

완전 자율주행을 위한 실시간 엣지 컴퓨팅 기술과 데이터 압축 알고리즘의 통합적 접근

한정수*
백석대학교 컴퓨터공학부 교수

An Integrated Approach to Real-Time Edge Computing and Data Compression Algorithms for Full Autonomous Driving

Jung-Soo Han*
Professor, Division of Computer Engineering, Baekseok University

요약 완전 자율주행은 차량이 인간의 개입 없이 주변 환경을 인식하고 실시간으로 판단하여 주행하는 기술을 의미한다. 이를 위해 차량은 대량의 데이터를 실시간으로 처리하고 최적의 주행 결정을 내려야 한다. 그러나 기존의 클라우드 기반 접근 방식은 네트워크 지연과 대역폭 문제로 인해 한계를 갖는다. 이러한 문제를 해결하기 위해 엣지 컴퓨팅과 데이터 압축 알고리즘이 필수적이다. 본 연구에서는 완전 자율주행을 가능하게 하는 엣지 컴퓨팅의 역할과 데이터 압축 알고리즘과 이를 통해 효율적인 자율주행 시스템을 구축하는 방법을 논의한다.

주제어 : 완전 자율주행, 엣지 컴퓨팅, 데이터 압축, 양자화 신경망, 실시간 데이터 처리

Abstract Fully autonomous driving refers to the technology that enables a vehicle to perceive its surroundings, make decisions, and drive without any human intervention. To achieve this, the vehicle must process vast amounts of data in real time and make optimal driving decisions. However, traditional cloud-based approaches face limitations due to network latency and bandwidth constraints. To overcome these challenges, edge computing and data compression algorithms are essential. This study discusses the role of edge computing and data compression algorithms in enabling fully autonomous driving and explores methods for building efficient autonomous driving systems.

Key Words : Fully Autonomous Driving, Edge Computing, Data Compression, QNN, Real-Time Data Processing

1. 서론

최근 자율주행 기술은 인간의 개입 없이 차량이 스스로 주행하는 완전 자율주행 시스템을 목표로 빠르게 발전하고 있다. 완전 자율주행 차량은 주변 환경을 실시간으로 인식하고, 다양한 센서에서 수집되는 대용량 데이터를 분석하여 최적의 주행 결정을 내려야 한다. 이러한

시스템은 안전성과 효율성을 확보하기 위해 수 밀리초(ms) 단위의 신속한 반응이 필수적이다.

그러나 기존의 클라우드 기반 처리 방식은 네트워크 지연과 대역폭 한계, 그리고 신뢰성 문제로 인해 실시간성이 요구되는 자율주행 환경에서 한계를 드러낸다. 특히, 클라우드에 의존할 경우 데이터 전송 과정에서 발생하는 지연이 사고로 이어질 수 있어 매우 위험하다.

본 논문은 2025년도 백석대학교 지원을 받아 수행된 것임.

*교신저자 : 한정수(jshan@bu.ac.kr)

접수일: 2025년 03월 01일 수정일: 2025년 03월 23일 심사완료일: 2025년 04월 10일

이러한 문제를 해결하기 위해 엣지 컴퓨팅(Edge Computing) 기술이 자율주행 시스템의 핵심 요소로 부상하고 있다. 엣지 컴퓨팅은 차량 내에서 데이터를 직접 처리함으로써 네트워크 의존도를 줄이고 지연 시간을 최소화하며, 시스템의 안정성과 응답성을 향상시킨다. 이를 통해 자율주행 차량은 보다 빠르고 정확한 실시간 의사결정을 수행할 수 있다[1][2][3].

엣지 컴퓨팅은 자율주행에서는 실시간 데이터 처리, 네트워크 대역폭 절감, 보안 강화의 세 가지 핵심 역할을 수행한다. 실시간 데이터 처리를 위해 병렬 처리, 비동기 데이터 스트리밍, GPU·TPU·FPGA 등 하드웨어 가속기 활용이 필요하다. 또한 신경망 경량화와 모델 압축 기술을 적용해 계산량을 줄이고, 센서 데이터와 주행 상황을 신속하게 분석할 수 있도록 한다.

네트워크 대역폭 절감을 위해서는 불필요한 데이터를 필터링하고, 핵심 정보만을 선택적으로 클라우드로 전송하는 전략이 필요하다. 이 과정에서 적응형 데이터 압축 및 이벤트 기반 전송 기법을 활용하여 통신 비용과 시스템 부하를 줄일 수 있다. 아울러, 입력 데이터의 특성을 분석해 손실(Lossy) 또는 비손실(Lossless) 압축 방식을 동적으로 선택하는 적응형 압축 알고리즘과, 오토인코더(Autoencoder)나 웨이블릿 변환(Wavelet Transform)과 같은 특징 기반 압축 기법이 효율성을 높인다. 보안 측면에서도 엣지 컴퓨팅의 중요성이 커지고 있다. 데이터의 로컬 처리로 인해 외부 전송 과정에서 발생할 수 있는 사이버 공격이나 데이터 유출 위험을 최소화할 수 있기 때문이다[4][5][6].

본 연구에서는 완전 자율주행 구현을 위해 필수적인 엣지 컴퓨팅과 데이터 압축 알고리즘의 역할을 분석하고, 이를 통해 고성능·고신뢰성의 효율적인 자율주행 시스템을 구축하는 방안을 논의하고자 한다.

2. 자율주행에서 엣지 컴퓨팅의 역할

엣지 컴퓨팅은 자율주행 차량에서 대표적인 3가지 주요 역할을 수행한다. 먼저 실시간 데이터 처리를 해야 한다. 이는 센서 및 카메라에서 발생하는 데이터를 빠르게 처리하여 즉각적인 반응을 보장하기 때문이다[7]. 이를 위해 병렬 처리를 활용하여 여러 연산을 동시에 수행하고, 비동기 데이터 스트리밍(Asynchronous Data Streaming) 기법을 적용하여 데이터 처리 속도를 높일 수 있다. 또한, 하드웨어 가속기(GPU, TPU, FPGA)를

활용하여 복잡한 연산을 병렬로 수행하며, 신경망 경량화(Neural Network Pruning) 및 모델 압축(Model Compression) 기법을 통해 연산량을 줄이고 실시간 처리를 가능하게 한다[8]. 다음으로는 네트워크 대역폭 절감이다. 이는 모든 데이터를 클라우드로 전송하지 않고 필수 데이터만 송신함으로써, 데이터 필터링과 중요도 분석을 수행하여 핵심 정보를 선별한다. 이 과정에서 적응형 데이터 압축(Adaptive Data Compression)과 이벤트 기반 데이터 전송(Event-Driven Data Transmission) 기법을 적용하여 불필요한 데이터를 제거하고 네트워크 부하를 줄인다. 그리고 적응형 데이터 압축 알고리즘이 필요하다. 이 알고리즘은 입력 데이터를 분석하여 주파수 성분과 정보량을 평가하고, 필요에 따라 손실 또는 비손실 압축 기법을 선택한다. 또한 중요한 특징을 유지하면서 덜 중요한 정보를 제거하는 오토인코더 또는 웨이블릿 변환인 특징 기반 압축기법이 필요하다. 그리고 실시간 상황에 맞게 압축률을 동적으로 조정하는 적응적 압축 비율 결정과 압축된 데이터를 전송한 후, 수신 측에서 복원하여 원본 데이터에 가까운 형태로 재구성하는 전송 및 복원을 해야 한다. 이과정에서는 이벤트 기반 데이터 전송 알고리즘이 사용되는데 센서 데이터를 지속적으로 모니터링하여 특정 조건(예: 이상 감지, 특정 임계값 초과)이 충족되었을 때 데이터 전송을 시작하는 이벤트 감지와 전송할 데이터의 우선순위를 평가하고 중요성이 낮은 데이터는 버퍼링하거나 압축하는 중요도 분석, 네트워크 상태 및 시스템 부하를 고려하여 전송 간격을 동적으로 조정하는 전송 빈도 조절, 최적화된 패킷 구성 기능으로 전송 데이터를 패킷 단위로 조립하여 최적화된 데이터 패킷을 생성하고 네트워크 효율성을 향상시켜야 한다[9][10]. 또한 지능형 데이터 캐싱 기술을 활용 최근에 전송된 데이터와 유사한 패턴을 학습하여 중복 전송을 방지하고 불필요한 네트워크 사용을 줄이고, 지능형 데이터 샘플링(Intelligent Data Sampling) 알고리즘을 활용하여 특정 조건에서만 데이터를 송신하도록 최적화함으로써 실시간 처리 성능을 향상시켜야 한다. 끝으로 보안 강화를 위해 데이터를 로컬에서 처리하여 사이버 공격 및 데이터 유출 위험을 최소화해야 한다.

3. 실시간 데이터 처리

자율주행 차량에서 실시간 데이터 처리는 수 밀리초 단위의 신속한 응답이 요구되며, 센서에서 수집한 데이

터를 즉각 분석하고 활용해야 한다. 이를 위해 딥러닝, 강화학습, 고속 데이터 스트리밍 기법이 적용된다.

3.1 YOLO 기반 객체 인식

YOLO(You Only Look Once)는 실시간 객체 탐지를 위한 딥러닝 기반 알고리즘으로, 단일 신경망을 사용하여 이미지 내 모든 객체를 한 번의 전방향 패스로 탐지하는 방식이다[11]. 기존의 R-CNN 계열과 달리, YOLO는 입력 이미지를 SxS 그리드로 나누고 각 그리드에서 객체를 예측하는 방식으로 빠른 속도를 제공한다. YOLO 알고리즘의 동작은 입력된 이미지를 고정된 크기로 조정 후 신경망에 전달하고 이미지의 픽셀 값을 정규화하여 학습 안정성을 증가시키는 역할을 한다. 그리고 그리드 기반 탐지 기술로 이미지를 SxS 그리드로 나누고 각 그리드 셀에서 객체를 예측한다. 이때 각 셀은 B개의 바운딩 박스를 예측하며, 각 박스는 좌표 및 신뢰도 점수(confidence score)를 포함한다. 그리고 객체 분류 및 점수 계산을 위해 각 바운딩 박스는 특정 클래스(예: 자동차, 보행자 등)에 대한 확률을 갖는다. 이때의 신뢰도 점수는 바운딩 박스 내 객체 존재 확률과 객체가 특정 클래스에 속할 확률을 곱한 값이다. 또한 비최대 억제(Non-Maximum Suppression, NMS) 기술을 사용하여 중복된 바운딩 박스를 제거하여 최종 탐지 결과를 최적화한다. 또한 IoU (Intersection over Union) 임계값을 설정하여 신뢰도가 높은 박스만 유지하도록 한다. 그 결과 최종적으로 탐지된 객체의 클래스와 위치 정보를 출력하게한다. 이때 속도와 정확도의 균형을 맞추기 위해 앵커 박스(Anchor Boxes) 및 다중 스케일 예측 기법이 사용될 수 있다. 이처럼 YOLO 알고리즘은 단일 패스로 전체 이미지를 분석하는데 높은 속도 제공하며 더 나은 탐지 성능기능을 갖고 있어 실시간 애플리케이션(예: 자율주행 차량, 보안 감시)에 적합한 알고리즘이지만 작은 객체 탐지 성능이 상대적으로 낮으며고해상도 이미지에서 정밀한 탐지가 어려울 수 있다는 단점이 있다.

YOLO는 지속적인 연구 개발을 통해 YOLOv4, YOLOv5, YOLOv7 등으로 발전하고 있으며, 정확도와 속도를 개선하기 위해 다중 스케일 피쳐 맵 및 어텐션 메커니즘이 추가되는 등 다양한 최적화가 이루어지고 있다. 결과적으로 YOLO는 단일 신경망을 활용하여 이미지 내 객체를 빠르게 인식하고, 실시간으로 주변 차량, 보행자, 장애물을 탐지하고 충돌 가능성을 분석하는 기능으로 적합한 알고리즘이다[12][13]. 최근에는 경량화

모델(YOLOv5n, YOLOv8n)도 개발되어 엣지 디바이스와 같은 저사양 환경에서도 실시간 처리가 가능해지고 있다. 또한, 고해상도 이미지와 복잡한 장면에서도 정확한 탐지 위해 패널티 함수와 학습 최적화 기법이 적용되고 있다. 이러한 발전을 통해 YOLO는 자율주행뿐만 아니라 드론, CCTV, 로봇 등 다양한 실시간 객체 인식 분야에서 활용 범위를 넓혀가고 있다.

3.2 Kalman Filter 기반 데이터 융합

Kalman Filter는 실시간으로 다중 센서 데이터를 결합하여 차량의 위치 및 속도를 정확하게 예측하는 알고리즘이다. 이 필터는 상태 추정(State Estimation) 문제를 해결하기 위해 확률 모델을 사용하며, 센서의 노이즈를 제거하고 더 정확한 상태 정보를 제공한다[14][15]. Kalman Filter 알고리즘 과정은 다음과 같다.

- ① 초기화(Initialization): 초기 예측값을 센서 데이터의 평균이나 기존 시스템의 초기 상태 값으로 설정 (초기 상태 벡터 x_0 와 초기 공분산 행렬 P_0 설정)

- ② 예측 단계(Prediction Step)

- 상태 벡터를 동적 모델을 기반으로 예측:

$$x_k^- = Fx_{k-1} + Bu_k$$

(F는 상태 전이 행렬, B는 제어 입력 행렬, u_k 는 제어 입력 벡터)

- 공분산 행렬을 갱신: $P_k^- = FP_{k-1}F^T + Q$

(Q는 프로세스 노이즈 공분산 행렬)

- ③ 갱신 단계(Update Step) - 칼만 이득 계산:

$$K_k = P_k^- H^T ((HP_k^- H^T + R)^{-1})$$

(H: 측정 모델 행렬, R은 측정 노이즈 공분산 행렬)

- 새로운 측정값 반영하여 상태 벡터 업데이트:

$$x_k = x_k^- + K_k(z_k - Hx_k^-) \quad (z_k \text{는 측정값})$$

- 공분산 행렬 갱신: $P_k = (I - K_k H)P_k^-$

(여기서 I는 항등 행렬)

Kalman Filter의 장점은 다중 센서 데이터를 효율적으로 융합하여 노이즈를 제거하고 신뢰성 높은 데이터를 제공하며, 실시간 연산이 가능하여 자율주행 차량에서의 빠른 의사 결정에 적합하다. 또한 확률적 모델을 기반으로 하기 때문에 측정 오차와 시스템 불확실성을 효과적으로 처리할 수 있다. 이처럼 Kalman Filter는 상태 추정 및 데이터 융합을 위한 강력한 알고리즘으로, 다중 센서 환경에서 신뢰성 높은 추정치를 제공하는 데 중요한

역할을 한다. 이를 통해 차량 위치 및 속도를 더욱 정확한 예측을 수행하는 알고리즘이다.

4. 데이터 압축 알고리즘

딥러닝 기반 데이터 압축은 인공지능을 활용하여 데이터를 효율적으로 압축하고, 원본 데이터를 복원할 때 정보 손실을 최소화하는 방식이다. 대표적인 기술로는 오토인코더 및 양자화 신경망(Quantized Neural Network)이 있다.

4.1 오토인코더 기반 데이터 압축

오토인코더는 입력 데이터를 저차원 잠재 공간(Latent Space)으로 변환한 후 다시 원래 데이터로 복원하는 비지도 학습 모델이다. 오토인코더 알고리즘은 입력 데이터(X)를 인코더 신경망에 입력하여 저차원 표현(Z)으로 압축한다. 이 과정에서 CNN(Convolutional Neural Network), RNN(Recurrent Neural Network) 또는 Transformer 기반 구조를 활용하여 특징을 추출하고, 차원 축소를 수행할 수 있다. 또한, 변분 오토인코더(Variational Autoencoder, VAE)를 활용하여 확률 분포를 학습함으로써 더욱 일반화된 압축 표현을 생성할 수 있다. 특정한 응용에서는 스파스 오토인코더(Sparse Autoencoder)를 활용하여 희소 표현을 극대화함으로써 모델의 해석력을 향상시킬 수 있다.

압축된 데이터(Z)를 디코더(Decoder) 신경망에 입력하여 원본 데이터(X')로 복원한다. 이 과정에서 업샘플링(Upsampling) 기법과 CNN 기반의 전이 학습(Transfer Learning)을 활용하여 복원 성능을 높일 수 있다. 또한, 주파수 변환(예: Discrete Wavelet Transform, DWT) 기법을 결합하여 원본 데이터의 특징을 더욱 효과적으로 복원할 수 있다.

그리고 손실 함수(예: Mean Squared Error, MSE)를 활용하여 원본 데이터(X)와 복원 데이터(X')의 차이를 최소화하는 방향으로 학습한다. 또한, 구조적 유사도 지수(Structural Similarity Index, SSIM) 및 절대 평균 오차(Mean Absolute Error, MAE)와 같은 추가적인 손실 함수를 도입하여 시각적 품질을 개선할 수 있다. 더 나아가, 생성적 적대 신경망(GAN) 기반 손실 함수를 적용하면 더욱 자연스러운 복원이 가능하다[8][10]. 최근에는 Perceptual Loss와 같이 사람의 시각적 인지 특성을 반영한 손실 함수도 활용되어 복원 데이터의 질적 향상을

도모하고 있다. 이러한 복합적 손실 함수 설계는 데이터 압축 및 복원 과정에서 균형 잡힌 성능과 시각적 품질 확보에 중요한 역할을 한다. 손실 함수 작동 방식은 다음과 같다.

- ① Mean Squared Error (MSE):

$$MSE = (1/n) * \sum (X_i - X'_i)^2$$

원본데이터(X)와 복원데이터(X') 간의 차이를 제공하여 평균으로 계산하며 작은 차이에도 민감하여 전체적인 오류를 최소화하는 데 유용함

- ② Structural Similarity Index Map(SSIM): 시각적 품질을 고려한 비교 지표로 명암, 대비, 구조적 차이 기반 유사성을 평가하며 MSE보다 사람이 인지하는 품질 차이에 더 민감함

$$SSIM(X, X') = \frac{(2\mu_X \mu'_X + C_1)(2\sigma_{XX} + C_2)}{(\mu_X^2 + \mu'^2_X + C_1)(\sigma_X^2 + \sigma'^2_X + C_2)}$$

- ③ Mean Absolute Error (MAE):

$$MAE = (1/n) * \sum |X_i - X'_i|$$

각 데이터 포인트 간 차이의 절대값 평균, 이상치(outlier) 영향이 적고, 단순한 차이 평가에 유용.

- ④ Generative Adversarial Network(GAN) 기반 손실 : 생성적 적대 신경망을 활용하여 복원된 데이터가 실제 데이터 분포와 유사하도록 학습하며, 판별기(Discriminator)와 생성기(Generator)를 경쟁적으로 학습시켜 현실적인 데이터 복원을 유도한다.

이러한 손실 함수들을 조합하여 자율주행 시스템에서 효율적인 데이터 복원을 수행할 수 있으며, 이를 통해 센서 융합 데이터의 신뢰성을 높이고 실시간 주행 성능을 향상시킬 수 있다.

4.2 양자화 신경망

양자화 신경망(Quantized Neural Network, QNN)은 모델의 가중치와 활성화 함수를 저비트 표현(예: 8비트, 4비트)으로 변환하여 연산 속도를 높이고 저장 공간을 줄이는 기술이다[9][15]. 이를 위해 정수 연산(Integer Arithmetic)을 활용하여 부동소수점 연산을 줄이고, 적응형 양자화(Adaptive Quantization) 및 정규화(Clipping & Normalization) 기법을 적용하여 정확도를 유지하면서도 효율적인 계산이 가능하도록 한다. 또한, 행렬 곱셈 연산을 최적화하는 Sparse Computation 기법과 LUT(Look-Up Table) 기반 연산을 적용하여 연산 속도

를 최적화할 수 있다. Sparse Computation은 모델의 가중치 행렬에서 0이 많은 희소 행렬 구조를 활용하여 불필요한 연산을 줄이는 방식이다. 이를 통해 메모리 사용량을 감소시키고 연산 속도를 높일 수 있다. LUT 기반 연산은 사전에 계산된 결과 값을 테이블에 저장하여 복잡한 연산을 빠르게 수행하는 방식으로, 딥러닝 모델의 활성화 함수 계산 및 양자화된 연산을 효율적으로 처리하는 데 유용하다. 이러한 최적화 기법을 하드웨어 가속기(NPU, FPGA)와 결합하면 엣지 장치에서 실시간 처리가 가능해지며 연산 속도를 더욱 향상시킬 수 있다.

5. 결론

본 연구에서는 완전 자율주행 시스템 구현을 위해 필수적인 엣지 컴퓨팅 기술과 데이터 압축 알고리즘의 역할과 적용 방안을 살펴보았다. 완전 자율주행 차량은 복잡하고 방대한 양의 데이터를 실시간으로 처리하고 분석해야 하며, 이를 통해 안전하고 효율적인 주행 결정을 내려야 한다.

기존 클라우드 기반 접근 방식은 네트워크 지연과 대역폭 한계로 인해 실시간성이 중요한 자율주행 환경에서는 한계가 있다. 이를 해결하기 위해 차량 내에서 데이터를 직접 처리하는 엣지 컴퓨팅 기술이 핵심적 역할을 수행한다. 엣지 컴퓨팅은 실시간 데이터 처리, 네트워크 대역폭 절감, 보안 강화라는 세 가지 핵심 기능을 수행한다. 병렬 처리, 비동기 스트리밍, 하드웨어 가속기 활용을 통해 복잡한 연산을 신속하게 처리하고, 신경망 경량화 및 모델 압축으로 연산 부담을 줄여 실시간성을 확보할 수 있다. 또한, 적응형 데이터 압축과 이벤트 기반 전송 기술을 통해 네트워크 대역폭 사용을 최소화하고, 중요 데이터만 전송하는 방식으로 효율적인 데이터 관리가 가능하다. 이는 통신 비용 절감과 함께 시스템 부하 감소로 이어진다.

데이터 최적화 측면에서는 오토인코더 기반 압축과 양자화 신경망 기술이 핵심으로 작용한다. 오토인코더는 고차원 데이터를 저차원으로 변환 및 복원하는 과정에서 중요한 특징만 보존하여 데이터 크기를 줄이는 동시에 시각적 품질을 유지한다. 양자화 신경망은 모델 가중치와 연산을 저비트로 변환해 연산 속도를 높이고 메모리 사용량을 줄이며, 하드웨어 가속과 결합할 경우 엣지 환경에서도 고성능 처리가 가능해진다.

실시간 객체 인식에서는 YOLO 알고리즘이 높은 처

리 속도와 정확도로 자율주행 차량의 시각 인식 성능을 향상시키며, Kalman Filter는 센서 융합과 상태 추정을 통해 위치 및 속도 예측의 정확도를 높여준다. 이러한 기술적 요소들은 상호보완적으로 작용하여 자율주행 시스템의 실시간성, 안정성, 효율성을 극대화한다. 특히, 적응형 알고리즘과 지능형 데이터 관리 기술은 상황 변화에 맞춰 시스템 성능을 최적화하고, 자원 소모를 최소화하는 효과를 제공한다.

향후 연구에서는 다양한 도로 환경과 복잡한 주행 상황에서도 안정적으로 작동할 수 있도록 알고리즘의 정교화와 하드웨어 최적화가 필요하다. 또한, 사이버 보안 위협에 대응하기 위한 엣지 보안 기술의 고도화도 필수적이다. 결론적으로, 엣지 컴퓨팅과 데이터 압축 기술은 완전 자율주행 구현의 핵심 기반으로, 이들의 효율적 융합과 실시간 처리 기술 확보가 안전하고 신뢰성 높은 자율주행 시스템 구축의 관건이 될 것이다.

REFERENCES

- [1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, Vol.19, No.4, pp.2322-2358, Fourthquarter 2017.
- [2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, Vol.3, No.5, pp.637-646, 2016.
- [3] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet of Things Journal*, Vol.3, No.6, pp.854-864, 2016.
- [4] H. Guo, J. Liu, and J. Zhang, "Mobile Edge Computation Offloading for Ultra-Dense IoT Networks," *IEEE Internet of Things Journal*, Vol.5, No.6, pp.4977-4988, 2018.
- [5] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, Vol.5, No.1, pp.450-465, 2018.
- [6] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing," *IEEE Transactions on Signal and Information Processing over Networks*, Vol.1, No.2, pp.89-103, 2015.
- [7] M. Chen, Y. Hao, L. Hu, M. S. Hossain, and A. Ghoneim, "Edge-CoCaCo: Toward Joint Optimization of Computation, Caching, and Communication on Edge Cloud," *IEEE Wireless Communications*, Vol.25, No.3, pp.21-27, 2018.
- [8] J. Park, S. Samarakoon, M. Bennis, and M. Debbah,

- "Wireless Network Intelligence at the Edge," Proceedings of the IEEE, Vol.107, No.11, pp.2204-2239, 2019.
- [9] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D Communications Meet Mobile Edge Computing for Enhanced Computation Capacity in Cellular Networks," IEEE Transactions on Wireless Communications, Vol.18, No.3, pp.1750-1763, 2019.
- [10] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, Honolulu, HI, pp.63-71, 2018.
- [11] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, "DeepDecision: A Mobile Deep Learning Framework for Edge Video Analytics," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, Honolulu, HI, pp.1421-1429, 2018.
- [12] Y. Kang et al., "Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," ACM SIGARCH Computer Architecture News, Vol.45, No.1, pp.615-629, 2017.
- [13] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating While Computing: Distributed Mobile Cloud Computing over 5G Heterogeneous Networks," IEEE Signal Processing Magazine, Vol.31, No.6, pp.45-55, 2014.
- [14] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds," IEEE Transactions on Vehicular Technology, Vol.66, No.4, pp.3435-3447, 2017.
- [15] J. Jang, K. Tulkinbekov, and D.-H. Kim, "Task Offloading of Deep Learning Services for Autonomous Driving in Mobile Edge Computing," Electronics, Vol.12, No.15, Article 3223, 2023.

한 정 수(Jung-Soo Han)

[정회원]



- 1992년 8월 : 경희대학교 컴퓨터 공학부(공학석사)
- 2000년 8월 : 경희대학교 대학원 컴퓨터공학부(공학박사)
- 2001년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 교수

<관심분야>

AI 교육, 자율주행, 데이터 분석, SW 모델링