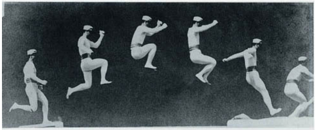
# Object Tracking in videos (cours 2)
# Rob 313

Gianni FRANCHI
ENSTA-Paris

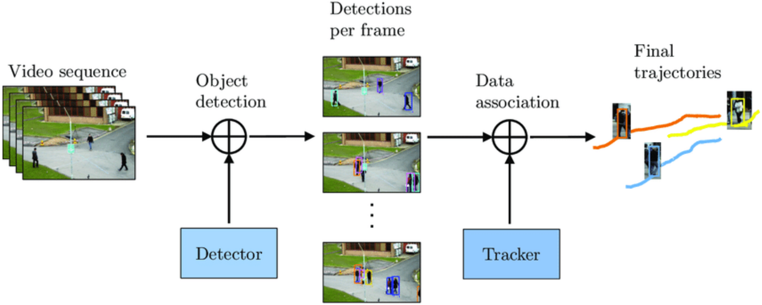


ENSTA
IP PARIS

# Fundamental Components of Tracking by detection

An object tracking algorithm is made of two fundamental elements:

- **Association**: associate the objects on two different frames
- **Detection**: localize the objects.

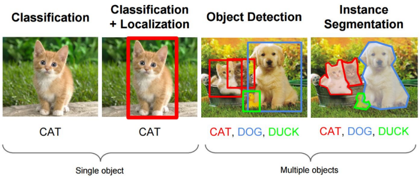


# Fundamental Components of Tracking by detection

Some of the most important challenges of Tracking by detection include:

- **Missed detections:** long-term occlusions are usually present in semi-crowded scenarios. In this case, it is very hard for the tracker to re-identify the pedestrian.
- **False alarms:** the detector can be triggered by regions in the image that actually do not contain any pedestrian, creating false positive.
- **Similar appearance:** one source of information commonly used for pedestrian iden-tification is appearance. However, in some videos similar clothing can lead to virtually identical appearance models for two different pedestrians.
- **Groups and other special behaviors:** when dealing with semi-crowded scenarios, it is very common to observe social behaviors like grouping.

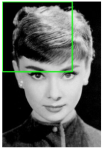
# What is object detection?



# Before Deep Learning

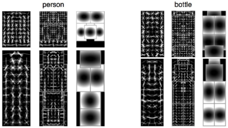
Sliding windows.

- Score every subwindow.



Deformable part models (DPM)

- Uses HOG features
- Very fast



# Before Deep Learning

# Different type of Object detection

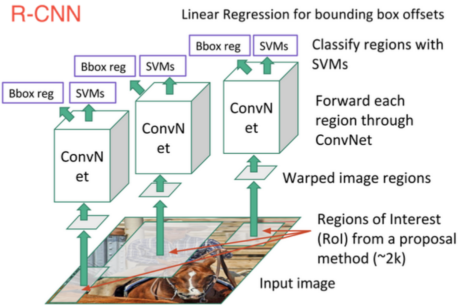
Two stages strategy:

- RCNN
- Fast RCNN
- Faster RCNN

One stage strategy:

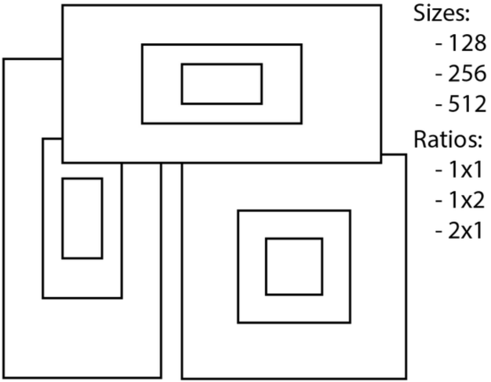
- Yolo
- SSD
- RetinaNet

# RCNN


R-CNN
Linear Regression for bounding box offsets
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Classify regions with SVMs
ConvNet
ConvNet
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

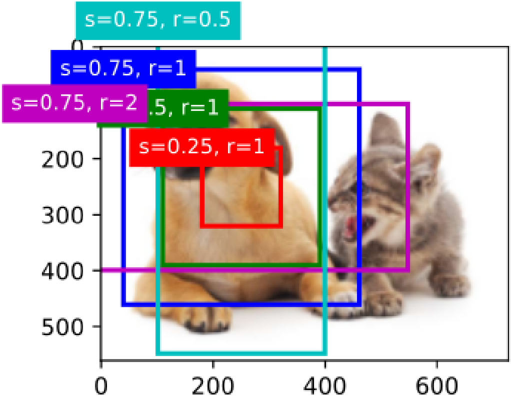
# Region proposal Network (RPN)

RPN will predict the best anchor. But what is an Anchor?



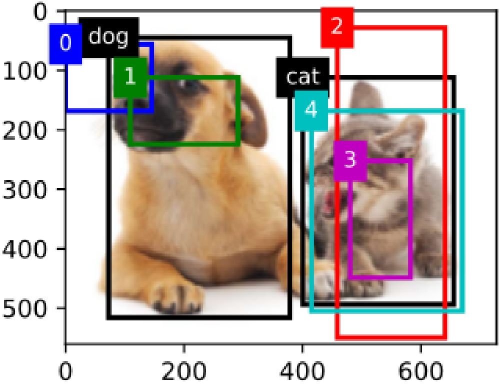
# Region proposal Network (RPN)

How do we define the best anchor?

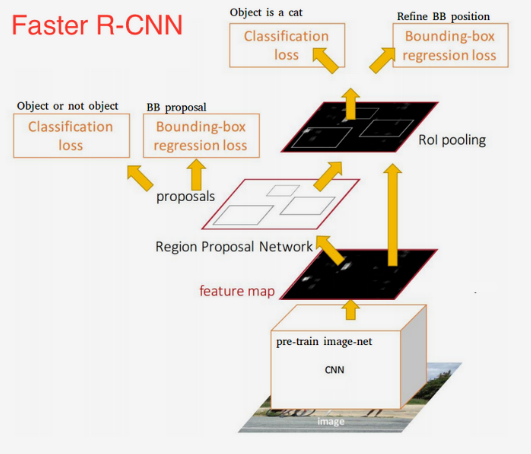


# Region proposal Network (RPN)

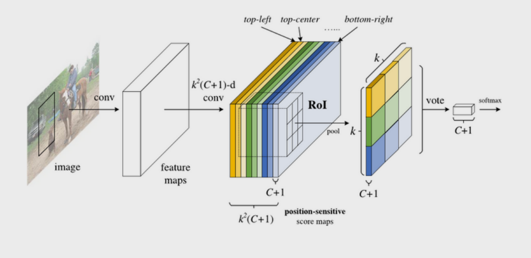
How do we define the best anchor?



# Faster RCNN


Faster R-CNN
Object is a cat
Classification loss
Refine BB position
Bounding-box regression loss
Object or not object
Classification loss
BB proposal
Bounding-box regression loss
RoI pooling
proposals
Region Proposal Network
feature map
pre-train image-net
CNN
image

# R-FCNN

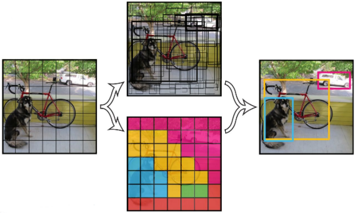

top-left
top-center
bottom-right
image
conv
feature
maps
$k^2(C+1)$-d
conv
RoI
pool
$k$
$k$
vote
softmax
$C+1$
$C+1$
$C+1$
$k^2(C+1)$
position-sensitive
score maps

# Yolo: You Only Look Once

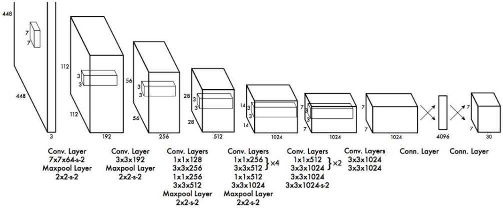
The following predictions are made for each cell in an $S \times S$ grid. output of YOLO:

- $C$ conditional class probabilities $P(\text{Classi}|\text{Obj})$
- $B$ bounding boxes (4 parameters each)
- $B$ confidence scores $P(\text{Obj})$
- Output is $SxSx(5B + C)$ tensor

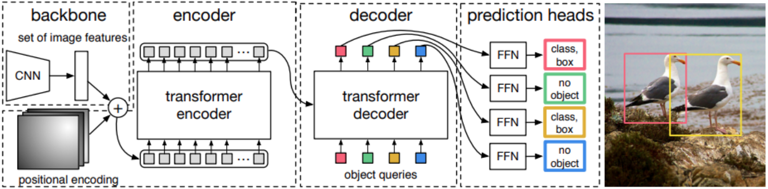
conditional class probabilities $P(\text{Classi}|\text{Obj})$



# Yolo: You Only Look Once



# Introduction to DETR [Carion2020]



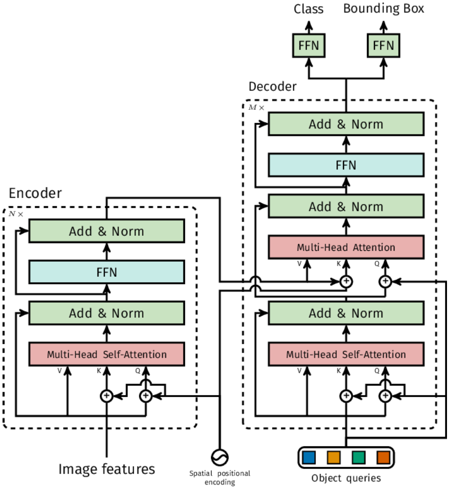
# DETR Architecture Overview

- **Components:**
  - **Backbone:** A CNN extracts image features.
  - **Transformer:** Processes features and models relationships.
    - **Encoder:** Self-attention captures global image context.
    - **Decoder:** Attention mechanism predicts objects.
  - **Prediction Heads:** Feedforward networks (FFNs) for class and bounding box predictions.
- **Object Queries:**
  - Fixed embeddings used by the decoder to predict objects.

# How DETR Works

- **End-to-End Workflow:**
  1. CNN backbone processes the image into a feature map.
  2. Transformer encoder processes the feature map with positional encodings.
  3. Decoder uses learned object queries to predict a fixed set of objects.
  4. FFNs predict class probabilities and bounding boxes for each object query.
- **Parallel Processing:**
  - Predictions are made in a single pass, unlike traditional detectors.

# Introduction to DETR [Carion2020]



# Bipartite Matching Loss

- **Purpose:**
  - Matches predictions to ground truth objects uniquely, avoiding duplicates.
- **Matching Process:**
  - Uses the Hungarian algorithm to compute the optimal one-to-one assignment.
- **Loss Function** that combines:
  - Classification loss: Predicting the correct object class.
  - Bounding box loss: L1 loss and generalized IoU (GIoU) loss.

**Loss Equation:**

$$\mathcal{L}_{\text{total}} = \sum_i \left[ \mathcal{L}_{\text{class}}(\hat{y}_i, y_{\sigma(i)}) + \lambda \mathcal{L}_{\text{box}}(\hat{b}_i, b_{\sigma(i)}) \right] \tag{1}$$

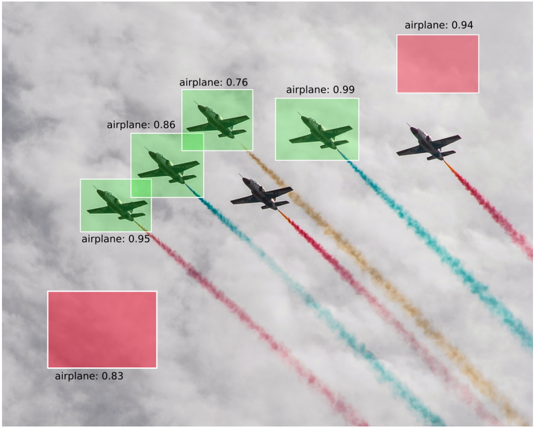
- $\mathcal{L}_{\text{class}}$: Cross-entropy for classification.
- $\mathcal{L}_{\text{box}}$: L1 loss + GIoU for bounding boxes.
- $\sigma(i)$: Optimal assignment from Hungarian matching.

# Key Advantages of DETR

- **Simplification:**
  - Removes the need for anchors and non-maximum suppression.
- **Global Context:**
  - Self-attention allows modeling relationships between objects.
- **Generalization:**
  - Performs well on large objects due to non-local reasoning.

# Criterion : mAP

How do we evaluate object detection



# Criterion : mAP

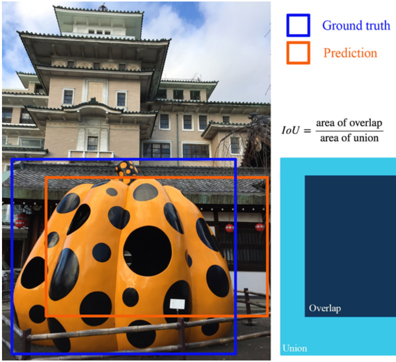
We need to define two metrics : **Precision** measures how accurate is your predictions. i.e. the percentage of your predictions are correct. **Recall** measures how good you find all the positives.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

# Criterion : mAP

We need to define TP, FP and FN for object detection.




$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

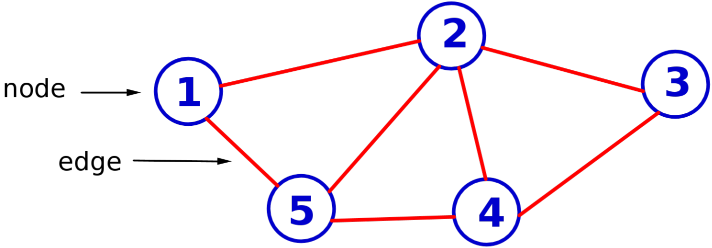
# Criterion : mAP

For the PASCAL VOC challenge, a prediction is positive if $IoU \geq 0.5$. First, we divide the recall value from 0 to 1.0 into 11 points = 0, 0.1, 0.2, . . . , 0.9 and 1.0. Next, we draw the **Recall-Precision curve** for this 11 value and integrate it. This process is applied for every class, and the mAP is the average.

# Graph Theory Basics (Remember)

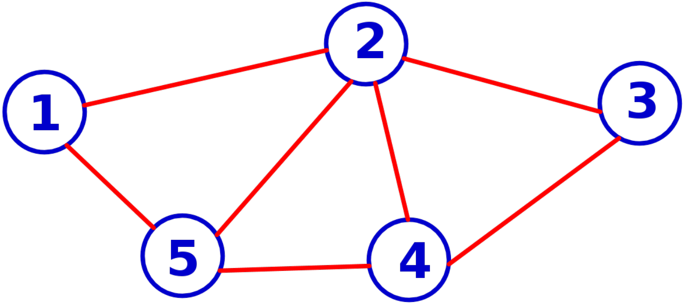
A graph is a data structure that is defined by two components :

- edges
- nodes (vertices)



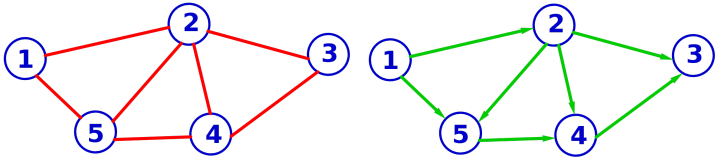
# Graph Theory Basics

We write a graph $G = (V, E)$ where $V$ is the set of nodes $E$ is the set of edges. $E \subseteq \{(x, y)|(x, y) \in V^2\}$
On the following case $V = \{1, 2, 3, 4, 5\}$ and
$E = \{(1, 2), (1, 5), (2, 5), (2, 4), (2, 3)\}$



# Directed/Undirected graphs (Remember)

First, let us make a distinction between a directed graph and an undirected graph. A directed graph or digraph is a graph in which edges have orientations, while it doesn't for an undirected graph.

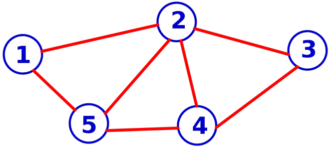



Undirected

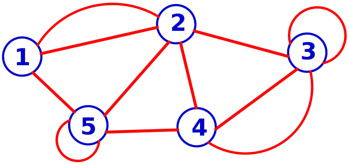
Directed

# Simple graphs / Multigraphs (Remember)

A simple graph is a graph without any loop in which two nodes are connected by at most one edge.
A multigraph is a graph that can have multiple edges that have the same nodes. Thus two nodes may be connected by more than one edge. One node can also have a self-loop.



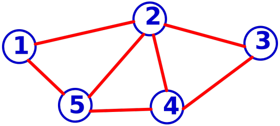
Simple graph



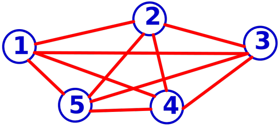
Multigraph

# Connected, Complete, Bipartite graphs (Remember)

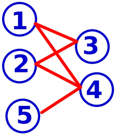
A connected graph is a graph composed of at least one vertex and there is a path (=finite or infinite sequence of edges which joins two nodes) between every pair of nodes.
A complete graph is a graph whose each nodes are connected to all other nodes .
A bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets $U$ and $V$ such that all edges connect a node in $U$ to one in $V$ .




Connected graph

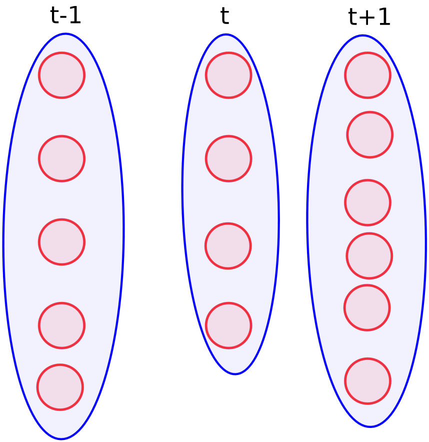



Complete graph



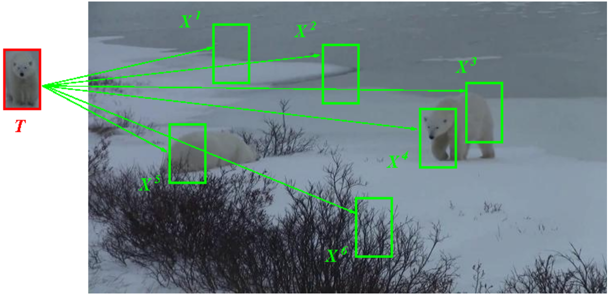

Bipartite graph

# Matching problem



# Local Matching

Global (di)similarity measures are applied between two vectors $T$ and $X$ of the same dimension $n$, one of which being the model (template) of the object, (usually) represented by a rectangular patch, and the other a patch extracted from the current image, that corresponds to a location hypothesis.

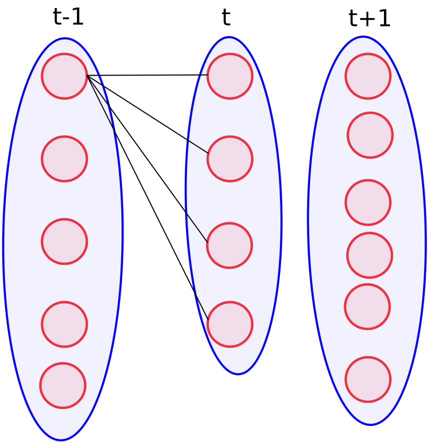


# Local Matching

## Principle

- At each frame we build all the possible associations between the set of active tracks and the current set of detections.
- We select first the association with the smallest distance and the corresponding track and detection $\mathcal{D}_k(X_{t-1,i}, X_{t,i})$.
- we discard this object from the association problem.
- we repeat the previous step up to when there is not object at time $t-1$
- Remaining detections are used to create new tracks, while non associated tracks are ended.

# Local Matching



# Global Matching

## Principle

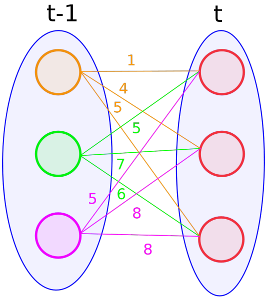
We look for the set of associations between tracks at time $t-1$ and $N$ detections at time $t$ with the smallest sum of distances : $\sum_{i=1}^{n_{t-1}} \sum_{j=1}^{n_t} \|X_{t-1,i} - X_{t,j}\|^2$. In practice, it is necessary to enumerate all the possible combinations of associations, which may turn out to be time consuming. On efficient algorithm to perfom this task is the hungarian algorithm.

# Hungarian Algorithm

Converting this problem to a formal mathematical definition we can form the following equations:

- let us define a cost matrix $C \in M_n(\mathbb{R})$, where $C_{ij}$ is the matching cost of object $i$ at time $t-1$ to object $j$ at time $t$.
- let us define a result binary matrix $R \in M_n(\mathbb{R})$, where $R_{ij} = 1$ if and only if object $i$ at time $t-1$ is assigned to object $j$ at time $t$.
- One object $i$ at time $t-1$ has just one object $j$ at time $t$ assign $\sum_{j=1}^{n} R_{ij} = 1$
- one object $j$ at time $t$ is assign to just one object $i$ at time $t-1$ assign $\sum_{i=1}^{n} R_{ij} = 1$
- the total cost function is :$\sum_{i=1}^{n} \sum_{j=1}^{n} R_{ij} C_{ij}$

# Hungarian Algorithm




$$\begin{pmatrix} 1 & 4 & 5 \\ 5 & 7 & 6 \\ 5 & 8 & 8 \end{pmatrix} \tag{4}$$

# Hungarian Algorithm

- This problem is known as the assignment problem.
- It is a special case of the transportation problem, which in turn is a special case of the min-cost problem
- It could also be optimized thanks to linear programming problem

# Hungarian Algorithm

- For each row of the matrix, find the smallest element and subtract it from every element in its row.
- Do the same (as first step) for all columns.
- Cover all zeros in the matrix using minimum number of horizontal and vertical lines.
- Test for Optimality: If the minimum number of covering lines is n, an optimal assignment is possible and we are finished. Else if lines are lesser than n, we haven't found the optimal assignment, and must proceed to the next step
- Determine the smallest entry not covered by any line. Subtract this entry from each uncovered row, and then add it to each covered column. Return to step

# Hungarian Algorithm: example

$$\begin{pmatrix} 1 & 4 & 5 \\ 5 & 7 & 6 \\ 5 & 8 & 8 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 3 & 4 \\ 0 & 2 & 1 \\ 0 & 3 & 3 \end{pmatrix}$$

# Hungarian Algorithm: example

$$\begin{pmatrix} 0 & 3 & 4 \\ 0 & 2 & 1 \\ 0 & 3 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 3 \\ 0 & 0 & 0 \\ 0 & 1 & 2 \end{pmatrix}$$

# Hungarian Algorithm: example

$$\begin{pmatrix} 0 & 1 & 3 \\ 0 & 0 & 0 \\ 0 & 1 & 2 \end{pmatrix}$$

$$\begin{pmatrix} \color{red}{0} & 1 & 3 \\ \color{red}{0} & \color{red}{0} & \color{red}{0} \\ \color{red}{0} & 1 & 2 \end{pmatrix}$$

just 2 lines, so we need to do one more step.

# Hungarian Algorithm: example

$$\begin{pmatrix} \color{red}{0} & 1 & 3 \\ \color{red}{0} & \color{red}{0} & \color{red}{0} \\ \color{red}{0} & 1 & 2 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 0 & 2 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \color{blue}{0} & 0 & 2 \\ 1 & 0 & \color{blue}{0} \\ 0 & \color{blue}{0} & 1 \end{pmatrix}$$

the cost is 15.

# Multiple object tracking as Maximum likelihood as a global objective function

## Principle

- We propose here to consider the tracking task as the construction of the set of the most probable trajectories given the observed data.
- Let $\theta_i = \{o_i^{t_{\text{init}}}, \dots, o_i^{t_{\text{end}}}\}$ denote a track: a sequence of an arbitrary number of states at discrete time .
- Let $\Theta = \cup_{i=1}^{N} \theta_i$ denotes the set of trajectories composed of N tracks
- Let $Z^l = \{o_i^{t_k}\}_{i,k}$ is the observed data over the time sequence.

For tracking purposes we want to find the tracks $\Theta$ that maximize the posterior (MAP) of the tracks:

$$\mathcal{L}(\Theta) = P(\Theta / Z^l) \tag{5}$$

# Multiple object tracking as a Maximum A Posteriori probability as a global objective function

The posterior probability of the tracks is

$$\mathcal{L}(\Theta) = P(\Theta/Z^l) \tag{6}$$

By applying the Bayes rule, we can rewrite it:

$$\mathcal{L}(\Theta) = \frac{P(\Theta, Z^l)}{P(Z^l)} \tag{7}$$

We can just maximize the numerator then we have $P(\Theta, Z^l) = P(Z^l/\Theta)P(\Theta)$.
We also make the usual assumption that objects are detected and move independently, that the tracks are independent.

$$P(\Theta, Z^l) = \prod_{i,k} P(o_i^{t_k}/\Theta) \prod_i P(\theta_i) \tag{8}$$

# Multiple object tracking as a Maximum A Posteriori probability as a global objective function

$P(o_i^{t_k}/\Theta)$ is the probability that $o_i^{t_k}$ is a good detection that belongs to the tracks.

$$P(o_i^{t_k}/\Theta) = \begin{cases} 1 - \beta_{i,k} & \text{if } \exists \theta_j \text{ such that } o_i^{t_k} \in \theta_j \\ \beta_{i,k} & \text{otherwise.} \end{cases} \quad (9)$$

The likelihood function $P(o_i^{t_k}/\Theta)$ can model that the observations that are associated are true detections, and those that are not associated are false alarms

# Multiple object tracking as a Maximum A Posteriori probability as a global objective function

$$P(\theta_i) = P(\{o_i^{t_{\text{init}}}, \ldots, o_i^{t_{\text{end}}}\} \tag{10}$$

By assuming the track follow te Markov assumption :

$$P(\theta_i) = P_{in}(o_i^{t_{\text{init}}})P_{in}(o_i^{t_{\text{init}}+1}/o_i^{t_{\text{init}}})...P_{out}(o_i^{t_{\text{end}}}) \tag{11}$$

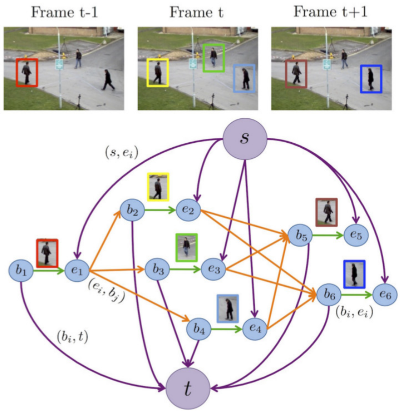
where $P_{in}(o_i^{t_{\text{init}}})$ is the probability that a trajectory $i$ is initiated with detection $o_i^{t_{\text{init}}}$ , $P_{out}(o_i^{t_{\text{end}}})$ is the probability the probability that the trajectory is terminated at $o_i^{t_{\text{end}}}$ and $P_{in}(o_i^{t_{\text{init}}+1}/o_i^{t_{\text{init}}})$ that $o_i^{t_{\text{init}}}$ is followed by $o_i^{t_{\text{init}}+1}$ in the trajector

# Multiple object tracking as a Maximum A Posteriori probability as a global objective functionn

How to solve this global objective function?

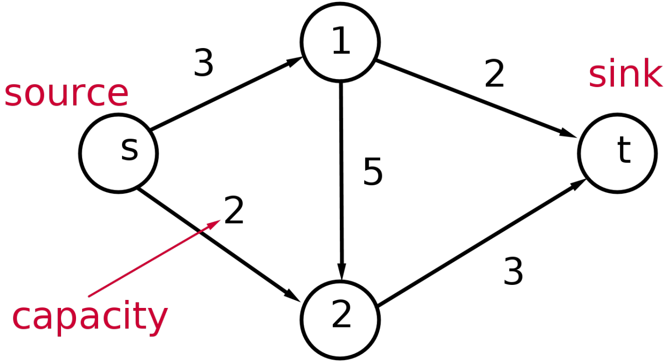
- Linear Programming
- Multiple hypotesis testing
- Max flow **[Zhang2008]**

# Multiple object tracking as Max flow problem [Laura14]



# Maximum flow:Introduction

The objective of the algorithm is to calculate the maximum amount of flow passing from the source to the sink.



# Maximum flow: Problem Definition

- Each edge is labeled with a capacity, that represents the maximum amount of stuff that it can carry.
- The goal is to figure out how much stuff can be pushed from the source to the sink while respecting all edges' capacitie

# Maximum flow: Problem Definition

Formally:

- Let us consider a directed graph $G = (V, E)$,
- One special node is the souce $s \in V$,
- each edge $e \in E$ has a non nonnegative and integer capacity $u_e$,
- we want to find for each edge $e \in E$ has a non nonnegative and integer flow $f_e$,

The goal is to find the flow $f_e$

- **Capacity constraints:** $f_e \leq u_e \ \forall e \in E$
- **Conservation constraints:** for every node $v$ expect $s$ and $k$ amount of flow entering $v$ = amount of flow exiting $v$.

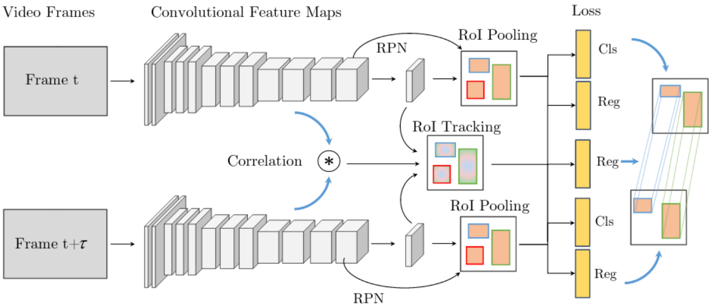
# Simple online and realtime tracking (SORT)[Bewley2016]

An object detection tracking algorithm with the following algorithm:

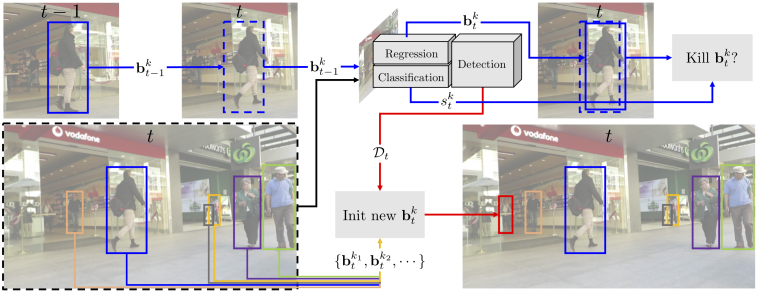
1. at time t, you apply the object detection DNN to have the objects.
2. you apply a Kalman filter with these objects between time t and t+1
3. at time t+1, you apply the object detection DNN to have the objects.
4. you use the IoU to check if the objects have an intersection and the Hungarian algorithm assigns the object to the tracks.
5. you change the object state. If new objects are found you build a new track.

https://www.youtube.com/watch?v=tq0BgncuMhs

# Detect to track and track to detect [Feichtenhofer2017]

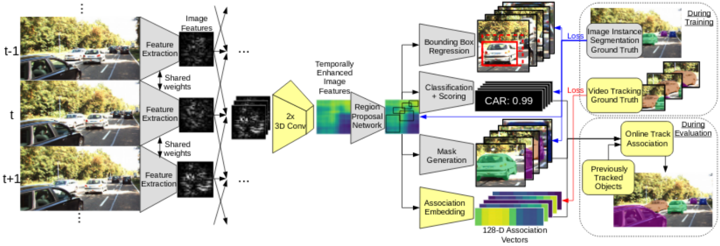


# Tracking with just an object detection algorithm [Bergmann2019]

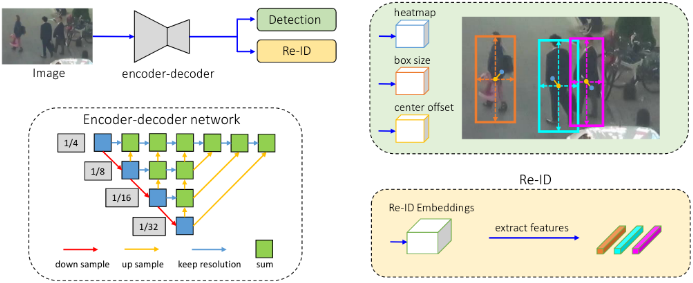



The presented Tracktor accomplishes multi-object tracking only with an object detector and consists of two primary processing steps, indicated in blue and red, for a given frame $t$. First, the regression of the object detector aligns already existing track bounding boxes $\mathbf{b}_{t-1}^{k}$ of frame $t-1$ to the object's new position at frame $t$. The corresponding object classification scores $s_t^k$ of the new bounding box positions are then used to kill potentially occluded tracks. Second, the object detector (or a given set of public detections) provides a set of detections $\mathcal{D}_t$ of frame $t$. Finally, a new track is initialized if a detection has no substantial Intersection over Union with any bounding box of the set of active tracks $B_t = \{\mathbf{b}_{t}^{k_1}, \mathbf{b}_{t}^{k_2}, \cdots\}$.

# Tracking with just an object detection and REID**[Voigtlaender2019]**



# Winner of MOT challenge 2020 [Wang2020]



# Bibliography - Tracking

[Jalal12] A.S. JALAL and V. SINGH
The State-of-the-Art in Visual Object Tracking
Informatica 36 (2012) 227-248

[VOT14] M. KRISTAN et al.
The Visual Object Tracking VOT2014 challenge results
Visual Object Tracking Workshop 2014 at ECCV2014, 2014

[Comaniciu03] D COMANICIU, V. RAMESH and P.MEER
Kernel-based object tracking
Pattern Analysis and Machine Intelligence, 25(5), 564-575 (2003)

# Bibliography - Tracking


- **[Isard98]** M. ISARD and M. BLAKE
  CONDENSATION - CONditional DENSity propagATION for visual tracking
  Int. Journal of Computer Vision (1998) 29(1), 5-28 (1998)
- **[Welsh01]** G. WELSH and G. BISHOP
  An Introduction to the Kalman Filter
  Tutorial of ACM SIGGRAPH (2001)
- **[Gordon93]** Gordon, Neil J., David J. Salmond, and Adrian FM Smith
  Novel approach to nonlinear/non-Gaussian Bayesian state estimation
  Digital Library, 1993

# Bibliography - Tracking

[Cheng95] Cheng, Yizong
"Mean shift, mode seeking, and clustering.
IEEE transactions on pattern analysis and machine intelligence 17.8 (1995) 790-799.

[Laura14] Leal-Taixe, Laura.
"Multiple object tracking with context awareness."
arXiv preprint arXiv:1411.7935 (2014).

[Zhang2008] Zhang, Li, Yuan Li, and Ramakant Nevatia
"Global data association for multi-object tracking using network flows."
IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008.

# Bibliography - Tracking

**[Feichtenhofer2017]** Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman.
"Detect to track and track to detect."
Proceedings of the IEEE International Conference on Computer Vision. 2017.

**[Bergmann2019]** Bergmann, Philipp, Tim Meinhardt, and Laura Leal-Taixe.
"Tracking without bells and whistles."
Proceedings of the IEEE international conference on computer vision. 2019.

**[Voigtlaender2019]** Voigtlaender, Paul, etal.
"MOTS: Multi-object tracking and segmentation."
Proceedings of the IEEE conference on computer vision and pattern recognition. 2019.

# Bibliography - Tracking

**[Wang2020]** Wang, Yongxin, Kris Kitani, and Xinshuo Weng. "Joint Object Detection and Multi-Object Tracking with Graph Neural Networks."

**[Carion2020]** Carion, Nicolas, et al. "End-to-end object detection with transformers." European conference on computer vision. Cham: Springer International Publishing, 2020.