ISSN:1226-7244 (Print)
ISSN:2288-243X (Online)
논문번호 18-02-08
j.inst.Korean.electr.electron.eng.Vol.22,No.2,281~288,June 2018
http://dx.doi.org/10.7471/ikeee.2018.22.2.281
55

쿠다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템

Intelligent Face Recognition and Tracking System to Distribute GPU Resources using CUDA

김 재 형*, 이 승 호*★

Jae-Heong Kim*, Seung-Ho Lee*★

Abstract

In this paper, we propose an intelligent face recognition and tracking system that distributes GPU resources using CUDA. The proposed system consists of five steps such as GPU allocation algorithm that distributes GPU resources in optimal state, face area detection and face recognition using deep learning, real time face tracking, and PTZ camera control. The GPU allocation algorithm that distributes multi-GPU resources optimally distributes the GPU resources flexibly according to the activation level of the GPU, unlike the method of allocating the GPU to the thread fixedly. Thus, there is a feature that enables stable and efficient use of multiple GPUs. In order to evaluate the performance of the proposed system, we compared the proposed system with the non - distributed system. As a result, the system which did not allocate the resource showed unstable operation, but the proposed system showed stable resource utilization because it was operated stably. Thus, the utility of the proposed system has been demonstrated

요 약

본 논문에서는 쿠다(CUDA)를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템을 제안한다. 제안한 시스템은 GPU 리소스를 최적의 상태로 분배하는 GPU 할당 알고리즘, 딥러닝을 이용한 얼굴 영역 검출, 딥러닝을 이용한 얼굴 인식, 실시간 얼굴 트래킹, PTZ 카메라 제어 등의 5단계로 구성되어진다. 멀티 GPU 리소스를 최적의 상태로 분배하는 GPU 할당 알고리즘은 고정적으로 스레드에 GPU를 할당하는 방식과 달리 GPU의 활성화 정도에 따라 유동적으로 GPU 리소스를 분배한다. 따라서 안정적이고 효율적인 멀티 GPU 사용을 가능하게 하는 특징이 있다. 제안된 시스템에 대한 성능을 평가하기 위하여 리소스 분배를 하지 않은 시스템과 제안한 시스템을 비교한 결과, 리소스를 분배하지 않은 시스템은 불안정한 동작을 보이는 반면에 제안한 시스템에서는 안정적으로 구동됨으로서 효율적인 리소스 사용을 보였다. 따라서 제안된 시스템의 효용성이 입증되었다.

Key words : Deep-learning, CUDA, GPU Resource Distribution, Face Detection, Face Recognition

* Dept. Electronics&Control Engineering, Hanbat National University
★ Corresponding author
E-mail: shlee@cad.hanbat.ac.kr, Tel: +82-42-821-1137
※ Acknowledgment
Manuscript received Jun. 9, 2018; revised Jun. 18, 2018; Accepted Jun. 19, 2018

Ⅰ. 서 론

각종 범죄로부터 시민을 안전하게 보호하기 위해, 실종 사건을 해결하기 위한 방법 등으로 CCTV는 매년 10만대가 증가하고 있는 추세다. 현재 중앙행정기관과 지방자치단체가 사용하는 CCTV만 해도 약 73만대로 집계되었다. 하지만 이 영상을 보고 관리하는 인원수는 턱없이 적어 일부 지역에서는 3교대 기준으로 한 사람이 273대의 CCTV를 모니터링 하고 있는 실정이다. 이는 과도한 업무량일뿐더러, 실질적으로 제대로 관제가 이루어지는지 조차 의심스러운 상황이다. 이를 해결하기 위해 CCTV 상에서 지능적으로 실시간 영상 안에서 얼굴을 인식하고 추적하는 시스템에 대한 관심이 증대되고 있다. 이런 지능형 관제 시스템을 구축하기 위해 과거에는 컴퓨터의 사양이 부족해 연산량을 줄이기 위한 알고리즘 개발[1]이 주된 목표 중 하나였다. 그러나 인식 성능의 향상을 추구하면 속도와 안정성이 저하되는 문제가 발생했다.

따라서 본 논문에서는 쿠다를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템을 제안한다. 기존의 시스템보다 높은 성능을 발휘하기 위하여 검출과 인식 과정 모두에 딥러닝 기법을 사용한다. 딥러닝 기법 사용에 따른 속도 및 안정성이 저하되는 것을 방지하기 위해 멀티 GPU를 이용한 병렬처리 기법을 사용하며, 각각의 GPU에 대한 활성화 정도를 파악해 GPU를 스레드에 할당하여 리소스를 최적의 상태로 분배한다.

Ⅱ. 본 론

1. 쿠다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템의 전체 구성도

본 논문에서 제안하는 쿠다를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템은 그림 1과 같은 구성도로 이루어져 있다. 고해상도 PTZ (Pan-Tilt-Zoom) 카메라 3대를 이용하여 실제 현장에서 다수의 사람들 중 찾고자 하는 대상 인물을 입력하면, 해당 인물을 검출 및 인식하여 PTZ 카메라의 Pan, Tilt 등의 제어를 통해 실시간으로 추적하여 3대의 모니터에 디스플레이 할 수 있는 시스템을 구축하는 것을 목표로 한다. 이때 3개의 스레드에 쿠다를 사용하여 GPU 리소스를 최적의 상태로 분배한다.

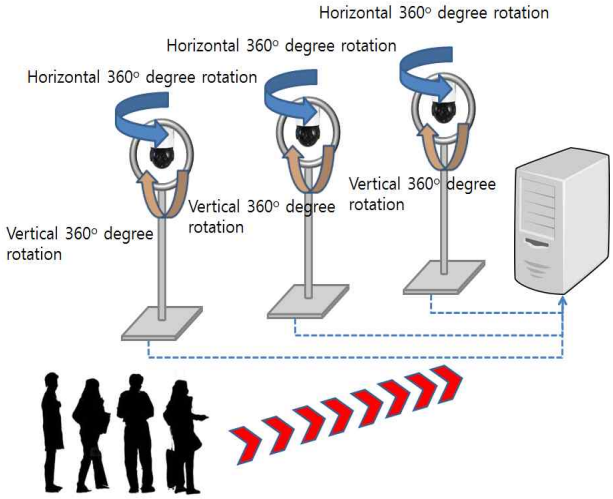


Fig 1. Configuration of Intelligent face recognition and tracking system to distribute GPU resources using CUDA

그림 1.쿠다를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템의 구성도

2. 전체 흐름도

본 논문의 전체 시스템 흐름도는 그림 2와 같다. 첫 번째로 쿠다를 사용하여 GPU의 활성화 정도를 파악해 3개의 스레드에 GPU 리소스를 최적의 상태로 분배한다. 두 번째로 PTZ 카메라를 통해 입력된 영상에서 딥러닝을 이용해 얼굴 영역을 검출하여 흰색 박스 형태로 표시한다. 세 번째로 검출된 얼굴 영역을 딥러닝을 이용해 사용자가 찾고자 하는 타겟과 동일 인물인지를 인식결과로 확인한다. 찾고자 하는 인물과 동일 인물이라면, 빨간색 박스 형태로 RoI(Region of Interest)를 지정하고 박스 위에 해당 인물의 이름 정보를 표시한다. 매 프레임마다 네 번째로 첫 번째에서 세 번째 과정을 반복하여 트래킹을 수행한다. 다섯 번째로 트래킹 도중에 카메라 화각의 중앙 지점에서 일정 범위 이상 벗어날 경우, PTZ 카메라 제어를 통해 해당 RoI가 다시 화각의 중심부에 올 수 있도록 카메라 화각을 조정한다.

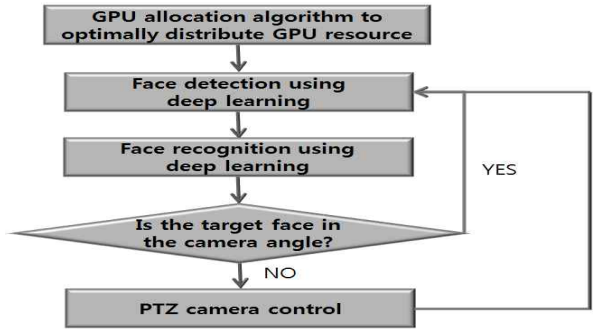


Fig 2. Flow chart of system
그림 2. 시스템 전체 흐름도

3. GPU 리소스를 최적의 상태로 분배하는 GPU 할당 알고리즘

멀티 GPU를 사용하는 과정에서 GPU 리소스를 효율적으로 할당하지 않을 경우에는 하나의 GPU에 과부하가 걸려 안정적인 시스템 구동에 문제가 발생하게 된다. 따라서 본 논문에서는 3개의 스레드에 쿠다를 사용하여 GPU 리소스를 최적의 상태로 분배하는 알고리즘을 제안한다. 제안하는 GPU 할당 알고리즘의 흐름도는 그림 3과 같다.

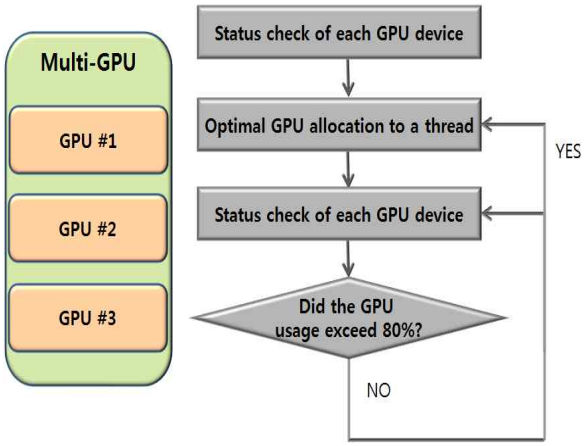


Fig 3. Flow chart of GPU allocation algorithm
그림 3. GPU 할당 알고리즘의 흐름도

GPU 할당 알고리즘은 첫 번째로 모든 GPU 디바이스의 활성화된 코어 수와 메모리 활성화 정도를 파악한다. 이 과정을 통해 각 GPU가 현재 어느 정도 사용되고 있는지를 파악하게 된다. GPU의 현재 리소스 사용 상태를 체크하는 과정을 그림 4에서 나타내고 있다.

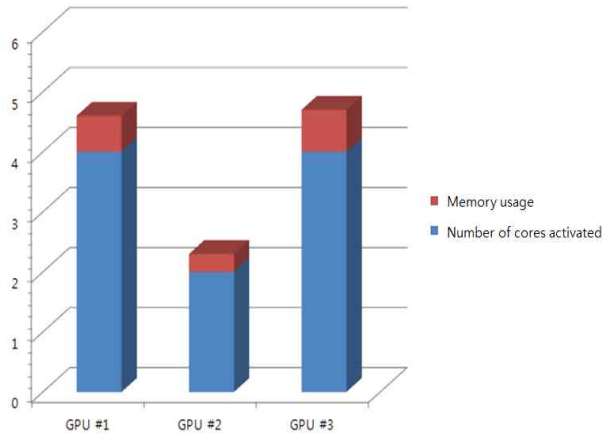


Fig 4. Status check of GPU resources usage
그림 4. GPU 리소스 사용의 상태 체크

두 번째로 각 GPU의 현재 리소스 사용 상태를 따라 가장 점유율이 낮은 GPU를 스레드에 할당한다. 이때 메모리 사용량과 활성화된 코어의 개수를 모두 고려하여 최적의 GPU를 할당한다. 그림 5에서는 스레드 #1에 대하여 최적의 GPU 할당하는 과정을 나타내고 있다. 리소스 점유율이 가장 낮은 GPU #2를 스레드 #1에 할당하게 된다.

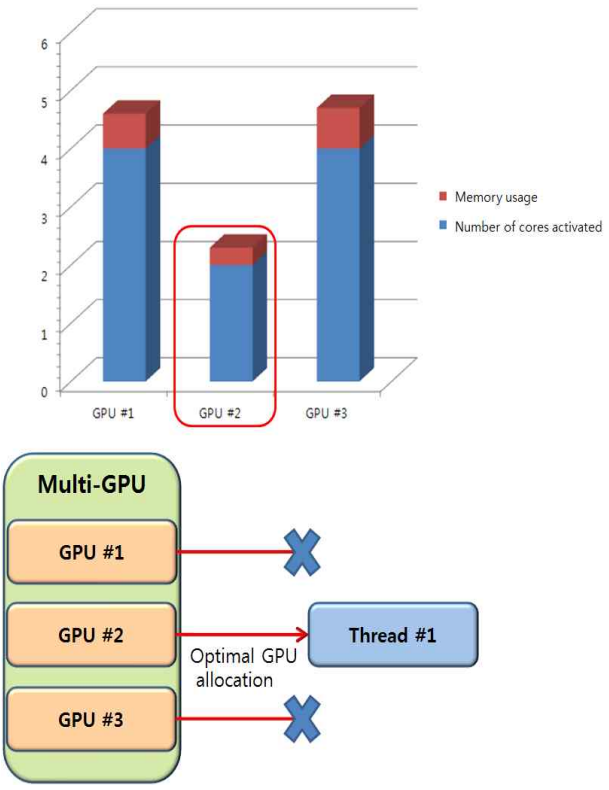


Fig 5. Optimal GPU allocation to a thread #1
그림 5. 스레드 #1에 대하여 최적의 GPU 할당

스레드 #2와 스레드 #3 들에 대해서도 첫 번째와 두 번째 과정을 수행하여 그림 6과 같이 모든 스레드에 대하여 최적의 GPU를 할당하게 된다.

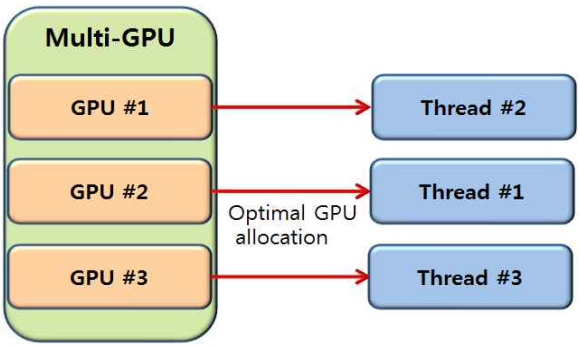


Fig 6. Optimal GPU allocation to all threads
그림 6. 모든 스레드에 대하여 최적의 GPU 할당

세 번째로 모든 스레드에 대하여 최적의 GPU를 할당한 후에는 첫 번째와 두 번째 과정을 매 프레임 반복시 오히려 GPU에 과부하가 발생될 우려가 있다. 따라서 GPU 점유율이 일정 수준 이상으로 높아질 경우에만 다시 첫 번째와 두 번째 과정을 수행하게 된다. 기준치를 높여 알고리즘을 실행할 경우에는 예상보다 빠르게 GPU 점유율이 높아져 GPU를 스위칭하기 전에 과부하가 걸릴 우려가 있다. 반대로 기준치를 낮추는 경우에는 GPU 스위칭이 너무 빈번하게 발생해 오히려 성능이 저하되는 현상이 발생하는 문제가 있다. 따라서 다시 수행하는 기준인 GPU 점유율 80%는 실험에 의해 정해졌다. 그림 7에서는 스레드에 대하여 GPU #2를 재할당하는 과정을 나타내고 있다. 리소스 점유율이 가장 낮은 GPU #2를 스레드 #1에 할당하였으나, GPU #2의 점유율이 다시 수행하는 기준인 GPU 점유율 80%를 오버하였으므로 리소스 점유율이 가장 낮은 GPU #1을 스레드 #1에 재할당하게 된다.

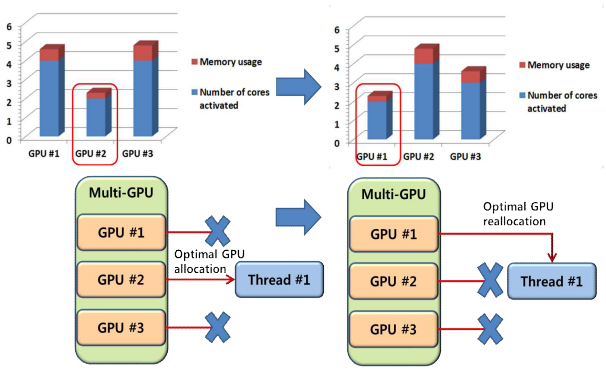


Fig 7. Process of reallocation GPU # 2 to a thread #1
그림 7. 스레드 #1에 대하여 GPU #2를 재할당하는 과정

4. 딥러닝을 이용한 얼굴 영역 검출

기존 얼굴 검출 알고리즘으로 널리 쓰이는 Haar-like feature[2] 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. 따라서 본 논문에서는 이러한 단점을 극복하고자 얼굴 영역을 다양한 각도의 회전까지 딥러닝 라이브러리인 Darknet[3][4]을 사용하여 학습시킨다. 먼저 여러 장의 사진에서 얼굴 영역인 부분을 x, y 상대 좌표로 입력한 후에 각 사진에 x, y 좌표를 라벨링 하여 학습시킨다. 얼굴 영역이 다수 검출되는 사진 역시 x, y 상대 좌표를 여러 개 입력한 후에 한 장의 이미지에서 동시에 여러 개의 얼굴이 검출 될 수 있도록 학습시킨다. 학습된 데이터들은 실시간 영상에서 얼굴 검출을 위해 사용된다. PTZ 카메라를 통해 실시간으로 입력되는 영상 내에서 얼굴이라고 판단되는 부분은 흰색 박스 형태로 표시한 후에 다음 단계인 얼굴 인식을 위해 잘라내어 임시로 저장한다. 그림 5는 Darknet을 사용한 얼굴 영역을 검출하는 딥러닝 수행 과정이고, 가장 오른쪽의 이미지가 얼굴영역이 검출된 결과를 나타낸다

.

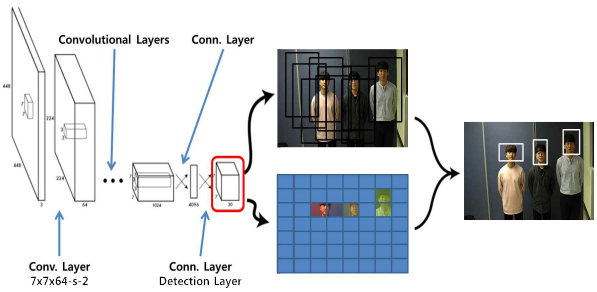


Fig 8. Deep learning process using Darknet
그림 8. Darknet을 사용한 딥러닝 수행과정

5. 딥러닝을 이용한 얼굴 인식

본 논문에서는 얼굴 인식을 위해 공개 딥러닝 라이브러리 중 Caffe[5]를 사용하였다. 첫 번째로

앞서 서술한 얼굴 검출 알고리즘을 이용해 사진이나 영상 등에 담긴 찾고자 하는 대상의 얼굴 영역을 검출하고 클래스를 분류해 저장 및 학습을 수행한다. 두 번째로 학습이 끝난 후에는 실시간으로 입력되는 영상에서 얼굴 검출 단계를 거쳐 임시로 저장된 얼굴 영역의 이미지를 받아와 사용자가 찾고자 하는 클래스와 일치하는 대상인지를 인식한다. 세 번째로 찾고자 하는 대상을 찾게 되면 빨간 박스 형태로 영상에 표시하고 박스 위에 대상의 이름을 표시한다. 그림 9는 인식이 완료된 결과를 보여주는 이미지이다.

Fig 9. Result image of face recognition
그림 9. 얼굴 인식의 결과 이미지

6. 실시간 얼굴 트래킹

기존의 트래킹 알고리즘으로는 템플릿 매칭[6]과 HOG(Histogram of Orented Gradient) 알고리즘[7] 등이 있다. 템플릿 매칭은 대상의 형태가 바뀔 경우 매칭이 잘 되지 않아 얼굴의 각도가 틀어질 경우 매칭에 실패할 확률이 높은 단점이 있다. HOG 알고리즘의 경우에는 전 프레임의 RoI 영역을 특성값을 추출해 사용하기 때문에 RoI 영역이 장애물에 가려져 영상 내에서 사라졌다 나타나는 경우에 취약하다는 단점이 있다. 따라서 본 논문에서는 실시간 얼굴 트래킹을 효과적으로 수행하기 위하여 앞서 서술한 2.3절에서 2.5절의 과정(GPU 할당, 얼굴 검출, 얼굴 인식)을 반복한다. GPU할당, 얼굴 검출, 얼굴 인식 과정을 반복하게 되면, RoI 영역이 사라졌다 나타날 경우나 카메라 화각 밖에서 안으로 들어올 경우에도 효율적으로 트래킹이 가능하게 된다. 그림 10은 실시간 얼굴 추적 결과 이미지이다.

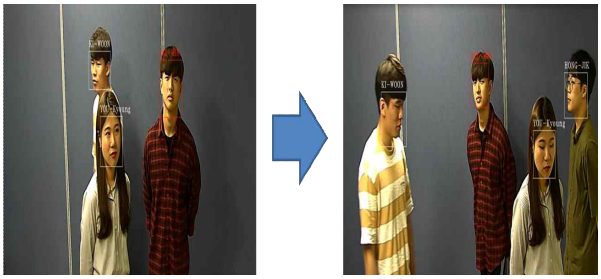


Fig 10. Result image of real-time face tracking
그림 10. 실시간 얼굴 추적의 결과 이미지

7. PTZ 카메라 제어

PTZ 카메라의 Pan-Tilt 제어는 웹 브라우저를 이용하여 카메라에 명령 신호를 보내 실시간으로 제어한다. 실시간으로 입력되는 영상에서 인식 결과에 부합하는 RoI 영역이 있을 경우, 해당 RoI 영역의 좌표와 영상의 중심 좌표를 비교해 일정 범위 이상 벗어나면 그림 11과 같이 카메라의 Pan-Tilt 모터를 제어한다. Pan-Tilt 모터를 제어하는 수식은 수식 (1)과 같다. 영상의 중심점과 RoI 영역의 중심점의 거리를 계산하여 Pan-Tilt 모터를 제어해 RoI 영역의 중심점이 영상의 중심점에 오도록 한다. 그림 12는 PTZ 카메라의 Pan-Tilt 모터를 제어한 결과를 나타내고 있다.

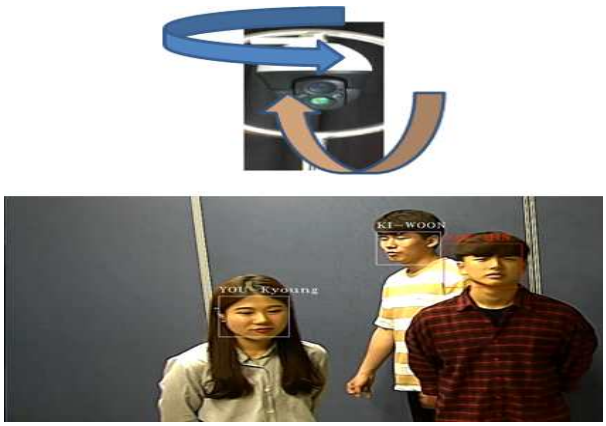


Fig 11. Image of Pan-Tilt control process
그림 11. Pan-Tilt 제어 과정의 이미지

$$x = x_{ori} - x_{roi} \qquad (1)$$
$$y = y_{ori} - y_{roi}$$

x_{ori}, y_{ori} = 원본 영상의 x, y 좌표
x_{roi}, y_{roi} = RoI 영역의 x, y 좌표

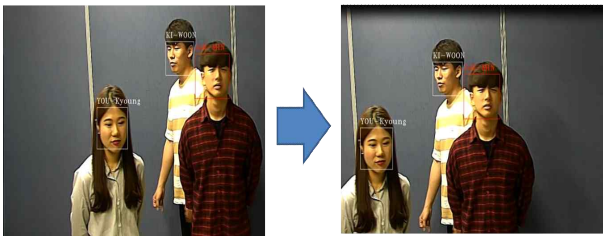


Fig 12. Image of Pan-Tilt control result
그림 12. Pan-Tilt 제어 결과의 이미지

8. 성능 실험

가. 실험 방법

쿠다를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템 구현에 사용된 시스템 하드웨어는 그림 13과 같이 PTZ 카메라 3대와 3개의 멀티 GPU를 사용한 고해상도 지원 컴퓨터, 모니터 3대로 구성되어 있다.

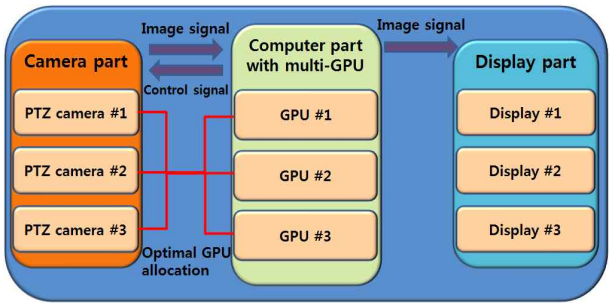


Fig 13. Hardware Diagram of Intelligent face recognition and tracking system using CUDA to distribute resources
그림 13. 쿠다를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템 하드웨어 블록도

나. 실험 결과

본 논문에서 제안한 쿠다를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템의 효율성을 측정하기 위하여, 성별, 나이, 체형이 각기 다른 150여 명의 데이터베이스를 구축하여 공개 딥러닝 라이브러리인 Darknet과 Caffe를 사용하여 학습시켰다. 학습이 완료 된 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 fps 수를 측정하고 GPU의 활성화 정도를 체크하였다. PTZ 카메라 원본 영상은 30 fps으로 입력되었다. 표 1에서 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 2에서 GPU 리소스를 분배했을 때의 실험 결과는 얼굴 검출율과 인식율의 결과가 영상에 등장하는 인원수에 관계없이 일정했으며, 결과 동영상의 fps 역시 꾸준히 25 fps 정도를 유지했다. 그림 14에 나타난 그래프로 보면 더욱 확실한 차이를 보인다. 영상에 등장하는 인원이 4명이었을 때의 결과가 약간 내려갔는데, 이는 처리 속도의 문제가 아닌 실험 영상의 조명 차이인 것으로 판단된다. 표 3은 GPU 리소스 분배 전의 GPU 점유율을, 표 4는 GPU 리소스 분배 후의 GPU 점유율을 나타내고 있다. 그림 15에서는 GPU 리소스 분배 차이에 따른 GPU 점유율 상태를 그래프로 나타내고 있다. 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. 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 논문에서 제안한 쿠다를 사용하여 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템의 효율성이 입증되었다.

Table 1. Result of GPU resource non-distribution
표 1. GPU 리소스 분배를 하지 않았을 경우의 결과

Measurement / Number of peoples	Face Detection Rate	Face Recognition Rate	fps of Video Results
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

Table 2. Result of GPU resource distribution
표 2. GPU 리소스 분배를 한 경우의 결과

Measurement / Number of peoples	Face Detection Rate	Face Recognition Rate	fps of Video Results
1	97%	90%	26 fps
2	97%	89%	25 fps
3	96%	89%	24 fps
4	94%	87%	25 fps
5	95%	88%	25 fps

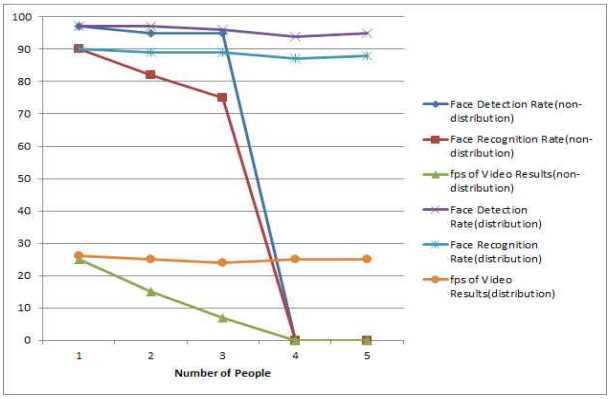


Fig 14. Result graph of GPU Resource Distribution Difference
그림 14. GPU 리소스 분배 차이에 따른 결과 그래프

Table 3. GPU usage before GPU resource distribution
표 3. GPU 리소스 분배 전의 GPU 점유율

GPU / Number of Peoples	#1	#2	#3
1	32%	0%	0%
2	65%	0%	0%
3	97%	0%	0%
4	100%	0%	0%
5	100%	0%	0%

Table 4. GPU usage after GPU resource distribution
표 4. GPU 리소스 분배 후의 GPU 점유율

GPU / Number of Peoples	#1	#2	#3
1	32%	0%	0%
2	31%	33%	0%
3	28%	32%	30%
4	26%	35%	32%
5	31%	32%	36%

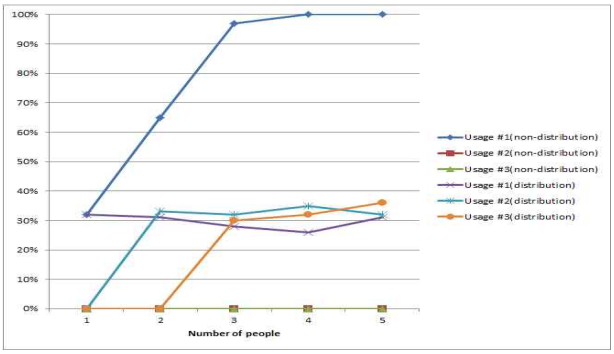


Fig 15. Graph of GPU usage
그림 15. GPU 점유율의 그래프

Ⅲ. 결론

본 논문에서는 딥러닝을 사용하고 GPU 리소스를 효율적으로 분배하는 얼굴 인식 및 트래킹 시스템을 구축하였다. 제안한 시스템의 멀티GPU 리소스를 최적의 상태로 분배하는 GPU 할당 알고리즘은 고정적으로 스레드에 GPU를 할당하는 방식과 달리 GPU의 활성화 정도에 따라 유동적으로 GPU 리소스를 분배함으로써 안정적이고 효율적인 멀티 GPU 사용이 가능한 특징이 있다. 구축된 시스템에 대한 실험 결과 GPU 리소스를 분배했을 때의 실험 결과는 얼굴 검출율과 인식율의 결과가 영상에 등장하는 인원수에 관계없이 일정했으며, 결과 동영상의 fps 역시 꾸준히 25 fps 정도를 유지하였다. 따라서 본 논문에서 제안한 쿠다를 사용하여 GPU 리소스를 분배하는 지능형 얼굴 인식 및 트래킹 시스템의 효율성이 입증되었다.

향후 연구 방향은 360° 카메라 영상[8]에서의 시스템 적용 방식에 대한 연구와 얼굴의 측면부에 대한 학습을 강화하는 방법에 대한 연구가 필요하다고 사료된다.

References

[1] Hee-Yeol Lee and Seung-Ho Lee, "A Study On Memory Optimization for Applying Deep Learning to PC," *Journal of IKEEE,* Vol. 21, No. 2, pp. 136~141, 2017.DOI:10.7471/ikeee.2017.21.2.136

[2] Lienhart, Rainer, and Jochen Maydt. "An extended set of haar-like features for rapid object detection." Image Processing. 2002. *Proceedings.*

2002 International Conference on. Vol. 1. IEEE, 2002.DOI: 10.1109/ICIP.2002.1038171

[3] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016.

[4] Redmon, Joseph. "Darknet: Open source neural networks in c." Pjreddie. com.[Online]. Available: https://pjreddie. com/darknet/.[Accessed: 21-Jun-2017] (2016).

[5] Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." *Proceedings of the 22nd ACM international conference on Multimedia.* ACM, 2014. DOI:10.1145/2647868.2654889

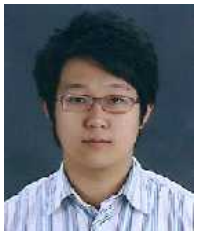
[6] Lewis, John P. "Fast template matching." *Vision interface.* Vol. 95. No. 120123. 1995.

[7] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* Vol. 1. IEEE, 2005. DOI: 10.1109/CVPR.2005.177

[8] Hee-Yeol Lee, Sun-Gu Lee and Seung-Ho Lee, "Development of 360° Omnidirectional IP Camera with High Resolution of 12Million Pixels," *Journal of IKEEE,* Vol. 21, No. 3, pp. 268~271, 2017 DOI:10.7471/ikeee.2017.21.3.268

BIOGRAPHY

Jae-Heong Kim (Member)



2010 ~ current : BS degree course of Electronics&Control Engineering, Hanbat National University

Seung-Ho Lee (Member)

1986 : BS degree in Electronic Engineering, Hanyang University

1989 : MS degree in Electronic Engineering, Hanyang University

1994 : Ph .D degree Electronic Engineering, Hanyang University

1994 ~ current : Professor, Department of Electronics&Control Engineering, HanbatNational University