Computer Vision

CS308

Feng Zheng

SUSTech CS Vision Intelligence and Perception

Week 14

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

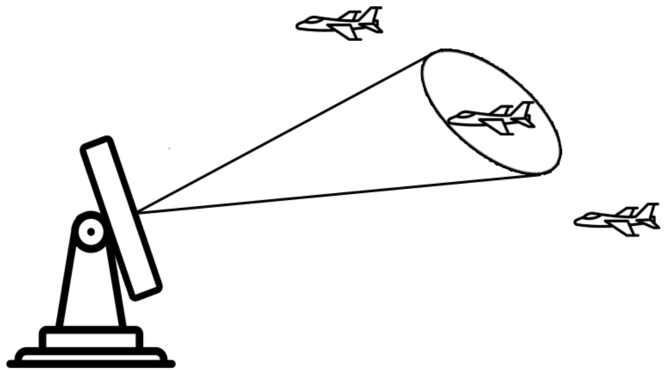
Content

- Object Tracking: General
- Siamese-based Tacker
- GOTURN Tracker
- Deep SORT Tracker

Object Tracking

What is tracking about?

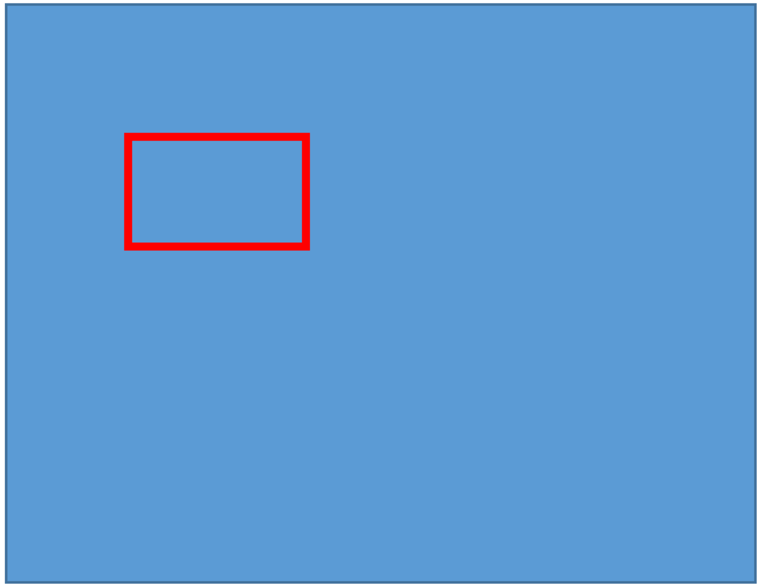
- All started in the early 60s
  - ➢ With Kalman filter for military
- Methods (location and appearance)
  - ➢ Data association (location)
  - ➢ Similarity measurement
  - ➢ Correlation
  - ➢ Matching/Retrieval
- Reasoning with "strong" priors: model-based
- Detection with very similar examples



Problem statement

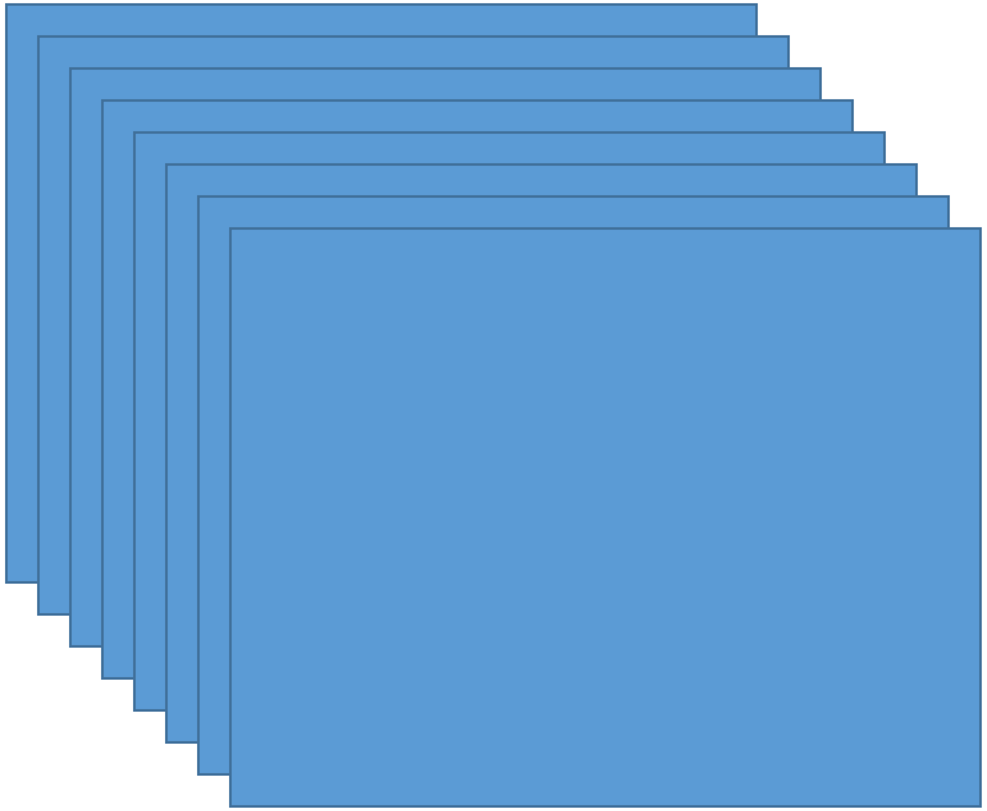
- Input: target
- Objective: Estimate target state over time
- State:
  - Position, Appearance, Shape, Velocity…
  - And affine transformation w.r.t. previous patch
- Choice: (O. S. S.)
  - Object representation
  - Similarity measure
  - Searching process

General Setting of Visual Tracking



Initialization in the first frame

......



Localization in the following frames

4 stages of the Object Tracking process

- Target initialization
- Appearance modeling
  - **Visual representation**: construct robust features and representation that can describe the object (template)
  - **Statistical modeling**: use statistical learning techniques to build mathematical models for object identification effectively.
- Motion estimation
- Target positioning

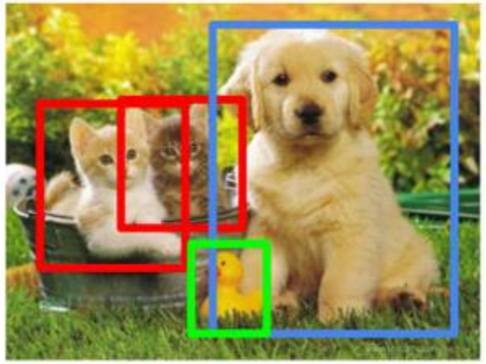
Levels of Object Tracking

- Single Object Tracking(SOT)
- Multiple Object Tracking(MOT): it aims to track objects of multiple classes as we see in self-driving cars

Object Tracking vs. Object detection

- Object tracking is to estimate or predict the position of a target object in each consecutive frame in a video once the initial position of the target object is defined
- Object detection is the process of detecting a target object in an image or a single frame of the video

Object Detection



CAT, DOG, DUCK

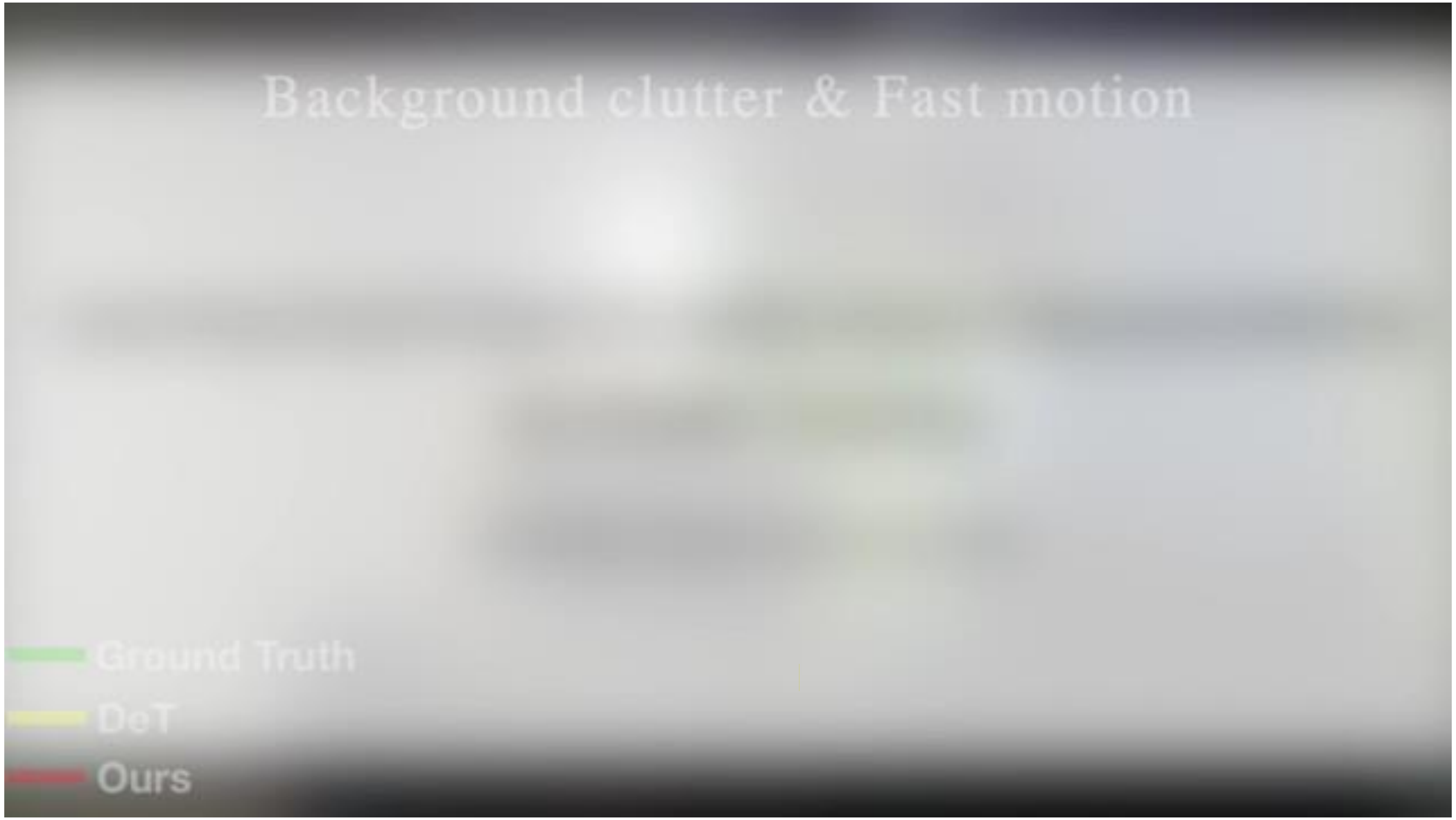
What are the challenges?

- Illumination Variation - the illumination in the target region is significantly changed
- Scale Variation – the ratio of the bounding boxes of the first frame and the current frame is out of the range
- Occlusion – the target is partially or fully occluded
- Deformation – non-rigid object deformation
- Motion Blur – the target region is blurred due to the motion of target or camera
- Fast Motion – the motion of the ground truth is larger than 20 pixels

What are the challenges?

- In-Plane Rotation – the target rotates in the image plane
- Out-of-Plane Rotation – the target rotates out of the image plane
- Out-of-View – some portion of the target leaves the view
- Background Clutters – the background near the target has the similar color or texture as the target
- Low Resolution – the number of pixels inside the ground-truth bounding box is less than 400

Object Tracking: challenges

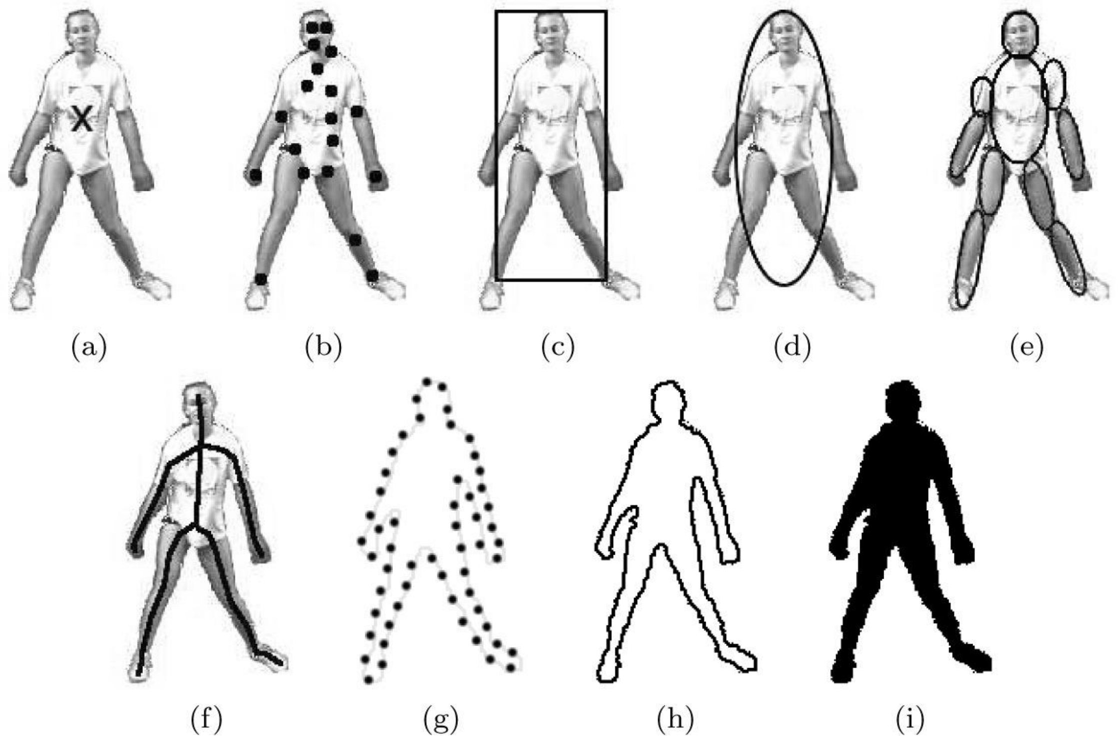


Object representation

- Taxonomy
  - Low/mid/high level features
  - Grid/Pyramid/Cascade
  - Patch/keypoints
- Goal: we want a representation that is:
  - Descriptive enough to disambiguate target VS background
  - Flexible enough to cope with:
    - ✓ Scale
    - ✓ Pose
    - ✓ Illumination
    - ✓ Partial occlusions

Object representation

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



Measuring Affinity

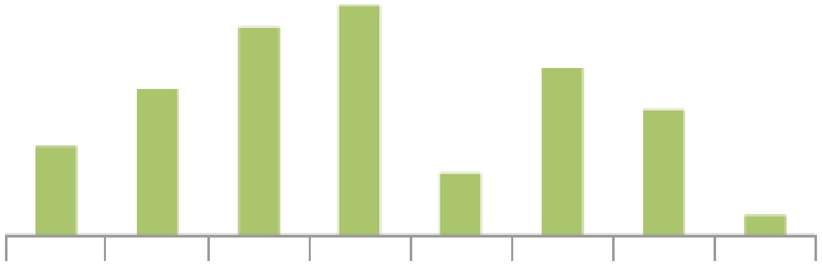
- In general: $$aff(x,y) = \exp\left(-\frac{1}{2\sigma_d^2}\|f(x) - f(y)\|^2\right)$$
- Examples:
  - Distance: $f(x) = location(x)$
  - Intensity: $f(x) = intensity(x)$
  - Color: $f(x) = color(x)$
  - Texture: $f(x) = filterbank(x)$

Pixels => Regions

- Note: Can also modify distance metric

Object representation: From light to useful information

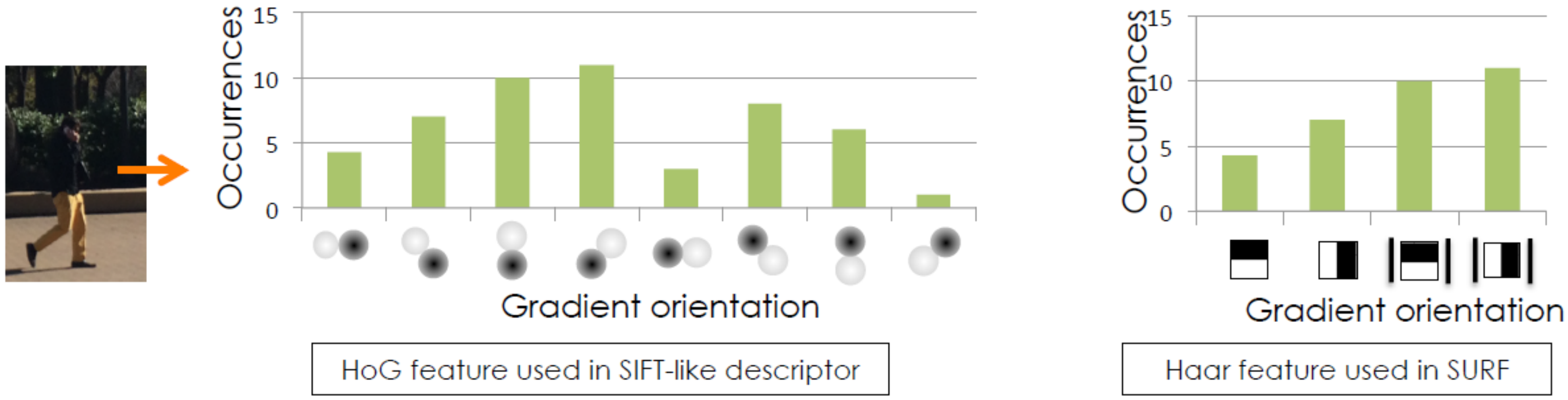
- Low/mid/high level features



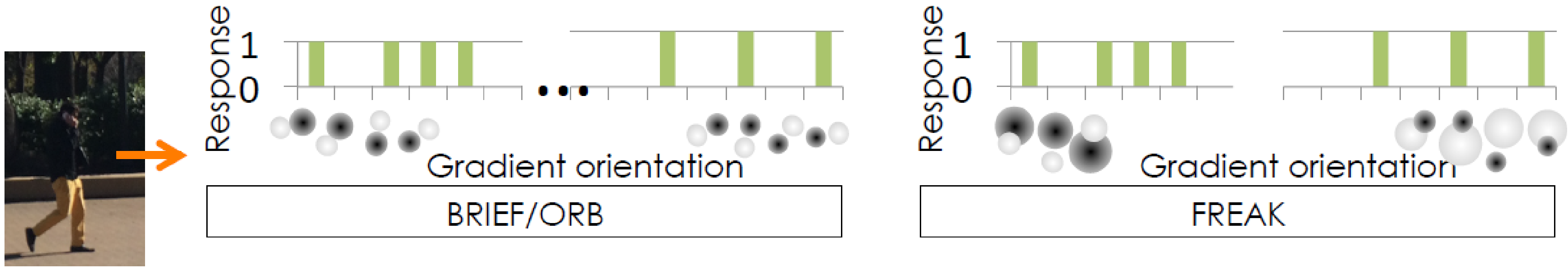
histograms

Low-level features

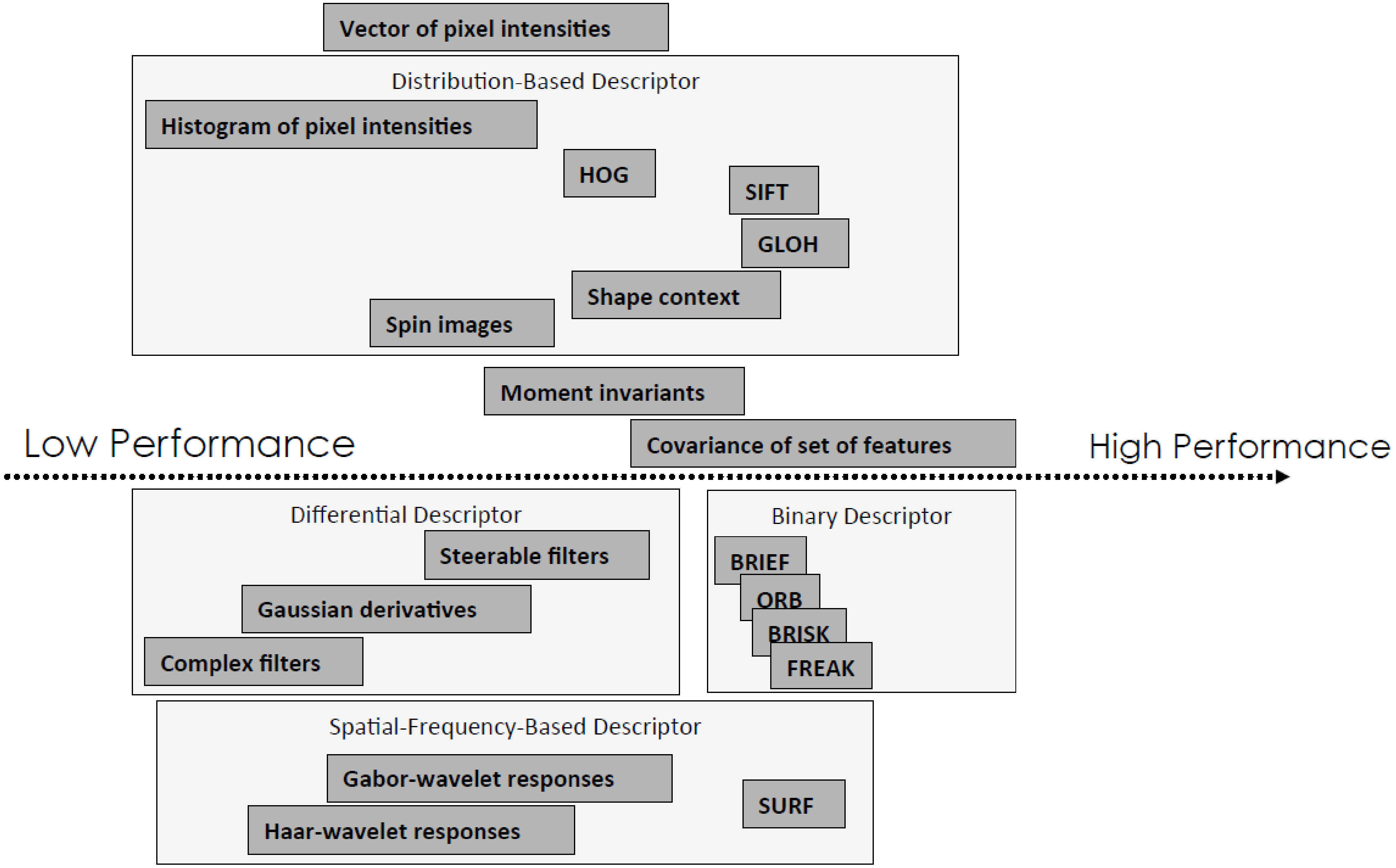
- Integer responses



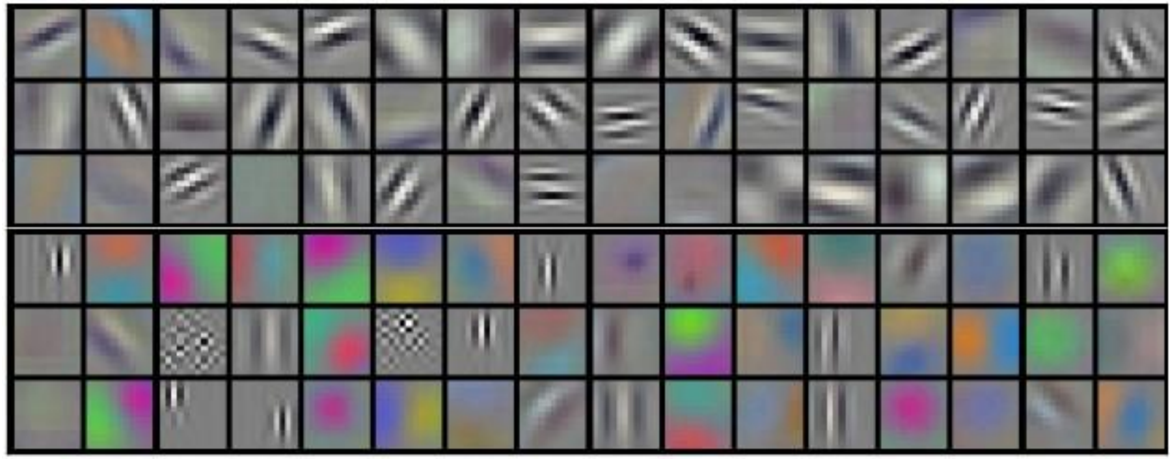
- Binary responses

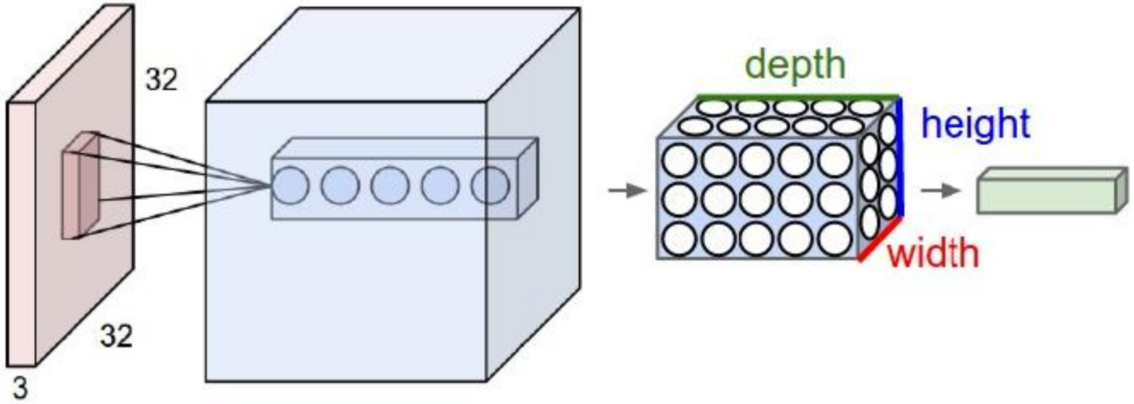


A bulk of Low-level features



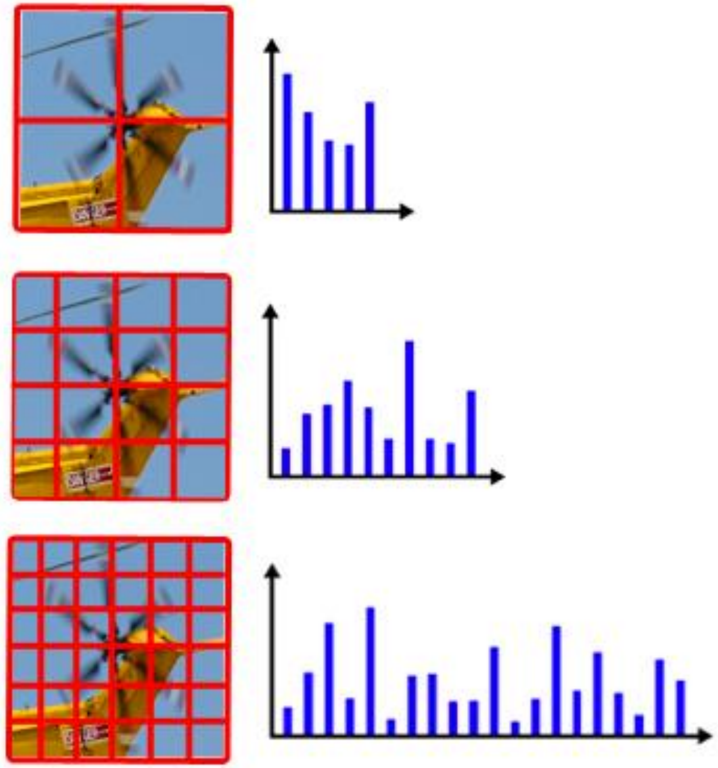
Recent trend: CNN features





Object representation: Sampling strategies

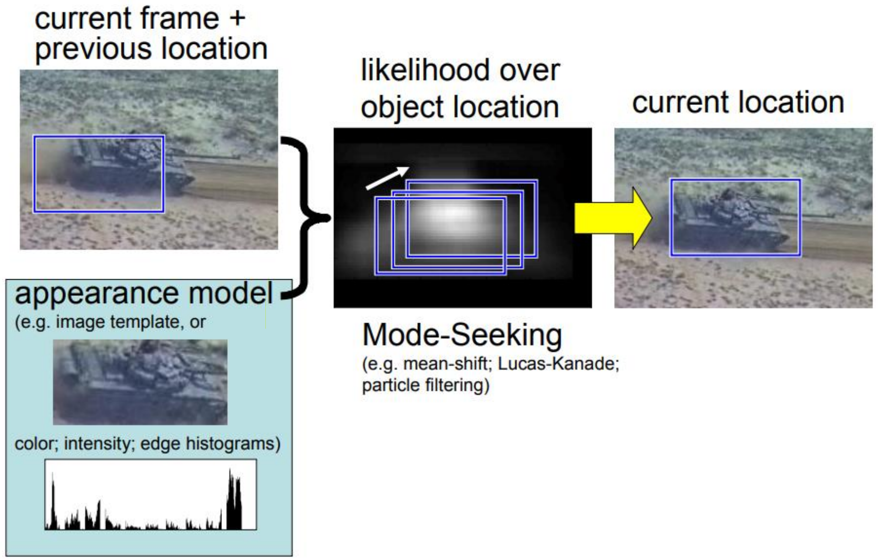
- Grid/pyramid/cascade of coarse-to-fine

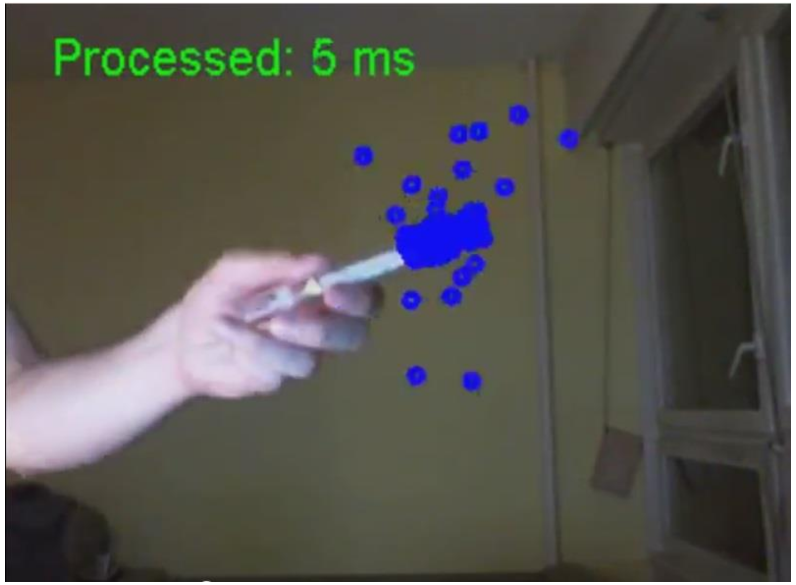


Classical Methods

- Optimal flow
- Kalman filter
- Particle filter
- Mean-shift

https://www.kalmanfilter.net/default.aspx



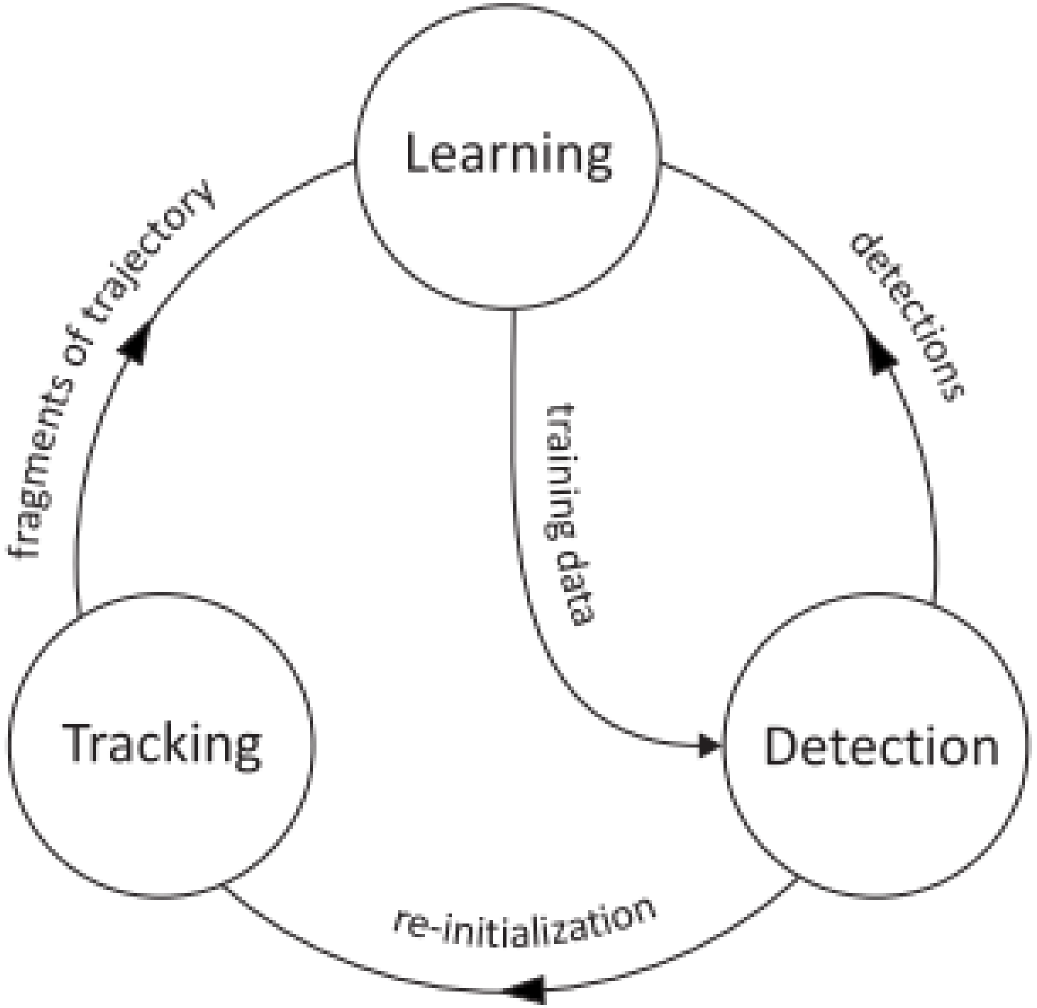


Online object tracking : Tracking-Learning-Detection

Tracking-Learning-Detection

Built in the first or last frame

Focus more on: Similarity



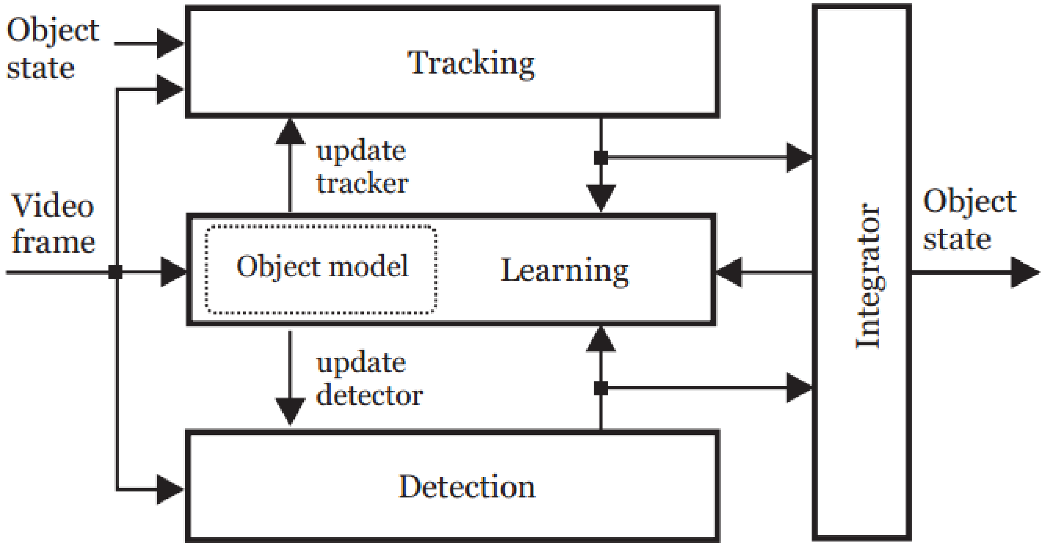
Built offline or in past frames

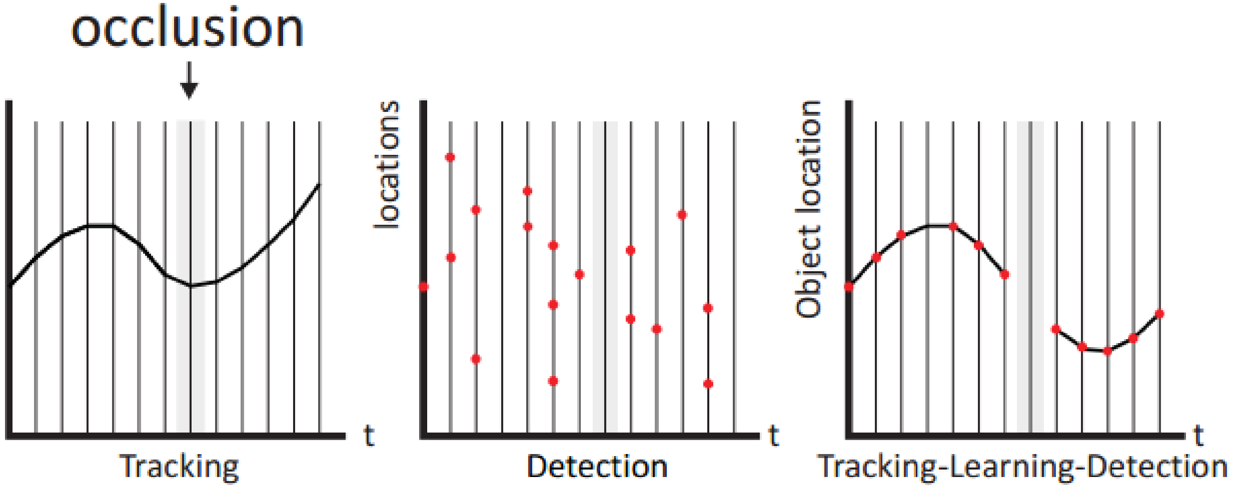
Focus more on: categories

https://github.com/gnebehay/TLD

Tracking-Learning-Detection

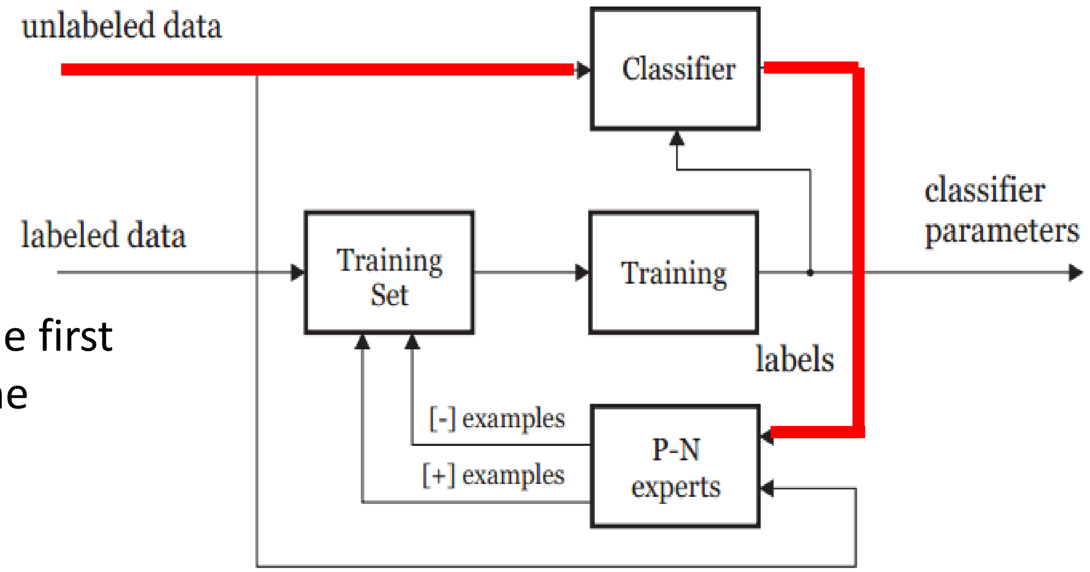
- Multiple models:
  - Tracking model (motion models: Kalman filter or Optimal flow)
  - Object model (features or templates)
  - Detection model (build in last frames)
- Cooperate and supervise each other





PN-learning

In the following frames



In the first frame

(i) A classifier for detection to be learned

(ii) Training set – a collection of labeled training examples

(iii) Supervised training– a method that trains a classifier from training set

(iv) P-N experts (object model)

Offline object tracking: GOTURN Tracker

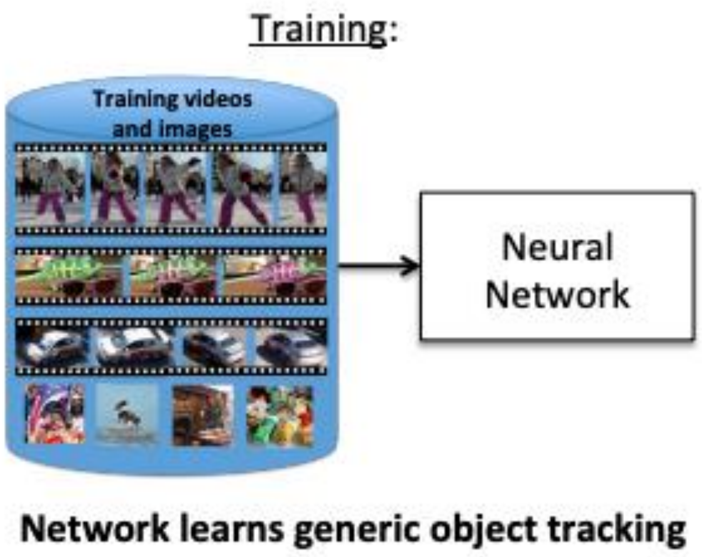
GOTURN Tracker

- GOTURN: Generic Object Tracking Using Regression Network
- Motivation
  - ➢ Previous trackers are trained from scratch online and do not benefit from the large number of videos
  - ➢ No real-time speed (<20FPS) using CNN models

- Solution
  - ➢ Offline training
  - ➢ Simple feed-forward network (no online training required, 100FPS)

"Learning to Track at 100 FPS with Deep Regression Networks." ECCV (2016).

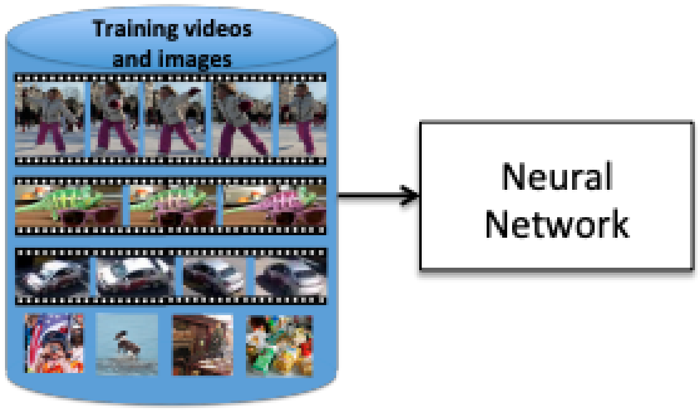
GOTURN Tracker



Offline training from large videos

GOTURN Tracker

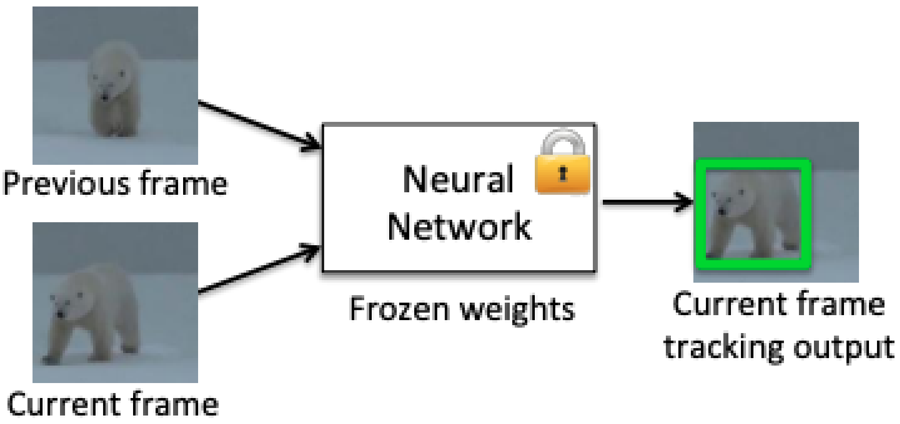
Training:



Network learns generic object tracking

Offline training from large videos

Test:

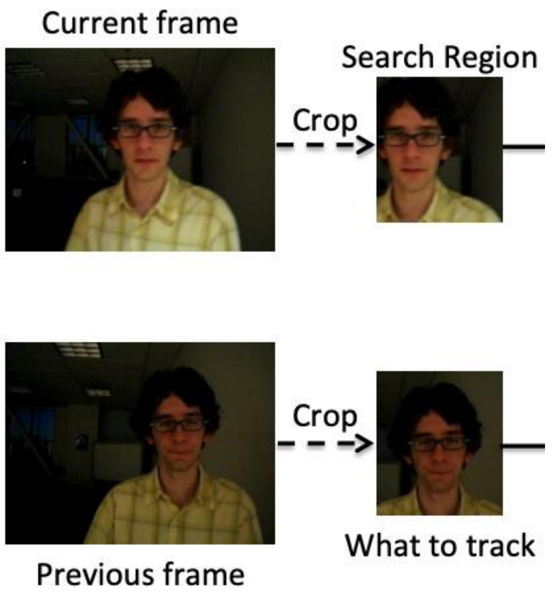


Network tracks novel objects (no finetuning)

No online learning (single feed-forward network)

high speed!

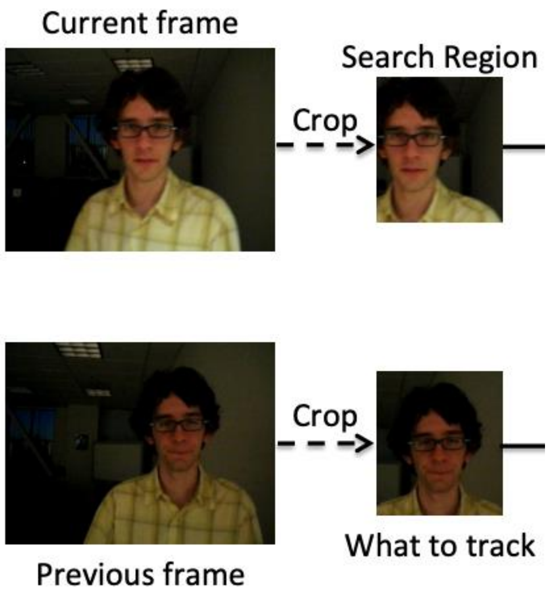
GOTURN Tracker



What to track?

- Crop and scale the previous frame to be centered on the target object
- Padding-receive extra contextual information about the surroundings

GOTURN Tracker



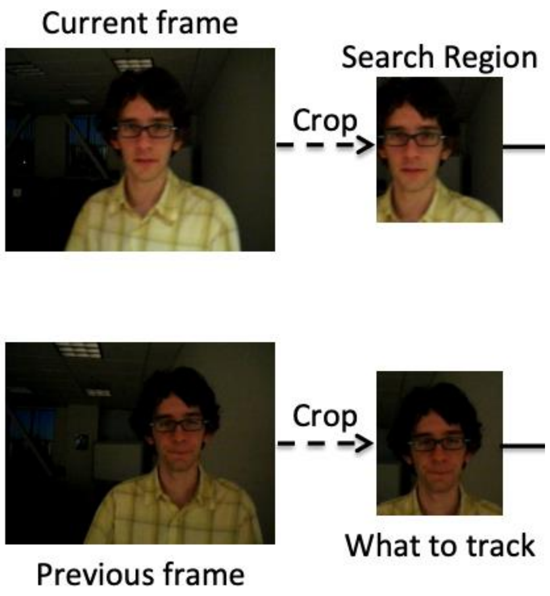
What to track?

- Crop and scale the previous frame to be centered on the target object
- Padding-receive extra contextual information about the surroundings

Where to look?

- Hypothesis: objects tend to move smoothly through space
- Choose a search region in current frame based on the object’s previous location—> crop

GOTURN Tracker



Where to look?

- Hypothesis: objects tend to move smoothly through space
- Choose a search region in current frame based on the object's previous location—> crop

$$c'_x = c_x + w \cdot \Delta x \qquad w' = w \cdot \gamma_w$$
$$c'_y = c_y + h \cdot \Delta y \qquad h' = h \cdot \gamma_h$$

– (c'_x, c'_y): the center of the bounding box in the current frame
– (c_x, c_y): the center of the bounding box in the previous frame
– w, h: the width and height
– $\Delta x, \Delta y$: position change of the bounding box relative to its size (a Laplace distribution with a mean of 0 ,in practice)
– γ_w, γ_h: the size change of the bounding box (modeled by a Laplace distribution with a mean of 1)

GOTURN Tracker

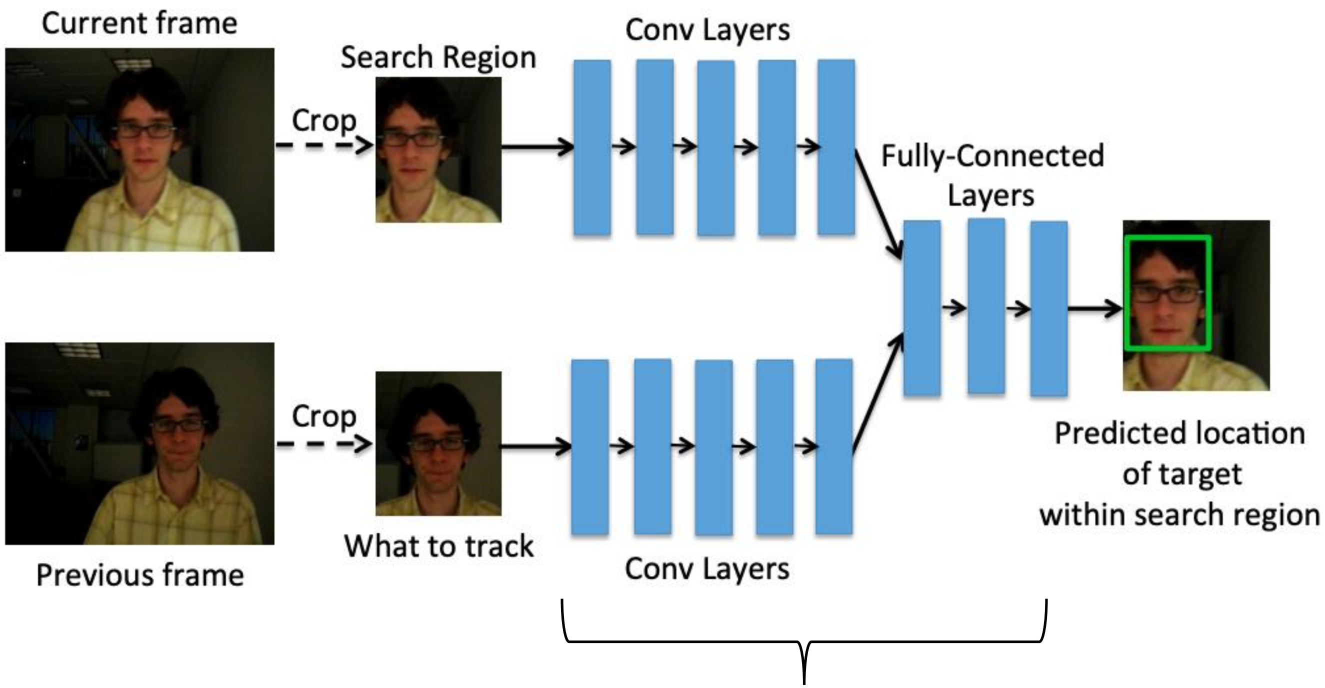


image-comparison tracking architecture

- Conv Layers: extract image features
- Fully connected layers: compare the features from the target object to the features in the current frame to find where the target object has moved and output bounding box.

GOTURN Tracker

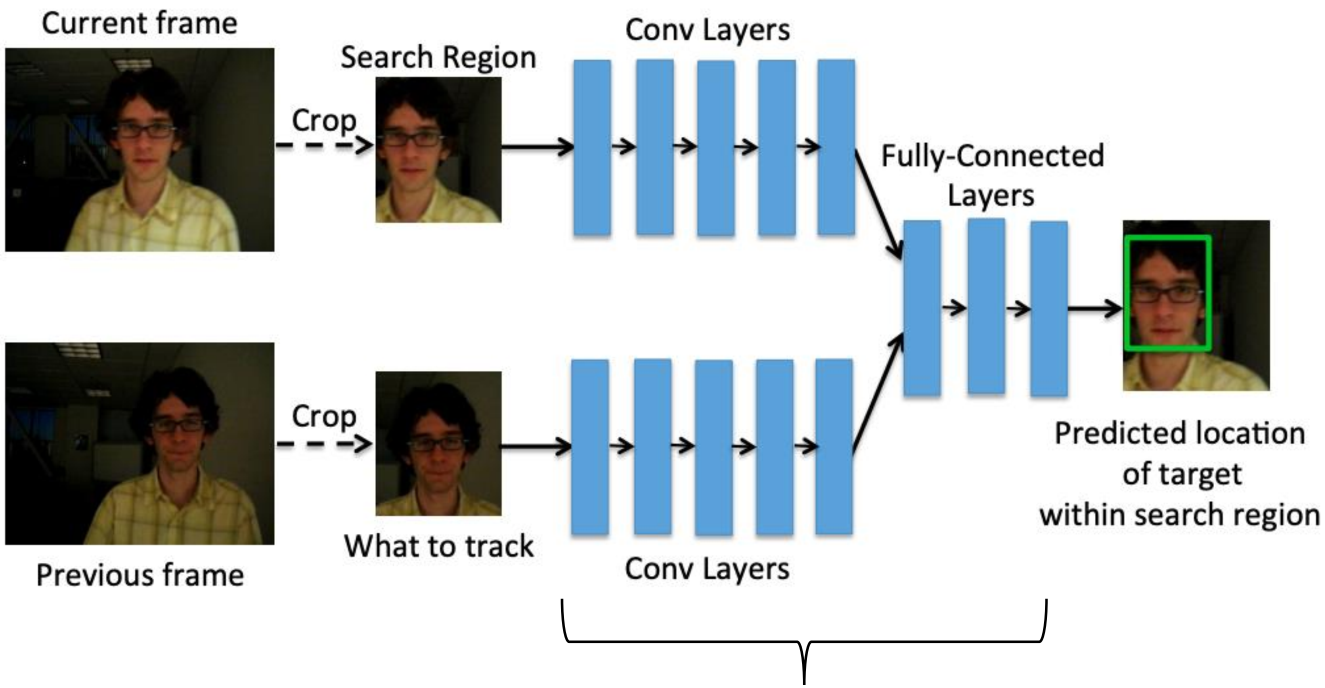
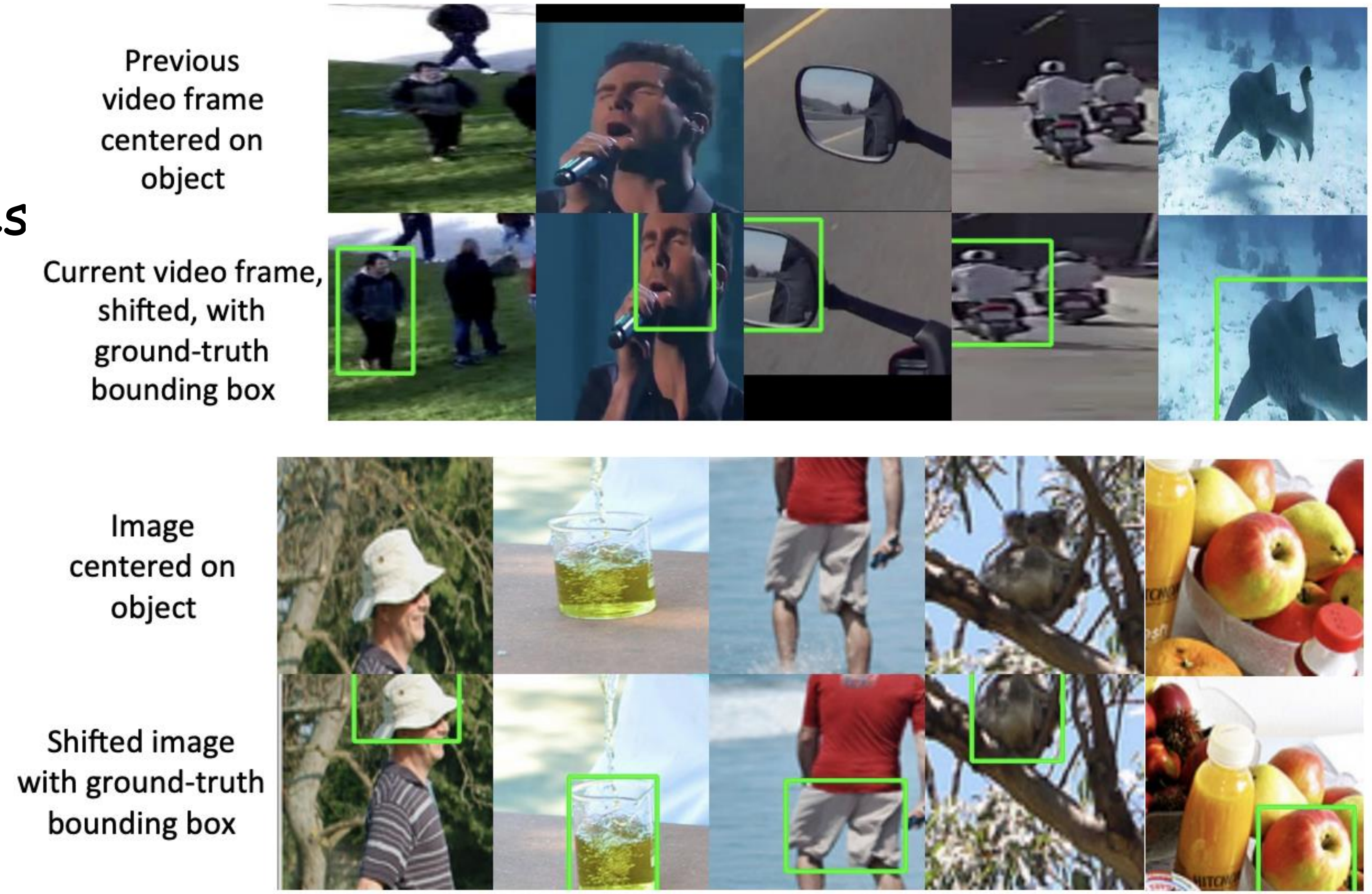


image-comparison tracking architecture

- Loss: L1 loss between predicted bounding box and ground truth.

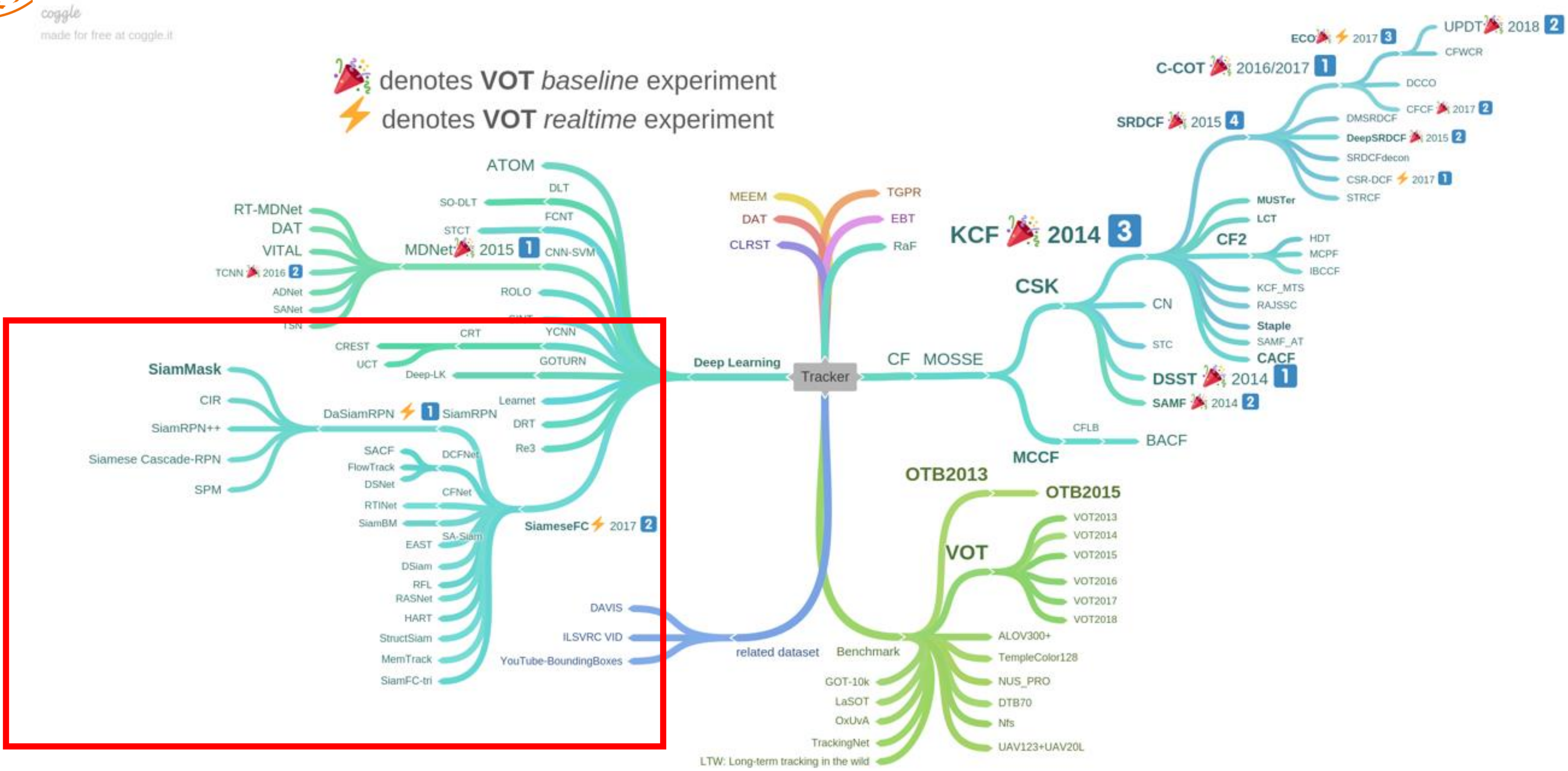
GOTURN Tracker

- Two types of training sets
  - A pair of images in a video
  - A image and a shifted image



Siamese-based Tracker

Siamese-based Tracker



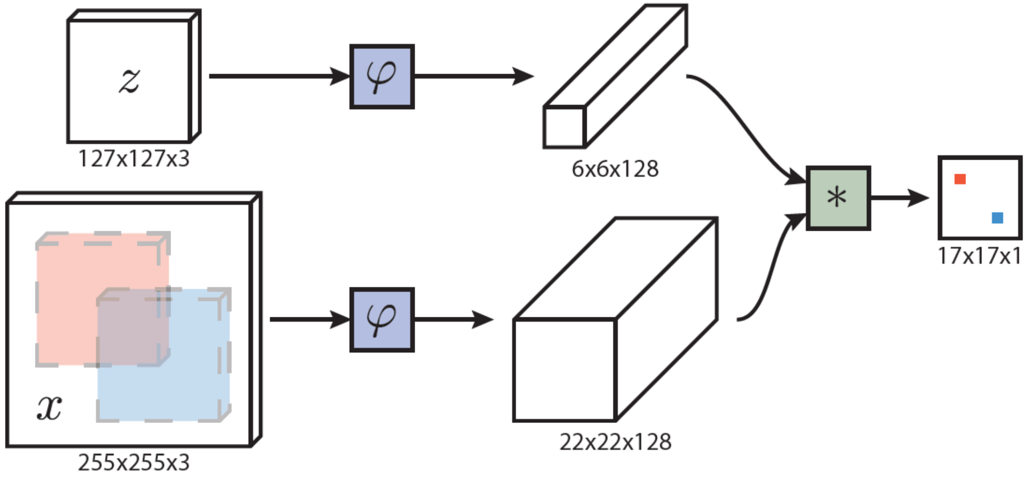
https://github.com/foolwood/benchmark_results

Siamese-based Tracker

- SiamFC
- SiamRPN
- SiamRPN++

SiamFC

- Architecture

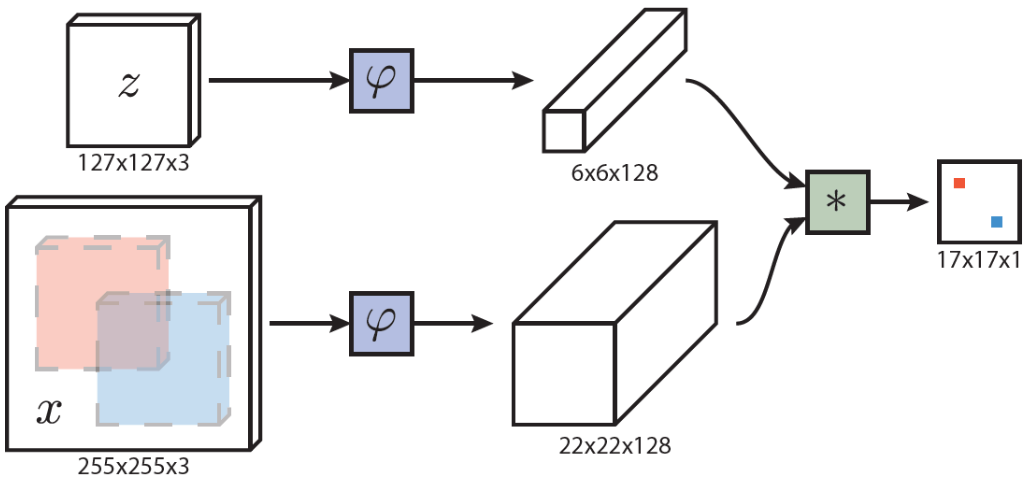


Layer	Support	Chan. map	Stride	Activation size for exemplar	Activation size for search	chans.
				127×127	255×255	$\times 3$
conv1	11×11	96×3	2	59×59	123×123	$\times 96$
pool1	3×3		2	29×29	61×61	$\times 96$
conv2	5×5	256×48	1	25×25	57×57	$\times 256$
pool2	3×3		2	12×12	28×28	$\times 256$
conv3	3×3	384×256	1	10×10	26×26	$\times 192$
conv4	3×3	384×192	1	8×8	24×24	$\times 192$
conv5	3×3	256×192	1	6×6	22×22	$\times 128$

"Fully-Convolutional Siamese Networks for Object Tracking." ECCV workshop (2016).

SiamFC

- Architecture



Images are scaled such that the bounding box, plus an added margin for context, has a fixed area.

- **Scaled Input image**

$$s(w + 2p) \times s(h + 2p) = A$$

scale factor s
A = 127^2
bounding box has size (w, h)
context margin is $p = (w + h)/4$

*** Network architecture**

*** Similarity measurement**

$$f(z, x) = \varphi(z) * \varphi(x) + b\mathbb{1}$$

*** Object location**

"Fully-Convolutional Siamese Networks for Object Tracking." ECCV workshop (2016).

SiamFC

- *** Loss function**

$$\ell(y, v) = \log(1 + \exp(-yv))$$

$$L(y, v) = \frac{1}{|\mathcal{D}|} \sum_{u \in \mathcal{D}} \ell(y[u], v[u])$$

the centre

$$y[u] = \begin{cases} +1 & \text{if } k\|u - c\| \le R \\ -1 & \text{otherwise} \end{cases} .$$

- v real-valued score of a single exemplar-candidate pair
- y ground-truth label
- D score map
- R radius

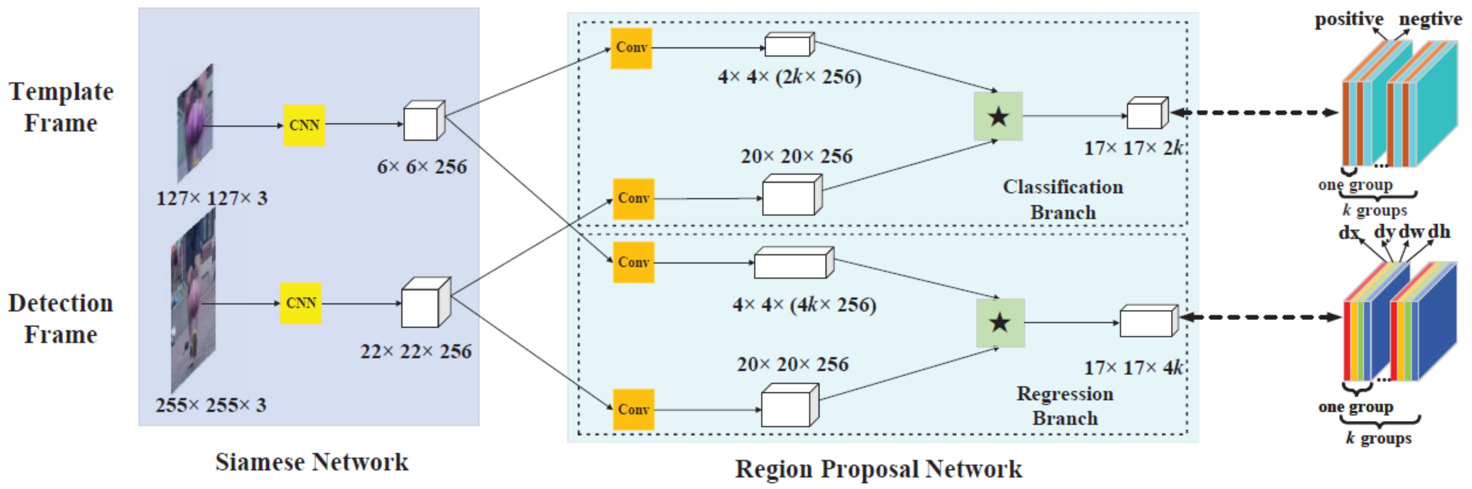
Siamese-based Tracker

- SiamFC
  - Lack of bounding box regression
  - Need to do multi-scale test
- SiamRPN
- SiamRPN++

SiamRPN

- Architecture

$2k$ represents for negative and positive activation of each anchor at corresponding location on original map



If there are k anchors, network needs to output $2k$ channels for classification and $4k$ channels for regression.

$$A^{cls}_{w\times h\times 2k} = [\varphi(x)]_{cls} \star [\varphi(z)]_{cls}$$
$$A^{reg}_{w\times h\times 4k} = [\varphi(x)]_{reg} \star [\varphi(z)]_{reg}$$

$4k$ represents for dx, dy, dw, dh measuring the distance between anchor and corresponding groundtruth

Can we have a deeper backbone? AlexNet->Resnet50/101

"High Performance Visual Tracking with Siamese Region Proposal Network." CVPR (2018 Spotlight).

SiamRPN

- Proposal generation
  - Collect the top K points in all $A^{cls}_{w \times h \times 2k}$
  - Get the corresponding refinement coordinates

$$A^{cls}_{w \times h \times 2k} = \{(x^{cls}_i, y^{cls}_j, c^{cls}_l)\} \Rightarrow ANC^* = \{(x^{an}_i, y^{an}_j, w^{an}_l, h^{an}_l)\} \Rightarrow$$

$$\begin{aligned} x^{pro}_i &= x^{an}_i + dx^{reg}_l * w^{an}_l \\ y^{pro}_j &= y^{an}_j + dy^{reg}_l * h^{an}_l \\ w^{pro}_l &= w^{an}_l * e^{dw_l} \\ h^{pro}_l &= h^{an}_l * e^{dh_l} \end{aligned}$$

- Proposal selection
  - Discard the bounding boxes generated by the anchors too far away from the center
  - Use cosine window and scale change penalty to re-rank the proposals' score to get the best one

$$penalty = e^{k*max(\frac{r}{r'}, \frac{r'}{r})*max(\frac{s}{s'}, \frac{s'}{s})}$$ Ratio and overall scale between proposal and last frame

Siamese-based Tracker

- SiamFC
- SiamRPN
  - ➢ Too shallow to meet the strict translation invariance (no padding)
  - ➢ Imbalance of parameter distribution (i.e. the RPN module contains 20M parameters while the feature extractor only contains 4M parameters)
- SiamRPN++

SiamRPN++

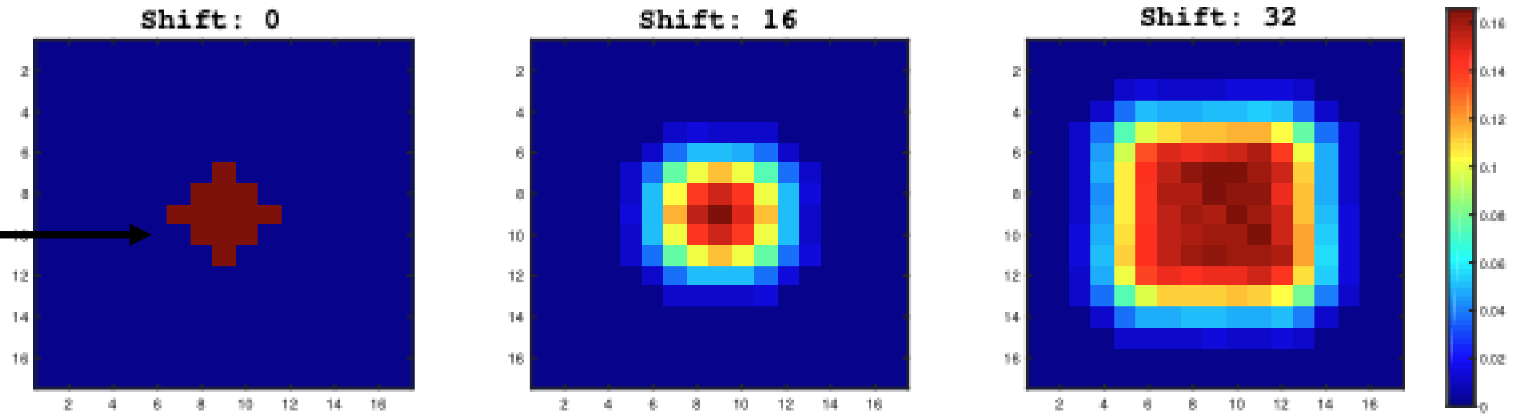
- Motivation: Can we have a deeper backbone?
  - The deeper, the better?
  - Need padding (resNet)
- Two restrictions in correlation : $f(z, x) = \Phi(z) * \Phi(x) + b$
  - Strict translation invariance. Padding ✗
  - Correlation symmetry. RPN requires asymmetrical for Cls and Reg.
- Solution
  - Spatial aware sampling strategy
  - Depthwise Cross Correlation

"SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks." CVPR Oral (2019).

Spatial aware sampling strategy

- Positive samples are evenly distributed within a certain range, not always at the center
- The range is the distance from the center point, which means shift

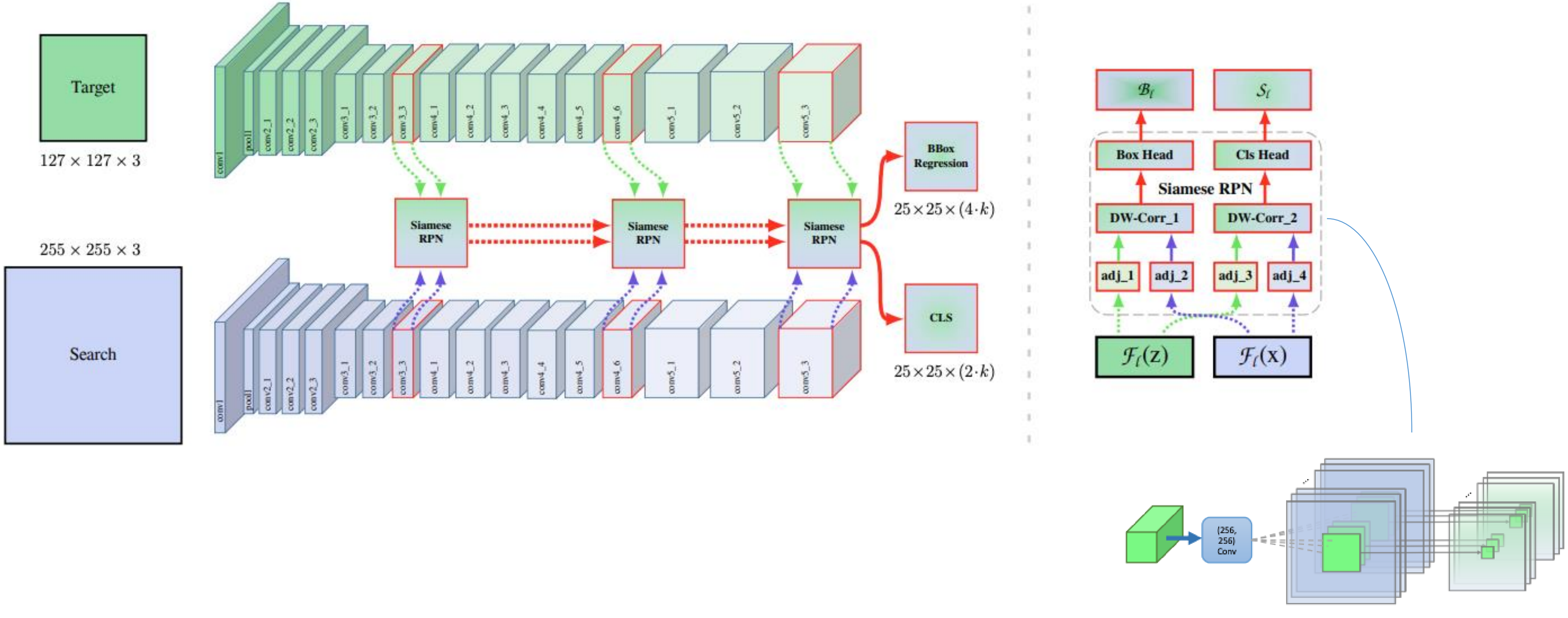
The disadvantage of a deep network that breaks strict translational invariance is that it learns positional bias.



Target may appear at any position in the search region and earned feature representation should stay spatial invariant

Layer-wise Aggregation

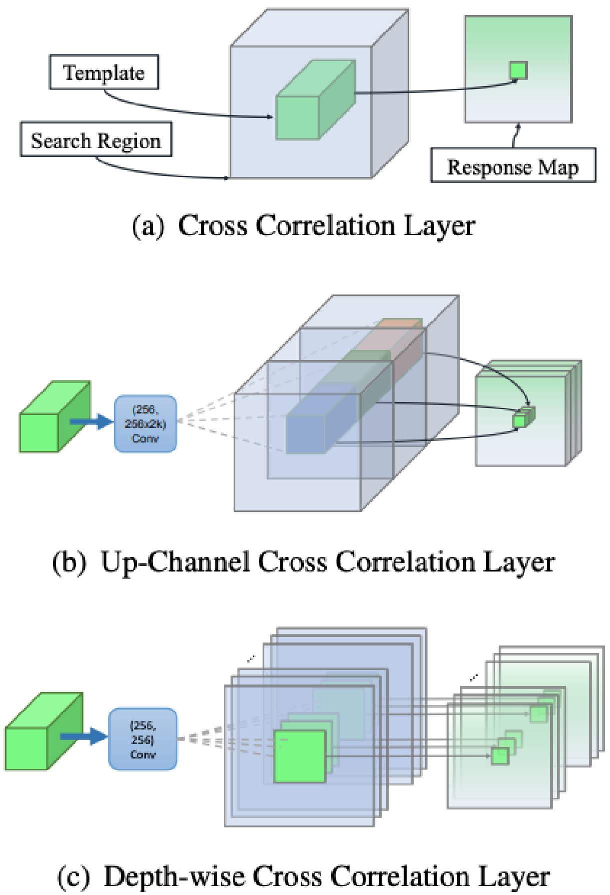
- Architecture



(c) Depth-wise Cross Correlation Layer

Depthwise Cross Correlation

- (a) SiamFC: Cross Correlation (XCorr) layer predicts a single channel similarity map between target template and search patches
- (b) SiamRPN: Up-Channel Cross Correlation (UP-XCorr) layer outputs a multi-channel correlation features by cascading a heavy convolutional layer with several independent XCorr layers
- (c) SiamRPN++: Depth-wise Cross Correlation (DW-XCorr) layer predicts multi-channel correlation features between a template and search patches



The same category have high response on same channels

Results

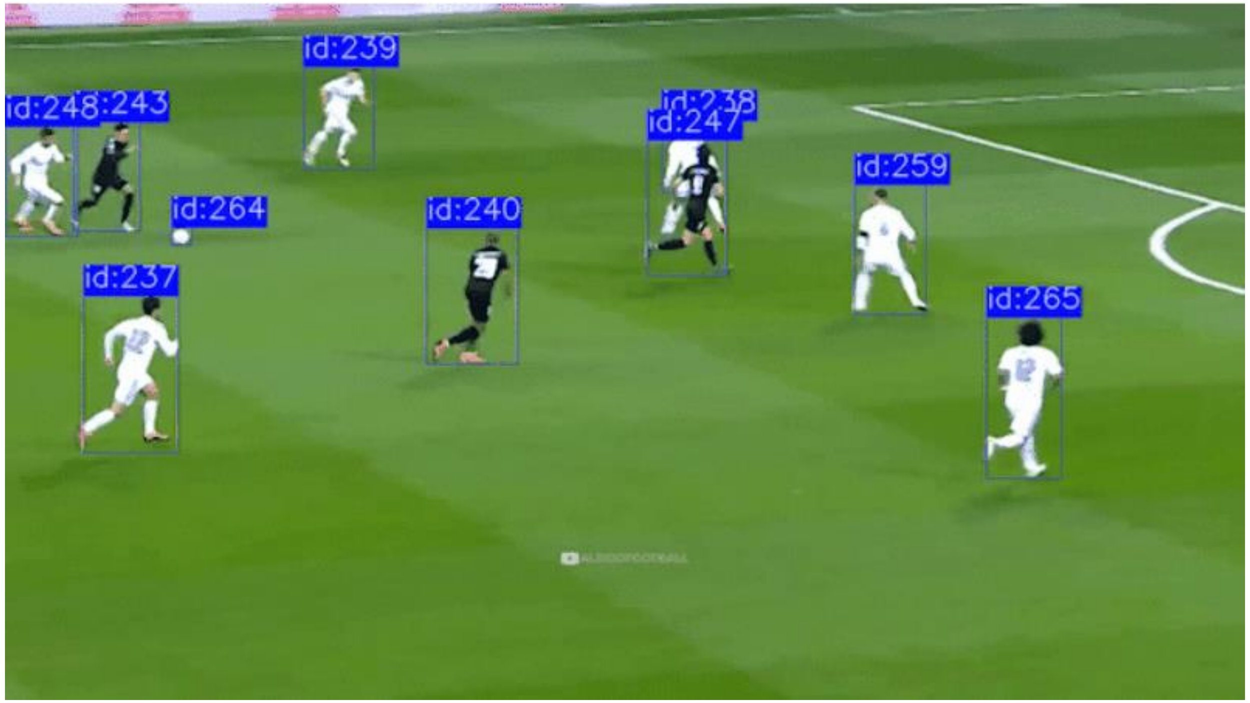
	CSRDCF [28]	ECO [5]	SiamFC [1]	CFNet [41]	MDNet [32]	DaSiamRPN [52]	**Ours**
AUC (%)	53.4	55.4	57.1	57.8	60.6	63.8	**73.3**
P (%)	48.0	49.2	53.3	53.3	56.5	59.1	**69.4**
P_{norm} (%)	62.2	61.8	66.3	65.4	70.5	73.3	**80.0**

Deep SORT Tracker

What is multiple object tracking(MOT)?

- What is MOT:

Locate multiple objects of interest in a given video simultaneously, maintain their IDs and record their trajectories.



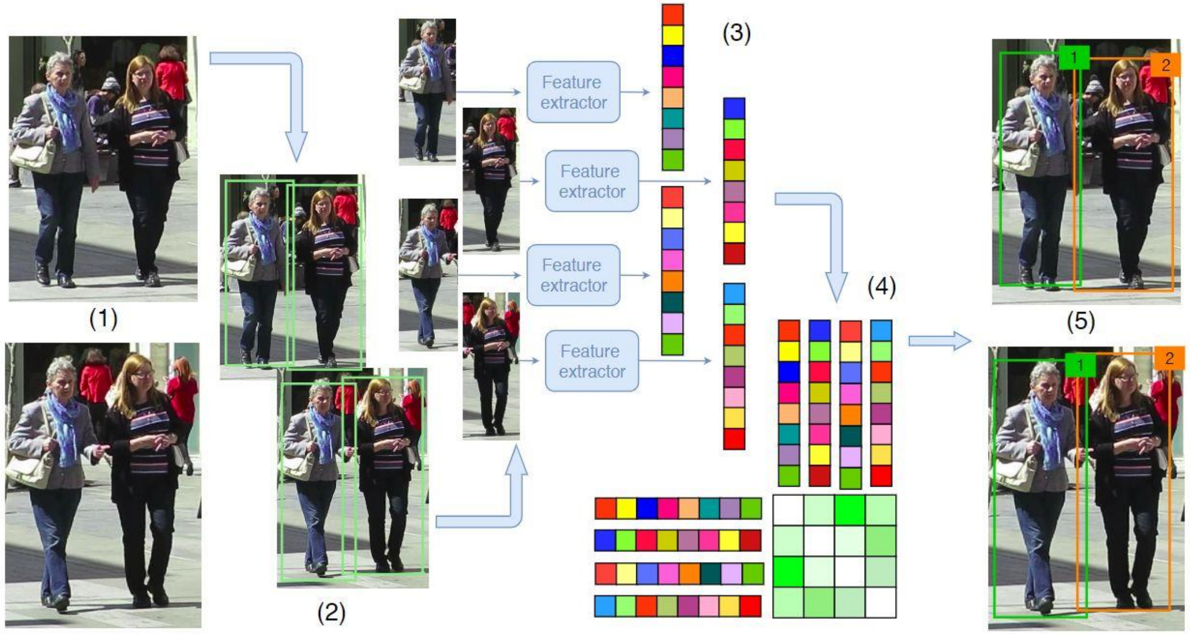
https://www.youtube.com/watch?v=xOGQ1lP7pt8

What is multiple object tracking(MOT)?

- Popular solution:

First, all detections are extracted via YOLO, Faster RCNN, and so on.

Then, an association algorithm is performed to link these detections to different tracks. Usually, the association algorithm considers motion (direction, speed, …) information and appearance information.



What is multiple object tracking(MOT)?

- Popular solution:

  First, all detections are extracted via YOLO, Faster RCNN, and so on.

  Then, an association algorithm is performed to link these detections to different tracks. Usually, the association algorithm considers motion (direction, speed, …) information and appearance information.

- Compared to single object tracking(SOT)

  MOT requires two additional tasks to be solved: (1) determining the number of objects, and (2) maintaining their identities, since in each moment, the algorithm should decide whether a new track appears and an existing track vanishes.

Motivation

- Deep SORT: Simple online and realtime tracking with a deep association metric
- The motivation of deep sort:

Integrate appearance information and motion information (Mahalanobis distance for motion information, cosine distance for appearance informtion)

"SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC" Wojke, et al.(2017).

Motion information

- The state of each target at some point is modelled as:

$$(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$$

where (u, v) is bounding box center position, γ is aspect ratio, h is height, overdot means their respective velocities in image coordinates.

- A standard Kalman filter with constant velocity motion and linear observation model, is used to update the above target state.

Motion information

- The state of each target at some point is modelled as:

$$(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$$

where (u, v) is bounding box center position, γ is aspect ratio, h is height, overdot means their respective velocities in image coordinates.

- A standard Kalman filter with constant velocity motion and linear observation model, is used to update the above target state.
- What is Kalman filter?

Kalman filtering is an algorithm that uses a series of measurements observed over time, and produces estimates of unknown variables.

Motion information

- The state of each target at some point is modelled as:

$$(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$$

where (u, v) is bounding box center position, γ is aspect ratio, h is height, overdot means their respective velocities in image coordinates.

- A standard Kalman filter with constant velocity motion and linear observation model, is used to update the above target state.
- Link detections to existing tracks:

$$d^{(1)}(i,j) = (\boldsymbol{d}_j - \boldsymbol{y}_i)^{\mathrm{T}} \boldsymbol{S}_i^{-1} (\boldsymbol{d}_j - \boldsymbol{y}_i) \quad \text{(cost)}$$

where $\boldsymbol{d}_j$ is the j-th bounding box detection, $\boldsymbol{S}_i$ is the covariance matrix of the Kalman filter prediction, $\boldsymbol{y}_i$ is the Kalman filter prediction bounding box.

The equation calculates the Mahalanobis distance of groudtruth detection and the Kalman filter prediction.

Motion information

- The state of each target at some point is modelled as:

$$(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$$

where (u, v) is bounding box center position, γ is aspect ratio, h is height, overdot means their respective velocities in image coordinates.

- A standard Kalman filter with constant velocity motion and linear observation model, is used to update the above target state.
- Link detections to existing tracks:

$$d^{(1)}(i,j) = (\boldsymbol{d}_j - \boldsymbol{y}_i)^{\mathrm{T}} \boldsymbol{S}_i{}^{-1} (\boldsymbol{d}_j - \boldsymbol{y}_i) \quad \text{(cost)}$$

IOU is another kind of cost on motion information[1]. IOU simply calculates the maximum overlap ratio of any bounding box in the track and the new detection bounding box.

[1] High-Speed tracking-by-detection without using image information." *Advanced Video and Signal Based Surveillance (AVSS), 2017*

Motion information

- The state of each target at some point is modelled as:

$$(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$$

where (u, v) is bounding box center position, γ is aspect ratio, h is height, overdot means their respective velocities in image coordinates.

- A standard Kalman filter with constant velocity motion and linear observation model, is used to update the above target state.
- Link detections to existing tracks:

$$d^{(1)}(i, j) = (\boldsymbol{d}_j - \boldsymbol{y}_i)^{\mathrm{T}} \boldsymbol{S}_i{}^{-1} (\boldsymbol{d}_j - \boldsymbol{y}_i) \quad \text{(cost)}$$

- Then apply a threshold $t^{(1)}$ to $d^{(1)}(i, j)$, to check whether it's feasible to accept this link:

$$b_{i,j}^{(1)} = \mathbb{1}[d^{(1)}(i, j) \leq t^{(1)}] \quad \text{(gate)}$$

Appearance information

- For each bounding box detection $\boldsymbol{d}_j$, we compute an appearance descriptor $\boldsymbol{r}_j$ with $\|\boldsymbol{r}_j\| = 1$ by L2 normalization, where $\boldsymbol{r}_j$ comes from a convolutional neural network(wide residual).
- Keep a gallery $\mathcal{R}_k = \{\boldsymbol{r}_k^{(i)}\}_{k=1}^{L_k}$ of the $L_k = 100$ associated appearance descriptors for each track k , i.e, only keep the last 100 descriptors.
- Distance between the *i*-th track and *j*-th detection in appearance space is the smallest cosine distance the *i*-th track and *j*-th detection that:

$$d^{(2)}(i,j) = \min\{1 - \boldsymbol{r}_j{}^{\mathrm{T}}\boldsymbol{r}_k^{(i)} \mid \boldsymbol{r}_k^{(i)} \in \mathcal{R}_i\}$$

- Then, apply a threshold $t^{(2)}$ to $d^{(2)}(i,j)$, check whether it's feasible to accept this link:

$$b_{i,j}^{(2)} = \mathbb{1}[d^{(2)}(i,j) \leq t^{(2)}]$$

Combine motion & appearance information

- Combine both metrics using a weighted sum:

$$c_{i,j} = \lambda \, d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \qquad (5)$$

This term is interpreted as the cost of associating the *i*-th track and *j*-th detection.

- And check whether motion and appearance are both less than the threshold:

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)}. \qquad (6)$$

This term is interpreted as the gate of associating the *i*-th track and *j*-th detection.

Symbols in pipeline

Listing 1 Matching Cascade

Input: Track indices $\mathcal{T} = \{1, \dots, N\}$, Detection indices $\mathcal{D} = \{1, \dots, M\}$, Maximum age $A_{\max}$

```
Compute cost matrix C = [c_{i,j}] using Eq. 5
Compute gate matrix B = [b_{i,j}] using Eq. 6
Initialize set of matches M ← ∅
Initialize set of unmatched detections U ← D
for n ∈ {1, ..., A_max} do
    Select tracks by age T_n ← {i ∈ T | a_i = n}
    [x_{i,j}] ← min_cost_matching(C, T_n, U)
    M ← M ∪ {(i, j) | b_{i,j} · x_{i,j} > 0}
    U ← U \ {j | Σ_i b_{i,j} · x_{i,j} > 0}
end for
return M, U
```

- $\mathcal{T} = \{1, \dots, N\}$: every track contains all the past detections in that track.
- $\mathcal{D} = \{1, \dots, M\}$: detections in all frames.
- $A_{\max}$: tracks has no new added frame in the past $A_{\max}$ frames are thought dead.
- $\boldsymbol{C} = [c_{i,j}]$: $c_{i,j}$ is the cost of associating the *i*-th track and *j*-th detection.
- $\boldsymbol{B} = [b_{i,j}]$: $b_{i,j}$ is the gate of associating the *i*-th track and *j*-th detection.

Pipeline

Listing 1 Matching Cascade

Input: Track indices $\mathcal{T} = \{1, \ldots, N\}$, Detection indices $\mathcal{D} = \{1, \ldots, M\}$, Maximum age $A_{\max}$

Get detections from another alogrithm(YOLO, faster RCNN, ...)

```
Compute cost matrix C = [c_{i,j}] using Eq. 5
Compute gate matrix B = [b_{i,j}] using Eq. 6
```

Compute the association cost matrix and the matrix of admissible associations.

```
Initialize set of matches M ← ∅
Initialize set of unmatched detections U ← D
for n ∈ {1, ..., A_max} do
```

n is in how many frames a tracker has not been updated.

```
    Select tracks by age T_n ← {i ∈ T | a_i = n}
```

select the subset of tracks $\mathcal{T}_n$ that have not been associated with a detection in the last n frames

```
    [x_{i,j}] ← min_cost_matching(C, T_n, U)
```

Solve the linear assignment between tracks in Tn and unmatched detections U

```
    M ← M ∪ {(i, j) | b_{i,j} · x_{i,j} > 0}
    U ← U \ {j | Σ_i b_{i,j} · x_{i,j} > 0}
end for
return M, U
```

Update the set of matches and unmatched detections