

OpenFaaS 기반 AI 분석 서비스 시스템 구축

Development of an AI Analysis Service System based on OpenFaaS

장래영, 이용, 박민우, 이상환
한국과학기술정보연구원

Rae-young Jang(raezero@kisti.re.kr), Ryong Lee(ryonglee@kisti.re.kr),
Min-woo Park(pminwoo@kisti.re.kr), Sang-hwan Lee(sanglee@kisti.re.kr)

요약

5G 네트워크와 사물인터넷 기술의 빠른 발전과 보급으로 빅 데이터 분석 기술 및 서비스 시스템에 대한 요구가 증가하고 있다. 특히, AI 기술 활용에 대한 폭발적인 수요 증가로 수집된 데이터에서 새로운 의미를 추출할 수 있는 머신/딥러닝 모델의 활용을 위한 경쟁이 치열해 지고 있다. 다양한 분야에서 AI 기술을 도입하기 위해서는 고성능 GPU를 탑재한 시스템을 구축하고 딥러닝 모델을 실행하기 위한 복잡한 설정을 할 필요가 있다. AI 기술을 활용하기 위해 소요되는 노력을 줄이기 위해, AIaaS 플랫폼은 사전준비과정 및 운영을 위한 복잡함을 클라우드 인프라에 감추고, AI 연구개발자들이 고성능 AI 분석기술을 쉽게 활용할 수 있게 하는 온라인 서비스로써 큰 주목을 끌고 있다. 본 논문에서는 딥 러닝 모델 등록부터 온라인 서비스 운영에 이르기까지 Docker 및 OpenFaaS 기반 AI 서비스 생성을 지원할 수 있는 새로운 AIaaS 시스템을 제안한다. 또한 제안 시스템에서 AI 서비스를 쉽게 생성, 활용하는 방법을 보여주는 사례 연구도 설명한다.

■ 중심어 : | AlaaS | OpenFaaS | 도커 | 딥러닝 | 데이터 분석 |

Abstract

Due to the rapid development and dissemination of 5G communication and IoT technologies, there are increasing demands for big data analysis techniques and service systems. In particular, explosively growing demands on AI technology adoption are also causing high competitions to take advantages of machine/deep-learning models to extract novel values from enormously collected data. In order to adopt AI technology to various research and application domains, it is necessary to prepare high-performance GPU-equipped systems and perform complicated settings to utilize deep learning models. To relieve the efforts and lower the barrier to utilize AI techniques, AlaaS(Al as a service) platform is attracting a great deal of attention as a promising on-line service, where the complexity of preparation and operation can be hidden behind the cloud side and service developers only need to utilize the high-level AI services easily. In this paper, we propose an AlaaS system which can support the creation of AI services based on Docker and OpenFaaS from the registration of models to the on-line operation. We also describe a case study to show how AI services can be easily generated by the proposed system.

■ keyword : | AlaaS | OpenFaaS | Docker | Deep Learning | Data Analysis |

* 본 연구는 한국과학기술정보연구원(KISTI) '연구데이터 공유·확산체제 구축(K-20-L01-C04-S01)' 과제를 통해 수행되었습니다.

* 본 논문은 ICC2019 국제학술대회 우수논문입니다.

접수일자 : 2020년 06월 12일

수정일자 : 2020년 06월 29일

심사완료일 : 2020년 06월 29일

교신저자 : 이용, e-mail : ryonglee@kisti.re.kr

I. 서론

5G 고속통신망 및 사물인터넷 기술의 빠른 발전과 보급은 다양한 형태의 빅데이터의 증가로 연결되고, 특히 IoT 관련 데이터들은 실시간 발생, 시공간 정보, 대용량의 특징을 갖는다. 수집되는 원시데이터는 정제되지 않아 사용에 어려움이 있기 때문에 전처리 과정과 이에 대한 분석 기술 및 응용 서비스에 대한 수요 또한 증가하고 있다. 특히, AI 기술을 활용한 데이터 분석에 대한 폭발적인 수요 증가와 함께 손쉽게 머신/딥러닝 모델 기반으로 빅데이터에 전처리 및 고급 분석을 통해 의미 있는 데이터를 얻기 위한 경쟁이 치열해지고 있다.

일반적으로 AI 기술을 활용하기 위해서는 고성능 GPU를 탑재한 시스템 인프라 구축부터 딥러닝 모델 등을 구동시켜야 하는 복잡한 과정까지 필요하기 때문에 요구되는 필요성에 비해 실제 AI 비전문 연구자 및 AI 연구개발자들이 손쉽게 다가가기 어려운 실정이다.

이러한 문제를 해결하기 위해 AI 모델을 손쉽게 OpenAPI 방식으로 서비스하는 기술이 주목 받고 있다. AIaaS(AI as a Service)는 AI 비전문 사용자도 API 형태로 제공되는 AI 분석서비스를 이용해 결과물을 활용하는 서비스를 제공할 수 있다. 일반적으로 AIaaS는 빠르고 안정적으로 AI 기술을 사용할 수 있게 해 준다는 장점이 있어 자체적으로 AI 기술을 개발할 역량이 부족한 연구자나 기업들에게 클라우드형 또는 설치형 AIaaS에 대한 관심이 높아지고 있다.

인프라 구축이 필요 없는 클라우드형 AIaaS로는 Amazon의 AWS(Amazon Web Service), Google의 GCP(Google Cloud Platform)가 대표적이다. AWS는 사용자가 직접 학습 모델들을 만들지 않고 인공지능 서비스를 사용하면서 기존의 머신러닝 과정을 획기적으로 줄일 수 있는 관리형 플랫폼 환경을 제공하고 있다. Amazon SageMaker, CodeGuru 등 AI 분석기술 개발서비스, Amazon Polly, Amazon Rekognition 등 AI 분석서비스를 제공한다. 구글의 GCP는 인공지능 분야 데이터 라벨링, 데이터 모델, AI 플랫폼 등 개발 서비스와 자연어 처리, 번역, Vision 인식, Video 동영상 분류 및 객체 추적 등 분석서비스를 제공한다.

그러나, AWS나 GCP의 AI분석기술은 개발환경과 분석 서비스를 제공하지만 자사 클라우드 서비스에 종속적이며 제한적이기 때문에 개발기술을 서비스로 배포하기에 어려움이 있고 새로운 개발환경에 익숙해져야 한다는 단점을 가지고 있다.

가트너의 2019년 발표보고서에 의하면 75%에 해당하는 기업이 AI가 새로운 비즈니스 창출에 기여한다고 응답했고, 84%에 해당하는 기업이 AI 기술은 경쟁력 강화에 도움을 주고 있다고 조사되었다. 그럼에도 불구하고 실제 AI 기술을 도입한 기업은 14%수준, 1년 이내 도입 계획이 있는 기업도 23%수준에 불과했다. 기업의 AI 기술에 대한 수요와 실제 AI 기술 도입 수준 차이는 추가적인 하드웨어 구매에 대한 부담과 전문인력 고용 및 교육에 따른 부담감으로 해석된다.

이를 해결하기 위해 AWS, Google Cloud 등 클라우드 형태의 AI 서비스가 이용되지만 보안의 문제, 맞춤형 분석기술 개발 요구 등의 이유로 설치 가능한 오픈소스 형태 AIaaS의 필요성이 요구된다. 이러한 설치형 AIaaS는 Apache OpenWhisk, OpenFaaS, bitnami Kubeless, Fn Project, Platform9 Fission 등이 공개되어 있지만 이를 AI 비전문가가 활용하는 것은 쉽지 않다.

본 논문에서는 오픈소스 형태의 AIaaS 프레임워크를 대상으로 인프라 자원에 비종속적이고 AI 연구개발자에게 친화적인 개발 구조를 통해 기존 문제점을 해결하기 위한 시스템을 제안한다. 특히, AI 분석기술 개발을 위한 주요 환경 제공과 AI 분석기술을 AIaaS형태로 제공하기 위해 서비스 등록 자동화를 위한 개발 파일 구조 규격화를 제안한다. 대형 클라우드 인프라에 비종속적이고 개발환경과 개발구조를 규격화하였기 때문에 관리자는 개발된 AI 분석 기술을 손쉽게 서비스로 등록하여 제공할 수 있다. 결과적으로, AI 연구개발자는 AI 모델 개발에 집중할 수 있고, AI 비전문 사용자는 AI 분석서비스를 OpenAPI 방식으로 활용해 원하는 연구나 서비스를 구축할 수 있다.

본 논문은 2장에서 관련연구, 3장에서 OpenFaaS 기반 AI 분석 서비스, 4장 AI 분석 기술 서비스 사례, 5장 기존 서비스 대비 제안 기술 순으로 소개하고 결론 및 향후 연구 제시로 끝을 맺는다.

II. 관련연구

1. 보편적인 의미의 AIaaS

일반적으로 AIaaS라 하면, AI 구현을 위한 H/W를 제공하는 IaaS, AI 개발에 필요한 개발툴을 지원하는 PaaS, 구현된 S/W를 제공하는 SaaS를 통칭하는 개념이다. 최근 AIaaS는 클라우드 기반으로 서비스되는 것이 주류인데 AWS, GCP가 대표적이다. AWS는 AI 솔루션 및 분석서비스를 제공하는데, AWS의 분석기술 개발을 위한 AI 솔루션은 다음과 같다.

표 1. AWS의 AI 주요 솔루션

솔루션명	기능
Amazon Kendra	검색 기능 제공
Amazon CodeGuru	머신러닝 코드 리뷰 자동화 및 애플리케이션 성능향상
Amazon FraudDetector	머신러닝 기반 자동 이상 거래 탐지 기능 제공
Amazon SageMaker	머신러닝 통합 개발환경

아마존 AWS는 복잡한 머신러닝 알고리즘이나 모델의 생성, 구성을 통한 사용 등을 고민할 필요 없이 다양한 AI 애플리케이션을 개발하도록 10여종의 서비스를 제공하며 주요 AI 분석 서비스는 다음과 같다.

표 2. AWS의 AI 주요 분석서비스

서비스명	기능
Amazon Polly	텍스트를 음성으로 변환
Amazon Rekognition	애플리케이션에 이미지와 비디오 분석 기능을 추가
Amazon Lex	음성 및 텍스트 이용에 대화형 인터페이스 구축
Amazon Transcribe	음성을 텍스트로 전환
Amazon Personalize	단순 API 호출로 맞춤형 추천 서비스를 구축

구글의 GCP는 인공지능 분야에서 데이터 라벨링, AI Platform을 솔루션으로 제공하고, 자연어처리, 번역, Vision API, Video Intelligence API를 서비스로 제공한다. 그 중 AI Platform은 AI Hub를 통해 개발자와 데이터 과학자 간 협업이 가능하도록 개발환경(Jupyter Notebook), VM Image, ML Container, 학습모델, AI Service 등 AI 분석기술에 필요한 대부분을

제공하며 주요 AI 분석 서비스는 다음과 같다.

표 3. GCP의 AI 주요 분석서비스

서비스명	기능
자연어처리	텍스트 문서 콘텐츠 분류, 분석, 인식
번역	100여개 이상 언어 간 번역 서비스
Vision	이미지를 분류하거나 객체를 감지 및 처리
Video Intelligence	동영상 객체와 동작 감지 및 처리

AWS는 SageMaker를 통해 개발환경을 제공하지만 지역 제한이 있어 국내에서는 이용하는데 불편함이 있고, GCP는 AI Hub의 Notebook을 통해 협업 개발환경을 제공하지만 클라우드 인프라를 사용해야 한다는 불편함이 존재한다. 또한, AWS와 GCP 모두 개발된 AI 분석기술을 서비스로 제공하려면 클라우드에 종속되어야 한다는 문제점이 있다.

본 연구에서는 이러한 문제점들을 해결하기 위해 Docker를 이용해 AI 연구개발자에게 동일한 개발환경을 제공하고자 하였고, OpenFaaS를 이용해 AI 비전문 사용자에게 AI 분석기술을 API 서비스 형태로 제공하는 AIaaS를 제안한다[1-3].

2. 동일한 개발환경 구현을 위한 Docker

도커는 컨테이너 기반의 오픈소스 가상화 플랫폼으로 다양한 프로그램, 실행환경을 컨테이너로 추상화하고 동일한 인터페이스를 제공하여 프로그램의 배포 및 관리를 단순하게 해준다[4].

가상화 기술 분야에서 CPU 가상화 기술을 이용한 전 가상화(Full-Virtualization) 방식의 KVM(Kernel-based Virtual Machine)과 반가상화(Para-Virtualization) 방식의 Xen 등장은 주로 사용되었던 VirtualBox같은 OS 가상화방식에 비해 성능을 향상시켰다. 그러나, 전 가상화든 반가상화든 추가적인 OS를 설치하여 가상화하는 방법은 결과적으로 성능 문제가 있었고 이를 개선하기 위해 프로세스를 격리하는 방식이 등장하였다. 이러한 방식을 리눅스에서는 컨테이너라고 하는데 단순히 프로세스를 격리시키기 때문에 가볍고 빠르게 동작한다. CPU나 메모리는 프로세스가 필요한 만큼만 추가로 사용하고 성능 상 손실도 거의 없다.

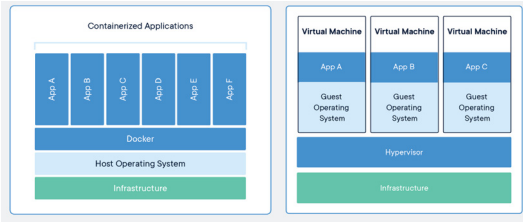


그림 1. 도커와 가상머신 비교

하나의 서버에 여러 개의 컨테이너를 실행해도 서로 영향을 미치지 않고 독립적으로 실행되어 마치 가벼운 VM(Virtual Machine)을 사용하는 수준이며 실행 중인 컨테이너에 접속하여 명령어를 입력할 수 있고 추가적인 패키지를 설치할 수 있으며 여러 개의 프로세스를 백그라운드로 실행할 수도 있다. 컨테이너 별로 CPU나 메모리 사용량을 제한할 수 있고 호스트의 특정 포트와 연결하거나 호스트의 특정 디렉토리를 내부 디렉토리인 것처럼 사용할 수 도 있다[5].

이미지는 컨테이너와 함께 도커에서 가장 중요한 개념 중 하나로 컨테이너 실행에 필요한 파일과 설정 값 등을 포함하고 있는 것을 말한다. 상태 값을 가지지 않고 변하지도 않는다. 컨테이너는 이미지를 실행한 상태라고 볼 수 있고 추가되거나 변하는 값은 컨테이너에 저장된다. 하나의 이미지에서 여러 개의 컨테이너를 생성할 수 있고 컨테이너의 상태가 바뀌거나 컨테이너가 삭제되더라도 이미지는 변하지 않고 원본 그대로 남는다. 이미지는 컨테이너를 실행하기 위한 모든 정보를 가지고 있다. DockerFile은 이런 이미지를 생성할 수 있게 하는 명령어를 모아둔 것으로 이미지에 설치해야 하는 패키지, 추가해야 하는 소스코드, 실행해야 하는 명령어와 셸 스크립트 등을 하나의 파일로 기록해 둔 것이다. DockerFile을 이용하면 직접 컨테이너를 생성하고 이미지로 변경해야하는 번거로움을 덜고 Git과 같은 개발도구를 통해 애플리케이션의 빌드 및 배포를 자동화할 수 있다.

AI 분석기술 개발 시 AI 연구개발자와 분석기술 내용에 따라 개발환경이 다를 수 있는 문제점이 있어 개발 후 서비스 등록 진행 시 개발환경의 차이로 인해 정상적인 동작이 어려운 불편함이 있다. Docker는 AI 서비스 확장성을 고려하여 다양한 개발환경의 이질성을 극

복할 수 있는 가상개발환경을 제공해줄 수 있다.

3. 서버리스컴퓨팅을 위한 OpenFaaS

OpenFaaS는 Docker/Kubernetes 기반 Serverless Function Service를 가능하게 하는 오픈소스 프레임워크다. 개발자로 하여금 단순 반복 작업을 줄이고 이벤트 중심 함수 서비스를 쉽게 배포할 수 있게 해준다. Docker ReplicaSet 기능과 연계하여 자동 확장이 가능하고 DockerFile을 사용하여 유지보수를 쉽게 할 수 있다[6].

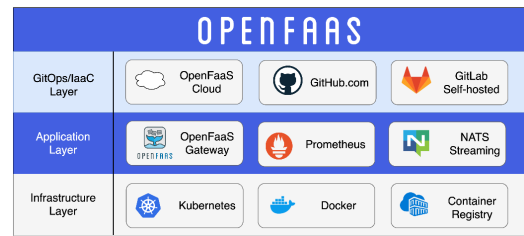


그림 2. OpenFaaS Stack&Layer

서버리스 컴퓨팅의 구조를 살펴보면 Function을 호출하기 위한 Trigger, Function이 구동되는 환경 FaaS, 처리 결과를 저장하거나 알리기 위한 BaaS (Back as a Service) 등 3가지로 구성되어 있다.

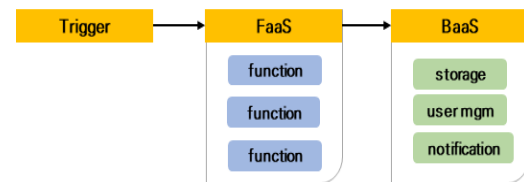


그림 3. 서버리스 구성요소

상용 클라우드 서버리스컴퓨팅은 대부분 위의 3가지 (Trigger, FaaS, BaaS) 요소를 모두 갖추고 있어 플랫폼으로써 다양한 기능을 제공하여 쉽게 서비스를 생성하고 관리할 수 있지만, 제공되는 서비스 공급자에 종속되는 경향이 있다. 따라서 이를 탈피하기 위해 자체 서버리스 환경을 구축하고자 하는 사용자를 위한 오픈소스 서버리스 프레임워크들이 발표되었다.

본 논문에서는 개발된 AI 분석기술을 OpenFaaS를 이용하여 OpenAPI형태로 빠르게 서비스할 수 있게 하였고, AI 분석기술 서비스 등록 자동화를 위해 컨테이너 내부 파일 구조의 규격화 및 서비스 설계를 최적화하여 제공한다.

III. OpenFaaS 기반 AI 분석 서비스

1. AI 분석 서비스 요구 사항 분석

IoT 빅데이터 시대 늘어나는 AI 기반 데이터 분석 수요에 대응하기 위해 KISTI 연구데이터공유센터에서는 IoTDA라는 사물인터넷 데이터 수집/분석 시스템을 개발하고 있다[7-10]. 본 시스템은 사물데이터를 활용하는 연구자 및 서비스 개발자들이 손쉽게 다양한 사물데이터를 수집, 검색, 공유하고, 필요시에 AI 기반으로 사물데이터를 분석해주는 AIaaS 서비스 기능을 포함하고 있다[11-13].

특히, 증가하는 AI 모델의 다양성과 웹 기반 서비스 제공에 대한 요구에 대응하기 위해 다음과 같은 전략을 취하고 있다.

- 1) **AI 서비스 확장성:** 머신/딥러닝 모델을 개발하는 AI 연구개발자들은 독자적인 하드웨어 및 소프트웨어 환경에서 개발에 집중하며 개발 후 공개 시 소스코드와 학습된 모델을 제공한다. 따라서 연구자들의 다양한 개발환경의 이질성을 흡수하고, 서비스를 계속 확장시켜 가기 위해 Docker 기반 가상화를 채택한다.
- 2) **웹 서비스화 용이성:** AI 연구개발자가 개발한 분석기술을 빠르게 서비스화 하기 위해 Docker 내 파일 구조를 규격화하고, OpenFaaS 기반으로 OpenAPI까지 손쉽게 생성할 수 있도록 한다.

2. Docker 기반 AI 동작 환경의 가상화

머신/딥러닝 모델을 개발하는 AI 분석기술은 AI 연구개발자와 기술내용에 따라 다양한 환경에서 개발되고 있다. 하드웨어 측면에서 GPU 사용유무와 드라이버 운용, 소프트웨어 측면에서 개발언어 및 사용 라이브러리 등이 무엇인지에 따라 개발환경이 달라진다. 어떤

AI 연구개발자는 ubuntu운영체제에서 python을 기반으로 개발하고, 다른 AI 연구개발자는 Jupyter Notebook에서 웹으로 개발할 수도 있다.

이러한 다양한 개발환경의 이질성을 동일하게 맞추기는 불가능하지만 Docker를 이용하면 주요 개발환경을 DockerFile과 이미지로 제공할 수 있고, AI 연구개발자가 제공이미지를 통해 AI 개발환경을 생성한다면 AI 동작 환경의 편의화를 줄일 수 있다. 이를 위해 본 논문에서는 [표 4]의 하드웨어 환경에서 동작하는 DockerFile과 이미지를 제공한다. 제공되는 주요 AI 개발환경은 다음 [표 5]와 같다.

표 4. 기본 이미지 하드웨어 환경 정보

운영체제	개발언어	GPU	CUDA	CUDNN
ubuntu18.04_64	python3.6.8	nVidia Titan XP	9.0	7.1.4

표 5. 주요 개발환경 제공 이미지 일부

구분	베이스이미지	적용라이브러리	비고
darknet	dev_base	darknet	
yolo34gpu	dev_base	oepncv, yolo34gpu	
tensorflow	dev_base	tensorflow	
opencv_rc	dev_opencv	torch, scikit-image, scipy	객체인식
opencv_ln	dev_opencv	numpy, cython, yolo34	차선검출및 차량계수
darknet_tf	darknet	tensorflow, keras, matplotlib	개인정보 익명화

AI 연구개발자는 개발에 필요한 라이브러리에 따라 이미지를 선택하여 사용하거나 또는 DockerFile을 수정하여 라이브러리를 추가하거나 변경하여 사용할 수 있다. 개발하는 분석기술에 맞는 모든 개발환경을 이미지로 제공할 수 없기 때문에 주요 개발환경은 이미지로 제공하고 AI 연구개발자가 수정하여 사용한 이미지는 필요에 따라 이미지 저장소에 반영되어 재사용할 수 있게 한다.

제공되는 AI 개발환경을 활용하면 AI 연구개발자들은 개발에만 집중할 수 있어 효율적이고 관리자는 이미지 단위로 AI 분석기술을 관리할 수 있어 AI 서비스 확장성에서 용이하다.

3. AI 서비스 워크플로우

본 절에서는 제안구조를 적용한 AI 연구개발자의 분석기술 개발 및 관리자의 서비스등록 업무, AI 비전문 사용자의 서비스 활용에 관한 전체 구조를 설명한다. 제안기술 적용 전체 구조도는 [그림 4]와 같으며 단계별 설명은 다음과 같다.

- ① AI 연구개발자는 제공되는 DockerFile과 이미지로 개발환경을 구성하고, 필요 시 라이브러리 추가 등 세부작업을 진행하여 새로운 개발환경을 구성한다. KISTI는 기본 사용법을 안내한다.
- ② AI 연구개발자는 이미지를 통해 생성된 컨테이너 개발환경에서 분석기술을 개발하고, AI 분석기술 개발 완료 시 이를 제공한다.
- ③ 관리자는 AI 분석기술이 포함된 이미지를 제공받아 확인한다.
- ④ 관리자는 필요 시 개발에 사용된 분석기술 이미지를 Docker Pool에 포함시켜 재사용할 수 있다.
- ⑤ 관리자는 기능이 확장된 OpenFaaS를 통해 생성된 프로젝트로 제공받은 AI 분석기술 이미지를 OpenAPI 형태로 서비스할 수 있게 등록한다.
- ⑥ AI 비전문 사용자는 필요한 AI 분석서비스를 선택하여, 원시데이터를 분석할 수 있고 결과물은 JSON 형태로 전달받아 연구에 활용할 수 있다.

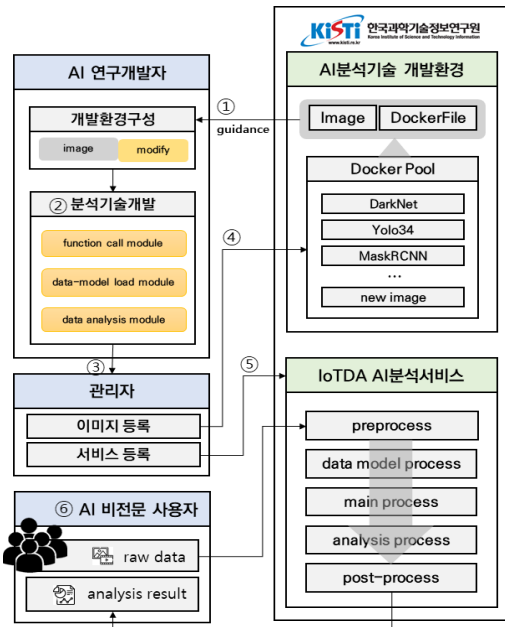


그림 4. 제안기술 전체구조도

4. AI 서비스 등록 자동화

개발이 완료된 AI 기술을 서비스로 제공하려면, OpenAPI 서비스로 등록하는 과정이 필요하다. 앞서 실제 AI 분석기술을 개발하는 개발환경과 서비스하는 인프라 환경의 이질성은 제공하는 개발환경 이미지를 통해 해결하였으나 개발자에 따라 개발환경의 파일구조는 다를 수 밖에 없다. AI 연구개발자와 관리자가 다른 경우 개발된 AI 분석기술이 실제 서비스로 제공되기 위해서는 상당한 시행착오가 발생한다. 등록과정은 다음과 같은 순서로 진행된다.

- 1) 개발환경 확인 및 Docker 이미지 생성
- 2) 개발된 AI 기술 개발코드 분석
- 3) OpenFaaS 프로젝트 생성 및 개발코드 적용
- 4) 데이터모델 적재여부에 따른 개발코드 수정
- 5) 서비스 동작 확인 및 배포

수행해야 하는 등록 작업은 개발된 기술에 따라 과정이 복잡하고 장애요소가 다양하다. 이는 AI 연구개발자들의 개발코드를 빠르게 서비스화 하는데 문제가 된다. 관리자는 개발코드 및 파일구조의 분석이 어려워 하나의 분석기술을 서비스로 등록하기 위해 많은 노력과 시간, 비용이 소모된다. 따라서, 분석기술 개발 초기단계부터 정형화된 파일 구조를 지정하여 AI 분석기술 서비스 등록 자동화를 지향하고자 하였다.

본 논문에서는 AI 분석기술을 빠르게 서비스할 수 있는 AI 서비스 등록 자동화를 위해 다음 사항을 제안한다.

- 데이터모델 시스템 상주 형 구조 설계
- 폴더 및 파일 구조 설계

KISTI는 제안 사항을 적용하여 Docker 내 파일구조를 규격화하고 DockerFile 및 이미지를 제공하며 AI 연구개발자가 구조를 확인하여 개발할 수 있게 사용법을 제공한다.

OpenFaaS는 Docker를 기반으로 하여 개발된 분석기술을 OpenAPI형태로 제공할 수 있게 하는 오픈소스 프로젝트이며, 본 논문에서는 OpenFaaS를 제안에 맞게 최적화하여 사용하였다. OpenFaaS는 faas-cli 명령어를 통해 서비스를 생성(new)하고, 구축(build)하고, 배포(deploy)한다. AI 연구개발자가 OpenFaaS 내부의 과정을 무시하고 개발한다면 서비스를 등록해야

하는 관리자는 일일이 파일 구조와 소스코드를 수정해야 한다. 반대로 AI 연구개발자가 OpenFaaS 구조를 전부 이해해야 한다면 개발기간은 길어지게 될 것이다. 그렇기 때문에 본 논문에서는 최소한의 파일 구조 규격화를 통한 제한을 두고자 하였다.

단순히 개발환경을 제공하고, 제공한 개발환경으로 생성되는 컨테이너의 내부 파일 구조를 일부 규격화하는 것만으로 DevOps처럼 관리자가 개발기술을 배포하고 AI 비전문 사용자가 활용할 수 있다. 소프트웨어적인 측면에서만 본다면, 단순히 데이터모델을 적재하고, 분석하는 기능만 필요하기 때문에 기능을 모듈 단위로 개발하여 불러와 사용할 수 있게 다음과 같은 제안을 할 수 있다.

OpenAPI 방식의 서비스를 하기 위해 본 논문에서는 데이터모델을 적재하는 방식을 제안하고, 적합한 파일 구조로 제한하였다. 분석기술 개발은 데이터 모델을 적재하는 load 모듈과 데이터를 분석하는 work 모듈로 분리하여 개발할 것을 제안하며, API 서비스를 고려한 AI 분석기술 개발 구조는 다음 그림과 같다.

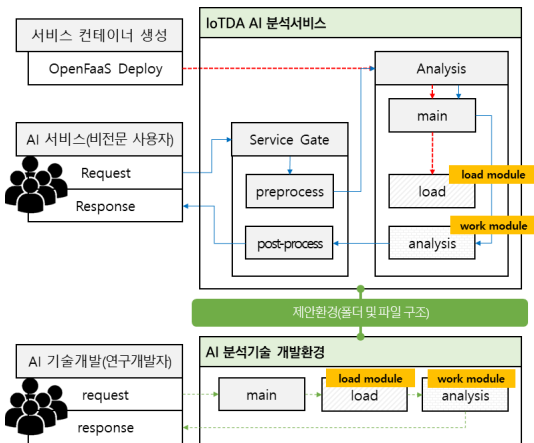


그림 5. 모듈 개발 구조

AI 연구개발자는 기술개발 시 데이터모델을 적재하는 기능을 load 모듈로, 분석기능을 work 모듈로 개발하면 OpenFaaS를 이용해 AI 분석서비스 생성 시 정해진 위치에서 개발된 모듈만 불러들여 사용할 수 있다.

AI 연구개발자가 분석기술을 개발할 때는 일반적으로 데이터모델은 상주하지 않고 필요 시 적재하여 분석하는 방법으로 주로 개발하는데, OpenAPI로 제공하고 할 때는 빠른 분석 결과 제공을 위해 데이터모델은 메모리에 적재하는 형태로 제안한다. 이를 위해 전체 소스코드를 수정할 필요 없이 주어진 구조로 모듈을 개발하고 AI 분석서비스에서는 모듈구조를 가져와 사용할 수 있게 하였다.

제안 파일구조가 적용된 DockerFile을 AI 연구개발자에게 제공하여 정해진 위치의 주어진 모듈만 개발하고 관리자는 개발된 AI 분석기술을 최소한의 수정과정을 거쳐 손쉽게 등록할 수 있게 하였다. 관리자는 AI 분석기술을 서비스 등록하기 위해 다음 과정만 진행하게 된다.

- 1) 개발된 분석기술 이미지 또는 DockerFile 확인
- 2) 개발된 AI 기술과 OpenFaaS 프로젝트 결합
- 3) DockerFile 수정 및 서비스 구동

IV. AI 분석기술 서비스 사례

본 장에서는 KISTI의 IoTDA 플랫폼에서 AIaaS로 제공하는 AI 분석기술 중 IoTDA Image Recognition 서비스를 사례로 제안 구조를 확인한다. AI 분석기술은 기술개발, 등록 과정을 거쳐 서비스로 제공된다.

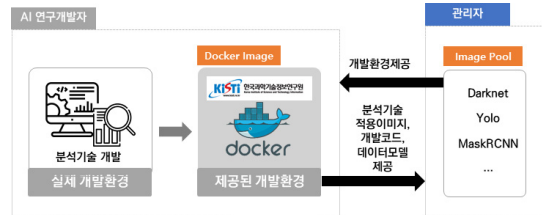


그림 6. AI 분석 기술 개발 사례

AI 연구개발자는 분석기술 개발에 맞춰 KISTI가 제공하는 Docker 이미지를 참조하여 개발환경 컨테이너를 구축할 수 있다. 컨테이너는 개발 라이브러리 설치 및 파일 구조가 설정되어 있다. AI 연구개발자는 분석기술 개발 후 이미지로 생성하여 도커 이미지와 개발

문서, 데이터모델 등을 내부 양식에 맞춰 관리자에게 전달한다.

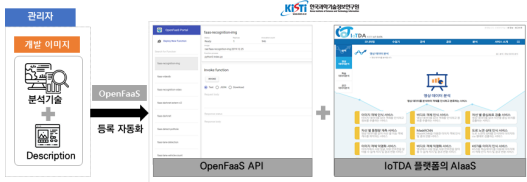


그림 7. AI 분석 기술 등록 사례

관리자는 분석기술 개발 결과물을 OpenFaaS의 프로젝트를 생성하여 IoTDA 플랫폼의 AlaaS로 등록한다. 분석기술은 API 형태로 등록되고 서비스 별로 분류되어 AI 비전문 사용자에게 공개된다. 등록 과정에서 알 수 있듯이 관리자는 개발환경을 신경 쓰지 않아도 되고, 규격화된 구조에 맞게 도커 이미지 내 소스 파일과 학습데이터 파일이 존재하는지 확인하고, 새로운 FaaS 프로젝트를 생성하고, 구축, 배포까지 한 번에 진행할 수 있게 등록 절차가 간소화되었다.

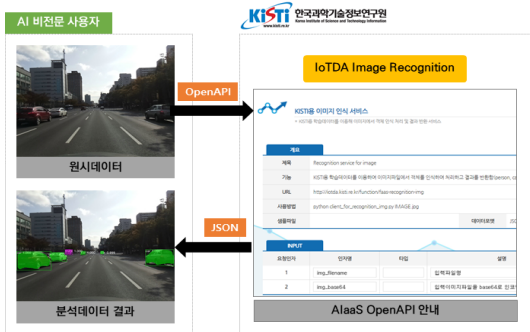


그림 8. AI 분석 서비스 사례

AI 비전문 사용자는 각 서비스 별 OpenAPI 사용 방법 안내를 통해 개요, 기능, 서비스 호출 주소, 필수 요청인자, 출력변수 및 예시를 제공받아 이용하고자 하는 서비스의 API 사용방법에 따라 원시데이터를 대상으로 서비스를 호출하여 분석결과를 구해 연구에 활용할 수 있다.

사례에서 보여지는 IoTDA Image Recognition 서비스는 대전시 도로영상데이터를 바탕으로 학습데이터

모델을 제작하여 승용차, 버스, 트럭, 사람, 오토바이 등 5종 객체를 식별하도록 개발한 분석서비스이다. 사용된 이미지는 1초 간격으로 촬영된 도로영상 이미지파일 100장 기준이며, 기본해상도는 640x480이다. 제안구조에 따라 분석을 진행하였을 때 일반연구자가 분석결과를 되돌려 받을 수 있는 평균 분석 시간은 장당 0.4초 내외였다. 해상도에 따른 평균 분석 시간은 다음과 같았다. 일반연구자 환경을 고려하여 진행하였기 때문에, 원시데이터 크기에 따라, 네트워크 사정에 따라 API방식으로 원시데이터를 전달하고 전달받는 시간까지 포함된 결과이다. 해당 서비스 사례 분석결과에 따르면 FHD 해상도 이미지를 1초 이내 촬영하였을 경우 실시간 분석서비스가 가능한 것으로 기대할 수 있다.

표 6. IoTDA Image Recognition 분석서비스 성능

해상도	640x480(VGA)	1920x1080(FHD)	3840x2160(4K)
평균파일크기	80 KB	556 KB	2194 KB
평균분석시간	0.5418 sec	0.7010 sec	4.8029 sec

V. 기존 서비스 대비 제안 기술 장점

제안기술은 AWS, GCP 등 기존 클라우드형 AlaaS와 다음과 같은 차이를 갖는다.

표 7. 클라우드형 AlaaS와 제안 기술 비교

	AWS	GCP	제안기술
개발환경 파편화	낮음	없음	낮음
인프라 종속성	높음	높음	낮음
분석기술 자유도	낮음	낮음	높음

개발환경에서 AWS는 머신러닝을 위한 이미지를 제공하고 있고, GCP는 AI Hub의 Jupyter-Notebook으로 웹에서 협업이 가능하게 제공하고 있다[14][15]. 제안기술은 DockerFile과 이미지를 통해 개발환경 초기 구성의 편리함과 최적화를 제공한다.

기존 기술은 클라우드 인프라에 종속되어 개발 및 서비스에 제한이 있었지만 제안기술은 설치형 기술로 인프라 자원에 상대적으로 종속이 낮다.

제공되는 분석기술에서 AWS와 GCP는 다양한 종류

의 분석 서비스를 제공하고 있지만 수정이 불가능하고 개발에 제약사항이 많아 맞춤형 분석기술 개발 및 서비스가 어렵다. 제안기술은 영상데이터 뿐만 아니라 다양한 수요에 맞춰 AI 연구개발자가 분석 기술을 개발하고 손쉽게 서비스로 제공할 수 있어 자유도가 높다.

이런 차이에 따라 제안 기술은 요구되는 다양한 분석 기술을 어느 환경에서나 Docker 기반으로 개발하고 관리자의 개입을 최소화하여 FaaS 형태로 서비스를 등록하고 제공할 수 있다. AI 연구개발자가 분석기술을 서비스로 제공하고자 할 때 기존 클라우드형 AIaaS의 제한을 해결하고 물리적인 시간의 소요없이 즉시 서비스할 수 있으며 기존 기술과 연계를 통해 새로운 수요를 충족시킬 수 있는 가능성을 갖는다.

VI. 결론

본 논문에서는 AI 기술에 대한 전문지식이 없어도 OpenAPI 방식으로 AI 분석서비스를 활용할 수 있는 AIaaS를 소개하고, KISTI에서 제공 중인 AI 분석서비스와 AIaaS를 구축하기 위해 고려한 AI 서비스 확장성과 분석기술 웹 서비스 용이성 등 제안 사항을 확인하였다.

AWS, GCP와 다르게 본 연구기술은 AI 분석기술 수요에 맞춰 AI 비전문 사용자 중심으로 진행되고 실제 AI분석기술을 자세히 알지 못하더라도 OpenAPI로 제공되는 사용법에 따라 필요한 서비스를 제공받아 활용할 수 있게 하며, AI 연구개발자의 분석기술 개발부터 관리자의 서비스 등록까지 자동화할 수 있는 구조를 제안한다.

AI 서비스 확장성을 위해 Docker기반 개발환경 이미지 제공으로 AI 연구개발자가 모델 개발에 집중할 수 있게 하였고, 관리자의 업무를 최소화하였으며, AI 비전문 사용자의 AI 분석서비스 활용을 손쉽게 하였다.

본 연구의 접근 방식은 이기종 데이터 및 서비스를 처리하기 위해 많은 스마트 응용 프로그램에 필요한 서비스 확장성 측면에서 장점이 있다. 기존 분석기술 개발에서 실제 서비스로 진행하기 위해 발생했던 문제점들을 해결하고 서비스 등록 과정을 원활하게 하여 새로

운 AI 분석기술 개발에 의한 확장이 가능하게 된다. 마지막으로, AI 분석기술 사례를 소개하고, AI 분석기술 서비스 성능을 확인하였다.

본 연구에서는 일부 다음과 같은 한계점을 갖는다. 개발환경을 이미지로 제공하여 개발환경 구축에 소모되는 불필요한 자원을 줄임으로써 분석기술 개발에 집중하게 하나 AI 연구개발자가 요구하는 모든 개발환경을 전부 제공하지 못하고 있다. 이는 Dockerfile의 구성을 조합하여 다양한 요구사항에 대응하고자 한다. 또한, 제안기술에 적용된 데이터 모델 적재 형태는 대량의 서비스 요청에 유연하게 대응하기 어렵고 시스템 자원 소모가 높은 문제점을 갖고 있으나 서비스 요청 형태에 따른 데이터 모델 적재 방식을 개선하여 해결하고자 한다.

향후 본 연구는 스마트 시티 애플리케이션의 필수 데이터 분석에 중점을 둔 서비스 풀 향상을 위한 AI 분석 서비스 간 기술 연동 및 성능에 중점을 두고 데이터 분석을 위한 IoTDA 플랫폼 고도화로 진행할 계획이다.

참고 문헌

- [1] V. Ishakian, V. Muthusamy, and A. Slominski, "Serving Deep Learning Models in a Serverless Platform," 2018 IEEE International Conference on Cloud Engineering (IC2E). IEEE, 2018.
- [2] K. Y. Lee and N. Ha, "AI Platform to Accelerate API Economy and Ecosystem," 2018 International Conference on Information Networking (ICOIN), IEEE, 2018.
- [3] E. Bisong, "An Overview of Google Cloud Platform Services." *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress, Berkeley, CA, pp.7-10, 2019.
- [4] Docker, <https://docs.docker.com/>
- [5] 배유미, 정성재, 소우영, "웹 서버 구성을 통한 가상머신과 컨테이너 방식 비교 분석," 한국정보통신학회논문지, 제18권, 제11호, pp.2670-2677, 2014.
- [6] OpenFaaS, <https://docs.openfaas.com/>
- [7] 이용, 박민우, 장래영, 이상환, "KISTI Vehicle-based IoT Dataset: Sensing a City with Vehicles," 한국

콘텐츠학회 ICCC 논문집, pp.223-224, 2019.

- [8] M. Park, R. Lee, R. Y. Jang, and S. H. Lee, "KISTI Vehicle-Based Urban Sensing Dataset," The First International Workshop on Research Data Analysis and Utilization (RDAU2020), pp.601-604, 2020.
- [9] R. Lee, R. Y. Jang, M. Park, G. Y. Jeon, J. K. Kim, and S. H. Lee, "Making IoT Data Ready for Smart City Applications," The First International Workshop on Research Data Analysis and Utilization (RDAU2020), pp.605-608, 2020
- [10] R. Lee, M. Park, and S. H. Lee, "An Advanced IoT Data Collection Service for Data-centric Smart Cities," International Journal of Innovative Technology and Exploring Engineering, Vol.8, Issue.2, 2019.
- [11] 이용, 박민우, 장래영, 이상환, "IoTDA: An Integrated IoT Data Service Platform for Smart City Applications," 한국콘텐츠학회 ICCC 논문집, pp.49-50, 2019.
- [12] 장래영, 이용, 박민우, 이상환, "An IoT Data Analysis Service based on OpenFaaS," 한국콘텐츠학회 ICCC 논문집, pp.225-226, 2019.
- [13] 이용, 전가예, 김재광, 최인빈, 이상환, "A Case Study: Road Traffic Analysis on KISTI IoTDA Platform," 한국콘텐츠학회 ICCC 논문집, pp.227-228, 2019.
- [14] AWS, <http://aws.amazon.com/>
- [15] GCP, <http://cloud.google.com/>

저 자 소 개

장 래 영(Rae-Young Jang)

정회원



- 2018년 8월 : 한남대학교 컴퓨터공학과(공학박사)
- 2019년 9월 ~ 현재 : 한국과학기술정보연구원 연구데이터공유센터 박사후연구원

<관심분야> : DevOps, Docker, Kubernetes, 인공지능 등

이 용(Ryong Lee)

정회원



- 2003년 : 일본교토대학교 사회정보학과(박사)
- 2011년 ~ 2013년 : 일본NICT연구소 연구원
- 2013년 9월 ~ 현재 : 한국과학기술정보연구원 연구데이터공유센터 책임연구원

<관심분야> : 공간데이터, 사물인터넷, 스마트시티, 인공지능 등

박 민 우(Min-Woo Park)

정회원



- 2004년 : 충남대학교 컴퓨터과학과(이학석사)
- 1996년 8월 ~ 현재 : 한국과학기술정보연구원 연구데이터공유센터 책임연구원

<관심분야> : 시스템 아키텍처, 정보보안, 인공지능 등

이 상 환(Sang-Hwan Lee)

정회원



- 2018년 : 서울시립대학교 컴퓨터공학과(공학박사)
- 1995년 4월 ~ 현재 : 한국과학기술정보연구원 연구데이터공유센터 책임연구원

<관심분야> : 빅데이터 분석, 데이터 생태계, 인공지능 등