



Object Detection

Innopolis University
Applied Machine learning
BEKKOUCH Imad Eddine Ibrahim



Learning Outcomes

Object detection - RCNN

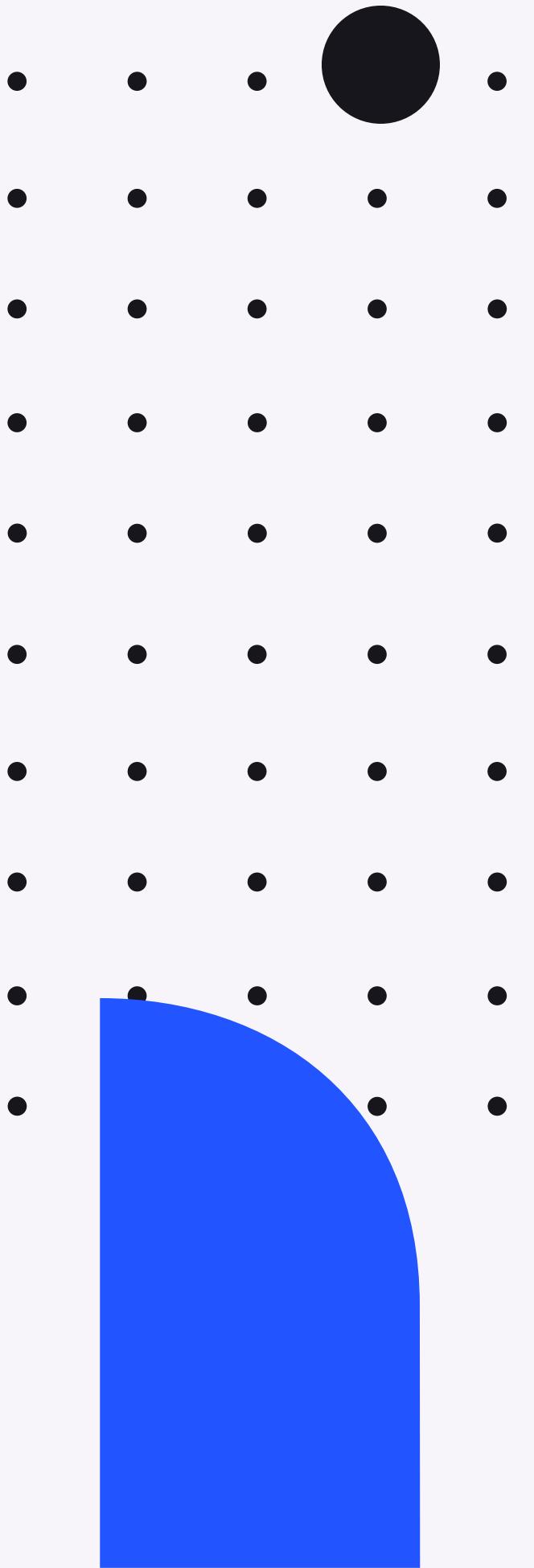
What's object detection? How to perform it?

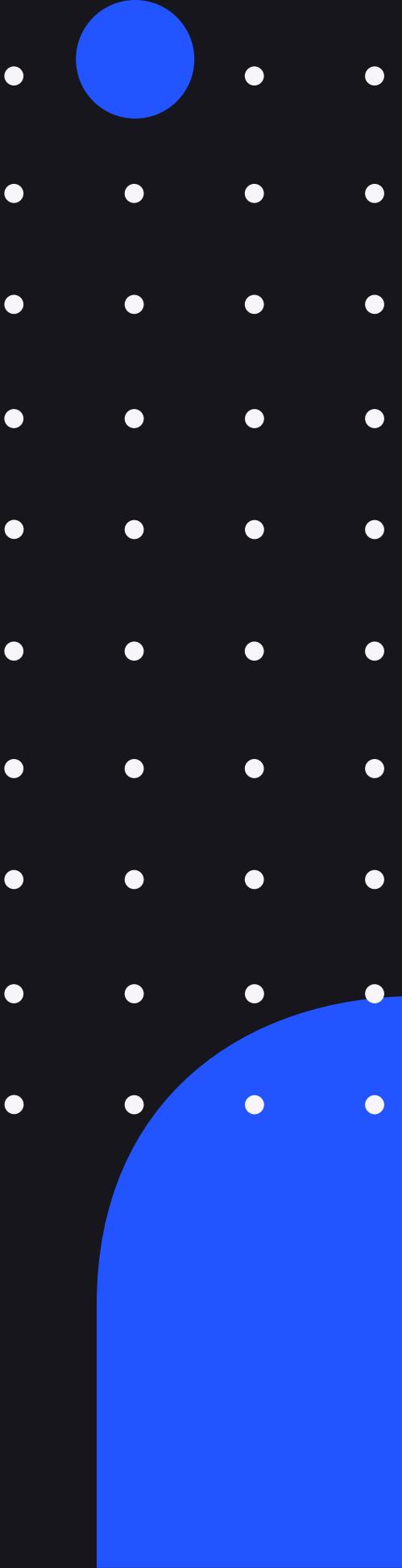
Fast/Faster RCNN

How to make the models faster

Yolo & Detectron

Yolov3 & Detectron2





Object Detection - RCNN

Region Based Convolutional Neural Networks

- **Object detection**

Expected output, examples of applications

- **Basic Model**

Sliding window and discuss improvements

- **RCNN**

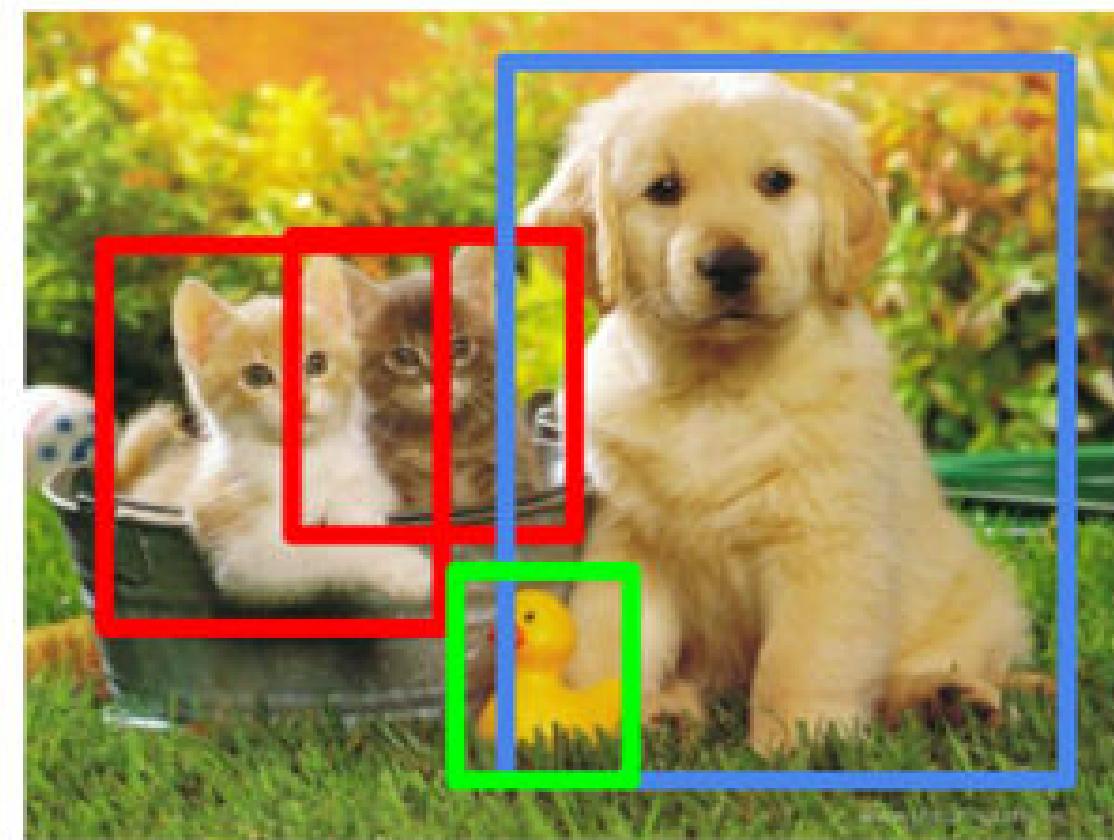
Mean Squared Error/Cross Entropy

Object Detection

Comparison between Object detection and image classification.



CAT

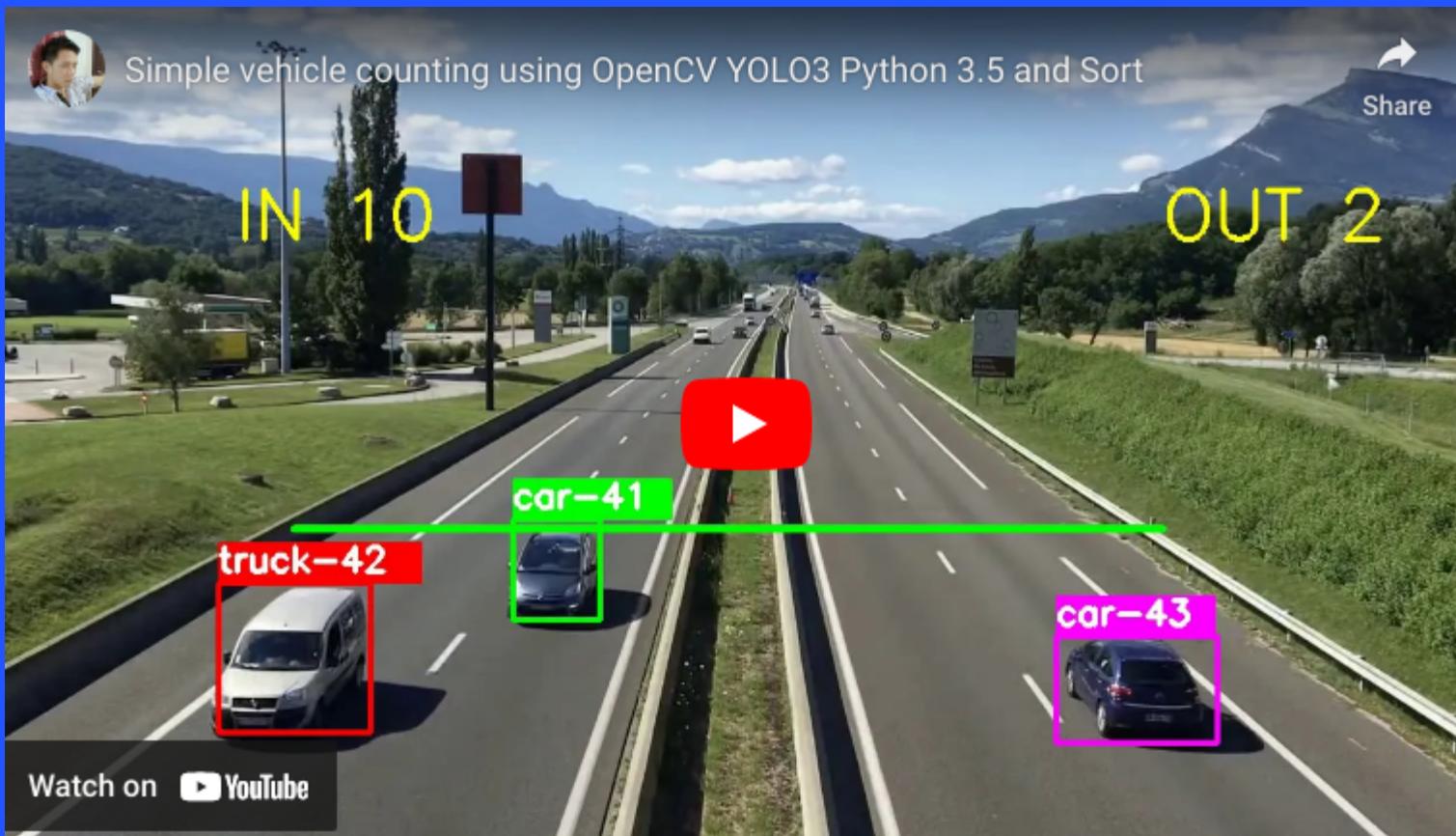


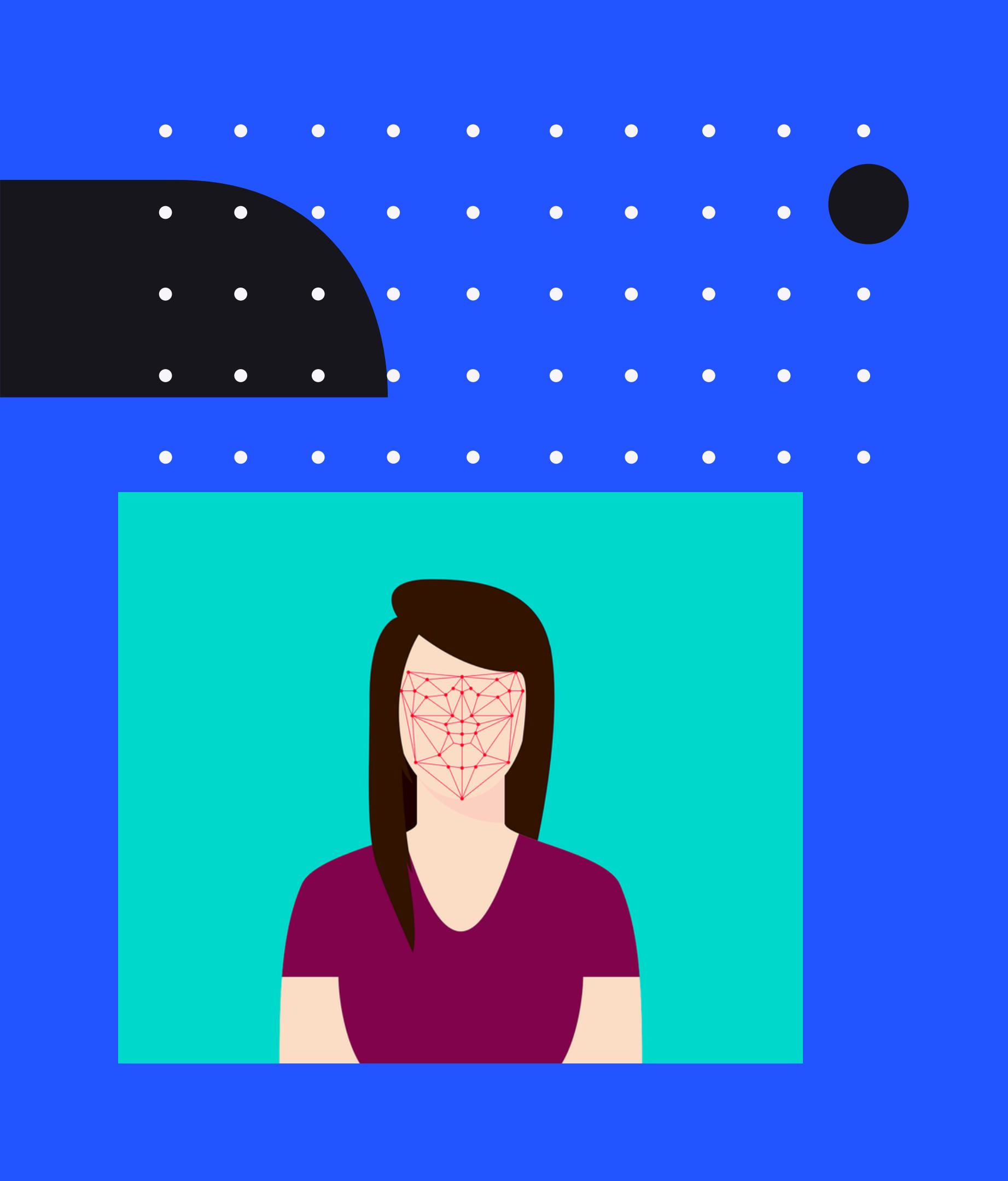
CAT, DOG, DUCK



Vehicle counting

A combination of object tracking and
object detection





Facial Detection & Recognition

A combination of object detection and facial recognition



Parking Lot monitoring

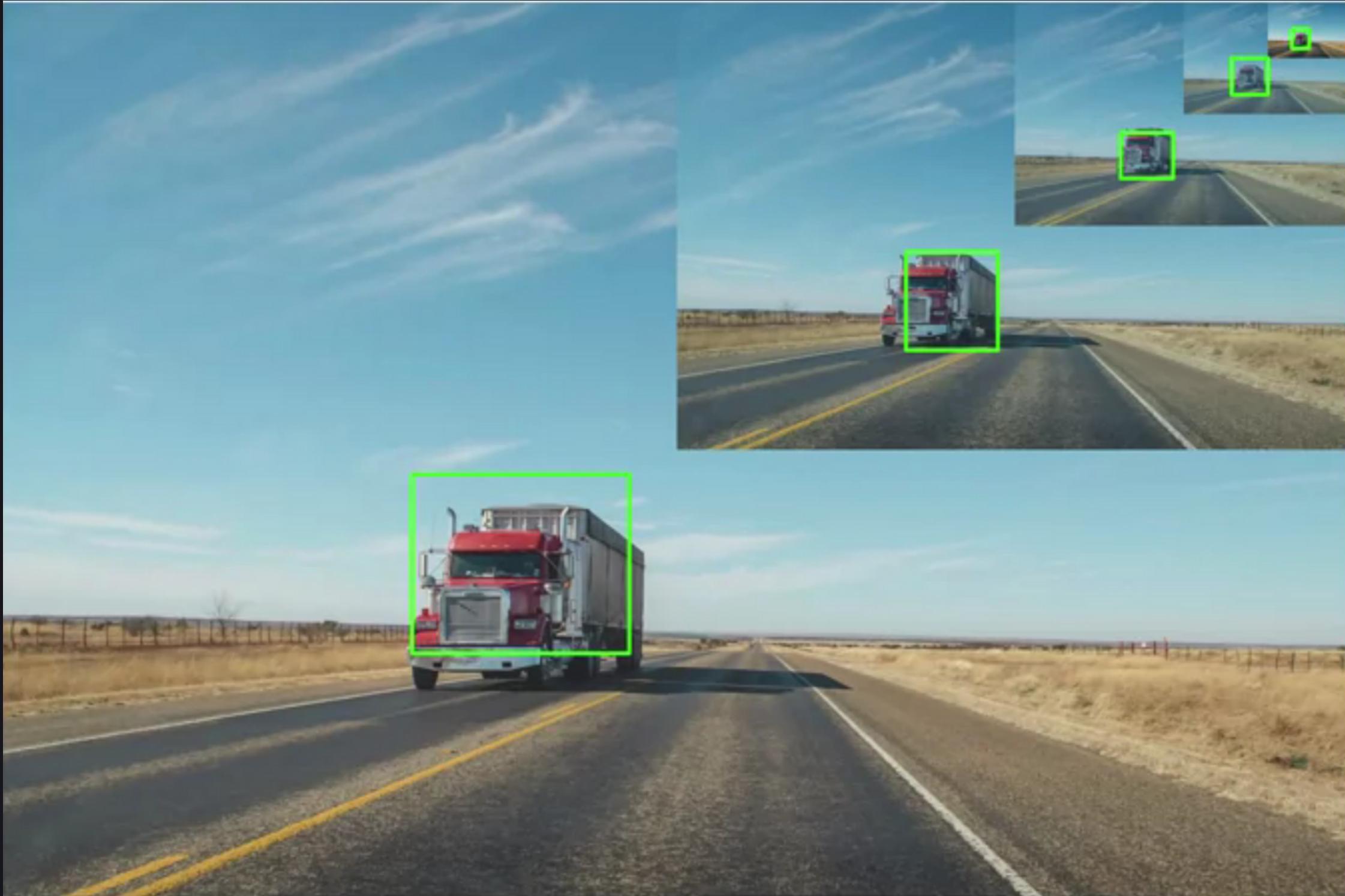
Detecting free slots inside parking lots.



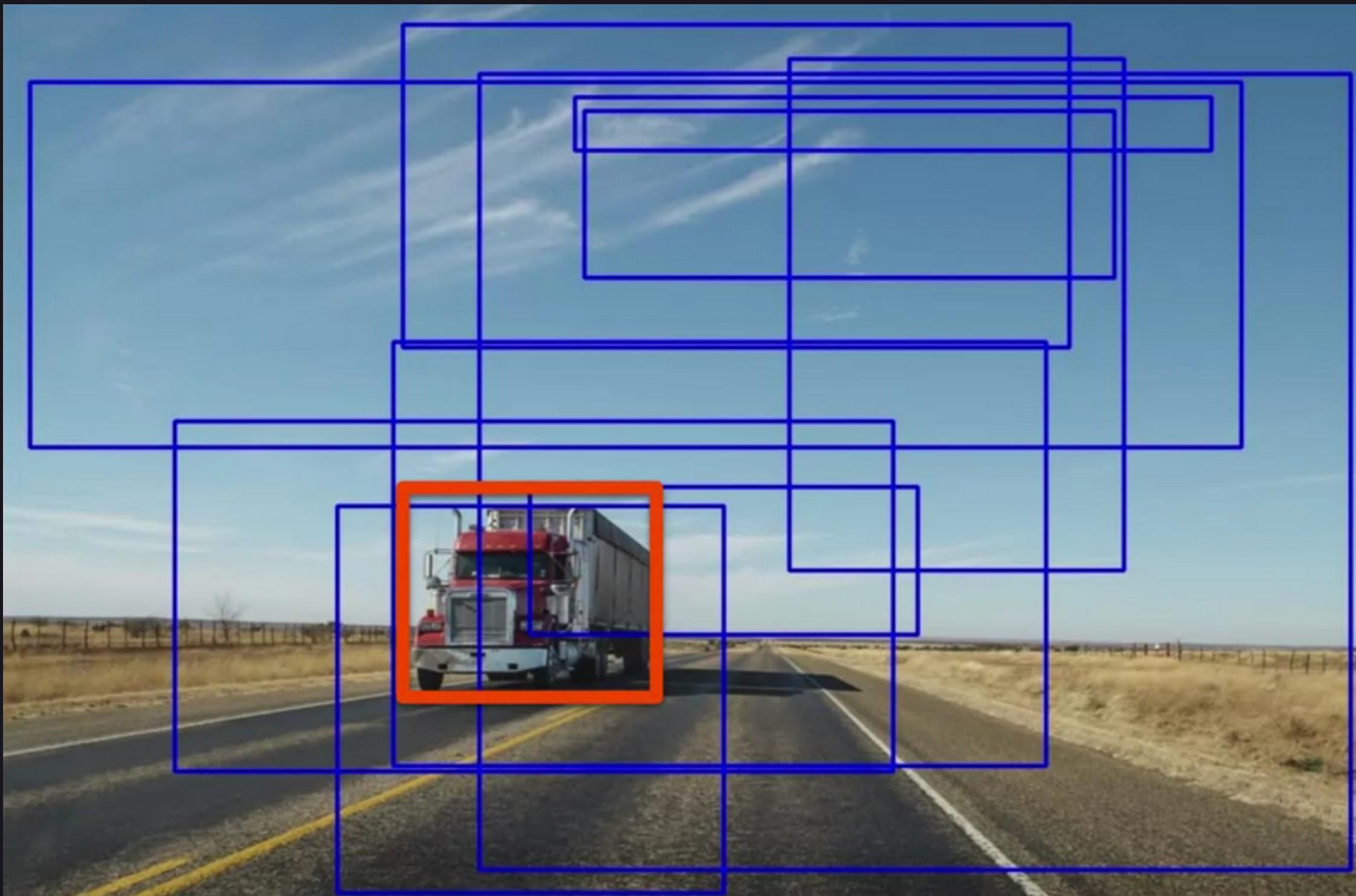
Basic Model - Sliding 2D CNN



Improvement - Speed

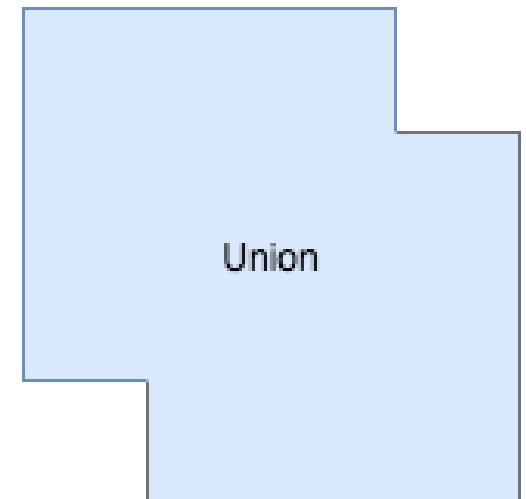
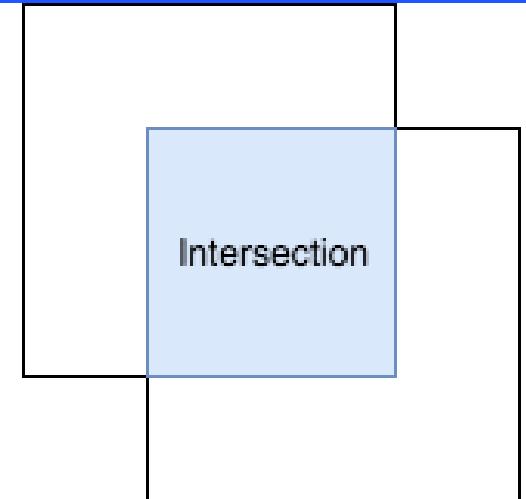


Selective search





IoU = _____

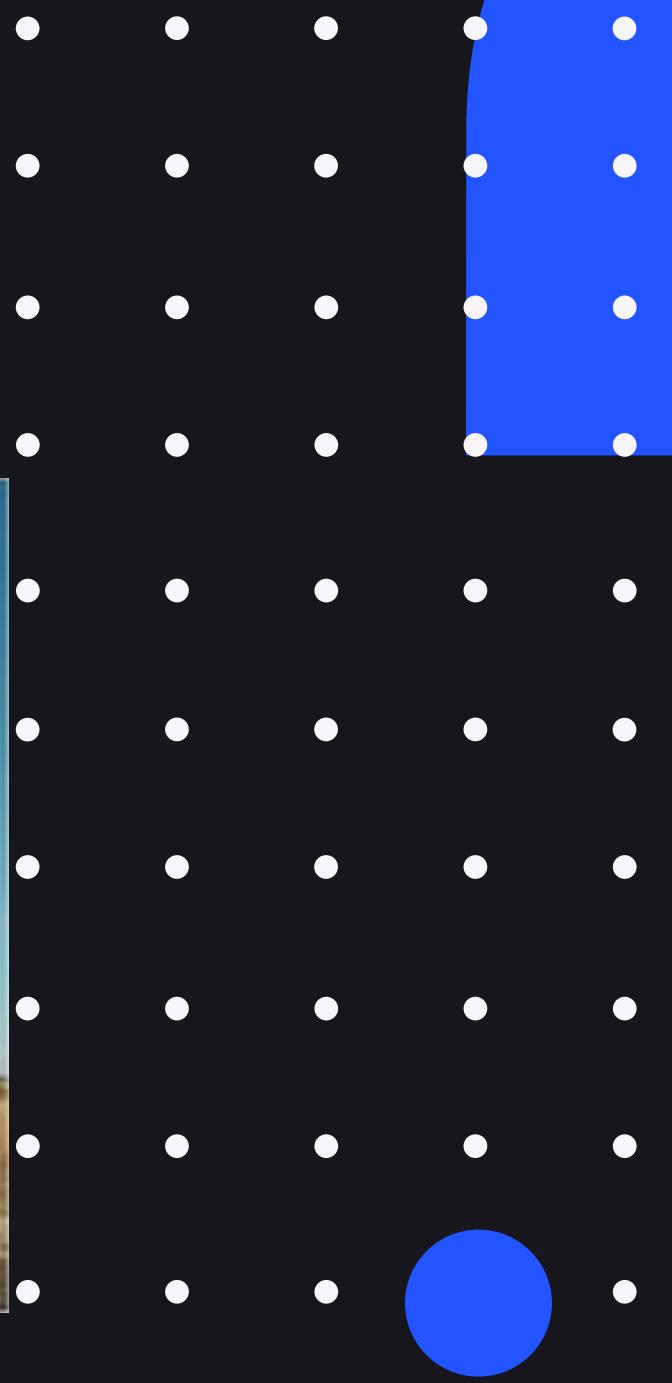
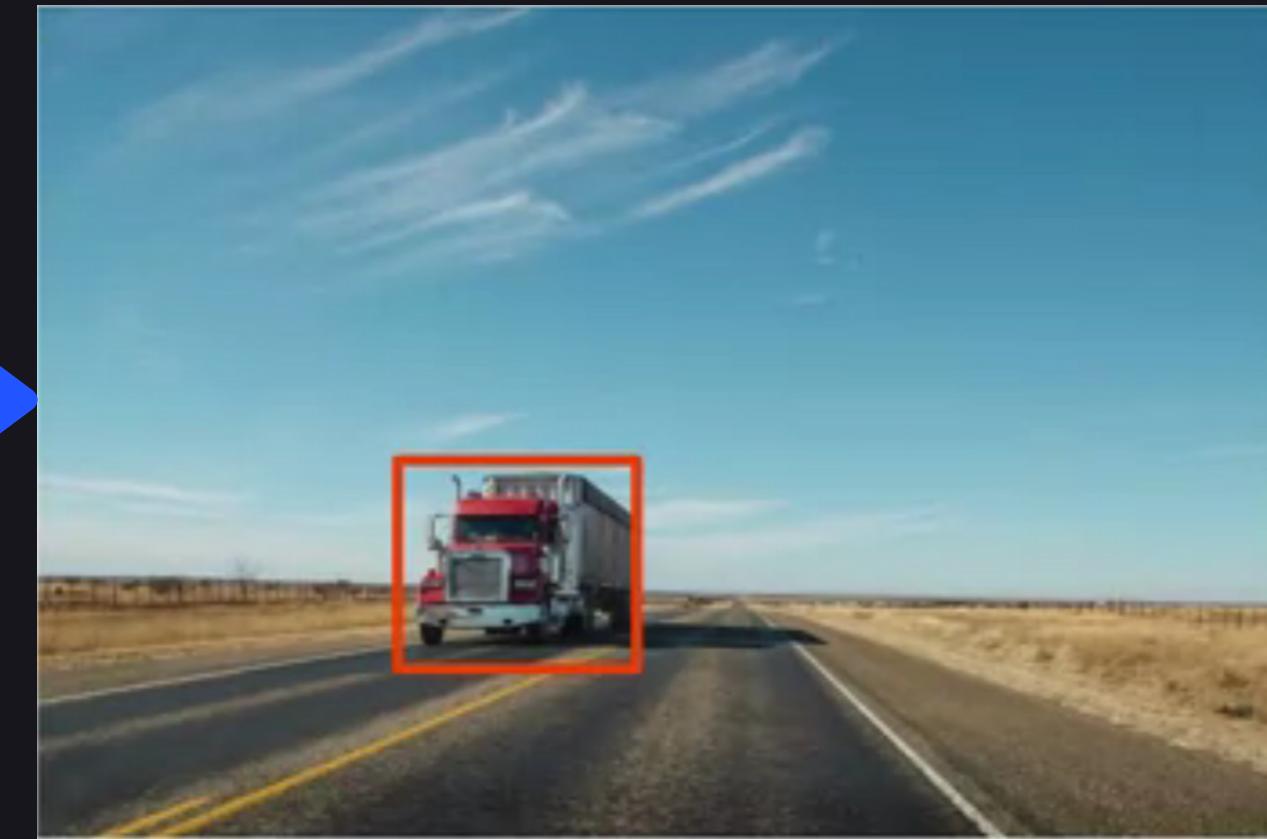
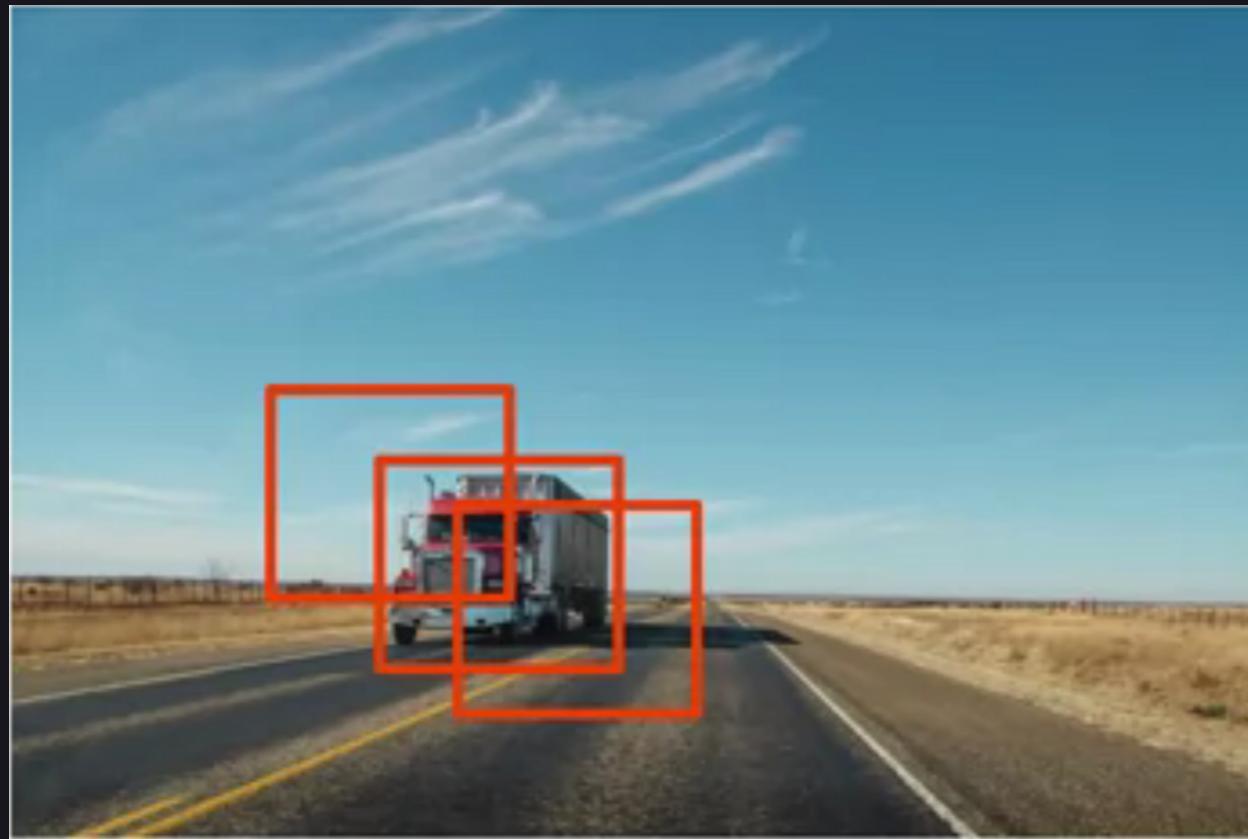


Intersection Over Union

The equivalent of accuracy in classification for object detection.



Non Maximum Supression



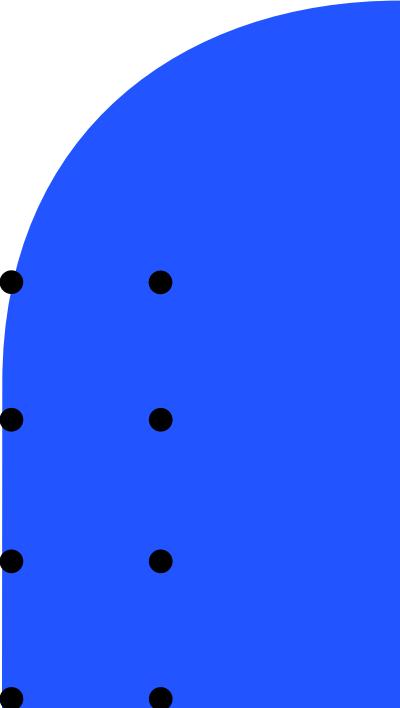
Non Maximum Supression

- Select the proposal with highest confidence score, remove it from B and add it to the final proposal list D. (Initially D is empty).
- Now compare this proposal with all the proposals – calculate the IOU (Intersection over Union) of this proposal with every other proposal. If the IOU is greater than the threshold N, remove that proposal from B.
- Again take the proposal with the highest confidence from the remaining proposals in B and remove it from B and add it to D.
- Once again calculate the IOU of this proposal with all the proposals in B and eliminate the boxes which have high IOU than threshold.
- This process is repeated until there are no more proposals left in B.

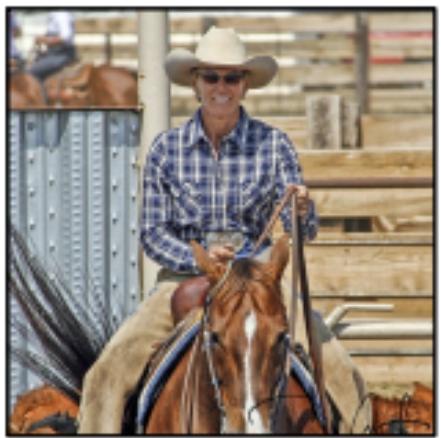


>

RCNN



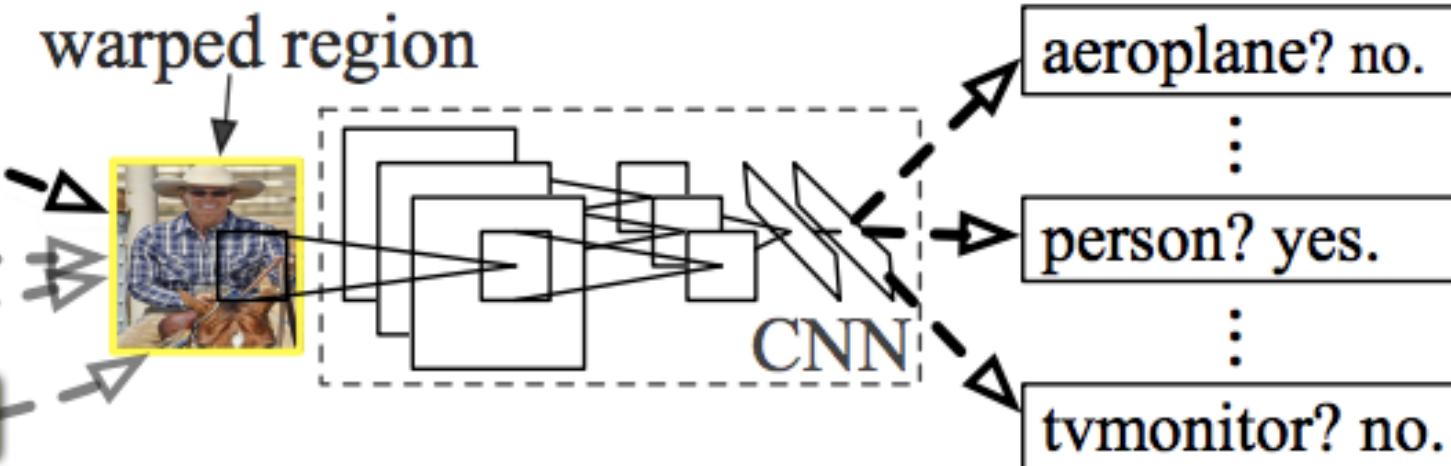
R-CNN: *Regions with CNN features*



1. Input image



2. Extract region proposals (~2k)



3. Compute CNN features

4. Classify regions



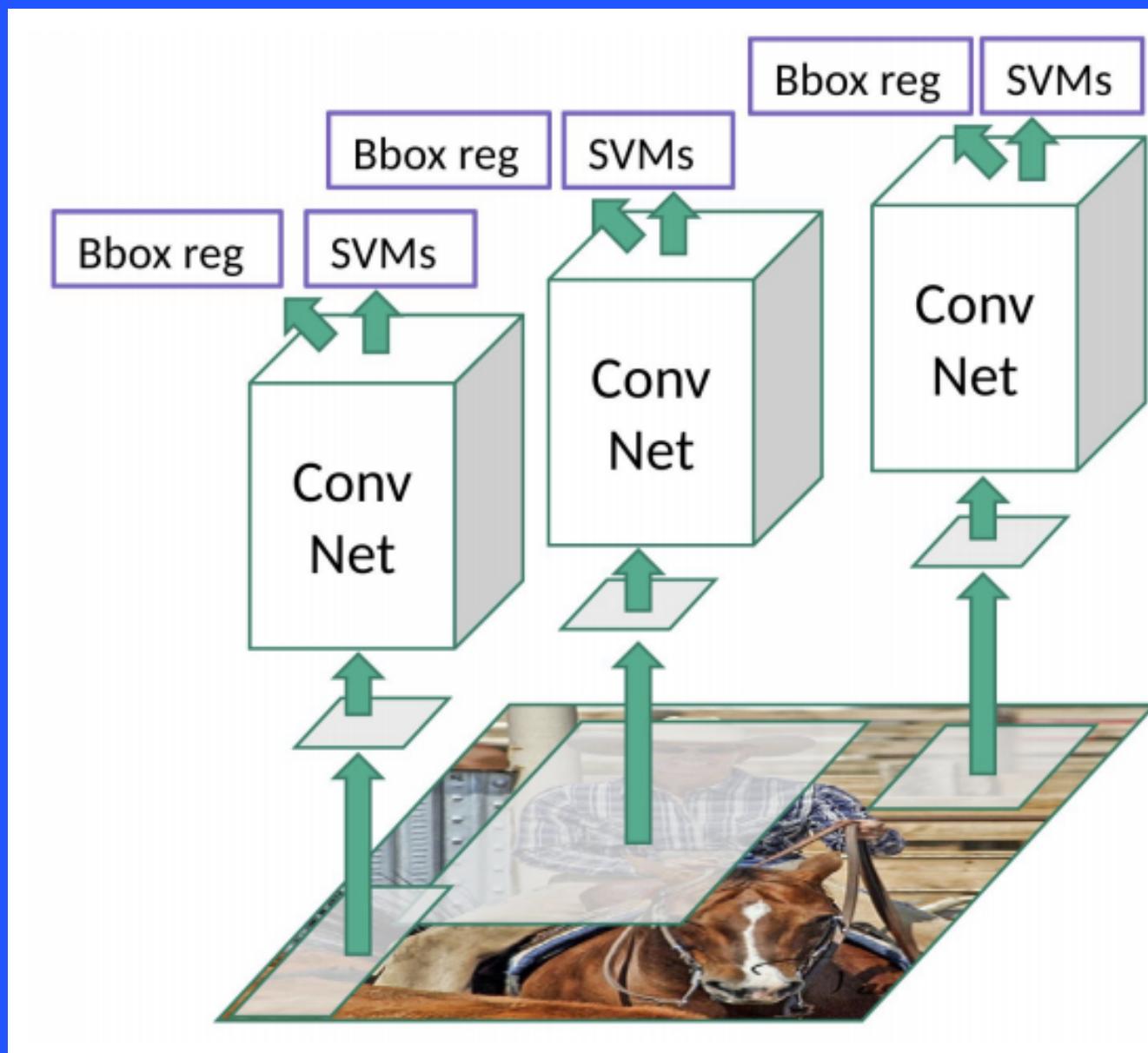
Selective Search

- Generate initial sub-segmentation, we generate many candidate regions
- Use greedy algorithm to recursively combine similar regions into larger ones
- Use the generated regions to produce the final candidate region proposals



Problems RCNN

- Training time, 2000 classification per image.
- Inference time around 47s.
- Selective search is not dataset specific.



Object Detection

1. Object Detection

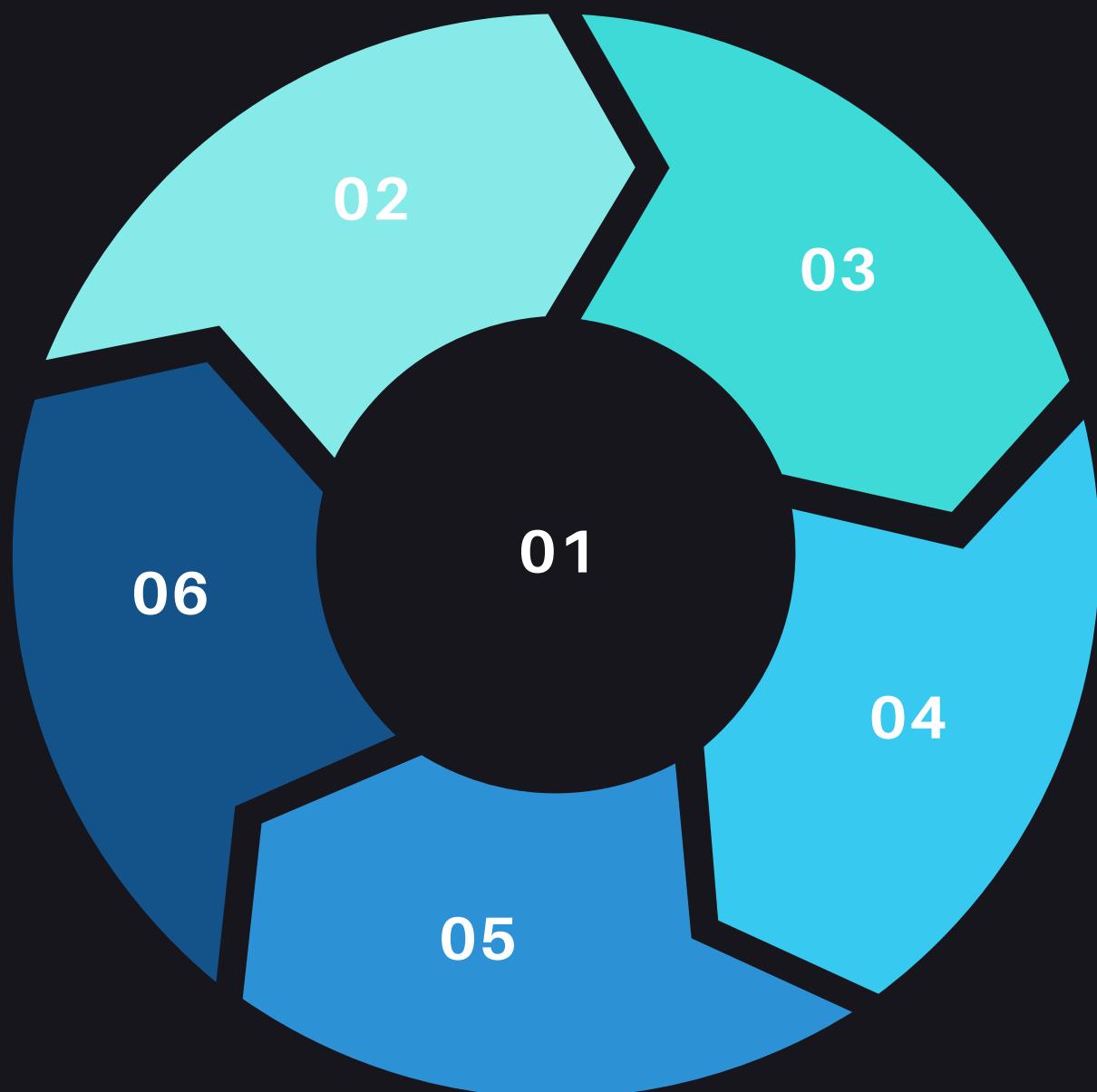
predicting the class and location of objects

2. IOU & NMS

intersection over union and non maximal suppression

3. RCNN

Region Based Convolutional Neural Networks



4. Fast RCNN

Reducing the convolutional operations

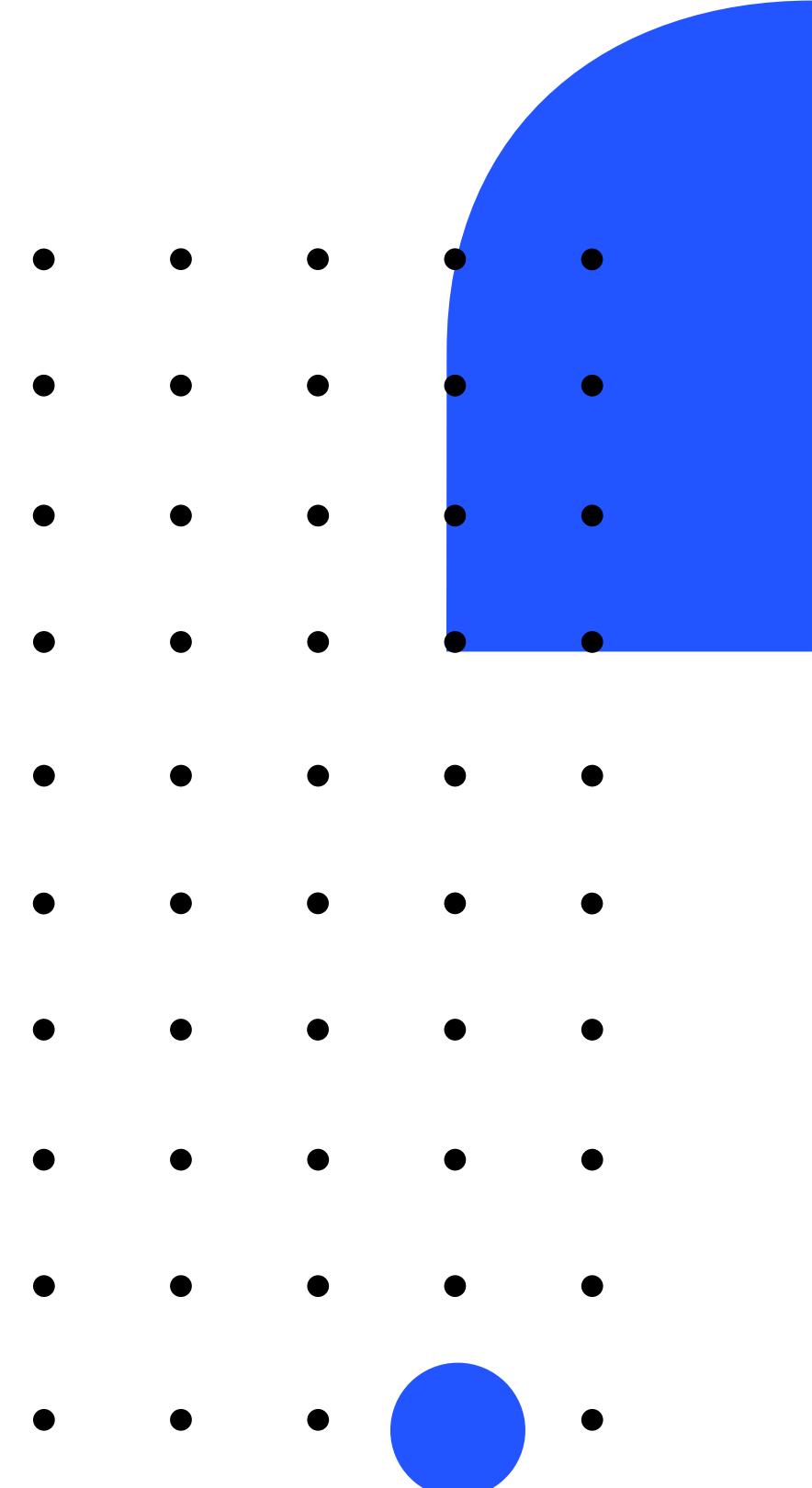
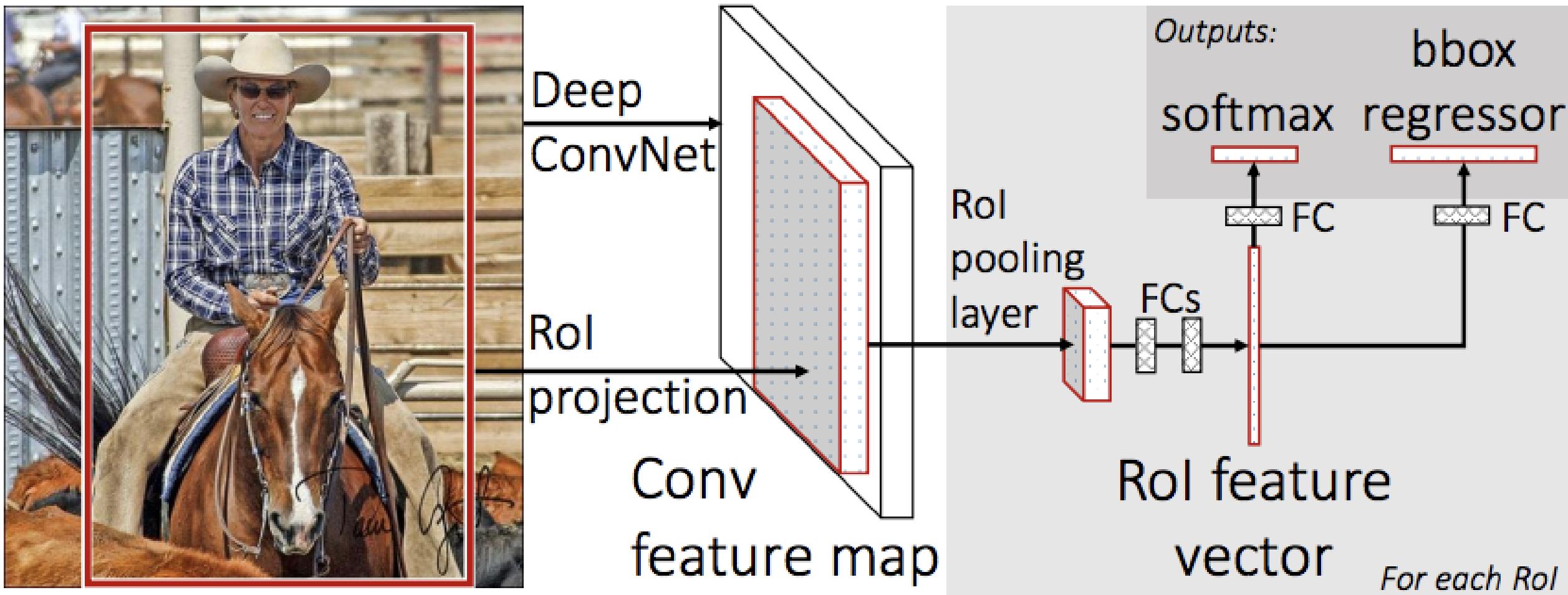
5. Faster RCNN

Removing selective search

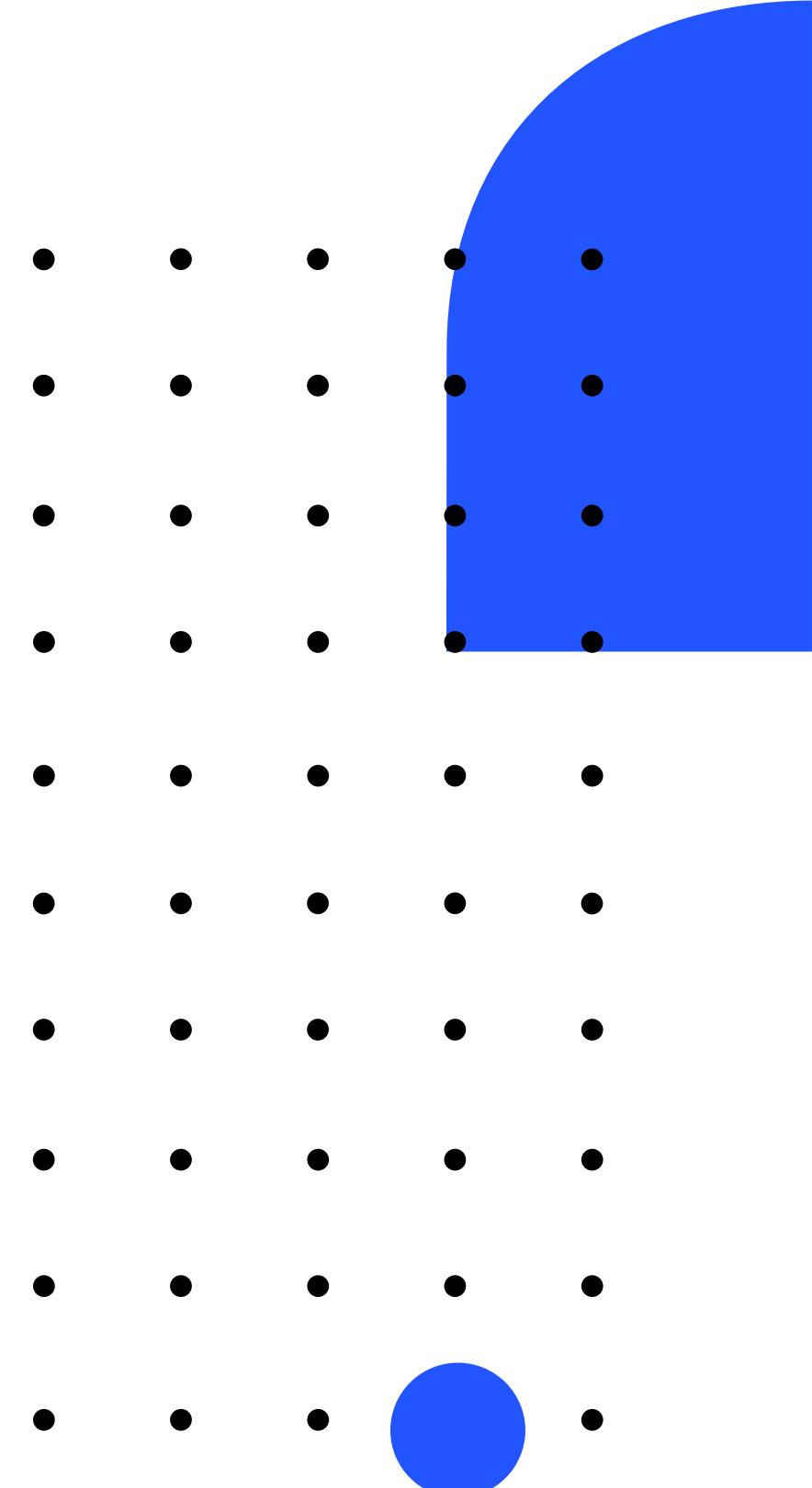
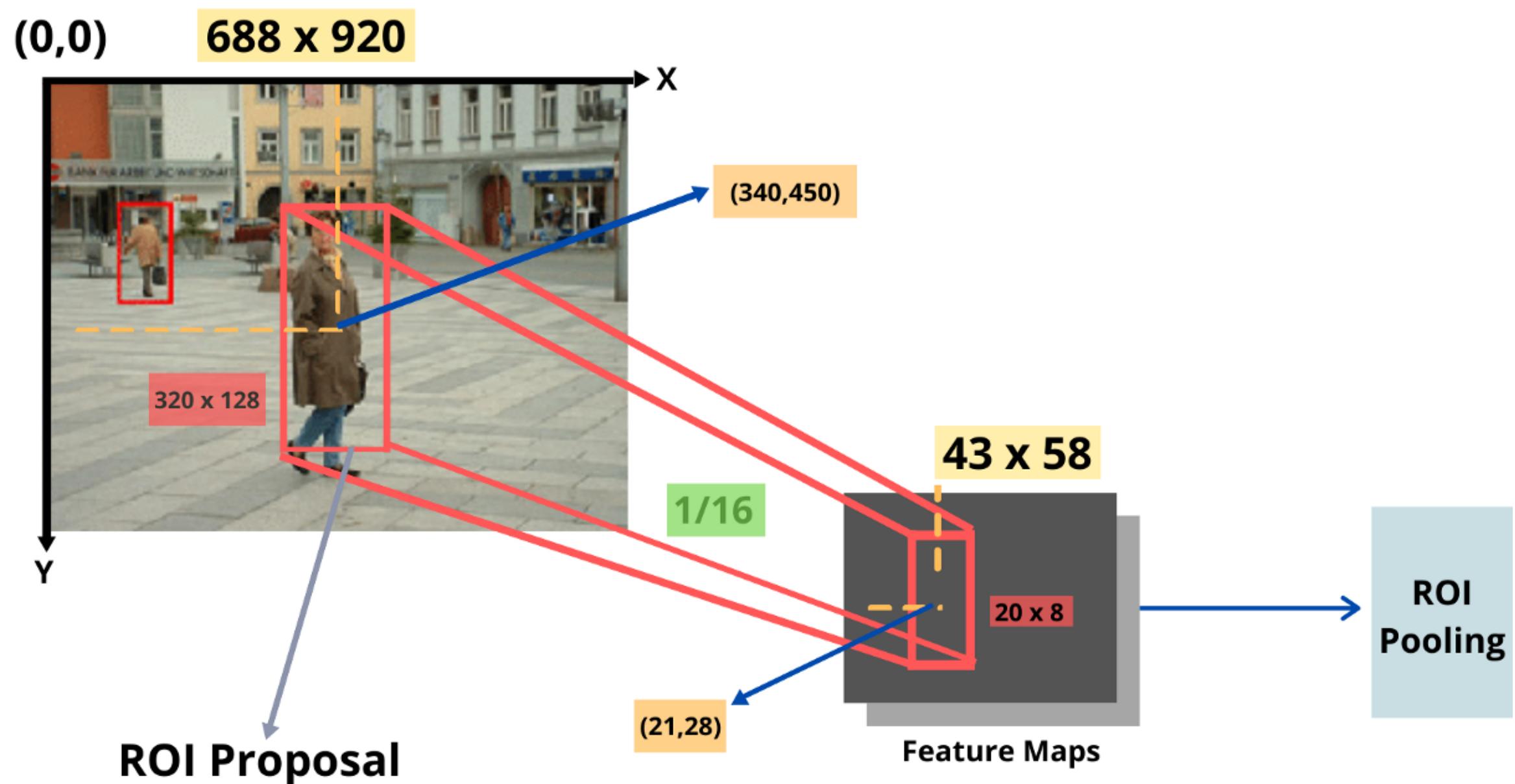
6. Detectron vs YOLO

RCNN vs one shot

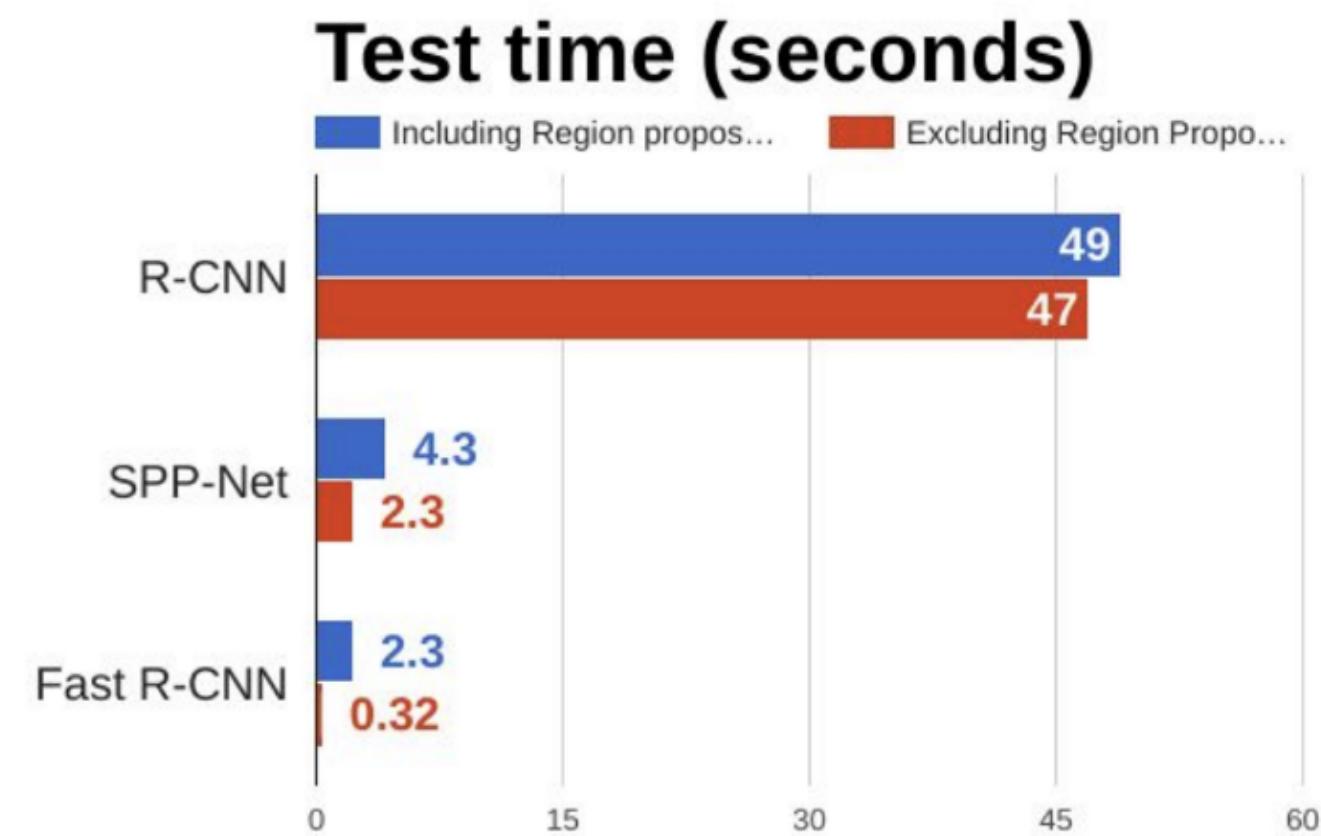
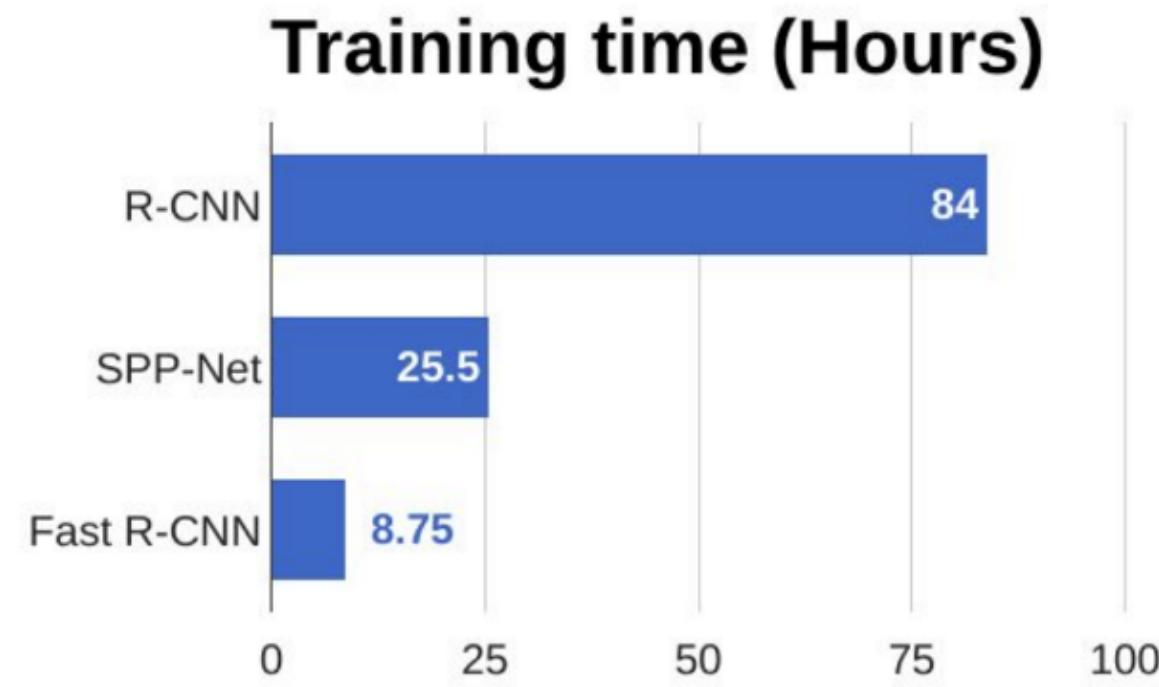
Fast RCNN



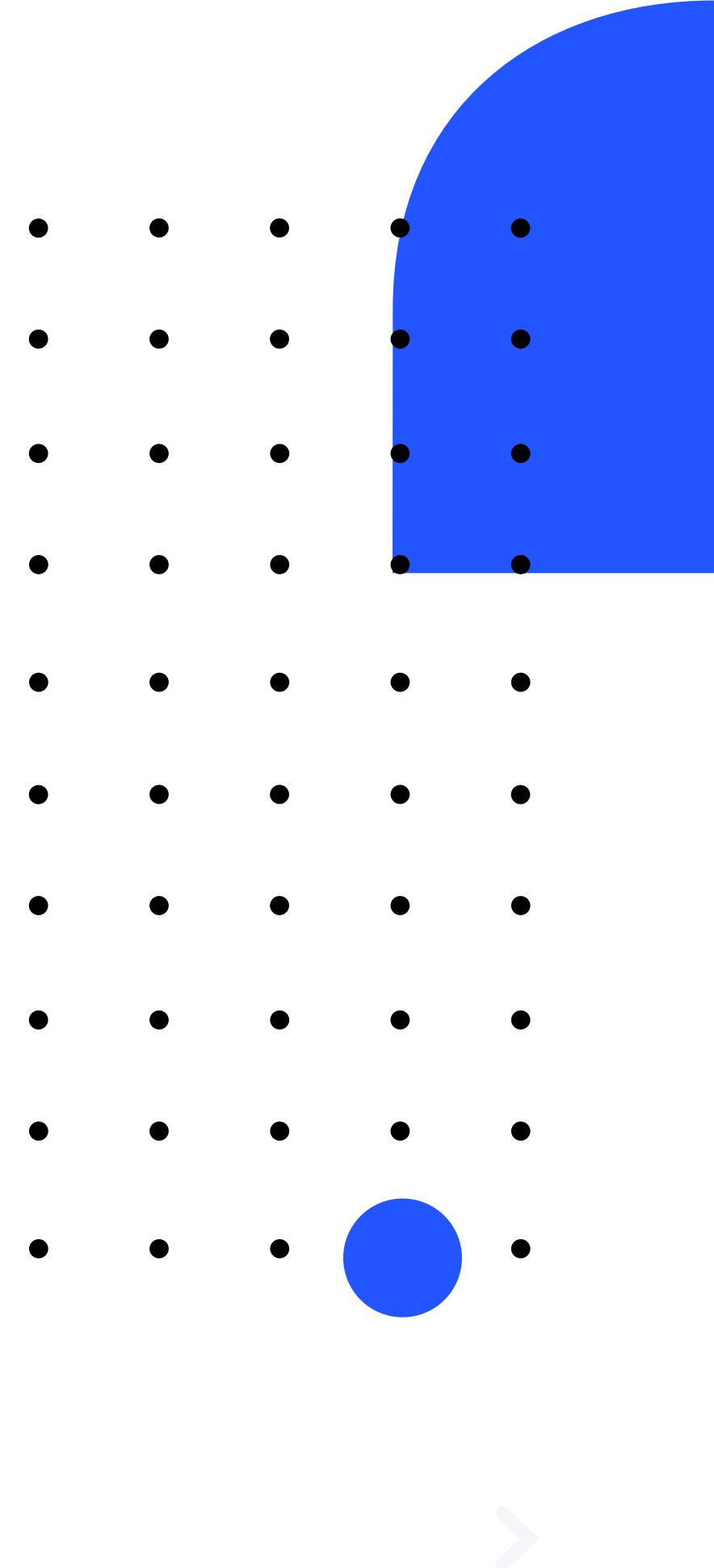
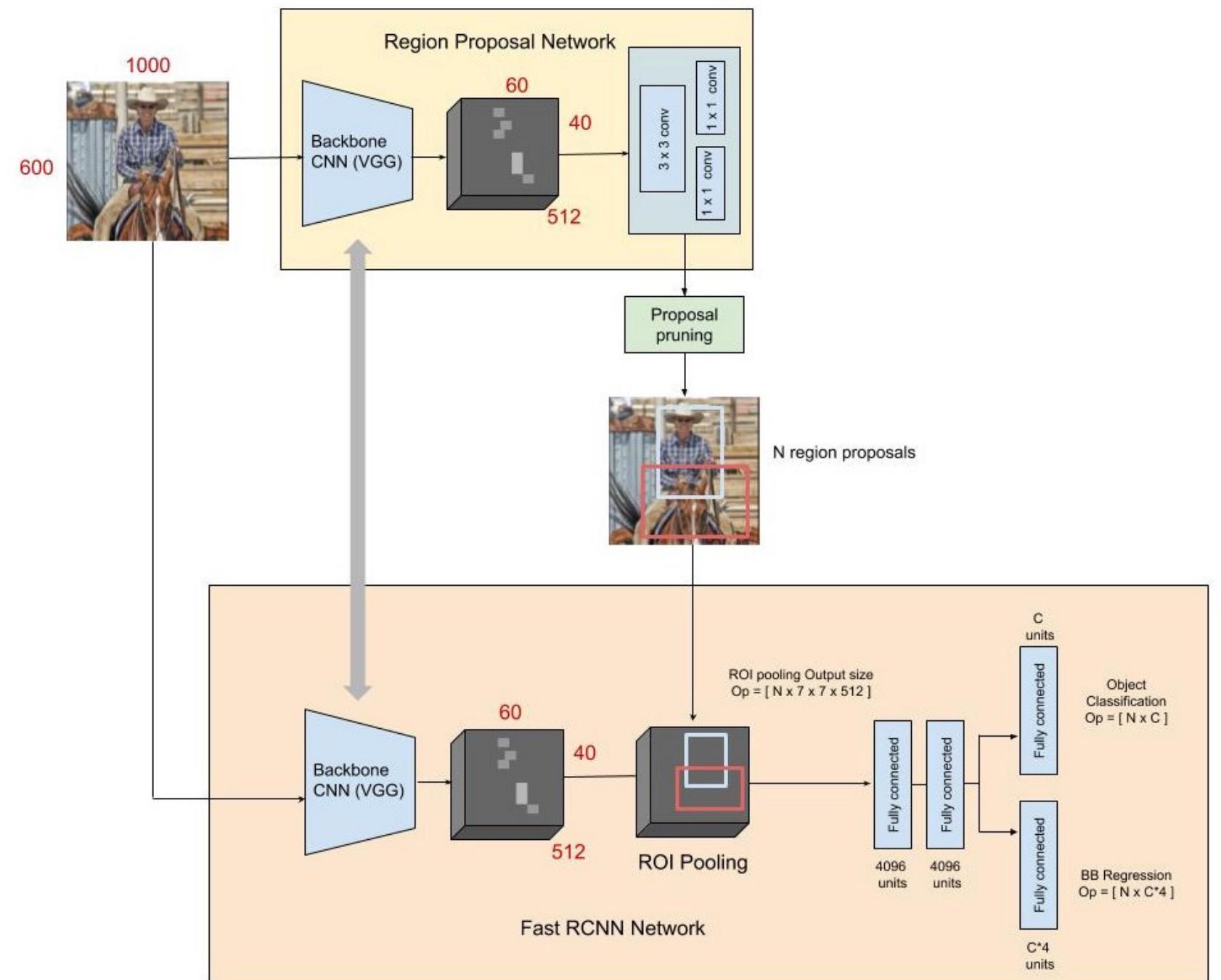
Fast RCNN - regions



Problems of Fast RCNN

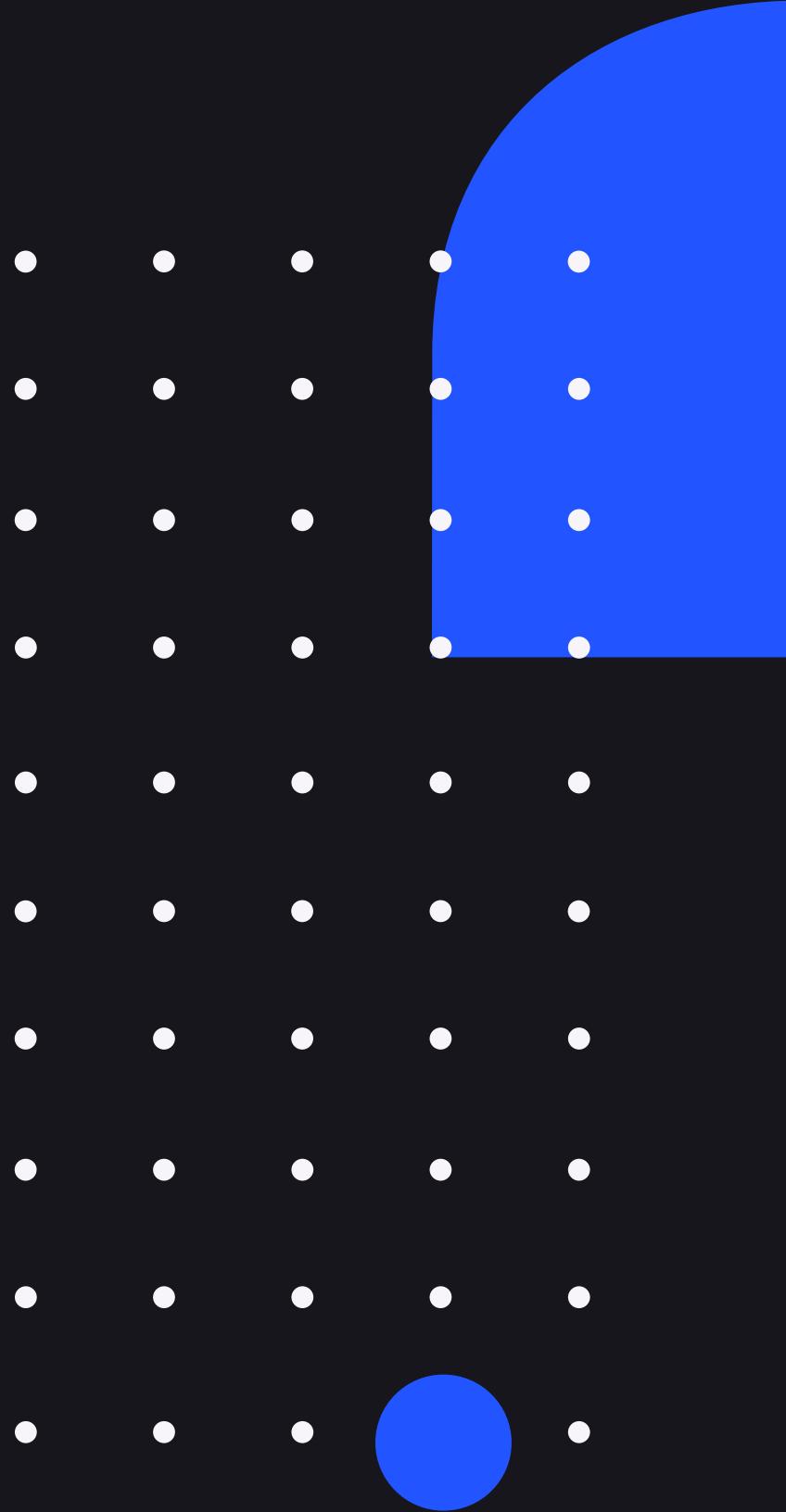


Faster RCNN



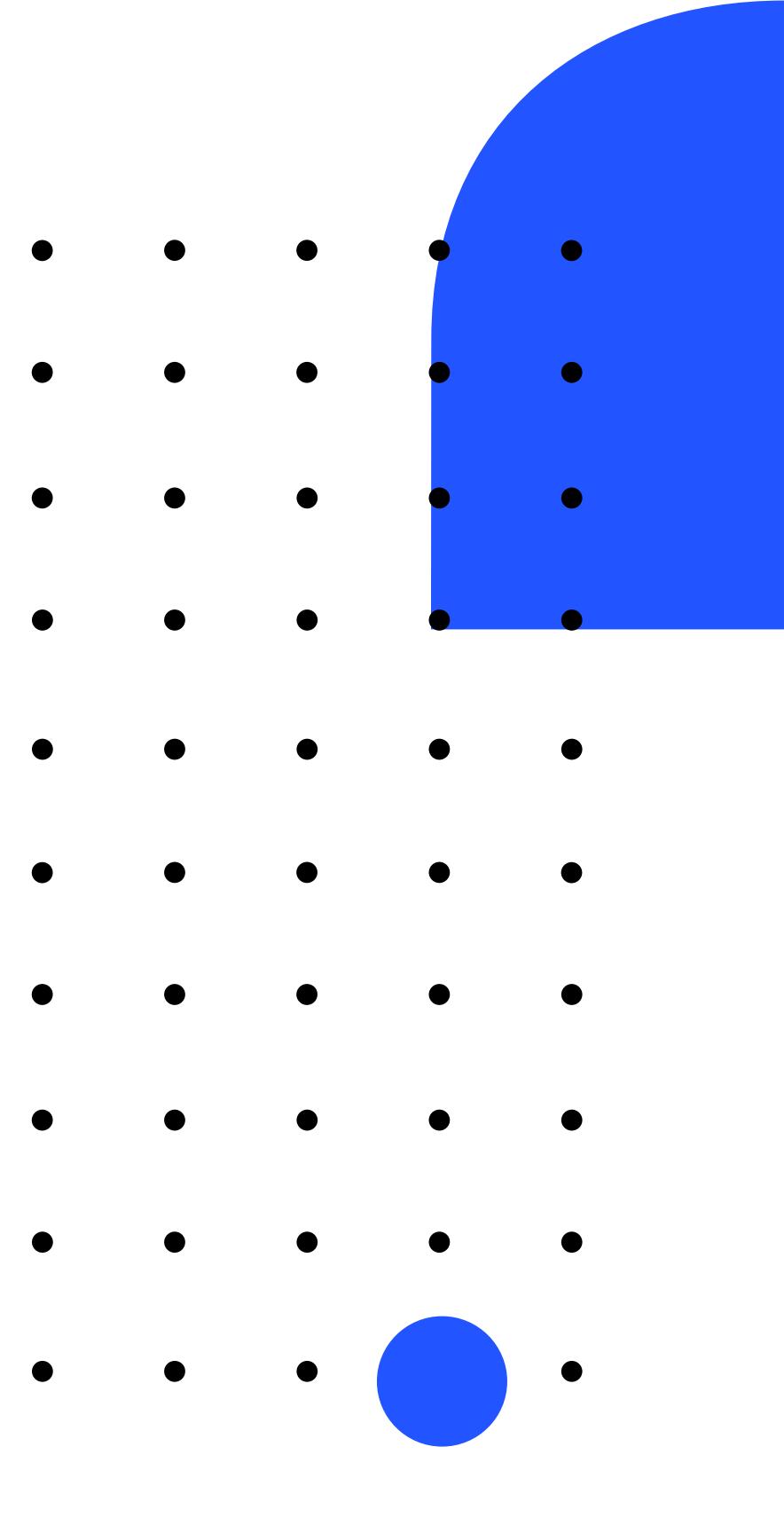
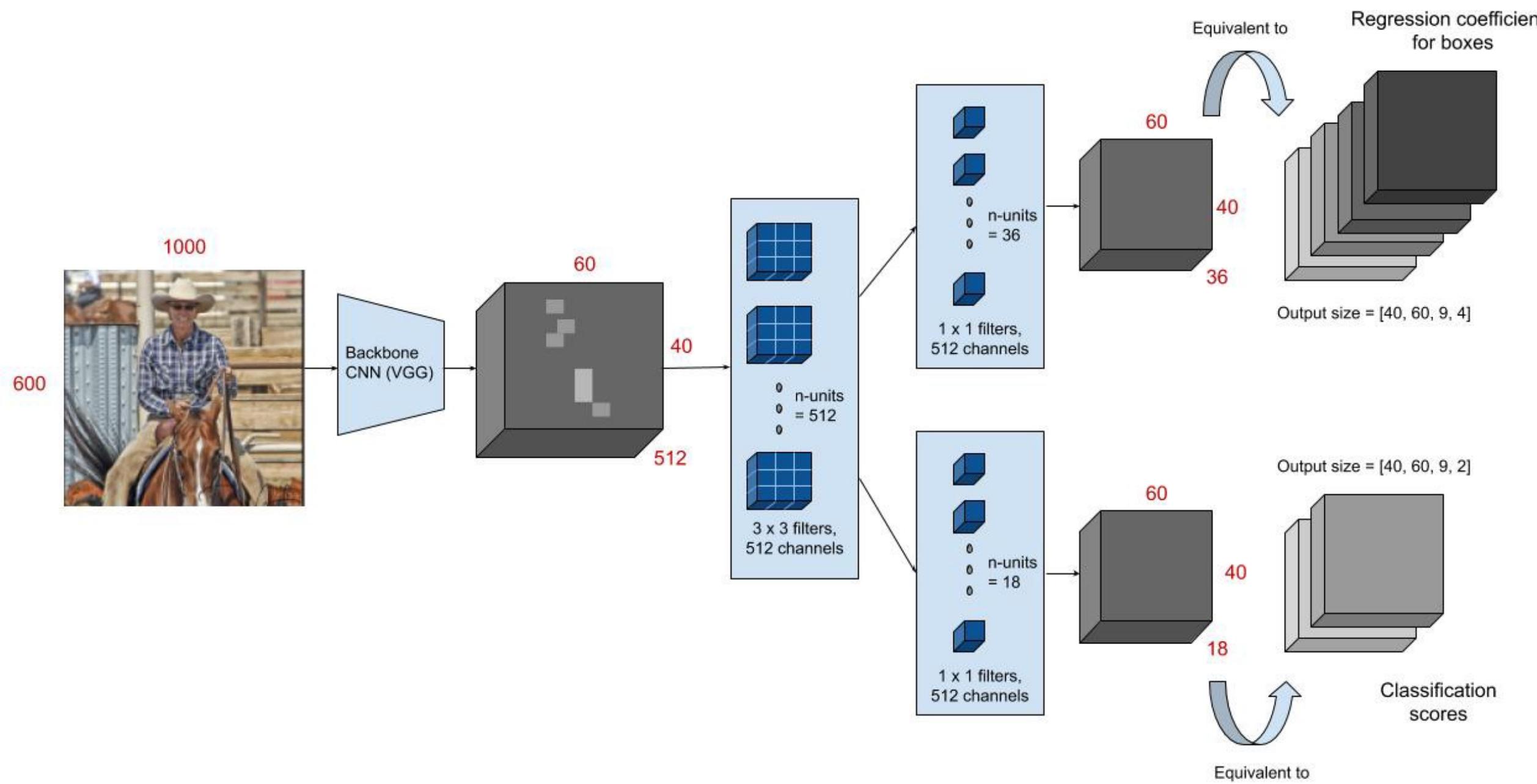
Faster RCNN

- Resize and input the image through a CNN backbone
- Get a feature map of the last layer and build a mapping between the points.
- For every point on FM put anchors on the image and try to predict the existence of an object and its coordinates.

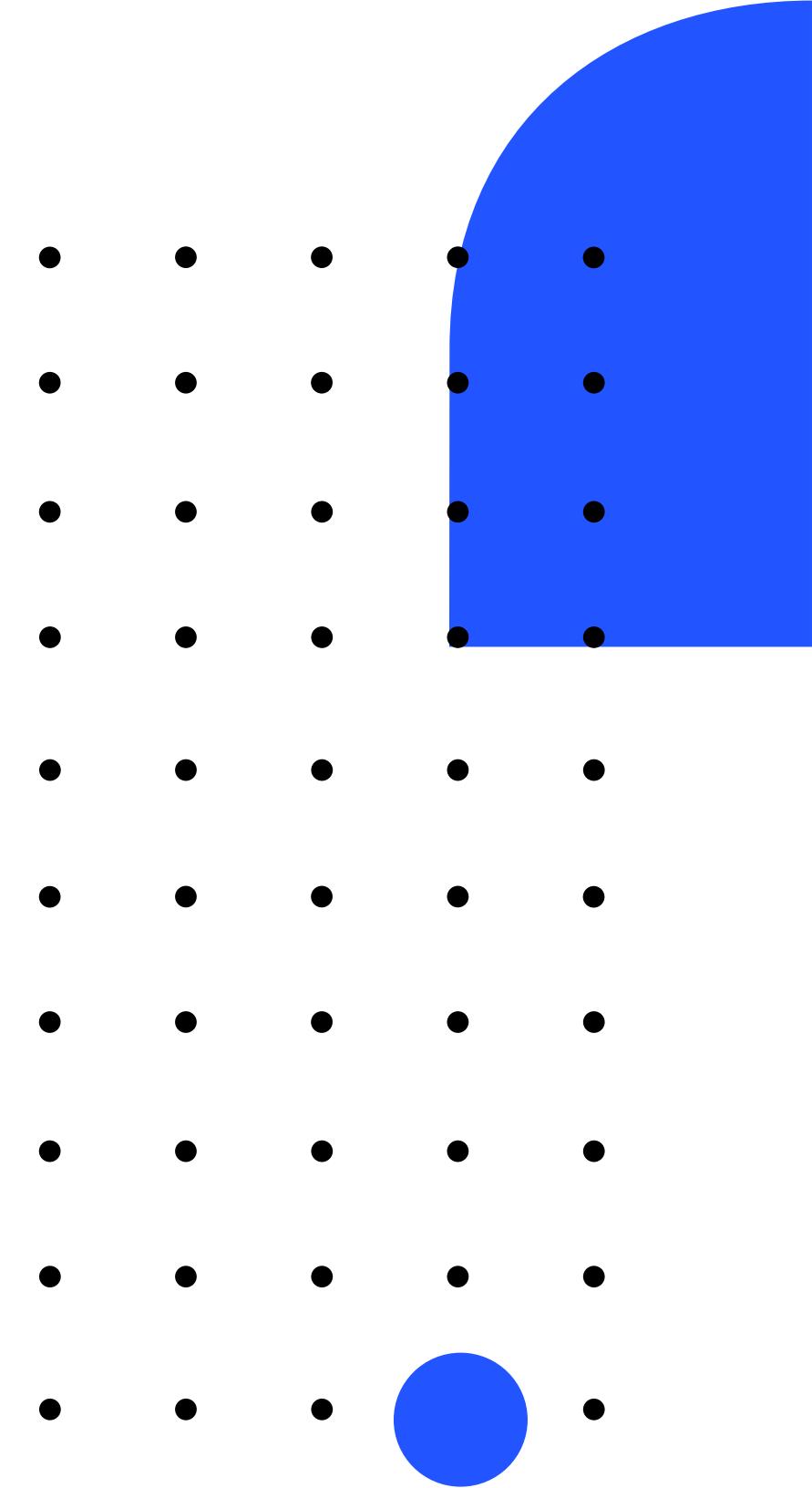
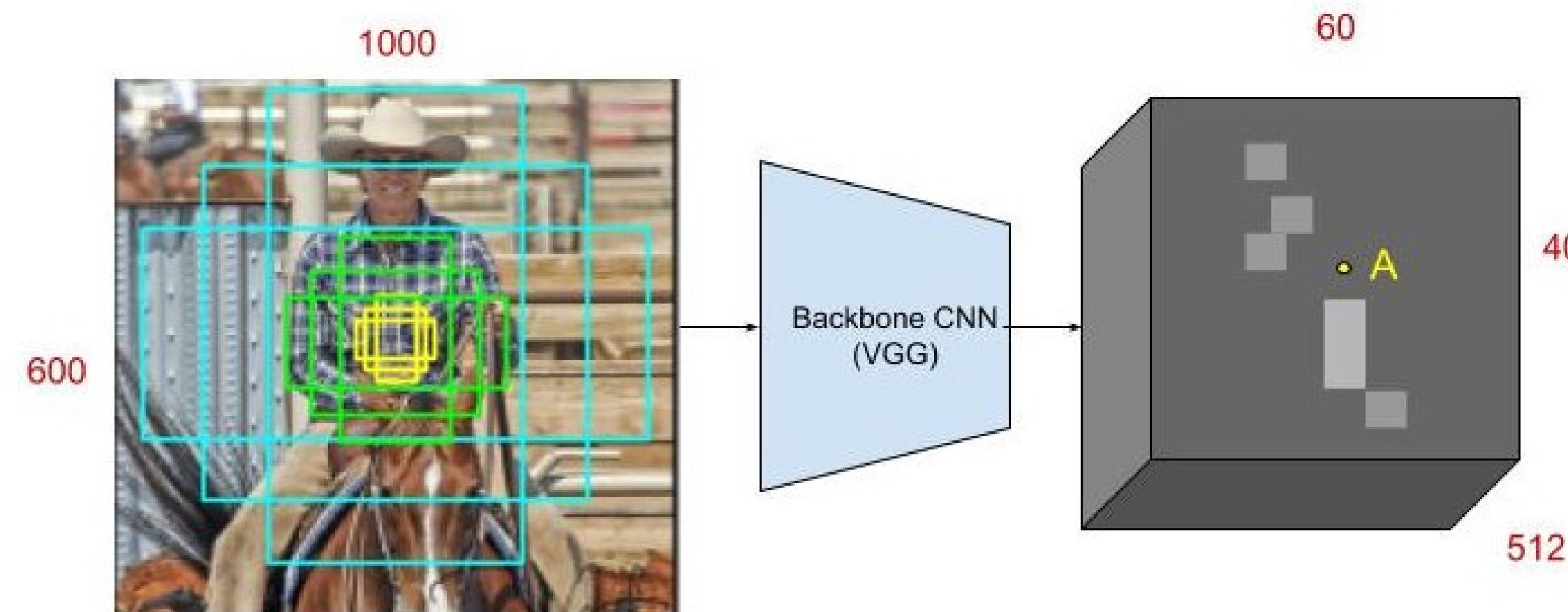


>

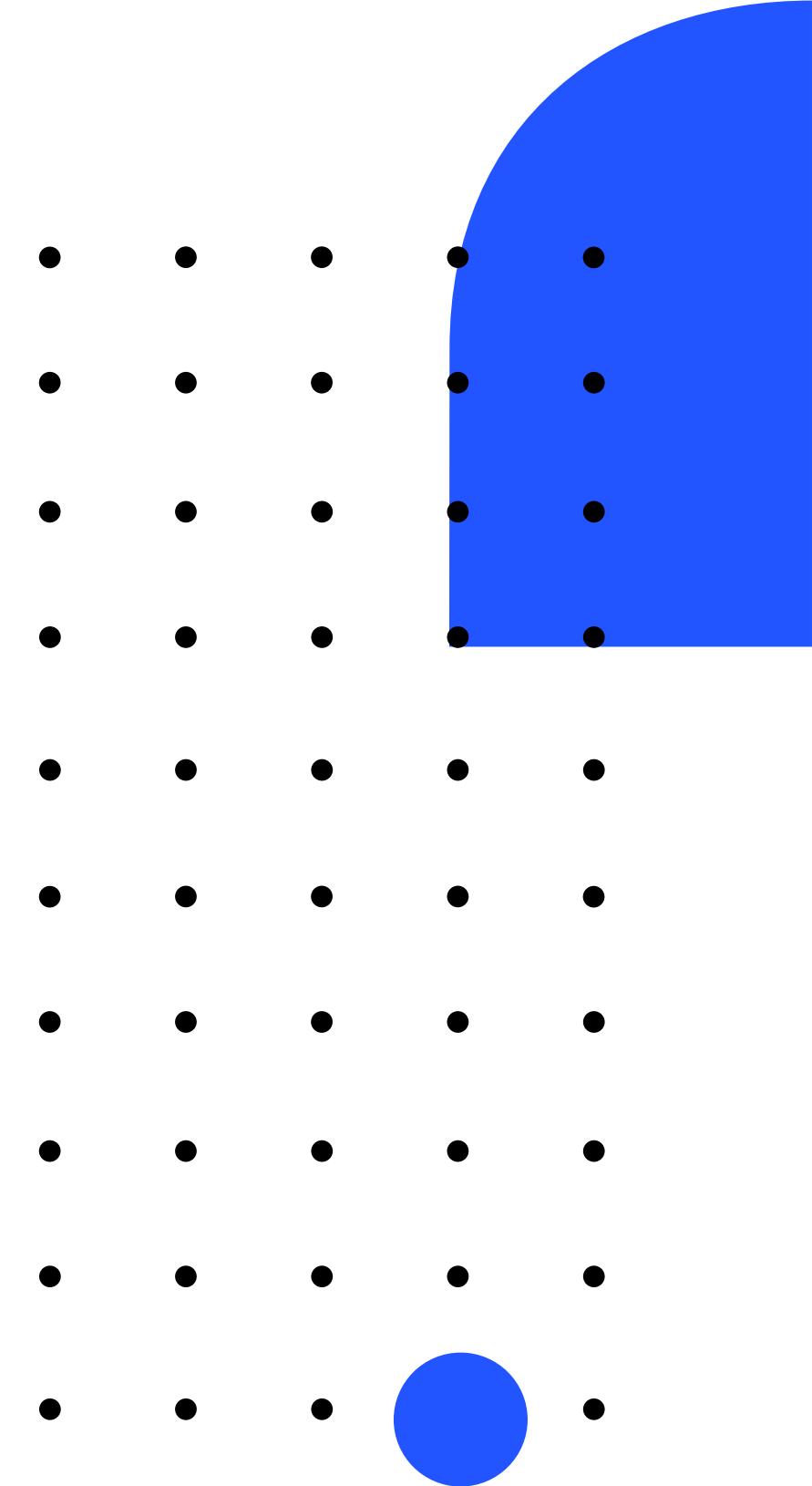
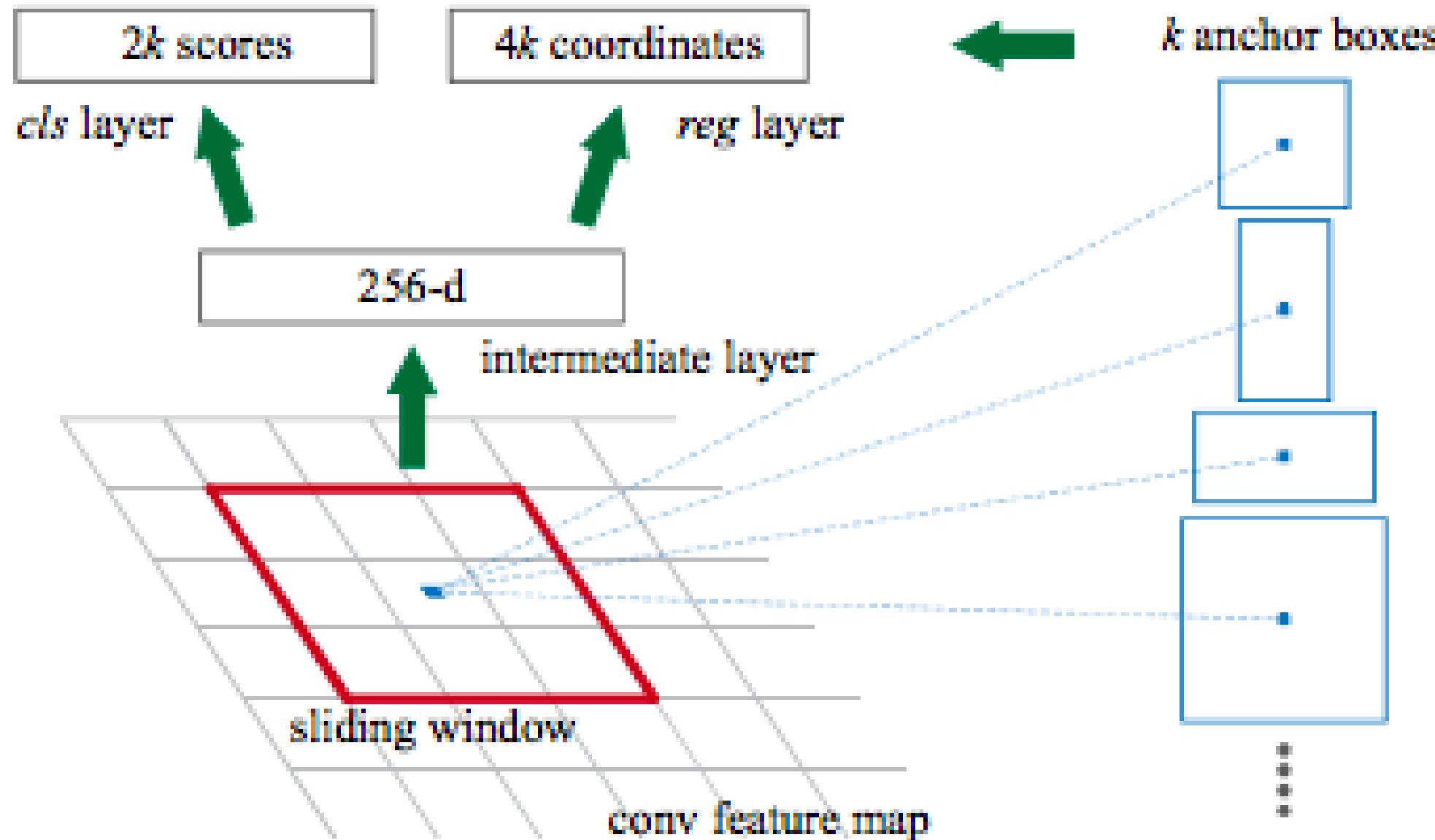
Region Proposal Networks



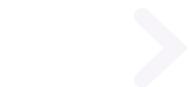
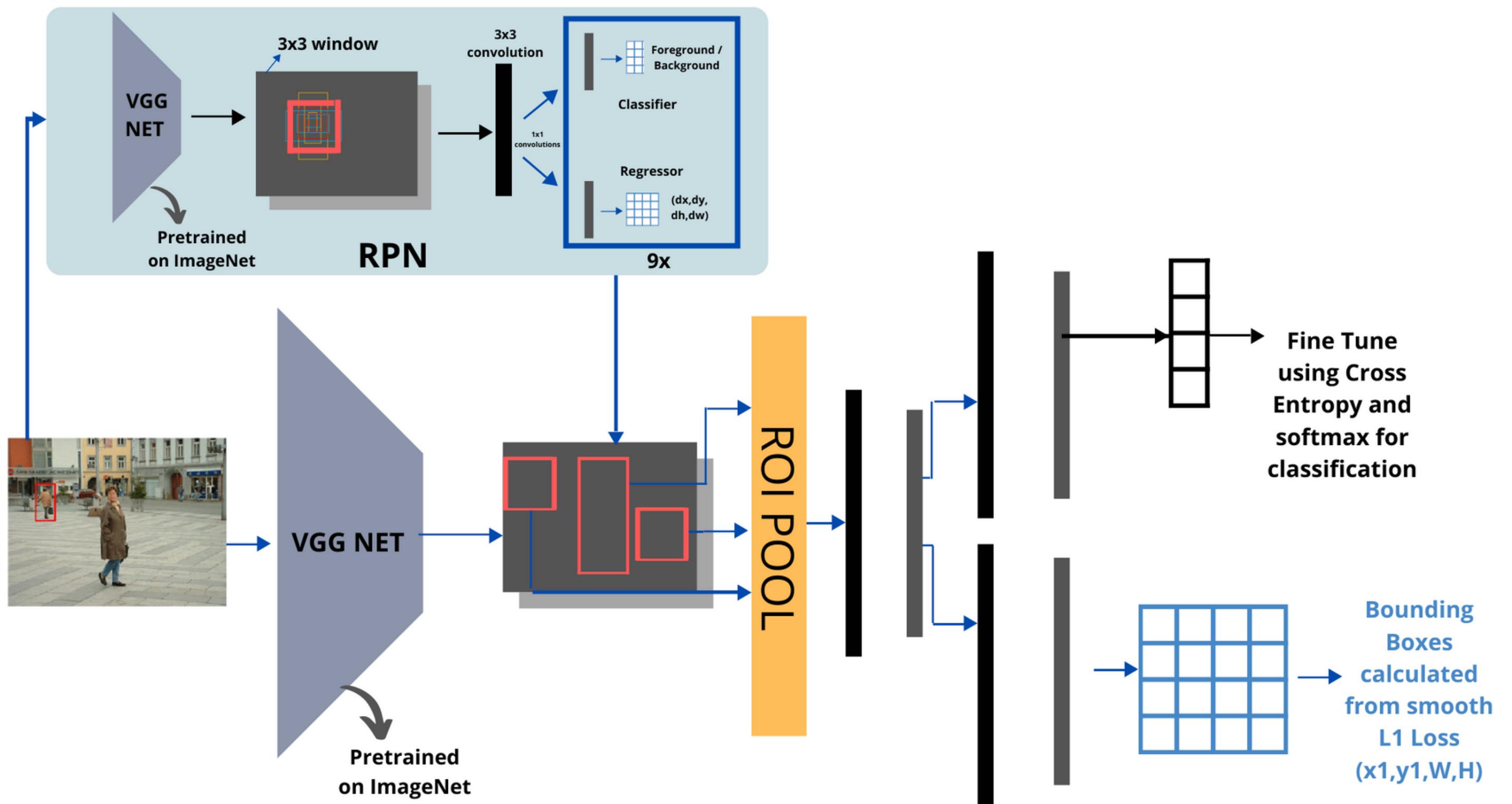
Anchors



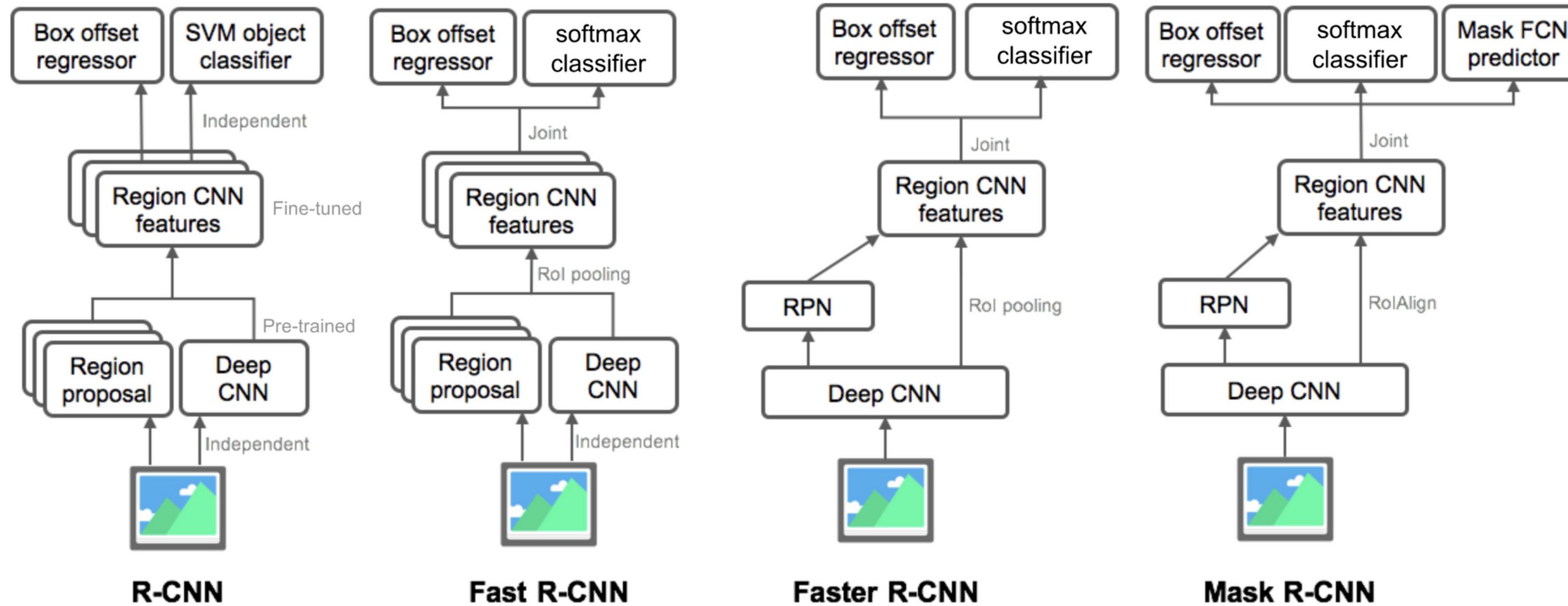
Region Proposal Networks



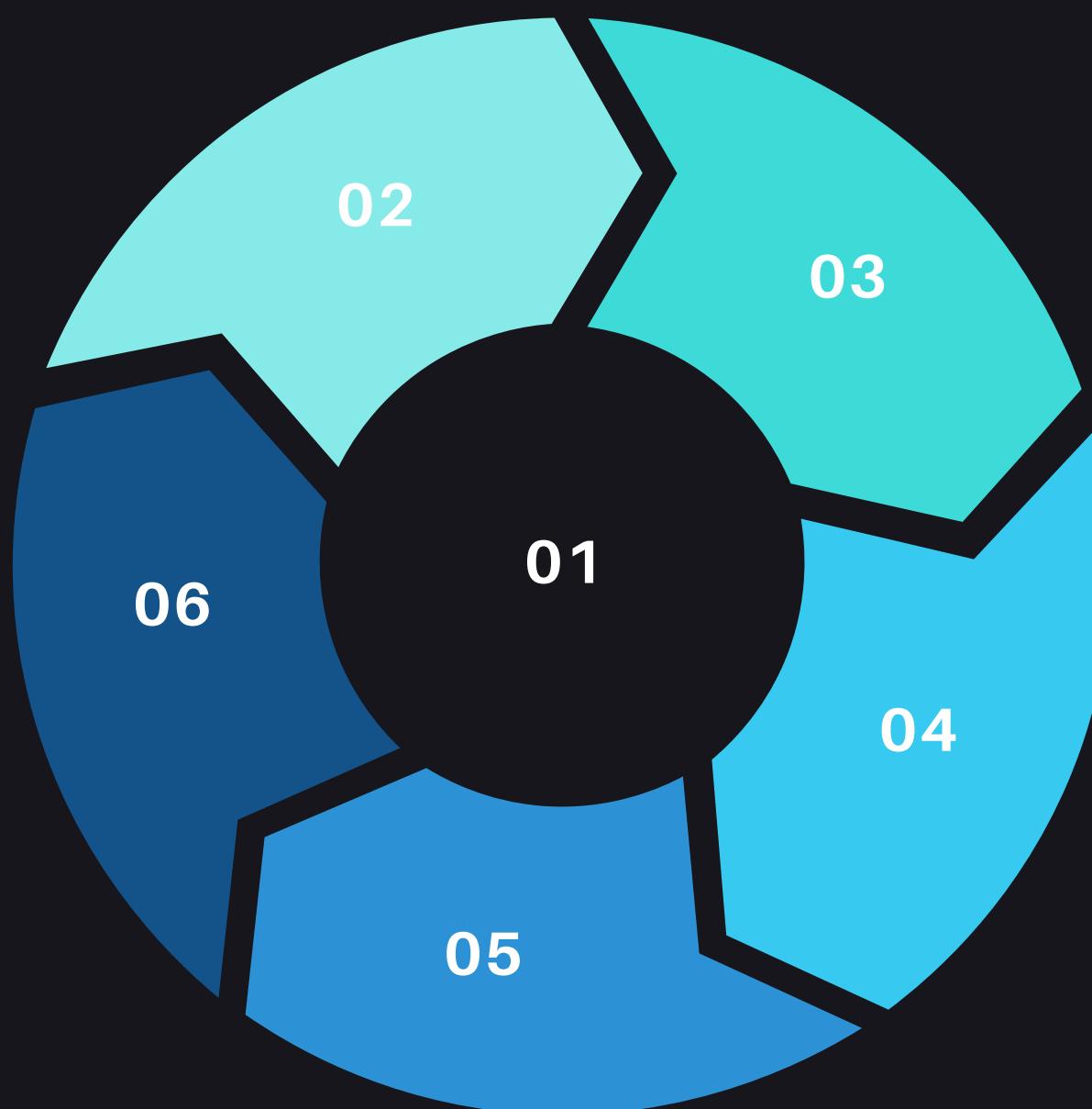
Faster RCNN



RCNN family



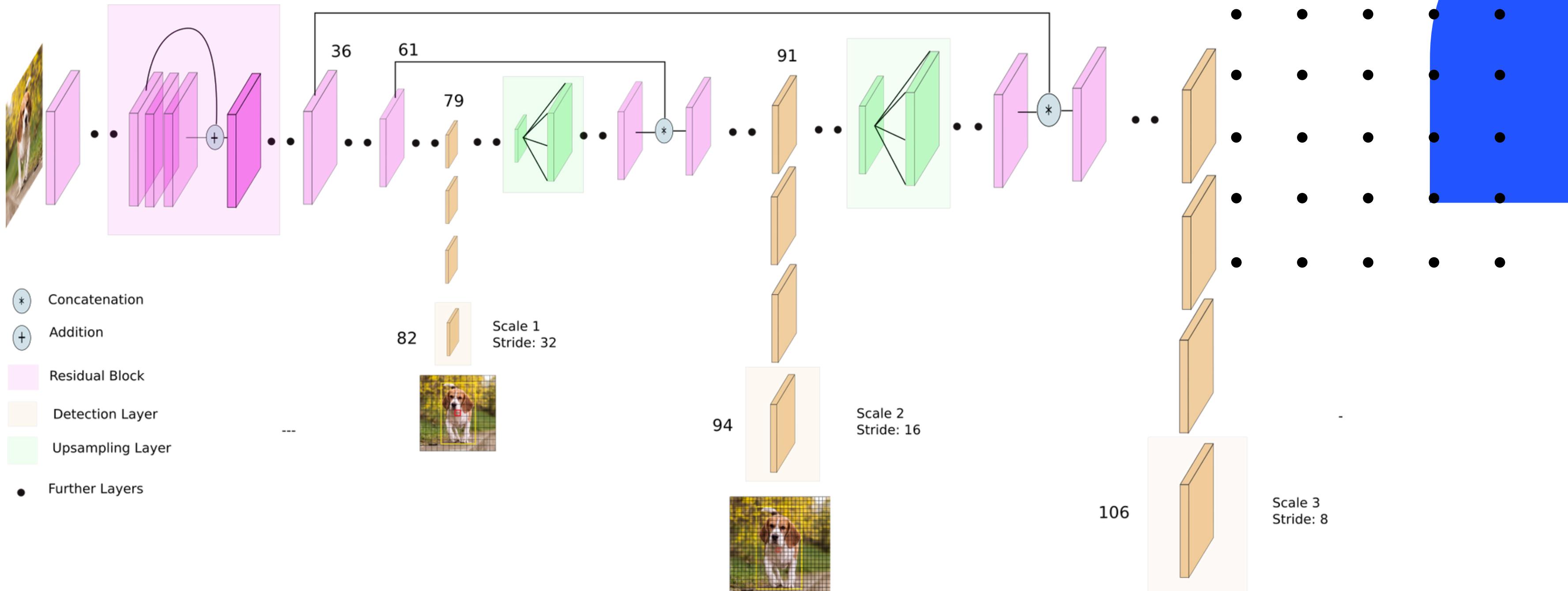
Object Detection

- 
- 1. Object Detection**
predicting the class and location of objects
 - 2. IOU & NMS**
intersection over union and non maximal suppression
 - 3. RCNN**
Region Based Convolutional Neural Networks
 - 4. Fast RCNN**
Reducing the convolutional operations
 - 5. Faster RCNN**
Removing selective search
 - 6. Detectron vs YOLO**
RCNN vs one shot

Detectron

- Detectron is a library made by facebook
- The goal is to implement many computer vision models
- It is written fully in pytorch
- It provides trained models for a variety of tasks such as:
 - Object detection
 - Object segmentation
 - Dense Pose
 - Panoptic Segmentation
 - Landmark detection

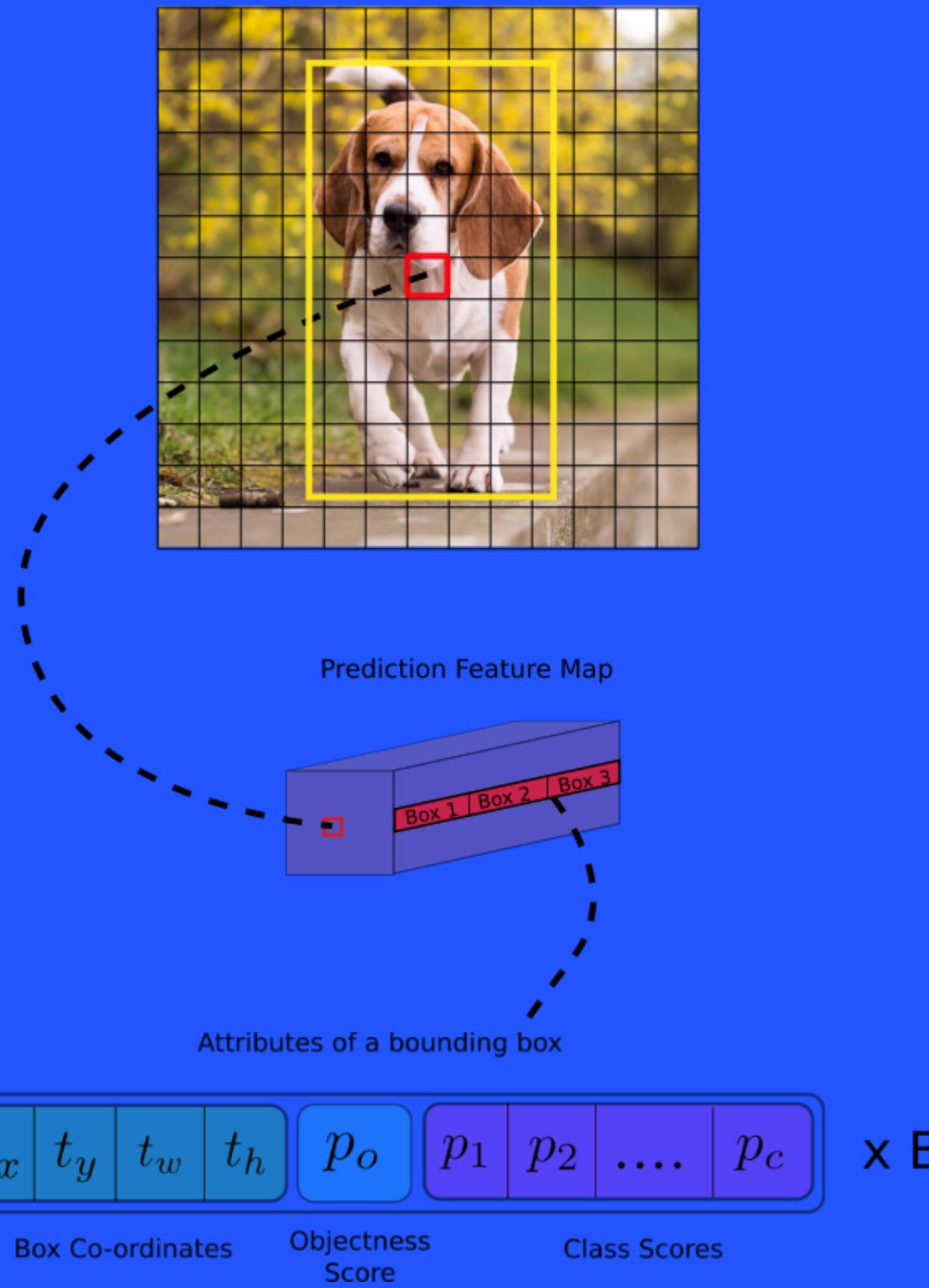




YOLO v3 network Architecture



Image Grid. The Red Grid is responsible for detecting the dog

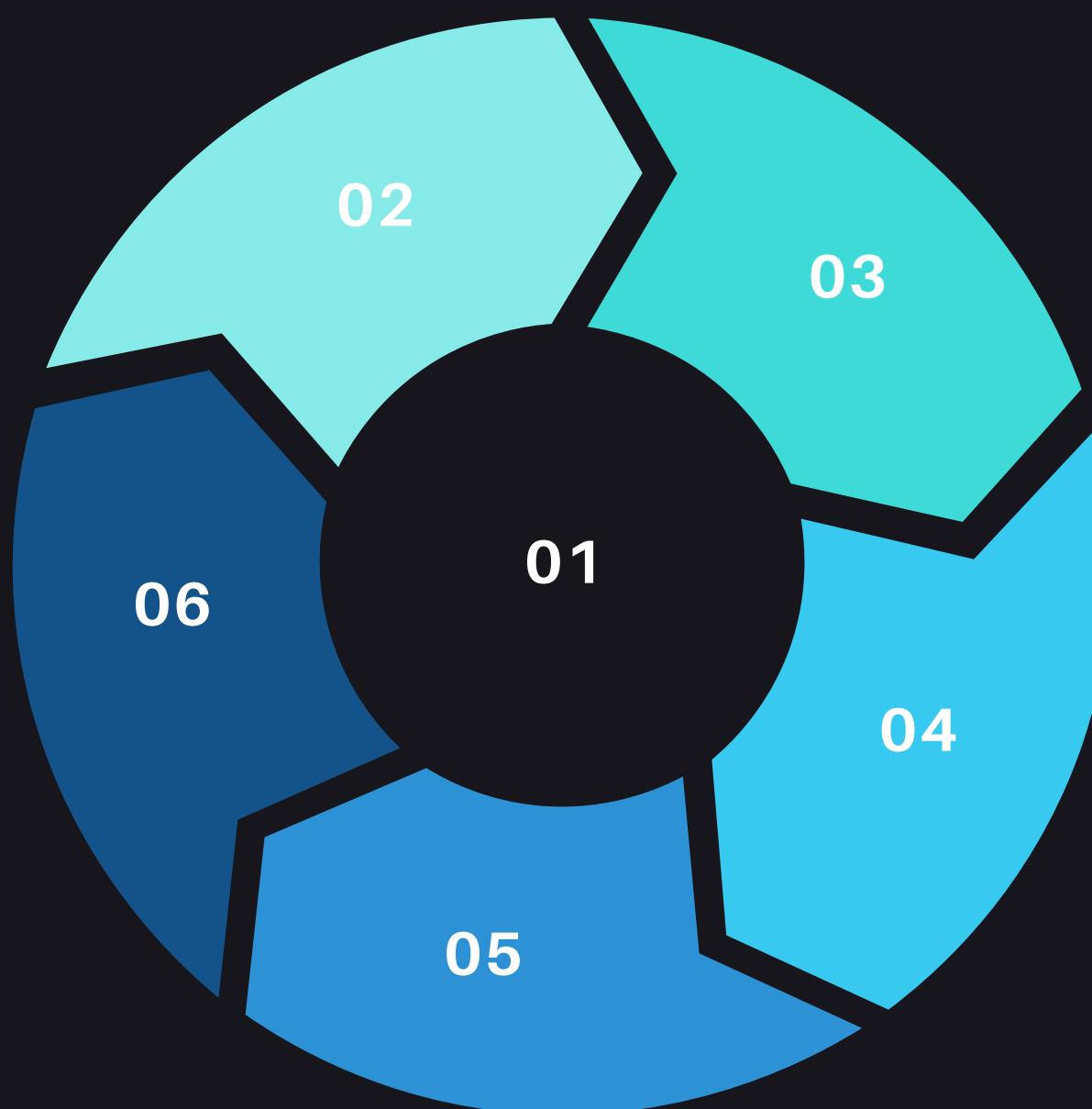


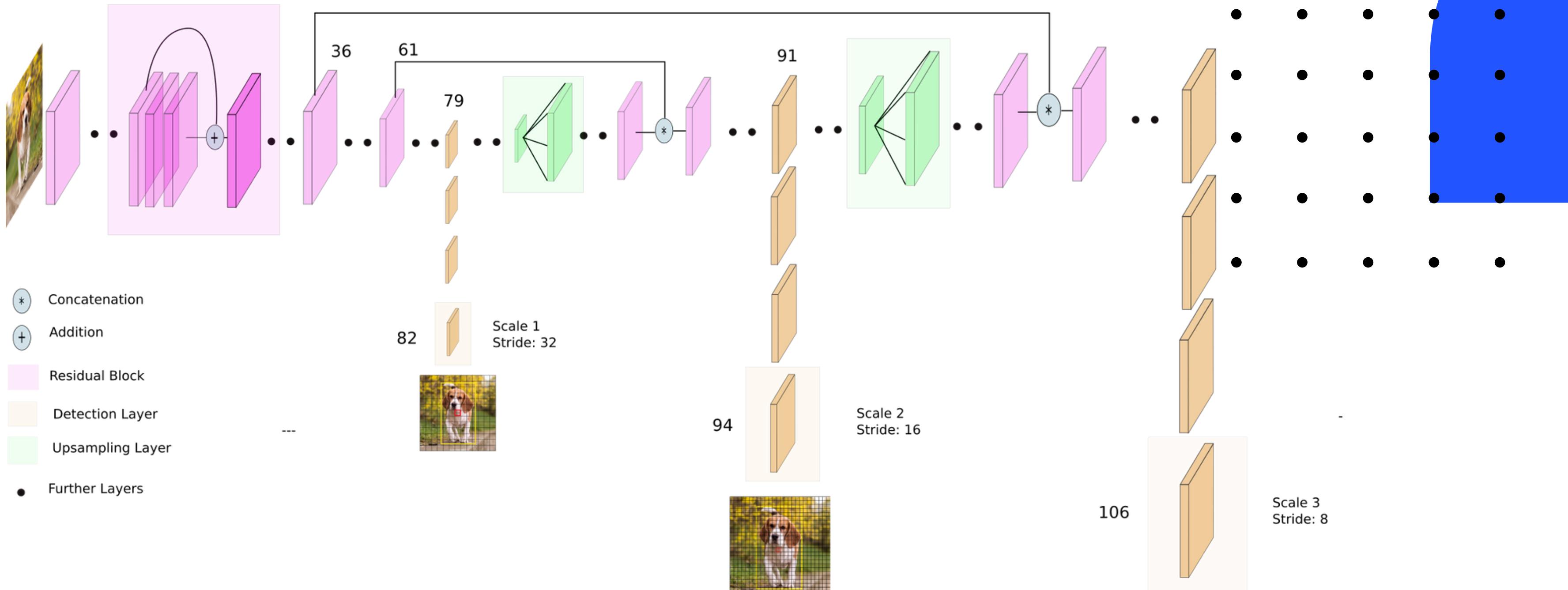
Configuring Yolo vx

- Why do we use the formula
 $\text{filters} = (\text{classes} + 5) \times 3$



Object Detection

- 
- 1. Object Detection**
predicting the class and location of objects
 - 2. IOU & NMS**
intersection over union and non maximal suppression
 - 3. RCNN**
Region Based Convolutional Neural Networks
 - 4. Fast RCNN**
Reducing the convolutional operations
 - 5. Faster RCNN**
Removing selective search
 - 6. Detectron vs YOLO**
RCNN vs one shot



YOLO v3 network Architecture



Ressources

- [Yolov3 explained](#)
- [Dive Really Deep into YOLO v3: A Beginner's Guide](#)
- [YOLOv3 Architecture: Best Model in Object Detection](#)
- [YOLO V3 Explained](#)
- [Advanced Computer Vision with tensorflow](#)