

인공지능 기반 객체 인식을 위한 최적 학습모델 구축 방안에 관한 연구*

양 환 석*

요 약

4차 산업혁명으로 많은 산업 분야에 커다란 변화가 일어나고 있으며, 그중에서도 인공지능을 활용한 융합기술에 활발한 연구가 진행되고 있다. 그중에서도 인공지능을 활용한 객체 인식과 인식 결과를 활용한 디지털 전환(Digital Transformation) 분야에서 그 요구가 나날이 증가하고 있다. 본 논문에서는 이미지내에 글자, 심볼, 선을 정확하게 인식하고 인식 결과를 시뮬레이션에 활용할 수 있도록 표준화 포맷의 파일로 저장하기 위해 최적의 학습모델 구축 방법을 제안하였다. 이미지내 글자, 심볼, 선을 인식하기 위하여 인식 대상별 특성을 분석한 후 최적의 인식 기법을 선택하였다. 그다음으로 인식 대상별 인식률을 향상시키기 위하여 최적의 학습 모델 구축 방안을 제안하였다. 글자, 심볼, 선 인식의 순서와 가중치를 다르게 설정하여 인식 결과를 확인하였으며, 인식 후처리에 대한 방안도 마련하였다. 최종적인 인식 결과는 시뮬레이션 등 다양한 처리에 활용될 수 있는 표준화 포맷으로 저장하였다. 본 논문에서 제안한 최적의 학습 모델 구축에 대한 우수한 성능은 실험을 통해 확인할 수 있었다.

A Study on How to Build an Optimal Learning Model for Artificial Intelligence-based Object Recognition

Yang Hwan Seok*

ABSTRACT

The Fourth Industrial Revolution is bringing about great changes in many industrial fields, and among them, active research is being conducted on convergence technology using artificial intelligence. Among them, the demand is increasing day by day in the field of object recognition using artificial intelligence and digital transformation using recognition results. In this paper, we proposed an optimal learning model construction method to accurately recognize letters, symbols, and lines in images and save the recognition results as files in a standardized format so that they can be used in simulations. In order to recognize letters, symbols, and lines in images, the characteristics of each recognition target were analyzed and the optimal recognition technique was selected. Next, a method to build an optimal learning model was proposed to improve the recognition rate for each recognition target. The recognition results were confirmed by setting different order and weights for character, symbol, and line recognition, and a plan for recognition post-processing was also prepared. The final recognition results were saved in a standardized format that can be used for various processing such as simulation. The excellent performance of building the optimal learning model proposed in this paper was confirmed through experiments.

Key words : Object Recognition, YOLO, CNN, Artificial Intelligence

접수일(2023년 11월 28일), 게재확정일(2023년 12월 28일)

* 중부대학교/정보보호학과

★본 연구는 2023년도 중소벤처기업부의 산학연 Collabo R&D (사업화R&D) 지원에 의한 연구임. [RS-2023-00224780]

1. 서 론

최근들어 이미지 또는 영상 내에서 담겨져 있는 다양한 객체들을 인식하고 이를 활용하기 위한 기술들이 다양한 분야에서 활발히 진행되고 있다. 특히 다양한 센서 기술과 인공지능을 이용한 자율주행 자동차, 디지털 전환(Digital Transformation, DX), 가상 제어 시스템 등에서 많은 연구가 진행되고 있다[1]. 이와 같이 영상내 다양한 객체 인식 정보를 공학적으로 응용할 수 있다면 그 활용도는 매우 높을 것이다. 특히 최근에는 영상내의 관심 영역 설정 및 객체 식별 단계에 인공지능 알고리즘을 적용하여 신속하고 정확한 결과를 얻기 위한 다양한 처리 기술들이 나오고 있다[2]. 영상내 객체 인식에 인공지능을 적용하기 위해서는 학습 데이터에 대한 라벨링을 통해 보다 정확한 학습 모델 구축이 매우 중요하다고 할 수 있다[3]. 객체 인식을 위한 인공지능 기법은 CNN(Convolutional Neural Network), LSTM(Long Short-Term Memory), YOLO(You Only Look Once) 등 다양한 기법이 사용되고 있으며 이들 모두 특징을 추출하고 객체를 분류한 후 인식을 수행하는 특징이 있다[4].

본 논문에서는 도면 이미지의 특징을 분석하여 인식 대상인 선, 글자, 심볼 인식을 위해 입력 이미지에 대한 전처리를 수행하고 객체 인식률을 향상시키기 위하여 각 인식 대상별 최적의 인공지능 알고리즘 선정 및 인식 과정을 최적화하여 객체를 효과적으로 인식할 수 있는 학습 모델 구축 방안을 제안한다. 또한 이러한 인식 결과를 다양한 응용에 활용될 수 있도록 표준 포맷에 저장하는 방안을 제시하고자 한다.

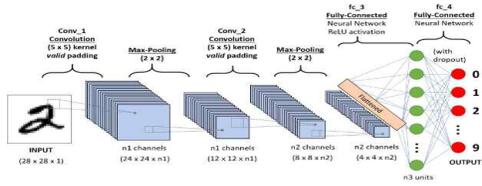
본 논문의 구성은 다음과 같다. 2장에서는 객체 인식을 위해 사용되는 대표적인 인공지능 기법에 대하여 살펴보고 3장에서는 본 논문에서 제안한 최적 학습모델 구축 방안 기법에 관해 기술하였다. 4장에서는 제안한 기법의 성능 평가를 위해 실험하고 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

머신러닝은 컴퓨터 시스템이 데이터로부터 학습하도록 설계되는 인공 지능의 한 분야로서 명시적인 프로그래밍 없이 컴퓨터가 데이터를 기반으로 스스로 학습하여 작업을 수행하게 하는 것이다. 머신 러닝은 크게 지도 학습, 비지도 학습, 강화 학습으로 나눌 수 있다. 딥 러닝은 머신 러닝의 한 분야로, 다층 신경망을 사용하여 복잡한 패턴과 특징을 학습하도록 설계된 것이다. 딥러닝은 대량의 데이터와 강력한 컴퓨팅 리소스를 필요로 하는데, 이러한 리소스의 확장으로 심층적인 구조를 갖는 모델이 효과적으로 학습될 수 있게 되었다. 머신 러닝과 딥러닝의 차이점은 크게 데이터 표현과 학습 방법, 모델의 복잡성, 적용 분야에 있다.

2.1 CNN (Convolutional Neural Network)

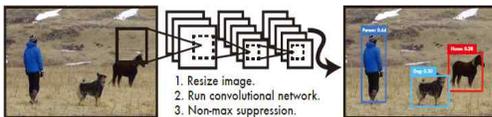
CNN은 주로 이미지 처리에 사용되는 딥러닝 모델로, 이미지에서 패턴 및 특징을 효과적으로 추출할 수 있다. CNN은 합성곱층(Convolutional Layer)과 풀링층(Pooling Layer)으로 구성되며, 일반적으로 여러 층으로 쌓아서 사용된다. 합성곱층은 특징 맵 (Feature Map), 스트라이드 (Stride), 패딩 (Padding)이로 이루어진다[5]. 먼저 특징 맵은 여러 개의 필터(또는 커널)를 사용하여 입력 이미지에서 특징을 추출한다. 각 필터는 작은 영역의 가중치와 연결되어 이 영역의 특징을 감지한다. 스트라이드는 필터가 입력 위를 이동하는 간격을 나타낸다. 더 큰 스트라이드는 출력 크기를 줄이고 계산 비용을 감소시킨다. 패딩은 입력 주변에 추가되는 가상의 픽셀로, 출력 크기를 조절하고 모서리의 정보 손실을 막는다. 풀링층은 공간적인 불변성(Spatial Invariance)을 제공하고, 작은 변화에 강인한 모델을 만드는 역할을 수행하며, 최대 풀링(Max Pooling) 또는 평균 풀링(Average Pooling)을 사용하여 특징 맵의 크기를 줄이는 풀링 연산을 수행한다.



(그림 1) CNN 동작 과정

2.2 YOLO(You Only Look Once)

YOLO는 객체 감지(Object Detection) 작업을 위한 딥러닝 기반 알고리즘 중 하나로, 실시간으로 객체를 탐지하는 데 특히 효과적이다. YOLO의 주요 특징은 다음과 같다[6]. 한 번에 바라보기 특징은 이미지를 한 번에 바라보면서 모든 객체를 탐지한다. 다른 기존 객체 감지 알고리즘과는 달리 여러 단계의 프로세스를 거치지 않고, 전체 이미지에 대한 객체를 한 번에 예측한다. YOLO를 이용한 객체 탐지 과정은 다음과 같다. 먼저, 입력 이미지를 S x S 크기의 그리드로 나눈 후 각 그리드 셀은 여러 개의 바운딩 박스와 클래스 확률을 예측하는 그리드 기반 객체 탐지를 수행한다. 그런 후 각 그리드 셀에서는 여러 개의 바운딩 박스를 예측한다. 각 바운딩 박스는 중심 좌표(x, y), 너비와 높이(w, h)로 표현된다. 각 바운딩 박스에 대해 해당 객체의 클래스 확률을 예측한다. 클래스 확률은 softmax 함수를 사용하여 정규화된다. YOLO는 실시간 객체 감지에 특히 효과적이며, 높은 성능을 제공한다. 이 특성은 실시간 비디오나 실제 환경에서의 응용에 적합하다. <그림 2>는 YOLO 물체 인식 과정을 보여주고 있다.



(그림 2) YOLO 물체 인식 과정

2.3 LSTM(Long Short-Term Memory)

순환 신경망(Recurrent Neural Network, RNN)

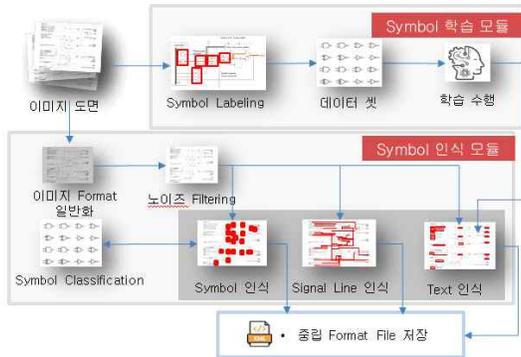
의 한 변형으로, 시퀀스 데이터를 처리하는 데 강점을 가지고 있다. LSTM은 시간에 따른 의존성을 효과적으로 학습할 수 있어서, 긴 시퀀스 데이터에서 발생하는 그레디언트 소실 문제를 완화할 수 있다[7]. LSTM은 셀 상태라고 불리는 중요한 구조를 가지고 있다. 셀 상태는 정보를 나타내는 메모리 역할을 하며, 정보의 추가 및 삭제가 가능하다. 셀 상태는 입력 게이트와 망각 게이트를 사용하여 업데이트되며, 이러한 업데이트는 시그모이드 함수를 통해 게이트의 열림 여부를 결정한다. LSTM은 이러한 구조를 통해 장기 의존성 문제를 해결하고, 긴 시퀀스에서 발생하는 그레디언트 소실 문제를 완화하여 효과적으로 시퀀스 데이터를 모델링할 수 있다.

3. 인공지능 기반 최적 학습모델 구축

본 장에서는 인공지능을 기반으로 한 정확한 객체 인식을 위해 본 논문에서 제안한 최적 학습모델 구축 기법에 대하여 설명하였다. 제안한 기법은 이미지내 객체 인식 성능을 최대화 시키기 위해 객체들의 특성을 고려한 알고리즘을 적용하였다.

3.1 시스템 개요

본 논문에서는 다양한 이미지 도면내의 선, 심볼, 글자들을 인공지능 기반으로 인식한 후 인식된 결과를 표준화된 데이터 포맷으로 생성될 수 있도록 하는 것을 목표로 한다. 이를 위해서는 이미지내의 객체들을 정확하게 인식시키는 과정이 매우 중요하다고 할 수 있다. 따라서 이를 위해서는 선, 심볼, 글자들에 대한 최적의 학습모델을 구축하는 것이 중요하며, 이를 위해서는 첫 번째 단계로 각 인식 대상들에 대한 특징을 정확하게 분석해야 한다. 본 논문에서는 이러한 특성들을 고려하여 다음과 같은 절차를 통해 객체 인식을 수행한다. <그림 3>은 본 논문에서 제안한 기법의 최적 학습모델 구축을 위한 과정을 보여주고 있다.



(그림 3) 최적 학습모델 구축 과정

3.2 객체 인식 최적 모델

영상내 객체 인식 대상은 선, 심볼, 글자로 이루어져 있다. 각 객체를 위한 최적 모델을 구축하기 위해서는 각 객체들의 특징을 고려해야 한다. 먼저, 선 인식은 인공지능 알고리즘을 이용하는 것은 매우 비효율적이며, 기존의 직선 인식 알고리즘 등을 적용하는 것이 효율적이다. 도면내 객체별 인식 순서는 첫 번째 단계로 글자를 인식한다. 그 이유는 심볼내 글자가 존재하고, 선 인식을 먼저 수행하면, 심볼의 수직, 수평선 그리고 글자를 이루고 있는 부분도 선으로 인식하기 때문에 글자를 가장 먼저 인식하게 된다. 두 번째 단계로 심볼을 인식한다. 그리고 마지막 단계로 선 인식을 수행하게 된다.

3.2.1 글자 인식

분할된(500×300) 이미지에서 YOLO를 이용하여 문자를 인식한 후 인식 문자들에 대한 각 좌표를 추출하여 원본 도면에서 좌표를 계산한다. 그리고 도면을 분할할 때 경계 부분에서 잘린 글자를 인식하기 위하여 x와 y 좌표값에 각각 50만큼 더한 후 재분할하여 다시 인식을 실행한다. 인식된 글자들에 대해 원본 도면에서 좌표를 계산한 후 여러 번 겹쳐서 인식된 글자는 하나만 남기고 모두 삭제한다. 문자 인식 결과 정보를 저장한다. 인식된 문자들에 대해 다음과 같은 후처리를 실시한다. 첫 번째로 한 문자씩 인식된 문자들을 단어 단위로 묶어서 저장할 수 있도록 한다. 두 번째로 단어 단위로

보정된 인식 문자를 수평 방향으로 가지런하게 배치될 수 있도록 한다.

3.2.2 심볼 인식

문자 인식에서 사용했던 YOLO를 사용해서 분할된 도면에서 심볼을 인식한 후 인식된 심볼들에 대한 좌표를 계산한다. 도면 분할할 시 경계 부분에서 잘린 심볼을 인식하기 위하여 x와 y 좌표 값에 각각 심볼 중 가장 긴 길이만큼 더한 후 분할하여 재인식을 실행한다. 인식된 심볼들에 대해 원본 도면에서 좌표를 계산 후 여러 번 겹쳐서 인식된 심볼은 하나만 남기고 모두 삭제한다. LINK-OFF 심볼을 인식할 때는 심볼 안의 문자로 인해 인식률이 낮아져 문자를 제거한 후 인식을 실행한다. 심볼 인식 결과 정보를 저장한다.

3.2.3 선 인식

인식 영역 설정 및 잡음 제거가 이루어진 도면의 좌측상단을 시작으로 좌측에서 우측 방향으로 모든 픽셀의 색을 검사하여 수평선을 검출한다. 수평선을 찾는 방식과 비슷한 방식으로 좌측상단을 시작으로 위에서 아래 방향으로 이미지 마지막까지 모든 픽셀의 색상(검정)을 검사하여 수직선을 검출한다. 수집된 수평선과 수직선의 교차하는 위치를 수집한 후 선 정보에서 시작, 끝점의 좌표, 길이, 교차점의 위치와 교차되는 수직선 또는 수평선의 인덱스를 저장한다. 인식된 심볼 영역 내에서 인식된 선들은 삭제 처리한다. 선 인식의 후처리는 문자와 심볼의 일부가 선으로 인식되는 현상을 없애기 위해서 인식된 문자와 심볼의 위치와 중첩 여부를 검사한 후 오인식된 선은 제거하는 과정을 거치게 된다.

앞에서 설명한 세 단계의 인식 과정이 모두 끝나면 그 다음 단계는 인식 결과를 확인하는 단계이다. 원본 도면위에 인식된 문자, 심볼, 선을 색으로 구분하여 출력한 후 인식 결과를 확인한다. 잘못 인식된 문자, 심볼, 선에 대해서 사용자가 수정할 수 있는 과정을 거친 후 최종 적으로 글자, 심볼,

선의 인식이 완료되면 XML 파일로 그 결과를 저장한다. <그림 4>는 XML 변환 요소의 일부를 보여주고 있다.

```

<DrawingDataDTOs>
  <DrawingDataDTO>
    <DrawingDTO>
      </DrawingDTO>
    <DwgNodeDTOs>
      </DwgNodeDTOs>
    <SymbolDTOs>
      </SymbolDTOs>
    <TextDTOs>
      </TextDTOs>
    <LineDTOs>
      </LineDTOs>
    </DrawingDataDTO>
  </DrawingDataDTOs>
  
```

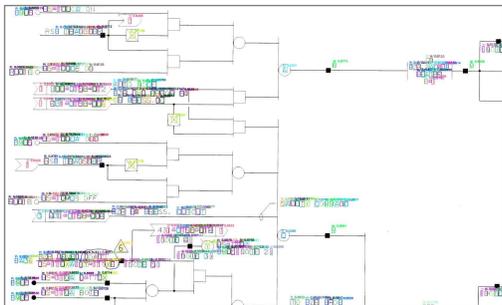
(그림 4) XML 변환 요소

4. 실험 및 결과

4.1 실험 환경

본 논문에서는 제한한 기법의 성능 평가를 위하여 글자와 심볼들에 대해서 각각 50개씩 라벨링을 수행하였다. 그리고 학습모델을 구축하기 위하여 CPU i9, 메모리 128G, GPU Geforce RTX 3070을 이용하였다. 학습은 10만번을 수행하였으며, 각 객체별로 서로 다른 가중치를 적용하였다.

4.2 실험 결과

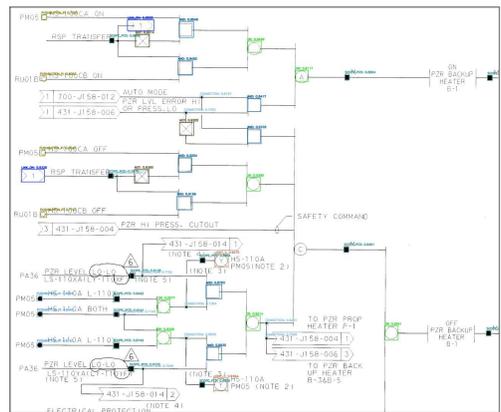


(그림 5) 글자 인식 결과

글자 인식은 각 영문자와 특수문자 한 글자씩 라벨링을 수행한 후 학습 모델을 구축하였다. 그리

고 글자를 하나의 객체로 인식하는 YOLO를 적용하여 글자 인식을 수행하였다. 모든 글자들에 대한 가중치를 동일하게 적용하였을 때 인식률이 떨어지는 결과를 보이는 글자로 인해 글자별 가중치는 다르게 설정하였다. <그림 5>에서 보여주고 있다.

심볼 인식의 성능을 향상시키기 위하여 도면 전체 크기를 그대로 학습하지 않고 도면을 분할하여 라벨링 및 학습 모델을 구축하였다. 탐지가 되는 confidence score를 조절하거나 추가 학습을 통해 미탐지를 방지하여 인식률을 향상시킬 수 있었다. 글자 인식 후 이미지에서 인식된 글자를 삭제한 후 심볼 인식을 수행하여 보다 좋은 인식 결과를 얻을 수 있었다. <그림 6>에서는 심볼 인식 결과를 보여주고 있다.



(그림 6) 심볼 인식 결과

히프 변환을 이용한 선 인식에서는 최적의 인자 값을 찾는 것이 어려워 선 인식 결과가 좋지 않았다. 따라서 본 논문에서는 직접 개발한 선 인식 알고리즘을 이용하여 인식을 수행하였다. 선 인식은 수평선, 수직선, 교차점 인식과정으로 수행하게 된다. 그리고 도면 이미지의 해상도 문제로 인해 하나의 선에서 여러 개의 선이 인식되는 문제 역시 해결하였다. 선 인식 결과는 <그림 7>에서 보여주고 있다.

