Real-Time Image Recognition Using Collaborative IoT Devices

Ramyad Hadidi, Jiashen Cao, Matthew Woodward,
Michael S. Ryoo, and Hyesoon Kim

in Proceedings of the 1st on Reproducible Quality-Efficient Systems Tournament
on Co-designing Pareto-efficient Deep Learning (ReQuEST '18),
ACM, New York, NY, USA.

Musical Chair

- IoT 시스템은 강력한 서버를 사용해 데이터로부터 정보를 추출
- 서버와의 통신으로 인해 가공되지 않은 민감 정보가 인터넷을 통해 전송
- 네트워크 부하, 서버 의존
- Solution: Musical Chair
  - IoT 네트워크 내 장치로부터 집계된 연산 능력을 모아 효율적이고 지역적이며 동적인 실시간 인식을 함

Musical Chair

- Musical Chair는 여러 장치에 DNN 모델을 분산 처리하기 위한 기법
- DNN 모델은 fc layer와 convolution layer에 연산이 집중
- Musical Chair는 두 종류의 레이어의 연산을 분배함으로써
  연산 비용을 줄여 리소스의 한계를 극복

분산 처리 방식

- Model Parallelism
  - 주어진 레이어나 레이어 그룹을 장치에 분배하는 방식
- Data Parallelism
  - 입력 데이터를 네트워크의 여러 장치에 보내는 방식
- Convolution Layer는 Data Parallelism을 활용할 때 이득
- FC Layer는 장치의 리소스 환경에 따라 다름

분산 처리 방식

- AlexNet

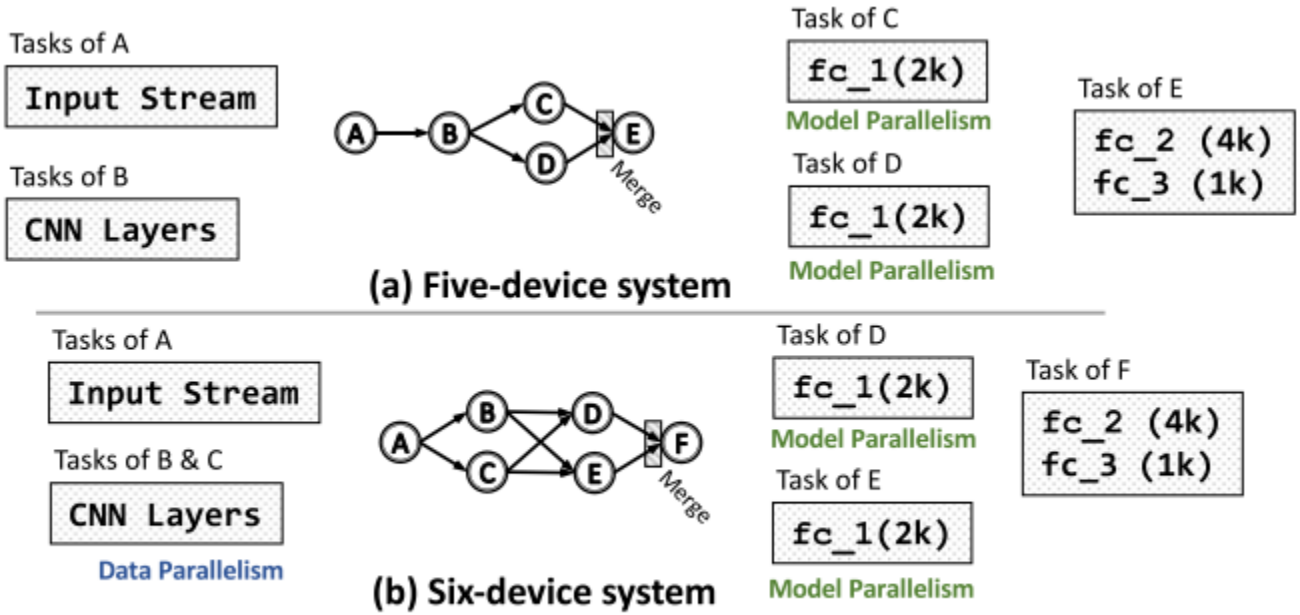


Figure 3. System architectures for AlexNet.

분산 처리 방식

- VGG16

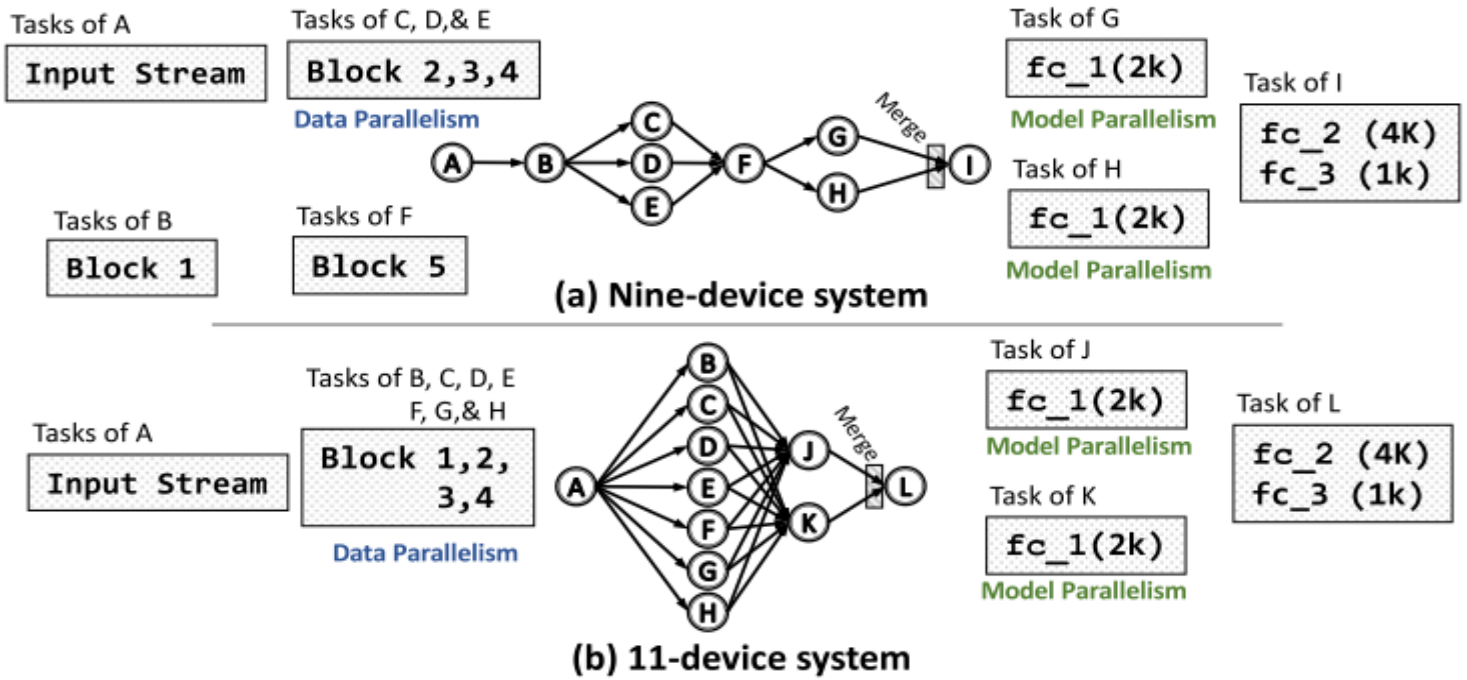


Figure 4. System architectures for VGG16.

성능 평가

- AlexNet
  - 6개의 장치를 사용했을 때 TX2 CPU와 비슷한 성능
  - TX2 GPU보다는 30% 낮은 성능

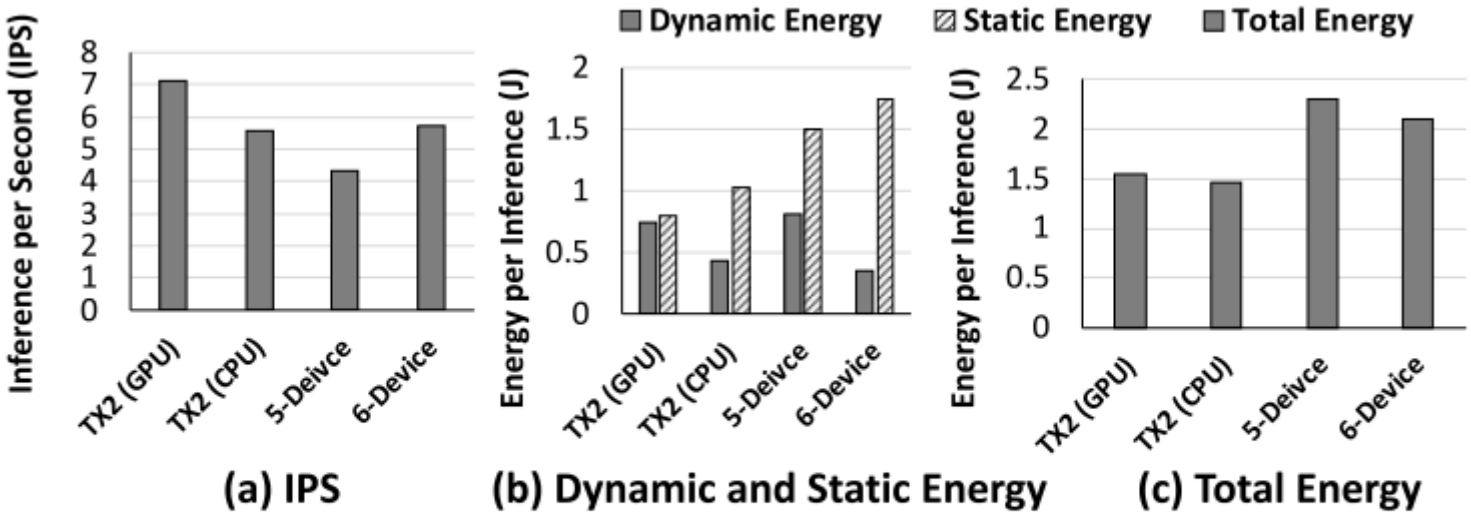


Figure 5. AlexNet: Measured IPS (a), static and dynamic energy consumption (b), and total energy consumption (c).

성능 평가

- VGG16
  - 더 많은 장치를 사용하여 CNN에 더 최적화된 data parallelism을 활용
  - 11개의 장치를 사용했을 때 TX2 GPU 대비 15% 낮은 성능

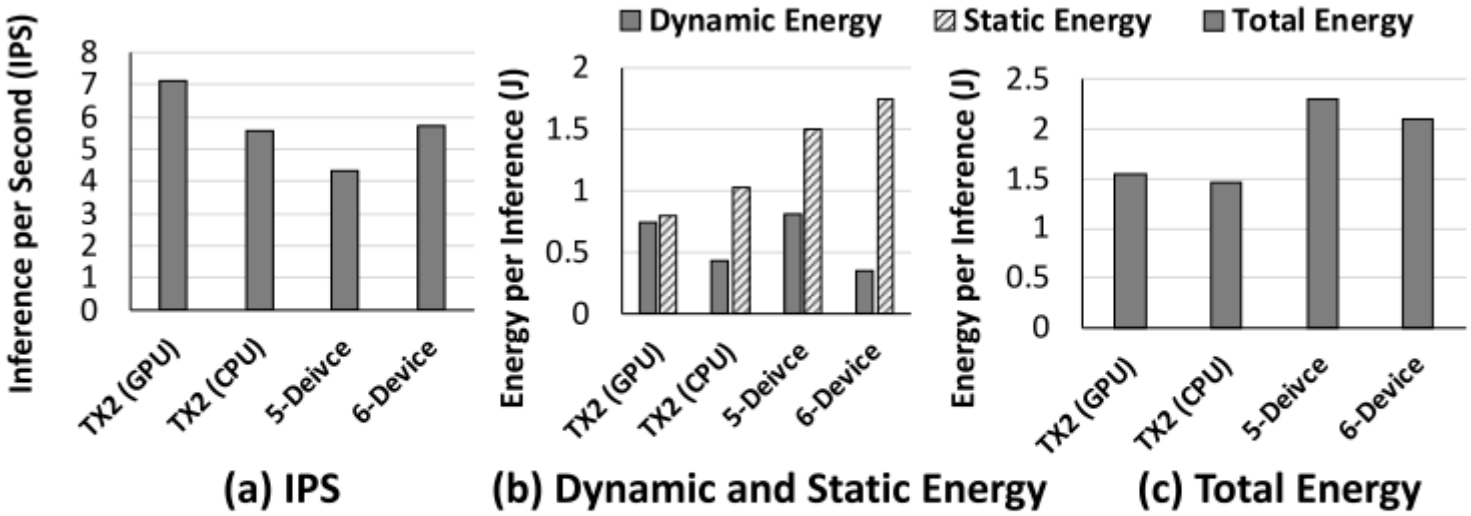


Figure 5. AlexNet: Measured IPS (a), static and dynamic energy consumption (b), and total energy consumption (c).

Musical Chair: Efficient Real-Time Recognition Using Collaborative IoT Devices

Ramyad Hadidi, Jiashen Cao, Matthew Woodward, Michael S. Ryoo, and Hyesoon Kim

ArXiv e-prints:1802.02138.

Musical Chair

- Musical Chair는 리소스가 제한된 저전력 IoT 장치의 공동 작업을 통해 비용 효율적인 실시간 동적 DNN 추론을 가능하게 함

- 입력 조건을 동적으로 조정하여 IoT 장치의 집단 연산을 통해 실시간 인식을 수행

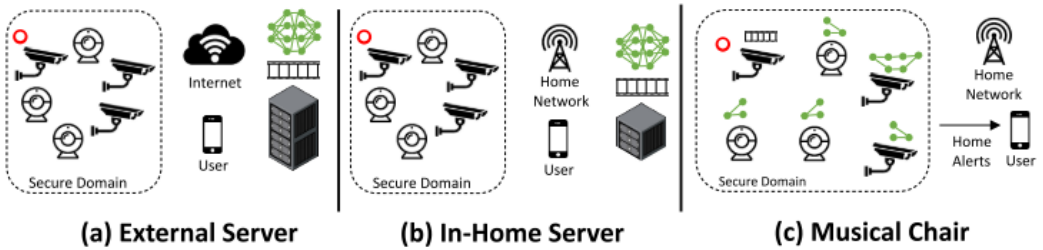


Figure 1: Three approaches for real-time DNN.

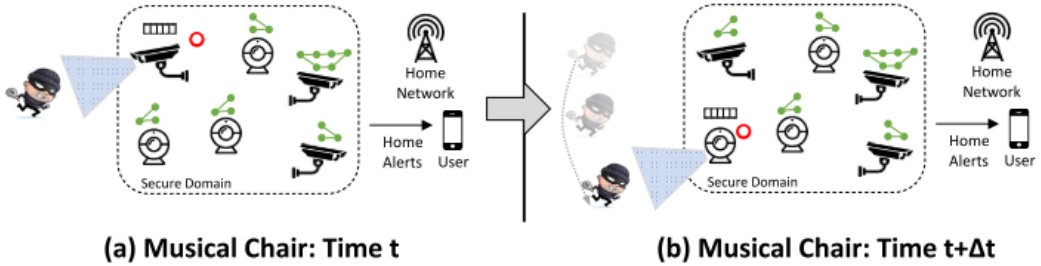
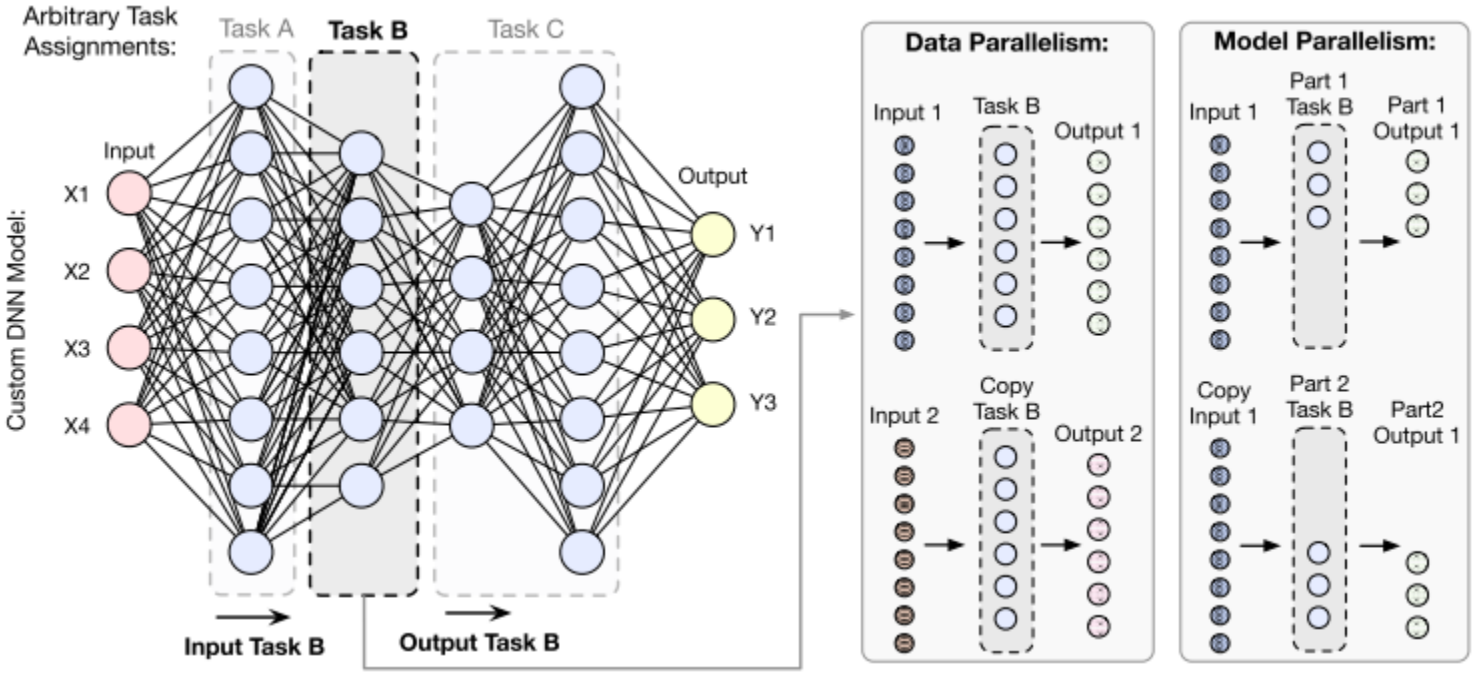


Figure 2: Musical Chair dynamically adjusts to input.

DNN 분산 처리

- Data Parallelism
- Model Parallelism

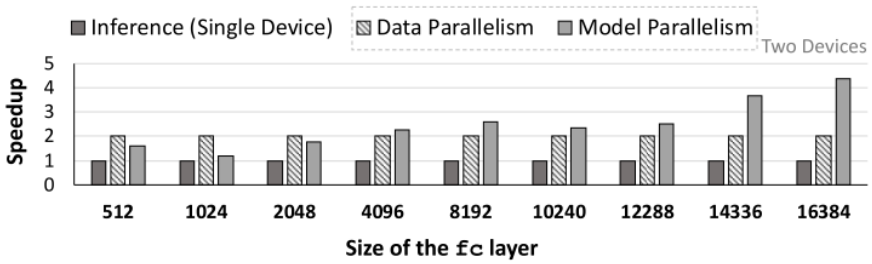


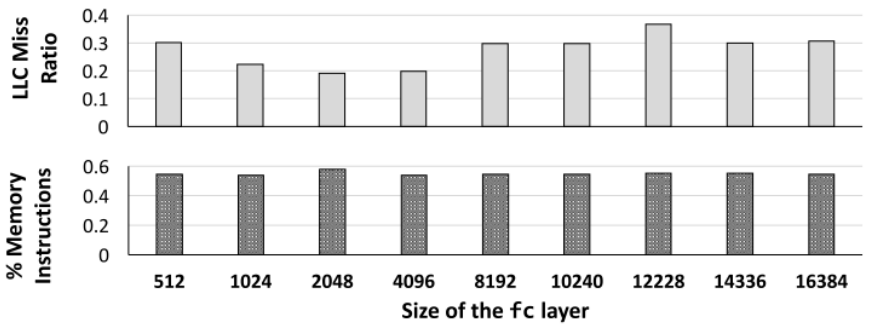
FC Layer

- 모든 입력 값을 각각의 장치에 복사해 각 하위 연산을 진행
- 출력 값을 다음 레이어의 입력으로 활용하기 위해 병합
- 동일한 모델을 사용하면서 장치 당 연산을 줄일 수 있음

FC Layer

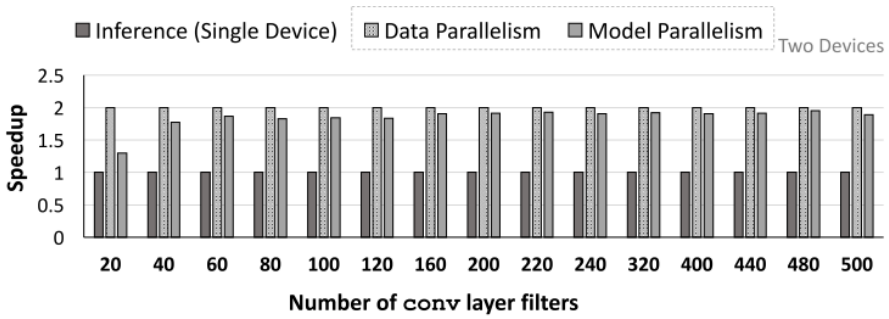
- 데이터 병렬 처리는 두 배의 성능 향상
- 모델 병렬 처리는 절반 크기의 fc layer의 성능에 따라 달라짐
- FC layer의 메모리 명령어 비율과 LLC 미스 비율이 높음
- FC layer의 크기가 10240 이상일 경우 메모리 스왑 공간을 사용
  - 모델 병렬 처리를 이용할 경우 스왑 공간 작업을 피할 수 있음

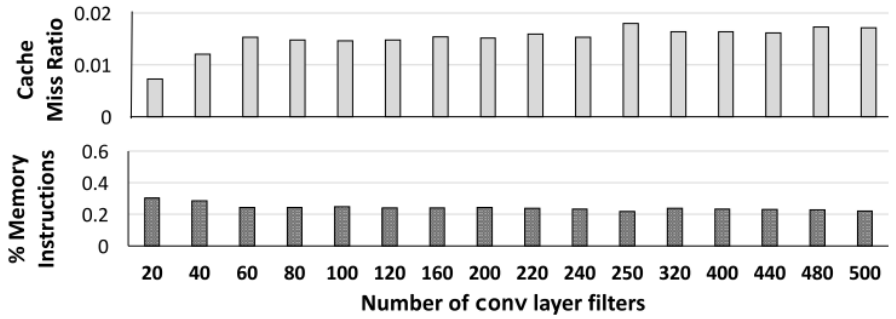




Convolution Layer

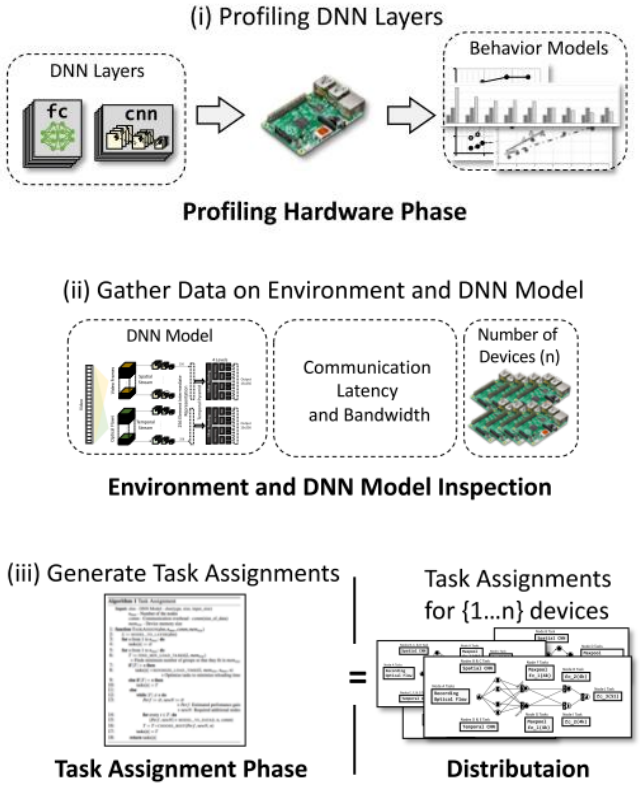
- 일반적으로 입력보다 필터의 수가 크므로
  필터를 분산시켜 병렬 처리를 진행
- LLC 미스와 메모리 명령어 비율이 상대적으로 낮음
- 데이터 병렬 처리가 모델 병렬처리보다 좋은 성능을 보임





Musical Chair

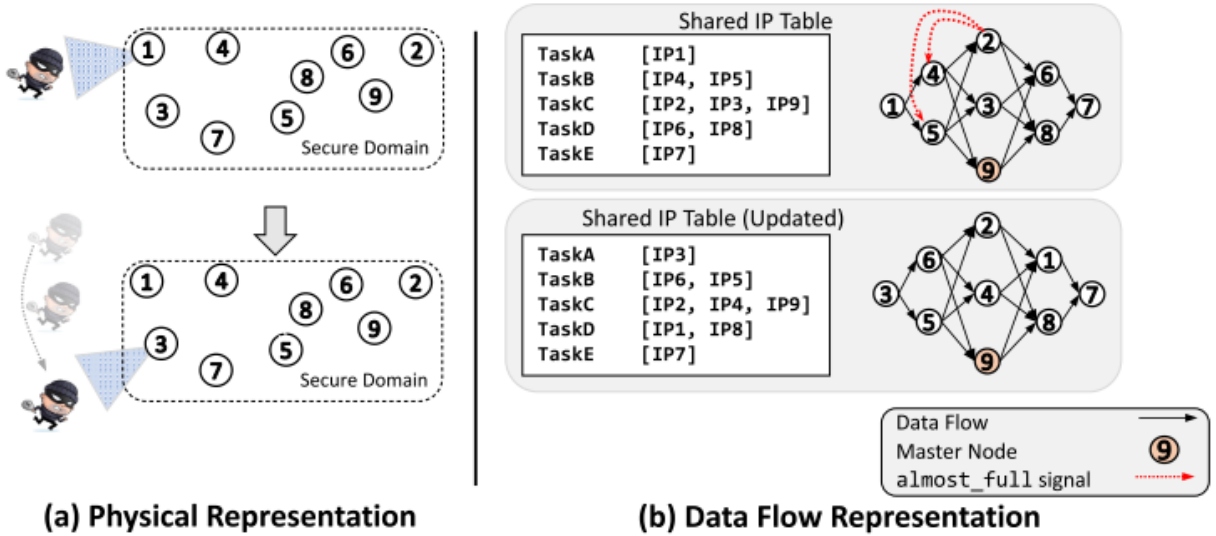
- DNN 레이어의 동작 모델을 만들기 위해 프로파일링을 수행
  - 최적의 작업 수행을 찾기 위해 시스템 장치 수를 고려한 프로파일링 진행
- DNN 모델과 장치 환경, 통신 지연 등을 분석
- 프로파일링 데이터를 기반으로 작업 할당
  - 최적의 작업의 수 결정
  - 동적인 작업 할당을 위해 모든 장치에 대해 모든 작업을 할당



Musical Chair

- 장치들은 IP Table을 가지고 이전, 이후 노드를 확인
  작업 노드가 변경되면 IP Table을 업데이트

- 효율적인 데이터와 명령 전송을 위해 Musical Chair에
  Apache Avro(원격 프로시저 호출 및 데이터 직렬화 프레임워크) 통합



(a) Physical Representation (b) Data Flow Representation

Action Recognition

- Input Data
- Two-Stream CNN
- Temporal Pyramid(pooling)
- Final Dense Layer

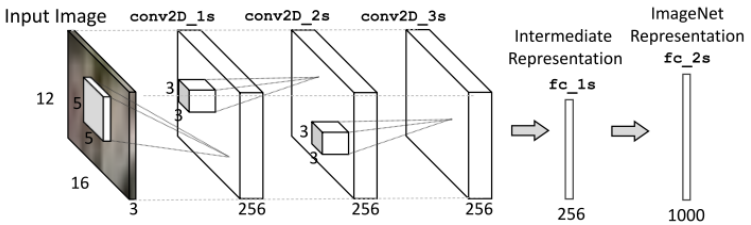


Figure 4: Spatial stream CNN.

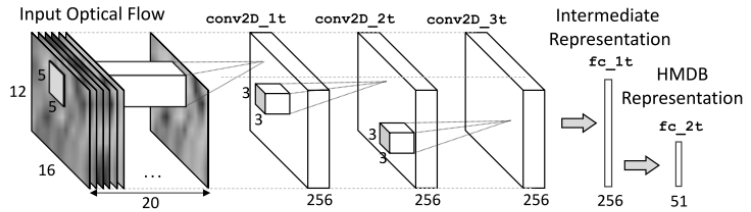


Figure 5: Temporal stream CNN

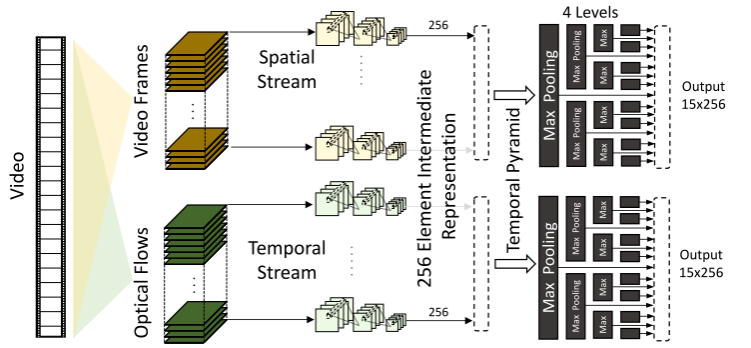
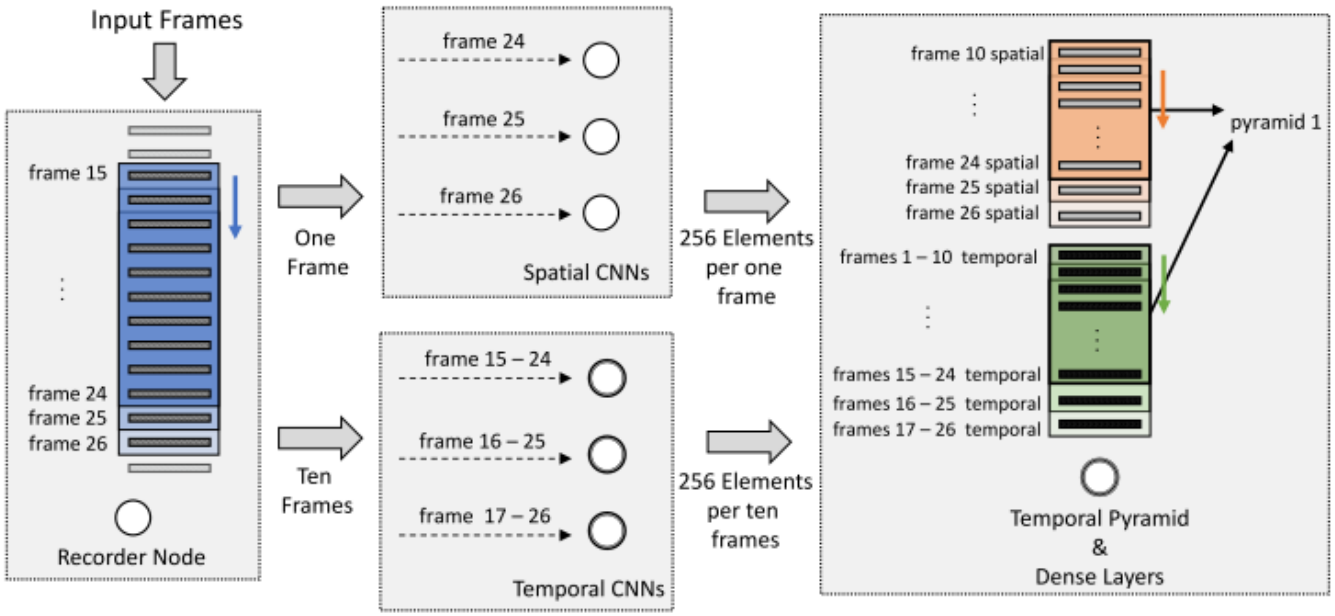


Figure 6: Four-level temporal pyramid generation in two-stream CNN.

Action Recognition

- 영상의 실시간 처리를 위해 수집된 데이터를 슬라이딩 윈도우로 활용



결론

- IoT 네트워크 내에서 리소스가 제한된 장치들의 공동 연산 방법을 통해 DNN 모델에서 상당한 연산 능력을 얻을 수 있다.