CS537 Final Project Report

Fanghai Ge, Mingzhao Liu, Da Lin

1. Problem statement

In this final project, we selected single target object tracking as the theme. The main purpose of the project is Given a bounding box of the target object in the first video frame track the target in the subsequent video frames.

We use the VOT2015 data set to test the algorithm performance. In the vot2015 data set, it has 60 short sequences and 11 attributes related to target tracking. The 11 attributes can evaluate different performances of program and these attributes also have more challenges for our program we need to solve more problems. The fast movement of objects will have a considerable impact on the algorithm. The fast motion is mainly the boundary effect, and the wrong samples generated by the boundary effect will cause the classifier to have insufficient discriminative power.

At present, we use the HCF algorithm to complete the single target tracking task, and we use VGG-19 witch has 19 layers to train the ImageNet datasets.

Vision tracking technology is an important subject in the field of computer vision and has wide application prospects in many aspects such as video surveillance, robot visual navigation, human-computer interaction, and medical diagnosis. In recent years, the target tracking technology has made breakthrough progress due to the combination of neural network algorithms and related filters, and has three criteria for robustness, accuracy, and stability.

2. Approach

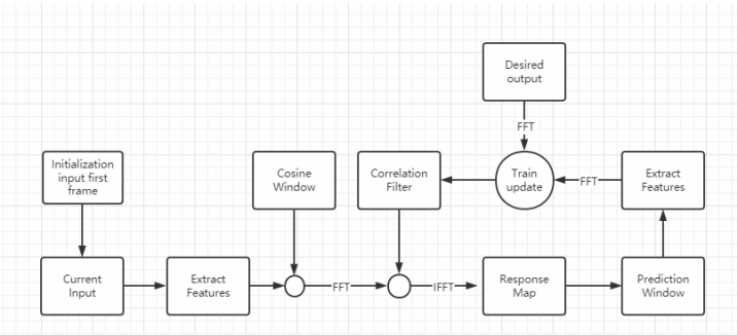


Figure 1. Components of HCF

In this approach, first, we use CNN to detect the deep feature for all frame and use correlation filter to find the object in other frames which same as the first frame. Using deep learning can better extract the characteristics of the target and express the target better. Low-level features have higher resolution and can accurately locate the target. High-level features contain more semantic information, can handle larger target changes, and prevent tracker drift, and can target the range.

Pseudocode:

Algorithm 1: Proposed tracking algorithm

Input: Initial target position P_0

Output: Estimated object position $P_t = (x_t, y_t)$, and learned correlation filters $\{w_t^l\}$

Repeat

Crop out the searching window in frame t centered at (x_{t-1}, y_{t-1}) and extract convolutional features with spatial interpolation

foreach layer l **do** computing confidence score f_l using $f_l = F^{-1}\left(\sum_{d=1}^{D} w^d \odot \bar{z}^d\right)$;

Coarse-to-fine estimate the new position (x_t, y_t) on response map set $\{f_l\}$

Crop out new patch centered at $P_t = (x_t, y_t)$ and extract convolutional features with interpolation.

foreach layer l **do** updating correlation filters $\{w_t^l\}$.

Until End of video sequences.

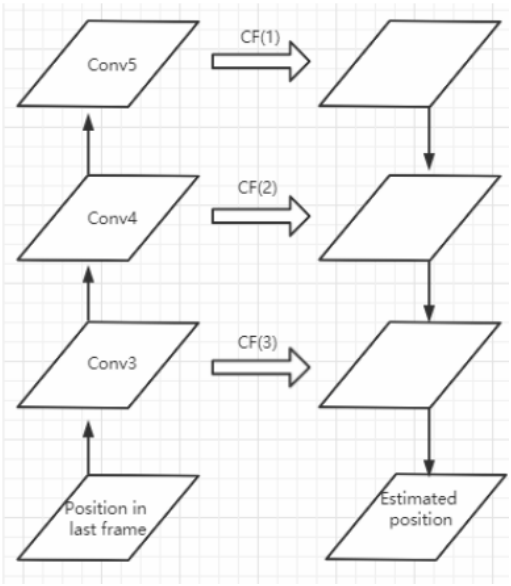


Figure 2. Step of training layers

The features in conv3 can clearly see the outline of the target, but it is difficult to see the details in conv5. However, what can be seen in conv5 is that this layer of features contains more semantic information. Among the features, we can easily find the approximate area of the target based on the semantic information of the extracted features, and then gradually use the lower-level features to accurately locate the target.

3. Evaluation

Implementation details:

Libraries: Pytorch, OpenCV, NumPy, matplot, VGG-16, ImageNet, VOT2015.

Open-source code: CF2 url: https://github.com/jbhuang0604/CF2.git.

https://github.com/vtddggg/Visual_Tracking_api.git

Epoch: 20. We test performance of distance precision (DP) rate at a threshold of 50 pixels and overlap success rate at an overlap threshold of 0.5.

Dataset(s) :

VOT 2015 DATASET

It has 60 short sequences and 11 attributes related to target tracking, including illumination changes, scale changes, occlusion, deformation, motion blur, fast motion, inplane rotation, out-of-plane rotation, out-of-field, background interference, and low pixels.

Ground truth: the VOT 2015 dataset support the ground truth that include correct target bounding box points of each frame in the sequence.

Evaluation matrix:

- Distance precision (DP) rate at a threshold of 20 pixels: Accuracy is measured with the Intersection Over Union for each frame: we compute the Intersection of the area of the predicted bounding box, and the area of the ground-truth bounding box and divide by the Union of the two areas.

- center location error (CLE): One way to measure robustness is to count the number of times a tracker loses the center location throughout an entire sequence, by reinitializing the tracker each time this happens.

- overlap success (OS) rate at an overlap threshold of 0.5.

Simpler versions of your approach:

We just use VGG net in torchvision lib to get deep feature and use ground truth to find target in first frame then use one to one match to find the most closed (smallest distance) position in other frames and find the target box around the matching position. Then we get a simpler result. We call it FT (Features Tracker) and we call the full approach HCF (Hierarchical Convolutional Features with Correlation Filter)

Table 1. Comparisons with state-of-the-art trackers. Our approach performs favorably against existing methods in distance precision (DP) rate at a threshold of 50 pixels, overlap success (OS) rate at an overlap threshold of 0.5 and center location error (CLE).

	HCF	FT	DLT	KCF	Struck	SCM	LSHT	CSK	TLD	MEEM	TGPR
DP	45.3	20.5	54.8	74.1	65.6	64.9	56.1	54.5	60.8	83.0	70.5
OS	32.8	15.8	47.8	62.2	55.9	61.6	45.7	44.3	52.1	69.6	62.8
CLE	73.2	85.6	65.2	35.5	50.6	54.1	55.7	88.8	48.1	20.9	51.3

We can find HCF is better than FT because we use correlation filter because it can find relation between two frames. The more similar the features, the higher the correlation. Then we can use this filter to find the target object

Training runtime: average 00:57 each Epoch.

Test runtime: 04:28 each sequence.

Computing hardware:

VGA compatible controller: NVIDIA Corporation GP102 [GeForce GTX 1080 Ti]

Memory at 91000000 (32-bit, non-prefetchable) [size=16M]

Memory at 3bfe0000000 (64-bit, prefetchable) [size=256M]

Memory at 3bff0000000 (64-bit, prefetchable) [size=32M]

I/O ports at 2000 [size=128]

[virtual] Expansion ROM at 92080000 [disabled] [size=512K]

Kernel driver in use: Nvidia

Kernel modules: nouveau, Nvidia, Nvidia

Qualitative evaluation:

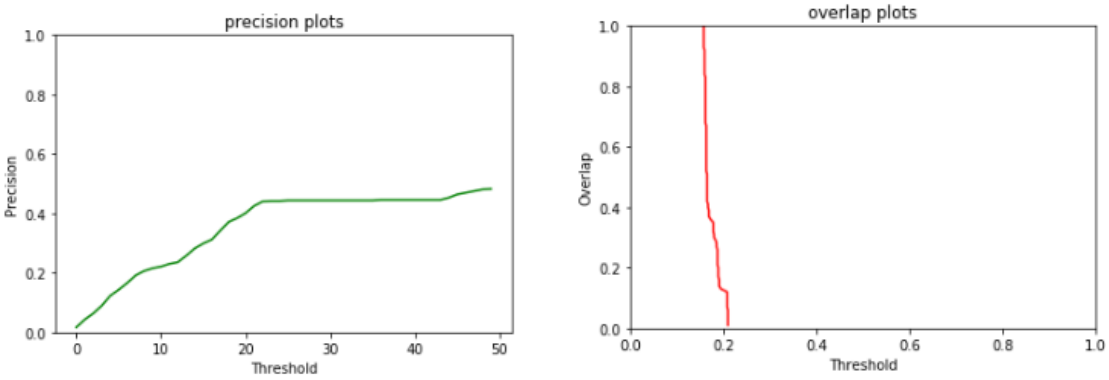
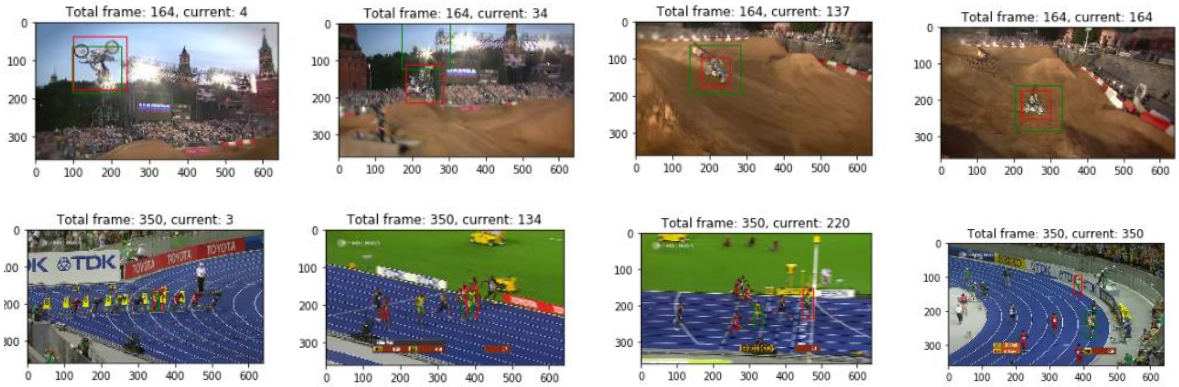


Figure 3. figure of precision and overlap plots



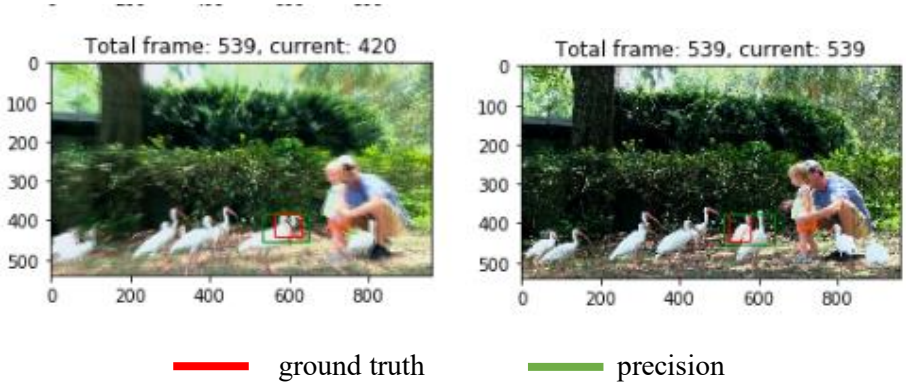


Figure 4. Qualitative evaluation of the proposed algorithm, on three challenging sequences (from top to down are Motocross1, Bolt1, Birds2)

These three sequences evaluated fast motion, rotation, motion blur, background interference and out-of-field. We can find in Bolt1 sequence we can find at first the precision is well but then the motion are fast to much it has not good performance and same as rotation. In the birds2 sequence there has motion blur, background interference and out-of-filed. We can find the precision is well in motion blur and out-of-filed but has not good performance when the sequence has background interference. Because we do not train correlation filter with new frame precision.

4. labor distribution

Fanghai Ge, Mingzhao Liu, Da Lin
Find data sets and do basic processing: Da Lin
Code of approach 1: Lin Da
Code of approach 2: Mingzhao Liu, Fanghai Ge
Compare the results of this project with the results of existing open source code and provide data analysis: Fanghai Ge
Final PPT and reports: Mingzhao Liu, Fanghai Ge

Reference

1. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., et al.: The visual object tracking vot2015 challenge results. In: ICCV2015 Workshops, Workshop on visual object tracking challenge. (2015)
2. C. Ma, J. Huang, X. Yang and M. Yang, "Hierarchical Convolutional Features for Visual Tracking," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 3074-3082, doi: 10.1109/ICCV.2015.352.