



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년08월16일
(11) 등록번호 10-2011402
(24) 등록일자 2019년08월09일

(51) 국제특허분류(Int. Cl.)
G06K 9/00 (2006.01) G06K 17/00 (2006.01)
G06T 7/246 (2017.01)
(52) CPC특허분류
G06K 9/00221 (2013.01)
G06T 7/246 (2017.01)
(21) 출원번호 10-2018-0109613
(22) 출원일자 2018년09월13일
심사청구일자 2018년09월13일
(30) 우선권주장
1020180097608 2018년08월21일 대한민국(KR)
(56) 선행기술조사문헌
KR1020140058091 A*
KR1020140095333 A*
KR1020170084657 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
한밭대학교 산학협력단
대전광역시 유성구 동서대로 125 (덕명동)
(주)큐디스
서울특별시 금천구 가산디지털1로 168, 우림라이온스밸리 비동 404호, 405호, 406호 (가산동)
(72) 발명자
이승호
대전광역시 유성구 엑스포로 448, 402동 402호(전민동, 엑스포아파트)
김재형
대전광역시 동구 성남로 15, 108동 605호(성남동, 스마트뷰)
이선구
대전광역시 유성구 덕명로 63, 106동 802호(덕명동, 하우스토리아파트)
(74) 대리인
이은철

전체 청구항 수 : 총 3 항

심사관 : 노용완

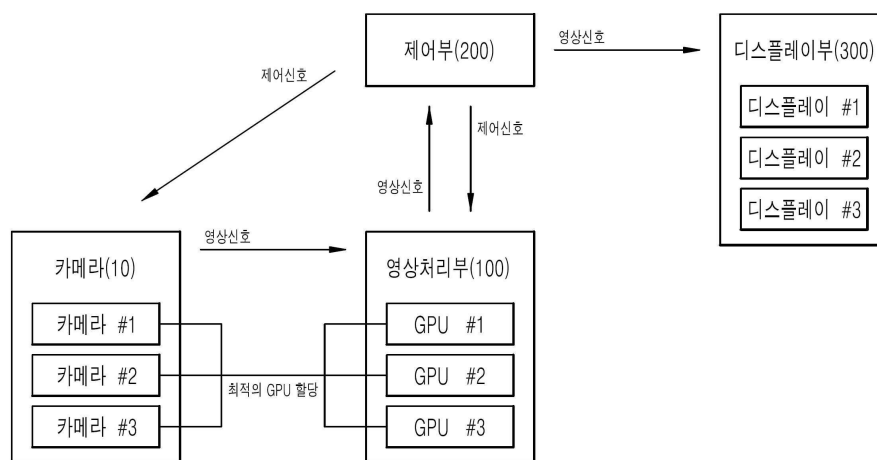
(54) 발명의 명칭 쿠다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템

(57) 요약

본 발명은 쿠다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템에 관한 것으로서, GPU로 구성되어 카메라를 통해 입력된 영상으로부터 그리드 방식으로 얼굴 객체를 검출 및 트래킹을 수행하는 영상처리부; 및 쿠다를 통해 GPU 각각의 메모리 사용량과 활성화된 코어의 수를 지표를 토대로 GPU를 선별하고, 선별한 GPU를 스레드에 할당하는 제어부를 포함한다.

상기와 같은 본 발명에 따르면, 종래의 얼굴 검출용 알고리즘을 딥러닝을 사용한 방식으로 대체하고, 이에 따라 증가하는 연산량을 GPU 리소스 분배 알고리즘으로 해소함으로써, 효율적인 리소스 분배를 통해 실시간으로 얼굴 검출 및 인식이 가능하고, CCTV 등을 관제하는 인력 낭비, 시간 낭비를 줄이고, 공항 블랙리스트 식별 및 트래킹, 실종사고 조기 해결 등의 서비스가 가능하다.

대표도 - 도1



S

(52) CPC특허분류

G06K 2017/0045 (2013.01)

G06T 2200/28 (2013.01)

G06T 2207/30201 (2013.01)

공지예외적용 : 있음

명세서

청구범위

청구항 1

GPU로 구성되어 카메라를 통해 입력된 영상으로부터 그리드 방식으로 얼굴 객체를 검출 및 트래킹을 수행하는 영상처리부; 및

쿠다를 통해 상기 GPU 각각의 메모리 사용량과 활성화된 코어의 수를 지표를 토대로 GPU를 선별하고, 선별한 GPU를 스레드에 할당하는 제어부를 포함하되,

상기 영상처리부는,

상기 카메라를 통해 입력된 영상으로부터 그리드 방식으로 객체를 검출하고,

딥러닝 학습을 통해 검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는지 여부를 판단하며,

검출한 객체가 찾고자 하는 타겟과 일치하는지 여부를 판단하되,

검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는 경우, 일치하는 영상에 박스 형태로 RoI(Region of Interest)를 지정하고,

상기 제어부는,

상기 GPU 각각의 메모리 사용량과 활성화된 코어의 수를 토대로 각 GPU의 현재 사용량을 도출하여 리소스 점유율이 가장 낮은 GPU를 스레드에 할당하되,

상기 얼굴 객체를 검출 및 트래킹할 때 마다 각 GPU의 현재 사용량을 도출하고, 스레드가 할당된 GPU의 점유율이 기 설정된 수준 이상으로 높아질 경우, 각 GPU의 현재 사용량을 도출하여 리소스 점유율이 가장 낮은 GPU를 스레드에 재할당하는 것을 특징으로 하는 쿠다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 영상처리부는,

상기 입력된 영상으로부터 얼굴 객체 검출, 및 검출한 얼굴 객체가 타겟과 일치하는지 여부를 판단하는 절차를 기 설정된 기준에 부합하도록 반복 수행하여 상기 RoI에 대한 트래킹을 수행하는 것을 특징으로 하는

쿠다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템.

청구항 4

제1항에 있어서,

상기 영상처리부는,

상기 RoI에 대한 트래킹 중에 카메라 화각의 중앙 지점에서 일정 범위 이상 벗어나는 경우, PTZ 제어를 통해 상기 RoI가 화각의 중심부에 위치하도록 상기 카메라의 화각을 조정하는 것을 특징으로 하는

쿠다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템.

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

삭제

발명의 설명

기술 분야

- [0001] 본 발명은 쿠다(Compute Unified Device Architecture, CUDA)를 사용하여 GPU(Graphics Processing Unit) 리소스를 분배하는 지능형 얼굴 인식/트래킹 시스템 및 그 방법에 관한 것으로 더욱 상세하게는, 종래의 얼굴 인식 및 트래킹 시스템의 단점을 극복하고 시스템의 안정성과 성능 향상을 위해 다중 GPU를 구성하고, GPU 리소스를 효율적으로 분배하여 실시간 영상에 찾고자 하는 대상을 빠르게 찾고 트래킹이 가능한 기술에 관한 것이다.

배경 기술

- [0002] 얼굴 인식 및 트래킹 시스템은 첫 번째로 얼굴 검출용 알고리즘 사용방식, 두 번째는 색상 기반의 트래킹 방식, 세 번째는 샘플링 된 영상에서의 얼굴 인식 방식, 네 번째는 GPU 리소스를 분배하지 않는 방식으로 4가지 방식이 존재하고 있으나, 이러한 종래의 얼굴 인식 및 트래킹 시스템에는 아래와 같은 문제점이 있다.
- [0003] 먼저, 얼굴 검출용 알고리즘을 사용하는 방식은, 얼굴의 각도에 따라 특징이 달라지기 때문에 얼굴의 각도에 따라 다른 알고리즘으로 얼굴을 검출해야 하는 번거로움이 있다.
- [0004] 또한, 색상 기반의 트래킹 방식은 현재 RoI 영역의 색상 정보를 활용하는 방식으로, 트래킹 대상이 기둥 등의 장애물에 가려졌다 나타나거나 카메라 화각 밖으로 나갔다 들어오는 등의 상황에서 트래킹 실패 확률이 매우 높다는 문제점이 있다.
- [0005] 또한, 샘플링 된 영상에서의 얼굴 인식 방식은, 실시간 인식을 할 수 없고 샘플링 된 영상에서 영상이 떨리거나 초점이 흐트러질 경우, 검출 및 인식률이 확연하게 떨어지는 문제점이 있다.
- [0006] 그리고, GPU 리소스 분배를 하지 않는 방식에서는, 연산량이 적은 시점에서는 문제가 발생하지 않으나 많은 연산량이 발생할 경우 속도가 느려지거나 GPU에 과부하가 걸려 시스템의 안정성에 문제가 발생하는 단점이 있다.

선행기술문헌

특허문헌

- [0007] (특허문헌 0001) 한국등록특허 제10-1850286호

발명의 내용

해결하려는 과제

- [0008] 본 발명의 목적은, 종래의 얼굴 인식 및 트래킹 시스템에서 사용하는 얼굴 검출용 알고리즘을 딥러닝을 사용한 방식으로 대체하고, 이에 따라 증가하는 연산량을 효율적으로 분배하는 GPU 리소스 분배 알고리즘으로 해소함으로써, 효율적인 리소스 분배를 통해 실시간으로 얼굴 검출 및 인식이 가능하고, 이에 따라 검출 및 인식을 반복해 실시간 트래킹 시스템을 구현하는데 있다.
- [0009] 본 발명의 목적은, GPU 리소스 분배 알고리즘을 적용하여 실시간 트래킹 시스템을 제공함으로써, CCTV 등을 관제하는 인력 낭비, 시간 낭비를 줄이고, 공항 블랙리스트 식별 및 트래킹, 실종사고 조기 해결 등의 서비스를 가능하게 하는데 있다.

과제의 해결 수단

- [0010] 이러한 기술적 과제를 달성하기 위한 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템은, GPU로 구성되어 카메라를 통해 입력된 영상으로부터 그리드 방식으로 얼굴 객체를 검출 및 트래킹을 수행하는 영상처리부; 및 쿼터를 통해 GPU 각각의 메모리 사용량과 활성화된 코어의 수를 지표를 토대로 최적의 GPU를 선별하고, 선별한 GPU를 스레드에 할당하는 제어부를 포함한다.
- [0011] 바람직하게는, 영상처리부는 카메라를 통해 입력된 영상으로부터 그리드 방식으로 객체를 검출하고, 검출한 객체가 찾고자 하는 타겟과 일치하는지 여부를 판단하되, 검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는 경우, 일치하는 영상에 박스 형태로 RoI(Region of Interest)를 지정하는 것을 특징으로 한다.
- [0012] 영상처리부는, 입력된 영상으로부터 얼굴 객체 검출, 및 검출한 얼굴 객체가 타겟과 일치하는지 여부를 판단하는 절차를 기 설정된 기준에 부합하도록 반복 수행하여 상기 RoI에 대한 트래킹을 수행하는 것을 특징으로 한다.
- [0013] 영상처리부는, RoI에 대한 트래킹 중에 카메라 화각의 중앙 지점에서 일정 범위 이상 벗어나는 경우, PTZ 제어를 통해 RoI가 화각의 중심부에 위치하도록 카메라의 화각을 조정하는 것을 특징으로 한다.
- [0014] 제어부는, GPU 각각의 메모리 사용량과 활성화된 코어의 수를 토대로 각 GPU의 현재 사용량을 도출하여 가장 점유율이 낮은 GPU를 스레드에 할당하되, 스레드가 할당된 GPU의 점유율이 기 설정된 수준 이상으로 높아질 경우, 각 GPU의 현재 사용량을 도출하여 가장 점유율이 낮은 GPU를 스레드에 재할당하는 것을 특징으로 한다.
- [0015] 전술한 시스템을 기반으로 하는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 방법은, 제어부가 다수의 GPU 리소스를 분배하는 (a) 단계; 및 디스플레이부가 영상처리부가 입력받은 영상과, 영상으로부터 검출한 얼굴 객체 및 트래킹을 출력하는 (b) 단계를 포함하는 것을 특징으로 한다.
- [0016] 바람직하게는 (a) 단계는, 제어부가 GPU 각각의 메모리 사용량과 활성화된 코어의 수를 검출하여 각 GPU가 현재 어느 정도 사용되고 있는지를 파악하는 (a-1) 단계; 제어부가 각 GPU의 현재 리소스 사용 상태에 따라 가장 점유율이 낮은 GPU를 스레드에 할당하는 (a-2) 단계; 제어부가 GPU 점유율이 기 설정된 수준 이상으로 높아지는지 여부를 판단하는 (a-3) 단계; 및 (a-3) 단계의 판단결과, GPU 점유율이 기 설정된 수준 이상으로 높아지는 경우, 제어부가 (a-1) 단계로 절차를 이행하는 (a-4) 단계를 포함하는 것을 특징으로 한다.
- [0017] (b) 단계는, 영상처리부가 입력된 영상으로부터 그리드 방식으로 얼굴 객체를 검출하는 (b-1) 단계; 영상처리부가 검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는지 여부를 판단하는 (b-2) 단계; (b-2) 단계의 판단결과, 검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는 경우, 영상처리부가 일치하는 영상에 빨간색 박스 형태로 RoI(Region of Interest)를 지정하는 (b-3) 단계; 영상처리부가 지정한 RoI에 대한 트래킹을 수행하는 (b-4) 단계; 영상처리부가 지정한 RoI에 대한 트래킹 중에 카메라 화각의 중앙 지점에서 일정 범위 이상 벗어나는지 여부를 판단하는 (b-5) 단계; (b-6) 단계의 판단결과, 카메라 화각의 중앙 지점에서 일정 범위 이상 벗어나는 경우, 영상처리부가 PTZ(Pan Tilt Zoom) 제어를 통해 해당 RoI가 화각의 중심부에 위치하도록 카메라 화각을 조정하는 단계를 포함하는 것을 특징으로 한다.

발명의 효과

- [0018] 상기와 같은 본 발명에 따르면, 종래의 얼굴 인식 및 트래킹 시스템에서 사용하는 얼굴 검출용 알고리즘을 딥러닝을 사용한 방식으로 대체하고, 이에 따라 증가하는 연산량을 효율적으로 분배하는 GPU 리소스 분배 알고리즘으로 해소함으로써, 효율적인 리소스 분배를 통해 실시간으로 얼굴 검출 및 인식이 가능하고, 이에 따라 검출 및 인식을 반복해 실시간 트래킹 시스템 구현이 가능한 효과가 있다.
- [0019] 본 발명에 따르면, GPU 리소스 분배 알고리즘을 적용하여 실시간 트래킹 시스템을 제공함으로써, CCTV 등을 관제하는 인력 낭비, 시간 낭비를 줄이고, 공항 블랙리스트 식별 및 트래킹, 실종사고 조기 해결 등의 서비스가 가능한 효과가 있다.

도면의 간단한 설명

- [0020] 도 1은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템을 도시한 구성도.

도 2는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 GPU 리소스 분배 알고리즘의 흐름도.

도 3은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 카메라 화각 조정을 도시한 예시도.

도 4는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 순서도.

도 5는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 GPU 리소스 사용 상태의 예시도.

도 6은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 스레드 #1에 최적의 GPU 할당을 도시한 예시도.

도 7은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 모든 스레드에 최적의 GPU 할당 순서도.

도 8은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 스레드 #1에 대하여 GPU #2를 재할당하는 과정을 도시한 예시도.

도 9는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 내부 Network 설명도.

도 10은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 딥러닝을 이용한 얼굴영역 검출 과정을 도시한 예시도.

도 11은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 딥러닝을 이용한 얼굴 인식 과정을 도시한 예시도.

도 12는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 학습과정에 사용되는 Network 설명도.

도 13은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 인식과정에서 사용되는 Network 설명도.

도 14는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 딥러닝을 이용한 실시간 얼굴 추적 결과 이미지를 도시한 예시도.

도 15는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 일반 카메라와 PTZ 카메라의 화각을 비교한 예시도.

도 16은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 PTZ 제어를 통한 트래킹 대상 추적을 도시한 예시도.

도 17은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 GPU 리소스 분배 차이에 따른 결과 그래프.

도 18은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템의 GPU 리소스 분배 전/후의 GPU 점유율 확인 결과 그래프.

도 19는 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 방법을 도시한 순서도.

도 20은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 방법의 제S100단계를 도시한 순서도.

도 21은 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 방법의 제S200단계를 도시한 순서도.

발명을 실시하기 위한 구체적인 내용

- [0021] 본 발명의 구체적인 특징 및 이점들은 첨부 도면에 의거한 다음의 상세한 설명으로 더욱 명백해질 것이다. 이에 앞서, 본 명세서 및 청구범위에 사용된 용어나 단어는 발명자가 그 자신의 발명을 가장 최선의 방법으로 설명하기 위해 용어의 개념을 적절하게 정의할 수 있다는 원칙에 입각하여 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야 할 것이다. 또한, 본 발명에 관련된 공지 기능 및 그 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는, 그 구체적인 설명을 생략하였음에 유의해야 할 것이다.
- [0022] 도 1에 도시된 바와 같이, 본 발명의 일 실시예에 따른 쿼다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템(S)은, 다수의 GPU로 구성되어 카메라(1)를 통해 입력된 영상으로부터 그리드 방식으로 얼굴 객체를 검출 및 트래킹을 수행하는 영상처리부(100) 및 쿼다(CUDA)를 통해 GPU 각각의 메모리 사용량과 활성화된 코어의 수를 지표를 토대로 최적의 GPU를 선별하고, 선별한 GPU를 스레드(thread)에 할당하는 제어부(200), 및 영상처리부(100)가 입력받은 영상과, 영상으로부터 검출한 얼굴 객체 및 트래킹을 출력하는 디스플레이부(300)를 포함한다.
- [0023] 이때, 쿼다를 이용해 다수의 GPU들 중에서 최적의 GPU 리소스 분배하는 알고리즘에 대한 일련의 과정은 도 2에 도시된 바와 같다.
- [0024] 또한, GPU를 사용하기 위한 API(Application Programming Interface)로 쿼다를 사용하며, 쿼다를 사용하는 이유는 많은 양의 연산을 동시에 처리하기에 적합한 하드웨어가 GPU이기 때문이다. 하나의 코어만을 놓고 비교했을 때 CPU의 연산속도가 월등하지만, 일반적으로 쓰이는 CPU는 4코어 내지 8코어인데 비해 GPU는 1500개 이상의 코어를 가지고 있다.
- [0025] 따라서, GPU를 이용해 많은 양의 연산을 동시에 처리할 수 있으며, 본 발명의 일 실시예에서는 GPU를 프로그래밍하기 쉽게 만들어주는 API로 쿼다를 적용하였고, 영상처리부(100)는 적어도 3개 이상의 GPU를 구비하도록 구성된다.
- [0026] 도 3을 참조하면, 영상처리부(100)는 다수의 GPU를 포함하는 멀티 GPU로 구성되어 제어부(200)의 GPU 리소스 분배에 따라 작업을 수행하되, 카메라(1)를 통해 입력된 영상으로부터 그리드 방식으로 얼굴 객체를 검출하고, 검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는지 여부를 판단한다.
- [0027] 이때, 영상처리부(100)가 검출한 객체는 얼굴 영역 또는 특정 사물 중에 어느 하나로 설정될 수 있고, 검출한 객체가 찾고자 하는 타겟과 일치하는지 여부의 판단은 딥러닝 학습을 통해 도출한다.
- [0028] 영상처리부(100)는 검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는 경우, 일치하는 영상에 빨간색 박스 형태로 RoI(Region of Interest)를 지정하고, 박스 위에 해당 얼굴 객체의 이름 정보를 표시한다. 이때, RoI 지정 박스의 색상은 사용자 설정에 의해 변경될 수 있다.
- [0029] 영상처리부(100)는 전술한 최적의 GPU에 스레드 할당, 카메라(1)를 통해 입력된 영상으로부터 얼굴 객체 검출, 및 검출한 얼굴 객체가 타겟과 일치하는지 여부를 판단하는 절차를 기 설정된 기준에 부합하도록 반복 수행하여 지정한 RoI에 대한 트래킹을 수행한다.
- [0030] 영상처리부(100)는 지정한 RoI에 대한 트래킹 중에 카메라(1) 화각의 중앙 지점에서 일정 범위 이상 벗어나는 경우, 도 4에 도시된 바와 같이 PTZ(Pan Tilt Zoom) 제어를 통해 해당 RoI가 화각의 중심부에 위치하도록 카메라(1) 화각을 조정한다.
- [0031] 이때, 카메라(1)는 영상처리부(100)에 구비된 GPU와 대응하는 개수로 구성될 수 있다.
- [0032] 이하, 제어부(200)가 영상처리부(100)의 GPU 디바이스 제어를 위한 세부절차에 대해 살펴보면 아래와 같다.
- [0033] 첫 번째로, 제어부(200)는 도 5에 도시된 바와 같이, GPU 각각의 메모리 사용량과 활성화된 코어의 수를 검출하여 각 GPU가 현재 어느 정도 사용되고 있는지를 파악한다.
- [0034] 두 번째로, 제어부(200)는 각 GPU의 현재 리소스 사용 상태에 따라 가장 점유율이 낮은 GPU를 스레드에 할당한다.
- [0035] 이때, 메모리 사용량과 활성화된 코어의 개수를 모두 고려하여 최적의 GPU를 할당하며, 도 6에 도시된 예와 같이, 제어부(200)가 리소스 점유율이 가장 낮은 GPU #2를 스레드 #1에 할당하게 된다.
- [0036] 아울러, 제어부(200)는 스레드 #2와 스레드 #3 들에 대해서도 첫 번째와 두 번째 과정을 수행하여 그림 7에 도

시된 바와 같이 모든 스레드에 대하여 최적의 GPU를 할당하게 된다.

- [0037] 이때, 제어부(200)는 모든 스레드에 대하여 최적의 GPU를 할당한 후에는 첫 번째와 두 번째 과정을 매 프레임 반복시 오히려 GPU에 과부하가 발생될 우려가 있다.
- [0038] 따라서, 세 번째로, 제어부(200)는 GPU 점유율이 기 설정된 수준 이상으로 높아질 경우에만 다시 첫 번째와 두 번째 과정을 수행한다.
- [0039] 이때, 기준치를 높여 알고리즘을 실행할 경우에는 예상보다 빠르게 GPU 점유율이 높아져 GPU를 스윙칭하기 전에 과부하가 걸릴 우려가 있다. 반대로 기준치를 낮추는 경우에는 GPU 스윙칭이 너무 빈번하게 발생해 오히려 성능이 저하되는 현상이 발생하는 문제가 있다.
- [0040] 따라서, 전술한 첫 번째와 두 번째 과정을 다시 수행하는 기준은 GPU 점유율 80%인 경우로 상정할 수 있고, 이 기준은 실험에 의해 도출되었으나 GPU의 성능에 따라 변경될 수 있다.
- [0041] 도 8에 도시된 바와 같이, 제어부(200)가 리소스 점유율이 가장 낮은 GPU #2를 스레드 #1에 할당하였으나, GPU #2의 점유율이 다시 수행하는 기준인 GPU 점유율 80%를 초과하였으므로 리소스 점유율이 가장 낮은 GPU #1을 스레드 #1에 재할당하도록 구성된다.
- [0042] 이하, 도 9 내지 도 13을 참조하여 본 발명의 일 실시예에 따라 영상처리부가 딥러닝을 이용해 얼굴영역 검출 및 인식하는 절차에 대해 살펴보면 아래와 같다.
- [0043] 기존 얼굴 검출 알고리즘으로 널리 쓰이는 Haar-like feature 알고리즘은 눈썹, 눈동자 등의 영역간의 특징적인 밝기 차이를 이용하여 얼굴 영역을 검출하며, 기본적으로 물체의 기하학적 정보를 유지하며 사용되기 때문에 영역 내부에서의 변화나 약간의 위치 변화 등에서는 강점을 나타낸다. 그러나 영상의 밝기가 크게 변화하는 상황, 대상 영역이 회전하는 경우, 얼굴 각 영역의 색상 대비가 뚜렷하지 않은 경우 등에서 매우 취약한 단점을 가지고 있다.
- [0044] 따라서, 본 발명의 일 실시예에는 이러한 단점을 극복하고자 얼굴 영역을 다양한 각도의 회전까지 딥러닝으로 학습시킨다.
- [0045] 얼굴 영역의 위치를 검출하기 위한 학습정보는 총 5가지의 채널이다. 첫 번째는 어떤 객체인지 인식하는 채널, 두 번째는 위치 정보 중의 중심좌표 x 값, 세 번째는 위치 정보 중의 중심좌표 y 값, 네 번째는 비율인 객체의 너비 w 값, 다섯 번째는 비율인 객체의 높이 h 값이다.
- [0046] 즉, 본 발명의 일 실시예는 기존의 딥 러닝과 학습과정은 같지만 객체를 검출하는 방식에서 차이점이 있다. 객체를 검출하는 방식은 신경망으로 검출하는 딥 러닝과 달리 그리드로 객체 검출을 한다.
- [0047] 본 발명의 일 실시예는 입력된 영상을 그리드로 분할하므로 처리속도가 빠르다는 장점이 있다.
- [0048] 도 9에 도시된 바와 같이 얼굴 영역 검출에 사용되는 딥러닝 Network는 총 26개의 layer로 이루어지며, 학습된 데이터들은 실시간 영상에서 얼굴 검출을 위해 사용된다.
- [0049] 영상처리부는 카메라로부터 실시간으로 입력되는 영상 내에서 얼굴이라고 판단되는 부분은 흰색 박스 형태로 표시한 후에, 다음 단계인 얼굴 인식을 위해 잘라내어 임시로 저장한다.
- [0050] 영상처리부의 얼굴영역 검출은 2가지로 나누어 객체 검출을 수행하게 되는데, 먼저 입력 이미지의 전체 크기를 '1'로 계산한다. 이에 따라 중심좌표 값과 너비, 높이는 소수점으로 도출된다.
- [0051] 이때, 입력 이미지는 x , y , w , h 및 class 다섯 가지로 값이 저장되는데, 이것은 학습한 클래스의 개수만큼 생성되며, 생성된 데이터에서 인식률을 가지고 최종 객체가 검출된다.
- [0052] 즉, 도 10에 도시된 객체검출 과정의 예와 같이, 딥러닝을 이용한 얼굴 영역 검출의 전체 흐름과 20개의 클래스로 구성될 수 있다.
- [0053] 또한, 딥러닝을 이용한 얼굴 인식은 도 11에 도시된 바와 같이, 영상처리부가 전술한 얼굴 검출 알고리즘을 이용해 사진이나 영상 등에 담긴 찾고자 하는 대상의 얼굴 영역을 검출하고 클래스를 분류해 저장 및 학습을 수행한다.
- [0054] 이어서, 영상처리부가 학습이 끝난 후에는 실시간으로 입력되는 영상에서 얼굴 검출 단계를 거쳐 임시로 저장된 얼굴 영역의 이미지를 받아와 사용자가 찾고자 하는 클래스와 일치하는 대상인지를 인식한다.

- [0055] 그리고, 영상처리부가 세 번째로 찾고자 하는 대상을 찾게 되면 빨간 박스 형태로 영상에 표시하고 박스 위에 대상의 이름을 표시한다.
- [0056] 본 발명의 일 실시예에서는 Convolution Filter와 Fully Connected Layer를 fine-tuning 하여 인식 결과의 향상을 도모하였으며, Overfitting을 방지하기 위해 Maxpooling Layer를 추가하고, Train set와 Test set을 나누어 학습한다.
- [0057] Train set은 학습 과정 중 역전파 알고리즘을 이용해 가중치를 수정하고, Test set은 일정 횟수 이상 학습 시 Train set와 다른 DB를 이용해 가중치가 Overfitting 되었는지 판단한다. Train set과 Test set은 마지막 레이어에 Softmax를 사용해 학습한다.
- [0058] 최종적으로 학습이 완료된 후에 역전파 알고리즘 없이 순방향으로만 레이어를 통과해 특징 벡터를 추출하고 그를 이용해 인식을 실시한다.
- [0059] 이때, 얼굴 인식에 사용된 Network의 각 Layer의 크기, Layer의 역할은 도 12에 도시된 바와 같고, 도 13에 도시된 바와 같이 얼굴 인식에 사용되는 딥 러닝 Network는 총 16개의 Layer로 이루어진다.
- [0060] 이하, 도 14를 참조하여 본 발명의 일 실시예에 따른 얼굴을 트래킹하는 절차에 대해 살펴보면 아래와 같다.
- [0061] 기존의 템플릿 매칭은 대상의 형태가 바뀔 경우 매칭이 잘 되지 않아 얼굴의 각도가 틀어질 경우 매칭에 실패할 확률이 높은 단점이 있고, HOG 알고리즘의 경우에는 전 프레임의 RoI 영역을 특성값을 추출해 사용하기 때문에 RoI 영역이 장애물에 가려져 영상 내에서 사라졌다 나타나는 경우에 취약하다는 단점이 있다.
- [0062] 따라서, 본 발명의 일 실시예에는 실시간 얼굴 트래킹을 효과적으로 수행하기 위하여 앞서 서술한 과정(GPU할당, 얼굴 검출, 얼굴 인식)을 반복한다.
- [0063] 이에 따라 영상처리부가 GPU 할당, 얼굴 검출, 얼굴 인식 과정을 반복하게 되면, RoI 영역이 사라졌다 나타날 경우나 카메라 화각 밖에서 안으로 들어올 경우에도 효율적으로 트래킹이 가능하며, 도 14에 도시된 바와 같이, 딥러닝을 이용한 실시간 얼굴 추적 결과 이미지로서 찾고자 하는 대상을 찾게 되면 빨간 박스 형태로 영상에 표시하고 그렇지 않은 경우에는 흰색 박스 형태로 표시한다.
- [0064] 이하, 도 15 및 도 16을 참조하여 본 발명의 일 실시예에 따른 카메라 제어에 대해 살펴보면 아래와 같다.
- [0065] 도 15에 도시된 바와 같이, 본 발명의 일 실시예에 따른 카메라는 PTZ(Pan Tilt Zoom) 제어가 가능하도록 구성되어 종래 고정형 카메라에 비해 넓은 화각을 가질 수 있고, 찾고자 하는 대상을 집중적으로 트래킹할 수 있다. 따라서, 단일 카메라의 구성으로도 여러대의 카메라가 지켜봐야할 범위에서 인식 및 트래킹이 가능하다.
- [0066] 또한, 도 16에 도시된 바와 같이, 트래킹 대상이 화각의 중앙에서 일정 거리 이상 벗어날 경우, PTZ 제어를 통해 트래킹 대상이 다시 화각의 중앙에 위치할 수 있도록 제어할 수 있다.
- [0067] 이하, 도 17 및 도 18을 참조하여 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템(S)의 GPU 리소스 분배, 얼굴 검출, 얼굴 인식 및 트래킹에 대한 실험결과에 대해 살펴보면 아래와 같다.
- [0068] 먼저, 쿼터를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템의 효율성을 측정하기 위하여, 성별, 나이, 체형이 각기 다른 150여 명의 데이터베이스를 구축하여 딥러닝을 사용하여 학습시켰다.
- [0069] 학습이 완료된 후에 GPU 리소스를 분배했을 때와 분배하지 않았을 경우로 나누고 각각의 경우에 입력되는 영상에 나타나는 사람의 수를 1명에서 5명까지 바꾸어 가면서 얼굴 검출율과 인식률, 인식 결과를 표시하는 동영상의 fps수를 측정하고 GPU의 활성화 정도를 체크하였다.
- [0070] 카메라 원본 영상은 30fps로 입력되었다. [표 1]에서 나타난 바와 같이 GPU 리소스를 분배하지 않을 경우에 영상에 등장하는 사람의 수가 많아질수록 검출 및 인식률이 급격히 저하되며, 4명이 넘는 사람이 등장 할 경우 아예 영상이 로드되지 않는 결과를 산출하였다.

표 1

GPU 리소스 분배를 하지 않았을 경우의 인식을 결과

측정 인원수	얼굴 검출률	얼굴 인식률	초당 프레임 수 (fps)
1	97%	90%	25 fps
2	95%	82%	15 fps
3	95%	75%	7 fps
4	0%	0%	0 fps
5	0%	0%	0 fps

[0071]

[0072]

[표 2]에서 GPU 리소스를 분배했을 때의 실험 결과는 얼굴 검출율과 인식율의 결과가 영상에 등장하는 인원수에 관계없이 일정했으며, 결과 동영상의 fps 역시 꾸준히 25 fps 정도를 유지했다.

[0073]

이러한 GPU 리소스 분배 차이에 따른 결과는 도 17의 그래프로 보면 더욱 확실한 차이를 보인다.

표 2

GPU 리소스 분배를 한 경우의 인식을 결과

측정 인원수	얼굴 검출률	얼굴 인식률	초당 프레임 수 (fps)
1	97%	90%	26 fps
2	97%	89%	25 fps
3	96%	89%	25 fps
4	95%	88%	25 fps
5	95%	88%	25 fps

[0074]

[0075]

[표 3]은 GPU 리소스 분배 전의 GPU 점유율을, [표 4]는 GPU 리소스 분배 후의 GPU 점유율을 나타내고 있다. 도 18은 GPU 리소스 분배 차이에 따른 GPU 점유율 상태를 그래프로 도시하였다.

표 3

GPU 리소스 분배 전의 GPU 점유율

측정 인원수	#1	#2	#3
1	32%	0%	0%
2	65%	0%	0%
3	97%	0%	0%
4	100%	0%	0%
5	100%	0%	0%

[0076]

표 4

GPU 리소스 분배 후의 GPU 점유율

인원수 \ GPU	#1	#2	#3
1	32%	0%	0%
2	31%	33%	0%
3	28%	32%	30%
4	26%	35%	32%
5	31%	32%	36%

[0077]

[0078]

GPU 리소스를 분배하지 않은 경우, 1번 GPU의 사용량이 순식간에 100%까지 올라가면서 영상이 출력되지 않는 상황이 발생하였고, 그로 인하여 더 이상의 실험이 불가능했다. 실험 도중 GPU 2, 3번은 아예 로드조차 되지 않은 것 역시 확인 할 수 있다.

[0079]

GPU 리소스를 분배한 경우는, 1, 2, 3번의 GPU 리소스가 고르게 사용되는 것을 확인 할 수 있다. 등장 인원수가 많아질수록 연산량이 증가하며, 연산량이 증가할수록 GPU 리소스 분배 여부에 따라 점유율 및 실험 결과의 차이가 크게 벌어지는 것을 볼 수 있다. 따라서 본 발명에서 제안한 쿼리를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템의 효율성이 입증되었다.

표 5

걸음 속도별 추적 정확도

걸음 속도	정지상태		느은 걸음 (1.0m/sec)		보통 걸음 (1.3m/sec)		빠른 걸음 (1.6m/sec)	
	정면	측면(1~45°)	정면	측면(1~45°)	정면	측면(1~45°)	정면	측면(1~45°)
인식률	98.2	97.3	94.6	91.7	93.4	89.1	93.1	89

[0080]

[0081]

아울러, [표 5]에서 나타난 바와 같이 정지상태에서는 정면인 경우 약 98.2%, 측면인 경우 약 97.3%, 느은 걸음 속도(1.0m/sec)에는 정면인 경우 약 94.6%, 측면인 경우 약 91.7%, 보통 걸음 속도(1.3m/sec)에는 정면인 경우 약 93.4%, 측면인 경우 약 89.1%, 빠른 걸음 속도(1.6m/sec)에는 정면인 경우 약 93.1%, 측면인 경우 약 89%의 평균 인식률을 확인하였다. 측면은 1초 내지 45초 회전 범위 내에서 측정하였다.

[0082]

이하, 도 19를 참조하여 본 발명의 일 실시예에 따른 쿼리를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 방법에 대해 살펴보면 아래와 같다.

[0083]

먼저, 제어부(200)가 다수의 GPU 리소스를 분배한다(S100).

[0084]

그리고, 디스플레이부(300)가 영상처리부(100)가 입력받은 영상과, 영상으로부터 검출한 얼굴 객체 및 트래킹을 출력한다(S200).

[0085]

이하, 도 20을 참조하여 본 발명의 일 실시예에 따른 쿼리를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 방법의 제S100단계의 세부과정에 대해 살펴보면 아래와 같다.

[0086]

먼저, 제어부(200)가 GPU 각각의 메모리 사용량과 활성화된 코어의 수를 검출하여 각 GPU가 현재 어느 정도 사용되고 있는지를 파악한다(S102).

[0087]

이어서, 제어부(200)가 각 GPU의 현재 리소스 사용 상태에 따라 가장 점유율이 낮은 GPU를 스레드에 할당한다(S104).

- [0088] 뒤이어, 제어부(200)가 GPU 점유율이 기 설정된 수준 이상으로 높아지는지 여부를 판단한다(S106).
- [0089] 제S106단계의 판단결과, GPU 점유율이 기 설정된 수준 이상으로 높아지는 경우, 제어부(200)가 상기 제S102 단계로 절차를 이행한다(S108).
- [0090] 이하, 도 21을 참조하여 본 발명의 일 실시예에 따른 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 방법의 제S200단계의 세부과정에 대해 살펴보면 아래와 같다.
- [0091] 제S100단계 이후, 영상처리부(100)가 입력된 영상으로부터 그리드 방식으로 얼굴 객체를 검출한다(S202).
- [0092] 이어서, 영상처리부(100)가 검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는지 여부를 판단한다(S204).
- [0093] 제S204단계의 판단결과, 검출한 얼굴 객체가 찾고자 하는 타겟과 일치하는 경우, 영상처리부(100)가 일치하는 영상에 빨간색 박스 형태로 RoI(Region of Interest)를 지정한다(S206). 이때, 박스 위에 해당 얼굴 객체의 이름 정보를 표시할 수 있다.
- [0094] 뒤이어, 영상처리부(100)가 지정한 RoI에 대한 트래킹을 수행한다(S208).
- [0095] 이어서, 영상처리부(100)가 지정한 RoI에 대한 트래킹 중에 카메라(1) 화각의 중앙 지점에서 일정 범위 이상 벗어나는지 여부를 판단한다(S210).
- [0096] 제S210단계의 판단결과, 카메라(1) 화각의 중앙 지점에서 일정 범위 이상 벗어나는 경우, 영상처리부(100)가 PTZ(Pan Tilt Zoom) 제어를 통해 해당 RoI가 화각의 중심부에 위치하도록 카메라(1) 화각을 조정한다(S212).
- [0097] 이상으로 본 발명의 기술적 사상을 예시하기 위한 바람직한 실시예와 관련하여 설명하고 도시하였지만, 본 발명은 이와 같이 도시되고 설명된 그대로의 구성 및 작용에만 국한되는 것이 아니며, 기술적 사상의 범주를 일탈함이 없이 본 발명에 대해 다수의 변경 및 수정이 가능함을 당업자들은 잘 이해할 수 있을 것이다. 따라서 그러한 모든 적절한 변경 및 수정과 균등물들도 본 발명의 범위에 속하는 것으로 간주되어야 할 것이다.

부호의 설명

- [0098] S: 쿼터를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식과 트래킹 시스템

1: 카메라

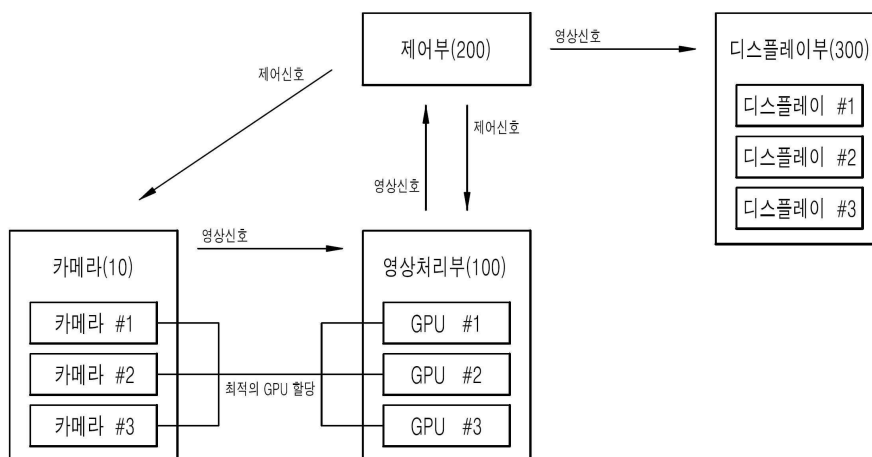
100: 영상처리부

200: 제어부

300: 디스플레이부

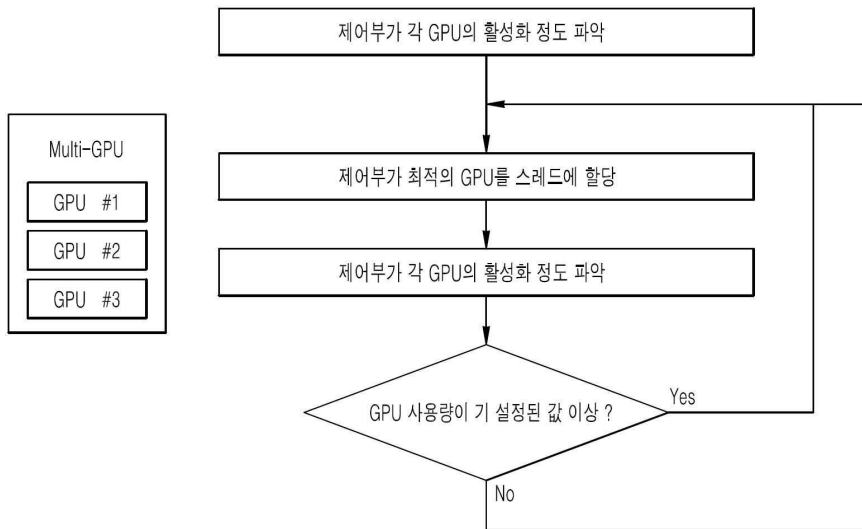
도면

도면1

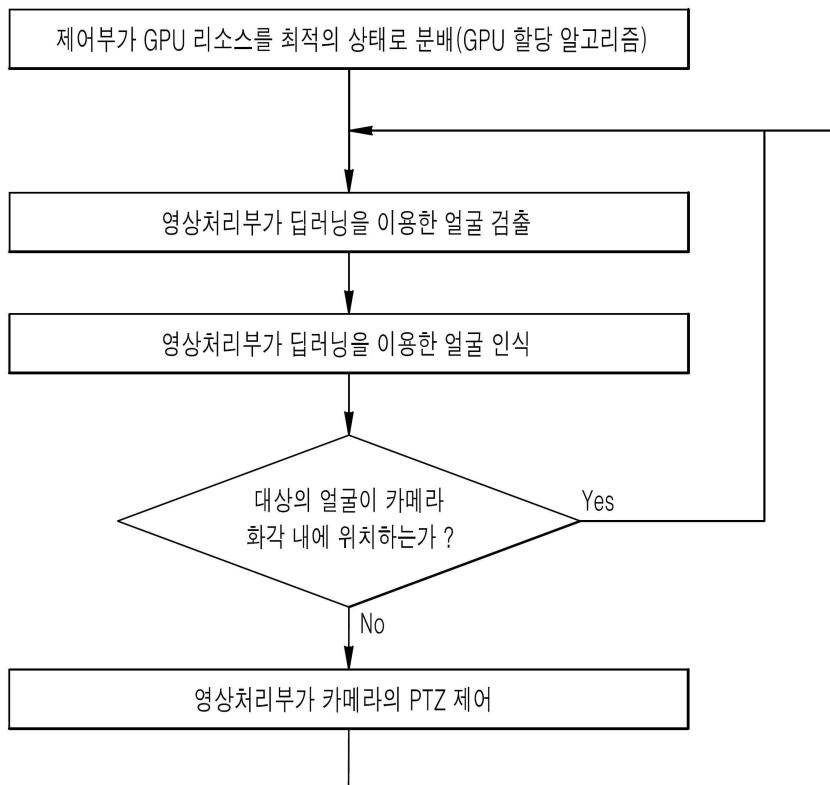


S

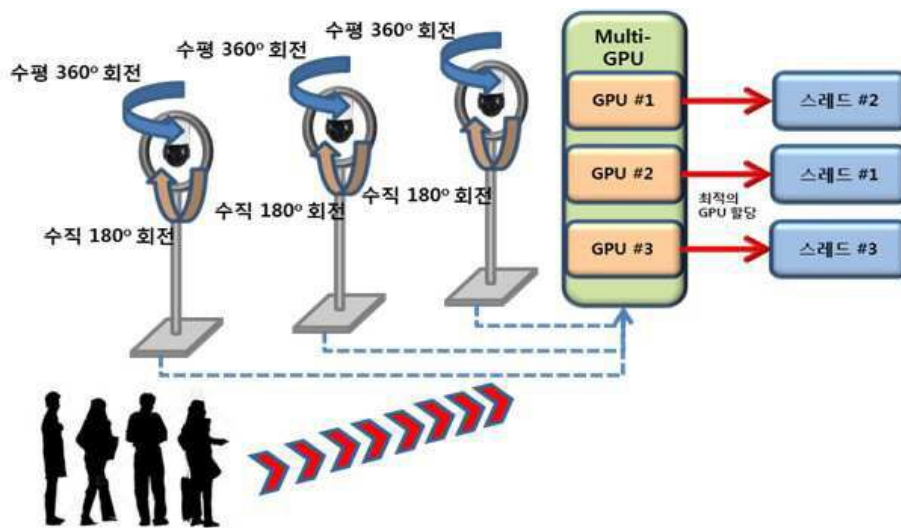
도면2



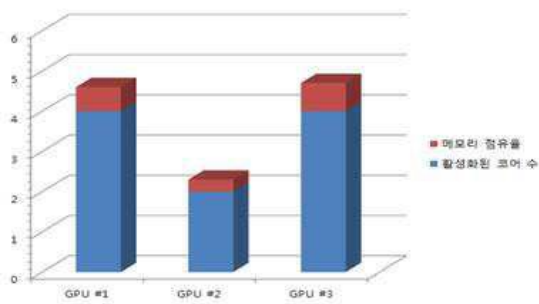
도면3



도면4



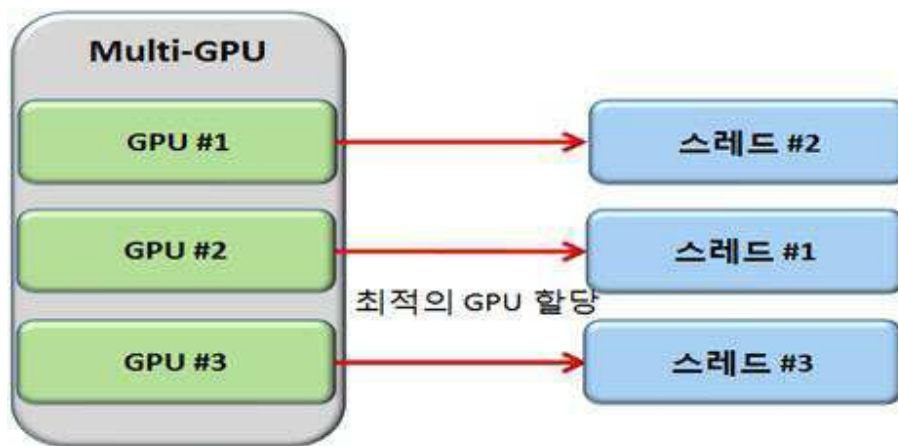
도면5



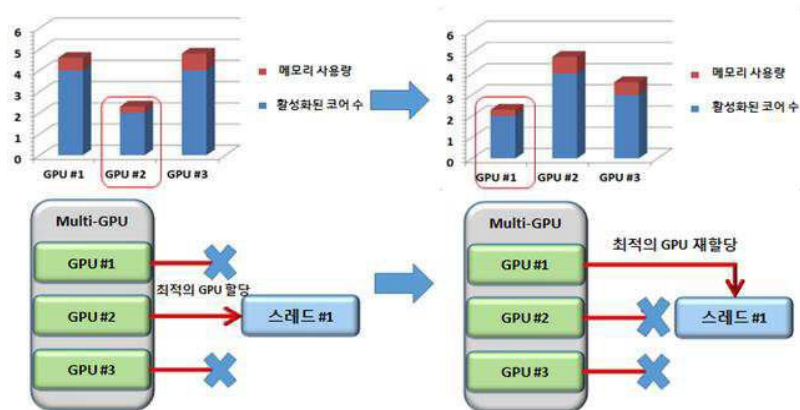
도면6



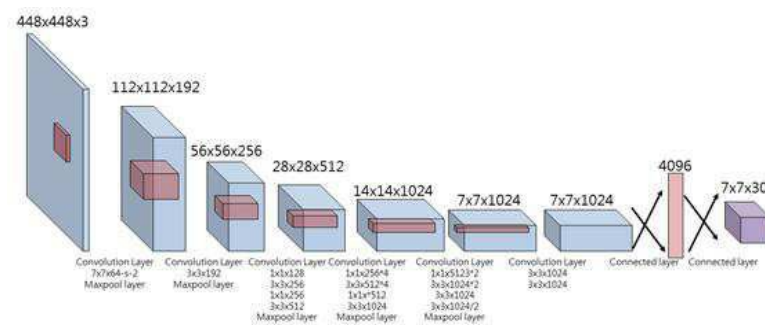
도면7



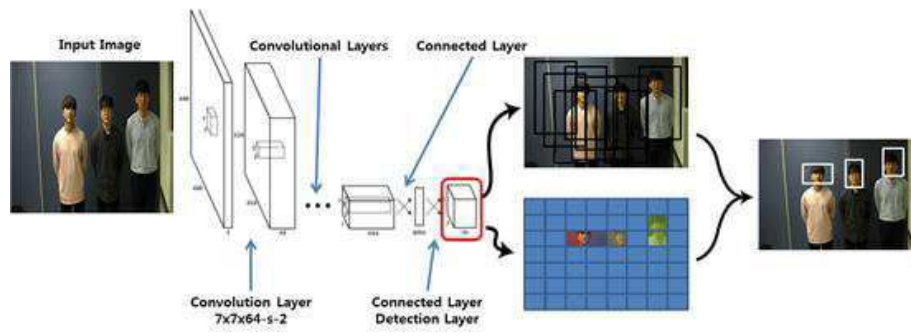
도면8



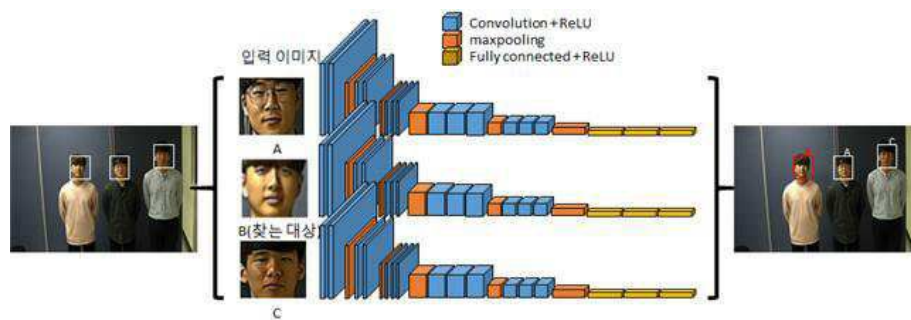
도면9



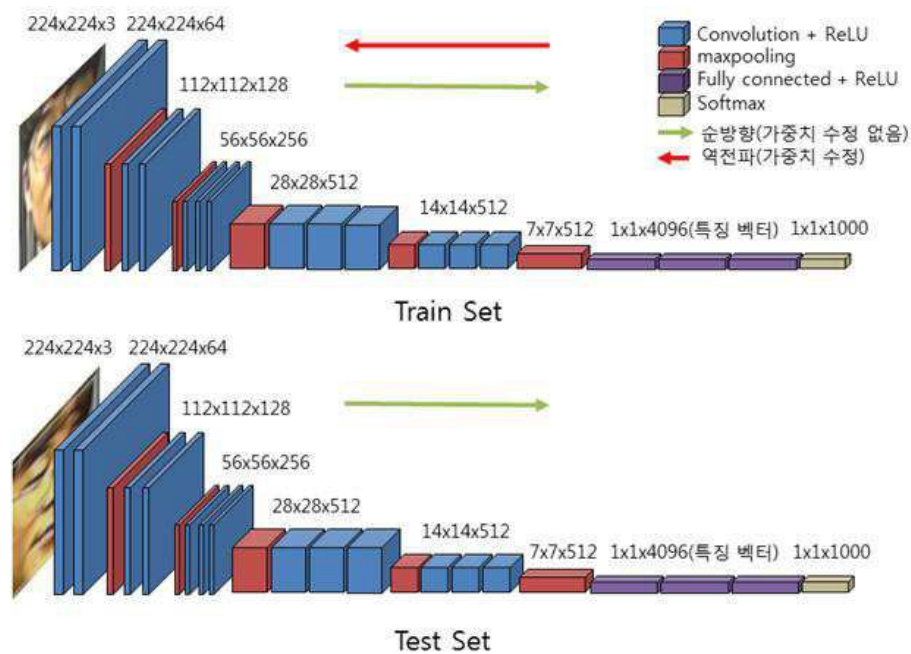
도면10



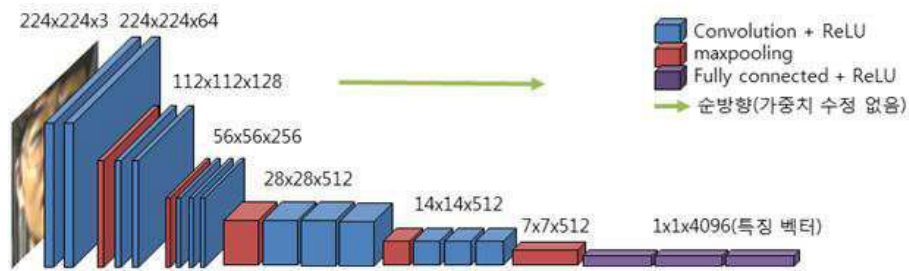
도면11



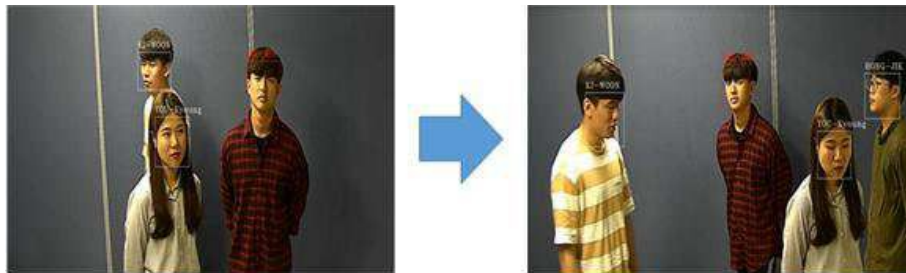
도면12



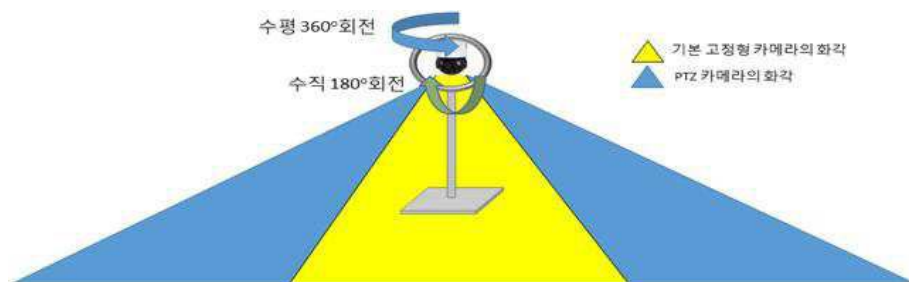
도면13



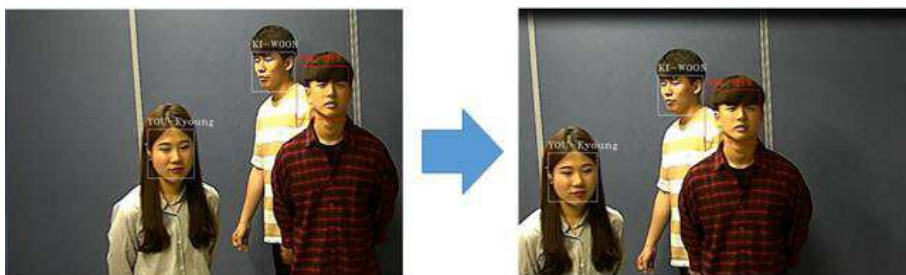
도면14



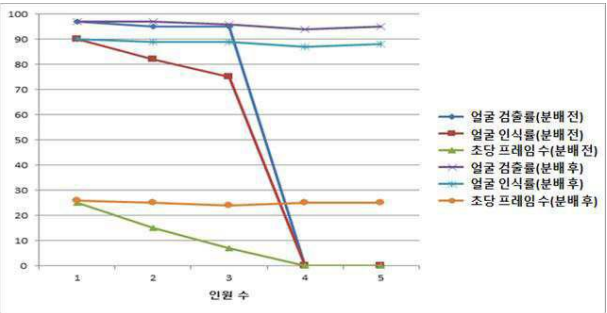
도면15



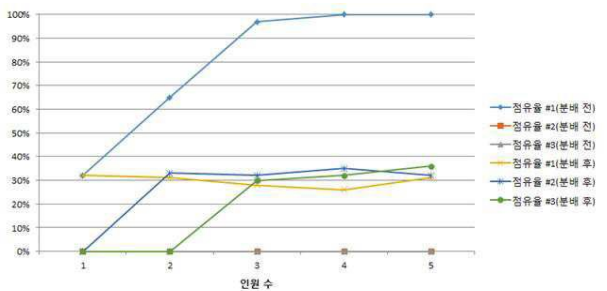
도면16



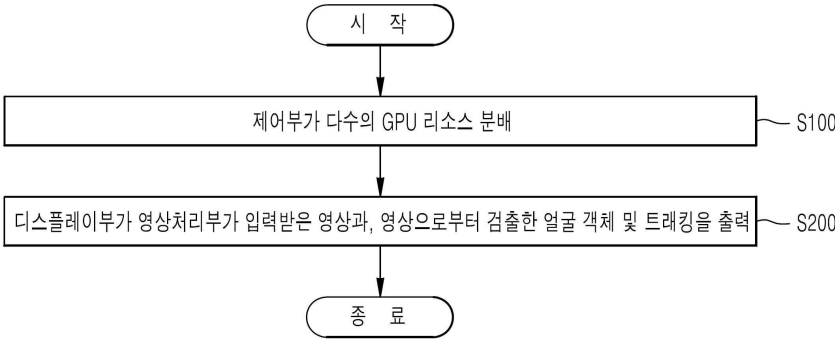
도면17



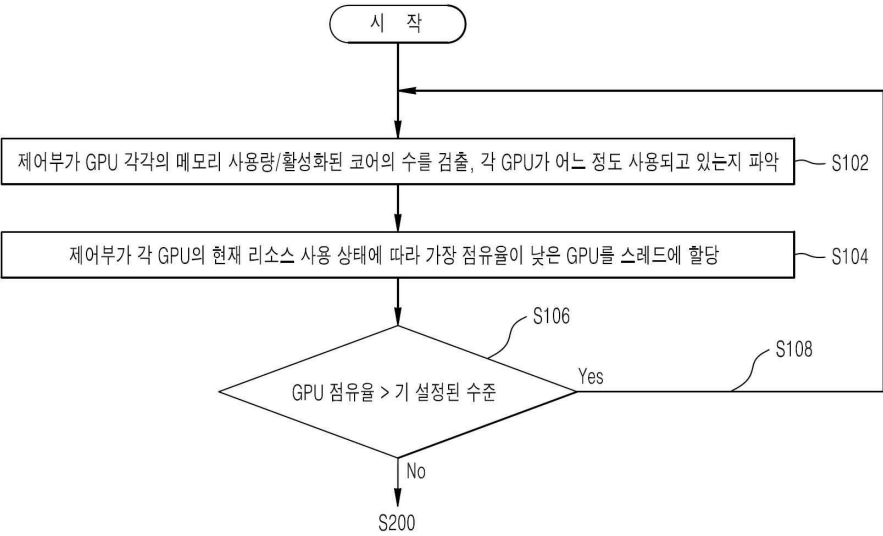
도면18



도면19



도면20



도면21

