MIDDLE EAST TECHNICAL UNIVERSITY

DI504 Foundations of Deep Learning

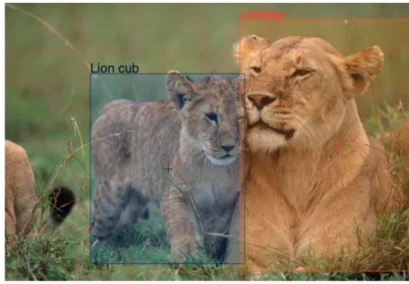
Advanced Architectures (Object detection)

This week:

- This week we continue our talks on some advanced architectural designs in deep learning.
- We will continue with object detection problem.

METU

Object Detection



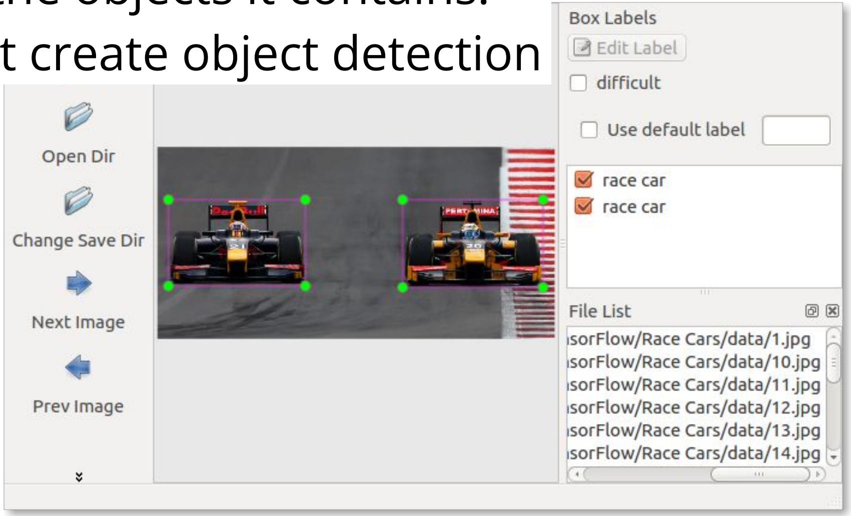
- While an image classification network can tell whether an image contains a certain object or not, it won’t say where in the image the object is located.
- Object detection networks provide both the class of objects contained in an image and a bounding box that provides the coordinates of that object.
- Object detection networks bear much resemblance to image classification networks and use convolution layers to detect visual features.
- In fact, most object detection networks use an image classification CNN and repurpose it for object detection.

METU

Object Detection

- Object detection is a supervised machine learning problem, which means you must train your models on labeled examples.
- Each image in the training dataset must be accompanied with a file that includes the boundaries and classes of the objects it contains.
- There are several open-source tools that create object detection annotations.

An easy format to use for image annotations is the PASCAL VOC file format. This is an XML file format used by Image Net. The LabelImg program is an excellent tool that can be used to generate and modify annotations of this format.



✓ race car
✓ race car

isorFlow/Race Cars/data/1.jpg
isorFlow/Race Cars/data/10.jpg
isorFlow/Race Cars/data/11.jpg
isorFlow/Race Cars/data/12.jpg
isorFlow/Race Cars/data/13.jpg
isorFlow/Race Cars/data/14.jpg

Object Detection

- The major reason why you cannot proceed with this problem by building a standard convolutional network followed by a fully connected layer is that, the length of the output layer is variable — not constant, this is because the number of occurrences of the objects of interest is not fixed.
- A naive approach to solve this problem would be to take different regions of interest from the image, and use a CNN to classify the presence of the object within that region.
- The problem with this approach is that the objects of interest might have different spatial locations within the image and different aspect ratios. Hence, you would have to select a huge number of regions and this could computationally blow up.

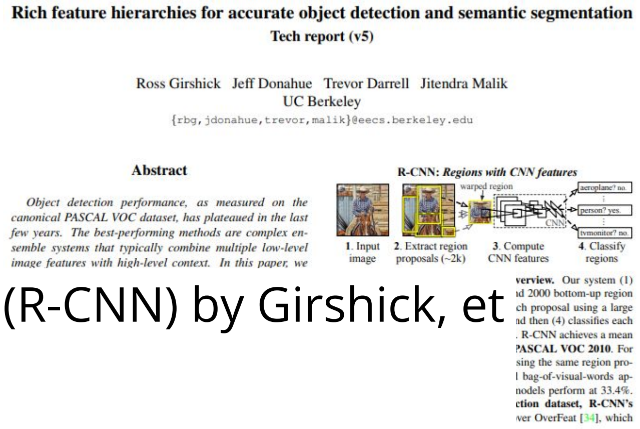
METU

Object Detection

- The object detection network is trained on the annotated data until it can find regions in images that correspond to each kind of object.
- Now let’s look at a few object-detection neural network architectures:
    - R-CNN
    - Fast R-CNN
    - Faster R-CNN
    - YOLO
- These architectures are examples that aptly summarize evolutionary development in object detection.

METU

R-CNN


Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

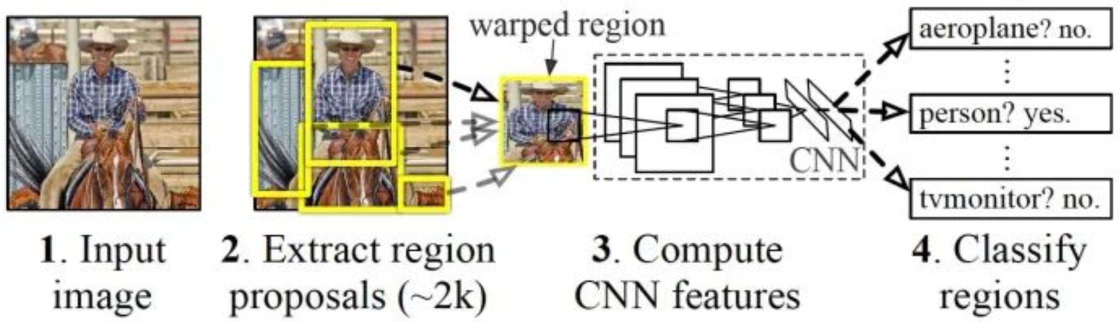
Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we

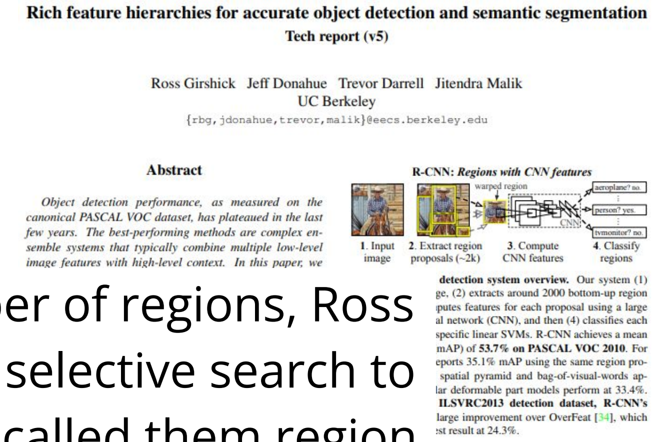
R-CNN: *Regions with CNN features*

verview. Our system (1)
ıd 2000 bottom-up region
ch proposal using a large
ıd then (4) classifies each
. R-CNN achieves a mean
PASCAL VOC 2010. For
sing the same region pro-
l bag-of-visual-words ap-
nodels perform at 33.4%.
ction dataset, R-CNN's
ver OverFeat [34], which

- The Region-based Convolutional Neural Network (R-CNN) by Girshick, et al, 2014, is composed of three key components:
  - Region selector
  - CNN
  - Classifier



R-CNN (Region Selection)



Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

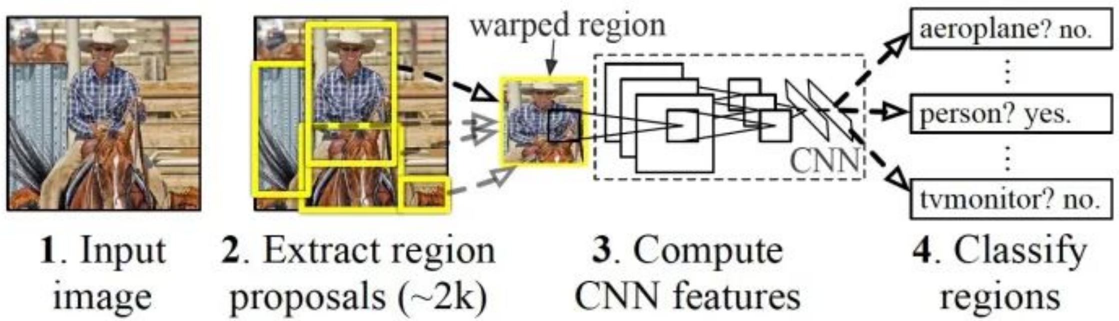
Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we

R-CNN: *Regions with CNN features*

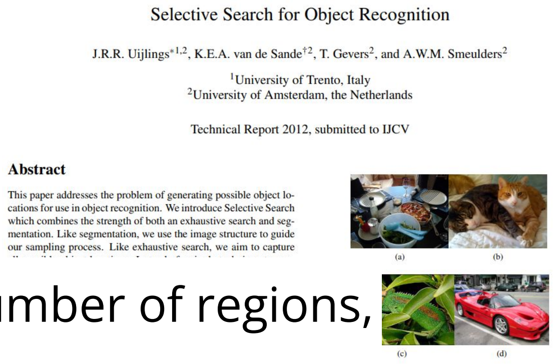
detection system overview. Our system (1) ge, (2) extracts around 2000 bottom-up region iputes features for each proposal using a large al network (CNN), and then (4) classifies each specific linear SVMs. R-CNN achieves a mean mAP) of **53.7% on PASCAL VOC 2010**. For eports 35.1% mAP using the same region pro- spatial pyramid and bag-of-visual-words ap- lar deformable part models perform at 33.4%. **ILSVRC2013 detection dataset, R-CNN's** large improvement over OverFeat [34], which st result at 24.3%.

- To bypass the problem of selecting a huge number of regions, Ross Girshick et al. proposed a method where we use selective search to extract just 2000 regions from the image and he called them region proposals.



METU

R-CNN (Region Selection)



Selective Search for Object Recognition

J.R.R. Uijlings[*1,2], K.E.A. van de Sande[†2], T. Gevers[2], and A.W.M. Smeulders[2]

[1]University of Trento, Italy
[2]University of Amsterdam, the Netherlands

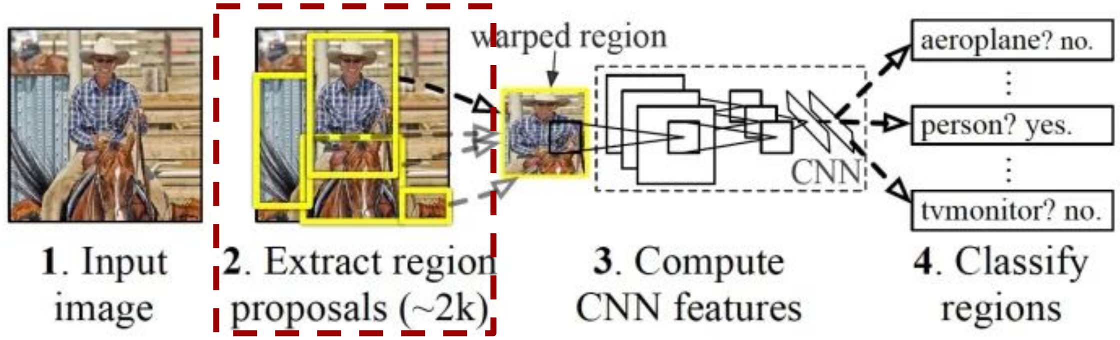
Technical Report 2012, submitted to IJCV

Abstract

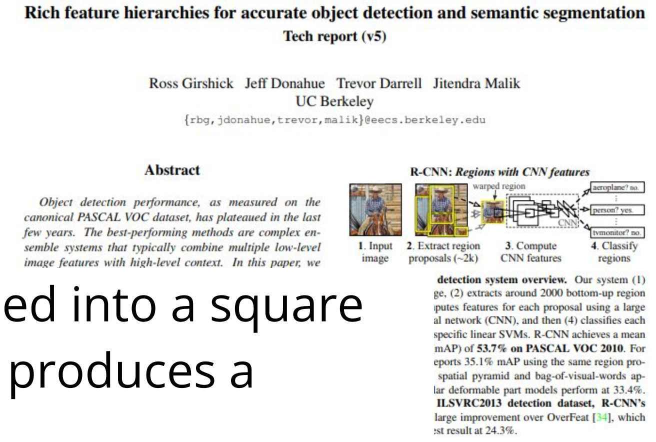
This paper addresses the problem of generating possible object locations for use in object recognition. We introduce Selective Search which combines the strength of both an exhaustive search and segmentation. Like segmentation, we use the image structure to guide our sampling process. Like exhaustive search, we aim to capture

(a) (b) (c) (d)

- Therefore, now, instead of trying to classify a huge number of regions, you can just work with 2000 regions.
- These 2000 region proposals are generated using the selective search algorithm
  a. Generate initial sub-segmentation, we generate many candidate regions
  b. Use greedy algorithm to recursively combine similar regions into larger ones
  c. Use the generated regions to produce the final candidate region proposals



METU



Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Abstract

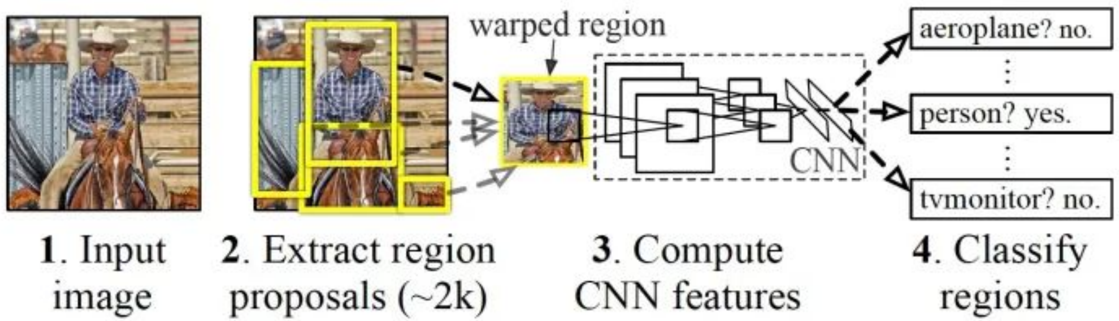
Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we

R-CNN: *Regions with CNN features*

detection system overview. Our system (1) ge, (2) extracts around 2000 bottom-up region iputes features for each proposal using a large al network (CNN), and then (4) classifies each specific linear SVMs. R-CNN achieves a mean mAP) of **53.7% on PASCAL VOC 2010**. For eports 35.1% mAP using the same region pro- spatial pyramid and bag-of-visual-words ap- lar deformable part models perform at 33.4%. **ILSVRC2013 detection dataset, R-CNN's** large improvement over OverFeat [34], which est result at 24.3%.

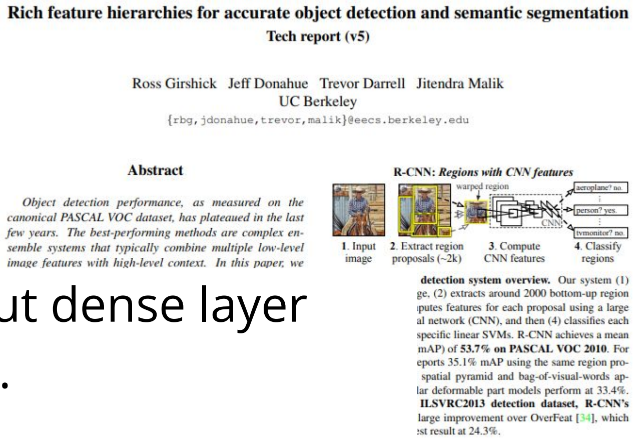
R-CNN (Region Selection)

- These 2000 candidate region proposals are warped into a square and fed into a convolutional neural network that produces a 4096-dimensional feature vector as output.



METU

R-CNN (the CNN)



Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

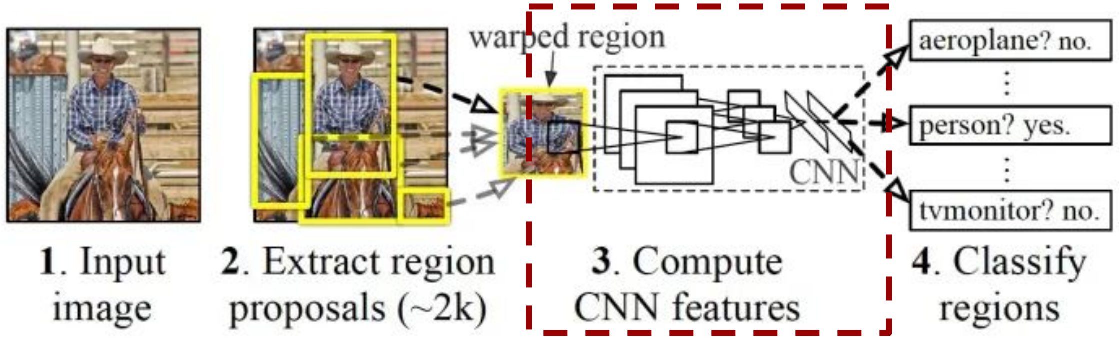
Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we

R-CNN: *Regions with CNN features*

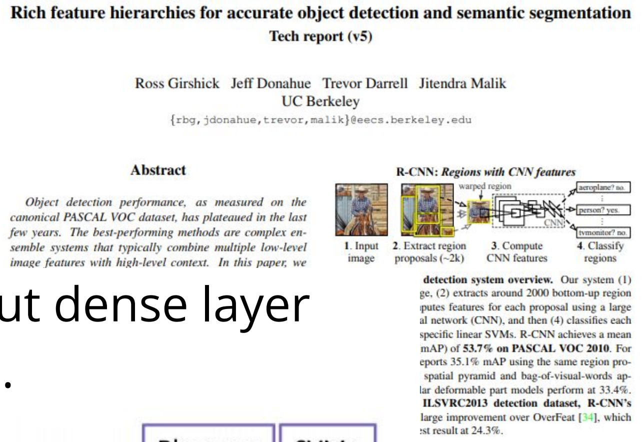
detection system overview. Our system (1) ge, (2) extracts around 2000 bottom-up region iputes features for each proposal using a large al network (CNN), and then (4) classifies each specific linear SVMs. R-CNN achieves a mean mAP) of **53.7% on PASCAL VOC 2010**. For eports 35.1% mAP using the same region pro- spatial pyramid and bag-of-visual-words ap- lar deformable part models perform at 33.4%. **ILSVRC2013 detection dataset, R-CNN's** large improvement over OverFeat [34], which :st result at 24.3%.

- The CNN acts as a feature extractor and the output dense layer consists of the features extracted from the image.



METU

R-CNN (the CNN)



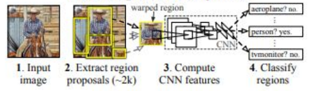
Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we

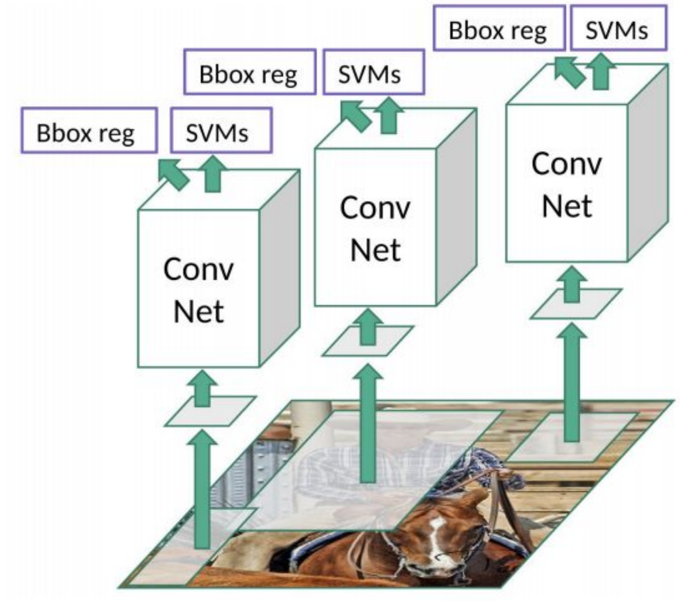
R-CNN: *Regions with CNN features*



detection system overview. Our system (1)
ge, (2) extracts around 2000 bottom-up region
putes features for each proposal using a large
al network (CNN), and then (4) classifies each
specific linear SVMs. R-CNN achieves a mean
mAP) of **53.7% on PASCAL VOC 2010**. For
eports 35.1% mAP using the same region pro-
spatial pyramid and bag-of-visual-words ap-
lar deformable part models perform at 33.4%.
ILSVRC2013 detection dataset, R-CNN's
large improvement over OverFeat [34], which
st result at 24.3%.

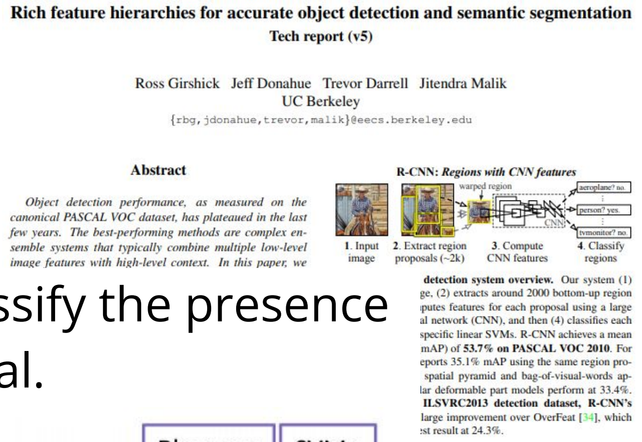
- The CNN acts as a feature extractor and the output dense layer consists of the features extracted from the image.

- The CNN processes every region separately extracts the features through a series of convolution operations. The CNN uses fully connected layers to encode the feature maps into a single-dimensional vector of numerical values.



METU

R-CNN (classifier)


Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we

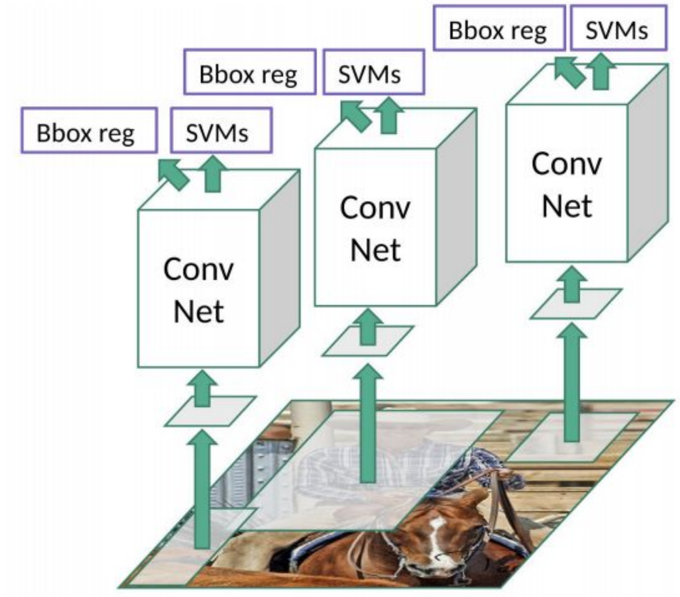
R-CNN: *Regions with CNN features*

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

detection system overview. Our system (1) ge, (2) extracts around 2000 bottom-up region iputes features for each proposal using a large al network (CNN), and then (4) classifies each specific linear SVMs. R-CNN achieves a mean mAP) of **53.7% on PASCAL VOC 2010**. For eports 35.1% mAP using the same region pro- spatial pyramid and bag-of-visual-words ap- lar deformable part models perform at 33.4%. **ILSVRC2013 detection dataset, R-CNN's** large improvement over OverFeat [34], which :st result at 24.3%.

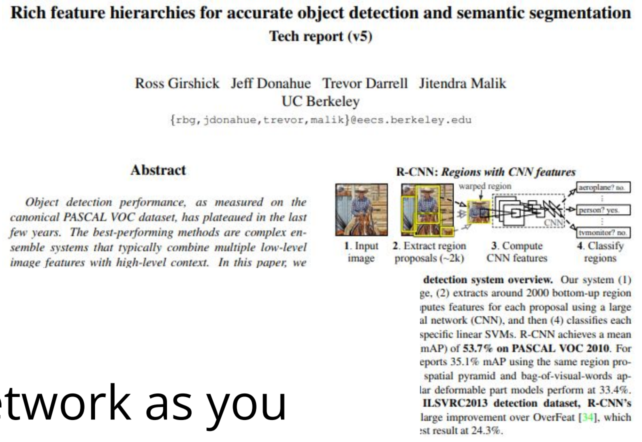
- The extracted features are fed into an SVM to classify the presence of the object within that candidate region proposal.

- Finally, the SVM classifier model maps the encoded features obtained from the CNN to the output classes. The classifier has a separate output class for "background," which corresponds to anything that isn't an object.



METU

R-CNN



Rich feature hierarchies for accurate object detection and semantic segmentation
Tech report (v5)

Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we

R-CNN: *Regions with CNN features*

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

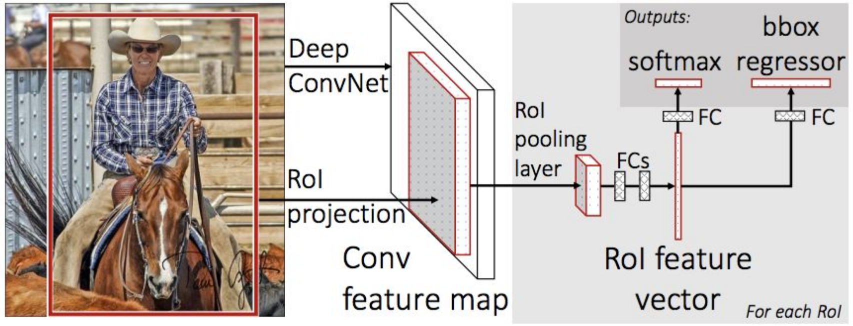
detection system overview. Our system (1) ge, (2) extracts around 2000 bottom-up region putes features for each proposal using a large al network (CNN), and then (4) classifies each specific linear SVMs. R-CNN achieves a mean mAP) of **53.7% on PASCAL VOC 2010**. For eports 35.1% mAP using the same region pro- spatial pyramid and bag-of-visual-words ap- lar deformable part models perform at 33.4%. **ILSVRC2013 detection dataset, R-CNN's** large improvement over OverFeat [34], which st result at 24.3%.

Problems with R-CNN

- It still takes a huge amount of time to train the network as you would have to classify 2k region proposals / image. So, it cannot be implemented real time.
- The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals.
- In the end, **y**ou **o**nly **l**ook **o**nce to an image. Not 2k times!

Fast R-CNN

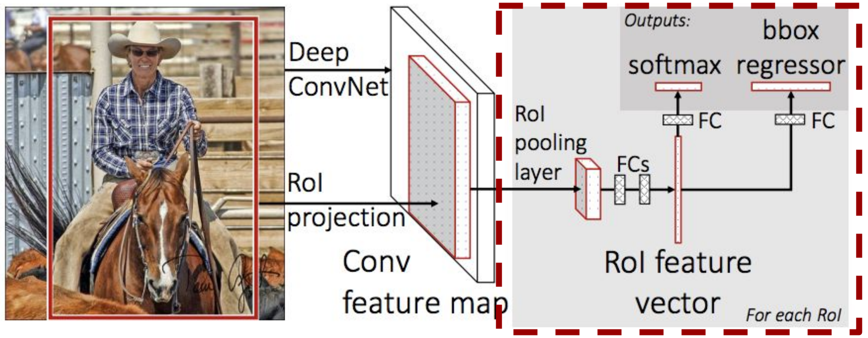
- Same guys solved some of the drawbacks of R-CNN to build a faster object detection algorithm and they called the Fast R-CNN.
- The approach is similar to the R-CNN algorithm. But, instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map.



METU

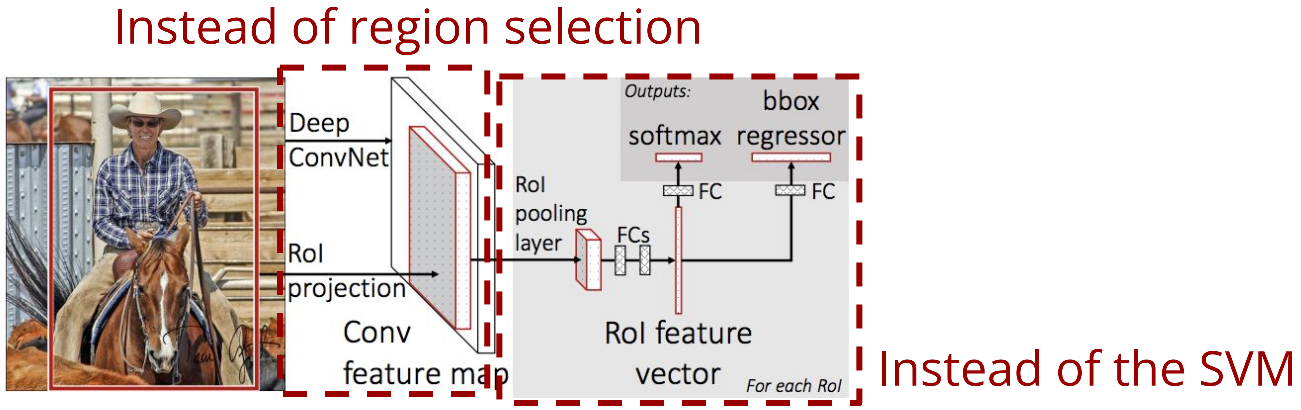
Fast R-CNN

- From the convolutional feature map, they identify the region of proposals and warp them into squares and by using a **RoI** pooling layer they reshape them into a fixed size so that it can be fed into a fully connected layer.
- From the RoI feature vector, we use a softmax layer to predict the class of the proposed region and also the offset values for the bounding box.



Fast R-CNN

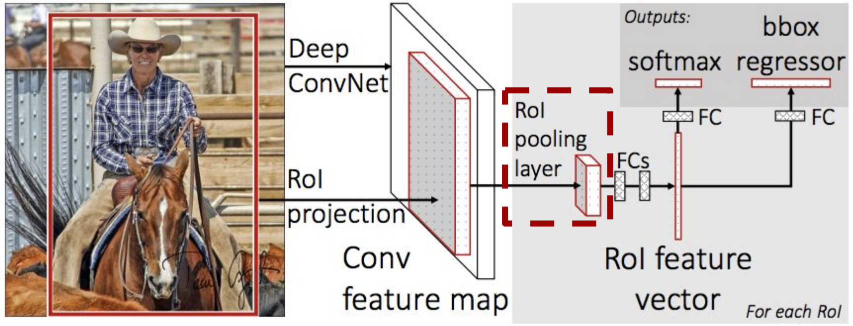
- The reason "Fast R-CNN" is faster than R-CNN is because you don't have to feed 2000 region proposals to the convolutional neural network every time.
- Instead, the convolution operation is done only once per image and a feature map is generated from it.



METU

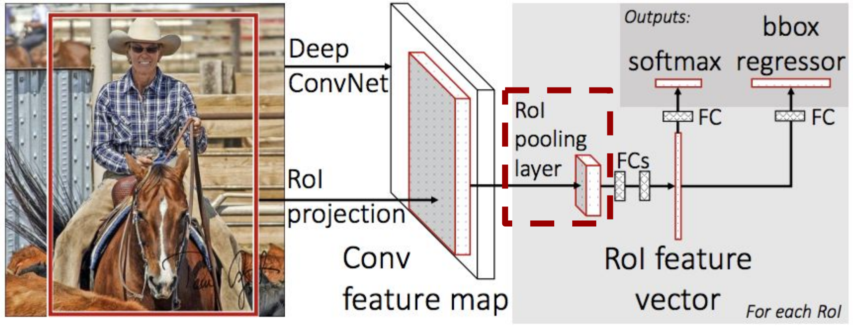
Fast R-CNN

- One of the key innovations in Fast R-CNN was the “RoI pooling layer,” an operation that takes CNN feature maps and regions of interest for an image and provides the corresponding features for each region.
- This allowed Fast R-CNN to extract features for all the regions of interest in the image in a single pass as opposed to R-CNN, which processed each region separately. This resulted in a significant boost in speed.



Fast R-CNN

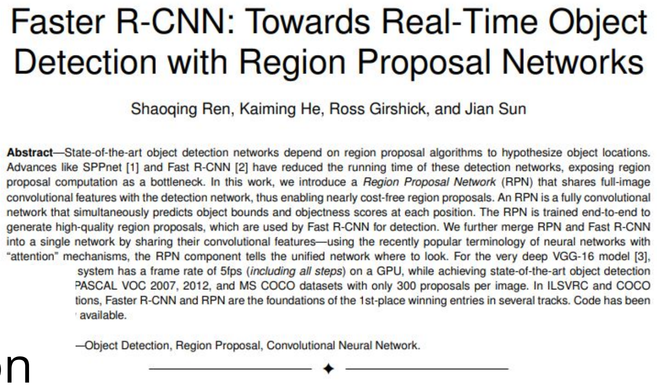
- However, one issue remained unsolved. Fast R-CNN still required the regions of the image to be extracted and provided as input to the model. Fast R-CNN was still not ready for real-time object detection.
- In the end, **y**ou **o**nly **l**ook **o**nce to an image.



METU

Faster R-CNN

- Faster R-CNN, introduced by the same group in 2016, solves the final piece of the object-detection puzzle by integrating the region extraction mechanism into the object detection network.
- Faster R-CNN takes an image as input and returns a list of object classes and their corresponding bounding boxes.

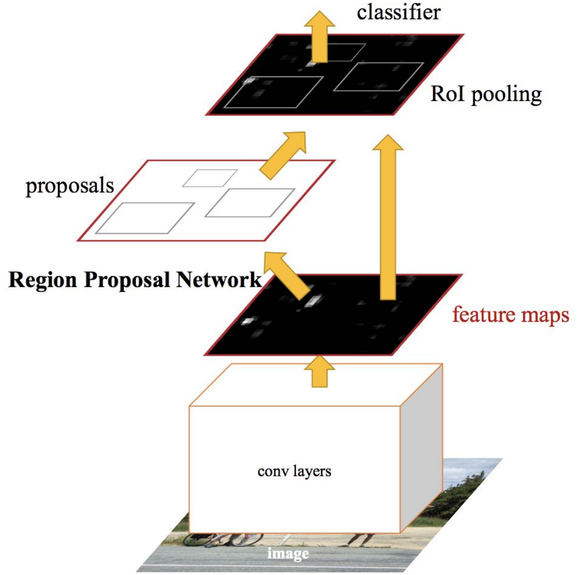


Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

Abstract—State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a *Region Proposal Network* (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with "attention" mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model [3], system has a frame rate of 5fps (*including all steps*) on a GPU, while achieving state-of-the-art object detection PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO tions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been available.

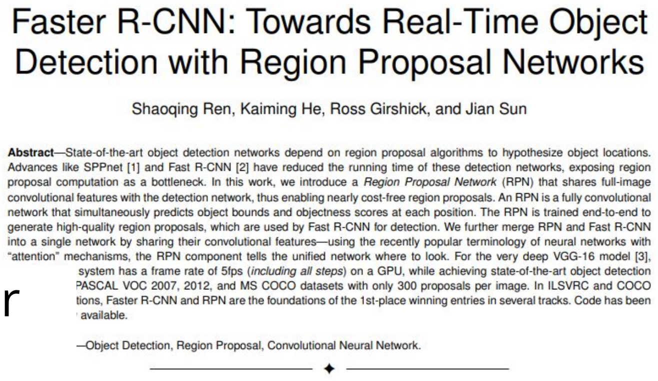
—Object Detection, Region Proposal, Convolutional Neural Network.



METU

Faster R-CNN

- The architecture of Faster R-CNN is largely similar to that of Fast R-CNN. Its main innovation is the “region proposal network” (RPN), a component that takes the feature maps produced by a convolutional neural network and proposes a set of bounding boxes where objects might be located. T
- he proposed regions are then passed to the RoI pooling layer. The rest of the process is similar to Fast R-CNN.

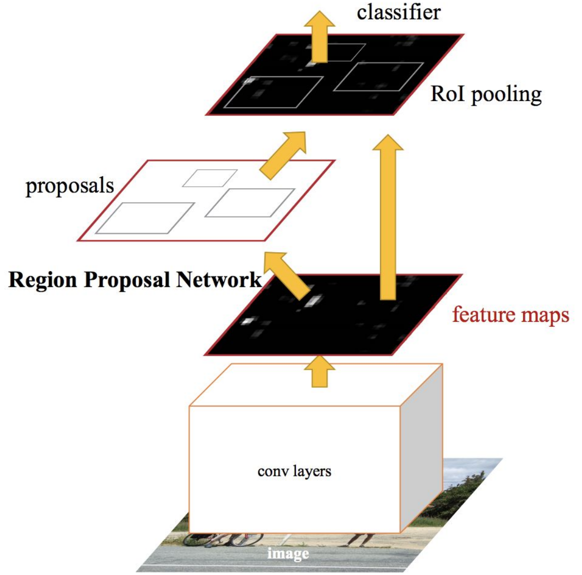


Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

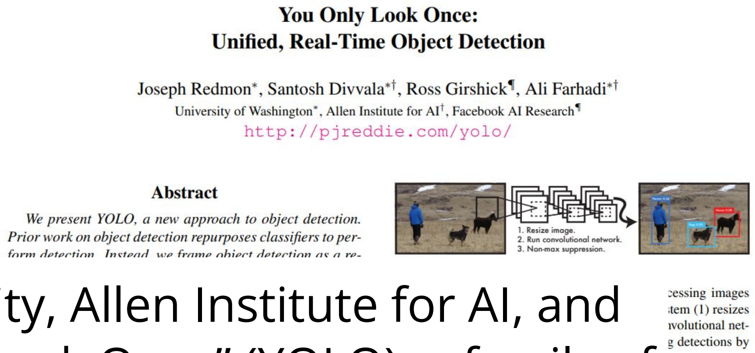
Abstract—State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a *Region Proposal Network* (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with “attention” mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model [3], system has a frame rate of 5fps (*including all steps*) on a GPU, while achieving state-of-the-art object detection PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO tions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been available.

—Object Detection, Region Proposal, Convolutional Neural Network.



METU

YOLO



You Only Look Once:
Unified, Real-Time Object Detection

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to per- form detection. Instead, we frame object detection as a re-

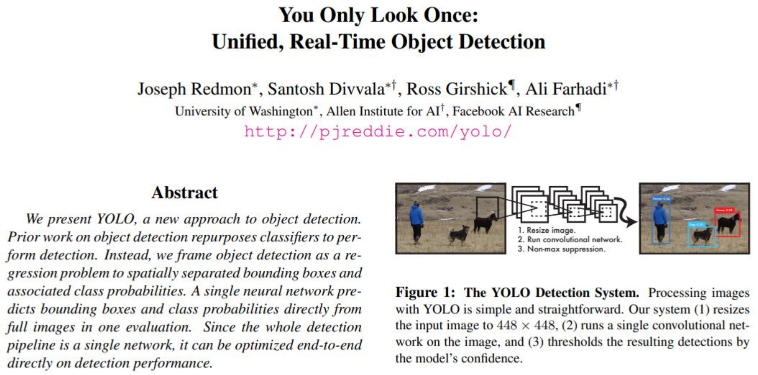
1. Resize image.
2. Run convolutional network.
3. Non-max suppression.

cessing images
tem (1) resizes
nvolutional net-
g detections by

- In 2016, researchers at Washington University, Allen Institute for AI, and Facebook AI Research proposed "You Only Look Once" (YOLO), a family of neural networks that improved the speed and accuracy of object detection with deep learning.
- The main improvement in YOLO is the integration of the entire object detection and classification process in a single network. Instead of extracting features and regions separately, YOLO performs everything in a single pass through a single network, hence the name "You Only Look Once."
- YOLO can perform object detection at video streaming frame rates and is suitable applications that require real-time inference.

YOLO

- How YOLO works is that we take an image and split it into an SxS grid, within each of the grid we take m bounding boxes.
- For each of the bounding box, the network outputs a class probability and offset values for the bounding box.
- The bounding boxes having the class probability above a threshold value is selected and used to locate the object within the image.



You Only Look Once:
Unified, Real-Time Object Detection

Joseph Redmon[*], Santosh Divvala[*†], Ross Girshick[¶], Ali Farhadi[*†]
University of Washington[*], Allen Institute for AI[†], Facebook AI Research[¶]
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

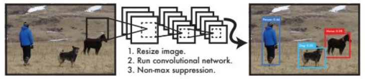
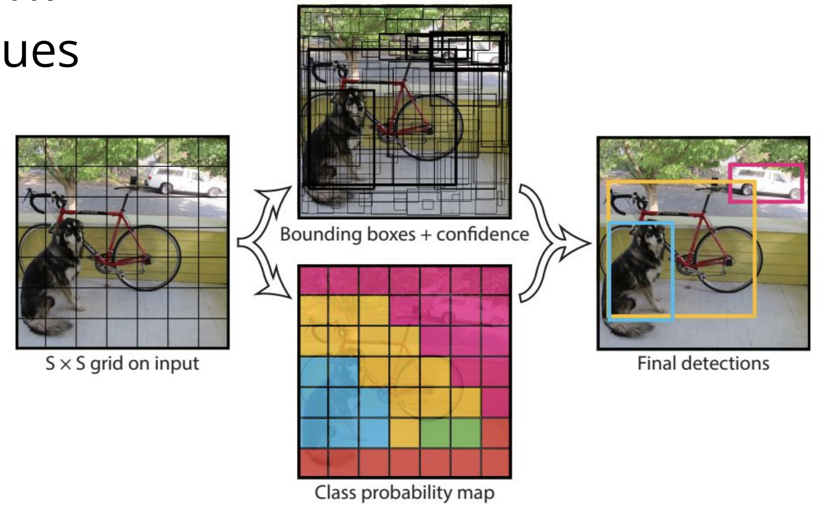
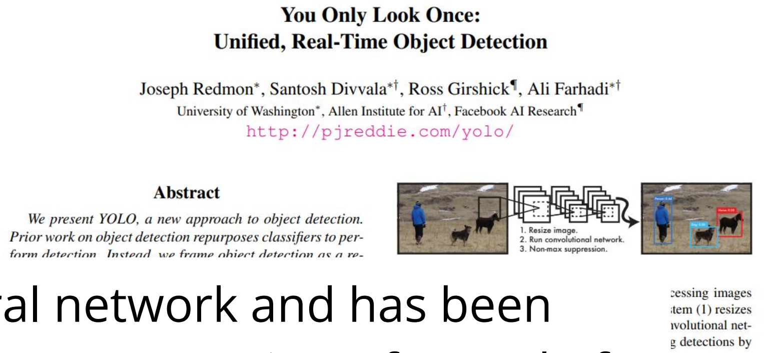


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.



YOLO



**You Only Look Once:
Unified, Real-Time Object Detection**

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a re-

cessing images
tem (1) resizes
volutional net-
g detections by

- YOLO is implemented as a convolution neural network and has been evaluated on the PASCAL VOC detection dataset. It consists of a total of 24 convolutional layers followed by 2 fully connected layers.

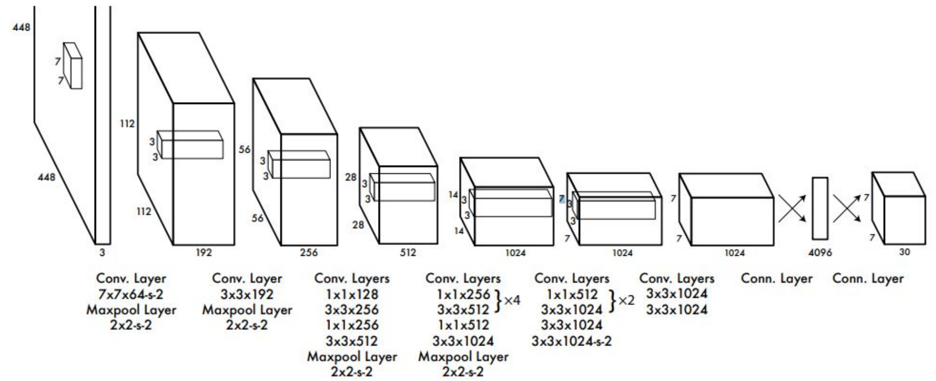
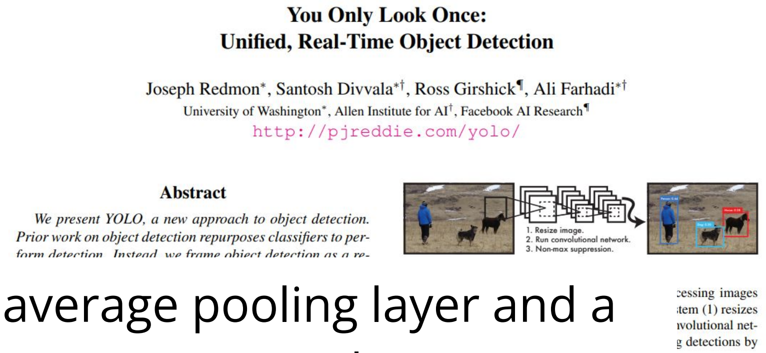


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

METU

YOLO (v1)


**You Only Look Once:
Unified, Real-Time Object Detection**

Joseph Redmon[*], Santosh Divvala[*†], Ross Girshick[¶], Ali Farhadi[*†]
University of Washington[*], Allen Institute for AI[†], Facebook AI Research[¶]
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a re-

cessing images
stem (1) resizes
nvolutional net-
g detections by

- First 20 convolutional layers followed by an average pooling layer and a fully connected layer is pre-trained on the ImageNet 1000-class classification dataset (dataset with resolution 224 x 224).

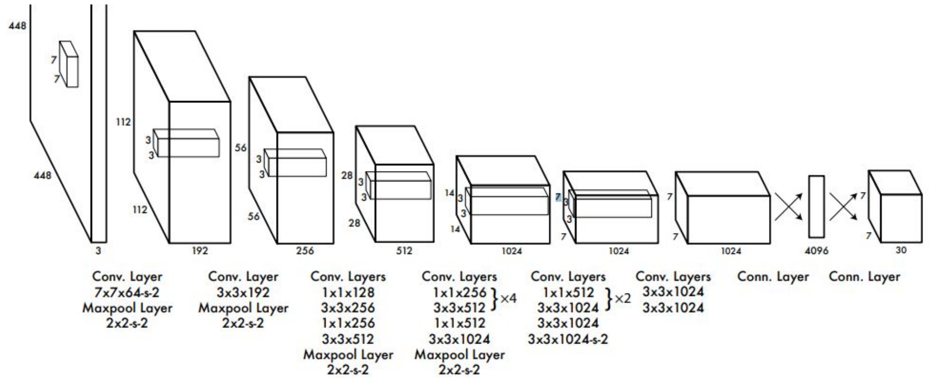
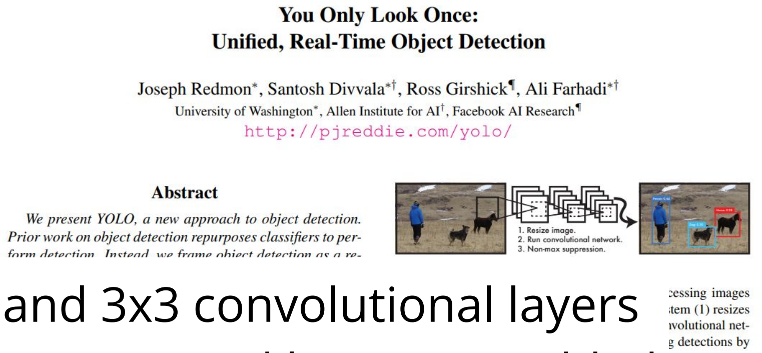


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

METU

YOLO (v1)



**You Only Look Once:
Unified, Real-Time Object Detection**

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a re-

cessing images
tem (1) resizes
volutional net-
g detections by

- The layers comprise of 1x1 reduction layers and 3x3 convolutional layers
- Last 4 convolutional layers followed by 2 fully connected layers are added to train the network for object detection
- The final layer predicts the class probabilities and bounding boxes.

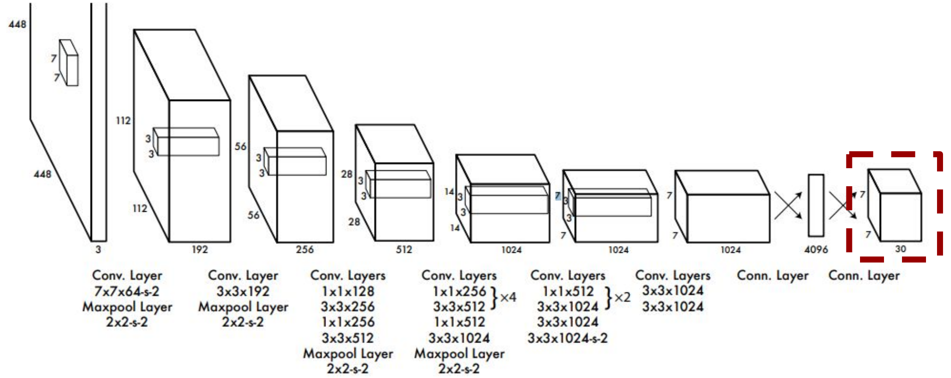
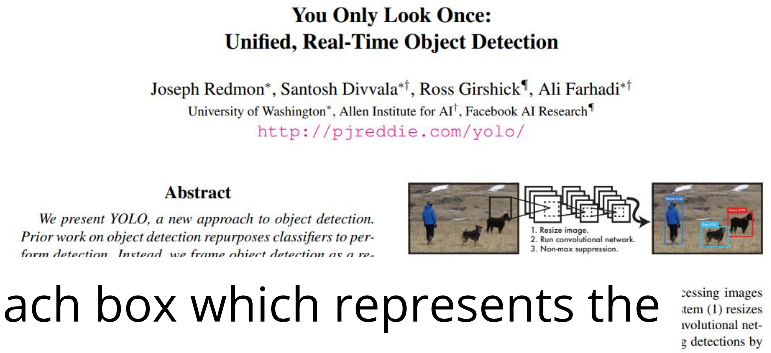


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

METU

YOLO (v1)


You Only Look Once:
Unified, Real-Time Object Detection

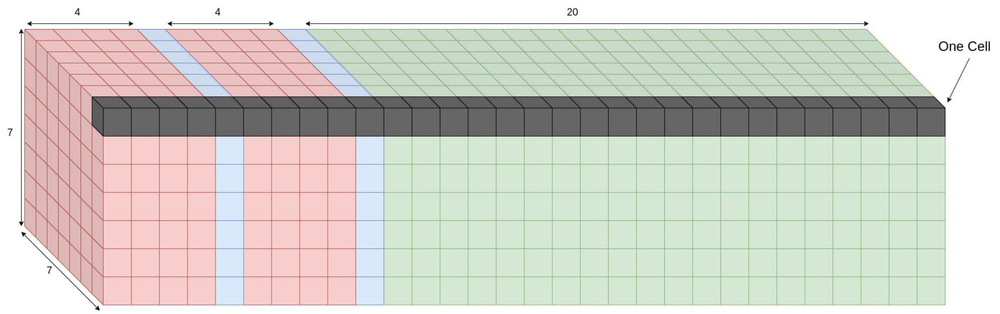
Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

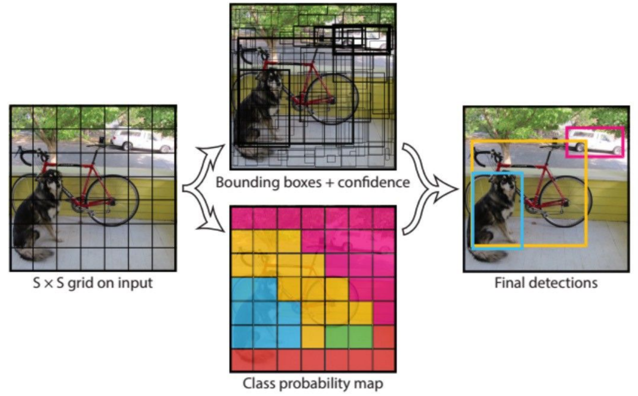
Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a re-

cessing images
item (1) resizes
nvolutional net-
g detections by

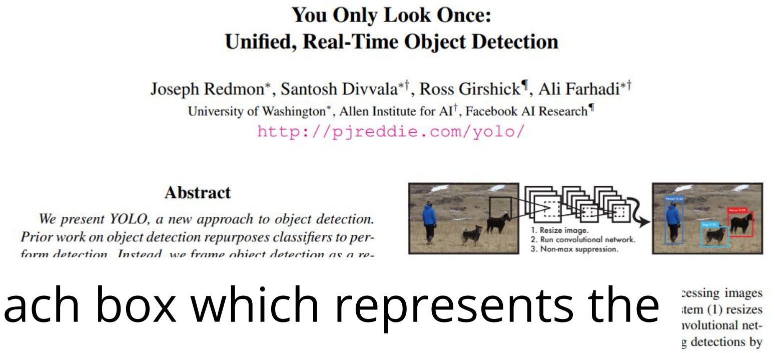
- YOLO also predicts a confidence score for each box which represents the probability that the box contains an object.
- The first five values encode the location and confidence of the first box, the next five encode the location and confidence of the next box, and the final 20 encode the 20 classes (because Pascal VOC has 20 classes).





METU

YOLO (v1)



You Only Look Once:
Unified, Real-Time Object Detection

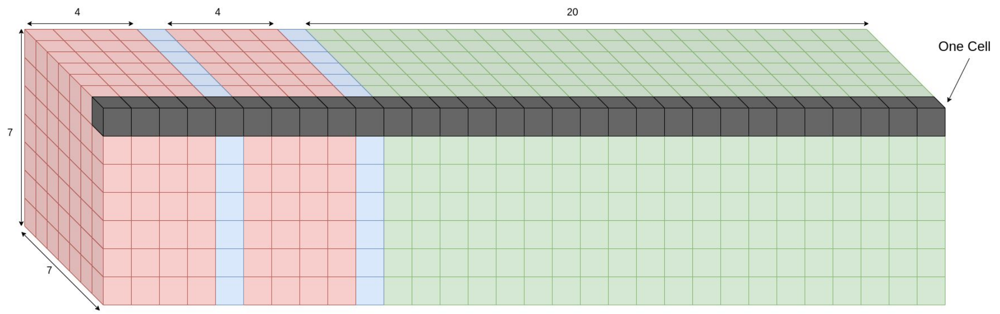
Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

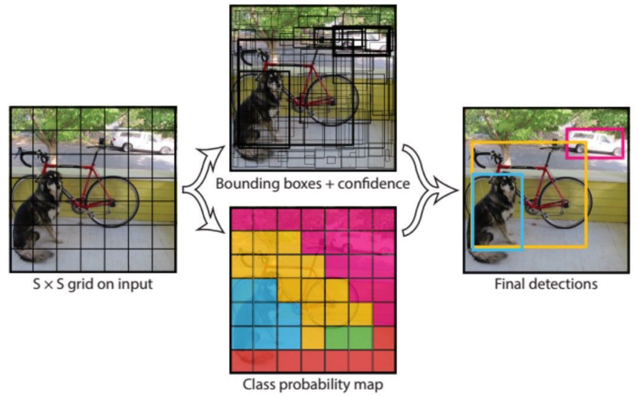
Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a re-

cessing images
stem (1) resizes
nvolutional net-
g detections by

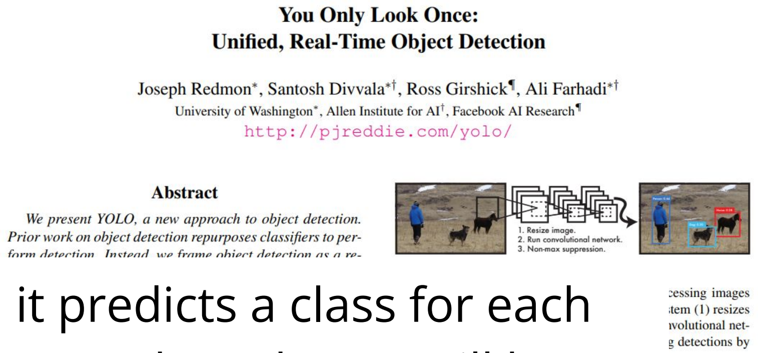
- YOLO also predicts a confidence score for each box which represents the probability that the box contains an object.
- YOLO predicts a class, which is represented by a vector of C values, and the predicted class is the one with the highest value. Now, here's the catch.





METU

YOLO (v1)



You Only Look Once:
Unified, Real-Time Object Detection

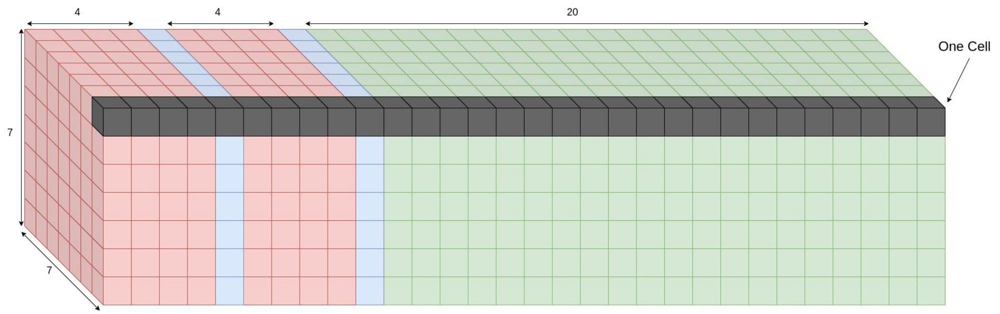
Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

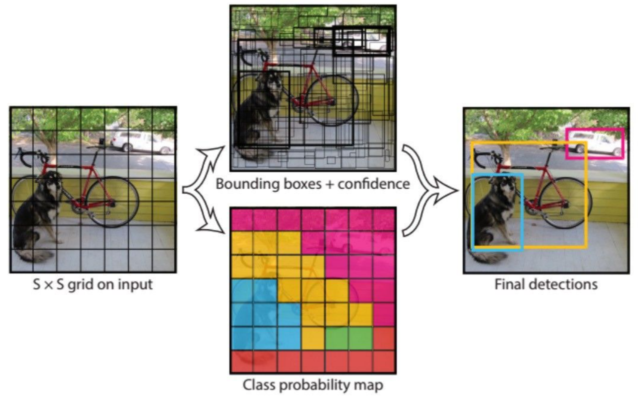
Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a re-

cessing images
tem (1) resizes
nvolutional net-
g detections by

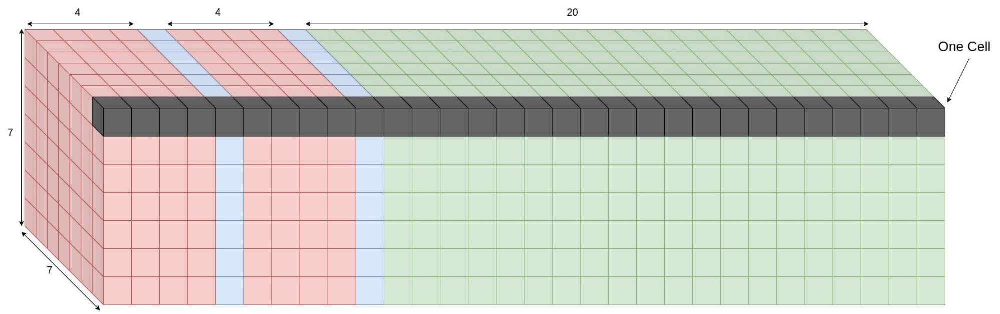
- YOLO does not predict a class for every box, it predicts a class for each cell. But each cell is associated with two boxes, so those boxes will have the same predicted class, even though they may have different shapes and positions.

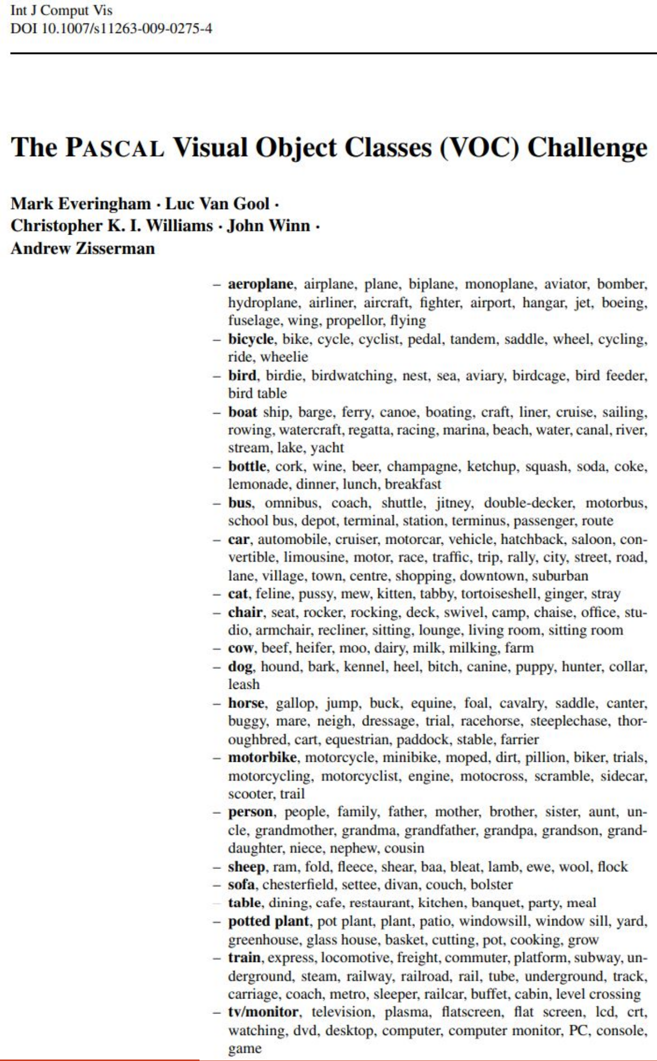




YOLO (v1)

- Pascal VOC has 20 classes:




Int J Comput Vis
DOI 10.1007/s11263-009-0275-4

The PASCAL Visual Object Classes (VOC) Challenge

Mark Everingham · Luc Van Gool · Christopher K. I. Williams · John Winn · Andrew Zisserman

- **aeroplane**, airplane, plane, biplane, monoplane, aviator, bomber, hydroplane, airliner, aircraft, fighter, airport, hangar, jet, boeing, fuselage, wing, propellor, flying
- **bicycle**, bike, cycle, cyclist, pedal, tandem, saddle, wheel, cycling, ride, wheelie
- **bird**, birdie, birdwatching, nest, sea, aviary, birdcage, bird feeder, bird table
- **boat** ship, barge, ferry, canoe, boating, craft, liner, cruise, sailing, rowing, watercraft, regatta, racing, marina, beach, water, canal, river, stream, lake, yacht
- **bottle**, cork, wine, beer, champagne, ketchup, squash, soda, coke, lemonade, dinner, lunch, breakfast
- **bus**, omnibus, coach, shuttle, jitney, double-decker, motorbus, school bus, depot, terminal, station, terminus, passenger, route
- **car**, automobile, cruiser, motorcar, vehicle, hatchback, saloon, convertible, limousine, motor, race, traffic, trip, rally, city, street, road, lane, village, town, centre, shopping, downtown, suburban
- **cat**, feline, pussy, mew, kitten, tabby, tortoiseshell, ginger, stray
- **chair**, seat, rocker, rocking, deck, swivel, camp, chaise, office, studio, armchair, recliner, sitting, lounge, living room, sitting room
- **cow**, beef, heifer, moo, dairy, milk, milking, farm
- **dog**, hound, bark, kennel, heel, bitch, canine, puppy, hunter, collar, leash
- **horse**, gallop, jump, buck, equine, foal, cavalry, saddle, canter, buggy, mare, neigh, dressage, trial, racehorse, steeplechase, thoroughbred, cart, equestrian, paddock, stable, farrier
- **motorbike**, motorcycle, minibike, moped, dirt, pillion, biker, trials, motorcycling, motorcyclist, engine, motocross, scramble, sidecar, scooter, trail
- **person**, people, family, father, mother, brother, sister, aunt, uncle, grandmother, grandma, grandfather, grandpa, grandson, granddaughter, niece, nephew, cousin
- **sheep**, ram, fold, fleece, shear, baa, bleat, lamb, ewe, wool, flock
- **sofa**, chesterfield, settee, divan, couch, bolster
- **table**, dining, cafe, restaurant, kitchen, banquet, party, meal
- **potted plant**, pot plant, plant, patio, windowsill, window sill, yard, greenhouse, glass house, basket, cutting, pot, cooking, grow
- **train**, express, locomotive, freight, commuter, platform, subway, underground, steam, railway, railroad, rail, tube, underground, track, carriage, coach, metro, sleeper, railcar, buffet, cabin, level crossing
- **tv/monitor**, television, plasma, flatscreen, flat screen, lcd, crt, watching, dvd, desktop, computer, computer monitor, PC, console, game

METU

YOLO (v1)

You Only Look Once: Unified, Real-Time Object Detection

Joseph Redmon[*], Santosh Divvala[*†], Ross Girshick[¶], Ali Farhadi[*†]
University of Washington[*], Allen Institute for AI[†], Facebook AI Research[¶]
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a re-

1. Resize image.
2. Run convolutional network.
3. Non-max suppression.

- The loss function is made up of five parts.

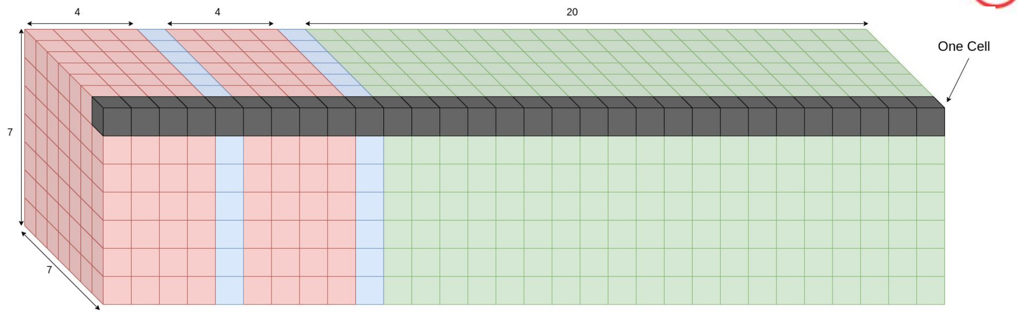
1 $$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

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

3 $$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

4 $$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

5 $$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (3)$$



YOLO versions

- The original YOLO (v1) model was the first object detection network to combine the problem of drawing bounding boxes and identifying class labels in one end-to-end differentiable network.
- YOLO v1 was written by Joseph Redmon in a custom framework called Darknet. Darknet is a very flexible research framework written in low level languages and has produced a series of the best real-time object detectors in computer vision: YOLO, YOLOv2, YOLOv3, and now, YOLOv4.
- The YOLO family of models has continued to evolve since the initial release in 2016.

METU

Additional Reading & References

- https://bdtechtalks.com/2021/06/21/object-detection-deep-learning/
- http://andrew.carterlunn.co.uk/programming/2018/01/24/annotating-large-datasets-with-tensorflow-object-detection-api.html
- https://arxiv.org/abs/1311.2524
- https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
- https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013/UijlingsIJCV2013.pdf
- https://towardsdatascience.com/evolution-of-yolo-yolo-version-1-afb8af302bd2
- https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109
- https://www.harrysprojects.com/articles/yolov1.html