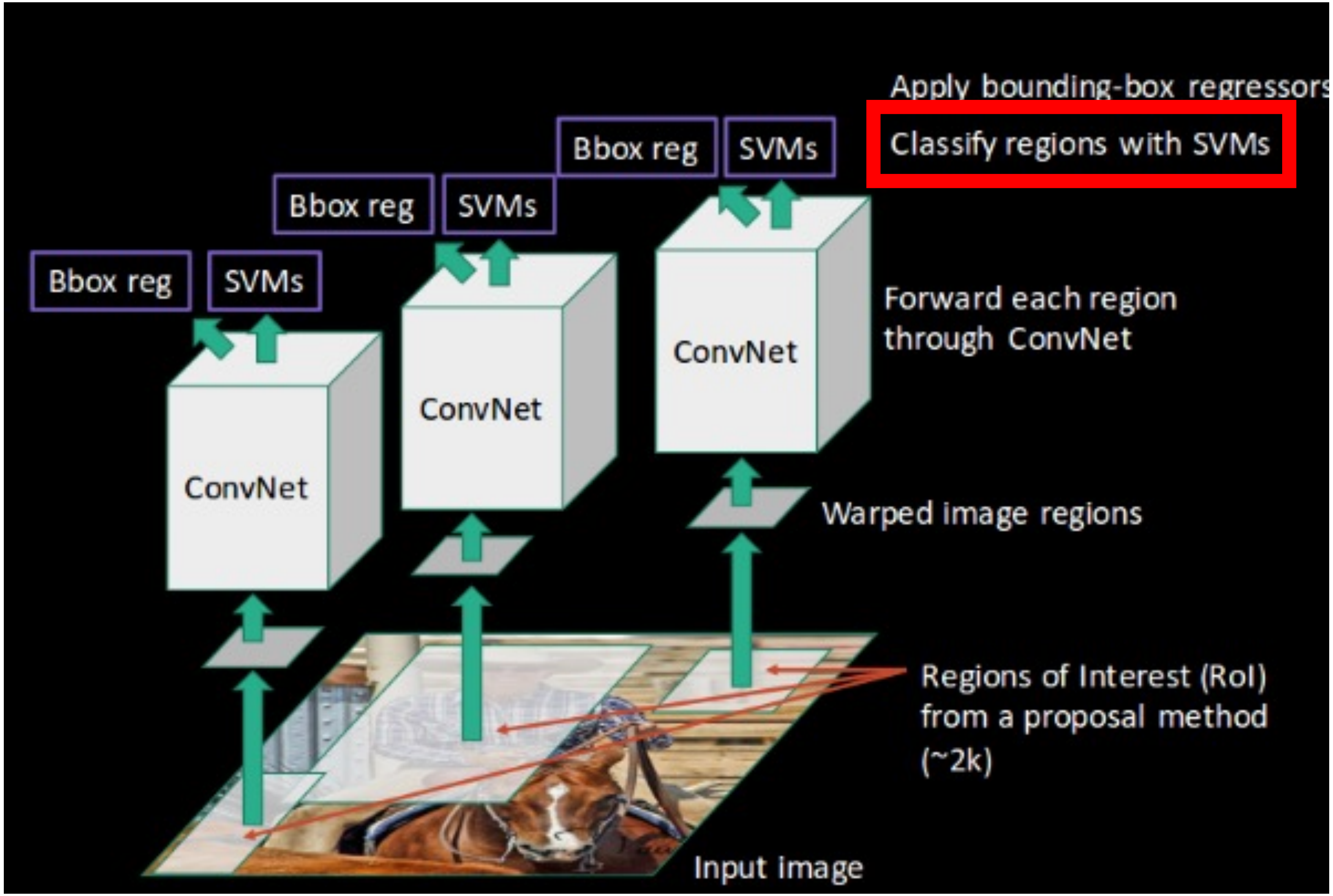


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture

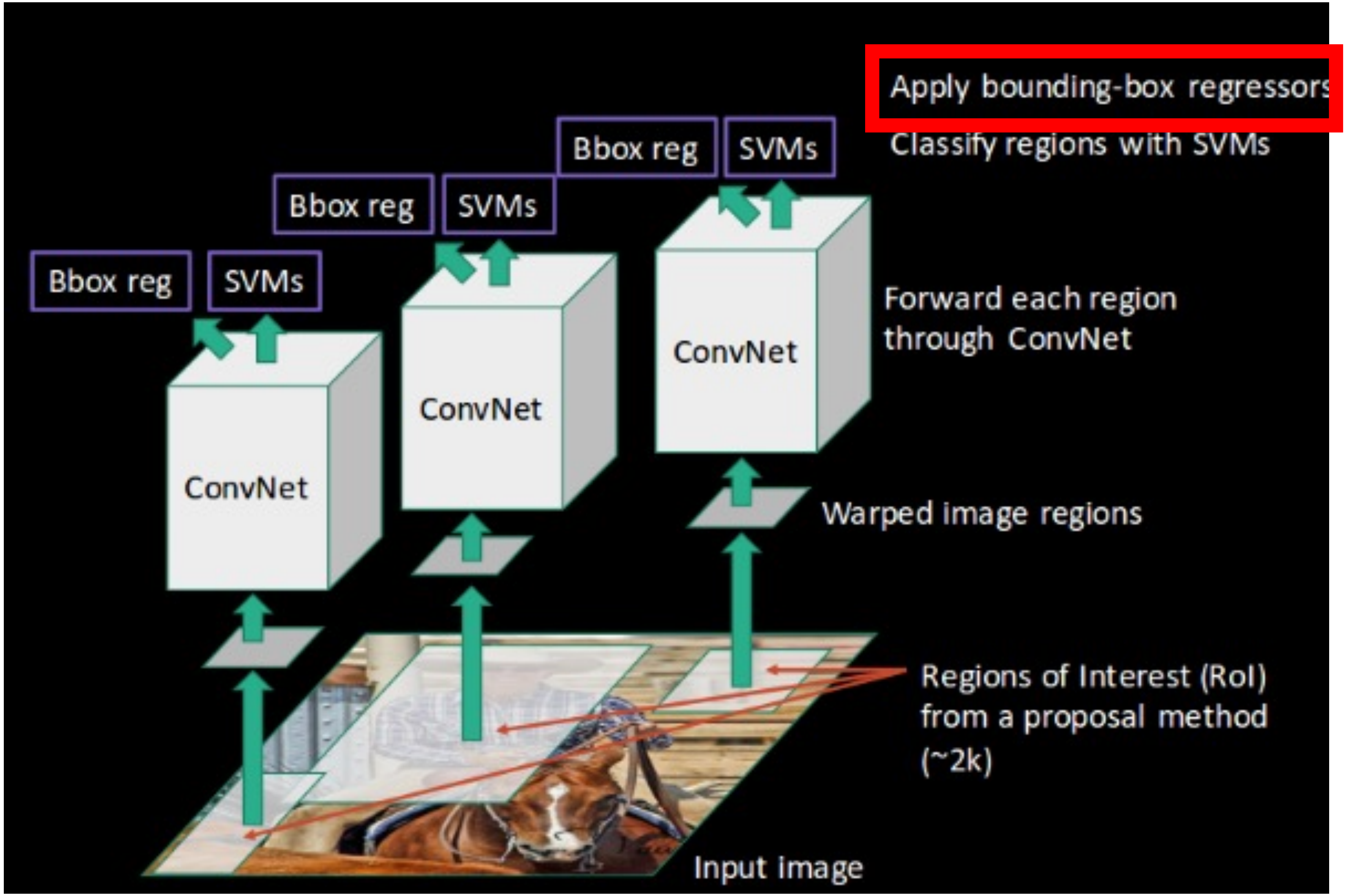


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture: **Region Selection** and Refinement

- Problem: ~2000 regions per image
- Solution: remove redundant regions through **non-maximum suppression;** for each class:
    1. Pick region with maximum score obtained from the SVM.
    2. Discard all regions belonging to that class with IoU score > 70%
    3. Select next highest score region and then repeat steps 1 and 2
    4. Repeat step 3 until all regions are either discarded or kept

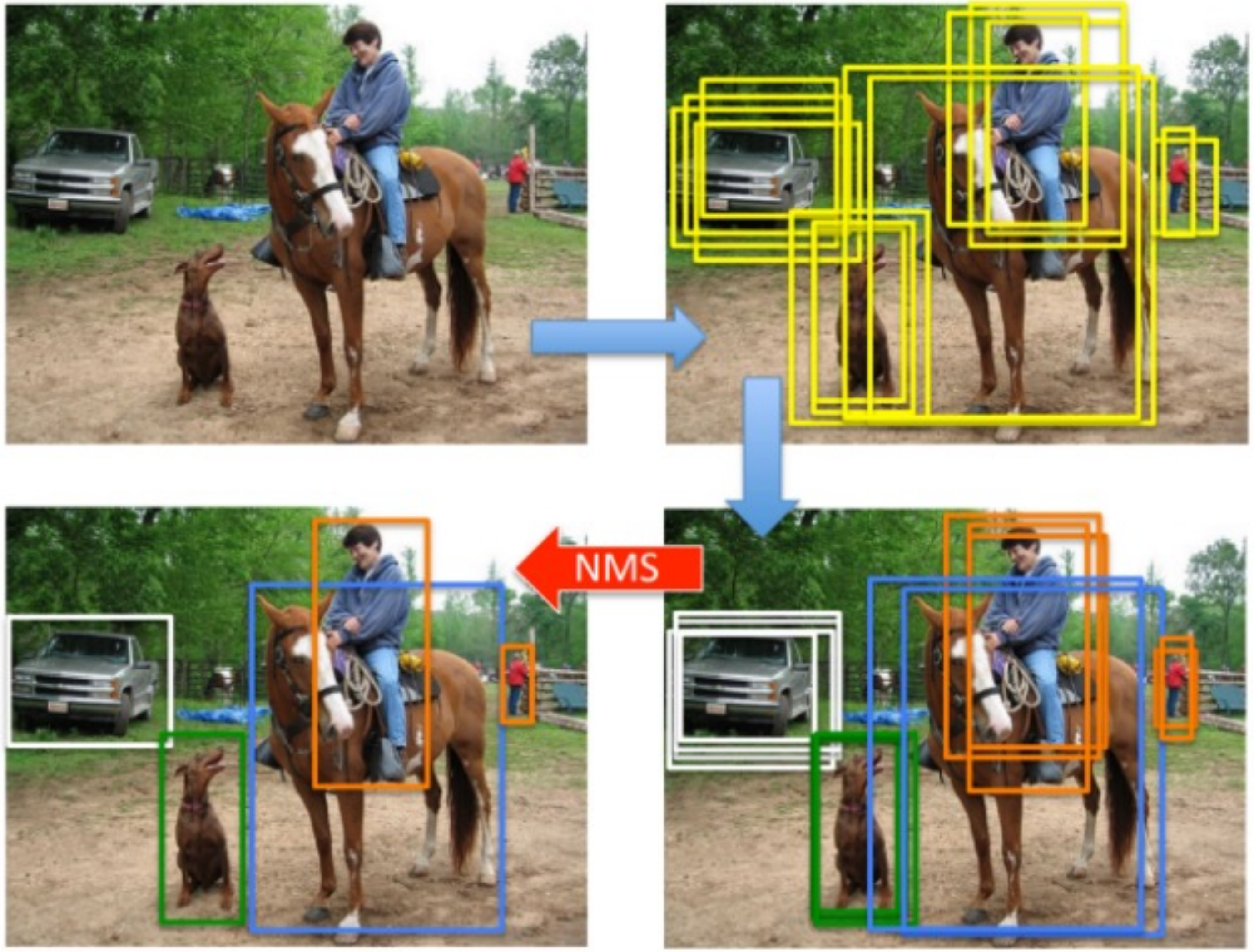


Image Source: https://towardsdatascience.com/deep-learning-method-for-object-detection-r-cnn-explained-ecdadd751d22

Architecture: Region Selection and **Refinement**

- Given an observed dominant issue that region mislocalization is common, each region was then postprocessed/refined by tweaking its position (x, y) and width (w) and height (h)

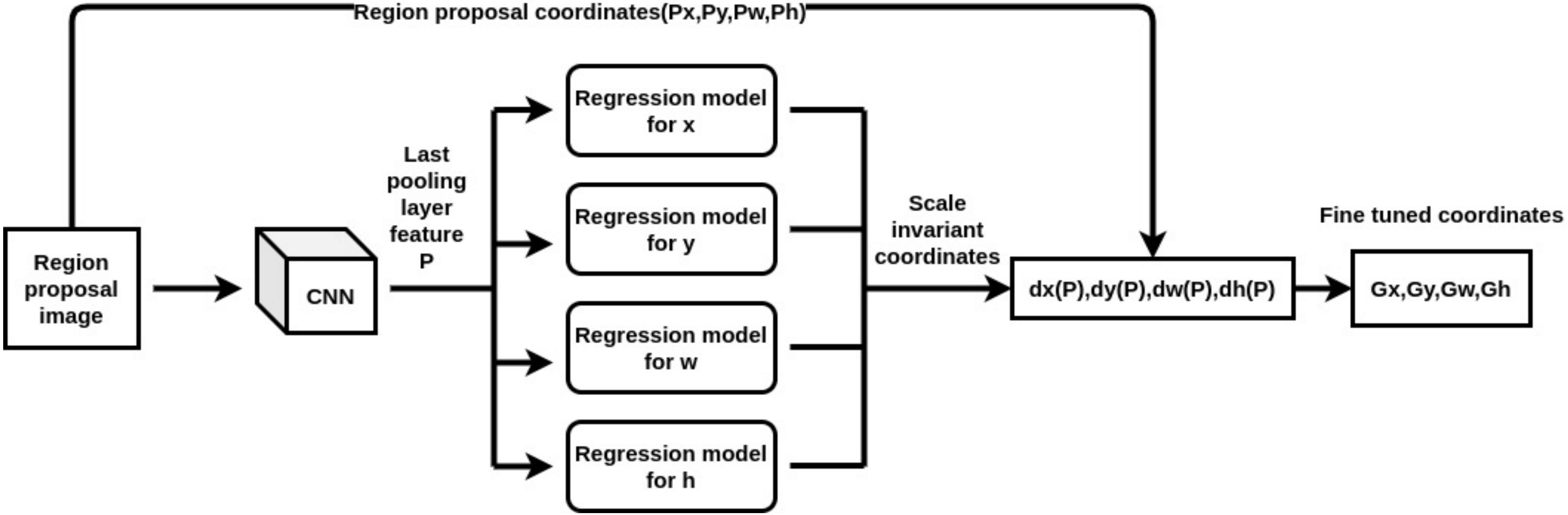


Image Source: https://towardsdatascience.com/deep-learning-method-for-object-detection-r-cnn-explained-ecdadd751d22

Key Contributions of R-CNN

1. Demonstrate how to accurately localize objects with a neural network (NN)
    - First time a CNN outperformed hand-crafted features on VOC, achieving mAP of 54% compared to 33% for previous HOG based model (VOC 2010)

2. Demonstrate how train an accurate (high-capacity) NN with a scarce amount of annotated detection data

Algorithm Training

Three models are trained independently in a series:

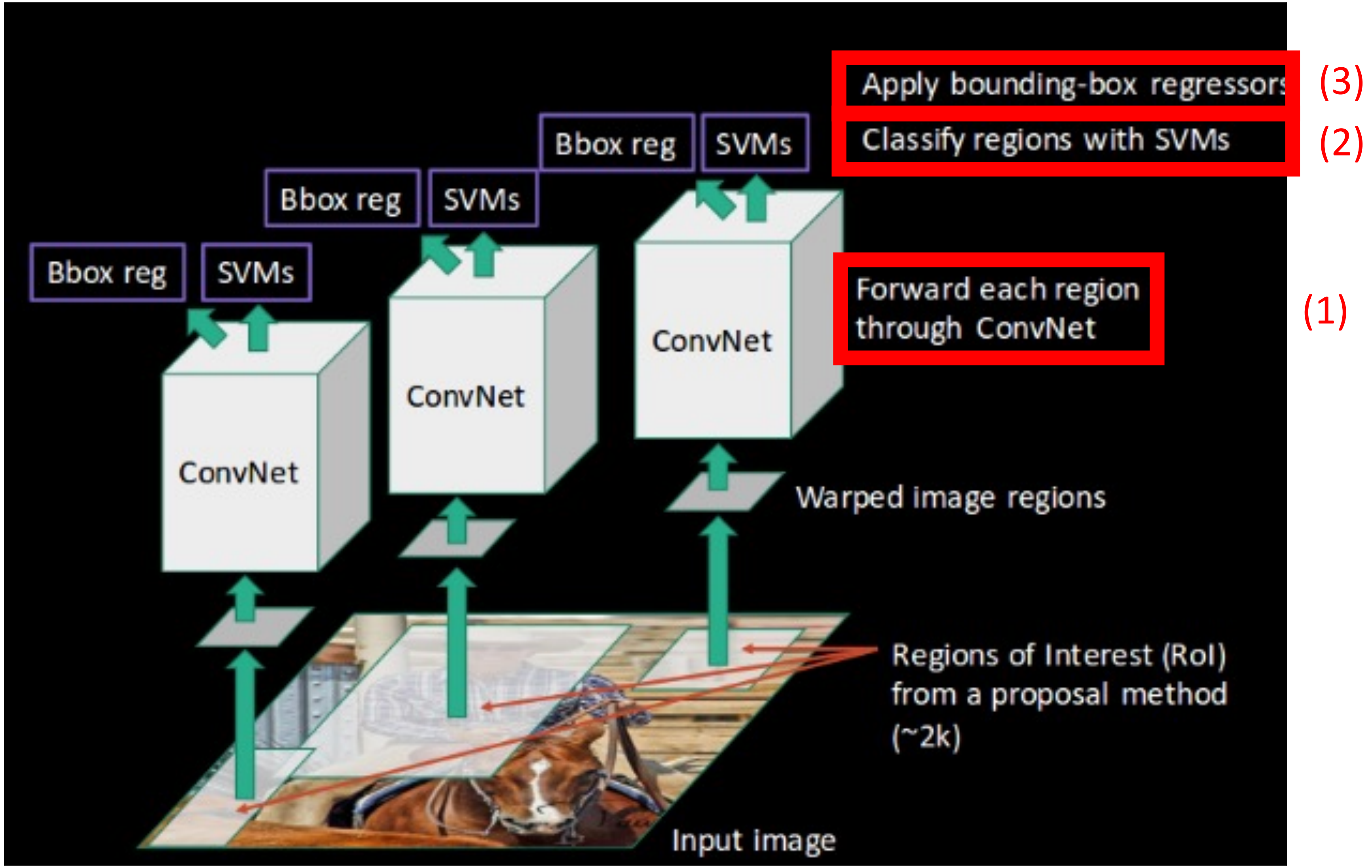
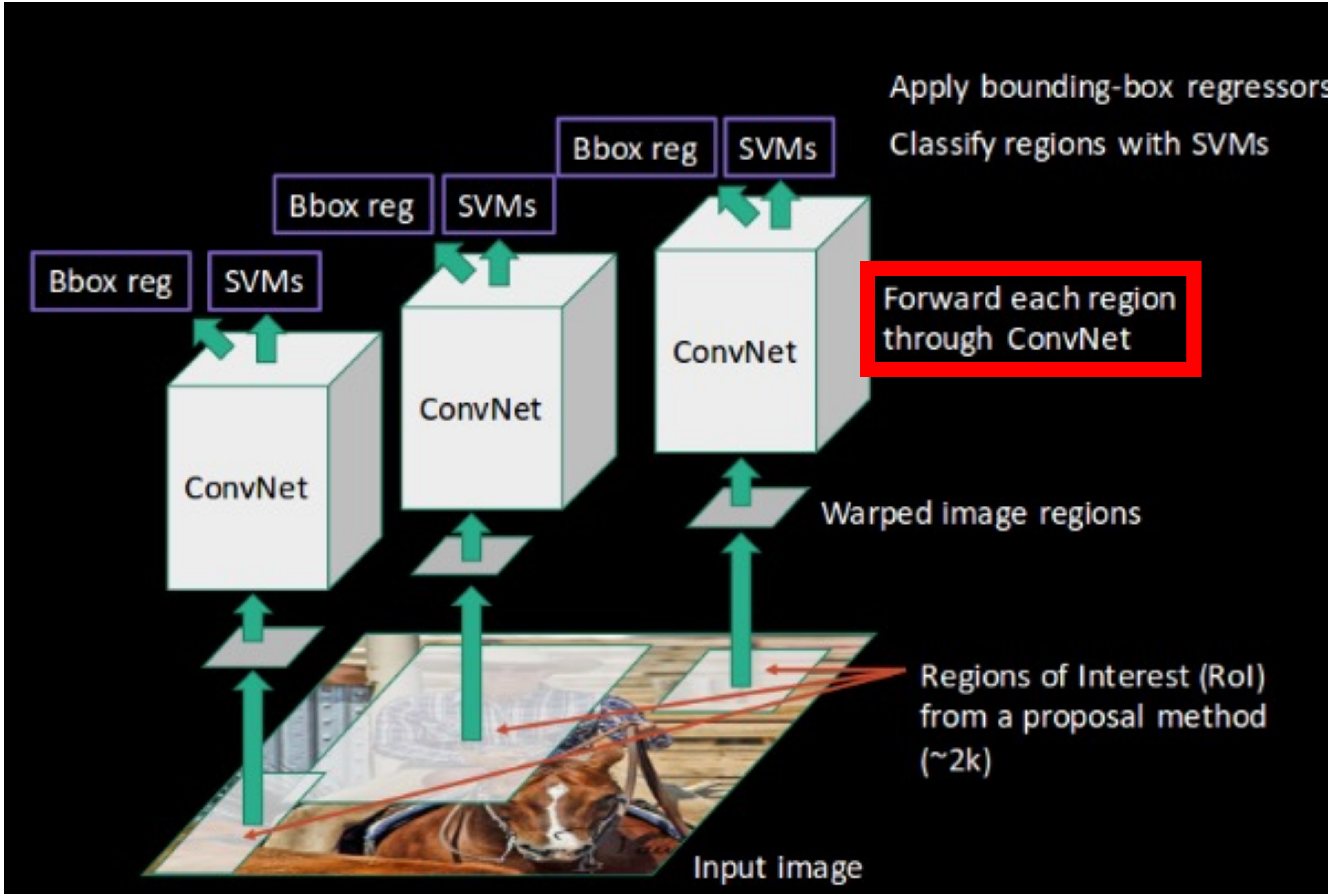


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Algorithm Training: CNN (Key Contribution)

Challenge: scarce amount of training data available in detection datasets



(1)

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

Algorithm Training: CNN (Key Contribution)

1. Train AlexNet architecture for image classification on ILSVRC (ImageNet)

Challenge: scarce amount of training data available in detection datasets

Solution: supervised pretraining on a large auxiliary dataset

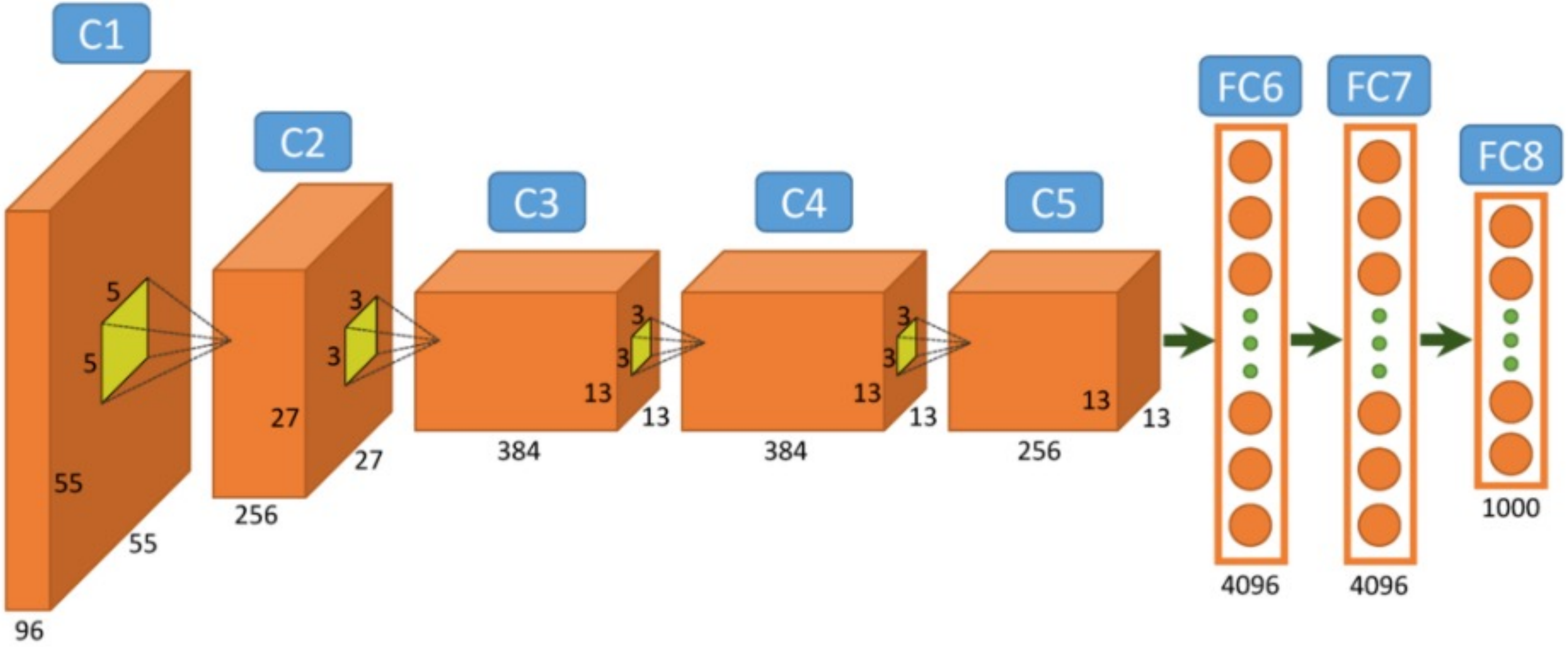
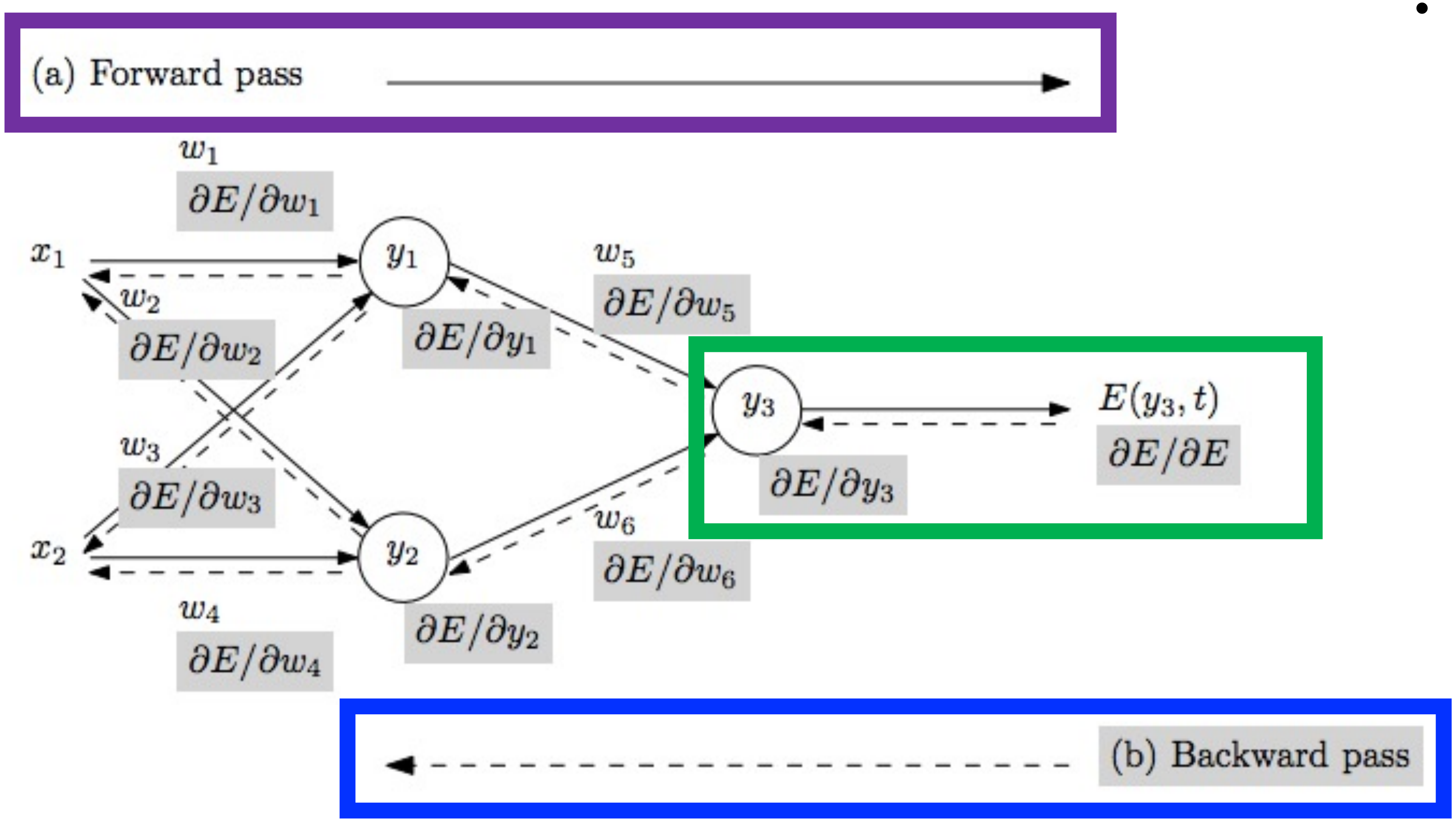


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

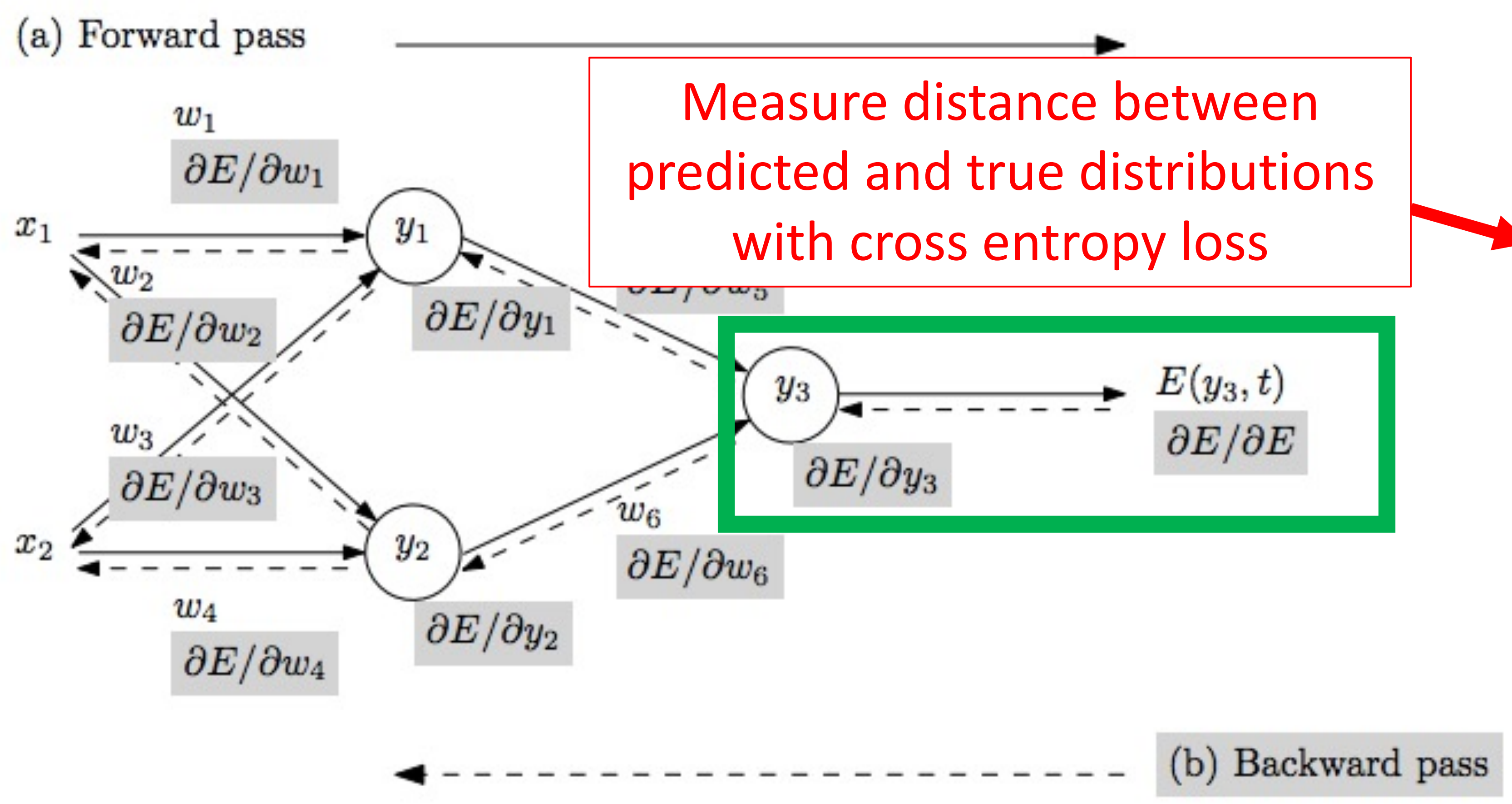
Algorithm Training: Recall How NNs Learn



- 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

Algorithm Training: CNN



Measure distance between predicted and true distributions with cross entropy loss

- 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

Algorithm Training: CNN

1. Train AlexNet architecture for image classification on ILSVRC (ImageNet)
2. Change final layer (FC8) to reflect number of categories in VOC (20 + background)
3. Train pretrained architecture for image classification on VOC (choose max IoU class; positive if IoU >= 0.5)

Key challenge: relatively little training data available in detection datasets

Solution: supervised pretraining on a large auxiliary dataset

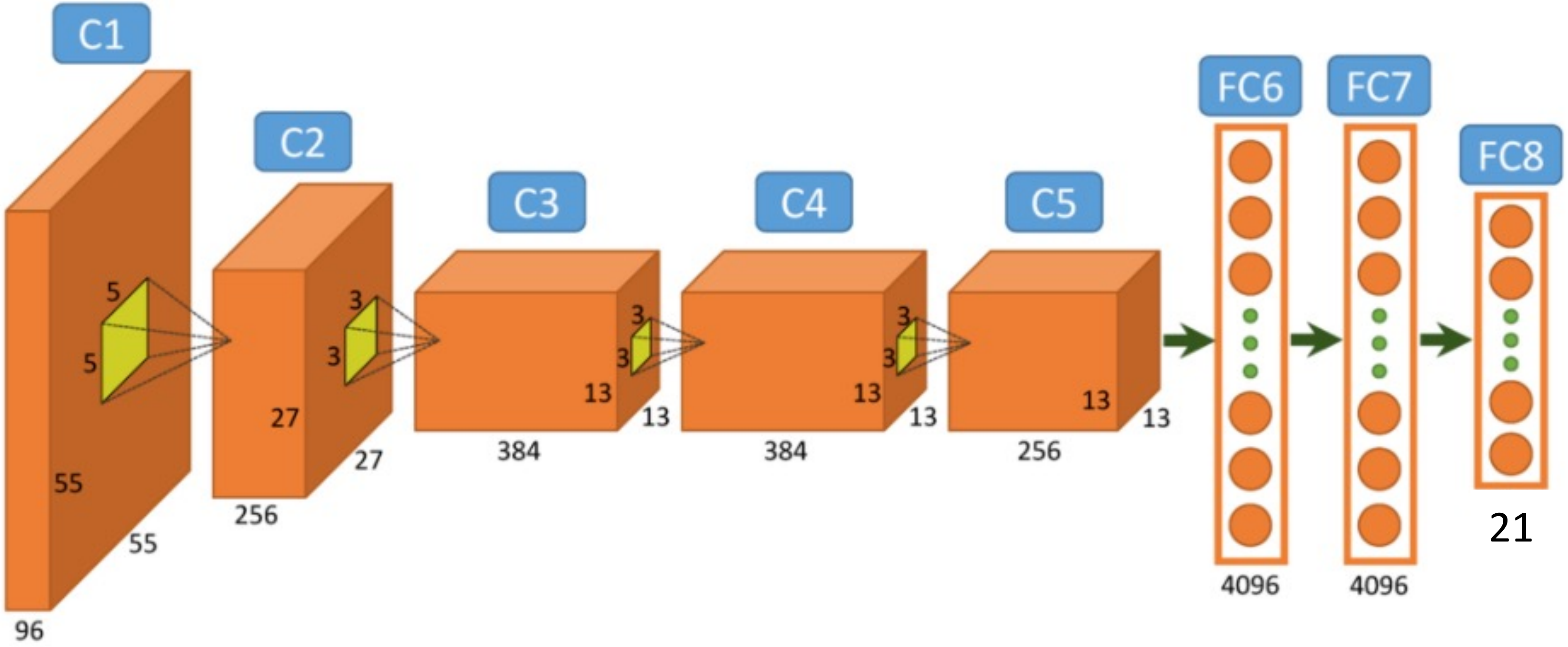
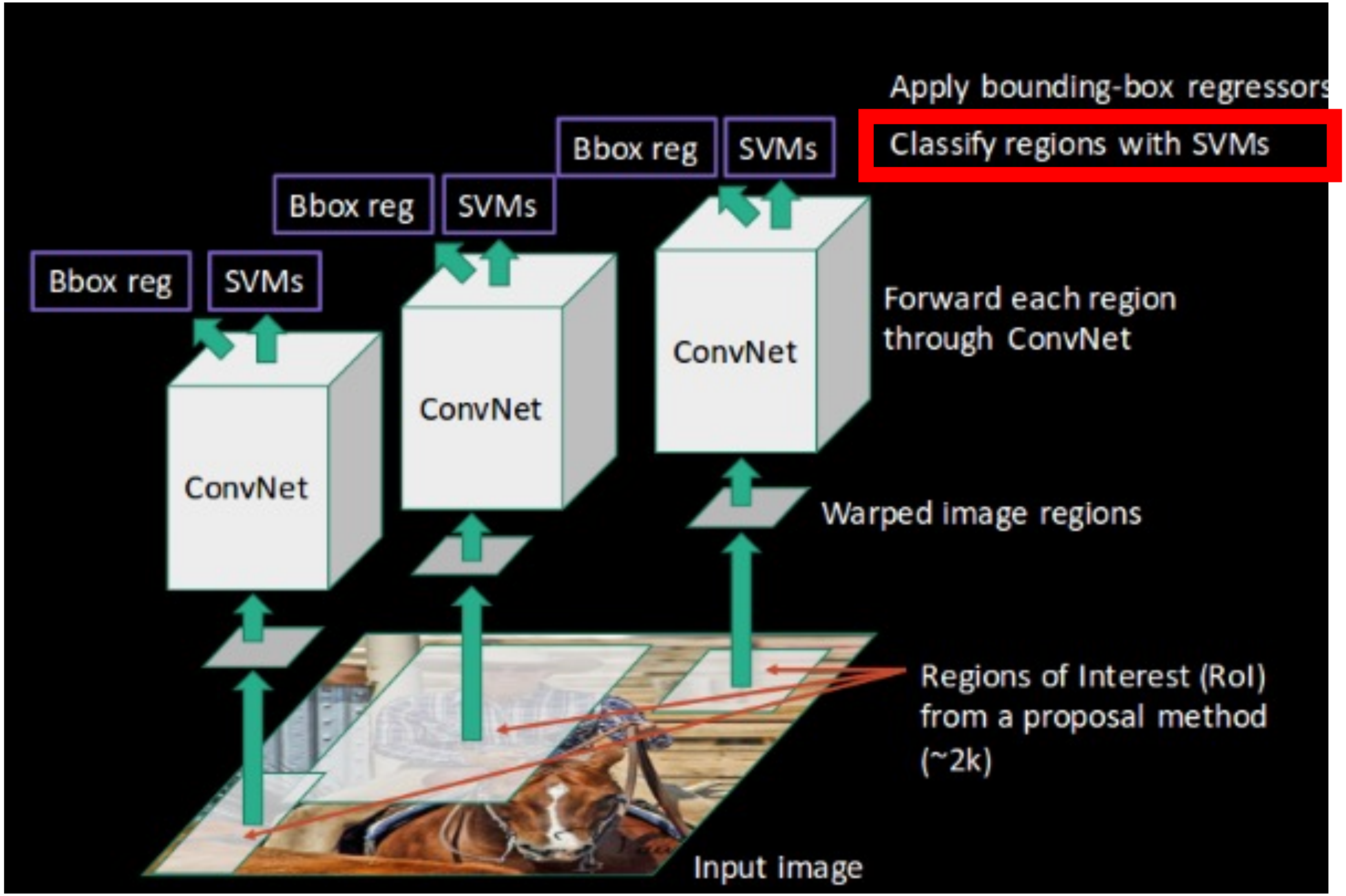


Image Source: https://www.researchgate.net/figure/Architecture-of-Alexnet-From-left-to-right-input-to-output-five-convolutional-layers_fig2_312303454

Algorithm Training

Three models are trained independently in a series:



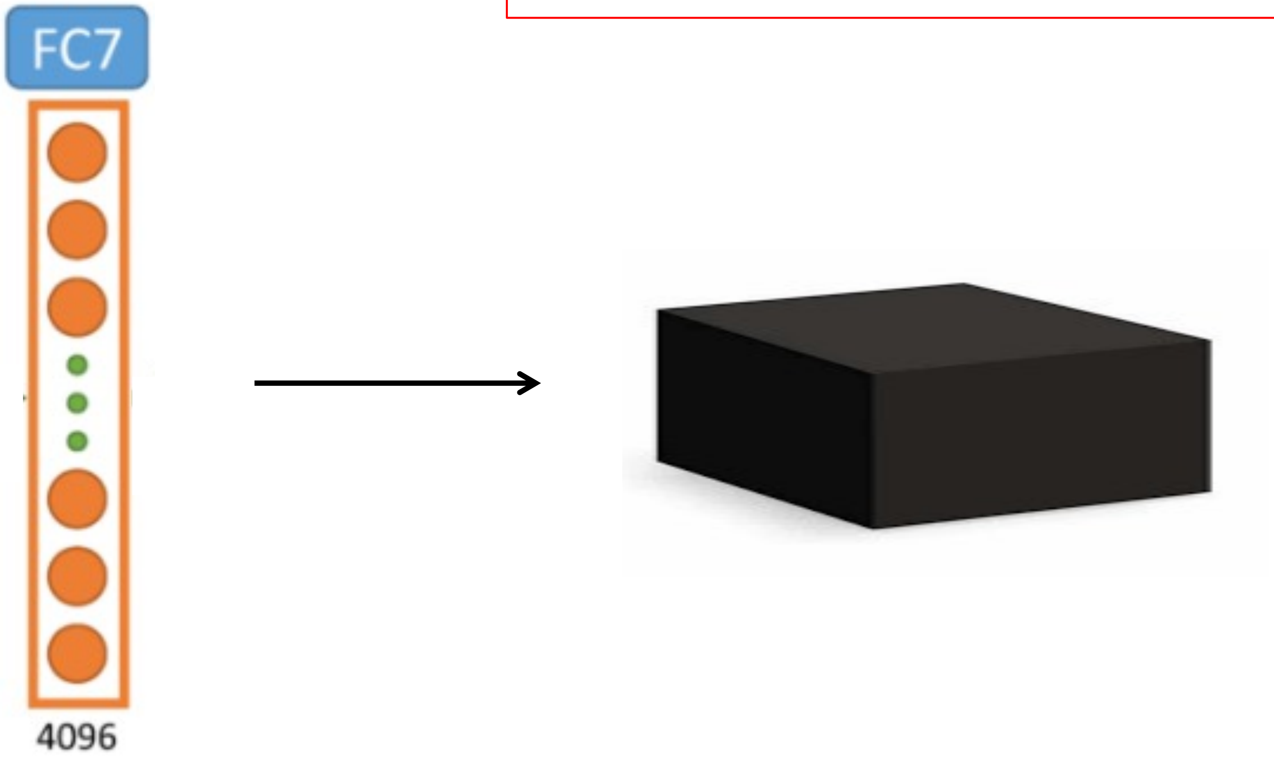
(2)

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

Algorithm Training: SVM Per Category

Input: Feature descriptor
of a region) + yes/no label

Yes label: GT BB
No label: IoU < 0.3 because of testing



Algorithm Training

Three models are trained independently in a series:

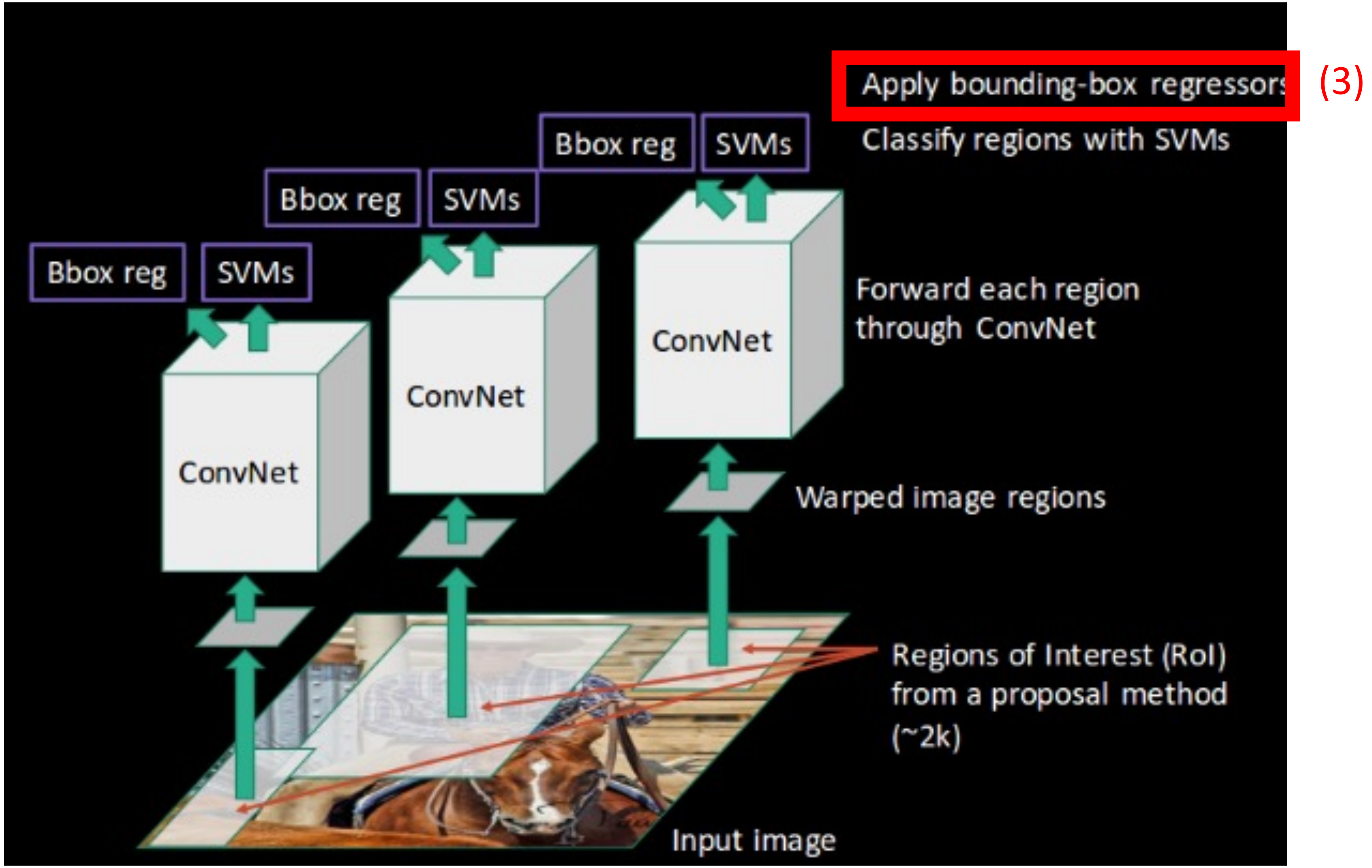


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

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

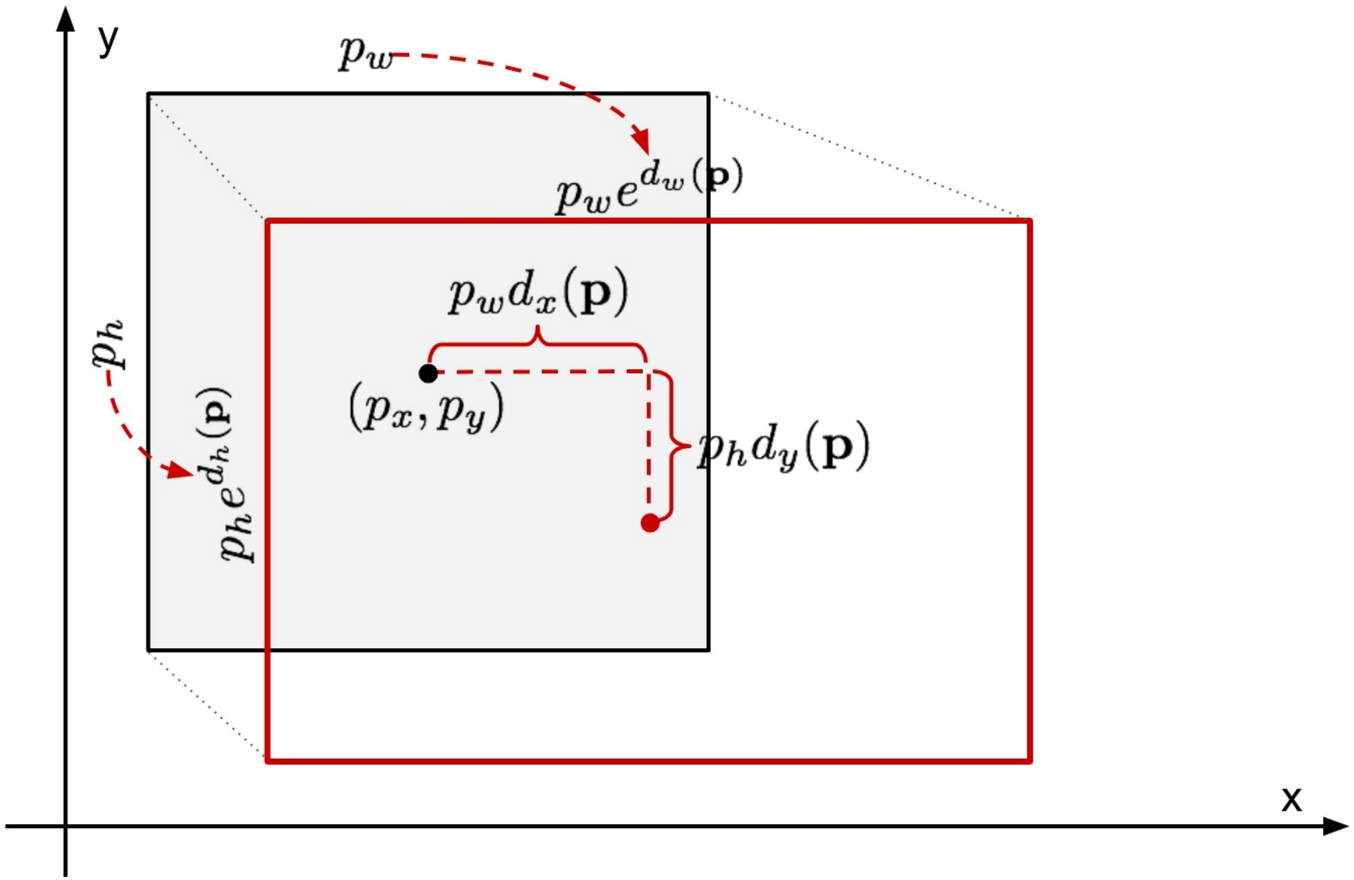


Image Source: https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html#bounding-box-regression

Limitations

- Slow at test time
  (~1 minute per image)
- Slow/complex training procedure
  - Must train three models
- Inefficient/complex architecture
  - Must store feature descriptor for each region proposal
  - Must refine initial region proposals

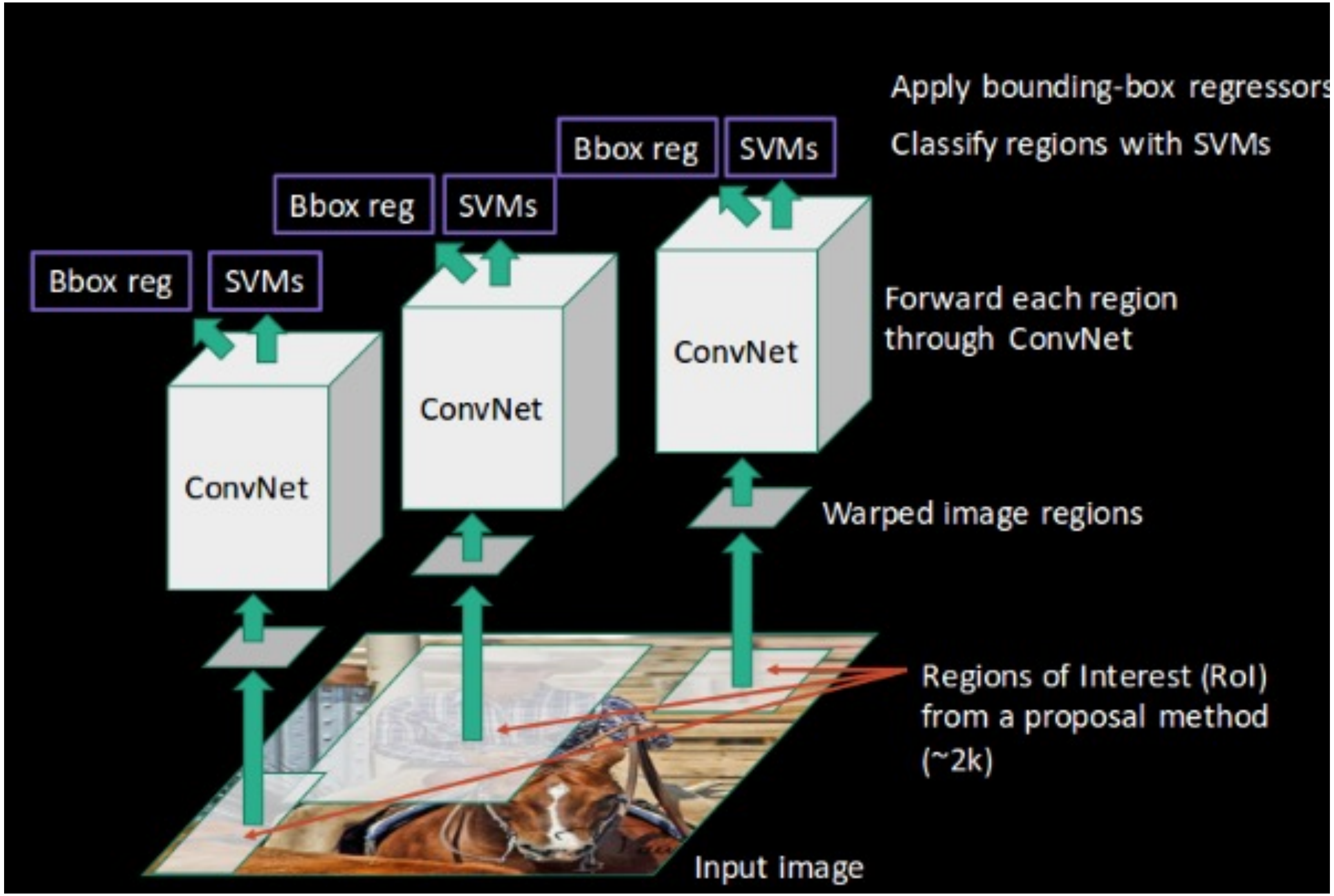
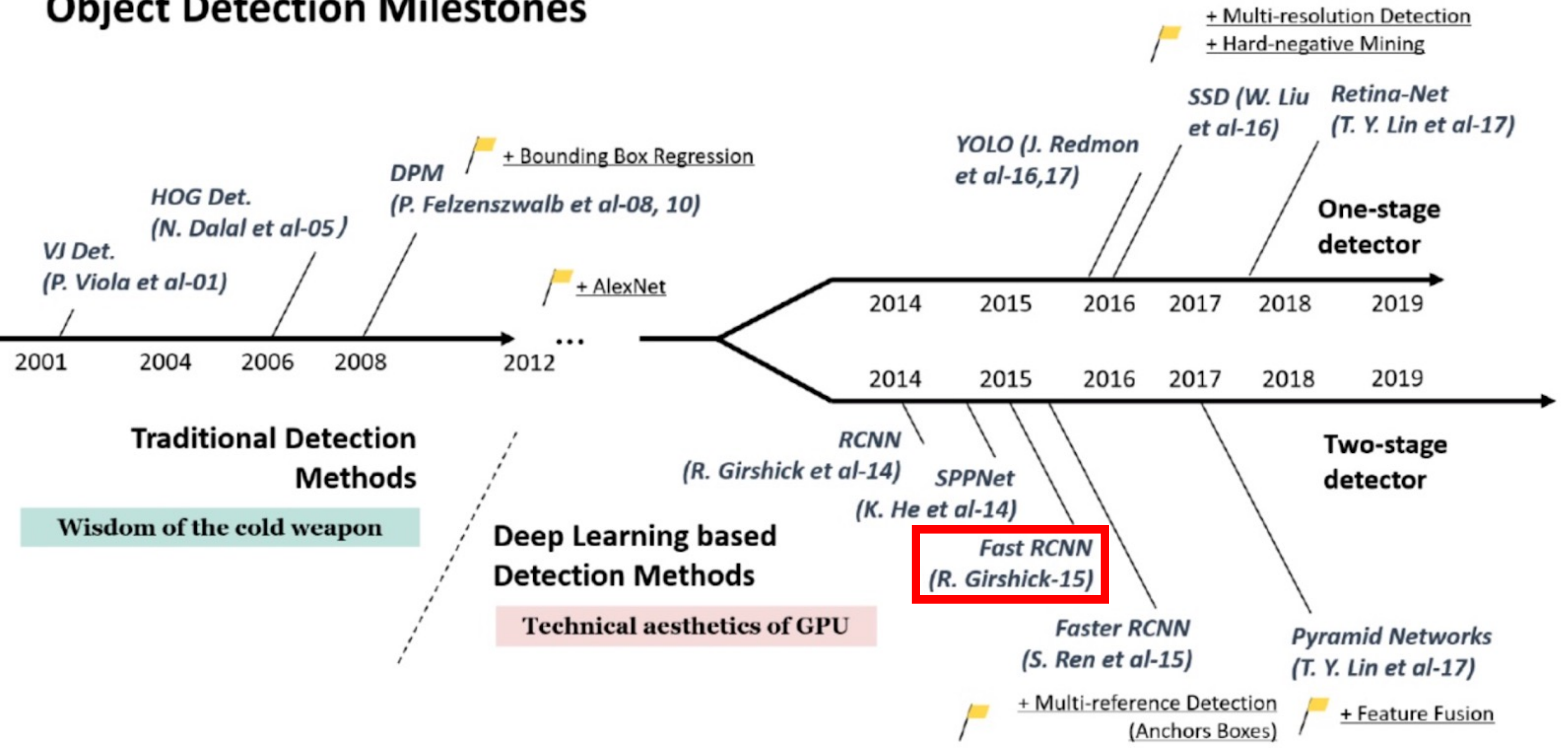


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Object Detection: Today's Topics

- Overview of object detection algorithms
- Baseline Model: R-CNN
- Fast R-CNN
- Faster R-CNN
- YOLO
- Discussion

Object Detection Milestones



Zhou et al. Object Detection in 20 Years: A Survey. arXiv 2019.

Key Contributions of Fast R-CNN

1. State of art object detection model in terms of accuracy and speed
   1. 9x faster than R-CNN
   2. mAP of 66% vs 62% for R-CNN on VOC2012

2. Reduced storage requirements by not requiring features to be stored for each region proposal

3. A training algorithm that learns in a single stage (rather than the three stages required by R-CNN)

Key Contributions of Fast R-CNN

1. State of art object detection model in terms of accuracy and speed
    1. 9x faster than R-CNN
    2. mAP of 66% vs 62% for R-CNN on VOC2012

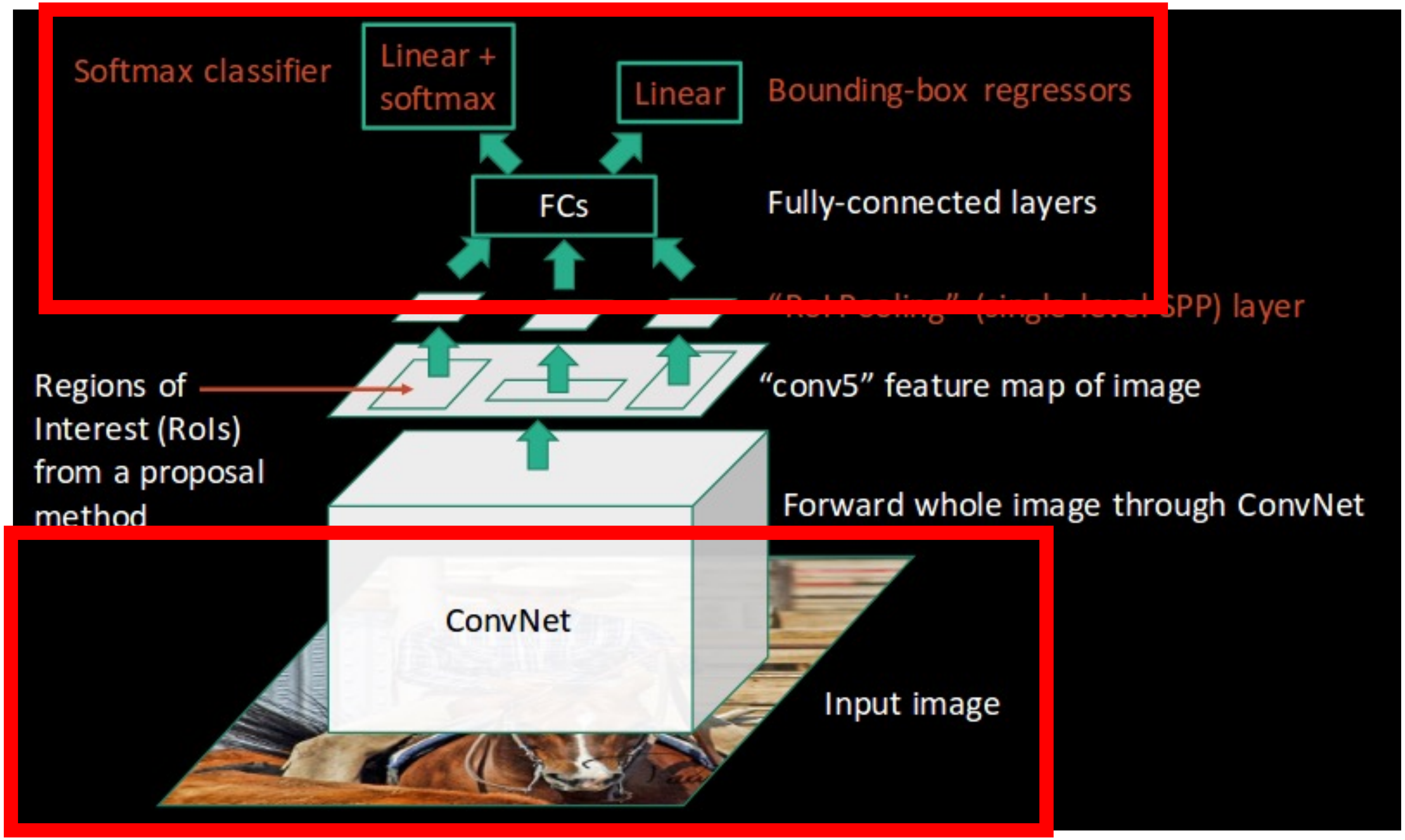
2. Reduced storage requirements by not requiring features to be stored for each region proposal

3. A training algorithm that learns in a single stage (rather than the three stages required by R-CNN)

Architecture

Output: (left) probability distribution over all classes plus background and (right) four real valued numbers for each object class

How many nodes would be at the bounding box regressor assuming 20 object classes?



Input: any image size

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

Architecture

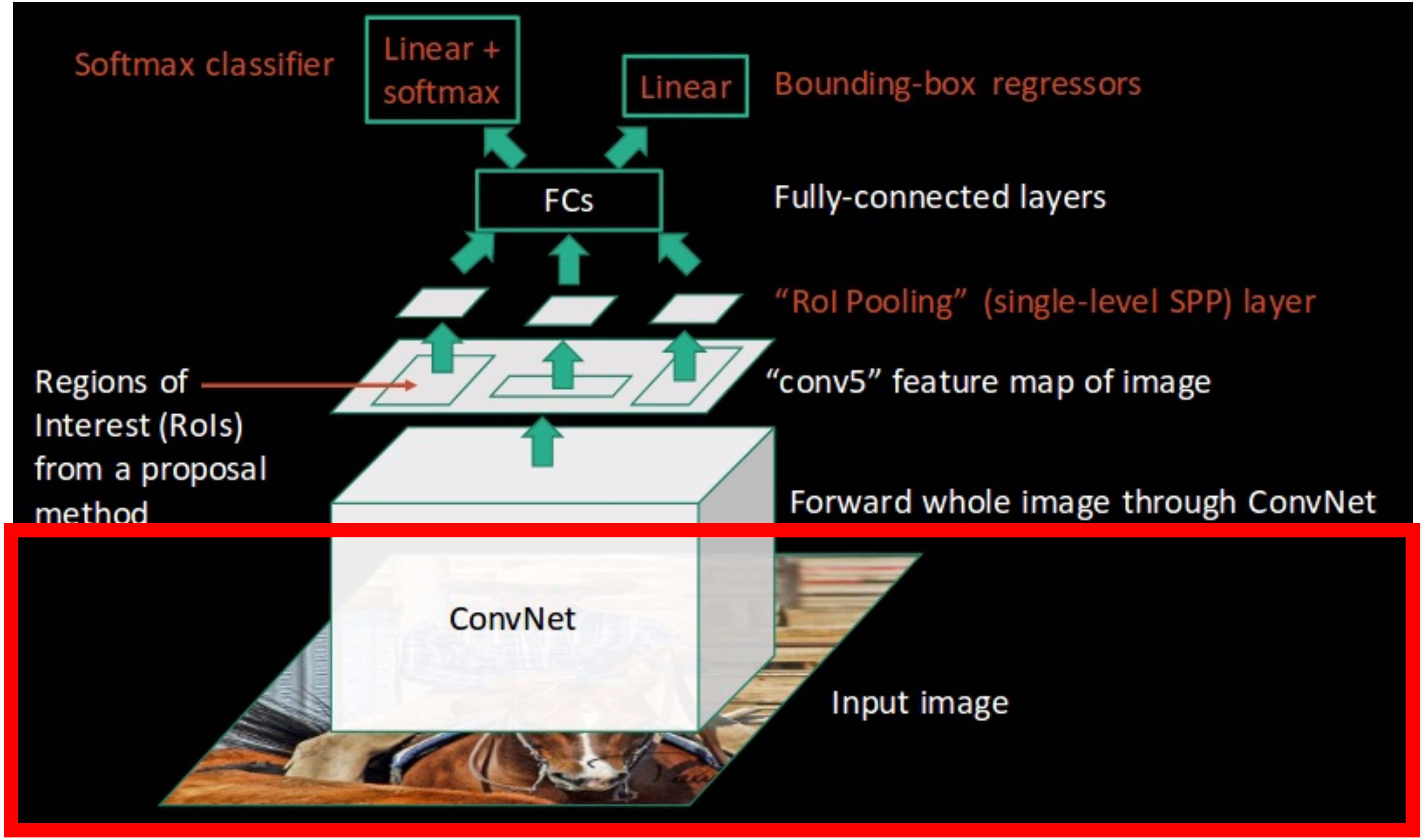


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture: Define Candidate Detection Regions

Given an image, produce bounding boxes around
“object”-like using selective search (same as R-CNN)

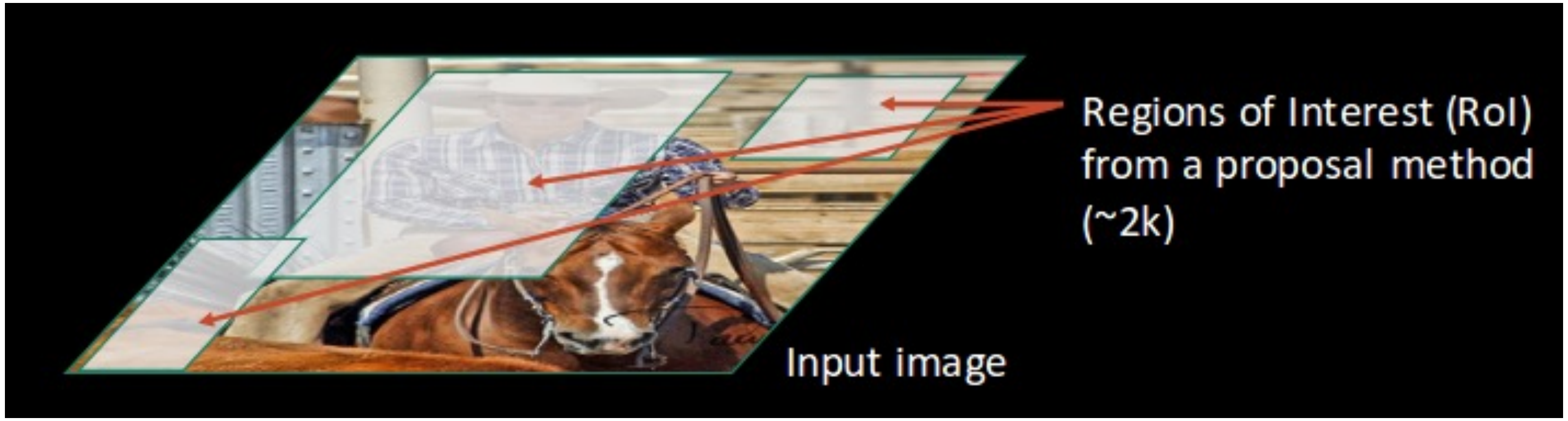


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture (Key Contribution)

Rather than run a pretrained network repeatedly to extract a feature descriptor per region, just use the section of the feature maps of last correspond to the region.

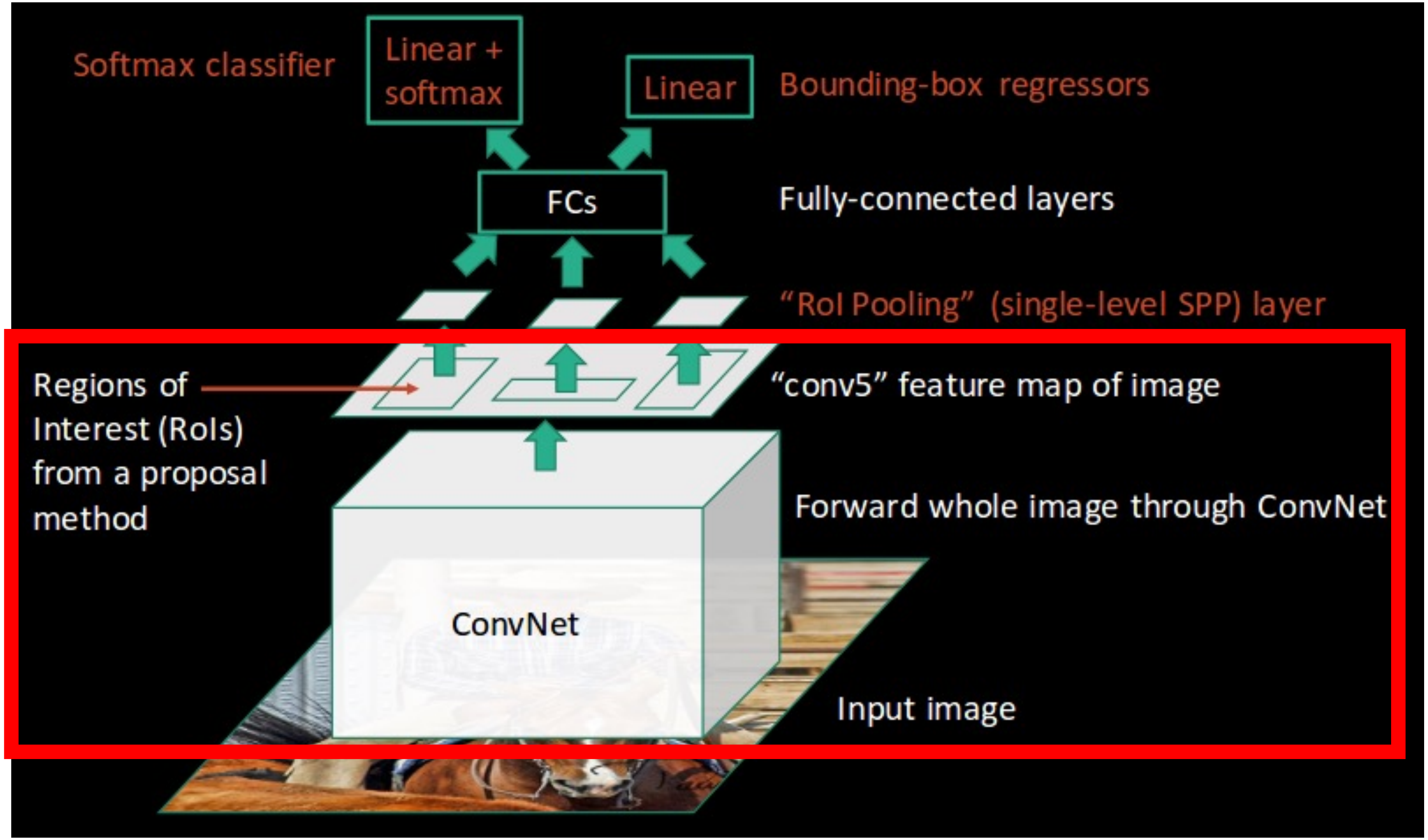


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture: Support Arbitrary Region Shapes

Use spatial pyramid pooling to support any input image since it converts an arbitrary size into a fixed size needed by the fully connected network

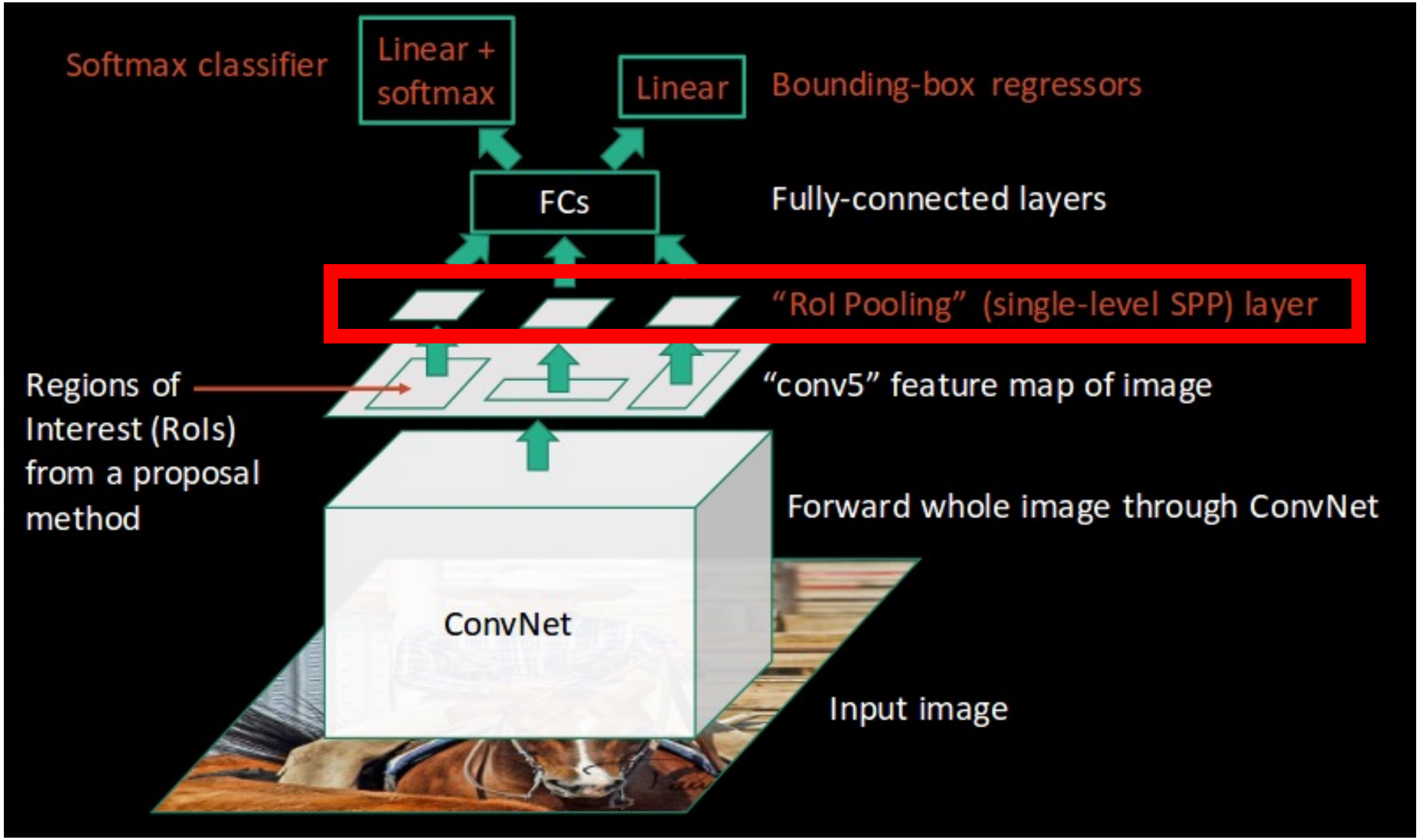


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture: Region Classification & Refinement

Each region descriptor is assigned to a class and final location

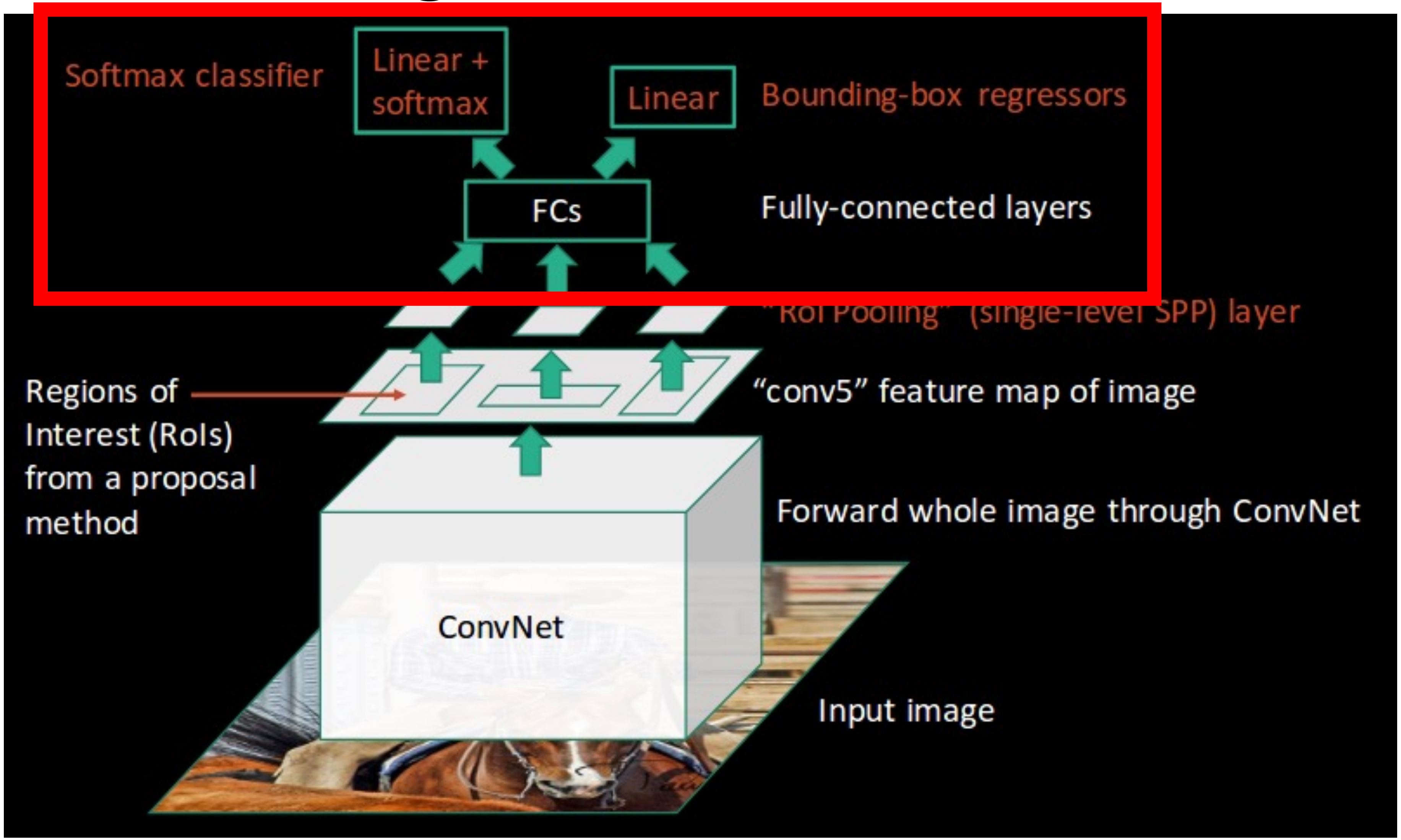
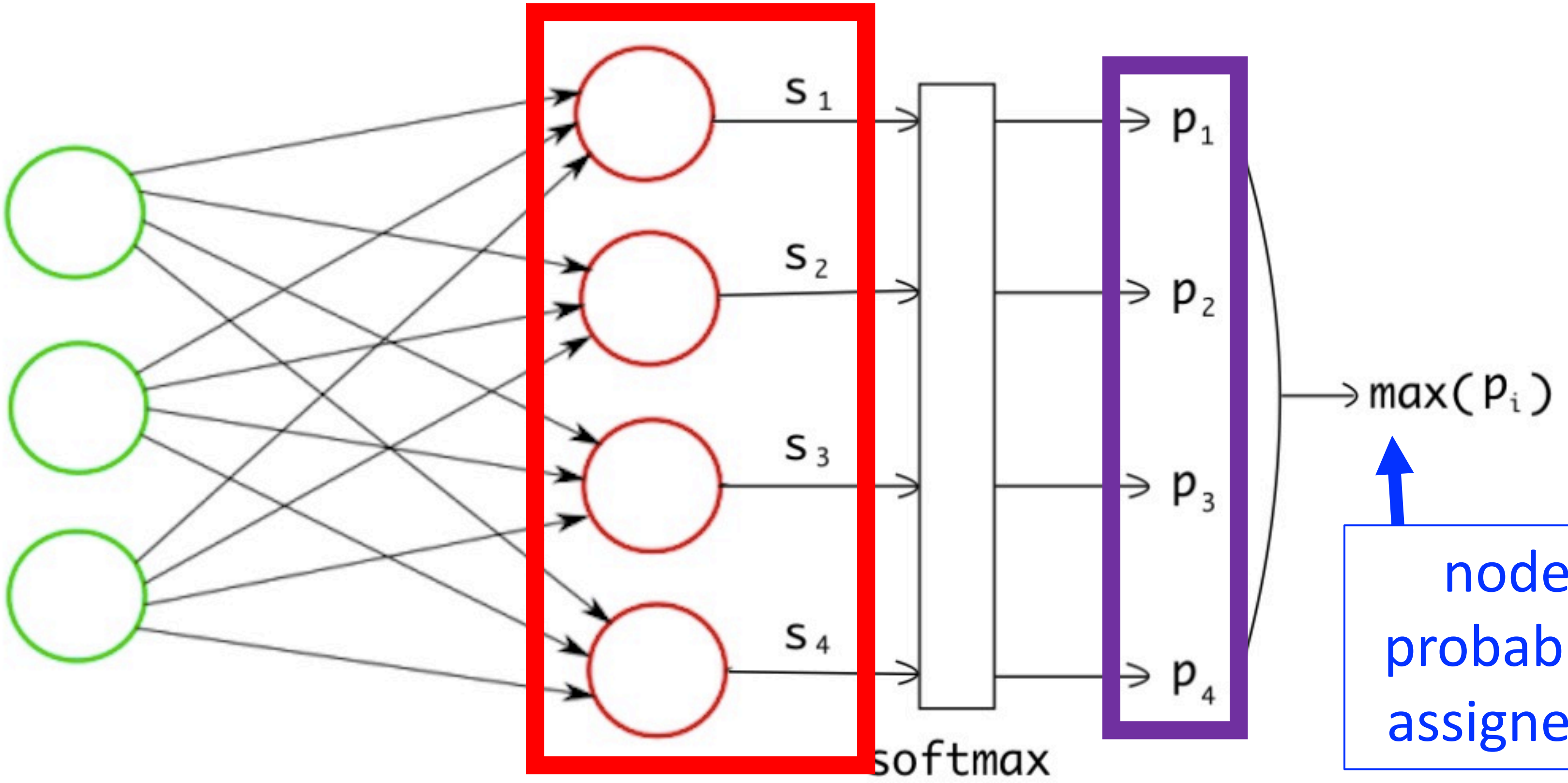


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

Architecture: Region **Classification** & Refinement

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,



node with largest probability is the class assigned to the image

Figure Source: https://towardsdatascience.com/multi-label-image-classification-with-neural-network-keras-ddc1ab1afede

Architecture: Region Classification & **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_y)

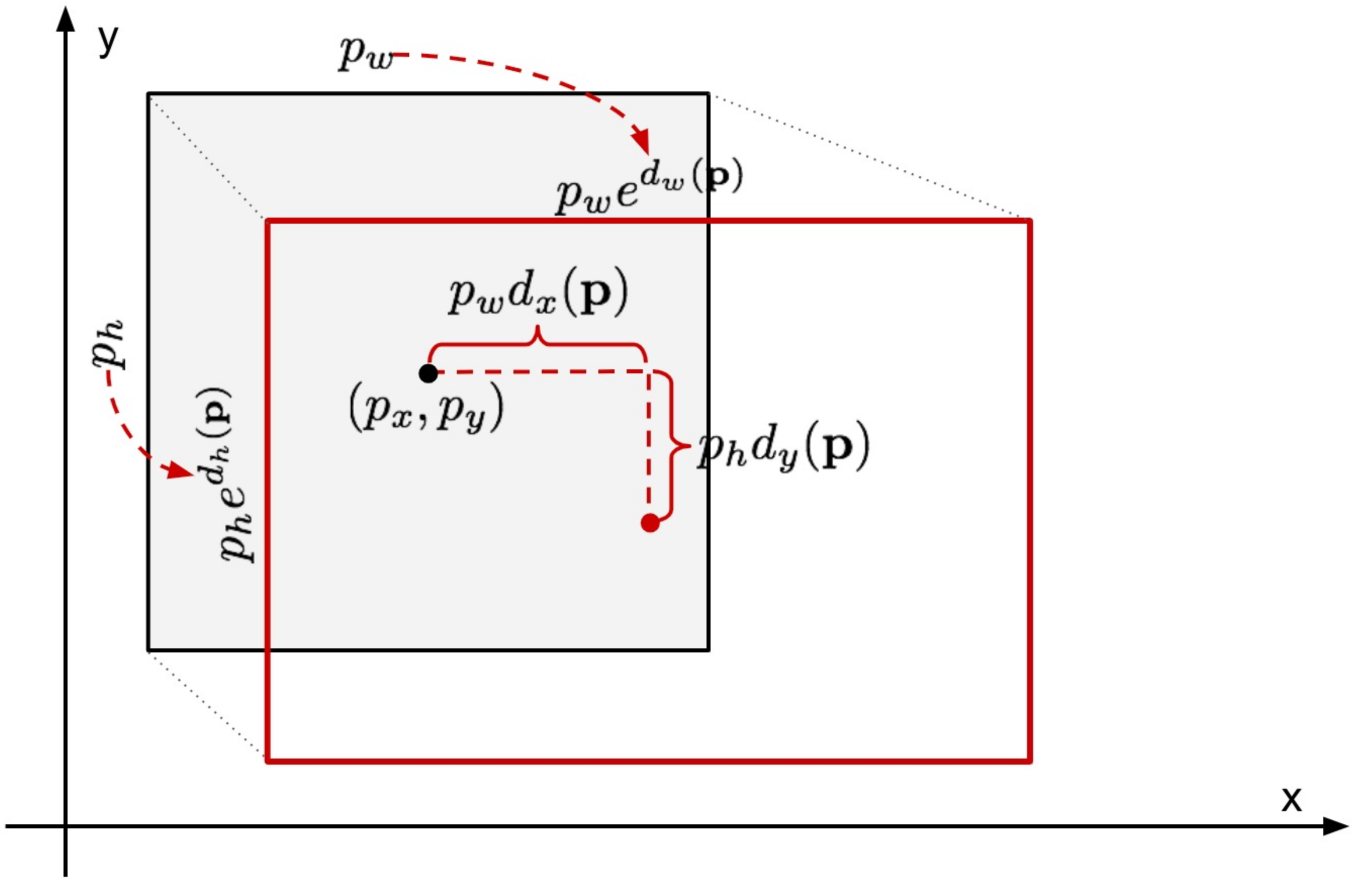


Image Source: https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html#bounding-box-regression

Key Contributions of Fast R-CNN

1. State of art object detection model in terms of accuracy and speed
   1. 9x faster than R-CNN
   2. mAP of 66% vs 62% for R-CNN on VOC2012

2. Reduced storage requirements by not requiring features to be stored for each region proposal

3. A training algorithm that learns in a single stage (rather than the three stages required by R-CNN)

Algorithm Training

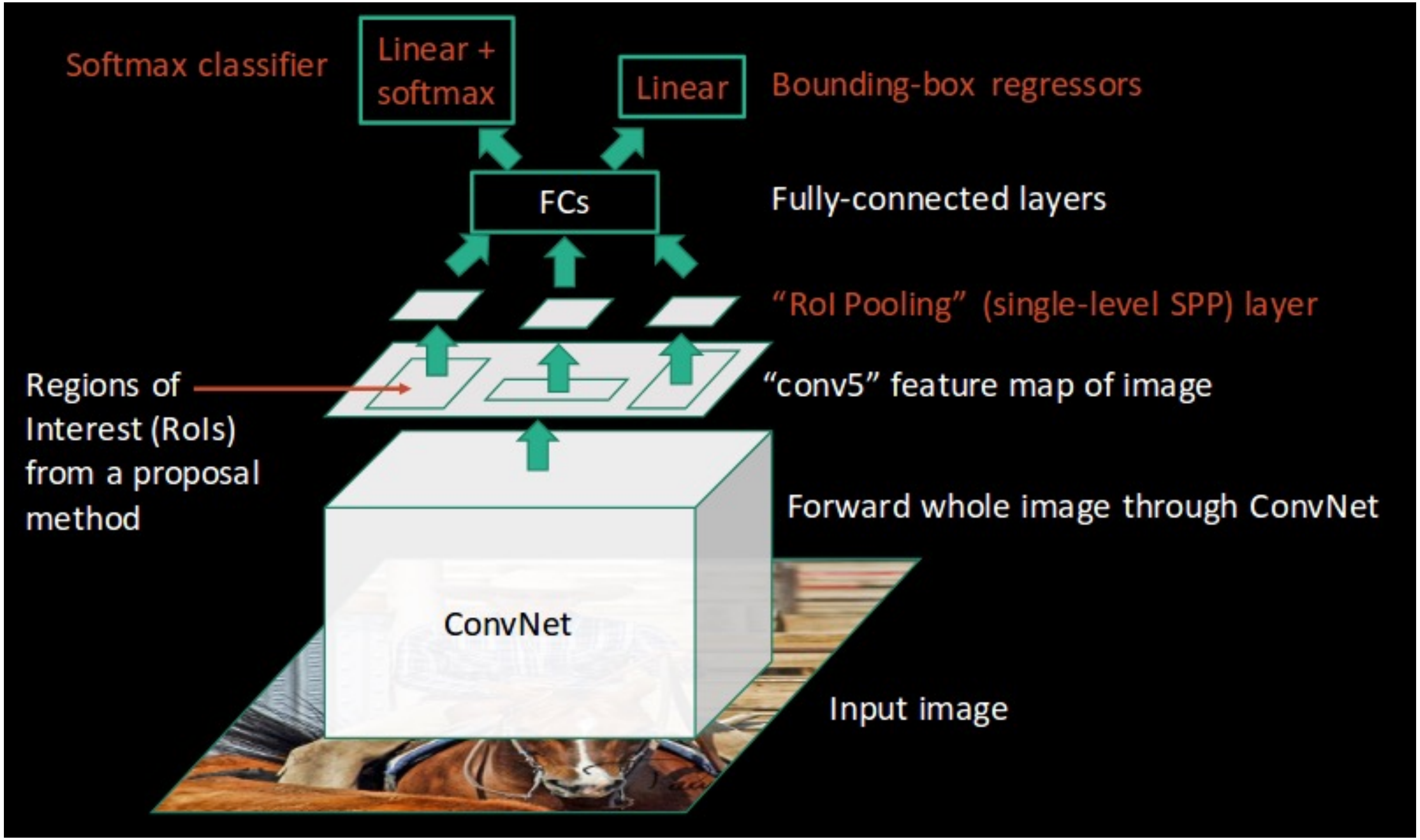
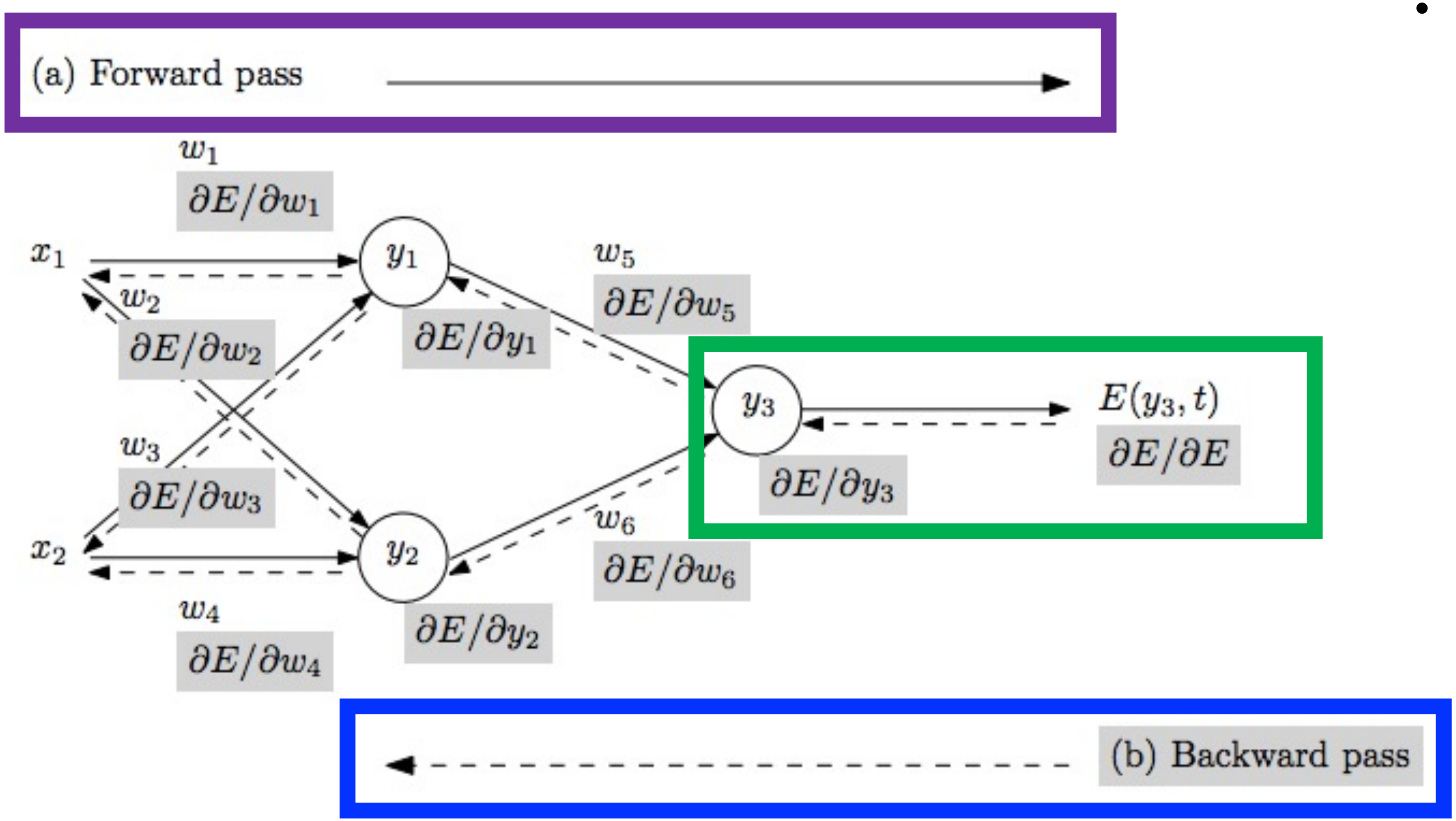


Figure Source: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

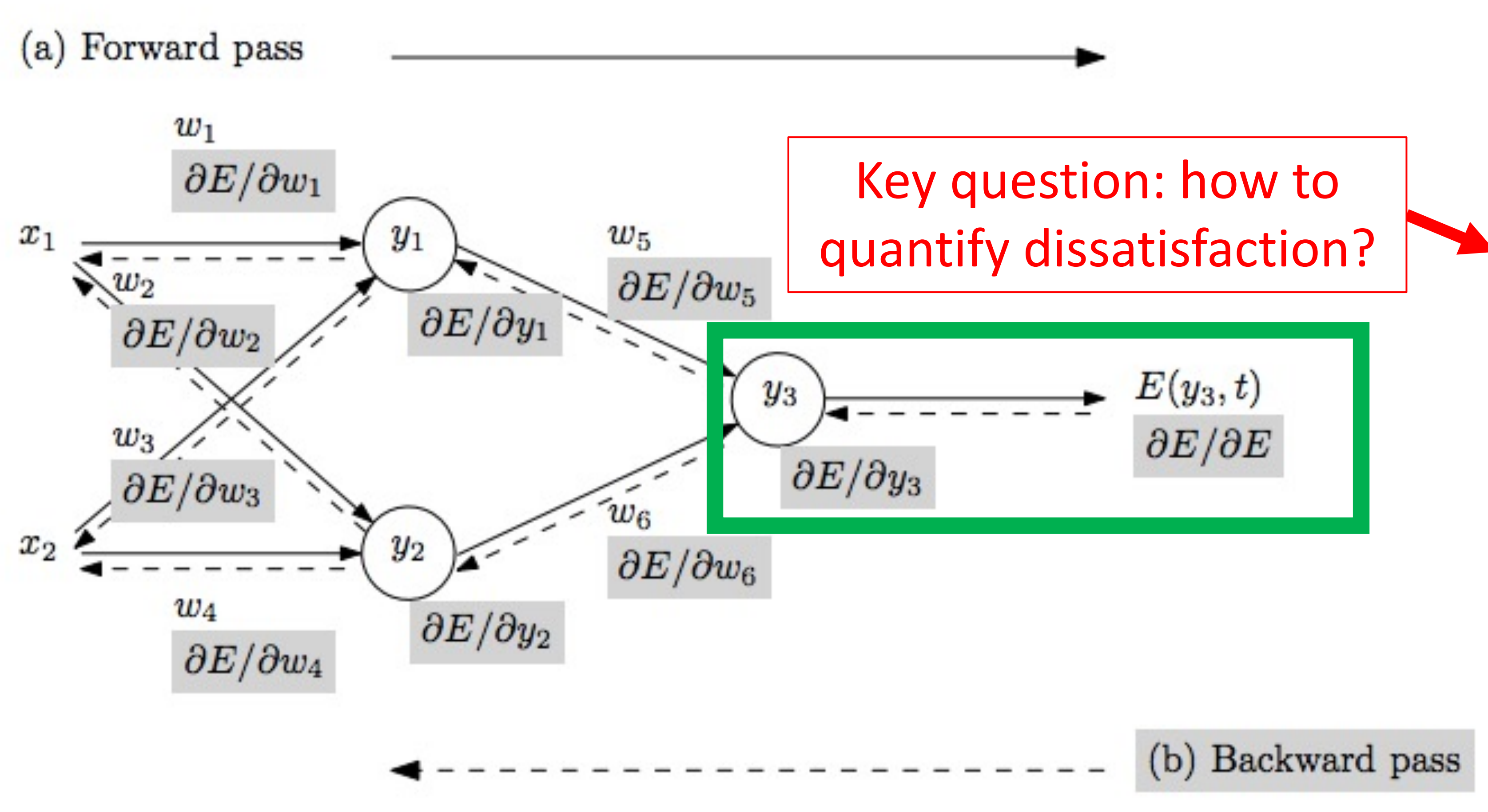
Algorithm Training: Recall How NNs Learn



- 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

Algorithm Training

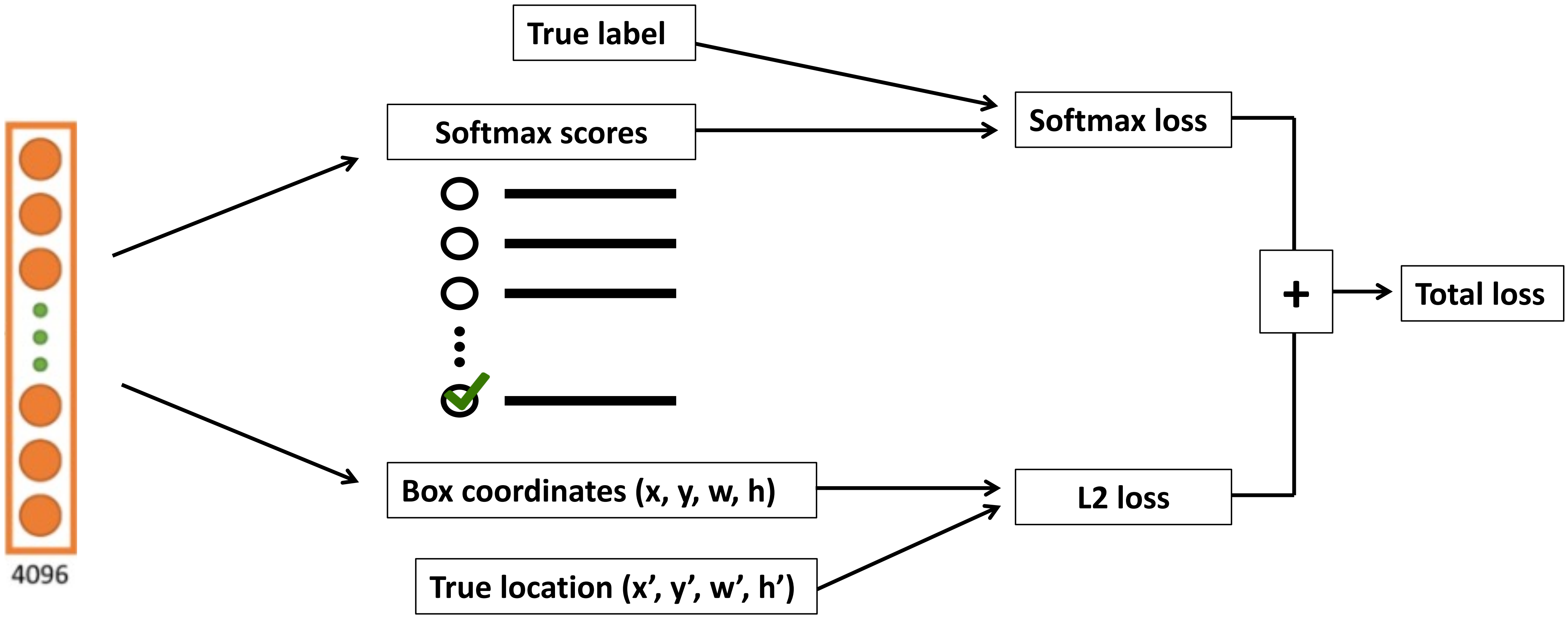


- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make prediction
  2. Quantify the dissatisfaction with a model's results on the training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model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

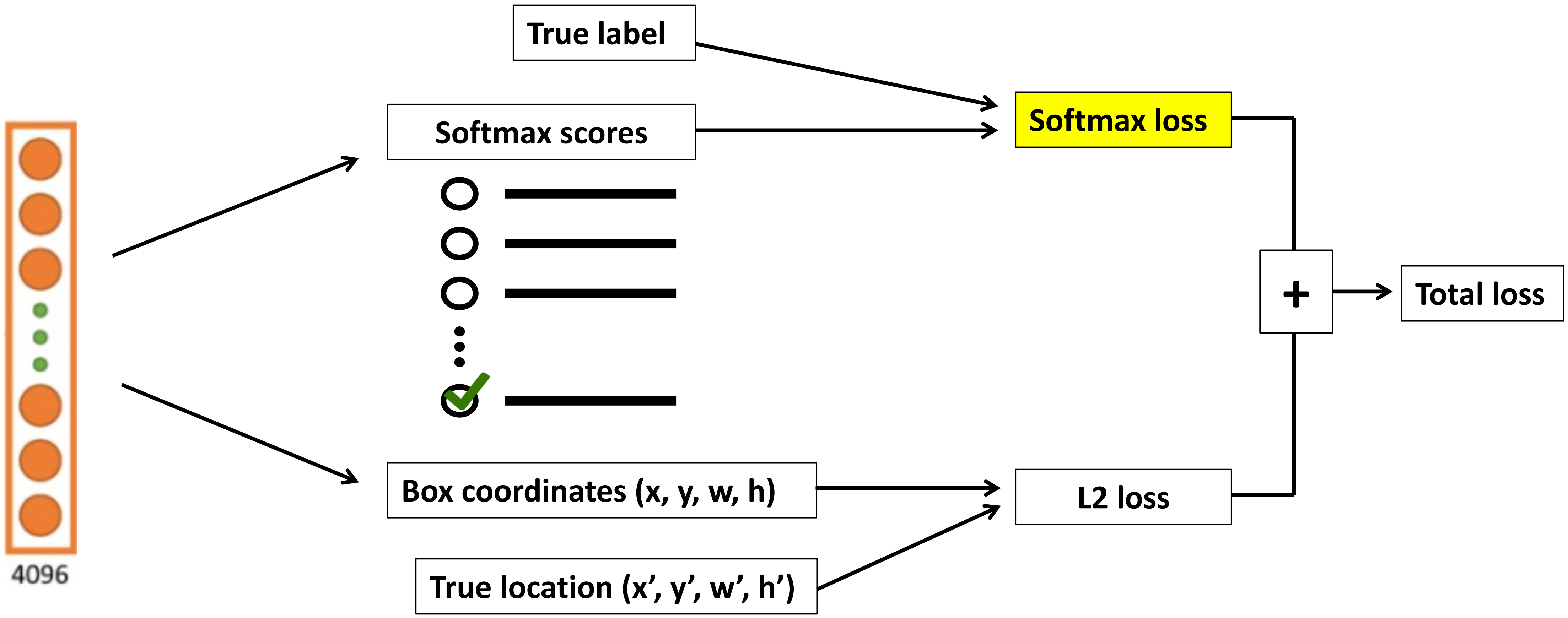
Algorithm Training: Multi-task Loss

Loss for each region proposal is sum of classification and localization losses

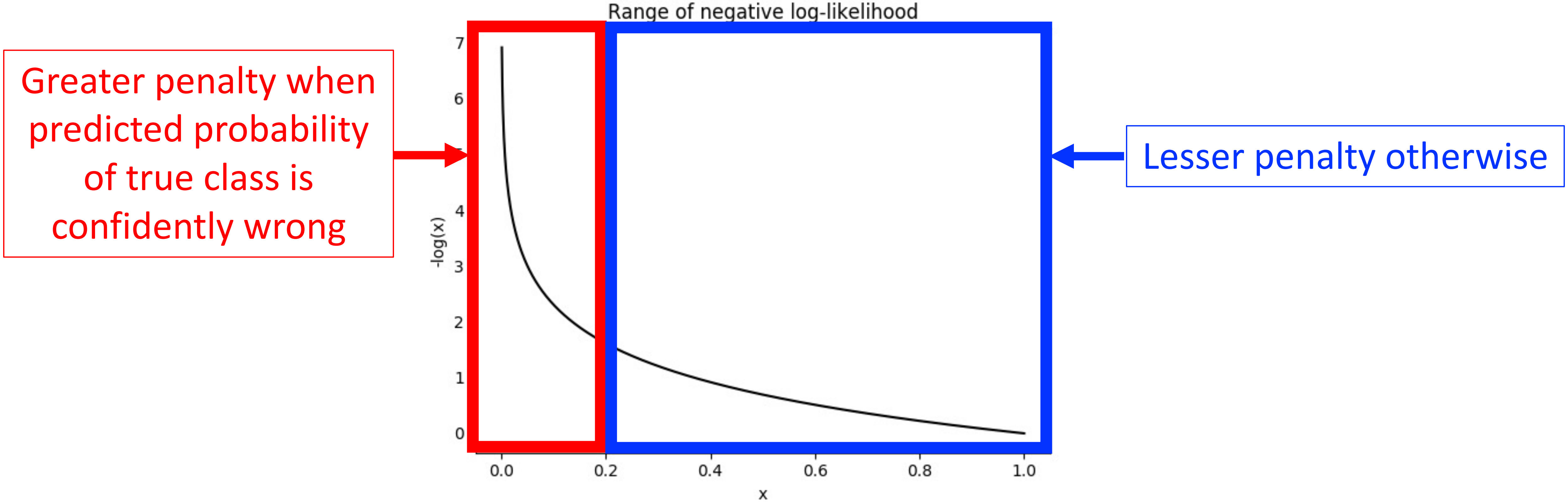


Algorithm Training: Multi-task Loss

Loss for each region proposal is sum of classification and localization losses



Algorithm Training: Classification Loss (Recap)



What is the range of possible values?

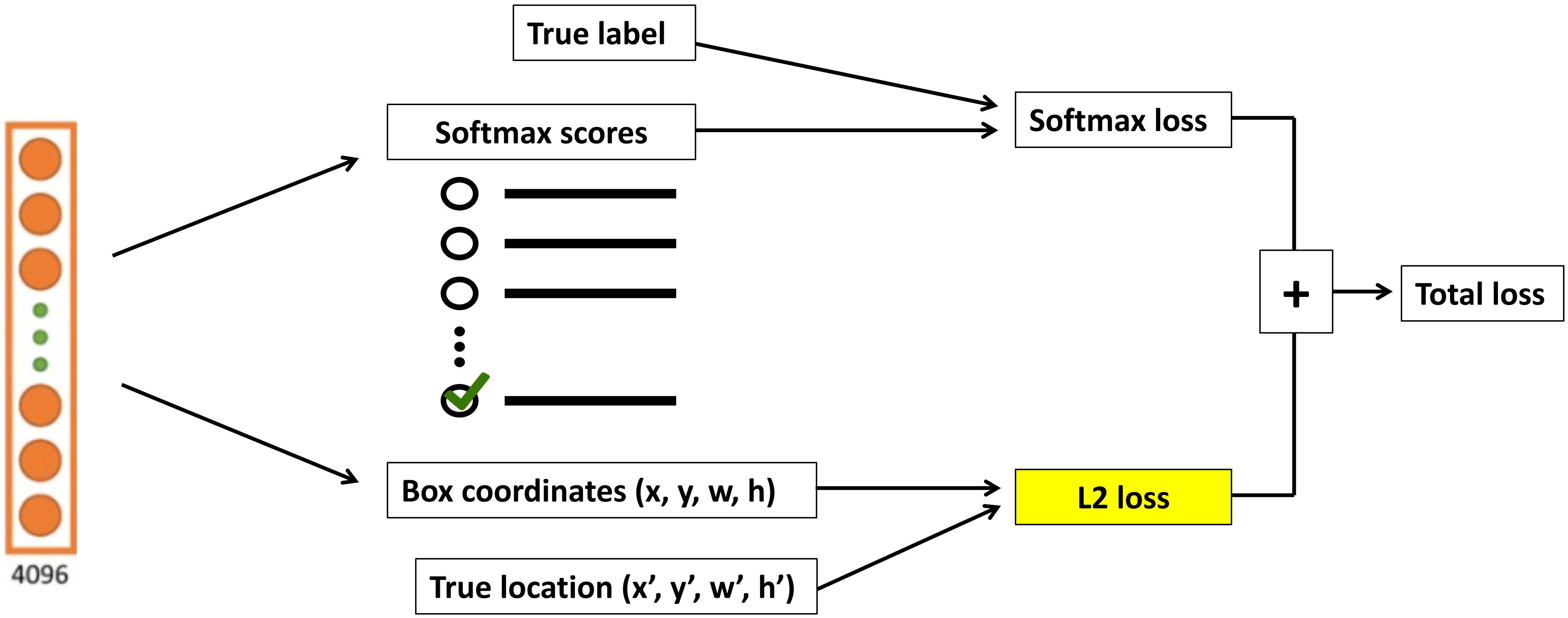
- Minimum: 0 (negative log of 1)
- Maximum: Infinity (negative log of 0)

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

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

Algorithm Training: Multi-task Loss

Loss for each region proposal is sum of classification and localization losses



Algorithm 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

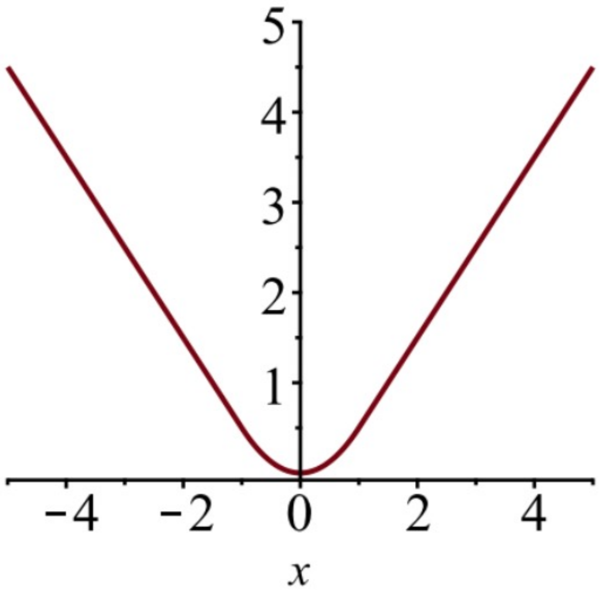


Image Source: https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html#bounding-box-regression

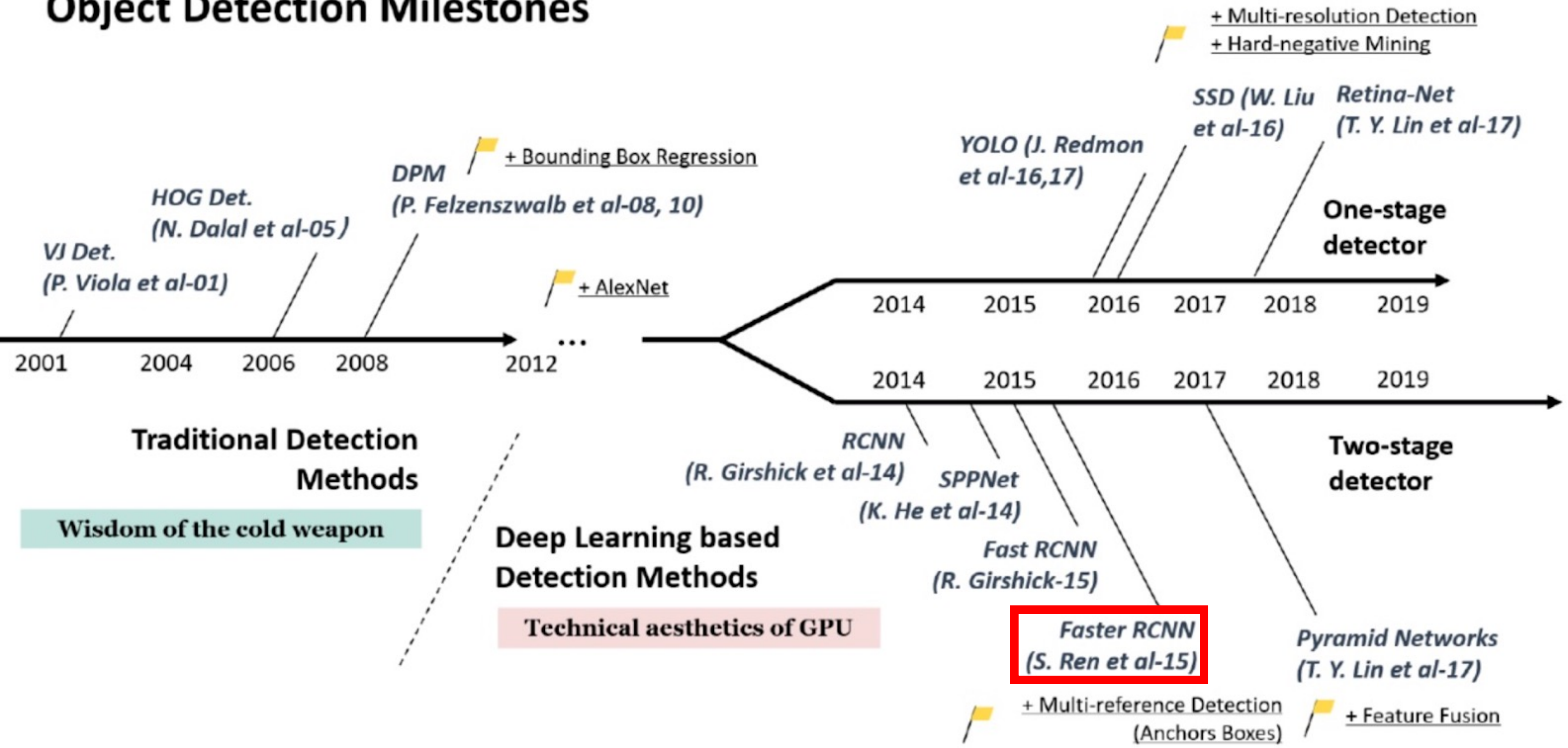
Key Limitation: Still Slow

- Requires time spent generating region proposals; i.e., selective search

Object Detection: Today’s Topics

- Overview of object detection algorithms
- Baseline Model: R-CNN
- Fast R-CNN
- Faster R-CNN
- YOLO
- Discussion

Object Detection Milestones



Zhou et al. Object Detection in 20 Years: A Survey. arXiv 2019.

Key Contributions of Faster R-CNN

1. State of art object detection model in terms of accuracy and speed

2. An end-to-end trained model that learns all parts of the pipeline, particularly with the addition of region proposals

Key Contributions of Faster R-CNN

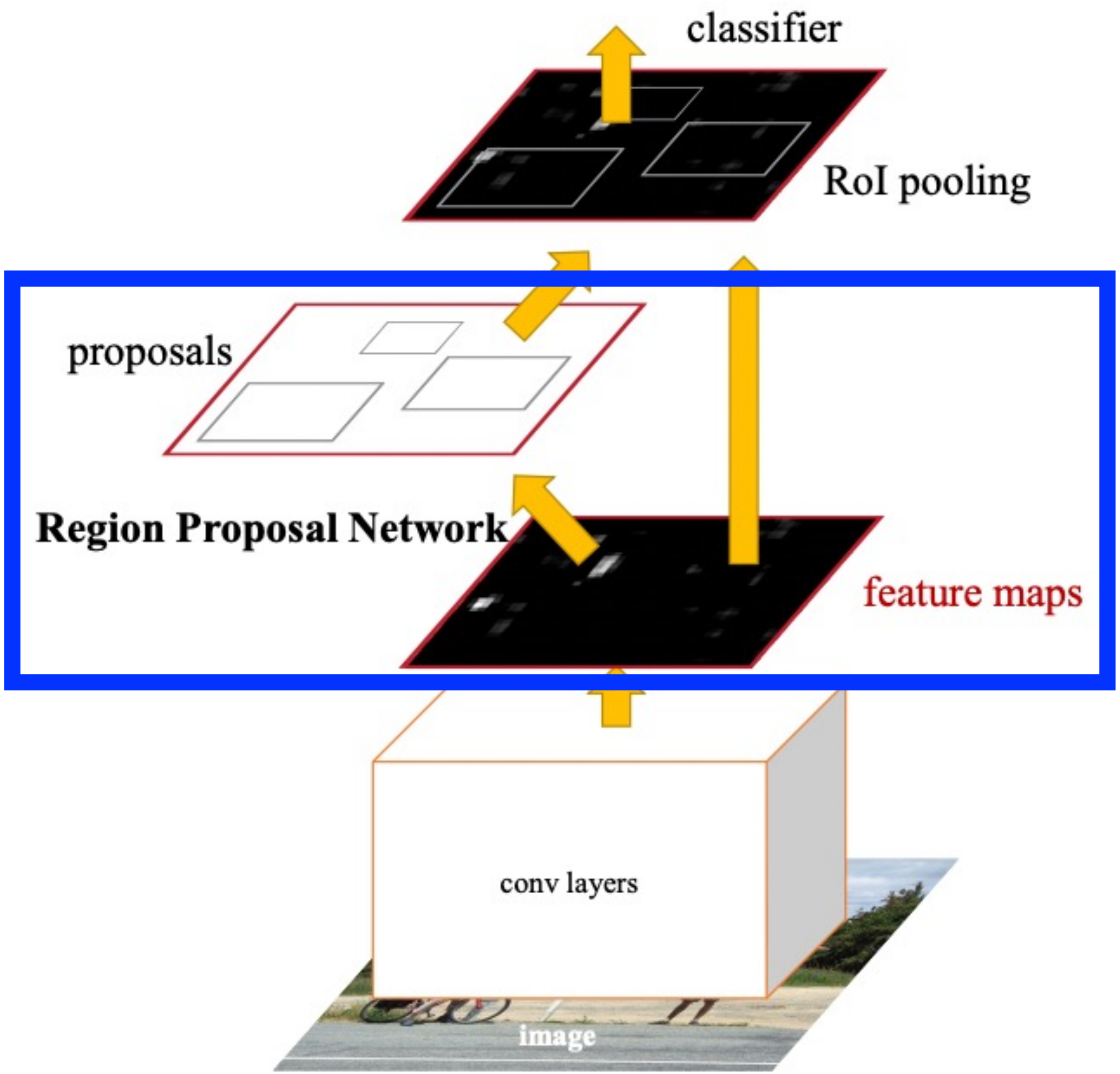
1. State of art object detection model in terms of accuracy and speed

2. An end-to-end trained model that learns all parts of the pipeline, particularly with the addition of region proposals

Architecture

Embeds a region proposal network in Fast R-CNN

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.

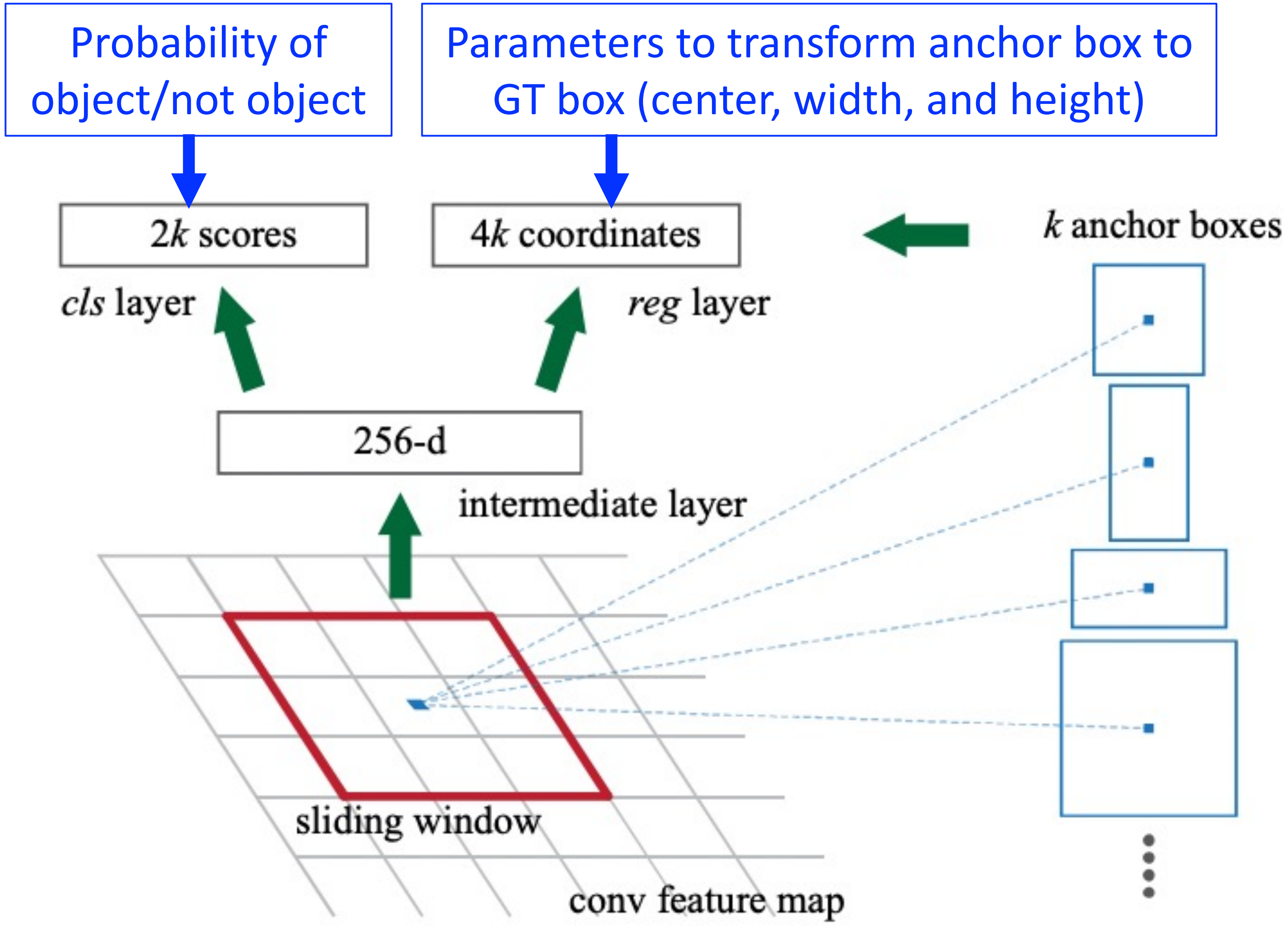
Architecture: Region Proposal Network

How many region proposals are considered for a convolutional feature map of size W X H (~2400)

- Note: post-processing is done to reduce number to ~2000 proposals

Based on convolution, so uses sliding window

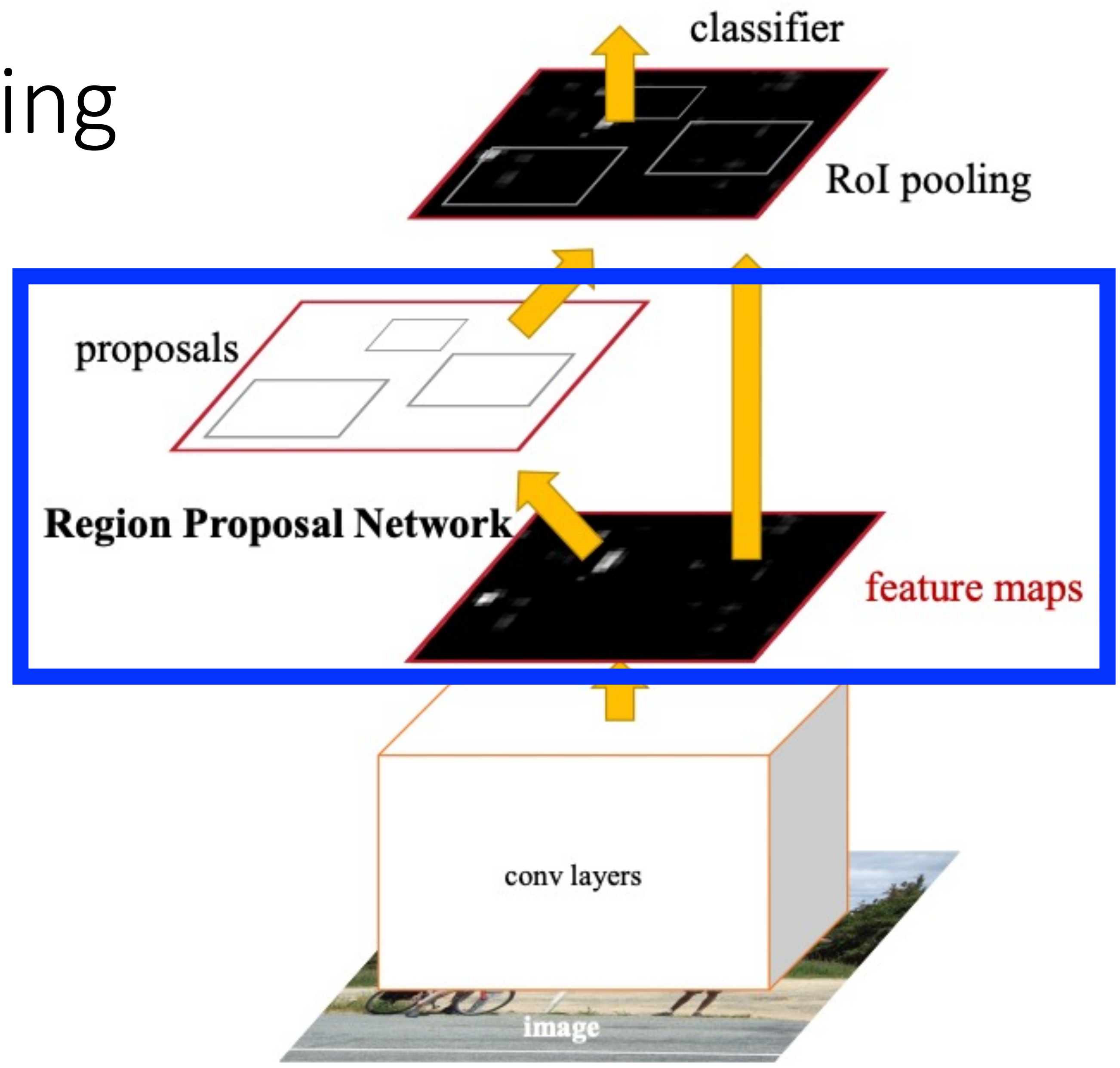
- At each sliding window position, region proposals are predicted with respect to an anchor point (i.e., center of sliding window position)
- At each anchor point, k = 9 anchors are used to represent 3 scales and 3 aspect ratios



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

Algorithm 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.

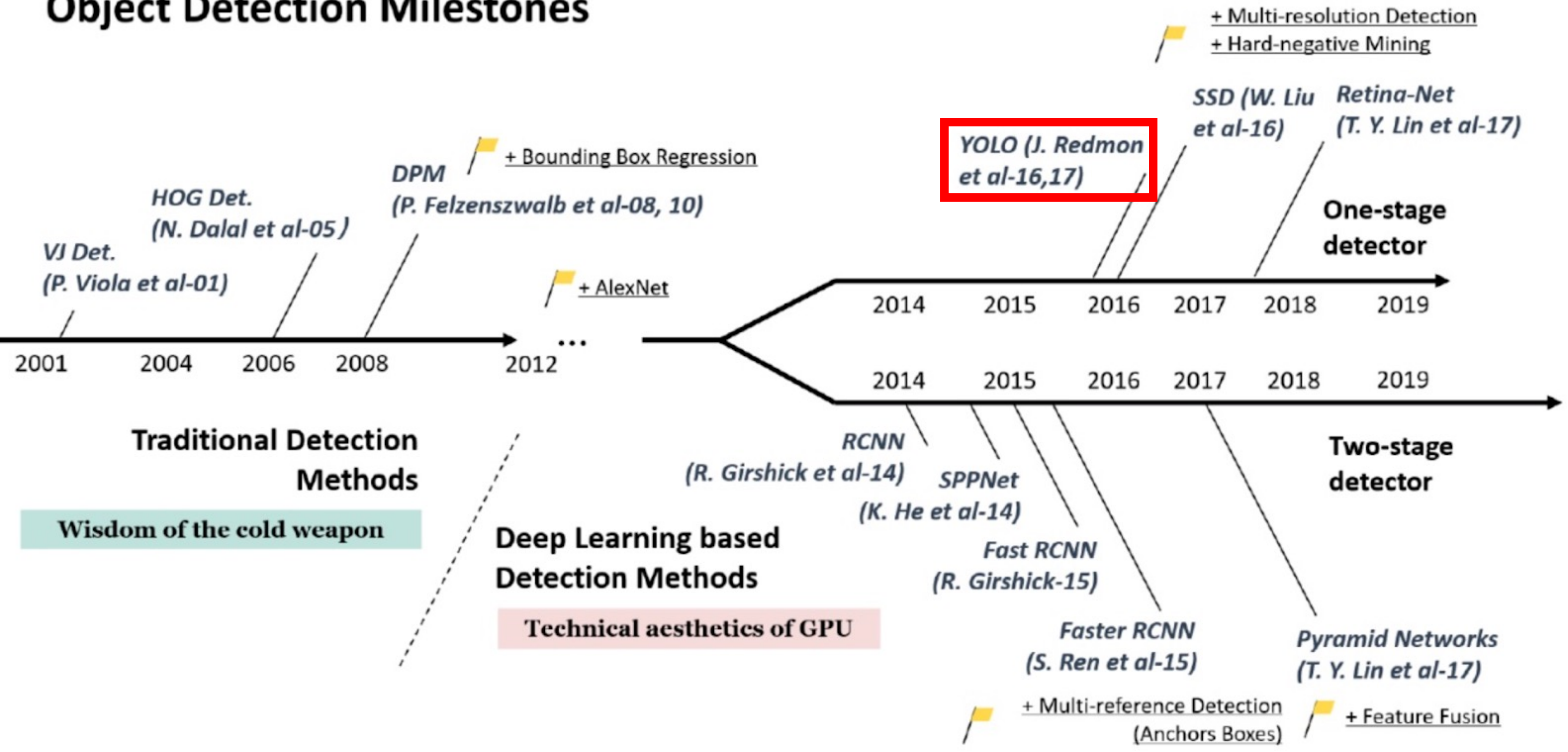
Limitations

- Still relatively slow; i.e., does not support real-time performance

Object Detection: Today's Topics

- Overview of object detection algorithms
- Baseline Model: R-CNN
- Fast R-CNN
- Faster R-CNN
- YOLO
- Discussion

Object Detection Milestones



Zhou et al. Object Detection in 20 Years: A Survey. arXiv 2019.

Why YOLO?

Named after the proposed technique: **Y**ou **O**nly **L**ook **O**nce

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. “You Only Look Once: Unified, Real-Time Object Detection.” CVPR 2016.

Key Contributions

- Most accurate *real-time* object detection system (i.e., 30+ frames per second), achieving twice the mean average precision of previous approach

- A simple CNN architecture that directly detects objects in an image and thus can consider the context when locating objects
  - i.e., a paradigm shift away from using classifiers to label image regions

- The single CNN learns a representation that simultaneously can detect a variety of objects and generalizes better than existing approaches

Key Contributions

- Most accurate *real-time* object detection system (i.e., 30+ frames per second), achieving twice the mean average precision of previous approach

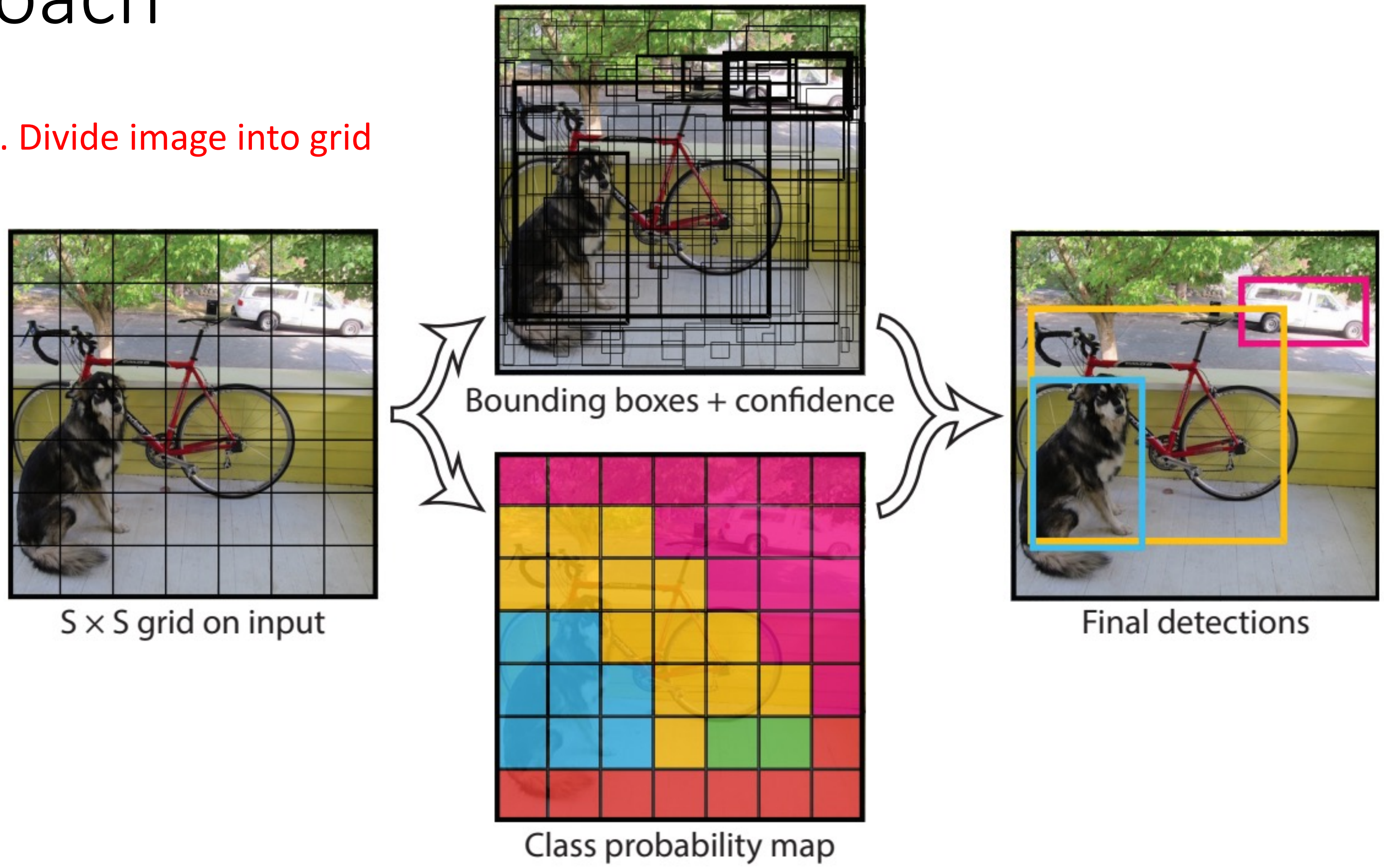
- A simple CNN architecture that directly detects objects in an image and thus can consider the context when locating objects
  - i.e., a paradigm shift away from using classifiers to label image regions

- The single CNN learns a representation that simultaneously can detect a variety of objects and generalizes better than existing approaches

Approach

2. For each grid cell, predict a probability distribution for class labels (assuming an object Is present) and **locate (potentially multiple) objects**

1. Divide image into grid



Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection.” CVPR 2016.

Approach: BB Prediction Per Grid Cell

Center of BB relative to grid cell

Width and height relative to image

$$y = (\boxed{p_c}, \boxed{b_x, b_y}, \boxed{b_h, b_w}$$

Confidence that an object is present and the predicted box's resemblance to GT (via IoU)

1. What should p_c equal if no object is present?
- 0

2. What should p_c equal if an object is present?
- IoU between predicted and ground truth boxes

Key Contributions

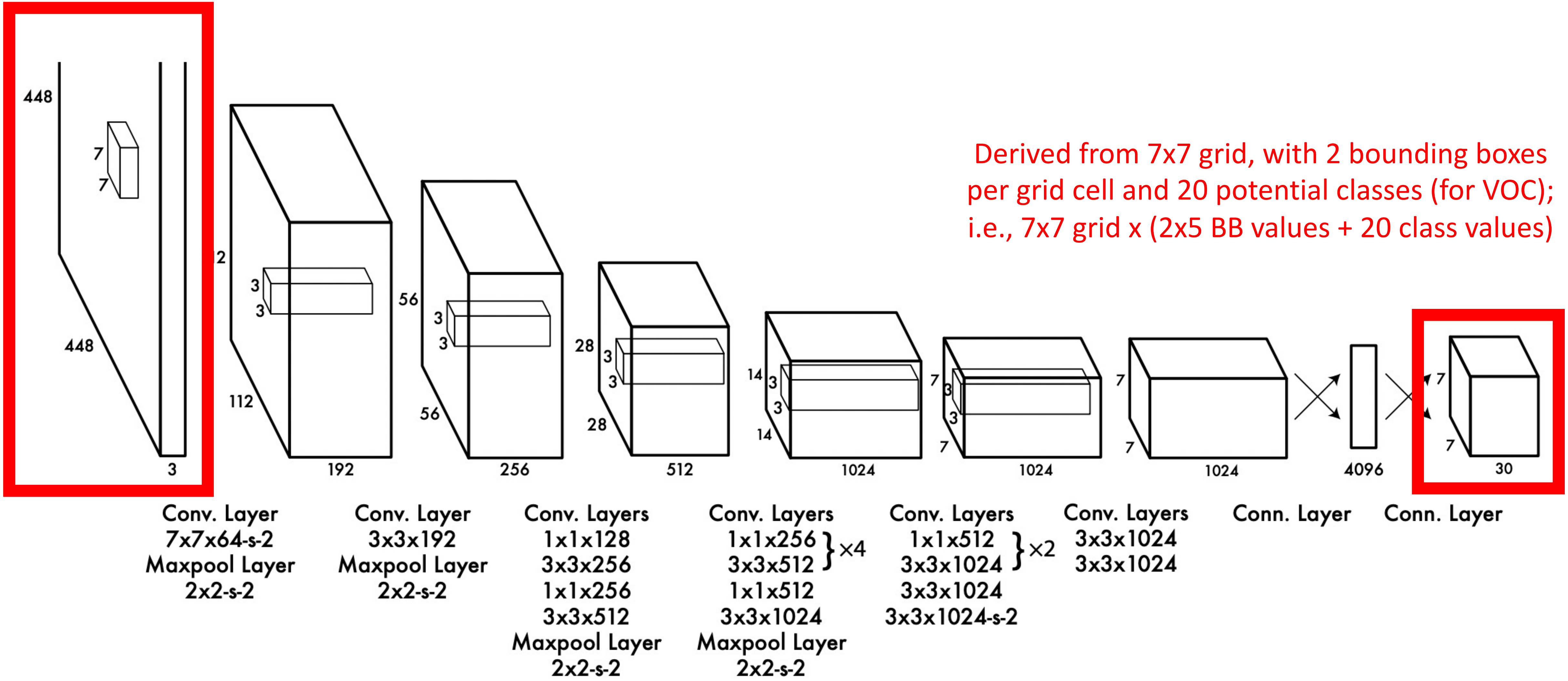
- Most accurate *real-time* object detection system (i.e., 30+ frames per second), achieving twice the mean average precision of previous approach
- A simple CNN architecture that directly detects objects in an image and thus can consider the context when locating objects
    - i.e., a paradigm shift away from using classifiers to label image regions
- The single CNN learns a representation that simultaneously can detect a variety of objects and generalizes better than existing approaches

Architecture

Input: RGB image resized to fixed input size

Output: 98 BB per image w/ class probabilities (i.e., 7x7 grid x 2 BB per grid cell = 98 BB)

Derived from 7x7 grid, with 2 bounding boxes per grid cell and 20 potential classes (for VOC); i.e., 7x7 grid x (2x5 BB values + 20 class values)



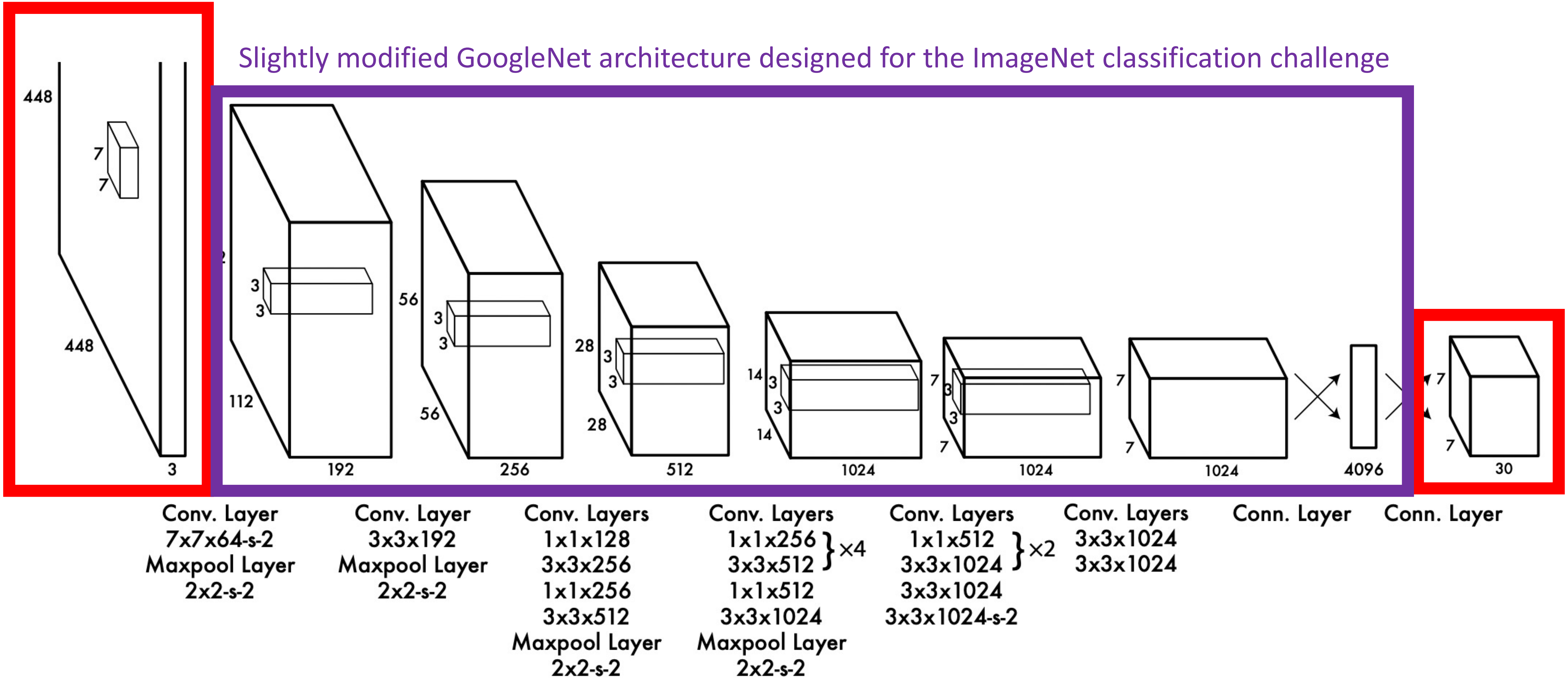
Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection.” CVPR 2016.

Architecture

Input: RGB image resized to fixed input size

Output: 98 BB per image w/ class probabilities
(i.e., 7x7 grid x 2 BB per grid cell = 98 BB)

Slightly modified GoogleNet architecture designed for the ImageNet classification challenge



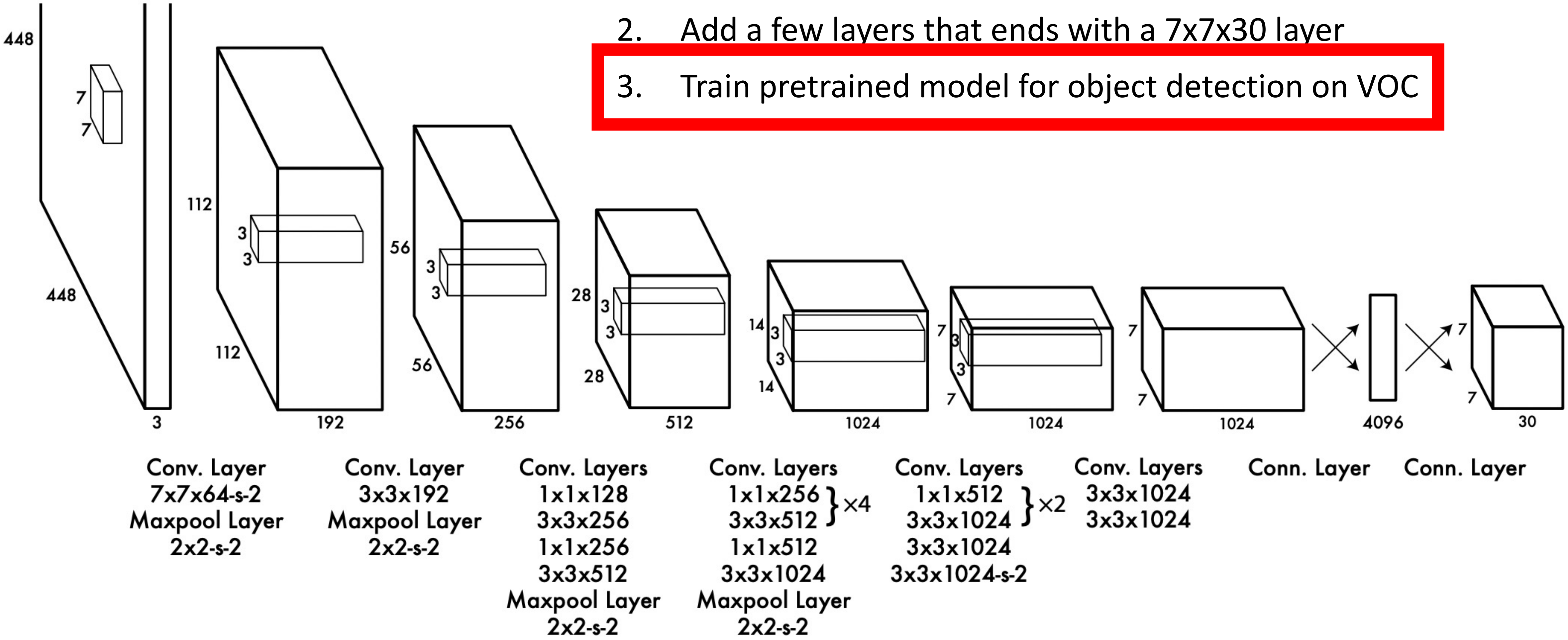
Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection." CVPR 2016.

Key Contributions

- Most accurate *real-time* object detection system (i.e., 30+ frames per second), achieving twice the mean average precision of previous approach
- A simple CNN architecture that directly detects objects in an image and thus can consider the context when locating objects
  - i.e., a paradigm shift away from using classifiers to label image regions
- The single CNN learns a representation that simultaneously can detect a variety of objects and generalizes better than existing approaches

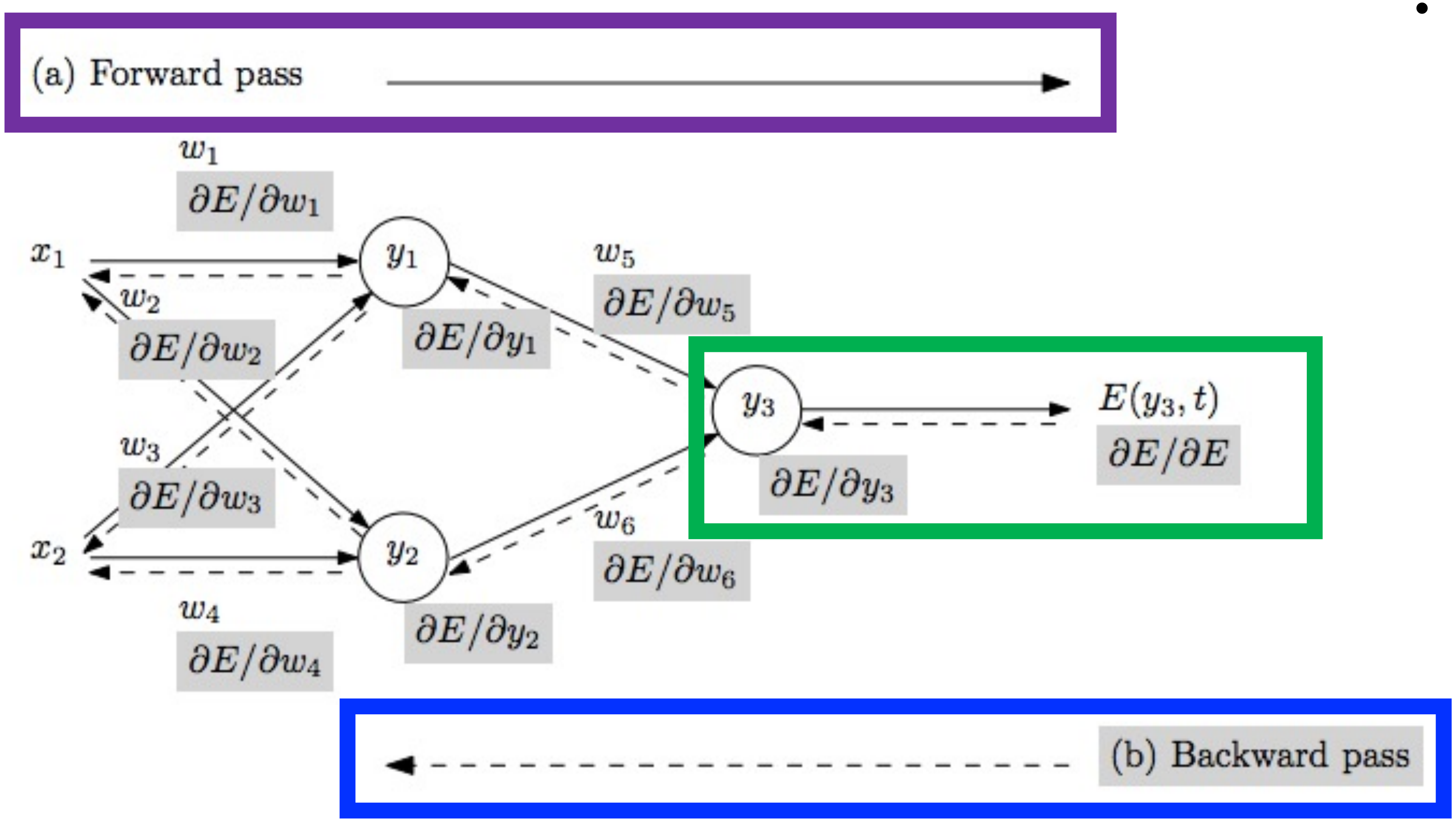
Algorithm Training: CNN

1. Pretrain first 20 convolutional layers for ImageNet classification
2. Add a few layers that ends with a 7x7x30 layer
3. Train pretrained model for object detection on VOC



Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection.” CVPR 2016.

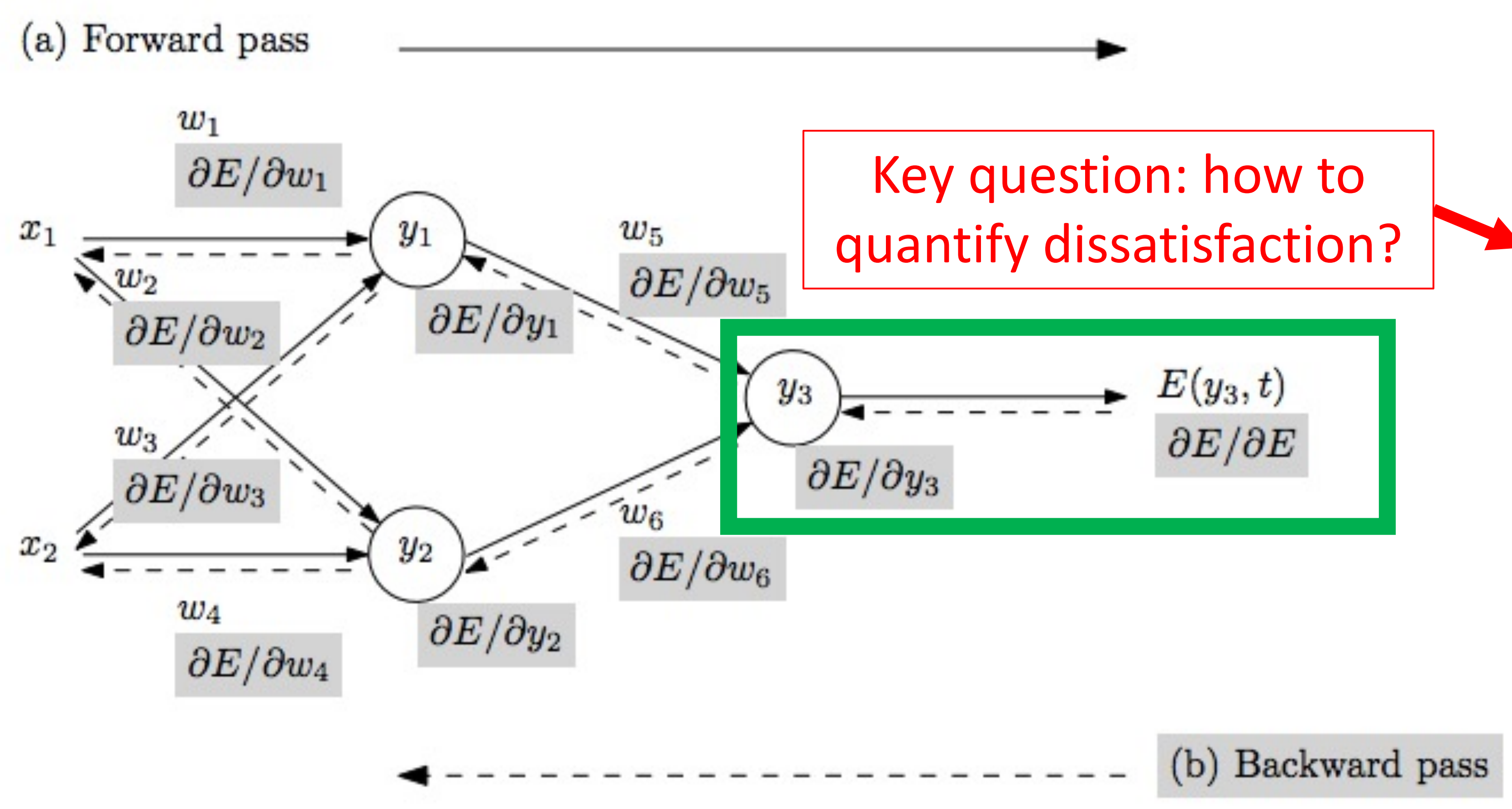
Algorithm Training: Recall How NNs Learn



- 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

Algorithm Training



Key question: how to quantify dissatisfaction?

- 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

Algorithm Training: Multi-Part Loss Function

Number of grid cells; what does YOLO use?

Number of bounding box predictors per cell; what does YOLO use?

Penalty only for most confident BB predictor from j predictors when an object is present

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \boxed{\mathbb{1}_{ij}^{\text{obj}}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

Penalizes imperfect positioning of object centers (for each BB)

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \boxed{\mathbb{1}_{ij}^{\text{obj}}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Penalizes imperfect widths and heights (for each BB)

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \boxed{\mathbb{1}_{ij}^{\text{obj}}} \left(C_i - \hat{C}_i \right)^2$$

Pushes the BB confidence score to match the IoU between the prediction and GT

$$+ \lambda_{\textrm{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Punishes the network for predicting an object is present when no object is in the grid cell (i.e., pushes confidence to 0)

Penalizes incorrect classifications (for each grid cell)

Penalty for classification occurs ONLY if an object is actually present in the grid cell

$$+ \sum_{i=0}^{S^2} \boxed{\mathbb{1}_{i}^{\text{obj}}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Algorithm Training: Loss Function

$$\boxed{\lambda_{\textbf{coord}}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \boxed{\lambda_{\textbf{coord}}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\boxed{\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)}^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \boxed{\lambda_{\textbf{noobj}}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Set coefficient of 5 to prioritize loss from BB shape predictions over misclassification predictions

Square root used to weight a small deviation in large boxes less than for small boxes

Set coefficient of 0.5 to reduce loss from boxes that don't contain objects to mitigate learning to push the confidence to 0 since most cells don't contain objects

Object Detection: Today’s Topics

- Problem
- Applications
- Datasets
- Evaluation metric
- Computer vision models


The End