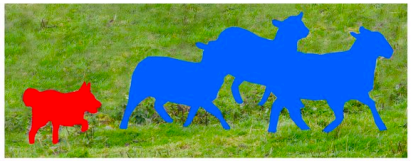
LOGIROAD
AI proof of concept

Detection/Tracking and Segmentation networks

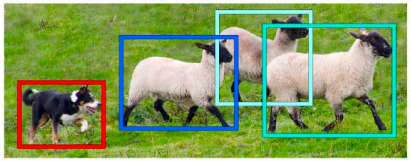
T. Chateau

Several correlated task

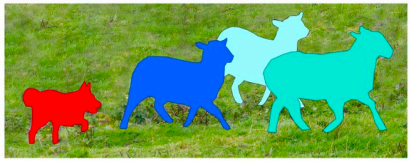
Image Recognition / Classification



Semantic Segmentation



Object Detection

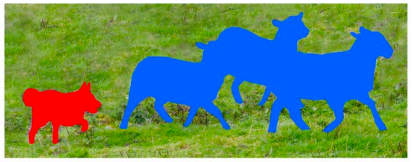


Instance Segmentation

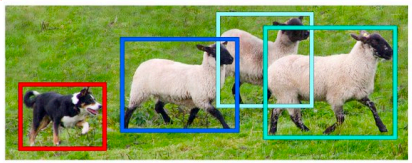
LOGIROAD
AI proof of concept

Several correlated task

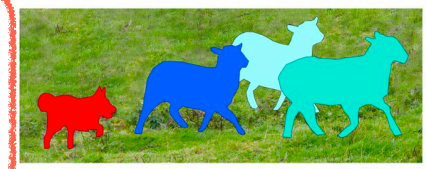
Image Recognition



Semantic Segmentation



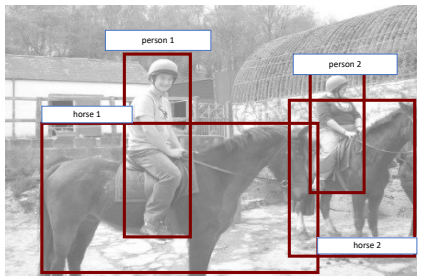
Object Detection



Instance Segmentation

LOGIROAD
AI proof of concept

The Task



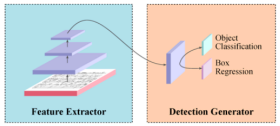
https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD
AI proof of concept

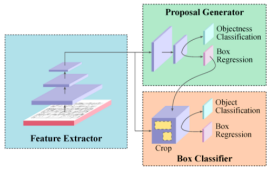
Object detection networks

Two main object detector structures exist:

- One-Stage Detectors
- Two-Stage Detectors



(a) Basic architecture of a one-stage detector.



(b) Basic architecture of a two-stage detector.

LOGIROAD AI proof of concept

History of object detection Datasets

- Face detection
- One category: face
- Frontal faces
- Fairly rigid, unoccluded

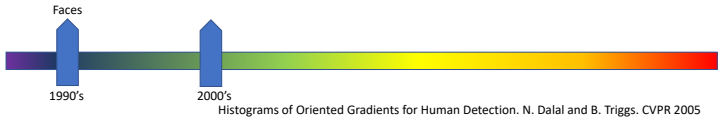
Human Face Detection in Visual Scenes. H. Rowley, S. Baluja, T. Kanade. 1995.

https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

Pedestrians

- One category: pedestrians
- Slight pose variations and small distortions
- Partial occlusions



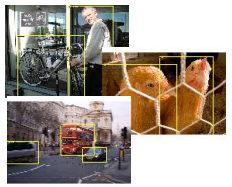
Histograms of Oriented Gradients for Human Detection. N. Dalal and B. Triggs. CVPR 2005

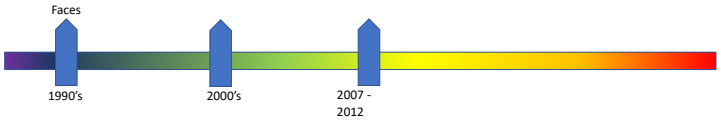
https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

PASCAL VOC

- 20 categories
- 10K images
- Large pose variations, heavy occlusions
- Generic scenes
- Cleaned up performance metric



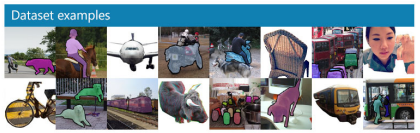


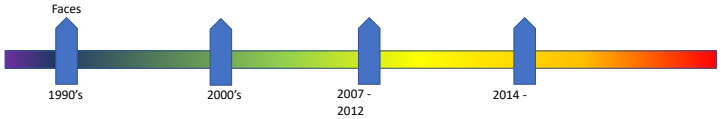
https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

Coco

- 80 diverse categories
- 200+K images
- Heavy occlusions, many objects per image, large scale variations





https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

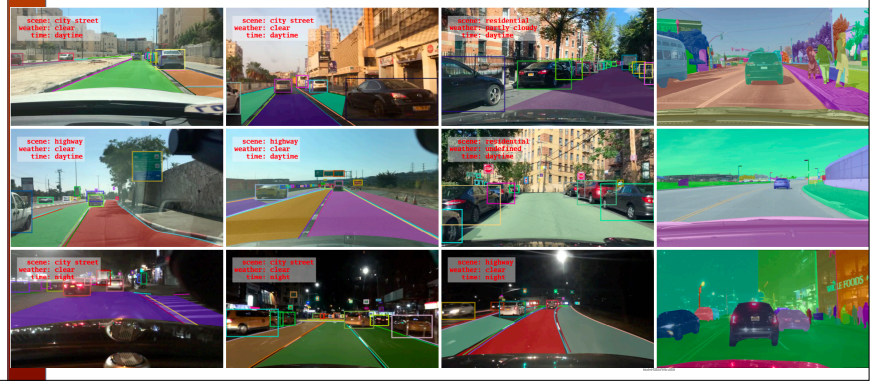
LOGIROAD AI proof of concept

Public dataset

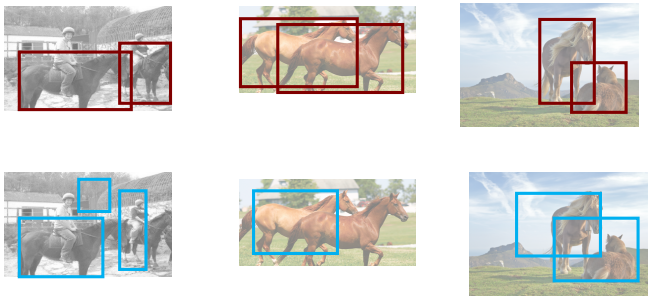
BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Many datasets exist

BDD100K



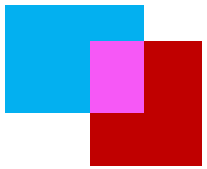
Evaluation metric



https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

Matching detections to ground truth



$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

Matching detections to ground truth

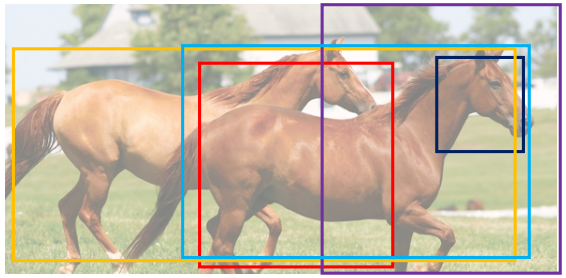
- Match detection to most similar ground truth
  - highest IoU
- If IoU > 50%, mark as correct
- If multiple detections map to same ground truth, mark only one as correct
- **Precision** = #correct detections / total detections
- **Recall** = #ground truth with matched detections / total ground truth

https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

Why is detection hard(er)?

- Precise localization



https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

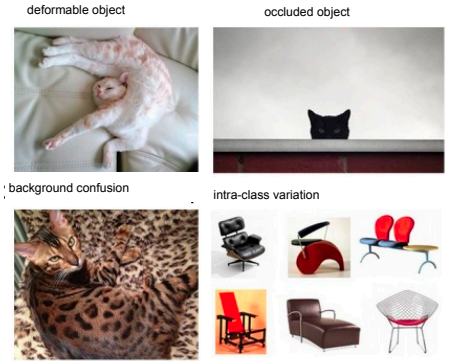
Why is detection hard(er)?

- Much larger impact of pose

https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

Why is detection hard(er)?



https://www.cs.cornell.edu/courses/cs4670/2018sp/lec36-obj-detn.pptx

LOGIROAD AI proof of concept

Why is detection hard(er)?

- light conditions make detection difficult

Credit: A Zisserman VGG Oxford

Why is detection hard(er)?

- Counting

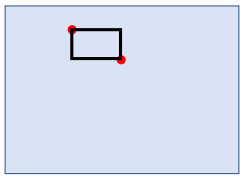
Why is detection hard(er)?

- Small objects / scale

Andrew Zisserman, VGG, Oxford

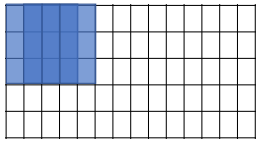
Detection as classification

- Run through every possible box and classify
- How many boxes?
  - Every pair of pixels = 1 box
  - $\binom{N}{2}$ = $O(N^2)$
  - For 300 x 500 image, N = 150K
  - 2.25 x 10^{10} boxes!



Idea 1: scanning window

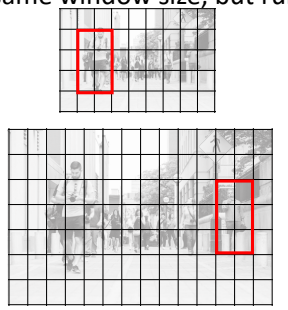
- Fix size
  - Can take a few different sizes
- Fixed stride
- Convolution with a filter
  - Classic: compute HOG features over entire image



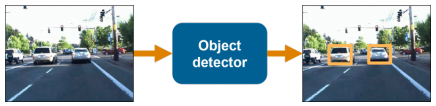
Dealing with scale

Dealing with scale

- Use same window size, but run on *image pyramid*



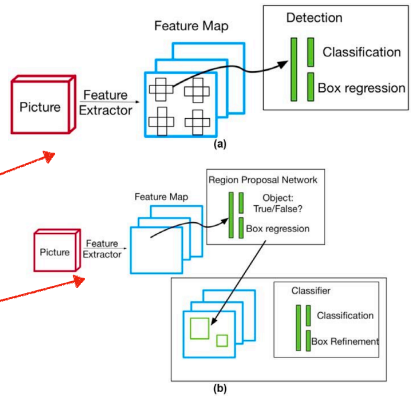
Object detection networks



Two main object detector structures exist:

One-Step detector

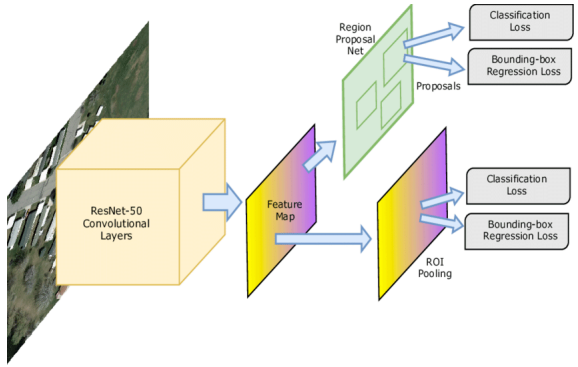
Two-Steps detector



Object detection networks

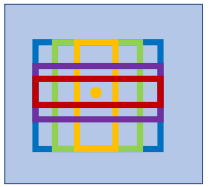
Example of two-stages-detector: Faster-Rcnn

Ren, Shaoqing, Kaiming He, Ross Girshick, et Jian Sun. « Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks ». In *Advances in Neural Information Processing Systems 28*, édité par C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, et R. Garnett, 91-99. Curran Associates, Inc., 2015. http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.



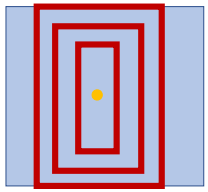
Faster R-CNN

- At each location, consider boxes of many different sizes and aspect ratios



Faster R-CNN

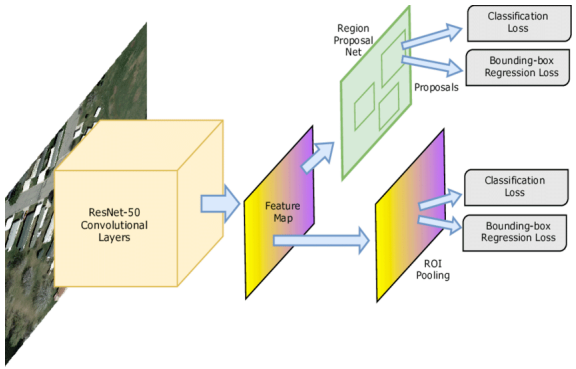
- At each location, consider boxes of many different sizes and aspect ratios



Object detection networks

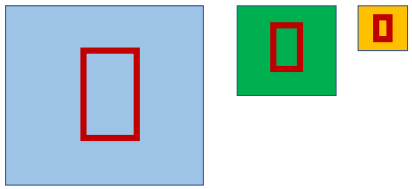
Example of two-stages-detector: Faster-Rcnn

Ren, Shaoqing, Kaiming He, Ross Girshick, et Jian Sun. « Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks ». In *Advances in Neural Information Processing Systems 28*, édité par C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, et R. Garnett, 91-99. Curran Associates, Inc., 2015. http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.



ROI Pooling

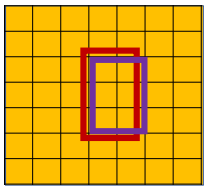
- How do we crop from a feature map?
- Step 1: Resize boxes to account for subsampling



Fast R-CNN. Ross Girshick. In ICCV 2015

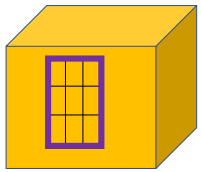
ROI Pooling

- How do we crop from a feature map?
- Step 2: Snap to feature map grid



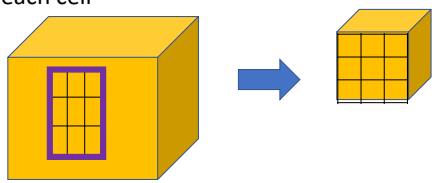
ROI Pooling

- How do we crop from a feature map?
- Step 3: Place a grid of fixed size



ROI Pooling

- How do we crop from a feature map?
- Step 4: Take max in each cell



Faster-RCNN Loss

Fast R-CNN

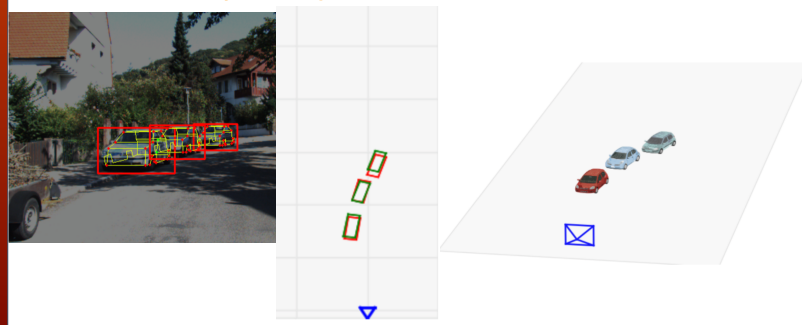
- Multi-task loss

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

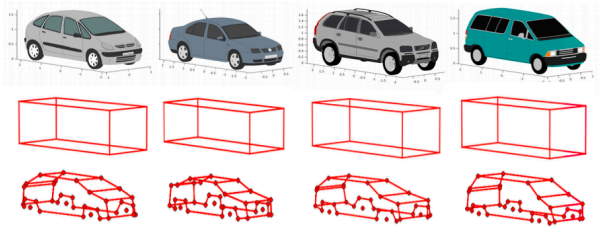
Where $$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

System Outputs

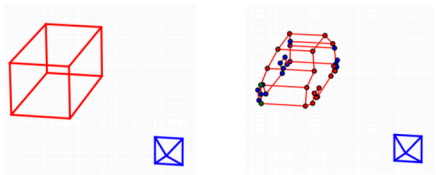


3D samples of shape and template dataset

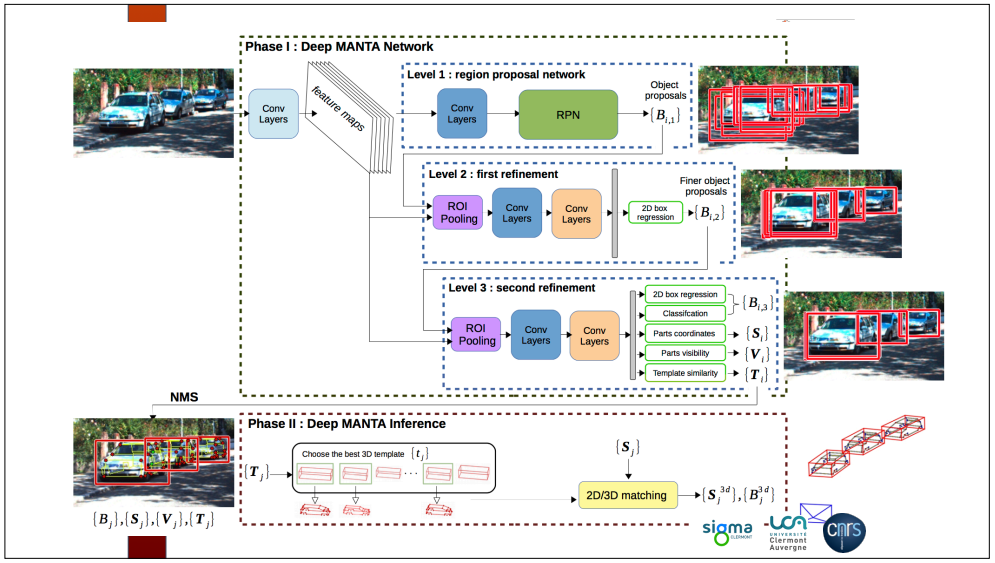


Bounding box and part detection
(with visibility estimation, green and blue)

(a) (b)



(c) (d)



Deep Learning for 3D vehicle understanding from monocular images

Loss functions

RPN Loss

Detection loss

Parts Loss

Visibility Loss

Template similarity loss

$$\mathcal{L} = \mathcal{L}^1 + \mathcal{L}^2 + \mathcal{L}^3$$

with

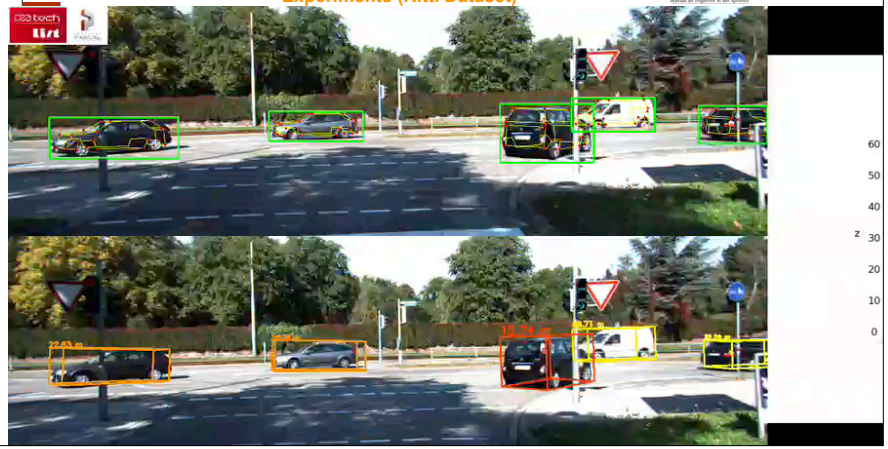
$$\mathcal{L}^1 = \mathcal{L}_{rpn},$$

$$\mathcal{L}^2 = \sum_i \mathcal{L}^2_{det}(i) + \mathcal{L}^2_{parts}(i),$$

$$\mathcal{L}^3 = \sum_i \mathcal{L}^3_{det}(i) + \mathcal{L}^3_{parts}(i) + \mathcal{L}_{vis}(i) + \mathcal{L}_{temp}(i),$$

Deep Learning for 3D vehicle understanding from monocular images

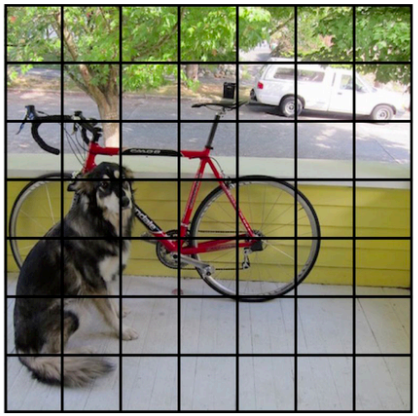
Experiments (Kitti Dataset)



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

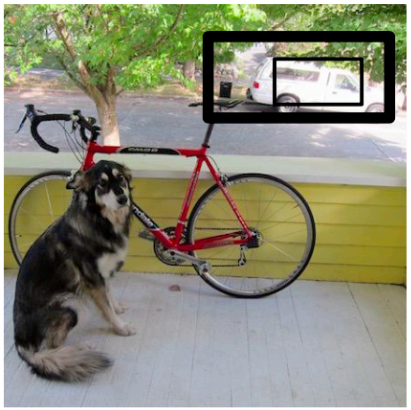
Presenter: Liyang Zhong Quan Zou

We split the image into an S*S grid

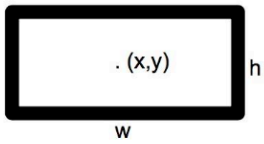


Each cell predicts B boxes(x,y,w,h) and confidences of each box: P(Object)

B = 2

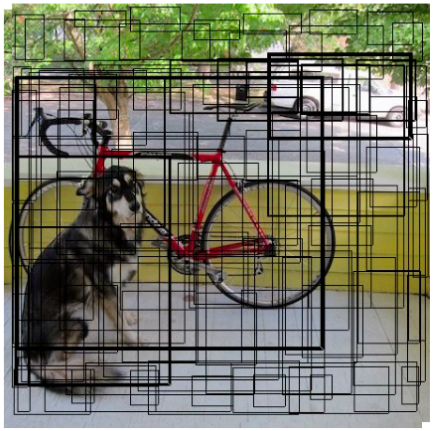


each box predict:

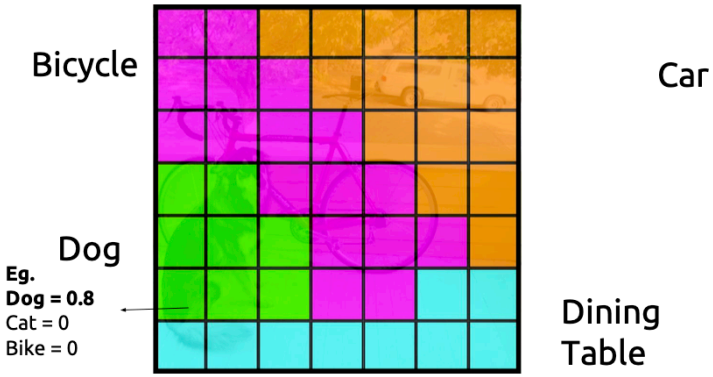


P(Object): probability that the box contains an object

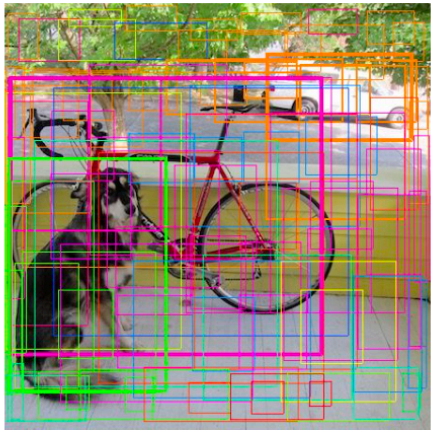
Each cell predicts boxes and confidences: P(Object)



Class Probability Conditioned on object: P(Car | Object)



Then we combine the box and class predictions.

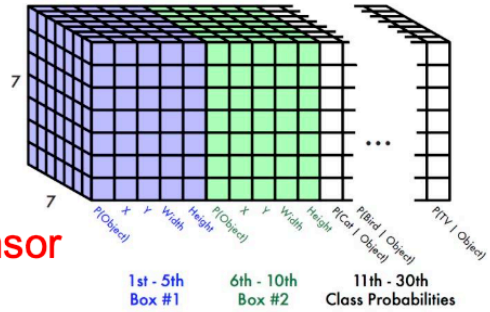


P(class|Object) * P(Object)
=P(class)

OUTPUT

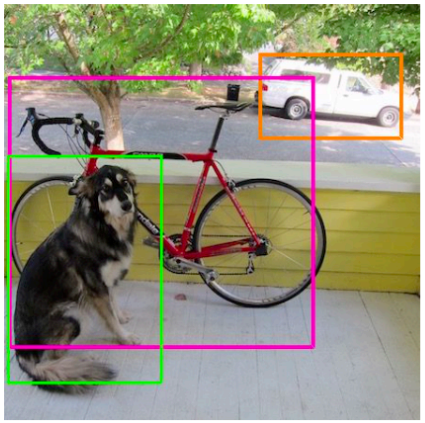
Each cell predicts:

- For each bounding box:
  - 4 coordinates (x, y, w, h)
  - 1 confidence value
- Some number of class probabilities

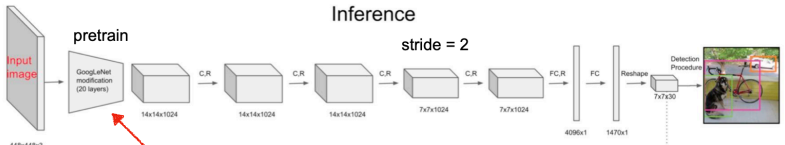


S * S * (B * 5 + C) tensor

Finally we do threshold detections and NMS



STRUCTURE



C: Convolutional
R: Relu

Backbone

Other details - Non-max suppression

How do we deal with multiple detections on the same object?

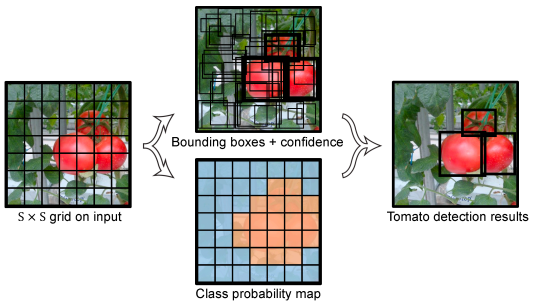
Other details - Non-max suppression

- Go down the list of detections starting from highest scoring
- Eliminate any detection that overlaps highly with a higher scoring detection
- Separate, heuristic step

Deep Learning for Visual Tracking

Object detection networks

Example of one-stage-detector: YOLO



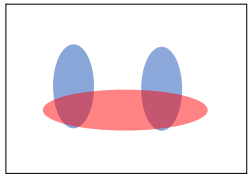
Redmon, Joseph, Santosh Divvala, Ross Girshick, et Ali Farhadi. « You Only Look Once: Unified, Real-Time Object Detection ». In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Semantic Segmentation

The Task

Evaluation metric

- Pixel classification!
- Accuracy?
  - Heavily unbalanced
  - Common classes are over-emphasized
- *Intersection over Union*
  - Average across classes and images
- Per-class accuracy
  - Compute accuracy for every class and then average



Things vs Stuff

THINGS

- Person, cat, horse, etc
- Constrained shape
- Individual instances with separate identity
-

STUFF

- Road, grass, sky etc
- Amorphous, no shape
- No notion of instances
- Can be done at pixel level
-

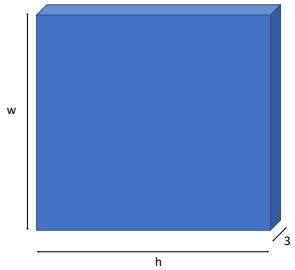
Challenges in data collection

- Precise localization is hard to annotate
- Annotating every pixel leads to heavy tails
- Common solution: annotate few classes (often things), mark rest as “Other”
- Common datasets: PASCAL VOC 2012 (~1500 images, 20 categories), COCO (~100k images, 20 categories)

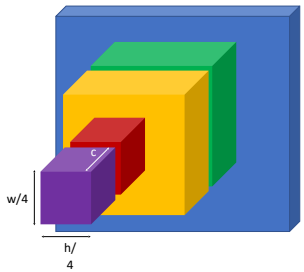
Pre-convnet semantic segmentation

- Things
  - Do object detection, then segment out detected objects
- Stuff
  - ”Texture classification”
  - Compute histograms of filter responses
  - Classify local image patches

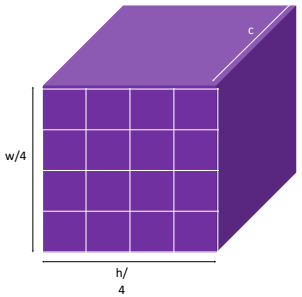
Semantic segmentation using convolutional networks



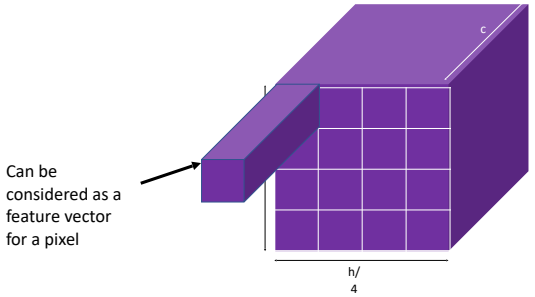
Semantic segmentation using convolutional networks



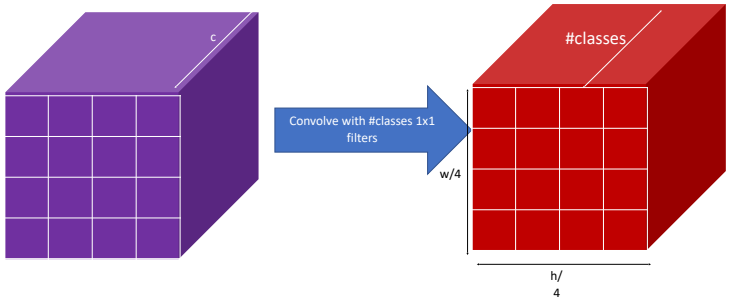
Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks

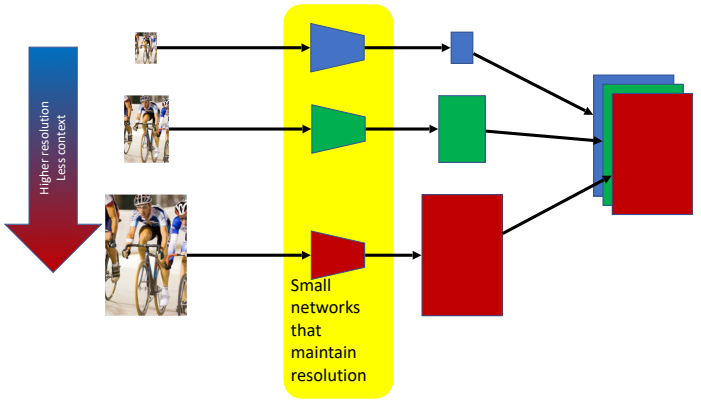
- Pass image through convolution and subsampling layers
- Final convolution with #classes outputs
- Get scores for *subsampled* image
- Upsample back to original size

Semantic segmentation using convolutional networks

The resolution issue

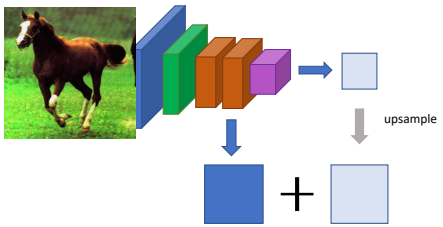
- Problem: Need fine details!
- Shallower network / earlier layers?
  - Deeper networks work better: more abstract concepts
  - Shallower network => Not very semantic!
- Remove subsampling?
  - Subsampling allows later layers to capture larger and larger patterns
  - Without subsampling => Looks at only a small window!

Solution 1: Image pyramids



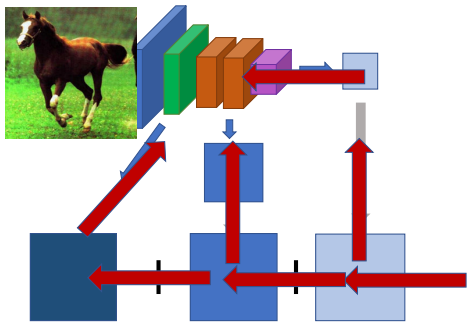
Learning Hierarchical Features for Scene Labeling. Clement Farabet, Camille Couprie, Laurent Najman, Yann LeCun. In *TPAMI*, 2013.

Solution 2: Skip connections



Compute class scores at multiple layers, then upsample and add

Solution 2: Skip connections



Red arrows indicate backpropagation

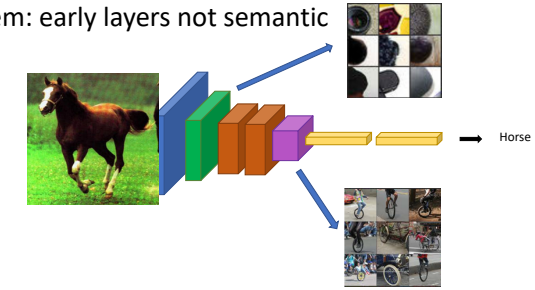
Skip connections

without skip with skip

Fully convolutional networks for semantic segmentation. Evan Shelhamer, Jon Long, Trevor Darrell. In *CVPR* 2015

Skip connections

- Problem: early layers not semantic



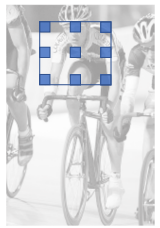
Visualizations from : M. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV* 2014.

Solution 3: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?

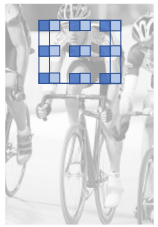
Solution 3: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?



Solution 3: Dilation

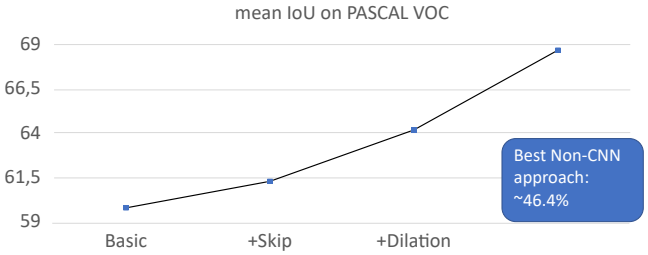
- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?



Solution 3: Dilation

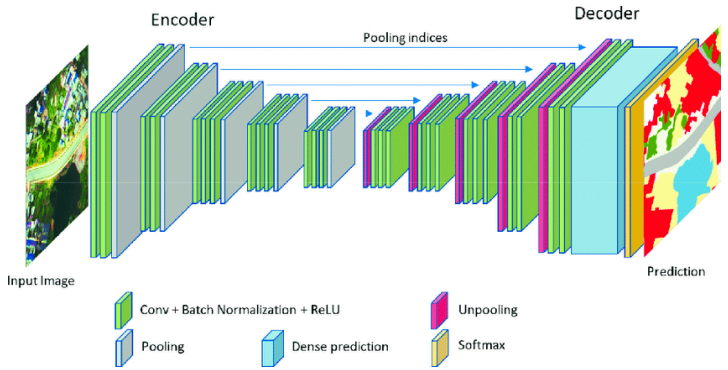
- Instead of subsampling by factor of 2: dilate by factor of 2
- Dilation can be seen as:
  - Using a much larger filter, but with most entries set to 0
  - Taking a small filter and “exploding”/ “dilating” it

Putting it all together

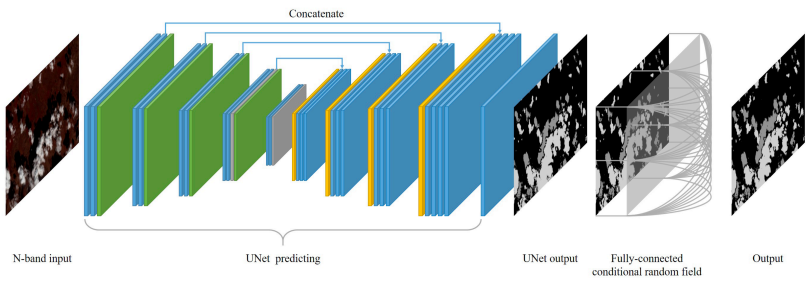


Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. In *ICLR*, 2015.

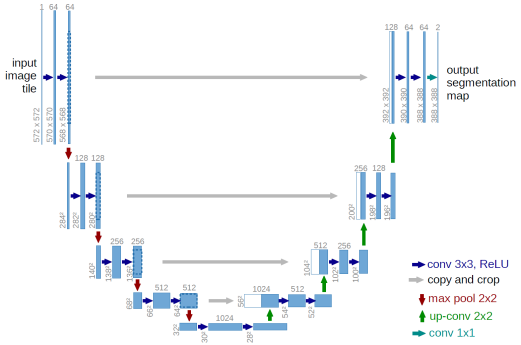
Popular Segmentation networks (segnet)



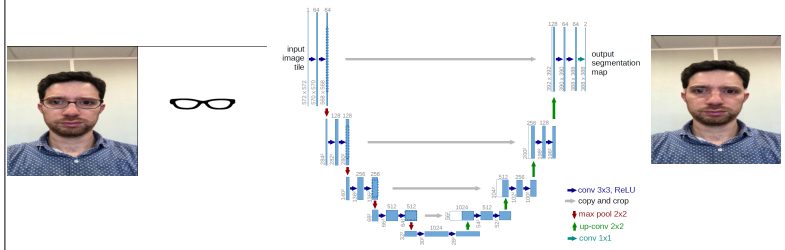
Popular Segmentation networks: UNET

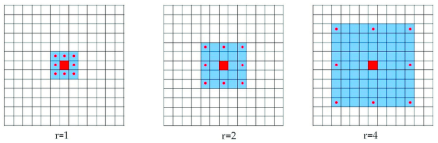


Popular Segmentation networks: UNET



Using UNET for Inpainting



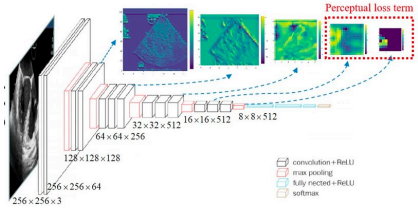


Using partial convolution

Several losses

$$\mathcal{L}_{perceptual} = \sum_{p=0}^{P-1} \frac{\|\Psi_p^{\mathbf{I}_{out}} - \Psi_p^{\mathbf{I}_{gt}}\|_1}{N_{\Psi_p^{\mathbf{I}_{gt}}}} + \sum_{p=0}^{P-1} \frac{\|\Psi_p^{\mathbf{I}_{comp}} - \Psi_p^{\mathbf{I}_{gt}}\|_1}{N_{\Psi_p^{\mathbf{I}_{gt}}}}$$

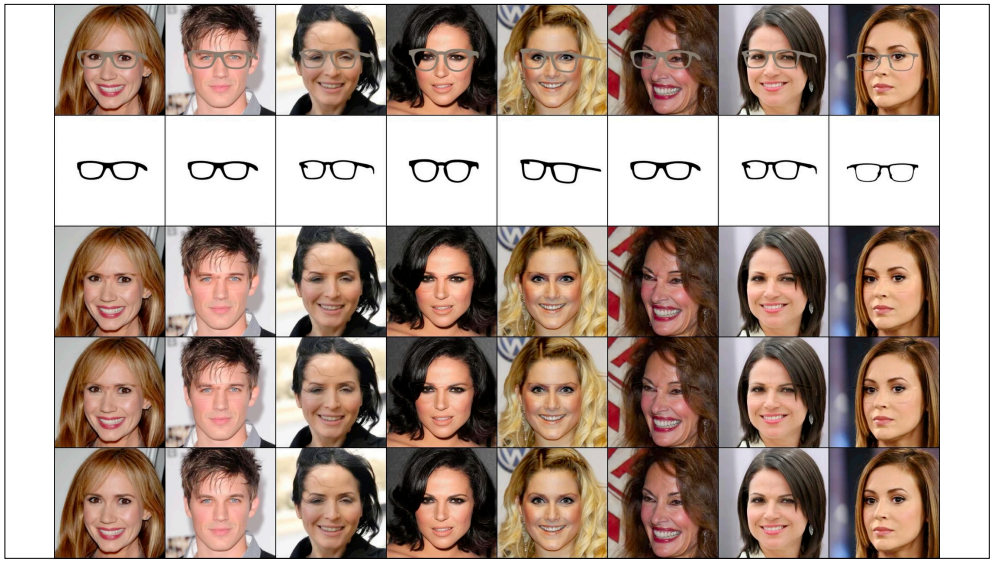
Perceptual loss: L1 norm from feature maps of a pertained backbone (VGG16)



$$\mathcal{L}_{style_{out}} = \sum_{p=0}^{P-1} \frac{1}{C_p C_p} \left\| K_p\left(\left(\Psi_p^{\mathbf{I}_{out}}\right)^{\intercal}\left(\Psi_p^{\mathbf{I}_{out}}\right) - \left(\Psi_p^{\mathbf{I}_{gt}}\right)^{\intercal}\left(\Psi_p^{\mathbf{I}_{gt}}\right)\right)\right\|_1$$

$$\mathcal{L}_{style_{comp}} = \sum_{p=0}^{P-1} \frac{1}{C_p C_p} \left\| K_p\left(\left(\Psi_p^{\mathbf{I}_{comp}}\right)^{\intercal}\left(\Psi_p^{\mathbf{I}_{comp}}\right) - \left(\Psi_p^{\mathbf{I}_{gt}}\right)^{\intercal}\left(\Psi_p^{\mathbf{I}_{gt}}\right)\right)\right\|_1$$

Style loss: Kernel on dot products on feature maps



Deep Learning for Visual Tracking

What is Visual Tracking? From single view single object

Deep Learning for Visual Tracking

What is Visual Tracking? To multi view Muti-non rigid-objects

Deep Learning for Visual Tracking

What is Visual Tracking?

State Vector

The dynamic configuration of the the tracked object at time k is modelled by a State vector denoted:

$$\mathbf{x}_k$$

State Sequence

The state sequence is given by the set (sequence) of State vectors, denoted:

$$\mathbf{X} \doteq \{\mathbf{x}_k\}_{k=1,\ldots,K}$$

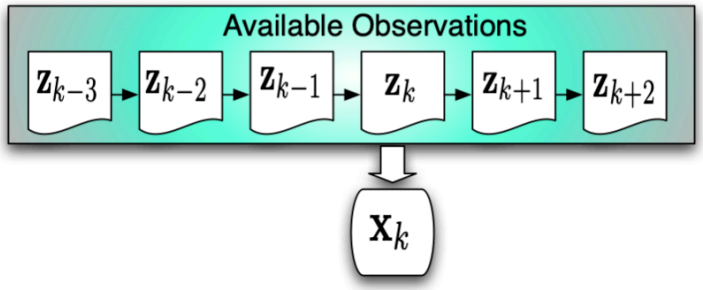
Observation

Observation: $\mathbf{Z} \doteq \{\mathbf{z}_k\}_{k=1,\ldots,K}$

Deep Learning for Visual Tracking

Off-line Tracking (Deferred Tracking)

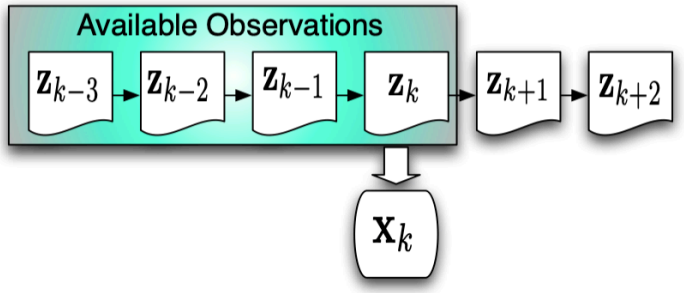
Estimation of the state $\mathbf{x}_k$ uses the entire observation sequence $\mathbf{Z} \doteq \{\mathbf{z}_k\}_{k=1,\ldots,K}$



Deep Learning for Visual Tracking

On-line Tracking

Estimation of the state $\mathbf{x}_k$ uses the current and past observation: $\mathbf{z}_{0:k}$



Deep Learning for Visual Tracking

Why is Visual Tracking difficult?

Hidden State

The state $\mathbf{X}$ is a hidden state and must be deduced from observation

Tracking Challenges

- Object Modeling: how to define what an object is in terms that can be interpreted by a computer ?
- Appearance Change: The observation of an object changes according to many parameters (illumination conditions, occlusions, shape variation...)
- Kinematic Modelling: How to inject priors on object kinematic and interactions between objects.

Deep Learning for Visual Tracking

Why is Visual Tracking difficult? (Object representation)

- Object approximation:
  - Segmentation / Polygonal approximation
  - Bounding ellipse/box
  - Position only
- Goal: Measure affinity

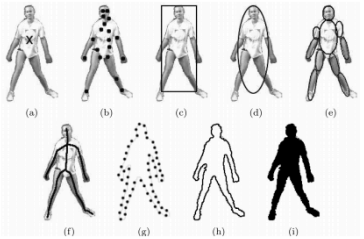
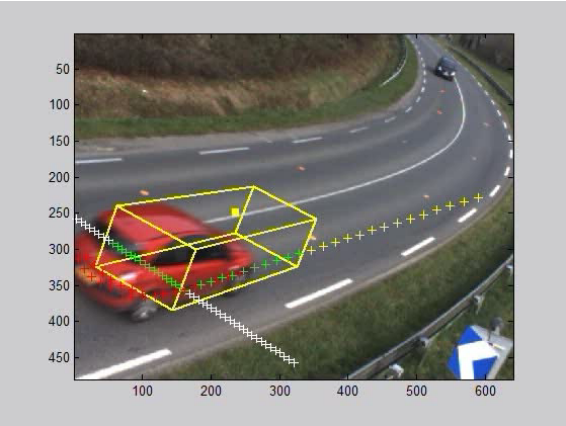


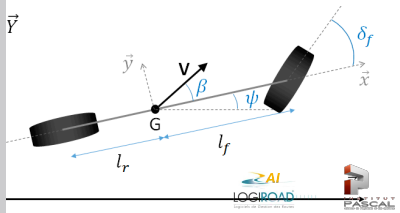
Image from A. Yilmaz et. Al : Object tracking: A survey. ACM Computing Surveys, 2006

Deep Learning for Visual Tracking

Why is Visual Tracking difficult? (Kinematic modelling)

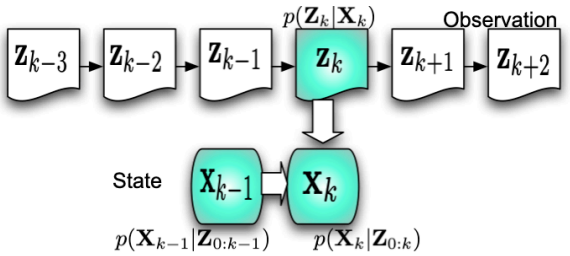


State Vector:
- 2D position on the ground plane
- 2D motion vector on the ground plane
- Steering angle
- Acceleration



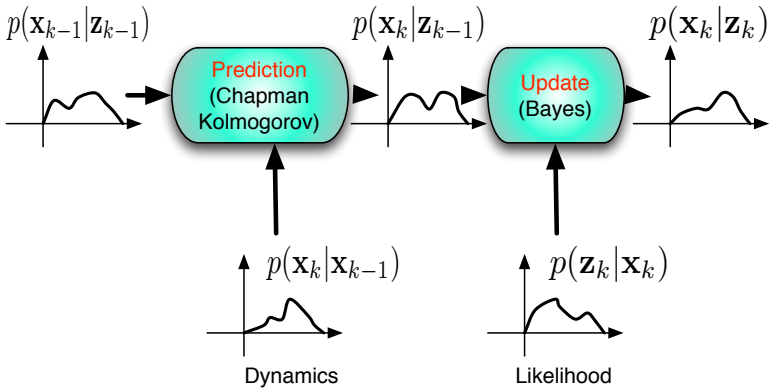
Deep Learning for Visual Tracking

The classical (probabilistic) view of tracking



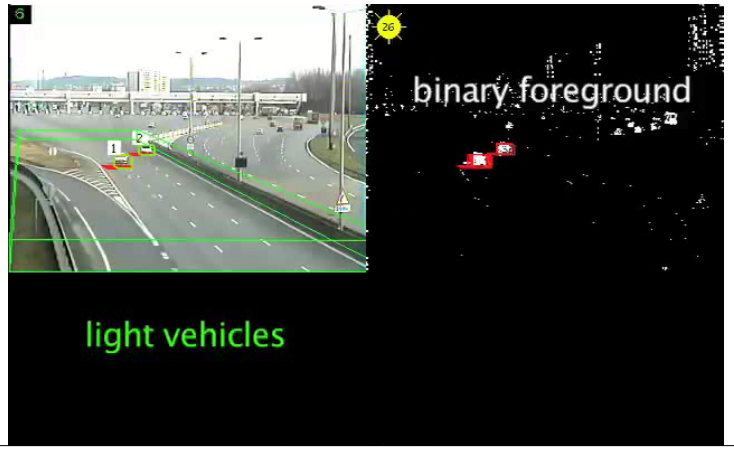
Deep Learning for Visual Tracking

The classical (probabilistic) view of tracking



Deep Learning for Visual Tracking

The classical (probabilistic) view of tracking



Deep Learning for Visual Tracking

The classical (optimisation) view of tracking

State

The State vector is an unknown parameter vector which can be estimated using optimisation techniques :

$$\hat{\mathbf{x}}_k = \arg\min_{\mathbf{x}_k \in \mathcal{X}} \mathcal{E}(\mathbf{x}_k, \mathbf{z}_k)$$

The search space $\mathcal{X}$ is often reduced using priors on motion and previous estimation.

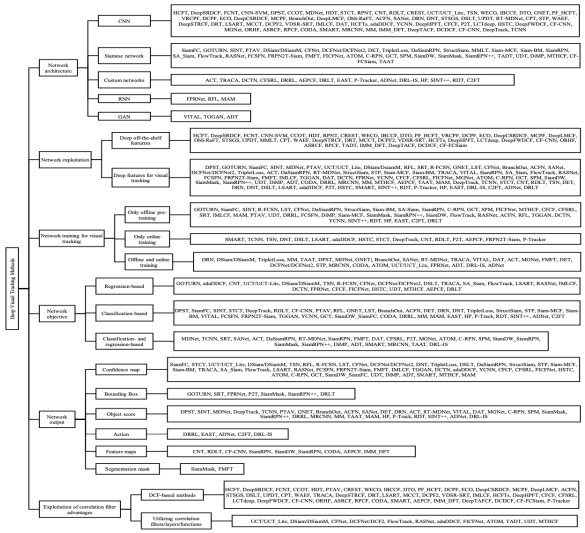
Deep Learning for Visual Tracking

The classical (optimisation) view of tracking (Meanshift)

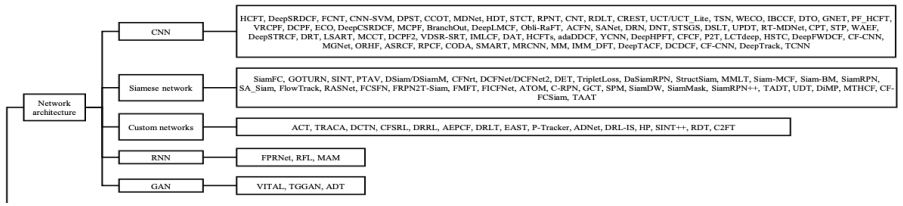
Deep Learning for Visual Tracking

Overview of Visual tracking

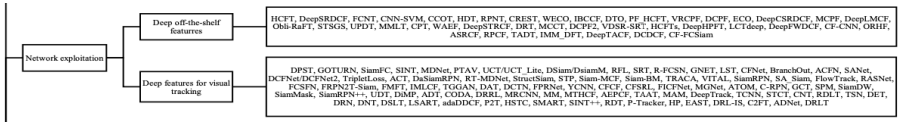
Deep Learning for Visual Tracking



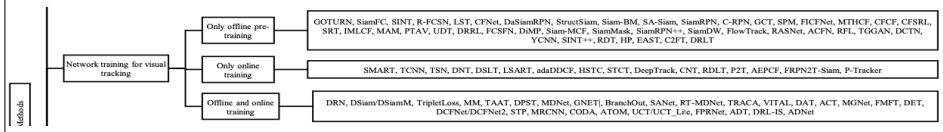
Deep Learning for Visual Tracking



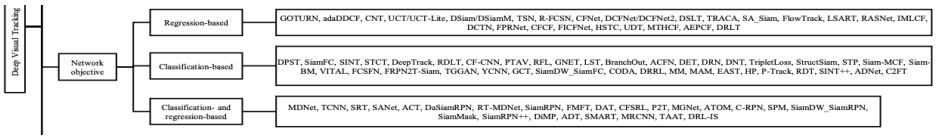
Deep Learning for Visual Tracking



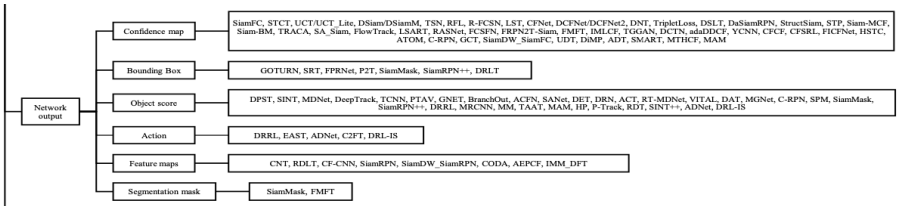
Deep Learning for Visual Tracking



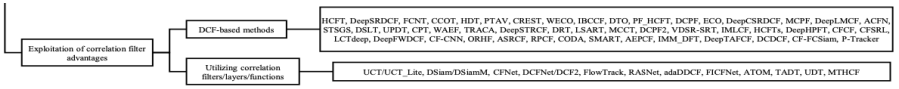
Deep Learning for Visual Tracking



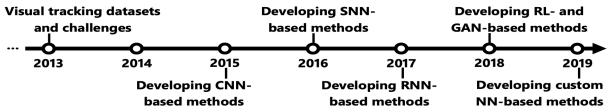
Deep Learning for Visual Tracking



Deep Learning for Visual Tracking



Deep Learning for Visual Tracking

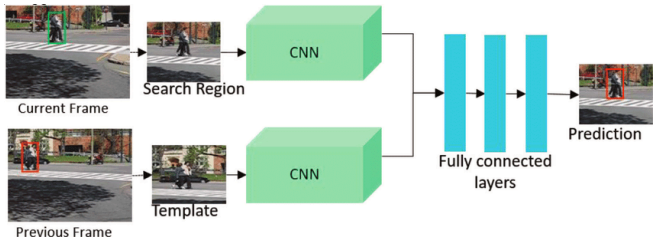


Recent history of Visual tracking

Deep Learning for Visual Tracking

SNN based model: GOTURN

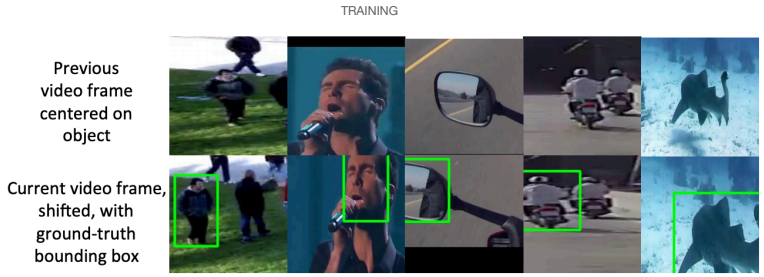
Generic Object Tracking Using Regression Networks



Deep Learning for Visual Tracking

SNN based model: GOTURN

Generic Object Tracking Using Regression Networks



Held, David, Sebastian Thrun, et Silvio Savarese. « Learning to Track at 100 FPS with Deep Regression Networks ». CoRR abs/1604.01802 (2016). http://arxiv.org/abs/1604.01802.

Deep Learning for Visual Tracking

multi-object tracking

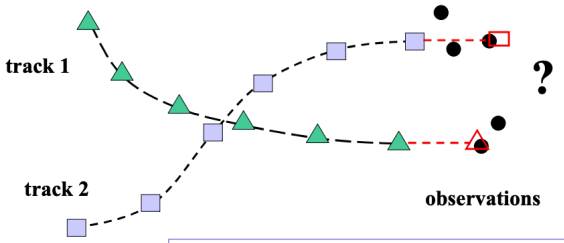
Based on Tracking-by-detection

1) **Object detection**
2) **Metric estimation between detected objects and targets (set of objects with the same identity)**
3) **Association between object and target**
4) **target birth, death and loss.**

Deep Learning for Visual Tracking

multi-object tracking

Intuition: predict next position along each track.

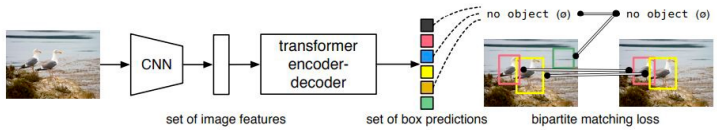


How to determine which observations to add to which track?

Deep Learning for Visual Tracking

Object detection networks

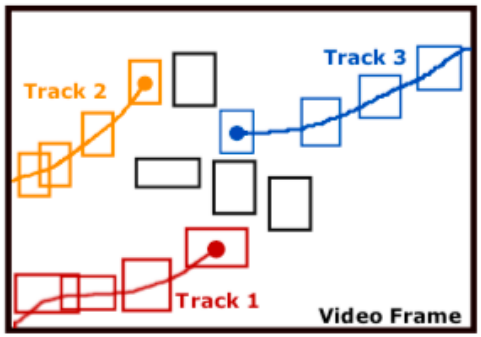
2020: Using Transformers for object detection



Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, et Sergey Zagoruyko. *End-to-End Object Detection with Transformers*, 2020.

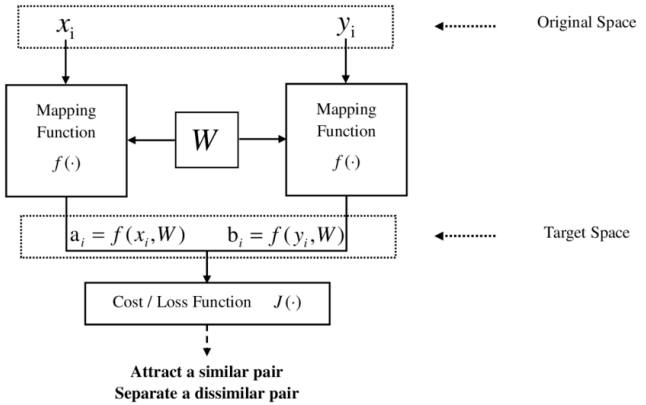
Deep Learning for Visual Tracking

Association: define metric and match objects and targets



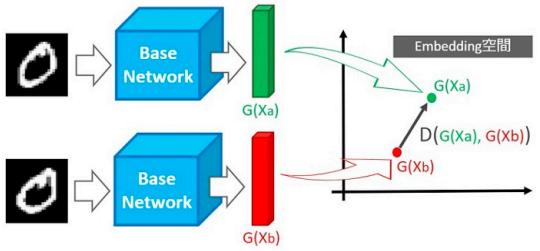
Deep Learning for Visual Tracking

Association: define metric



Deep Learning for Visual Tracking

Association: define metric

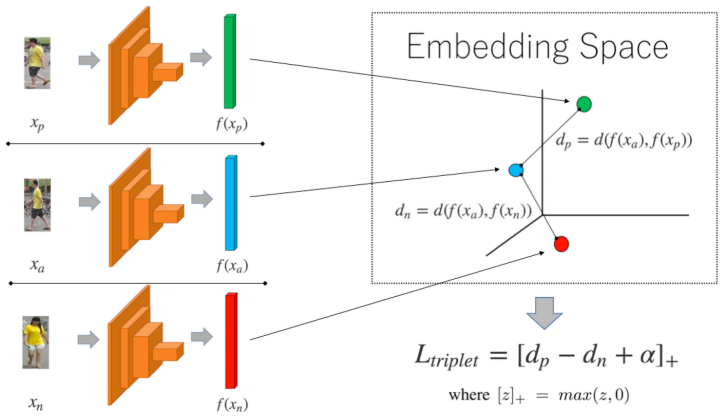


Y=0, same category, minimize Dw

$$L = (1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$

Y=1, different category, maximize Dw to m

Deep Learning for Visual Tracking



Association: define metric

Deep Learning for Visual Tracking

Association: define metric and match objects and targets (association matrix)

We have N objects in previous frame and M objects in current frame. We can build a table of match scores m(i,j) for i=1...N and j=1...M. For now, assume M=N.

	1	2	3	4	5
1	0.95	0.76	0.62	0.41	0.06
2	0.23	0.46	0.79	0.94	0.35
3	0.61	0.02	0.92	0.92	0.81
4	0.49	0.82	0.74	0.41	0.01
5	0.89	0.44	0.18	0.89	0.14

problem: choose a 1-1 correspondence that maximizes sum of match scores.

Deep Learning for Visual Tracking

Association: define metric and match objects and targets (association matrix)

5x5 matrix of match scores

0.95	0.76	0.62	0.41	0.06
0.23	0.46	0.79	0.94	0.35
0.61	0.02	0.92	0.92	0.81
0.49	0.82	0.74	0.41	0.01
0.89	0.44	0.18	0.89	0.14

working from left to right, choose one number from each column, making sure you don't choose a number from a row that already has a number chosen in it.

How many ways can we do this?

5 x 4 x 3 x 2 x 1 = 120 (N factorial)

Deep Learning for Visual Tracking

Association: define metric and match objects and targets (association matrix)

0.95	0.76	0.62	0.41	0.06
0.23	**0.46**	0.79	0.94	0.35
0.61	0.02	**0.92**	0.92	0.81
0.49	0.82	0.74	**0.41**	0.01
0.89	0.44	0.18	0.89	**0.14**

score: 2.88

0.95	0.76	0.62	**0.41**	0.06
0.23	0.46	0.79	0.94	0.35
0.61	0.02	**0.92**	0.92	0.81
0.49	**0.82**	0.74	0.41	0.01
0.89	0.44	0.18	0.89	**0.14**

score: 2.52

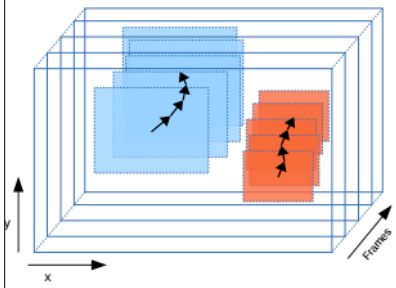
0.95	**0.76**	0.62	0.41	0.06
0.23	0.46	0.79	**0.94**	0.35
0.61	0.02	0.92	0.92	**0.81**
0.49	0.82	**0.74**	0.41	0.01
0.89	0.44	0.18	0.89	0.14

score: 4.14

Deep Learning for Visual Tracking

Object detection networks

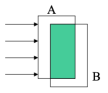
SORT: Tracking-by-detection



State Vector : Position, scale, ratio $\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$,

Trajectory prediction: Kalman filter

Association: IOU distance and Hungarian Algorithm



$$\text{score} = \frac{2 * \text{area(A and B)}}{\text{area(A)} + \text{area(B)}}$$

Traj

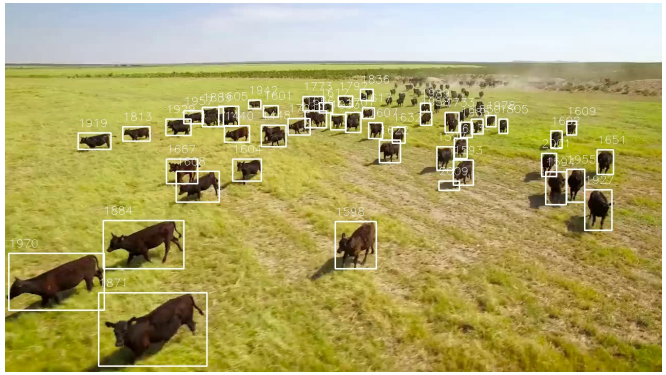
Detections

a1	a2	a3	a4
b1	b2	b3	b4
c1	c2	c3	c4
d1	d2	d3	d4

Bewley, Alex, ZongYuan Ge, Lionel Ott, Fabio Ramos, et Ben Upcroft. « Simple Online and Realtime Tracking ». *CoRR* abs/1602.00763 (2016). http://arxiv.org/abs/1602.00763.

Deep Learning for Visual Tracking

SORT: Tracking-by-detection

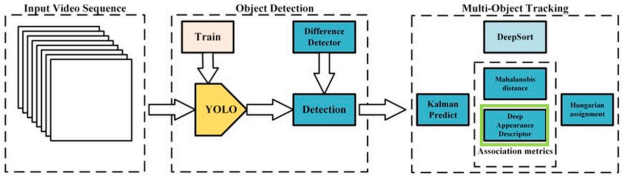


https://medium.com/neuromation-blog/tracking-cows-with-mask-r-cnn-and-sort-fcd4ad68ec4f

Deep Learning for Visual Tracking

DeepSORT: Tracking-by-detection

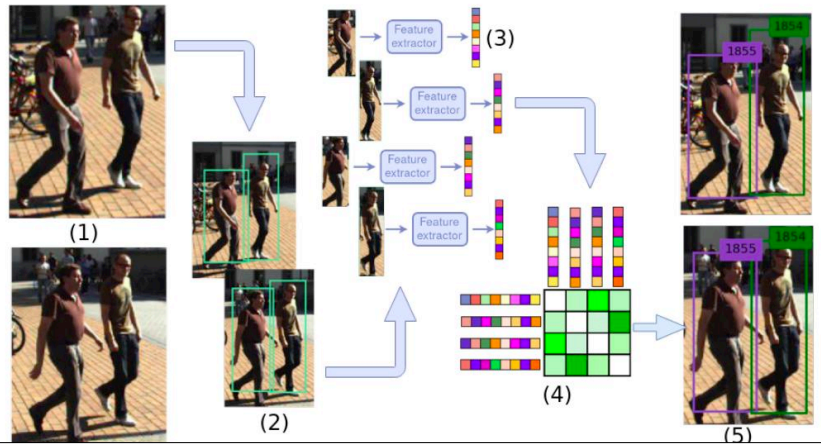
SORT WITH DEEP ASSOCIATION METRIC



https://medium.com/neuromation-blog/tracking-cows-with-mask-r-cnn-and-sort-fcd4ad68ec4f

Deep Learning for Visual Tracking

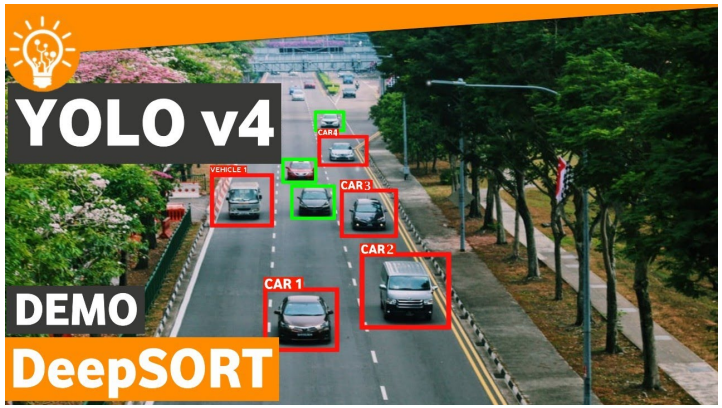
DeepSORT: Tracking-by-detection SORT WITH DEEP ASSOCIATION METRIC



Deep Learning for Visual Tracking

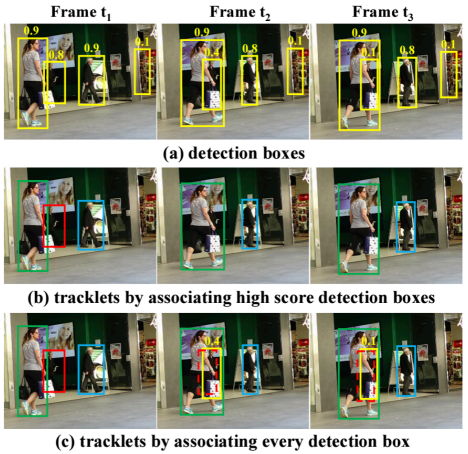
DeepSORT: Tracking-by-detection

SORT WITH DEEP ASSOCIATION METRIC



Deep Learning for Visual Tracking

ByteTrack: Multi Object Tracking



Deep Learning for Visual Tracking

Yellow: high score boxes
red: low score boxes

ByteTrack: Multi Object Tracking

Deep Learning for Visual Tracking

Yellow: high score boxes
red: low score boxes

ByteTrack: Multi Object Tracking

The power of video interlacing

Introduction

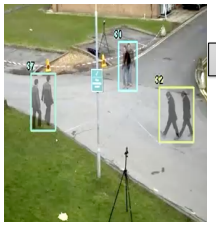
The classical tracking-by-detection scheme:

- object detection (for each frame of the video sequence)
- **association (spatio-temporal and/or appearance models)**
- birth and death trajectories algorithms

The power of video interlacing

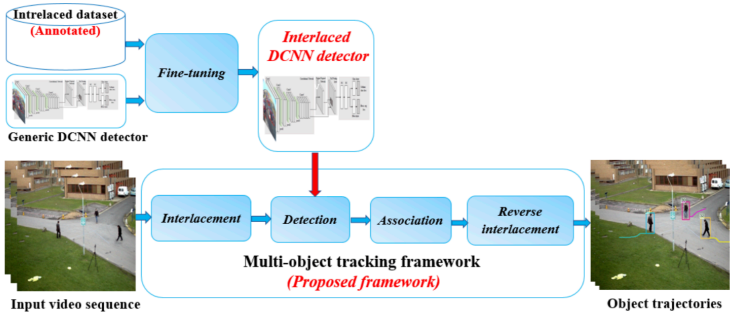
The key idea

build an **interlaced** video and train an interlaced pedestrian detector



to:
- increase overlapping between successive frames
- learn appearance association within a deep convolution neural network

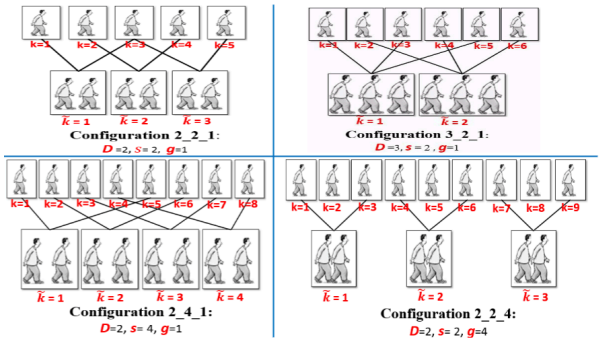
The power of video interlacing



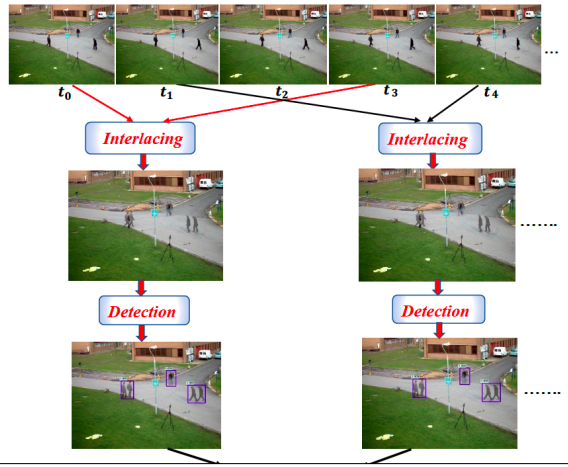
The power of video interlacing

Build a interlaced video

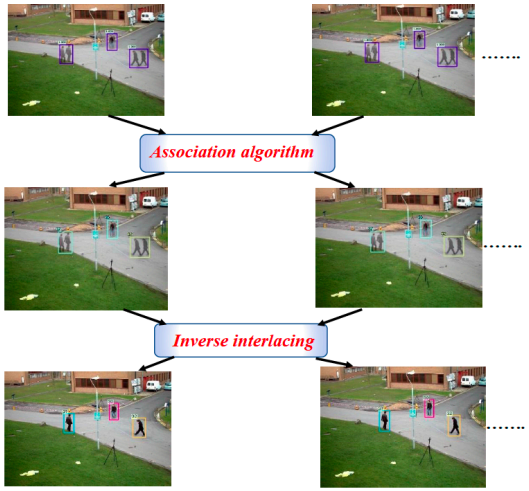
$$\tilde{\mathrm{I}}_{\tilde{k}}(x,y) \doteq \sum_{d=0,..,(D-1)} \mathrm{I}_{(\tilde{k}g+ds)}(x,y).\delta(y[D]-d)$$



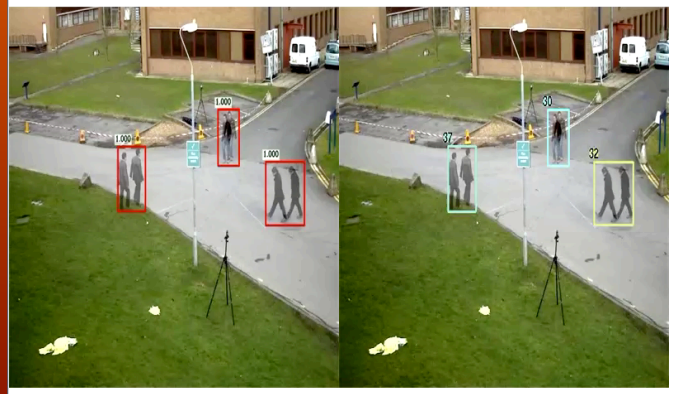
The power of video interlacing



The power of video interlacing



The power of video interlacing



Detection Tracking