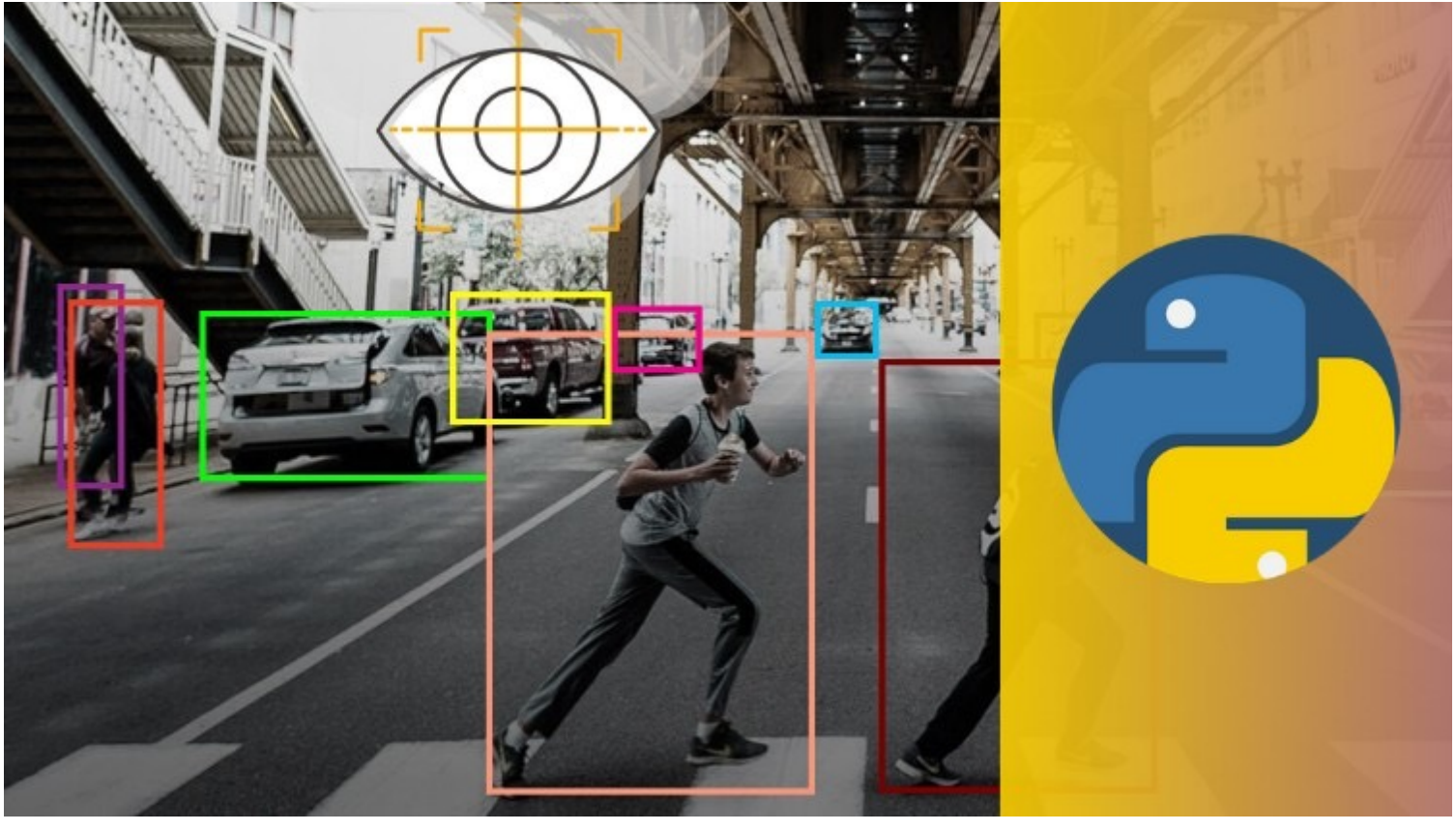
OBJECT TRACKING USING PYTHON AND OPENCV

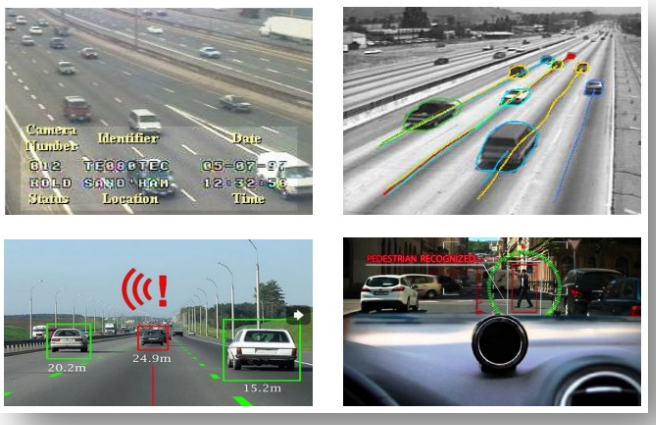


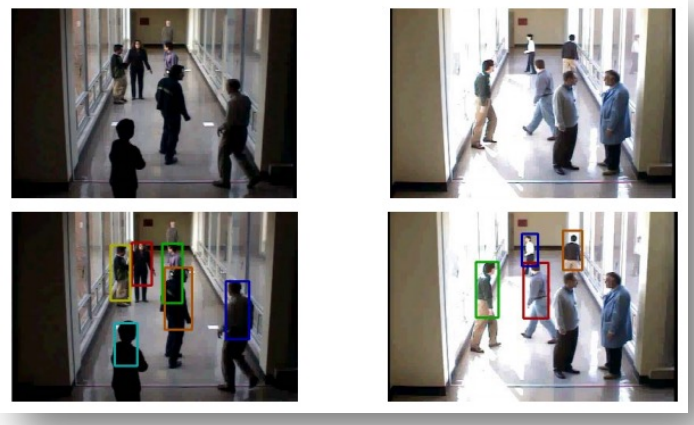
EXPERT ACADEMY

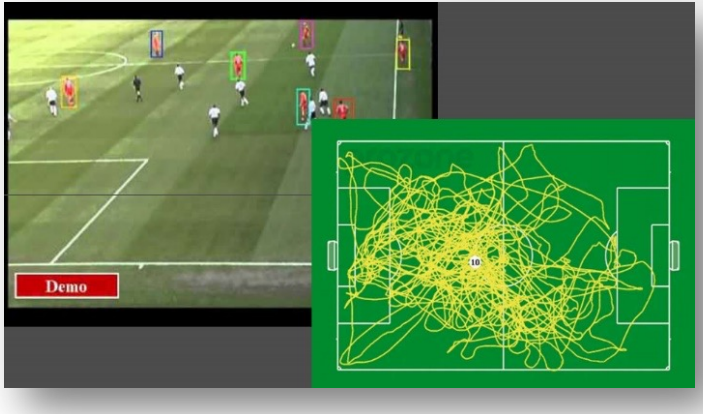
ALGORITHMS

- BOOSTING
- MIL
- KCF
- TLD
- MedianFlow
- MOSSE
- CSRT
- Goturn
- Meanshift
- CAMshift
- Opticalflow (sparce and dense)

APPLICATIONS

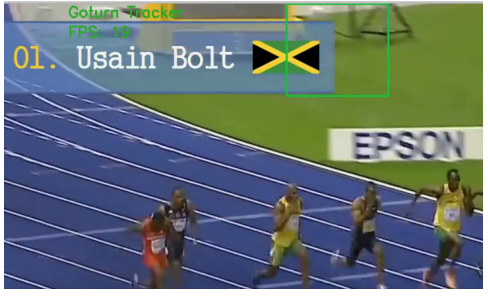




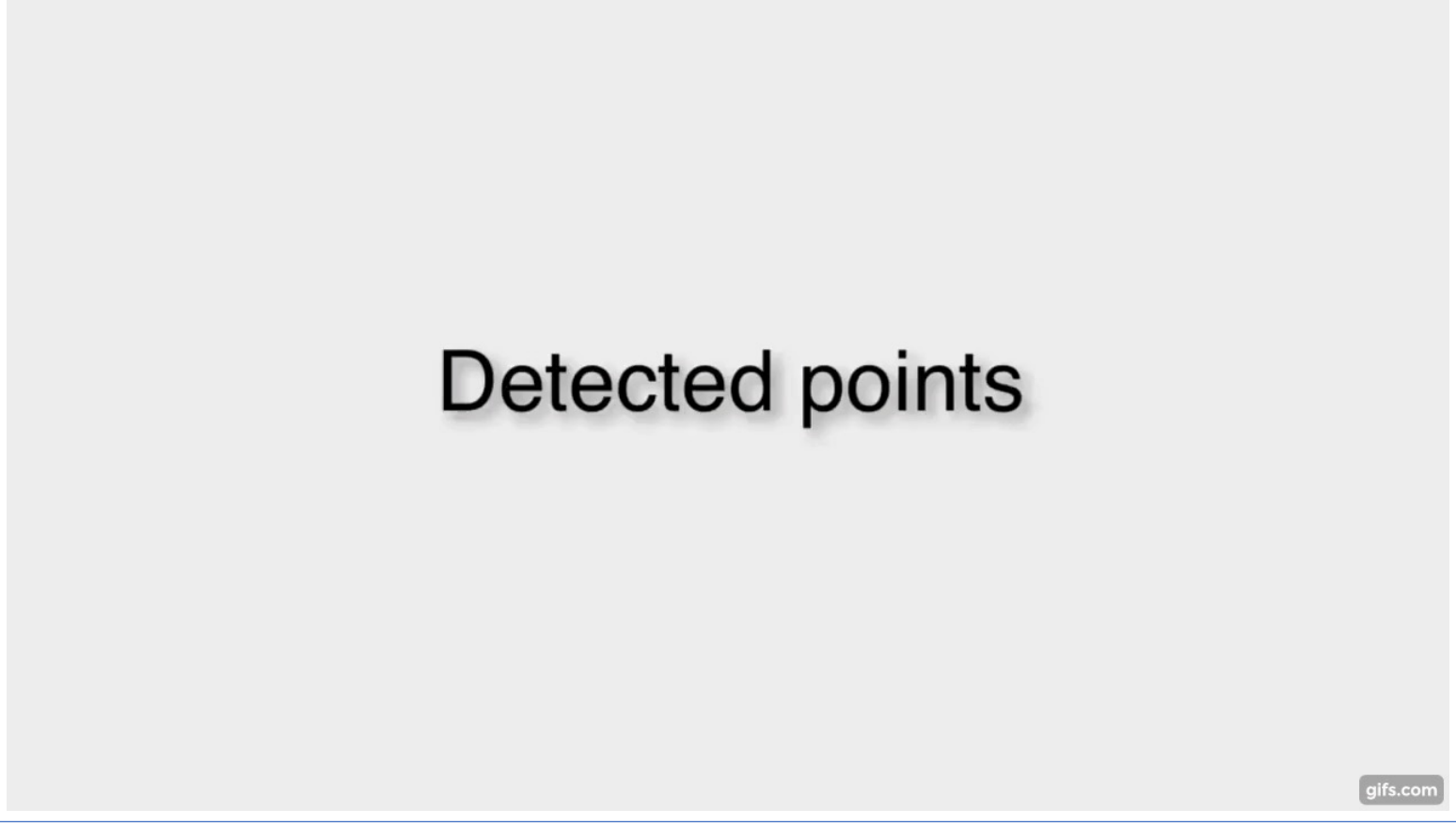


EXPERT ACADEMY

OBJECT TRACKING vs. OBJECT DETECTION



OBJECT TRACKING vs. OBJECT DETECTION



OBJECT TRACKING ALGORITHMS

- Motion model
  - Location + speed in previous frames
  - Predicts the approximate location of the object
- Appearance model
  - Appearance of the object (shape)
  - Search a nearby position to predict the location

- The motion model predicts the approximate location of the object. The appearance model adjusts the location to provide a more accurate location based on the appearance of the object

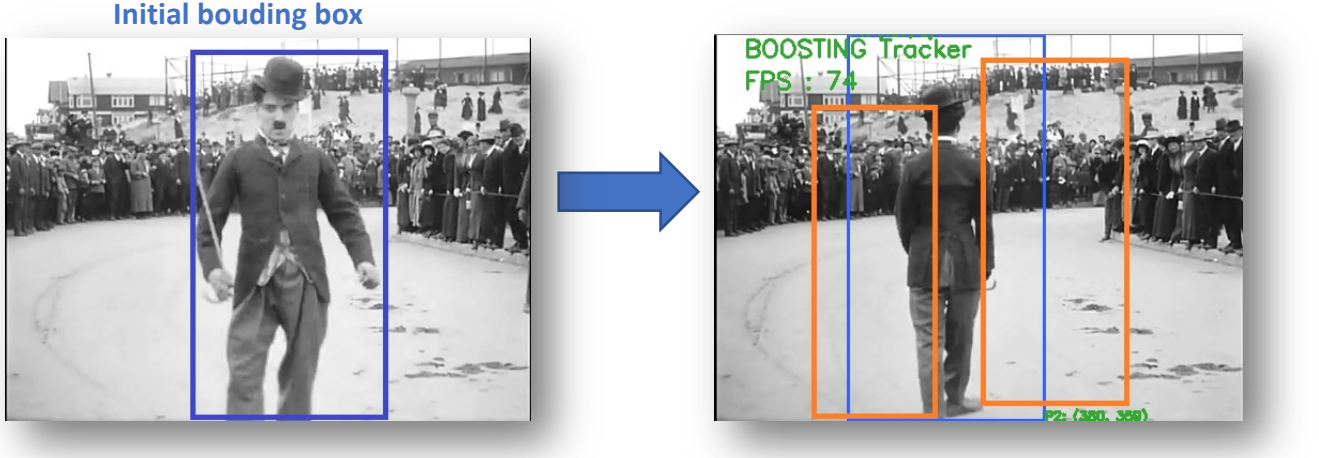
OBJECT TRACKING ALGORITHMS

- Classify a rectangular region as an object or background
- It receives an “image patch” as input and returns a probability (1 – object, 0 = background)
- Online learning vs. Offline learning

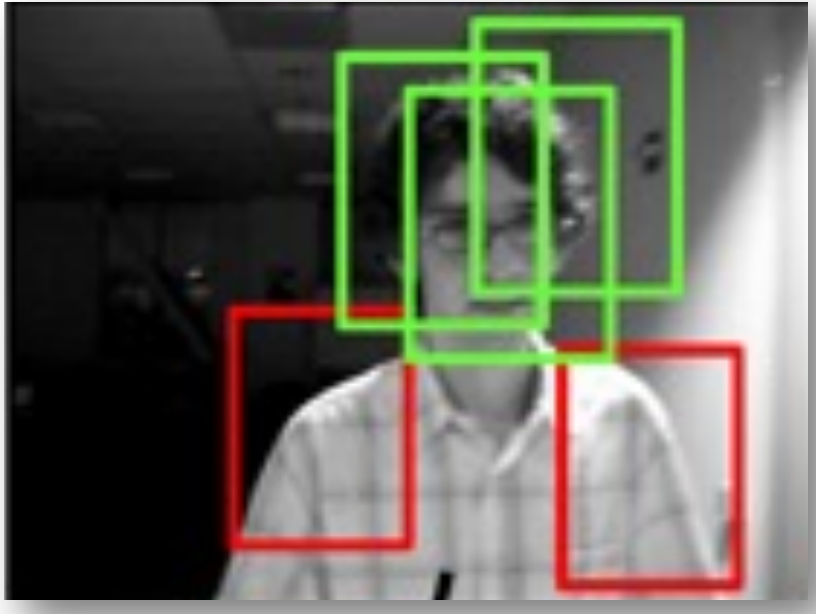
- Source of explanations: Satya Mallick (learnopencv.com) and Adrian Rosebrock (pyimagesearch.com): https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/

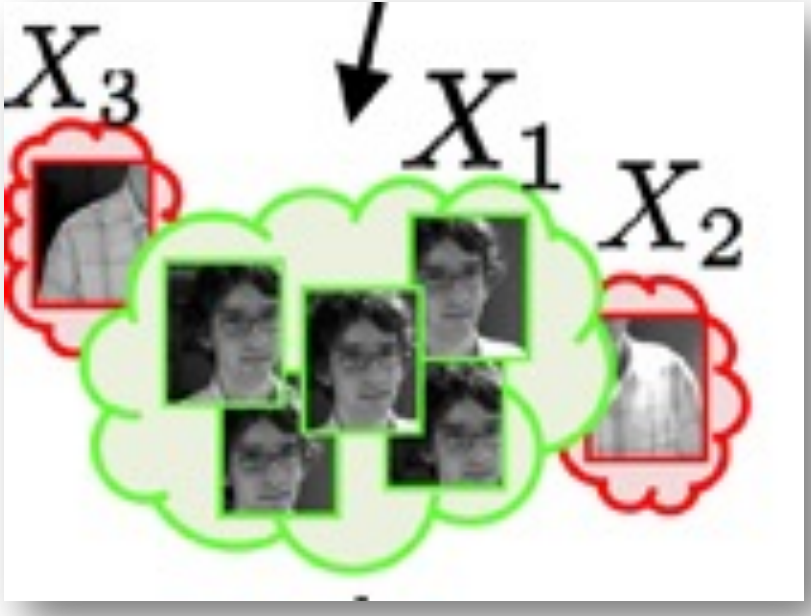
BOOSTING

- It uses Haarcascade classifier (AdaBoost algorithm)
- On-line training
- The initial bounding box is the positive example. The rest of the image is treated as the background (negative examples)

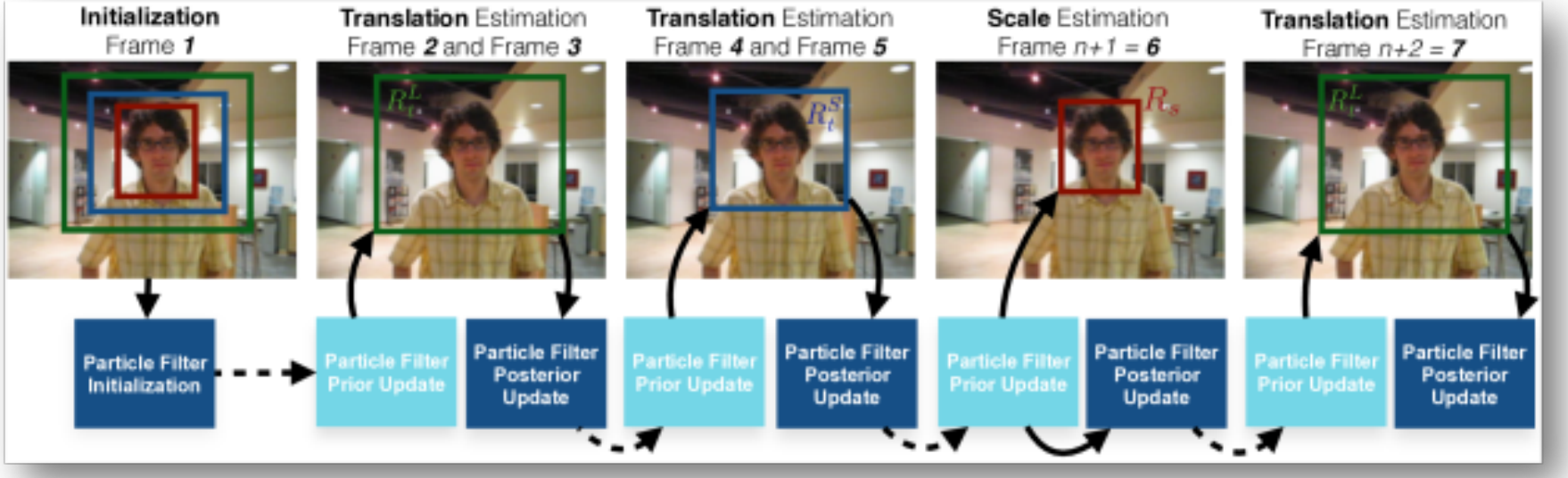


MIL (MULTIPLE INSTANCE LEARNING)



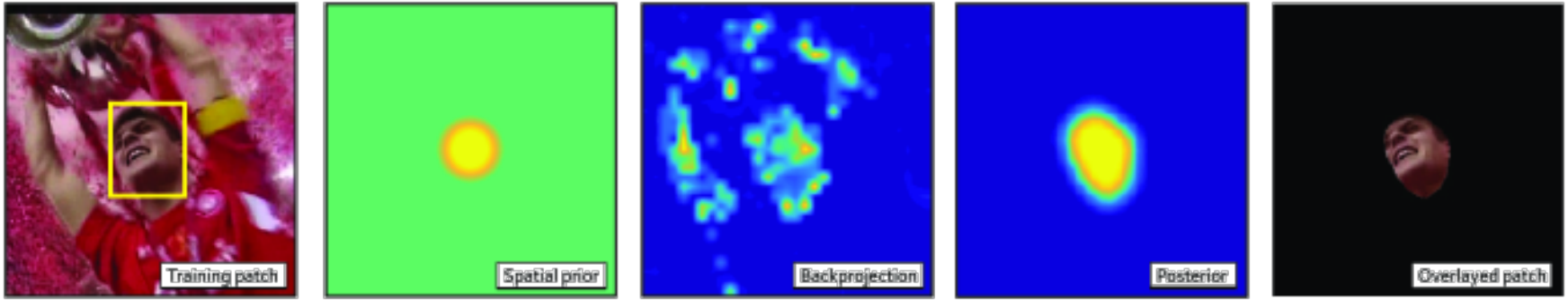


KCF (KERNAL CORRELATION FILTERS)



Source: https://www.groundai.com/project/enkcf-ensemble-of-kernelized-correlation-filters-for-high-speed-object-tracking/1

CSRT (DISCRIMINATIVE CORRELATION FILTER WITH CHANNEL AND SPATIAL RELIABILITY)

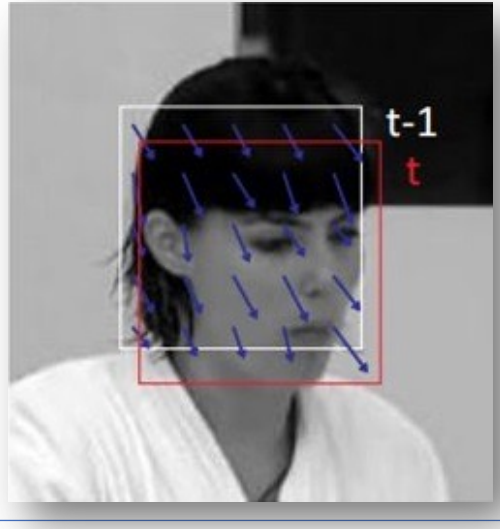


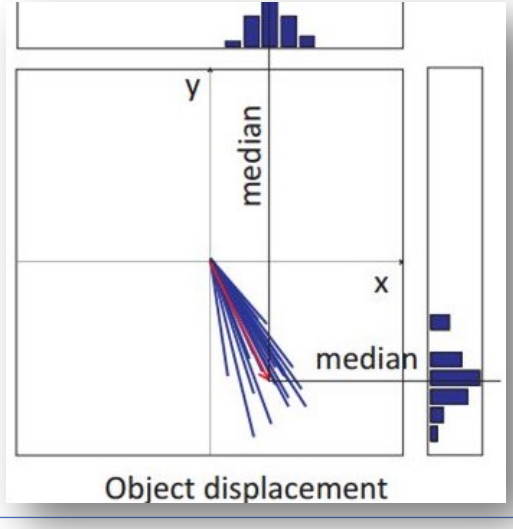
1. From left to right: training patch with the bounding box of the object
2. HOG to extract useful information of the image
3. Application of Random Markov Test to generate probabilities
4. Training patch masked using the confidence map

Source: https://www.arxiv-vanity.com/papers/1611.08461/

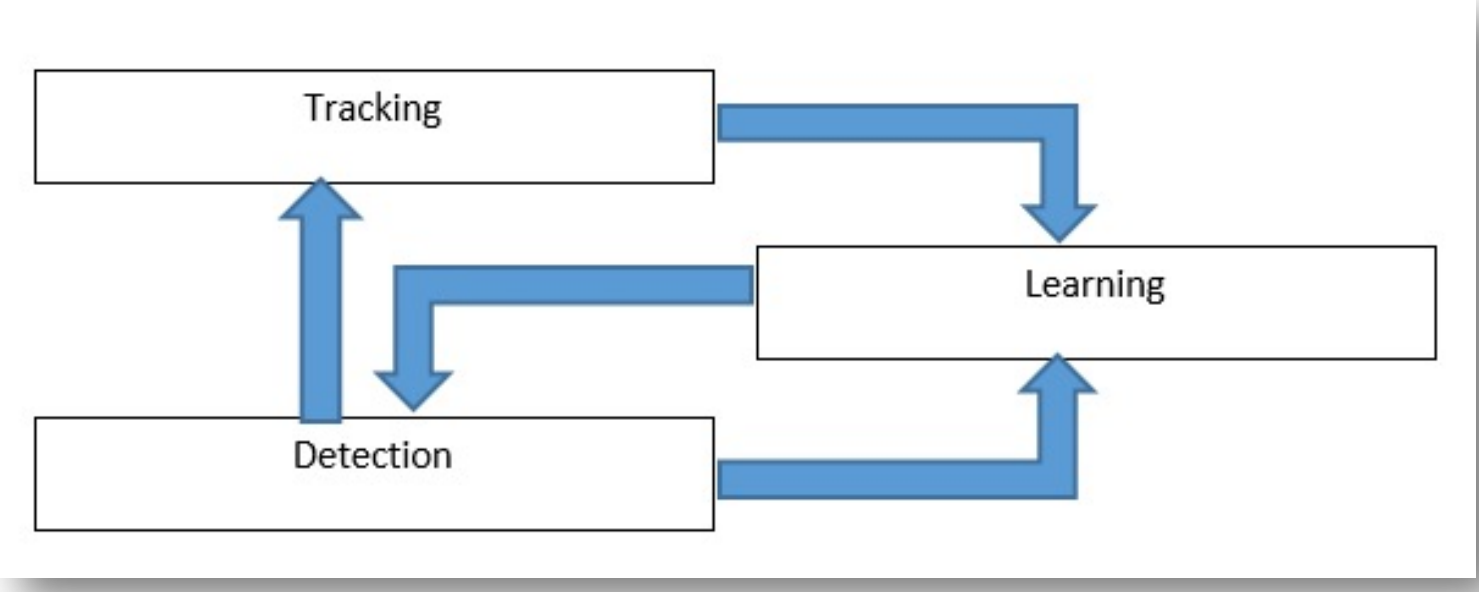
MEDIANFLOW

- Not suitable for fast moving objects or objects that change their appearance quickly
- Tracks the object in forward and backward direction and measures the diferences between these two trajectories
- Allows the detection of tracking failures



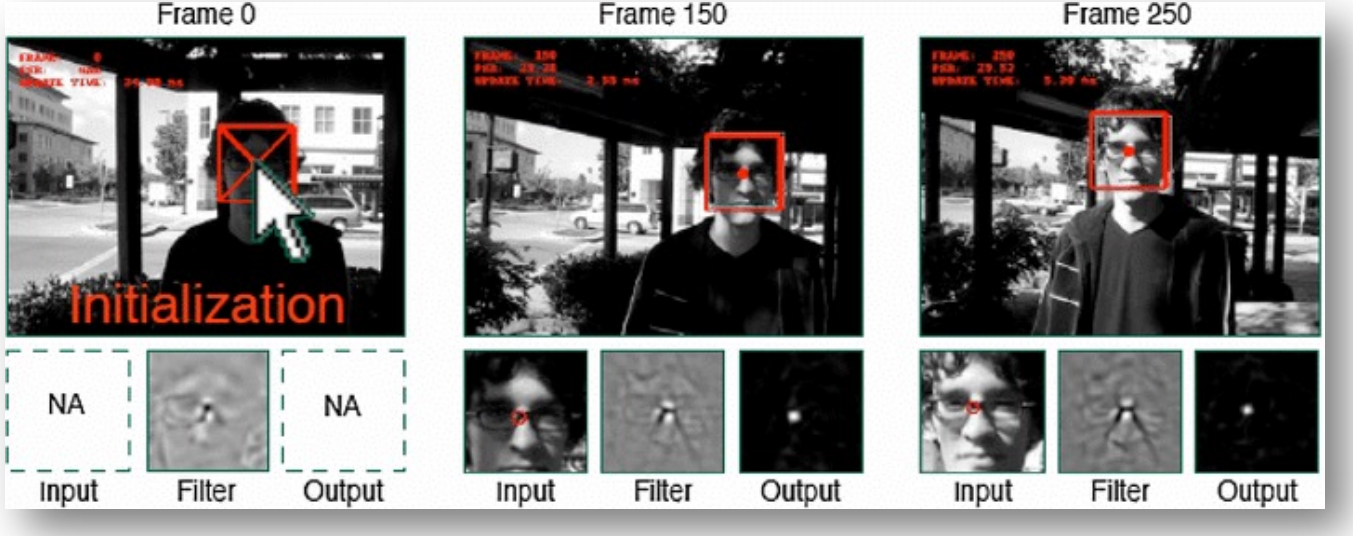


TLD (TRACKING LEARNING DETECTION)



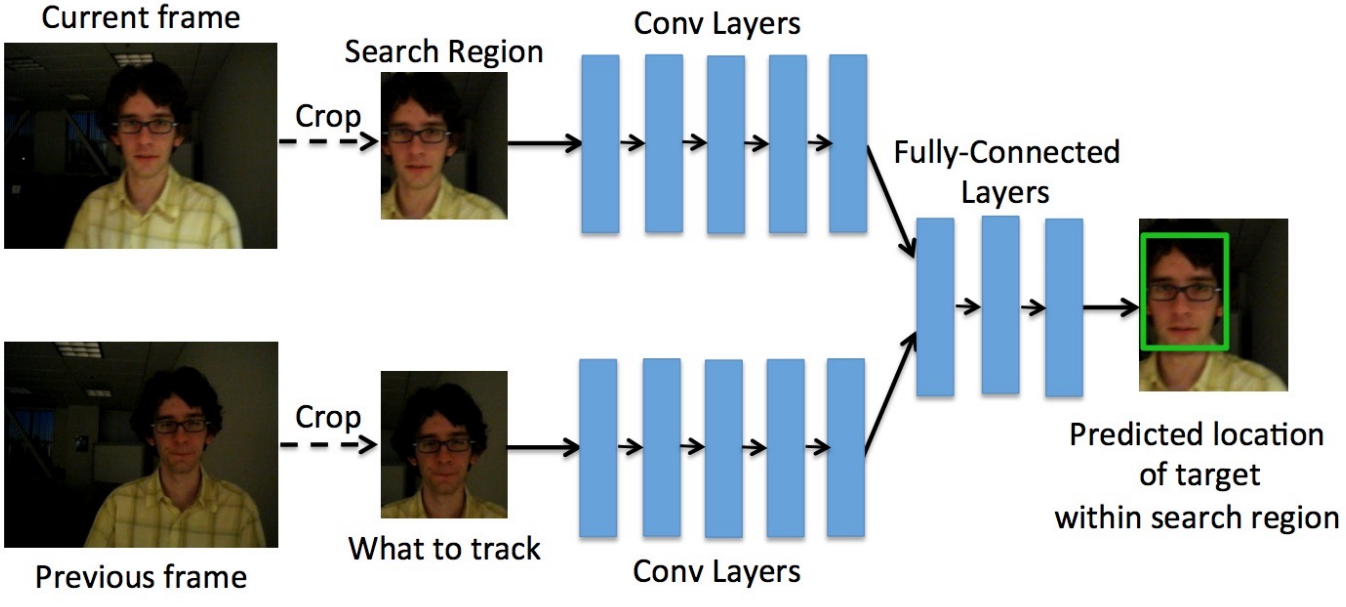
MOSSE (MINIMUM OUTPUT SUM OF SQUARED ERROR)

- Robust to lighting variations, scale and pose
- Correlation filters



GOTURN (GENERIC OBJECT TRACKING USING REGRESSION NETWORKS)

- Off-line learning

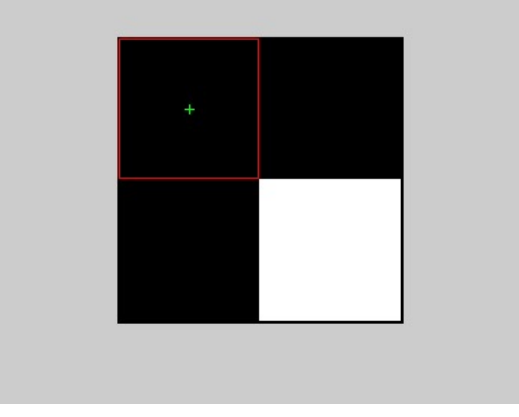


TRACKING ALGORITHMS

- CSRT: good accuracy, but it is slower than others
- KCF: not so good accuracy, but it is fast
- MOSSE: it is the fastest

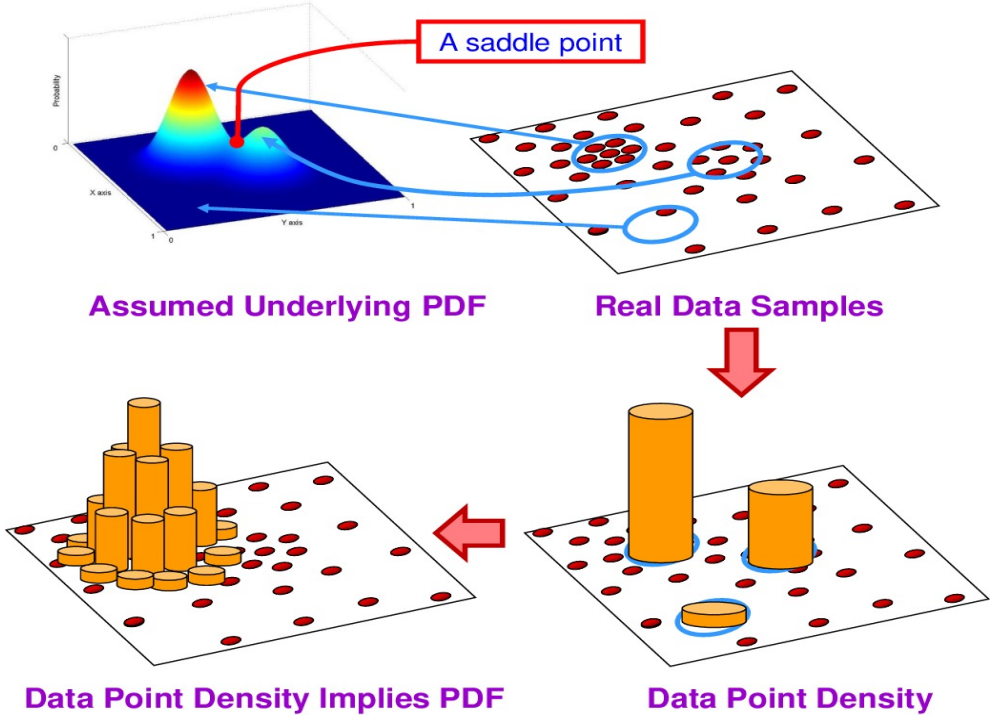
MEANSHIFT

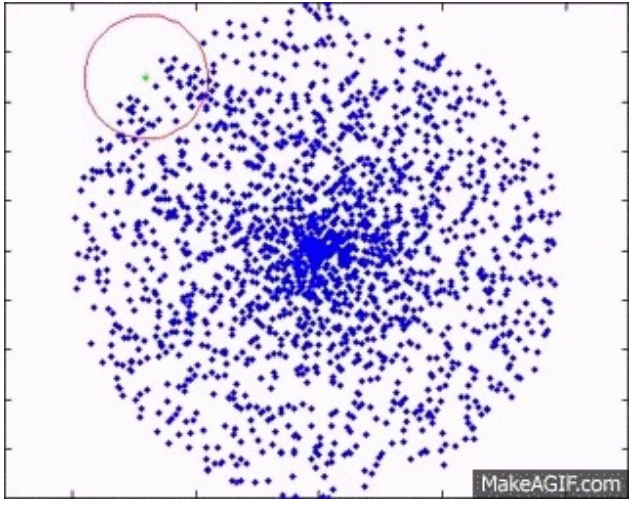
- It uses the **feature space analysis** to find the maxima of a **density function**
- Determines the weight of nearby points based on the average estimate (based on colors)



Source: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_meanshift/py_meanshift.html

MEANSHIFT


A saddle point
Probability
X axis
Y axis
Assumed Underlying PDF
Real Data Samples
Data Point Density Implies PDF
Data Point Density

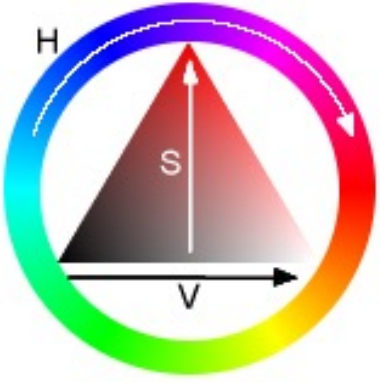

MakeAGIF.com

CAMSHIFT (CONTINUOSLY ADAPTIVE MEANSHIFT)

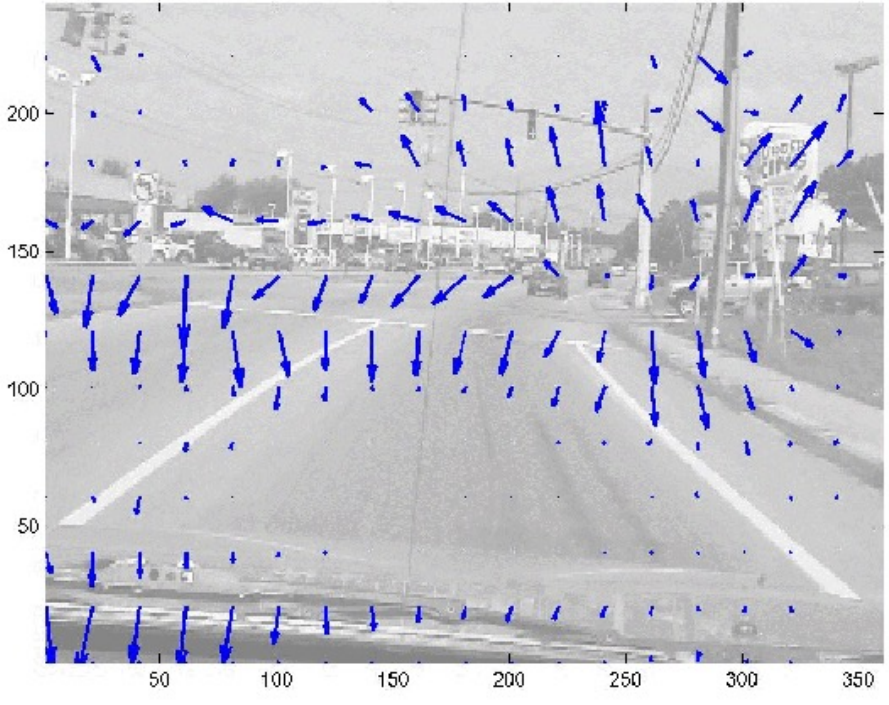
Source: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_meanshift/py_meanshift.html

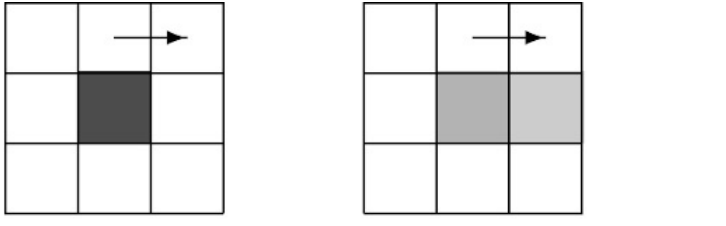
OPTICALFLOW

- Direction and speed of pixels
- Hue (H) is used to indicate the direction of the pixels (where each pixel is moving)
- Saturation (S) is used to indicate the speed of the pixels



OPTICALFLOW – SPARSE





(a) I(x,y,t) (b) I(x+v,y+v,t+1)

OPTICALFLOW – SPARSE

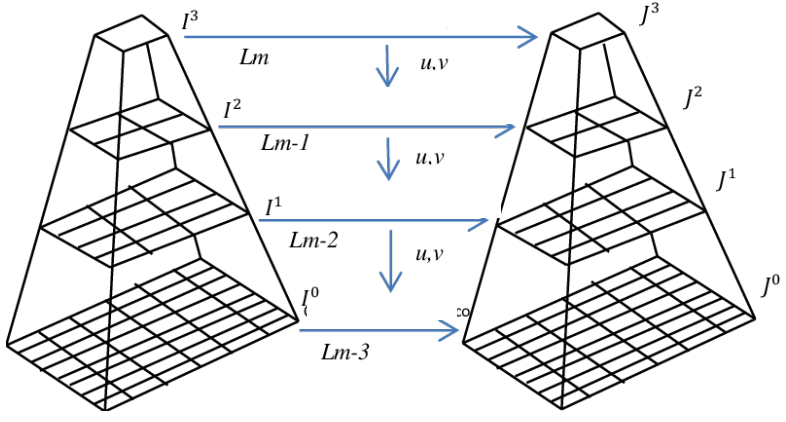
- Initially, the algorithm detects corners and edges to determine the orientation of the vectors using Harris Corner Detector algorithm

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

$$R = \det M - k(\operatorname{trace} M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$



- Shi and Tomasi modified the Harris Corner Detector equation, so the R-score is calculated

$$R = \min(\lambda_1, \lambda_2)$$

- KLT method (Kanade, Lucas and Tomasi): uses pyramids, it analyses smaller images of the same frame

OPTICALFLOW – DENSE (FANERBÄCK)