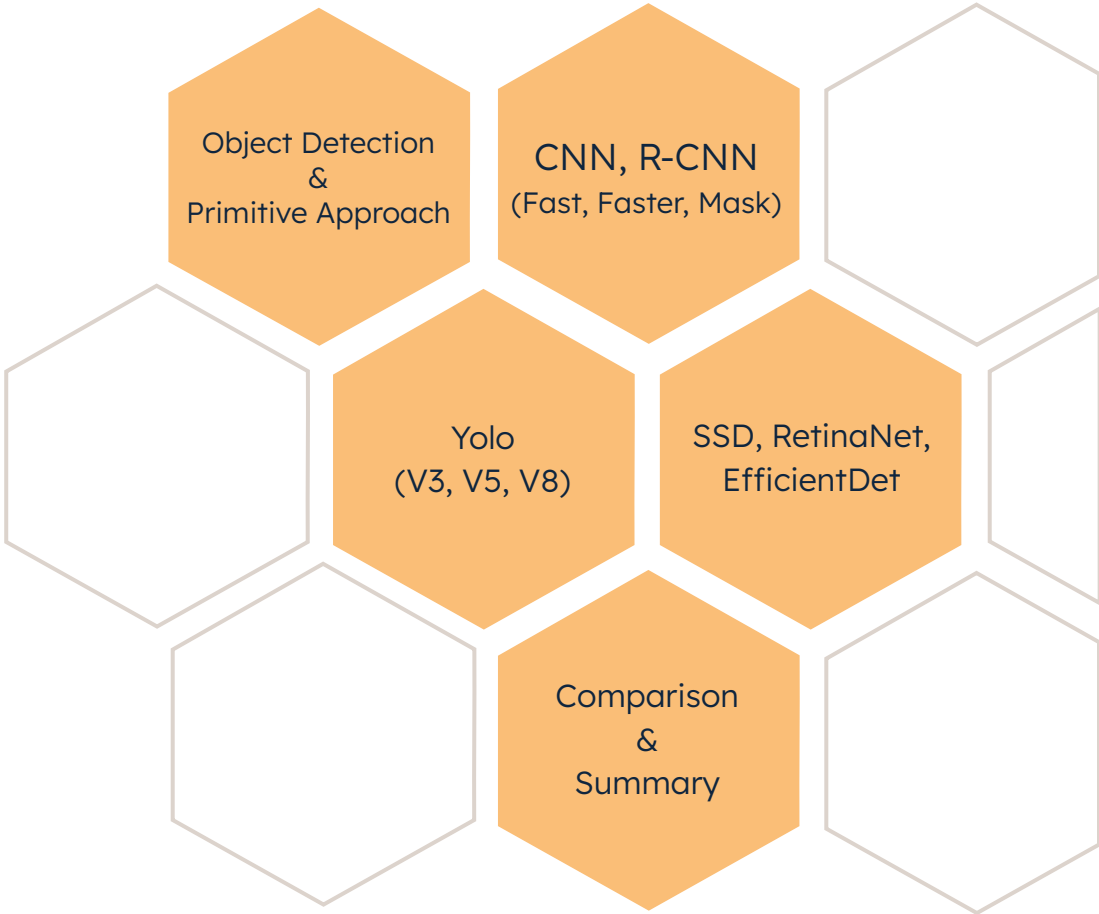
Regions with CNN, RCNNs, Yolos & more: Algorithm, Structure, & Comparison

Presenter's

Nimesh Arora (Team Lead).
Ashish Gurung.
Dikshant Sagar.
Harika Yerramalli.
Ismael Valenzuela.
Kavya Reddy.
Safal Rijal.
Shreyas Teli.

Agenda of the Seminar

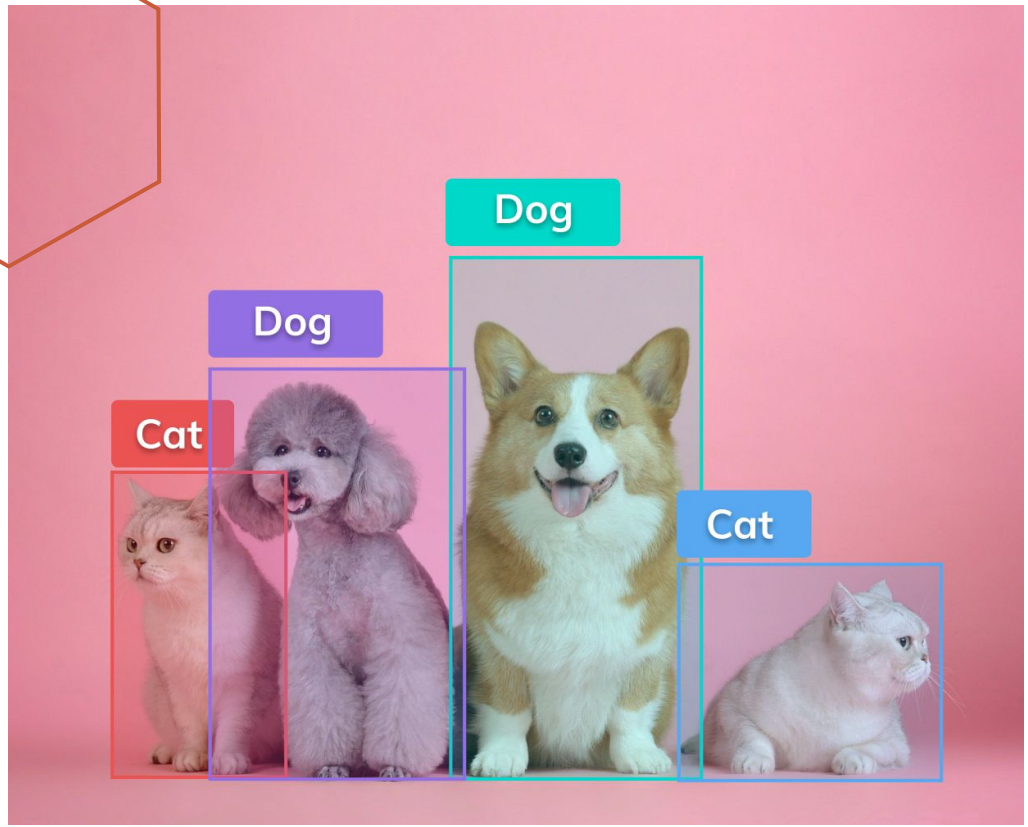


Object Detection

“Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos.”

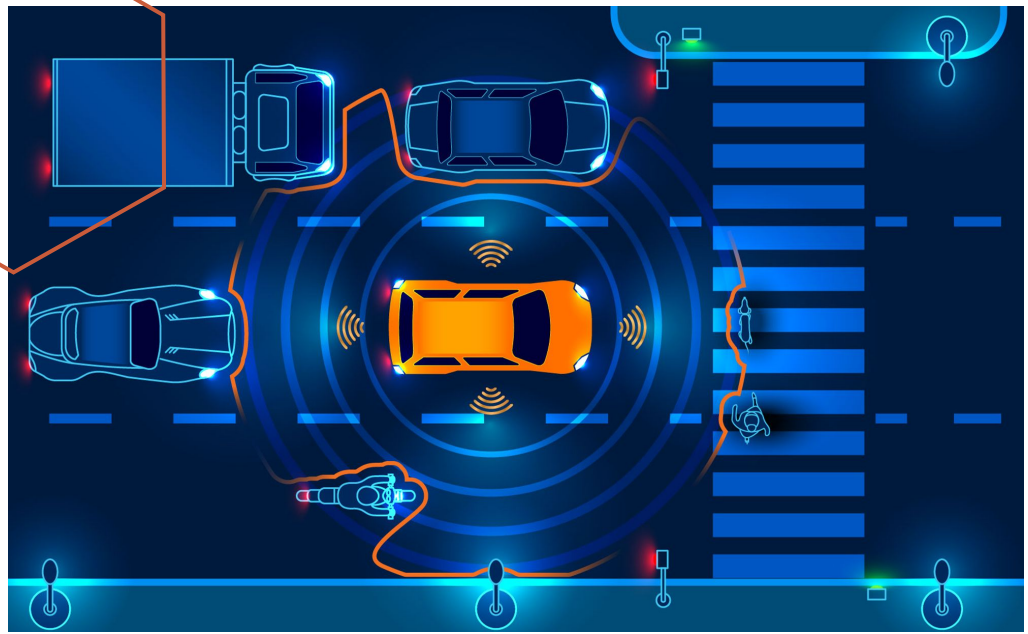
- Wikipedia

In other words,
Object Detection is a computer vision technique that allows us to identify and locate objects in an image or video.



Application
Object Detection

- Autonomous Driving
- Medical imaging
- Surveillance and Security
- Robotics
- Human - Computer interaction



Template Matching

How does this work?

- Select a template image
- Choose a similarity metric
  Squared Differences(SD)
  Normalized Cross Correlation(NCC)
  Correlation Coefficient
- Slide the template across the image
- Thresholding and non-maximum suppression
- Verification and Refinement

Disadvantages

- Sensitive to change in scale, rotation, and lighting.
- Limited to rigid objects
- High Computational Cost
- Limited Accuracy

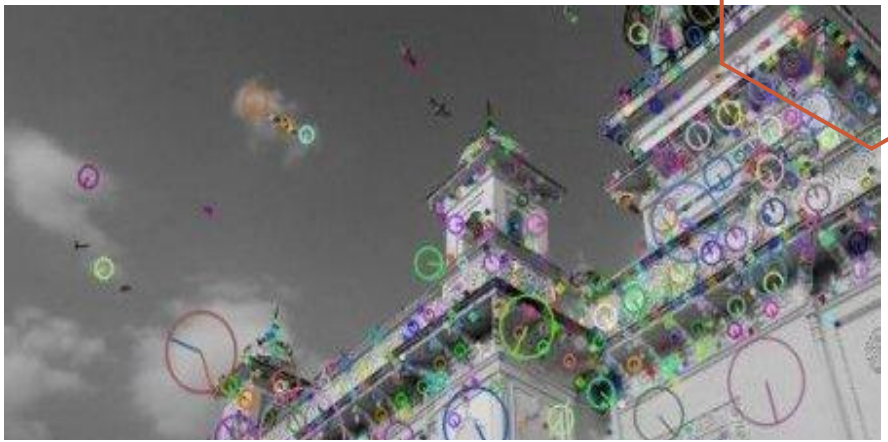
Edge Detection

How does this work?

- Preprocessing with Gaussian Smoothing and Histogram
- Gradient Computation
- Thresholding and non-maximum suppression
- Verification and Refinement

Disadvantages

- Sensitivity to noise
- Limited ability to handle variations
- Lack of contextual information
- Incomplete Edge Detection
- Difficult in setting thresholds



Feature Extraction

How does this work?

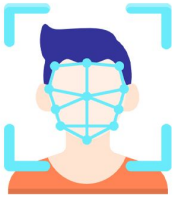
- Preprocessing with Gaussian Smoothing and Histogram
- Feature detection
- Feature Description
- Feature Matching
- Post Processing

Disadvantages

- Limited to hand-picked features
- Sensitivity to variations
- Prone to false matches
- Limited ability to handle complex objects
- Requires manual tuning

Regions with CNN (RCNN), Fast RCNN, Faster RCNN: Algorithm, Structure, Comparison

The Voila-Jones Algorithm



Developed in 2001

Address the challenges of Real-time Face Detection

One of the 1st Algorithm to combine other Algorithms

High Accuracy and Low Computational Cost

Major breakthrough in Object Detection

The Voila-Jones Algorithm

Real-time Face Detection

- Detected faces in real time
- Even in low-power devices
- Major breakthrough

High Accuracy

- Combining Haar Feature with Ada Boost
- Low False Positive Rate

Efficiency

- Highly Efficient
- Low Computational Cost
- Integral images and a cascade of classifiers.

Scalability

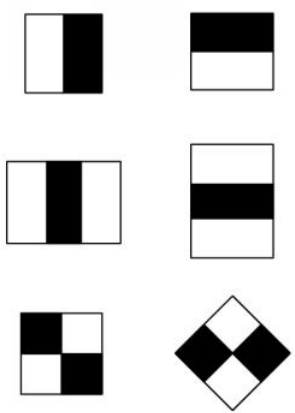
- Large dataset with 1000's of images
- Cars, Pedestrians, Animals Detection

Inspiration for future research

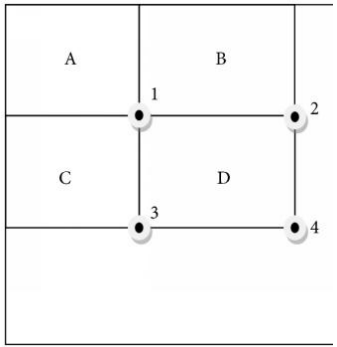
- Deep Learning
- Convolutional Neural Network
- Combining Simple features Machine Learning Algorithm

How did this Algorithm Work?

Haar-like Features

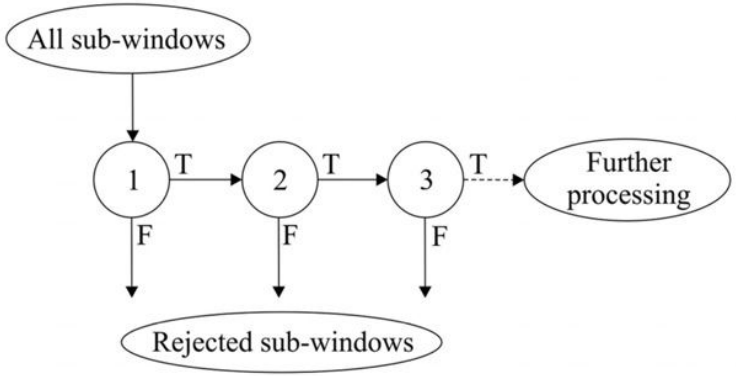


Integral Images



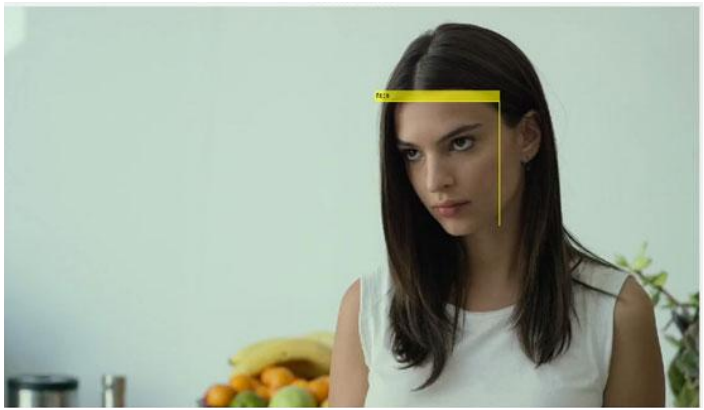
AdaBoost Learning

Attentional Cascade



Limitations

The Voila-Jones Algorithm



- Limited to detecting Rectangular objects
- Limited to detecting objects with fixed sizes
- Sensitive to changes in lightning and orientation
- Requires a large amount of training data
- Can be computationally expensive.

Convolutional Neural Network(CNN)

What’s in it for you?

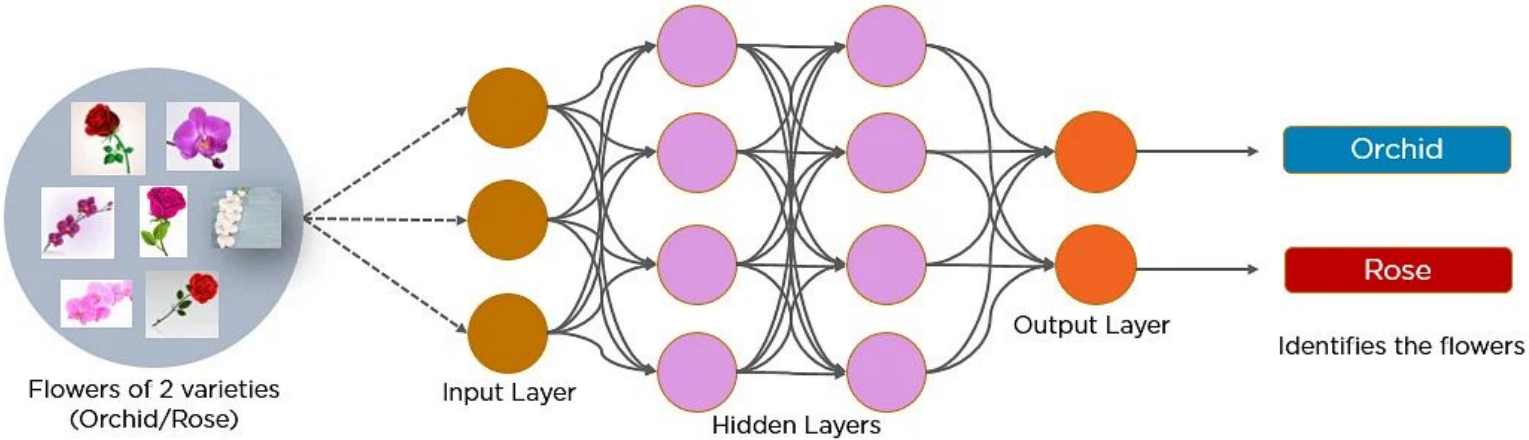
Do you know how Deep Learning recognizes the objects in an image?

It does using a Convolutional Neural Network

- Introduction to CNN?
- What is CNN?
- CNN architecture
- Applications of CNN
- Advantages and Disadvantages

Introduction to CNN

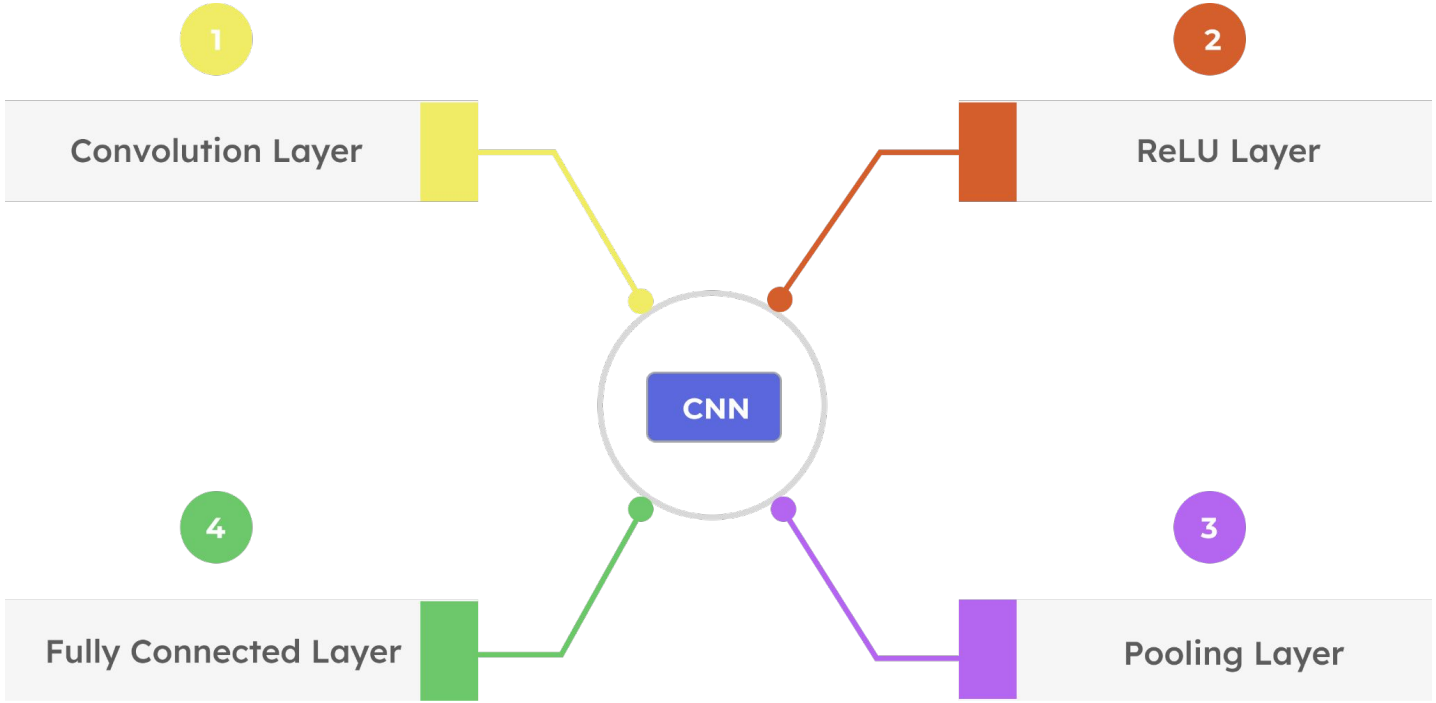
- CNN is a feed neural network that is generally used to analyze visual images by processing data with grid like topology. A CNN is also called as “ConvNet”



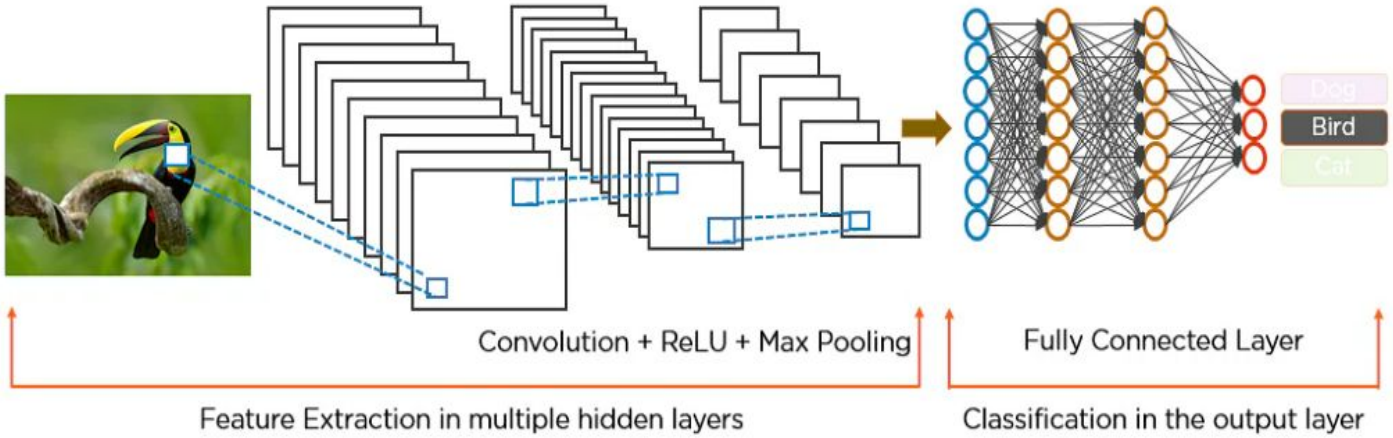
CNN Origin

- Convolutional neural network's were first developed by postdoctoral computer researcher Yann LeCun. It was built to recognize handwritten digits which was named as LeNet.
- Lecun is now the chief AI scientist at META(often abbreviated as FAIR) and is also a professor of NYU
- The most important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback at that period, it worked only on low resolution images and it wasn't until Alexnet was released.

CNN Operation

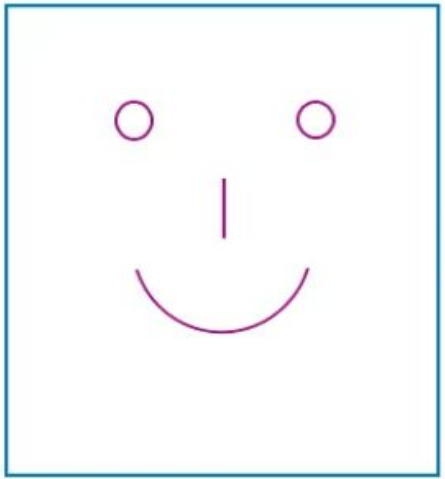


CNN Architecture

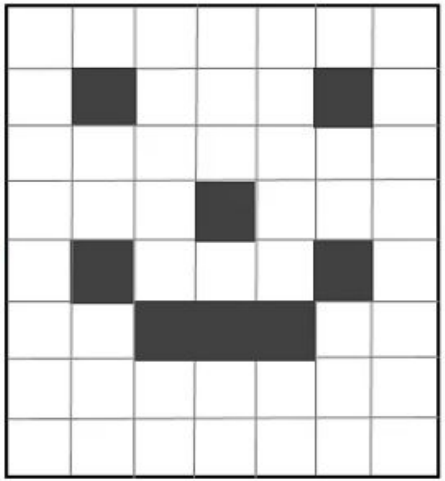


CNN Architecture

Here is another example to depict how CNN recognizes an image:



Real Image



Represented in the form of black and white pixels

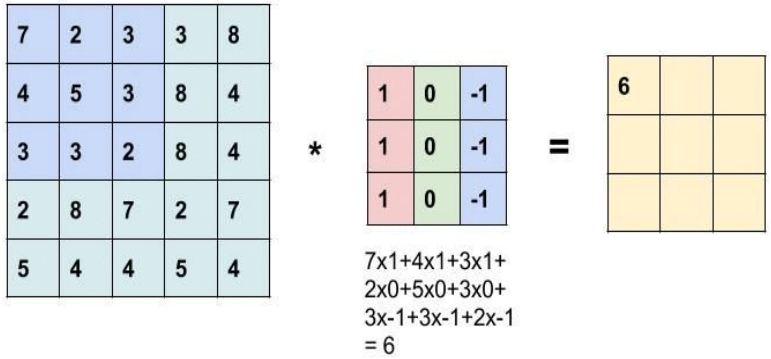
0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Image represented in the form of a matrix of numbers

Convolution:

- Convolution reduces spatial dimension only. CNN can be used to increase channels too based on number of filters/kernels.

- In convolution, we can reduce the no.of features but on the other hand, we keep the features intact. convolution is in charge of executing convolution operations.
  Example: We are dealing with images



Input Image Kernel/Filter Feature Map

Kernel:

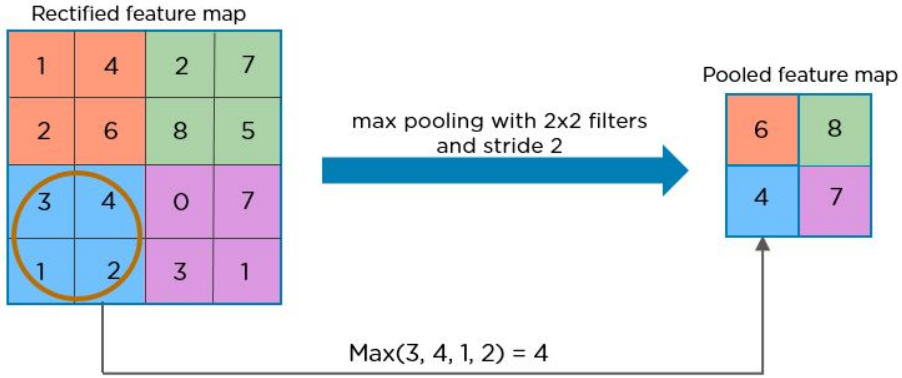
- Here we apply the kernel/filter as the component in this layer that performs convolution operation(matrix). Until the complete image is scanned. The kernel makes horizontal and vertical adjustments depending on the stride rate. The kernel is lesser in size but it has more depth. This means if the image has 3 RGB channels, the kernel height and width will be modest spatially but the depth will span all the three channels.

Activation Layer:

- Another important part of the convolution layer known as Non- Linear activation function. The output of the convolution layer passed through the non linear activation function. The most commonly used non-linear activation function is ReLu. f(x) = max(0,x). It is a threshold activation layer. ReLu can never be negative. If we get negative values, the negative values will be zero.

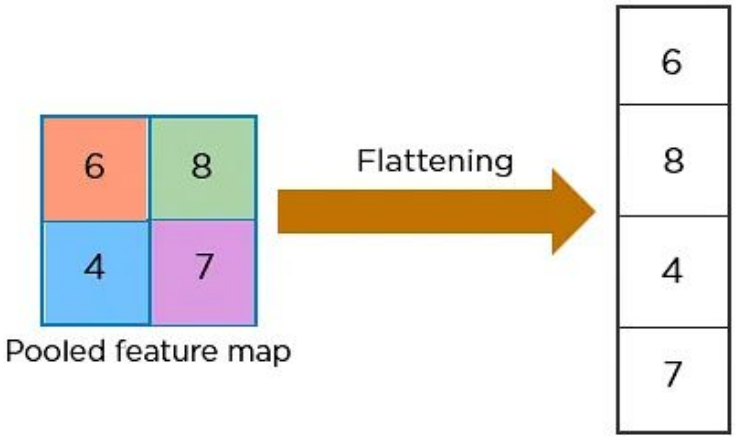
Pooling Layer:

- The rectified feature map now goes through a pooling layer. Pooling is a down-sampling operation that reduces the dimensionality of the feature map

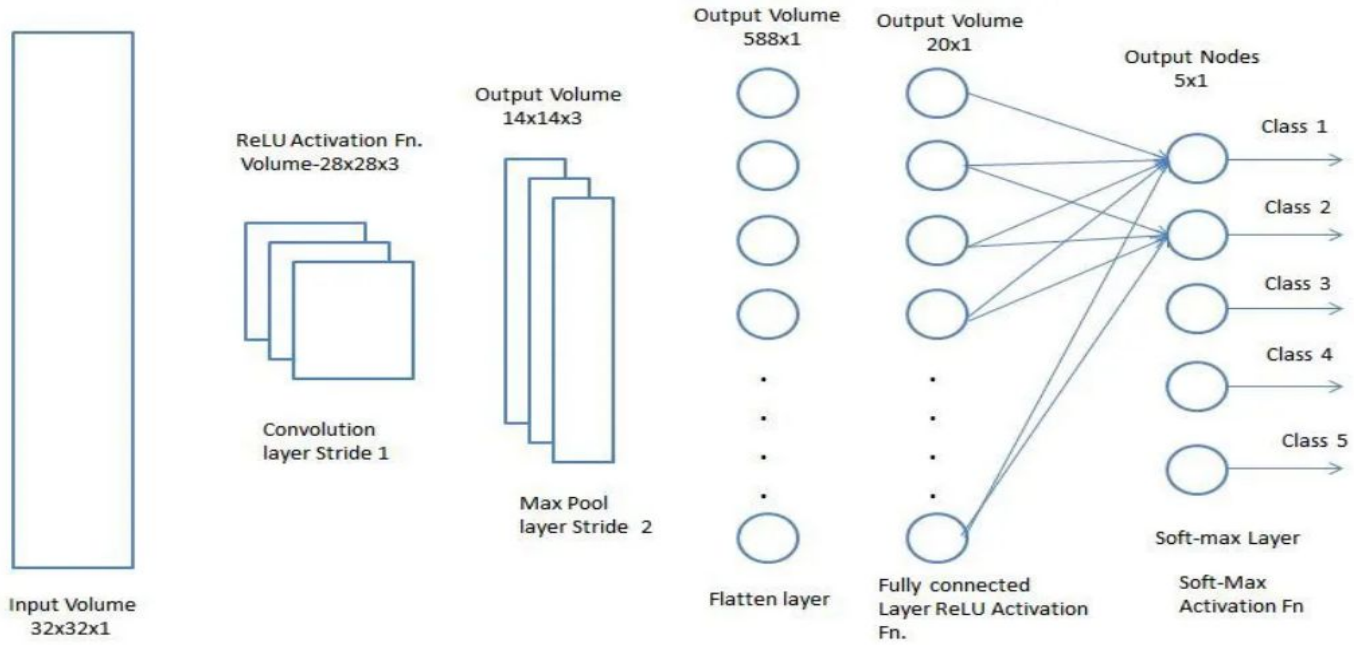


Flattening:

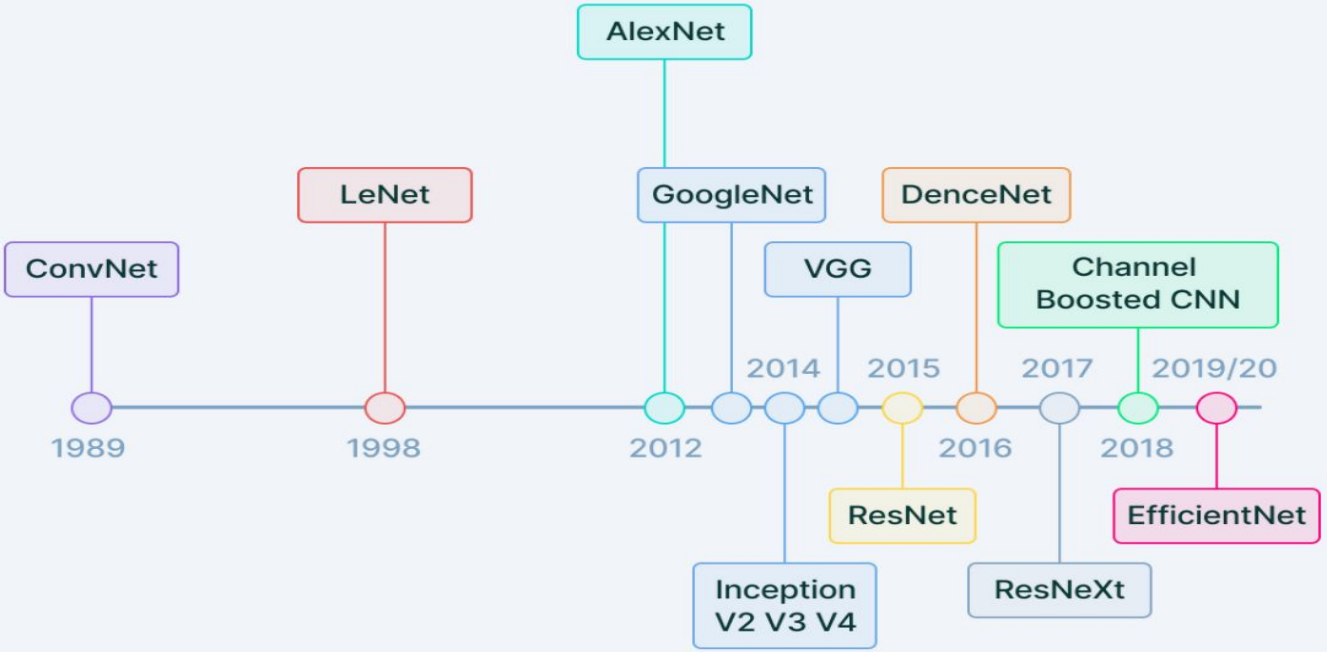
- It is the process of converting all the resultant 2 dimensional arrays from pooled feature map into a single long continuous linear vector



Fully Connected Layer

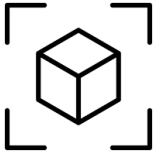


Types of CNN


AlexNet
LeNet
GoogleNet
DenceNet
ConvNet
VGG
Channel
Boosted CNN
2014
2015
2017
2019/20
1989
1998
2012
2016
2018
ResNet
EfficientNet
Inception
V2 V3 V4
ResNeXt

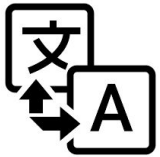
Applications of CNN

Face Detection



Object Detection

Self Driving or autonomous car



Auto Translation

X-Ray Image analysis

Cancer Detection

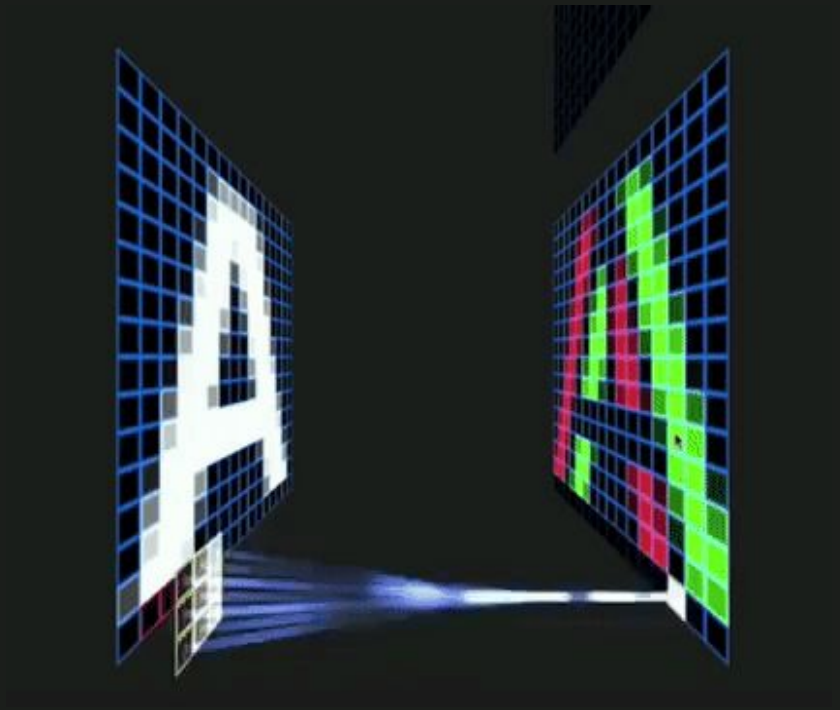
Advantages

1. High Accuracy at Image recognition
2. Good at detecting patterns and features in images, videos, and audio signals

Disadvantages

1. A lot of training data is needed
2. CNN's tend to be much slower
3. Training process takes a long time
4. Fail to encode the position and orientation of objects
5. Cannot do multi-object detection.

Region based Convolutional Neural Network (R-CNN)



Why R-CNN?

- CNN is a fully connected layer
- Unable to deal with frequency of occurrences
- Brute force (Sliding window) as an alternative
- Objects have different size and aspect ratio regions
- Computationally very expensive

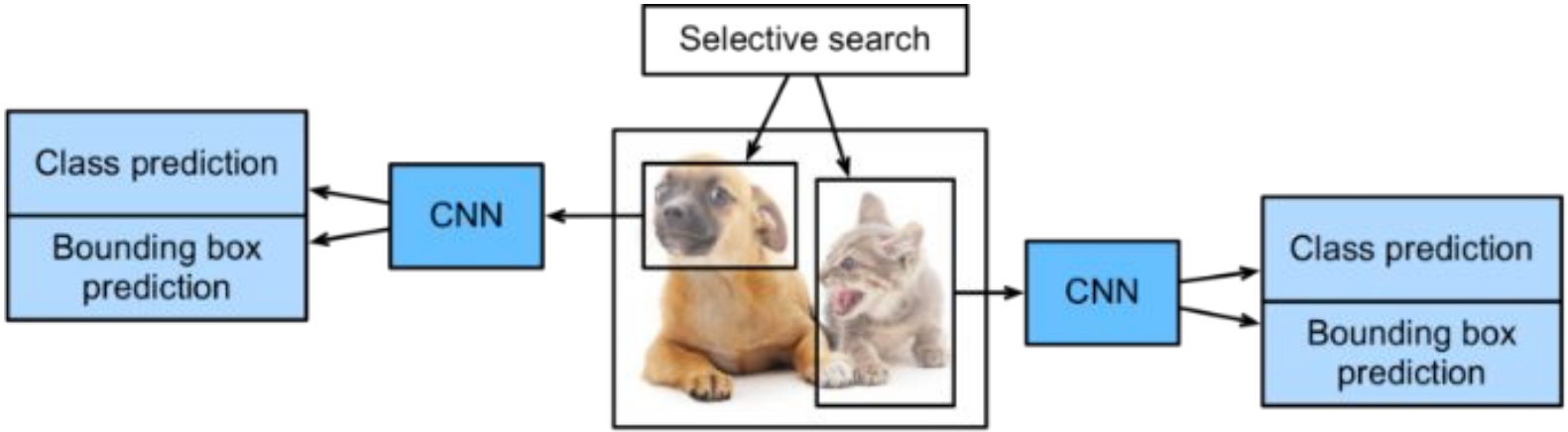
- Research Scientist

Meta / Facebook AI Research

- Well known for developing **R-CNN architecture** for object detection
- Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5) - 2014
- Learn more about Ross - visit www.rossgirshick.info

Ross Girshick

R-CNN



- Instead of working on massive regions on image, RCNN proposes bunch of boxes in the image called **bounding boxes**
- It checks for objects inside this bounding box
- RCNN uses **selective search** which generates ~**2000** region proposals
- Region proposals are provided to CNN architecture that computes CNN features
- It used **SVM model** to classify the object present in these region proposal

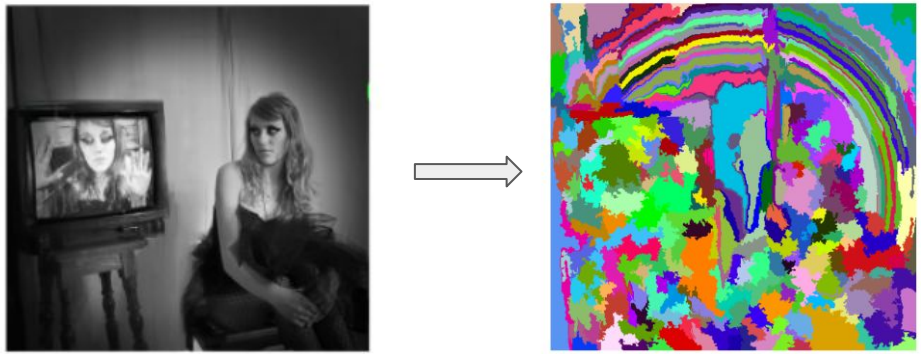
R-CNN

- Prior to R-CNN object detection with PASCAL VOC dataset was sluggish
- R-CNN made an impressive **30%** (relative) improvement
- CNN trained on images from ImageNet can work well
- Use of **Selective Search** for selecting regions of interest

Region Proposals: Selective search

Input image

Region Proposals: Selective search



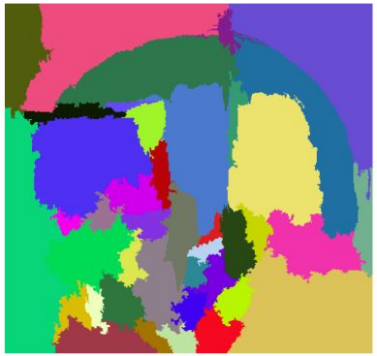
Input image

Initial Segmentation

Region Proposals: Selective search

Input image

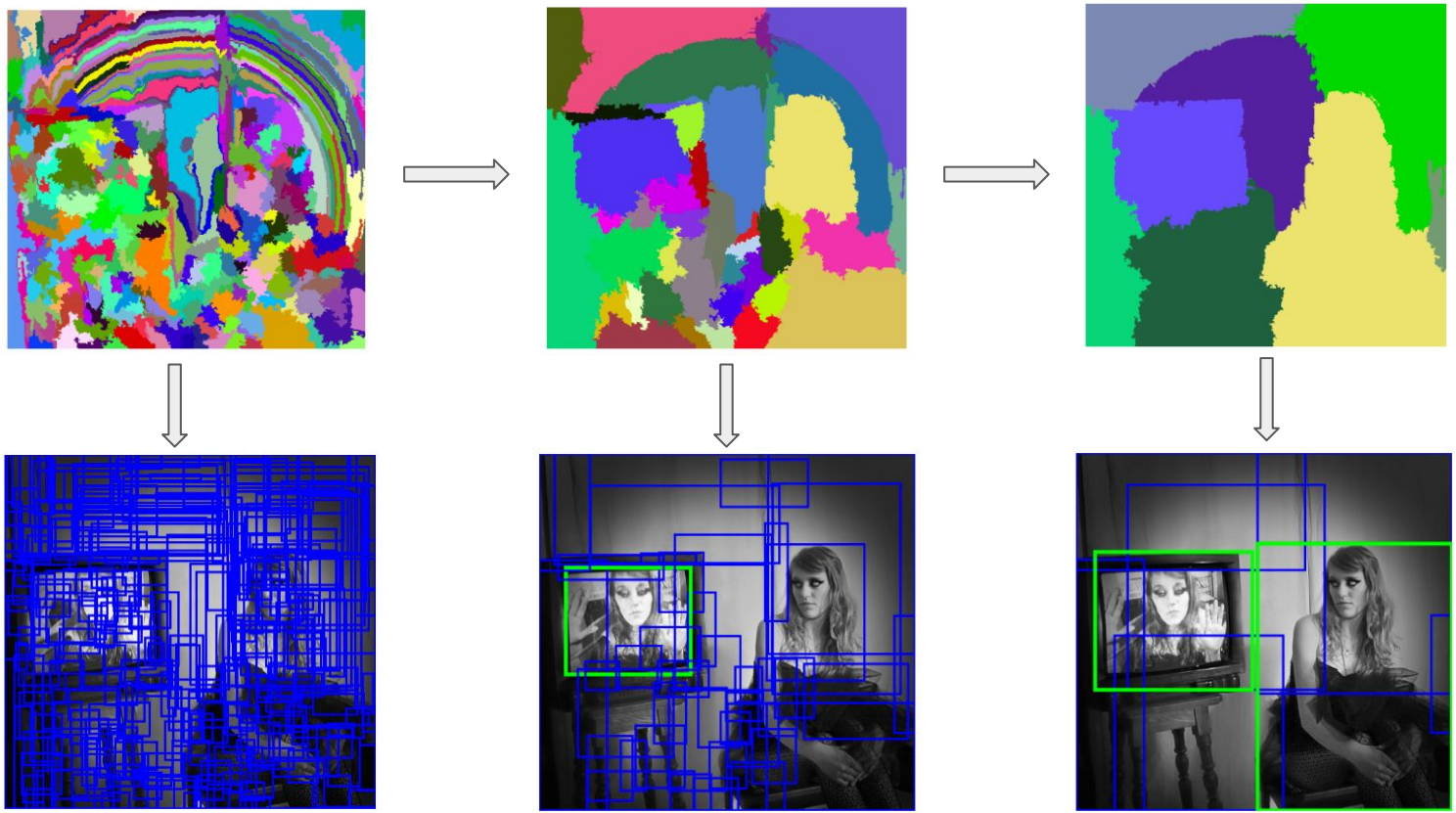
Initial Segmentation

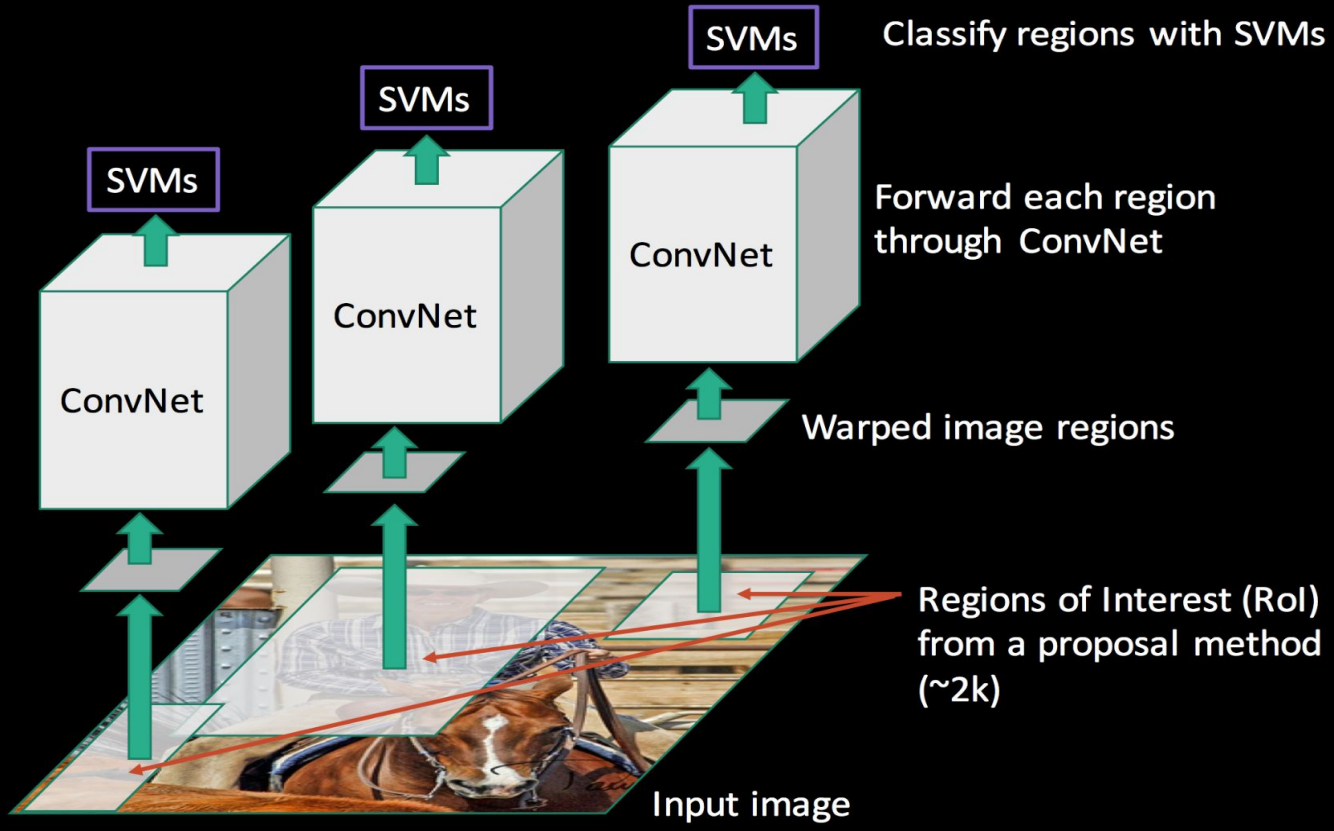


After many iterations

Region Proposals: Selective search

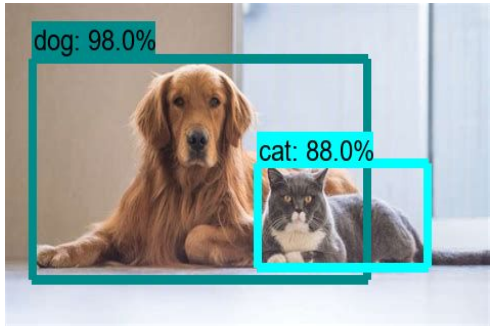
Converting regions to boxes





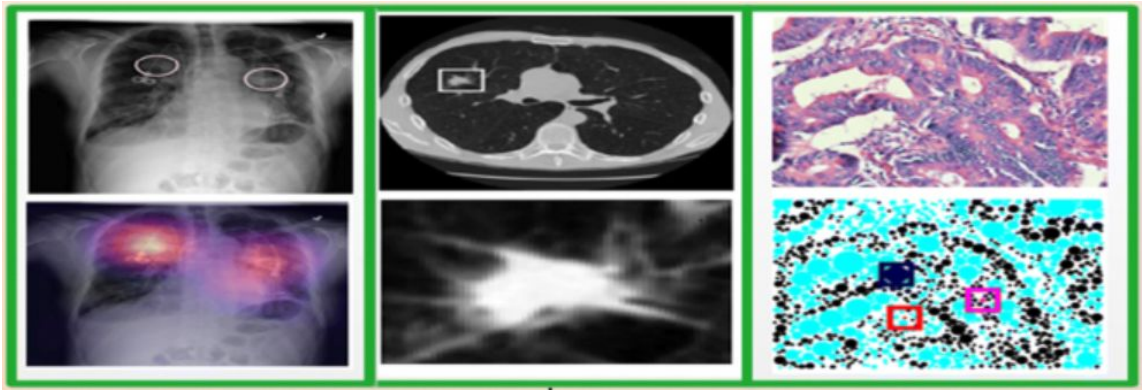
Girshick et al. CVPR14.

Applications



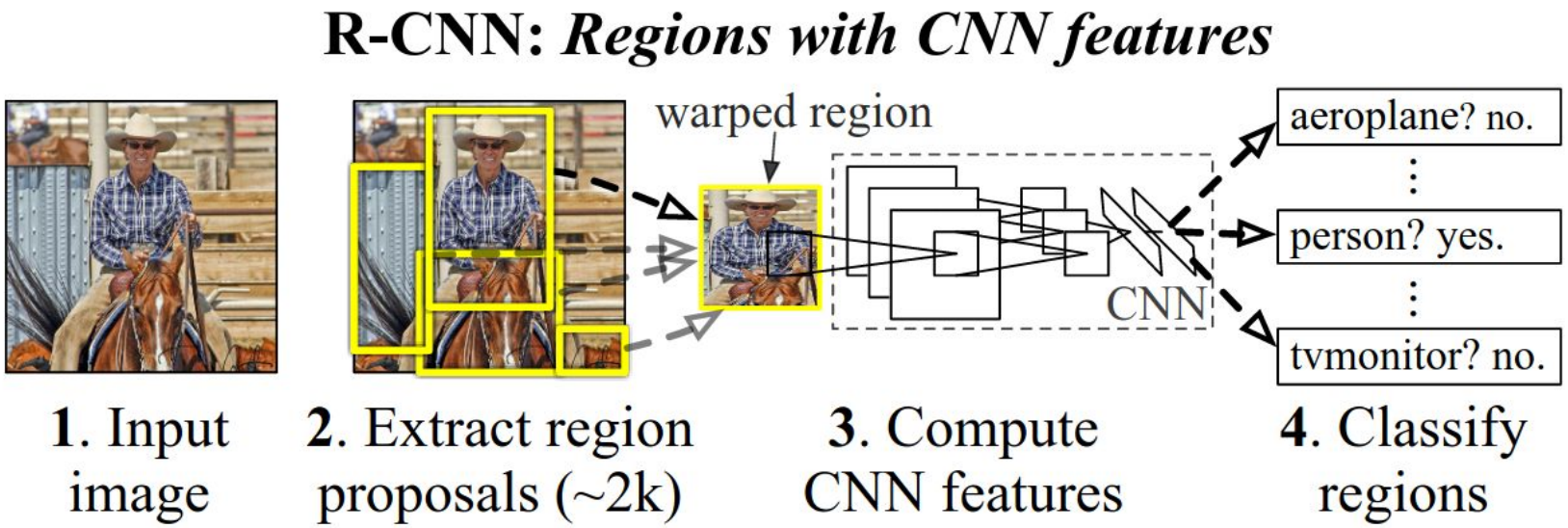
Object Detection in image

Face Detection



Medical image analysis

R-CNN


R-CNN: *Regions with CNN features*
warped region
CNN
aeroplane? no.
person? yes.
tvmonitor? no.
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

Selective Search

- Color Similarity

  $C_i^k, c_j^k = k^{th}\ value\ of\ histogram\ bin\ of\ region\ r_i\ and\ r_j\ respectively$

  $\mathbf{S_{color}(r_i, r_j) = \sum_{k=1}^{n} min(c_i^k, c_j^k)}$

- Texture Similarity

  $t_i^k, t_j^k = k^{th}\ value\ of\ texture\ histogram\ bin\ of\ region\ r_i\ and\ r_j\ respectively$

  $\mathbf{S_{texture}(r_i, r_j) = \sum_{k=1}^{n} min(t_i^k, t_j^k)}$

- Size Similarity

  $\mathbf{S_{size}(r_i, r_j) = 1 - (size\,(r_i) + size\,(r_j)) \div size\,(img)}$

  $where\ size\,(r_i)\ ,\ size\,(r_j)\ and\ size\,(img)\ are\ the\ sizes\ of\ regions\ r_i\ ,\ r_j\ and\ image\ respectively\ in\ pixels$

- Fill Similarity

  $\mathbf{S_{fill}(r_i, r_j) = 1 - (size\,(BB_{ij}) - size\,(r_i) - size\,(r_j)) \div size\,(img)}$

  $size\,(BB_{ij})\ is\ the\ size\ of\ bounding\ box\ around\ i\ and\ j$

- Combined Similarity

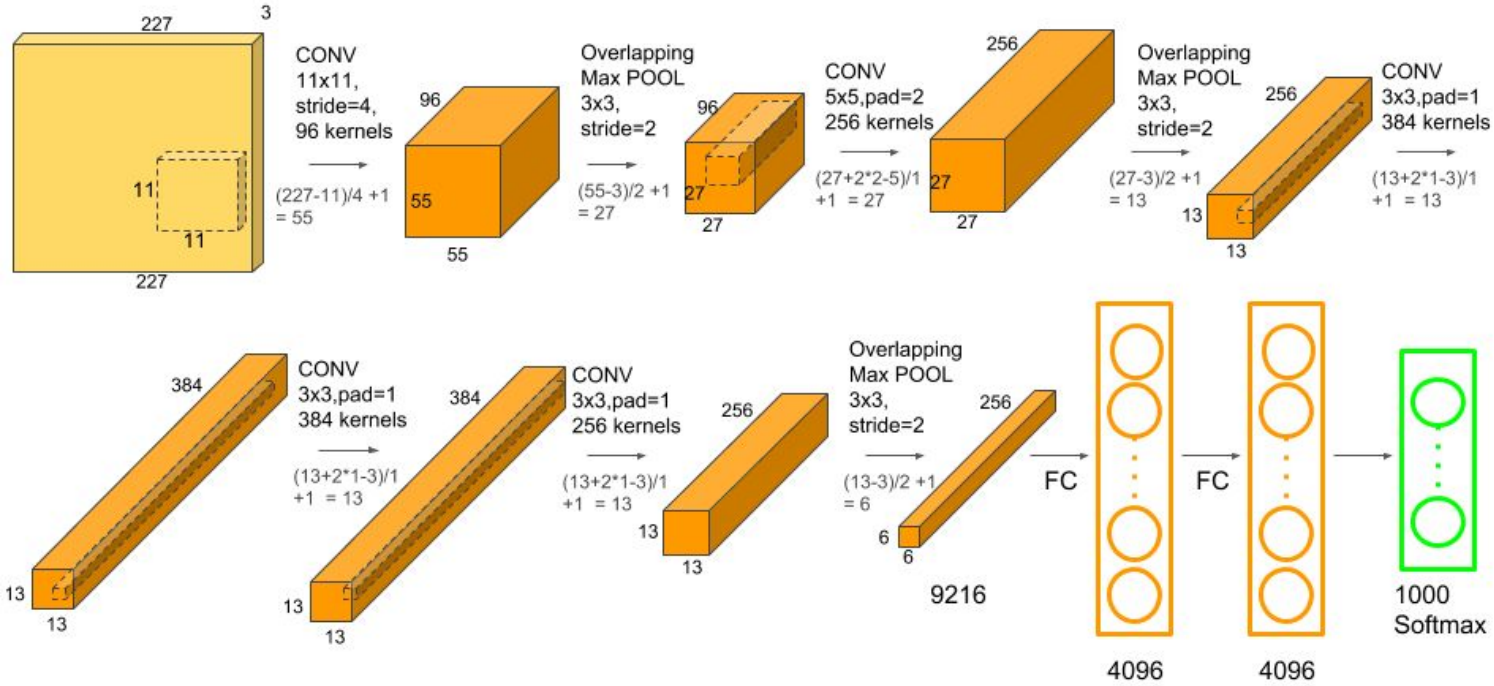
  $\mathbf{S_{(r_i, r_j)} = a_1 * s_{color}(r_i, r_j) + a_2 * s_{texture}(r_i, r_j) + a_3 * s_{size}(r_i, r_j) + a_4 * s_{fill}(r_i, r_j)}$

  $where\ a_i\ is\ either\ 0\ or\ 1\ depending\ upon\ we\ consider\ this\ similarity\ or\ not\ .$

Warped Region

- (B) Tightest square with context
- (C) Tightest square without Context

Computing CNN (Alexnet)

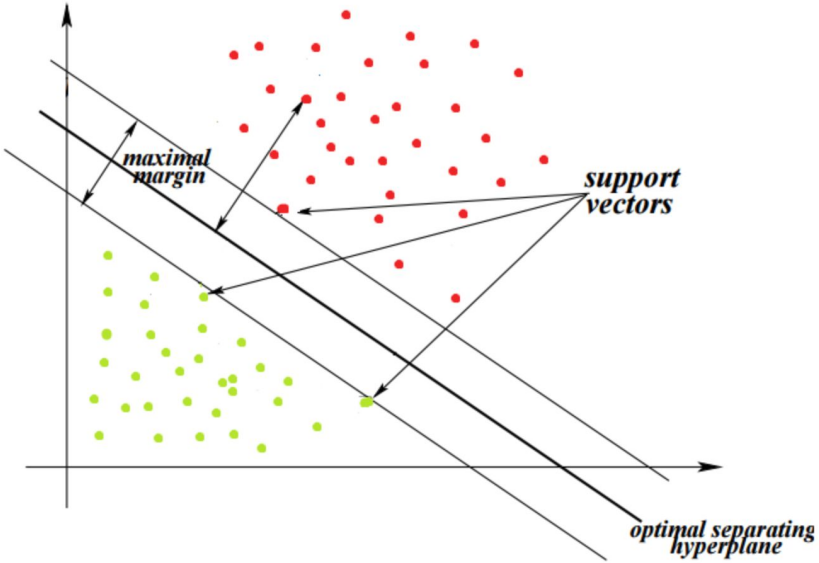


Classifying Regions

- The output from CNN produces a (1, 4096) feature vector.

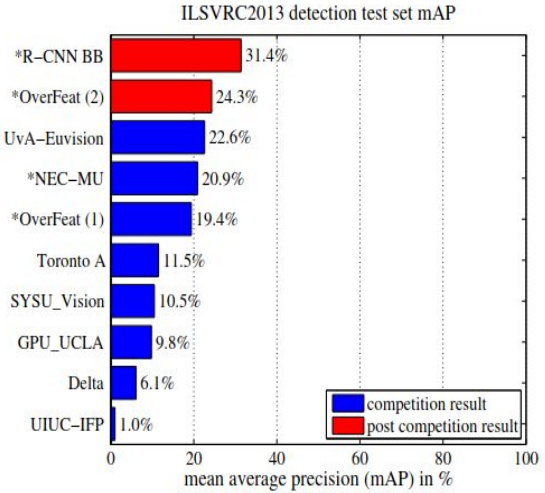
- Output is passed to a SVM model for classification and bounding-box regression for localization

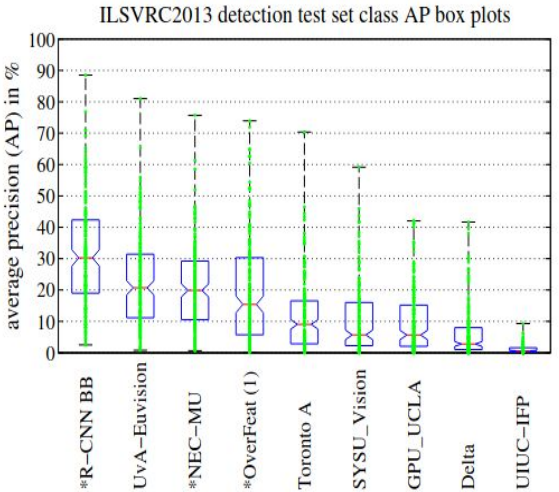
- One linear SVM per class



source:**stackexchange.com**

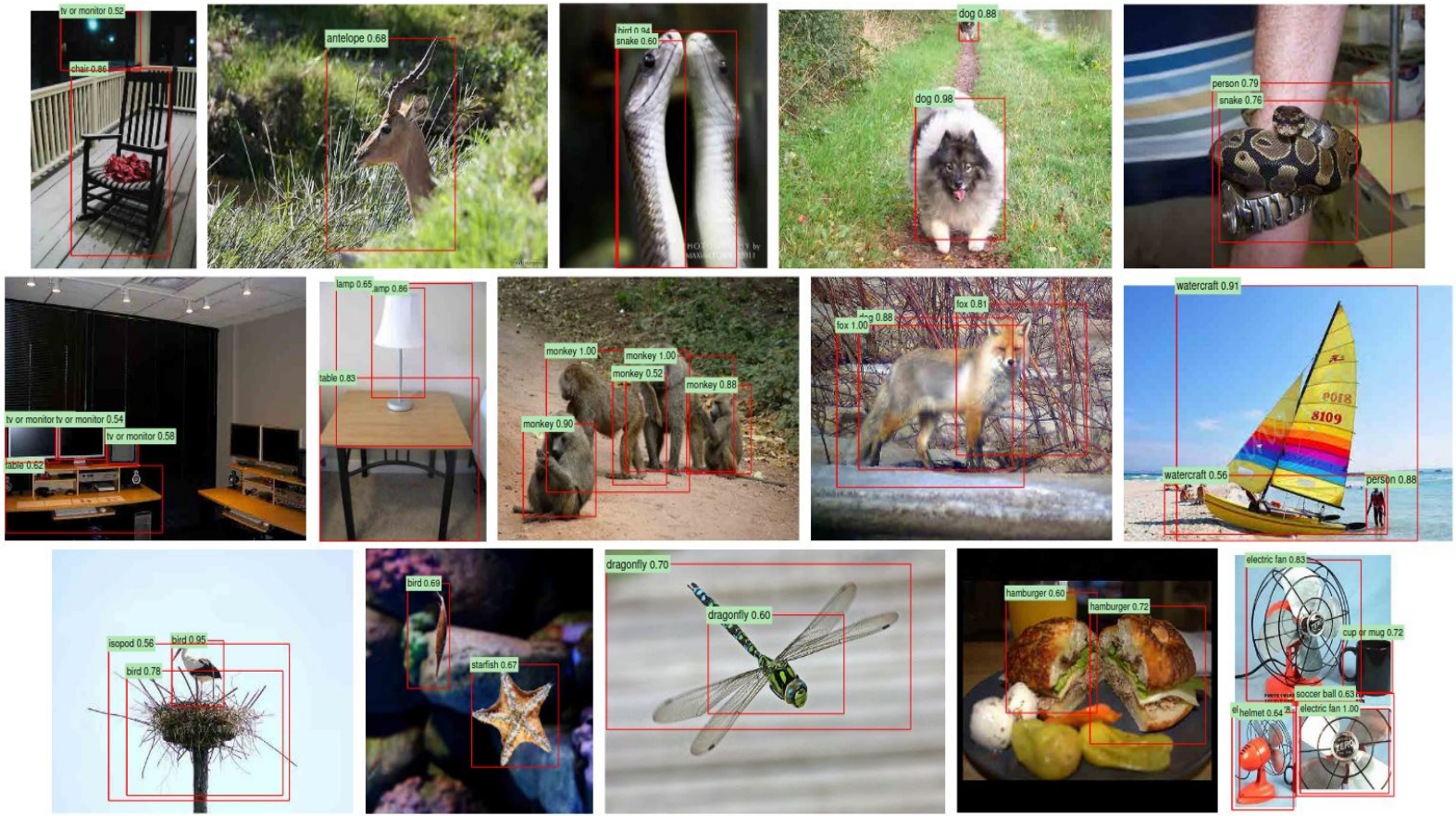
Comparing to Other Methods





VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [39]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [41]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [18]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

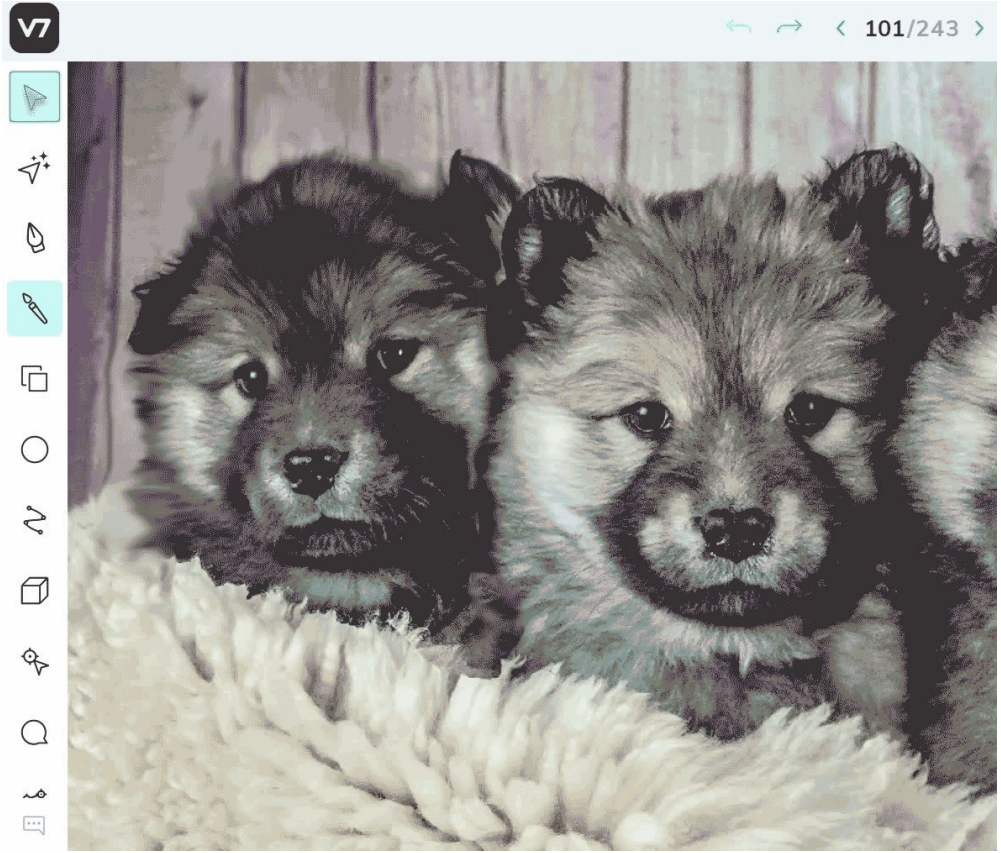
Conclusion


tv or monitor 0.52
chair 0.86
antelope 0.68
snake 0.60
dog 0.88
dog 0.98
person 0.79
snake 0.76
tv or monitor tv or monitor 0.54
tv or monitor 0.58
table 0.62
lamp 0.65
lamp 0.86
table 0.83
monkey 1.00
monkey 1.00
monkey 0.52
monkey 0.88
monkey 0.90
fox 1.00
fox 0.81
watercraft 0.91
watercraft 0.56
person 0.88
isopod 0.56
bird 0.95
bird 0.78
bird 0.69
starfish 0.67
dragonfly 0.70
dragonfly 0.60
hamburger 0.60
hamburger 0.72
electric fan 0.83
cup or mug 0.72
soccer ball 0.63
helmet 0.64
electric fan 1.00

Fast R-CNN

Fast R-CNN

- Fast RCNN is an object detection algorithm proposed by Ross Girshick in 2015.
- The approach is similar to the R-CNN algorithm. But, instead of feeding the regions proposals to the CNN, the input image is fed to CNN to generate a convolutional feature map.
- It is a single unified network.

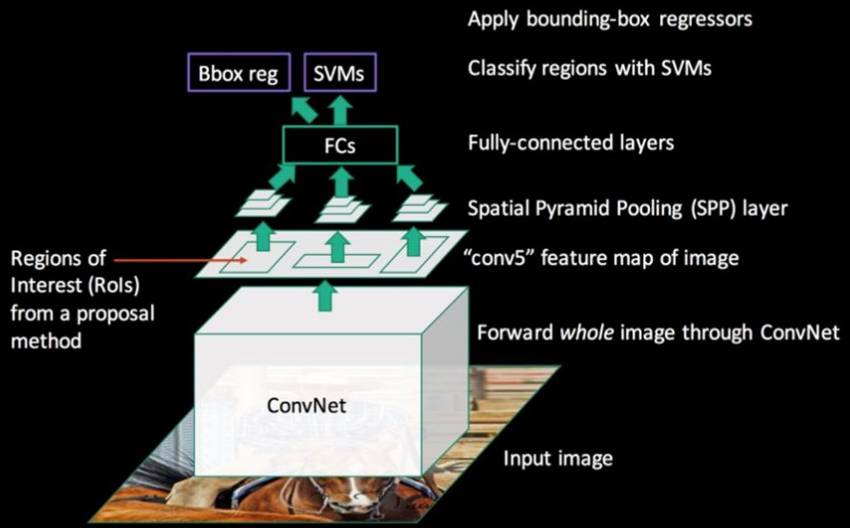


Problems with R-CNN

- It still takes a huge amount of time to train the network.
- Consider training the network with dataset of 1000 images, that would be a total pass of 2M(2000 *1000). Which is huge.
- It cannot be implemented in real time as it takes around 47 seconds for each image.
- If we have numerous images, it is hard to compute and requires huge volume of memory.Therefore we require another algorithm like Fast RCNN

Fast R-CNN

- Fast R-CNN is a powerful object detection model that utilizes a deep convolutional neural network to predict the locations of objects accurately and efficiently in an image.
- Spatial pyramid pooling networks(SPP-Nets) were used to speed up the algorithm.
- It accelerates by **10 to 100x** at test time and training time is reduced by **3x times**.



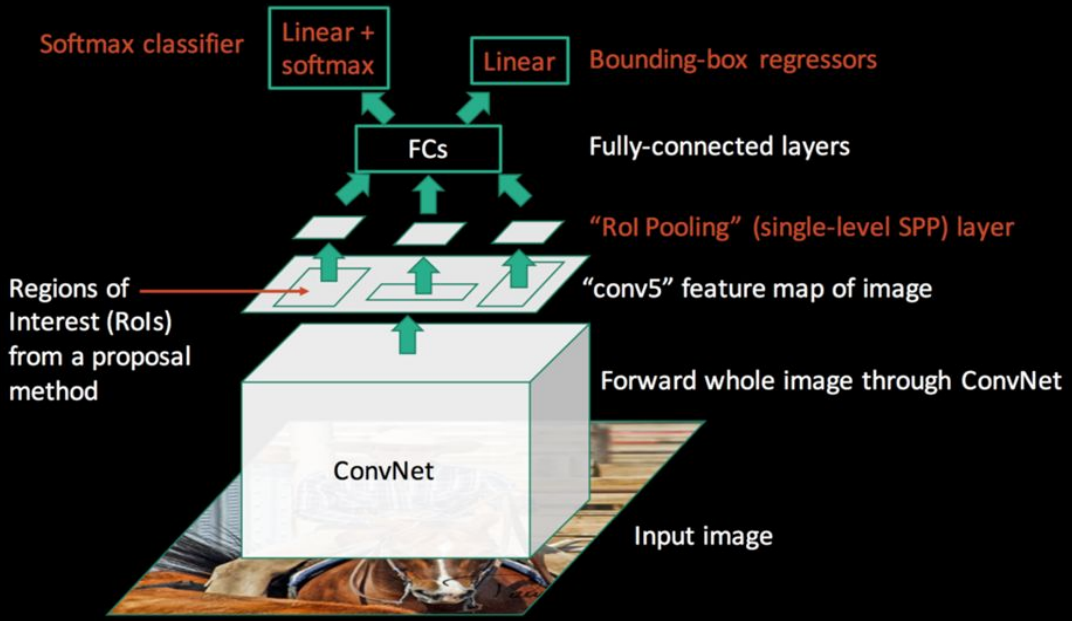
Spatial pyramid pooling network

- Convolutional neural architecture.
- Add on the last convolutional layer.
- Fixed length outputs.

Disadvantages of SPP-Net

- Training is a multi-stage pipeline.
- Pipeline involves extracting features.
- Pipeline involves fine tuning a network with log loss.
- Fixed Convolutional layers limits the accuracy.

Working of Fast R-CNN



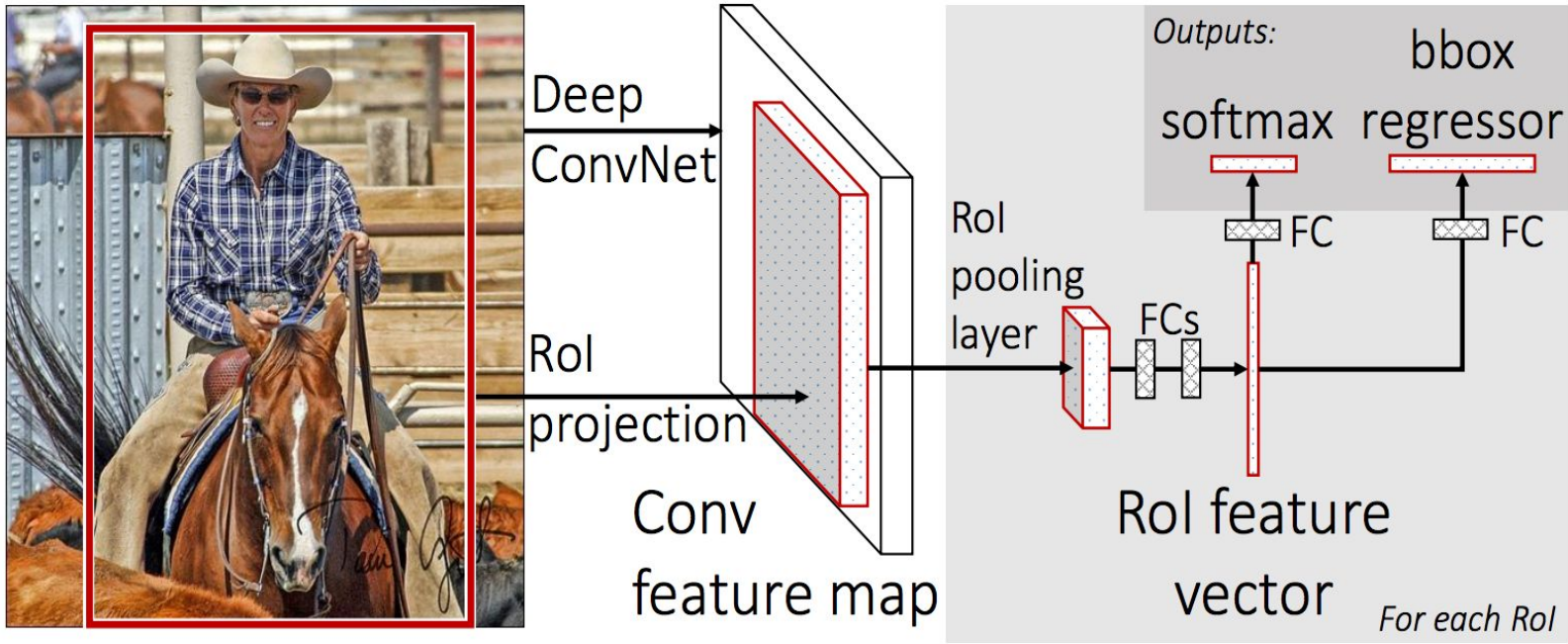
Applications with Fast R-CNN

Auto Pilot Cars

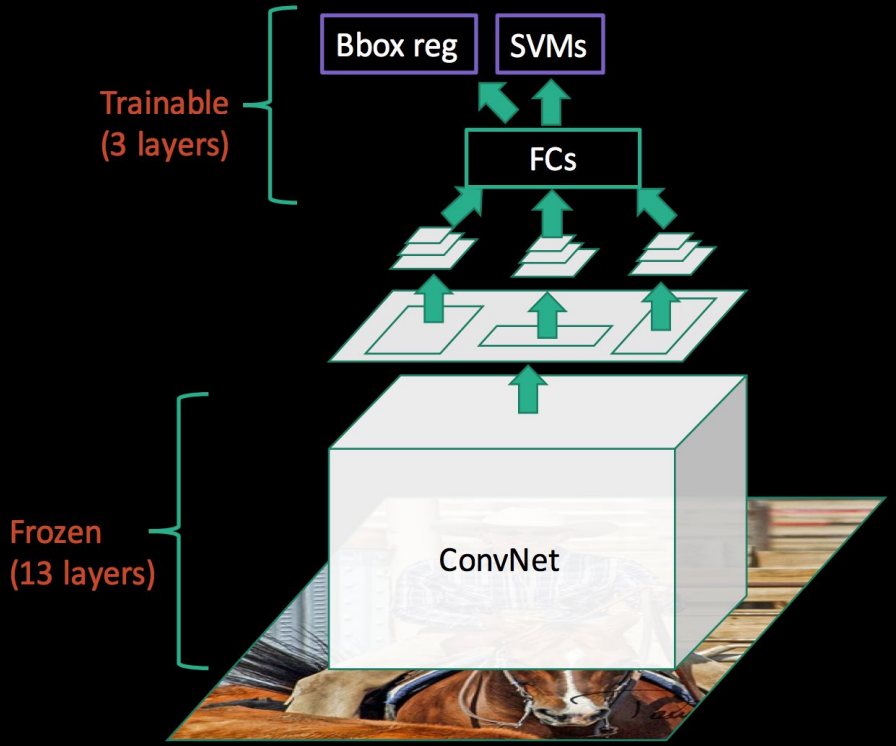
Human Face Detection

Detecting the presence of baby and other objects in a sequence of 2D images.

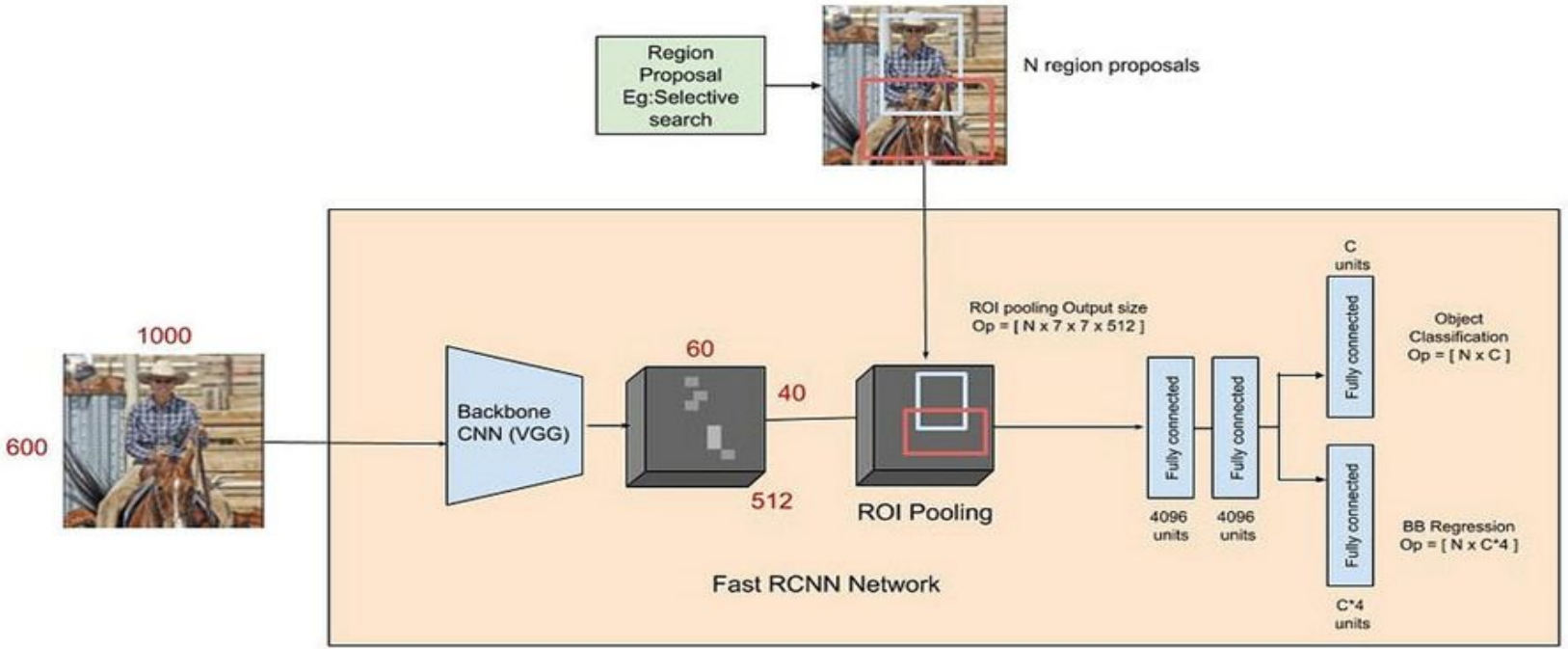
One pyramid SPP Net of Fast R-CNN



Difference Between Fast-RCNN and SPPnet



Fast R-CNN Architecture



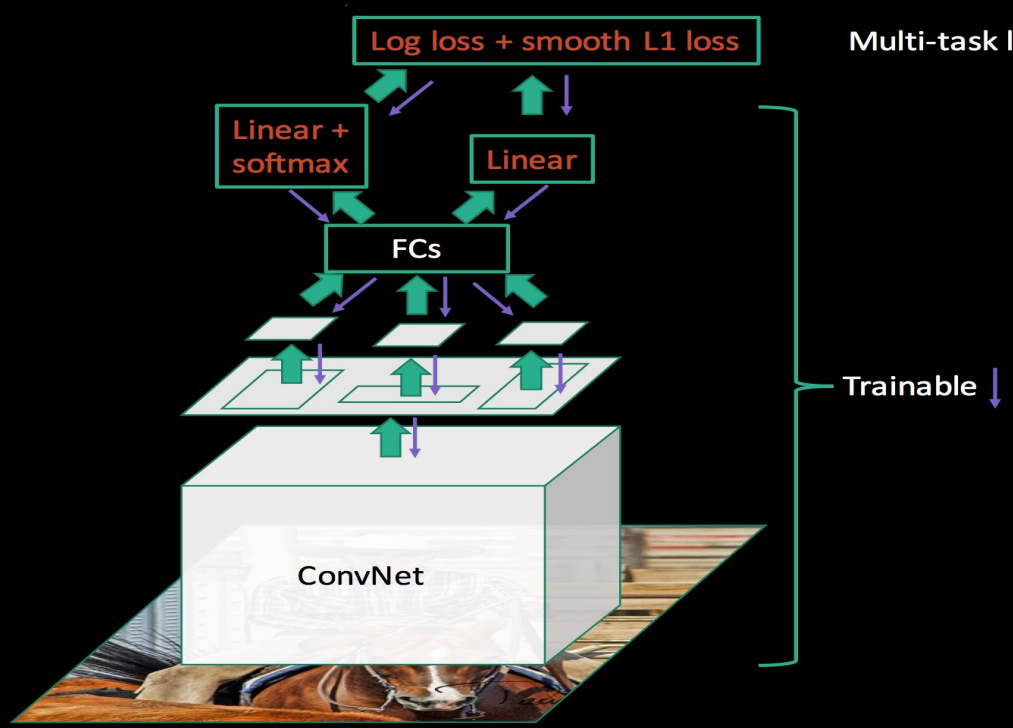
Fast R-CNN Architecture

- ROI Pooling layer
- Initializing from pre-trained networks
- Fine tuning for detection
  - **Multi - task loss**
  - **Mini batch sampling**
  - **Back propagation through ROI pooling layer**
  - **SGD hyper-parameters**
- Scale invariance

Fast R-CNN detection

- Truncated SVD for faster detection.
- Computational time is less.
- Detection of no. of ROI’s is large.
- Weight matrix is used.
- Reduces parameter count.
- Compression method gives good speedups.

Class Classification and L1 loss function



Equations

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

Equation 1: smooth L1 loss for BB regression

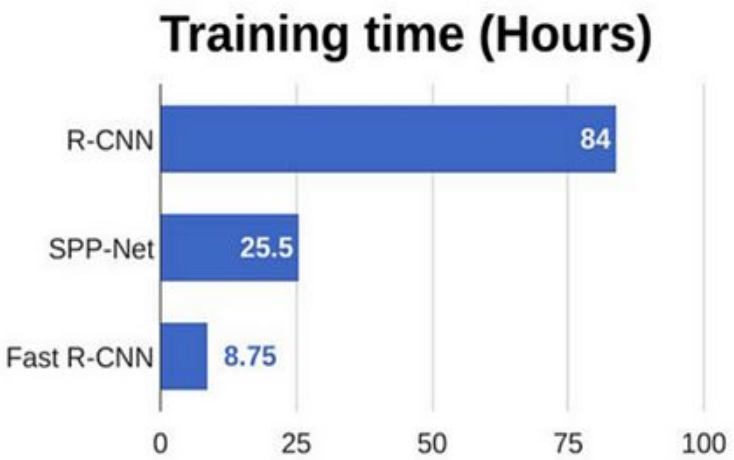
$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v)$$

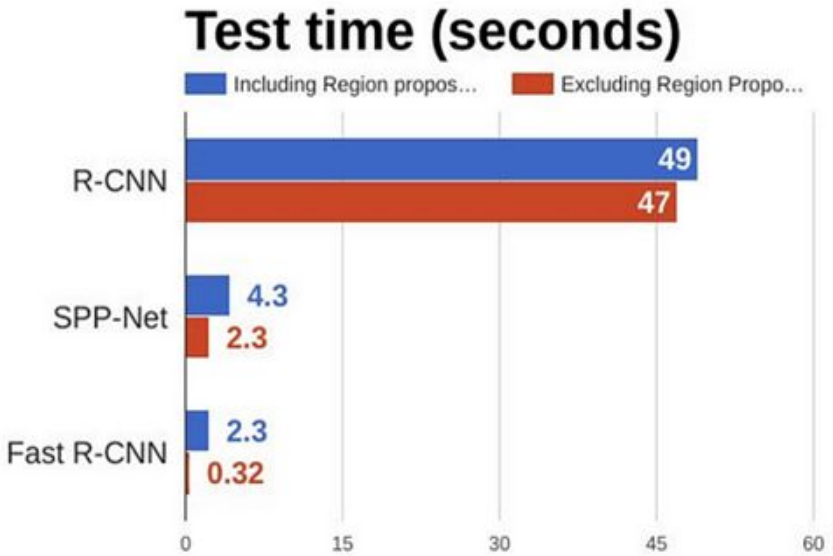
Equation 2: Joint loss for multi-task training

Results

- Fast R-CNN achieved state of the art performance at the time of its publication on VOC12, VOC10 and VOC07.
- Fast R-CNN processes images 45x faster than R-CNN at test time and 9x faster at train time
- It also trains 2.7x faster and runs test images 7x faster than SPP-Net.
- The detection time of the network is reduced by more than 30% with just a 0.3 drop in mAP.
- Multi-task training is not only easier but also improves performance because the tasks influence each other while training.

Performance of Fast-RCNN



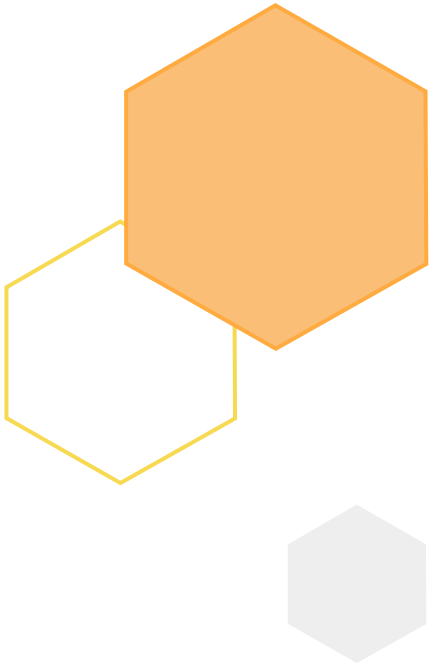


Pros of Fast R-CNN

- It achieves higher detection quality (measured in mean average precision, or mAP) than R-CNN and SPPnet.
- Training is single-stage, using a multi-task loss.
- Training can update all network layers.
- No disk storage is required for feature caching.

Cons of Fast R-CNN

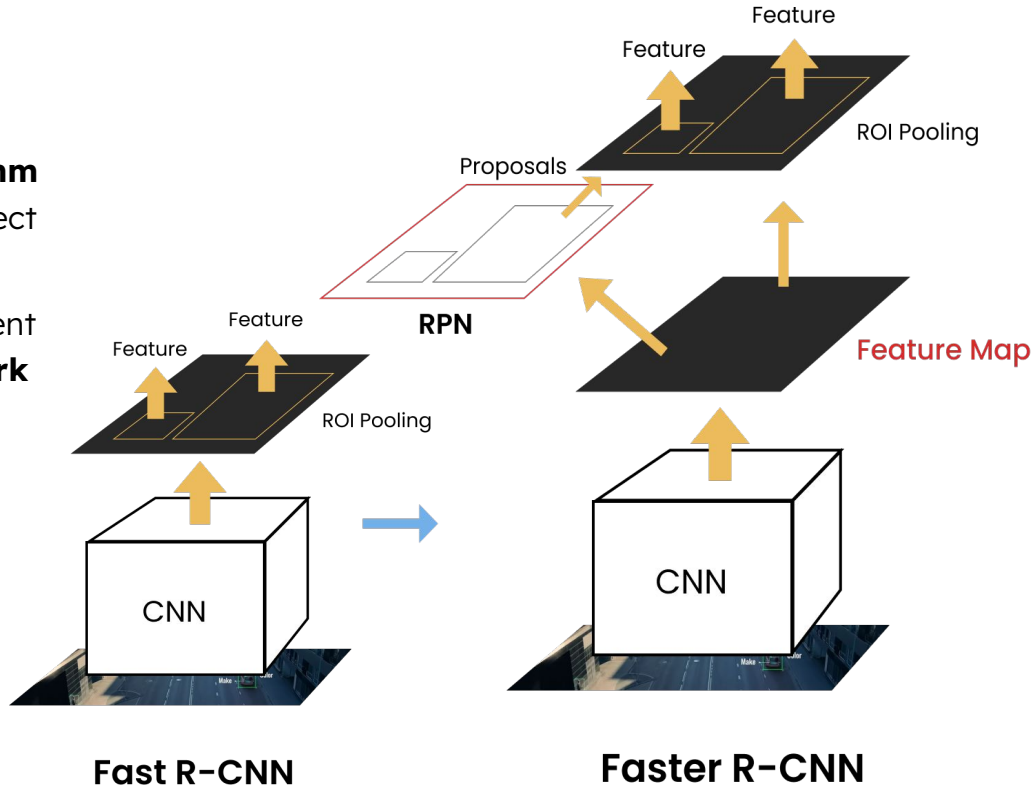
- Fast R-CNN uses selective search as a method for finding regions of interest.
- During testing, using region proposals significantly slows down the algorithm compared to not using region proposals, making them a bottleneck that affects performance.



Faster-RCNN

Faster R-CNN

- **state-of-art object detection algorithm** that can accurately and efficiently detect objects within an image
- Faster R-CNN introduced a new component called **the Regional Proposal Network (RPN).**



Why Faster RCNN?

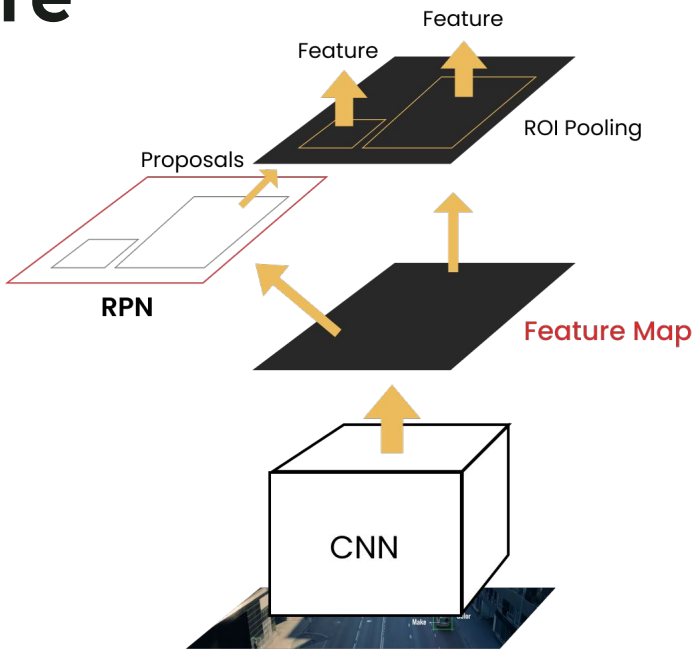
- For improvements in object detection **speed** and **accuracy**
- Fast RCNN suffered from a bottleneck caused by the region proposal step, which was slow and computationally expensive.

Limitations Faster RCNN addresses

Faster R-CNN Architecture

- It consists of two modules:
  - **a. RPN**: For generating region proposals
  - **b. Fast R-CNN:** For detecting objects

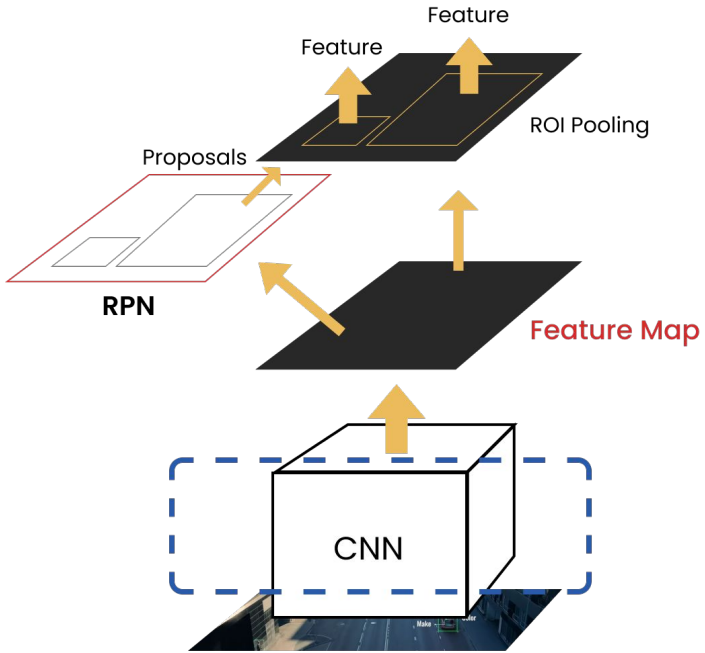
- How Faster R-CNN Works:
  - a. Image is provided as input to a convolutional network.
  - b. RPN is used to predict the region proposals.
  - c. The predicted region proposals are then reshaped using ROI pooling layer which is used to classify image within the proposed region and predict the offset values of the bounding boxes



Faster R-CNN

Backbone Layer

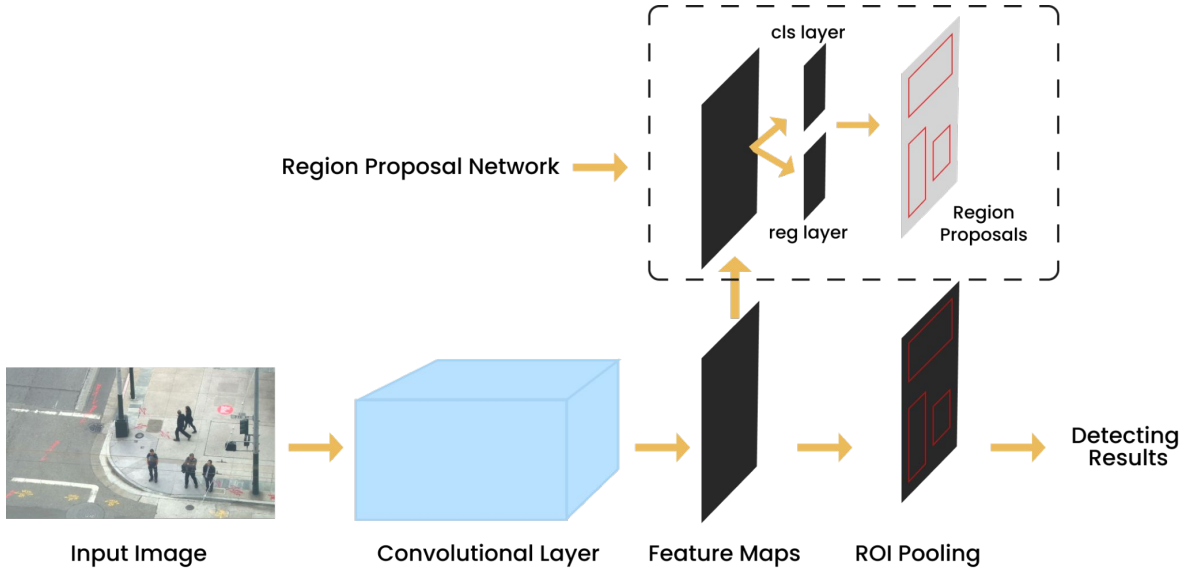
- The first step in the Faster R-CNN is to input an image that needs to be analyzed for objects.
- The input image is fed into a pre-trained convolutional neural network (CNN) such as VGG, ResNet, or Inception, which serves as the backbone layer of the model.
- The backbone layer extracts high-level features from the image.



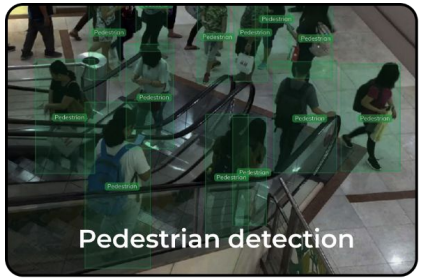
Faster R-CNN

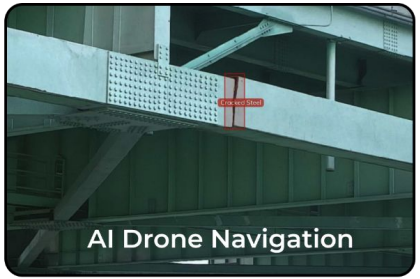
Regional Proposal Network (RPN)

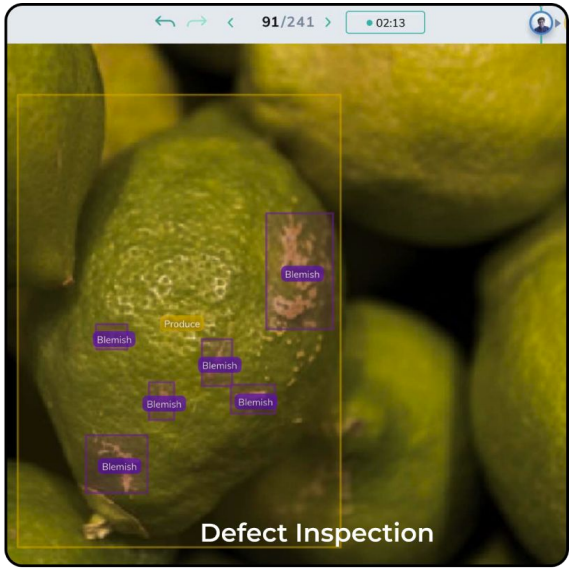
- The feature maps generated by the backbone layer are then fed into the Region Proposal Network (RPN), which is a small neural network that generates object proposals or regions of interest (RoIs) in the image.
- The RPN scans the feature maps at different scales and locations and predicts a set of rectangular boxes, each with an objectness score, which represent potential objects in the image.



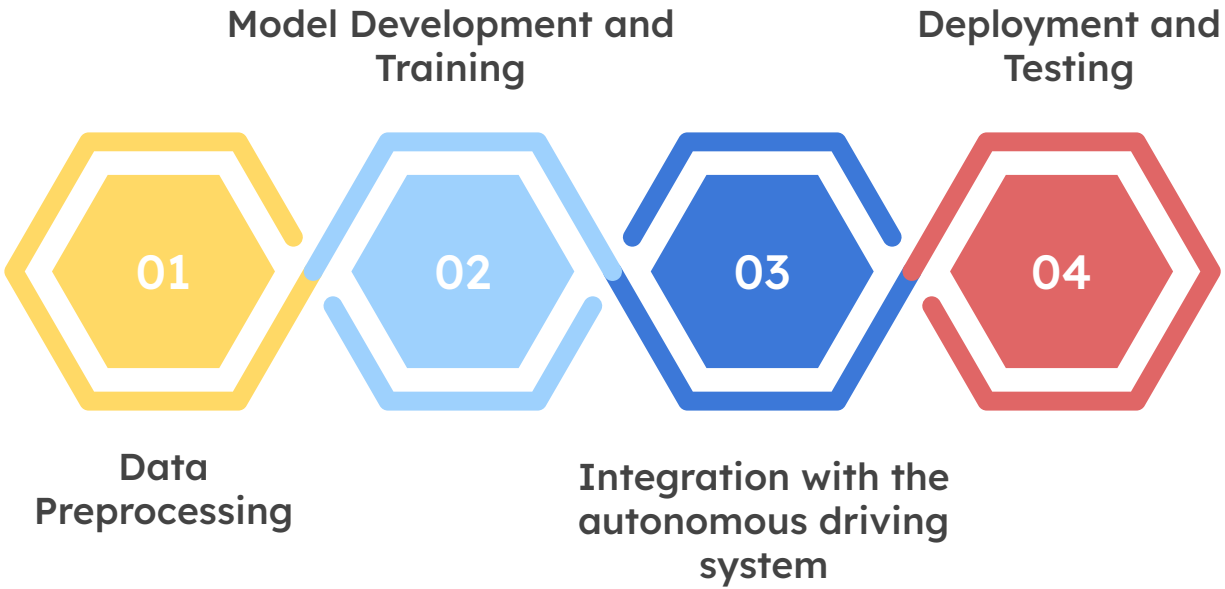
Applications



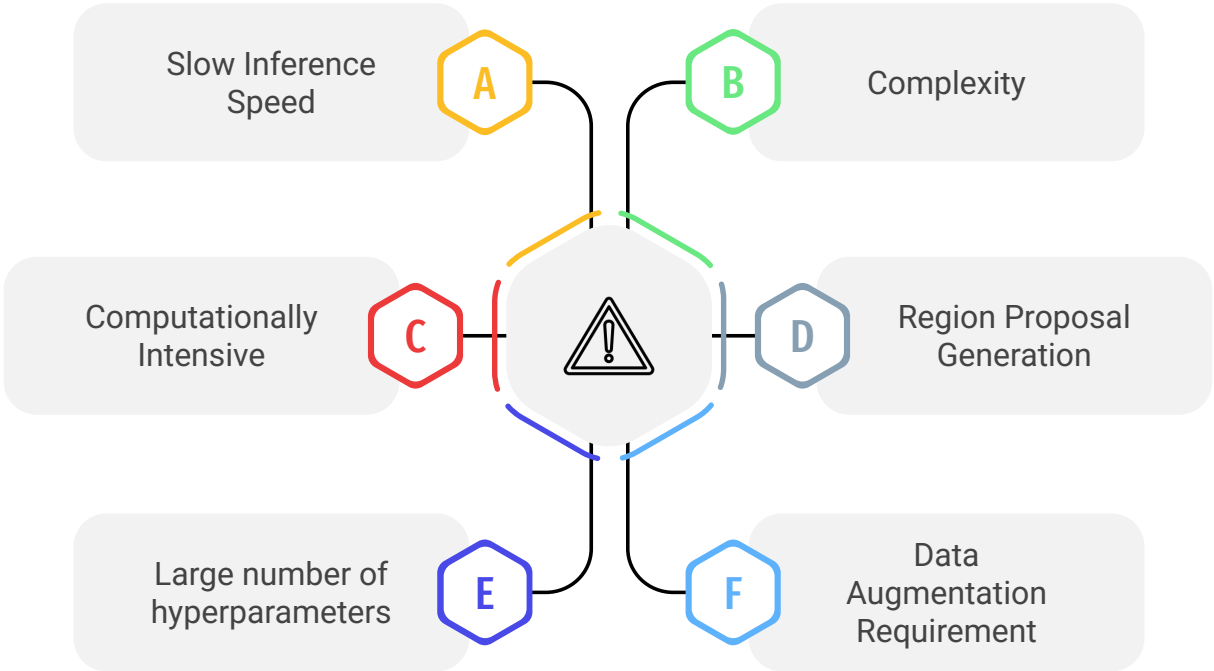




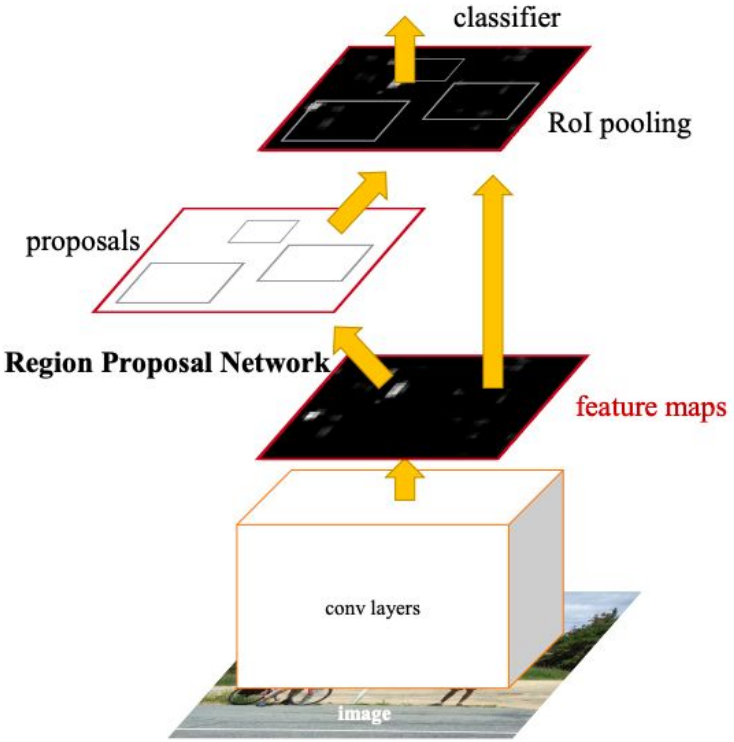
Implementation in Autonomous Driving



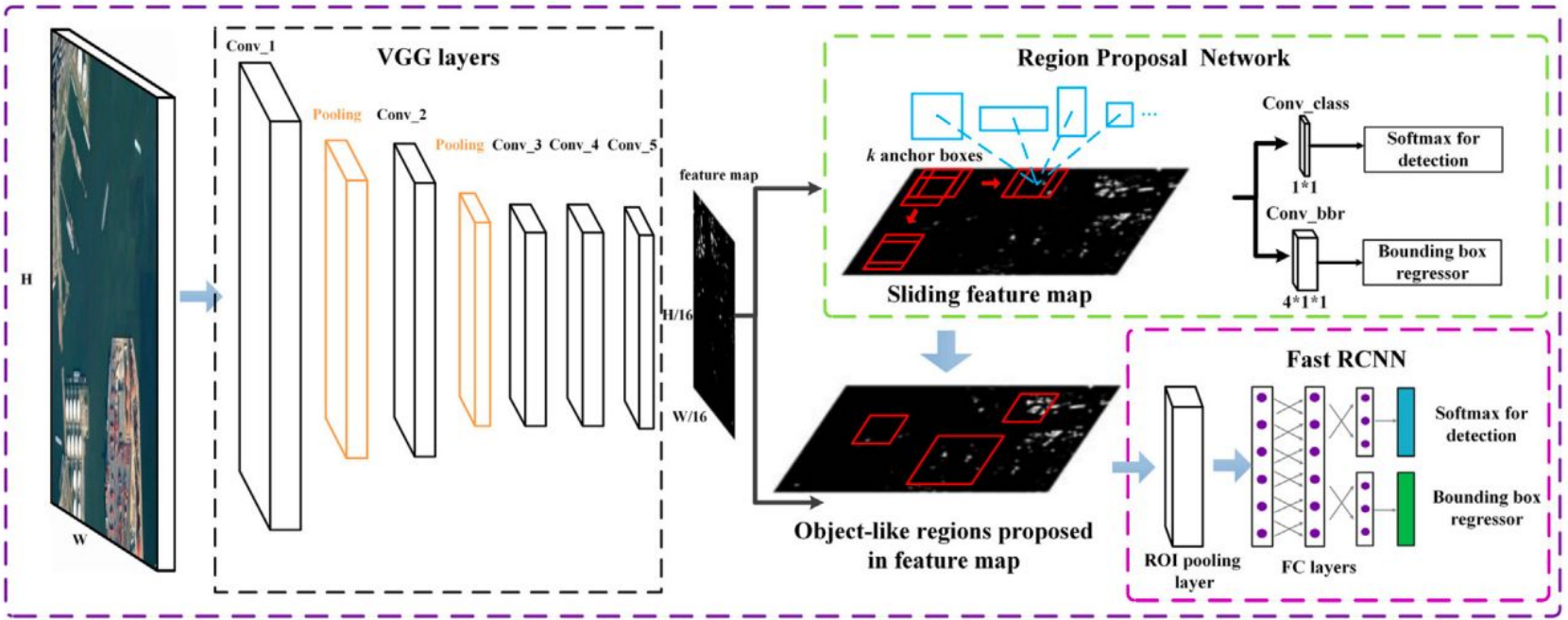
Limitation of Faster R-CNN



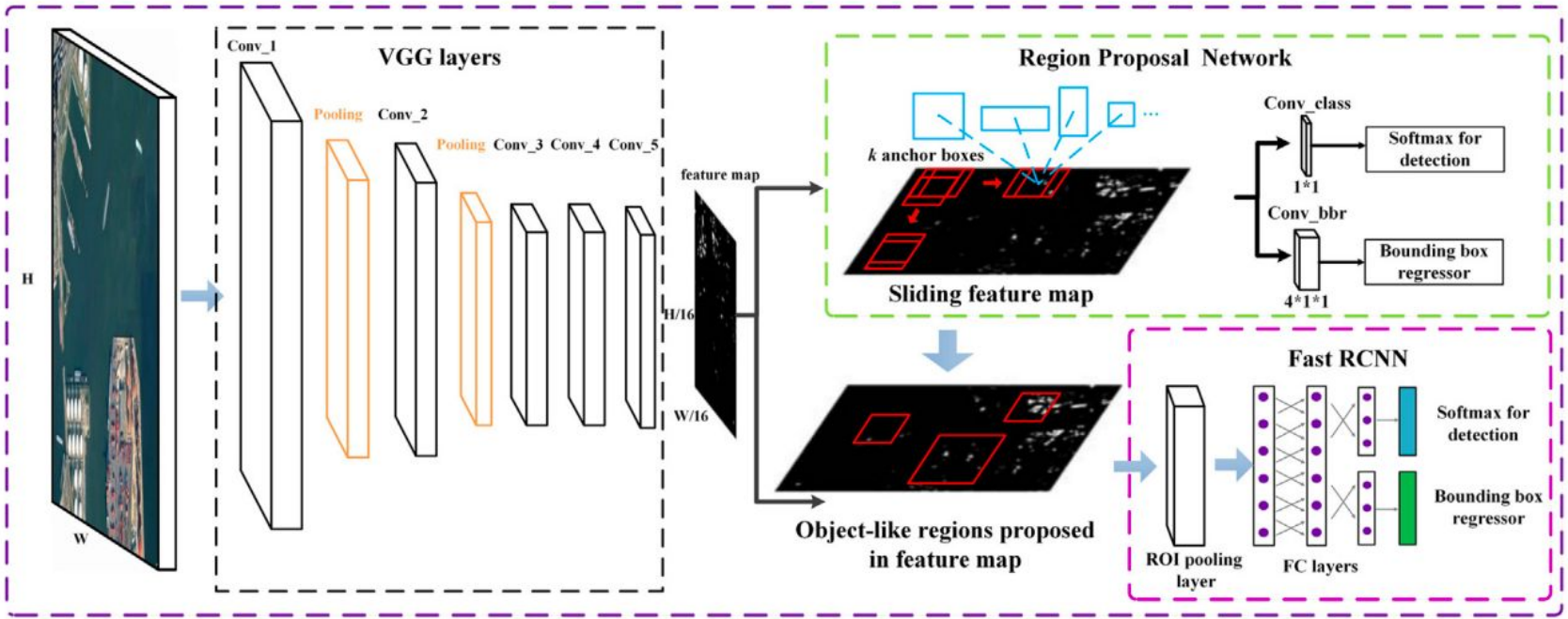
Faster-RCNN


classifier
RoI pooling
proposals
Region Proposal Network
feature maps
conv layers
Image

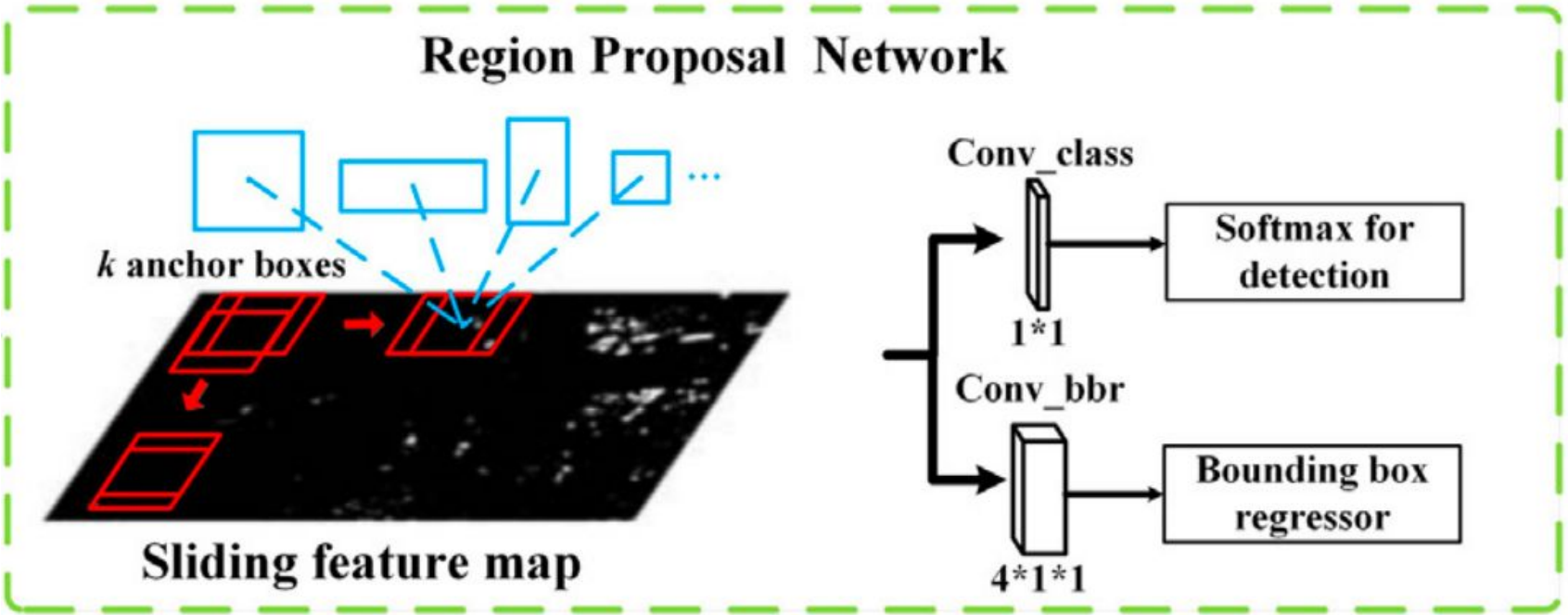
Faster-RCNN


H
W
Conv_1
VGG layers
Pooling
Conv_2
Pooling
Conv_3
Conv_4
Conv_5
feature map
H/16
W/16
Region Proposal Network
k anchor boxes
Sliding feature map
Conv_class
1*1
Softmax for detection
Conv_bbr
4*1*1
Bounding box regressor
Object-like regions proposed in feature map
Fast RCNN
ROI pooling layer
FC layers
Softmax for detection
Bounding box regressor

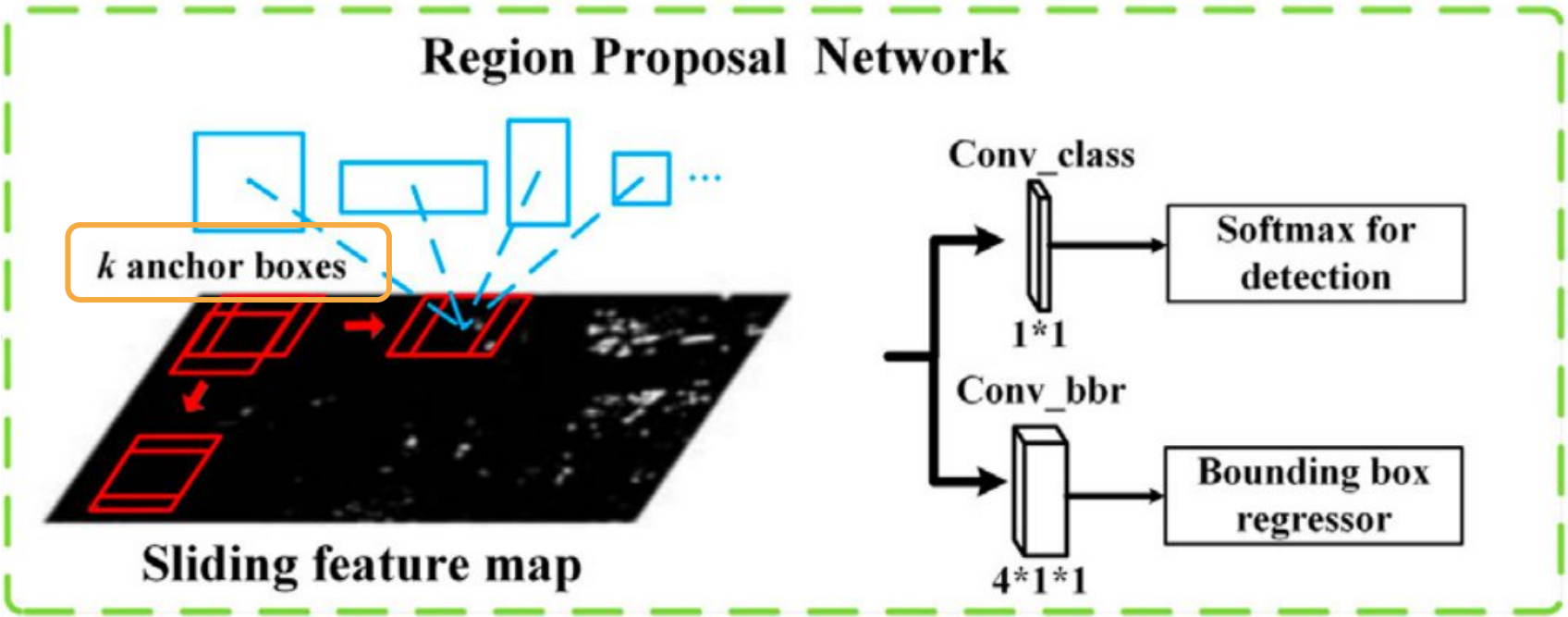
What's New ?


H
W
Conv_1
VGG layers
Pooling
Conv_2
Pooling
Conv_3
Conv_4
Conv_5
feature map
H/16
W/16
Region Proposal Network
k anchor boxes
Sliding feature map
Conv_class
1*1
Softmax for detection
Conv_bbr
4*1*1
Bounding box regressor
Object-like regions proposed in feature map
Fast RCNN
ROI pooling layer
FC layers
Softmax for detection
Bounding box regressor

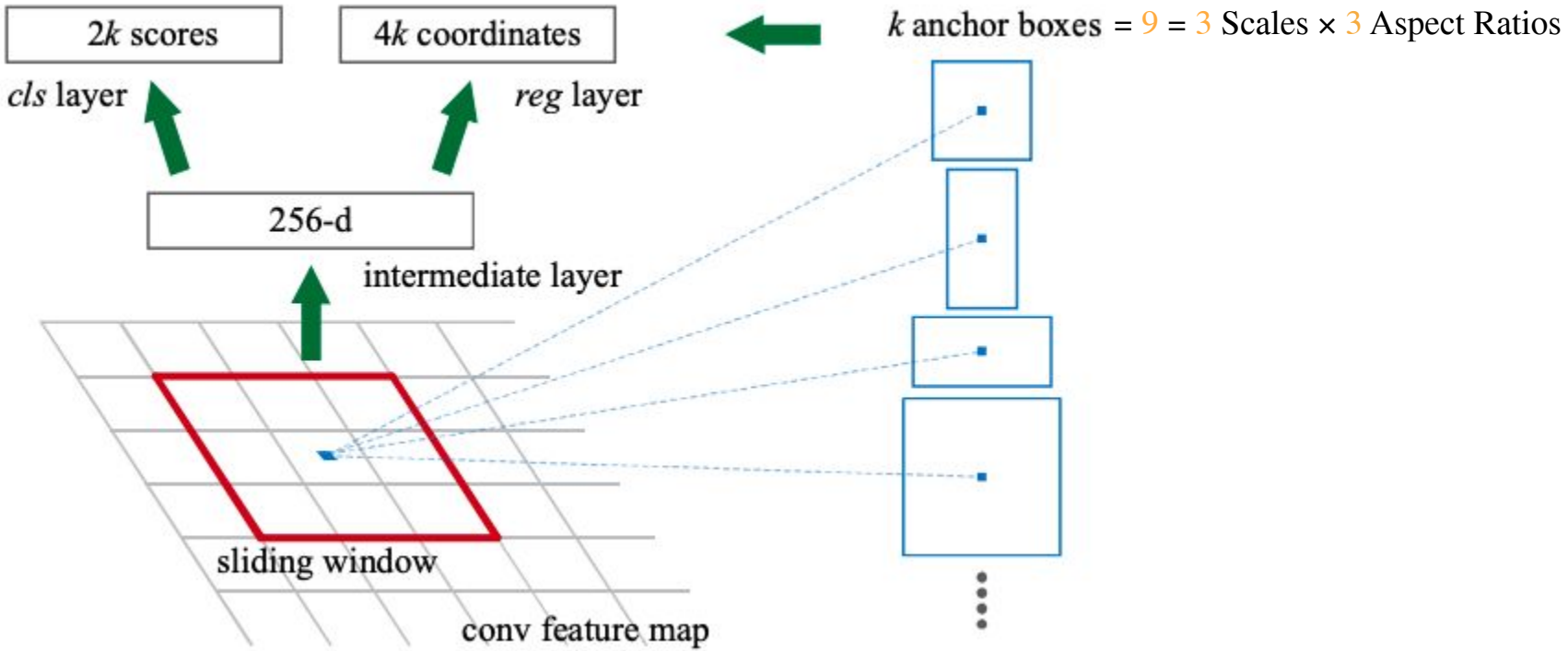
Region Proposal Network (RPN)



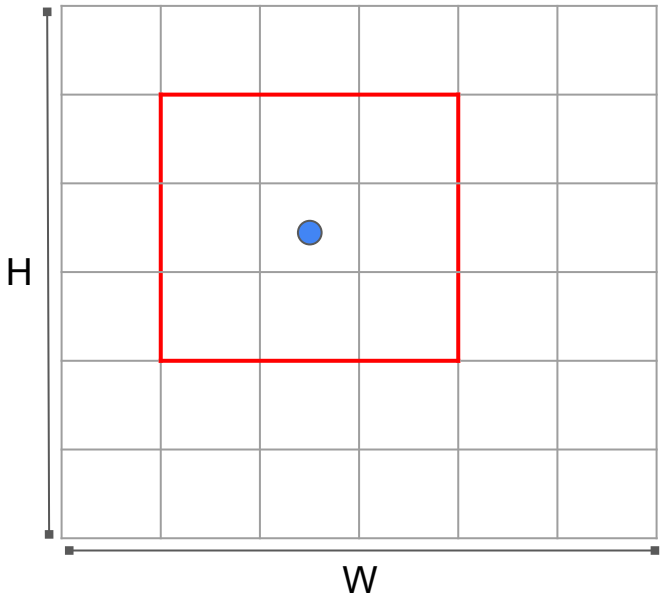
Region Proposal Network (RPN)



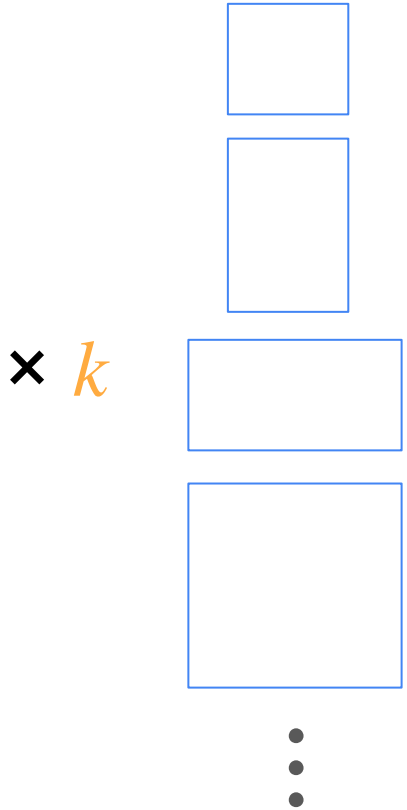
Anchor Boxes



Anchor Boxes



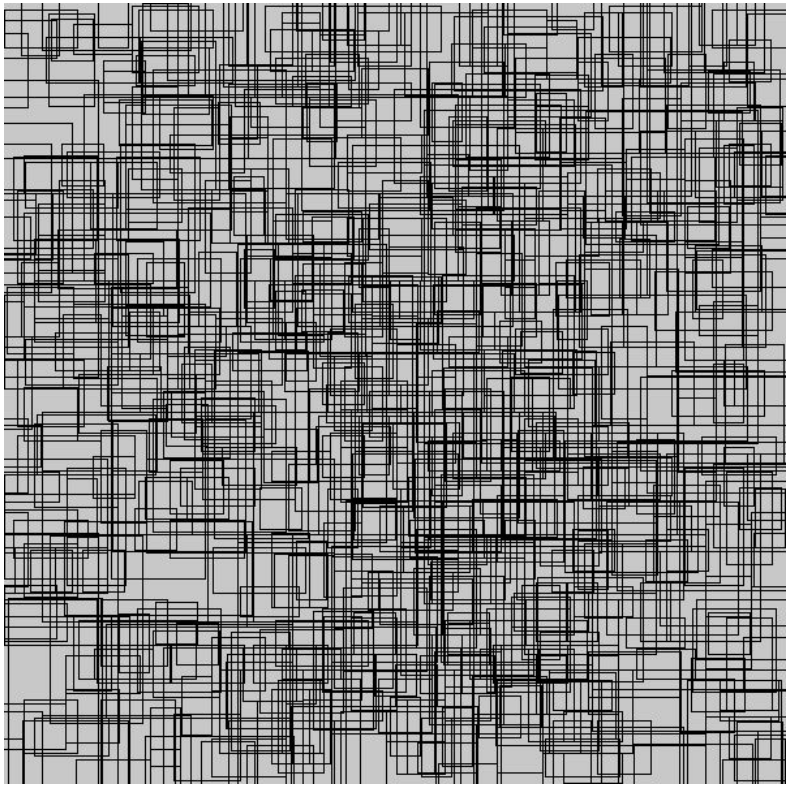
Feature Map



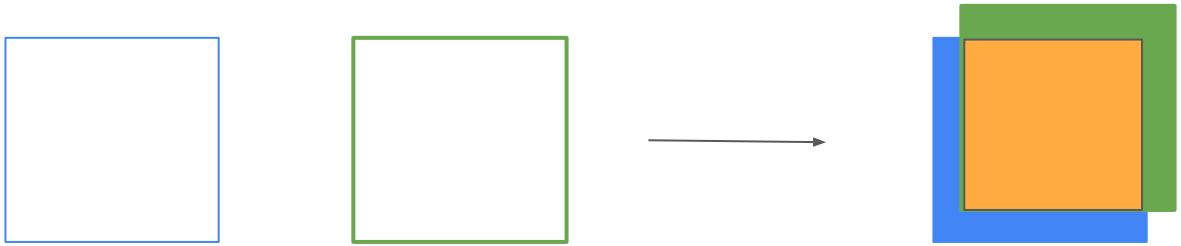
× k

= W.H.k anchor boxes

Anchor Boxes



Anchor Boxes



Anchor Box

Ground Truth

Intersection-over-Union (IoU)

Anchor with highest IoU with Ground Truth
OR
IoU > 0.7

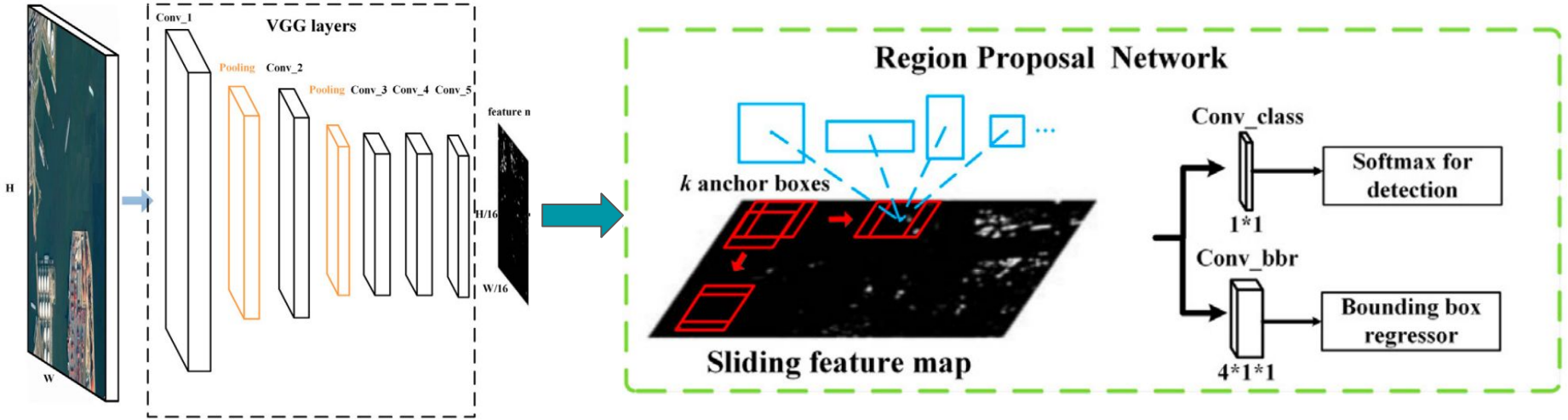
Object / Not Object

Training Faster-RCNN

- 4-Step Alternating Training

➔ Train RPN only (Initialization)
➔ Freeze RPN + Train (Backbone, Box Head, Class Head)
➔ Freeze (Backbone, Box Head, Class Head) + Train RPN
➔ Freeze Backbone + Train (RPN, Box Head, Class Head)

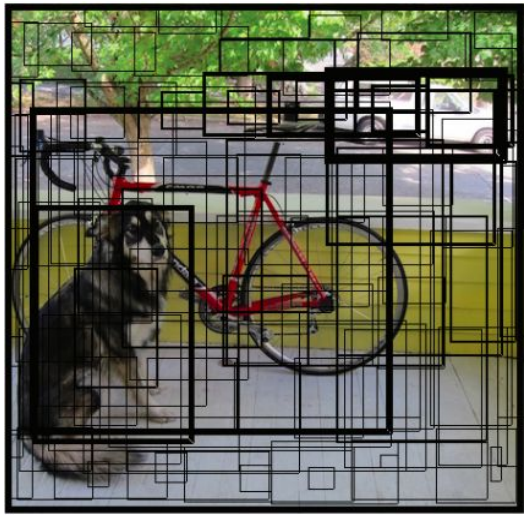
Training Faster-RCNN - Step 1

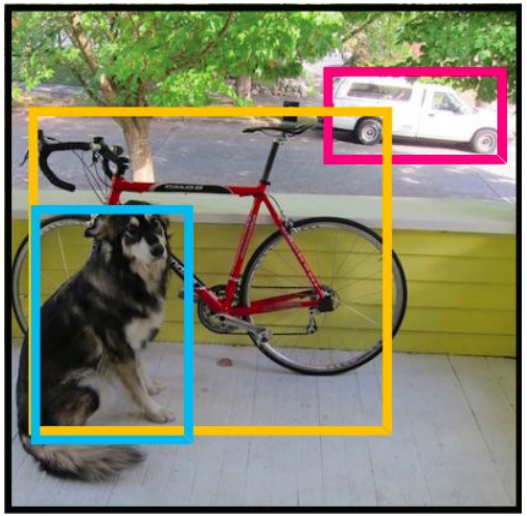


Backpropagation (ImageNet Initialization)

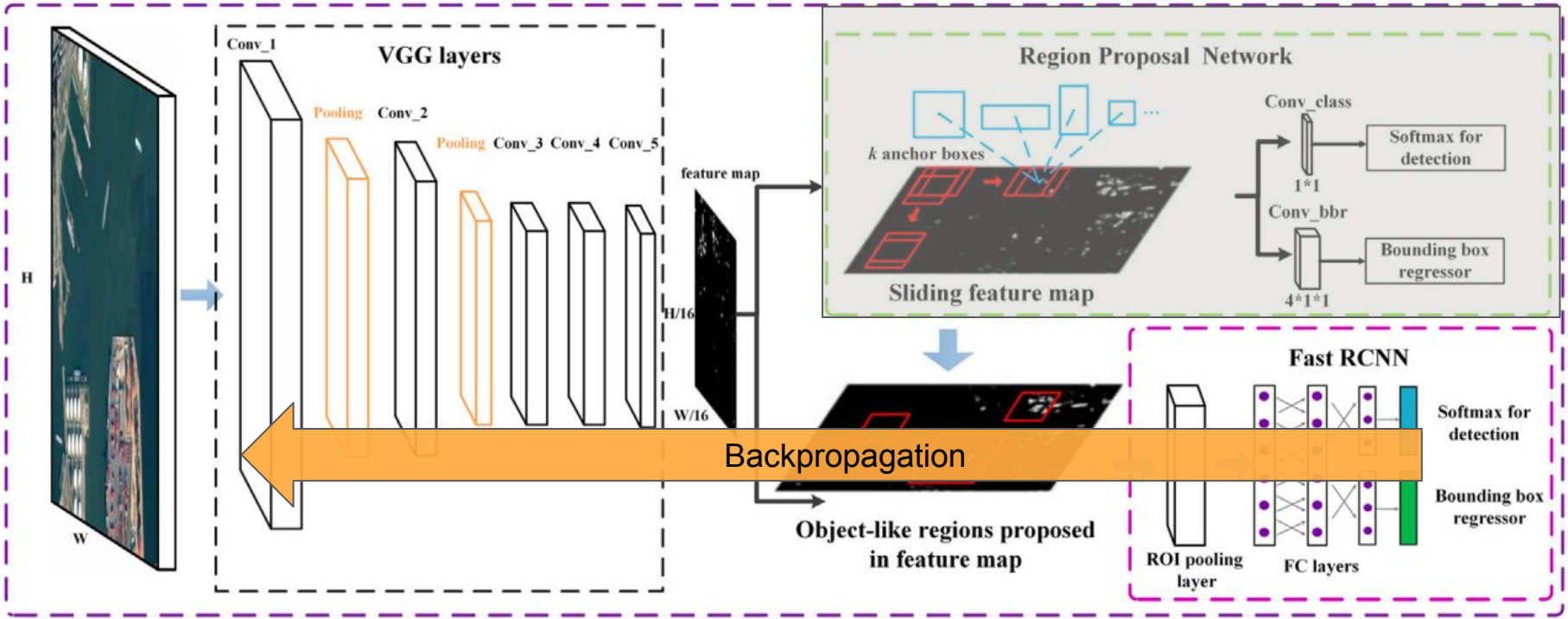
Training Faster-RCNN - Step 1

- Trained separately - End to End - BackProp - SGD.
- RPN Loss computed over 256 randomly sampled anchor boxes.
- To prevent bias from high number of negative anchors.

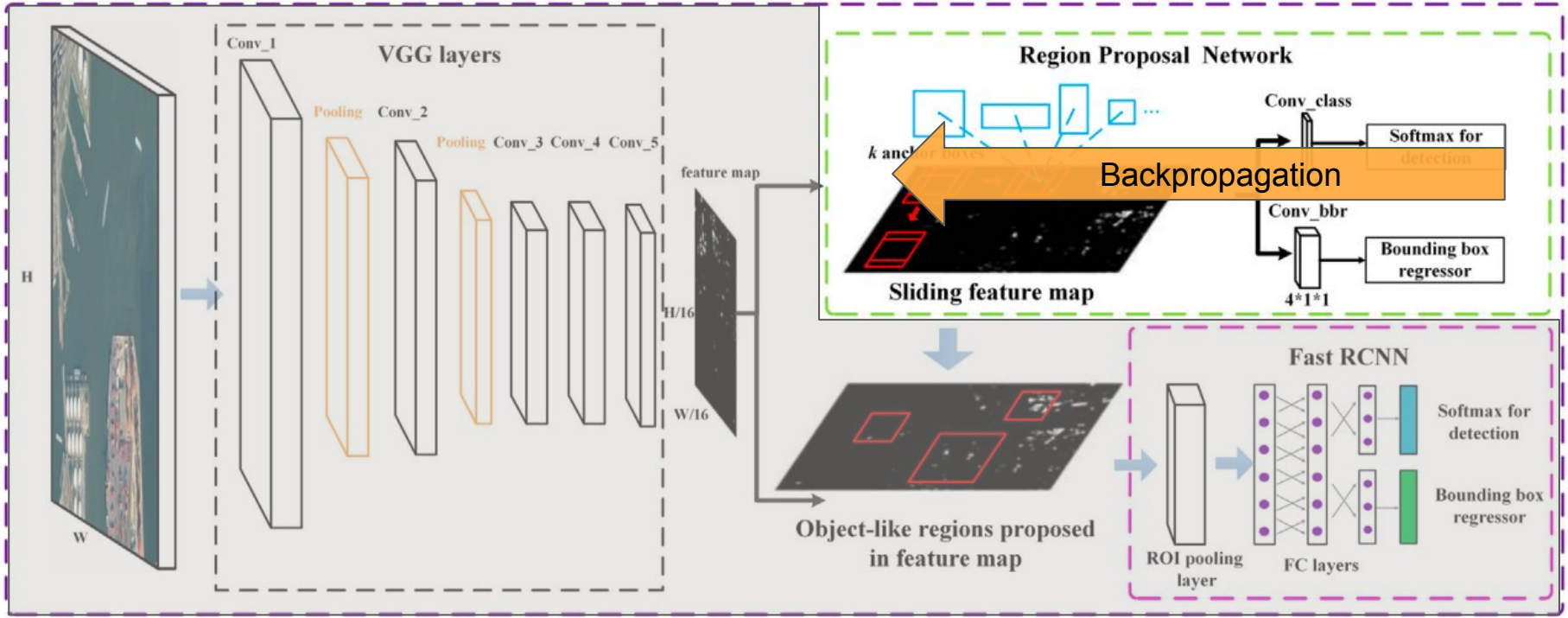




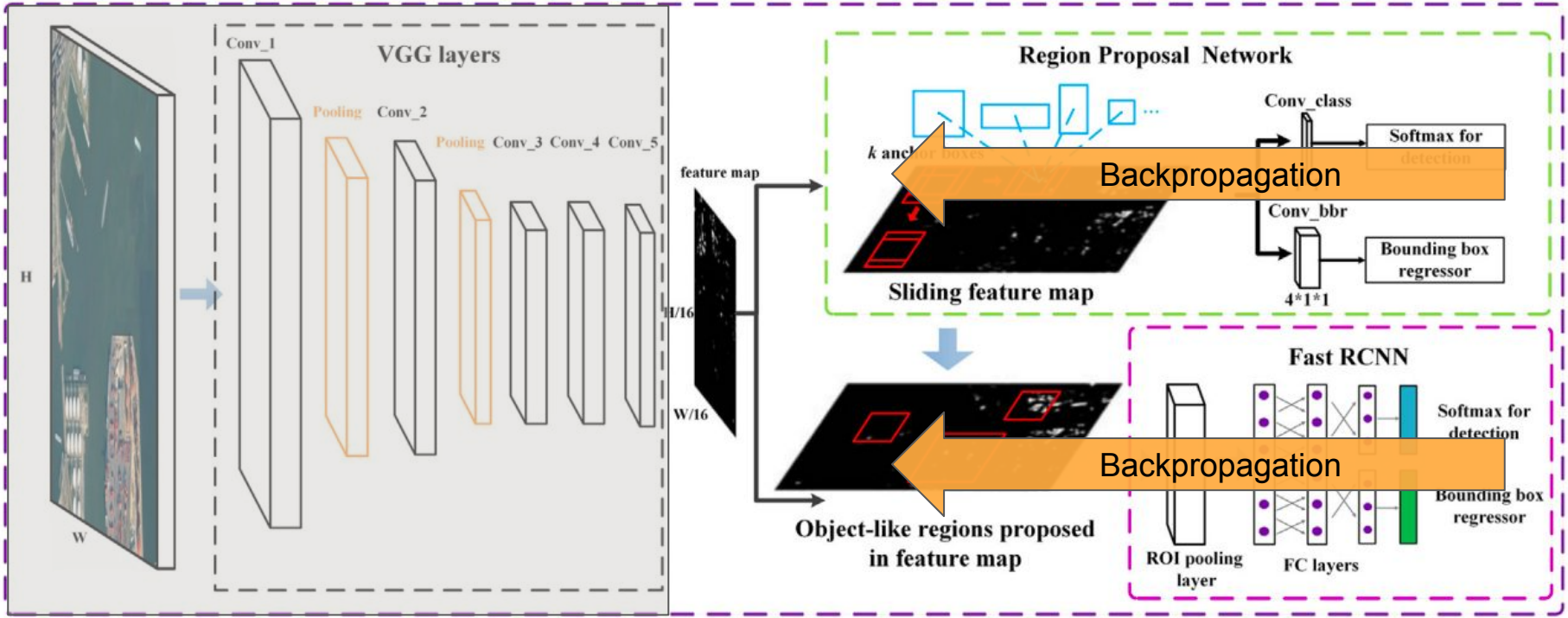
Training Faster-RCNN - Step 2



Training Faster-RCNN - Step 3

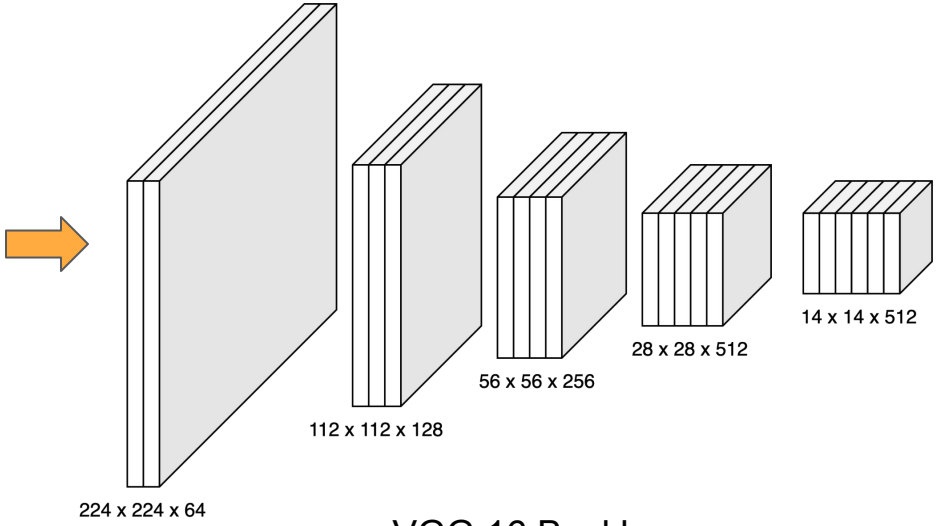


Training Faster-RCNN - Step 4



Deep Dive

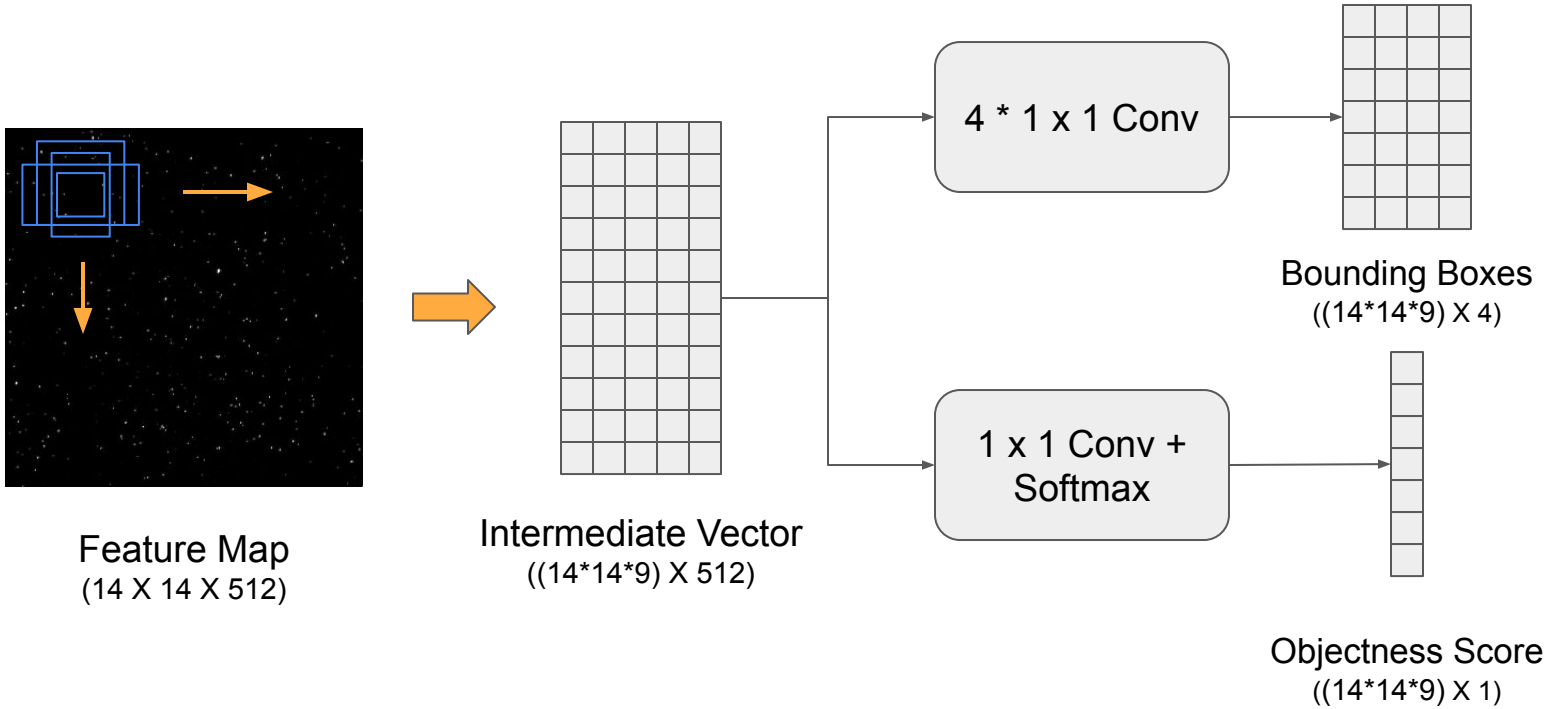
224 x 224 x 3



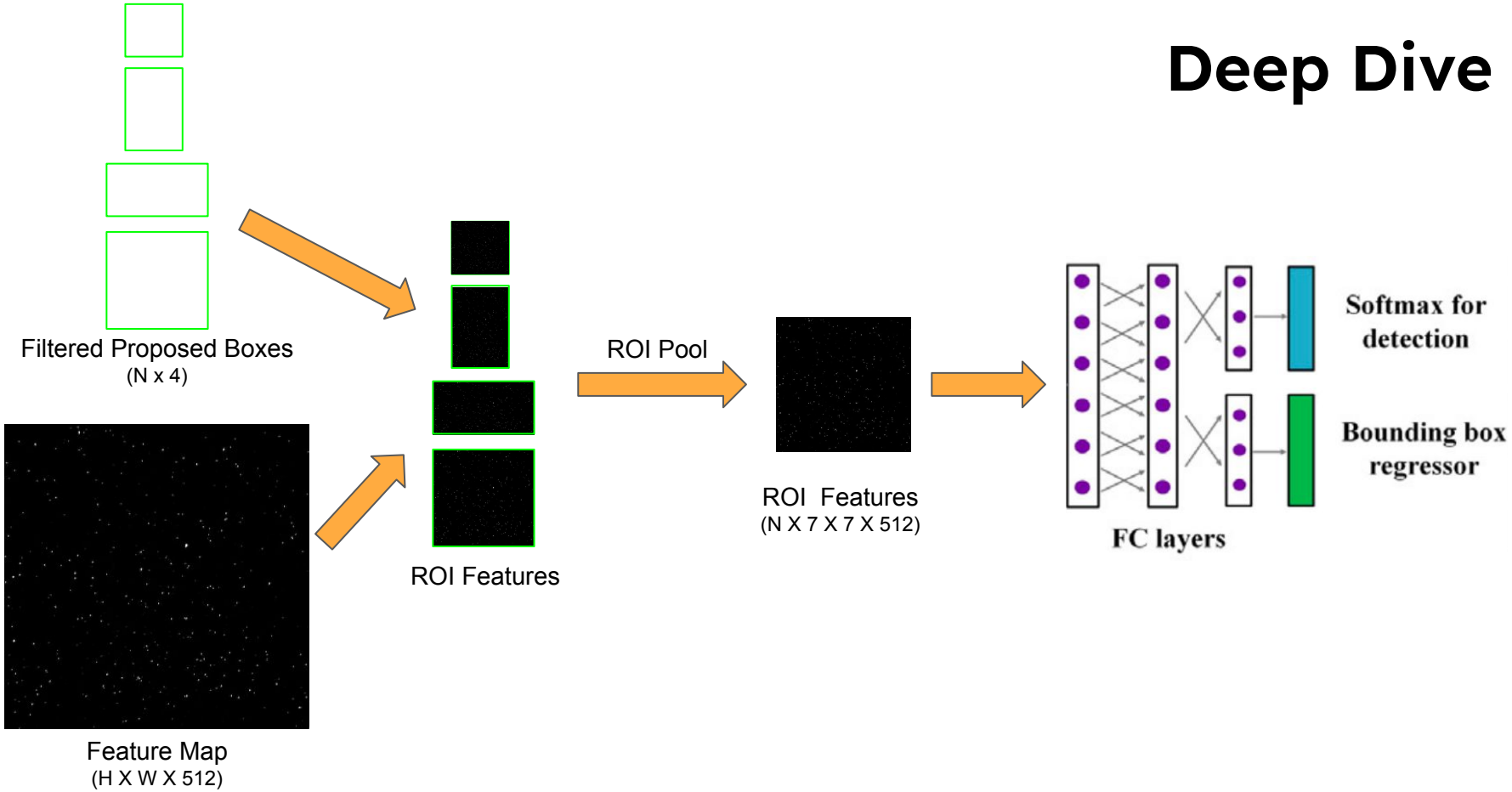
VGG 16 Backbone

Feature Map
(14 X 14 X 512)

Deep Dive


4 * 1 x 1 Conv
Bounding Boxes
((14*14*9) X 4)
1 x 1 Conv +
Softmax
Objectness Score
((14*14*9) X 1)
Intermediate Vector
((14*14*9) X 512)
Feature Map
(14 X 14 X 512)

Deep Dive


Filtered Proposed Boxes
(N x 4)
Feature Map
(H X W X 512)
ROI Features
ROI Pool
ROI Features
(N X 7 X 7 X 512)
FC layers
Softmax for detection
Bounding box regressor

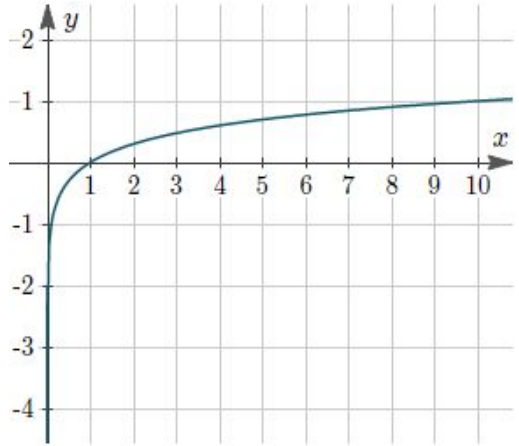
Loss Function

Categorical Cross Entropy Loss (Softmax Loss) for Class Prediction

$$Softmax \Rightarrow f(\hat{y})_i = \frac{e^{\hat{y}_i}}{\sum_j^C e^{\hat{y}_j}}$$

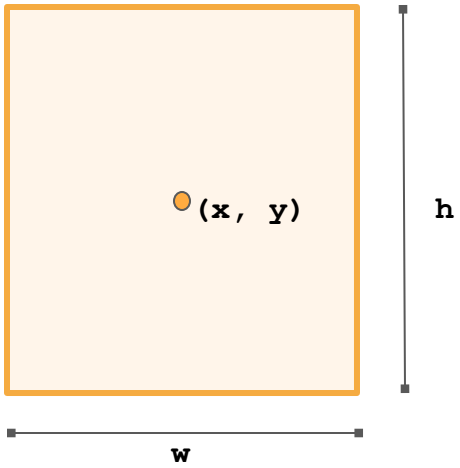
$$Cross\ Entropy \Rightarrow CE\left(f(\hat{y})_i\right) = -\sum_i^C y_i \log\left(f(\hat{y})_i\right)$$

$y =$	0	0	1	0	0	0
$\hat{y} =$	5	5	6	5	5	5
$f(\hat{y}) =$	.12	.12	.40	.12	.12	.12



Loss Functions

L2 Loss for Bounding Box coordinates Regression

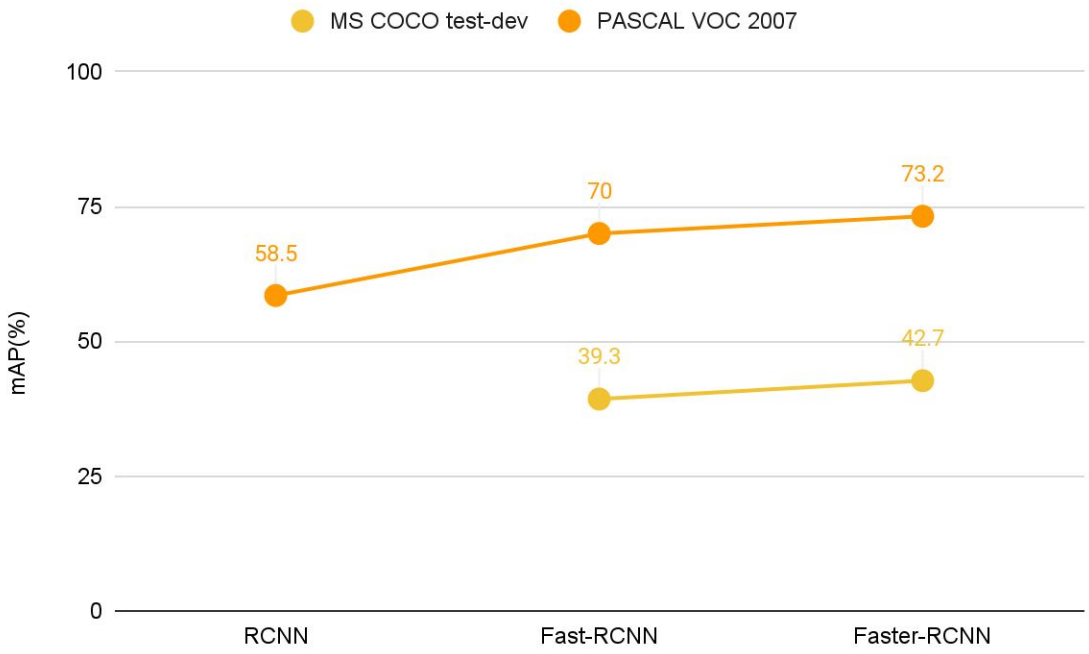


$$L_{reg} = \left(x - \hat{x}\right)^2 + \left(y - \hat{y}\right)^2 + \left(w - \hat{w}\right)^2 + \left(h - \hat{h}\right)^2$$

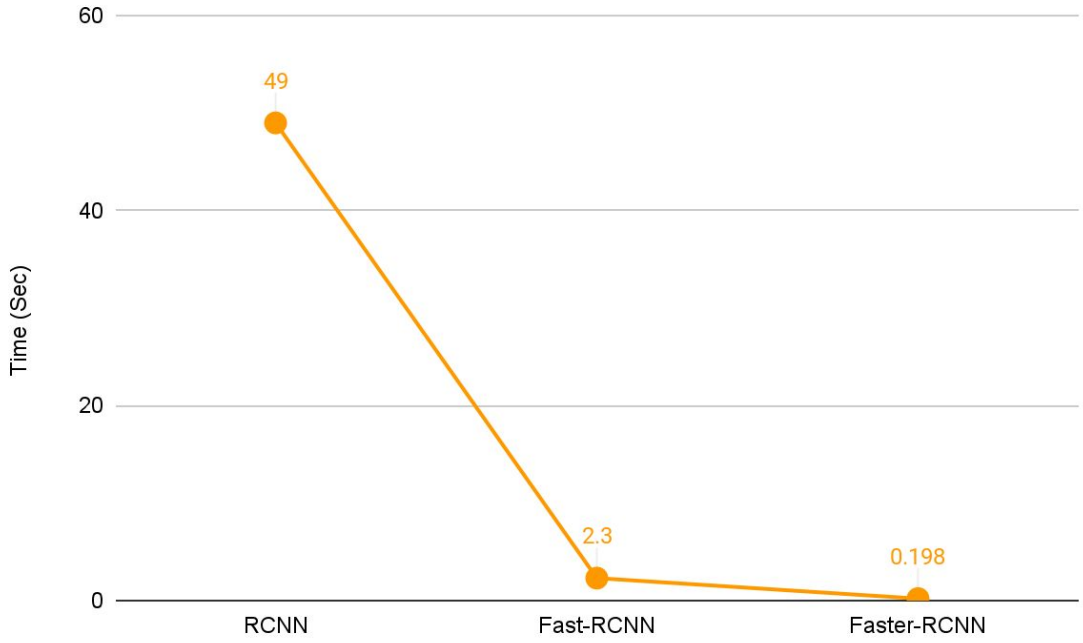
Multitask Loss Function

$$L\left(\{p_i\},\{t_i\}\right) = \frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda\sum_i p_i^* L_{reg}\left(t_i, t_i^*\right)$$

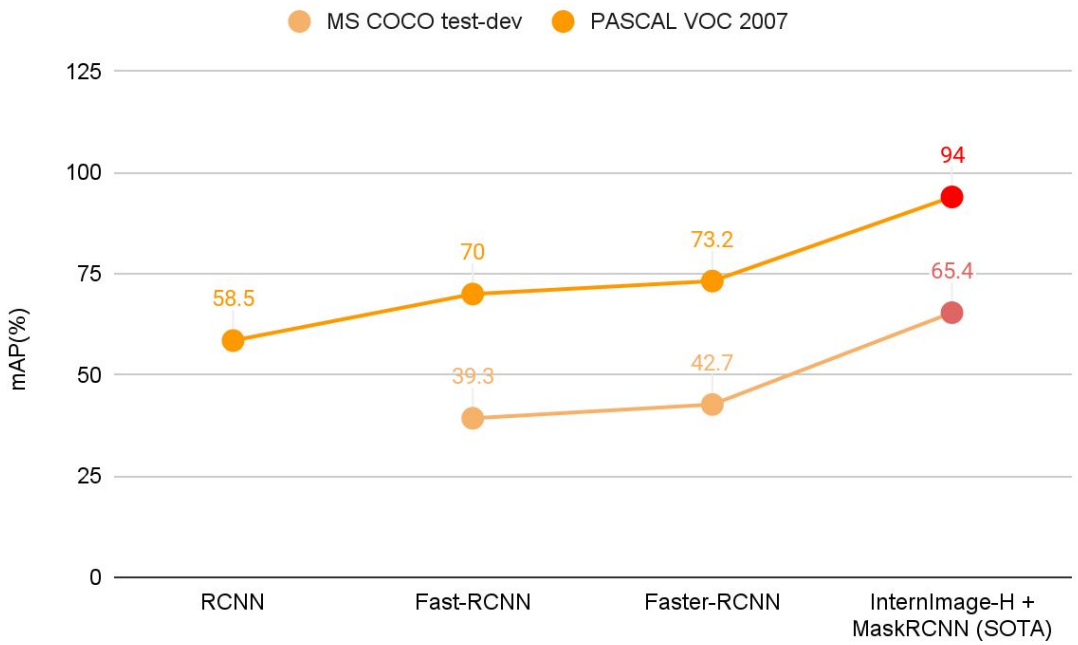
Performance Comparison



Speed Comparison



Performance Comparison Today

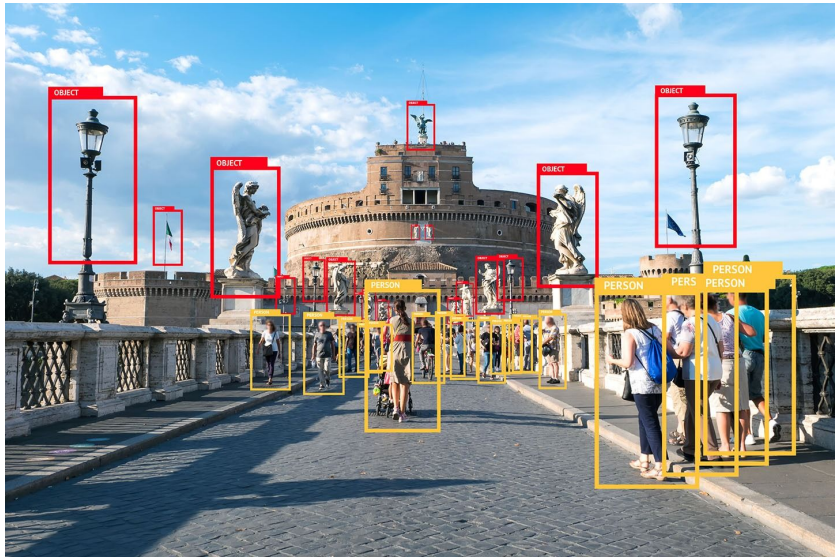


Further Advancement on Object detection

- After Faster R-CNN, there have been several advancements and improvements in the field of object detection. One of the most significant developments has been the introduction of **one-stage object detection models,** which have surpassed the accuracy of two-stage models while being significantly faster.

Some one-stage detection models include:

a. **YOLO**
b. **SSD**
c. **RetinaNet**
d. **EfficientDet**



YOLO (You Only Look Once)

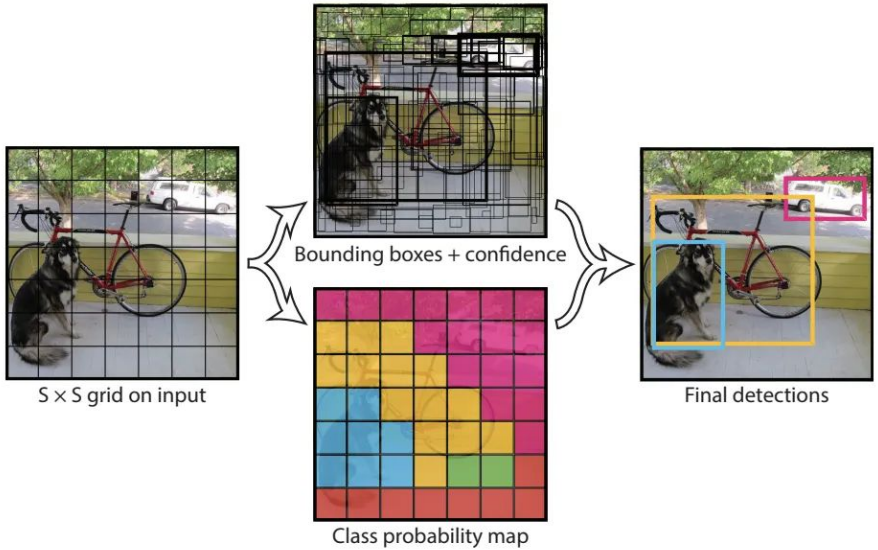
- The YOLO model was first published (by Joseph Redmon et al.) in 2015 and subsequently revised in two following papers.
- Directly predicts object bounding boxes and class probabilities from input images, without the need for an explicit region proposal step.
- YOLO achieves state-of-the-art accuracy in real-time object detection while being faster than two-stage models.

Why the name “You Only Look Once”?

- As typical for object detectors, the features learned by the convolutional layers are passed onto a classifier which makes the detection prediction. In YOLO, the prediction is based on a convolutional layer that uses 1×1 convolutions.
- The YOLO algorithm is named “you only look once” because its prediction uses 1×1 convolutions; this means that the size of the prediction map is exactly the size of the feature map before it.

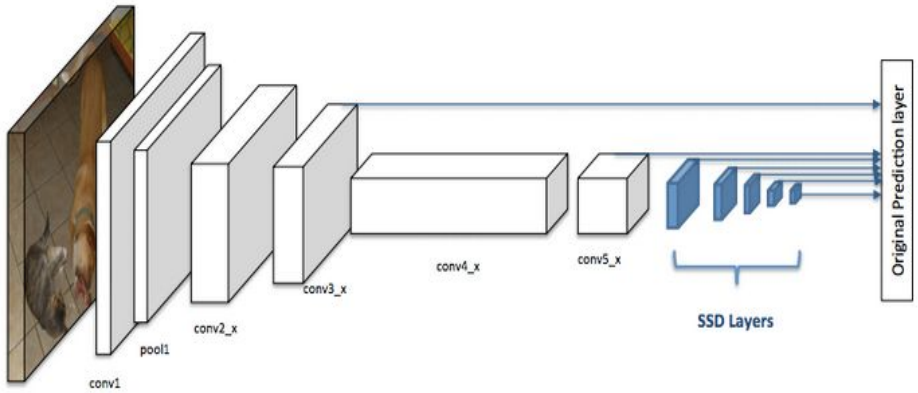
How YOLO works

- Divide the input image into a grid of cells.
- Predict bounding boxes, confidence scores, and class probabilities for each cell.
- Combine predictions across all cells and apply non-maximum suppression to output the final set of detected objects with their bounding boxes and class probabilities.



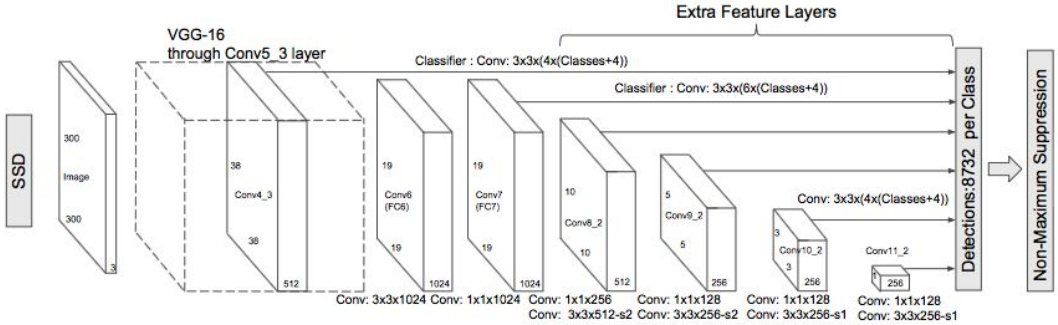
SSD (Single Shot Detection)

- SSD is an object detection algorithm that predicts object class and location in an image using a single neural network.
- It divides the image into fixed-size boxes and predicts class probabilities and offset values for each box.
- Predictions are filtered based on confidence scores to produce the final detections, making SSD accurate and efficient for real-time applications.



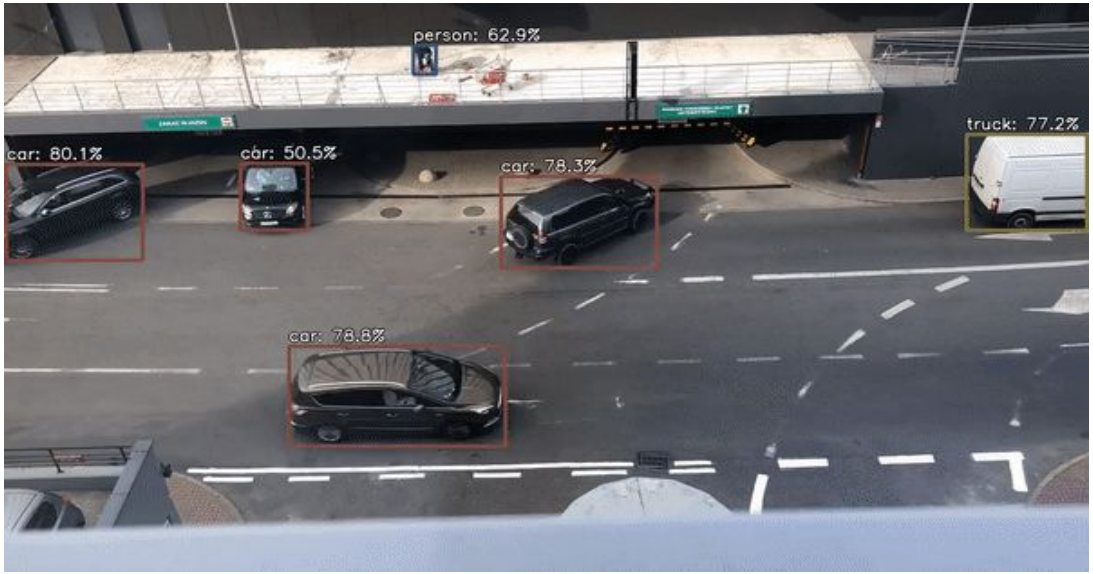
How SSD Works?

- SSD divides the input image into a set of default boxes at different scales and aspect ratios.
- It passes the input image through a convolutional neural network that predicts class probabilities and offsets for each default box to adjust its position and size to match the object in the image.
- The algorithm applies non-maximum suppression to filter overlapping bounding boxes and keep only the ones with the highest confidence scores.
- The final set of filtered bounding boxes, along with their corresponding class labels and confidence scores, is outputted as the object detection results.



RetinaNet

- RetinaNet is a state-of-the-art one-stage object detection algorithm introduced in 2017.
- It uses a feature pyramid network to extract feature maps of different resolutions and scales from an input image.

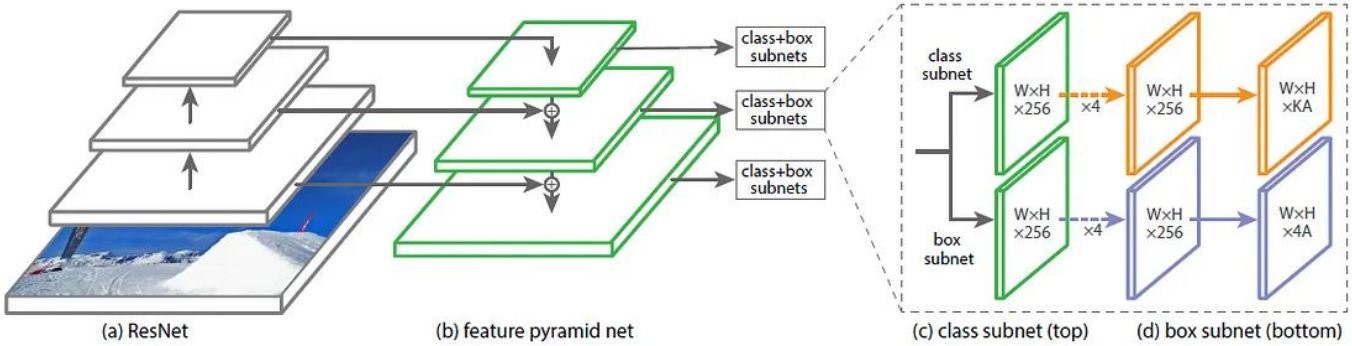


A Demo of RetinaNet on Parking Lot Entrance

- The algorithm then applies a novel focal loss function that assigns higher weights to hard examples during training to address the class imbalance problem in object detection.
- The final output is a set of bounding boxes with their corresponding class labels and confidence scores, which can be used for tasks such as object recognition and localization.

How RetinaNet Works?

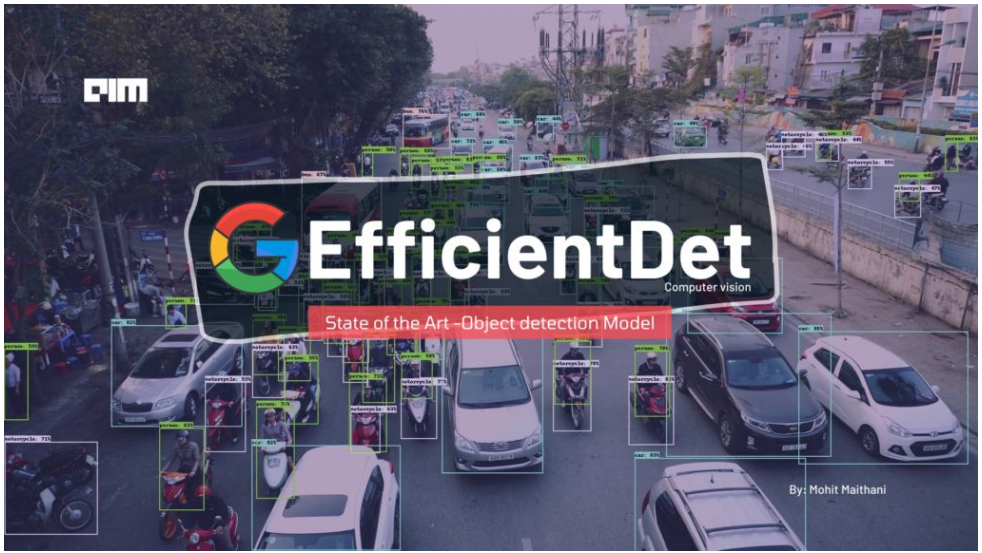
- RetinaNet uses a feature pyramid network (FPN) to extract features of different scales.
- It uses a novel focal loss function to improve detection accuracy.
- The architecture has a two-stage detection framework for generating candidate bounding boxes and classifying each box.
- A lightweight network, called a "head," performs the final classification and regression tasks on the candidate boxes.



RetinaNet Detector Architecture

EfficientDet

- EfficientDet is based on EfficientNet, a scalable neural network architecture that achieves high accuracy with low computational cost.
- EfficientDet uses a compound scaling method that scales up the network dimensions in a balanced way to achieve high accuracy and efficiency.
- The architecture uses a bi-directional feature network and a weighted feature fusion module to extract features at different scales and improve detection accuracy.



How EfficientDet works?

- EfficientDet is based on EfficientNet, a scalable neural network architecture that achieves high accuracy with low computational cost.
- EfficientDet uses a compound scaling method to scale up the network dimensions in a balanced way
- The architecture uses a bi-directional feature network that extracts features at multiple scales in both forward and backward directions

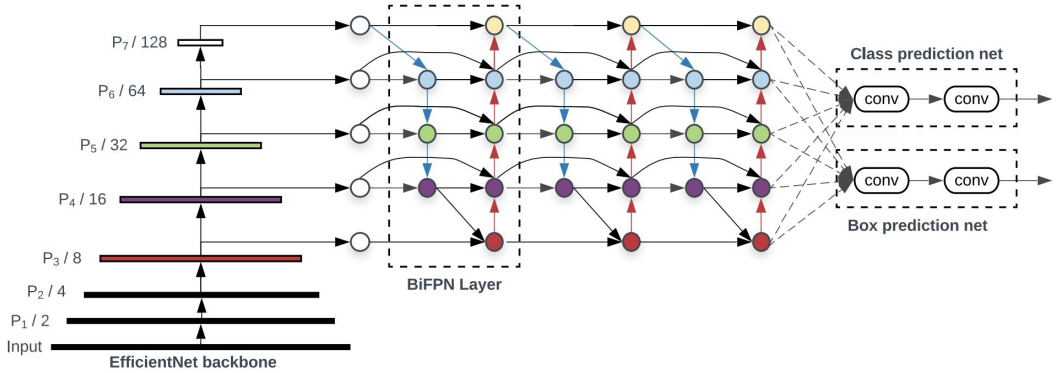


Figure 3: **EfficientDet architecture –** It employs EfficientNet [36] as the backbone network, BiFPN as the feature network, and shared class/box prediction network. Both BiFPN layers and class/box net layers are repeated multiple times based on different resource constraints as shown in Table 1.

Comparison

RCNN Models

- Multi-stage design leading to slower inference time
- Achieved state-of-the-art accuracy when first introduced
- Limited flexibility to perform tasks beyond object detection
- Can be computationally expensive to train
- Lack interpretability

Current state-of-the-art object detection frameworks:

- Designed to be faster and more efficient
- Achieve higher accuracy than RCNN models
- Offer greater flexibility to perform multiple tasks
- More efficient to train
- Aim to improve model interpretability

Conclusion

In conclusion, RCNN, Fast RCNN, and Faster RCNN have been crucial in shaping the field of object detection. However, with advancements such as Single Shot Detector (SSD), You Only Look Once (YOLO), and RetinaNet object detection has become faster and more accurate. Despite these advancements, RCNN models and their variants continue to be relevant and have paved the way for modern object detection methods.

thank you