

Deep Learning Description Language for Referring to Analysis Model Based on Trusted Deep Learning

Jong Hyeok Mun[†] · Do Hyung Kim^{††} · Jong Sun Choi^{†††} · Jae Young Choi^{††††}

ABSTRACT

With the recent advancements of deep learning, companies such as smart home, healthcare, and intelligent transportation systems are utilizing its functionality to provide high-quality services for vehicle detection, emergency situation detection, and controlling energy consumption. To provide reliable services in such sensitive systems, deep learning models are required to have high accuracy. In order to develop a deep learning model for analyzing previously mentioned services, developers should utilize the state of the art deep learning models that have already been verified for higher accuracy. The developers can verify the accuracy of the referenced model by validating the model on the dataset. For this validation, the developer needs structural information to document and apply deep learning models, including metadata such as learning dataset, network architecture, and development environments. In this paper, we propose a description language that represents the network architecture of the deep learning model along with its metadata that are necessary to develop a deep learning model. Through the proposed description language, developers can easily verify the accuracy of the referenced deep learning model. Our experiments demonstrate the application scenario of a deep learning description document that focuses on the license plate recognition for the detection of illegally parked vehicles.

Keywords : Trusted Deep Learning, Model Reference, Deep Learning Description Language, Traffic Situation Analysis Model

신뢰성있는 딥러닝 기반 분석 모델을 참조하기 위한 딥러닝 기술 언어

문종혁[†] · 김도형^{††} · 최종선^{†††} · 최재영^{††††}

요약

최근 딥러닝은 하드웨어 성능이 향상됨에 따라 자연어 처리, 영상 인식 등의 다양한 기술에 접목되어 활용되고 있다. 이러한 기술들을 활용해 지능형 교통 시스템(ITS), 스마트홈, 헬스케어 등의 산업분야에서 데이터를 분석하여 고속도로 속도위반 차량 검출, 에너지 사용량 제어, 응급상황 등과 같은 고품질의 서비스를 제공하며, 고품질의 서비스를 제공하기 위해서는 정확도가 향상된 딥러닝 모델이 적용되어야 한다. 이를 위해 서비스 환경의 데이터를 분석하기 위한 딥러닝 모델을 개발할 때, 개발자는 신뢰성이 검증된 최신의 딥러닝 모델을 적용할 수 있어야 한다. 이는 개발자가 참조하는 딥러닝 모델에 적용된 학습 데이터셋의 정확도를 측정하여 검증할 수 있다. 이러한 검증을 위해서 개발자는 학습 데이터셋, 딥러닝의 계층구조 및 개발 환경 등과 같은 내용을 포함하는 딥러닝 모델을 문서화하여 적용하기 위한 구조적인 정보가 필요하다. 본 논문에서는 신뢰성있는 딥러닝 기반 데이터 분석 모델을 참조하기 위한 딥러닝 기술 언어를 제안한다. 제안하는 기술 언어는 신뢰성 있는 딥러닝 모델을 개발하는데 필요한 학습데이터셋, 개발 환경 및 설정 등의 정보와 더불어 딥러닝 모델의 계층구조를 표현할 수 있다. 제안하는 딥러닝 기술 언어를 이용하여 개발자는 지능형 교통 시스템에서 참조하는 분석 모델의 정확도를 검증할 수 있다. 실험에서는 제안하는 언어의 유효성을 검증하기 위해, 번호판 인식 모델을 중심으로 딥러닝 기술 문서의 적용과정을 보인다.

키워드 : 신뢰성있는 딥러닝, 모델 참조, 딥러닝 기술 언어, 교통상황 분석 모델

1. 서론

딥러닝은 기계학습의 한 분야로서 컴퓨터가 데이터에 대한

패턴을 학습하기 위한 기술로[1], 하드웨어의 성능 향상으로 자연어 처리, 영상 인식 등의 기술 분야와 지능형 교통 시스템, 스마트홈, 헬스케어 등의 산업분야에서 활용한다[2-7].

딥러닝 기술을 활용하는 산업 분야 중, 지능형 교통 시스템(Intelligent Transportation System, 이하 ITS)은 교통 수단 및 시설에 첨단 기술을 접목하여 지능형 교통 서비스를 제공하는 자동화 교통 시스템이다[8-9]. 최근에 ITS는 딥러닝 기술을 활용한 차량 이미지 분석을 수행하는 교통상황 분석 모델을 통해 차량 및 차량 번호와 같은 교통상황 정보를

* 이 논문은 2019년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구 사업임(No.2019R1A2C1007861).

[†] 준 회원 : 송실대학교 컴퓨터학과 박사과정

^{††} 비 회원 : 송실대학교 컴퓨터학과 석사과정

^{†††} 정 회원 : 송실대학교 컴퓨터학부 부교수

^{††††} 종신회원 : 송실대학교 컴퓨터학부 교수

Manuscript Received : October 28, 2020

Accepted : November 29, 2020

* Corresponding Author : Jong Sun Choi(jongsun.choi@ssu.ac.kr)

수집하여 사고 판독, 불법주차 단속, 과속 차량 검출, 스마트 요금징수 등과 같은 자동화 교통 서비스를 통해 교통의 편리성을 제공하고 있다[10-11].

다른 산업분야로, 스마트홈은 딥러닝과 IoT 기술을 결합하여 사용자의 행동 및 사용 패턴에 따라 IoT 센서를 통해 에너지 사용량 제어, 온도 제어, 보안, 모니터링 등의 맞춤형 스마트 제어 서비스를 제공한다[12-13]. 이러한 데이터 분석을 통한 서비스에서 결함이 발생할 경우 지능형 교통 시스템에서는 부정확한 교통 정보 수집으로 인한 위법차량 및 통행 요금의 미징수 등의 비용 문제로 이어질 수 있고, 스마트홈 분야에서는 센서의 오작동으로 인한 전력 낭비, 화재 등의 문제가 발생할 수 있다. 그러므로 시스템에 적용되는 딥러닝 모델은 신뢰성이 요구된다. 여기서 딥러닝 모델의 신뢰성은 딥러닝 모델의 결과에 대해 이해하는 것을 중점으로 하는 해석적 측면과 딥러닝 모델의 예측, 인지 결과의 정확도와 같은 성능적 측면으로 구분될 수 있다[14]. 딥러닝 모델을 참조하고자 할 때, 참조하는 딥러닝 모델은 안정적인 성능을 보장해야 하며, 이러한 성능을 바탕으로 개발될 수 있어야 한다. 따라서 딥러닝 모델의 참조는 정확도와 같은 성능적 측면에 기반한다. 이에 데이터 분석을 수행하는 딥러닝 모델을 개발할 때, 개발자는 성능이 검증된 딥러닝 모델을 적용할 수 있어야 한다[15-17]. 신뢰성있는 딥러닝 기반 분석 모델을 적용하기 위해서는 개발자는 참조 모델의 학습 데이터셋, 개발 환경 및 딥러닝의 계층구조 등과 같은 딥러닝 모델의 정보를 이용하여 참조 모델을 개발하여 검증해야 한다[18-19]. 그리고 제공된 학습 데이터셋을 통해 모델의 성능을 검토하고 이를 기반으로 서비스 환경에 맞추어 개선할 수 있어야 한다. 이러한 과정에 필요한 딥러닝 모델의 정보를 제공하기 위해서는 딥러닝 모델을 문서화하기 위한 구조가 요구된다. 딥러닝 모델의 추상화 연구들[18-23]에서는 기계학습 모델을 기술하기 위한 기계학습 모델 스키마를 정의하였으나, 딥러닝 모델을 구조적으로 문서화하기 위해서는 구체적인 정보를 기술할 수 있는 요소가 요구되며, 이에 스키마를 보완할 필요가 있다.

본 논문에서는 여러 데이터 형태 중에서 이미지를 분석하는 딥러닝 모델의 문서화를 위한 기술 언어를 제안한다. 제안하는 딥러닝 기술 언어(Deep Learning Description Language for Image Analysis, 이하 DL2IA)는 딥러닝 모델을 개발하고 검증하는데 필요한 데이터셋, 개발환경, 자원 참조 정보 등과 함께 딥러닝 모델의 계층구조, 특히 합성곱 신경망을 표현하기 위한 정보를 포함한다. DL2IA를 통해 개발자는 XML 형식의 딥러닝 기술 문서를 활용하여 참조하는 딥러닝 모델의 성능을 검증하여 신뢰성있는 이미지 분석을 위한 딥러닝 모델을 개발할 수 있다.

2. 관련 연구

본 절에서는 기계학습 모델을 추상화하여 스키마로 표현한 연구들에 대해 기계학습 모델 관리, 데이터 프로비넌스, 기계

학습 모델 공유 관점으로 살펴본다.

설명가능한 인공지능(eXplainable Artificial Intelligence, 이하 XAI)에서는 인공지능 모델을 구조화하기 위한 연구가 진행되고 있다[24]. XAI는 인공지능 행위와 판단에 대해 사람이 이해 가능한 형태로 설명할 수 있는 인공지능으로 [25-26], 인공지능 모델의 이해를 돕기 위한 딥러닝 모델의 추상화가 연구되었다[18, 19, 20, 23]. 이 연구들에서는 기계학습 모델을 구조화하여 기술하기 위한 기계학습 스키마를 제안하였으며, 인공지능 모델을 실험, 데이터, 모델의 계층구조 등의 요소들로 구조화하였다.

기계학습 모델링은 학습, 평가, 배치 등의 과정이 반복되어 이루어진다. 이러한 과정은 목적과 용도에 따라 하나의 정형화된 워크플로우로 표현하여 스키마로 정의할 수 있고, 이를 통해 기계학습 모델은 데이터베이스 상에서 관리가 가능해진다. ModelDB[18]는 기계학습 개발의 전반적인 생명주기를 관리하기 위한 시스템으로, 기계학습 모델을 처리 단계에 따라 데이터베이스화하여 관리한다. ModelDB에서 정의한 기계학습 처리 단계는 기계학습 모델링을 위한 데이터 준비, 특징 추출, 학습, 검증, 배치, 유지 등으로 구성된다. 각 처리 단계는 워크플로우와 스키마로 표현될 수 있고 이는 관계형 데이터베이스에 저장되어 관리가 가능하다.

데이터의 출처 추적은 연구의 버전관리에 있어 필수적인 기능이다. 데이터 프로비넌스(Data Provenance)[27]는 데이터의 근원 정보를 제공하기 위한 메타데이터의 한 종류로서 데이터를 이해하고 연구를 구현하는데 사용된다. 기계학습 수명주기 동안 데이터의 출처가 불명확하여 데이터를 추적하지 못할 경우, 기계학습 모델을 다시 개발해야 하는 상황이 발생할 수 있다. 따라서 기계학습 모델의 관리를 위해서는 학습에 사용된 데이터의 출처가 명확하도록 데이터 프로비넌스를 제공해야 한다[28]. 이에 Renan[20]는 기계학습 모델 개발에 참여하는 이해 관계자를 설정하고 사용되는 데이터셋의 출처를 제공하기 위한 스키마인 PROV-ML을 제안하였다. PROV-ML은 기계학습 데이터 프로비넌스를 위한 스키마로서 선행연구인 W3C-PROV[21]와 ML-Schema[19]를 기반으로 설계되었으며, 데이터셋의 프로비넌스를 위한 메타데이터를 제공할 수 있도록 요소를 확장하였다. 또한 지진 감지를 위한 지층 데이터를 중심으로 한 예시를 통해 데이터 프로비넌스를 검증하였다.

기계학습 모델은 성능이 향상되는 만큼 복잡성 또한 증가하여 모델의 이해와 분석 결과에 대한 해석이 어려워지는 문제가 있다. 이에 Publio[19]는 기계학습 모델의 해석이 수월하도록 개발자들 간에 기계학습 모델을 공유할 수 있는 ML-Schema를 개발하였다. ML-Schema는 온톨로지 기반 데이터 모델인 PROV-O[22]에 바탕을 두며, 이는 데이터셋, 알고리즘, 모델 특성, 실험 등의 클래스와 속성을 갖는다.

Marcio[23]의 연구에서는 ML-Schema를 개선하여 딥러닝 모델을 위한 스키마를 개발하였다. 이 스키마는 기존 연구의 구성을 따르고 딥러닝 모델의 계층 구조를 표현하기 위해

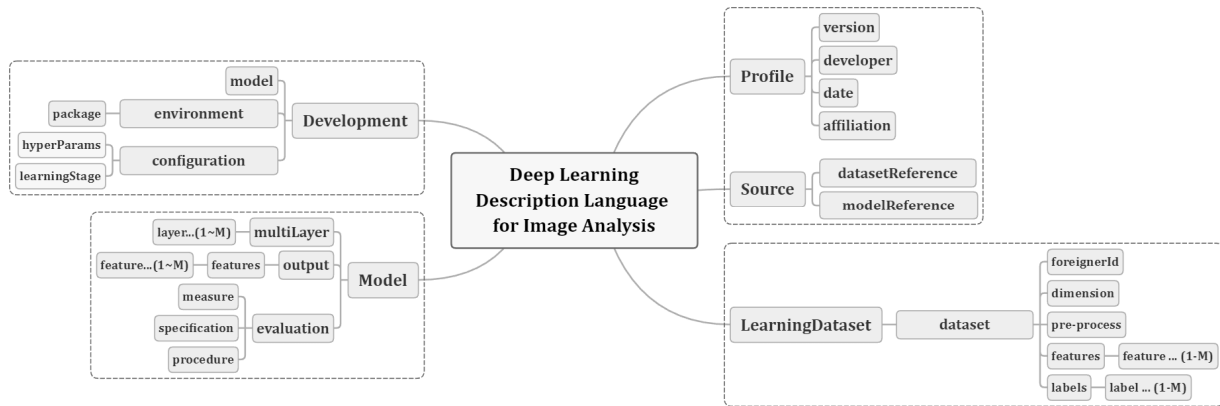


Fig. 1. Core Elements of Deep Learning Description Language for Image Analysis of Traffic Situation

구성요소를 확장하였지만, 이미지 분석을 위한 합성곱 신경망의 정보를 표현하기 위해서는 파라미터, 출력 구조 등의 추가 요소가 요구된다.

이미지 분석을 위한 딥러닝 모델을 문서화하고 이를 기반으로 모델을 개발 및 검증하기 위해서는 데이터셋, 딥러닝 모델의 계층 구조 등의 학습을 위한 정보뿐만 아니라 딥러닝 모델을 개발하기 위한 개발환경, 개발 설정 등의 정보 또한 구조적이고 체계적으로 기술할 수 있어야 한다. 선행연구들은 기계학습 모델을 구조화하여 스키마로 표현하였지만, 제안된 스키마를 통해 딥러닝 모델을 개발하기 위해서는 구성요소가 추가될 필요가 있다. 또한, 딥러닝 모델의 다층구조를 표현하기 위해서 구성요소의 확장이 요구된다.

본 연구에서는 딥러닝 모델을 구조적으로 기술하기 위한 언어를 제안한다. 이는 선행연구에서 설계한 스키마의 학습데이터셋, 모델, 실험 등의 요소뿐만 아니라 모델을 검증하기 위해 제공되어야 하는 개발환경 및 설정 정보, 그리고 딥러닝 모델의 계층 구조를 기술할 수 있다. 제안하는 딥러닝 모델 기술 언어에 대한 내용은 3장에서 구체적으로 설명하며, 이에 대한 검증 및 기존연구와의 비교 분석은 각각 4, 5장에서 보인다.

3. 제안하는 딥러닝 기술 언어

신뢰성있는 딥러닝 모델의 적용을 위해서는 딥러닝 모델의 정확도를 검증할 수 있어야 한다. 이를 위해서는 딥러닝 모델의 데이터셋, 개발환경, 자원 참조 정보, 계층 구조 등의 정보를 구조적으로 문서화하여 개발자에게 제공할 수 있어야 한다. 이에 정확도 검증에 필요한 정보를 제공하기 위한 딥러닝 모델 기술 언어는 다음과 같은 요구사항을 갖는다.

- 딥러닝 모델과 데이터셋의 참조 정보 제공을 위한 요소
- 참조하는 딥러닝 모델의 검증을 위한 학습 데이터셋의 정보를 기술하기 위한 요소
- 딥러닝 모델의 다층 구조를 표현하기 위한 요소
- 딥러닝 모델을 개발하는데 필요한 환경 및 설정 정보를 기술하기 위한 요소

본 절에서는 위 요구사항을 반영한 딥러닝 기반 교통상황 분석 모델을 기술하기 위한 딥러닝 기술 언어(DL2IA)에 대해 설명한다.

본 논문에서 제안하는 DL2IA는 ML-Schema[19]와 PROV-ML[20]을 기반으로 설계되었다. 핵심 구성요소(element)는 Fig. 1과 같다. Fig. 1은 딥러닝 기반의 교통상황 분석 모델을 구조적으로 표현하기 위한 요소의 개요를 보인 것이다. 상위요소는 Fig. 1에서 보이는 각 영역의 특성을 대표하는 요소로, 이는 개발(Development), 모델(Model), 프로파일(Profile), 자원(Source), 그리고 학습데이터셋(LearningDataset)이다.

DL2IA는 참조하는 딥러닝 모델의 검증을 위한 정보를 제공하기 위해 개발 환경 및 설정 등의 개발 정보를 기술하기 위한 상위 요소를 가진다. 참조 모델의 검증을 위한 정보를 기술하기 위해서는 개발 환경 위에서 동작하는 모델의 학습 정보가 필요하며 이는 요구사항 위 세 개에 해당한다. 이를 위해 추가적으로 프로파일, 자원, 학습데이터셋, 그리고 모델의 상위 요소를 정의하였으며 이는 아래 세부 절에 기술한다.

3.1 프로파일(Profile)

〈Profile〉은 참조하는 모델의 개발자 프로파일을 기술하기 위한 것으로 개발자의 이름 또는 닉네임, 개발 버전 및 일자, 참고 사이트 등의 정보를 포함한다. 속성(attributes)은 개발자 식별 정보 〈developer〉, 개발 버전 〈version〉, 개발일자 〈date〉 참조 문서 〈reference〉로 구성된다.

3.2 자원(Source)

〈Source〉는 딥러닝 모델과 데이터셋의 출처를 기술하기 위한 요소로 자원을 참조할 수 있는 정보를 가진다. 하위 요소로서 모델 참조 개체(이하, 〈modelReference〉)와 데이터셋 참조 개체(이하, 〈datasetReference〉)로 구성되며 각각 구현된 모델과 데이터셋의 식별 정보(〈id〉)와 저장 위치 정보(〈value〉)를 갖는다. 〈id〉를 통해 다른 요소들에서는 데이터셋과 모델의 참조 정보를 활용할 수 있으며, 참조는 〈id〉를 기반으로 한 key/value 방식이다.

Table 1. Elements of LearningDataset in DL2IA

Elements	Description
Dataset	A description of the characteristics of the dataset used in the learning stage
ForeignerId	An element to match the reference information with the ID of the dataset
Dimension	The size of the dataset, its format is like [28 x 28].
Data pre-processing	Imputation and outlier elimination etc.
Features (option)	Describe information about each feature of the dataset
Labels	Image labels in dataset (Supervised Learning)

3.3 학습데이터셋(⟨LearninDataset⟩)

⟨LearningDataset⟩은 기술된 딥러닝 모델의 학습에 사용된 데이터셋을 기술하기 위한 상위 요소이며, 각 학습 단계에 사용된 데이터셋을 나타내기 위해 하나의 데이터셋을 개체로 표현한다.

데이터셋(⟨dataset⟩)은 하나의 학습 데이터셋을 기술하기 위한 요소로 데이터셋의 식별자, 형식 등의 속성과 참조 정보, 품질, 입력 인자(feature) 등의 요소로 구성된다.

⟨dataset⟩은 ⟨Source⟩의 ⟨datasetReference⟩와 일대일로 연결되어 학습 단계에서 ⟨datasetReference⟩ 함께 참조되어 사용된다. 학습 단계는 학습, 검증, 테스트로 나뉘며 각 단계마다 다른 학습 데이터셋이 사용될 수 있어 최대 세 개의 학습데이터셋이 기술될 수 있다. 세부 구성요소는 Table 1과 같다. 참조 정보⟨foreignerId⟩는 학습에 사용된 데이터셋의 실제 위치 정보를 가진 ⟨datasetReference⟩을 연결하기 위한 정보이다.

다른 세부 구성요소들은 데이터셋의 특성을 기술 위한 것으로 차원 정보⟨dimension⟩은 데이터셋의 크기를 나타내며 ‘[28x28]’과 같은 형식으로 표현한다. 전처리 정보⟨pre-process⟩는 데이터셋 품질의 정도를 보이기 위한 요소로 전처리 과정에서 데이터셋의 데이터 대체, 이상치 제거 등의 처리된 비율을 0-1의 값으로 나타낸다.

3.4 모델(⟨Model⟩)

딥러닝 모델은 문제를 해결하기 위한 알고리즘과 레이어를 구성하고 이에 데이터를 학습한 구현체이다. DL2IA의 ⟨Model⟩은 이러한 딥러닝 모델의 딥러닝 알고리즘과 계층 구조와 평가 정보를 기술하기 위한 요소를 갖는다. 이에 ⟨model⟩의 하위 요소는 다층 구조를 표현하기 위한 ⟨multiLayer⟩, 출력 구조 ⟨output⟩, 모델 평가 정보 ⟨evaluation⟩ 등으로 구성되며 Fig. 2와 같다.

구체적으로 ⟨multiLayer⟩는 딥러닝 모델의 다층 레이어를 기술하기 위한 요소이며 각 레이어를 하나의 개체로써 레이어 ⟨layer⟩로 표현한다. ⟨layer⟩는 순서와 파라미터를 가지며 특

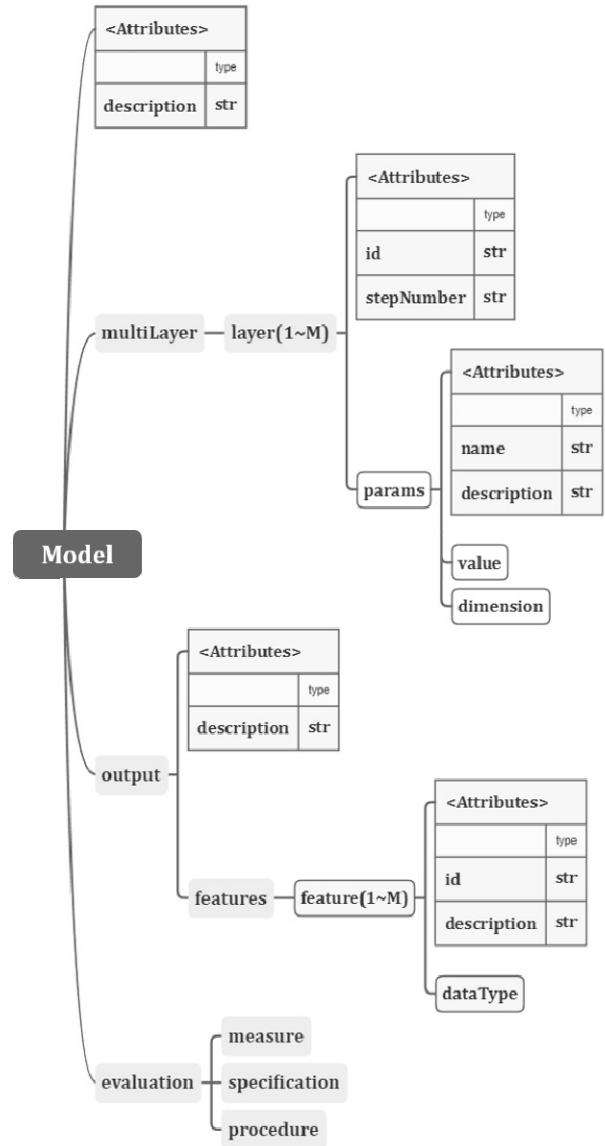


Fig. 2. Elements and Attributes of Model in DL2IA

히 이미지 분석을 위한 컨볼루션 레이어(Convolutional Layer)는 파라미터가 이 차원으로 표현된다. 이에 ⟨layer⟩는 파라미터의 차원 크기와 값을 저장할 수 있는 ⟨dimension⟩와 ⟨value⟩요소를 포함한다. 다층 레이어를 통해 얻게 되는 최종 출력 정보는 ⟨output⟩에 작성된다. ⟨output⟩은 딥러닝 모델의 최종 출력 구조로 출력 인자들을 ⟨feature⟩로 나타낸다. ⟨output⟩의 내용을 토대로 한 출력 결과는 ⟨evaluation⟩ 요소에 기술된 평가 정보를 기반으로 딥러닝 모델의 정확도를 평가한다. 이에 ⟨evaluation⟩은 모델 평가 정보가 기술되는 요소로써 모델 평가 알고리즘, 평가 결과, 평가 척도 등을 나타낸다. 따라서 참조하는 딥러닝 모델은 ⟨evaluation⟩을 기준으로 평가될 수 있다. 마지막으로 딥러닝 모델의 알고리즘과 구조에 대한 이해를 돕기 위한 설명은 ⟨description⟩ 속성에 기술되어 개발자를 지원한다.

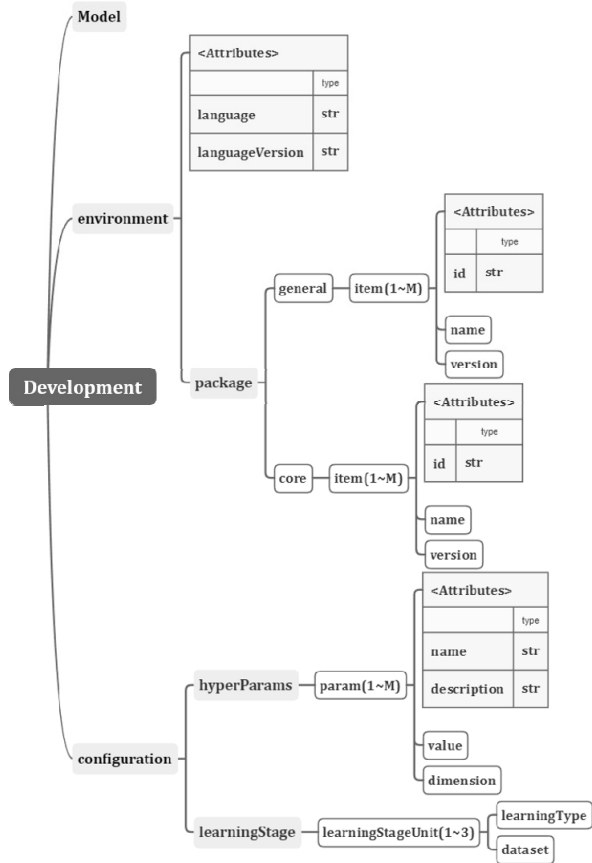


Fig. 3. Elements and Attributes of Development Environment in DL2IA

3.5 개발(Development)

딥러닝 모델 개발자는 기획한 모델을 개발하기 위한 개발 환경을 구축하고 구축된 환경 위에서 설정된 알고리즘을 기반으로 파라미터의 세부조정과 학습 및 성능 검토 과정을 반복하여 딥러닝 모델을 구현한다.

〈Development〉 요소는 딥러닝 모델을 개발하는데 필요한 정보를 기술하기 위한 것으로 Fig. 3과 같으며 〈LearningDataset〉, 〈Source〉, 〈model〉 요소들을 활용한다. 주요 세부요소는 두 가지로 개발 환경 〈environment〉과 설정 정보 〈configuration〉이다. 첫 번째로 〈environment〉는 문서에 기술된 모델을 개발하고 실행할 수 있는 환경을 구축하기 위한 구성 정보를 갖는다. 딥러닝 모델을 재구현하기 위해서는 기본적으로 개발 언어와 사용된 핵심 모듈, 패키지 등이 요구된다. 이에 개발 언어를 표시하기 위한 하위 속성을 가지고 사용된 패키지를 목록화하여 표현하는 요소를 포함한다. 두 번째로 〈configuration〉은 모델의 학습에서 사용되는 요소를 갖는다. 〈configuration〉의 세부 요소 중 〈hyperParams〉은 학습 과정을 조정하는데 사용되는 파라미터인 하이퍼파라미터의 설정값을 저장하기 위한 요소로, 이는 〈Model〉에 기술된 레이어의 파라미터 정보 또한 구현하기 위한 파라미터로 표현한다. 다른 요소인 〈learningStage〉는 모델의 각 학습 단계에 이용한 데이터셋

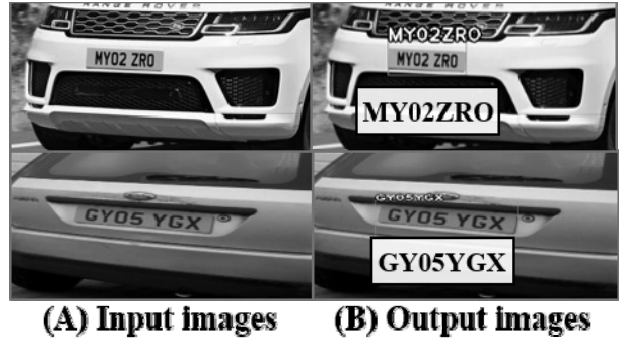


Fig. 4. Results According to the Input Images of the License Plate Recognition Model

을 나타낸다. 학습 단계는 학습, 검증, 테스트 단계로 나뉘며, 이에 사용된 데이터셋은 〈Source〉와 〈LearningDataset〉에 있는 정보를 참조한 것이다. 개발자는 〈Development〉를 통해 참조하는 딥러닝 모델을 재구현하고 〈Model〉에 기술된 평가 방법 및 정확도를 기반으로 검증할 수 있다.

4. 제안하는 기술 언어 문서의 적용 검증

본 절에서는 신뢰성있는 딥러닝 기반 분석 모델을 참조하기 위해 DL2IA 문서를 적용하는 과정을 보인다. 이를 위해 지능형 교통 시스템에서 차량 및 차량 번호 등의 교통상황 정보 수집과 스마트 단속 서비스에 활용되고 있는 차량번호판 인식 모델의 개발 시나리오를 예시로 든다. 본 개발 시나리오는 개발자가 참조하는 모델을 개발하는 일반적인 과정에 제안하는 기술 언어 문서를 적용한 것이다. 이는 참조 모델의 신뢰성을 검토하고 서비스 환경에 맞추어 수정 및 개발하는 과정으로, 참조 모델을 DL2IA 문서의 내용을 토대로 개발하고 정확도를 검증하는 과정을 포함한다.

본 절의 시나리오에서 예시로 사용하는 딥러닝 모델은 합성곱 신경망을 기반으로 한 차량번호 인식 모델[29]이다. Fig. 4는 이러한 차량번호 인식 모델에 차량 이미지(A)를 입력하여, 차량번호판 영역을 검출하고 번호를 인식한 결과(B)를 나타낸 것이다. 개발자는 참조하는 딥러닝 모델의 신뢰성을 DL2IA 문서를 통해 검증할 수 있으며, 이는 원시 개발자(Original Developer)로부터 작성된다. 이러한 문서가 작성되고 적용되는 시나리오는 Fig. 5와 같다.

Fig. 5에서 왼쪽에 위치한 개발자(A)는 개발자(B)가 참조하는 차량번호 인식 모델의 원시 개발자로 차량 자동 인식 서비스를 제공하기 위한 차량번호 인식 모델을 개발하였다. 개발자(A)는 개발 과정 동안 딥러닝 모델의 관리와 그리고 배포를 목적으로 차량번호 인식 모델을 DL2IA 문서(DL2IA document)로 작성하며, 개발 코드(Model code)와 함께 기술 문서를 배포한다. 기술 문서의 구체적인 예시는 Fig. 6, 7과 같다.

Fig. 6, 7은 차량번호판 인식 모델의 DL2IA 문서로 각각 개발 정보와 모델 정보를 기술한 것이다. 두 곳 모두 기재된

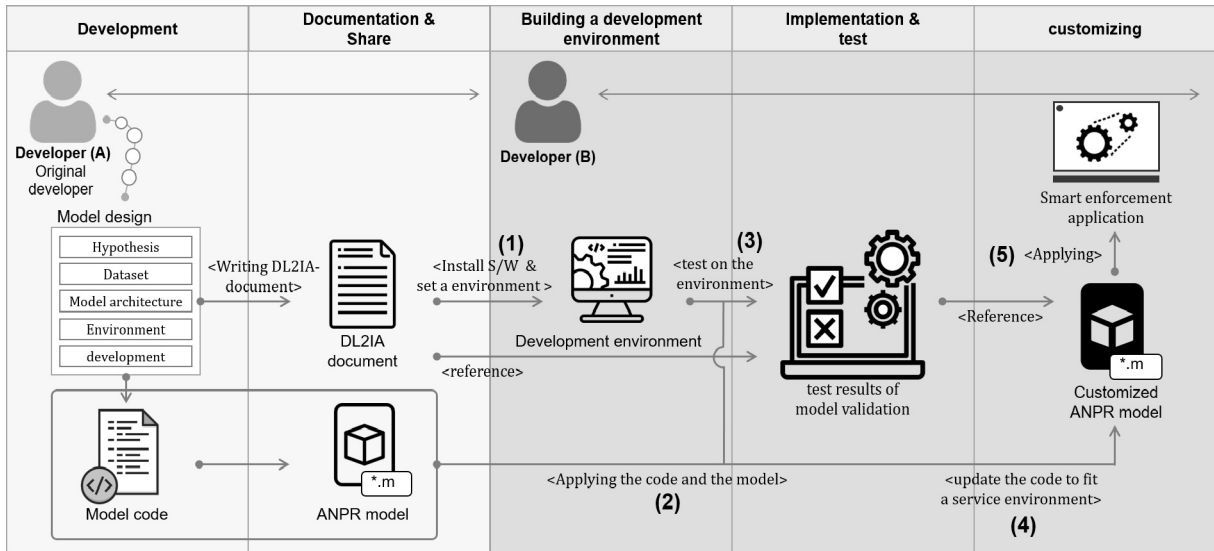


Fig. 5. Overview of Applying DL2IA Document for the Development of Deep Learning-based Automatic Number Plate Recognition (ANPR)

<DL2IA>는 최상위 요소로 개발 대상의 식별자를 가지며 본 예제의 차량번호판 인식 모델은 'deep_ANPR'로 명명된다.

Fig. 6에는 참조하는 모델의 레이어 구조, 출력, 평가 기준 등의 모델의 구성을 보인다. 출력 정보는 (A)영역과 같으며 <output>에 인자별로 구분하여 기술될 수 있다. 본 예제에서는 차량번호를 문자열로 출력한다. 레이어 정보는 <multiLayer>에 기술되어 한 레이어당 하나의 <layer>로 구성될 수 있으며 (B) 영역에 해당한다. 레이어는 번호, 파라미터 등의 정보를 포함하며 차량 인식 모델은 여러 개의 합성곱 레이어와 활성화 함수로 구성되어 최종적으로 문자열 형태의 출력 결과를 제공한다. (C) 영역은 모델을 평가할 때 사용된 척도(measure), 평가 방법(procedure), 그리고 기준과 방법에 따른 정확도(specification)를 보인다. 본 예제에서는 'cross entropy loss'를 척도로 하여 데이터셋으로부터 정확도를 산출한다. 평가 방법은 'cross validation', 'holdout method' 등이 있으나, 본 예제에서는 별도의 평가 방법이 사용되지 않았다.

Fig. 7은 차량번호 인식 모델의 개발 환경 및 설정 정보이다. (A)영역에서 볼 수 있듯이, 개발에 사용된 언어는 'python-3.7'이며, 핵심 패키지로 'tensorflow-1.14'가 사용되었다. 모델은 학습(training)과 검증(validation)의 학습 과정이 수행되었으며, 학습데이터셋은 'deepANPR'로 동일하게 적용되었다(Fig. 7(B)).

개발자(B)는 고속도로에서 자동화 단속 서비스를 제공하기 위해 스마트 단속 프로그램을 개발하면서 다른 개발자가 개발한 차량번호 인식 모델을 이용하고자 한다. 다른 개발자로부터 개발된 차량번호 인식 모델을 활용하기 위해서는 이러한 모델의 성능을 검토하고 수정하여 서비스에 적용하는 일련의 과정이 수행되어야 한다. 이 예제에서는 개발자(A)의 개발물을 활용한다. 이러한 과정은 Fig. 5에 표시되어 있는 (1)~(5)이며 구체적인 내용은 다음과 같다.

```

<DL2IA id="deep_ANPR">
  <Model description="algorithm: CNN">
    <output>
      <features>
        <feature id="carPlateResult">
          <dataType>str</dataType>
        </feature>
      </features>
    </output>
    <multiLayer>
      <layer id="deepANPR004Layer01",
        stepNumber="1">
        <name>convloutional_layer1</name>
        <params>
          <param id="conv1_weight">
            <name>weight</name>
            <value>[5, 5, 1, 48]</value>
            <dimension>[4]</dimension>
          </param>
          <param id="conv1_bias">
            <name>bias</name>
            <value>[48]</value>
            <dimension>[1]</dimension>
          </param>
          <!-- skip -->
        </params>
      </layer>
    </multiLayer>
    <!-- skip -->
    <evaluation>
      <measure> cross-entropy loss </measure>
      <procedure> none </procedure>
      <specification> 91 </specification>
    </evaluation>
  </Model>
  <!-- skip -->
</DL2IA>
    
```

Fig. 6. Document Describing Model Architecture of DL2IA-based Plate Number Recognition

```

<DL2IA id="deep_ANPR">
  <-- skip -->
  <Development>
    <environment language="python"
      languageVersion="3.7">
      <package>
        <general>
          <item>
            <name>opencv-python</name>
            <version>4.2.0.34</version>
          </item>
          <item>
            <name>numpy</name>
            <version>1.18.1</version>
          </item>
        </general>
        <core>
          <item>
            <name>TensorFlow</name>
            <version>1.1.4</version>
          </item>
        </core>
      </package>
    </environment>
    <configuration>
      <hyperParams>
        <param name="window_shape">
          <value>[64, 128]</value>
          <dimension>[2]</dimension>
        </param>
      </hyperParams>
      <LearningStage>
        <LearningUnit>
          <learningType>training</learningType>
          <dataset>deepANPR004</dataset>
        </LearningUnit>
        <LearningUnit>
          <learningType>validation</learningType>
          <dataset>deepANPR004</dataset>
        </LearningUnit>
      </LearningStage>
    </configuration>
  <-- skip -->
</Development>
</DL2IA>
  
```

Fig. 7. Document Describing Development Environment of DL2IA-based Plate Number Recognition Model

- (1) DL2IA 문서의 개발 정보를 참고하여 필요한 S/W를 설치하고 환경 또는 실행 변수를 설정하여 참조하는 차량번호 인식 모델(이하 참조 모델)을 실행할 수 있는 환경을 구축한다. S/W는 운영체제, 컴파일러, 가상머신, 라이브러리 등이 있다. 본 예제에서는 개발 언어는 'python-3.7'를 사용하기 위한 컴파일러 설치와 실행하기 위해서는 'tensorflow-1.14' 라이브러리 설치가 요구된다.
- (2) 구축된 개발 환경에서 코드 또는 바이너리 형태의 참조 모델을 실행하여 작동여부를 확인한다. 작동여부를 확인하는 과정에서 하이퍼파라미터의 조정이 이루어진다.
- (3) DL2IA 문서에 기술된 평가 정보와 데이터셋을 기반으로 정확도를 산출하여 참조 모델의 신뢰성을 검토하고,

이후 서비스 환경에서 획득한 이미지를 참조 모델에 적용하는 것을 통해 사용 여부를 결정한다.

- (4) 참조 모델의 사용이 가능하다고 판단되면, 개발자는 단속 서비스에 맞도록 참조 모델을 수정, 학습, 검증을 반복하는 과정을 수행한다. 이 때, DL2IA 문서가 작성되어 또 다른 개발자가 이를 활용할 수 있다.
- (5) 수정 및 개발이 완료된 새로운 모델은 서비스 환경에 배치되어 서비스에 적용된다.

위 과정에서 (1)~(3) 과정은 참조 모델의 신뢰성을 확인하는 과정이며, 이를 통해 신뢰성이 검증된 참조 모델은 (4)~(5) 과정을 통해 서비스 환경에 맞추어 수정 및 보완되어 서비스에 적용되어진다.

5. 제안하는 딥러닝 기술 언어 평가

딥러닝 모델을 문서화하기 위해서는 딥러닝 모델의 레이어 구조, 학습 데이터셋, 개발 환경, 참조 주소 등의 정보가 구조적으로 기술될 수 있어야 한다. 이에 본 장에서는 3장에서 요구사항과 함께 FAIR 데이터 원칙(FAIR Data Principles) [30]에 따라 제안하는 딥러닝 기술 언어와 기존에 연구된 기계 학습 모델 스키마들을 구성요소를 중심으로 비교 분석을 수행한다. FAIR 데이터 원칙[30]은 온라인상에서 과학자 간의 연구 데이터를 투명하고 효율적으로 공유하여 활용하기 위한 표준방안으로 검색가능성(Findable), 접근가능성(Accessible), 상호 운용성(Interoperable), 재사용성(Reusable)의 4가지 원칙을 기반으로 한다. 이 중, 재사용성은 메타데이터를 통해 공개된 연구를 재사용할 수 있도록 상세히 기술해야 한다는 원칙으로, 본 실험에서는 이를 기준으로 기계학습 모델 스키마를 평가한다. FAIR 데이터 원칙을 기반으로 본 연구에서 제시하는 기계 학습 또는 딥러닝 연구를 위한 재사용성의 기준은 다음과 같다.

- 딥러닝 모델과 학습데이터셋의 출처의 기술을 위한 요소
- 딥러닝 모델의 학습데이터셋에 대한 생성 목적과 날짜, 속성 및 범위 등의 정보를 기술하기 위한 요소
- 참조하는 딥러닝 모델의 계층 구조를 표현하기 위한 요소
- 참조하는 딥러닝 모델을 개발하여 재사용하는데 필요한 사용된 소프트웨어 및 개발환경, 그리고 매개변수와 같은 설정 정보를 기술하기 위한 요소

Table 2는 DL2IA와 기계학습 모델 스키마들을 위의 요구사항에 따라서 신뢰성있는 딥러닝 모델을 개발할 수 있는지에 대해 구성요소를 통해 비교 분석한 것이다.

PROV-O를 확장한 ML-Schema는 비교 대상이 되는 연구들의 기반 연구로서, 개발자 간에 모델을 수월하게 공유하는 것을 목적으로 한다. 이에 공유를 위한 학습데이터셋, 모델 평가 정보, 운영체제와 개발 언어와 같은 개발 요소 등의 핵심 요소로만 구성되어 낮은 복잡성을 가진다. 하지만 딥러닝 모델을 개발하기 위한 구체적인 개발 환경 및 모델의 구조

Table 2. Comparison of Schema Studies to Describe the Machine Learning Models

Schema	ML-Schema	PROV-ML	Hyperknowledge Model	DL2IA
Requirements	-	DataStoreInstance	-	datasetReference
	-	-	-	ModelReference
Data characteristic	Dataset	LearningDataset Reference	Dataset	LearningDataset
	Feature	Featureset	Feature	Features
Deep Learning Model characteristic	Model	ModelSchema	Model	Model
	-	ModelHyperParameter & Value	-	Output
	ModelEvaluation	ModelEvaluation	ModelEvaluation	Evaluation
Implementation Environment	-	-	Layer	MultiLayer
	Implementation	Implementation	Implementation	Development
	-	-	-	Environment
	-	LearningStage	-	LearningStage
-	LearningStageType	-	LearningStageUnit	

를 기술할 수 없어 모델을 구현하여 성능을 검증하기엔 어렵다. 이러한 기계학습의 특성에 대한 핵심 요소만을 갖고 있는 ML-Schema를 딥러닝 모델에 적용할 수 있도록 확장한 Hyperknowledge Model은 딥러닝의 계층 구조를 표현하기 위한 레이어 클래스를 추가하였다. 그러나 레이어 클래스를 통해 여러 종류의 레이어의 특성을 기술하기에는 한정이며, 이에 합성곱 신경망을 표현하기 위해서는 구성요소의 확장이 요구된다. DL2IA는 ML-Schema의 핵심 요소를 기반으로 설계되어 기반이 되는 스키마와 마찬가지로 공유를 위한 핵심 요소를 가진다. 또한 <datasetReference>, <modelReference>와 같은 요소들을 통해 모델 개발에 필요한 소스 제공하여 딥러닝 모델과 학습데이터셋에 대한 출처를 통해 교통상황 분석 모델을 개발할 수 있으며, 요소들을 구체화하여 딥러닝 모델 문서화가 가능하다.

PROV-ML은 딥러닝 모델의 특성 요소들을 포함하면서 데이터의 프로비던스를 위한 요소들을 확장하였다. 이를 통해 학습 단계를 구체화하고 각 단계별 사용된 데이터의 참조 정보 및 데이터의 특성 정보가 추가되어 이전 작업에서 사용된 데이터에 대한 추적이 가능하다. 그러나 위 두 스키마와 같이 구체적인 딥러닝 모델의 레이어를 구조화할 수 있는 요소가 필요하며, 참조 모델의 검증을 위한 개발 환경 정보가 제공되기 위한 요소가 필요하다. DL2IA는 딥러닝 모델의 구조를 표현하기 위한 요소(<multiLayer>)와 함께 개발 환경 정보를 기술할 수 있는 요소(<environment>)와 <configuration> 및 데이터셋을 표현할 수 있도록 데이터셋 요소(<features>, <dimension>)가 확장되었다. 또한 구체적인 참조 정보와 개발 환경, 모델의 계층구조를 구현 수준으로 정보를 제공한다. 이러한 요소들을 통해 재사용성 원칙에 따라 딥러닝 모델은 문서로 기술될 수 있으며 이를 통해 개발자는 참조하는 딥러닝 모델을 개발하여 검증할 수 있다.

6. 결 론

다양한 산업분야에서 활용되는 딥러닝 기반 데이터 분석 모델은 고품질의 서비스를 제공하기 위해 정확성이 요구된다. 이에 개발자는 검증된 모델의 적용을 고려하며, 적용 모델은 신뢰성이 보장되어야 한다. 이에 개발자는 딥러닝 모델 기술 문서를 제공받아 정확도를 검증하여 신뢰성을 확인할 수 있어야 한다. 이에 본 논문에서는 신뢰성있는 이미지 분석을 위해 딥러닝 모델을 구조적으로 표현하기 위한 딥러닝 기술 언어(DL2IA)를 제안하였다. 이를 통해 개발자는 참조 딥러닝 모델의 정확도를 검증하여 신뢰성을 보장할 수 있다.

DL2IA는 ML-Schema와 PROV-ML를 기반으로 개발되었으며, 이를 통해 딥러닝 모델 개발자는 데이터셋, 계층구조, 개발환경 및 설정 정보를 구조적으로 문서화할 수 있다. 실험에서는 DL2IA의 적용 과정을 차량번호 인식 모델을 활용한 시나리오를 기반으로 필요성을 입증하였다. 또한 기존 연구들과 비교 분석을 통해 DL2IA의 유효성을 보였다. 이에 개발자는 DL2IA 기반의 딥러닝 모델 기술 문서를 통해 참조하는 딥러닝 모델을 개발하여 검증 과정을 수행할 수 있다.

DL2IA는 단일 신경망을 기술하기 위한 기술 언어이다. 최근 딥러닝 연구들은 다중 네트워크를 기반으로 하여, 이를 하나의 기술 언어로 표현하기 위한 연구가 추가 진행될 필요가 있다. 이에 향후 연구로 범용적인 활용을 위해 다중 신경망을 기술하기 위한 확장 연구를 진행할 것이다. 또한 설명가능한 인공지능 분야에서 딥러닝 모델을 설명가능한 인공지능 모델로 확장하기 위해 필요한 딥러닝 기술 문서로 개선할 것하고 보다 이해를 돕기 위해 워크플로우 차트로 기술하기 위한 방법을 연구할 것이다.

References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, Vol.521, No.7553, pp.436-444, 2015.
- [2] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational intelligence magazine*, Vol.13, No.3, pp.55-75, 2018.
- [3] M. Y. Liu et al., "Few-shot unsupervised image-to-image translation," *The IEEE International Conference on Computer Vision (ICCV)*, pp.10551-10560, 2019.
- [4] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961-2969, 2017.
- [5] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1125-1134, 2017.
- [6] S. J. Choi, E. W. Kim, and S. H. Oh, "Human behavior prediction for smart homes using deep learning," *2013 IEEE RO-MAN*, IEEE, pp.173-179, 2013.
- [7] S. U. Amin, M. S. Hossain, G. Muhammad, M. Alhussein, and M. A. Rahman, "Cognitive smart healthcare for pathology detection and monitoring," *IEEE Access* 7, pp. 10745-10753, 2019.
- [8] C. N. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Transactions on Intelligent Transportation Systems*, Vol.7, No.3, pp.377-392, 2006.
- [9] S. H. An, B. H. Lee, and D. R. Shin, "A survey of intelligent transportation systems," *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*, IEEE, pp.332-337, 2011.
- [10] R. Akhawaji, M. Sedky, and A. H. Soliman, "Illegal parking detection using Gaussian mixture model and kalman filter," *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, pp.840-847, 2020.
- [11] J. P. Lin, and M. T. Sun, "A YOLO-based traffic counting system," *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, IEEE, pp.82-85, 2018.
- [12] J. Dahmen, D. J. Cook, X. Wang, and W. Honglei, "Smart secure homes: A survey of smart home technologies that sense, assess, and respond to security threats," *Journal of Reliable Intelligent Environments*, Vol.3, No.2, pp.83-98, 2017.
- [13] D. Popa, F. Pop, C. Serbanescu, and A. Castiglione, "Deep learning model for home automation and energy reduction in a smart home environment platform," *Neural Computing and Applications*, Vol.31, No.5, pp.1317-1337, 2019.
- [14] IBM AI research, "Trusting AI," IBM, accessed September, 11, 2020 [Internet], <https://www.research.ibm.com/artificial-intelligence/trusted-ai/>.
- [15] D. Sirohi, N. Kumar, and P. S. Rana, "Convolutional neural networks for 5G-enabled Intelligent Transportation System: A systematic review," *Computer Communications*, Vol.153, pp.459-498, 2020.
- [16] H. Hendry and R. C. Chen, "Automatic license plate recognition via sliding-window darknet-YOLO deep learning," *Image and Vision Computing*, Vol.87, pp.47-56, 2019.
- [17] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, Vol.11, No.2, pp.68-75, 2017.
- [18] M. Vartak and S. Madden, "MODELDB: Opportunities and challenges in managing machine learning models," *IEEE Data Eng. Bull.*(플네임표기), Vol.41, No.4, pp.16-25, 2018.
- [19] G. C. Publio et al., "ML-Schema: Exposing the semantics of machine learning with schemas and ontologies," *arXiv*, arXiv:1807.05351, 2018.
- [20] R. Souza et al., "Provenance data in the machine learning lifecycle in computational science and engineering," *2019 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*, IEEE, pp.1-10, 2019.
- [21] World Wide Web Consortium, "PROV-DM: the PROV data model," W3C, April 30, 2013, accessed December 7, 2020 [Internet] <https://www.w3.org/TR/prov-dm/>.
- [22] T. Lebo et al., "PROV-O: The PROV Ontology," *W3C recommendation* 30, 2013.
- [23] M. Moreno et al., "Managing machine learning workflow components," *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, IEEE, pp.25-30, 2020.
- [24] A. Seeliger, M. Pfaff, and H. Krcmar, "Semantic web technologies for explainable machine learning models: A literature review," *PROFILES/SEMEX@ ISWC*, pp.30-45, 2019.
- [25] D. Gunning, "Explainable artificial intelligence (XAI)," *Defense Advanced Research Projects Agency (DARPA)*, accessed June 16, 2020.
- [26] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical XAI," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [27] W. C. Tan, "Research problems in data provenance," *IEEE Data Engineering Bulletin*, Vol.27, No.4, pp.45-52, 2004.

- [28] N. Baracaldo, B. Chen, H. Ludwig, and J. A. Safavi, "Mitigating poisoning attacks on machine learning models: A data provenance based approach," *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp.103-110, 2017.
- [29] M. Earl, "Using neural networks to build an automatic number plate recognition system," *GitHub*, August 30, accessed June 30, 2020 [Internet, <https://github.com/matthewearl/deep-anpr>].
- [30] M. D. Wilkinson et al., "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific Data*, Vol.3, No.1, pp.1-9, 2016.



최 종 선

<https://orcid.org/0000-0001-9648-0667>

e-mail : jongsun.choi@ssu.ac.kr

2000년 송실대학교 컴퓨터학부(학사)

2002년 송실대학교 컴퓨터학부(석사)

2008년 ~ 2010년 유한대학교 e-비즈니스과
전임교원

2010년 송실대학교 컴퓨터학부(박사)

2011년 ~ 2012년 송실대학교 지능형로봇연구소 연구원

2012년 ~ 2013년 서일대학교 인터넷정보과 전임교원

2013년 ~ 현 재 송실대학교 컴퓨터학부 부교수

관심분야 : 로봇 소프트웨어 플랫폼, 클라우드 컴퓨팅



문 종 혁

<https://orcid.org/0000-0002-4919-0058>

e-mail : jonghyeokmun@soongsil.ac.kr

2017년 한경대학교 컴퓨터공학과(학사)

2019년 송실대학교 컴퓨터학과(석사)

2019년 ~ 현 재 송실대학교 컴퓨터학과
박사과정

관심분야 : 로봇 미들웨어, 빅데이터 처리, MLOps, XAI



김 도 형

<https://orcid.org/0000-0002-8951-8679>

e-mail : dohyungkim@soongsil.ac.kr

2019년 한경대학교 컴퓨터공학과(학사)

2019년 ~ 현 재 송실대학교 컴퓨터학과
석사과정

관심분야 : 클라우드 컴퓨팅, 빅데이터
처리, 인공지능 모델링



최 재 영

<https://orcid.org/0000-0002-7321-9682>

e-mail : choi@ssu.ac.kr

1984년 서울대학교 제어계측공학과(학사)

1986년 미국 남가주대학교 전기공학과
(컴퓨터공학)(석사)

1991년 미국 코넬대학교 전기공학부
(컴퓨터공학)(박사)

1992년 ~ 1994년 미국 국립오크리지연구소 연구원

1994년 ~ 1995년 미국 테네시 주립대학교 연구교수

1995년 ~ 현 재 송실대학교 컴퓨터학부 교수

관심분야 : 로봇 미들웨어, 시스템소프트웨어, 병렬/분산처리,
고성능컴퓨팅