

인공지능 기반 객체인식 기법에 관한 연구*

양 환 석*

요 약

최근 들어 4차산업 연관기술인 사이버물리시스템(CPS) 구축을 위해 물리 모델과 제어회로 시뮬레이션을 위한 가상 제어시스템 구축 작업이 다양한 산업 분야에서 요구가 점점 증가하고 있다. 전자 문서화 되지 않은 문서들에 대한 직접 입력을 통한 변환은 시간과 비용이 많이 소모된다. 이를 위해 이미 출력된 대량의 도면을 인공지능을 이용한 객체 인식을 통해 디지털화 작업은 매우 중요하다고 할 수 있다. 본 논문에서는 도면내 객체를 정확하게 인식하고 이를 다양한 응용에 활용할 수 있도록 하기 위하여 도면내 객체의 특징을 분석하여 인공지능을 활용한 인식 기법을 제안하였다. 객체 인식의 성능을 높이기 위하여 객체별 인식 후 그 정보를 저장하는 중간 파일을 생성하게 하였다. 그리고 인식 결과를 도면에서 삭제하여 다음 인식 대상의 인식률을 향상시켰다. 그리고 그 인식 결과를 표준화 포맷 문서로 저장하여 이를 제어시스템의 다양한 분야에 활용할 수 있도록 하였다. 본 논문에서 제안한 기법의 우수한 성능은 위해 실험을 통해 확인할 수 있었다.

A Study on Object Recognition Technique based on Artificial Intelligence

Yang Hwan Seok*

ABSTRACT

Recently, in order to build a cyber physical system(CPS) that is a technology related to the 4th industry, the construction of the virtual control system for physical model and control circuit simulation is increasingly required in various industries. It takes a lot of time and money to convert documents that are not electronically documented through direct input. For this, it is very important to digitize a large number of drawings that have already been printed through object recognition using artificial intelligence. In this paper, in order to accurately recognize objects in drawings and to utilize them in various applications, a recognition technique using artificial intelligence by analyzing the characteristics of objects in drawing was proposed. In order to improve the performance of object recognition, each object was recognized and then an intermediate file storing the information was created. And the recognition rate of the next recognition target was improved by deleting the recognition result from the drawing. In addition, the recognition result was stored as a standardized format document so that it could be utilized in various fields of the control system. The excellent performance of the technique proposed in this paper was confirmed through the experiments.

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

접수일(2022년 09월 29일), 게재확정일(2022년 12월 30일)

* 중부대학교/정보보호학과

★ 본 연구는 2022년도 중소벤처기업부의 산학연 Collabo R&D(예비연구) 지원에 의한 연구임 [S3253352]

1. 서 론

영상 내에서 많은 정보의 가치를 얻기 위해 객체의 종류, 위치, 특징 등을 인식하기 위해 많은 연구가 진행되고 있다[1]. 이와 같이 시각정보 처리능력을 영상인식 등에 공학적으로 응용될 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]. 인공지능 기법을 적용하기 위해서는 사전에 많은 데이터를 이용한 라벨링 및 학습 모델 구축이 객체 인식의 성능을 좌우한다고 할 수 있다[3]. 영상내 객체 인식을 위한 인공지능 기법은 R-CNN, Faster R-CNN, SSD, YOLO 등 다양한 기법이 사용되고 있으며 이들 모두 CNN(Convolutional Neural Network)을 이용하여 특징을 추출하고 입력값을 0에서 1사이의 확률 값으로 변환해주는 softmax를 이용하여 객체를 분류한 후 인식을 수행하는 특징이 있다[4][5].

본 논문에서는 도면의 특징을 분석하여 객체(선, 글자, 심볼) 인식을 위해 입력 이미지에 대한 전처리를 수행하고 객체 인식률을 향상시키기 위한 최적의 인공지능 알고리즘을 선정하여 객체를 효과적으로 식별할 수 있는 모델을 제안한다. 또한 이러한 인식 결과를 표준화 포맷인 XML로 저장하여 이를 다양한 응용에 활용될 수 있도록 하고자 한다.

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

2. 관련 연구

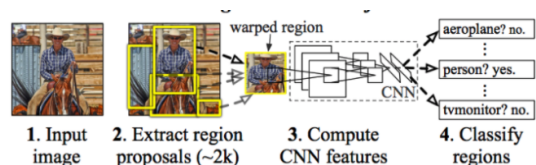
머신러닝은 데이터를 이용해 기계를 훈련해서 스스로 결정을 내리거나 예측을 할 수 있도록 하는

인공지능의 한 방법이다. 이러한 기계학습 알고리즘에는 지도학습 기법, 비지도학습 기법, 준지도학습 기법, 강화학습 기법으로 분류할 수 있다. CNN을 이용한 이미지 대상 머신러닝은 주로 지도학습 기법이 사용된다. 학습할 데이터셋을 작성할 때 직접 이미지에 라벨링을 수행한다. CNN 알고리즘은 이러한 데이터셋을 바탕으로 출력층에서 출력된 값을 피드백하고 각각에 대한 가중치를 조정함으로써 입력층과의 오차를 최소화하는 과정을 반복하게 된다. 학습이 시작이 되는 시점에서는 각각의 타일이 동일하지만 학습이 진행됨에 따라 식별 대상과 관계가 있는 타일에게는 추가적인 가중치가 부여되고 오차가 일정 수준 이하로 줄어들게 되면 학습률을 줄임으로써 목표치에 보다 가깝게 접근하게된다.

2.1 R-CNN

R-CNN은 Object Detection에서 대표적인 2-stage Detector 기법으로서 Image classification을 수행하는 CNN과 localization을 위한 regional proposal 알고리즘을 연결한 모델이다[6]. R-CNN에서는 다음의 세 가지 모듈로 구성되어 있다. 첫 번째는 Object가 있을법한 영역을 찾는 모듈이다. 두 번째는 각각의 영역으로부터 고정된 크기의 Feature Vector를 뽑아낸다. 세 번째는 Classification을 위한 선형 지도학습 모델이다.

R-CNN의 단점은 selective search로 2000개의 region proposal을 뽑고 각 영역마다 CNN을 수행하기 때문에 CNN 연산 * 2000 만큼의 시간이 걸려 수행시간이 매우 느리다.



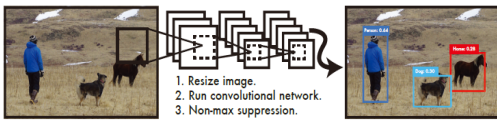
(그림 1) R-CNN 동작 과정

2.2 Faster R-CNN

Fast R-CNN은 2016년에 S. Ren 등이 제안한 기법으로 R-CNN의 느린 처리 속도를 향상시킨 모델이다[7]. Fast R-CNN의 핵심은 RoI(Region of Interest) Pooling이다. R-CNN에서 CNN output이 FC layer의 input으로 들어가야 했기 때문에 CNN input을 동일 size로 맞춰줘야 했다. 따라서 원래 이미지에서 추출한 RoI를 crop, warp을 통해 동일 size로 조정했었다. 그러나 실제로 FC layer의 input이 고정인거지 CNN input은 고정이 아니다. 따라서 CNN에는 입력 이미지 크기, 비율 관계 없이 input으로 들어갈 수 있고 FC layer의 input으로 들어갈 때만 size를 맞춰주기만 하면 된다.

2.3 YOLO

YOLO는 you Only Look Once, 즉 이미지를 한 번 보는 것만으로 Object의 종류와 위치를 추측하는 딥러닝 기반의 물체 인식 알고리즘이다[8]. YOLO는 1-stage detector로서 간단한 처리 과정으로 다른 딥러닝 알고리즘에 비해 속도가 매우 빠르며, 높은 mAP(mean Average Precision)을 보인다. 하지만 작은 object에 대해서 정확도가 낮았으나 YOLOv3에서는 각 칸에 Bounding box를 생성하는 방법에 대한 집중적인 개선이 이루어졌으며, 더 안정적인 학습과 객체 식별이 가능하게 되었다.



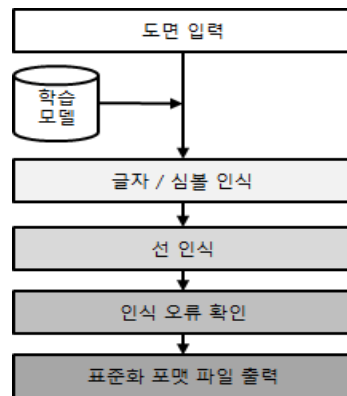
(그림 2) YOLO 물체 인식 과정

3. 인공지능 기반 객체인식 기법

본 장에서는 인공지능을 기반으로 한 정확한 객체인식을 위해 본 논문에서 제안한 기법에 대하여 설명하였다. 제안한 기법은 객체 인식의 성능을 향상시키기 위하여 객체들의 특성을 고려한 알고리즘을 적용하였다.

3.1 시스템 개요

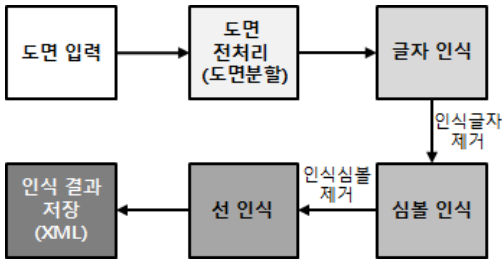
본 논문에서는 산업 플랜트에서 활용되는 아날로그 형태의 다양한 도면들의 활용도를 높이기 위하여 도면내의 객체들(선, 심볼, 글자)을 인공지능을 기반으로 인식한 후 인식된 결과를 표준화된 데이터 포맷으로 생성될 수 있도록 하는 것을 목표로 한다. 도면내의 객체들을 인식하기 위해서는 다음과 같은 특징들을 고려해야만 한다. 먼저, 첫 번째로 도면의 크기이다. 산업 플랜트에서 사용되는 도면들은 다양한 크기를 갖고 있는데, 이러한 특징은 인식을 위한 객체들의 상대 위치 계산에 어려움을 줄 수 있다. 두 번째는 도면내의 객체들의 상대적인 크기이다. 도면의 크기에 따른 객체들의 상대적 크기는 인공지능을 이용한 인식에 큰 영향을 미칠 수 있다. 따라서 본 논문에서는 이러한 특성들을 고려하여 다음과 같은 절차를 통해 객체 인식을 수행한다. 1 단계로 심볼과 글자들에 대하여 최적의 학습 모델을 구축한다. 학습 모델 구축시 심볼의 복잡도 및 크기 등을 고려하여 가중치를 다른 학습 모델을 구축한다. 2 단계로 입력 이미지를 일정 크기로 분할하여 글자 인식을 진행한다. 3 단계로 인식된 글자를 제거한 후 심볼 인식을 실시한다. 4 단계로 선 인식의 성능을 향상시키기 위하여 글자와 심볼 인식 결과를 도면에서 제거한 후 선 인식을 수행한다. 5 단계에서 최종 인식 결과에 대한 오류를 점검한 후 표준화 포맷 형태의 파일로 결과를 출력한다. <그림 3>은 본 논문에서 제안한 기법의 인식 과정을 보여주고 있다.



(그림 3) 객체인식 과정

3.2 객체인식

도면내 객체 인식의 과정은 도면 전처리, 글자인식, 심볼인식, 선인식 과정으로 이루어졌다. 입력 도면에 대한 전처리는 도면내 객체 인식의 성능을 향상시키기 위하여 수행하였으며, 입력 도면을 6개의 서브 도면으로 분할을 하였다. 이는 전체 도면 대비 객체들의 크기가 상대적으로 매우 작기 때문에 객체들의 특징을 추출하기가 매우 어려운 점이 있다. 따라서 이러한 문제를 해결하기 위한 전처리 과정이 수행되었다. <그림 4>는 객체 인식을 위한 세부 과정에 대하여 보여주고 있다.



(그림 4) 객체인식을 위한 세부 절차

도면내 글자와 심볼 인식을 인식하기 위하여 각각의 글자와 심볼들에 대한 학습모델을 구축한 후 이를 기반으로 인식을 수행하였다. 인식의 순서는 글자 인식 후 심볼 인식을 진행하였다. 이러한 이유는 일부 심볼들이 글자를 포함하고 있고 같은 심볼이지만 내부에 포함되어 있는 글자들이 포함되어 있어 학습 모델 구축시 이러한 모든 내용을 반영하기가 쉽지 않기 때문이다. 즉, 학습 모델 구축시에도 글자를 제외한 심볼들을 이용한 모델을 구축하였고 글자를 인식한 후에 도면에서 글자를 제거하면 심볼의 인식률을 향상시킬 수 있게 하였다. 따라서 글자를 인식한 후 인식 결과에 대한 정보를 저장하는 중간 파일을 생성하였다. 이 파일에는 인식된 글자와 각 글자들의 위치 정보를 포함하게 하였다. 이렇게 인식된 글자 정보를 중간 파일에 저장한 후 도면에서 글자를 삭제하게 된다. 그런 후에 같은 방법으로 심볼 인식을 수행하였다. 심볼 인식도 마찬가지로 인식 후 중간 파일에 심볼들에

정보와 위치 정보를 중간 파일에 저장한 후 심볼들을 삭제한다. 선 인식 전에 심볼 인식 결과를 삭제하는 이유는 심볼들에 포함되어 있는 수직, 수평선이 선 인식 단계에서 인식되는 것은 불필요하기 때문이다. 선 인식의 단계는 크게 수평선 인식, 수직선 인식, 교차점 인식 과정을 수행하게 된다. 먼저 도면의 가장 왼쪽 상단을 기준으로 하여 오른쪽으로 한 픽셀씩을 읽어 들이면서 픽셀의 농도값을 확인한다. 만약 검은색의 픽셀이 존재한다면 계속해서 읽어 들인다. 이렇게 읽어 들이면서 픽셀의 값이 변화가 생긴다면 수평선의 끝으로 인식하게 된다. 이렇게 도면의 마지막 픽셀까지 읽어 들인다. 같은 방법으로 도면의 왼쪽 상단의 시작점을 기준으로 아래 방향을 픽셀값을 읽어 들이면서 수직선을 인식해나간다. 마지막으로 수집된 수평선과 수직선의 교차하는 위치를 수집한 후 선 정보에서 시작, 끝점의 좌표, 길이, 교차점의 위치와 교차되는 수직선 또는 수평선의 인덱스를 저장한다. 인식된 심볼 영역내에서 인식된 선들은 삭제하게 된다.

선 인식이 완료되면 해당 정보를 활용하기 위해 표준화 파일 포맷인 XML 파일로 그 결과를 저장하게 된다. <그림 5>인 인식 결과를 저장한 XML 파일을 보여주고 있다.

```

<DwgNodeDTO>
  <DwgNodeType>LINK_ON</DwgNodeType>
  <IO_No>26</IO_No>
  <DwgNodeUID>558771a9-bf8a-45dd-a1cf-7deb73feee25</DwgNodeUID>
  <IO_From></IO_From>
  <DwgNode_FromUID>00000000-0000-0000-0000-000000000000</DwgNode_FromUID>
  <IO_To></IO_To>
  <DwgNode_ToUID>00000000-0000-0000-0000-000000000000</DwgNode_ToUID>
  <CenterPt>
    <Point>
      <PointX>1266.000</PointX>
      <PointY>1814.000</PointY>
    </Point>
  </CenterPt>
  <Bound>
    <MinPoint>
      <Point>
        <PointX>1200.000</PointX>
        <PointY>1780.000</PointY>
      </Point>
    </MinPoint>
    <MaxPoint>
      <Point>
        <PointX>1332.000</PointX>
        <PointY>1848.000</PointY>
      </Point>
    </MaxPoint>
  </Bound>
</DwgNodeDTO>
  
```

(그림 5) 인식 결과를 저장한 XML

4. 실험 및 결과

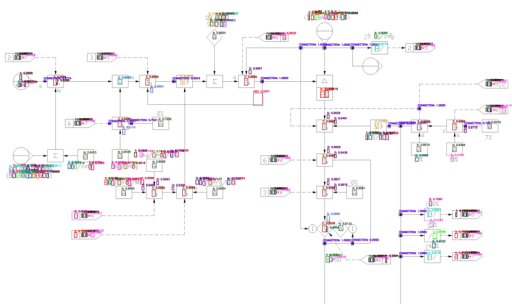
4.1 실험 환경

본 논문에서는 제안한 기법의 성능 평가를 위하여 다음과 같은 환경을 이용하였다. 먼저 글자와 심볼들에 대해서 각각 50개씩 라벨링을 수행하였다. 그리고 학습모델을 구축하기 위하여 CPU i7, 메모리 64G, GPU Geforce RTX 3070을 이용하였다. 학습은 20만번을 수행하였으며, 각 객체별로 서로 다른 가중치 파일을 수행하여 인식률을 향상시켰다.

4.2 실험 결과 분석

도면내 객체 인식은 크게 선, 글자, 심볼 인식 세 가지로 구분된다.

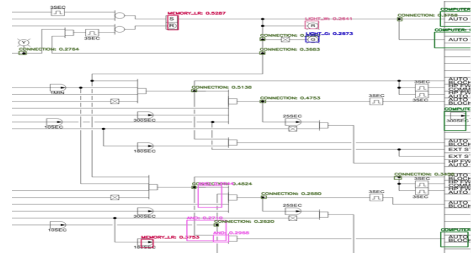
글자 인식은 각 영문자와 특수문자 한 글자씩 라벨을 수행한 후 학습 모델을 구축하였다. 그리고 모든 글자에 대한 가중치는 동일하지 않으며, 인식 실험을 반복하면서 글자별 가중치를 설정하였다. 글자를 하나의 객체로 인식하는 YOLO를 적용하여 글자 인식을 수행하였으며, 그 결과는 <그림 6>에서 보여주고 있다.



(그림 6) 글자 인식 결과

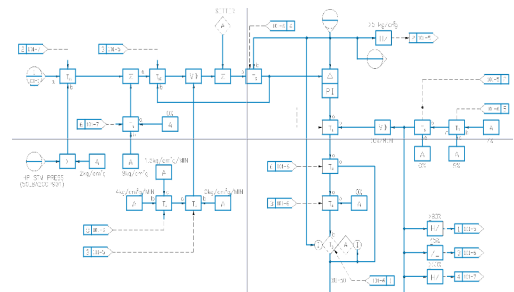
심볼 인식은 속도와 정확도 측면에서 R-CNN과 Faster R-CNN 기법에 비해 YOLO를 적용하는 것이 우수한 결과를 얻을 수 있었다. 인식의 성능을 향상시키기 위하여 도면 전체 크기를 그대로 학습하지 않고 도면을 분할하여 라벨링 및 학습 모델을

구축하였다. 앞 단계의 글자 인식 후 글자가 삭제된 도면에서 심볼 인식을 진행하여 보다 좋은 인식 결과를 얻을 수 있었다. 심볼에 대한 인식 결과는 <그림 7>에서 보여주고 있다.



(그림 7) 심볼 인식 결과

선 인식은 기존의 선 인식을 위해 많이 활용되는 Hough line 알고리즘과 비교하여 수행해보았지만, 도면내 선의 간격이 좁고 정확히 연결되어 있지 않는 부분이 많아 오인식률이 높게 나타났다. 따라서 직접 개발한 선 인식 알고리즘을 이용하여 인식을 수행하였다. 선 인식은 수평선, 수직선, 교차점 인식과정으로 수행하게 된다.



(그림 8) 선 인식 결과

<표 1>에서는 본 논문에서 실험한 인식 실험의 평균 결과를 보여주고 있다. 실험 결과를 통해서 본 논문에서 제안한 기법이 객체 인식에 우수한 성능을 보여주는 것을 확인할 수 있다.

<표 1> 평균 인식 결과

항목	TPR	FPR	Precision
글자	0.923	0.091	0.917
심볼	0.946	0.074	0.922
선	0.961	0.033	0.951

5. 결 론

최근들어 다양한 분야에서 인공지능 혹은 머신러닝을 적용하는 기술 개발이 활발하게 이루어지고 있다. 이러한 기술들은 이미지, 동영상 또는 실시간 영상 등 그 대상 또한 매우 다양하다.

본 논문에서는 도면내 객체를 인식하여 그 결과를 여러 응용 분야에 활용할 수 있도록 하기 위한 기법을 제안하였다. 특히 도면내 객체를 인식하기 위한 효율적 기법을 제시하였으며, 인식 결과의 활용을 위해 표준화 포맷 파일에 저장하는 방법 또한 제시하였다. 먼저 입력 이미지의 특성을 고려하여 도면을 분할하고 반복 인식을 수행하는 전처리 기법을 제시하였다. 또한 심볼 인식의 성능을 높이기 위하여 전단계의 글자 인식 후 결과를 저장하는 중간 파일을 생성한 후 인식 결과를 입력 이미지에서 삭제하는 방법을 적용하였다. 심볼 인식도 같은 방법을 적용하였다. 그리고 인식 성능을 높이기 위해 객체별 다른 가중치를 적용하도록 하였다. 본 논문에서 제안한 기법의 우수한 성능은 실험 결과를 통해 확인할 수 있었다.

참고문헌

- [1] T.-Yi Lin, P. Dollár, R. B. Girshick et al., "Feature pyramid networks for object detection," IEEE CVPR, vol. 43, pp. 936-944, 2017.
- [2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems, vol. 61, pp. 91-99, 2015.
- [3] Chen, L., Wang, F., Wang, L.: Research on warehouse object detection algorithm based on fused DenseNet and SSD. In: Chinese Conference on Image and Graphics Technologies, pp. 602-611. Springer, Berlin (2019)
- [4] W.-L. Chu, C.-J. Lin, and K.-N. Chang, "Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine," Appl. Sci., vol. 9, no. 21, p. 4579, 2019.
- [5] Asaoka, T., Nagata, K., Nishi, T.: Detection of object arrangement patterns using images for robot picking. ROBOMECH J. 5(1), 23-32 (2018)
- [6] BHARATI, Puja; PRAMANIK, Ankita. Deep learning techniques—R-CNN to mask R-CNN: a survey. Computational Intelligence in Pattern Recognition, 2020, 657-668.
- [7] Sharif Razavian, Ali, et al. "CNN features off-the-shelf: an astounding baseline for recognition." Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2014.
- [8] Diwan, T., G. Anirudh, and J. V. Tembhurne. "Object detection using YOLO: challenges, architectural successors, datasets and applications." Multimedia Tools and Applications (2022): 1-33.

[저자소개]



양 환 석 (Hwan-seok Yang)
1998년 2월 조선대학교 이학석사
2005년 2월 조선대학교 이학박사
2007년 3월 호원대학교 연구교수
2011년 9월 ~ 현재 중부대학교
정보보호학과 부교수
email : yanghs@joongbu.ac.kr