

HOG 알고리즘과 CNN을 이용한 객체 검출 시스템에 관한 연구

박 병 준* · 김 현 식**

Research on Objects Tracking System using HOG Algorithm and CNN

Park Byungjoon · Kim Hyunsik

〈Abstract〉

For the purpose of predicting credit card customer churn accurately through data analysis Detecting and tracking objects in continuous video is essential in self-driving cars, security and surveillance systems, sports analytics, medical image processing, and more. Correlation tracking methods such as Normalized Cross Correlation(NCC) and Sum of Absolute Differences(SAD) are used as an effective way to measure the similarity between two images. NCC, a representative correlation tracking method, has been useful in real-time environments because it is relatively simple to compute and effective. However, correlation tracking methods are sensitive to rotation and size changes of objects, making them difficult to apply to real-time changing videos. To overcome these limitations, this paper proposes an object tracking method using the Histogram of Oriented Gradients(HOG) feature to effectively obtain object data and the Convolution Neural Network(CNN) algorithm. By using the two algorithms, the shape and structure of the object can be effectively represented and learned, resulting in more reliable and accurate object tracking. In this paper, the performance of the proposed method is verified through experiments and its superiority is demonstrated.

Key Words : Image Processing, Objects Tracking, Histogram of Oriented Gradients(HOG), Convolution Neural Network(CNN)

I. 서론

최근 보안 및 감시 시스템, 위치 추적, 사고 예방, 인명 구조, 자율주행, 의료, 제조 등 다양한 분야에서 인공

지능 기반의 영상분석을 통한 다양한 활용과 연구가 진행되고 있다. 이때, 가장 기본적으로 이러한 영상분석을 위한 영상 처리 시스템의 효율과 신뢰성을 높이기 위해서는 먼저 배경과 객체를 정확하게 식별하는 것이 필수적이다. 물체를 정확하게 추적하려면 배경과 물체를 구별하는 작업이 필수적이며, 이는 시스템 생산의 성공에 큰 영향을 미치기 때문에 매우 중요한 과정이라 할 수

* 서일대학교 소프트웨어공학과 교수(주저자)

** 서일대학교 소프트웨어공학과 교수(교신저자)

있다. 배경 환경을 효과적으로 추적하기 위해 배경 변화를 모니터링하여 배경과 객체를 효과적이고 정확하게 구별하는 방법은 다음과 같다.

먼저, 객체를 추적하기 위해 배경과 객체를 분리한다. 객체 배경을 분리하는 과정은 아래와 같은 과정을 통해 진행된다.

- ① 객체와 배경이 무엇인지, 어디에 있는지 파악
- ② 객체를 명확히 파악하기 위해 배경을 제거
- ③ 객체를 포함하는 영상에서 객체가 아닌 부분 제거
- ④ 배경을 모두 제거하여 객체만 남기고, 잔여 객체 (노이즈) 제거

배경이 분리된 객체는 다음 과정으로 학습과 추적 과정이 이루어진다.

- ① 객체 등록(학습)
- ② 시간별 프레임 단위 움직임 추적
- ③ 추적된 데이터를 기초로 객체 다음 프레임의 객체를 분석

위 과정에서 하나의 프레임 내에서 물체를 정밀하게 추적하고 감지하려면 배경을 정확하게 식별하는 전처리 단계를 통해 객체 이외의 픽셀을 배경에서 제거해야 한다. 일반적으로 각 픽셀의 배경 값과 객체 값을 구분하기 어려운 경우가 많으며, 그 결과 픽셀이 통합된 배경 이미지로 인식되는 경우가 많다. 때문에 기본적으로 단일 이미지에서 배경과 물체를 구별하는 것은 거의 불가능하다. 따라서 처음 저장된 이미지와 최근에 변경된 이미지 데이터를 비교하여 변경된 데이터를 기반으로 객체 영역을 정확히 찾아내는 과정이 필수적이다.

영상 추적은 움직임이 발생하였을 때 이미지 프레임 내에 등록된 객체의 데이터를 이용한다. 이 과정 전반에 걸쳐 물체의 움직임과 위치 이동에 대한 지속적인 경계가 유지되므로 정확한 객체 추적의 결과를 얻을 수 있다 [1].

일반적으로 객체가 없는 깨끗한 상태의 배경 화면이 있다면 배경 차분 방식을 사용하여 물체의 움직임이 없더라도 배경이 외부적 요인에 의해 변화 하였으면 객체를 찾을 수 있다. 하지만 기존의 이러한 방식은 처음의 배경 화면이 완전한 배경 영상이라는 확신이 있어야만 가능하다. 그리고 같은 배경의 영상이라도 시간의 지남에 객체의 회전, 크기 변화에 따라 변화 할 수 있기에 객체의 위치를 찾는 상황에서 배경과 객체 그리고 또 다른 객체 안에 숨겨져 있던 물체를 잘못 식별하는 문제가 발생할 수 있다. 때문에 이러한 문제를 적절하게 해결하기 위해서는 보다 복잡한 배경 식별 방법을 채택할 필요가 있다. 그리고 동적 객체를 모니터링 하는 과정에서 객체의 영역이 아닌 다른 객체의 움직임이 발생하여 식별된 변화 영역의 픽셀에서 움직임을 감지하면 시스템은 동작 데이터를 기반으로 물체를 영역별로 분류한 후 배경 설정, 물체 전환 추적 및 움직임 추정을 통하여 객체의 변화된 형태와 움직임을 실시간으로 추적하게 되는데 이때, 객체 구분에 상당한 혼란과 실시간적으로 많은 계산량이 요구된다.

본 논문에서는 이러한 문제점을 해결하고 영상인식의 효율을 높이기 위해서 비디오 시퀀스 내의 이전 프레임과 후속 프레임을 비교하여 시간 경과에 따른 배경 변화를 모니터링하여 불필요한 노이즈를 제거, 변화가 발생하는 영역 내에서 객체의 움직이는 영역을 식별한다. 즉 배경에서 객체의 변화를 추적하여 움직임의 상대적 움직임을 측정한다. 이와 함께, 끊임없이 변화하는 동영상 프레임을 지속적으로 모니터링하기 위해 방향성 기울기 히스토그램 HOG 알고리즘을 활용하여 배경과 물체를 분리하고, 이때 검출된 객체 데이터를 CNN 알고리즘을 이용하여 변화된 데이터를 업데이트 하여 추적하는 방식을 이용한다[2]. 이러한 접근 방식을 채택함으로써 배경 식별 기술과 관련된 제약을 효과적으로 우회하여 객체 인식 정확도를 향상시킬 수 있다. 논문의 2장은 관련 연구를 검토하고, 3장에서는 객체와 배경 분리하는 전처리 과정에 대하여 알아보고 4장에서는 시스템 구현에 대해

설명한다. 이어서 5장에서는 실험 결과에 대한 분석을 제공하고, 6장에서는 결론으로 마무리한다.

II. 관련 연구

영상인식과 분석, 검출을 위해서 프레임 간의 객체의 움직임을 찾기 위해 특징을 비교하여 유사점을 식별하는 객체 추적 알고리즘에 대한 많은 연구가 수행되었고 이러한 다양한 연구를 통해 객체 추적의 정밀도를 향상시키기 위한 다양한 접근법이 제안되었다.

HOG(Histogram of Oriented Gradients) 알고리즘의 탄생은 대상 객체와 관련된 경계 정보를 활용하면서 물체를 감지해야 하는 필요성에서 비롯되었다. 모서리의 크기와 방향을 확인하기 위해 64×128 크기로 보정된 1차원 커널을 사용하여 X축과 Y축을 따라 일련의 콘볼루션 연산을 실행한다. 이어서, 결과 출력을 미리 정해진 크기의 셀로 단편화하고, 각 셀의 크기를 집계하여 방향성 히스토그램으로 구성된 디스크립터를 통해 적절한 매칭 대상을 식별한다. HOG 알고리즘의 각 셀이 포함하는 각도 범위는 일반적으로 180° 이지만 본 논문에서는 360° 에 이르는 확장된 범위 내의 벡터로 묘사되며, 8개 셀의 방향과 데이터는 셀당 45° 세그먼트로 더 세분화된다. 또한 히스토그램은 8개의 서로 다른 방향으로 분할하여 도출되며, 방향에 해당하는 크기를 누적하여 관련 객체를 쉽게 추적할 수 있다. 물체 추적 메커니즘이 미리 정해진 간격으로 주기적으로 업데이트되므로 환경 내에서 진화하는 물체의 움직임을 실시간으로 신속하게 식별할 수 있다[3].

CNN(Convolutional Neural Network) 알고리즘은 이미지나 비디오 등의 시각적 데이터를 처리하는데 많이 사용되는데, Y축(수직)을 따라 대칭성을 나타내는 방식으로 두 이미지를 합친 다음 해당하는 각 픽셀을 해당 픽셀에 대해 콘볼루션 연산을 수행한다[4]. 이때, 정보가

손실되면서 특징들이 합쳐져 하나로 뭉뚱그려지는 값이 되는데 이 과정에서 정보가 크게 손실되기 때문에 이러한 현상을 완화하기 위해 제로 패딩(Zero Padding)이라는 기술을 활용하며 맥스 풀링(Max Pooling) 기술 필터를 거치면서 최댓값만 보존하고 나머지 값은 삭제하므로 정보가 크게 손실되어 데이터의 상당 부분이 삭제되며 중요한 특징만 보존되고 대부분의 데이터는 손실된다. 그리고 특히 이 방법을 실시간으로 프레임 상에서 적용할 때 복잡성과 리소스 집약적인 특성으로 인해 많은 계산이 요구 되므로 실시간 적용이 용이하지 않는 특징을 가지고 있다. 하지만 CNN 알고리즘은 컬러와 다양한 환경적 변화에 빠르게 데이터를 갱신할 수 있다는 장점을 가진다. 이러한 CNN은 이미지에서 하나의 객체는 변위 벡터가 존재하며, 영상에 속한 객체의 수에 따라 변위 벡터가 존재한다. 벡터들은 집합이 이루는 히스토그램에서 이전 변화의 값과 가장 높은 밀도를 가지는 벡터를 객체의 이동 벡터로 결정한다.

III. 전처리 과정

본 논문에서는 보다 효율적인 영상인식을 위해 연속적인 프레임 신호에서 선행 프레임과 후속 프레임의 데이터를 병치하여 고정 프레임 간에 전환되는 물체를 식별하고 배경의 변경 사항을 파악하여 객체와 배경을 구분할 수 있도록 한다. 이 과정을 통해 물체와 배경 간의 경계를 더 명확하게 구분할 수 있다. 이때, 움직이는 요소가 지정된 임계값을 초과하는지 여부를 확인하며, <식 1>에서 볼 수 있듯이 다음으로 이어지는 배경 이미지 프레임과 대조하는 작업을 수행한다[5].

$$x_{mp} = \left\{ \begin{array}{l} (I_1(x) - I_{1-1}(x)) > T_{1(x)} \\ \parallel \{ I_n(x) - I_{n-1}(x) \} > T_{n(x)} \end{array} \right\} \quad \langle \text{식 1} \rangle$$

'I'는 현재 이미지를 나타내고 'n'은 시간의 경과를 나

타낸다. 현재 이미지의 픽셀의 값이 이전 프레임의 이미지 값과 비교하여 사전 정해진 임계값을 초과하면 움직이는 픽셀로 분류된다. 이때 객체로 식별된 부분이 <그림 1>에서 보논바와 같이 배경 섹션과 구별된다. 성능 평가에 사용된 첫 번째 영상(<그림 1> 영상 A)은 760 x 570 해상도의 컬러영상(24Frame/초)이고 두 번째 영상(<그림 1> 영상 B)은 1024 * 768 그레이 영상(24Frame/초)이다.



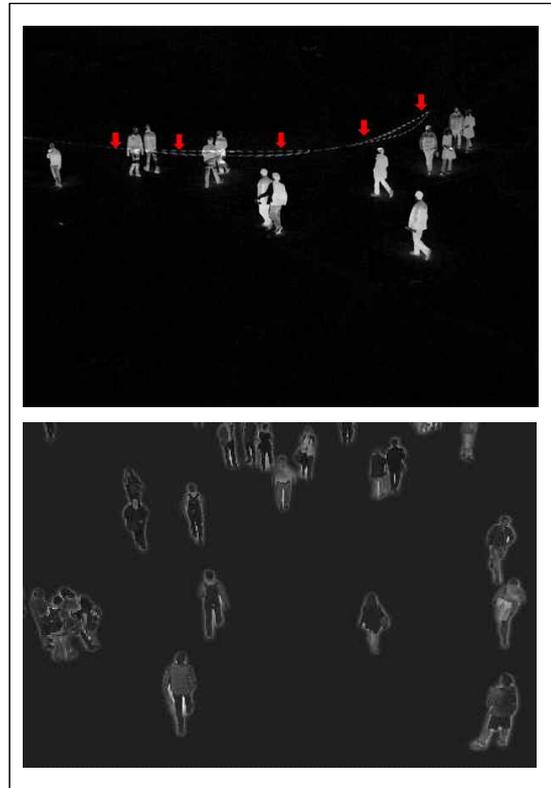
<그림 1> 객체 추적 영상 이미지
(위 : 영상 A, 아래 : 영상 B)

이러한 객체로 식별된 움직이는 객체의 영역이 식별되면 움직이는 분할 프로세스가 시작된다. 이 분할 프로세스에는 기본적으로 움직이는 물체가 포함된 관심 영역과 배경 이미지의 일부인 관심 영역을 구분하는 작업을 말한다. 이때 <식 2>를 적용하면 배경 요소와 움직이는 물체 사이에 명확한 경계가 설정되므로 이미지 데이터

내의 동적 요소에 대한 추가 분석 및 처리가 보다 용이해진다.

<식 2>에서 N은 배경 이미지에서 물체에 해당하는 픽셀의 값을 뺀 값을 나타내고, 움직이는 픽셀의 값은 mp로, 움직이지 않는 픽셀의 값은 nmp이다. 기본적으로 배경 이미지의 픽셀의 값에서 현재 프레임 이미지의 픽셀의 값을 뺀 값이 특정 임계값을 초과하면 이전 프레임 이후 변경된 픽셀의 값을 나타내며, 이를 움직이는 물체로 분류할 수 있다.

$$N = \{x : (I_n(x) - B_n(x)) > T_n(X), x \in R\} x_{mp} \quad \text{<식 2>} \\ !N = x_{nmp}$$



<그림 2> 움직임 검출 이미지
(위 : 영상 A, 아래 : 영상 B)

하지만 움직임을 보이는 모든 픽셀이 객체는 아니다. 일부 움직임은 바람이나 조명 조건의 변화 등의 요인의 영향을 받기 때문이다. 따라서 지속적으로 변화하는 배경 이미지에서 움직이는 객체와 <그림 2>의 위쪽 영상 A에 표시된 화살표와 같이 객체는 아니지만 움직이고 있는 영역을 지속적으로 업데이트해야 한다[6]. 단순 비교에만 의존하면 객체와는 관련이 없지만 조명이나 바람과 같은 외부 요인에 의해 <그림 2>의 위쪽 영상 A 이미지의 화살표에 표시된 것처럼 흔들리는 픽셀의 값이 객체로 식별되어 객체의 증가로 연산이 늘어나게 되며 이는 객체 추적의 정확성을 떨어뜨리며 연산 시간 또한 늘리는 변수가 된다. 따라서 이미지 프레임 내에서 시간이 지남에 따라 변화하는 픽셀을 정확하게 평가할 수 있는 기술이 필요하다.



<그림 3> 잔여 노이즈 객체가 있는 이미지
(위 : 영상 A, 아래 : 영상 B)

이러한 노이즈 객체의 문제는 입력 프레임과 후속 프레임 간의 가중치 이동을 고려하여 평균값을 누적하고, 후속 프레임에서 입력 이미지 프레임 간의 불일치를 재계산하여 움직이는 객체 정보를 업데이트함으로써 <그림 3>에서 볼 수 있듯이 객체를 제외한 배경 화면에서의 객체가 아닌 노이즈를 제거함으로써 이를 해결할 수 있다. 이러한 방법을 사용하면 정밀한 객체 추적이 용이해지고 배경 이미지가 더 선명해진 객체 데이터를 학습에 이용할 수 있다.

$$O(x, y) = I_{1Frame}(x, y) - I_{2Frame}(x, y) \quad \text{<식 3>} \\ W = (1 - \alpha) \cdot I_{MoveFrame} \\ O_{Move}(x, y) = I_{1MoveFrame} + I_{2MoveFrame} + \dots \rightarrow W$$

이러한 과정을 통해서 일정 크기 이상의 노이즈는 제거되었지만, 아직 남아있는 잔여 노이즈를 제거하기 위해 <식 3>에서 보는 바와 같이 입력 프레임과 다음 프레임 간의 움직임의 가중치를 집계하여 이미지를 평균값으로 계산하고, 각 프레임 간의 차이를 재계산하여 이동 객체 정보를 누적한다.

이러한 반복 프로세스를 통해 이미지 내 노이즈를 더 정확하게 감지할 수 있다. 10프레임의 시간 간격으로 이전 프레임과 후속 프레임을 비교하여 움직임을 표시하는 픽셀의 변화된 영역을 명확하게 표시한다. O 는 이전 프레임에서 현재 프레임의 값을 뺀 변화된 픽셀이며 W 는 누적 가중치이다. 상수로 작용하는 가중치는 업데이트된 픽셀의 값의 비율을 결정하며, α 값이 1에 가까울수록 데이터 업데이트 비율은 높아진다. 하지만 움직임이 없을 경우가 지속되면 그만큼 누적 가중치 W 가 감소한다.

영상 프레임 간에는 프레임 내에서 발생하는 사소한 움직임이나 다른 여러 가지 요인으로 인하여 상당한 양의 자잘한 노이즈가 남아있다. 이러한 노이즈를 제거하기 위해 침식 및 팽창 연산을 활용하여 관심 영역 외부에 있는 픽셀을 제거한다. 이러한 계산을 통해 영상 프레임의 전체 품질을 왜곡시킬 수 있는 불필요한 요소를 제

거하여 최종 영상의 선명도와 정밀도를 향상시킨다.

집합 A와 B 사이의 침식 과정은 수학적으로 $A \ominus B$ 로 표현하고 <식 4>와 같이 정의한다. 반면에 팽창은 <식 5>와 같이 정의하며 이는 물체 내부의 돌출부를 줄이는 동시에 바깥쪽 돌출부를 확대하여 객체의 전체 크기를 확장하고 배경의 영향을 줄이는 것을 목표로 한다. 이러한 전략적 접근 방식은 객체의 모양을 최적화하고 외부 교란을 최소화하여 시각적 데이터를 향상시킨다[7].

$$A \ominus B = w : B_w \subseteq A \quad \text{<식 4>}$$

$$A \oplus B = (a,b) + (u,v) : (a,b) \in A, (u,v) \in B \quad \text{<식 5>}$$

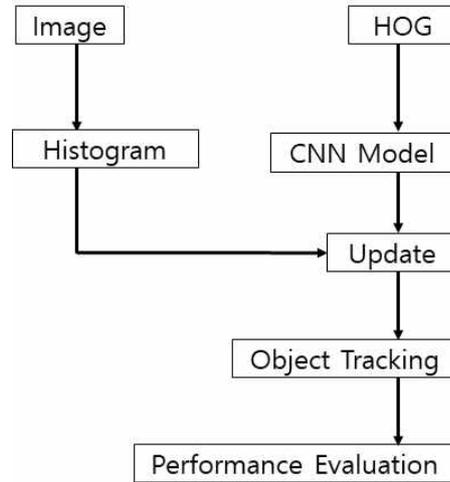


<그림 4> 노이즈 제거 후 객체 이미지

이러한 과정을 통해서 <그림 4>에서 보는 것과 같이 이미지 영역 전체에 존재하는 관심이 없는 영역을 완전히 제거 한 것을 볼 수 있다.

IV. 시스템 구현

본 논문에서 사용한 학습의 기반이 되는 데이터는 Caltech Vision Lab의 보행자 데이터를 이용하였다[8]. 처리 속도를 높이기 위해, 컬러(RGB)로 이루어진 영상의 화소를 그레이 영상으로 변환 하였다. 이는 컬러 영상으로 프레임들을 비교할 때 발생하는 계산량을 줄이기 위함이다. 그레이 영상을 이용하면 실시간 영상 처리 속도를 크게 향상시킬 수 있으며, 객체의 추적 시간을 효과적으로 줄이며 측정이 가능하다.



<그림 5> 시스템 구조

본 논문에서는 이전 프레임이 기준 영상이 되며, 다음에 들어올 프레임이 새로운 영상이 된다. 입력 프레임과 후속 프레임 간에 전환되는 히스토그램 가중치의 누적을 통해 도출된 평균값을 계산하여 이미지의 히스토그램 데이터를 생성한다. 이어서, 움직이는 객체의 정보는 다음 프레임 내에서 움직이는 차이만큼 다시 계산하여 객체 정보를 업데이트한다[9].

이러한 업데이트 과정은 이미지를 Block단위로 나누고 Block 내부의 위치한 각 셀들에 대해 기울기 히스토

그램을 만든다. 이 블록은 특징 벡터가 되며, 모든 블록을 이어 붙이면 최종 벡터값을 구할 수 있다. HOG 객체 추적 알고리즘은 벡터값을 이용하여 객체인지 아닌지 파악 하지만, 본 논문에서는 이 데이터를 CNN 알고리즘의 특징 추출기로 사용하는 것이다. 즉, HOG에서 파생된 히스토그램 데이터의 변위 벡터 데이터는 CNN 알고리즘에서 다음 움직이는 객체를 찾을 때 이전 변화의 값과 가장 높은 밀도를 가지는 벡터를 객체의 이동 벡터로 결정 하여 학습 시간을 크게 줄이고 실시간 데이터 처리를 용이하게 한다.

이 과정에서 CNN 방식은 초기 동영상의 프레임 데이터를 입력으로 그대로 사용할 경우, 실시간 처리 자체가 불가능할 정도로 계산량이 많아지는 단점을 가지고 있다. 따라서 이러한 문제를 해결하기 위해 객체로부터 HOG 기능의 히스토그램 데이터를 수집한 후 이 데이터를 CNN 모델의 입력 데이터로 추출하는 프로세스를 거친다. CNN 학습 과정에서 입력 데이터를 가지고 성능 분석을 진행하며 학습된 Model의 성능 향상 실험을 진행한다. 이를 통해서 본 논문에서 제시한 HOG + CNN Model 혼합 방식이 성능 향상에 미치는 영향을 알아본다[10].

V. 실험 평가

측정 실험은 Intel i5 10400 CPU, Windows 10, Microsoft Visual Studio 2022를 이용하여 평가하였다. 이 실험은 기존의 영상 처리 방법과 제안하고 있는 알고리즘을 테스트하는 것을 목적으로 하였다. 특히, 관심 영역을 설정하여 전체 처리 시간을 실시간 처리에 맞게 단축하는 방법을 연구하였다.

<표 1>은 기존 알고리즘과 제안하고 있는 알고리즘의 계산 속도와 객체 탐지율을 나타내고 있다. 기존 물체 감지 프로세스는 최소한의 계산 리소스가 필요한 간단

계산 방법을 이용하지만 객체 추적에 대한 민감도 때문에 객체 추적의 경우에는 실패하는 경우가 많다. 이는 <표 1>에서 확인할 수 있는데, HOG 방법이 기존의 NCC(Normalized Cross Correlation) 방법에 비해 시간당 계산량(Calculation time)을 약 15% 증가시켰으나 객체 탐지율(Object Tracking Rate)의 경우 약 18% 높아졌음을 확인할 수 있다. 그럼에도 불구하고 HOG 방법만으로는 응용 프로그램 등에 적용하여 효율적 객체 탐지를 수행하기에는 탐지율이 부족한 것을 확인할 수 있다. 이와 달리, 본 논문에서 제안하고 있는 HOG와 CNN을 결합한 탐지 방식의 경우, 영상의 객체를 추적하여 평균값을 구한 결과 계산량이 증가하기는 하지만 실시간 탐지를 수행하는데 문제가 되지 않는 수준이며 오탐지가 최소화되며 객체 탐지율이 상당히 높아졌음을 <표 1>에서와 같이 실험을 통해 확인할 수 있었다. 앞선 시스템 구현 과정에서 설명한 것처럼 실험에서 CNN 단일 방식은 초기 동영상의 프레임 데이터를 입력으로 그대로 사용할 경우 실시간 처리 자체가 불가능할 정도로 계산량이 많아지는 단점을 가지기 때문에 비교에서 제외하였다.

<표 1> 알고리즘 별 계산 속도

Algorithm \ Calculate	Calculation Time	Object Tracking Rate
NCC	200 ms	65 %
HOG	230 ms	83 %
HOG + CNN	320 ms	97 %

실험을 통해서 제안한 알고리즘이 두 가지 알고리즘을 결합함으로써 기존의 영상 차감 알고리즘 객체 추적 방법에 비하여 연산량이 늘어나 연산 속도는 늦어졌지만 더 정밀하게 객체 추출 및 추적이 가능하다는 것을 확인하였다. 객체를 추적하는 과정에서 처리 속도는, 30프레임 이상으로 기존방식에 비해 연산과정이 많은 만큼 6~7프레임 정도의 속도 저하가 있었다.

〈표 2〉 개별 데이터별 객체 인식 정확도

	영상 A	영상 B
실험 이미지 1	100%	100%
실험 이미지 2	95%	97%
실험 이미지 3	93%	98%



〈그림 6〉 영상 A에 대한 객체 추적 결과

개별 실험 데이터를 확인해 봄으로써 검출 과정의 문제점과 특징을 확인해 보았다. 〈표 2〉는 실험 영상 별 개별 프레임에 따른 객체 인식 정확도를 나타내고 있으며 〈그림 6〉과 〈그림 7〉은 객체를 추적하여 인식하는 과정을 보여주고 있다[11]. 실험 결과를 살펴보면 영상 A의 경우, 가운데 기둥을 통과하면서 검출율이 약간 감소하지만 영상 B는 장애물이 없는 상태에서 초기 검출율이 거의 일정하게 유지되고 있음을 보여준다. 또한 두 영상 데이터 모두에서 보행자 검출 성능이 단일 이미지에 의존하는 기존 객체 검출 방식에 비해 상대적으로 높은 것을 확인할 수 있었다.



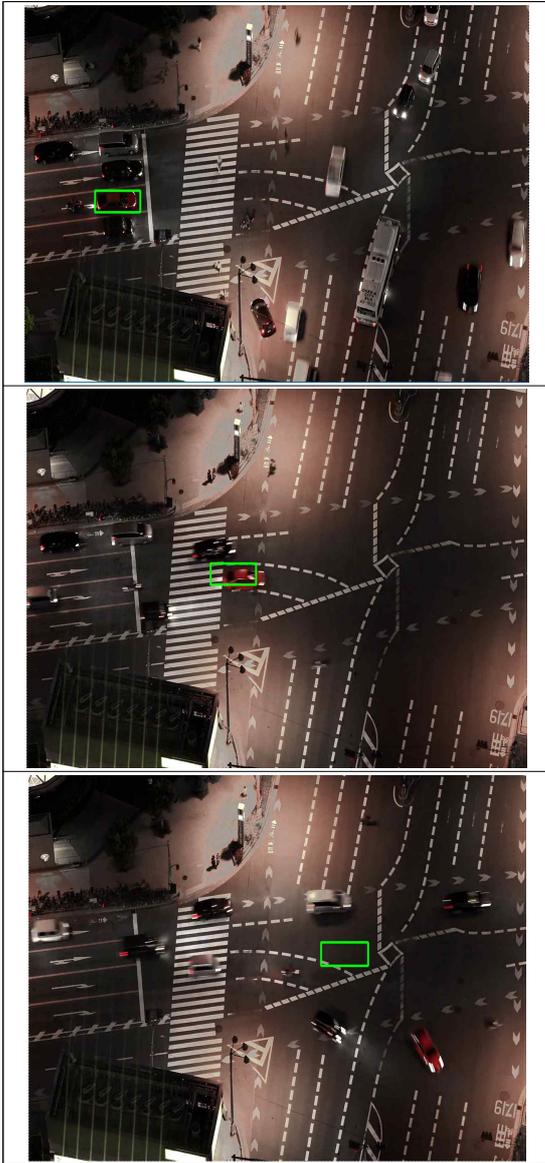
〈그림 7〉 영상 B에 대한 객체 추적 결과

영상 A의 경우, 가운데 기둥을 통과할 때 검출율이 다소 떨어지는 경향이 있었다. 하지만 영상 B의 경우에는 별다른 장애물이 없기에 초기 감지율이 거의 일정하게 유지되고 있는 것을 볼 수 있다. 이를 통해서 단일 이미지를 사용하는 기존의 객체 검출 방식에 비해 보행자 검출 성능이 우수함을 확인하였다.

성능을 높이면서도 계산량 증가를 최소화 하고자 하였는데 이를 위해서, 이미지의 처리는 이미지 내 모든 영역에 균일하게 적용되는 것이 아니라 특정 관심 영역만 선택적으로 처리하여 효율성과 효과를 향상시켰기 때문으로 분석된다. 이러한 방식으로 주의와 분석이 필요한 특정 영역에만 초점을 맞추면 알고리즘의 계산 시간을 크게 줄일 수 있다[12].

실험 데이터에서 침식 확장 과정을 거친 후 이미지를 자세히 살펴보면 모든 노이즈 사레가 성공적으로 제거되는 않았음을 알 수 있으며, 추가 관찰 결과 해당 물체와 관련된 픽셀 데이터가 손실된 것으로 나타났다. 본 연구에서는 이 잔존하는 노이즈 문제를 해결하기 위해 관

심 영역에만 특수 가중치 값을 포함한 재계산 절차를 거치는 표적 접근법을 채택하였다. 제안된 방법론에서 나타나는 탐지 정확도의 저하는 기동 근처에서 또는 지나가는 다른 사람들과의 상호 작용 중에 추가 계산이 갑자기 급증하는 경우에 기인할 수 있는 것으로 분석된다.



〈그림 8〉 객체 추적 실패 영상

〈그림 8〉은 Caltech Vision Lab의 Cars 데이터를 이용하여 학습을 진행 하여 테스트를 수행한 결과를 보여 준다. 초기에는 객체의 특징을 추적에 성공했지만 자동차의 속도가 빨라지면서 추적에 실패 하였다. 자동차가 가속하며 속도가 빨라지면서 자동차의 잔상 이미지가 생성되며 단 시간에 히스토그램이 급격히 변화를 하여 빠른 데이터 변경을 따라가지 못한 것이다. 즉 데이터 업데이트 속도 이상으로 움직이는 자동차의 변화를 업데이트가 따라가지 못하고 있는 것을 볼 수 있다. 즉 300ms 내외의 업데이트 보다 빠르게 히스토그램 일정 이상으로 변화하여 다음 객체를 찾지 못하고 있는 것이다. 실시간 동작에서 이러한 연산 문제를 해결하려면 데이터 업데이트 시간을 줄이던지 하드웨어 상에서 멀티프로세싱 계산 또는 최적화 기능을 활용하여 타이밍 속도를 높이는 것이 매우 중요하다. 하지만 이러한 기술 발전에도 불구하고 업데이트 시간과 타이밍에 대한 지속적인 딜레마는 여전히 중요한 문제이므로 향후 지속적인 연구가 필요하다.

VI. 결론

객체의 추적을 시작하기 전에 배경 이미지를 분리한 후 객체의 각 프레임을 기반으로 업데이트 및 분석하여 객체와 배경을 구별하는 기술은 객체 추적 시스템 초기 단계이며 필요한 객체를 보다 정밀하게 추적함으로써 모니터링 시스템의 정밀도를 높이는 데 매우 효과적이라는 것이 입증되었다. 본 연구에서는 단순한 추론 방법을 통해 배경을 업데이트하지 않는 픽셀 기반 업데이트 문제를 해결하기 위해 학습 목적의 데이터를 생성하였다. 그 결과 프레임별 객체 비교를 통해 객체 감지의 정확도가 크게 향상되었다. 각 정보가 배경 상태를 기반으로 하는 프레임 간 데이터와 통합되므로 프로세스의 계산 효율이 향상되었고 특히 배경 환경이 변화하는 관심 영역 내의 객체와 관심 없는 객체(노이즈)를 보다 정확하게 구분할

수 있었다. 빛과 객체의 변화와 같은 외부의 변화에 민감한 기존의 이미지에서의 객체를 추적하는 방법과 관련된 문제는 계층적 객체 감지, 업데이트 및 프레임 간 조사를 포함하는 객체 추적을 위한 전처리 단계를 통해 보다 효과적으로 완화할 수 있었다.

본 논문에서는 단순 차감법 알고리즘을 이용한 픽셀 기반 객체 추적의 단점인 객체가 변화 하였을 때 제대로 추적하지 못하는 문제를 해결하기 위해 움직임 가중치를 히스토그램으로 업데이트하며 다음 찾을 객체를 추적하였다. 그 결과 프레임별 객체 비교를 통해 물체 감지의 정확도를 크게 향상시켰으며, 프레임 간 데이터와 통합되므로 프로세스의 계산 효율성 또한 향상되었다. 특히 환경이 변화하는 상태에서 관심 영역 내의 객체와 관심 없는 노이즈를 보다 정확하게 구분할 수 있었다. 또한 CNN Model에 프레임 이미지를 그대로 데이터로 입력하면 이미지의 규모에 따라 학습 계산의 복잡성이 기하급수적으로 급증한다는 것을 알 수 있는데 HOG의 히스토그램 데이터를 학습 프로세스에 통합함으로써 CNN 모델에서 원 프레임의 이미지를 학습하는 것에 비해 계산 작업량이 실시간 처리를 할 수 있을 정도로 눈에 띄게 감소하였다. 이러한 감소는 학습 단계에서 계산에 필요한 양을 최소화 할 수 있기 때문이며 결과적으로 HOG 함수를 학습 데이터 생성에 활용하면 학습 부담을 줄이면서 실시간 데이터 분석 영역에 이용할 수준으로 크게 줄일 수 있다는 것을 확인할 수 있었다.

이러한 연구 결과는 앞으로 더 효율적이고 정밀한 객체 추적 시스템의 발전을 위한 기초 데이터로 활용될 수 있을 것으로 기대되며 소규모 시스템에서 보다 효율적인 시스템 개발을 가능하게 할 것이다. 본 논문은 정지영상이 아닌 동영상에서 전 처리를 진행하며 연속적으로 추적을 수행하기 때문에 기존 방법들에 비해서 시간 단위 연산이 증가하는 문제점이 발생하였다. 때문에 향후 연구에서는 알고리즘의 최적화 및 간소화를 통해 연산량을 줄이고 다양한 데이터와 보다 정밀한 알고리즘을 이용하는 방법에 대한 지속적인 연구를 수행할 예정이다.

참고문헌

- [1] Feng Zhao, Qingming Huang and Wen Gao, "Image Matching by Normalized Cross-Correlation," IEEE International Conference on Communications, 2006, pp.729-732.
- [2] Miranto, S. R. Sulistiyanti and F. X. Arinto Setyawan, "Adaptive Background Subtraction for Monitoring System," International Conference Information Communication Technology, 2019, pp.153-156.
- [3] N.Dalal and B.Triggs, "Histograms of oriented gradients for human detection," Conference on Computer Vision and Pattern Recognition, Vol.2, June 2005, pp.886-893.
- [4] Y. Taigman, M. Yang, M. Ranzato, & L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," IEEE conference on computer vision and pattern recognition, 2014, pp.1701-1708.
- [5] J. G. Choe, J. H. Lee, D. G. Kang, and S. H. Seo, "Comparison and Verification of Deep Learning Models for Automatic Recognition of Pills," Journal of Digital Contents Society, Vol.21, no.3, Mar. 2020, pp.445-452.
- [6] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," ACM Multimedia Systems, Vol.1, 1993, pp.10-28.
- [7] H.B.Lu Y.J.Zhang, Y.R.Yao, "Robust Gradual Scene Change Detection," International Conference on Image Processing, Vol.3, 1999, pp.304-308.
- [8] Caltech Vision Lab, https://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/
- [9] D.S. Bolme, J.R. Beveridge, B.A. Draper, and Y.M. Lui, "Visual Object Tracking Using Adaptive Correlation

Filters," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544-2550.

- [10] 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, Vol.37, 2015, pp.583-596.
- [11] H. Nam and B. Han, "Learning Multi-Domain Convolutional Neural Networks for Visual Tracking," IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp.4293-4302.
- [12] Bo Zhang, W eifeng T ian and Zhihua Jin. "Joint tracking algorithm using particle filter and mean shift with target model updating," Chinese Optics Letters, Vol.4, 2006, pp.569-572.



김 현 식
(Kim Hyunsik)

2011년 3월~현재 서울대학교
소프트웨어공학과 조교수

2010년 8월 경기대학교 일반대학원
전자계산학과(이학박사)

2004년 2월 경기대학교 일반대학원
전자계산학과(이학석사)

2001년 2월 경기대학교 전자계산학부
전자계산전공(이학사)

관심분야 : 자동계획, 시멘틱웹, 지능형
시스템, 데이터마ining, 기계학습,
모바일 응용 S/W

E-mail : hskim@seoul.ac.kr

논문접수일 : 2024년 8월 22일
수정접수일 : 2024년 9월 03일
게재확정일 : 2024년 9월 14일

■ 저자소개 ■



박 병 준
(Park Byungjoon)

2013년 3월~현재 서울대학교
소프트웨어공학과 조교수

2010년 2월 국민대학교 전산학과(이학박사)

2002년 8월 고려대학교 의료정보기학과
(공학석사)

2000년 2월 고려대학교 전자정보공학과
(공학사)

관심분야 : 영상처리, 패턴인식, 객체추적,
IoT, 인공지능

E-mail : 20130029@seoul.ac.kr