

EdgeCPS 플랫폼을 위한 지식 공유 그래프를 활용한 컴포넌트 기반 AI 응용 지원 시스템

김영주*

Component-based AI Application Support System using Knowledge Sharing Graph for EdgeCPS Platform

Young-Joo Kim*

*Principal Researcher, Electronics and Telecommunications Research Institute, Daejeon, 34129 Korea

요약

AI 관련 산업의 급속한 발전으로 인해 무수히 많은 엣지 디바이스가 실세계에서 동작되고 있고, 이들 디바이스로 구성된 스마트 공간에서 발생하는 데이터가 상상을 초월함으로 엣지 디바이스가 처리하는 것이 점점 어려워지고 있다. 이러한 문제를 해결하기 위해서 EdgeCPS 기술이 등장하게 되었다. EdgeCPS는 엣지 디바이스와 엣지 서버간 연동과 자원증강 및 기능증강을 통하여 AI 응용 서비스를 포함한 다양한 응용 서비스의 원활한 수행을 지원하기 위한 기술이다. 따라서, 본 논문에서는 EdgeCPS 플랫폼에 적용 가능한 지식 공유 그래프 기반의 컴포넌트화된 AI 응용 지원 시스템을 제안한다. 지식 공유 그래프는 AI 응용 작성에 필수적인 요소인 학습데이터, 학습된모델, 학습알고리즘, 디바이스 등에 대한 정보를 효과적으로 저장할 수 있도록 설계된다. 그리고 EdgeCPS 플랫폼의 지원 하에서 자원증강 및 기능증강을 손쉽게 변경할 수 있도록 AI 응용이 컴포넌트화 되어 동작한다. AI 응용 지원 시스템은 사용자가 손쉽게 응용을 작성할 수 있고 테스트 해 볼 수 있도록 지식 공유 그래프와 연동되고, 응용에 대한 파이프라인을 통해서 응용의 실행 양상을 사용자에게 시각화를 해 준다.

ABSTRACT

Due to the rapid development of AI-related industries, countless edge devices are working in the real world. Since data generated within the smart space consisted of these devices is beyond imagination, it is becoming increasingly difficult for edge devices to process. To solve this issue, EdgeCPS has appeared. EdgeCPS is a technology to support harmonious execution of various application services including AI applications through interworking between edge devices and edge servers, and augmenting resources/functions. Therefore, we propose a knowledge-sharing graph-based componentized AI application support system applicable to the EdgeCPS platform. The graph is designed to effectively store information which are essential elements for creating AI applications. In order to easily change resource/function augmentation under the support of the EdgeCPS platform, AI applications are operated as components. The application support system is linked with the knowledge graph so that users can easily create and test applications, and visualizes the execution aspect of the application to users as a pipeline.

키워드 : EdgeCPS, 엣지 디바이스, 인공지능, 컴포넌트, 지식 공유 그래프

Keywords : EdgeCPS, Edge Device, AI, Component, Knowledge Sharing Graph

Received 13 July 2022, Revised 26 July 2022, Accepted 27 July 2022

* Corresponding Author Young-Joo Kim(E-mail:kr.yjkim@etri.re.kr, Tel:+82-42-860-1092)

Principal Researcher, Electronics and Telecommunications Research Institute, Daejeon, 34129 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.8.1103>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

엣지 디바이스(edge device)는[1] 데이터를 발생하는 기기로 데이터를 생성 또는 수집하는 사물 인터넷(IoT) 센서들, 비디오/감시 카메라, 인터넷에 연결된 가전 기기, 스마트폰과 같은 스마트 기기 등을 포함한다. 이러한 엣지 디바이스가 활용되는 다양한 분야 (스마트 공장, 스마트시티, 자율주행, 국방)에서 무수히 많은 데이터가 서비스의 형태에 따라 저지연 혹은 고속 처리될 수 있고, 서비스 고도화를 위해서 5G 망 연동이나 AI 기술 등을 활용할 수 있다. 이러한 것을 실현하기 위해서는 대부분 고사양 엣지 디바이스에서 가능하다. 하지만, 다양한 분야에서 서비스 되고 있는 엣지 디바이스들은 저성능부터 고성능까지 다양하게 사용되고 있다. 저성능 디바이스는 AI 기술과 같은 많은 연산량과 복잡한 기술이 적용된 서비스가 탑재되어 실행되는 것이 쉽지 않기 때문에 최적화[2]나 경량화[3]등으로 문제를 해결하려는 노력이 지속적으로 이루어지고 있다.

각 디바이스 특성과 AI 서비스 특성이 다를 뿐만 아니라, 사용자 요구사항도 다르기 때문에 최적화나 경량화로 이 모든 것을 해결하는 것은 쉽지 않다. 이것을 효과적으로 해결할 수 있는 방법 중에 하나는 사용자가 원하는 요구사항을 만족할 수 있도록 디바이스를 제공하거나 프로그램의 기능을 조절해 주는 것이다. 이러한 것을 가능하게 하는 것이 EdgeCPS 플랫폼 기술[4, 5]이다. EdgeCPS 기술은 자원에 대한 성능 증강 또는 기능증강을 통해서 사용자가 원하는 서비스 요구사항을 만족시켜 실제계를 지능적으로 제어할 수 있다. 이 기술의 장점을 극대화하려면 AI 응용들을 포함한 다양한 응용들이 제공되어 서비스되어야 한다. 하지만, 사용자가 AI 응용을 손쉽게 작성하여 실행 및 배포하는 것이 쉽지 않다.

따라서, 본 논문에서는 EdgeCPS 플랫폼에 적용 가능한 지식 공유 그래프 기반의 컴포넌트화된 AI 응용 지원 시스템을 제안한다. 제안된 시스템은 AI 응용 작성에 필요한 여러 가지 정보들을 제공하는 지식 공유 그래프를 활용한다. 이 지식 공유 그래프는 학습데이터, 학습된 모델, 학습알고리즘, 디바이스 등에 대한 정보를 효과적으로 저장하도록 설계된다. 그리고 컴포넌트 단위로 AI 응용 블록을 제공하여 사용자가 손쉽게 응용을 작성할 수 있다. 작성된 응용에 대해서 쿠버네티스[6, 7] 상에서 동작하는 쿠버플로우[8] 파이프라인 기능[9]을 이용하

여 워크플로우를 생성한다. AI 응용을 컴포넌트 단위로 지원하는 장점은 필요에 따라 자원증강 및 기능증강 등이 용이할 수 있기 때문이다.

2장에서는 연구배경으로 EdgeCPS의 개념에 대해서 설명하고, 3장에서는 EdgeCPS 플랫폼에서 활용할 수 있는 컴포넌트 기반의 AI 응용 지원 시스템에 대해서 설명한다. 4장에서는 제안된 시스템의 구현 결과를 소개한다. 마지막으로 5장에서는 결론과 향후과제로 마무리를 한다.

II. 연구배경

본 절에서는 EdgeCPS 기술에 대한 소개와 이 기술이 왜 필요한 것인지에 대해서 설명한다.

2.1. EdgeCPS

그림 1은 EdgeCPS의 개념도를 보인 것이다. HW적 관점에서의 성능 증강과 SW적 관점에서의 기능 증강으로 나눌 수 있다. 전자의 경우는 목적에 맞게 증강하기 위해서 코어, 저장공간, 메모리, 디바이스 등을 할당받을 수 있다. 예를 들어, 그림 1의 왼쪽에 있는 “Devices”에서 드론을 선택하여 얼굴 인식을 할 수 있는 AI 응용을 실행하고자 할 때 자원 부족 등으로 동작이 원활하지 않아서 목적에 맞게 증강을 받는다. “EdgeCPS Platform”에서 자원 증강을 받아 GPU를 사용할 수 있는 드론으로 얼굴 인식 AI 응용을 원활하게 동작시킬 수 있다.

후자의 경우는 기존 기능에 대한 고도화 및 요구사항에 맞는 최적화된 AI 응용을 제공하기 위해서 ML, CNN, RNN 등의 인공지능 요소 등을 이용하여 Non-AI 응용을 AI 응용으로 변경할 수 있고, AI 에 관련된 기술인 최적화, 개별화, 군집화, 분산화 등 이용하여 AI 응용

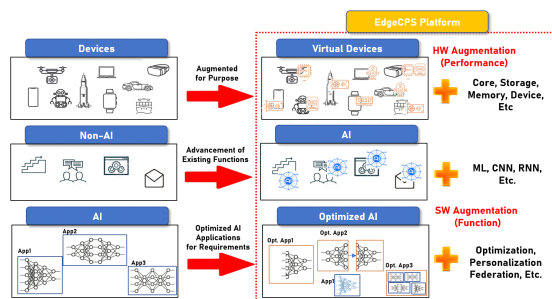


Fig. 1 The Concept of EdgeCPS

을 Optimized AI 응용으로 변경할 수 있다. 이렇게 변경된 응용으로 해당 디바이스에서 수행할 수 있다.

실세계에 무수한 많은 디바이스들이 실제로 배포되어 다양한 서비스들이 실행되고 있지만, 산업의 급속한 발전과 AI 기술의 발전 등으로 인해서 기존 디바이스를 그대로 활용하면서 실세계를 정밀하고 지능적으로 제어하기 쉽지 않다. 이러한 문제를 해결하기 위해서 EdgeCPS 기술이 등장했다. 이 기술은 엣지 디바이스와 엣지 서버의 성능차이를 극복하면서 엣지 디바이스에서 실행되는 응용들이 사용자 요구사항에 맞게 원활하게 수행할 수 있도록 지원하는 것이다. 이러한 기술의 장점을 극대화하려면 AI 응용들을 포함한 다양한 응용들이 제공되어 서비스되어야 한다. 하지만, 사용자가 AI 응용을 손쉽게 작성하여 실행 및 배포하는 것이 쉽지 않다.

III. EdgeCPS 지원 지식 공유 그래프를 활용한 컴포넌트 기반 AI 응용 지원 시스템

본 절에서는 EdgeCPS 플랫폼 상에서 수행 가능하도록 컴포넌트 기반의 AI 응용을 손쉽게 구성할 수 있는 시스템에 대해서 설명한다.

3.1. 전체 시스템

그림 2는 EdgeCPS를 위한 지식 공유 그래프 기반의 AI 응용 지원 시스템을 보인 것이다. 이 시스템은 “Programming Interface”를 통해서 컴포넌트 관리

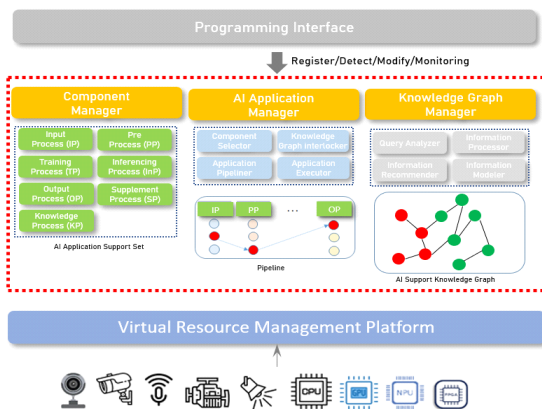


Fig. 2 Whole Architecture of Component based AI Application Support System for EdgeCPS

(Component Manager), 지식 그래프 관리(Knowledge Graph Manager), 그리고 AI 응용 관리(AI Application Manager)를 한다. 작성된 AI 응용은 쿠버플로우[8] 같은 가상자원관리플랫폼[6, 8] 상에서 동작한다. 컴포넌트 관리기에서는 AI 응용 지원을 위해서 입력처리(Input Processing), 전처리(Pre Processing), 학습처리(Training Processing), 추론처리(Inferencing Processing), 출력처리(Output Processing), 부가처리(Supplement Processing), 지식처리(Knowledge Processing)의 7개 파트로 구분된 AI 응용 지원 집합 (AI Application Support Set)이 정의되어 있다. 지식 그래프 관리기에서는 AI 지원을 위한 지식 그래프를 구축 및 관리를 한다. 이 지식 그래프는 Coarse-grained 기반의 정보를 저장한다. Coarse-grained 기반의 정보가 의미가 하는 것은 AI 응용 작성에 필요한 학습데이터, 학습된 모델, 그리고 학습 알고리즘 등을 사용자가 추가적인 처리없이 프로그램에서 바로 사용할 수 있는 정보이다. 이 관리기는 4개의 모듈로 구성되어 있다. 그리고 AI 응용 작성기를 통해서 사용자가 원하는 AI 응용 프로그램을 작성한다. 응용 작성기는 세부적으로 컴포넌트 선택기(Component Selector), 지식 그래프 연동기(Knowledge Graph Interlocker), 응용 파이프라인기(Application Pipeliner), 응용 실행기(Application Executor)로 구성되어 있다. 컴포넌트 선택기는 컴포넌트 관리기에서 AI 응용에 필요한 컴포넌트를 사용자가 순서적으로 선택할 수 있다. 지식 그래프 연동기는 AI 응용 작성시에 필요한 학습데이터, 학습된 모델, 학습된 알고리즘 등이 저장된 지식 그래프에서 가져올 수 있는 환경을 제공한다. 그리고 응용 파이프라인기는 사용자가 AI 관련 컴포넌트들과 지식 그래프의 정보를 참고하여 작성된 응용에 대해서 파이프라인을 생성한다. 응용 실행기는 생성된 파이프라인 기반의 응용을 가상자원 관리플랫폼 상에서 실행한다.

3.2. EdgeCPS를 위한 지식 공유 그래프

지식 공유 그래프는 그래프 기반의 정보 저장소이므로 노드와 관계로 구성되어 있다. 따라서, EdgeCPS 플랫폼 상에서 AI 응용 작성을 효과적으로 지원하기 위해서 노드와 관계에 대한 정의가 필요하다.

첫 번째, 표 1은 지식 공유 그래프를 위한 노드와 그 노드에 대한 속성을 정의한 것이다. 노드는 총 6 가지로 구성된다.

Table. 1 Node Definition for Knowledge Sharing Graph

Label Name	Properties	Examples
ModelType	Name,Path	ResNetType
ModelDetailType	Name,InputType, TotalParams, TotalSizeMB, MinimumModelSize, Path	ResNet18, ResNet50, ResNet152
Dataset	name,DataType, ImSize,ImType, ImNum,ClassNum, Path	Cifar
SubDataset	name,DataType, ImSize,ImType, ImNum,ClassNum, ParentData, Path	Cifar10, Cifar20, Cifar100
Device	name,CPUSpec, GPUSpec,NPUSpec, Memory	Raspberry PI4, Xavier
TrainedModel	name, ModelSize,Path	ResNet50_Cifar10_Epoch50

- ModelType: AI 알고리즘의 대표성을 나타내는 이름을 표시 (예, ResNet_XXX)
- ModelDetailType: 실제 AI 응용에 사용되는 알고리즘을 표시 (예, ResNet18, ResNet50)
- Data: AI 학습에 필요한 데이터의 대표성을 나타내는 이름을 표시 (예, Cifar_XXX)
- SubData: AI 학습 시에 실제적으로 필요한 학습데이터에 대한 것을 표시 (예, Cifar10, Cifar100)
- Device: AI 기반의 학습기 및 추론기가 실제적으로 동작하는 하드웨어를 표시 (예, RaspberryPI4)
- TrainedModel: 임의 학습데이터와 임의의 AI 알고리즘으로 학습된 모델을 표시 (예, ResNet50_Cifar10_Epoch50.pt)

Table. 2 Relation Definition for Knowledge Sharing Graph

Label Name	Relation
SUBModel	ModelType — ModelDetailType
SUMData	Dataset — SubDataset
ModelIn	ModelDetailType — TrainedModel
TrainedOn	Device — TrainedModel
Trained{ }	SubDataset — TrainedModel
Embedded{ }	ModelDetailType — Device
Resulted{ }	TrainedModel — Device

두 번째, 표 2는 지식 공유 그래프를 위한 노드간의 관계를 정의한 것이다. 관계의 종류는 총 7 가지로 구성된다. 표 1과 표 2에서 정의된 노드와 관계를 이용하여 AI 응용 지원을 위한 AI 전용 지식 공유 그래프를 구성할 수 있다. 본 논문에서 의미하는 지식은 인공지능에 관련 응용 작성에 필요한 정보이며, 이들 정보에 대해서 추가적인 처리없이 프로그램에 적용할 수 있다.

3.3. 컴포넌트 기반의 AI 응용 지원 시스템

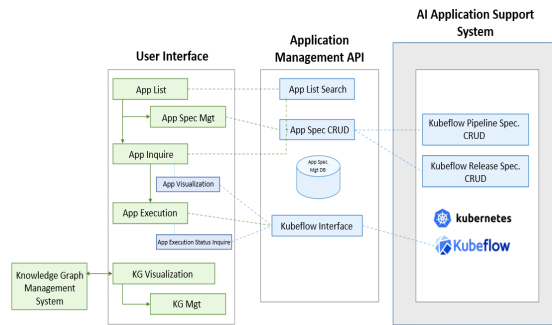


Fig. 3 Component based AI Application Support System

그림 3은 컴포넌트 기반의 AI 응용 지원 시스템을 보인 것이다. 이 시스템은 쿠버네티스 위에서 파이프라인으로 구현되어 동작하는 AI 응용을 사용자 인터페이스를 통해서 생성, 조회, 삭제, 실행 등의 기능을 제공한다. AI 응용은 컴포넌트로 구성되고, 이런 컴포넌트의 흐름은 파이프라인으로 구성하여 하나의 응용으로 구현된다. 그림 2에서 컴포넌트 관리기는 AI 응용에 관련된 컴포넌트 관리하여 응용 작성을 효과적으로 지원한다. 이를 위해서 컴포넌트 명세에 대한 정의가 되어 있어야 한다. 그림 4는 컴포넌트 관리를 위한 명세를 보인 것이다. 이 명세에서 중요한 부분은 “Component Image Registry”, “Component Image Repository”, “Component Image Tag”, “Component IO” 부분이다. 그리고 작성된 응용에 대해서 파이프라인을 생성하기 위해서 컴포넌트 기반의 응용 명세가 정의되어야 한다. 그림 5는 컴포넌트 기반의 응용 명세를 보인 것이다. 이 두 개의 명세를 이용하여 사용자 인터페이스를 통해서 컴포넌트와 응용을 관리할 수 있다.

응용을 구성하는 컴포넌트는 특정 기능을 수행하는 응용의 구성요소로, Docker 컨테이너 이미지 하나는 갖는다. 이미지 하나를 생성하기 위해서는 Fig. 6와 같이

“Program Code”를 실행시킬 명령어 (Cmd), 입력 값 (Input), 출력 값(Output)을 설정하고, DockerFile을 이용하여 빌드하여 이미지 파일 (Image file)을 생성한다. 이런 이미지 파일을 이용하여 컴포넌트를 생성하게 된다. 본 논문에서는 그림 7과 같이 컴포넌트 생성은 KubeFlow에서 지원하는 방법을 사용한다. 이렇게 생성된 이미지를 이용하여 컴포넌트 기반의 응용을 작성한다. 이렇게 작성된 응용을 파이프라인화하고, 실행을 위한 백엔드 (사용자 인터페이스와 연동)에 해당하는 응용 관리 API가 있다. 응용 관리 API는 크게 응용관리, 컴포넌트 관리, kubeFlow 인터페이스로 구성되어 있다. 응용 관리와 컴포넌트 관리는 앞에서 설명한 명세를 기반으로 한다. 그리고 AI 응용 지원 시스템의 핵심 부분인 kubeFlow 인터페이스는 kubeFlow에서 제공하는 kfp SDK를 이용하여 응용에 대한 파이프라인화와 파이프라인화된 응용을 실행하는 역할을 담당한다. 이렇게 파이프라인화를 제공하는 이유는 EdgeCPS를 자원 증강과 기능 증강을 효과적으로 지원하기 위함이다. 각 파이프라인을 구성하는 노드에 대한 성능정보나 하드웨어 정보를 알 수 있으므로, 사용자가 손쉽게 변경할 수 있다.

```

- Component name:
- Component Description:
# Repository that stores docker container images including components.
- Component Image Registry:
# Name of docker container images including components.
- Component Image Repository:
# Tag of docker container image including components
- Component Image Tag:
# Inputs and outputs used by the component
- Component IO:
# Input argument, type, and description of components
  · Inputs
# Output argument, type, and description of components
  · Outputs
# Describes how to pass the command to run the component and
input/output arguments in json format
- Component Command:
    
```

Fig. 4 Specification for Component Management

```

- Application name:
- Application Description:
# Tags to identify applications
- Application Tag:
# A set of components that make up an application
- Component
    
```

```

# Components included in the application
- Component:
# Component execution order
  · Sequence:
# Input specification for each component input argument
  - Inputs
# Output specification for each component input argument
  - Outputs
    
```

Fig. 5 Specification for Application Management



Fig. 6 Method to Create Docker Container Image

```

name: DownloadData Component
description: Download data for learning datasets

outputs:
- {name: Data, type: LocalPath, description:
'Pathwheredatawillbestored.'}

implementation:
container:
image: akates/ml-test:download_data_v1
command: [
python3, download_data.py,

--data,
{outputPath: Data},
]
    
```

Fig. 7 KubeFlow based Component

IV. 구현

본 절에서는 EdgeCPS 위한 지식 공유 그래프 기반의 AI 응용 지원 시스템 구축을 위해서 실험환경과 시나리오에 대해서 설명하고, 구현된 시스템에 대해서 시나리오 기반으로 설명한다.

4.1. 실험환경

본 논문에서 제안한 컴포넌트 기반의 AI 응용 지원 시스템을 구축하기 위해서 클러스터 시스템 기반의 실험환경을 구성한다. 실험환경은 하드웨어 구성과 소프트웨어 구성으로 나누어서 설명한다.

- 하드웨어 구성
 - 컴퓨터: 3대

- CPU: Core 16 3.8GHz (i7-9800)
- Memory: 126G
- GPU: GTX 1660
- 소프트웨어 구성
 - OS: 우분투 20.04 LTS
 - 관리 플랫폼: 쿠버네티스 1.18.19
쿠버플로우 1.2.0
 - 그래프 관리: Neo4j

4.2. AI 응용 시나리오

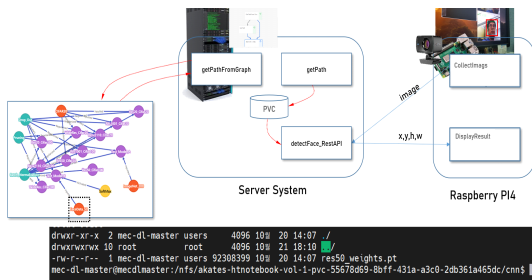


Fig. 8 Scenario of AI Application

그림 8은 본 논문에서 제안한 AI 응용 지원 시스템에서 실제로 동작되는 것을 보여주기 위한 시나리오 화면이다. 이 시나리오는 ResNet50 기반 얼굴 인식 과정을 보이는 것이다. 사전에, 수집된 얼굴 데이터를 이용하여 ResNet50 알고리즘으로 pytorch로 학습기를 만들어서 res50_weights.pt 파일을 생성한다. 생성된 학습모델은 지식 그래프에 추가된다. 그림 8의 지식 그래프에서 점선으로 표시된 부분이다. 그리고 Server System에는 쿠버네티스와 쿠버플로우가 설치되어 있고, 지식 그래프를 위해서 Neo4j가 설치되어 있다. 그리고 Raspberry PI4는 카메라가 설치되어 얼굴 이미지를 생성하고 있는 상태이다.

이런 상태에서 Server System에서는 지식 그래프에서 학습된 모델의 경로명을 읽어오는 컴포넌트, 읽어온 경로명으로 쿠버플로우 PVC에 저장하는 컴포넌트, PVC에 저장된 학습모델로 얼굴을 추론하는 컴포넌트가 동작한다. 그리고 Raspberry PI4에서는 카메라에서 생성된 얼굴 이미지를 Server System 쪽으로 지속적으로 전송을 하고, Server System쪽에서 추론하고 그 결과와 이미지의 좌표값을 Raspberry PI4 쪽으로 전송한다.

4.3. 구현

그림 9는 지식 공유 그래프를 활용한 컴포넌트 기반의 AI 응용 지원 시스템의 구현된 인터페이스 메인화면을 보인 것이다. 이 화면에서 등록된 응용 수는 7개, 등록된 컴포넌트는 7개, 지식 그래프 관련 개수도 표시되어 있다. 제안된 시스템은 Components, Applications, 지식그래프 총 3개의 메인 메뉴로 구성되어 있다.

Components 메뉴에서는 Docker image를 이용하여 컴포넌트를 생성, 수정, 삭제할 수 있다. 그림 10는 컴포넌트 등록 과정을 보인 것이다. 그림 4에 명세된 것을 기반으로 컴포넌트를 등록한다. Applications 메뉴에서는 시스템에 등록된 컴포넌트를 이용하여 응용 프로그램을 생성, 수정, 삭제할 수 있다. 그림 11은 컴포넌트 기반의 응용을 등록하는 과정을 보인 것이다. 그림 5에 명세된 것을 기반으로 응용을 등록한다. 그리고 지식 그래프 메뉴에서는 AI 응용 지원을 위해서 Neo4j로 지식 그래프가 구현된 결과를 보인 것이다. 4개의 학습데이터 노드, 5개의 지원모델 (ModelType) 노드, 12개의 지원 모델알고리즘 (ModelDetailType) 노드, 2개의 디바이스 노드로 구현되었다. 이 지식 그래프는 그림 11의 응용 생성 과정에서 “지식그래프에서 찾기” 버튼을 클릭하면 그림 12의 지식 그래프에서 사용자가 해당되는 것을 클릭하면 자동으로 그 정보가 등록된다.

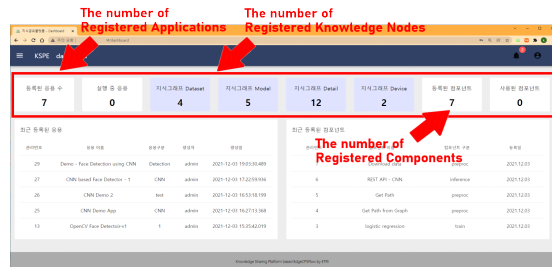


Fig. 9 Knowledge Sharing Graph



Fig. 10 Component Registration using Docker images

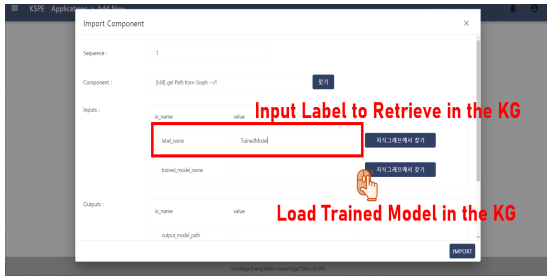


Fig. 11 Application Registration using Components

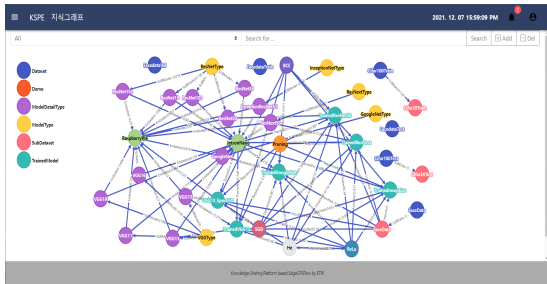


Fig. 12 Knowledge Sharing Graph

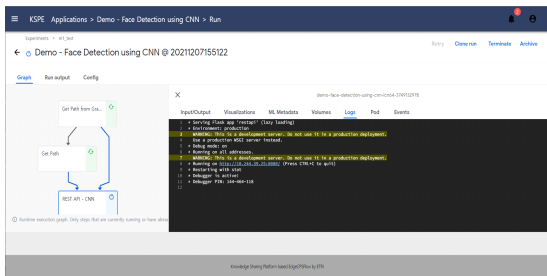


Fig. 13 pipeline and execution of AI Application

이렇게 생성된 컴포넌트 기반의 응용을 그림 13처럼 파이프라인화해서 실행을 시킬 수 있다. 그림의 왼쪽에 파이프라인화된 컴포넌트들 중에서 “Get Path from Graph”와 “Get Path”는 각각 지식 공유 그래프에서 학습된 모델을 가져와서 로딩을 하는 것이고, 로딩된 학습 모델로 실제 추론을 하는 부분은 “REST API-CNN” 부분이다. 이 부분은 데몬 형태로 계속 실행되는 부분으로써 외부의 입력 장치로부터 이미지가 들어오면 실시간으로 추론을 진행한다.

V. 결론

본 논문에서는 EdgeCPS 플랫폼에서 사용 가능하도록 컴포넌트 기반의 AI 응용 지원 시스템을 구축했다. 그리고 AI 응용을 보다 쉽게 생성할 수 있도록 AI 응용에 필수적인 요소인 학습데이터, 학습된모델, 학습알고리즘, 디바이스 등에 대한 정보를 그래프 기반의 정보로 제공함으로써, 사용자가 보다 손쉽게 필요한 정보를 사용할 수 있도록 제공했다. 향후과제로는 보다 많은 지식 공유 그래프를 구축하고, 사용자가 보다 다양한 응용을 생성해서 실행해 볼 수 있는 환경을 제공하고자 한다.

ACKNOWLEDGEMENT

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [21ZS1300, Research on High Performance Computing Technology to overcome limitations of AI processing]

References

- [1] TTA, Keyword: Edge Device. [Internet]. Available: <http://terms.tta.or.kr/main.do>.
- [2] H. J. Choi, B. G. Ko, J. S. Lee, E. S. Kang, J. O. Kim, and B. K. Lee, “Optimization function analysis for tower AI learning,” in *Proceeding of Journal of Korea Society of Computer Information*, Jeju, Korea, vol. 28, no. 2, pp. 351-353, 2020.
- [3] J. W. Park and Y. B. Ko, “A Lightweight CNN Algorithm based on Explainable AI for UWB Localization in Disaster Environments,” in *Proceedings of Korean Institute of Next Generation Computing*, Gwangju, Korea, pp. 30-33, 2021.
- [4] I. G. Chun, S. J. Kang, and G. J. Na, “EdgeCPS Technology Trend for Massive Autonomous Things,” *Electronics and Telecommunications Trends*, vol. 37, no. 1, pp. 32-41, Nov. 2021.
- [5] Y. J. Kim, I. G. Chun, S. J. Kang, G. N. Na, Y. Y. Kim, J. H. Jeon, and J. H. Lee, “A Design Plan for Constructing the Knowledge Sharing Middleware based on EdgeCPS for Harmonious Execution of AI Applications),” in *Proceeding of The 16th IEMEK Symposium on Embedded Technology*,

- Jeju, South Korea, pp. 210-213, 2021.
- [6] Kubernetes Federation [Internet]. Available:
<https://github.com/kubernetes-sigs/kubefed>.
- [7] Y. S. Kim and Y. H. Kim, "A Case Study of Orchestration for Kubernetes based Multi-Cluster," in *Proceeding of the KICS Winter Conference*, Pyeongchang, South Korea, pp. 153-154, 2021.
- [8] Kubeflow [Internet]. Available: kubeflow.org.
- [9] Y. J. Kim, G. J. Na, and I. G. Chun, "A Performance Test for Supporting Microservice of AI Applications," in *Proceeding of The 17th IEMEK Symposium on Embedded Technology*, Jeju, South Korea, pp. 307-310, 2022.



김영주(Young-Joo Kim)

1999 국립경상대학교 컴퓨터학과 졸업 (학사)
2001 국립경상대학교 컴퓨터학과 졸업 (석사)
2007 국립경상대학교 컴퓨터학과 졸업 (박사)
2007 ~ 2010 한국과학기술원 연구교수
2010 ~ 2012 Texas A&M University 연구원
2012 ~ 현재 한국전자통신연구원 (책임연구원)
※관심분야: 인공지능, 고신뢰 컴퓨팅, 임베디드 컴퓨팅, 병렬/분산처리