

Siame-FPN기반 객체 특징 추적 알고리즘

김종찬[†], 임수창^{**}

Object Feature Tracking Algorithm based on Siame-FPN

Jong-Chan Kim[†], Su-Chang Lim^{**}

ABSTRACT

Visual tracking of selected target objects is fundamental challenging problems in computer vision. Object tracking localize the region of target object with bounding box in the video. We propose a Siam-FPN based custom fully CNN to solve visual tracking problems by regressing the target area in an end-to-end manner. A method of preserving the feature information flow using a feature map connection structure was applied. In this way, information is preserved and emphasized across the network. To regress object region and to classify object, the region proposal network was connected with the Siamese network. The performance of the tracking algorithm was evaluated using the OTB-100 dataset. Success Plot and Precision Plot were used as evaluation matrix. As a result of the experiment, 0.621 in Success Plot and 0.838 in Precision Plot were achieved.

Key words: Computer Vision, Convolutional Neural Networks, Deep Learning, Object Tracking, Siamese Network

1. 서 론

시각 기반 객체 추적 알고리즘은 컴퓨터 비전에서 주목받는 분야이다. 객체 추적은 침입자 감시 및 보안, 드론의 목표 객체 고정 추적, 스마트 팩토리, 자동차 자율주행 등과 같은 다양한 시각 응용 어플리케이션의 핵심 기술로 사용된다[1-3].

객체 추적의 핵심 목표는 첫 번째 프레임에서 설정된 정보를 사용하여 연속 시퀀스에서 목표 객체를 찾고 추적하는 것이다. 객체 추적은 크게 3단계로 구성된다. 첫 번째로 첫 번째 프레임의 어노테이션(Annotation)을 사용해 목표 객체의 영역 정보를 초기화한다. 어노테이션에는 객체의 위치를 나타내는 좌상단 X, Y좌표와 객체의 넓이(Width), 높이

(Height) 값을 포함한다. 두 번째로 목표 객체 영역에서 특징을 추출하여 추적에 사용하는 외형모델을 제작한다. 마지막으로 외형 모델을 사용하여 연속 시퀀스에서 목표객체가 위치한 영역을 찾아내고 바운딩 박스로 지역화(Localization)를 진행한다.

객체 추적은 첫 번째 프레임에서 획득되는 제한된 정보만을 사용하기 때문에 성능에 한계점을 가지고 있다. 연속 시퀀스의 객체는 장애물에 객체가 가려지는 폐색(Occlusion) 현상, 객체 크기 변화(Scale Variation), 조명 변화(Illumination Variation), 잡음 또는 목표 객체와 유사한 복잡한 배경(Background Clutters)등의 여러 요소에 의해 초기 설정된 상태와 상이하다. 제한된 정보는 모든 프레임에 나타나는 목표 객체의 형태 변화와 다양한 환경 변화에 강건히

* Corresponding Author: Su-Chang Lim, Address: (57922) Jungang-ro 255, Suncheon, Jeonnam, Korea, TEL: +82-61-750-3628, FAX: +82-61-750-3620, E-mail: suchanglim@sunchon.ac.kr

Receipt date: Jan. 17, 2022, Approval date: Jan. 24, 2022
[†] Dept. of Computer Engineering, Suncheon National University (seaghost@snu.ac.kr)

^{**} Dept. of Computer Engineering, Suncheon National University

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2021R111A1A01044223)

대응하기 위한 충분한 특징을 가지고 있지 않기 때문에 어렵고 이는 객체 추적알고리즘이 표류(Drift)하는 결과를 얻게된다. 이러한 문제를 해결하기 위해서 추적에 사용하는 다양한 특징으로 구성된 객체 외형 모델을 설계하는 것이다. 외형 모델이 충분한 정보를 가지고 있을 때 추적 성능이 향상된다[4].

객체 추적에 사용되는 특징은 크게 수작업(Hand-crafted)을 이용해 추출된 특징과 심층 특징(Deep Feature)으로 분류된다. 수작업 방식은 주로 HOG(Histogram of Oriented Gradients), SIFT(Scale Invariant Feature Transform), LBP(Local Binary Pattern) 알고리즘과 같은 전통적인 특징 추출 알고리즘을 사용한다. 하지만 이러한 특징 추출 알고리즘은 특징을 표현하기 위한 알고리즘의 이상적인 파라미터가 존재하지 않기 때문에 특정 환경에 제한되어 있다는 단점이 존재한다. 심층 특징은 특징 추출에 딥러닝(Deep Learning)을 사용한다는 점에서 수작업 특징과 다르다. ILSVRC(ImageNet Large Scale Visual Recognition Challenge)에서 딥러닝을 사용한 객체 분류 결과가 기존 고전 알고리즘보다 더 좋은 성능을 보여주었다.

딥러닝이 핵심 기술로 주목받았고 객체 인식[5], 분할[6]과 같은 다양한 컴퓨터 비전 분야에 적용하려는 연구가 진행되었다. 특히, 컴퓨터 비전에서 CNN(Convolutional Neural Networks)은 이미지 처리에 우수한 성능을 보여주어 대표적인 딥러닝 아키텍처로 자리 잡았다. 심층 특징은 CNN의 다층 컨볼루션 레이어를 통해 추출된 의미론적(Semantic) 특징을 의미한다. 대상 모양 변화에 더 많은 불변성을 보이기 때문에 수작업 특징에 비해 많은 이점을 지니고 있다.

일반적으로 객체 추적은 생성된 외형모델을 사용하여 추적하는 과정에서 오프라인 방식과 온라인 방식으로 구분된다. 오프라인(Offline) 방법으로 외형 모델을 작성할 경우 충분한 정보를 얻을 수 없어 추적에 실패할 수 있다. 이런 문제를 해결하기 위해서 테스트 단계에서 외형 모델을 업데이트하는 온라인(Online) 방식이 외형 모델 적응성에 있어서 이점을 얻을 수 있었다. 온라인 방식에서 외형모델은 지속적인 업데이트를 통해 갱신할 수 있지만, 계산에 있어서 비효율적이다. 또한, 긴 프레임의 경우 목표 객체의 모양이 프레임 길이에 따라 변화도가 심할 때 추

적 실패 문제를 야기한다.

본 논문에서는 오프라인 방식을 적용한 Siam-FPN(Feature Preserve Network)를 설계했다. 추적 오버랩 정확도와 중점 에러를 감소하기 위해서는 목표 객체의 특징 영역이 강조될 필요성이 존재한다. 주변 영역의 특징으로 인해 배경 영역을 추적하는 결과를 얻기 때문이다. 이를 위하여 객체 특징 정보 강화를 위한 구조로 커스텀 CNN을 설계했다. 커스텀 CNN의 경우 레이어간 부분연결을 사용해 객체 특징 정보를 강화했다. 이 구조를 통해 특징 정보 흐름이 첫 번째 레이어에서 마지막 레이어까지 유지되는 특성을 지닌다.

객체 추적 프로세스를 유사도 비교 문제로 간주하고 목표 객체 이미지와 찾기 영역 이미지의 유사성을 학습하고 해당 영역을 추론하는 것을 목표로 했다. 객체 특징 추출을 위한 커스텀 CNN을 이미지 유사도 비교와 상관관계 문제에 자주 사용되는 삼 네트워크(Siamese Network) 구조로 설계했다[7-9]. 삼 네트워크는 단일 네트워크를 Y 형태의 구조로 만들어 주는 것이다. 이 구조는 동일한 네트워크와 동일한 커널을 사용하여 연산이 이루어지기 때문에, 유사한 두 장의 이미지가 입력될 경우 유사한 특징맵을 출력한다. 이는 비슷한 영역일수록 더 강한 신호를 얻을 수 있다.

논문의 구성은 다음과 같다. 2장에선 외형모델 생성 기반 추적알고리즘과 딥러닝 기반 추적알고리즘의 연구를 기술했다. 3장에서 제안한 알고리즘을 서술하였고, 4장에서 제안한 알고리즘의 성능평가 결과를 상세히 설명했다. 마지막 5장에서는 결론을 맺는다.

2. 관련연구

2.1 외형모델 생성 기반 추적 알고리즘

외형모델은 객체를 대표하는 특징으로 구성되어 있다. 이러한 외형모델은 다양한 방법으로 모델링 되는데 대표적으로 상관 필터(Correlation Filter)를 사용한다. 상관 필터는 FFT(Fast Fourier Transform)를 사용해 생성된 주파수 도메인(Frequency Domain)에서 진행되는 연산은 높은 계산 효율을 보여준다. 따라서 빠른 연산 속도를 보장받아야하는 객체 추적 분야에서 상관 필터를 사용한 다양한 알고리즘

이 연구되어 왔다[10]. 상관 필터 기반 추적 알고리즘은 첫 번째 프레임에서 주어지는 목표 객체의 중심을 기준으로 서브 윈도우를 추출한다. 추출된 서브 윈도우에 FFT 연산을 적용하고 일련의 연산을 거쳐 외형 모델로 변환된다. 변환된 외형 모델은 다음 프레임에서 추출되고 FFT로 변환된 서브 윈도우와 상관 연산을 진행한다. 이후 새로 추출된 객체와 생성된 외형 모델의 유사도에 따라 부여된 가중치를 적용하여 온라인 외형 업데이트를 진행한다. 상관 필터를 추적 분야에 최초로 적용한 알고리즘은 MOSSE이다[11]. 이 알고리즘에서 사용한 개념은 목표 객체의 외형을 적응적으로 온라인 학습하며, PSR(Peak to Sidelobe Ratio)를 이용해 강한 신호를 포착한다. 다른 방법으로 KCF(Kernelized Correlation Filter)기반의 DCF(Dual Correlation Filter) 추적 알고리즘이 연구되었다[12]. 이 알고리즘은 상관 필터를 사용해 신뢰도 맵(Coherence Map)을 제작하고, 객체 위치 및 영역을 베이지안 확률을 사용하여 포착한다. 단일 커널인 KCF에 다중 커널 개념인 MKL(Multi Kernel Learning)을 도입한 MKCFup(Multi Kernel Correlation Filter up)알고리즘이 연구되었다[13]. 객체와 전경을 분리하기 위한 외형 모델링을 적용한 연구도 진행되었다. HOG와 같은 수작업 특징 추출 알고리즘을 사용하여 BACF(Background-Aware CF)를 제안했다[14].

2.2 딥러닝 기반 추적 알고리즘

딥러닝 기반 추적 알고리즘의 핵심은 특징 추출 또는 추적 프로세스에 딥러닝 프레임워크를 적용한다는 점이다. 대량의 데이터를 통해 네트워크를 학습함으로써 다양한 특징을 추출할 수 있다. 처음으로 멀티 레이어 오토 인코더(Multi Layer Auto Encoder)를 사용한 추적 알고리즘이 제작되었다[15]. 최근에는 컴퓨터 비전 분야에서 CNN이 보여준 성능으로 인해, 추적에 CNN을 사용하는 연구들이 진행되었다[16-17]. 대부분의 연구들은 CNN을 특징 추출기의 역할로 사용한다. 동일한 네트워크를 사용할 경우, 비슷한 이미지를 입력하면 유사한 특징이 출력될 것이라는 가설로 CNN을 삼 네트워크 구조로 변환하여 추적에 사용한 연구들이 진행되었다. 삼 네트워크의 구조는 Fig. 1과 같다.

SiamFC와 GOTURN의 경우 사전 학습된 Alex

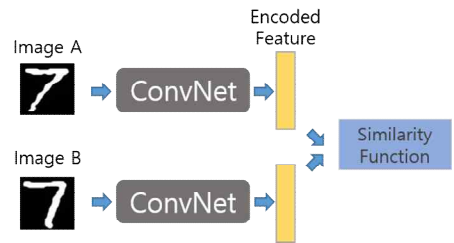


Fig. 1. Basic Architecture of Siamese Network.

Net을 특징 추출기로 사용하였다[18-19]. SiamFC는 AlexNet에서 Fully Connected Layer를 제거하였고, 삼 네트워크 끝단에 교차 상관관계 연산(Cross-Correlation Operation) 레이어를 구현했다. GOTURN은 Fully Connected Layer에 AlexNet에서 추출된 특징을 입력하여 객체의 영역을 추론했다.

3. 제안하는 추적 알고리즘

3.1 제안하는 특징 보존 네트워크 구조

본 논문에서는 특징 보존 및 추출을 위한 목적으로 커스텀 CNN을 제작했다. Fig. 2는 제안하는 특징 보존 네트워크를 보여준다.

활성화 함수를 제외한 네트워크 모든 레이어는 컨볼루션 레이어로만 구성되어있으며 특징맵의 크기 축소 또한 컨볼루션 레이어의 스트라이드 값을 이용하여 축소했다. CNN의 특징은 적층된 다수의 레이어에서 개별적으로 출력되며 다양한 형태의 특징을 표현할 수 있다. 일반적으로 입력 계층과 가까운 레이어에서는 경계(Edge), 모서리, 색상 및 질감과 같은 낮은 수준의 특징을 추출하고, 네트워크 말단으로 갈수록 고수준의 의미론적(Semantic)인 특징을 추출한다. 이러한 CNN의 이점을 살려 첫 번째 네트워크부터 마지막 네트워크까지 정보 흐름을 보존할 목적으로 설계했다. 입력 이미지는 적층된 컨볼루션 레이어를 지나며 특징이 추출되고 더 작은 특징맵으로 변환된다. 이 과정에 특징 연결(Feature Connection)을 적용했다. 특징 연결을 효율적으로 관리하기 위해

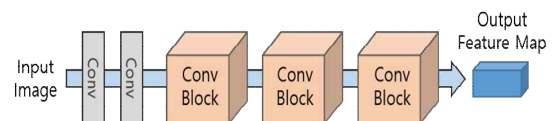


Fig. 2. Architecture of Custom CNN for Feature Preserve.

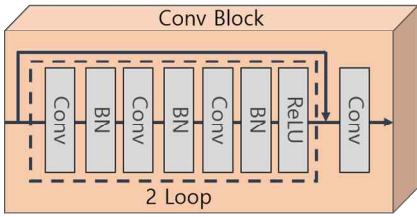


Fig. 3. Detailed Structure of Convolution Block.

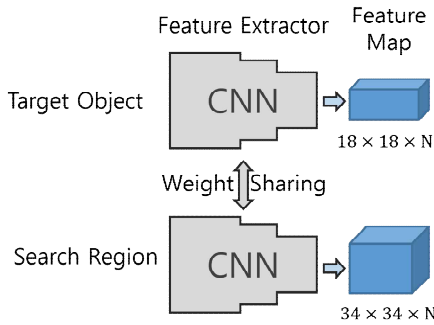


Fig. 4. Siamese Network with Custom CNN.

특징 추출, 특징 연결부, 특징맵 크기 축소를 위한 컨볼루션 레이어로 이루어진 컨볼루션 블록으로 구성했다. 컨볼루션 블록 내부의 특징 추출부는 패딩 (Padding)을 부여해 입력 특징 맵과 동일한 크기를 유지하고 블록 끝단에서 입력 특징맵과 연결했다. Fig. 3은 특징 추출부 및 연결부가 포함된 컨볼루션 블록을 보여준다.

연결된 특징맵은 풀링 레이어를 거치며 크기가 줄어들고, 다음 컨볼루션 블록의 입력으로 사용된다.

이러한 과정을 통해 네트워크 말단까지 정보 흐름을 유지하였다. 제안한 커스텀 네트워크를 추적 문제 해결에 적용하고자 Fig. 4와 같은 Y자형 구조의 삼 네트워크 형태로 구성했다. 삼 네트워크의 최종 출력 특징맵의 크기는 입력 이미지의 크기에 대응하여 결정된다.

삼 네트워크는 목표 객체 이미지와 찾기 영역 이미지를 입력받는다. 두 이미지는 동일한 커스텀 CNN을 지나며 특징맵을 각각 출력한다. 찾기 영역 이미지에 목표 객체가 존재한다면 목표 객체 특징맵과 동일하거나 유사한 값을 출력한다. 출력된 두 개의 특징맵을 사용하여 목표 객체의 위치와 영역을 찾는 작업이 진행된다. 이미지 유사도 비교 작업의 개념과 동일하게 목표 객체 특징맵을 사용하여 찾기 영역 특징맵에서 목표객체를 찾는다. 동일하거나 유사한 특징 위치에서 강한 신호를 얻고 배경일 경우 낮은 신호를 얻는다. 이러한 작업을 위하여 지역 제안 네트워크(Region Proposal Network)를 서브 네트워크로 사용했다. Fig. 5는 삼 네트워크와 지역 제안 네트워크가 결합된 네트워크의 전체 구조를 보여준다.

지역 제안 네트워크는 N개의 앙카 박스(Anchor Box)를 사용하여 객체의 영역을 회귀(Regression)하고 박스 내부에 객체가 존재하는지를 분류(Classification)한다. 앙카 박스는 객체 영역의 중점 좌표와 박스 넓이 높이 4개의 값으로 구성된다. 제안하는 추적 알고리즘에서는 5개의 앙카 박스를 사용했다. 지역 제안 네트워크의 입력으로는 삼 네트워크에서 출력된 목표 객체 특징맵과 찾기 영역 특징맵이 사용

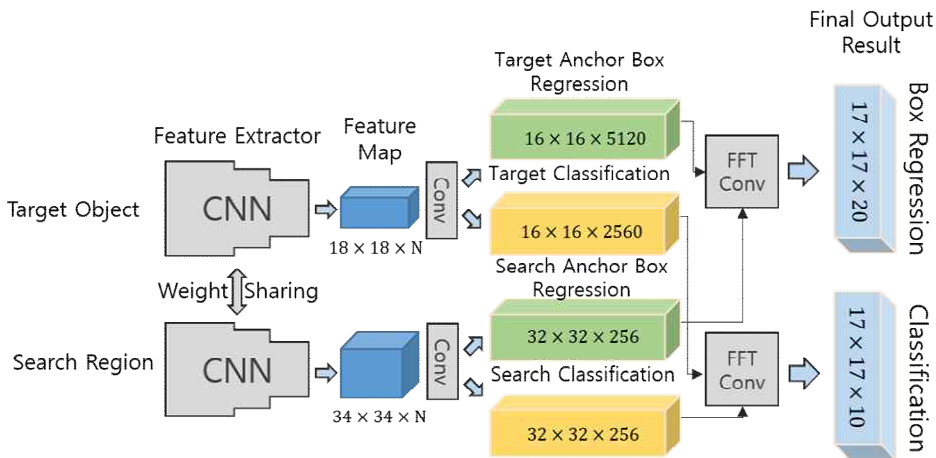


Fig. 5. Proposed Siam-FPN Architecture with Siamese Network and Region Proposal Network.

된다. 이 두 특징맵을 상관 연산하여 박스 좌표와 객체 분류를 진행한다. 이때 커널은 목표 객체 특징맵으로 설정하고 커널이 적용되는 영역은 찾기 영역 특징맵으로 지정한다. 상관 연산과정을 위하여 목표 객체 특징맵과 찾기 영역 특징맵에 FFT(Fast Fourier Transform) 연산을 적용하여 주파수 도메인으로 변환한다. 목표 객체 특징맵은 찾기 영역 특징맵보다 작은 크기로 구성되어 있으므로 주변 영역을 0으로 채워넣어 동일한 크기로 매칭한 후 작업을 진행했다. 주파수 도메인으로 변환된 각각의 특징맵을 아다마르 곱셈을 사용하여 대응되는 위치의 값끼리 연산한다. 마지막으로 Inverse FFT를 적용하여 공간 도메인으로 변환한다. 최종 결과물은 앙카 박스 회귀 값과 객체 분류 2가지로 구분된다. 앙카 박스 회귀 값의 경우 $17 \times 17 \times 20$ 의 형태를 가지는데, 17은 상관관계 연산을 통해 계산된 특징맵의 크기이고 20은 앙카 박스 개수 5개와 각 박스가 지닌 중점 좌표, 넓이, 높이 4개의 값을 포함한다. 분류 값은 $17 \times 17 \times 10$ 형태를 가진다. 17은 앙카 박스와 동일한 특징맵의 크기이고 10은 5개의 앙카 박스의 2가지 분류 결과값을 포함한다.

4. 실험 결과

4.1 실험 환경 및 데이터셋 구성

제안한 추적 알고리즘을 실험하기 위해 사용된 하드웨어 및 소프트웨어 사양은 Table 1과 같다. 네트워크 및 전처리 작업과 같은 모든 과정은 파이썬(Python)으로 구현되었고, 딥러닝 프레임워크는 파이토치(Pytorch)를 사용했다.

유사도 비교를 위한 삼 네트워크는 3,862개의 학습 영상과 555개의 검증 비디오를 포함하는 ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2015 데이터셋을 사용해서 학습했다. 이 데이

터셋은 개별 비디오의 프레임 단위로 추출된 이미지와 대응하는 어노테이션(Annotation)으로 구성되어 있다. 어노테이션에 기록된 GT(Ground Truth) 영역 좌표를 기준으로 객체 영역을 추출한다. 네트워크 학습에는 2장의 이미지를 사용한다. 각 이미지는 목표 객체 이미지와 찾기 영역 이미지로서 각 비디오의 임의 프레임에서 추출한다. 전처리 과정에서 목표 객체가 이미지의 정 중앙에 위치되도록 배치하고 이미지 다양성을 부여하기 위해 객체의 크기는 0.8~1.2 배 크기로 랜덤하게 변화한다.

학습에 사용되는 좌표 레이블은 GT 박스값과 생성된 앙카 박스값을 식 (1)~(4)에 대입하여 계산한다. GT의 좌표는 좌상단을 기준으로 입력되어 있기 때문에 중점 좌표인 앙카박스와 동일하게 매칭하는 작업을 진행한다. 또한, 전처리 과정에서 목표 객체의 크기가 변하면 대응하는 좌표도 변경되므로 새로운 영역에 대응되도록 수정한다. 레이블링 과정을 거치면 좌표값은 매개변수화(Parameterization)가 되어 학습 가능한 형태로 출력된다.

$$Reg_X = \frac{GT_X - Anchor\ Box_X}{Anchor\ Box_W} \quad (1)$$

$$Reg_Y = \frac{GT_Y - Anchor\ Box_Y}{Anchor\ Box_H} \quad (2)$$

$$Reg_W = LOG\left(\frac{GT_W}{Anchor\ Box_W}\right) \quad (3)$$

$$Reg_H = LOG\left(\frac{GT_H}{Anchor\ Box_H}\right) \quad (4)$$

앙카 박스의 객체 분류 작업을 위해 레이블 작업을 진행했다. 레이블은 이미지 겹침 정도를 나타내는 IoU(Intersection Over Union) 척도를 사용해서 계산한다. 사용하는 데이터는 GT와 앙카 박스 영역이다. IoU가 50% 이상이라면 앙카 박스의 분류 레이블을 1로 지정한다. 40% 이하라면 객체가 존재하지 않는 것을 표현하기 위해 0을 지정한다. 40% 초과 50% 미만일 경우는 네트워크 학습에 혼란을 주는 요인으로 판단하고 학습 과정중에 해당 레이블은 무시하도록 -1을 할당했다. 네트워크 학습에 사용되는 손실 함수(Loss Function)은 두 종류를 사용했다. 객체 분류에는 식 (5)와 같이 크로스 엔트로피(Cross Entropy) 함수를 사용했고, 앙카 박스 회귀에는 식 (6)의 SmoothL1Loss 함수를 사용했다. 최종 손실 값은 식

Table 1. Specification of Experiment Environment.

H/W and S/W		Version
H/W	O/S	Windows 10
	CPU	I7-8700K
	RAM	32G
	GPU	Titan XP
S/W	Deep learning Framework	Pytorch 1.8

(7)처럼 두 함수에서 출력되는 손실을 합산하여 얻는다. 네트워크는 이 손실값을 줄여주는 방향으로 학습이 진행된다.

$$\begin{aligned} loss_{cls} &= -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) \\ &= -x[class] + \log\left(\sum_j \exp(x[j])\right) \end{aligned} \quad (5)$$

$$smooth_{L1-reg} = \begin{cases} 0.5(x_n - y_n)^2/\beta, & \text{if } |x_n - y_n| < \beta \\ |x_n - y_n| - 0.5 \times \beta, & \text{otherwise} \end{cases} \quad (6)$$

$$loss_{total} = loss_{cls} + smooth_{L1-reg} \quad (7)$$

4.2 성능 평가

제안한 추적 알고리즘은 100개 비디오로 구성된 OTB-100[20] 데이터셋을 사용해 성능 평가를 진행했다. 데이터셋의 영상이 지닌 다양한 환경 속성은 Table 2와 같다. 각 영상은 최소 1개 이상의 속성을 포함하고 있다. 비교 대상 알고리즘은 BACF(Background-Awared Correlation Filter)와 DCF(Dual Correlation Filter)를 사용했다. 실험은 제안한 알고리즘과 동일한 환경에서 수행했다.

알고리즘의 성능평가는 OPE(One Pass Evaluation) 정량 지표를 사용했다. 이 지표는 Success plot과 Precision plot 두 가지 지표로 구성되어있다. 중첩 정확도(Success Plot)는 추적 알고리즘이 추정된 목표 객체 영역과 GT 영역의 오버랩 비율을 계산하여 성공률을 획득한다. 중점 정밀도(Precision Plot)는 박스 중점 사이의 픽셀 거리를 계산하는 지표로 0픽

셀에서 50픽셀까지 임계값을 두어 계산한다.

Fig. 6은 OTB-100 데이터셋을 사용한 성능 평가 그래프를 보여준다. Fig. 6(a)는 중첩 정확도를 (b)는 중점 정밀도를 나타낸다. 제안하는 알고리즘은 중첩 정확도에서 0.621, 중점 정밀도에서 0.838을 달성하며 가장 높은 수치를 보여주었다.

Fig. 7은 Motion Blur, Fast Motion 속성을 포함한 Blurcar2 영상의 추적 결과 이미지다. 영상 내부의 바운딩 박스 색상은 제안하는 알고리즘은 빨간색, GT는 초록색, BACF는 파란색, DCF는 노란색이다. 이 영상은 카메라의 빠르고, 흔들리는 움직임으로 인해 영상이 전반적으로 흐려지는 현상을 지니고 있다. 0번 프레임의 목표 객체 영역을 기준으로 추적을 진행했다. 파란색의 BACF 알고리즘의 경우 크기가 변화되는 자동차 이미지에 적응하지 못하고 번호판을 중심으로 일부 영역만 오버랩 하는 결과를 보여주었다. DCF는 BACF보다는 나은 성능을 보여주었지만, 제안하는 알고리즘이 GT영역과 거의 유사한 크기로 바운딩하며 추적하는 결과를 보여주었다.

Fig. 8은 Scale Variation 속성을 포함한 Red Team 영상의 추적 결과 이미지다. 영상의 목표 객체는 프레임이 진행될수록 점진적으로 크기가 변하는 양상을 보여준다. 제안하는 알고리즘과 BACF 알고리즘은 객체 크기에 적응적으로 대응하며 객체를 바운딩 하는 결과를 보여준다. 반면 노란색의 DCF의 경우에는 객체 크기 변화에 대응하지 못하고 초기 지정된 객체 크기만큼의 영역만 추적하는 결과를 보여준다.

Fig. 9는 11개 속성에 대한 결과를 내림차순 정렬

Table 2. The Sequence Attribute Table for Algorithm Evaluation.

Attribute	Description
Illumination Variation(IV)	Change the lighting
Scale Variation(SV)	Change in size of target object
Occlusion(OCC)	Object occlusion by obstacles
Deformation(DEF)	Target object deformation
Motion Blur(MB)	Blur on target object
Fast Motion(FM)	Fast motion of target object
In-Plane Rotation(IPR)	Rotation of an target object within an image
Out-of-plane Rotation(OPR)	Rotation of an target object outside an image
Out-of-View(OV)	Some areas of the target object move out of the image
Background Clutters(BC)	Generate target object-like color or texture
Low Resolution(LR)	Target object has low resolution

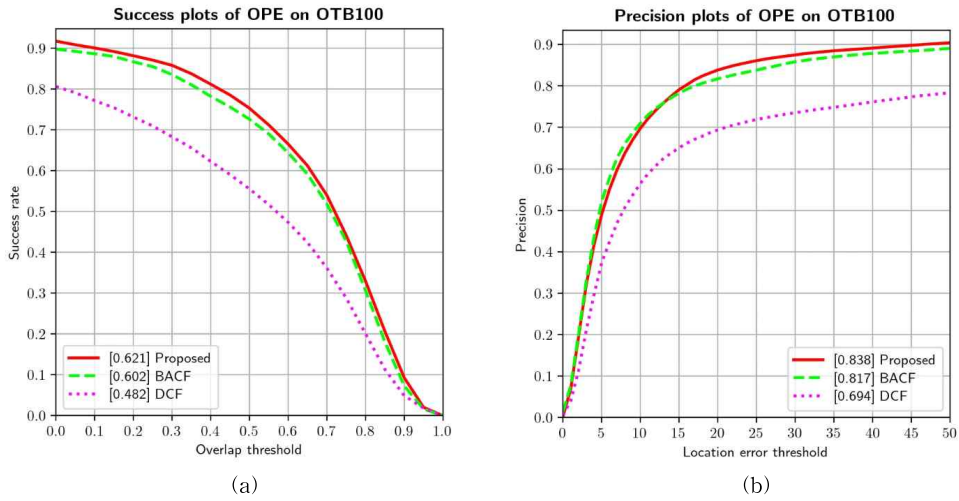


Fig. 6. Result of OTB-100 Datasets, (a) Result of Success Plot and (b) Result of Precision Plot,



Fig. 7. Result of Motion Blur Attribute.

하여 막대 그래프로 표현했다. 중첩 정확도에서는 MB속성이 가장 높은 수치를 보여주었고, LR 속성이 낮은 수치를 보여주었다. 중점 정밀도의 경우 BC 속성에서 가장 높은 수치를 보여주었고, OPR 속성에서 낮은 수치를 보여주었다.

Table 3은 제안하는 알고리즘과 비교 알고리즘의 결과를 11가지 속성에 대해 표로 정리했다. 제안하는 알고리즘은 중첩 정확도의 경우 OCC, OV, FM, MB,

DEF 속성에서 높은 수치를 보여주었고, 중점 정밀도의 경우 OCC, OV, FM, MB, SV, DEF, OPR 속성에서 높은 수치를 보여주었다.

5. 결론

본 논문에서는 객체 추적을 위해 특징 보존, 객체 분류, 객체 영역 추론 3가지 방법론을 제안했고, 이를

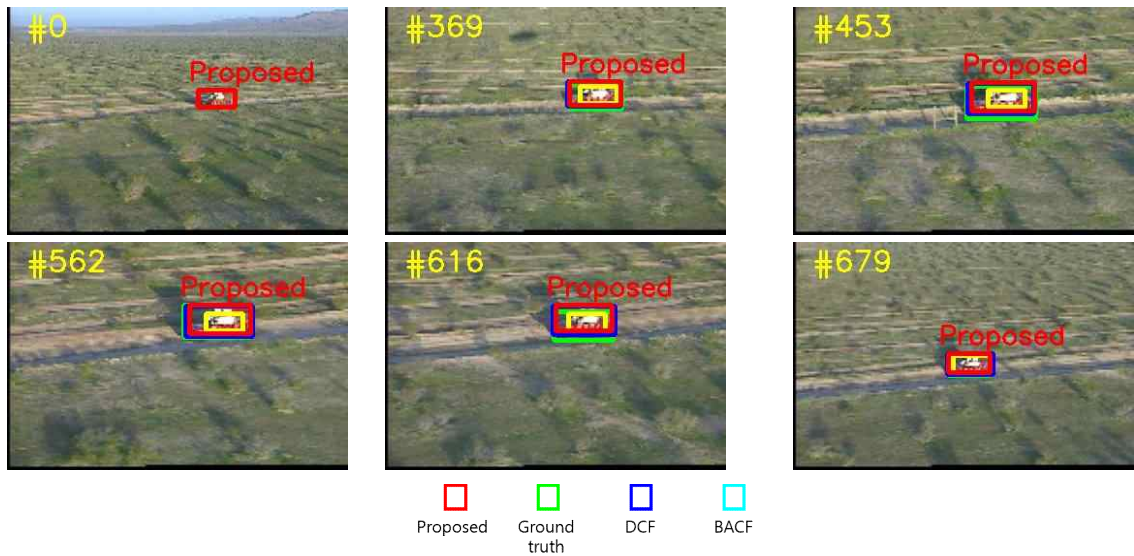


Fig. 8. Result of Fast Motion Attribute.

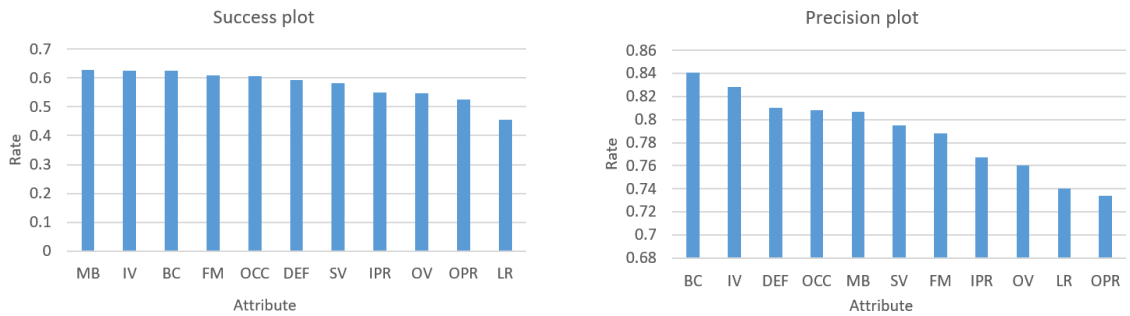


Fig. 9. Bar Graph of Proposed Tracking Algorithm.

위한 삼 네트워크와 지역 제안 네트워크가 결합된 Siam-FPN(Feature Preserve Network)를 제안했다. 제안한 네트워크는 유사도 비교를 위한 커널 학습 및 지역 제안 네트워크가 종단간(End-to-End) 훈련이 가능하도록 설계되었다. 특징 연결부로 이루어진 커스텀 CNN을 사용하여 특징이 보존되어 목표 객체 주변부의 특징이 강화되는 것을 보여주었다. 제안한 추적 알고리즘은 OTB-100 데이터 셋을 사용하여 성능평가를 진행했고, 중첩 정확도에서 0.621, 중점 정밀도에서 0.838을 달성하며 가장 높은 수치를 보여주었다.

향후 연구로는 객체 크기변환에 좋은 성능을 보이기 위해 각 컨볼루션 블록에서 추출되는 크기가 상이한 특징맵을 사용하여 특징 피라미드를 구축한 후, 해당 특징맵의 객체 영역 넓이 정보를 기준으로 객체

를 추적하는 연구를 진행하고자 한다.

REFERENCE

[1] S.H. Kim, S.C. Lim, and D.Y. Kim, "Intelligent Intrusion Detection System Featuring a Virtual Fence, Active Intruder Detection, Classification, Tracking, and Action Recognition," *Annals of Nuclear Energy*, Vol. 112, pp. 845-855, 2018.

[2] A. Koubâa and B. Qureshi, "DroneTrack: Cloud-Based Real-Time Object Tracking Using Unmanned Aerial Vehicles over the Internet," *IEEE Access*, Vol. 6, pp. 13810-13824, 2018.

Table 3. Overall Performance Evaluation Results for Attributes of OTB-100 Benchmark Dataset.

		BACF	DCF	Proposed
Total	Success	0.602	0.482	[0.621]
	Precision	0.817	0.694	[0.838]
IPR	Success	[0.567]	0.471	0.550
	Precision	[0.792]	0.754	0.767
OCC	Success	0.560	0.441	[0.607]
	Precision	0.756	0.619	[0.808]
OV	Success	0.483	0.328	[0.548]
	Precision	0.704	0.443	[0.760]
IV	Success	[0.627]	0.484	0.625
	Precision	[0.830]	0.727	0.828
LR	Success	0.446	0.264	0.456
	Precision	0.729	0.584	0.740
BC	Success	[0.646]	0.513	0.624
	Precision	[0.863]	0.724	0.841
FM	Success	0.572	0.464	[0.610]
	Precision	0.782	0.632	[0.788]
MB	Success	0.584	0.470	[0.628]
	Precision	0.777	0.614	[0.807]
SV	Success	0.538	0.397	0.583
	Precision	0.766	0.623	[0.795]
DEF	Success	0.555	0.450	[0.592]
	Precision	0.777	0.645	[0.810]
OPR	Success	0.518	0.394	0.524
	Precision	0.719	0.573	[0.734]

- [3] S.J. Lee and M.C. Lee, "A Vision Based People Tracking and Following for Mobile Robots Using CAMSHIFT and KLT Feature Tracker," *Journal of Korea Multimedia Society*, Vol. 17, No. 7, pp. 787-796, 2014.
- [4] V.A. Laurence, J.Y. Goh, and J.C. Gerdes, "Path-Tracking for Autonomous Vehicles at the Limit of Friction," *Proceeding of American Control Conference (ACC). IEEE*, pp. 5586-5591, 2017.
- [5] S.W. Park, S.H. Kim, S.C. Lim, and D.Y. Kim, "Performance Comparison of Commercial and Customized CNN for Detection in Nodular Lung Cancer," *Journal of Korea Multimedia Society*, Vol. 23, No. 6, pp. 729-737, 2020.
- [6] S.C. Lim and D.Y. Kim, "Semantic Segmentation using Convolutional Neural Network with Conditional Random Field," *The Journal of the Korea Institute of Electronic Communication Sciences*, Vol. 12, No. 3, pp. 451-456, 2017.
- [7] S. Zagoruyko and N. Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4353-4361, 2015.
- [8] J. Bromley, et al., "Signature Verification Using a Siamese Time Delay Neural Network," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7, No. 4, pp. 669-688, 1993.
- [9] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively with Application to Face Verification," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 539-546, 2005.
- [10] T. Liu, G. Wang, and Q. Yang, "Real-Time Part-Based Visual Tracking via Adaptive Correlation Filters," *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4902-4912, 2015.
- [11] D.S. Bolme, J.R. Beveridge, B.A. Draper, and Y.M. Lui, "Visual Object Tracking Using Adaptive Correlation Filters," *Proceeding of International Conference on Computer Vision and Pattern Recognition*, pp. 2544-2550, 2010.
- [12] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, No. 3, pp. 583-596. 2015.
- [13] M. Tang, B. Yu, F. Zhang, and J. Wang, "High-speed Tracking with Multi-kernel Correlation Filters," *Proceedings of the IEEE*

Conférence on Computer Vision and Pattern Recognition, pp. 4874-4883, 2018.

[14] H.K. Galoogahi, A. Fagg, and S. Lucey, "Learning Background-Aware Correlation Filters for Visual Tracking," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1135-1143, 2017.

[15] N. Wang and D.Y. Yeung, "Learning A Deep Compact Image Representation for Visual Tracking," *Advances in Neural Information Processing Systems*, 2013.

[16] S.C. Lim and D.Y. Kim, "Object Tracking Using Feature Map from Convolutional Neural Network," *Journal of Korea Multimedia Society*, Vol. 20, No. 2, pp. 126-133, 2017.

[17] C. Ma, J.B. Huang, X. Yang, and M.H. Yang, "Hierarchical Convolutional Features for Visual Tracking," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3074-3082, 2015.

[18] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H. Torr, "Fully Convolutional Siamese Networks for Object Tracking," *Proceedings of the European Conference on Computer Vision*, pp. 850-865, 2016.

[19] D. Held, S. Thrun, and S. Savarese. "Learning to Track at 100 fps with Deep Regression Networks," *European Conference on Computer Vision*, pp. 749-765, 2016.

[20] W. Yi, J.W. Lim, and M.H. Yang, "Online Object Tracking: A Benchmark," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411-2418, 2013.



김 종 찬

2000년 순천대학교 컴퓨터과학과 졸업(이학사)
 2002년 순천대학교 대학원 컴퓨터과학과 졸업(이학석사)
 2007년 순천대학교 대학원 컴퓨터과학과 졸업(이학박사)

2013년 서울대학교 자동차 시스템 연구소 선임연구원
 2021년~현재 순천대학교 컴퓨터공학과 조교수
 관심분야 : 영상처리, HCI, 콘텐츠, 컴퓨터그래픽스, 기계 학습, 객체추적, 컴퓨터비전, 데이터 분석 및 예측



임 수 창

2015년 순천대학교 컴퓨터공학과 졸업(공학사)
 2017년 순천대학교 대학원 컴퓨터공학과 졸업(공학석사)
 2022년 순천대학교 대학원 컴퓨터공학과 졸업(공학박사)

2020년~현재 (주)티이에프 기업부설연구소 수석연구원
 관심분야 : 컴퓨터비전, 영상처리, 딥러닝, 객체추적, 시계열분석