영상 콘텐츠에서의 장면 전환 인식을 사용한 다중 얼굴 추적

박한규 O1 이현민 1 유한봉 2
1 한양대학교 2 클레온
official03x@hanyang.ac.kr, gusals005@hanyang.ac.kr, hanbong.yoo@klleon.io

Multi Face Tracking using Shot Change Detection in Video Contents

Hankyu Park O1 Hyunmin Lee 1 Hanbong Yoo 2
1 Hanyang University 2 KLleon

요 약

기존의 다중 얼굴 추적 모델은 장면 전환이 빈번하거나 한 프레임에 동일 인물이 다수 존재하는 영상에 대해 제대로 얼굴을 추적하지 못하는 한계가 존재한다. 이러한 문제를 해결하기 위해 영상을 장면 단위로 구분하여 다른 장면에 있는 동일 인물의 얼굴 ID 를 통합하는 방법을 제안하였고 우수한 성능의 다중 얼굴 추적 결과를 확인하였다. 또한 같은 프레임에 존재하는 동일 인물의 얼굴 ID 를 통합하는 방법을 최초로 제안하였다. 다양한 형식의 영상 콘텐츠에서 우수한 성능의 다중 얼굴 추적 결과 값을 활용하여 다대 다 얼굴 변환, 얼굴 합성 등의 작업 성능이 더 향상될 것으로 기대한다.

1. 서 론

드라마, 영화, 뮤직비디오 등의 미디어 산업이 날이 갈수록 커지고 있고, 유튜브, 틱톡, 카멜로 등 동영상 기반 소셜 미디어 서비스의 이용자 수도 점점 많아지고 있다. 영상 콘텐츠를 이용하는 사용자가 많아지면서 영상에 등장하는 사람의 얼굴을 추적하는 작업도 매우 중요해지고 있다. 얼굴 추적 작업을 통해서 얼굴을 변환하거나 얼굴 정보를 추가로 활용하여 창작활동이 가능하고 부가가치를 창출할 수 있기 때문이다.

이러한 중요성으로 인해 현재 다양한 얼굴 추적 모델에 대한 연구가 이루어지고 있지만 두 가지의 문제점이 존재한다. 첫 번째는 콘텐츠에 장면 전환이 빈번하게 발생함에 따라 얼굴의 ID 를 유지하는 것이 힘들다는 점이다. 두 번째는 그림 1 과 같이 한 프레임에 동일 인물의 얼굴이 다수 존재하듯이 편집된 영상 콘텐츠에 대해서 동일한 ID 를 부여하지 못한다는 점이다.

첫 번째 문제점을 해결하기 위해 영상의 장면을 구별하여 다른 장면에 있는 동일 인물의 ID 를 통합하는 방법을 제안한다. 두 번째 문제점을 해결하기 위해 같은 프레임 안에 존재하는 얼굴들의 특징(feature) 정보를 추출하여 코사인 유사도 비교를 통한 ID 통합 방법을 제안한다. 성능을 측정하기 위하여 장면 전환이 빈번한 영상과 한 프레임에 동일 인물의 얼굴이 다수 등장하는 영상을 직접 수집하여 숏폼 데이터셋을 구성하였다. 본 논문의 주요 기여는 다음과 같이 정리할 수 있다.

1. 입력된 영상을 장면 단위로 구분하고, 장면 안에서의 ID 통합과 장면 간의 ID 통합 방법을 제안하였다.
2. 한 프레임에 존재하는 동일 인물 얼굴들의 ID 통합 방법을 제안하였다.

그림 1. 한 프레임에 동일 인물의 얼굴이 다수 존재하는 영상

2. 관련 연구

2.1 얼굴 탐지

얼굴 탐지는 사진 혹은 영상에서 얼굴이 존재하는 위치를 찾아내는 작업이다. MTCNN [1] 은 P-Net, R-Net, O-Net 으로 구성된 Cascade CNN(Convolutional Neural Network)으로 얼굴 탐지, 얼굴 경계 상자 회귀, 얼굴 정렬 작업을 동시에 학습시키는 방법을 제시하였다.

RetinaFace [2] 는 얼굴 경계 상자와 신뢰도, 랜드마크 정보, 3D 얼굴 위치 정보를 사용하여 동시에 학습하는 단일 단계 얼굴 탐지 방법을 제시하였다. 높은 성능과 가벼운 모델 구조로 이루어졌으며, 얼굴 변환 같은 후처리 작업을 위한 랜드마크 정보를 획득할 수 있기 때문에 본 연구에서는 해당 모델을 사용하였다.

2.2 얼굴 인식

얼굴 인식은 기존의 데이터베이스 혹은 새로 입력된 사진 또는 영상에 있는 얼굴을 식별하는 작업이다. CurricularFace [3] 는 마진 기반 손실 함수와 마이닝 기반 전략을 결합하여, 초기 훈련 단계에서 쉬운 샘플을 할당하고, 나중에 어려운 샘플을 할당하는 알고리즘이다.

ArcFace [4] 는 같은 사람 얼굴의 특징을 가깝게 하고, 다른 사람 얼굴의 특징을 멀리하여 얼굴 인식 분별력을 향상 시킨다. 얼굴 인식 분야에서 범용적으로 사용되며, RetinaFace [2] 와 주로 같이 사용되기 때문에 본 논문에서 해당 모델을 얼굴 인식 모델로 사용한다.

2.3 다중 객체 추적

다중 객체 추적은 영상에서 여러 객체를 자동으로 식별하고 추적하여 궤적의 집합으로 나타내는 작업이다. SOTMOT [5] 는 객체 감지를 위한 분기와 단일 객체 추적 분기를 함께 사용한 다중 객체 추적 구조를 제안하였다.

FairMOT [6] 는 경계 상자가 없는 탐지 구조에 Re-ID 방법을 결합한 다중 객체 추적 방법이다. Re-ID 는 영상에서 다른 프레임에 존재하는 같은 객체를 연결하는 작업이다. FairMOT 는 1) 인코더-디코더를 통과하여 고해상도 특징 지도(feature map)를 추출하고, 2) 특징 지도를 사용하여 탐지 분기에서 객체를 탐지하고, 3) Re-ID 분기에서 객체 추적의 과정을 통해 객체에 ID 를 부여한다.

객체와 다르게 얼굴을 추적할 때에는 얼굴의 특징을 기반으로 ID 를 연결해야 한다. 본 논문에선 FairMOT 의 다중 객체 추적 알고리즘을 얼굴 추적에 적용한다.

3. 제안 방법

3.1 개요

그림 2 는 제안 방법의 전체적인 진행 순서를 나타낸다. 먼저, 입력된 영상은 장면 인식 모델 [7] 을 사용하여 장면 단위로 분할된다. 하나의 장면은 세 단계를 거쳐 짧은 구간 경로(tracklet)를 생성한다. 짧은 구간 경로는 다중 객체 추적 알고리즘을 통해 임시로 생성된 객체 ID 의 집합을 의미한다.

첫 번째 단계는 얼굴 탐지 모델인 RetinaFace [2] 를 사용해서 모든 프레임에서의 얼굴 경계 상자 $b = (x_1, y_1, x_2, y_2)$ 를 탐지한다. 여기서 (x_1, y_1)과 (x_2, y_2)는 각각 경계 상자의 좌상단, 우하단 좌표를 뜻한다. 두 번째 단계는 얼굴 인식 모델인 ArcFace [4] 를 사용해서 얼굴 경계 상자에 해당하는 512 차원의 얼굴 특징 정보를 추출한다. 세 번째 단계는 다중 객체 추적 알고리즘 FairMOT [6] 에 얼굴 경계 상자와 얼굴 특징 정보를 넘겨 장면 내에서의 짧은 구간 경로(tracklet)를 생성한다.

마지막으로 각 장면에서 도출된 짧은 구간 경로를 장면끼리 비교하여 전체 구간 경로(trajectory)를 도출한다. 전체 구간 경로는 서로 다른 짧은 구간 경로에 존재하는 동일한 ID 의 객체를 연결하여 생성된 객체 ID 의 최종 집합을 의미한다. 짧은 구간 경로와 전체 구간 경로에 대한 내용은 각각 3.2 절, 3.3 절에서 설명한다.

3.2 장면 내 짧은 구간 경로(Tracklet) 생성 과정

입력으로 영상이 들어오면 영상을 프레임 단위로 나누어 장면 인식 모델 [7] 을 통해 장면 목록 $\{S_1, S_2, \dots, S_N\}$으로 구성한다. 여기서 N 은 입력 영상의 장면 개수를 뜻한다. 각 장면마다 얼굴 탐지 및 얼굴 인식 모델을 사용하여 얼굴 경계 상자와 얼굴 특징 정보를 추출한다.

그 다음 FairMOT 알고리즘을 활용하여 두 번의 결합을 통해 다중 얼굴 추적이 진행된다. 첫 번째 결합은 탐지된 얼굴 특징들의 코사인 유사도(cosine similarity)가 임계값 θ_1 이상일 경우에 이루어진다. 두 번째 결합은 인접한 프레임 간 탐지된 얼굴 경계 상자의 IoU(Intersection over Union) 값이 임계값 θ_2 이상일 경우에 이루어진다. 최종적으로 장면 안에서 각각의 ID 에 대한 짧은 구간 경로 집합 $\{t_1^n, t_2^n, \dots, t_{i_n}^n\}\ (n \in N)$이 생성된다. 여기서 i_n 은 n 번째 장면 S_n 에서의 짧은 구간 경로의 수를 의미한다.

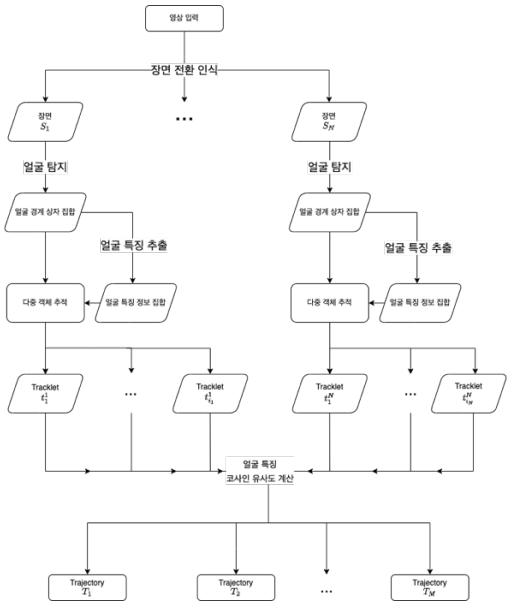

그림 2. 제안 방법의 순서도

3.3 장면 간 전체 구간 경로(Trajectory) 생성 과정

장면 간 동일 인물의 ID 를 통합하는 작업을 진행한다. 서로 다른 장면에 있는 짧은 구간 경로 t_i^n 와 t_i^{n+1} 에서 신뢰도가 가장 높은 얼굴의 썸네일 이미지를 추출한다. 얼굴 특징의 코사인 유사도를 비교하여 코사인 유사도가 임계값 θ_3 보다 높은 경우 두 짧은 구간 경로를 같은 ID 로 처리한다. 위 과정이 모든 장면의 모든 짧은 구간 경로에 대해 진행되면 최종적으로 얼굴에 대한 전체 구간 경로 집합 $\{T_1, T_2, \dots, T_M\}$ 이 도출된다. 여기서 M 은 전체 구간 경로의 개수를 의미한다.

4. 실 험

4.1 실험 정보

평가 방법은 다중 객체 추적 방법에서 널리 사용되고 있는 CLEAR MOT [8] 를 적용하였다. CLEAR MOT 를 구성하는 각 평가 요소에 대한 설명은 표 1 에 기술하였다. 평가 요소 옆에 기술된 화살표는 값이 높을수록 혹은 낮을수록 좋음을 뜻한다. MOTA 와 MOTP 는 식(2), (3)과 같다.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (1)$$

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \quad (2)$$

m_t 는 미탐지 횟수, fp_t 는 오탐지 횟수, mme_t 는 ID 연결 오류 횟수를, $\sum_t g_t$ 는 객체의 개수를 의미한다. c_t 는 시간 t 에서 발견된 ID 연결의 횟수, d_t^i 는 실제 객체와 탐지된 객체 사이의 거리를 의미한다.

데이터셋은 유튜브에 게재된 광고, 예능 클립 등 1 분 내외의 영상 20 개를 직접 수집하여 구성하였다. 수집된 데이터셋은 장면 전환이 빈번하거나 한 프레임에 동일 인물의 얼굴이 다수 등장하는 영상으로 구성되었다. 실험은 [9]를 기반으로 구현된 다중 얼굴 추적 알고리즘 [10]과 성능 및 정확도를 비교하는 방향으로 진행되었다. 수치 비교를 통한 정량적 성능 측정을 위해 파이썬 라이브러리 motmetrics [11] 를 활용하였다. 임계값 θ_1 은 0.35, θ_2 는 0.5, θ_3 는 0.25 로 설정하였다.

표 1. 평가 지표 설명

평가 요소	설명
Recall↑	경계 상자 탐지에 대한 정밀도
Precision↑	경계 상자 탐지에 대한 재현율
IDs↓	객체에 부여된 ID가 바뀌어 판별된 수
Frag↓	탐지를 놓쳐 추적이 중단된 경우의 수
MOTA↑	다중 객체 추적 정확도
MOTP↓	경계 상자 위치의 오류 비율

표 2. 제안 방법 실험 결과

방법	Recall	Precision	IDs	Frag	MOTA	MOTP
[10]	**0.889**	0.826	322	**255**	0.610	0.155
제안	0.770	**0.914**	**124**	471	**0.690**	**0.123**

4.2 실험 결과

[10]의 방법과 제안 방법으로 진행된 숏폼 데이터셋에 대한 다중 얼굴 추적 실험의 정량적 결과는 표 2 와 같다. Recall, Precision, F1, MOTA, MOTP 는 20 개 영상에서 구한 값을 각각 산술 평균하여 도출하였다. IDs 와 Frag 값은 20 개 영상에서 구한 값을 모두 합하여 구하였다. 표 2 에서 [10]의 방법보다 제안 방법이 Precision, IDs, MOTA, MOTP 에서 더 나은 결과 값을 나오는 것으로 볼 때 제안 방법이 우수한 성능으로 다중 얼굴을 추적하는 것을 확인할 수 있다. 반면 제안 방법의 Recall 값과 Frag 값은 [10]의 방법보다 낮게 나오는 것을 확인할 수 있는데 Face Detection 성능과 장면 인식 모델의 정확도가 향상된다면 충분히 개선될 여지가 있다.

한 프레임 내 동일 인물의 여러 얼굴을 추적한 결과는 그림 3 과 같다. 왼쪽 그림은 [10]으로 다중 얼굴 추적을 진행했을 때 나타난 결과 값이다. 한 프레임 안에서 탐지된 두 개의 얼굴이 동일 인물임에도 서로 다른 ID 를 부여한 것을 확인할 수 있다. 오른쪽 그림은 본 논문에서 제안한 방법으로 도출된 다중 얼굴 추적 결과 값이다. 왼쪽 그림과 다르게 두 개의 얼굴에 동일한 ID 를 부여하며 편집된 영상에서도 충분히 다중 얼굴 추적을 진행하는 것을 확인할 수 있다.

제안 방법을 통해 숏폼 데이터셋에서 장면 전환이 빈번할 때 다중 얼굴을 추적한 결과는 그림 4 와 같다. 그림 4 의 사진 4 장은 시계방향으로 각각 33 번째, 123 번째, 316 번째, 223 번째 프레임이다. 영상의 길이는 총 361 프레임으로 영상에 등장하는 얼굴들이 초반, 중반 및 후반까지 ID 의 변환이 발생하지 않고 추적이 이루어지는 것을 정성적으로 확인할 수 있다.

5. 결 론

본 논문에서 제안한 다중 얼굴 추적 방법은 장면이 전환될 때에도 추적 성능이 우수하고, 한 프레임 내 동일 인물의 여러 얼굴에 ID 를 올바르게 부여하는 모습을 보여준다. 또한 본 연구의 다중 얼굴 추적 결과를 바탕으로 영상 기반 소셜 미디어에서 높은 정확도를 가진 다대다 얼굴 변환, 얼굴 합성 등의 작업이 가능해질 것으로 기대된다.

본 논문에서 제안한 방법에서 얼굴 탐지 성능 향상, 더 좋은 장면 전환 알고리즘을 도입, 짧은 구간 경로 결합 시 다른 결합 방법 탐구 등을 통해 성능을 더 향상시키는 것을 향후의 목표로 한다.

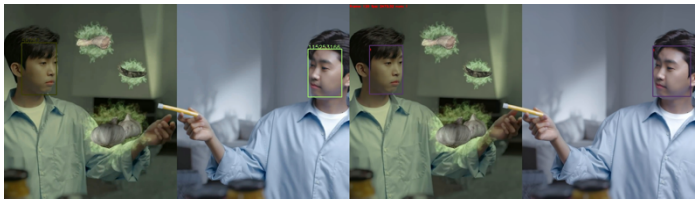

그림 3. 동일 인물 다중 얼굴 추적 결과

그림 4. 장면 전환이 빈번할 때의 다중 얼굴 추적 결과

6. 참고문헌

[1] Zhang, Kaipeng, et al. "Joint face detection and alignment using multitask cascaded convolutional networks." in IEEE signal processing letters 23.10, 2016

[2] Deng, Jiankang, et al. "Retinaface: Single-shot multi-level face localisation in the wild." in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

[3] Huang, Yuge, et al. "Curricularface: adaptive curriculum learning loss for deep face recognition." in proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020.

[4] Deng, Jiankang, et al. "Arcface: Additive angular margin loss for deep face recognition." in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.

[5] L. Zheng, M. Tang, Y. Chen, G. Zhu, J. Wang and H. Lu, "Improving Multiple Object Tracking with Single Object Tracking" in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[6] Zhang, Yifu, et al. "Fairmot: On the fairness of detection and re-identification in multiple object tracking." in International Journal of Computer Vision, 2021.

[7] https://github.com/Breakthrough/PySceneDetect

[8] Bernardin, Keni, and Rainer Stiefelhagen. "Evaluating multiple object tracking performance: the clear mot metrics." in EURASIP Journal on Image and Video Processing, 2008.

[9] C. Lin and Y. Hung, "A Prior-Less Method for Multi-face Tracking in Unconstrained Videos" in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[10] https://github.com/chuckan13/multi-face-tracking-project

[11] https://github.com/cheind/py-motmetrics