

Hochschule für Technik
und Wirtschaft Berlin

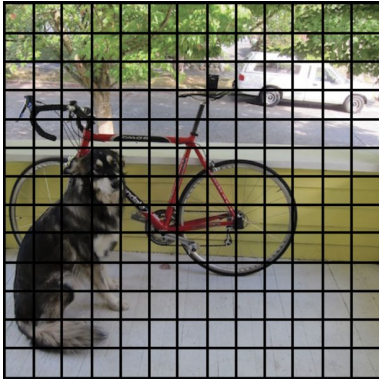
University of Applied Sciences

Region-based CNN (R-CNN) - Tuan Thanh Tran, Max Hager

Object detection

How does object detection work?

1

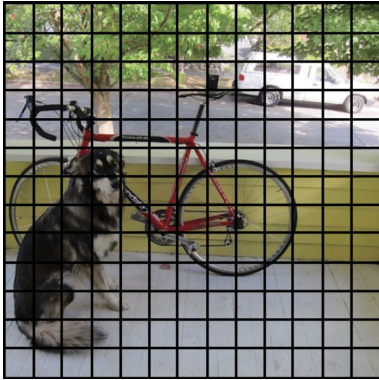


large set of bounding
boxes

https://www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-tensorflow-api/?utm_source=blog&utm_medium=Non_Max_Suppression_for_Object_Detection

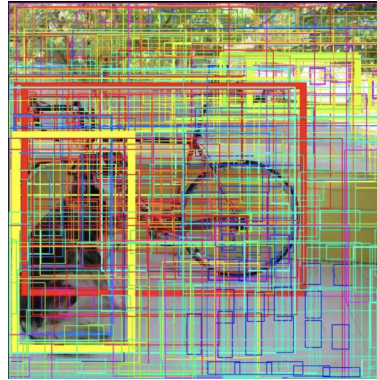
How does object detection work?

1



large set of bounding
boxes

2

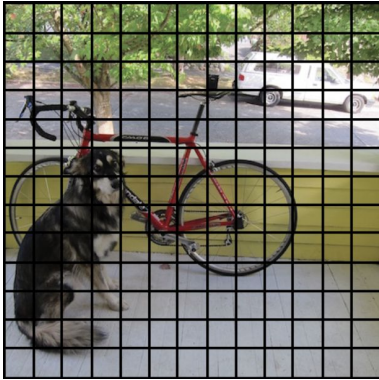


classification

https://www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-ten-sorflow-api/?utm_source=blog&utm_medium=Non_Max_Suppression_for_Object_Detection

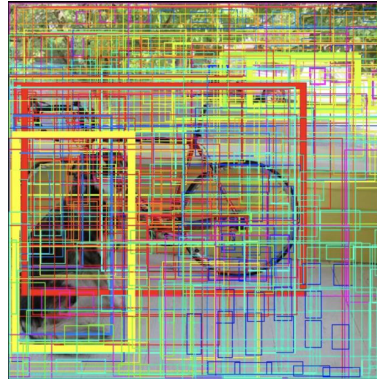
How does object detection work?

1



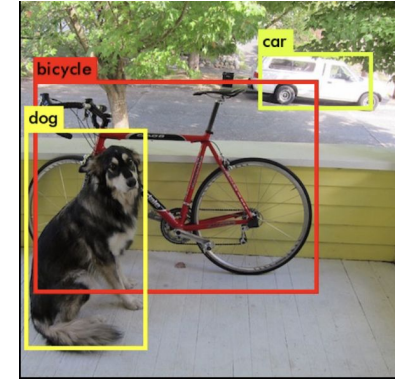
large set of bounding
boxes

2



classification

3

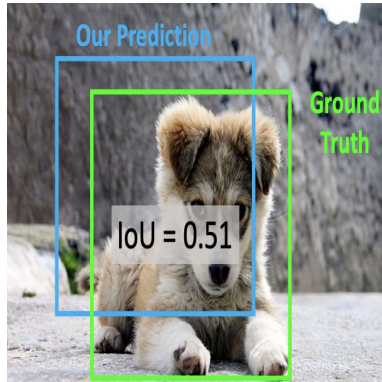



overlapping boxes
are combined into
single bounding box

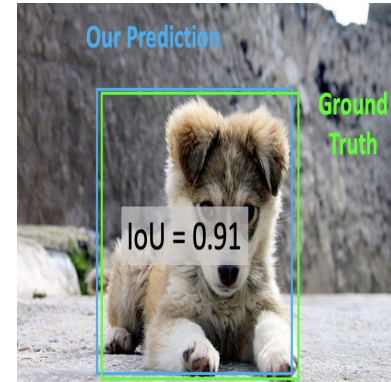
https://www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-ten-sorflow-api/?utm_source=blog&utm_medium=Non_Max_Suppression_for_Object_Detection

How to select bounding boxes?

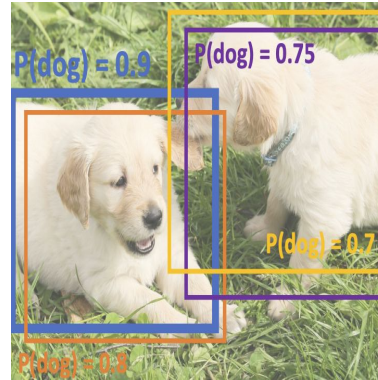
Comparing Boxes: Intersection over Union (IoU)



$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


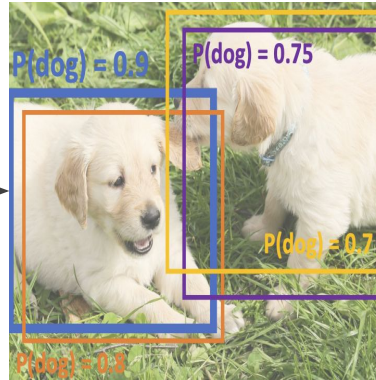


Overlapping Boxes: Non-Max Suppression (NMS)



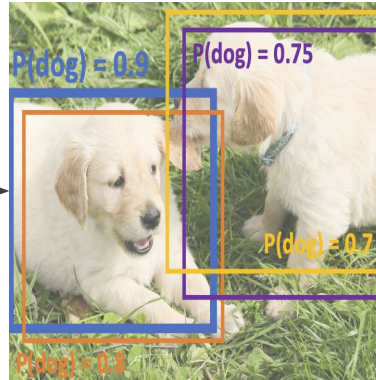
Overlapping Boxes: Non-Max Suppression (NMS)

1. Select next higher scoring box



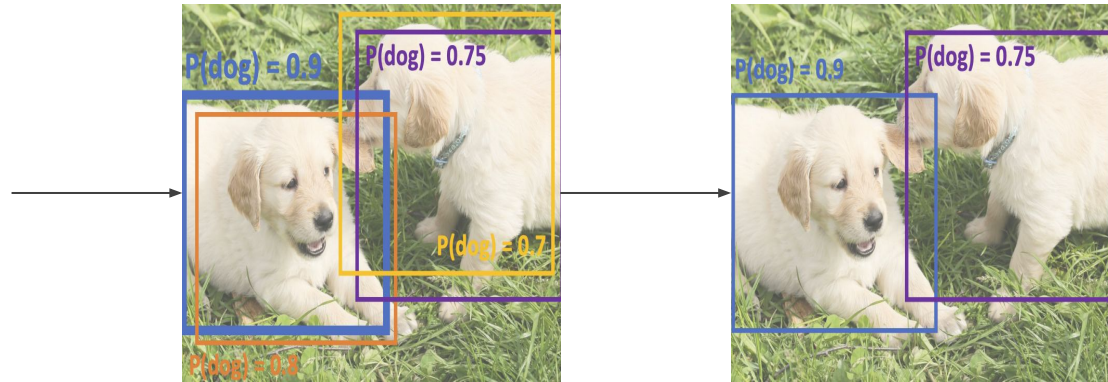
Overlapping Boxes: Non-Max Suppression (NMS)

1. Select next higher scoring box
2. Eliminate lower-scoring boxes with $\text{IoU} > \text{threshold}$



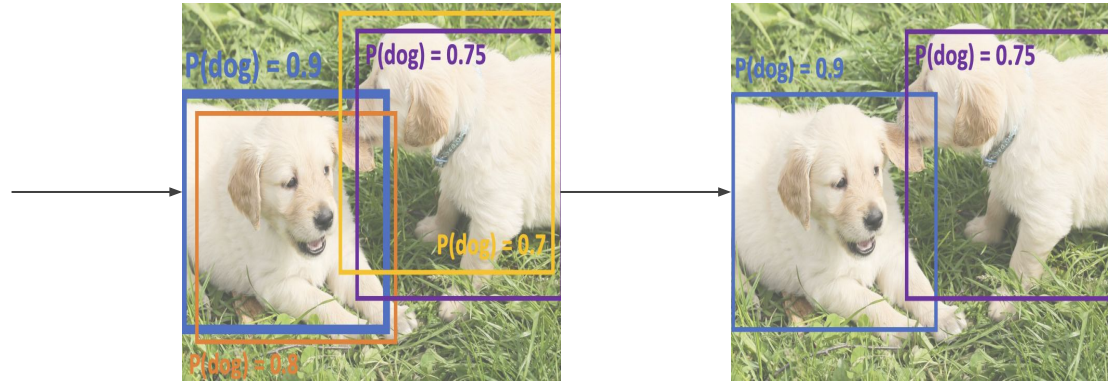
Overlapping Boxes: Non-Max Suppression (NMS)

1. Select next higher scoring box
2. Eliminate lower-scoring boxes with $\text{IoU} > \text{threshold}$

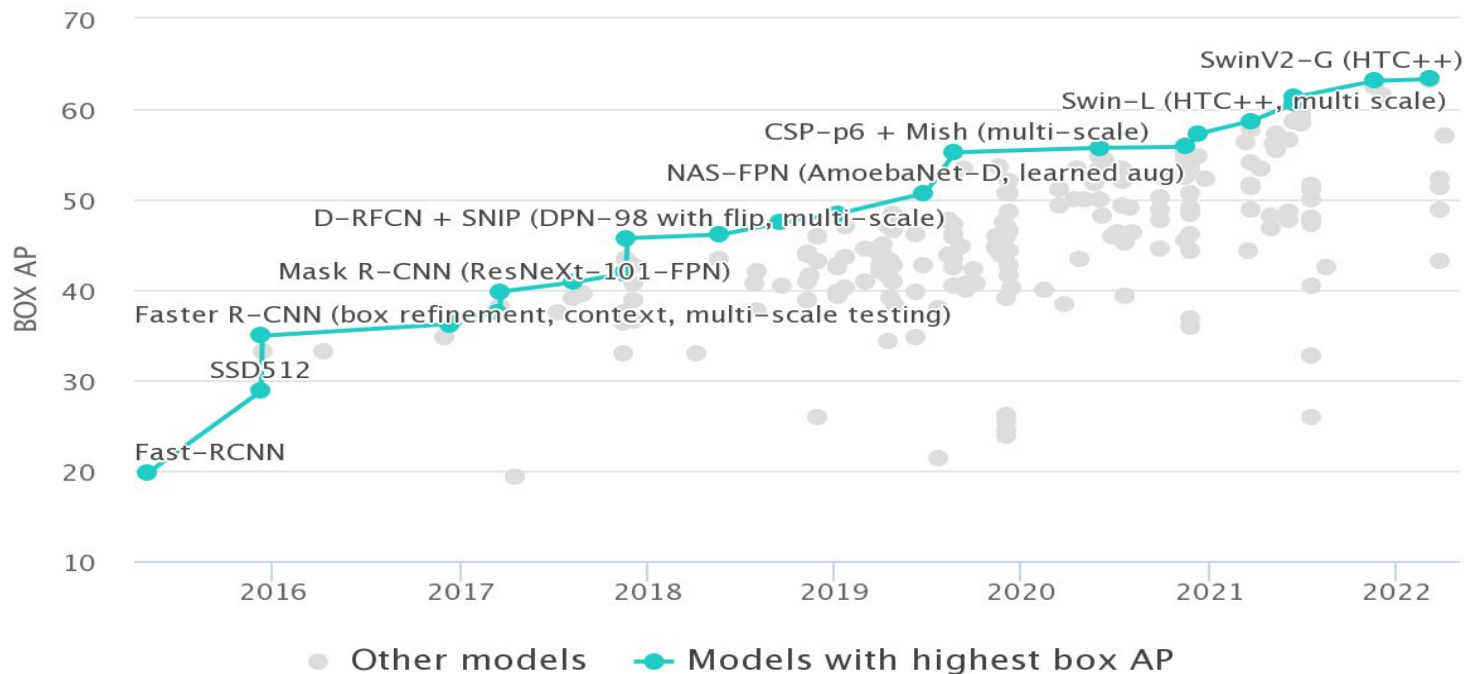


Overlapping Boxes: Non-Max Suppression (NMS)

1. Select next higher scoring box
2. Eliminate lower-scoring boxes with $\text{IoU} > \text{threshold}$
3. If any boxes remain, go to 1



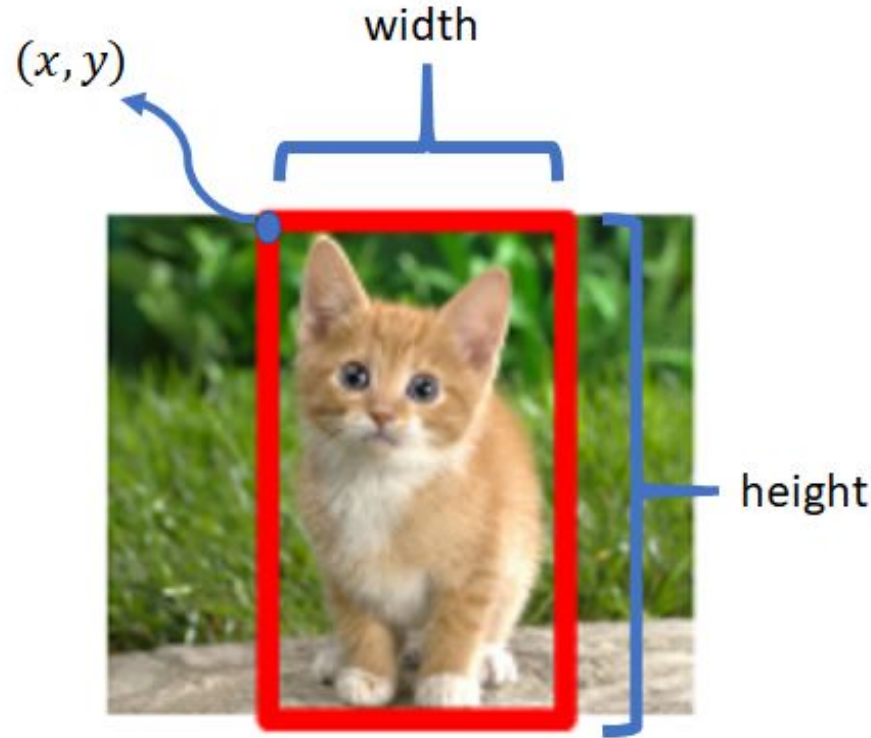
Overviews of famous models



<https://paperswithcode.com/sota/object-detection-on-coco>

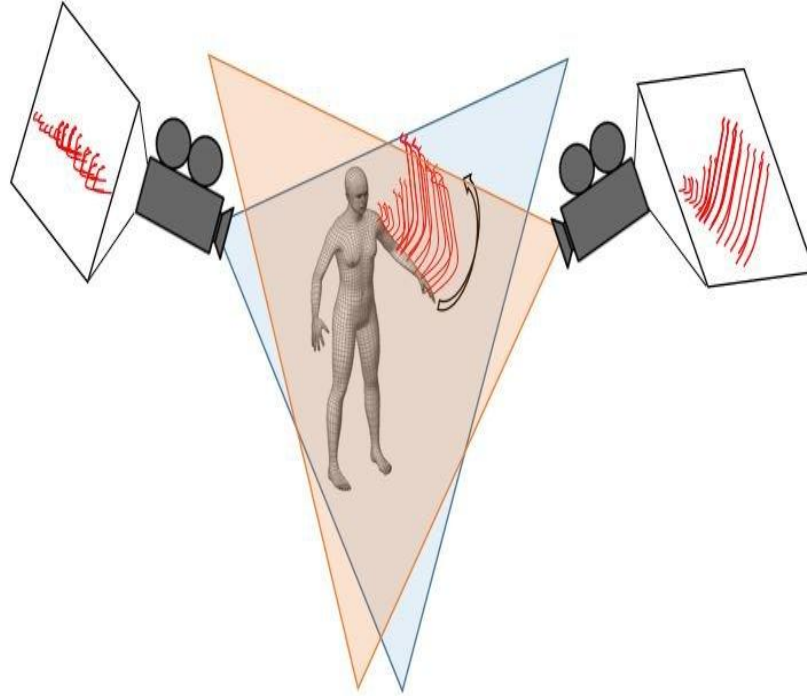
Challenges

Object localisation



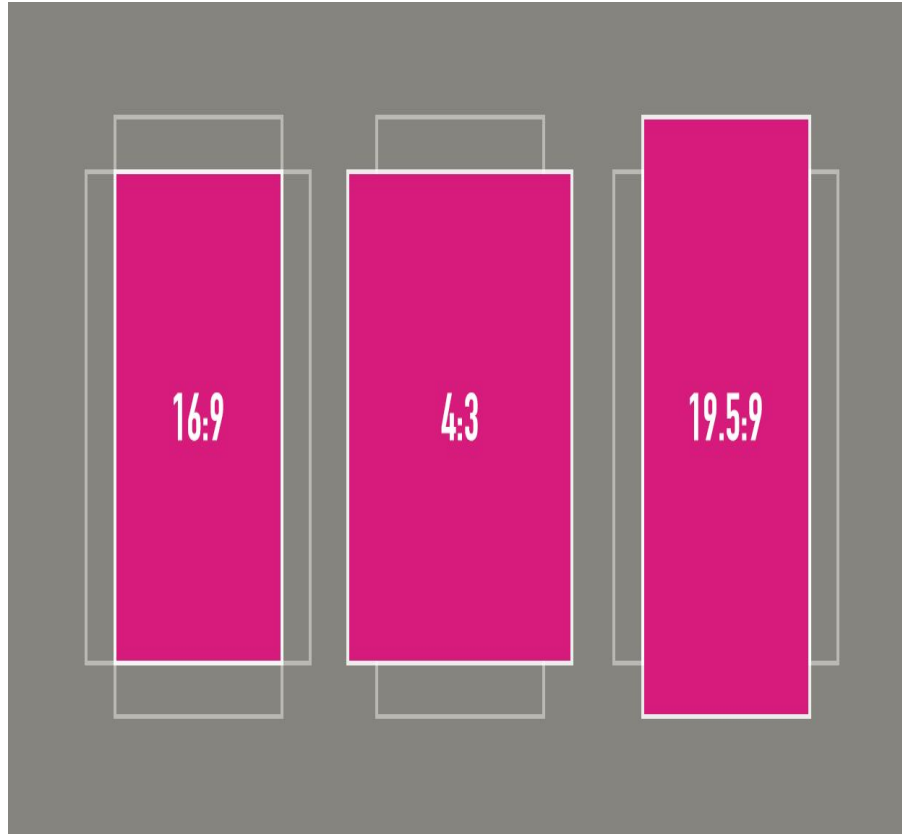
<https://medium.com/analytics-vidhya/object-localization-using-keras-d78d6810d0be>

Viewpoint variation



https://www.researchgate.net/figure/Illustration-of-the-issue-of-viewpoint-variation-in-the-context-of-action-recognition_fig1_338913739

Multiple aspect ratios and spatial sizes



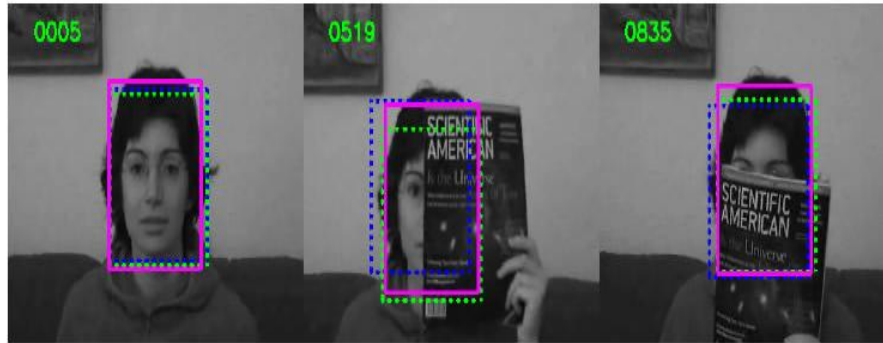
<https://medium.com/the-space-ape-games-experience/aspect-ratio-scaling-mobile-and-tablets-d574ab20a943>

Deformation



<https://www.exposit.com/blog/computer-vision-object-detection-challenges-faced/>

Occlusion



(a) Occluded Face 1



(c) Person

<https://stackoverflow.com/questions/2764238/image-processing-what-are-occlusions>

Lighting



https://www.researchgate.net/figure/An-example-of-low-illumination-objects-detection-Our-detector-have-achieved-amazing_fig1_342757964

Cluttered or textured background



<https://becominghuman.ai/computer-vision-object-detection-challenges-failed-9a927f9c5623>

Intra class variation



https://www.researchgate.net/figure/shows-an-example-of-Intra-Class-variation-on-the-Chair-class-But-how-will-a-machine_fig1_325311506

Limited data



<https://stats.stackexchange.com/questions/233512/how-to-get-the-data-set-size-required-for-neural-network-training>

RCNN

How does RCNN work?

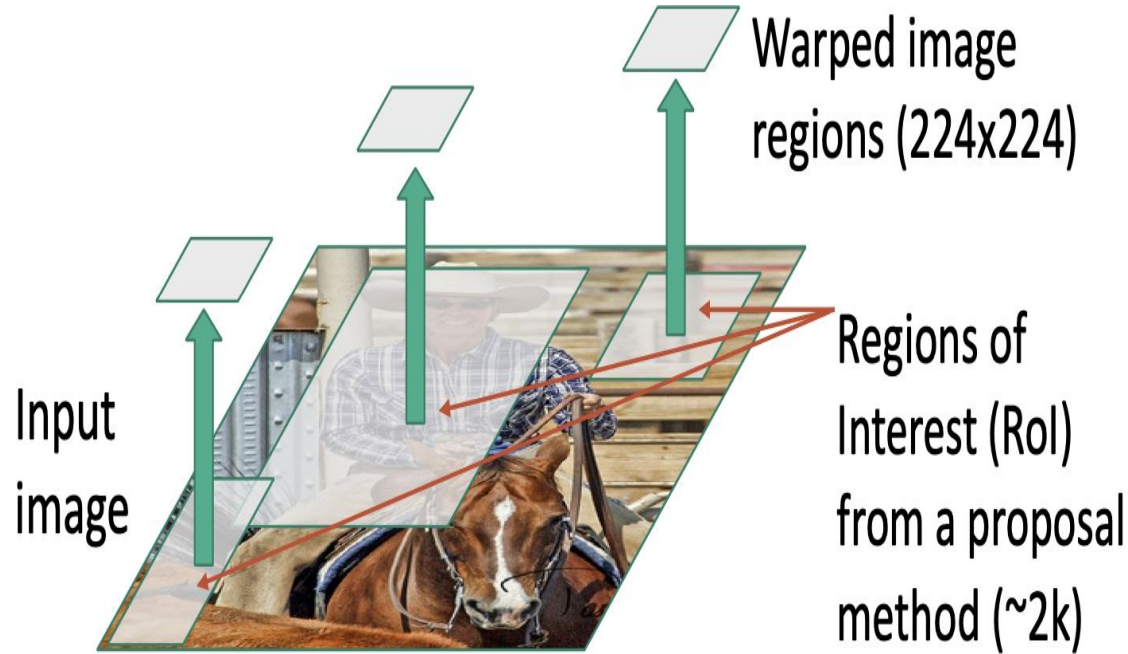
Input
image



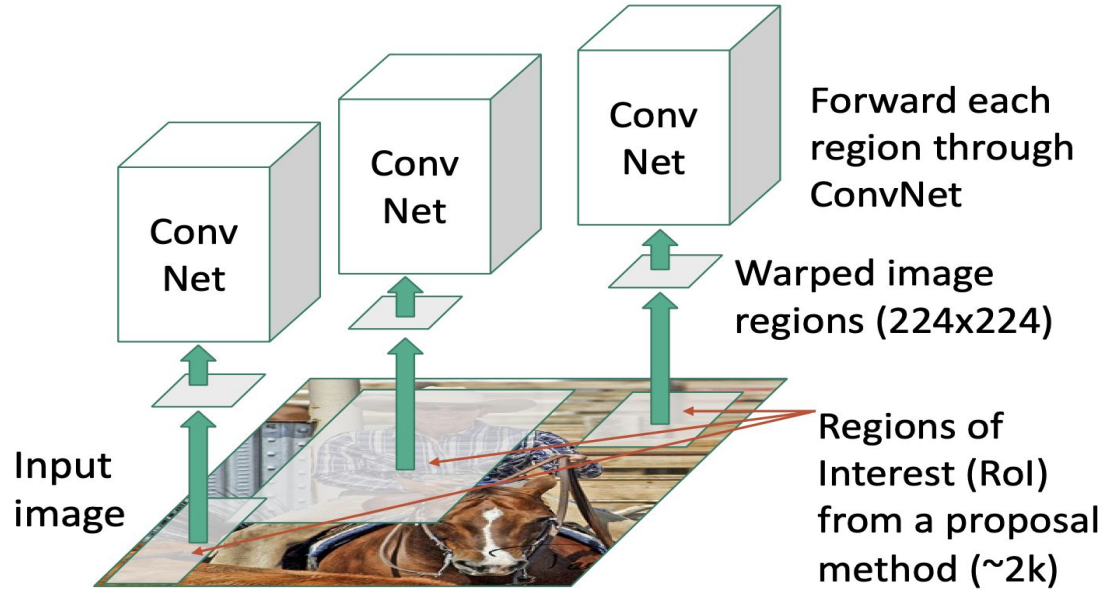
How does RCNN work?



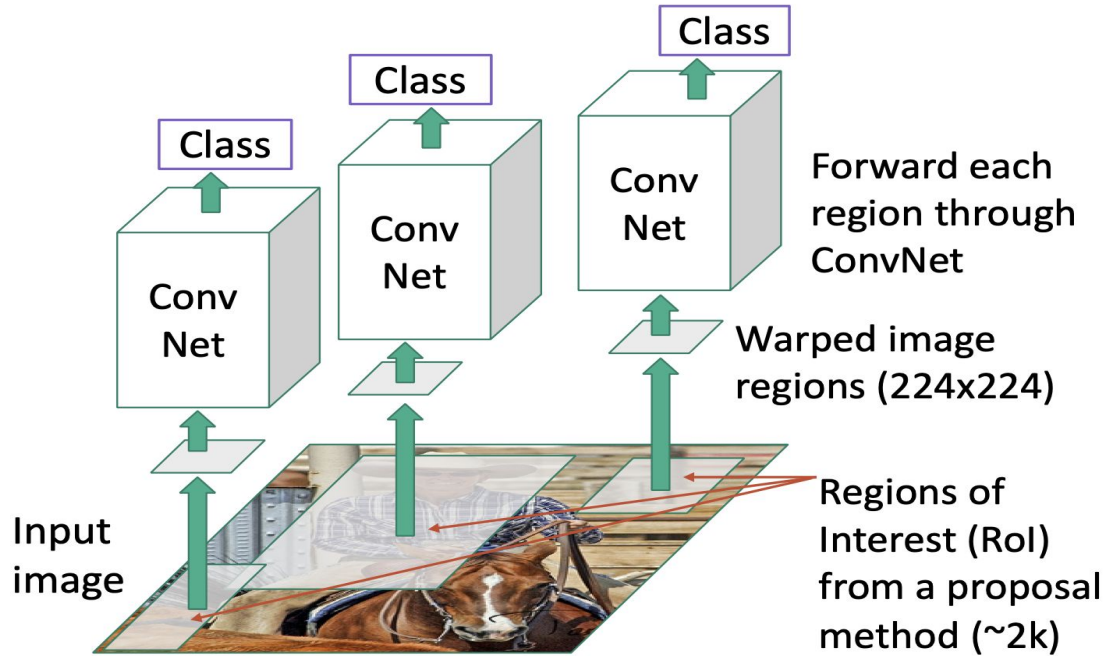
How does RCNN work?



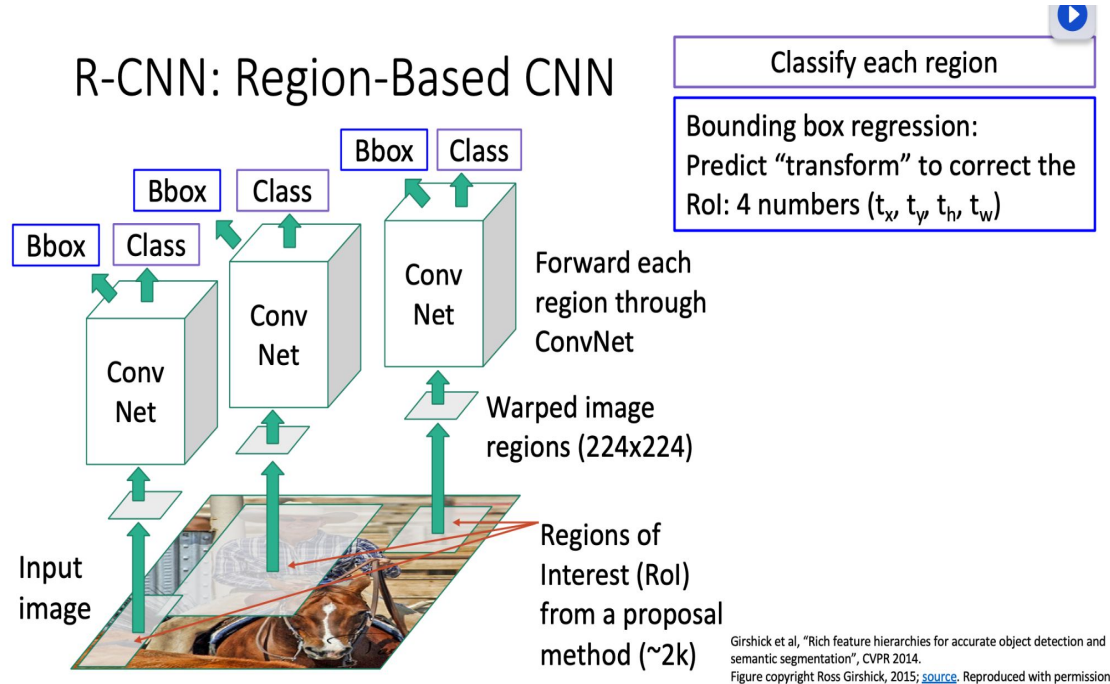
How does RCNN work?



How does RCNN work?



How does RCNN work?



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

Challenges

- Selective search no learning algorithm → sometimes result in bad region proposal generation for object detection
- Approx 2000 proposals → long training time
- Testing image with bounding box regressor takes 50 sec → no real time object detection

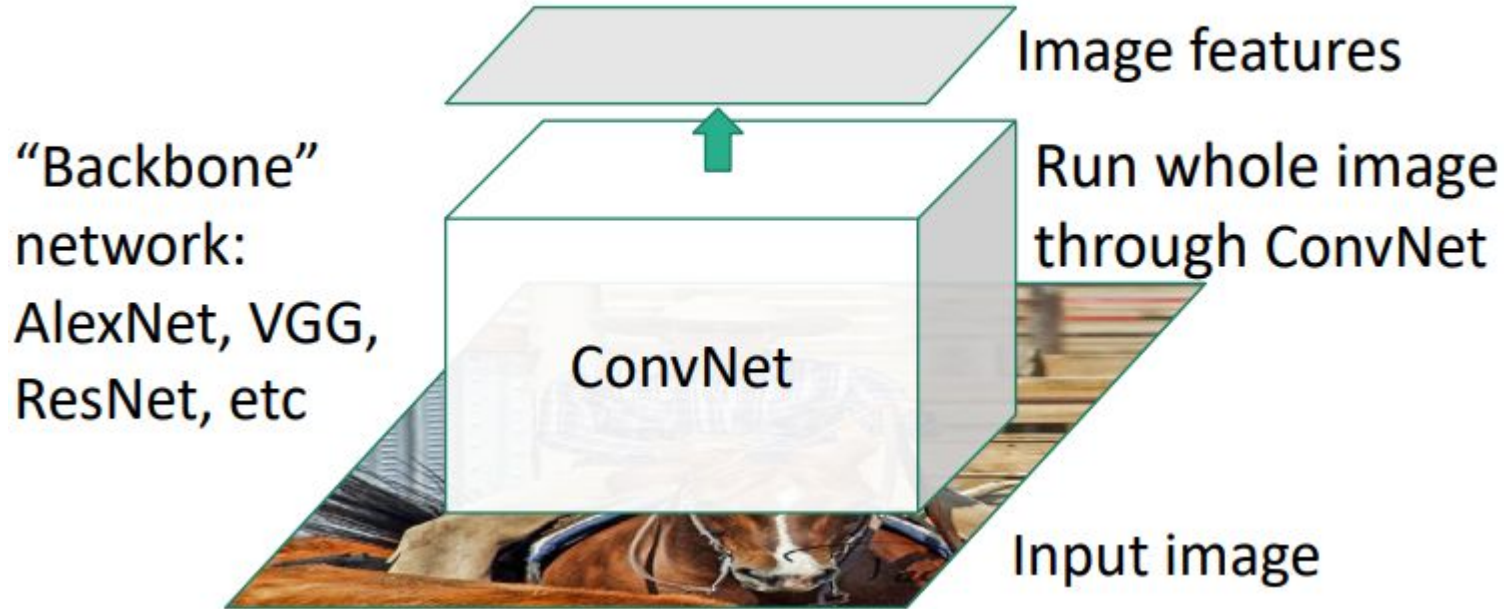
Fast RCNN

How does Fast-RCNN work?



Input image

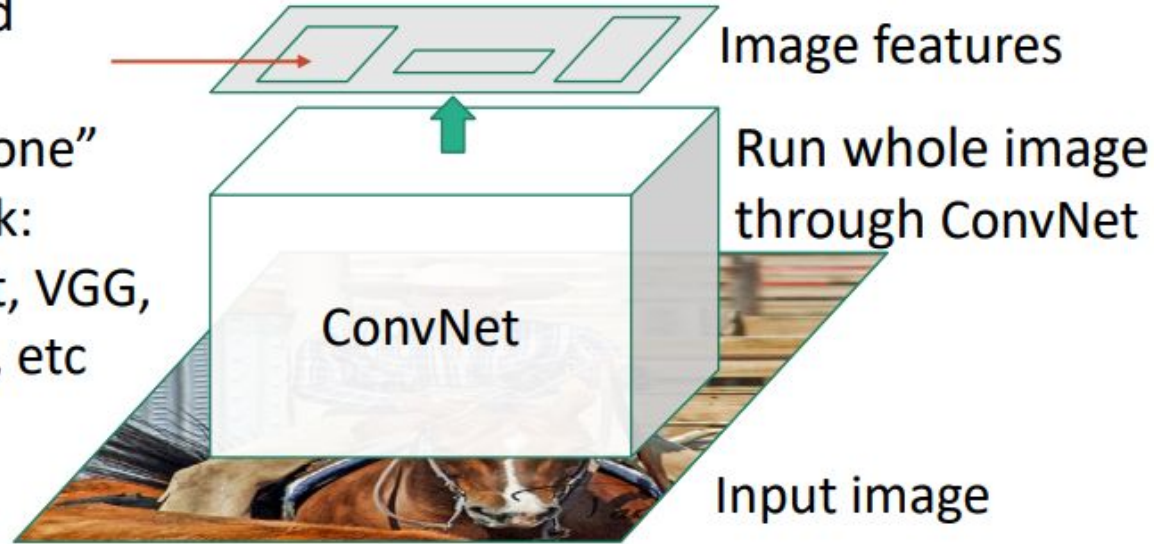
How does Fast-RCNN work?



How does Fast-RCNN work?

Regions of
Interest (Rols)
from a proposal
method

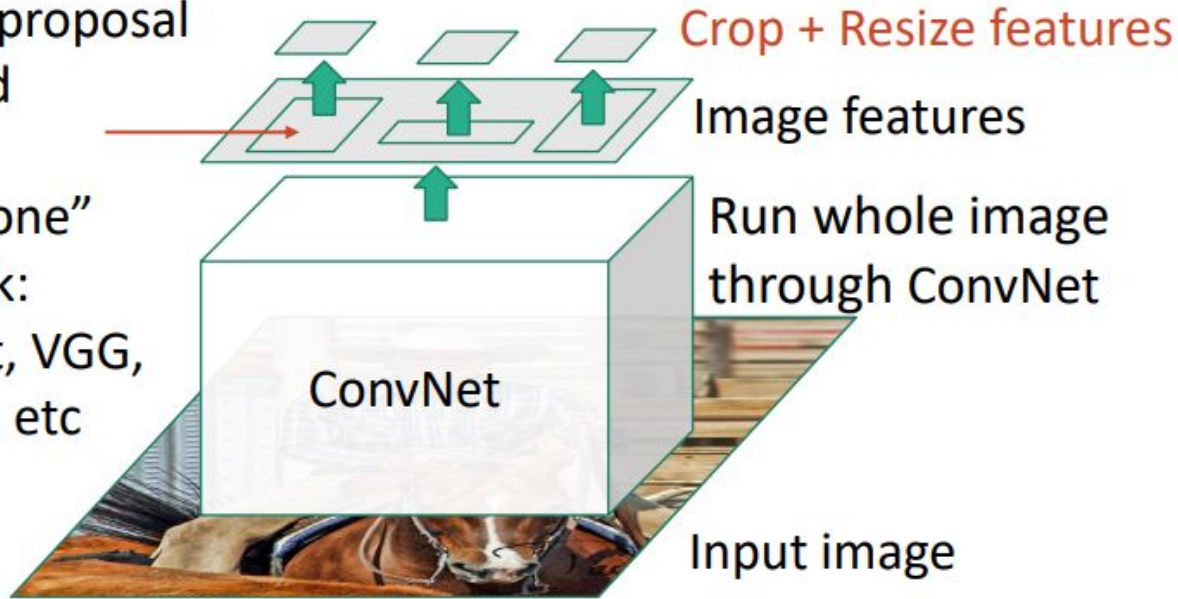
“Backbone”
network:
AlexNet, VGG,
ResNet, etc



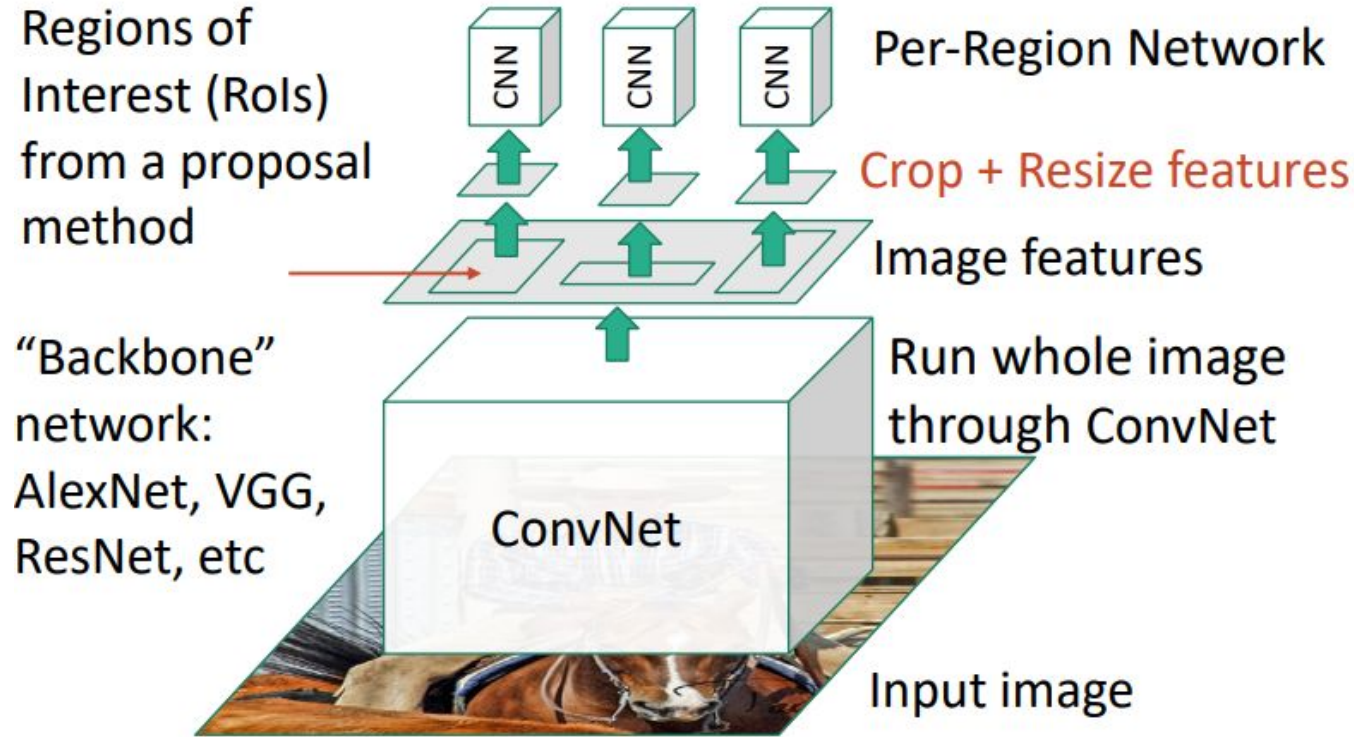
How does Fast-RCNN work?

Regions of
Interest (RoIs)
from a proposal
method

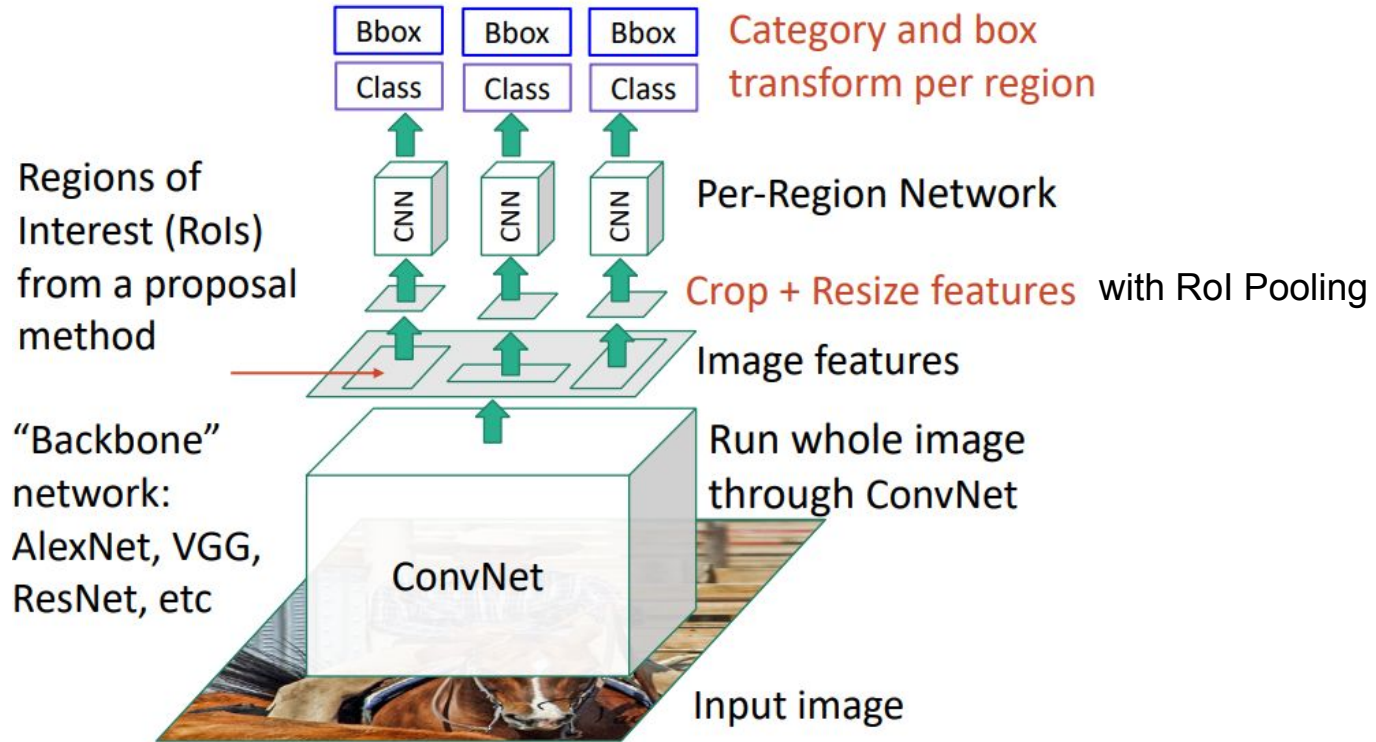
“Backbone”
network:
AlexNet, VGG,
ResNet, etc



How does Fast-RCNN work?



How does Fast-RCNN work?



Multi-task loss for training

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v)$$

$$L_{\text{cls}}(p, u) = -\log(p_u)$$

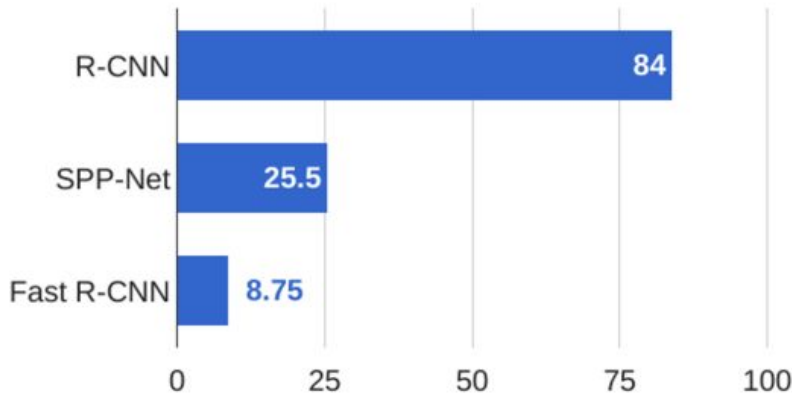
$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L1}(t_i^u - v_i)$$

For each region of interest (RoI):

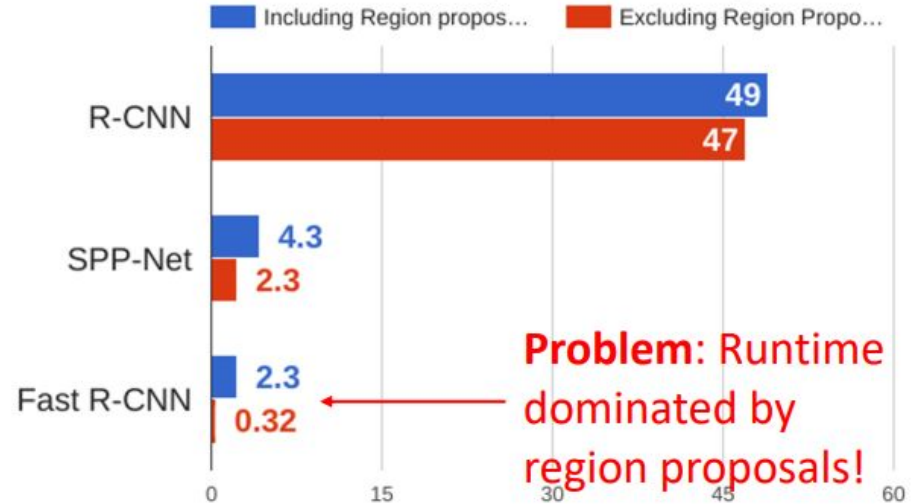
- u : ground-truth class
- $p = (p_0, p_1, \dots, p_K)$: probability distribution (per RoI) over $K + 1$ classes
- $t^u = (t_x^u, t_y^u, t_w^u, t_h^u)$: predicted bounding-box regression offsets for class u
- $v = (v_x, v_y, v_w, v_h)$: ground-truth bounding-box regression offsets
- $[u \geq 1]$: The Iverson bracket indicator function evaluates to 1 when $u \geq 1$ and 0 otherwise (Background class: $u = 0$)
- λ : Hyperparameter controls the balance between the two task losses.

Challenges

Training time (Hours)



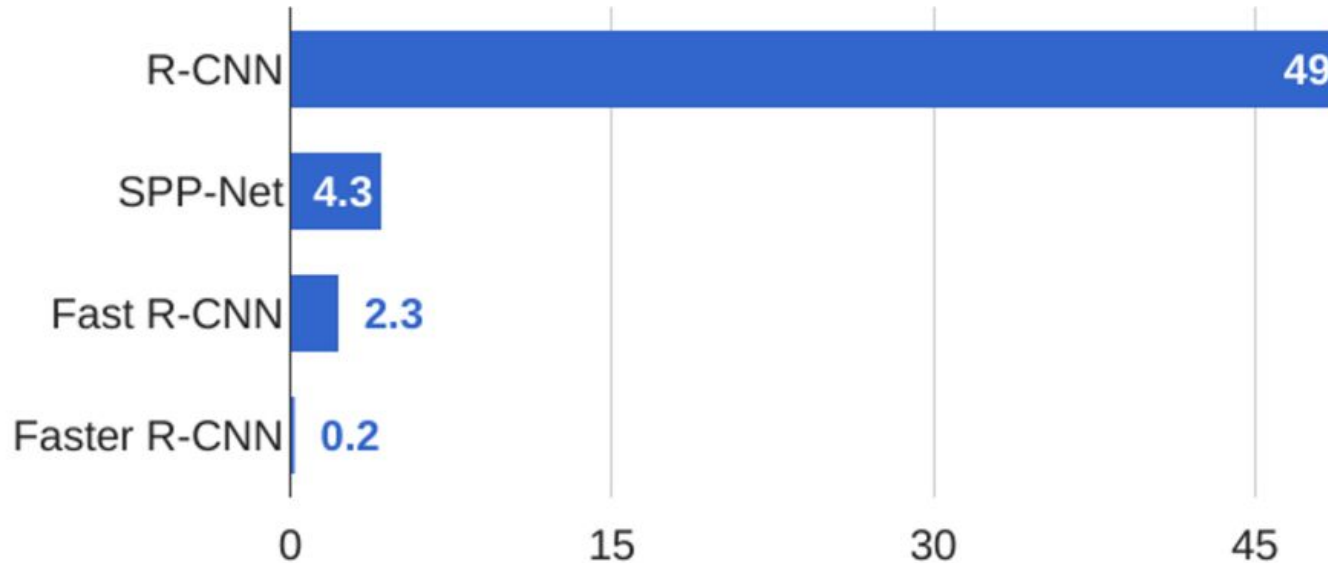
Test time (seconds)



Faster RCNN

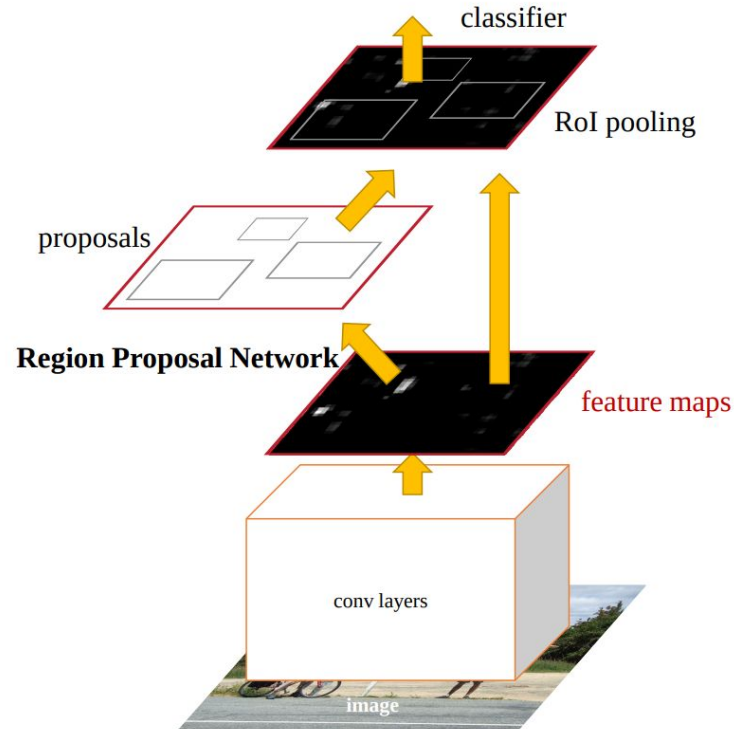
Faster-RCNN

R-CNN Test-Time Speed

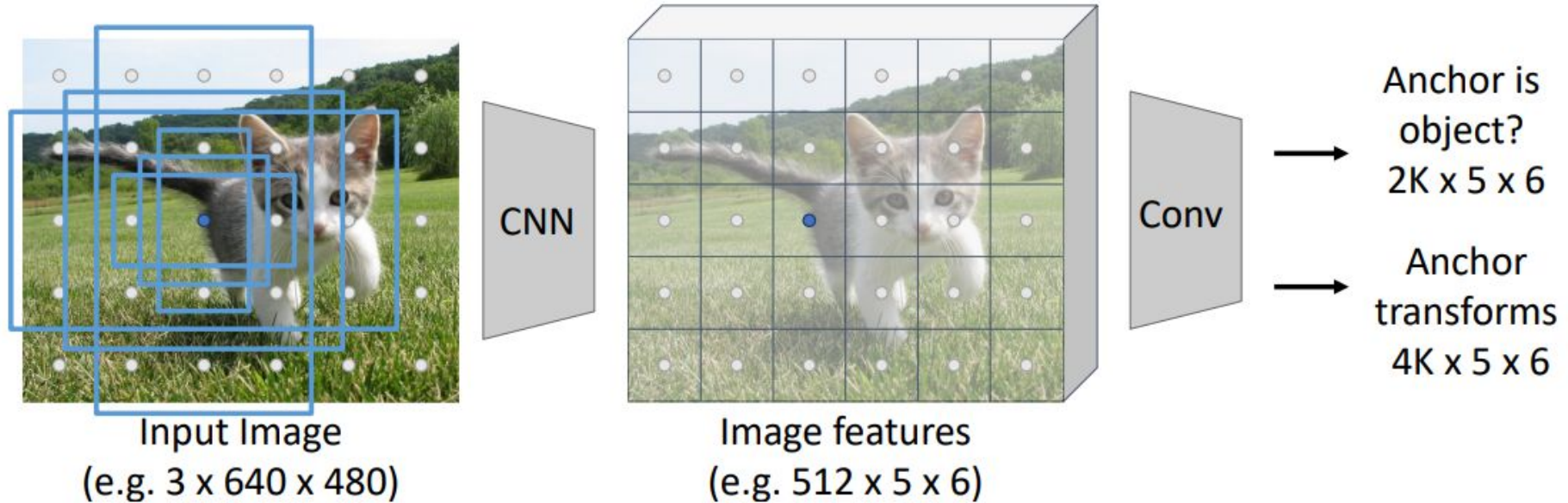


How does Faster-RCNN work?

- Same as Fast R-CNN
- Insert Region Proposal Network (RPN) to predict proposals from features

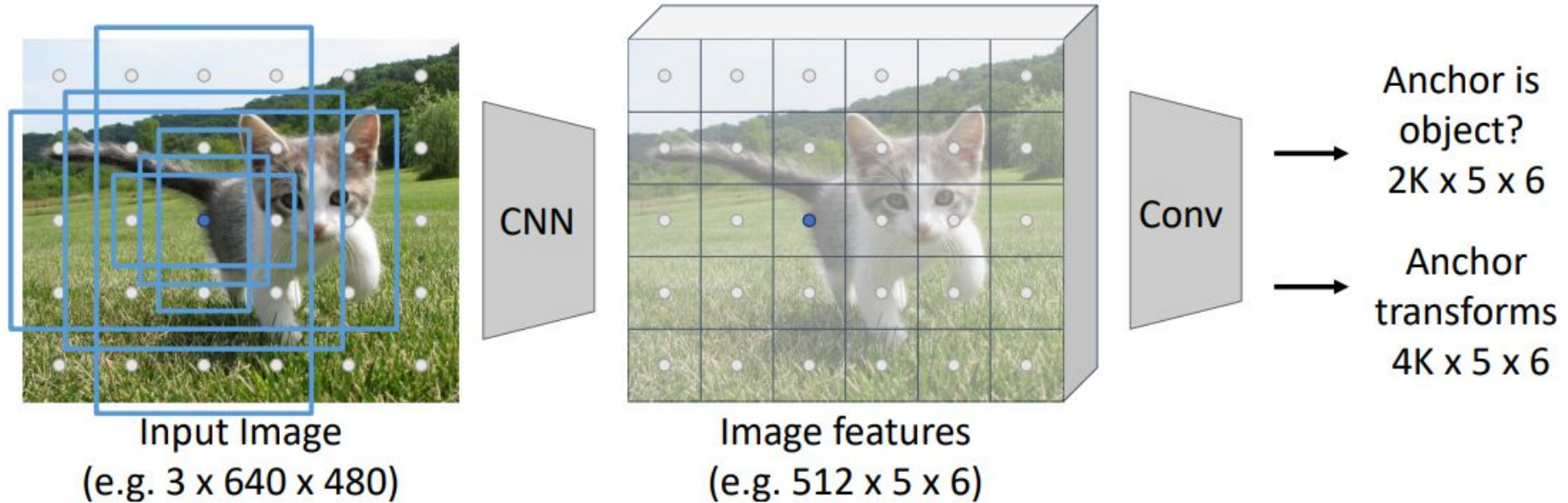


How does RPN work?



- Each feature corresponds to a point in the input
- K different anchors with different size and scale per point

How does RPN work?



- Sort all $K \times 5 \times 6$ boxes by their object score
- Take top n anchors as our region proposals

Joint multi-task loss for end-to-end training

$$L_{\text{Faster-RCNN}} = L_{\text{RPN}} + L_{\text{Fast-RCNN}}$$

$$L_{\text{RPN}} \text{ same as } L_{\text{Fast-RCNN}}$$

$$L_{\text{RPN}}(\{p_i\}, \{t_i\}) = \sum_i L_{\text{cls}}(p_i, p_i^*)/N_{\text{cls}} + \lambda \sum_i p_i^* L_{\text{reg}}(t_i, t_i^*)/N_{\text{reg}}$$

- i : the index of an anchor in a mini-batch
- p_i : the predicted probability of anchor i being an object
- p_i^* : the ground-truth label (1 if the anchor is positive, 0 if the anchor is negative)
- t_i : a vector representing the 4 parameterized coordinates of the predicted bounding box
- t_i^* : the ground-truth box associated with a positive anchor
- λ : hyperparameter controls the balance between the two task losses.
- N_{cls} : the mini-batch size
- N_{reg} : the number of anchor locations

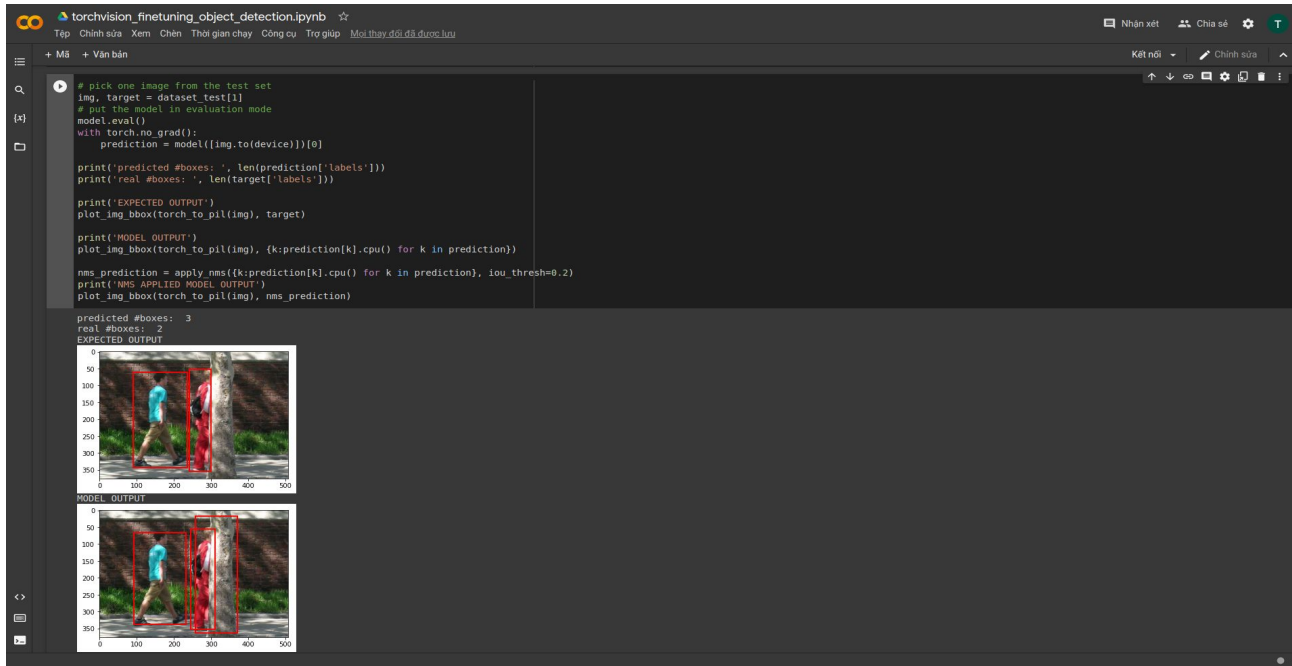
Challenges

2-stage object detector:

- First stage: run once per image
- Second stage: run once per RoI

→ Accurate but slow for real-time detection

Practical notebook



The screenshot shows a Jupyter Notebook interface with a dark theme. The notebook title is 'torchvision_finetuning_object_detection.ipynb'. The code in the cell is as follows:

```
# pick one image from the test set
img, target = dataset_test[1]
# put the model in evaluation mode
model.eval()
with torch.no_grad():
    prediction = model([img.to(device)])[0]

print('predicted #boxes: ', len(prediction['labels']))
print('real #boxes: ', len(target['labels']))

print('EXPECTED OUTPUT:')
plot_img_bbox(torch_to_pil(img), target)

print('MODEL OUTPUT:')
plot_img_bbox(torch_to_pil(img), {k:prediction[k].cpu() for k in prediction})

nms_prediction = apply_nms({k:prediction[k].cpu() for k in prediction}, iou_thresh=0.2)
print('NMS APPLIED MODEL OUTPUT:')
plot_img_bbox(torch_to_pil(img), nms_prediction)
```

The output of the cell shows the following text:

```
predicted #boxes: 3
real #boxes: 2
EXPECTED OUTPUT
```

Below the text, there are two side-by-side plots. The top plot, labeled 'EXPECTED OUTPUT', shows an image of a person walking on a path with two red bounding boxes. The bottom plot, labeled 'MODEL OUTPUT', shows the same image with three red bounding boxes. Both plots have x and y axes ranging from 0 to 500.

Faster-RCNN fine-tuning tutorial

https://colab.research.google.com/drive/1I3p5FfYgMzWpo6vHYRN_OKF68HKBZW-F?usp=sharing

References

www.analyticsvidhya.com, Machine learning blog

- https://www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-tensorflow-api/?utm_source=blog&utm_medium=Non_Max_Suppression_for_Object_Detection

commons.wikimedia.org, Wikipedia images

- https://upload.wikimedia.org/wikipedia/commons/c/c7/Intersection_over_Union_-_visual_equation.png

www.paperswithcode.com

- <https://paperswithcode.com/sota/object-detection-on-coco>

www.medium.com

- <https://medium.com/analytics-vidhya/object-localization-using-keras-d78d6810d0be>
- <https://becominghuman.ai/computer-vision-object-detection-challenges-faced-9a927f9c5623>
- <https://medium.com/the-space-ape/games-experience/aspect-ratio-scaling-mobile-and-tablets-d574ab20a943>
- <https://towardsdatascience.com/yolo-v4-optimal-speed-accuracy-for-object-detection-79896ed47b50>

www.researchgate.com

- https://www.researchgate.net/figure/Illustration-of-the-issue-of-viewpoint-variation-in-the-context-of-action-recognition_fig1_338913739
- https://www.researchgate.net/figure/shows-an-example-of-Intra-Class-variation-on-the-Chair-class-But-how-will-a-machine_fig1_325311506
- https://www.researchgate.net/figure/An-example-of-low-illumination-objects-detection-Our-detector-have-achieved-amazing_fig1_342757964

References

www.exposit.com

- <https://www.exposit.com/blog/computer-vision-object-detection-challenges-faced/>

www.stackoverflow.com

- <https://stackoverflow.com/questions/2764238/image-processing-what-are-occlusions>
- <https://stats.stackexchange.com/questions/233512/how-to-get-the-data-set-size-required-for-neural-network-training>

Justin Johnson, Lectures from EECS 442 - Computer Vision course at the University of Michigan:

- https://web.eecs.umich.edu/~justincj/slides/eecs498/WI2022/598_WI2022_lecture14.pdf
- https://web.eecs.umich.edu/~justincj/slides/eecs498/WI2022/598_WI2022_lecture13.pdf

Ross Girshick, RCNN, <https://arxiv.org/abs/1311.2524v5>

Ross Girshick, Fast-RCNN, <https://arxiv.org/pdf/1504.08083.pdf>

Shaoqing Ren et al., Faster-RCNN <https://arxiv.org/pdf/1506.01497.pdf>

Thank you for listening!



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

www.htw-berlin.de