

# 차분 영상을 이용한 객체 추적 방법

조진환 · 장시웅\*

동의대학교

## Object Tracking Method using Difference Images

Jin-Hwan Cho · Si-Woong Jang\*

Dong-Eui University

E-mail : 20206061@office.deu.ac.kr, swjang@deu.ac.kr

### 요 약

최근 딥러닝 환경의 확산으로 인하여 데이터셋 생성의 중요성이 높아지고 있다. 본 논문에서는 효율적인 데이터셋 생성을 위하여 객체의 회전 영상을 촬영하고 해당 영상에서의 객체 추적을 수행하는 방법을 설계하고 구현하고자 한다.

본 논문에서 구현하는 방법은 객체의 여러 각도를 촬영하기 위하여 객체를 회전시켜 영상 데이터를 획득하고 해당 영상에서의 배경 제거 및 차분 영상 처리 기법을 통하여 객체를 검출하고 추적하여 라벨링을 수행하여 사용자가 현재 프레임에서의 객체 추적 결과를 모니터링할 수 있도록 화면으로 보여 주며, 추후 데이터셋으로 활용하기 위하여 이미지 내에서의 객체 위치 데이터를 반환하도록 구현하였다.

### ABSTRACT

Recently, the spread of deep learning environments has increased the importance of dataset generation. In this paper, we aim to design and implement a method for capturing rotating images of objects and performing object tracking on them for efficient dataset generation.

The method implemented in this paper is to obtain image data by rotating objects to capture multiple angles of objects, detect and track objects through background removal and difference image processing techniques, showing them on screen to monitor object tracking results in the current frame. It was then implemented to return object location data within the image for use as a dataset.

### 키워드

딥러닝, 데이터셋, 객체 추적, 배경 제거, 영상 처리

### 1. 서 론

최근 딥러닝 관련 연구가 활발하게 이루어짐에 따라 데이터셋 구축을 위한 데이터 수집 방법에 대한 관심도 높아지고 있다.

기존 데이터 수집 방법은 웹 상에 존재하는 데이터들을 어플리케이션이나 툴과 같은 자신들의 고유한 방법을 사용하여 수집하는 방법이 주를 이루었다[1, 2, 3].

하지만 이러한 방법은 사용자가 원하는 이미지

뿐만 아니라 내용상 맞지 않는 이미지들도 많이 포함되어 있다는 문제점을 가지고 있으며[4], 웹 상에 존재하는 데이터 중에서 딥러닝 모델의 학습을 위하여 대규모 대표 훈련 데이터를 수집하기 위한 환경의 숫자가 너무 적거나 비현실적이고 사람 이미지 데이터의 경우 초상권과 같이 액세스하기에 위험하다는 단점이 있다[5].

또한, 위와 같은 문제점을 모두 극복하고 데이터를 수집한다고 하더라도 딥러닝 모델의 원하는 성능을 달성하기 위하여 딥러닝 모델을 훈련시키는 데 필요한 시간과 자원과 훈련 데이터셋을 생성하

\* corresponding author

기 위한 수동 라벨링 작업에 걸리는 시간이 많이 걸리고 오류가 발생하기 쉽다는 문제점이 있다[5].

본 연구에서는 위와 같은 기존 데이터 수집 방법에 대한 문제점을 해결하기 위하여 객체의 회전 영상을 획득하고 해당 영상에서의 차분 영상 처리 기법을 사용하여 객체를 검출하고 추적한다.

이렇게 검출한 객체를 기반으로 자동 라벨링을 수행하면 효율적으로 다면 데이터셋을 구축할 수 있다.

## II. 객체 추적 방법 구현

본 연구에서 객체 추적 방법은 객체의 회전 영상을 촬영하고 해당 영상에서 여러 영상 처리 기법을 활용하여 이전 프레임과 현재 프레임 간의 비교를 통해 변화된 부분을 객체로 인식하여 추적하는 방법을 사용한다.

먼저 촬영한 객체 영상 데이터를 읽어와 영상 내 각각의 프레임마다 이전 프레임과 비교를 통해 움직이지 않는 부분 즉, 배경을 제거하여 그림 1과 같이 차분 영상을 구한다. 그 후, 차분 영상에서 움직인 부분을 객체로 인식하고 인식한 객체의 범위를 추적하여 라벨링을 수행한다. 이때 촬영한 영상에서 바람과 같은 여러 변수로 인하여 배경이 움직여 원하는 객체가 아닌 다른 부분을 인식하는 것을 방지하기 위하여 객체가 위치한 범위를 ROI(Region of Interest)로 지정하여 둔다.

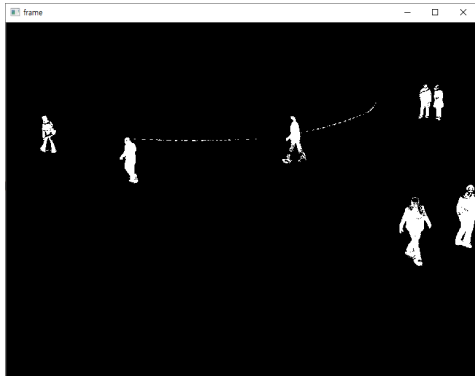


그림 1. 배경 제거를 통한 차분 영상 획득

차분 영상을 구할 때, 발생하는 잡음을 제거하기 위하여 여러 선택 인자를 추가하는데, 대표적으로 각 픽셀마다 가우시안 분산 임계 값의 개수를 지정해주는 Threshold 인자와 객체 그림자를 탐지할지 여부를 결정하는 detectShadow 인자가 있다 [6]. 본 연구에서는 정확도 높은 객체 추적을 구현하기 위하여 Threshold 값을 50, detectShadow 인자를 false로 선택하였다.

위와 같은 방법으로만 차분 영상을 구했을 때, 사람의 팔, 다리와 같이 얇은 부분이나 객체의 색상에 따라 제대로 인식이 되지 않는 경우가 발생

하였다. 이를 해결하기 위하여 그림 2와 같이 영상 처리 기법 중 팽창 연산을 통하여 차분 영상에서 이미지의 경계 부분을 팽창시켜 인식률을 높였다.

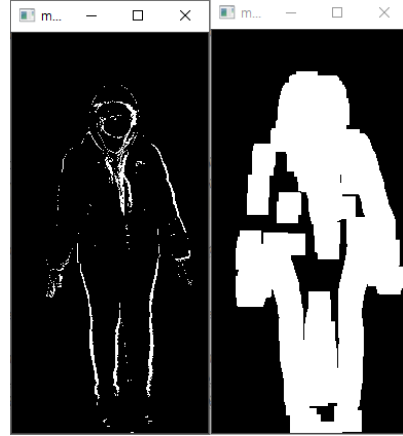


그림 2. 팽창 연산 전(좌), 연산 후(우)

그림 2에서 팽창 연산의 수행 횟수를 10번으로 지정하여 인식된 경계 부분을 크게 확장하였으며, 연산 수행 횟수를 줄인다면 경계 부분의 크기를 줄일 수 있다. 또한, 팽창으로 인하여 발생하는 잡음을 모폴로지 연산(열기)를 수행하여 제거하였다.

차분 영상에서 원하는 객체를 하나의 객체로 판단하지 않고 분리된 여러 개의 객체로 판단하는 경우가 발생하였다. 이를 해결하기 위한 방법으로 인접한 객체들의 좌·상단과 우·하단의 좌표를 묶어 하나의 객체로 판단하는 알고리즘을 추가로 구현하였다.

## III. 객체 추적 테스트

본 논문에서 구현한 시스템을 테스트하기 위해 데이터셋 라벨링 툴인 Yolo\_Mark를 활용하였다.

본 시스템에서 영상 처리를 기반으로 객체를 자동으로 추적하여 Bounding Box를 지정했을 때의 결과와 Yolo\_Mark를 사용하여 사람이 직접 객체의 Bounding Box를 지정했을 때의 결과는 그림 3, 4와 같다.

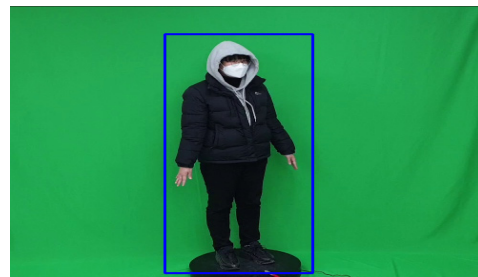


그림 3. 본 시스템의 객체 자동 추적 결과



그림 4. Yolo\_Mark 활용 객체 수동 추적 결과

위와 같은 결과에서 각 Bounding Box의 좌표 값(x, y, width, height)을 아래 공식 1과 같이 유클리디안 거리 공식을 활용하여 Bounding Box 간 거리를 계산했을 때의 결과를 구하였다.

이후 공식 2를 통해 Bounding Box 간 거리 결과를 기반으로 유사도를 구하였으며, 결과는 아래 표 1과 같다.

$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

공식 1. 유클리디안 거리 공식

$$Similarity(\%) = (1 - \sqrt{\sum_{i=1}^n (x_i - y_i)^2}) \times 100$$

공식 2. Bounding Box 유사도 계산 공식

표 1. 객체 자동 추적과 수동 추적 결과 비교

		자동추적 결괏값	수동추적 결괏값	유사도 (%)
사례1	x 좌표	0.516146	0.521484	98.17
	y 좌표	0.590741	0.595833	
	너비	0.148958	0.142969	
	높이	0.651852	0.636111	
사례2	x 좌표	0.515625	0.525000	93.14
	y 좌표	0.590741	0.577778	
	너비	0.147917	0.090625	
	높이	0.651852	0.686111	
사례3	x 좌표	0.526042	0.526953	93.49
	y 좌표	0.571296	0.597222	
	너비	0.114583	0.152344	
	높이	0.690741	0.644444	

테스트 결과, 본 시스템의 객체 자동 추적 결과와 Yolo\_Mark를 사용하여 수동으로 객체를 추적한 결과의 유사도는 최소 93% 이상으로 유사하게 나타나는 것을 확인할 수 있다. 추후 가중치를 객

체에 적용하여 추적한다면 더욱 정확한 결과를 얻을 수 있을 것이다.

## VI. 결 론

본 논문에서는 효율적인 데이터셋 생성 방법을 위하여 영상 처리를 기반으로 한 객체 회전 영상에서의 객체 자동 추적 방법에 대하여 제안하였다.

본 시스템은 객체 회전 영상을 촬영하고 해당 영상에서의 영상 처리 기법을 기반으로 하여 객체 자동 추적 과정을 수행한다.

이러한 방법은 실제 환경에서 데이터셋을 생성하기 위한 과정(훈련 데이터 수집, 유효 데이터 선별, 객체 인식 및 추적)에서 발생할 수 있는 시간 및 비용을 크게 줄여 효율적으로 정확도 높은 데이터셋을 생성할 수 있다.

추후 연구를 통해 가중치를 객체에 적용시킴으로써 객체 자동 추적의 결과의 정확도를 더욱 높이고자 한다.

## Acknowledgement

“본 논문은 2020년도 BB21+사업에 의하여 지원되었음. 본 연구는 과학기술정보통신부 및 정보통신 기획평가원의 지역지능화혁신인재양성 (Grand ICT연구센터) 사업의 연구결과로 수행되었음 (IITP-2021-2020-0-01791).”

## References

- [1] K. M. Choi, Y. M. Kim, J. P. Shin, S. M. Sung, B. K. Lee, “Data set design and implementation for Assistive walking device AI service construction,” in *Proceedings of the Korean Society of Computer Information Conference*, 29(1), pp. 227-229(3 pages), 2021. 1.
- [2] J. S. Lee, B. K. Ko, E. S. Kang, H. J. Choi, J. O. Kim, B. K. Lee, “AI Learning Cookie Image Data Set Construction,” in *Proceedings of the Korean Society of Computer Information Conference*, 28(2), pp. 347-349(3 pages), 2020. 7.
- [3] H. C. Lee, S. Y. Shin, “Development of Data Collection System using Google Environment.” in *Journal of the Korea Institute of Information and Communication Engineering*, 23(2), pp. 704-705(2 pages), 2019. 10.
- [4] E. S. Park, Y. J. Yang, J. H. Jeon, E. S. Ryu, “Image Web Crawling Program for Artificial Intelligence

- Datasets.” in *Proceedings of the Korean Society of Broad Engineers*, pp. 55-55(1 pages), 2018. 11.
- [5] J, H. Choi, Kevin M. Irick, Justin Hardin, Weichao Qiu, Alan Yuille, Jack Sampson, Vijaykrishnan Narayanan, “Stochastic Functional Verification of DNN Design through Progressive Virtual Dataset Generation” in *2018 IEEE International Symposium on Circuits and Systems(ISCAS)*, Italy: IT, pp. 1-5(5 pages), 2018.
- [6] OpenCV(Open Source Computer Vision). Motion Analysis - Video Analysis [Internet]. Available : [https://docs.opencv.org/master/de/de1/group\\_\\_video\\_\\_motion.html](https://docs.opencv.org/master/de/de1/group__video__motion.html)