논문 17-42-06-01 The Journal of Korean Institute of Communications and Information Sciences '17-06 Vol.42 No.06
https://doi.org/10.7840/kics.2017.42.6.1117

컨볼루션 신경망을 사용한 영상 객체 추적에서 경계 박스 분할을 통한 효과적인 온라인 학습 알고리듬

김 인 성◆, 황 선 영°

A Effective Online Training Algorithm by Partitioning Bounding Box for Visual Object Tracking Using Convolutional Neural Network

In-Sung Kim◆, Sun-Young Hwang°

요 약

본 논문은 영상객체추적 분야에서 경계 박스 재조정을 통하여 CNN을 위한 효율적인 온라인 학습 알고리듬을 제안한다. 제안된 알고리듬은 경계 박스 재조정을 위하여 분할된 경계 박스들을 이용한다. 이를 통해 영상 전반에 걸쳐 경계 박스의 크기가 거의 변하지 않는 이전 연구와 다르게, 제안된 알고리듬은 경계 박스의 크기를 재조정하여 효율성을 향상하였다. 제안된 알고리듬은 3가지 판별기로 구성된다. 앞단 분류기는 이전 프레임 경계 박스의 전체 특징을 이용하여 목표 객체를 추적하고, 뒷단 두 분류기들은 첫 번째 판별기로부터 얻어진 평가 점수가 정해진 임계 점수보다 작거나, 정해진 프레임 수를 처리 했을 때 경계박스를 재조정한다. 앞단 분류기를 위한 학습 데이터들은 단순히 이전 결과로부터 추출하고, 뒷단 분류기를 위한 학습 데이터들은 이전 결과를 분할하여 추출한다. 영상을 추적할 때 제안된 알고리듬은 경계 박스 재조정을 통하여 이전 알고리듬보다 정확한 결과와 다양한 크기의 학습 데이터를 만들 수 있다. 실험 결과 기존 연구들과 비교 했을 때, 성공률 평가 방법은 3%, 정확성 평가 방법은 5%의 성능 향상을 보인다.

Key Words : CNN, Deep learning, Visual tracking, Computer vision, machine learning, ANN

ABSTRACT

This paper proposes an effective online training algorithm which resizes bounding box for visual object tracking in Convolutional Neural Network (CNN). The proposed algorithm employs partitioned bounding boxes for resizing bounding box. Compared to previous algorithm where bounding box size is rarely changed, the proposed algorithm improves the efficiency in object tracking by resizing the size of bounding boxes. The proposed algorithm is composed of three classifiers; The front-end classifier is for tracking target object by using the entire feature of previous bounding box, and the other two back-end classifiers are for resizing bounding box when the score acquired by the first classifier is lower than threshold or the number of frame exceeds a determined number. Training data for front-end classifier are extracted from previous raw result, and training data for back-end classifiers king a target in a sequence, the proposed algorithm makes more accurate result and training data of various sizes than previous algorithm by resizing bounding box. Experimental results show that the success rate and the precision for visual object tracking are improved by 3% and 5%, respectively, when compared with previous works.

◆ First Author : Sogang University Department of Electronic Engineering, gioinsung@sogang.ac.kr, 학생회원
° Corresponding Author : Sogang University Department of Electronic Engineering, hwang@sogang.ac.kr, 종신회원
논문번호 : KICS2017-04-119, Received April 20, 2017; Revised May 9, 2017; Accepted May 19, 2017

Ⅰ. 서 론

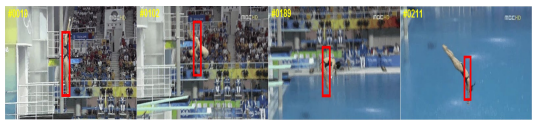
영상 객체 추적은 컴퓨터 비전 영역에서 중요하게 다루어지고 있는 문제 중 하나이다. 다양한 영역에서 인공지능(AI)을 통한 지능화 연구의 필요성이 부각됨에 따라 영상 객체 추적에서도 관련 연구가 진행 중이다[1-3]. 특히, 심화 학습의 제한사항이었던 데이터 부족과 하드웨어 한계성 문제가 GPU의 발전과 빅 데이터의 등장으로 인해 해결되며 심화 학습을 영상 객체 추적에 적용하려는 연구가 중심이되고 있다. 심화 학습을 적용한 영상 객체 추적은 객체 형태의 변화 및 사라짐, 조도의 변화, 빠른 움직임, 복잡한 배경 등을 극복해 대상 객체의 변형에 대응되도록 모델링되어야 한다[4].

객체의 다양한 변화에 효과적으로 대처하도록 대상 객체를 모델링 하는 방법이 활발하게 연구되고 있다. 학습을 통한 객체 모델링 방법은 생성적 (generative) 방법과 판별적 (discriminative)방법으로 나뉜다. 생성적 방법은 이전 프레임으로부터 대상 객체 특징을 확률 분포로 얻는다. 대상 객체에 대한 특징 x와 해당되는 클래스 y를 표현하는 확률 분포 P(x | y)와 P(y)를 학습한다. 대표적인 알고리듬으로 Sparse representation[5-9], online density estimation[10], incremental subspace learning[11]의 방법이 제안되었다. 판별적 방법은 특징 x와 클래스 y를 통하여 P(y | x)를 판별하는 boundary를 학습하는 방식으로 online boosting[12,13], multiple instance learning[14], structed SVM[15], online random forest[16,17]이 제안되었다. 위의 방법들은 객체를 표현하는 특징을 얻기 위하여 haar-like, template, histogram 등과 같은 사람에 의해 정의된 hand-craft 특징을 이용하나, 이들은 적용하는 영역에 적합하도록 추출되지만 성능의 한계를 가지고 있다.

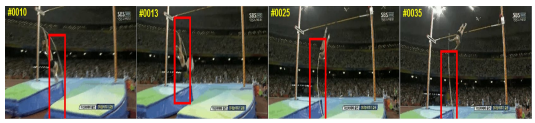
최근 이러한 한계를 뛰어넘는 방법으로 CNN (Convolutional Neural Network)이 주목받고 있다. CNN은 특징 추출에 사용되는 컨벌루션 필터를 학습하여 hand-craft보다 데이터에 적합한 특징 추출을 위한 필터값을 가진다. 영상 분류 분야의 권위 있는 경연 중 하나인 ImageNet challenge에서 CNN의 구조를 구현한 Alexnet[18]은 120만장의 영상 정보를 학습하여 ILSVRC-2012에서 우승함으로써 CNN의 성능을 입증하였다. CNN은 객체 탐색(object detection), 영상 분류(image classification), 영상 분할(semantic segmentation)등 여러 컴퓨터 영상 분야에 적용되고 있으나, 영상 추적 분야에서는 추적을 위한 학습 데이터 수집의 어려움과 부적절한 신경망의 구조로 인해 다른 분야에 비하여 활발하게 적용되고 있지 않으며 여전히 영상 객체 추적 분야에서는 hand-craft 특징을 사용하고 있다. 영상 추적 분야에서 CNN을 적용한 대표적인 알고리듬으로 CNN-SVM[19], MDNet[20]이 있다.

기존의 CNN을 적용한 영상 추적 연구들은 효과적인 특징 추출을 위하여 선행학습을 통한 오프라인학습을 중심으로 진행되었다. 그에 반하여 온라인학습은 단순하게 적용되어서 CNN을 위한 온라인학습 연구가 필요하다. 기존의 알고리듬은 온라인학습을 위한 데이터 생성을 이전 프레임 객체의 경계 박스 전체크기에 의존적이다. 이는 그림 1에서 제시한 것처럼 추적을 실패하거나 영상이 진행되어도 첫 프레임의 경계박스의 크기가 거의 변화하지 않는 문제를 발생시킨다. 그림 1-(a)에서 객체의 크기변화를 반영하지 못하여 결과가 정확하지 않는 문제를 제시하고, 그림 1-(b)에서 영상 추적에 실패하는 문제를 제시한다. 또한, 온라인학습에서 매 단계마다 사용되는 학습 데이터들은 이전 프레임 결과의 크기를 기준으로 생성되므로, 이전 결과가 잘못되었을 경우 잘못된 학습 데이터를 생성하는 문제점과, 좋은 성능을 얻기 위하여 다수의 후보 객체가 필요한 문제점을 가진다.

본 논문에서는 이러한 문제를 개선하는 CNN을 이용한 새로운 온라인학습 방법을 제안한다. 첫 프레임의 경계 박스 (Ground truth bounding box)를 분리하여 각각의 back-end 분류기들을 학습하고, back-end 분류기들의 결과를 취합하여 기존의 경계 박스 결과를 재조정한다. 이를 통하여 영상 추적의 문제인 학습 데이터 부족을 해소하고 보다 정확하게 경계박스를 결정할 수 있다. 특징을 추출하기 위한 컨벌루션 층은



(a)



(b)

그림 1. 기존의 CNN을 이용한 알고리듬의 영상 결과 (a) Diving 영상, (b) Jump 영상
Fig. 1. Result of previous algorithm using CNN (a) Diving sequence, (b) Jump sequence

기존의 방법과 마찬가지로 선행학습을 이용하고 온라인학습 시에는 이진 분류기만 학습한다. 제안된 방법은 기존 방법의 절반의 후보 객체만으로 충분한 성능을 가지므로 추가적인 back-end 분류기들을 학습하여도 처리시간이 크게 증가하지 않는다. 제안된 알고리듬의 성능을 확인하기 위하여 다양한 객체를 대상으로 실험을 수행하였으며, 이를 위해 영상 추적의 벤치마크인 OTB (Online Tracking Benchmark)[21]를 이용하였다.

본 논문은 다음과 같이 구성된다. 2절에서 관련 연구와 제안된 알고리듬에 대하여 설명한다. 3절에서 실험 결과와 기존의 알고리듬과 성능비교를 제시하고, 마지막으로 4절에서 결론 및 추후 과제를 제시한다.

Ⅱ. 본 론

2.1 이론적 배경 및 관련 연구

본 절에서는 영상처리 분야에서 좋은 성능을 보이는 심화학습 방법인 CNN과 기존의 CNN을 이용한 영상 추적 알고리듬에 대해 기술한다.

2.1.1 컨벌루셔널 신경망 (CNN)

최근 CNN을 이용한 영상 특징 추출은 컴퓨터 영상 분야에서 다양하게 적용되고 있다. 다중 퍼셉트론을 (MLP) 이용한 영상처리는 이미지의 모든 픽셀이 가중치를 갖지만 CNN은 필터값만 가중치로 갖는다. 전체 이미지에 필터를 이용한 동일한 컨벌루션 연산을 진행하여 다중 퍼셉트론보다 연산량을 줄일 수 있다. 또한, 필터를 사용함으로써 주변 픽셀과의 상관관계가 중요한 영상정보를 효과적으로 다룬다. 컨벌루셔널 신경망은 필터를 이용한 컨벌루션 층, Pooling (subsampling) 층과 분류기로 구성되어 있으며, 그림 2에서 CNN의 구조를 제시하였다.

(1) 필터를 이용한 컨벌루션 층

영상처리에서 컨벌루션은 특징 추출을 위한 방법으로 사용되고 있다. 기존의 특징 추출은 원하는 특징 추출을 위하여 정의된 값을 가진 필터를 사용하여 컨벌루션 연산을 수행하지만, 컨벌루셔널 신경망에서는 학습을 통하여 결정된 필터의 값을 사용한다. 각 층마다 k개의 필터가 존재할 경우 특징 h_{ijk}를 얻는 과정을 식 (1)에 표현하였다.

$$h_{ijk} = \tanh(W_k * X_{ij} + b_k) \quad (1)$$

여기서 X_{ij}는 컨벌루션 연산을 수행할 대상 이미지에서 i, j 위치를 기준으로 추출한 이미지를 나타낸다. 추출할 이미지의 크기는 컨벌루션 연산에 사용되는 필터 W_k와 같은 크기를 가진다. W_k는 n x n x K 의 크기를 가지는 k번째 필터이다. 필터의 크기 n x n x K 는 사용자가 정의하며, K는 영역의 크기를 나타내고 n은 같은 영역 내의 윈도우 크기를 의미한다. 이때, K는 특징맵 수에 의존적이고 n은 주로 3 x 3, 5 x 5의 크기를 갖는다. 예를 들어, 그림 2의 입력 층에서 필터 크기 5 x 5 x 3 은 n = 5의 윈도우 크기를 가지고 있으며 RGB 입력을 받아서 K = 3 이다. b_k는 bias 이고, tanh는 비선형 함수로 Sigmoid 혹은 ReLU함수로 대체 가능하다. 모든 W_k는 각 층이 다른 값을 갖고 독립적으로 적용된다. 영상 이미지 전반에 걸쳐서 추출된 h_{ijk}들을 합하여 특징맵을 생성한다.

(2) Pooling 층

컨벌루션 과정으로 각각의 특징맵을 얻은 후 Pooling 과정을 거쳐 특징맵을 압축한다. Pooling은 정의된 윈도우 크기 안에서 대푯값만 가지고, 대푯값은 최댓값, 최솟값, 평균을 통하여 결정한다. Pooling 과정은 미리 결정된 stride 변수만큼 특징맵을 압축한다. 예를 들면 stride 변수가 2일 경우는 특징맵의 크기가 1/2로 감소하고, 3일 경우에는 1/3으로 감소한다. Pooling 과정을 통하여 연산량을 줄일 수 있으며 영상 크기 변화에도 대응할 수 있다.

(3) 분류기

컨벌루션 층과 Pooling층은 특징 추출을 위하여 이용되고 추출된 특징을 위한 분류기 (Classifier)를 필요로 한다. Krizhevsky가 제안한 Alexnet[18]처럼 대부분의 컨벌루셔널 신경망은 다중신경망(MLP)을 분류기로 선택하고 있다. 그 외에도 SVM 같이 기존에 제안된 다양한 분류기와 결합도 가능하다.

컨벌루셔널 신경망은 컨벌루셔널 층과 Pooling층이 반복적으로 수행되어 특징의 크기가 작아지며 전체를 대표하는 특징들만 남게 된다. 이러한 특징들은 심화학습으로 이미지 전반을 고려한 특징이 된다. 컨벌루셔널 신경망의 학습은 기존의 방식과 동일하게 SGD (Stochastic Gradient Descent)를 통한 역전파 (Back-propagation)로 이루어진다.

2.1.2 CNN을 이용한 영상추적 알고리듬

CNN을 통한 대표적인 영상 추적 알고리듬은 CNN-SVM[19], MDnet[20]이 있다. 많은 양의 데이터를

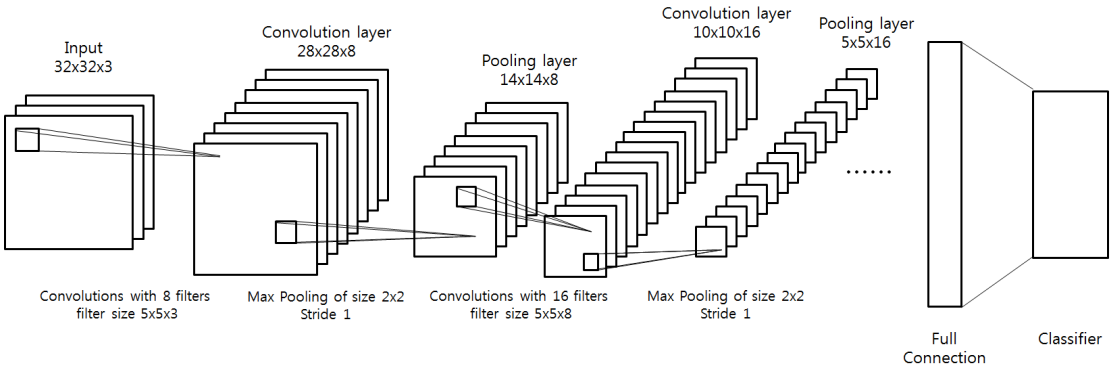


그림 2. CNN의 구조
Fig. 2. Structure of CNN

이용하여 선행학습 된 컨벌루션 층과 Pooling 층을 특징 추출에 이용한다는 공통점을 가지고 있고 추출된 특징을 이용하여 배경 (background)과 대상 객체 (foreground)를 이진 분류기로 판별하는 판별적 (Discriminative)모델이다.

(1) CNN-SVM

R-CNN[22]을 선행학습 모델로 특징 추출에 이용하고 SVM을 분류기로 적용한 모델이다. KKT (Karush-Kuhn-Tucker) 조건과 라그랑지안 함수 최적화를 이용하여 SVM을 학습한다. 동영상 정보는 순차적으로 영상정보가 얻어지기 때문에 온라인 학습을 위해서 Incremental SVM[23]을 사용하고 saliency map[24]을 통하여 Localization을 효과적으로 구현하였다.

(2) MDnet

R-CNN은 객체 탐색, 분류를 위한 모델이기 때문에 영상 추적에 최적화 된 특징 추출 모델이 아니다. MDnet은 선행 학습된 VGG-M[25] 모델을 영상 추적에 적합한 특징을 추출하도록 미세조정 (fine-tune) 하는 학습법을 제안하였고 OTB[21], VOT[26,27] 영상 데이터를 이용하여 학습하였다. MDnet은 모든 영상이 특징 추출을 위한 컨벌루션 층과 Pooling 층은 공유하지만 객체 분류를 위한 이진분류기는 독립적으로 존재하는 다중 영역(Multi-domain) 방법을 적용하여 학습함으로써 R-CNN보다 영상 추적에 적합한 특징 추출 필터를 얻는다. 그 외에도 MDnet은 객체 탐색에 사용되는 Bounding Box Regression[22]을 적용하여 더욱 정확한 경계박스를 갖도록 하고, 더 의미 있는 학습 데이터 추출을 위한 Hard-Mining을 적용했다.

2.2 제안된 온라인 학습 알고리듬

본 절에서는 제안한 알고리듬에 관해 기술한다. 2.2.1절에서는 기존의 영상 객체 추적 알고리듬과 제안된 알고리듬의 전체 흐름에 대해 기술하고 2.2.2절에서 제안된 알고리듬의 동작에 대해 기술한다.

2.2.1 영상 객체 추적 알고리듬의 전체 흐름

영상 추적에서 객체는 다양한 변화를 고려하여 모델링되어야 한다. 객체의 변화가 적은 영상은 객체가 일정한 형태만 가지므로 온라인 학습이 필요하지 않으나, 실제 영상들은 객체의 크기, 형태, 명암의 변화가 발생하여 이를 고려하기 위한 온라인 학습은 필수적이다.

기존의 CNN을 이용한 추적 알고리듬은 매 프레임마다 이전 프레임의 결과를 기준으로 후보 객체를 생성하고 후보 객체 중 분류기로 얻어진 평가 점수 (Score)가 가장 높은 후보 객체를 해당 프레임 결과로 선택한다. 온라인 학습을 위한 데이터들은 결과를 중심으로 가우시안 분포로 생성되고, 생성된 데이터들은 결과와 오버랩 비율을 기준으로 positive 데이터와 negative 데이터로 분류된다. 이때 생성된 학습 데이터들의 크기는 해당 프레임 결과와 학습 데이터 사이의 오버랩 비율을 통해 생성되어 학습 데이터와 결과의 크기가 대체로 같다. 이는 기존의 알고리듬이 다양한 크기의 데이터를 학습하지 못하여 정확하게 표현된 후보 객체가 높은 평가 점수를 받지 못하고 이전 프레임 결과와 크기가 유사한 후보 객체가 높은 점수를 받는 문제점을 그림 1에서 제시하였다. 그림 1에서 첫 프레임의 경계 박스의 크기가 영상 전체에 걸쳐 거의 변화하지 않는다.

제안한 알고리듬은 후보 객체 크기에 의존하는 문제점을 보완하기 위하여 기존의 분류기 외에 이전 프

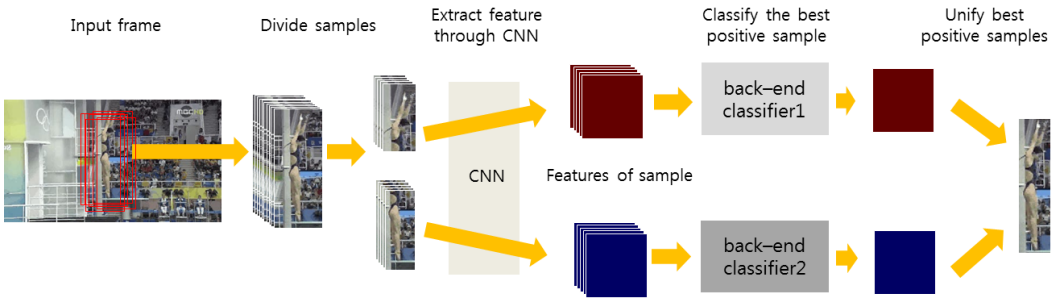


그림 3. 제안한 알고리듬의 경계 박스 재조정 과정
Fig. 3. Resizing bounding box process of proposed algorithm

레임의 경계박스를 분할하여 각 분할된 경계박스를 모델링하기 위해 분류기를 추가하여 학습과정을 수행한다. 앞단 front-end 분류기에서 평가된 점수가 임계값에 비해 작거나 정해진 프레임 수가 처리될 때마다 추적 대상을 분할하고 각각의 후단 back-end 분류기들의 결과를 이용하여 경계 박스를 재조정하는 과정을 거친다. 그림 3에 제안된 알고리듬의 경계 박스 재조정 과정을 보인다. 각각의 분류기를 위한 학습 데이터는 취합 전 결과를 기준으로 생성하여 학습시키고 위의 과정들을 마지막 프레임까지 반복한다. 경계 박스는 (x, y, w, h)로 정해지며, 각각의 변수 x, y는 경계 박스의 좌측 하단 좌표를, w, h는 너비와 높이를 나타낸다. 제안된 방식은 후보 객체의 수를 줄이면서 다양한 경우의 수를 고려할 수 있고 추적할 객체에 다양하게 표현할 수 있다. 또한 경계 박스의 크기변화를 고려할 수 있어 기존의 알고리듬보다 객체를 정확하게 표현하는 경계 박스를 가진다.

2.2.2 제안한 알고리듬의 동작

그림 4에 제안한 알고리듬의 pseudo-code를 제시하였다. 제안된 알고리듬에서 특징 추출은 CNN을 이용하고 기존이 알고리듬과 다르게 추적 객체 크기 재조정을 위한 분류기를 추가로 학습한다. 객체 추적은 첫 번째 프레임에 추적 대상 객체의 정보를 바탕으로 나머지 프레임에서 해당 객체를 추적 하므로 객체를 모델링 하는 첫 번째 프레임 처리 과정과 객체의 추적과 변화를 처리하기 위한 나머지 프레임 처리 과정으로 나뉜다. 모든 프레임에 걸쳐서 영상 추적을 위한 후보 객체들은 X_t^i 로 표현되고, 이때 아래 첨자 t는 처리하는 프레임의 index, i는 후보 객체의 index이다. 윗 첨자 *는 최적의 상태를 표현하여 i개의 후보 객체 중 추적 알고리듬을 통해서 얻어진 최적의 후보 객체는 X_t^*로 표현된다.

(1) 첫 번째 프레임 처리

첫 번째 프레임은 추적 객체 X_1^*의 정보를 가지고 있고, 이를 이용하여 객체를 추적하기 위한 front-end 분류기와 객체 크기를 재조정하기 위한 두 개의 back-end 분류기들을 초기화하고 학습한다. Front-end와 back-end 분류기들의 컨벌루션 필터값과 FC 가중치는 오프라인 학습된 CNN 값으로 초기화되고 이진분류기의 가중치는 random 값으로 초기화된다. Front-end 분류기를 위한 학습 데이터는 X_1^*을 평균으로 한 가우시안 분포로 생성하고 back-end들을 위한 학습 데이터는 X_1^*를 상, 하로 분할하여 평균으로 삼고 학습 데이터를 생성한다. 모든 학습 데이터는 생성 기준이 된 경계 박스와 학습 데이터 간의 오버랩 비율로 구별되고 비율에 따라 positive 데이터와 negative 데이터로 나뉜다. 과적합 (over-fitting)을 방지하기 위하여 positive 데이터 생성 오버랩 비율을 기존의 알고리듬이 통상적으로 적용한 0.7과 다르게 제안한 알고리듬은 0.9의 높은 비율을 적용하여도 2개의 back-end 분류기 결과를 취합함으로써 과적합 문제를 억제할 수 있다. 분류기 학습은 짧은 프레임 주기로 학습하는 short-term과 긴 프레임 주기로 학습하는 long-term으로 나누어 수행한다. 분류기의 성능은 positive 데이터보다 negative 데이터에 의해 좌우되므로[19] 빈번하게 수행하는 short-term 학습은 negative 데이터를 사용하고 상대적으로 빈번하지 않은 long-term 학습은 positive 데이터를 사용한다. 이때 short-term은 20 프레임, long-term은 100 프레임을 학습에 사용하는 최대 프레임 수로 정하였다. 연산량 증가를 방지하기 위하여 컨벌루션 연산에 사용되는 필터값은 온라인 학습 대상에서 제외되고 FC층의 가중치와 이진분류기의 가중치만 온라인 학습하게 된다.

Algorithm Online tracking algorithm

Input: Pretrained CNN filters
Initial target bounding box, X_1^*
Number of frames for periodic training, τ
Output: Estimated target bounding box, X_t^*

```
Initialize feature extract layers with pre-trained CNN layers ;

/* Process of 1st frame */
Generate training data for front-end classifier from X1* ;
Partition X1* into X1_p1* and X1_p2* ;
Generate training data for two back-end classifiers from X1_p1* and X1_p2* ;
Train front-end and two back-end classifiers ;

/* Process of remaining frames */
for (t = 2 ; t ≤ last frame number ; t++) {
    /* t is the frame index being processed */
    Generate candidate objects using Xt-1* bounding box ;
    Select Xt* from candidate objects by front-end classifier ;
    if (f(Xt*) > 0) {          /* When the score of Xt* is more than 0 */
        Generate training data from Xt* ;
        Partition Xt* into Xt_p1* and Xt_p2* ;
        Generate training data from Xt_p1* and Xt_p2* ;
    }
    else {                     /* When the score of Xt* is less than 0 */
        Resize Xt* by two back-end classifiers ;
        Train front-end and back-end classifiers ;
    }
    if (τ frames processed)
        /* Training performed for predetermined number of frames periodically */
        Train front-end and back-end classifiers ;
}
```

그림 4. 제안된 경계 박스 재조정의 전반적인 알고리듬
Fig. 4. Overall proposed algorithm for resizing bounding box

(2) 두 번째 이후 모든 프레임 처리

영상 처리는 2단계로 나누어 진행된다. 첫 번째 단계는 front-end 분류기를 이용한 통상적인 객체 추적 단계이고, 두 번째 단계는 back-end 분류기들을 통한 경계 박스 재조정 단계이다. 첫 번째 단계에선 객체 추적을 위하여 이전 프레임의 결과를 바탕으로 현재 프레임의 후보 경계 박스 샘플 (i = 128)을 생성한다. 생성된 후보 경계 박스 중 최적의 후보 X_t^*를 식 (2)와 front-end 분류기를 이용하여 결정한다.

$$X_t^*(C) = \text{argmax } f(X_t^i) \quad (2)$$

여기서 C는 평가에 사용된 분류기이다. 사용된 분류기에 따라 X_t^*(front)는 front-end 분류기로 평가된 최적의 후보, X_t^*(back1)과 X_t^*(back2)는 각각 첫 번째와 두 번째 back-end 분류기들로 평가된 최적의 후보이다. t는 처리하는 프레임의 index, i는 후보 객체의 index이고, $f(X_t^i)$은 t 프레임의 i번째 후보 객체의 positive 평가점수이다. 식 (2)로 결정된 X_t^*(front)의 평가점수가 임계점수 이상이면 두 번째 단계를 거치

지 않고 학습을 위한 데이터를 생성한 후 다음 프레임을 처리하고, 이하일 경우 두 번째 단계를 진행한다. 두 번째 단계에서 식 (2)와 back-end 분류기들을 이용하여 재조정을 위한 후보 객체 X_t^*(back1), X_t^*(back2)를 결정한다. 이때 X_t^*(back1), X_t^*(back2)는 각각의 back-end 분류기들로 얻어진 결과다. 얻어진 후보 객체의 평가점수를 고려하여 후보 객체를 병합하고 X_t^*를 재조정한다. 병합은 크게 두 부분으로 나뉜다. 첫 번째는 더 낮은 평가 점수를 받은 후보객체를 더 높은 점수를 받은 후보객체 쪽으로 옮기는 과정이다. X_t^*(back2)의 평가 점수가 더 낮을 경우 X_t^*(back2)의 위치를 조정하는 방법을 식 (3), (4)에 표현하였다.

$$x2 = (x1-x2)(1-f(X_t^*(back2)))+x2 \quad (3)$$

$$y2 = (y1-y2)(1-f(X_t^*(back2)))+y2 \quad (4)$$

식 (3), (4)에서 (x1, y1), (x2, y2)은 각각 X_t^*(back1)와 X_t^*(back2)의 좌측 하단 픽셀의 위치를 나타내며, $f(X_t^*(back2))$는 X_t^*(back2)의 positive 평가 점수로 0 ~ 1의 값을 갖는다. 즉, 평가점수가 0일 경우 (x2, y2)는 전혀 고려되지 못하고 1일 경우에는 온전히 고려됨을 의미한다. 반대로 X_t^*(back1)의 평가 점수가 더 낮을 경우에는 식 (3), (4)의 (x1, y1), (x2, y2)를 역으로 대입하고 $f(X_t^*(back2))$를 $f(X_t^*(back1))$로 대치한다. 두 번째 과정은 두 개의 경계박스를 하나로 합치는 과정이다. 각각의 평가점수를 고려하여 위치를 조정한 두 개의 경계 박스를 식 (5), (6), (7)을 통하여 구한다.

$$x^* = \min(x1,x2),\ y^* = \min(y1,y2) \quad (5)$$

$$w^* = \max((x1+w1),(x2+w2))-x^* \quad (6)$$

$$h^* = \max((y1+h1),(y2+h2))-y^* \quad (7)$$

최종적으로 back-end 분류기들을 이용한 재조정 과정을 통해 최적의 경계박스 $X_t^* = (x^*, y^*, w^*, h^*)$를 얻고 다음 프레임을 처리한다.

(3) 학습에 사용된 조건

오프라인 학습에서 MDnet은 100,000 epoch 학습을 수행하였으며 특징을 추출하기 위한 컨벌루션 층을 정확하게 학습하기 위해 FC층에 비해 1/10의 learning rate를 적용하여 각각 0.0001, 0.001을 사용했다. 첫 번째 프레임 학습은 분류기가 과적합되지 않도록 오프라인 학습보다 적은 30 epoch와 0.0001의 learning rate로 FC층과 이진분류기층만 학습하였다. 나머지 프레임의 온라인 학습은 빠른 처리를 위하여 10 epoch와 learning rate 0.0003을 사용하였다. 온라인 학습을 위한 positive 데이터와 negative의 데이터의 구별을 위한 오버랩 비율은 각각 ≥ 0.9, ≤ 0.3 이고 각각 50, 200개의 샘플을 생성하였다. 모든 parameter들은 기존의 알고리듬과 동일하게 적용하여 성능분석에 영향을 미치지 않도록 하였다. 단, 후보객체는 기존의 알고리듬의 절반인 128개를 각 프레임 별로 이전 프레임 결과 (X_{t-1}^*)의 가우시안 분포로 생성하여 성능을 비교하였다.

Ⅲ. 실 험

본 논문에서 제안한 알고리듬은의 성능을 영상 추적 벤치마크인 Object Tracking Benchmark (OTB)[21] 영상 데이터를 이용하여 기존의 알고리듬과 비교 평가하였다. 제안된 알고리듬은 MatConvNet[28] toolbox를 이용하여 VOT2013[27], VOT2014[26] 영상 정보를 선행 학습한 MDnet을 특징 추출에 이용하였고, 3.8Ghz Intel i7-2600, Nvida GeForce GTX 560 Ti 하드웨어 환경에서 시뮬레이션을 수행하였다. 평가에 사용된 OTB 영상들은 Benchmark Attributes에 따라 분류되어 있다. 제안된 알고리듬은 Hand-craft 특징을 사용하는 Struck[15] 과 지금까지 성능 좋은 것으로 알려진 MDnet[20] 추적 알고리듬과 비교하였다. 실험결과 제시에 있어 성공률 (Success Rate)과 정확성 (Precision)을 평가 기준으로 하였으며, One-Pass Evaluation (OPE)[21] 평가 방법을 이용하였다. 성공률은 Ground-Truth와 결과의 오버랩 비율을 기준으로 하였으며, 정확성은 Ground-Truth와 결과의 중앙 위치 차이를 기준으로 성공 여부를 평가하였다. 일반적인 평가 기준으로는 객체의 위치만 평가하는 정확성보다 객체의 크기까지 평가에 고려되는 성공률이 유효하게 채택된다.

표 1에 성공률 결과를 제시하였고, 표 2에 정확성 결과를 제시하였다. 제시된 성공률 결과는 추적 성공 여부를 평가하기 위하여 Ground-Truth와 추적 결과의 오버랩 비율 threshold를 0.5로 하였고, 정확성 결과는 Ground-Truth의 중앙 좌표와 추적 결과의 중앙 좌표의 차이 (Location error threshold)를 25로 적용하여 추적 성공 여부를 평가하였다. 모든 평가는 Benchmark Attributes인 IV (Illumination Variation), SV (Scale Variation), OCC (Occlusion), DEF

표 1. Benchmark attributes에 따른 성공률
Table 1. Success rate to benchmark attributes

Trackers / Benchmark Attributes	Success rate (Overlap threshold = 0.5)				
	MDnet	Struck	proposed	Comparison(%)	
				vs MDnet	vs Struck
IV	0.586	0.339	0.625	+3.9	+28.6
SV	0.597	0.358	0.612	+1.5	+25.4
OCC	0.561	0.341	0.583	+2.2	+24.2
DEF	0.565	0.333	0.594	+2.9	+26.1
MB	0.579	0.368	0.608	+2.9	+24.0
FM	0.598	0.375	0.597	-0.1	+22.2
IPR	0.569	0.367	0.616	+4.7	+24.9
OPR	0.573	0.330	0.588	+1.5	+25.8
OV	0.559	0.327	0.573	+1.4	+24.6
BC	0.562	0.357	0.620	+5.8	+26.3
LR	0.566	0.330	0.598	+3.2	+26.8
Average				+2.71	+25.4

(Deformation), MB (Motion Blur), FM (Fast Motion), IPR (In-Plane Rotation), OPR (Out-of-Plane Rotation), OV (Out-of-View), BC (Background Clutters), LR (Low Resolution) 으로 분류하여 수행하였다. 표 1에 제시한 성공률 평가 결과는 특징 추출에 CNN을 사용한 MDnet과 제안된 알고리듬이 Haar-like 특징을 사용한 Struck에 비하여 평균 25.4% 좋은 성공률을 보인다. 제안된 알고리듬은 같은 CNN 특징 추출을 이용한 MDnet보다 평균적으로 2.7%의 성공률 향상을 보였다. 표 2에 제시한 정확성 평가 결과는 제안된 알고리듬이 평균적으로 Struck보다 38.4%, MDnet보다 5.7% 좋은 정확성 결과를 보였다. 제안된 알고리듬은 성공률 평가의 FM을 제외한 모든 Attributes에서 성능 향상을 보였으며, 특히, BC, IV, IPR Attributes를 더욱 효과적으로 처리하였음을 보인다. 처리 속도 측면에서 제안된 알고리듬은 모든 프레임에서 경계 박스를 재조정 하지 않고 일정 주기를 가지고 재조정하므로 모든 영상 데이터의 처리 시간이 MDnet에 비해 평균적으로 0.01초 증가한다.

그림 5와 그림 6에 각각 Overlap threshold와 Location error threshold의 변화로 얻어진 성공률과 정확성 결과를 Struck[15], TLD[29], OAB[30], MDnet[20] 알고리듬과 비교하였다.

표 2. Benchmark Attributes에 대한 정확성
Table 2. Precision to benchmark attributes

Trackers / Benchmark Attributes	Precision (Location error threshold = 25)				
	MDnet	Struck	proposed	Comparison(%)	
				vs MDnet	vs Struck
IV	0.841	0.448	0.914	+7.3	+46.6
SV	0.858	0.523	0.870	+1.2	+34.7
OCC	0.787	0.477	0.844	+5.7	+36.7
DEF	0.835	0.465	0.873	+3.8	+40.8
MB	0.799	0.494	0.854	+5.5	+36.0
FM	0.822	0.512	0.838	+1.6	+32.6
IPR	0.800	0.486	0.893	+9.3	+40.7
OPR	0.805	0.438	0.861	+5.6	+42.3
OV	0.752	0.434	0.810	+5.8	+37.6
BC	0.788	0.449	0.881	+9.3	+43.2
LR	0.818	0.587	0.899	+8.1	+31.2
Average				+5.74	+38.4

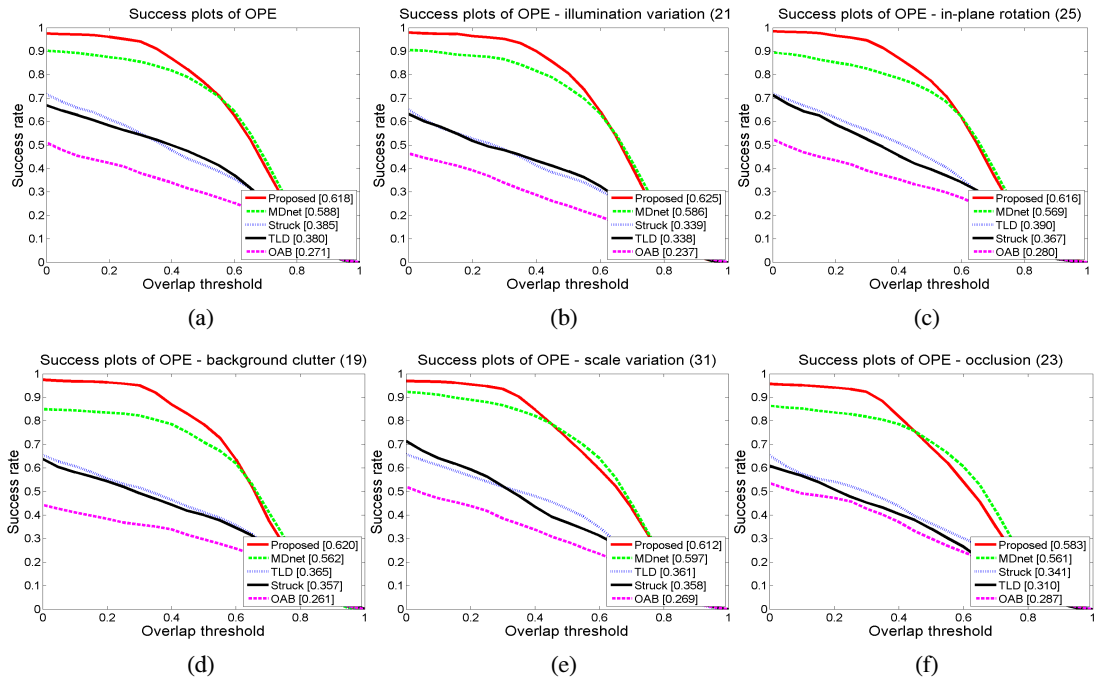


그림 5. Overlap threshold 변화에 따른 benchmark attributes의 성공률 (a) 모든 영상, (b) Illumination Variation, (c) In-Plane Rotation, (d) Background Clutter, (e) Scale Variation, (f) Occlusion

Fig. 5. Success rate to benchmark attributes according to overlap threshold (a) All image sequence, (b) Illumination Variation, (c) In-Plane Rotation, (d) Background Clutter, (e) Scale Variation, (f) Occlusion

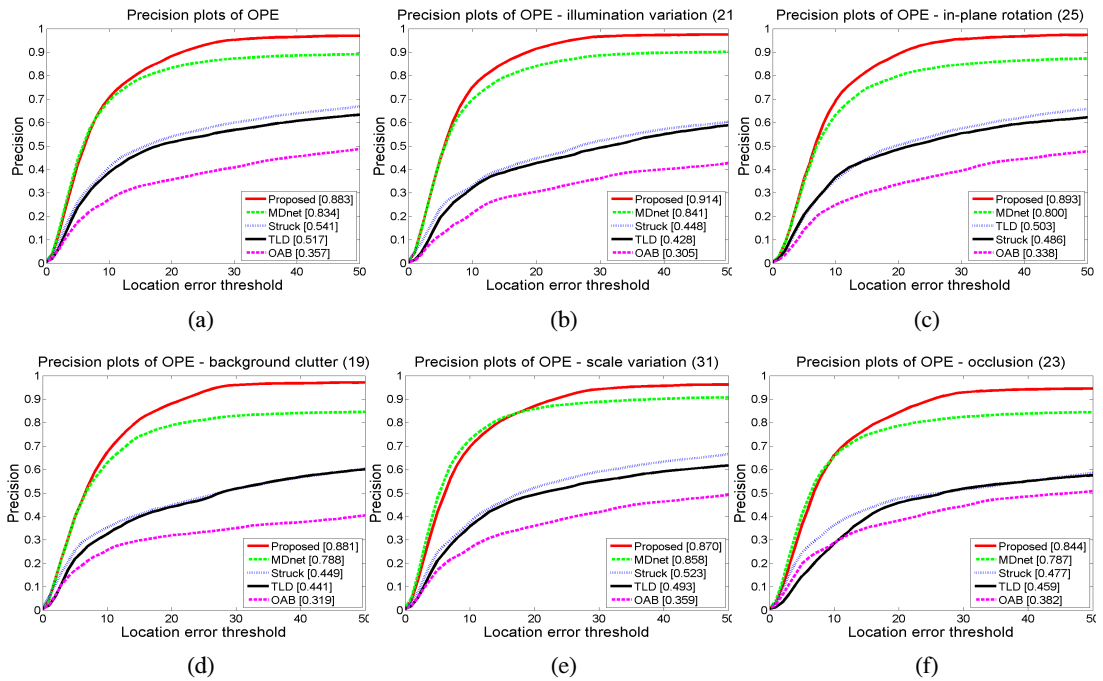
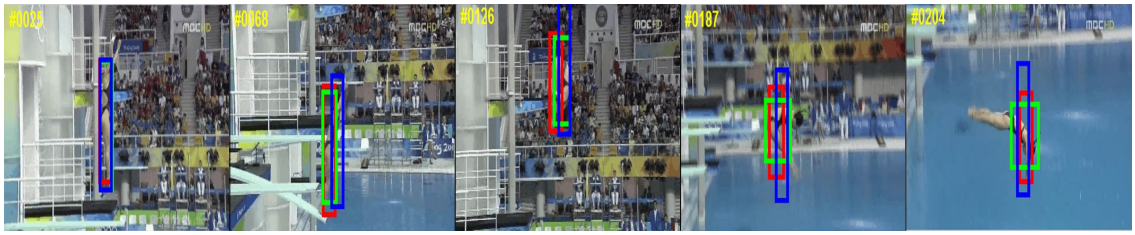
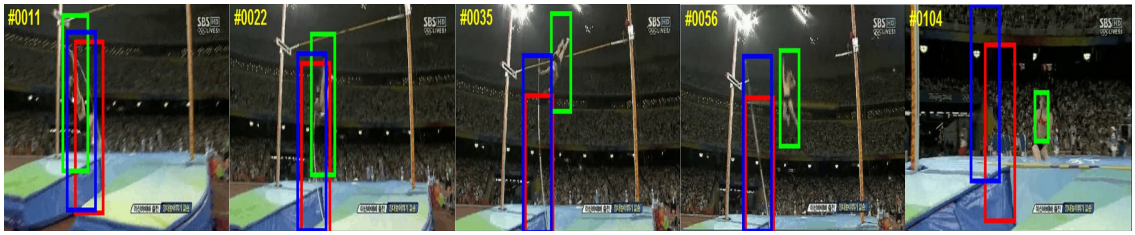


그림 6. Overlap threshold 변화에 따른 benchmark attributes의 정확성 (a) All sequence, (b) Illumination Variation, (c) In-Plane Rotation, (d) Background Clutter, (e) Scale Variation, (f) Occlusion.

Fig. 6. Precision to benchmark attributes according to overlap threshold (a) All image sequence, (b) Illumination Variation, (c) In-Plane Rotation, (d) Background Clutter, (e) Scale Variation, (f) Occlusion



(a)



(b)

그림 7. 제안된 알고리듬 (green), MDnet (red), Struck (blue) 의 추적 결과 (a) Diving 영상, (b) Jump 영상
Fig. 7. Tracking result to proposed algorithm (green), MDnet (red) and Struck (blue) (a) Diving sequence, (b) Jump sequence

그림 5와 그림 6에서 threshold에 변화를 주어 평가하고 이를 도식화하여 고정된 threshold 적용한 표 1과 표 2보다 확장된 결과를 제시하였다. 성공률 평가는 threshold가 커질수록 어려운 제한조건이 되고 정확성 평가는 threshold가 작아질수록 어려운 제한조건이 된다. 그림 5와 그림 6에서 threshold가 커질수록 전반적으로 성공률 곡선이 감소하고 정확성 곡선이 증가하는 것을 확인할 수 있다. 그림 5에서 제안된 알고리듬은 오버랩 threshold가 증가하여도 다른 알고리듬과 비교하여 성공률 곡선이 천천히 감소하고 높은 성공률을 보인다. 그림 6에서 Location error threshold가 커질수록 제안된 알고리듬의 정확성 곡선이 가장 급격하게 증가하고 높은 정확성을 보인다. 알고리듬의 성능은 AUC(Area Under Curve)를 통하여 수치적으로 나타내었고 그림의 우측하단에서 확인할 수 있다. 그림 5와 그림 6의 곡선과 수치를 통해 제안된 알고리듬은 다양한 조건의 오버랩 threshold와 location error threshold에도 기존의 추적 알고리듬에 비해 높은 성공률과 정확성 결과를 보임을 확인할 수 있다.

그림 7에 MDnet, Struck과 제안된 알고리듬의 영상 처리 결과를 제시하였다. 그림 7-(a) 에서 제안된 알고리듬이 모든 프레임에서 MDnet과 Struck 보다 객체의 크기 변화를 효과적으로 표현한다. 그림 7-(b) #35 프레임에서 MDnet과 Struck이 객체의 크기 변화를 고려하지 못하고 추적에 실패하는 반면 제안된 알고리듬은 객체의 크기변화를 고려하여 영상 마지막까지 추적에 성공함을 보인다.

Ⅳ. 결 론

기존의 영상객체추적을 위한 CNN은 효율적인 특징 추출을 위한 오프라인 학습이 주된 연구 방향이었으나, 오프라인 학습은 영상 내에서 객체의 변화에 대응하지 못하는 한계를 가지고 있다. 객체의 변화에 대응하기 위해선 온라인 학습이 필요하지만 기존의 CNN 온라인 학습은 추적대상 크기 변화에 거의 대응하지 못한다. 본 논문에서는 기존의 방법보다 정확하게 추적대상을 표현하기 위해 bounding box를 재조정하는 온라인학습 알고리듬을 제안하였다. 제안된 영상추적 알고리듬은 영상추적을 위한 front-end 분류기와 함께 bounding box를 재조정하기 위한 back-end 분류기들을 적용한다. partitioned bounding box로 back-end 분류기들을 학습시키고, back-end 분류기들의 결과를 취합하여 bounding box를 재조정 한다. 기존의 오프라인 학습과 제안된 CNN 온라인 학습 알고리듬을 이용하여 효과적으로 추적대상 크기 변화에 대응하였다. 이를 통해 기존의 영상 추적 알고리듬 보다 강건하게 객체를 학습시킬 수 있고 다양한 benchmark attributes를 효과적으로 추적할 수 있다. 추후과제로 많은 연산을 필요로 하는 CNN을 실시간 영상객체추적에 적용하기 위한 연구가 필요하다. CNN을 이용한 알고리듬들은 모든 후보객체의 특징을 독립적으로 얻는다. 이는 좋은 성능을 가지기 위해

후보객체의 수가 증가할수록 처리속도가 느려지는 한계를 가지고 있다. 모든 후보 객체를 CNN을 통하여 독립적으로 평가하지 않고 전체 이미지의 특징을 한 번에 추출한 후 후보객체에 해당되는 영역의 특징을 선택하면 CNN 과정을 최소화하여 빠른 처리속도를 기대할 수 있을 것이다.

References

[1] D. Park and Y. Woo, "Edge preserving image compression with weighted centroid neural network," *J. KICS*, vol. 24, no. 10, pp. 1946-1952, Oct. 1999.

[2] K. Park, "Gaze detection system by IR-LED based camera," *J. KICS*, vol. 29, no. 4C, pp. 494-504, Apr. 2004.

[3] H. Park, "Active object tracking system for intelligent video surveillance," *J. KIECT,* vol. 2, no. 7, pp. 82-85, Jun. 2004.

[4] Y. Kang and C. Bae, "License plates detection using a gaussian windows," *J. KICS*, vol. 37, no. 9, pp. 780-785, Sept. 2012.

[5] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. CVPR,* pp. 1830-1837, Providence, RI, Jun. 2012.

[6] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. CVPR,* pp. 1822-1829, Providence, RI, Jun. 2012.

[7] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Proc. ICCV,* pp. 1436-1433, Kyoto, Japan, Oct. 2009.

[8] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. CVPR,* pp. 2042-2049, Providence, RI, Jun. 2012.

[9] W. Zhong, H. Lu, and M. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. CVPR,* pp. 1838-1845, Providence, RI, Jun. 2012.

[10] B. Han, D. Comaniciu, Y. Zhu, and L. Davis, "Sequential kernel density approximation and its application to real-time visual tracking," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 30, no. 7, pp. 1186-1197, Jul. 2008.

[11] D. Ross, J. Lim, and M. Yang, "Adaptive probabilistic visual tracking with incremental subspace update," in *Proc. ECCV,* pp. 470-482, Prague, Czech, May 2004.

[12] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. British Machine Vision Conf.,* pp. 6.1-6.10, Edinburgh, UK, Sept. 2006.

[13] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof, "Online multi-Class LPBoost," in *Proc. CVPR,* pp. 3570-3577, San Francisco, CA, Jun. 2010.

[14] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. and Machine Intell.,* vol. 33, no. 8, pp. 1619-1632, Dec. 2011.

[15] S. Hare, A. Saffari, and P. Torr, "Struck: structured output tracking with kernels," in *Proc. ICCV,* pp. 263-270, Barcelona, Spain, Nov. 2011.

[16] J. Gall, A. Yao, N. Razavi, L. V. Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 33, no. 11, pp. 2188-2202, Apr. 2011.

[17] S. Schulter, C. Leistner, P. Roth, L. Gool, and H. Bischof, "Online hough forests," in *Proc. British Machine Vision Conf.,* pp. 128.1-128.11, Dundee, UK, Jan. 2011.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst. 25,* pp. 1106-1114, Lake Tahoe, NV, Dec. 2012.

[19] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Machine Learning,* pp. 597-606, Lille, France, Jul. 2015.

[20] H. Nam and B. Han, "Learning multi-domain convolutional neural network for visual tracking," in *Proc. CVPR,* pp. 4293-4302, Las

Vegas, NV, Jun. 2016

[21] Y. Wu, J. Lim, and M. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 37, no. 9, pp. 1834-1848, Sept. 2015.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR,* pp. 580-587, Columbus, OH, Jun. 2014.

[23] C. P. Diehl and G. Cauwenberghs, "SVM incremental learning, adaptation and optimization," in *Proc. Int. Joint Conf. Neural Netw.,* pp. 2685-2690, Portland, OR, Jul. 2003.

[24] S. Karen, V. Andrea, and Z. Andrew, "Deep inside convolutional networks: visualising image classification models and saliency maps," in *Proc. ICLR Workshop,* Scottsdale, AZ, May 2013.

[25] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: delving deep into convolutional nets," in *Proc. British Machine Vision Conf.,* Nottingham, UK, Sept. 2014.

[26] M. Kristan, et al., "The visual object tracking VOT2014 challenge results," in *Proc. ECCV,* pp. 191-217, Zurich, Switzerland, Sept. 2014.

[27] M. Kristan, et al., "The visual object tracking VOT2013 challenge results," in *Proc. ICCVW,* pp. 98-111, Sydney, Australia, Dec. 2013.

[28] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for Matlab," in *Proc. ACM Multimedia,* Brisbane, Australia, Oct. 2015.

[29] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: bootstrapping binary classifiers by structural constraints," in *Proc. CVPR,* pp. 49-56, San Francisco, CA, Jun. 2010.

[30] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. British Machine Vision Conf.,* pp. 6.1-6.10, Edinburgh, UK, Sept. 2006.

김 인 성 (In-Sung Kim)

2015년 8월 : 고려대학교 전자 및 정보공학과 졸업

2015년 9월~현재 : 서강대학교 전자공학과 CAD & ES 연구실 석사과정

<관심분야> Machine learning, Neural Network

황 선 영 (Sun-Young Hwang)

1976년 2월 : 서울대학교 전자공학과 학사

1978년 2월 : 한국과학원 전기 및 전자공학과 공학석사

1986년 10월 : 미국 Stanford 대학 전자공학 박사

1976년~1981년 : 삼성반도체(주) 연구원, 팀장

1986년~1989년 : Stanford 대학 Center for Integrated System 연구소 책임연구원 및 Fairchild Semiconductor Palo Alto Research Center 기술자문

1989년~1992 : 삼성전자(주) 반도체 기술 자문

1989년 3월~현재 : 서강대학교 전자공학과 교수

<관심분야> SoC설계 및 framework 구성, CAD시스템, Com. Architecture 및 DSP System Design 등