

광학 흐름과 스케일 리샘플링을 통한 실시간 얼굴 탐지 기법 *

김상정¹, 이동건², 서영석³
^{1,2,3}영남대학교 컴퓨터공학과

sangjung245@gmail.com, dklee77@ynu.ac.kr, ysseo@yu.ac.kr

A Method for Real-Time Face Detection through Optical Flow and Scale Resampling

Sang-Jeong Kim¹, Dong-Gun Lee², Yeong-Seok Seo³
^{1,2,3}Dept. of Computer Science, Yeungnam University

요 약

기존의 딥러닝 모델을 활용한 얼굴 탐지 시스템은 영상을 처리할 때 이미지의 양이 과도하여 추론 속도가 영상 재생 속도보다 느려지게 되고, 이로 인해 지연 현상이 발생한다. 본 논문은 이미지 크기 조정 및 광학 흐름을 활용하여 얼굴 탐지에 필요한 추론량을 줄이는 기법을 제안한다. 제안된 기법은 세 단계의 처리 과정으로 구성된다. 첫 번째 단계에서는 프레임의 크기를 줄여 프레임 처리 속도를 효과적으로 향상시킨다. 두 번째 단계에서는 비탐지 구간이 아닌 프레임만을 배치 처리하여 딥러닝 모델로 추론하여 처리 시간을 단축시킨다. 세 번째 단계에서는 광학 흐름 알고리즘을 이용하여 비탐지 구간에서 얼굴 추적을 함으로써 정확도는 유지하면서 탐지 시간을 단축한다. 본 논문에서 제안하는 이미지 크기 조정 및 광학 흐름 알고리즘 기반 얼굴 탐지 시스템은 처리 시간을 수십 배 이상 단축하여 영상에서의 얼굴 탐지에 있어서 우수한 성능을 입증하였다.

1. 서론

최근 학계에서는 얼굴 탐지의 정확도를 높이기 위해 딥러닝 모델을 활용한 연구가 활발히 진행되고 있다. 그러나 딥러닝 모델을 기반으로 한 얼굴 탐지 시스템의 느린 추론 속도는 신속한 얼굴 탐지에 있어 큰 장애물로 작용하고 있다. 이러한 문제점을 해결하기 위하여, 이미지 크기 조정 및 광학 흐름 알고리즘을 활용하여 얼굴 탐지 시스템의 추론량을 줄이는 기법을 제안한다. 제안된 기법은 추론량을 획기적으로 감소시켜 처리 시간을 줄임으로써 얼굴 탐색을 보다 효율적으로 수행할 수 있다.

2. 제안 기법

본 연구에서는 프레임 분배, 얼굴 탐지, 객체 추적과 출력 프로세스로 분리하여 각각을 처리하는 멀티프로세싱 기법을 제안한다. 각 도메인은 정해진 프로세스를 수행한 후 다음 프로세스로 결과를 전달하는 구조를 가지고 있다.

먼저, 얼굴 탐지 시스템의 구조에 사용되는 중요한 변수로는 배치 크기, 비탐지 구간의 크기와 프레임 스케일이 있다. 배치 크기는 한 번에 처리되는 프레임 수를 의미하며, 비탐지 구간의 크기는 한 번 탐지한

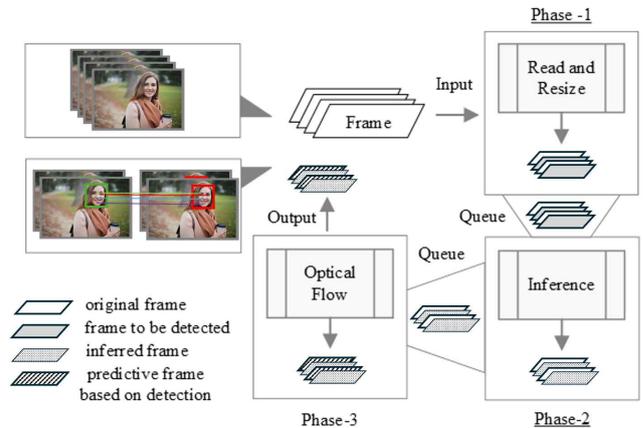


Fig. 1. Overall Approach

이후 탐지하지 않는 프레임의 수이다. 스케일은 영상에서 원본 프레임의 크기를 조정하고자 하는 값이다.

그림 1은 얼굴 탐지에 필요한 프레임 데이터 처리 과정의 세 단계를 보여준다. 각 단계에는 프로세서가 할당되어 작동하며, 프로세서 간 데이터 전송을 위해 큐가 사용된다. 큐를 이용한 데이터 전송을 통해, 프로세서를 복제하여 동일한 큐에 연결함으로써 같은 단계의 프로세스를 실행할 수 있다. 이를 통해 프로세서의 개수를 유동적으로 조절하며 처리 속도를 향상시킬 수 있다. 프로세서들은 프레임 데이터를 손실 압축하고 다른 데이터들과 함께 직렬화

* 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2023R1A2C1008134).

하여 전송한다. 손실 압축을 통해 데이터의 길이를 크게 줄임으로써 전송 시간을 절약할 수 있다.

(Phase-1) 단계 1에서는 주어진 영상을 읽고 각 프레임을 스케일 값에 맞게 조정한다. 스케일 조정을 통해 절대적인 데이터 크기를 줄일 수 있다. 조정된 프레임들은 배치 크기로 그룹화되어 단계 2로 전송된다.

(Phase-2) 단계 2에서는 첫 번째 프레임부터 시작하여 비탐지 구간의 크기만큼의 간격을 두어 배치 크기의 마지막 프레임까지의 프레임 데이터들만 배치 처리하여 GPU로 전송된다. 이 데이터는 딥러닝 모델의 추론에 사용된다. 배치처리 된 데이터가 한번에 전송되기에 GPU 연산에 최적화된다. 추론 결과로 얻어진 얼굴 위치는 각각의 프레임과 함께 단계 3으로 전송된다.

(Phase-3) 단계 3에서는 단계 2에서 전송받은 데이터 중 비탐지 구간이 아닌 프레임에서 탐지된 얼굴 위치 데이터를 기반으로 다음 프레임의 얼굴 위치를 추적한다. 비탐지 구간 동안 이전 프레임의 얼굴 위치를 기준으로 분석한 광학 흐름 데이터를 사용하여 다음 프레임의 얼굴 위치를 추적한다. 비탐지 구간이 아닌 프레임을 만날 경우, 현재 추적 중인 얼굴이 해당 프레임의 얼굴 데이터로 존재한다면 추적을 재시작하고, 존재하지 않는다면 지속적으로 추적한다. 추적된 데이터는 간단한 벡터 연산으로 얼굴 위치를 파악할 수 있다. 추적된 얼굴 데이터는 프레임 데이터와 함께 최종적으로 출력된다.

3. 실험 및 검증

본 논문에서는 얼굴 탐지 시스템 구조를 구현하기 위해 딥러닝 모델인 MTCNN (Multi-task Cascaded Convolutional Networks) [1]과 광학 흐름 알고리즘인 Lucas-Kanade [2]를 사용했다. 배치 크기는 60, 비탐지 구간의 크기를 10으로 설정했고, 스케일은 원본의 환경을 유지하기 위해 1로 설정했다.

성능 비교는 전체 처리 시간, 단일 프레임 처리 시간 및 GPU 사용률을 기준으로 한다. 제안된 시스템과 비교 대상 시스템에는 35개의 각기 다른 영상을 동일하게 입력하여, 비교 기준이 되는 데이터들의 평균을 구한다. 비교 대상으로는 MTCNN을 사용한 단일 프로세스 시스템을 사용했다.

실험 결과, 표 1은 T-검정 결과를 보여준다. 귀무가설 (H_0)은 제안된 기법과 비교 대상 사이에 성능 비교 기준이 통계적으로 유의미한 차이가 없음을 가정하며 대립가설 (H_1)은 제안된 기법과 비교 대

Table 1. T-test for Each Performance Metric ($p < 0.05$)

	P-value	T-statistic
Processing Time/Frame	1.64E-33	-41.01
Total Processing Time	1.83E-17	-15.47
GPU utilization	5.62E-11	7.61

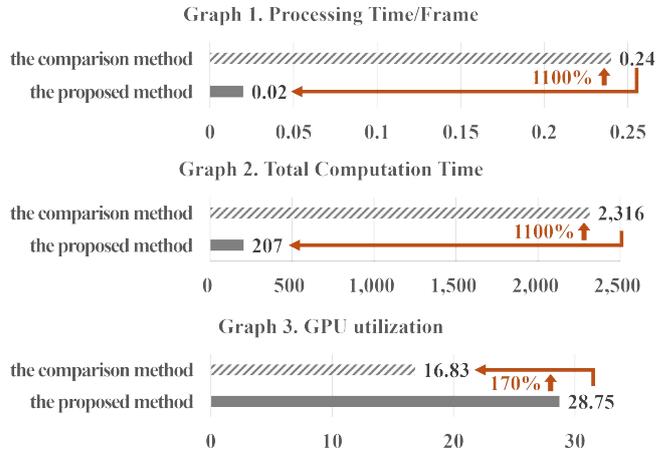


Fig. 2. Performance Evaluation

상 사이에 성능 비교 기준이 통계적으로 유의미한 차이가 있음을 가정한다. 표 1에서 모든 p가 0.05 미만이므로 (H_0)을 기각하고 (H_1)을 채택한다. 이 결과는 제안된 기법이 비교 대상보다 통계적으로 유의미한 차이가 있음을 의미한다. 그림 2는 성능 비교 기준의 평균값들을 보여준다. 제안된 기법을 통하여 비교 대상에 비해 메모리 사용률이 높은 대신 프레임 당 처리 시간과 전체 처리 시간은 약 11배 그리고 GPU 사용률은 약 1.7배 향상됐다.

4. 결론

본 논문에서는 프레임의 크기를 원하는 스케일로 조정하고, 광학 흐름 알고리즘을 통해 딥러닝 모델의 추론 횟수를 줄이는 기법을 제안하였다. 제안한 얼굴 탐지 시스템 구조는 3가지 단계를 통해 얼굴 탐지를 부분적으로 실행함으로써 연산량을 획기적으로 줄일 수 있었음을 보여줄 수 있었다.

참고문헌

[1] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks." IEEE Signal Processing Letters, 23(10), 1499-1503.
 [2] Lucas, B.D. & Kanade, T. (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision." Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), 674-679.