Journal of The Korea Convergence Society
Vol. 13. No. 3, pp. 01-09, 2022
e-ISSN 2713-6353
DOI : https://doi.org/10.15207/JKCS.2022.13.03.001

클래스 다이어그램 이미지의 자동 분류에 관한 연구

김동관
목포해양대학교 해양컴퓨터공학과 교수

A Study on Automatic Classification of Class Diagram Images

Dong Kwan Kim
Professor, Department of Computer Engineering, Mokpo National Maritime University

요 약 UML(Unified Modeling Language) 클래스 다이어그램은 시스템의 정적인 측면을 표현하며 분석 및 설계부터 문서화, 테스팅까지 사용된다. 클래스 다이어그램을 이용한 모델링이 소프트웨어 개발에 있어 필수적이지만, 경험이 많지 않은 모델러에게 쉽지 않은 작업이다. 도메인 카테고리별로 분류된 클래스 다이어그램 데이터 세트가 제공된다면, 모델링 작업의 생산성을 높일 수 있을 것이다. 본 논문은 클래스 다이어그램 이미지 데이터를 구축하기 위한 자동 분류 기술을 제공한다. 추가 정보 없이 단지 UML 클래스 다이어그램 이미지를 식별하고 도메인 카테고리에 따라 자동 분류한다. 먼저, 웹상에서 수집된 이미지들이 UML 클래스 다이어그램 이미지인지 여부를 판단한다. 그리고, 식별된 클래스 다이어그램 이미지에서 클래스 이름을 추출하여 도메인 카테고리에 따라 분류한다. 제안된 분류 모델은 정밀도, 재현율, F1점수, 정확도에서 각각 100.00%, 95.59%, 97.74%, 97.77%를 달성했으며, 카테고리별 분류에 대한 정확도는 81.1%와 95.2% 사이에 분포한다. 해당 실험에 사용된 클래스 다이어그램 이미지 개수가 충분히 크지 않지만, 도출된 실험 결과는 제안된 자동 분류 방식이 고려할 만한 가치가 있음을 나타낸다.

주제어 : 소프트웨어 융합, Unified Modeling Language, 객체지향 모델링, 클래스 다이어그램, 딥러닝

Abstract UML class diagrams are used to visualize the static aspects of a software system and are involved from analysis and design to documentation and testing. Software modeling using class diagrams is essential for software development, but it may be not an easy activity for inexperienced modelers. The modeling productivity could be improved with a dataset of class diagrams which are classified by domain categories. To this end, this paper provides a classification method for a dataset of class diagram images. First, real class diagrams are selected from collected images. Then, class names are extracted from the real class diagram images and the class diagram images are classified according to domain categories. The proposed classification model has achieved 100.00%, 95.59%, 97.74%, and 97.77% in precision, recall, F1-score, and accuracy, respectively. The accuracy scores for the domain categorization are distributed between 81.1% and 95.2%. Although the number of class diagram images in the experiment is not large enough, the experimental results indicate that it is worth considering the proposed approach to class diagram image classification.

Key Words : Software convergence, Unified Modeling Language, Object-oriented modeling, Class diagram, Deep learning

*This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1I1A3056172).

*Corresponding Author : Dong Kwan Kim (dongkwan@mmu.ac.kr)

Received February 11, 2022 Revised March 8, 2022
Accepted March 20, 2022 Published March 28, 2022

1. 서론

소프트웨어 모델링은 개발하고자하는 목표 시스템에 대한 추상화된 표현인 모델을 만드는 작업으로 소프트웨어 개발에 있어 필수 작업 중 하나이다. UML(Unified Modeling Language) [1]은 20년 이상 산업계 및 학계에서 객체지향 분석 및 설계를 효과적으로 지원하기 위해 사용되고 있다. UML은 다양한 관점에서 여러 유형의 시스템을 모델링할 수 있도록 사용 사례 다이어그램, 클래스다이어그램, 순차 다이어그램, 활동 다이어그램 등을 포함하는 다이어그램들을 제공한다. 특히, UML 클래스 다이어그램은 핵심이 되는 다이어그램으로 소프트웨어의 정적인 측면 즉, 소프트웨어 구조를 설계하고 설명하는 데 사용된다. 소프트웨어 모델링 작업이 소프트웨어생명주기 전 과정에 영향을 미칠 정도로 중요함에도 불구하고, 클래스 다이어그램을 이용한 모델링은 쉽지 않은 작업이다.

기존에 작성된 클래스 다이어그램들을 사례로 참고한다면, 클래스 다이어그램 작성을 보다 효과적으로 할 수 있을 것이다. 하지만, 클래스 다이어그램은 프로그램 소스코드와 같이 공개된 데이터 세트가 없으므로 클래스 다이어그램 데이터 세트 구축이 필요하다. 깃허브와 같은 공개 저장소에 소스 코드와 함께 UML 다이어그램들이 공유되어 있지만, 검색 기능이 제한되어 수동으로 검색해야 하는 경우도 있다. 또한, 브라우저들의 이미지 검색 기능이나 이미지 크롤링 소프트웨어를 이용하여 다이어그램 이미지를 수집할 수 있지만, 검색된 이미지가 실제로 클래스 다이어그램 이미지인지를 판단해야 한다. 이미지 검색 결과가 실제 클래스 다이어그램 이미지가 아닌 경우를 포함할 수 있다. 즉, 웹을 통해 수집된 이미지가 클래스 다이어그램인지 아닌지를 분류하는 작업이 요구된다. 이러한 작업은 이미지의 양이 많은 경우 수작업으로 하기 비효율적이므로 자동 분류 기능이 필요하다.

또한, 수집된 클래스 다이어그램 이미지들이 도메인 카테고리에 따라 좀 더 세부적으로 분류된다면 데이터 활용 측면에서 효과적이다. 예를 들어, UML 클래스 다이어그램 이미지가 '뱅킹', '교육', '주문" 등과 같은 도메인 카테고리별로 세부 분류된다면 모델링 교육에 보다 용이하게 활용될 수 있을 것이다. 이러한 데이터 세트들은 클래스 다이어그램을 대상으로 한 머신러닝이나 딥러닝 연구를 위한 데이터 세트로도 활용될 수 있다. 클래스 이름은 클래스 다이어그램의 핵심 구성요소이므로 클래스 이름을 통해 주어진 클래스 다이어그램이 속하는 카테고리를 결정할 수 있다. 클래스 다이어그램 이미지에 포함된 클래스 이름들을 추출하고 클래스 이름들과 도메인 카테고리와의 유사도를 계산하여 주어진 클래스 다이어그램 이미지의 카테고리를 결정한다.

본 논문은 도메인 카테고리별로 UML 클래스 다이어그램 이미지를 자동분류하여 클래스 다이어그램 이미지 데이터 세트를 제공하는 데 목적이 있다. 클래스 다이어그램에 대한 추가 정보없이 단지 이미지만을 입력으로 받아 실제로 클래스 다이어그램을 포함한 이미지인지를 판별하는 기능과 식별된 클래스 다이어그램 이미지를 대상으로, 사전에 정의된 도메인 카테고리에 따라 자동 분류 기능을 제공한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구를 소개한다. 3장에서는 클래스 다이어그램 분류를 위한 방법을 기술한다. 제안된 방법을 이용한 실험 결과와 이에 따른 분석은 4장에서 정리한다. 결론과 향후 연구 방향은 5장에 제시된다.

2. 관련 연구

본 논문에서는 이미지 형식으로 저장된 클래스 다이어그램을 고려하며 이미지에 포함된 객체를 식별하기 위한 머신러닝 또는 딥러닝 알고리즘들이 UML 다이어그램 이미지 분류에 적용되고 있다.

[2]는 Convolution Neural Network (CNN) 기반의 딥러닝 모델을 이용하여 UML 클래스 다이어그램을 자동 분류한다. UML 클래스 다이어그램과 비 UML 클래스 다이어그램을 포함하는 이미지를 데이터 세트로 사용한다. [3]은 UML 다이어그램 중 클래스 다이어그램, 활동 다이어그램, 순차 다이어그램, 사용 사례 다이어그램을 분류한다. 수집한 이미지가 어떤 유형의 UML 다이어그램을 포함하는지를 판단한다. 이미지 분류를 위해 ResNet, MobileNet 등을 포함하는 사전 훈련된 모델을 이용한 전이 학습을 사용한다. [4]는 머신러닝 알고리즘을 이용하여 웹 이미지가 UML 다이어그램인지 아닌지를 판단한다. 이미지에 대한 별도의 정보가 없는 웹 이미지의 그래픽 특성(예를 들어, 그레이스케일 히스토그램, 컬러 히스토그램, 기본 기하학적 형태 등)들을 추출하고 해당 정보들을 기반으로 규칙에 따라 자동 분류를 수행한다. [5]는 UML 다이어그램은 아닌 P&I 다

이어그램에 포함된 객체와 객체들의 연결을 탐지하기 위해 머신러닝 알고리즘을 사용한다. 종이에 작성된 P&I 다이어그램을 스캔 후 YOLO 알고리즘을 사용하여 디지털화하는 작업을 수행한다. 이 과정에서 다이어그램에 포함된 펌프, 밸브 등을 탐지한다. 또한 Optical Character Recognition (OCR) 기술을 사용하여 텍스트 정보를 추출한다. [6]은 서비스 지향 아키텍처(Service-Oriented Architecture, SOA) 개발을 위해 UML 다이어그램 이미지에 포함된 오퍼레이션과 서비스간 관계(화살표 이미지)를 추출한다. 클래스 다이어그램에 포함된 오퍼레이션 추출을 위해 OCR과 딥러닝 LSTM 알고리즘을 사용한다. 추출된 오퍼레이션 이름들은 서비스 이름들과 관계를 가진다. 또한, 기존의 ResNet50 CNN 모델 [7]을 이용하여 클래스 다이어그램에 포함된 의존성 관계(화살표 이미지) 유형 식별과 순차다이어그램에 포함된 동기화, 비동기화, 응답 메시지를 구별한다. 오퍼레이션 이름 추출과 화살표 이미지 식별은 SOA 시스템 개발에 활용된다. [8]은 UML 클래스 다이어그램을 FwCD(Forward Engineered Class Diagram)와 RECD(Reverse Engineered Class Diagram)로 구분한다. FwCD는 수작업을 통해 작성된 클래스 다이어그램이고, RECD는 소스 코드에서 리버스 엔지니어링 과정을 통해 자동 생성된 다이어그램을 의미한다. 해당 논문은 머신러닝 알고리즘들을 사용하여 FwCD와 RECD를 자동 분류한다.

3. 클래스 다이어그램 이미지 분류

본 장에서는 UML 클래스 다이어그램 이미지를 분류하고 도메인 카테고리별로 세부 분류 위한 방법에 대해 설명한다. 이미지 분류를 위해 딥러닝 모델을 사용하고 이미지 카테고리화를 위해 이미지로부터 텍스트를 추출한다.

3.1 CD 이미지 식별 모델

Fig. 1은 주문관리시스템에 대한 UML 클래스 다이어그램이다. 해당 클래스 다이어그램은 jpg 형식의 이미지이며 670 x 602 픽셀로 구성된다. 클래스 다이어그램은 *Order, OrderStatus, Account, Transaction, ShoppingBasket, StockItem, LineItem* 총 7개의 클래스로 구성된다. 주어진 이미지를 설명하는 별도의 부가 정보는 제공되지 않으며 보통의 이미지와 동일하다. 본 논문에서는 먼저, 입력된 이미지가 클래스 다이어그램인지 아닌지를 분류한다. 그 다음, 식별된 클래스 다이어그램 이미지를 대상으로 OCR 알고리즘을 적용하여 클래스 이름들을 추출한다. 일반적으로 하나의 이미지에 다수 개의 클래스 포함된다. 본 논문에서는 5개 이상의 클래스를 포함한 클래스 다이어그램 이미지만을 고려했다. 클래스 이름을 기준으로 도메인 카테고리가 결정된다. Fig. 1에 표현된 클래스 다이어그램은 “Order” 카테고리로 분류할 수 있다.

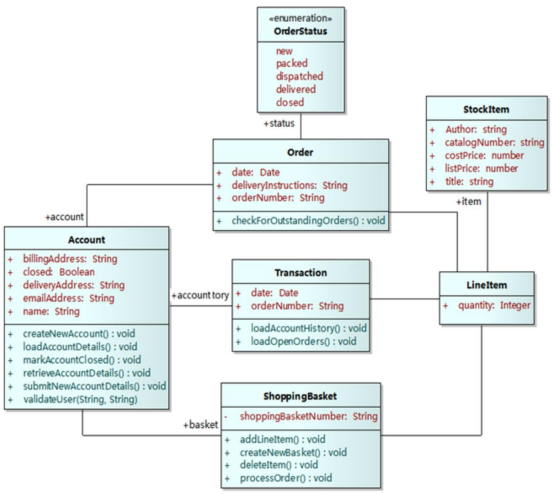

Fig. 1. UML Class Diagram for an Order Management System

Fig. 2는 클래스 다이어그램 이미지를 도메인에 따라 자동 분류하는 전체 과정을 표현한다. 분류 과정은 크게 클래스 다이어그램 이미지를 식별하는 단계(Fig. 2.에서 Step 1. 해당)와 식별된 실제 클래스 다이어그램 이미지를 도메인 카테고리별로 세부 분류하는 단계(Fig. 2.에서 Step 2. 해당)로 구성된다. 본 논문에서 제안하는 분류 모델의 입력은 이미지이며 최종 산출물은 도메인에 따라 분류된 이미지 카테고리이다. 클래스 다이어그램 이미지들은 깃허브나 검색엔진의 이미지 검색 기능을 통해 수집한다 [9,10]. 하지만, 검색 결과의 성능에 따라, 검색되거나 웹 크롤링을 통해 다운로드된 이미지들이 실제로 클래스 다이어그램일 수도 있고 그렇지 않은 경우도 발생한다. 그러므로, 인터넷 상에서 수집된 이미지들을 대상으로 실제 클래스 다이어그램 이미지를 분류하는 작업이 요구된다. 이미지 이진 분류는 딥러닝 모델의

특화된 분야로 기존 머신러닝 알고리즘들보다 우수한 성과를 달성하고 있다. 본 논문에서는 이미지 분류를 위해 ResNet50을 사전 모델로 사용한다. CNN 기반의 이미지 분류 모델을 새롭게 개발할 수도 있지만, 이미 방대한 이미지 데이터세트인 ImageNet으로 학습되고 다양한 이미지 분류 영역에서 의미 있는 성과를 도출한 ResNet50을 사용해도 본 연구에서 원하는 결과를 도출할 수 있을 것으로 기대된다. 이는 한 작업에서 학습한 지식을 다른 작업에 변경하여 적용하는 학습 유형인 전이 학습에 해당한다. ResNet50에서 특성 추출을 위한 컨볼루션 계층들은 원래 모델 그대로 사용된다. 훈련 과정 중에 컨볼루션 계층의 패러미터들은 변경되지 않는다. 컨볼루션 계층에서 생성된 활성화 맵은 완전히 연결된 소프트맥스 계층(Fully-Connected Softmax Layer)의 입력이 된다. 본 논문에서는 클래스 다이어그램 이미지인 경우와 아닌 경우 두 가지 경우만을 고려한다.

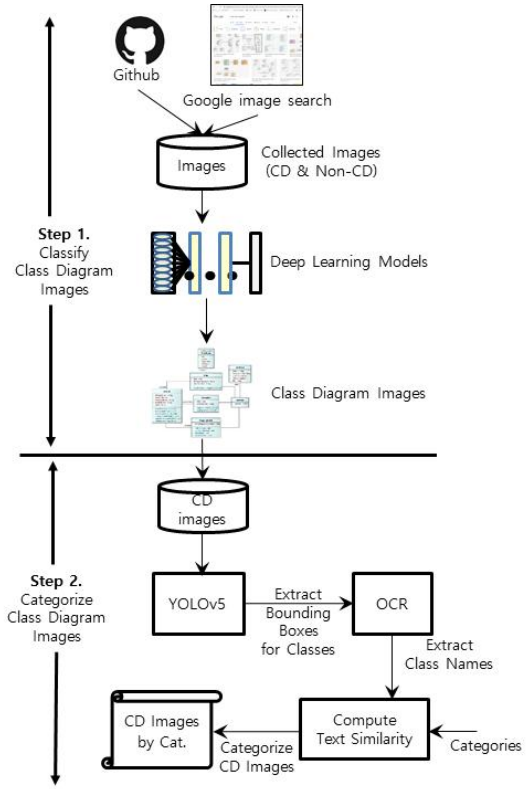

Fig. 2. The Overall Workflow of Categorizing Class Diagram Images

Fig. 3에서 보듯이, 이미지 분류기는 클래스 다이어그램 이미지를 이용한 지도 학습 과정을 통해 개발된다. 인터넷 상에서 수집된 원시 이미지 데이터는 데이터 사전 처리와 데이터 증강 과정 후 변경된 ResNet50 모델의 학습을 위해 사용된다. 완성된 이미지 분류 모델은 테스팅 이미지 데이터 세트를 사용하여 성능 평가를 수행한다. 클래스 다이어그램 이미지 분류기를 통과한 이미지들은 클래스 다이어그램과 비클래스 다이어그램으로 분류된다. 즉, 전체 작업의 단계 1에서는 실제 클래스 다이어그램 이미지를 추출할 수 있다.

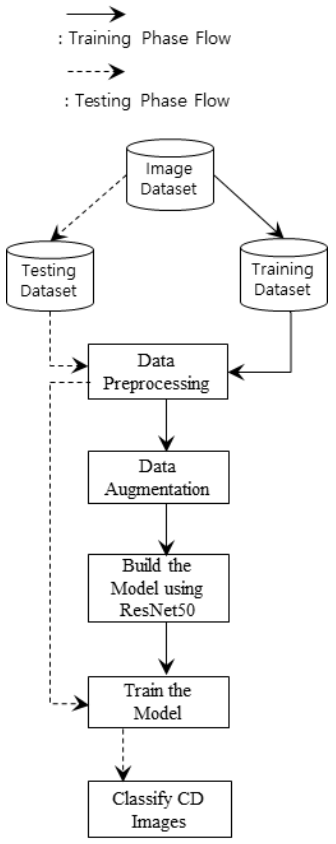

Fig. 3. Build a Class Diagram Image Classifier

3.2 CD 이미지 도메인 카테고리별 분류 모델

단계 1에서 식별된 클래스 다이어그램 이미지들은 다음과 같은 절차를 통해 카테고리에 따라 분류된다.

- ❑ 클래스 다이어그램에서 클래스 추출(YOLOv5 알고리즘 적용)
- ❑ 클래스에서 클래스 이름 추출(OCR 적용)

- ❑ 클래스 이름 목록과 카테고리 목록의 유사도 계산 (Gensim 라이브러리 사용)
- ❑ 주어진 CD 이미지는 유사도가 가장 높은 카테고리로 분류됨

실제 클래스 다이어그램 이미지들은 전체 분류 과정 중 단계 2의 입력으로 사용되며 최종적으로 카테고리별로 분류된다. 먼저, YOLOv5 알고리즘[11]을 사용하여 클래스 다이어그램 이미지에 포함된 클래스들을 추출한다. 예를 들어, Fig. 1에 표현된 *Order Management System* 클래스 다이어그램 이미지에서는 Fig. 4와 같은 클래스 이미지들이 추출될 수 있다. 클래스 다이어그램 이미지에 포함된 총 7개의 클래스들이 추출되었음을 알 수 있다. 이 과정에서 클래스들의 관계성은 제외되고 오직 클래스만이 별도의 이미지로 추출된다. OMG의 UML 표준 명세서 [12]는 클래스 다이어그램 관련 기호와 표기법을 제공한다. 일반적으로 클래스는 직사각형 표기법을 사용하며, 직사각형을 세 영역으로 나누어서 첫 번째 구획은 클래스 이름을 표현한다. 경우에 따라 〈〈 〉〉 표기법을 사용하여 스테레오타입(Stereotype)을 표기할 수 있다. 스테레오타입은 UML이 제공하는 다이어그램 확장성 메커니즘 중의 하나이며, 기본 어휘를 확장하여 특정 도메인이나 특수 용도에 적합한 속성을 표현할 때 사용된다. 예를 들어, 클래스 종류가 인터페이스나 추상 클래스인 경우 〈〈interface〉〉, 〈〈abstract〉〉와 같이 스테레오타입을 명시적으로 클래스 이름 상단에 표시한다. 클래스의 두 번째 및 세 번째 구획은 각각 클래스의 속성과 오퍼레이션을 표현하는 영역이다. 가장 단순한 형태의 클래스 표현은 클래스 이름만 나타내며, 속성이나 오퍼레이션 구획은 생략될 수 있다. 클래스 추출을 위해 상기와 같은 내용들을 고려해야 한다.

식별된 개개의 클래스 이미지들은 OCR 모듈을 통해 클래스 이름들이 텍스트로 추출된다. 즉, OCR 모듈의 입력은 클래스 이미지이며, 출력은 이미지에 포함된 클래스의 이름에 대한 문자열이다. 추출된 개개의 클래스 이미지는 별도의 이미지 파일 형식으로 저장되며 파일명은 클래스 이름을 동일하게 사용한다. 예를 들어, *Account* 클래스는 *Account.jpg*로 저장되며 파일명인 *Account*는 클래스 이름과 동일하다. 클래스 이름을 추출하기 위해 OCR 라이브러리인 EasyOCR [13]을 사용한다.

마지막으로 카테고리별 클래스 다이어그램 분류를 위해, 추출된 클래스 이름 목록과 사전에 정의된 카테고리 목록의 유사도를 계산한다. 즉, 카테고리별 이미지 분류를 위해 클래스 이름과 각각의 카테고리와의 유사도에 따라 특정 카테고리로 분류된다. 두 문자열 리스트들의 유사도 계산을 위해 Gensim 라이브러리 [14]를 사용한다. 유사도는 코사인 유사도를 사용하며 -1에서 +1까지의 값을 가진다. -1인 경우는 서로 완전히 반대, +1인 경우는 서로 완전히 같은 경우를 의미한다. 카테코리 목록은 사전에 정해지며, 입력된 클래스 이름 목록과 카테고리 목록들의 유사도를 점수화한다. 가장 큰 점수를 가진 카테고리가 입력된 클래스 이름 목록을 대표하는 카테고리로 결정된다. 예를 들어, Fig. 4에서 추출된 클래스 이름은 “Order”, “Status”, “Account”, “Transaction”, “Shopping”, “Basket”, “Stock”, “Item”, “Line”이다. 사전 정의된 카테고리 목록은 “Banking”, “Library”, “Flight”, “School”, “Order”을 포함한다.

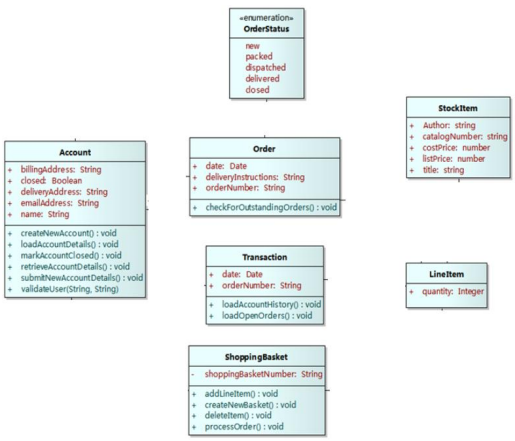

Fig. 4. Class Images in the *Order Management System* Class Diagram Image

4. 클래스 다이어그램 이미지 분류 실험

본 장에서는 클래스 다이어그램 이미지 분류 모델과 도메인 카테고리 세부 분류 모델을 이미지 데이터에 적용한 실험 및 실험 결과에 대한 분석을 기술한다.

4.1 클래스 다이어그램 이미지 분류기

클래스 다이어그램 이미지 분류를 위한 딥러닝 모델의 성능 평가를 위해 혼동 행렬(Confusion Matrix)을

사용한다. 혼동 행렬은 이미지 분류 결과를 True Positive (TP), False Negative (FN), False Positive (FP), True Negative (TN)로 고려한다. TP는 실제 클래스 다이어그램 이미지를 올바르게 예측한 경우를, FP는 클래스 다이어그램이 아닌 이미지를 클래스 다이어그램으로 잘 못 예측한 경우를 의미한다. FN은 실제 클래스 다이어그램 이미지를 올바르게 예측하지 못한 경우를, TN은 실제로 클래스 다이어그램을 포함하지 않은 이미지를 정확하게 비클래스 다이어그램 이미지로 예측한 경우를 나타낸다.

클래스 다이어그램 이미지 분류를 위한 딥러닝 모델의 성능 평가를 위해 대표적인 모델 평가 지표인 정밀도(Precision), 재현율(Recall), F1점수(F1-Score), 정확도(Accuracy)를 사용한다. 클래스 다이어그램 이미지 분류 모델의 정밀도는 분류기가 클래스 다이어그램이라고 예측한 이미지 중에서 실제로 클래스 다이어그램 이미지인 비율을 나타낸다. 재현율은 실제 클래스 다이어그램 이미지 중에서 분류기가 클래스 다이어그램 이미지라고 예측한 비율을 나타낸다. F1점수는 정밀도와 재현율의 조화평균으로 해당 값이 클수록 예측 모델의 성능이 좋다고 판단할 수 있다. 정확도는 시험용 입력 데이터를 얼마나 정확하게 예측 했는지를 나타낸다.

본 논문의 실험을 위한 딥러닝 개발환경으로 구글 코랩을 사용하며 개발 언어로 Python과 PyTorch 라이브러리가 사용된다. Table 1은 실험에 사용된 클래스 다이어그램과 비클래스 다이어그램으로 구성된 이미지 데이터 세트이다. 이미지 데이터는 훈련용, 검증용, 시험용 데이터로 분류된다.

Table 1. A Dataset of Class Diagram and Non-Class Diagram Images

Image Types	Training	Val.	Testing	
Class Diagrams	406	137	136	679
Non-Class Diagrams	399	133	133	665
Total	805	270	269	1,344

Fig. 5는 클래스 다이어그램 이미지 분류 모델의 훈련 과정 손실값과 정확도를 나타낸다. 훈련 과정은 훈련용 데이터 세트를 이용한 단계와 검증용 데이터 세트를 이용하는 단계로 구성된다. train_loss와 train_acc는 훈련 과정의 손실값과 정확도를 나타내고, val_loss와 val_acc는 검증 과정의 손실값과 정확도를 나타낸다. 에포크를 100으로 설정하여 훈련을 시작했으나, 조기 종료(Early Stopping)에 따라 에포크 값이 10이 된 경우에, 학습이 조기 종료되었다. train_loss와 val_loss는 각각 0.05와 0.09이며 train_acc와 val_acc는 각각 98.0%와 97.4%이다. 전반적으로 제안된 모델의 학습이 적절하게 수렴함을 알 수 있다.

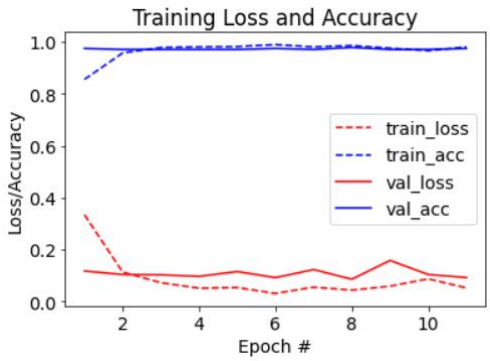

Fig. 5. Training Loss and Accuracy of Classifying UML Class Diagram Images

Table 2는 ResNet50 기반의 클래스 다이어그램 이미지 분류기의 성능 측정결과이다. 성능 평가 지표로는 정밀도, 재현율, F1점수, 정확도가 사용된다. 테스팅을 위해269개의 이미지 데이터가 사용되었으며 Table 2의 상단은 혼동 행렬을 나타낸다. 6장의 이미지에 대한 False Negative 예측을 제외한 나머지 이미지에 대해서는 적절하게 예측했음을 알 수 있다. Table 2의 하단은 성능 평가 지표에 대한 계산 결과이다. 정밀도, 재현율, F1점수, 정확도에서 각각 100.00%, 95.59%, 97.74%, 97.77%를 달성했다.

Table 2. Performance Measures of the Proposed Class Diagram Image Classifier

		Prediction Values	
		CD	NCD
Actual Values	CD	TP: 130	FN: 6
	NCD	FP: 0	TN: 133

Eval. Metrics	Formula	Values(%)
Precision	TP/(TP+FP)	100.00
Recall	TP/(TP+FN)	95.59
F1-Score	2*(P*R)/(P+R)	97.74
Accuracy	(TP+TN)/ (TP+FN+FP+TN)	97.77

학습이 완료된 클래스 다이어그램 이미지 분류기는 Fig. 6과 같이 임의의 클래스 다이어그램 이미지 한 장에 대한 예측을 수행할 수 있다. 분류기에 임의의 클래스 다이어그램 이미지 한 장(Fig. 6 상단 참고)을 입력으로 한 경우, 해당 이미지가 클래스 다이어그램인지를 판단한다 (Fig. 6 하단 참고). 주어진 이미지는 100%에 가까운 확신으로 클래스 다이어그램이라고 판단한다.

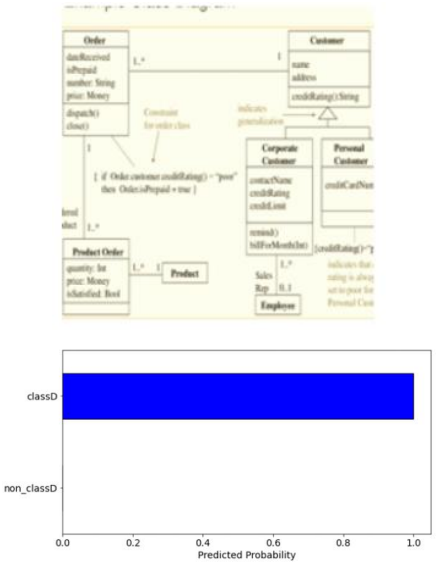

Fig. 6. A CD Image and the Prediction Result of the Proposed Image Classifier

4.2 클래스 다이어그램 이미지 카테고리별 세부 분류기

클래스 다이어그램 이미지 분류 모델을 통해 식별된 실제 이미지들은 카테고리 분류를 위한 데이터로 사용된다. 실제 클래스 다이어그램 이미지를 카테고리별로 분류하는 작업이 수행된다. Table 3은 클래스 이름들에 대한 카테고리별 유사도 점수를 계산하는 예를 제시한다. 클래스 이름 목록에는 “Account”, “Current”, “Customer”, “Bank”, “Deposit”, “Banking”, “System”, “Personal”, “Business”가 포함된다. 사전 정의된 카테고리 목록에는 “Banking”, “Library”, “Flight”, “School”, “Order”가 포함된다. 주어진 클래스 이름들을 대표하는 카테고리는 “Banking”으로 유사도 점수가 0.46임을 알 수 있다. 반면, 가장 거리가 있는 카테고리는 “Order”이며 0.04의 유사도를 가진다.

Table 3. Similarity Score (SS) by Class Names

Class Names	Account, Current, Customer, Bank, Deposit, Banking, System, Personal, Business				
Category	Banking	Library	Flight	School	Order
SS	**0.46**	0.19	0.08	0.06	0.04

Table 4는 총 196개의 이미지 데이터에 적용한 카테고리 분류 결과이다. 첫 번째 컬럼은 5개의 카테고리를 나타낸다. 두 번째 컬럼은 각 카테고리에 속하는 클래스 다이어그램 이미지 개수를 나타낸다. 세 번째 컬럼은 제안된 카테고리 분류 모델을 통해 식별된 이미지 개수를 나타낸다. 마지막 컬럼은 두 번째와 세 번째 값을 통해 카테고리 분류의 정확도를 제공한다. 본 실험에서는 카테고리 분류의 신뢰성을 높이기 위해 유사도 점수가 0.2 이상인 경우만을 고려했다. 유사도 점수의 임계 값이 1에 가까울수록 좀 더 유사함을 의미한다. 유사도의 임계값 설정에 따라 식별된 클래스 다이어그램 이미지의 개수는 변화한다. 임계 값이 클수록 식별된 이미지의 개수가 작아진다. 클래스 이름 기반의 카테고리별 분류 정확도 점수가 81.1%와 95.2% 사이에 분포함을 알 수 있다.

Table 4. Categorizing UML Class Diagram Images

Categories	# of CD Images	# of Found CD Images	Accuracy(%)
Banking	42	40	**95.2**
Flight	48	45	93.8
Library	37	30	**81.1**
Order	33	28	84.9
School	36	32	88.9
Total	196	175	89.3

클래스 이름에 따른 이미지 카테고리 분류 결과는 Table 5의 예제와 같이 JSON 형식으로 저장된다. 클래스 다이어그램 이미지 이름(CD_Img_Name), 식별된 클래스 다이어그램 이름 목록(Org_Class_Name), 확장된 클래스 다이어그램 이름 목록(EX_Class_Name), 카테고리별 유사도 점수(Similarity_Score), 최종 카테고리 이름(Category_Name)을 저장한다. 최종 카테고리는 입력된 클래스 다이어그램 이미지의 유사도 점수 중 가장 높은 점수를 가지는 카테고리를 나타낸다.

Table 5. A JSON Format for the Class Diagram Categorization

```
{
  "CD_Img_Name": a name of a class diagram
  "Org_Class_Name": a list of class names,
  "EX_Class_Name": a list of extended class names,
  "Similarity_Score": a list of pairs(category, similarity score)
  "Category_Name": a name of a category
}
```

Table 6은 UML 클래스 다이어그램 이미지를 분류하는 기존의 모델들과 본 논문에서 제시한 모델을 비교한다. 기존 분류 모델들의 프로그램 코드가 공개되어 있지 않아 본 논문의 데이터 세트를 사용하여 각 모델들의 성능을 평가할 수는 없지만, 각 논문에서 제공하는 모델의 분류 결과를 근거로 한다. 기존 분류 모델들이 사용하는 이미지 데이터의 크기와 알고리즘들이 상이하므로 직접적인 정확도 비교는 어려운 것이 실상이다. 제안된 이미지 분류 모델의 정확도는 제한적이지만 나쁘지 않으며 특히, 카테고리 별 분류 기능 추가는 의미가 있다고 판단된다.

Table 6. Comparison between the Proposed Model and Prior Classification Models

Authors	# of Images	Algorithm/Model	Accuracy(%)	Remarks
Gosals [2]	3,282	CNN w/ & w/o Regularization	w/ Reg.: 86.6 w/o Reg.: 85.4	UML Image Classification
Shcherban [3]	3,231	Modified MobileNetV3 & ResNet	98.7	Multiclass Image Classification Transfer Learning
Moreno [4]	18,899	Machine Learning	94.4	UML Image Classification
Munialo [6]	2,000	ResNet50 CNN Tesseract OCR	IC: 95 ~ 98 TE: 91 ~ 98	Image Classification (IC) Text Extraction (TE)
Proposed Approach	1,344	ResNet50 CNN EasyOCR	IC: 97.8 CCD: 89.3	Transfer Learning Image Classification (IC) Categorization of Class Diagrams (CCD)

5. 결론 및 향후연구방향

본 논문은 클래스 다이어그램 이미지 데이터를 도메인 카테고리에 따라 자동 분류하기 위한 딥러닝 기반의 자동 분류 접근법을 제안한다. 제안된 이미지 이진 분류 모델은 웹 상의 개방형 저장소나 인터넷 이미지 검색을 통해 수집된 이미지 데이터가 실제로 클래스 다이어그램인지를 딥러닝 모델을 사용하여 판단한다. 또한, 분류된 클래스 다이어그램 이미지에서 클래스 이름들을 추출하고 이를 기반으로 클래스 다이어그램 이미지를 세부 카테고리로 분류한다. 제안된 이미지 분류 모델은 정밀도, 재현율, F1점수, 정확도에서 각각 100.00%, 95.59%, 97.74%, 97.77%를 달성했다. 클래스 이름 기반의 카테고리별 분류 정확도 점수는 81.1%와 95.2% 사이에 분포한다. 본 논문의 실험에서 사용된 클래스 다이어그램 이미지 데이터의 개수가 충분히 크지 않았지만, 도출된 실험 결과는 의미가 있는 것으로 판단한다. 향후 연구 내용으로 UML 클래스 다이어그램 이미지 데이터의 개수를 늘려서 주어진 자동 분류 기법을 적용하는 실험이 필요하다. 또한 카테고리별 이미지 분류의 효과를 높이기 위해 클래스 이름들이 적절하게 모델링된 양질의 클래스 다이어그램 이미지를 수집하는 것이 요구된다.

REFERENCES

[1] Unified Modeling Language (UML). (2022) https://www.uml.org/

[2] B. Gosala, S. R. Chowdhuri, J. Singh, M. Gupta. & A. Mishra. (2021). Automatic Classification of UML Class Diagrams Using Deep Learning Technique: Convolutional Neural Network. *Applied Sciences 11(9).*
DOI : 10.3390/app11094267

[3] S. Shcherban, P. Liang, Z. Li & C. Yang. (2021). Multiclass Classification of Four Types of UML Diagrams from Images Using Deep Learning. *In Proceedings of the 33rd International Conference on Software Engineering and Knowledge Engineering (SEKE 2021).*
DOI : 10.18293/SEKE2021-185

[4] V. Moreno, G. Génova, M. Alejandres, & A. Fraga. (2020). Automatic Classification of Web Images as

UML Static Diagrams Using Machine Learning Techniques. *Applied Sciences, 10(7).*
DOI : 10.3390/app10072406

[5] J. K. Nurminen1, K. Rainio, J. Numminen, T. Syrjanen, N. Paganus & K. Honkoila. (2019). Object Detection in Design Diagrams with Machine learning. *Progress in Computer Recognition Systems CORES 2019*. Advances in Intelligent Systems and Computing.
DOI : 10.1007/978-3-030-19738-4_4

[6] S. W. Munialo, G. M. Muketha & K. K. Omieno. (2020). Automated Feature Extraction from UML Images to Measure SOA Size. *International Journal of Recent Technology and Engineering (IJRTE). 9(2).*
DOI : 10.35940/ijrte.B4131.079220

[7] ResNet50. (2022).
https://pytorch.org/hub/pytorch_vision_resnet/

[8] K. Mangaroliya & H. Patel. (2020). Classification of Reverse-Engineered Class Diagram and Forward-Engineered Class Diagram using Machine Learning. *arXiv:2011.07313*.

[9] G. Robles, T. Ho-Quang, R. Hebig, M. R. V. Chaudron & A. Fernandez. (2017). An extensive dataset of UML models in GitHub. *In Proceedings of the 14th International Conference on Mining Software Repositories (MSR)*, IEEE.
DOI : 10.1109/MSR.2017.48

[10] R. Hebig, T. H. Quang, M. R.V. Chaudron, G. Robles & M. A. Fernandez. (2016). The Quest for Open Source Projects that use UML Mining GitHub. *In Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems.*
DOI : 10.1145/2976767.2976778

[11] YOLOv5. (2022).
https://pytorch.org/hub/ultralytics_yolov5/

[12] UML Specification 2.5.1. (2022).
https://www.omg.org/spec/UML/

[13] EasyOCR. (2022).
https://github.com/JaidedAI/EasyOCR

[14] Gensim. (2022).
https://radimrehurek.com/gensim/index.html

김 동 관(Dong Kwan Kim) [정회원]

· 1993년 2월 : 숭실대학교 전산학과 (공학사)
· 1998년 2월 : 숭실대학교 전산학과 (공학석사)
· 2009년 7월 : Virginia Tech 컴퓨터과학과(공학박사)
· 2013년 3월 ~ 현재 : 목포해양대학교 해양컴퓨터공학과 교수
· 관심분야 : 딥러닝 알고리즘, 빅 데이터 분석, 소프트웨어 공학, 런타임 시스템
· E-Mail : dongkwan@mmu.ac.kr