Learning Multi-Domain Convolutional Neural Networks for Visual Tracking (MDNet)

CVPR 2016

2019.01.21

발표자 박성현

DAVIAN
Data and Visual Analytics Lab
KOREA UNIVERSITY

1 Introduction

Single Object Tracking

[Single Object Tracking]

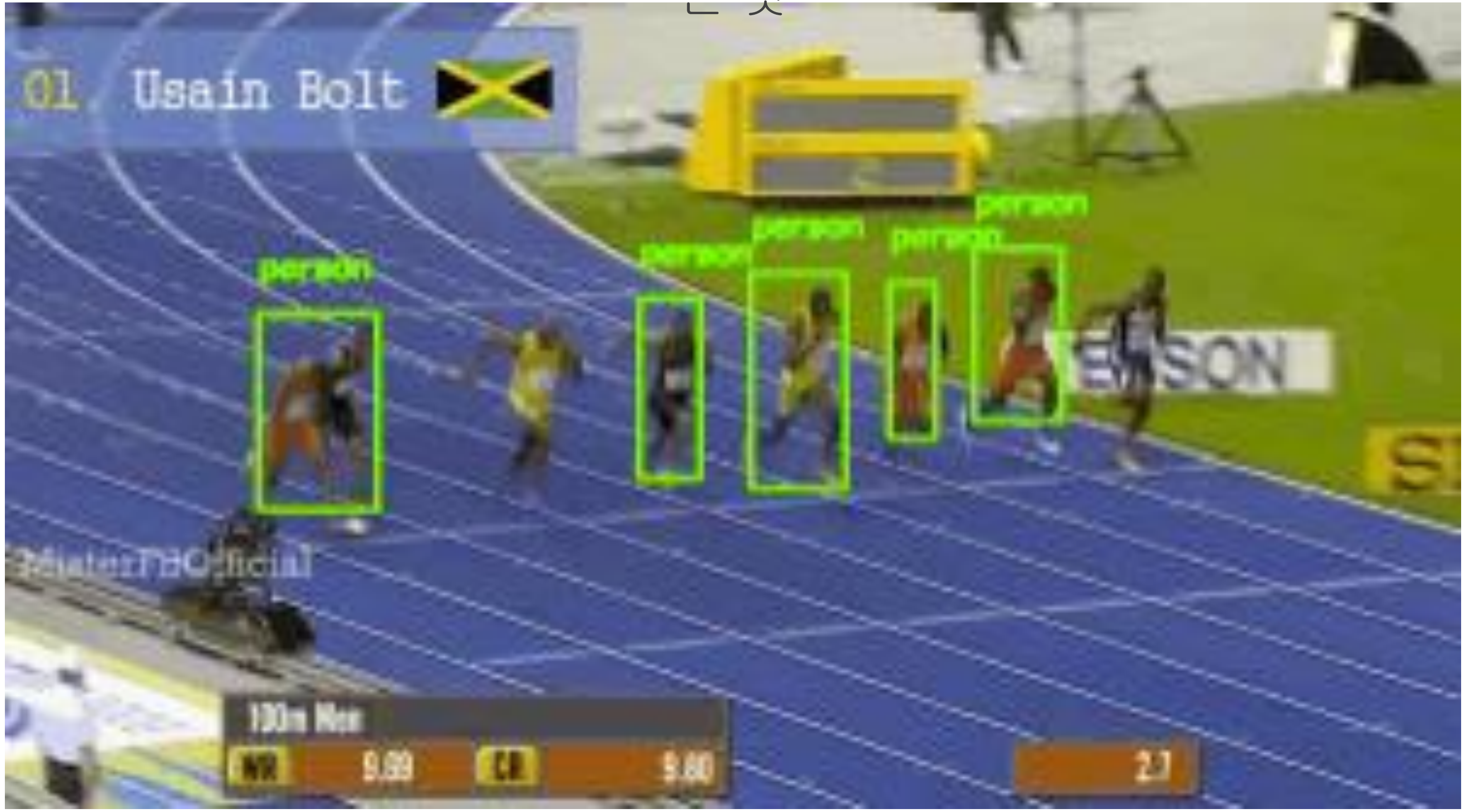
영상의 첫 프레임에서 추적하는 물체의 위치 정보가 주어지고, 이 물체를 추적하
는 것

1 Introduction

Multi Object Tracking

[Multi Object Tracking]

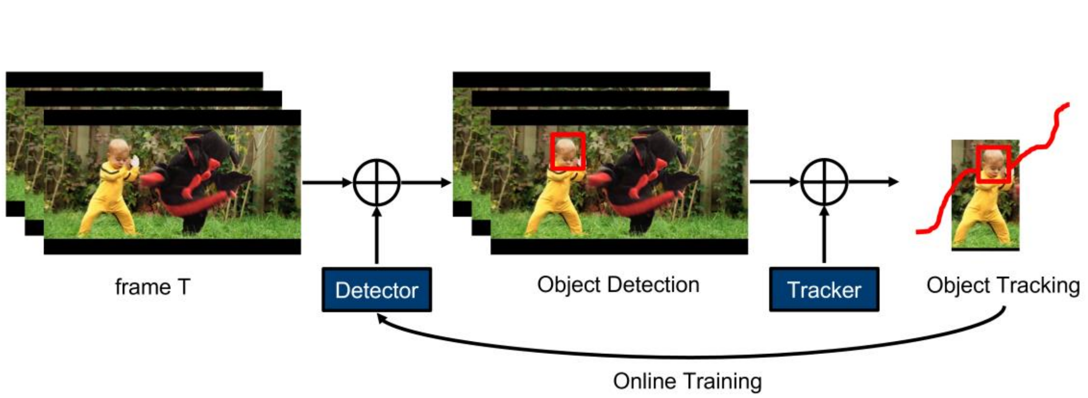
영상의 첫 프레임에서 추적하는 여러 물체들의 위치 정보가 주어지고, 여러 물체를 추적하는 것



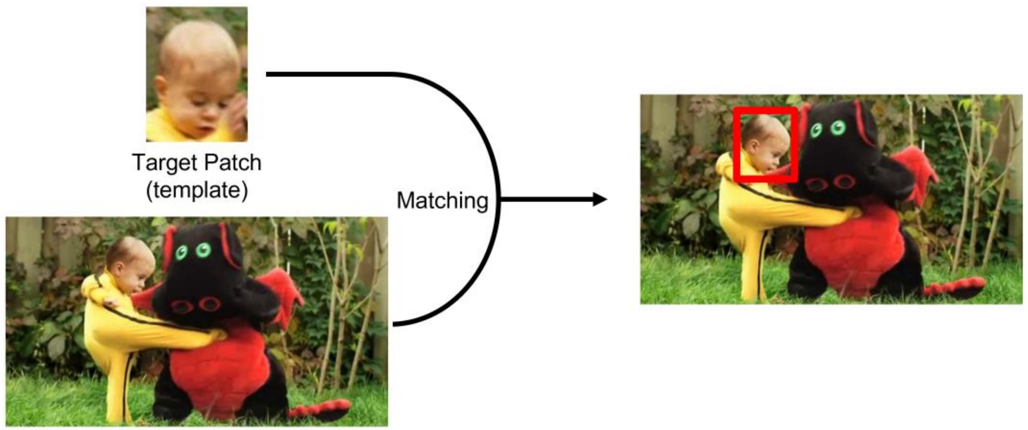
1 Introduction

Tracking Algorithm

	Tracking by Detection	Template Matching
Pros	Adapt to changing target	Real time tracker
Cons	Slow(< 2 fps)	No adapt to changing target
example	MDNet(CVPR 2016), ADNet(CVPR 2017)	siameseFC(ECCV 2016), SA-Siam(CVPR 2018), RASNet(CVPR 2018)



[Tracking by Detection]



[Template Matching]

2 Model

MDNet

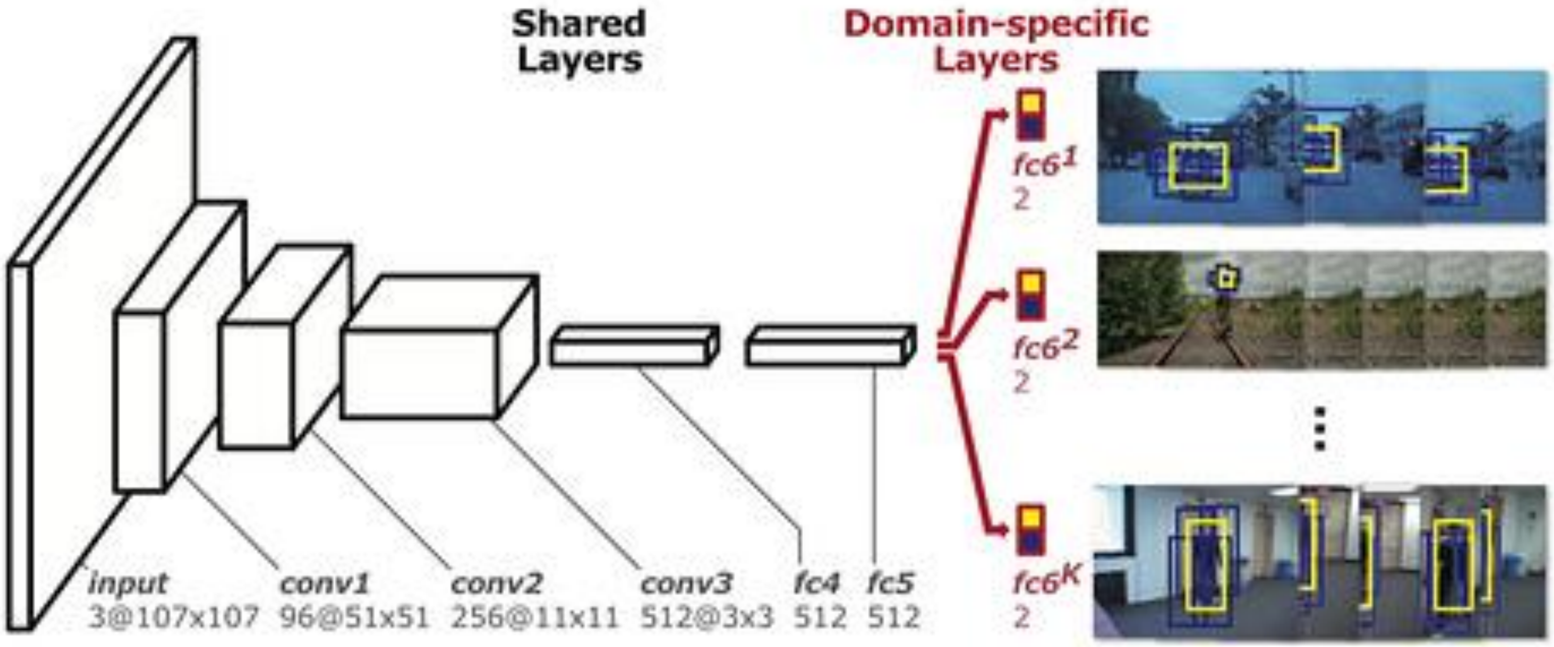


Figure 1: The architecture of our Multi-Domain Network, which consists of shared layers and K branches of domain-specific layers. Yellow and blue bounding boxes denote the positive and negative samples in each domain, respectively.

2

Model

MDNet

▶ Model 설명

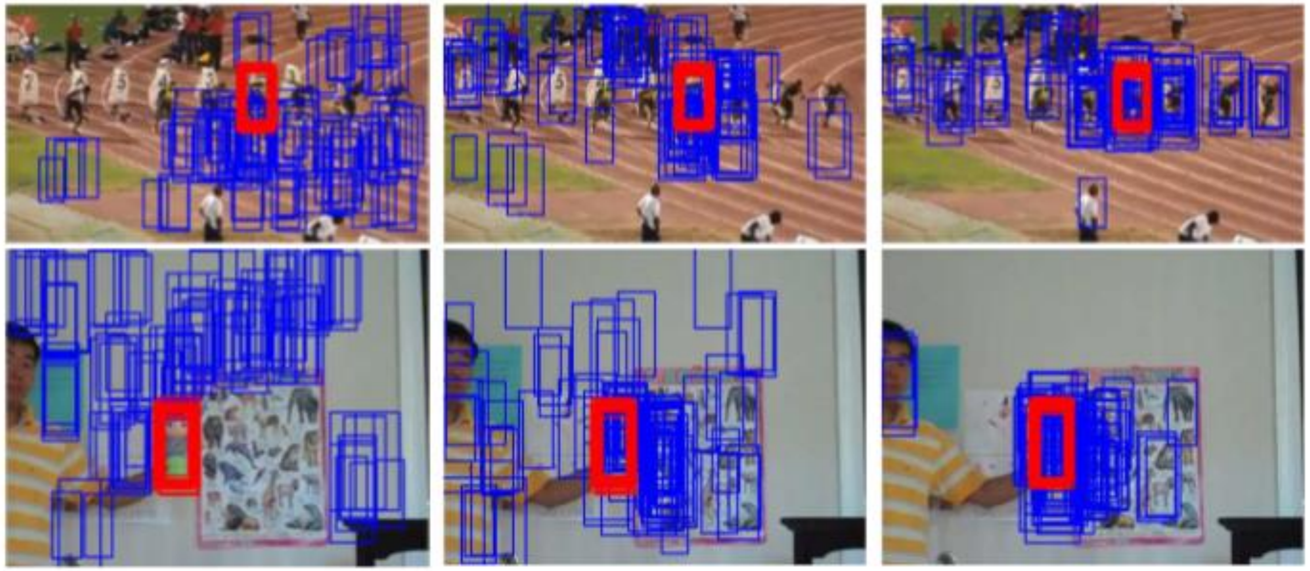
3개의 Conv Layer와 2개의 FC Layer로 이루어져 있습니다. 그 위에 K개의 Branch가 각각의 도메인에 맞게 나누어져 있습니다. 각 Branch는 해당 물체인지 아닌지를 판별하는 Binary Classification을 수행합니다.

▶ 작은 네트워크로 설계한 이유

(1) 트래킹 자체가 타겟과 배경을 구분하는 쉬운 문제임.
(2) 레이어가 많아지면 target localization에 상대적으로 취약한 부분도 있음.
(Spatial information이 layer가 깊어질수록 희석됨)
(3) Tracking 대상이 일반적으로 작다는 점
(4) 온라인 러닝에 더욱 효율적이라는 점에서 장점이 있음.

2 Model

Positive & Negative Samples



(a) 1^{st} minibatch (b) 5^{th} minibatch (c) 30^{th} minibatch

Figure 2: Identified training examples through our hard negative mining in *Bolt2* (top) and *Doll* (bottom) sequences. Red and blue bounding boxes denote positive and negative samples in each minibatch, respectively. The negative samples becomes hard to classify as training proceeds.

2 Model

Online Tracking Algorithm

Algorithm 1 Online tracking algorithm

Input : Pretrained CNN filters $\{\mathbf{w}_1, \ldots, \mathbf{w}_5\}$
Initial target state $\mathbf{x}_1$
Output: Estimated target states $\mathbf{x}_t^*$

Randomly initialize the last layer $\mathbf{w}_6$.
Train a bounding box regression model.
Draw positive samples S_1^+ and negative samples S_1^-.
Update $\{\mathbf{w}_4, \mathbf{w}_5, \mathbf{w}_6\}$ using S_1^+ and S_1^-;
$\mathcal{T}_s \leftarrow \{1\}$ and $\mathcal{T}_l \leftarrow \{1\}$.
repeat
  Draw target candidate samples $\mathbf{x}_t^i$.
  Find the optimal target state $\mathbf{x}_t^*$ by Eq. (1).
  if $f^+(\mathbf{x}_t^*) > 0.5$ **then**
    Draw training samples S_t^+ and S_t^-.
    $\mathcal{T}_s \leftarrow \mathcal{T}_s \cup \{t\}, \mathcal{T}_l \leftarrow \mathcal{T}_l \cup \{t\}$.
    if $|\mathcal{T}_s| > \tau_s$ **then** $\mathcal{T}_s \leftarrow \mathcal{T}_s \setminus \{\min_{v \in \mathcal{T}_s} v\}$.
    if $|\mathcal{T}_l| > \tau_l$ **then** $\mathcal{T}_l \leftarrow \mathcal{T}_l \setminus \{\min_{v \in \mathcal{T}_l} v\}$.
    Adjust $\mathbf{x}_t^*$ using bounding box regression.
  if $f^+(\mathbf{x}_t^*) < 0.5$ **then**
    Update $\{\mathbf{w}_4, \mathbf{w}_5, \mathbf{w}_6\}$ using $S_{v \in \mathcal{T}_s}^+$ and $S_{v \in \mathcal{T}_s}^-$.
  else if $t \bmod 10 = 0$ **then**
    Update $\{\mathbf{w}_4, \mathbf{w}_5, \mathbf{w}_6\}$ using $S_{v \in \mathcal{T}_l}^+$ and $S_{v \in \mathcal{T}_s}^-$.
until end of sequence

- **Target Candidate Generation**

N(=256)개의 sample 생성
(Previous frame의 target의 width와 height로 생성)

- **Training Data**

Positive sample은 0.7이상의 IoU
Negative sample은 0.3 이하의 IoU
1:4의 비율로 학습 진행

- **Hard Negative Mining**

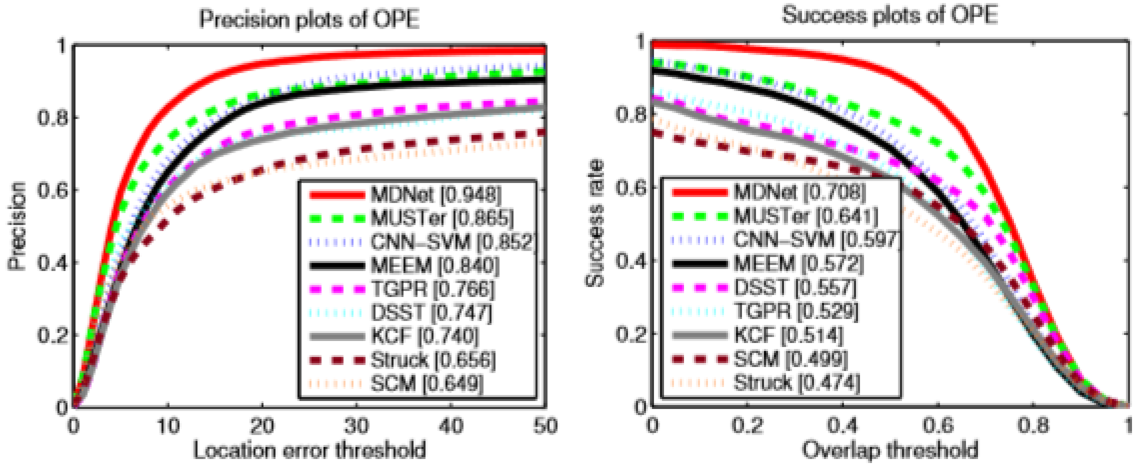
Negative sample들 중 Positive로 점수가 높은 것

- **Network Learning**

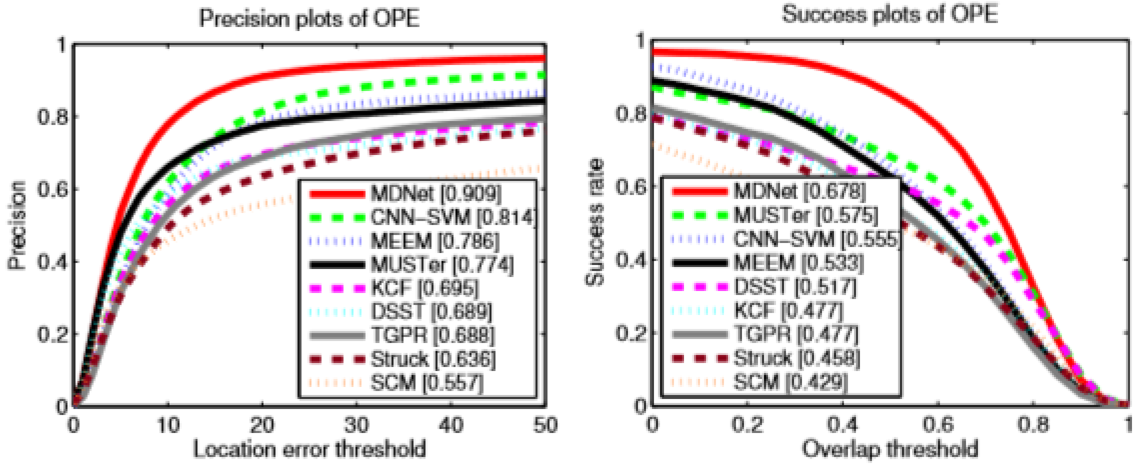
새로운 Last FC Layer를 먼저 학습시키고, Test 진행
(10 or 30 Iteration 학습 진행)
Bounding Box Regression을 추가해서 정밀하게 조정

3 Experiment

OTB Dataset



(a) OTB50 result



(b) OTB100 result

3

Experiment

VOT Dataset

3 Experiment

VOT Dataset

Tracker	Accuracy Score	Accuracy Rank	Robustness Score	Robustness Rank	Combined Rank
MUSTer	0.58	4.50	0.99	5.67	5.09
MEEM	0.48	7.17	0.71	5.50	6.34
DSST	0.60	4.03	0.68	5.17	4.60
SAMF	0.60	3.97	0.77	5.58	4.78
KCF	0.61	3.82	0.79	5.67	4.75
DGT	0.53	4.49	0.55	3.58	4.04
PLT_14	0.53	5.58	0.14	2.75	4.17
MDNet	0.63	2.50	0.16	2.08	2.29

(a) Baseline result

Tracker	Accuracy Score	Accuracy Rank	Robustness Score	Robustness Rank	Combined Rank
MUSTer	0.55	4.67	0.94	5.53	5.10
MEEM	0.48	7.25	0.74	5.76	6.51
DSST	0.58	4.00	0.76	5.10	4.55
SAMF	0.57	3.72	0.81	4.94	4.33
KCF	0.58	3.92	0.87	4.99	4.46
DGT	0.54	3.58	0.67	4.17	3.88
PLT_14	0.51	5.43	0.16	2.08	3.76
MDNet	0.60	3.31	0.30	3.58	3.45

(b) Region_noise result

Table 1: The average scores and ranks of accuracy and robustness on the two experiments in VOT2014 [26]. The first and second best scores are highlighted in red and blue colors, respectively.