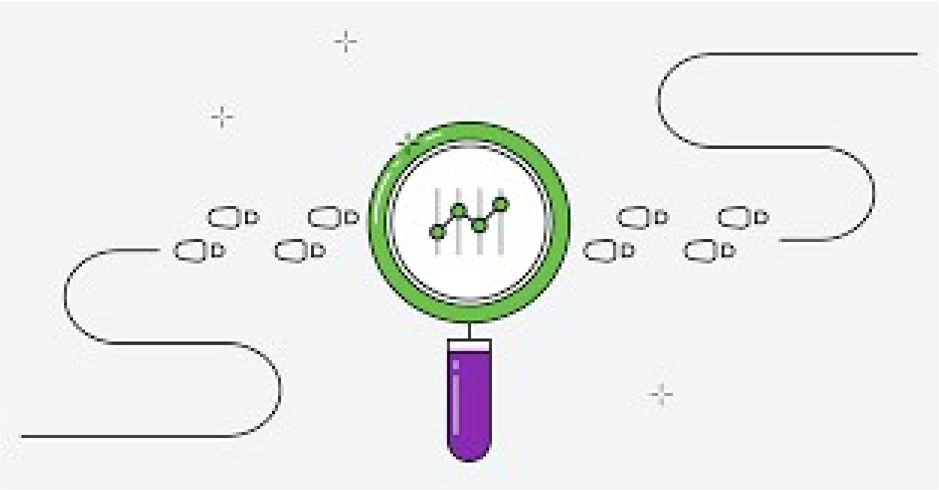
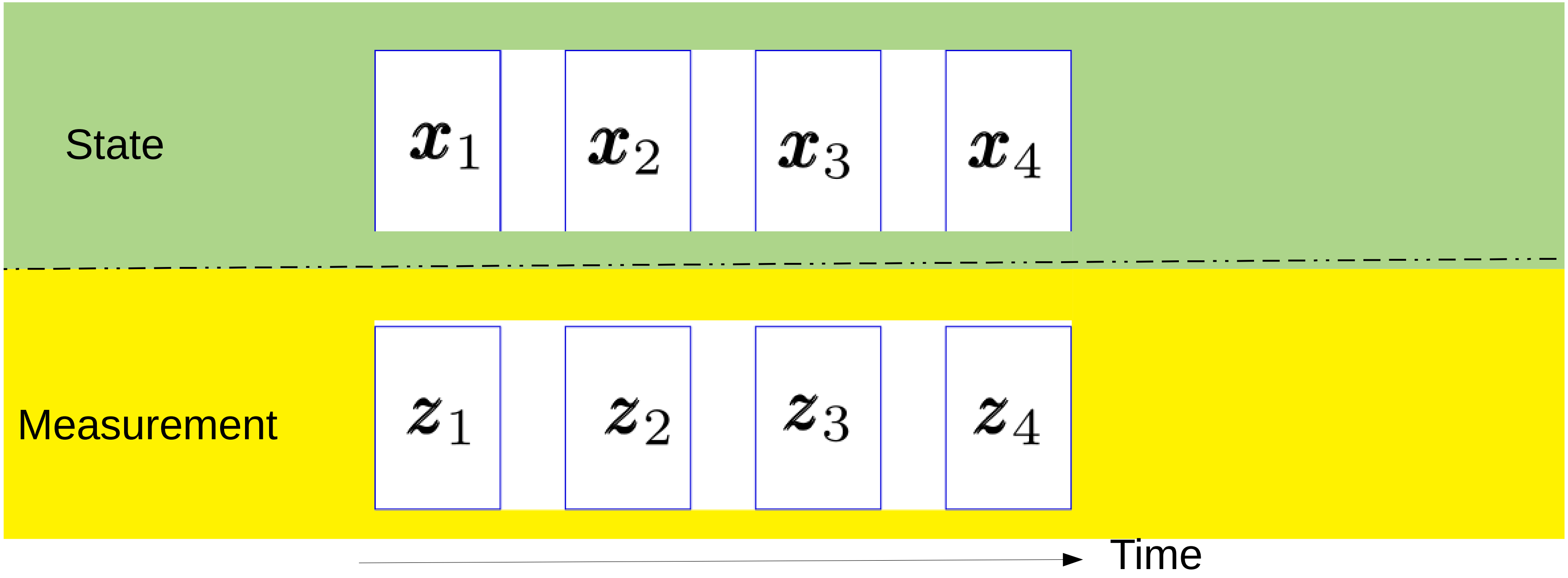
Object Tracking with Deep Learning

Narada Warakagoda



What is Tracking?

- Process of successively determining the state of an object or objects based on noisy measurements.
  - State: position, velocity, pose etc.
  - Measurements: Radar, Sonar, Camera etc.



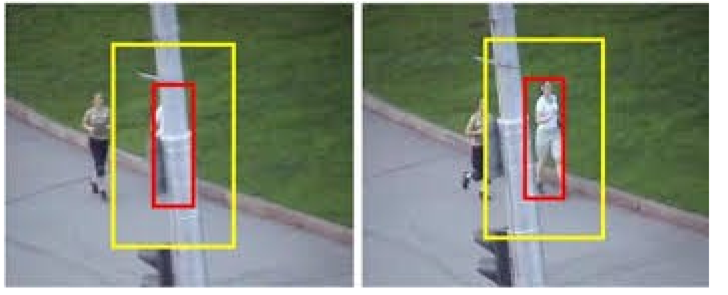
Visual Tracking

- The process of locating dynamic object(s) in successive frames of a video
  - Special case of general object tracking
  - Information exploited
    - Appearance of objects
    - History (motion) of objects

https://neurohive.io/en/datasets/new-datasets-for-object-tracking/

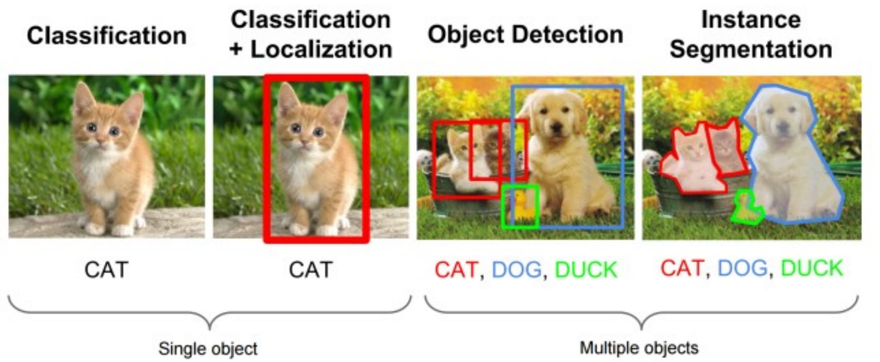
Challenges of Visual Tracking

- Occlusion
- Identity switches
- Motion blur
- Viewpoint variation
- Scale change
- Illumination variation
- Background clutter
- Low resolution



Classification of Tracking Algorithms

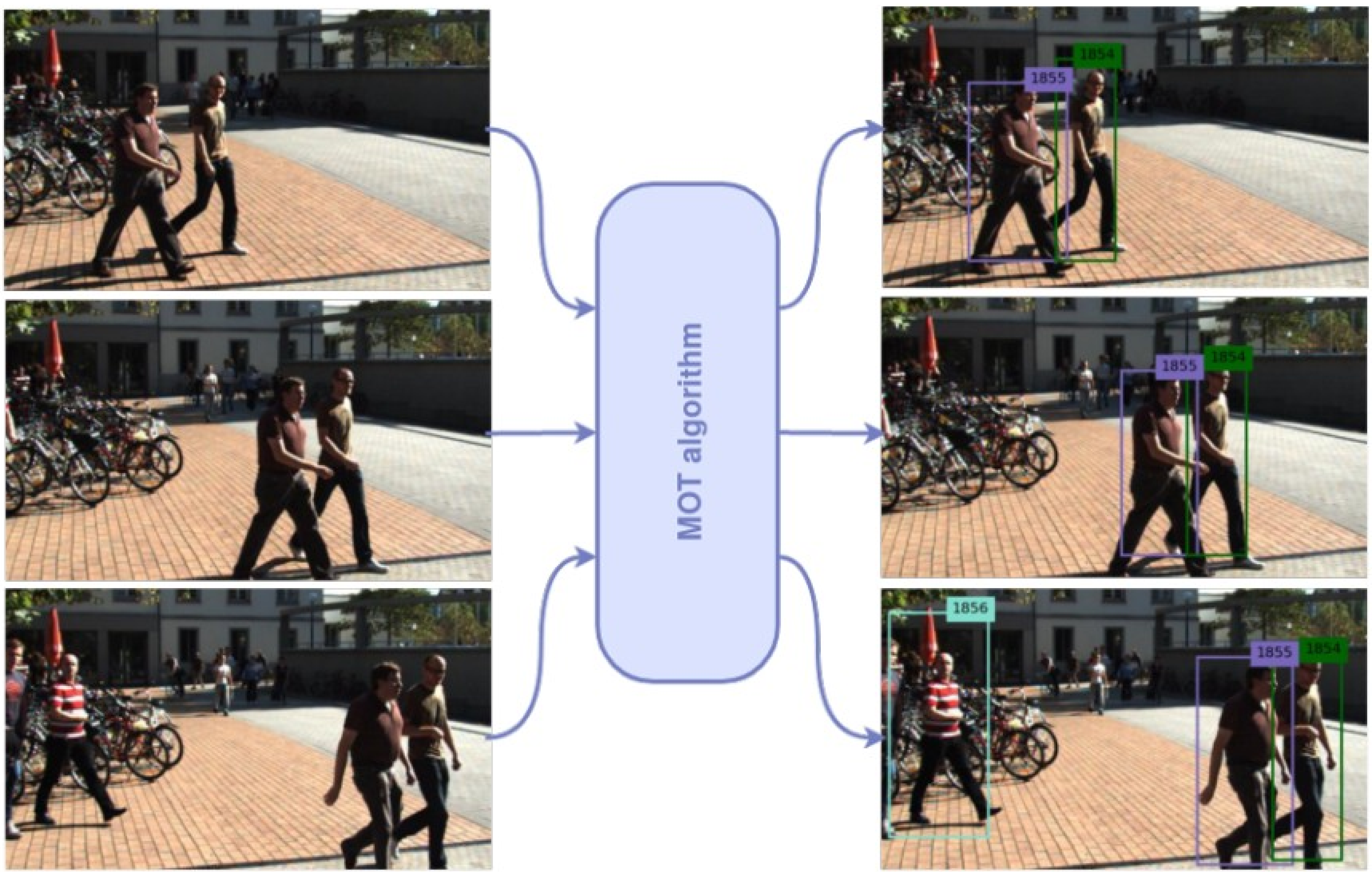
- Detection based vs detection free
  - Detection vs localization of objects



- Single object (target) vs multiple object (target)
  - How many objects are of interest?
- On-line vs off-line (bidirectional)
  - Can we use future frames?
- On-line training vs off-line training
  - Can the system adapt to data?

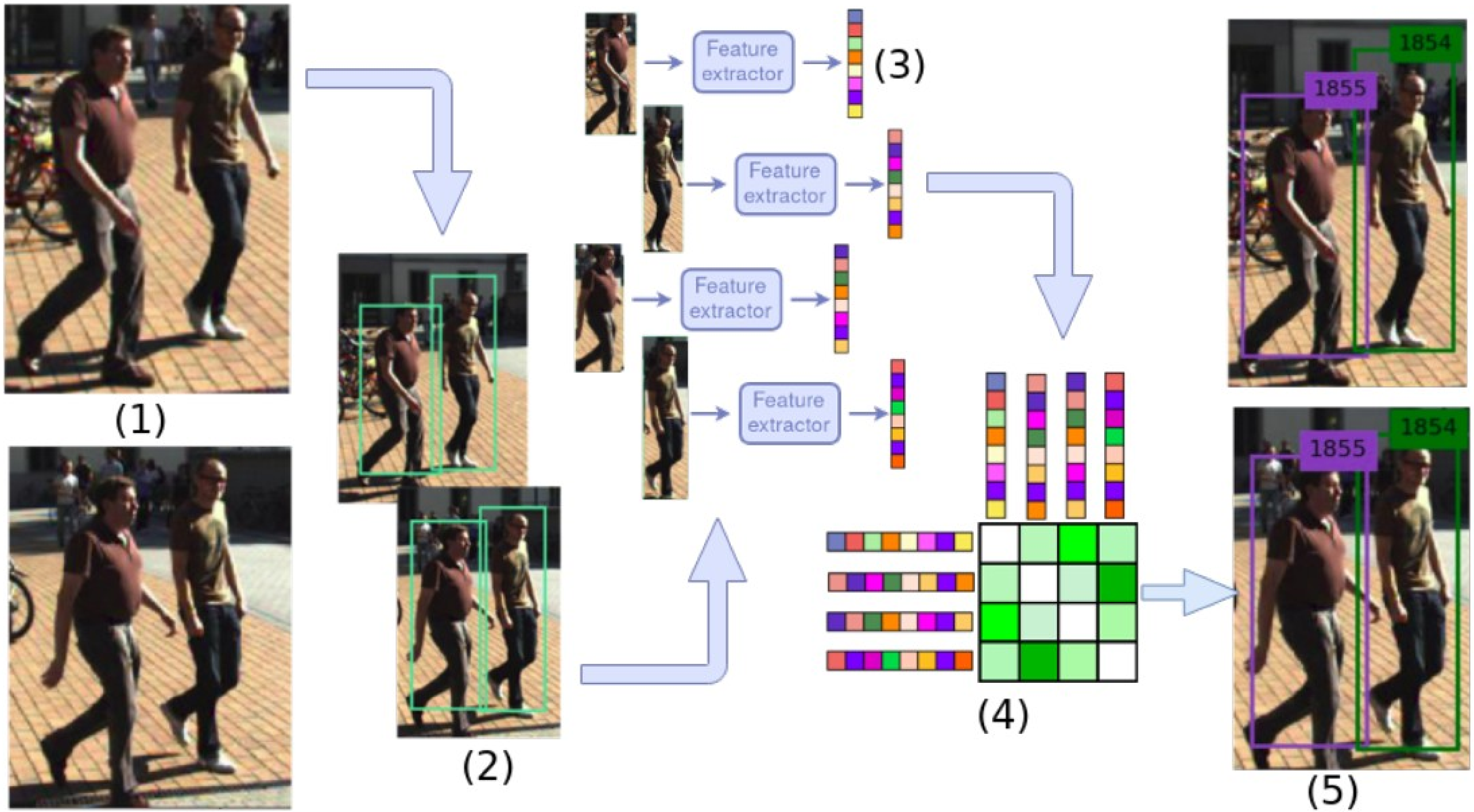
https://cv-tricks.com/object-tracking/quick-guide-mdnet-goturn-rolo/

Multi-Object Tracking (MOT)



Deep learning in video multi-object tracking, a survey, Gioele Ciaparrone et. Al 2019

MOT Workflow



1) Video frames
2) Detection
3) Feature extraction
4) Affinity computation
5) Association

Deep learning can be employed in all these steps

Deep learning in video multi-object tracking, a survey, Gioele Ciaparrone et. Al 2019

Deep Learning for Detection

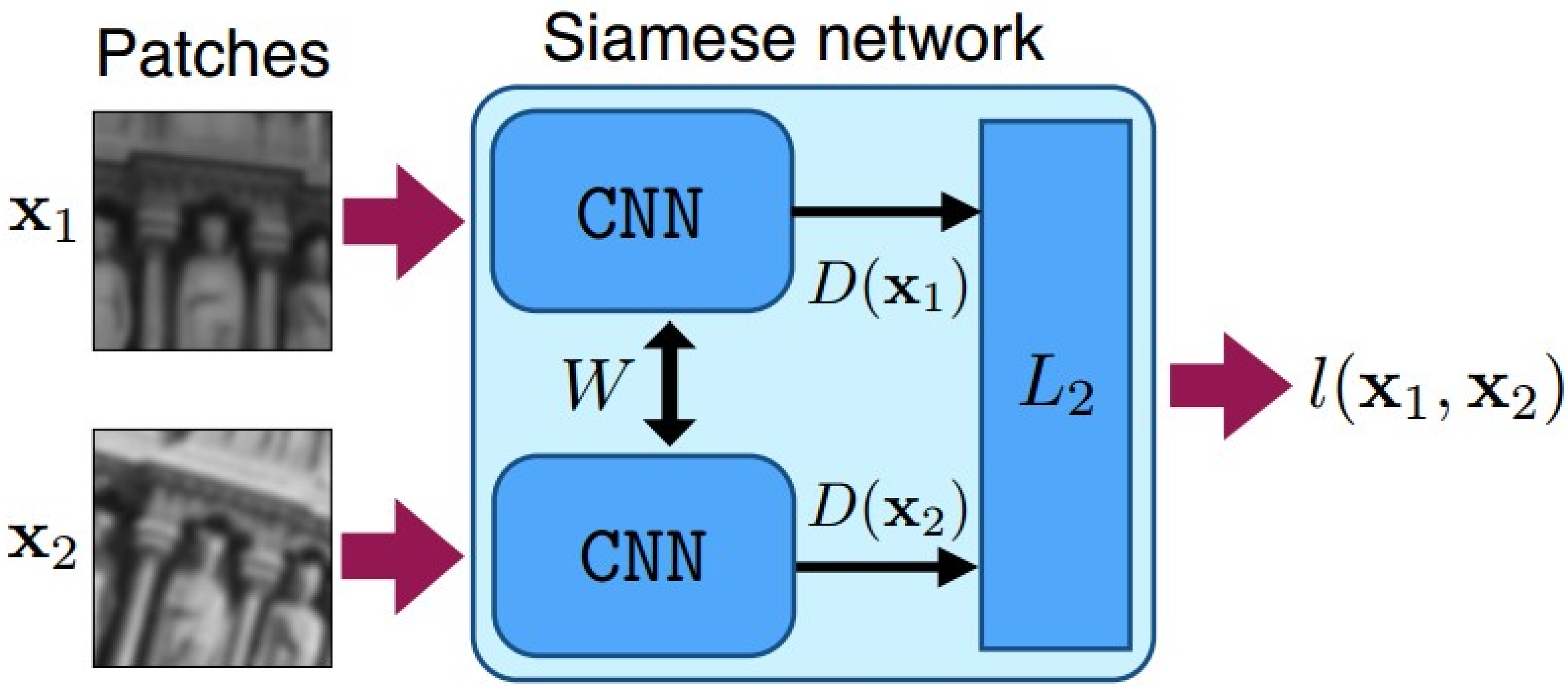
- Several well-known systems
  - Faster R-CNN
  - Single Shot Detector (SSD)
  - You Only Look Once (YOLO)
  - Detection Transformer (DETR)

- We do not discuss detection systems

Deep Learning for Feature Extraction

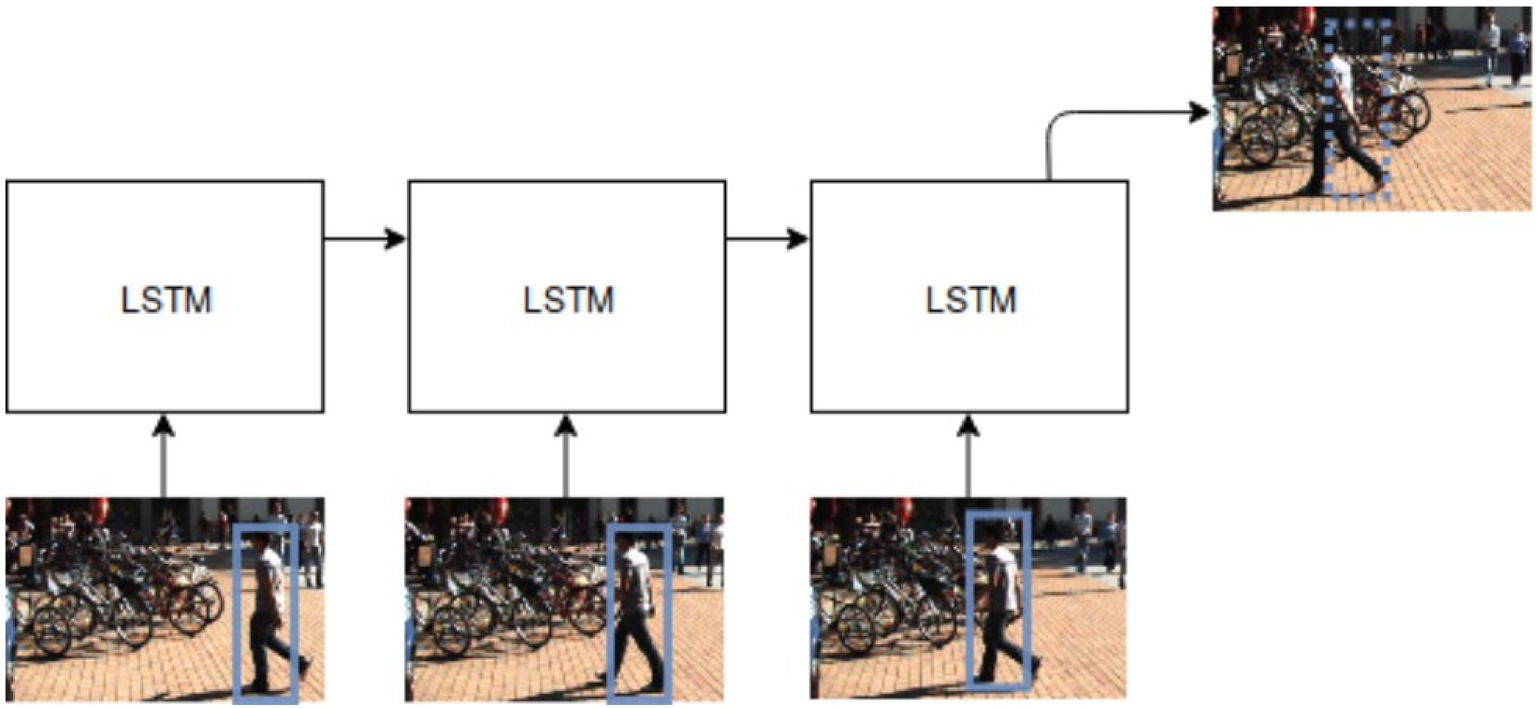
- Mainly two types of features
  - Visual appearance
  - Motion
- Visual feature extraction
  - Auto-encoders
  - CNN
  - **Siamese Networks**
- Motion feature extraction
  - **RNN/LSTM**
  - **CNN + Correlation filter**
  - Reinforcement Learning

Visual Feature Extraction
Siamese Networks



- In training minimize distance l between image patch pairs
- In inference CNN output gives features

Motion Feature Extraction
LSTM Networks



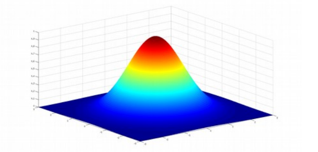
Deep learning in video multi-object tracking, a survey, Gioele Ciaparrone et. Al 2019

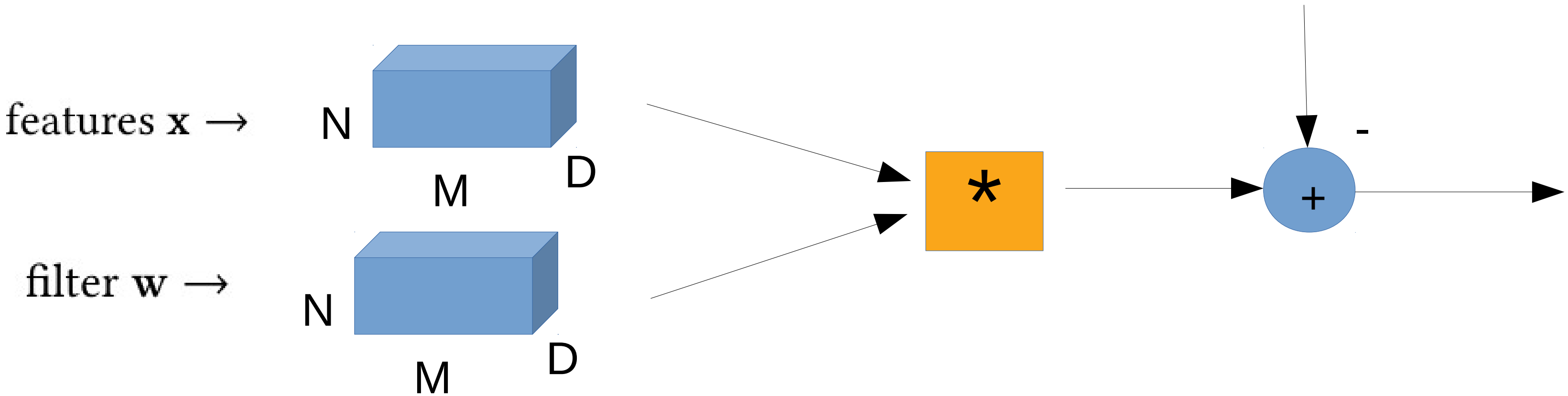
- LSTM network predicts the bounding box for the next frame based on the history
- Predicted bounding box can be used (as a feature) in computing affinity with other objects

Motion Feature Extraction
CNN + Correlation filter

- CNN features are good for appearance representation, not as good for location representation
- Use a correlation filter to estimate the location.
- Correlation filter is a linear filter trained to maximize response at the target location.

$$y(m,n) = e^{-\frac{(m-M/2)^2+(n-N/2)}{2\sigma^2}}$$



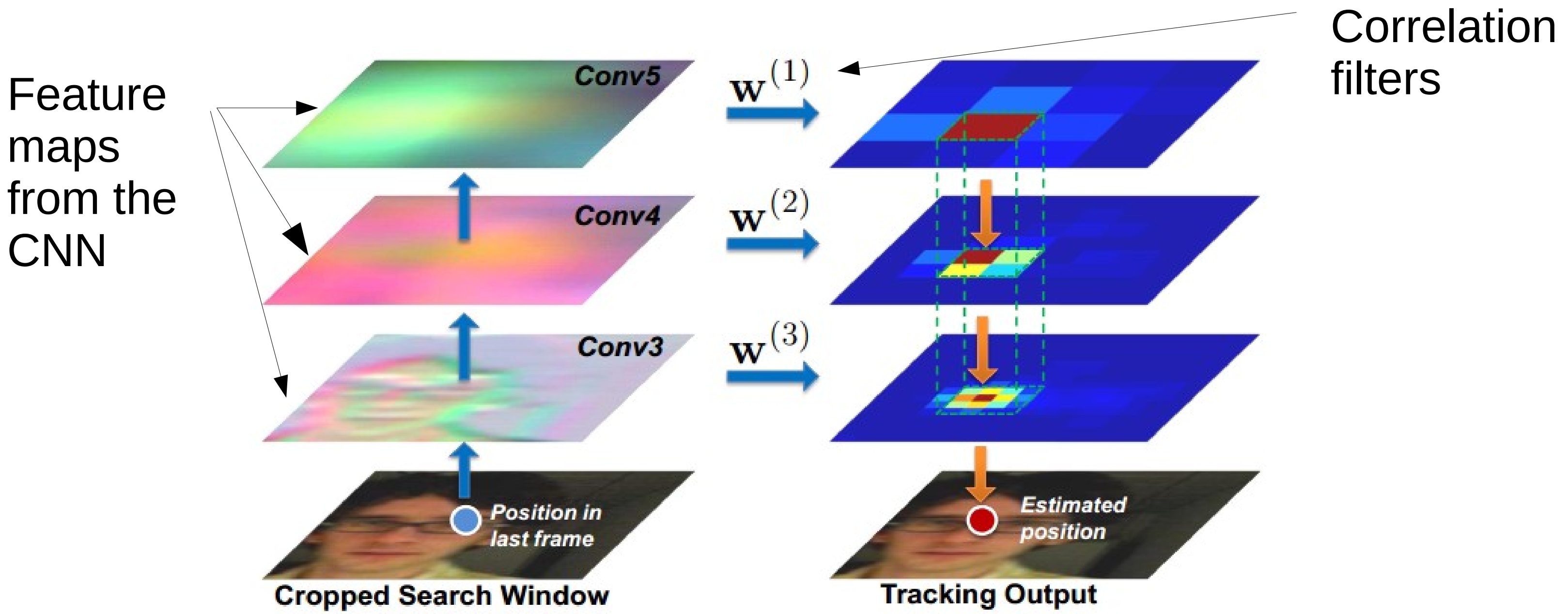


$$\mathbf{w}^{\star} = \arg\min_{\mathbf{w}} ||\mathbf{w} * \mathbf{x} - \mathbf{y}||^2 + \lambda ||\mathbf{w}||^2$$

Closed form solution available

Hierarchical Convolutional Features for Visual Tracking Chao Ma et. al, 2015

Motion Feature Extraction CNN + Correlation filter

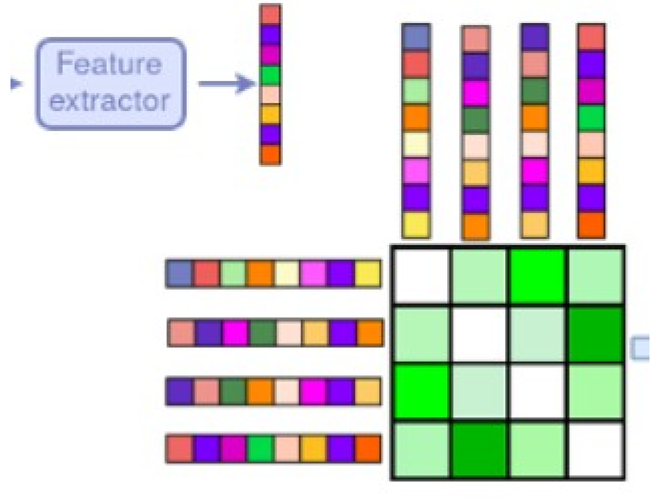


- Pre-trained CNN is used for generating feature maps at different layers
- For each frame of the video
  - Apply correlation filters and estimate the object location
  - Move to the estimated location
  - Update correlation filter parameters (use the closed form)
- Object location = motion feature

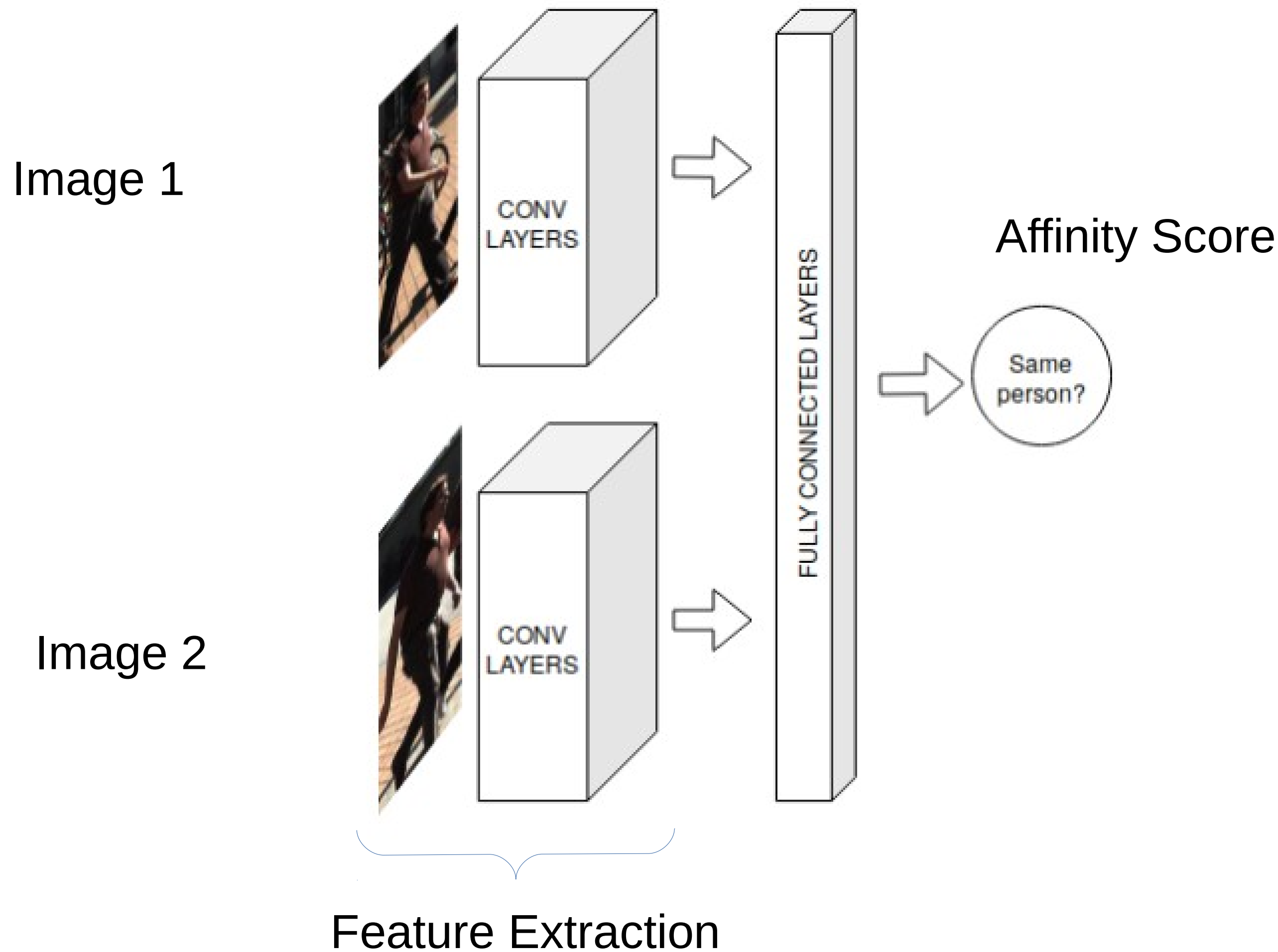
Hierarchical Convolutional Features for Visual Tracking Chao Ma et. al, 2015

Deep Learning for Affinity Computation

- Siamese CNNs
- Siamese LSTMs
- Affinity computation is often merged with feature extraction.

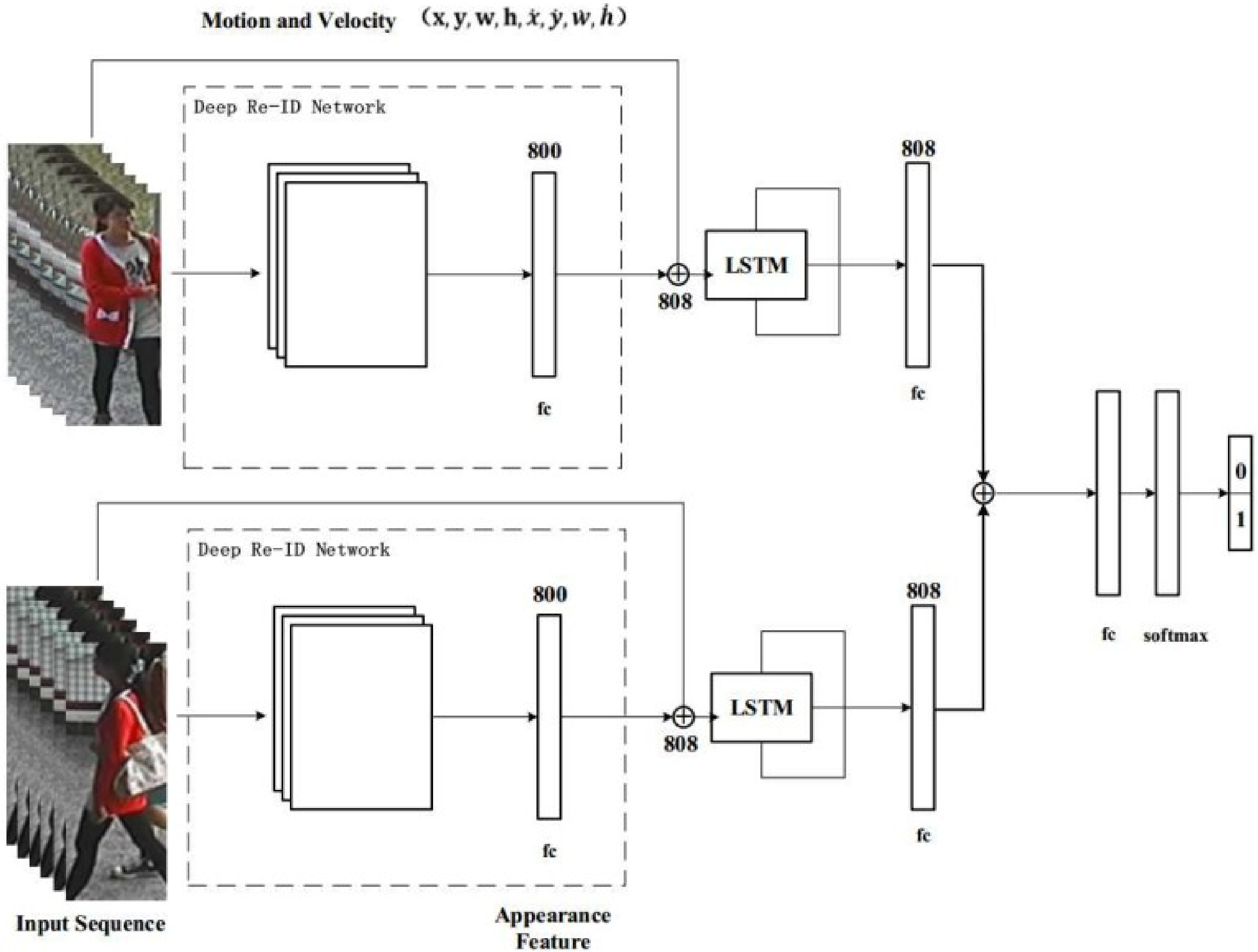


Siamese CNNs



Deep learning in video multi-object tracking, a survey, Gioele Ciaparrone et. Al 2019

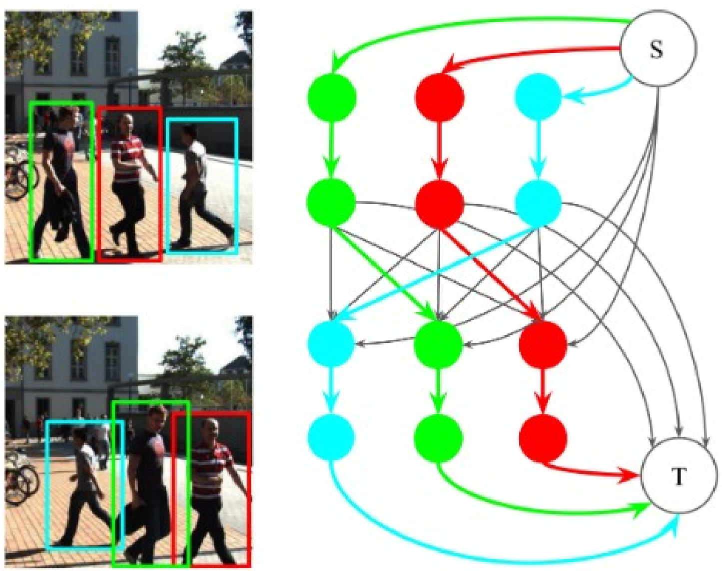
Siamese LSTMs



An Online and Flexible Multi-Object Tracking Framework using Long Short-Term Memory, Xingyu Wan et.al , 2018

Deep Learning for Association

- Can formulate association as a graph partition problem
  - Classical approach: Solve the graph partition problem by solving a constrained optimization problem



$\min_y \sum_{(i,j)\in E} c_{i,j} y_{i,j}$ subject to flow constraints

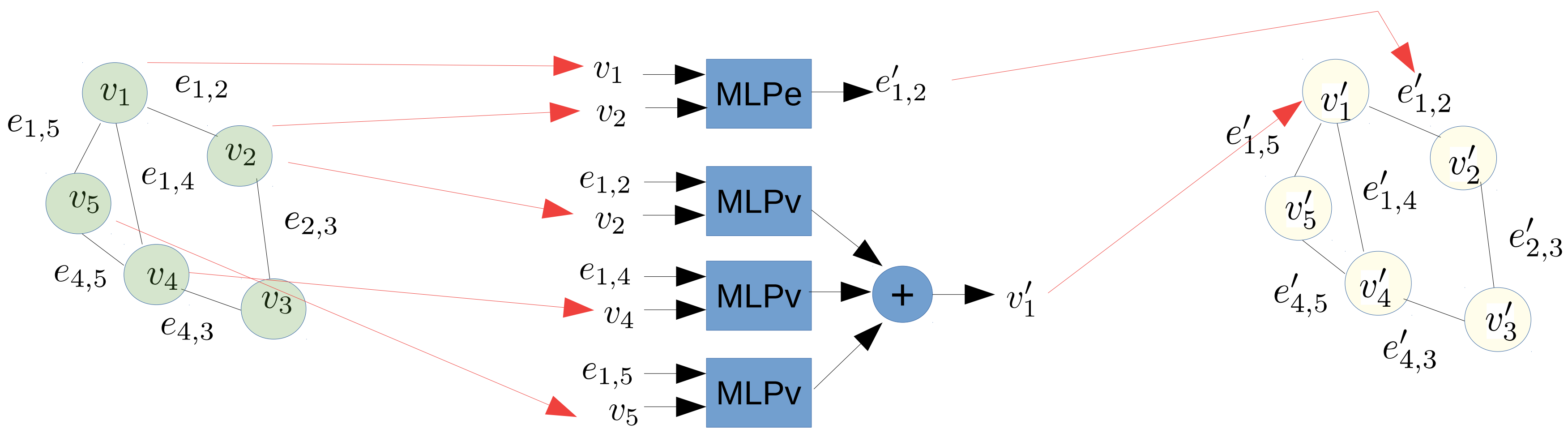
$y_{i,j}$ = binary variable associated with link

$c_{i,j}$ = cost associated with link

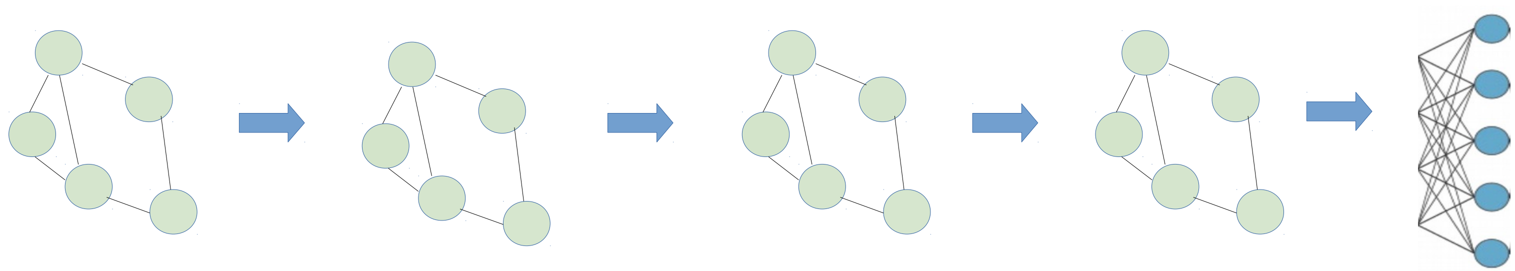
  - DL approach: Train a graph neural network to predict the binary variables $y_{i,j}$

Graph Neural Net

- Learning on graphs

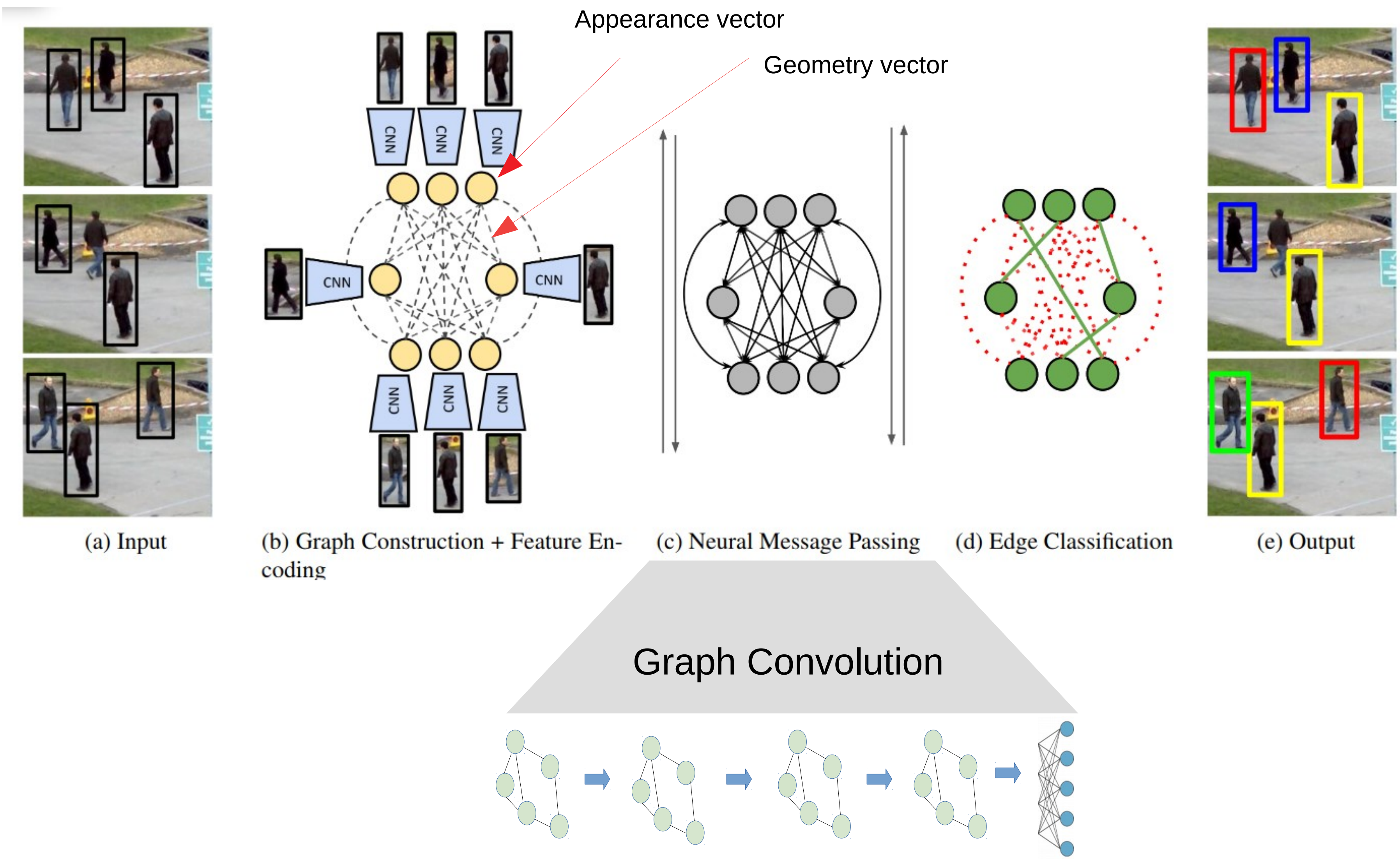


Graph Convolution Layer



Graph Convolution Network

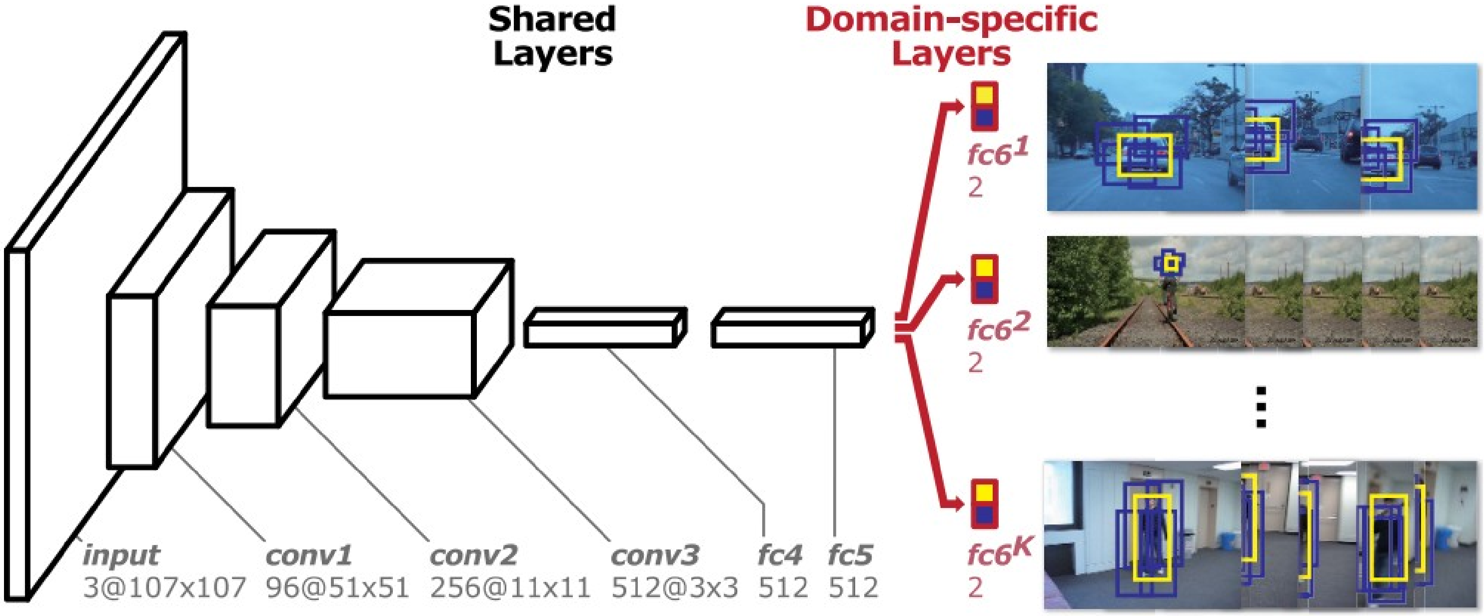
GNN Based Association



Learning a Neural Solver for Multiple Object Tracking, G. Braso L. Leal-Taixe 2020

Popular Deep Learning Based Tracking Systems

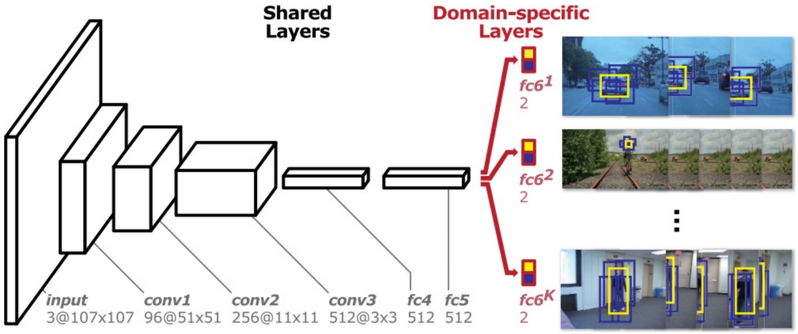
Multi-Domain Convolutional Net for Tracking (MDNet)



- Considers the problem of tracking of an arbitrary object.
- Object class in tracking can be a completely new class than in training
- Adaptation (re-training) in tracking is necessary
- Available data in tracking is limited
  - Make the trainable parts small
  - Shared layers are trained in the training phase (fine tuned in tracking phase).
  - Domain specific layers are trained in the tracking phase (from scratch).

Learning Multi-Domain Convolutional Neural Networks for Visual Tracking, Hyeonseob Nam and Bohyung Han, 2016

MDNet Operation - Training

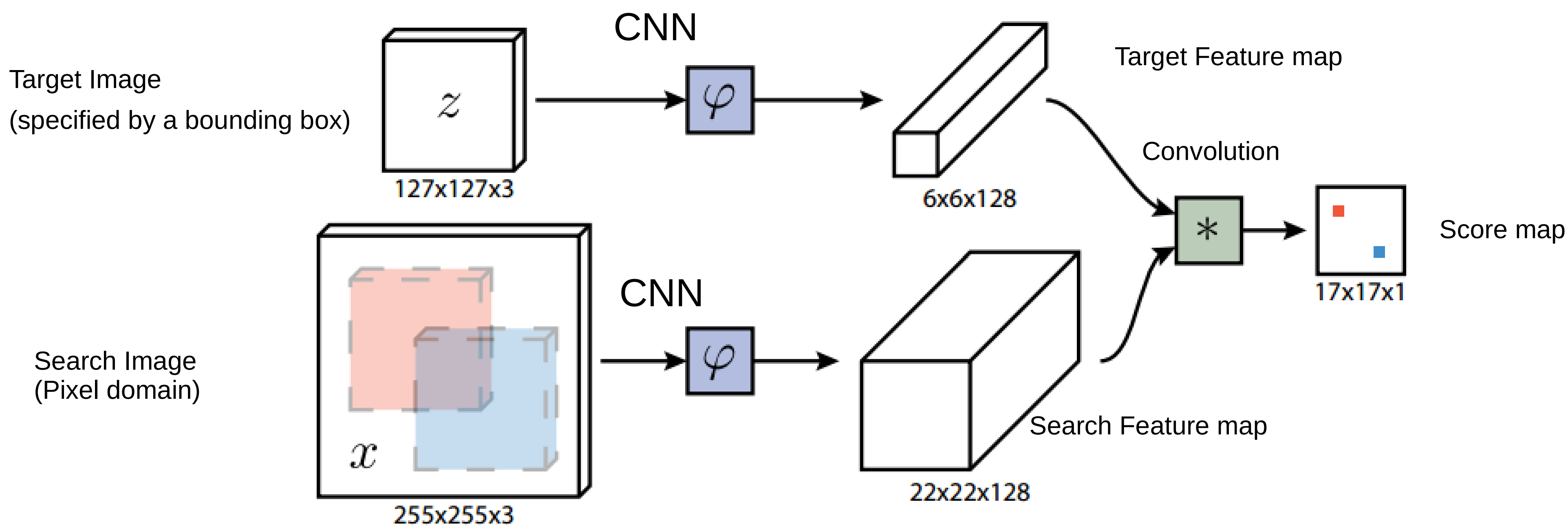


- Collect data from multiple domains
  - Images patches of objects (positive samples)
  - Image patches of background (negative samples)
- Train the system to classify the image patches
  - Positive samples to class 1
  - Negative samples to class 0
  - If the data sample belongs to domain D
    - Update the shared layer parameters in training
    - Update the classification head for domain D

MDNet Operation -Tracking

1. Remove all the domain specific heads
2. Add a new head (w) and initialize with random values
3. Take the test sequence and the start bounding box
4. Draw random patches around the start bounding box and categorize as positive samples (S+) and negative samples (S-)
   - S+ = samples which have overlaps with the start bounding box (eg: IoU > some threshold)
   - S- = Otherwise
5. Fine-tune shared layers and train (w) with S+ and S-
6. Let the current target x(t) be the image patch of the bounding box
7. From the next image draw N random patches in translation and scale using a Gaussian distribution whose mean is the position and scale of x(t)
8. Classify the random patches. Find the patch with the maximum probability x*(t)
9. If probability of x*(t) > 0.5
   - Draw samples around x*(t)
   - Classify them as as S+ and S-
   - Add them to an online training set
10. If probability of x*(t) < 0.5
    - Fine-tune shared layers and train w with the online training set
11. Let x(t) = x*(t) and go to step 7

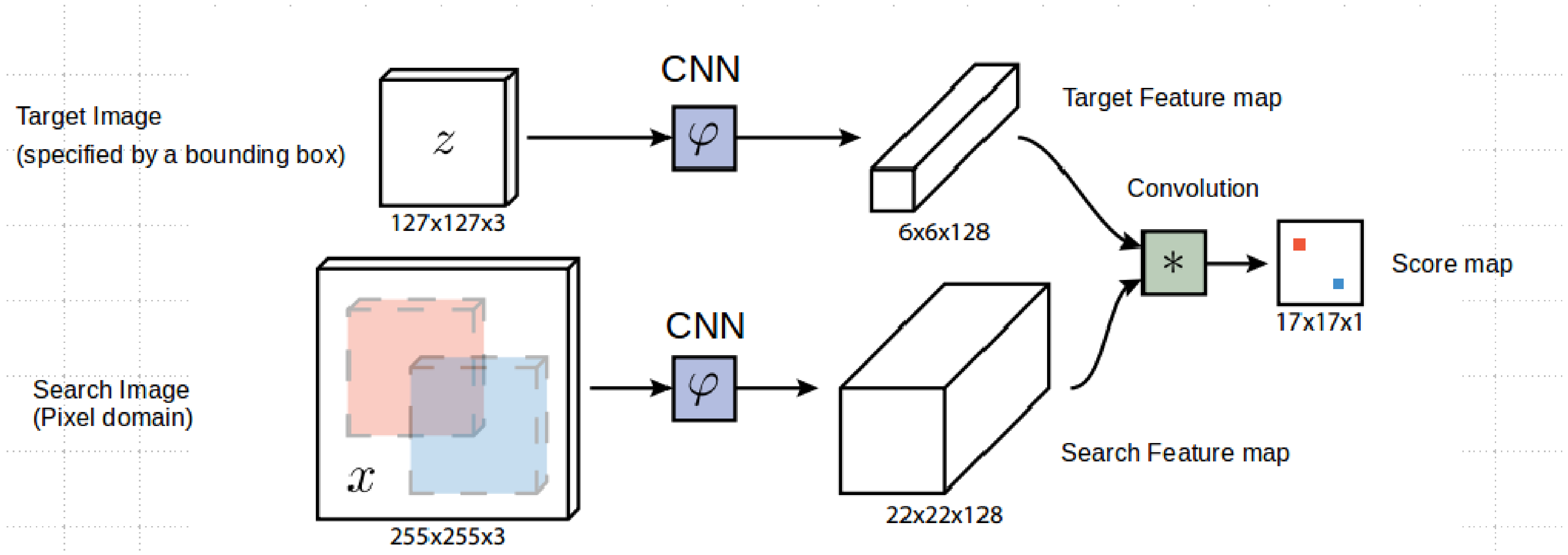
Fully Convolutional Siamese Network (SiamFC)



- Applied to Single Object Tracking (SOT)
- Arbitrary object tracking (i.e. object/object class to be tracked is not known a-priori)
  - Only the bounding box is given
- Trained to learn the similarity between the target image and the search image
- Fully Convolutional
  - Translations in pixel domain are uniquely corresponding to the translations in the feature domain
  - Usual convolutional network operations satisfy this, unless padding is used.

Fully-Convolutional Siamese Networks for Object Tracking, Luca Bertinetto et.al, 2016

SiamFC Training



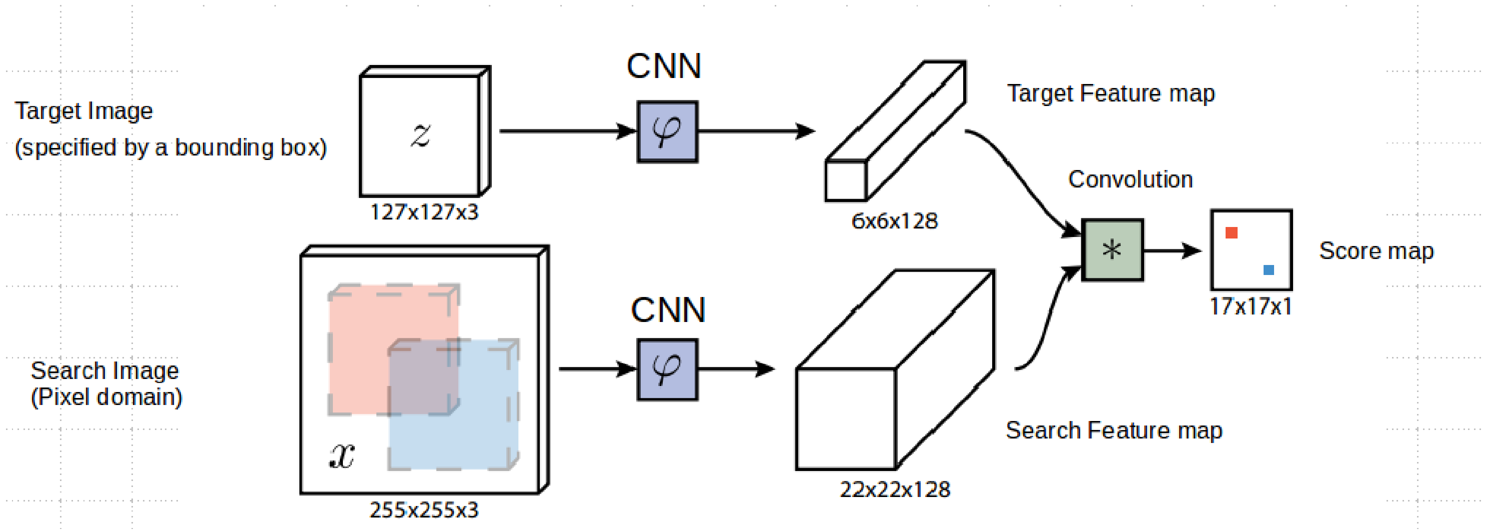
- Prepare a data set of image pairs taken from videos
  - Similar and different image pairs.
- Feed with each image pair and calculate the loss

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

  - v is the corresponding value in the score map
  - y is the label (+1 for similar images, -1 otherwise)
- Train the network to minimize the average loss

Fully-Convolutional Siamese Networks for Object Tracking, Luca Bertinetto et.al, 2016

SiamFC Tracking



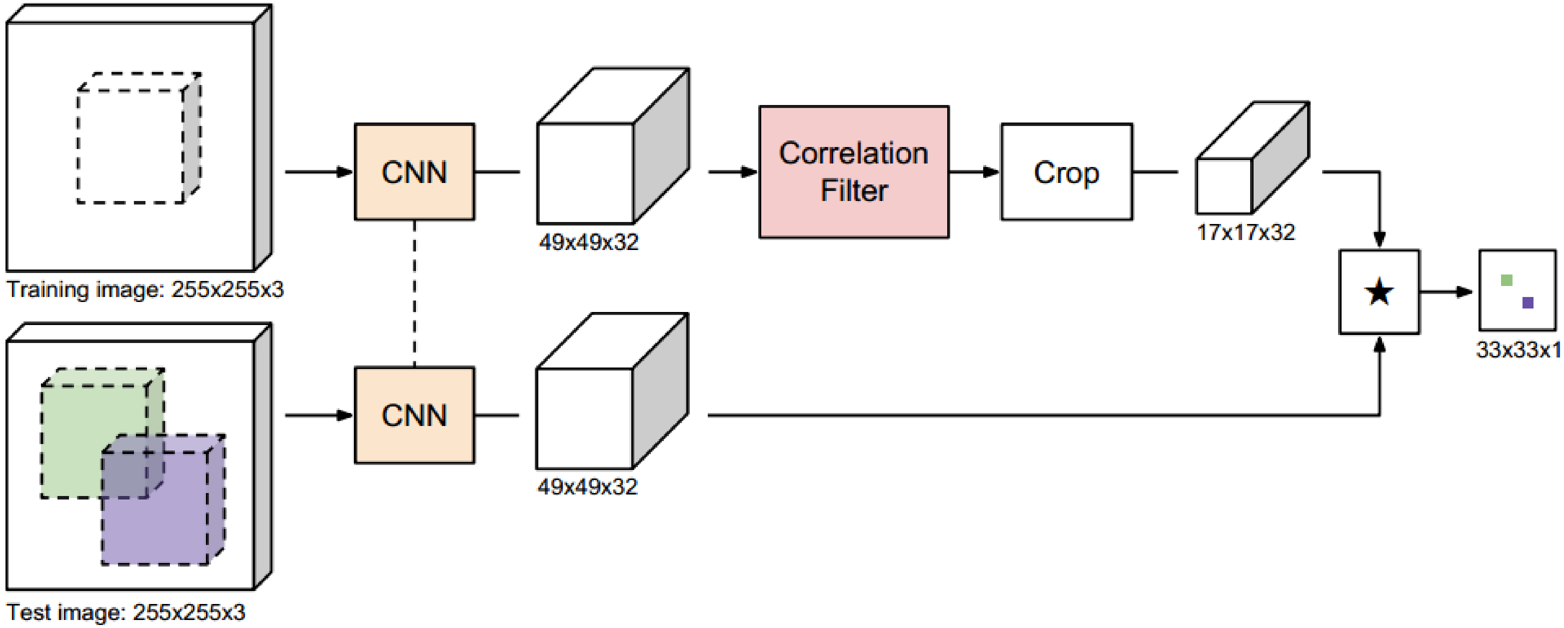
1) Get the initial target z (bounding box)
2) Prepare the search images
    - Get the next frame in video
    - Make several (3-5) copies and scale them with a predefined set of ratios
3) Run target and search images through the network and find the max score for each scale
4) Set the track location to be the location of the max score
5) Go to step 2 above and repeat

Fully-Convolutional Siamese Networks for Object Tracking, Luca Bertinetto et.al, 2016

Strengths & Weaknesses of SiamFC

- Strengths
  - Fast (due to efficient matching)
  - No re-training at test-time (in tracking)
- Weaknesses
  - Differences in trainset and testset can lead to performance degradation
  - Handling scale invariance is not very good
  - When the background is filled with other objects (distractors), SiamFC is not robust

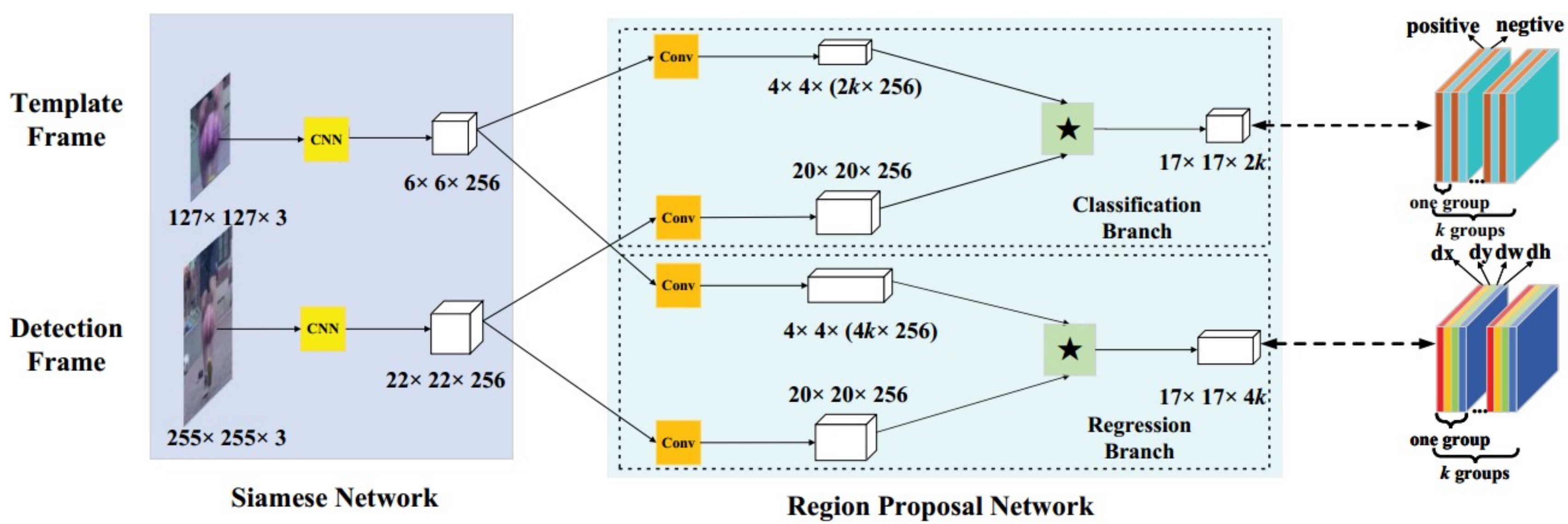
SiamFC with Correlation Filter



- Plain SiamFC is fixed in tracking
  - Differences in test set and train set can hurt the performance
- SiamFC with correlation filter adds an adaptable element (correlation filter) to the architecture
  - Correlation filter parameters are updated (re-trained) in tracking
  - Correlation filter is a linear operation with few parameters
    - Can be re-trained fast
    - Can be re-trained with few data samples.

End-to-end representation learning for Correlation Filter based tracking, Jack Valmadre et.al, 2017

SiamFC RPN



- Improves scale invariance
- Instead of running search images of different scales, use a regional proposal network (RPN)
  - Regression: Predict the bounding box
  - Classification: Classification score for the bounding box
- Involved procedure for picking the track location.

High Performance Visual Tracking with Siamese Region Proposal Network, Bo Li et.al, 2017

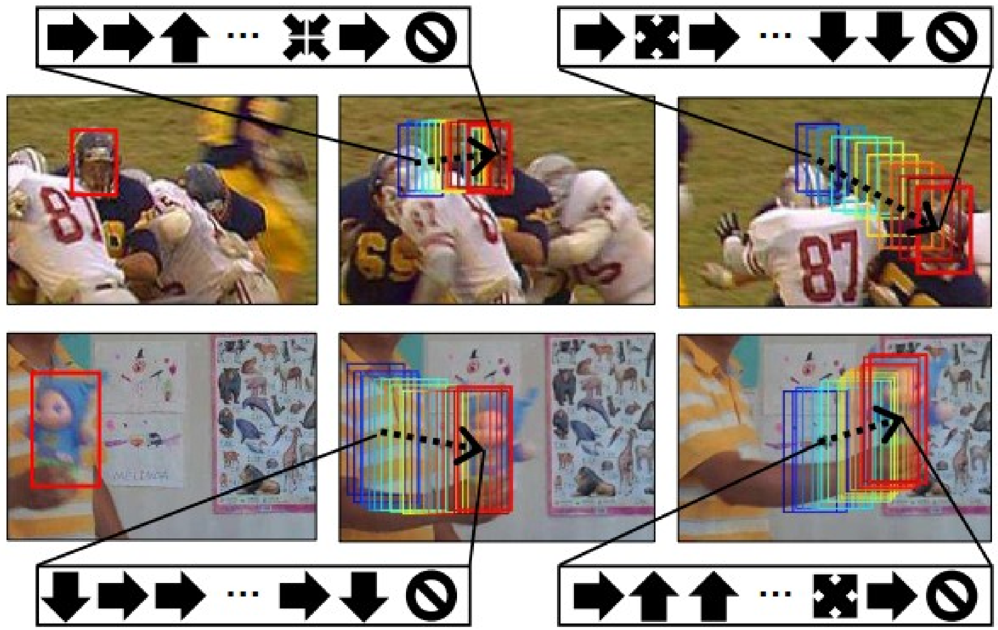
Distraction Aware Training

(a) detection pairs (b) negative pairs from the same categories (c) negative pairs from different categories

- Use a training database with
  - Augmented positive pairs
  - Semantically negative pairs

Distractor-aware Siamese Networks for Visual Object Tracking, Zheng Zhu et.al 2018

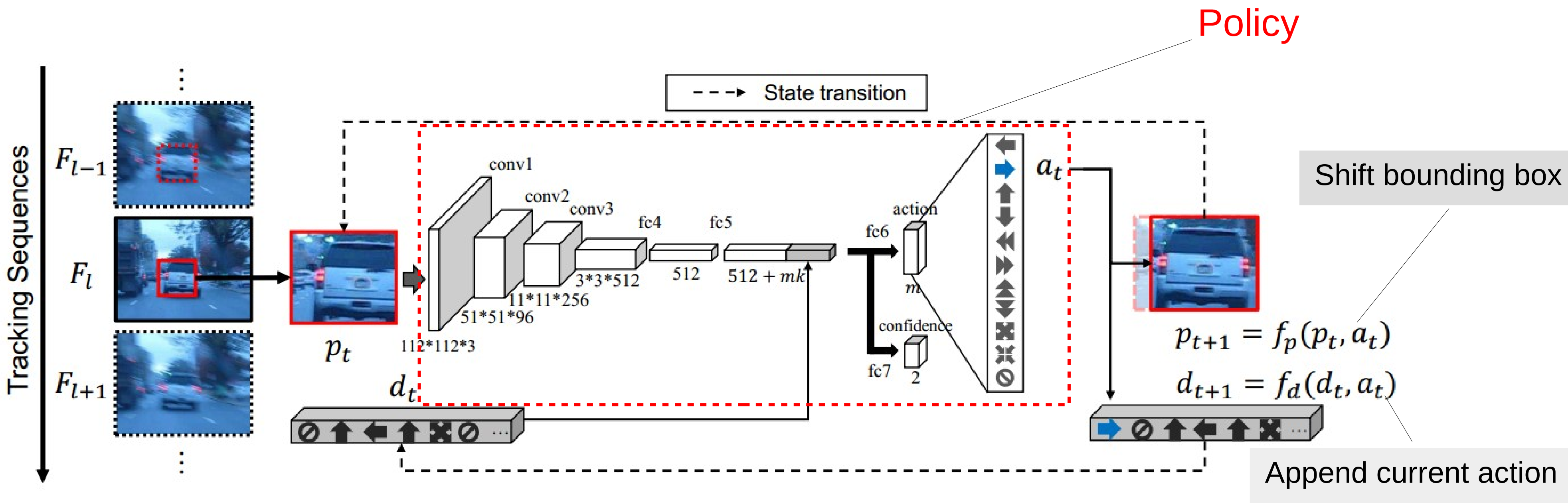
Action Desicion Network (ADNet)



- Detect objects by learning transitions to an initial bounding boxes
- Relatively light computation
- Better handling of scale and translation

Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning, Sangdoo Yun, 2017

ADNet Architecture



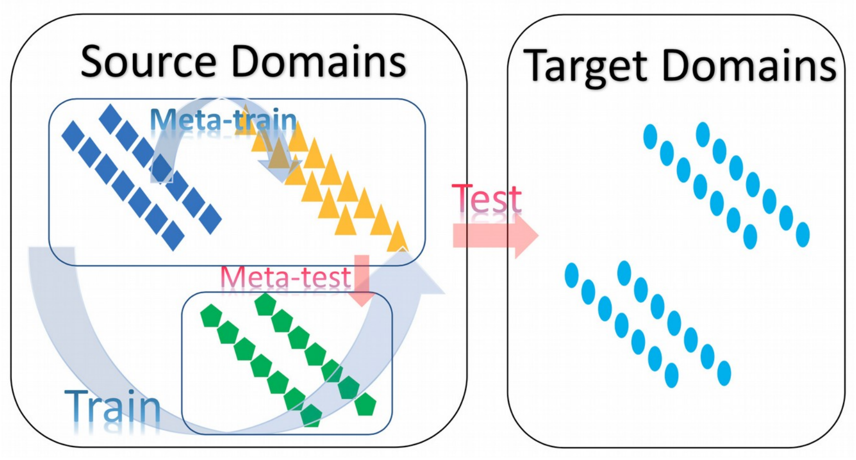
- State: $\begin{bmatrix} p_t \\ d_t \end{bmatrix}$ 　　p_t The current image patch
  d_t 10 previous actions
- Action a_t : One operation drawn from a set of 11 (***left, right, up, down, big-left, big-right, big-up, big-down, scale-up, scale-down, stop***)
- Environment (state transition) $\begin{bmatrix} f_p(p_t, a_t) \\ f_d(d_t, a_t) \end{bmatrix}$
- Reward: 1 if bounding box lands on the ground truth when action is *stop*, 0 otherwise

ADNet Operation

- Training (Two steps):
  1. Supervised training
     - Generate state-action pairs by taking random samples around the ground truths
     - Train the policy to map states to actions
  2. Reinforcement learning
     - Policy gradient based training for each image
     - Bounding box at ***Stop*** action is transferred to the next image
- Tracking (testing):
  1. Policy-environment interaction loop
  2. Online adaptation
     - Similar to the supervised training above
     - Currently predicted state (bounding box is used as the ground truth)

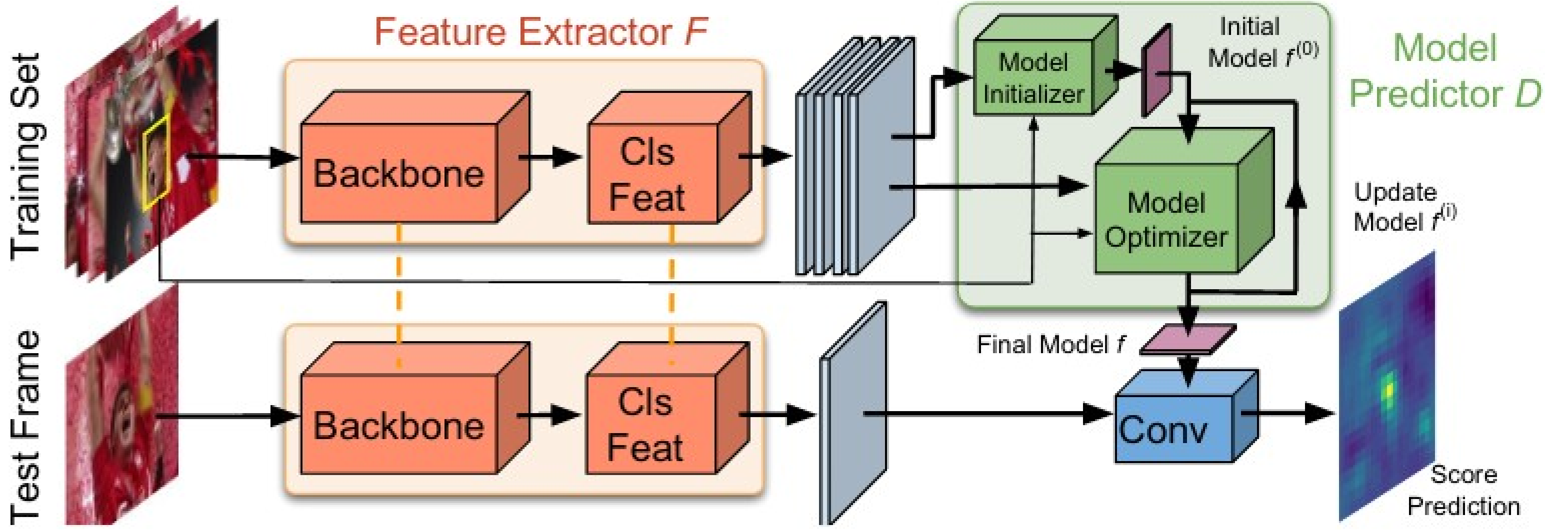
Tracking as Few Shot Learning

- A core problem of general visual tracking is "Class(es) of the object(s) to be tracked is (are) NOT known at training time"
- The tracker needs to adapt to new object classes after seeing few examples
- Tracking needs to solve a few-shot learning (meta-learning) problem.



Discriminative Model Prediction (DiMP)

- Contains meta-learning elements
- Upper branch estimates model parameters and transfers to the lower branch
- In tracking, the model parameters are adapted to the new classes and transferred to the lower class.



Learning Discriminative Model Prediction for Tracking Bhat G et.al, 2020