

# 딥러닝의 다수 입력 이미지 학습 및 추론 효율 향상을 위해 추가적인 처리 프로세스 연구

최동규 · 김민영 · 장종욱\*

동의대학교

## A Study on Additional Processing Processes for Learning Multiple-input Images and Improving Inference Efficiency in Deep Learning

Donggyu Choi · Minyoung Kim · Jongwook Jang\*

Dong-eui University

E-mail : dgchoi@office.deu.ac.kr / kmyco@deu.ac.kr / jwjang@deu.ac.kr

### 요 약

실생활에는 많은 카메라가 활용되고 있으며 단순한 추억을 위한 사진 촬영을 넘어서 문제 상황을 확인하기 위하여 감시, 방범을 위하여 많이 사용되고 있다. 이러한 감시와 방범은 일반적인 형태로 단순한 저장으로만 사용되고 있으며, 다수의 카메라를 활용하는 시스템에서는 추가 기능을 활용하는 것은 하드웨어의 추가적인 사양을 요구하게 된다. 본 논문에서는 일반적인 이미지 처리에서 벗어난 객체 감지 시스템을 수행하는 하나의 하드웨어 또는 서버에서 입력된 여러 개의 이미지 입력 처리하기 위해 이미지 입력 방법과 객체 감지 이후 처리 프로세스를 추가한다. 방법의 수행은 딥러닝을 수행하는 하드웨어의 학습과 추론에 모두 활용해 보며 개선된 이미지 처리 프로세스를 수행할 수 있도록 한다.

### ABSTRACT

Many cameras are used in real life, and they are often used for monitoring and crime prevention to check the situation of problems beyond just taking pictures for memories. Such surveillance and prevention are generally used only for simple storage, and in systems utilizing multiple cameras, utilizing additional features would require additional hardware specifications. In this paper, we add image input methods and post-object processing processes to process multiple image inputs from one hardware or server that perform object detection systems that deviate from typical image processing. The performance of the method is utilized in both learning and reasoning of the hardware performing deep learning, and allows improved image processing processes to be performed.

### 키워드

Deep learning, Image processing, Post-processing, Pre-processing, Object detection

### 1. 서 론

인공지능은 지금까지의 발전에 이르기 전, 하드웨어 성능으로 인하여 크게 발전하지 못했던 분야다. 자율적으로 단순한 작업을 수행하기 위한 기기 개발에는 큰 문제가 없었으나 복잡한 연산을 수행하기 위해서는 다량의 데이터를 처리해야 하며 이를 받쳐줄 성능의 하드웨어가 필요로 하였다. 물

론, 시간이 지나며 점차 하드웨어가 발전했으나 이전과 같은 관심도 부족으로 인공지능에 대한 새로운 알고리즘이나 처리 방법이 개발되지 못하였다. 그러던 도중 2016년 이세돌과 알파고의 바둑 대국으로 인공지능의 새로운 장을 열게 되었다[1].

알파고는 이세돌과의 대국 전에 약 3천만 개의 바둑판 상황을 학습하게 되었다. 이는 기존에 제기되었던 문제인 다량의 데이터를 수용 및 연산할 수 있는 하드웨어에 대한 이슈를 한 번에 타파하였다. 하지만, 알파고가 해결하지 못한 문제점이

\* corresponding author

있었다.

알파고가 다량의 데이터를 학습하기 위해 온라인을 통하여 사용에 동의한 다른 컴퓨터들의 리소스를 공유하는 방법을 택했다. 한 개의 하드웨어에서 다수의 입력 데이터를 한 번에 처리하기 쉽지 않기 때문이다. 이것은 문제를 해결하기 위한 좋은 방법이나, 단일 서버가 동작하는 시스템 또는, 보안을 요구하는 부분에서는 사용하기 어렵다.

본 논문에서는 인공지능 활용에 있어서 다수 입력에 대한 하나의 하드웨어에서 수행할 수 있는 학습 및 처리 프로세스를 제안한다. 이미지의 딥러닝 학습 수행 시의 특징과 결과물 출력에서의 추가적인 프로세스를 도입하여 다수의 입력에 대해 효율적으로 처리 방법을 설명한다.

## II. 관련 연구

### 2.1 딥러닝의 병렬처리

딥러닝의 경우 병렬처리가 가능하다. 학습 및 처리에서 속도를 개선하기 위하여 수행하는 것인데 단일 그래픽카드에서 여러 개의 모델을 수행하는 것이 아닌 다수의 그래픽카드를 연결하여 한 개의 모델을 병렬처리하는 것을 의미한다[2,3].

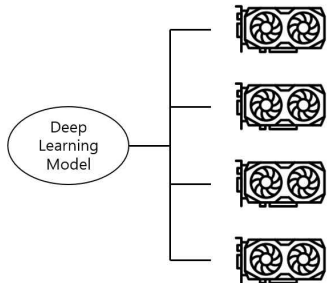


그림 1. Multi-GPU 사용 딥러닝 표현 이미지

그림 1는 딥러닝에서 1개의 모델에 대해 병렬처리를 수행하는 GPU를 표현한 것이다. 같은 1개의 인공지능 결과를 위한 처리를 수행할 때 분산하여 속도와 성능을 일부 개선할 수 있다. 기본적으로 입력 데이터를 다수로 받는 모델이 아니면 단일 입력 외에는 취약하다고 볼 수 있다.

### 2.2 딥러닝 사용 CCTV

실제로 딥러닝의 발전이 이루어짐에 따라 국가에서는 골목과 도로의 CCTV에 활용하기 위해서 다양한 연구과제를 진행하고 있다[3].

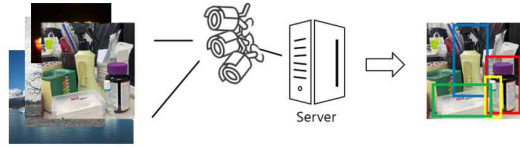


그림 2. Server의 객체 인식 수행 간략 이미지

그림 2은 카메라를 통해 서버로 입력 데이터를 전달하여 처리하는 일련의 과정을 나타내고 있다. 과정상에 문제가 없어 보이지만 서버는 앞서 설명한 것과 같이 단일 모델을 수행하기 위해서 정해진 입력만 받을 수 있다. CCTV에서 딥러닝 처리는 인력이 담당해야 하는 많은 부분을 보완할 수 있기에 필요하지만, 보통의 딥러닝 모델은 단일 입력에 대해 처리하기 때문에 다수의 입력을 처리하는 것이 일반적이지 않다.

## III. 설계 및 수행

보통의 딥러닝 수행 프로세스는 입력 이미지를 곧바로 결과 출력을 위해 반영시킨다.



그림 3. 제안하는 수행 방법 순서

그림 3은 제안하는 방법인 입력에서 다중이미지를 받아 단일이미지 크기와 같은 크기로 압축 및 이어 붙여 변환한 후에 검출된 이미지의 위치를 줄인 이미지 비율을 반영하여 원본 이미지에 나타내는 것을 보여준다. 위 방법은 OpenCV에 있는 간단한 함수를 활용해 수행할 수 있으며, 중요한 점은 원본 이미지를 입력 이후에 따로 저장해 놓아야 한다는 것이다. 압축하여 처리한 이후 결과물 출력에 위치 데이터만 가져와 원본에서 나타낸다.

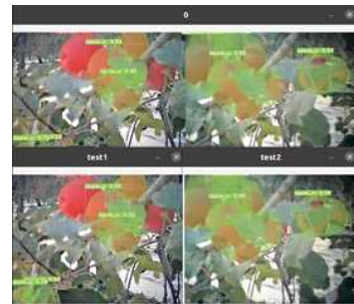


그림 4. 제안하는 방법 수행 이미지

실제로 해당 방법으로 수행 시 이미지를 이어 붙인 구간에는 간섭이 있었다. 추후 추가적인 연구를 통하여 프로세스 및 알고리즘을 도입해 간섭에서 생기는 객체를 제외시킬 수 있는 방법이 필요했다. 기본적인 기능인 객체 인식 자체는 전혀 문제가 없었으며, 결과물로 나온 객체 위치에 압축된 비율만 조정하면 원본에 도입될 좌표를 확인할 수 있었다.

#### IV. 결 론

인공지능은 이미 실생활의 범주로 많이 넘어왔다. 당장 사용하고 있는 스마트폰을 보아도 많은 기술이 들어가 있다. 그중에서 영상을 활용한 기술들은 다양하며 그 활용도가 높다. 단일 카메라 즉, 단일 데이터 입력을 통한 사용은 주차 시 번호판을 확인하거나 핸드폰 카메라로 찍은 이후 보정을 하는 등의 실사용이 많지만, 다수의 데이터를 실시간으로 한 번에 처리하는 것은 쉬운 일이 아니다.

본 논문에서는 아직 정확도 및 속도 면에서 부족하긴 하지만, 현재 범용으로 사용되는 실시간 객체 추적 딥러닝 모델을 활용하여 다수의 입력을 단일 하드웨어에서 수행하는 방안을 제시하였다. 기존에 CCTV 활용에서 단순 데이터 저장을 위하여 원본 이미지를 압축하는 경우는 많았지만, 처리를 위하여 임시 저장 및 추후 활용으로 이어지는 경우는 딥러닝 기술을 활용하는 특수 상황에서 가능한 일이기 때문에 저장의 목적으로는 효율적이지 않다. 하지만, 연산의 가중을 낮출 수 있으며 인공지능을 필요한 만큼만 사용하는 것은 소프트웨어 측면에서 매우 효율적인 일이다. 또한, 딥러닝 학습에 수행되는 Pooling의 작업에서는 이미 압축을 하고 있기에 학습 데이터를 한 번에 처리할 방법도 고안해 볼 수 있다.

#### Acknowledgement

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 Grand ICT연구센터지원사업의 연구결과로 수행되었음(IITP-2021-2016-0-00318). 추가적으로, 본 논문(저서)는 부산광역시 및 (재)부산인재평생교육진흥원의 BB21플러스 사업으로 지원된 연구임.

#### References

- [1] Lee Sedol lost in 186 moves. AlphaGo made an intended mistake and drew Lee Se-dol's carelessness. [Internet]. Available : <https://www.segye.com/newsView/20160309003374?O>

- utUrl=naver
- [2] Pytorch Multi-GPU Examples. [Internet]. Available : [https://pytorch.org/tutorials/beginner/former\\_torchies/parallelism\\_tutorial.html](https://pytorch.org/tutorials/beginner/former_torchies/parallelism_tutorial.html)
- [3] Y.M. Park, S.Y. Ahn, E.J. Lim, Y.S. Choi, Y.C. Woo, and W. Choi, "Deep Learning Model Parallelism", 2018 *Electronics and Telecommunications Trends*, Vol. 33, No. 4. pp. 1-13, Aug. 2018.
- [4] D.H. Shin, J.W. Baek, R.C. Park, and K.G. Chung, "Deep Learning-based Vehicle Anomaly Detection using Road CCTV Data", *Journal of the Korea Convergence Society*, Vol. 12, No. 5, pp. 1-6, Feb. 2021.