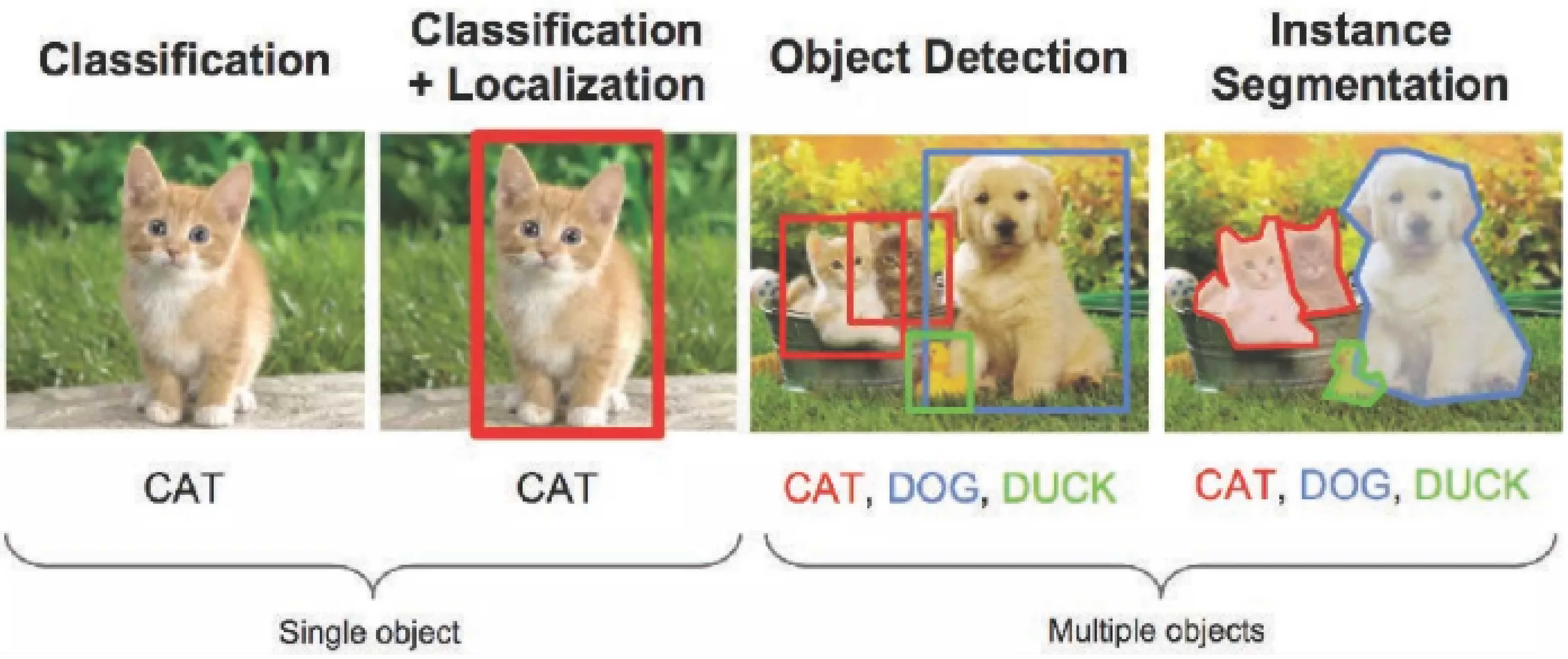
LAB 6 - RCNN AND FASTER RCNN

Dr. Pandiyaraju V
Shravan Venkatraman

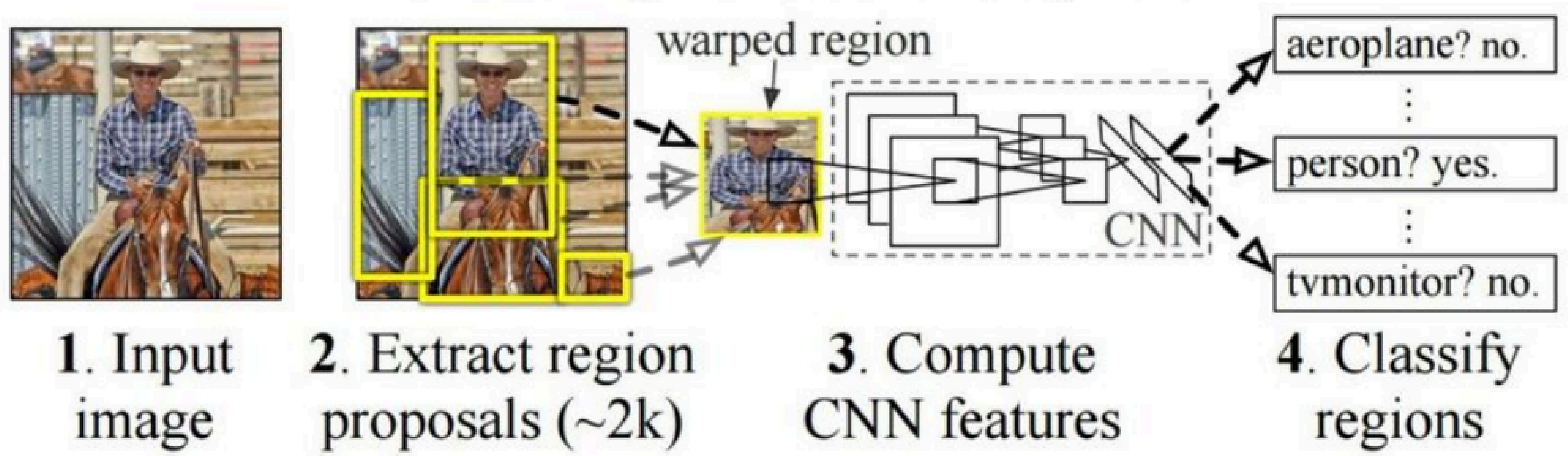
September 27, 2024

TABLE OF CONTENTS

COMPUTER VISION TASKS

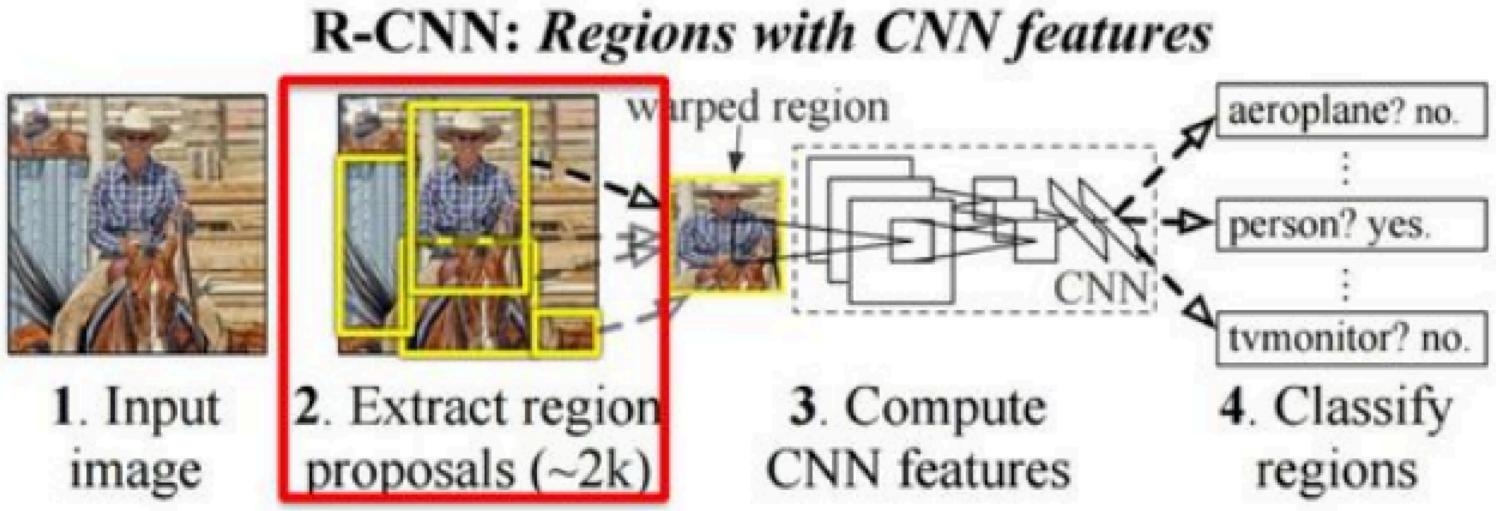


R-CNN: REGIONS WITH CNN FEATURES



R-CNN: REGIONS WITH CNN FEATURES

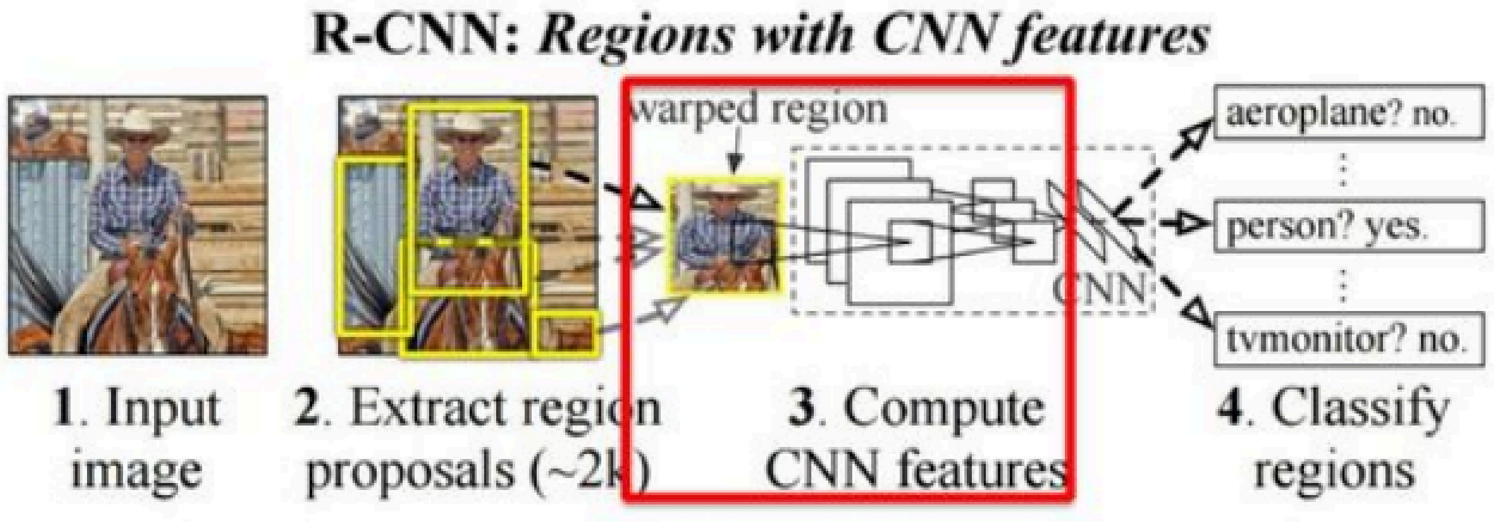
- First Stage: generate category-independent region proposals.
    - 2000 Region proposals every image.

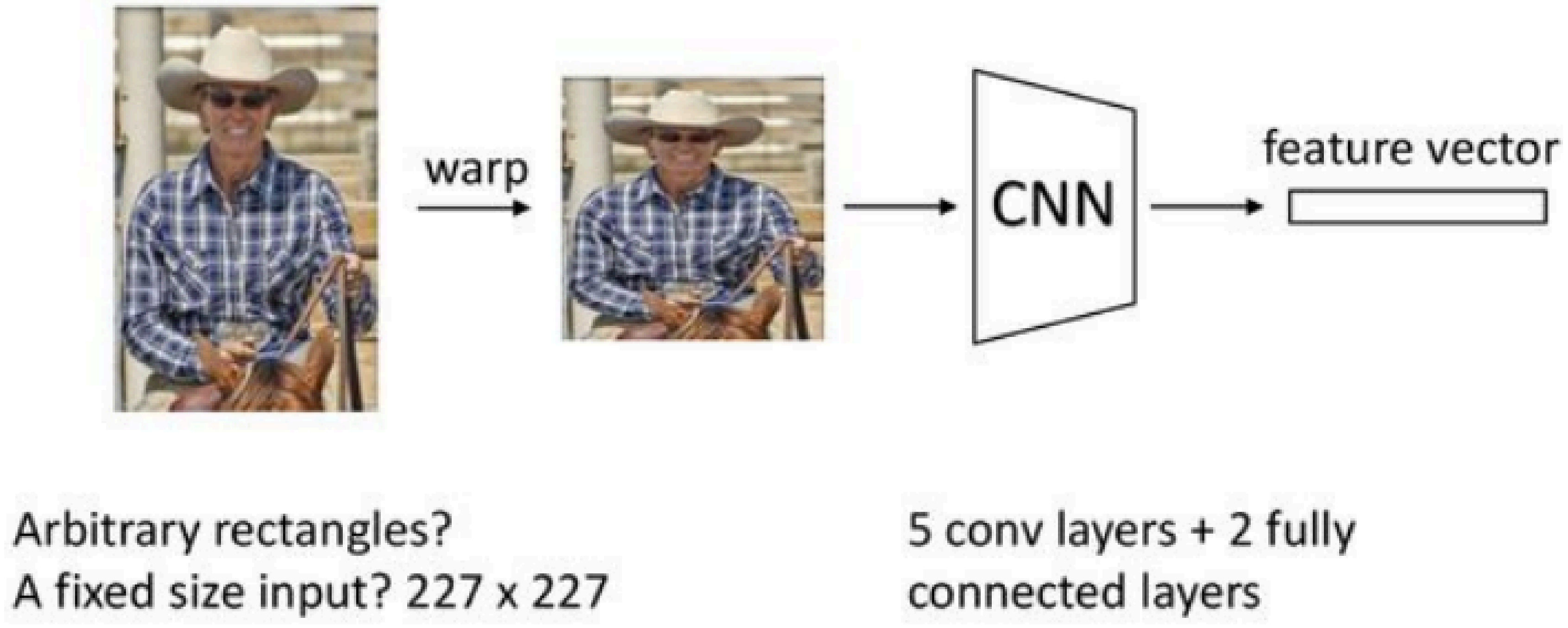


Selective Search: Combining strengths of both exhaustive search and segmentation.

R-CNN: REGIONS WITH CNN FEATURES

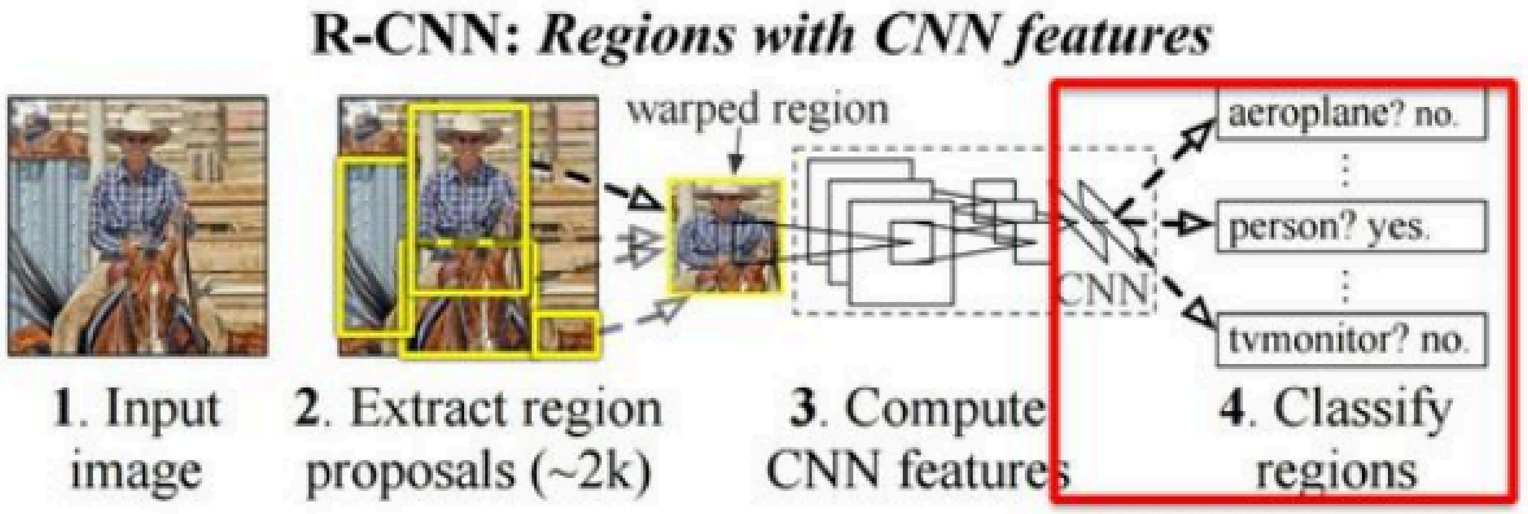
- First Stage: generate category-independent region proposals.
    - 2000 Region proposals every image.
- Second Stage: Extract a fixed length feature vector from each region.
    - a 4096-dimensional feature vector from each region proposal.

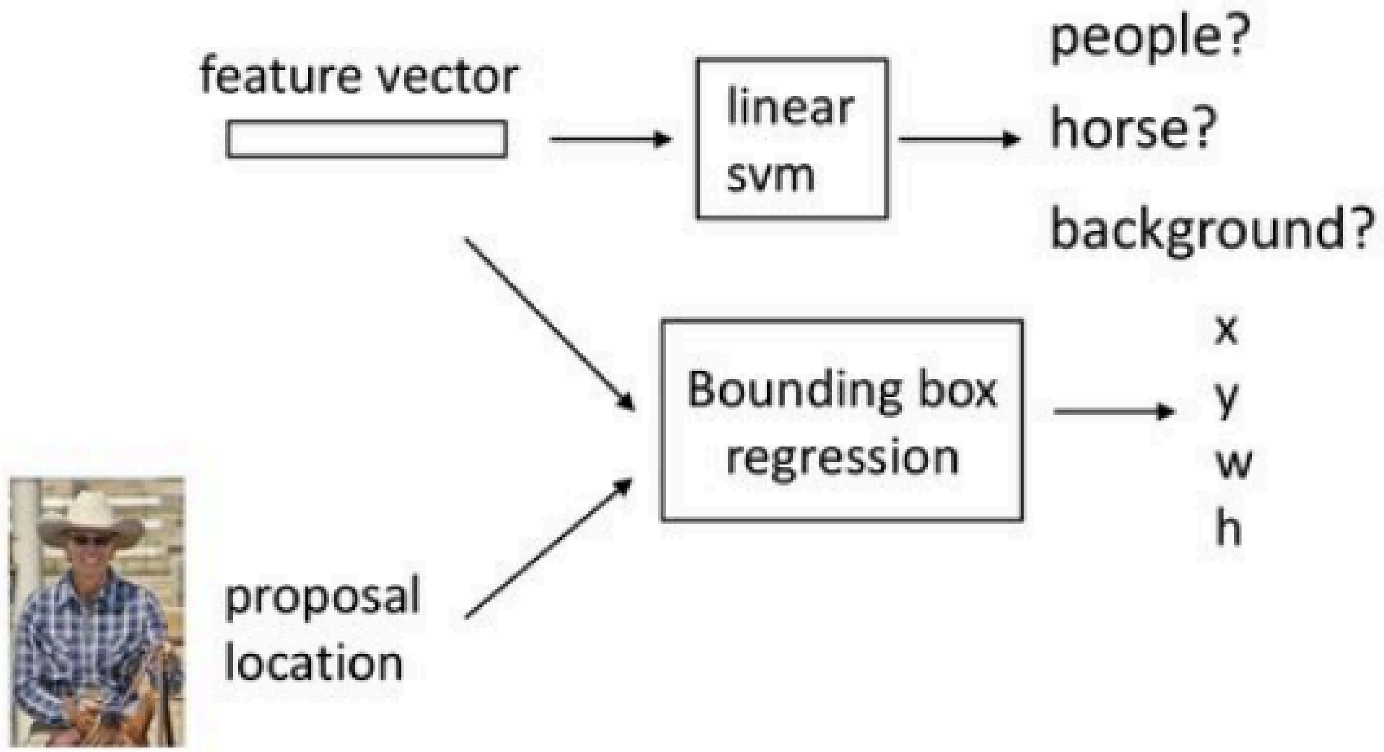




R-CNN: REGIONS WITH CNN FEATURES

- First Stage: generate category-independent region proposals.
  - 2000 Region proposals every image.
- Second Stage: Extract a fixed length feature vector from each region.
  - a 4096-dimensional feature vector from each region proposal.
- Third Stage: a set of class-specific linear SVMs.
  - object category and localization.





WHY DO WE NEED FAST RCNN?

RCNN

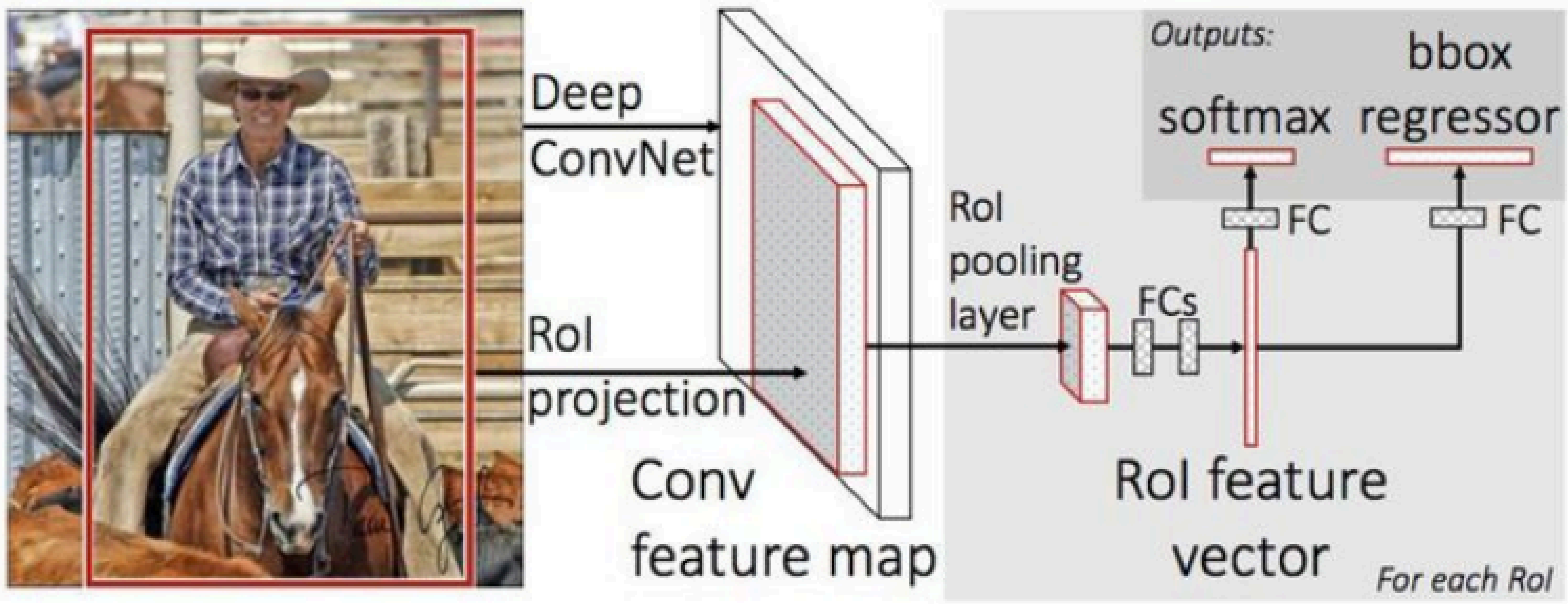
- Simple and scalable.
- improves mAP.

- A multistage pipeline.
- Training is expensive in space and time (features are extracted from each region proposal in each image and written into disk).
- Object detection is slow.

Fast-RCNN

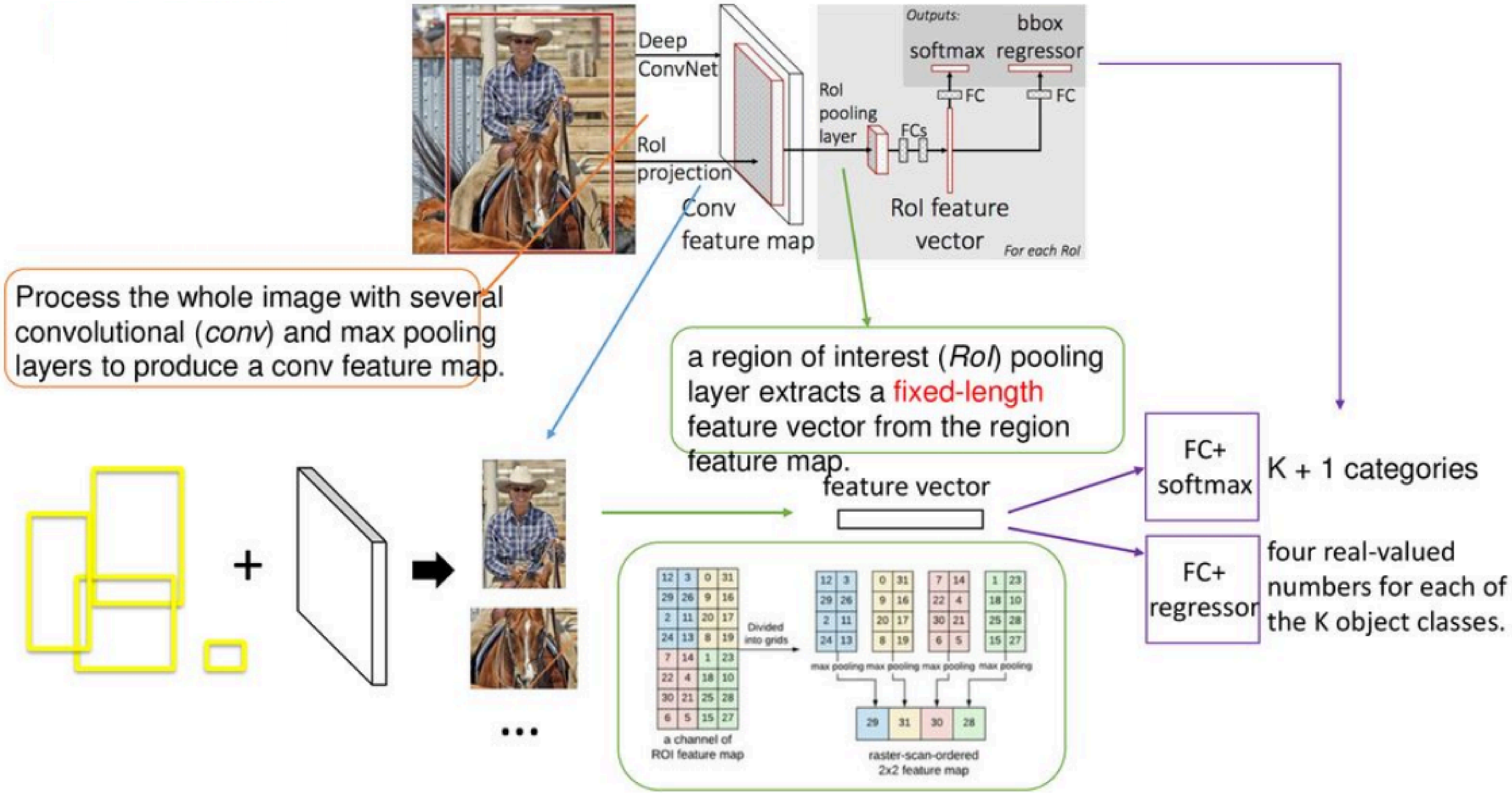
?

FAST R-CNN



Idea: No need to recompute features for each box independently.

FAST R-CNN



Deep
ConvNet
RoI
projection
Conv
feature map
Outputs:
softmax
bbox
regressor
FC
FC
RoI
pooling
layer
FCs
RoI feature
vector
For each RoI
Process the whole image with several convolutional (conv) and max pooling layers to produce a conv feature map.
a region of interest (RoI) pooling layer extracts a fixed-length feature vector from the region feature map.
feature vector
FC+
softmax
K + 1 categories
FC+
regressor
four real-valued numbers for each of the K object classes.
Divided into grids
max pooling
a channel of ROI feature map
raster-scan-ordered 2x2 feature map

WHY DO WE NEED FASTER RCNN?

RCNN

- Simple and scalable.
- improves mAP.

- A multistage pipeline.
- Training is expensive in space and time (features are extracted from each region proposal in each image and written into disk).
- Object detection is slow.

Fast-RCNN

- Higher mAP.
- Single stage, end-to-end training.
- No disk storage is required for feature caching.

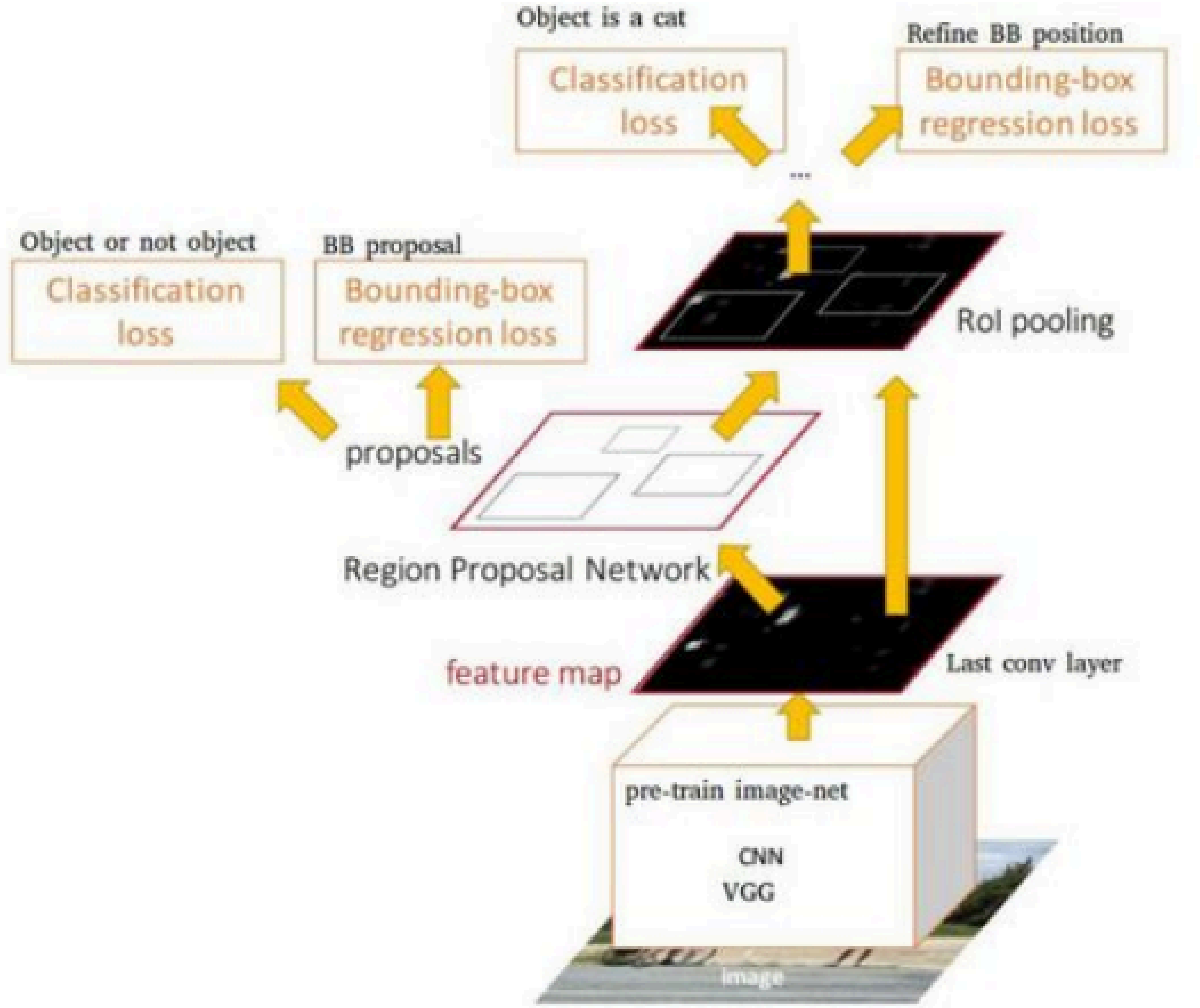
- proposals are the computational bottleneck in detection systems.

Faster-RCNN

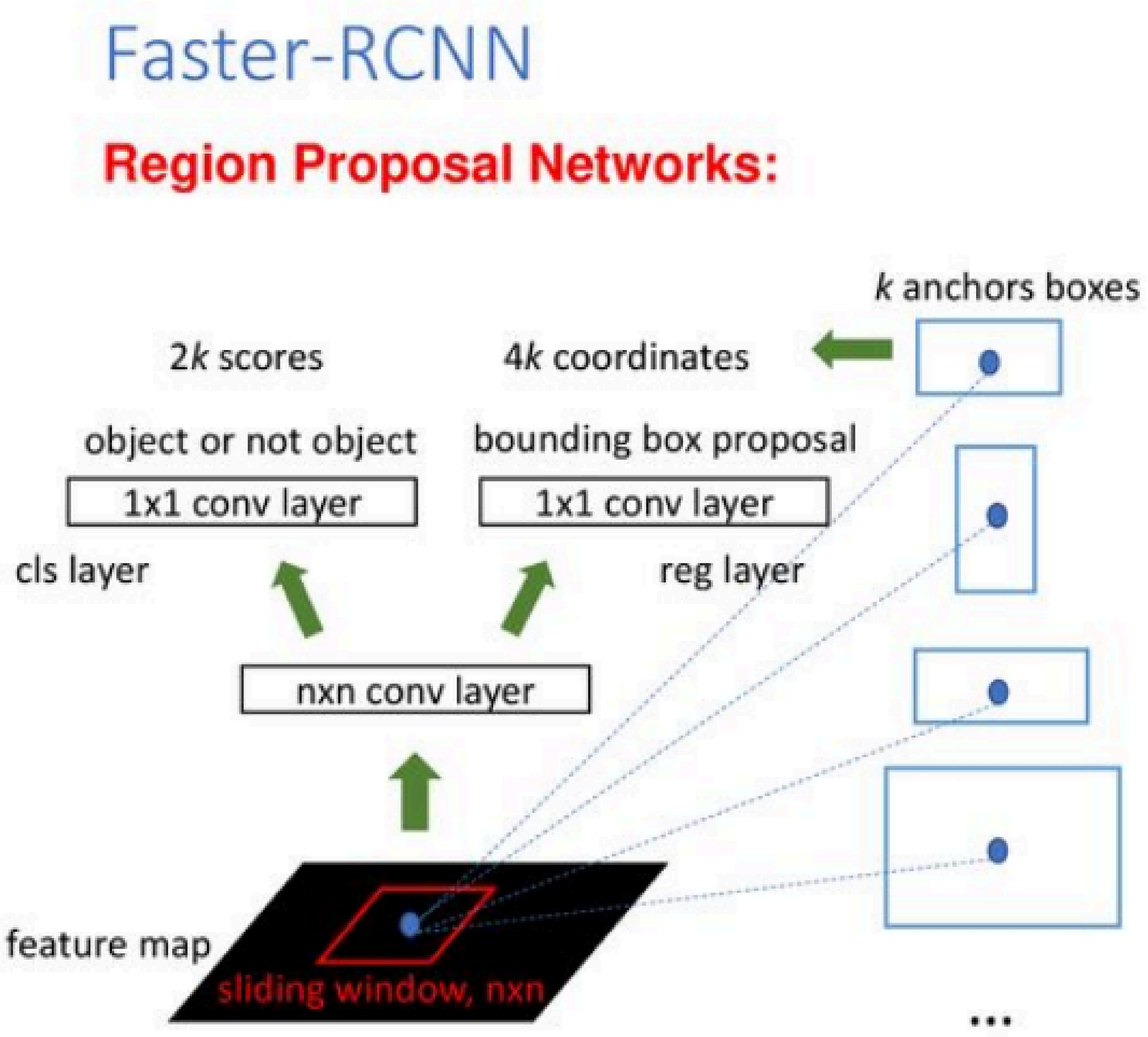
?

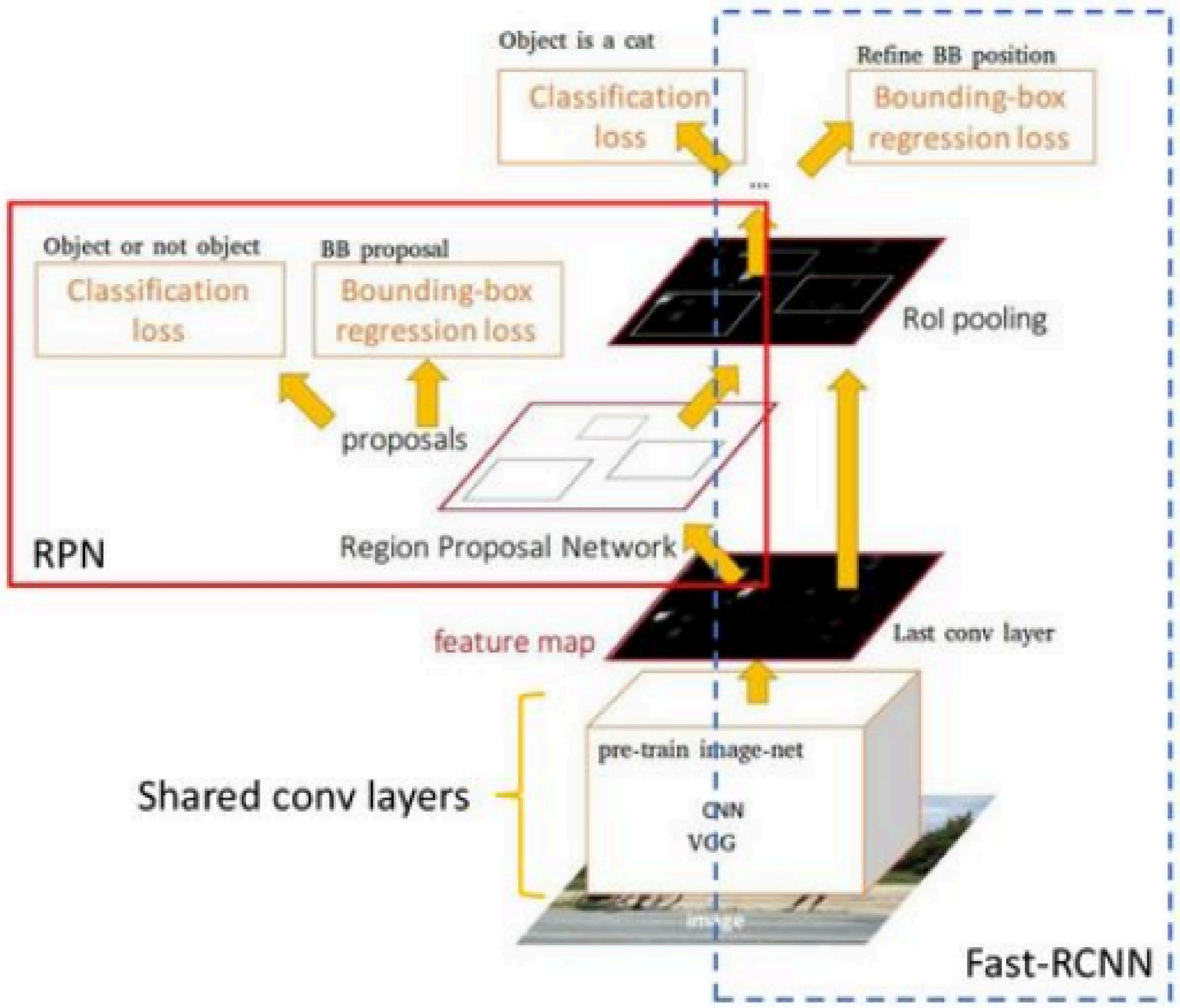
FASTER R-CNN

Idea: Integrate the bounding box proposals as part of the CNN predictions.



FASTER R-CNN


Faster-RCNN
Region Proposal Networks:
k anchors boxes
2k scores
4k coordinates
object or not object
bounding box proposal
1x1 conv layer
1x1 conv layer
cls layer
reg layer
nxn conv layer
feature map
sliding window, nxn
...


Object is a cat
Classification loss
Refine BB position
Bounding-box regression loss
Object or not object
BB proposal
Classification loss
Bounding-box regression loss
RoI pooling
proposals
RPN
Region Proposal Network
feature map
Last conv layer
pre-train image-net
CNN
VGG
Shared conv layers
Fast-RCNN

THEN WHY DOES YOLO EXIST?

RCNN

- Simple and scalable.
- improves mAP.

- A multistage pipeline.
- Training is expensive in space and time (features are extracted from each region proposal in each image and written into disk).
- Object detection is slow.

Fast-RCNN

- Higher mAP.
- Single stage, end-to-end training.
- No disk storage is required for feature caching.

- proposals are the computational bottleneck in detection systems.

Faster-RCNN

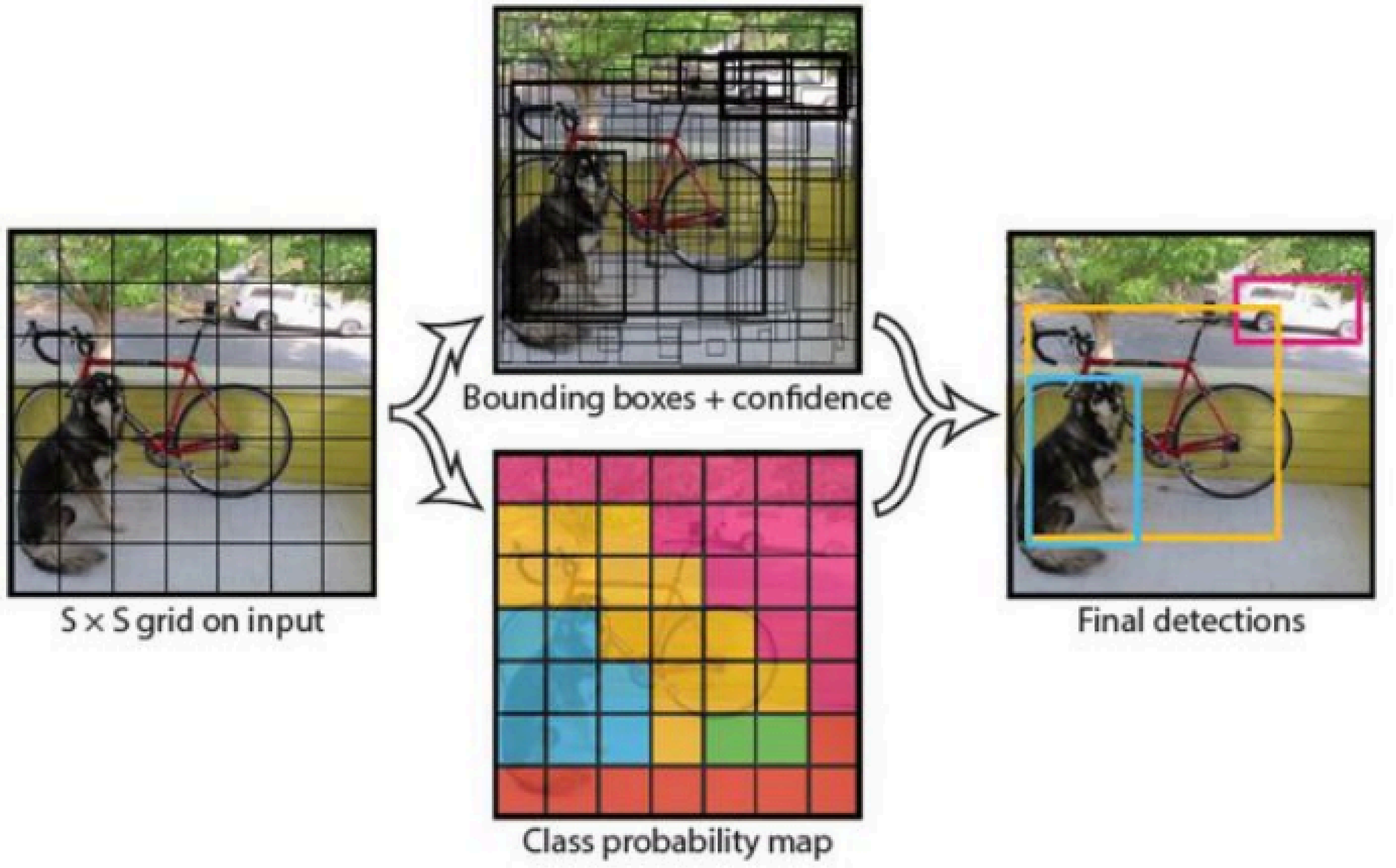
- compute proposals with a deep convolutional neural network --*Region Proposal Network* (RPN)
- merge RPN and Fast R-CNN into a single network, enabling nearly cost-free region proposals.

?

YOLO (YOU ONLY LOOK ONCE)

Idea: No bounding box proposal. A single regression problem, straight from image pixels to bounding box coordinates and class probabilities.

- Extremely fast
- Learns generalizable representations.



HANDS ON / CHALLENGE TASK

Task 1: In this task, you will use the provided Fast RCNN and Faster RCNN models to perform object detection on any object detection dataset of your choice. You are expected to fit the dataloaders, train both models, and evaluate their performance.

Dataset: any object detection dataset (Pascal VOC subset, COCO subset, etc)
Metrics: use mean average precision (mAP), inference time, and accuracy.

Reference code for Fast RCNN and Faster RCNN:
https://colab.research.google.com/drive/1E18GeirhjNQrVb_04en5ot_ch4ziH259?usp=sharing