

# 임베디드 시스템에 Exemplar Transformer를 활용한 시각적 객체 트래킹 구현에 관한 연구

김도완<sup>1</sup>, 임채연<sup>1</sup>, 이채원<sup>1</sup>, 문혜경<sup>2</sup>

<sup>1</sup>동서대학교 컴퓨터 소프트웨어과 학부생

<sup>2</sup>동서대학교 컴퓨터 소프트웨어과 교수

asd8217924@naver.com, mushr55ml5v2@naver.com, 021122chaewon@naver.com, hkmoon@du.ac.kr

## A Study on the implementation of Visual Object Tracking Using Exemplar Transformer in the Embedded System

Do-Wan Kim<sup>1</sup>, Chae-Yeon Lim<sup>1</sup>, Chae-Won Lee<sup>1</sup>, Hae-Kyung Moon<sup>2</sup>

<sup>1, 2</sup>Dept. of Computer Software, Dong-Seoul University

### 요 약

본 논문은 Exemplar Transformer를 활용하여 단일 프로세서 환경에서 동작하는 시각적 객체 추적(Visual Object Tracking) 모델인 ETTrack(Exemplar Transformer Track)을 리눅스(우분투 20.04) 운영체제를 사용하는 임베디드 시스템 라떼판다 알파(LattePanda Alpha)에 탑재하였다. 데스크톱 프로세서인 Intel i7-8700와 라떼판다 알파의 Intel m3-8100y에서의 객체 추적 성능과 속도를 AUC(Area Under the ROC Curve)와 FPS를 통해서 비교평가하였다. 평가 결과 기존 트랜스포머 기반 추적 모델(TransT)과 유사한 성능, 3 FPS 빠른 추적 속도를 나타내는 것을 구현 증명하였다.

### 1. 서론

트랜스포머(Transformer) 모델은 인공지능 분야에서 많은 성과를 이루었지만, 모델 크기가 커지는 문제가 발생하였다. 이 문제를 해결하기 위해 효율적인 트랜스포머에 대한 많은 연구가 진행되고 있으며[1], 본 논문에서는 시각적 객체 추적 분야에서 활용되는 Exemplar Transformer를 기반으로 한 ETTrack 모델을 임베디드 시스템 라떼판다 알파에 탑재하여 온 디바이스 A.I(On-Device A.I)를 구현하고자 한다.

### 2. 시각적 객체 추적(Visual Object Tracking)

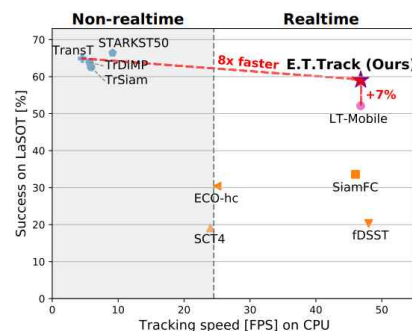


(그림 1) Visual Object Tracking의 동작화면.

객체의 움직임을 예측하여 추적하는 컴퓨터 비전의 한 분야로 하나의 객체를 추적하는 SOT(Single Object Tracking)와 여러 객체들을 추적하는 MOT(Multiple Object Tracking)로 나뉜다. SOT는 현재의 프레임에서 이전 프레임에 지정된 객체의 위치를 파악하는 방식이며, MOT는 학습된 객체들에 한 해 이전 프레임에서 인식된 여

러 객체의 위치를 현재 프레임에서 파악하는 방식이다. (그림 1)은 SOT와 MOT의 동작 화면을 표시한다[2].

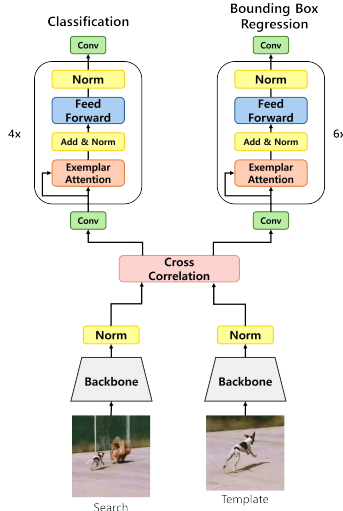
### 3. ETTrack(Exemplar Transformer Track)



(그림 2) 객체 추적 모델 성능 비교.

ETTrack은 LT-Mobile(Light Track Mobile)[3] 구조를 기반으로 하여 단일 CPU 환경(i7-8700)에서 47 FPS(Frame Per Second)의 추적 성능을 가진다. ETTrack은 LT-Mobile의 Classification과 Regression 부분의 합성곱(Convolution) 레이어를 트랜스포머로 교체하여 성능을 7% 향상시켰다. 단일 글로벌 쿼리를 사용하는 Exemplar Transformer를 통해 추적속도를 유지하면서도 성능을 개선하였다. (그림 2)은 여러 객체 추적 모델과 ETTrack의 추적 성능을 LaSOT 데이터 셋으로 실험 비교한 그래프이다[4]. ETTrack은 추적 객체를 포함하는 이전 프레임과 현재 프레임을 입력받아 사전

학습된 backbone을 통해 이미지 특징을 추출하고, 이미지 간의 유사도를 계산하여 Correlation Map을 구한다. 그리고 Classification의 Exemplar Transformer를 통해 배경과 객체를 구분하고 Regression의 Exemplar Transformer를 통하여 객체의 범위와 위치를 예측한다. (그림 3)은 ETTrack의 동작 구조도이다[4].



(그림 3) ETTrack의 동작구조도.

4. 구현 및 성능비교

<표 1>은 ETTrack을 구현한 임베디드 시스템(라떼판다 알파)의 규격, <표 2>는 이와 성능 비교 대상인 데스크톱 PC(Personal Computer)의 규격을 보여준다.

<표 1> 라떼판다 알파의 하드웨어, 소프트웨어 규격

구분	항목	내용
H/W	CPU	Intel m3-8100y
	GPU	Intel HD Graphics 615
	Ram	8GB
S/W	Library (version)	torch (1.8.1)
		torchvision (0.9.1)
		opencv-python (4.9.80)

<표 2> 데스크톱 PC의 하드웨어, 소프트웨어 규격

구분	항목	내용
H/W	CPU	Intel i7-8700
	GPU	Nvidia GTX TITAN X
	Ram	32GB
S/W	Library (version)	torch (1.8.1)
		torchvision(0.9.1)
		opencv-python (4.9.80)

성능 평가 실험은 라떼판다 알파 시스템에 ETTrack을 구현 탑재하여 UAV-123, OTB-100 데이터 셋에서의 추적 속도와 성능을 평가하고 데스크톱 PC 환경에서 비실시간 추적 모델인 TransT와 성능과 추적속도를 비교평가 하였다. <표 3>은 데스크톱 PC의 TransT와 임베디드 시스템 (라떼판다 알파)에서 ETTrack의 정확도와 추적속도를 모델 성능 지표인 AUC(Area Under the ROC Curve)와 추적속도 (FPS)로 평가하였다. (그림 4)는 임베디드 시스템 (라떼판다

알파)에서 구현한 ETTrack의 UAV-123 데이터와 OTB-100 데이터의 추적 결과 이미지이다.

<표 3> 데스크톱, 라떼판다 알파의 성능비교

실험 환경		데스크톱 PC (TransT)	라떼판다 알파 (ETTrack)
UAV-123	AUC	69.4	62.3
	FPS	5	8
OTB-100	AUC	69.1	67.1
	FPS	5	8



(a). UAV-123 추적 결과 (b). OTB-100 추적 결과 (그림 4) 라떼판다 알파 시스템에서의 추적결과화면.

본 연구를 통해 임베디드 시스템 (라떼판다 알파)에서 ETTrack을 탑재한 성능이 데스크톱 PC에서 기존 비실시간으로 동작하는 트랜스포머 기반의 시각적 객체 추적 모델 (TransT)보다 3 FPS 빠른 추적속도를 얻었지만 성능 면에서는 평균적으로 4.6 만큼 떨어진 AUC 스코어를 나타내었다. 그러나 ETTrack은 임베디드 시스템 (라떼판다 알파)과 같은 환경에서 실시간 동작한다는 장점이 있다.

5. 결론

임베디드 시스템에서 딥러닝 기반 시각적 객체 추적 시스템 구현은 여러 가지 제한조건으로 인해 어려움이 있었다. 본 논문에서 Exemplar Transformer를 사용한 ETTrack 모델을 활용하여 데스크톱 PC 환경과 비슷한 추적 성능과 실시간 추적 속도를 갖는 딥러닝 기반의 시각적 객체 추적 시스템을 임베디드 시스템 (라떼판다 알파) 환경에서 구현하여 그 성능을 증명하였다. 본 논문의 결과는 드론, 이동형 로봇과 같은 다양한 실시간 추적 시스템을 필요로 하는 분야에 활용할 수 있을 것으로 기대된다. 최근 CUDA를 활용할 수 있는 Nvidia의 젯슨나노(Jetson Nano)와 같은 고성능의 임베디드 시스템을 활용하면 더 향상된 성능을 구현할 수 있을 것으로 기대된다.

참고문헌

[1] Yi Tay et al., "Efficient Transformers: A Survey", arXiv:2009.06732, 2020  
 [2] Zahra Soleimanitaleb et al, "Single Object Tracking: A Survey of Methods, Datasets, and Evaluation Metrics", arXiv:2201.13066, 2022  
 [3] Bin Yan et al, "LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search", arXiv:2104.14545, 2021  
 [4] Philippe Blatter et al., "Efficient Visual Tracking with Exemplar Transformers", arXiv:2122.09686, 2021