Target Tracking Algorithm Based on a Broad Learning System

Dan Zhang[1,2], Tieshan Li[1,5], C. L. Philip Chen[1,3,4], He Yang[1]

[1]Navigation College, Dalian Maritime University, Dalian 116026, China;
[2]Innovation and Entrepreneurship Education College, Dalian Minzu University, Dalian 116600, China;
[3]Computer Science and Engineering College, South China University of Technology, Guangzhou 510641, China;
[4]Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 99999, China;
[5]School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Science China Information Sciences

Contents

Background

- Target tracking is a common and difficult task. During the tracking process, target variations are dramatic in terms of scale and position. Additionally, target signals are subject to interference such as occlusion, illumination changes, and background clutter.

- Although deep learning performs well at target tracking, real-time tracking must be improved in terms of its computational cost.

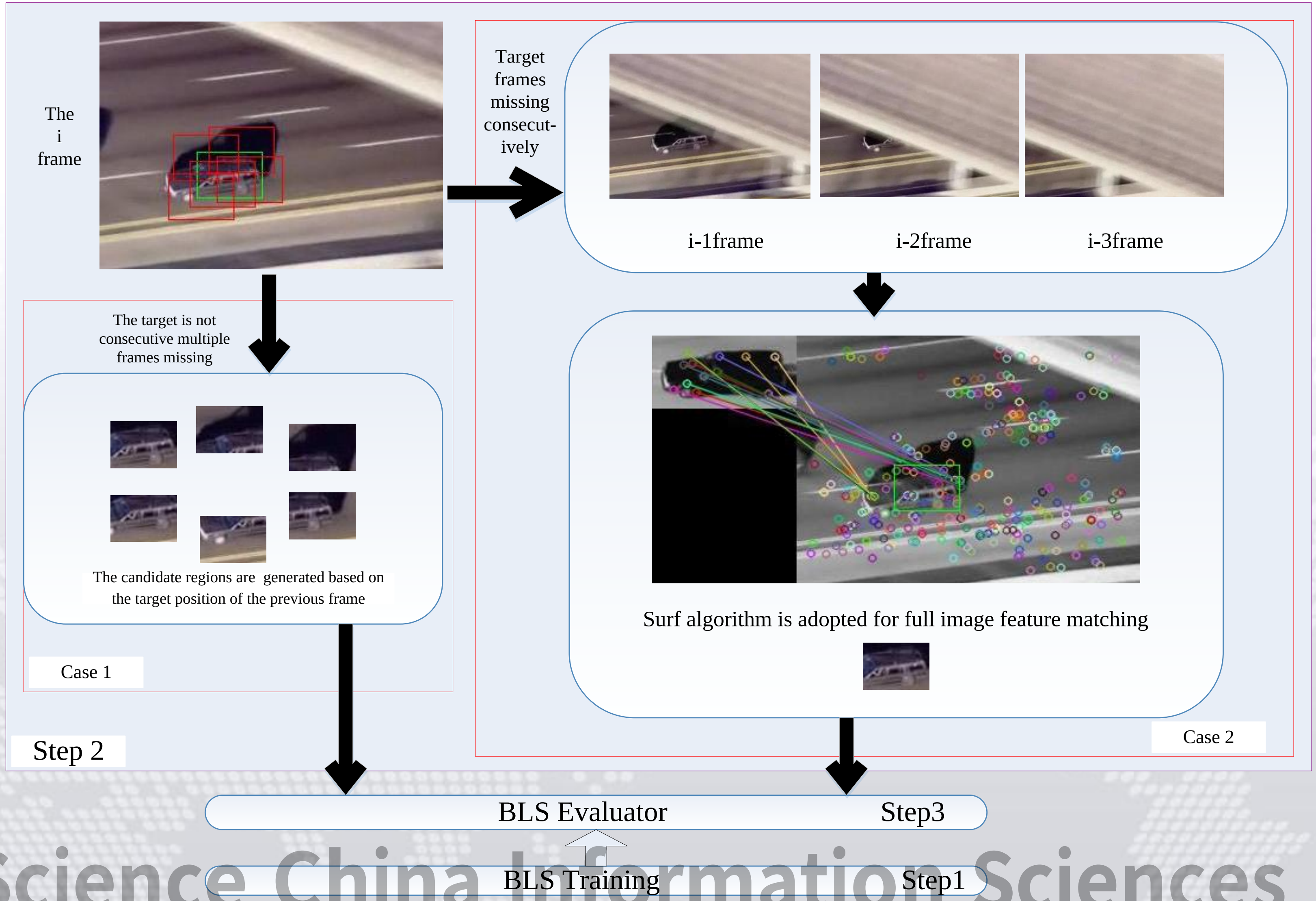
Background

- Broad learning system (BLS) was proposed in 2017-2018.

- As an alternative to a deep network architecture, its calculation speed is very fast.

- Additionally, the BLS can extract sparse features from training data and sparse feature learning models are attractive for exploring essential characterization.

Target tracking based on BLS



Flowchart of target tracking based on BLS

Step 1: Evaluator training based on BLS

- The tracking dataset for training is X , $X \in \mathbb{R}^{N\times M}$. ϕ_i is the transformation, and the i th mapped feature is denoted as Z_i. Then, $Z_i = \phi_i(XW_{ei} + \beta_{ei})$, $i = 1, 2\cdots, n$. Matrix $Z^n \equiv [Z_1, \cdots, Z_n]$ indicates that n groups of feature nodes. The j th group of enhancement nodes is denoted as H_j , then: $H_j = \xi_j(Z^n W_{hj} + \beta_{hj})$, $j = 1, 2, \cdots, m$. Matrix $H^m \equiv [H_1, \cdots, H_m]$ is used to denote group m of the enhancement nodes.

- The output is set Y , then, the broad learning model can be written as:

$$\begin{aligned} Y &= \left[Z_1, \cdots, Z_n \middle| \xi(Z^n W_{h1} + \beta_{h1}), \cdots, \xi(Z^n W_{hm} + \beta_{hm})\right] W_m \\ &\left[Z_1, \cdots, Z_n \middle| H_1, \cdots, H_m\right] W_m \\ &\left[Z^n \middle| H^m\right] W_m \end{aligned}$$

- $W_m = \left[Z^n \middle| H^m\right]^+ Y$, W_m are the connection weights of the BLS. W_m are computed using the ridge regression approximation, setting $\left[Z^n \middle| H^m\right]$ to A , for the pseudoinverse we have that:

$$A^+ = \lim_{\lambda \to 0} (\lambda I + AA^T)^{-1} A^T$$

Step 2: Multi-cue target tracking frame

- Case 1: If the object is not occluded or lost for a long period of time, to consume less resources, the positions around the target object are selectively searched. After removing any windows intersecting the image border, the remaining windows are selected as candidate regions.

- Case 2: If the target is occluded or lost for a long time (five frames), then feature matching based on the surf algorithm and a full-image target search are performed to ensure accuracy.

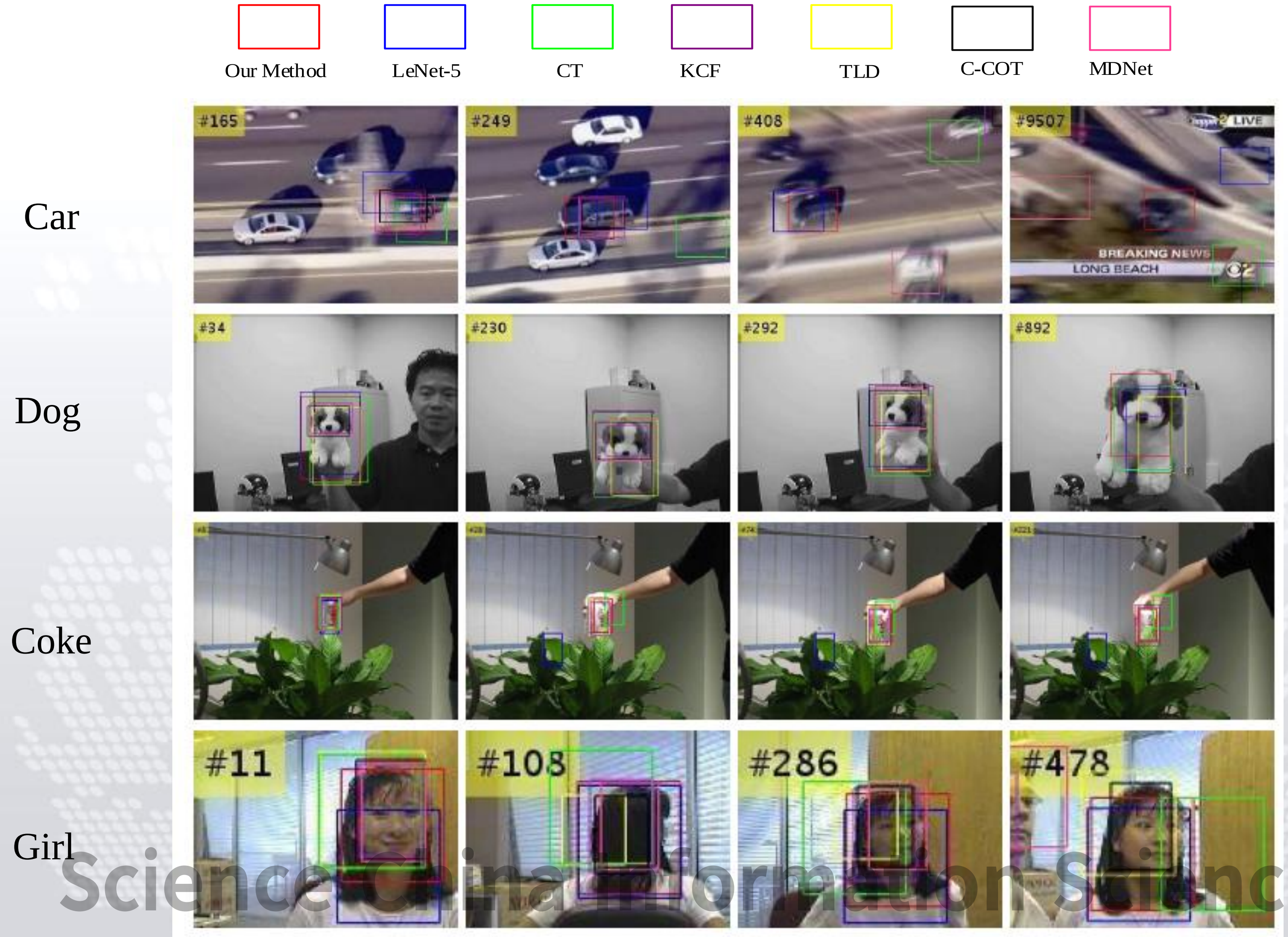
- Thus, the multiple clues tracking system is constituted which combined the two methods. The proposed method can adapt to target tracking with deformation, occlusion, and loss while maintaining acceptable tracking speed.

Step 3: Evaluating candidate regions

- The third step in our method is to evaluate the candidate regions and select the window with the highest evaluation score as the position of the target.

- However, if the scores of all candidate regions are very low (less than a predetermined threshold), then our method judges that the target is lost and counts the number frames in which the target is lost.

Experiments-results



Tracking effectiveness on four datasets

Experiments-results

Real-time performance of compared methods

Video sets / methods	Car(s)		Coke(s)		Dog(s)		Girl(s)	
	training time	tracking time	training time	tracking time	training time	tracking time	training time	tracking time
Ours	0.29	918.55	0.22	49.21	1.09	197.98	0.96	69.73
LeNet-5	95.99	1063.90	12.75	64.04	104.25	307.40	31.61	96.27
C-COT	NAN	6787.8	NAN	199.43	NAN	920.14	NAN	308.85

Conclusions and discussion

- Tracking adaptability: First, we trained an accurate information evaluator. BLS is the process of acquiring sparse features. Sparse feature learning models are attractive for exploring the essential characteristics of tracking data. Based on statistical target occlusion and loss, we can adjust the selective search and surf feature matching. By using such a tracking strategy, we can effectively enhance tracking adaptability.

- Time consumption: Our method has a small time overhead because BLS has few parameters and is solved using ridge regression.

- Considering tracking speed as an example, there is still room for further improvement. Additionally, robustness to long-term occlusion and loss, as well as scale variation, must be improved. Follow-up research will focus on these issues and additional studies will be required to develop faster and more robust online target tracking systems.

Thanks!