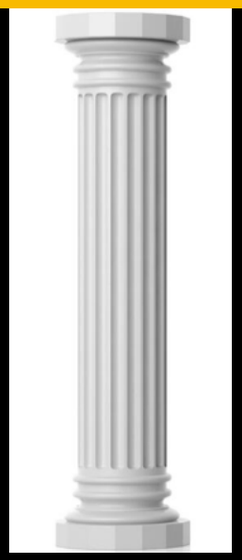
# Tracking

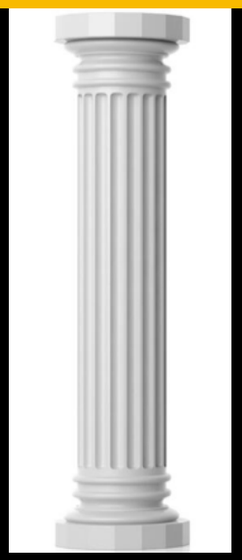
Earl Wong

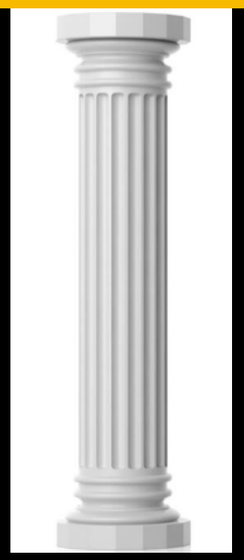
Labeled Dataset

Algorithm

Loss Function







**Labeled Dataset**

| Dataset | ImageNet Video | VOT18* | MOT20 | YouTube BB | Tracking Net | GOT-10K | LaSOT |
|---|---|---|---|---|---|---|---|
| **Categories** | 30 | 24 | 5 | 23 | 27 | 560 | 70 |
| **Sequences** | 3800+ | 60 (~356fr/ seq) | 8 (~1600fr/ seq) | 380K (~570fr/ seq) | 30K (~470fr/ seq) | 10K (10fr/sec) | 1400 (~2500fr/ seq) |
| **Bounding Boxes** | 860K+ | ~21K | 1.6M | 5.6M | 14M | 1.5M | 3.5M |

**Labeled Dataset**

- Easy sequences in VOT18 were replaced with more difficult sequences in VOT19.

- Bounding boxes in VOT19 were replaced with segmentation masks in VOT20.

- The MOT dataset was introduced in 2015 to address non-singular object tracking.  (MOT20 contained up to 246 people in a single frame)

- Prior to the introduction of larger tracking datasets, object detection datasets like ImageNet DET and COCO were also used for training.

- Illumination changes, occlusions, object deformations, scale changes, motion, etc. introduce increasing levels of difficulty for any tracking algorithm.

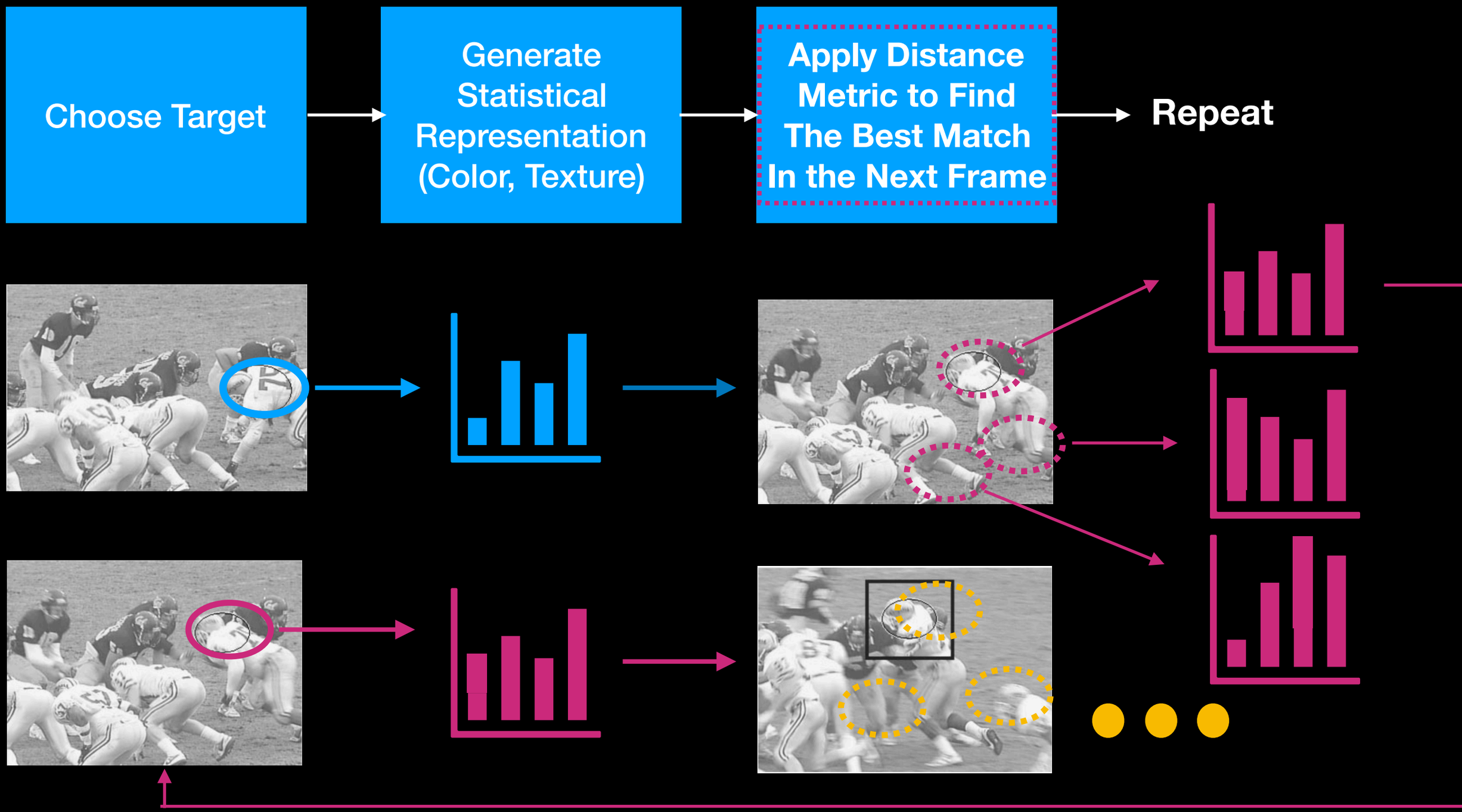
# Historical

- Unlike other areas of computer vision (image recognition, object detection, etc.), early deep learning (DL) based tracking algorithms actually underperformed existing benchmarks.

- Non-DL based tracking algorithms and DL based tracking algorithms competed head to head in different tracking benchmarks as late as 2017.

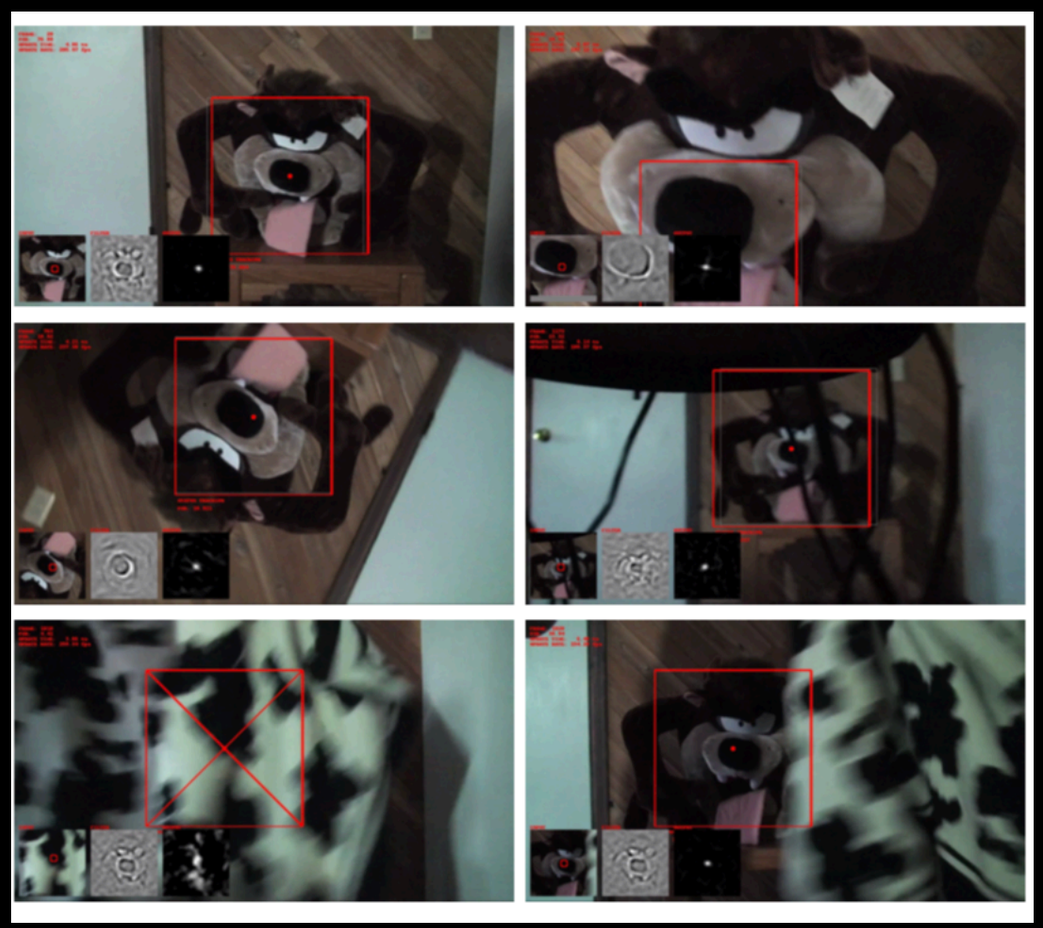
# Mean Shift

Introduced in 2000, the mean shift tracking algorithm was an industry workhorse. It's simplicity and beauty is shown below.



# Minimum Output Sum of Square Error (MOSSE)

MOSSE introduced the concept of using an adaptive correlation filter (H*) for tracking.



**The correlation filter adapted to changes in rotation, scale, lighting and occlusion.**

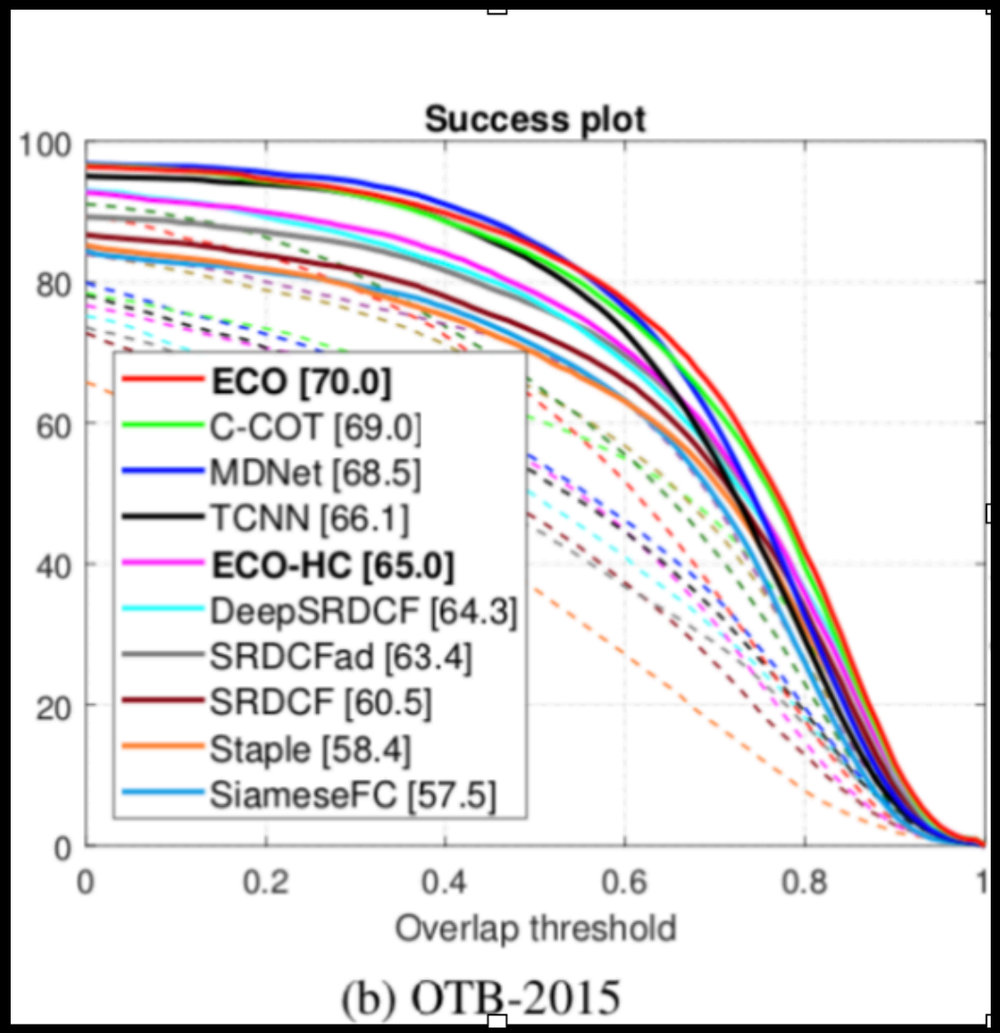
1) Let f denote the input. Let h denote the adaptive filter that we wish to apply.
2) Let F = fft(f) and H = fft(h), where fft denotes the fast fourier transform.
3) Let $G = F \cdot H^*$ (Elementwise multiplication) (Ideally, we want G to have a “peaky” output.)
4) With this in mind, we seek the following H*:

$$\min_{H^*} \sum_{i=1}^{n} \left| F_i \cdot H^* - G_i \right|^2$$

5) The solution for H* is given by:

$$H^* = \frac{\sum_{i=1}^{n} G_i \cdot F_i^*}{\sum_{i=1}^{n} F_i \cdot F_i^*}$$

# ECO (2017)



(b) OTB-2015

**As of 2017, deep learning approaches such as MDNet and SiameseFC had not overtaken hand crafted correlation filter based approaches like ECO.**

By 2017, the original MOSSE filter (now C-COT) had witnessed a huge increase in complexity.

Although C-COT produced a 2x performance improvement, C-COT was 1000x slower.

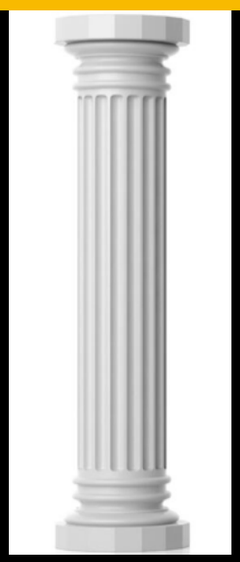
Three modifications were made to C-COT, resulting in an improved filter called ECO:

1) By applying a factorized convolution operator, the complexity was reduced from 800K parameters to 160K parameters.

2) By employing a compact generative model, the number of samples needed for online learning was reduced.

3) By decreasing the parameter update frequency, the tracking performance improved.

**Algorithm**

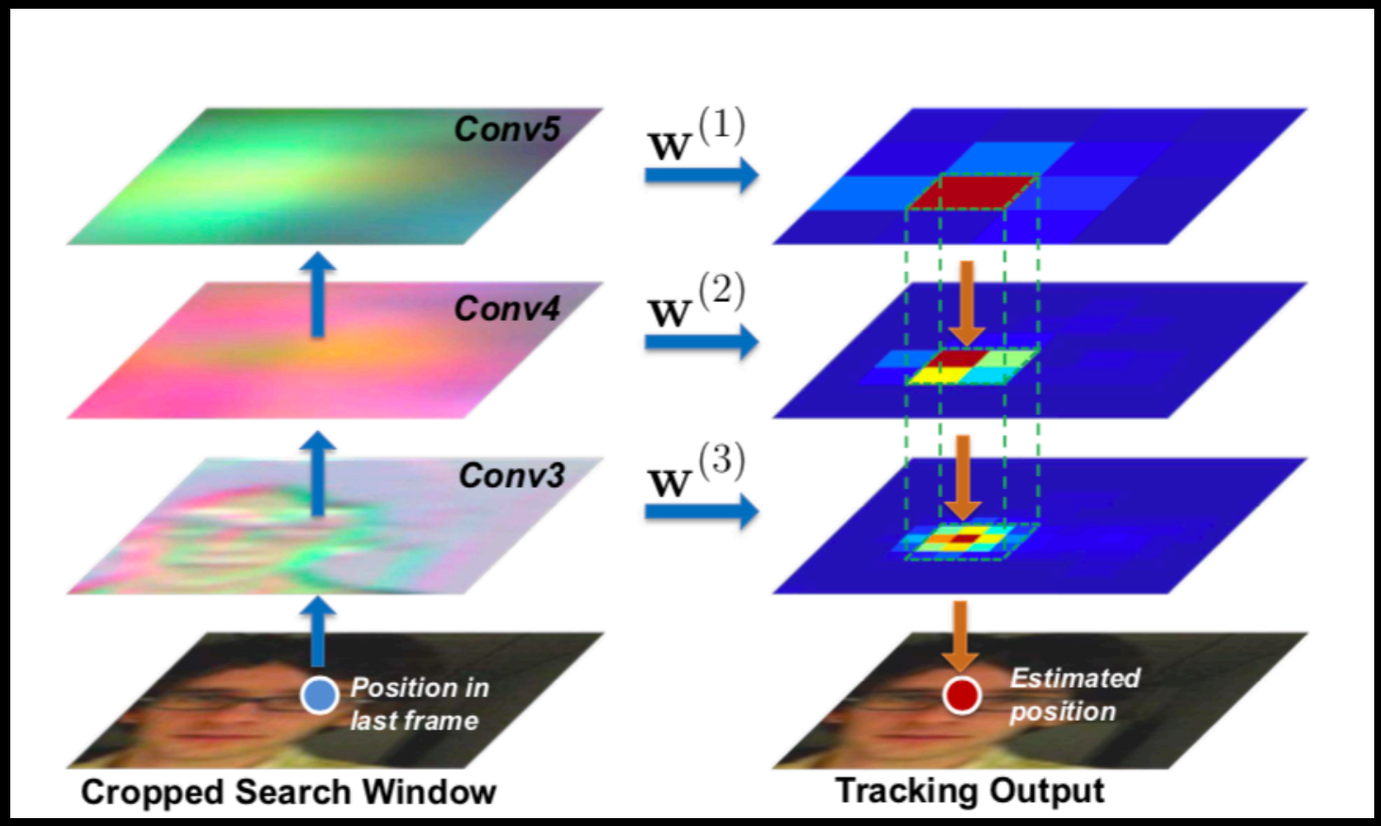


# The DL Era

- Convolutional Features / Hierarchical Convolutional Features (2015)
- Multimodal Domain Networks: MDNet (2016)
- Convolutional Siamese Networks (2016)
- SiameseRPN (2018)
- Accurate Tracking by Overlap Maximization: ATOM (2019)
- AlphaRefine (2020)
- Tracking Without Bells and Whistles (2019)

This list is only intended to highlight some of the main lines of thinking.  For a more comprehensive list, please refer to the VOT20 competition results.

# Hierarchical Convolutional Features




Initially, the feature maps and the hierarchical structure of deep CNN's were exploited.

First, response maps were computed by applying learned linear correlation filters (with weights w) to the feature maps.

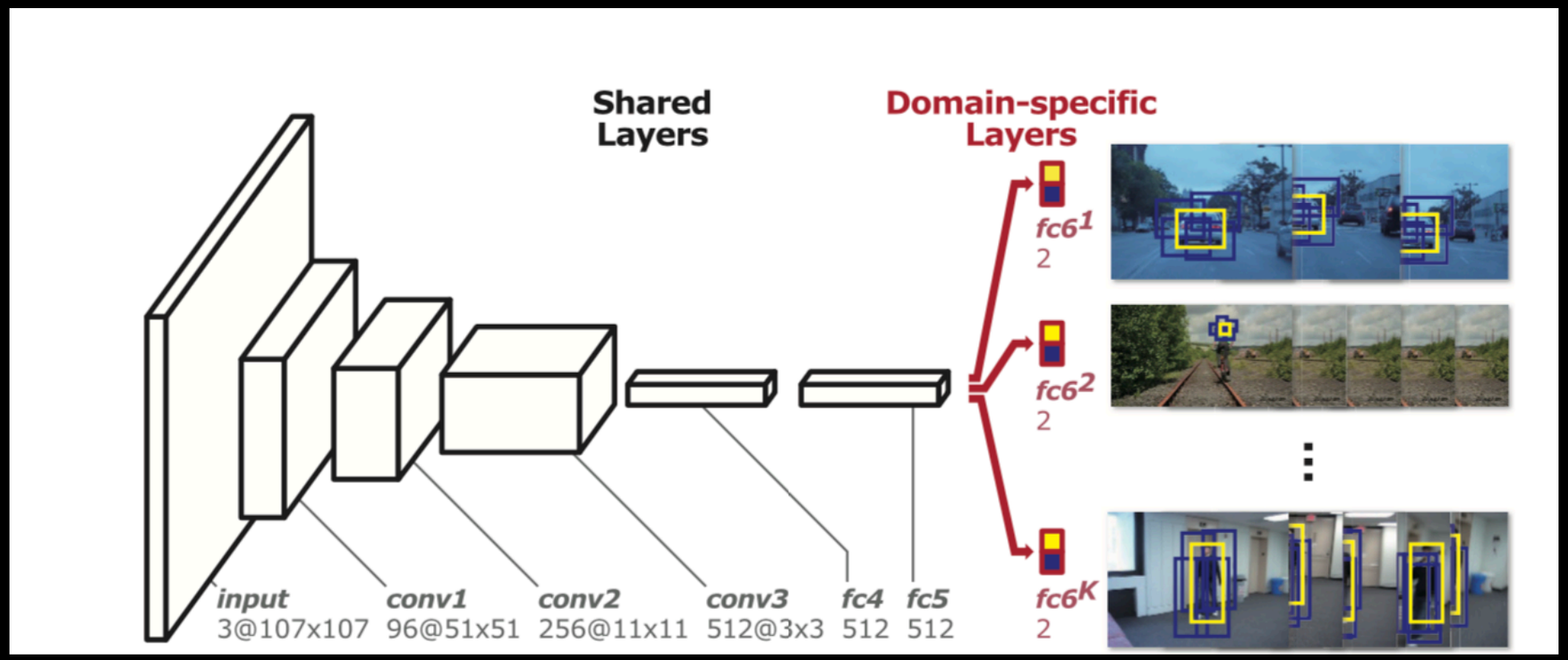
Next, the peaks of the response maps were tracked from coarse to fine.

The strongest responses were used to determine the target location and the target size.

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{m,n} \|\mathbf{w} \cdot \mathbf{x}_{m,n} - y(m,n)\|^2 + \lambda \|\mathbf{w}\|_2^2$$

**x = feature map(s), w = learned correlation filter(s) weights,**
**y = output label(s) corresponding to the target location**

# Multi-Domain Networks (MDNet)




Multi-domain networks represented an end to end deep learning solution for tracking.

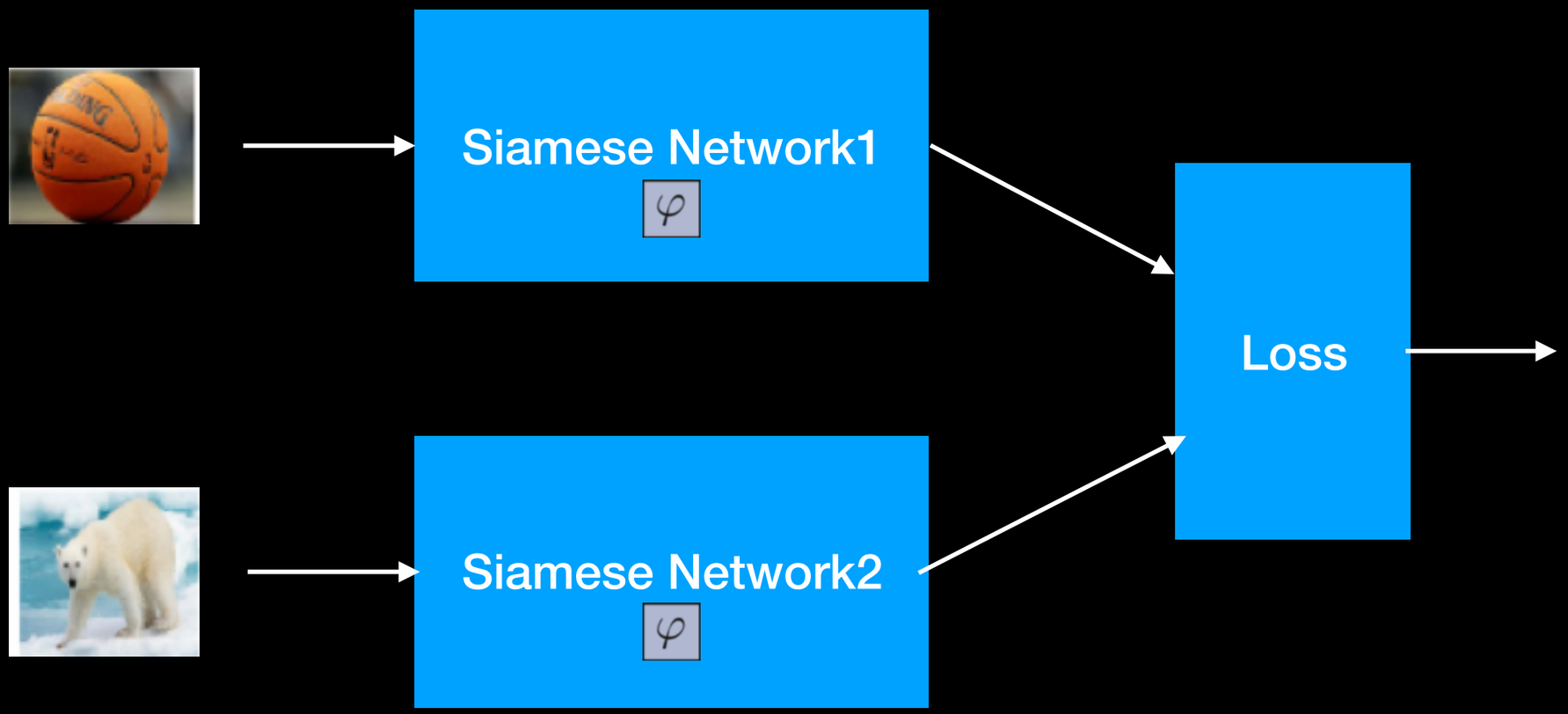
First, a network consisting of a base network (=shared layers) and K branches (=domain specific fully connected layers) was created.

Each connected branch was individually trained end to end using K unique training sequences.

This resulted in a base network containing a “shared representation” for K the training sequences.

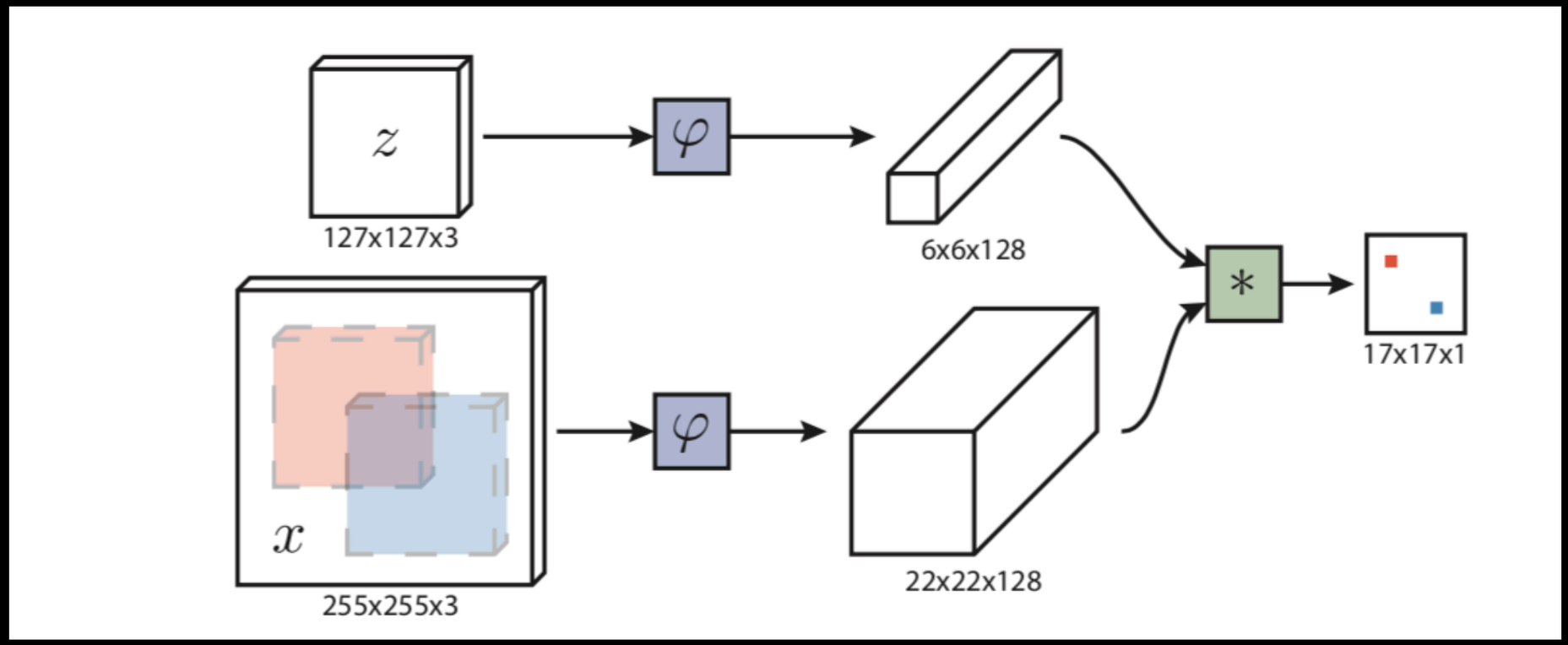
To apply the network, a new branch was created for a candidate target.  Online processing / updates then followed for all fully connected layers (= a frozen base network).

# Convolutional Siamese Networks




A fully convolutional siamese network was trained, using a contrastive loss function.

# Convolutional Siamese Networks




Target and search images were then processed by the siamese networks, producing two output embeddings.

A similarity function was applied to the output embeddings, to determine the spatial location associated with the best match.

No online processing was needed, since the siamese networks were pre-trained.

The authors of online tracking algorithms argued that this was a significant weakness.

# A Note on Bounding Box Accuracy

- Up to this point, not much emphasis was placed on producing accurate bounding box estimates.

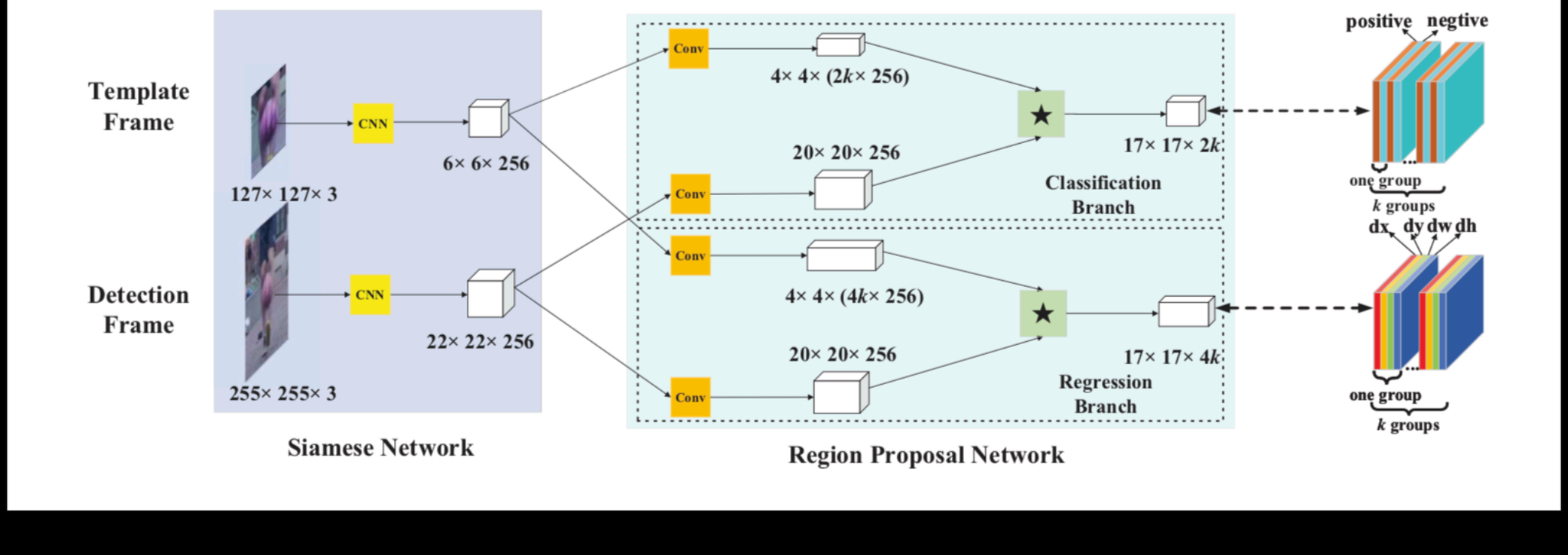
- In Convolutional Features, multiscale information was used to track the target location.

- In Siamese Networks, a similarity function was applied to the reference and target embeddings.

- In MDNet, bounding box estimates was produced using a network trained from targeted input sequences and the current online sequence.

- However, things were about to change.

# SiameseRPN




Previously, the output embeddings from a Siamese network were used to perform tracking.

Now, the output embeddings underwent additional processing.

Specifically, the embeddings were used as inputs to two network branches: a classification network branch and a bounding box network branch

The reference embedding was further processed by convolutional layer that increased the number of output channels by 2k and 4k, for use in the classification and regression network branches.

# SiameseRPN

The target embedding was processed by convolutional layers that maintained the number of original output channels.

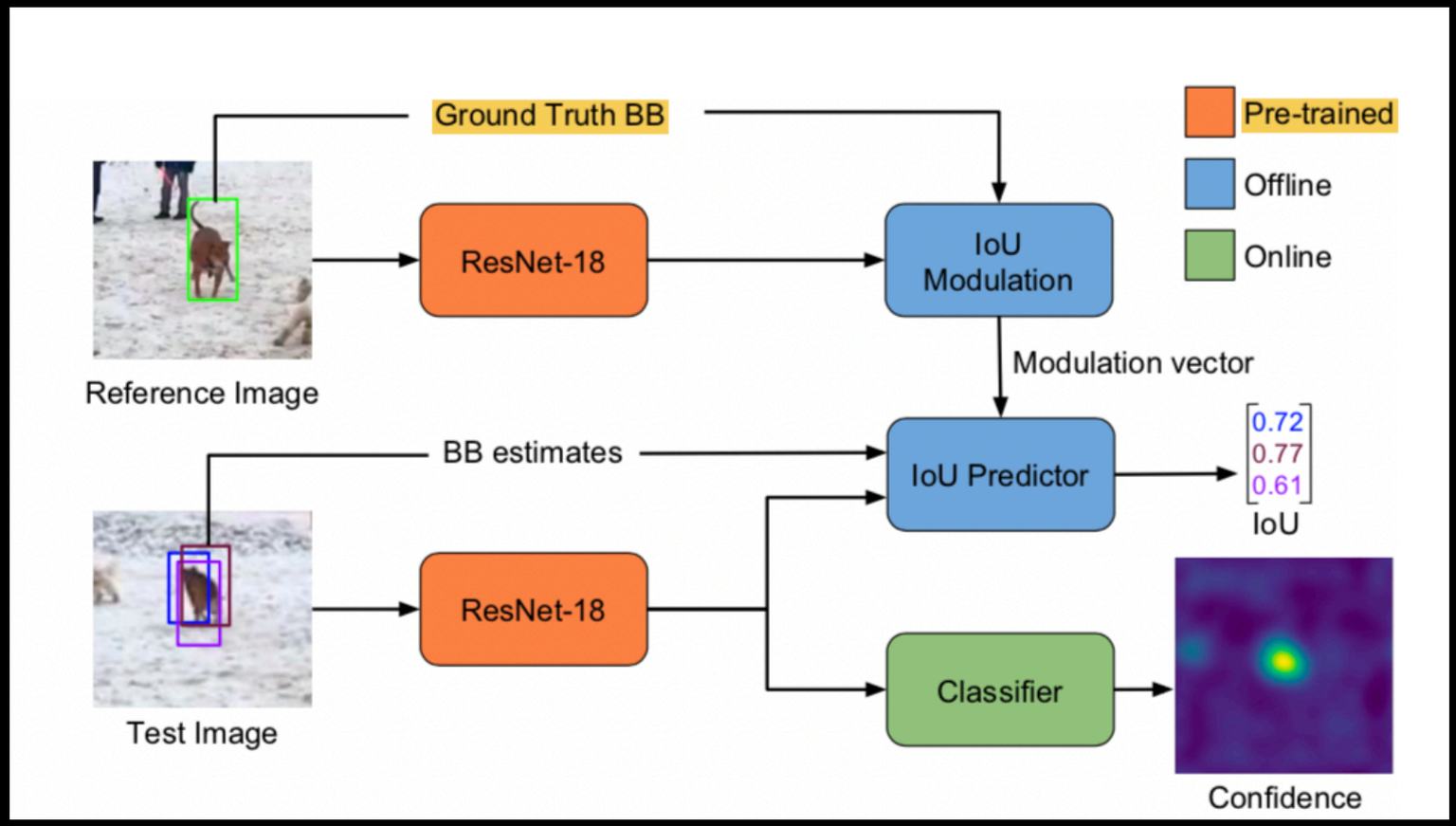
Next, for each branch, a convolution operation was performed between the transformed reference and transformed target embeddings.

This produced two new output tensors - one for the classification branch and one for the bounding box network branch.

The number of channels in each output tensor was determined by the number of anchor boxes selected for use.

When this new network was trained end to end, improved bounding box regression outputs were produced.

# Accurate Tracking by Overlap Maximization (ATOM)



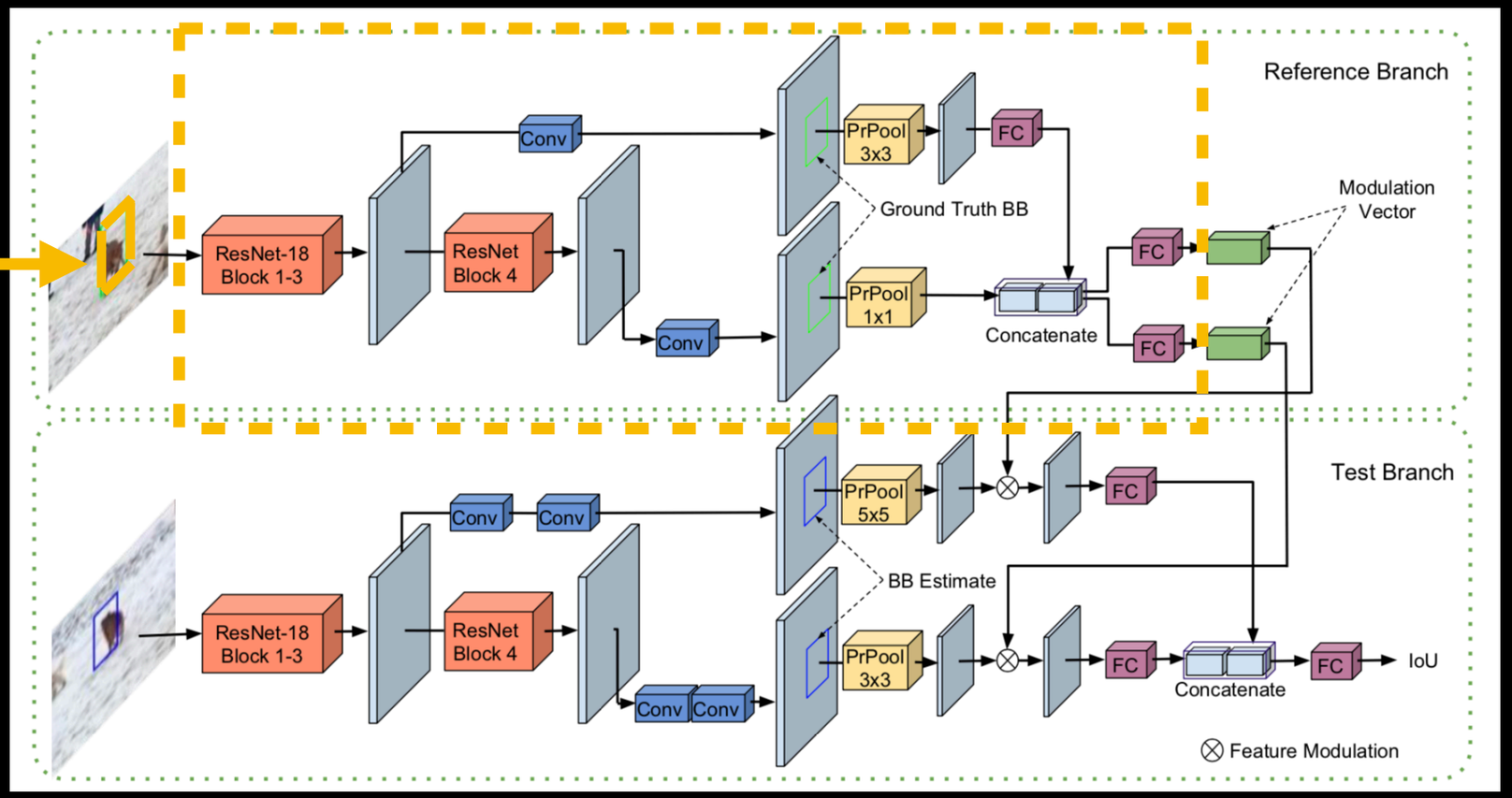

Like SiameseRPN, the authors also divided the tracking problem into a classification task and a bounding box estimation task.

In the classification task, a MLP was trained online using conjugate gradient.

In the estimation task, the authors proposed a new network that 1) was trained offline and 2) whose goal was to maximize  bounding box overlap with GT (IoU). (See next slide.)

Even when trained only on ImageNet VID data, the resulting network outperformed all current SOTA methods on recent test sets like LaSOT and TrackingNet.

# Accurate Tracking by Overlap Maximization (ATOM)



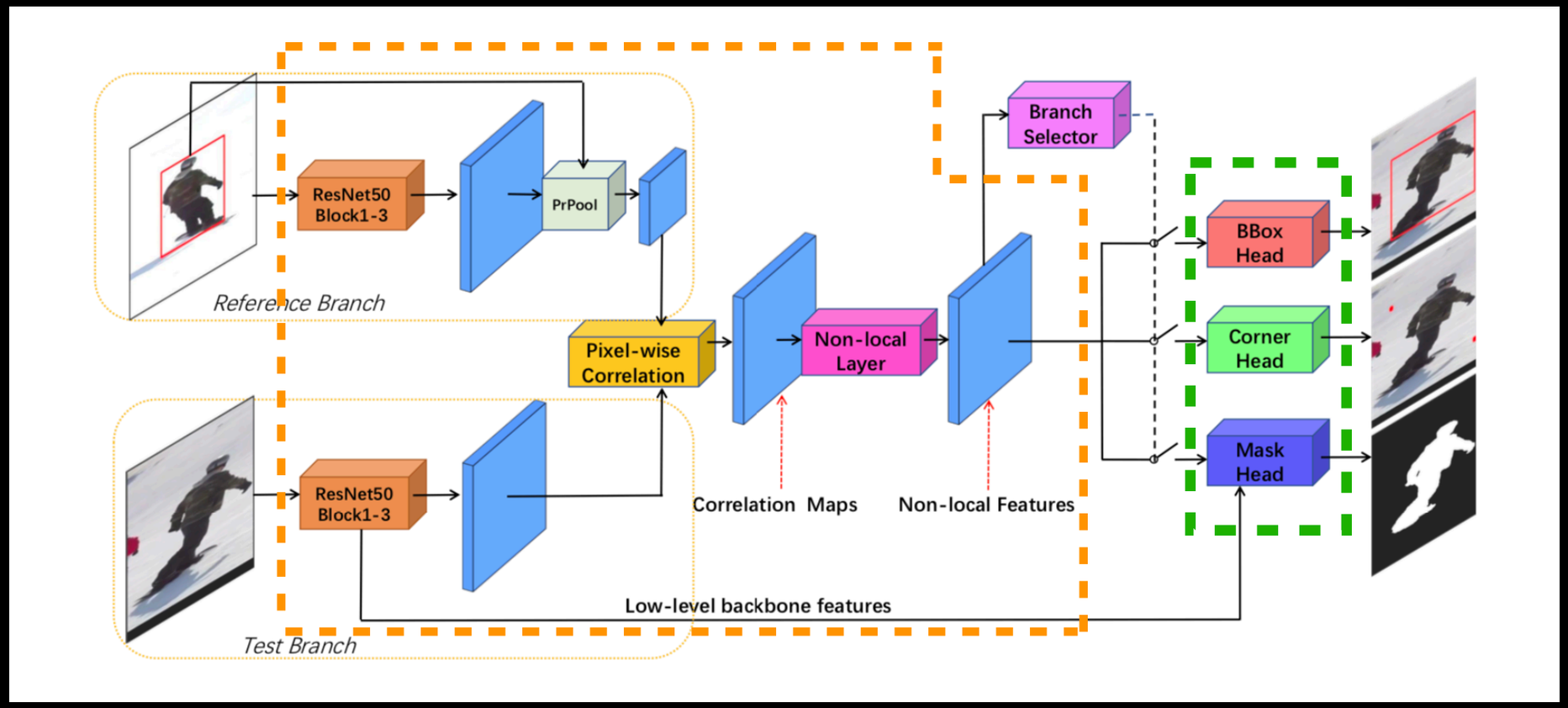

Key question: How do we incorporate the reference target appearance information into the test branch, to produce accurate bounding box estimates?

Based on empirical investigations, the authors concluded that the reference appearance information could be best summarized by computing a modulation vector, using the boxed network shown above.

The modulation vector was then used to modulate the feature layer representations in the test branch.

The resulting output was passed through several though FC layers, producing improved bounding box estimates that maximized overlap (IoU) with GT.

# AlphaRefine



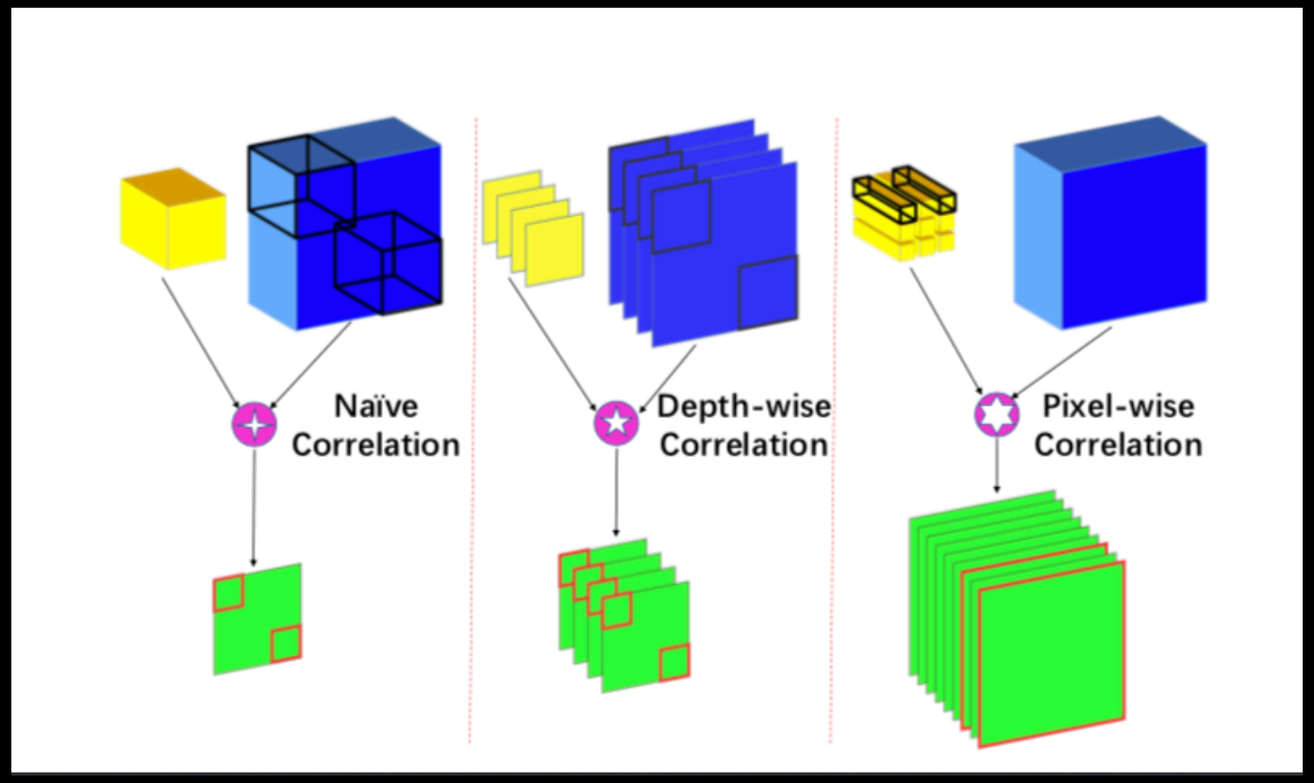

AlphaRefine was proposed as a “drop in” network to improve bounding box accuracy.

First, a base network and three output heads were trained: bounding box, corner points and mask (Note: Earlier, several authors had trained networks with mask heads instead of bounding box heads.)

Next, the base network was frozen, and a separate branch selector network was trained to select the result that best agreed with the GT data.

Internally, the network employed a correlation map (which was computed using pixelwise correlation) and a non local layer.  (See next slide.)

# AlphaRefine



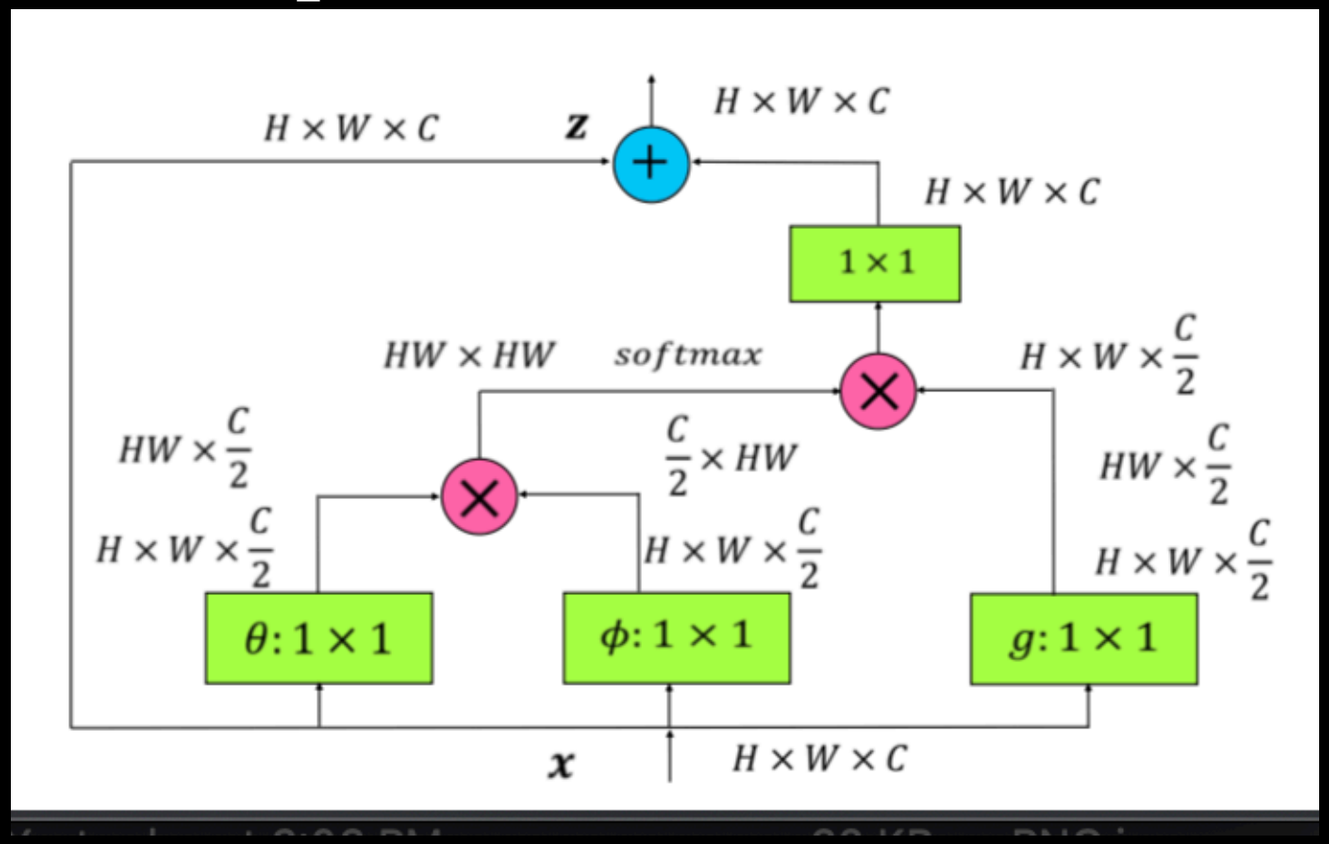

In ATOM, a modulation vector was computed to summarized the reference appearance information.

This modulation vector was then multiplied with the target output.

Here, the authors computed pixelwise correlations between the reference and target outputs, producing pixelwise based correlation maps.

Specifically, a correlation map was computed using each 1x1x N tensor output produced by the reference input.

# AlphaRefine



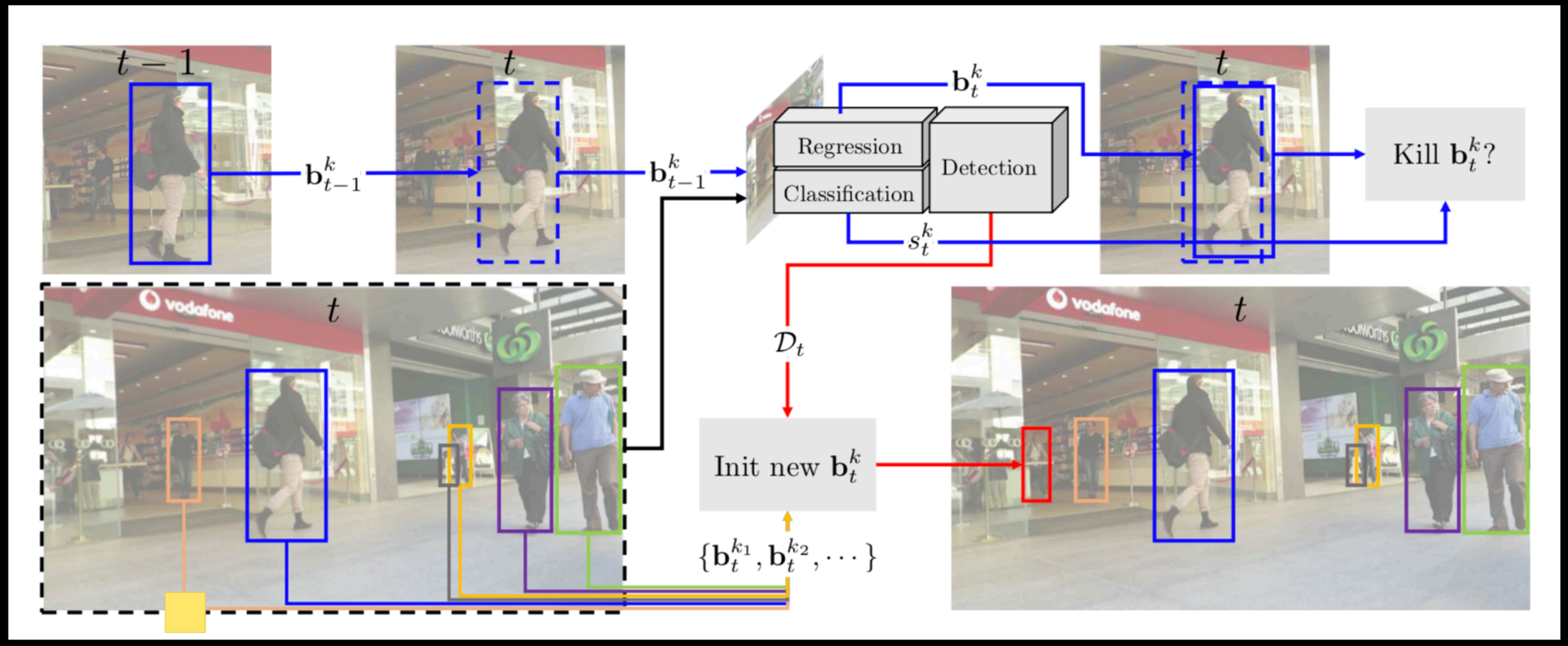

Next, a non-local layer was introduced to exploit any global contextual information contained in the correlation maps.

In general, a non-local layer (Non-local Neural Networks, Wang, et. al) computes a new response for a specific spatial position, by using a weighted sum of all of the candidate spatial positions.

Here, the authors employed a “covariance matrix like” calculation.

Note: Various elements in the above picture were not fully explained by the authors.

# Tracking Without Bells and Whistles




Taking a step back, we conclude with a very simple tracking algorithm that yields solid results.

Here, the tracking problem is viewed an object detection problem with temporal data association.

Tracking is performed by using the bounding box regressor and classifcation outputs from Faster R-CNN.

At each frame, exsiting objects were tracked, new objects were identified, and disappearing objects were removed.

To increase accuracy, an additional camera motion compensation model can be added, to account for large interframe motion and / or low frame rate updates.

## Algorithm

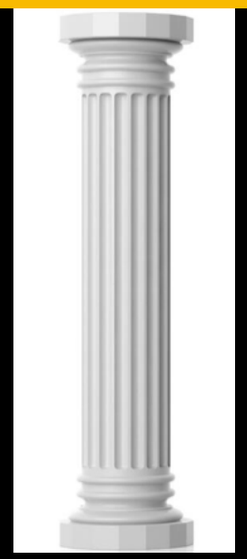
- Convolutional Features, MDNet and Siamese network employed CNN's of varying complexity.

- In SiameseRPN, two new convolutional branches were appended to the output embeddings produced by a Siamese network.

- In ATOM, the reference and tracking inputs were initially processed by indentical ResNet backbones.  A modulation vector was then created from the reference tensor output.  This vector was then introduced into the tracking path, to infuse the reference appearance information.

- In AlphaRefine, the reference and tracking inputs were initially processed by "near identical" ResNet backbones. Pixel correlation was then applied to generate correlation maps.  The maps were then processed by a non-local layer.  Three output heads were then trained on the output of the non-local layer.

**Loss Function**



## Loss Function

- Hierarchical Convolutional Features - Cross Entropy Loss
- MDNet - Regression Loss
- Siamese - Regression Loss
- SiameseRPN - Cross Entropy Loss + Regression Loss
- ATOM - IoU Loss + Regression Loss
- AlphaRefine - IoU Loss + Regression Loss + Binary Cross Entropy Loss

# References


- Real Time Tracking of Non Rigid Objects using Mean Shift, Comaniciu, et. al, CVPR 2000
- Visual Object Tracking using Adaptive Correlation Filters, Bolme, et. al., CVPR 2010
- ECO: Efficient Convolution Operators for Tracking, Danelljan, et. al., CVPR 2017
- Hierarchical Convolutional Features for Visual Tracking, Ma, et. al., CVPR 2015
- Convolutional Features for Visual Tracking, Danelljan, et. al. ICCV 2015
- Multi-Domain Convolutional Neural Networks for Visual Tracking, Nam and Han, CVPR 2016
- Fully Convolutional Siamese Networks for Object Tracking, Bertinetto, et. al., ECCV Workshop 2016
- High Performance Visual Tracking With Siamese Region Proposal Network, Li, et. al., CVPR 2018
- ATOM: Accurate Tracking by Overlap Maximization, Danelljan, et. al., CVPR 2019
- Learning Discriminative Model Prediction for Tracking, Bhat, et. al., ICCV 2019
- Fast Online Tracking and Segmentation: A Unifying Approach, Wang, et. al., CVPR 2019
- D3s-A Discriminative Single Shot Segmentation Tracker, Lukezic, et. al., arXiv November 2019
- AlphaRefine, Yan, et. al. arXiv July 2020
- The Eighth Visual Object Tracking VOT 2020 Challenge Results, Kristan, et. al., ECCV 2020 Workshops (Visual Object Tracking Challenge)
- Tracking Without Bells and Whistles, Bergmann, et. al., arxiv March 2019