특집논문 (Special Paper)
방송공학회논문지 제23권 제2호, 2018년 3월 (JBE Vol. 23, No. 2, March 2018)
https://doi.org/10.5909/JBE.2018.23.2.186
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

합성곱 신경망을 통한 강건한 온라인 객체 추적

길 종 인[a)], 김 만 배[a)‡]

Robust Online Object Tracking via Convolutional Neural Network

Jong In Gil[a)] and Manbae Kim[a)‡]

요 약

본 논문에서는 객체를 추적하기 위해 합성곱 신경망 모델을 이용한 온라인 추적 기법을 제안한다. 오프라인에 모델을 학습시키기 위해서는 많은 수의 훈련 샘플이 필요하다. 이러한 문제를 해결하기 위해, 학습되지 않은 모델을 사용하고, 실험 영상으로부터 직접 훈련 샘플을 수집하여 모델을 갱신한다. 기존의 방법들은 많은 훈련 샘플을 획득하여 모델의 학습에 사용하였지만, 본 논문에서는 적은 수의 훈련 샘플만으로도 객체의 추적이 가능함을 증명한다. 또한 컬러 정보를 활용하여 새로운 손실 함수를 정의하였고 이로부터 잘못 수집된 훈련 샘플로 인해 모델이 잘못된 방향으로 학습되는 문제를 방지한다. 실험을 통해 4가지 비교 방법과 동등하거나 개선된 추적 성능을 보임을 증명하였다.

Abstract

In this paper, we propose an on-line tracking method using convolutional neural network (CNN) for tracking object. It is well known that a large number of training samples are needed to train the model offline. To solve this problem, we use an untrained model and update the model by collecting training samples online directly from the test sequences. While conventional methods have been used to learn models by training samples offline, we demonstrate that a small group of samples are sufficient for online object tracking. In addition, we define a loss function containing color information, and prevent the model from being trained by wrong training samples. Experiments validate that tracking performance is equivalent to four comparative methods or outperforms them.

Keyword : visual tracking, convolutional neural network, on-line tracking, probability map, color histogram

a) 강원대학교 컴퓨터정보통신공학과(Department of Computer and Communications Eng., Kangwon National University)
‡ Corresponding Author : 김만배(Manbae Kim)
E-mail: manbae@kangwon.ac.kr
Tel: +82-33-250-6395
ORCID: http://orcid.org/0000-0002-4702-8276
※ 이 논문의 연구결과 중 일부는 “한국방송·미디어공학회 2017년 추계학술대회”에서 발표한 바 있음.
※ 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2017R1D1A3B03028806).
※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2017R1D1A3B03028806).
· Manuscript received November 28, 2017; Revised January 24, 2018; Accepted January 24, 2018.

Ⅰ. 서 론

객체 추적은 보안 감시 시스템 등의 분야에서 많은 응용을 가질 수 있는데, 기존의 객체 추적 방법은 컬러와 같은 특징을 이용한 템플릿 매칭으로부터 시작되어 컬러 기반 추적[1], 그래디언트 기반 추적[2,3], 커널 기반 추적[4], 움직임 기반 추적[5] 등 다양한 연구가 수행되어왔다. 그러나 사람과 같은 객체는 움직이면서 그 외형이 변하거나 조명 등으로 인해 색이 변하므로 이를 해결하기 위해 객체 모델을 갱신하는 적응적 객체 추적 기법이 연구되고 있다[6]. 이러한 추적 기법은 제한된 환경을 설정해놓고, 해당 상황에서 좋은 추적 성공률을 보일 수 있도록 설계되었다. 그러나 추적하고자 하는 객체가 존재하는 실험 영상 혹은 실제 장면은 여러 가지 시나리오를 가질 수 있고, 화질 및 해상도 또한 다양하다. 그러므로 앞서 언급한 추적기법들은 이러한 모든 상황에 대해서 높은 성공률을 보장할 수 없다.

최근에는 객체 검출과 추적을 결합한 방법 또한 연구되고 있는데, 이러한 방법을 검출에 의한 추적(Tracking-by-Detection)이라 한다. 객체를 검출하기 위한 검출기를 학습하기 위해 여러 가지 기계학습법이 사용될 수 있다. 이러한 검출에 의한 추적 기법은 모델을 사전에 학습하는 방법과 사전에 학습되지 않은 모델을 사용하는 방법으로 나뉠 수 있다. 사전에 학습된 모델을 사용할 경우, 모델의 학습을 위해 많은 수의 훈련 샘플이 필요하고, 이는 큰 비용을 초래한다. 비록 충분한 양의 훈련 샘플이 충족되었다 할지라도, 높은 정확도를 갖는 분류기의 학습을 위해 많은 시간이 필요하다. 그러나 일단 많은 수의 훈련 샘플이 확보가 된다면 해당 모델은 범용적인 목적으로 사용될 수 있다. 모델을 사전에 학습하지 않는 경우에는, 사전에 훈련 샘플을 준비하지 않아도 된다는 장점이 있다. 이는 실용성이 크므로 큰 장점이 될 수 있다. 추적을 수행하면서 순차적으로 훈련 샘플을 획득하고 모델을 업데이트한다. 이러한 방법은 비교적 최신에 획득한 훈련샘플에 과적합(overfitting)될 가능성이 있다. 그러나 이러한 문제는 객체의 추적에 있어서는 비교적 큰 문제가 되지 않는다.

Kalal 등은 이러한 검출에 의한 추적을 위해 TLD를 제안하였다[7,8]. 추적기와 검출기를 독립적으로 생성하고, 추적기는 연속된 프레임으로부터 움직임으로부터 객체를 추적하고, 검출기는 학습된 랜덤 포레스트(Random Forest) 모델을 이용하여 객체를 검출한다. 추적기가 객체를 추적함과 동시에 검출기도 계속 객체를 검출함으로써, 추적이 실패하였을 때를 대비할 수 있다. Babenko 등은 AdaBoost를 이용하여 객체를 온라인으로 추적하였다. 온라인으로 객체를 추적할 때, 항상 정확하게 객체의 위치를 예측하는 것은 불가능하다. 이러한 오차가 점차 누적되면 추적기는 표류(drift)현상이 발생하여 추적에 실패하게 된다. 이를 해결하기 위해 MIL(Multiple Instance Learning)이 제안되었다[9].

최근에는 현재 컴퓨터비전 분야에서 가장 널리 활용되는 합성곱 신경망(Convolutional Neural Network: CNN)을 활용한 온라인 객체 추적 기법이 연구되고 있다[10-12]. CNN은 객체 검출 및 인식에 탁월한 성능을 보여주는 것으로 알려져 있다. 이러한 방법들은 모두 검출에 의한 추적 메커니즘을 적용하고 있다. 본 논문에서는 클래스가 정해진 훈련 집합이 불필요한 온라인 학습 기반 추적 기법을 제안한다. 객체의 추적을 위한 모델로써 합성곱 신경망을 활용한다. 제

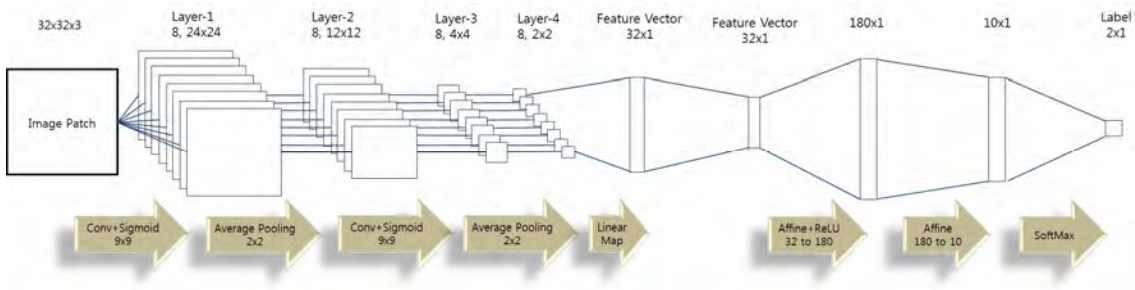


그림 1. 추적을 위한 CNN의 구조
Fig. 1. Architecture of CNN for visual tracking

안방법에서는 프레임마다 6개의 훈련 샘플만을 획득한다. 게다가, 모델을 학습시키기 위해서 많은 연산을 필요하기 때문에, 본 논문에서는 CNN을 이용하여 객체를 추적하는 과정에서 작은 수의 훈련샘플만으로도 CNN 모델을 충분히 학습시킬 수 있도록 하였다. 다음 그림 1은 객체 추적을 위한 CNN의 구조를 보여준다.

CNN은 두 개의 합성곱층(convolutional layer)과 두 개의 완전결합층(fully connected layer)으로 구성된다. 활성 함수로서 시그모이드가 사용되었고, 차원 축소를 위해 평균값 풀링이 사용되었다. 획득한 이미지 패치는 32x32x3의 크기로 변경되어 입력된다. 첫 번째와 두 번째 합성곱에서는 각각 3x3의 크기를 갖는 필터가 8개씩 존재한다. 패치에 콘볼루션이 적용되면 노드의 수는 30x30x8로 변경되고, 다시 평균값 풀링을 통해 15x15x8로 바뀐다. 동일한 과정을 한번 더 거치게 되면 노드의 수는 7x7x8로 축소되고, 392개의 노드가 완전결합층에 입력된다. 완전결합층에는 두 개의 은닉층(hidden layer)이 존재하고 각각의 층에는 100, 80개의 노드가 있다. 완전결합층의 마지막에는 활성함수로서 소프트맥스가 사용되고, 모멘텀을 적용하였다.

II. 훈련 샘플 수집

본 논문에서 제안하는 방법은 영상으로부터 훈련 샘플을 직접 수집한다. 수집된 훈련샘플은 모델을 학습하는데 이용된다. 이러한 방법은 많은 장점을 가질 수 있다. 먼저 사전에 훈련 샘플을 이용하여 검출기를 학습하는 경우, 훈련샘플과 실험영상에 많은 차이가 있을 때 (예를 들어, 해상도 및 화질의 차이) 충분한 성능을 내지 못할 가능성이 크다. 그러나, 실험영상에서 훈련 샘플을 직접 수집하게 되면 이러한 차이로부터 발생하는 성능의 차이를 극복할 수 있다. 또한, 검출기의 사전 훈련을 위해서는 많은 수의 훈련 샘플이 필요하다. 이렇게 많은 수의 샘플을 생성하는 것은 많은 비용이 필요하다. 따라서 실험영상으로부터 온라인으로 훈련샘플을 수집함으로써 이러한 문제의 해결이 가능하다.

제안방법에서 객체 추적을 위한 초기화는 필요하지 않다. 단 첫 프레임에서 추적될 객체의 위치는 알려져 있다고 가정한다. 해당 객체의 위치를 (x,y), 객체의 높이와 너비를 (w,h)이라 할 때, 해당 위치를 중심으로 바운딩 박스를 설정할 수 있다. 또한 획득한 이미지 패치에 좌우 반전을

그림 2. 추적되는 객체로부터 얻어지는 긍정 및 부정 바운딩 박스 (적색: 긍정 이미지 패치, 청색: 부정 이미지 패치)
Fig. 2. Positive and negative bounding boxes obtained from a tracked object. (red: positive image patch, blue: negative image patch)

수행함으로써 두 개의 긍정(positive) 이미지 패치를 획득할 수 있다.

추적 객체의 위치로부터 상하좌우 네 방향에 대해 동일한 크기의 바운딩 박스를 취한다. 즉, 네 위치의 좌표는 $(x\pm w, y\pm h)$가 된다. 이 위치에서 동일한 크기의 바운딩 박스를 생성하고, 이미지 패치의 클래스를 부정(negative)으로 설정한다. 이로써, 총 2개의 긍정 샘플과 4개의 부정 샘플을 획득할 수 있다. 그림 1과 같이 추적 객체가 영상 내부에 존재하는 동안 매 프레임마다 6개의 훈련 샘플을 획득한다.

훈련 샘플은 CNN의 학습 성능에 큰 영향을 미친다. 그러나 다양한 실험 영상에서는 조명의 변화 등으로 인해 밝기에 차이가 발생하는 경우가 존재한다. 이 문제를 해결하기 위해 훈련 샘플에 전처리 과정으로서 화이트닝(whitening) 변환을 수행하였다. 화이트닝은 다음 식 (1)로 정의할 수 있다.

$$y^T = \Lambda^{-\frac{1}{2}} \Phi^T x^T \tag{1}$$

여기서 x는 입력영상, y는 변환영상을 의미한다. Φ는 $d\times d$ 행렬로, 입력영상으로부터 계산된 공분산행렬로부터 획득한 고유벡터를 열벡터로 구성한 행렬이다. Λ는 대각행렬이고, i번째 대각요소는 i번째 고유값이다. 그림 3은 수집한 훈련 샘플에 화이트닝 변환을 수행한 결과를 보여주고 있다.

Ⅲ. CNN 모델 학습

CNN 모델로부터 현재 프레임의 객체 위치가 결정되면, 새로운 훈련샘플을 수집하여 CNN 모델을 갱신한다. Ⅱ장에서 설명한 바와 같이 훈련샘플을 수집할 수 있는데, 수집된 샘플의 신뢰도는 현재 CNN 모델의 정확성에 따라 결정된다. 만일 CNN 기반의 추적기가 현재 프레임에서의 객체의 위치를 올바르게 추적하지 못했다면, 훈련 샘플도 올바르게 수집되지 못할 것이다. 이를 위해 multiple instance learning 방법 등이 제안되었다[9]. 본 논문에서는 이러한 문제를 해결하기 위해 MeanShift 추적기에서 활용하는 컬러 모델을 이용한다[13]. 먼저 프레임 $t-1$에서 객체의 위치가 올바르게 추적되었다고 가정했을 때, 해당 패치로부터 컬러 히스토그램을 구한다. 히스토그램을 사각형 형태의 패치로부터 직접 생성할 경우, 배경이 히스토그램에 영향을 미칠 수 있다. 따라서 사각 바운딩 박스 대신 타원 형태로 경계를 설정하여 히스토그램을 생성한다. 다음으로 프레임 t에서 획득한 히스토그램을 역투영하여 확률맵(probability map) BP를 획득한다. 그림 4는 확률맵과 역확률맵을 보여주고 있다.

BP로부터 적분확률맵(Integral Probability Map) P는 다음 식에서 구해진다.

$$\begin{aligned} P_p(x_n, y_n) &= \sum_{a=0}^{h_n-1}\sum_{b=0}^{w_n-1} BP_p(x_n+a, y_n+b) \\ P_n(x_n, y_n) &= \sum_{a=0}^{h_n-1}\sum_{b=0}^{w_n-1} BP_n(x_n+a, y_n+b) \end{aligned} \tag{2}$$

여기서 BP_p는 확률맵, BP_n은 역확률맵이다. P_p는 BP_p의 합으로 얻어진 적분확률맵이고, P_n은 역적분확률맵이다.

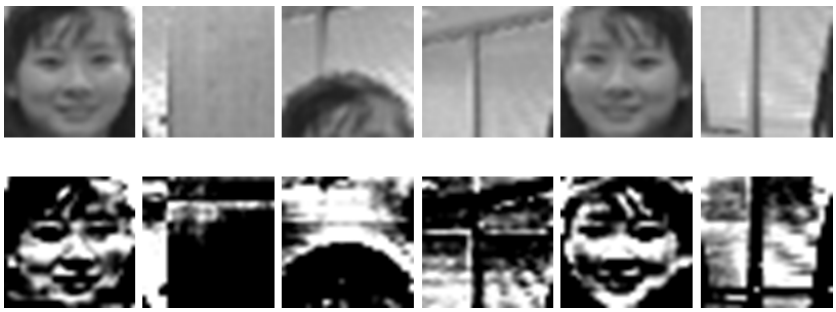


그림 3. 화이트닝 변환 (1행 : 원본 영상, 2행: 변환 영상)
Fig. 3. Whitening transformation (top row: original images, bottom row: transformed images)

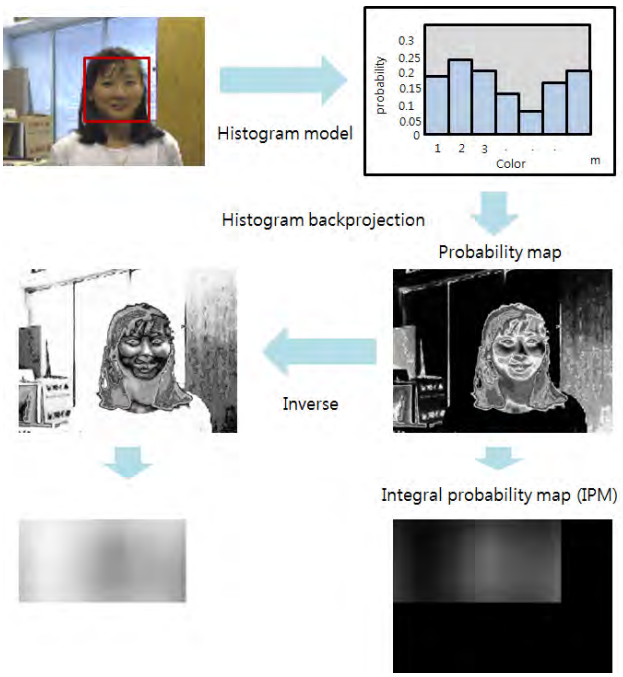


그림 4. 적분 확률맵 생성 과정
Fig. 4. Procedure of generating Integral Probability Map

그림 4와 같이 확률맵에서 나타나는 픽셀 값은 샘플의 레이블이 긍정일 가능성으로 해석할 수 있다. 반대로 그의 역확률맵은 훈련 샘플이 부정 패치일 가능성을 나타낸다고 볼 수 있다. 확률맵은 식 (2)을 이용하여 확률맵의 이미지 패치 내부의 모든 픽셀 값을 합산한 적분확률맵으로 변환한다. 적분확률맵의 특성을 이용하여 새로운 손실함수를 정의한다. 식 (3)은 손실 함수로써 주로 사용되는 Cross entropy이다.

$$J(l_n,d_n)=\sum_{i=1}^{M}\{-d_{n,i}\ln(l_{n,i})-(1-d_{n,i})\ln(1-l_{n,i})\} \quad (3)$$

여기서 ln은 자연로그이다. M은 출력 노드의 수이고, i는 출력 노드의 인덱스이다. 여기에서는 2-class만 존재하므로 $M=2$이다. 또한, n은 훈련 샘플의 인덱스이다. d_n은 훈련 샘플의 레이블(ground truth), l_n은 CNN의 출력 벡터이다. 식 (2)에서 획득한 확률값을 이용하여 식 (4)의 새로운 손실 함수를 정의한다. 이렇게 생성된 식 (3)과 (4)를 결합하여 식 (5)의 최종 손실 함수를 정의한다.

$$C(l_n,d_n)=\frac{1}{N}\sum_{n=1}^{N}\{d_nP_p(x_n,y_n)+(1-d_n)(1-P_n(x_n,y_n))\} \quad (4)$$

$$L(l_n,d_n)=C(l_n,d_n)\times J(l_n,d_n) \quad (5)$$

여기서 N은 훈련 이미지 패치의 수이고, 실험에서는 $N=6$으로 설정하였다. (x_n,y_n)은 n번째 훈련 이미지 패치의 위치, (w_n,h_n)은 너비와 높이이다.

이미지 패치의 레이블이 긍정일 경우, 해당 패치가 객체의 컬러모델과 유사한 색상 분포를 가지게 되면 CNN 모델 갱신에 있어서 상대적으로 높은 가중치를 얻게 된다. 만일 어떤 패치가 객체의 컬러 모델과 유사하지 않다면, 그것은 추적하고자 하는 객체일 가능성이 현저히 낮아진다. 따라서 해당 패치의 레이블이 부정일 경우, CNN 모델 갱신에 있어서 높은 가중치를 얻게 된다. 즉, 긍정 패치이지만 컬러모델과 큰 차이를 갖거나, 부정 패치이지만 컬러 모델과 유사한 경우에는 모호한 경우로 규정하고 낮은 가중치를 할당함으로써 CNN 모델 갱신의 오류를 줄이고자 하였다.

그러나 식 (5)의 손실 함수를 이용하여 CNN 모델을 학습하여도 학습에 사용된 훈련 패치의 수는 6개이므로 적은 수의 훈련으로는 CNN 모델이 원하는 값으로 수렴될 수 없다. 이러한 이유로 인해, 일반적으로 CNN 모델을 학습시킬 때, 전체 학습 데이터에 대해서 한번만 학습시키는 것이 아니라 재학습을 시키게 된다. 이렇게 전체 학습 데이터를 한 번씩 모두 학습시킨 횟수를 에폭(epoch)이라 한다. 보통 에폭을 고정시켜놓고 학습을 진행한다. 앞선 과정을 통해 훈련샘플을 수집한 후에, 전체 데이터를 한 번씩 학습할 때마다 식 (6)을 이용하여 훈련 오차를 계산한다.

$$e=\frac{1}{N}\sum(l_n-d_n)^2 \quad (6)$$

이러한 오차가 임계치보다 작을때까지 훈련을 반복한다. 즉, 프레임 t에서 객체의 위치를 예측하고, 이로부터 훈련 샘플을 수집하여 모델을 학습한 후, 훈련에 사용된 훈련 샘플을 이용하여 모델을 테스트한다. 훈련샘플로부터 모델의 오차가 수렴할 때까지 모델의 훈련을 반복한다.

Ⅳ. 객체 추적

새로운 프레임이 입력되면 학습된 CNN 모델을 이용하여 추적하고자 하는 객체의 위치를 예측한다. 이전 프레임에서 객체의 위치를 중심으로 탐색 범위를 설정한다. 탐색 범위의 수평 길이를 s_h, 수직 범위를 s_v라 할 때, 해당 탐색 범위에서 총 $s_h \times s_v$개의 후보 이미지 패치를 획득할 수 있다. 획득한 모든 패치에 대해 학습된 CNN 모델로 입력되면, 2D 벡터의 형태로 출력이 된다. 출력된 벡터는 $v = \{s_p, s_n\}$의 형태를 갖는다. s_p는 긍정 확률, s_n은 부정 확률이다. 출력에 소프트맥스를 적용하였으므로, $s_p + s_n = 1.0$이다. 따라서 결국 s_p가 해당 패치가 긍정일 확률이라고 규정할 수 있다. 여기에서는 s_p를 점수로 규정하여, 가장 높은 점수를 갖는 곳에 객체가 위치해있다고 판단할 수 있다.

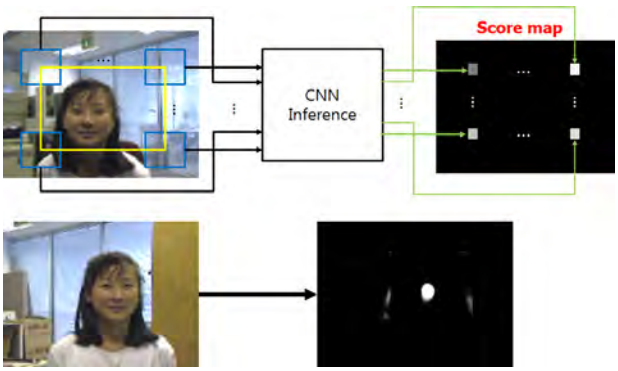


그림 5. 객체 탐색 영역(노란색)과 후보 이미지 패치(청색)
Fig. 5. Object search range (yellow) and candidate image patch(blue)

모든 후보 이미지 패치에 대해 점수를 획득하고, 이를 영상의 픽셀에 할당하면 점수맵(score map)을 획득할 수 있다. 이를 그림 5에서 보여주고 있다. 탐색 범위를 노란색으로 나타내었고, 개별 이미지 패치들을 파란색으로 표현하였다. 첫 프레임에서는 모델이 학습되어있지 않기 때문에 점수맵을 획득할 수 없다. 학습된 CNN 모델은 두 번째 프레임에서부터 적용이 가능하므로, $t = 2$부터 점수를 획득할 수 있다. 그림 5에서 보는바와 같이 추적하고자 하는 객체의 근처에서 높은 점수가 나타남을 확인할 수 있다. 그러나 1.0의 점수를 갖는 위치는 많이 존재한다. 따라서 점수맵의 무게 중심(center of gravity)을 예측된 위치로 판단하였다. 무게 중심은 다음과 같이 계산된다.

$$m_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} i^p j^q f(i,j) \tag{6}$$

$$x_c = \frac{m_{10}}{m_{00}}, y_c = \frac{m_{01}}{m_{00}} \tag{7}$$

$t = 2$일 때, 객체의 위치를 찾았다면, 그 위치로부터 다시 긍정샘플과 부정샘플을 앞서 기술한 바와 동일하게 획

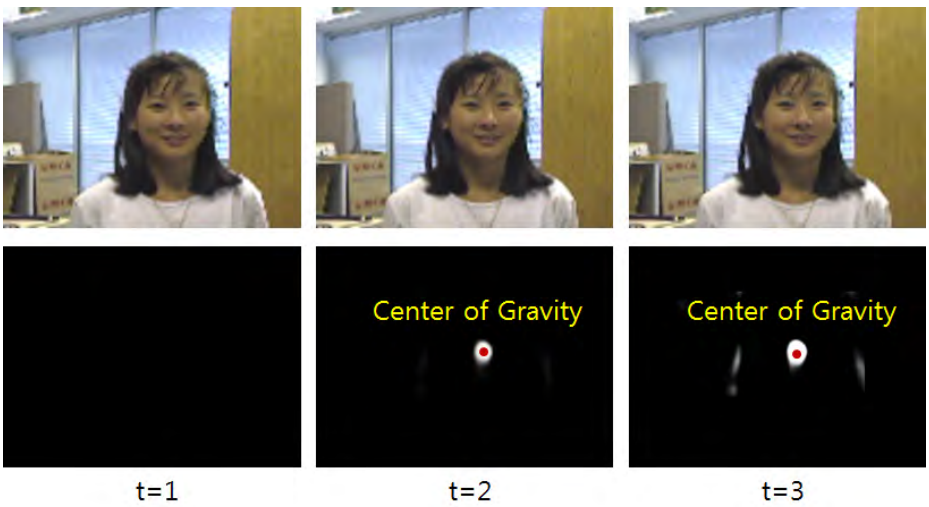
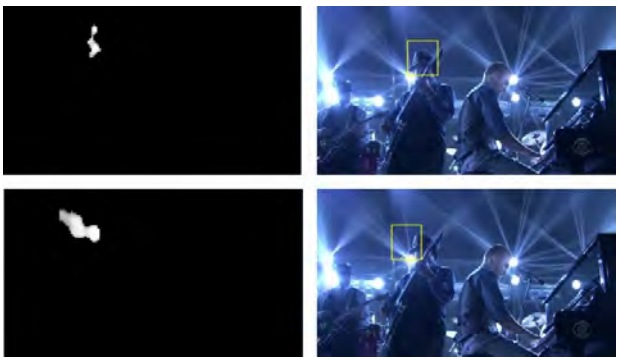


그림 6. 입력 영상과 점수맵
Fig. 6. Imput images and score maps

득한다. 이로부터 CNN 모델을 갱신하고, $t=3$일 때의 점수맵을 획득하여 객체의 위치를 예측한다. 이와 같은 과정을 반복함으로써 연속적으로 객체의 위치를 추적하게 된다.

여기에서는 앞서 설명한 화이트닝 변환의 효과에 대해 설명한다. 그림 7에서 화이트닝 변환의 적용 여부에 따른 점수맵과 추적결과의 차이를 비교하여 보여주고 있다. 해당 영상은 무대조명으로 인해 추적하고자 하는 객체의 명암에 큰 변화가 존재하는 실험 영상이다. 이러한 영상으로부터 훈련 샘플을 획득하게 되면 추적을 수행할 때 후보 이미지 패치들간에 변별력이 사라지는 것을 그림 7(a)에서 확인할 수 있다. 따라서 그림 7(b)에서와 같이 정확한 추적에 실패하게 된다.



(a) Score map (b) tracking result

그림 7. 화이트닝 변환에 따른 점수맵 및 추적 결과의 차이
Fig. 7. Difference in score map and tracking result by whitening

V. 실험 결과

제안하는 CNN 기반 추적기는 CVPR2013 Visual Tracker Benchmark의 데이터 중 일부를 이용하여 실험하였다[14]. 그림 8은 4개의 실험영상에 제안 방법을 이용하여 객체를 추적한 결과를 보여준다.

그림 9는 제안 방법을 이용하여 추적을 수행한 결과와

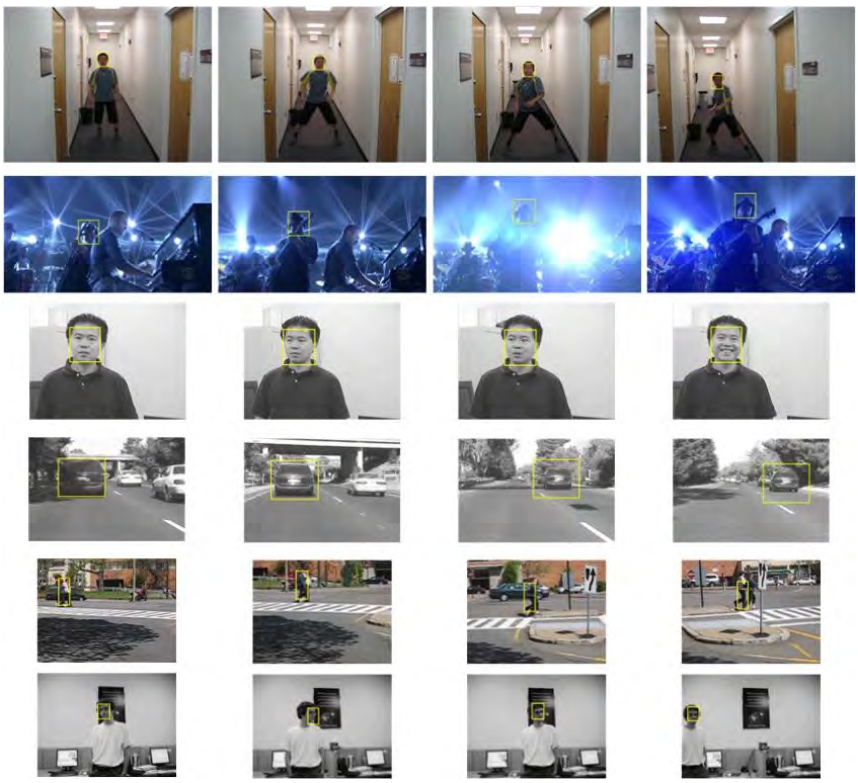


그림 8. 제안하는 방법의 실험 결과
Fig. 8. Experimental results of proposed method

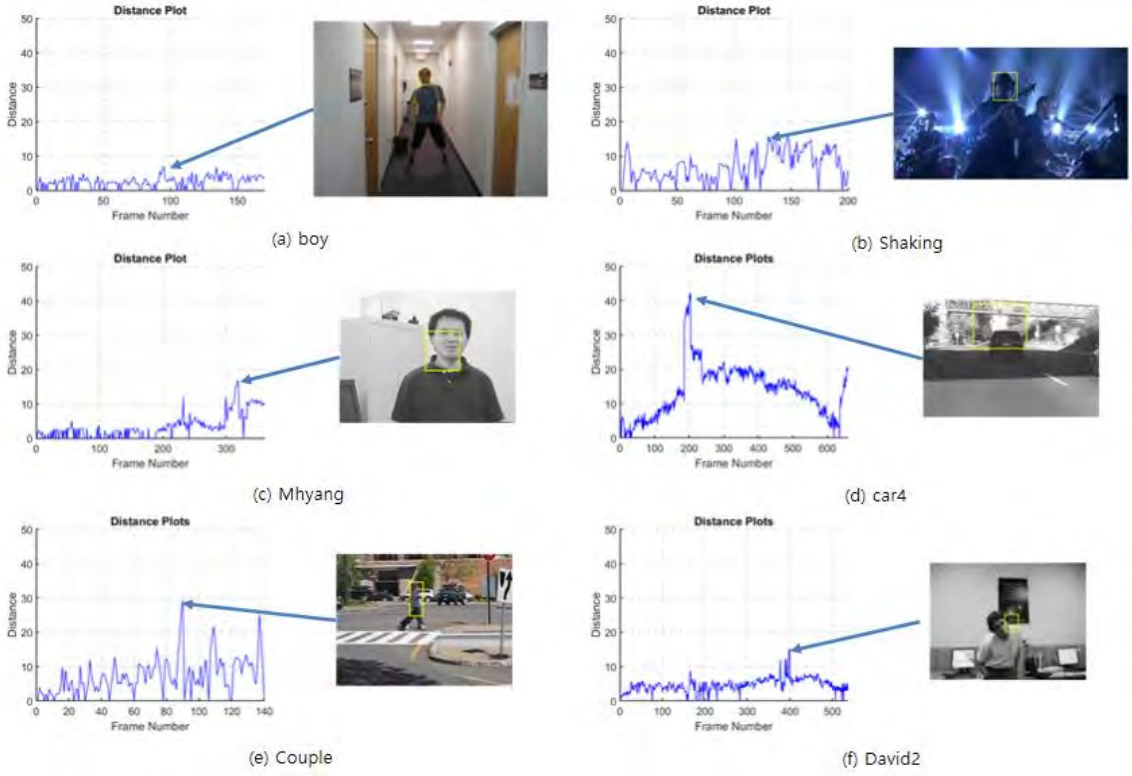


그림 9. 추적 거리 오차
Fig. 9. Tracking distance error

실측(ground-truth)의 거리 오차를 측정한 결과를 보여주고 있다. 그래프의 가로축은 프레임 번호, 세로축은 두 위치 사이의 거리를 나타낸다. 각각의 결과에 대해서 가장 큰 오차를 보였던 프레임을 오른편에서 보여주고 있다. *boy*는 가장 큰 오차를 보이는 부분이 10 픽셀 이하였다. 이로부터 추적이 원활하게 수행되었음을 알 수 있다. *Shaking*은 조명에 큰 변화를 가지고 있는 실험 영상이다. 그럼에도 불구하고 전체적으로 안정적으로 추적을 수행하고 있다. *Mhyang*에서는 *boy*에서와 유사하게 좋은 추적 결과를 보이고 있으며, *car4*에서는 전체적으로 변화가 큰 결과를 보이고 있다. 이로부터 알 수 있는 사실은, 중간에 추적하고자 하는 목표로부터 살짝 벗어나게 될지라도 모델에 오차가 누적되어 표류현상이 발생하지 않고 다시 올바르게 제자리로 돌아갈 수 있는 능력이 있음을 보여주고 있다. *Couple* 영상은 두 사람이 함께 걸어가고 있는 영상인데 비교적 *Shaking*과 유사한 결과를 보이고 있다. *David2*에서는 *boy*에서처럼 가장 훌륭한 성능을 보여주고 있다.

제안 방법의 객관적인 성능평가를 위해 4가지의 기존 알고리즘과 비교평가를 수행하였다. 비교 알고리즘인 CFP는 전통적인 컬러를 이용한 객체 추적 기법이고, MIL, OAB, TLD는 기계학습 모델을 활용한 온라인 객체 추적 기법이다. 이들은 각각 Color-based Probabilistic Tracking[1], Multiple instance learning[9], On-line AdaBoost[15], Tracking-Learning-Detection[8] 알고리즘이다.

표 1은 3개의 실험영상에 대하여 각 프레임마다 추적기가 예측한 객체의 위치와 실측 사이의 거리 오차로부터 평균값을 측정한 결과를 보여주고 있다. 거리는 L2-norm을 이용하여 계산하였다. 값이 낮을수록 우수한 추적기라고 판단할 수 있다. *Boy*에서는 제안방법과 기존 방법 모두 우수한 결과를 보여주고 있다. 제안방법에서는 최소한의 훈련샘플을 이용함으로써 모델을 학습하는데 최대한의 효율을 가질 수 있도록 설계하였다. 이를 통해 적은 수의 훈련샘

표 1. 거리 오차의 평균
Table 1. Mean value of distance error

Sequence	Proposed method	CFP	MIL	OAB	TLD
Boy	2.4725	2.0587	2.3752	1.0126	2.6546
Shaking	7.0906	60.9270	13.0536	113.7921	1.7704
Mhyang	3.3691	8.4343	2.5483	1.4461	1.6057
car4	13.6391	20.5605	23.9314	67.9511	11.6711
Couple	7.2579	5.1237	9.6042	33.6817	2.1000
David2	4.6815	3.6372	5.1816	18.3104	2.2713

플로도 효과적으로 객체를 추적할 수 있음을 확인할 수 있다. *Shaking*, *car4*에서는 제안 방법이 비교 알고리즘에 비해 월등히 우수한 결과를 보여주고 있다. 해당 실험 영상은 음악 공연에서 흔들리고 있는 머리를 추적해야 하는 영상인데, 배경 영역에서 조명으로 인해 밝기의 변화가 매우 심하게 나타나는 영상이다. 기존의 방법에 비해 제안하는 방법이 조명에 강건하므로 비교적 좋은 결과를 가질 수 있었다. *Mhyang*에서는 CFP보다는 비교적 매우 좋은 결과를 보이고 있고, 나머지 방법들과는 유사한 결과를 보이고 있다. *Couple*, *David2*는 비교 알고리즘들과 동등한 성능을 나타내고 있다.

추적 정확도를 측정하기 위해 추정된 위치가 ground-truth와의 거리 임계치 안에 존재하는 프레임의 비율을 측정하였다. 즉, 예측 오차가 거리 임계치보다 크면 추적 실패, 거리 임계치보다 작으면 성공으로 판단하였고, 이를 비율로 환산하였다. 거리 임계치는 0에서부터 50까지 달리해 가며 측정하였다. 측정한 추적 정확도를 그림 9에서 보여주고 있다. 실험영상 모두 그림 9에서 보았던 거리오차와 비

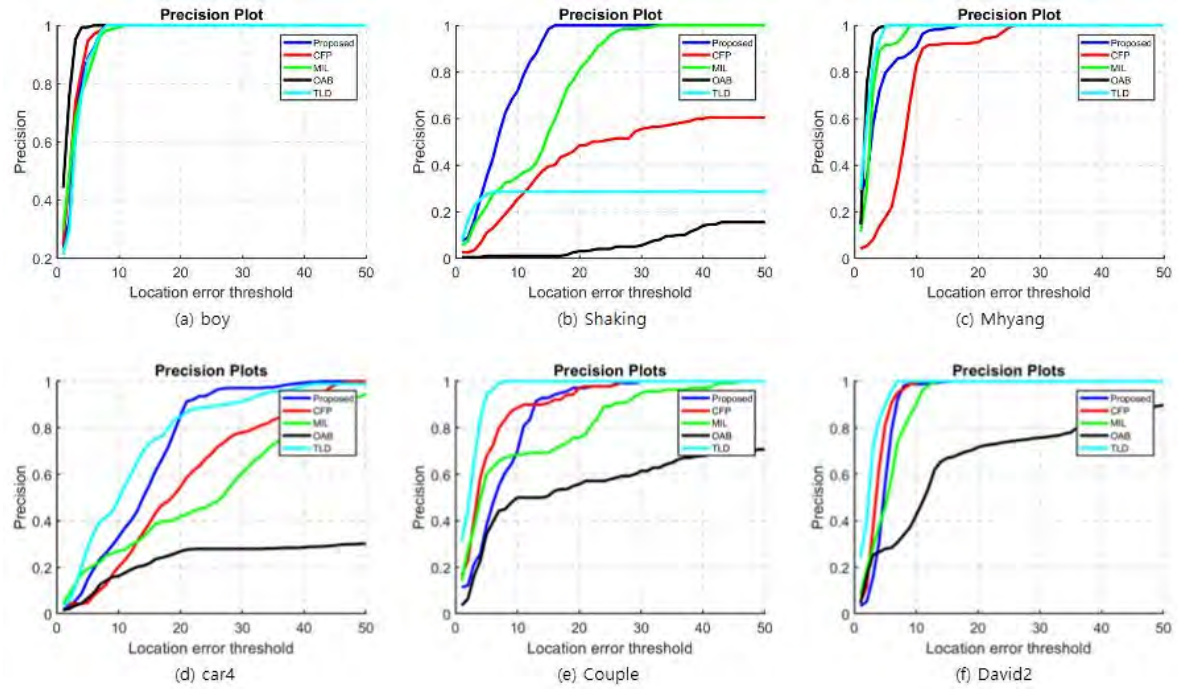


그림 10. 추적 정확도
Fig. 10. Tracking precision

슷한 양상을 보이고 있다. *Boy*에서는 대부분의 알고리즘이 좋은 정확도를 보이고 있었다. *Shaking*와 *car4*에서는 제안 방법의 우수함을 거리오차보다 더 명확하게 보여주고 있다. 해당 그래프 아래의 면적이 클수록 좋은 추적기임을 나타낸다. 제안방법이 가장 큰 면적을 가짐을 확인할 수 있다. *Mhyang*에서는 CFP보다는 좋고 MIL, OAB, TLD와는 유사한 성능을 보이고 있다. 나머지 실험 영상들에서는 OAB가 가장 안좋은 성능을 보이고 있고, 나머지 알고리즘들은 유사한 결과를 보이고 있다.

Ⅵ. 결 론

본 논문에서는 합성곱 신경망 모델을 이용하여 객체를 추적하는 기법을 제안하였다. 기존의 온라인 객체 추적에서는 많은 수의 훈련 집합을 필요로 하지만 제안 기법에서는 적은수의 훈련 집합으로도 충분히 학습 및 추적이 가능함을 증명하였다. 또한 컬러 정보를 결합하여 새로운 손실함수를 정의함으로써, CNN 모델 학습의 효율을 향상시켰다. 대부분의 객체 추적 알고리즘은 오랜 시간 객체를 추적하게 되면 오차가 누적되어 추적에 실패하게 되는 표류 현상이 발생한다. 많은 다른 알고리즘들에서는 이러한 문제를 해결하기 위해 재 검출과 같은 방법을 추가로 적용하였지만, 제안 방법에서는 이러한 방법을 적용하지 않았다. 따라서 비록 추적기가 완전히 객체를 벗어나게 되면 추적에 실패하게 되지만, 작은 차이로 인해 오차가 발생하더라도 쉽게 표류 현상이 발생하지 않는다. 추적에 실패했을 때, 객체를 재검출하는 기법을 도입한다면, 장기간 실패 없이 객체를 추적하는 것이 가능할 것으로 기대된다. 실험 결과를 통해 적은 훈련 집합을 사용했음에도 불구하고, 동등하거나 더 우수한 성능을 가짐을 확인할 수 있었다.

참 고 문 헌 (References)

[1] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, “Color-Based Probabilistic Tracking”, *Computer Vision-ECCV,* pp. 661-675, 2002

[2] D. Bruch and K. Takeo, “An Iterative Image Registration Technique with an Application to Stereo Vision”, *Int’ Joint Conf. on Artificial Intelligence*, pp. 674-679, Aug. 1981

[3] T. Carlo and K. Takeo, “Detection and Tracking of Point Features”, *Technical Report CMU-CS-91-132*, 1991

[4] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-Based Object Tracking”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, pp. 564-577, May 2003

[5] K. Lee, S. Ryu, S. Lee, and K. Park, “Motion based object tracking with mobile camera“, *Electronics Letters*, Vol. 34, No. 3, pp. 256-258, 1998.

[6] Y. Wu, J. Lim, and M. Y, “Object Tracking Benchmark”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 37. No. 9, pp. 1834-1848, Sep. 2015

[7] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints”, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 49-56, 2010

[8] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-Learning-Detection”, I*EEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 7, pp. 1409-1422, July 2012

[9] B. Babenko, M. Yang, and S. Belongie, “Robust Object Tracking with Online Multiple Instance Learning”, I*EEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 8, pp. 1619-1632. Aug. 2011

[10] H. Li, Y. Li, and F. Porikli, “DeepTrack: Learning Discriminative Feature Representations Online for Robust Visual Tracking“, *IEEE Trans. on Image Processing*, Vol. 25, No. 4, pp. 1834-1848, April 2016.

[11] K. Zhang, Q. Liu, and M. Yang, “Robust Visual Tracking via Convolutional Networks Without Training”, *IEEE Trans. on Image Processing*, Vol. 25, No. 4, pp. 1779-1792, April 2016.

[12] X. Zhou, L. Xie, P. Zhang, and Y. Zhang, “An Ensemble of Deep Neural Networks for Object Tracking”, *IEEE Conf. on Image Processing*, pp. 843-847, 2014.

[13] D. Comaniciu, V. Ramesh, and P. Meer, “Real-Time Tracking of Non-Rigid Objects using Mean Shift”, *IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 142-149, 2000.

[14] Y. Wu, J. Lim, and M. H. Yang, “Online object tracking: A benchmark”, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2411-2418, 2013.

[15] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised On-Line Boosting for Robust Tracking”, *British Machine Vision Conf.*, Vol. 1, No. 5, pp. 6. 2006.

저 자 소 개

길 종 인

- 2010년 8월 : 강원대학교 컴퓨터정보통신공학과 학사
- 2012년 8월 : 강원대학교 컴퓨터정보통신공학과 석사
- 2012년 9월 ~ 현재 : 강원대학교 IT대학 컴퓨터정보통신공학과 박사과정
- 주관심분야 : 객체 트랙킹, 얼굴인식, 점유센서, 머신러닝

김 만 배

- 1983년 : 한양대학교 전자공학과 학사
- 1986년 : University of Washington, Seattle 전기공학과 공학석사
- 1992년 : University of Washington, Seattle 전기공학과 공학박사
- 1992년 ~ 1998년 : 삼성종합기술원 수석연구원
- 1998년 ~ 현재 : 강원대학교 IT대학 컴퓨터정보통신공학과 교수
- 2016년 ~ 현재 : 강원대학교 정보통신연구소 소장
- ORCID : http://orcid.org/0000-0002-4702-8276
- 주관심분야 : 3D영상처리, 비전점유센서, 객체트랙킹