Deep Learning for Object Tracking

Semester work Presentation

Alexandre Carlier

EPFL, Lausanne

January 2019

Introduction

- **Object tracking**: track an object in any sequence, given only its first frame bounding box annotation.

Figure 1: SiamRPN++ tracker on the MountainBike sequence of OTB-2015.

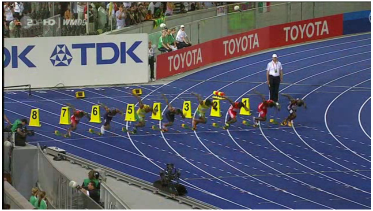
Tracking is hard!

To be successful, the tracker has to be:

- **Class-agnostic**
- Robust to severe **appearance changes** (lighting conditions, rotations, changes in aspect ratio, motion blur)
- Able to handle temporary **occlusions**
- Robust to semantic **distractors**

A challenging benchmark dataset: OTB-2015

Distractors:

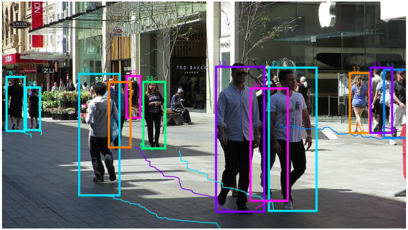


Rotations:

Scaling:

Variants of the tracking problem

People tracking



- Not class-agnostic.
- Tracking by detection paradigm.
- Benchmarked on the MOTChallenge [Milan et al., 2016].

Semi-supervised video segmentation

- No 'causal' requirement (all the frames are provided from the beginning).
- No real-time requirement.
- Benchmarked on the DAVIS Challenge [Perazzi et al., 2016].
- Very short sequences (2-4 seconds, mean number of frames per sequence: 69.7).

Outline

1. Introduction
2. Related Works
   - Object Detection literature
   - Real-time trackers
3. My work
   - Reproducing SiamRPN
   - Other approaches
4. Results
5. Demo
6. Conclusion

Outline

1. Introduction
2. Related Works
   - Object Detection literature
   - Real-time trackers
3. My work
   - Reproducing SiamRPN
   - Other approaches
4. Results
5. Demo
6. Conclusion

Single Shot MultiBox Detector (SSD)

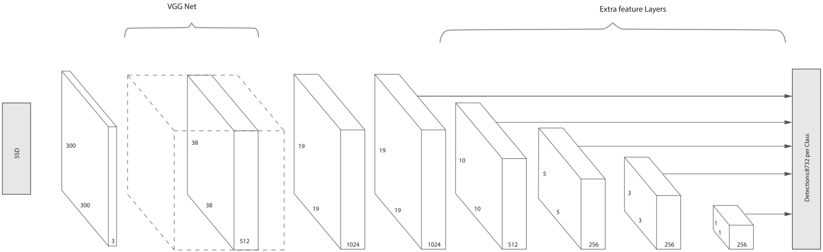


Figure 2: SSD architecture [Liu et al., 2016]

The default boxes

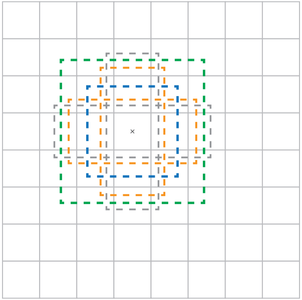


Figure 3: Default boxes as used in SSD. For every feature map (here 8×8) and at every feature map location center, we define 6 default boxes.

At the k^{th} feature map, we define the scale values s_k and s'_k.
For every aspect ratio value $a \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$, the default box has width and height:

$$\begin{cases} w = s_k\sqrt{a} \\ h = \frac{s_k}{a} \end{cases}$$

so that its area is $w \times h = s_k^2$.

Finally, we add the 1:1 default box of scale s'_k (the green one on figure 3).

Outline

1. Introduction
2. Related Works
   - Object Detection literature
   - Real-time trackers
3. My work
   - Reproducing SiamRPN
   - Other approaches
4. Results
5. Demo
6. Conclusion

Siamese Fully Convolutional network (SiamFC)

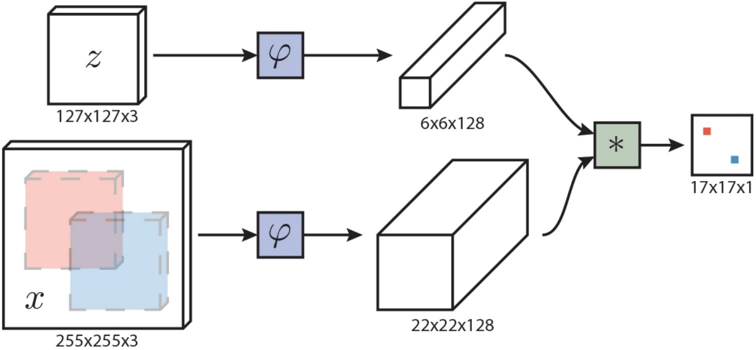


Figure 4: Siamese architecture [Bertinetto et al., 2016]

Siamese Region Proposal Network (SiamRPN)

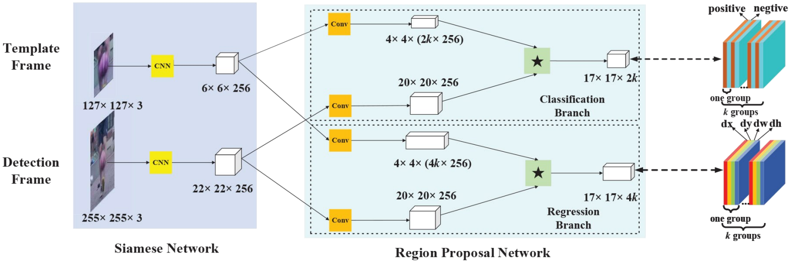


Figure 5: SiamRPN architecture [Li et al., 2018b]

Accurate Tracking by Overlap Maximization (ATOM)

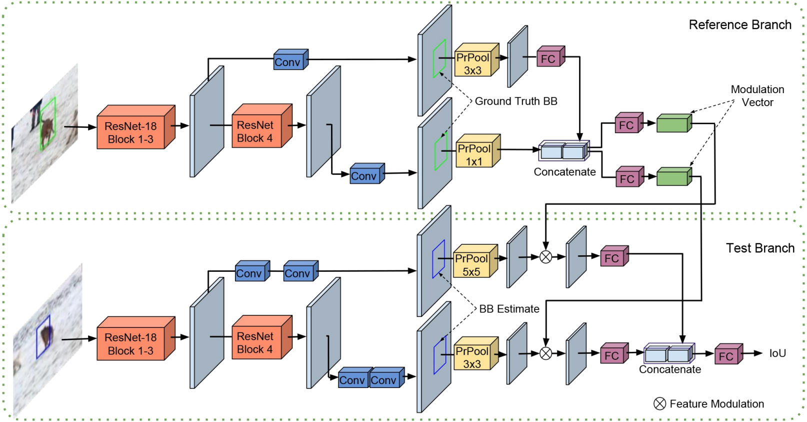


Figure 6: ATOM architecture [Danelljan et al., 2018]

SiamRPN++

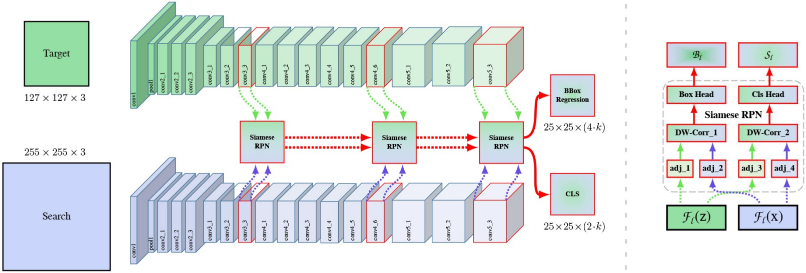


Figure 7: SiamRPN++ architecture [Li et al., 2018a]

(Submitted to arXiv.org on 31 Dec 2018!)

State-of-the-art on OTB-2015

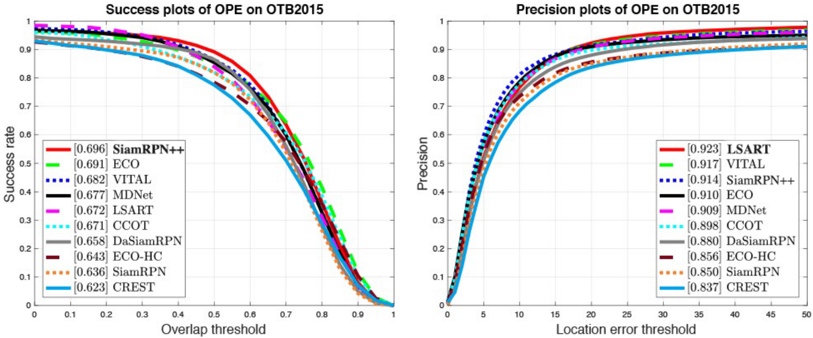


Figure 8: Comparison of the success and precision plots with the state-of-the-art trackers on the OTB-2015 dataset.

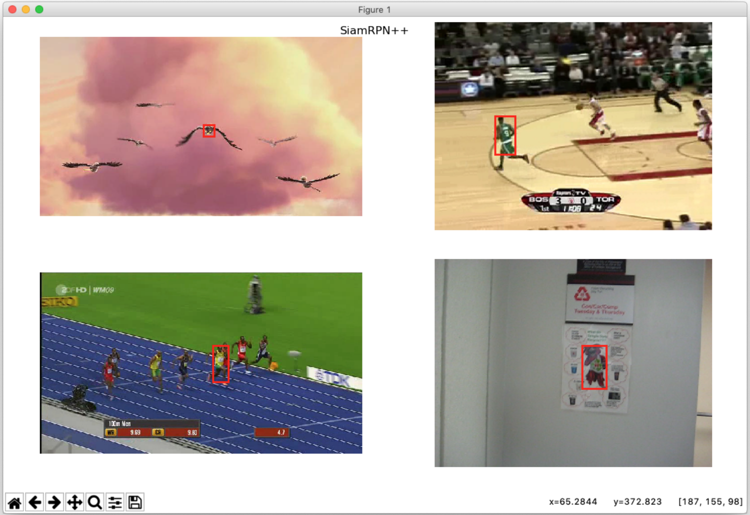
State-of-the-art on VOT2018

	DLSTpp	DaSiamRPN	SASiamR	CPT	DeepSTRCF	DRT	RCO	UPDT	SiamRPN	MFT	LADCF	ATOM	SiamRPN++
EAO	0.325	0.326	0.337	0.339	0.345	0.356	0.376	0.378	0.383	0.385	0.389	0.401	0.414
Acc.	0.543	0.569	0.566	0.506	0.523	0.519	0.507	0.536	0.586	0.505	0.503	0.590	0.600
Robust.	0.224	0.337	0.258	0.239	0.215	0.201	0.155	0.184	0.276	0.140	0.159	0.204	0.234

Table 1: Comparison with the state-of-the-art in terms of expected average overlap (EAO), accuracy and robustness (failure rate) on the VOT2018 benchmark.

SiamRPN++

[demo]



Outline

1. Introduction
2. Related Works
   - Object Detection literature
   - Real-time trackers
3. My work
   - Reproducing SiamRPN
   - Other approaches
4. Results
5. Demo
6. Conclusion

Training datasets

- TrackingNet [Müller et al., 2018]: 30,132 sequences (6 chunks / 12 were downloaded), 14,431,266 frames, 27 categories.
- ILSVRC-2015 video dataset [Russakovsky et al., 2015]: 3,862 / 555 train / validation videos, 1.3 million frames, 30 categories.
- COCO dataset [Lin et al., 2014]: 328,000 images, 2.5 million labeled instances, 91 categories.

COCO data augmentation

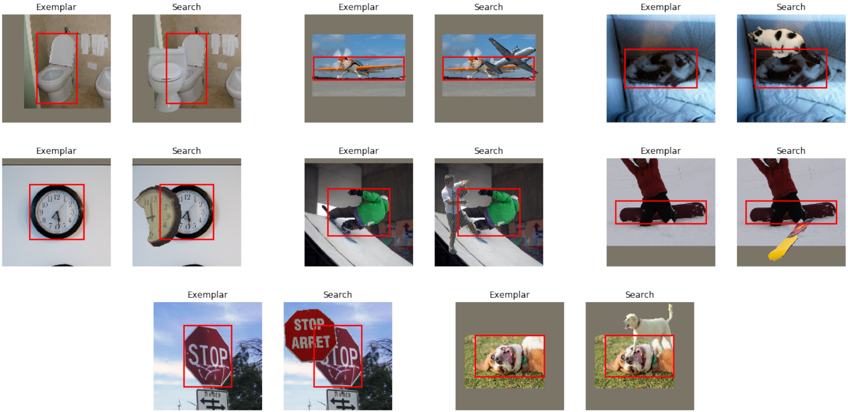


Figure 9: Some synthetic pairs including semantic distractors generated from the COCO dataset.

Image cropping

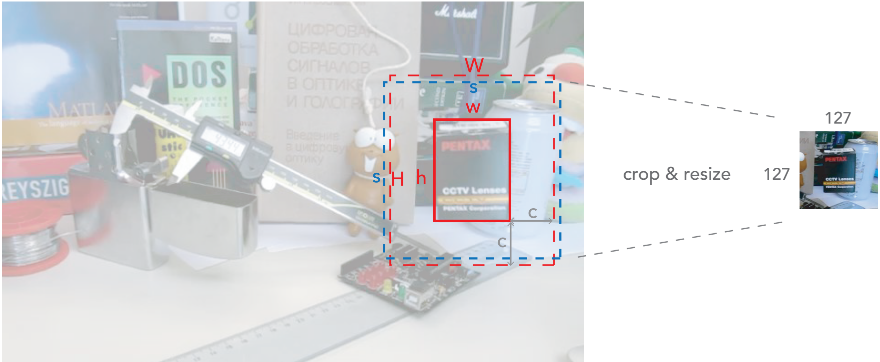


Figure 10: Image cropping: Given a bounding box (w, h) and a context amount (here 0.5), we compute the context $c = \texttt{context\_amount} \times (w + h)/2$. We then have $W = w + 2c, H = h + 2c$. The area to crop is the square of size $s = \sqrt{W \times H}$. Finally we resize the obtained region to 127 pixels.

The loss

- Similarly to SSD, we have the following variables:
  - D default boxes d_i ($i \in \{0, \ldots, D-1\}$): $\mathbf{d_i} = (d_i^{cx}, d_i^{cy}, d_i^{w}, d_i^{h})$
  - One ground-truth bounding-box: $\mathbf{g} = (g^{cx}, g^{cy}, g^{w}, g^{h})$
- For every default box index i, we further define:
  - The *normalized* ground-truth bounding-box: $\mathbf{\hat{g}_i}$:

$$\hat{g}_i{}^{cx} = (g^{cx} - d_i^{cx})/d_i^{w}, \quad \hat{g}_i{}^{cy} = (g^{cy} - d_i^{cy})/d_i^{h}$$

$$\hat{g}_i{}^{w} = \log(\frac{g^{w}}{d_i^{w}}), \quad \hat{g}_i{}^{h} = \log(\frac{g^{h}}{d_i^{h}})$$

  - The network output: confidence score $\mathbf{c_i} \in [0,1]$
    and offset location prediction $\mathbf{l_i} = (l^{cx}, l^{cy}, l^{w}, l^{h})$
  - The matching indicator:

$$x_i = \begin{cases} 1 & \text{if } \mathrm{IoU}(d_i, g) \geq \delta_{\text{high}} \quad \text{(positive match)} \\ 0 & \text{if } \mathrm{IoU}(d_i, g) \leq \delta_{\text{low}} \quad \text{(negative match)} \end{cases}$$

The loss

- We finally define the following loss:

$$L(x, c, l, g) = \frac{1}{N}(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g))$$

where

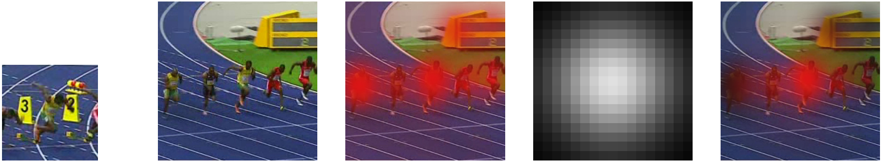
$$L_{\text{conf}}(x, c) = \text{BinaryCrossEntropyLoss}(c, x) \quad \text{and}$$

$$L_{\text{loc}}(x, l, g) = \sum_{i:x_i=1} \text{smooth}_{L1}(l_i - \hat{g}_i)$$

- Because of the heavy class imbalance (more negative matches than positives), we impose the ratio $\text{num}_{\text{negatives}}/\text{num}_{\text{positives}} = 3$.
- *Hard negative mining*: we choose the negative matches as the ones that contribute the most to the confidence loss.

Tracking engineering

- Similarly to SiamRPN [Li et al., 2018b], we use the following strategies during tracking:



(a) Exemplar (b) Search (c) Corr. map (d) Cosine window (e) Corr. map

Figure 11: Visualization of applying a cosine window to the correlation map. The confidence scores are then re-ranked in order to suppress large displacements.

Additionally, we penalize scale changes using the penalty $e^{k\max(\frac{r'}{r},\frac{r}{r'})\max(\frac{s'}{s},\frac{s}{s'})}$ where r and s represent the ratio and scale of the current prediction. The values of the last frame are noted with a prime symbol.

Some implementation details

- Developed using **PyTorch 0.4**
  - easier to debug than Tensorflow
  - more "Pythonic"
- Training visualization using **TensorboardX**
  - training curves
  - validation bounding boxes
- Model configuration management using **yacs**
  - readable .yaml config files
  - command-line overridable parameters

Remarks about SiamRPN

- Using correlation maps seems to work well for the confidence score.
- However, it is conceptually not clear why one could regress the bounding box from it.
- What's more, the ground-truth bounding box from the exemplar frame is used only to crop the image with the correct context amount. In particular, the ground-truth aspect ratio is not used.

Outline

1. Introduction
2. Related Works
   - Object Detection literature
   - Real-time trackers
3. My work
   - Reproducing SiamRPN
   - Other approaches
4. Results
5. Demo
6. Conclusion

Architectures: SiamConcatRPN

Inspired by *Fast Video Object Segmentation by Reference-Guided Mask Propagation* [Oh et al., 2018], we build the following network:

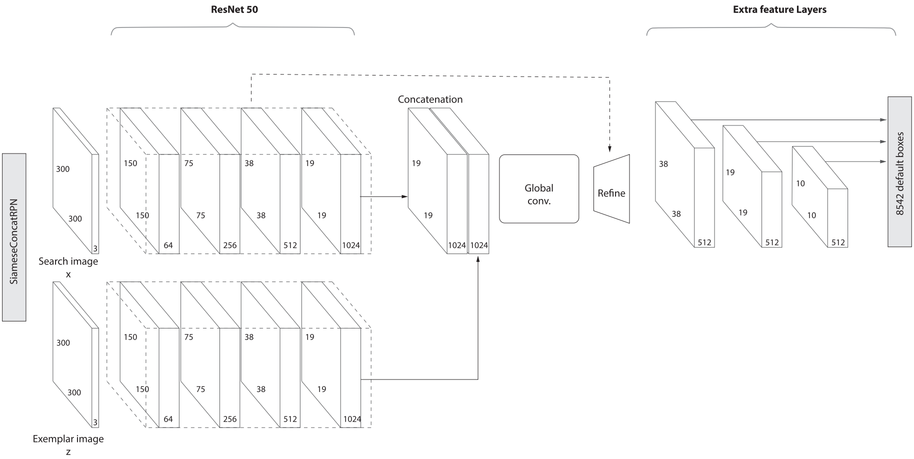


Figure 12: SiamConcatRPN architecture

Global convolution

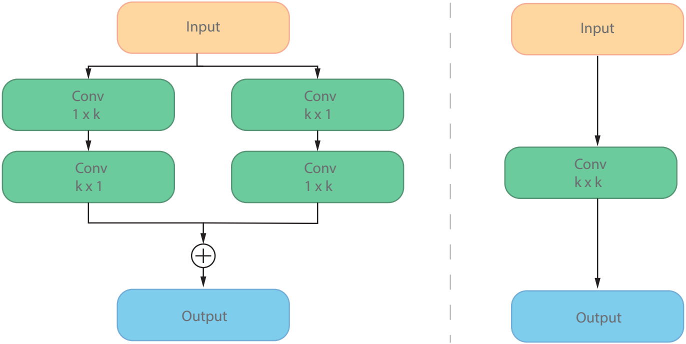
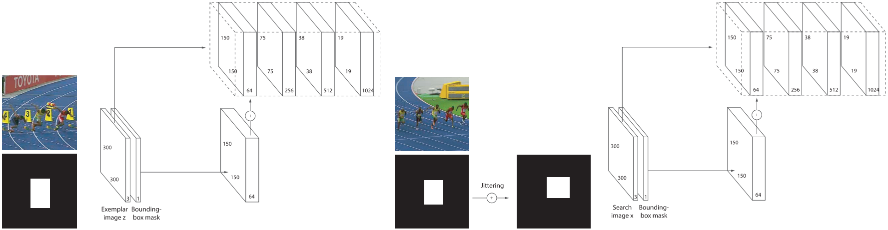


Figure 13: $k \times k$ Global convolution compared to a standard $k \times k$ convolutional layer.

Mask guide



(a) Exemplar frame (b) Search frame

Figure 14: Illustration of how the ground-truth exemplar and search bounding boxes are used in the SiamConcatRPN architecture to produce a binary mask. The latter is processed by a convolutional layer and added to the first layer of the ResNet network.

Remarks about SiamConcatRPN

- Relies only on convolutional layers to perform the matching.
- Missing a similarity map?

Architectures: SiamBroadcastRPN

Inspired by *Class-Agnostic Counting* [Lu et al., 2018], we build the following network:

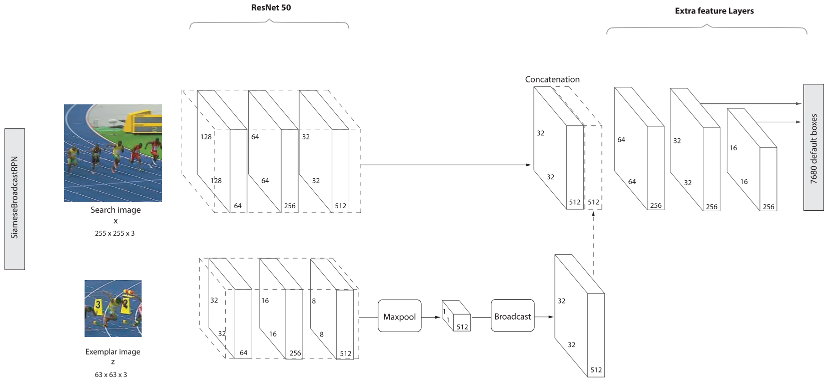


Figure 15: SiamBroadcastRPN architecture

Outline

1. Introduction
2. Related Works
   - Object Detection literature
   - Real-time trackers
3. My work
   - Reproducing SiamRPN
   - Other approaches
4. Results
5. Demo
6. Conclusion

Results on OTB2015

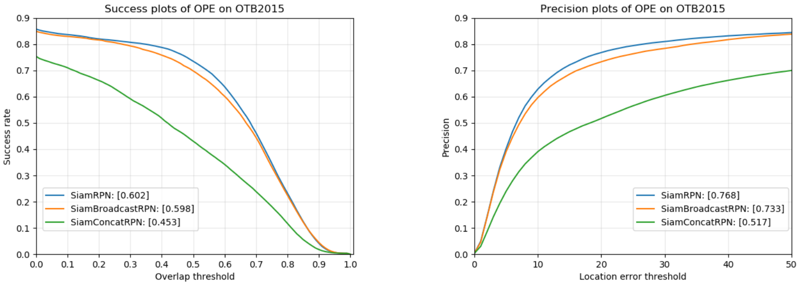


Figure 16: Success and Precision plots of the constructed networks on OTB-2015.

Results on OTB-2015

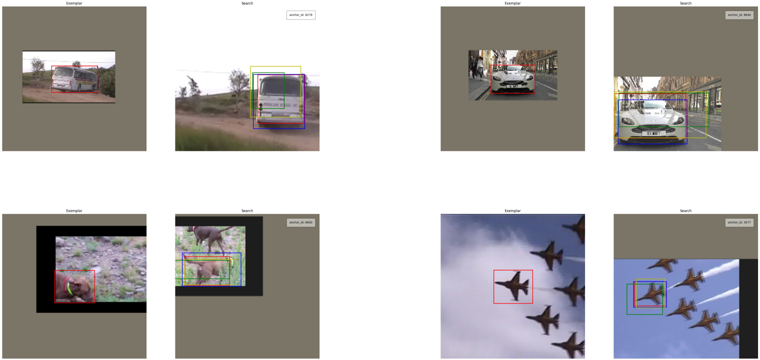


Figure 17: Validation results from the SiamConcatRPN model after training. Each image pair corresponds to an exemplar and search frame. In the search image, the bounding boxes correspond to: the ground-truth (in blue), the predicted box (in red), the best default box (in yellow), the jittered guide (in green).

Outline

1 Introduction

2 Related Works
- Object Detection literature
- Real-time trackers

3 My work
- Reproducing SiamRPN
- Other approaches

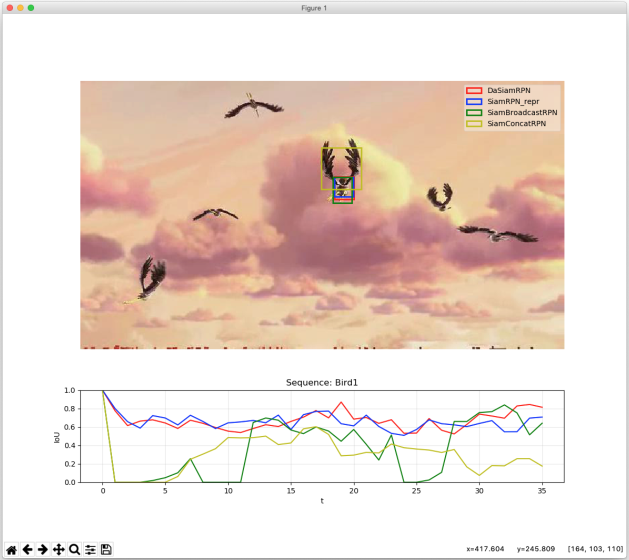
4 Results

5 Demo

6 Conclusion

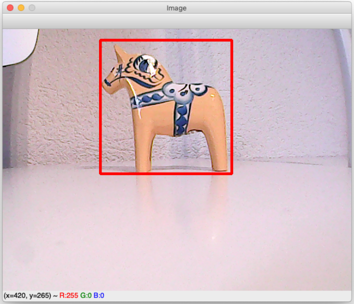
Example sequences

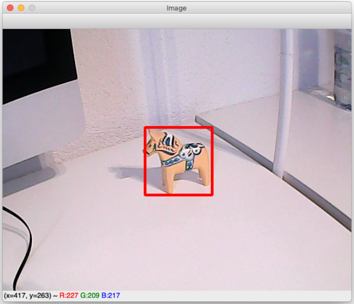
[demo]



Interactive demo

[demo]



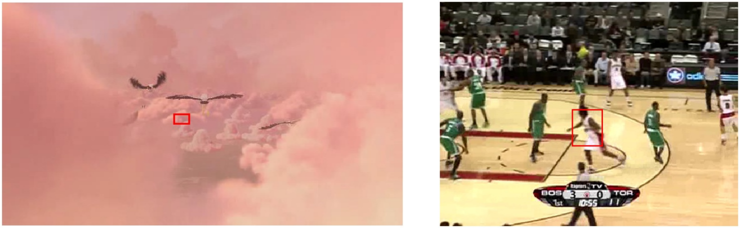


Conclusion

- Training a state-of-the-art deep tracker is hard.
- A very enriching experience (my first real-world application of the classes I took last year, like CS-433 Machine Learning and EE-559 Deep Learning).
- In the process, I learned a lot about object detection / object tracking and writing deep learning code.

Conclusion

There is still room for improvement!



(a) Bird1 (b) Basketball

Figure 18: Failures of SiamRPN++ on the OTB-2015 dataset.

Thanks for your attention!

References I

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. (2016).
Fully-convolutional siamese networks for object tracking.
In *European conference on computer vision*, pages 850–865. Springer.

Danelljan, M., Bhat, G., Khan, F. S., and Felsberg, M. (2018).
Atom: Accurate tracking by overlap maximization.
arXiv preprint arXiv:1811.07628.

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., and Yan, J. (2018a).
SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks.
arXiv preprint arXiv:1812.11703.

References II

Li, B., Yan, J., Wu, W., Zhu, Z., and Hu, X. (2018b).
High Performance Visual Tracking With Siamese Region Proposal Network.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014).
Microsoft coco: Common objects in context.
In *European conference on computer vision*, pages 740–755. Springer.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016).
Ssd: Single shot multibox detector.
In *European conference on computer vision*, pages 21–37. Springer.

References III

Lu, E., Xie, W., and Zisserman, A. (2018).
Class-agnostic counting.
arXiv preprint arXiv:1811.00472.

Milan, A., Leal-Taixé, L., Reid, I., Roth, S., and Schindler, K. (2016).
Mot16: A benchmark for multi-object tracking.
arXiv preprint arXiv:1603.00831.

Müller, M., Bibi, A., Giancola, S., Al-Subaihi, S., and Ghanem, B. (2018).
Trackingnet: A large-scale dataset and benchmark for object tracking in the wild.
arXiv preprint arXiv:1803.10794.

References IV

Oh, S. W., Lee, J.-Y., Sunkavalli, K., and Kim, S. J. (2018).
Fast video object segmentation by reference-guided mask propagation.
In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7376–7385. IEEE.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. (2016).
A benchmark dataset and evaluation methodology for video object segmentation.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015).
Imagenet large scale visual recognition challenge.
International Journal of Computer Vision, 115(3):211–252.