https://doi.org/10.7840/kics.2023.48.9.1169

웹 검색엔진 및 딥러닝 기반 한글 단어 인식 OCR 시스템

장 혁 수◆, 고 상 호*, 이 재 현*, 박 승 권°

The Deep Learning-Based OCR System for Korean Word with Web Search Engine

Hyuksoo Jang◆, Sangho Goh*, Jaehyun Lee*, Sungkwon Park°

요 약

Optical character recognition (OCR)은 이미지 내의 텍스트를 인식하여 이를 텍스트 데이터로 변환하는 기술이다. 외국에서는 OCR로 문서 처리를 자동화하여 비용과 시간을 절약하는 데 활용되고 있다. 그러나 한국에서는 한글의 언어적 특성 때문에 영어와 숫자에 비해 인식률이 낮아, OCR이 적극적으로 사용되지 않고 있다. 따라서 OCR의 한글 인식 정확도가 향상되면 한국에서도 OCR을 통한 업무 효율성 증가를 기대할 수 있다. 본 논문에서는 convolutional neural network (CNN)을 이용해 한글, 영어 및 숫자를 훈련시켰다. 이를 기반으로 문자가 복합적으로 구성된 단어에서 한글의 완성형 글자를 구분해 인식하고, 인식된 단어를 검색엔진에 검색 후 수정된 검색어가 존재하면 이를 최종 결과물로 출력해 인식 정확도를 향상시키는 시스템을 구현하였다. 인식률 측정 결과 한글, 영어 및 숫자가 복합적으로 구성된 영수증에서 최대 90.1%의 문자 인식률이 확인되었다.

키워드 : 광학 문자 인식, 딥러닝, 합성곱 신경망, 한글 단어 인식, 단어 분리
Key Words : OCR, Deep Learning, CNN, Korean Word Recognition, Word Segmentation

ABSTRACT

Optical character recognition (OCR) is the technology that recognizes text in an image and converts it into text data. In foreign countries, OCR enables automated document processing. Since the recognition rate of Hangul is lower than that of English and Numbers, the OCR is not widely used in Korea. If the OCR accuracy of Hangul is improved, we expect an increase in work efficiency through OCR in Korea as well. In this paper, the OCR system was based on the convolutional neural network (CNN) to train Hangul, English, and Numbers. Subsequently, the process was implemented that distinguishes the complex words to complete Hangul characters, recognizes the complete Hangul characters, and converts them into text data. Additionally, to further improve the accuracy of the OCR system, search the text data in a web search engine, and verify the existence of modified words. If a modified word is found in the web search results, it is considered the correct recognition result and included in the final text data. We conducted a recognition rate measurement and found that the OCR system was able to accurately recognize up to 90.1% of characters in documents containing Hangul, English, and Numbers.

※ 본 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2022R1F1A1064288).
◆ First Author : Hanyang University, Department of Electronic Convergence Engineering, wkd9146@hanyang.ac.kr, 학생회원
° Corresponding Author : Hanyang University, Department of Electronic Convergence Engineering, sp2996@hanyang.ac.kr, 종신회원
* Hanyang University, Department of Electronic Convergence Engineering, {cndgkrp43, lenny1205}@hanyang.ac.kr
논문번호 : 202303-040-C-RN, Received February 28, 2023; Revised April 24, 2023; Accepted May 2, 2023

Ⅰ. 서 론

광학 문자 인식 (Optical character recognition; OCR)은 이미지 내의 텍스트를 기계가 읽을 수 있는 텍스트 포맷으로 변환하는 기술이다[1]. 본래 컴퓨터에서 문서를 이미지로 저장하게 되면 이미지 내의 텍스트 데이터를 수정, 검색할 수 없다. 하지만 OCR을 사용해 이미지 내의 텍스트를 데이터로 변환하면 데이터의 수정, 검색 등이 가능해진다. 이러한 특징은 산업에서 문서의 데이터를 처리하는 방식에 변화를 일으켰다. 직원들이 수작업으로 문서의 데이터를 입력해 처리하던 업무를 OCR을 사용해 문서의 이미지로부터 텍스트 데이터를 추출함으로써 시간적, 비용적 이점을 가지게 되었다. 외국에서는 금융, 의료, 물류 등 다양한 산업에서 적극적으로 OCR을 활용해 업무 효율성을 높이고 있다. 그러나 한국에서는 언어의 구조적 특징으로 인해 널리 사용되지 않고 있다.

영어와 숫자는 한 음절이 한 글자를 구성해 하나의 단어에서 하나의 글자를 구분하는 것이 쉬우므로 convolutional neural network (CNN)을 통한 인식률이 95% 이상 높게 도출된다[2]. 하지만 한글의 경우 자음과 모음 세부적으로는 초성, 중성, 종성의 조합을 통해 글자가 구성되기 때문에 단어나 문장에서 한 글자를 구분하는 데 어려움이 있다. 이로 인해 한글의 OCR 정확도는 영어와 숫자에 비해 낮은 편이다. 한글의 구조적 특징 때문에 OCR을 통한 한글 인식에는 일반적으로 완성형 글자가 사용된다. 완성형 글자를 이용해 글자를 인식하면 한 글자의 인식 정확도는 영어, 숫자와 같이 높은 정확도를 보여준다. 하지만 한글, 영어 및 숫자가 복합적으로 구성된 단어에서 한글의 모음이 영어나 숫자로 인식될 가능성이 있어 완성형 글자 하나를 구분해 인식하는 데 어려움이 있다.

본 논문에서는 한글, 영어 및 숫자가 복합적으로 포함된 단어에서 각 글자를 구분하고, CNN을 통해 글자를 인식한다. 그리고 인식된 결과를 검색엔진을 통해 개선하는 글자 인식 시스템을 제안한다. 논문의 구성은 다음과 같다. 2장은 글자 인식 시스템을 설명하기 위해 학습 데이터, CNN 구조 그리고 문자 인식 과정을 소개하고, 3장은 영수증을 통해 개발한 시스템의 인식 정확도를 분석해 성능을 평가한다. 4장은 본 논문의 결론 및 시스템 성능 개선을 위한 방법을 제안한다.

Ⅱ. 글자 인식 시스템 구성

글자 인식 시스템은 학습 데이터, CNN 구조, 글자 인식 과정으로 구성된다. 수집한 학습 데이터를 기반으로 제안한 CNN 구조를 통해 문자별 특징을 학습시킨다. 학습 결과를 기반으로 단어에서 글자를 올바르게 인식하고, 인식 정확도를 높이기 위해 검색엔진을 활용한다. 이를 구현하기 위해 MATLAB R2022b 9.13.0.2080170을 사용하였다.

2.1 학습 데이터

일반적으로 문서는 한글, 영어 및 숫자로 구성된다. 이를 인식하기 위해 다양한 한글, 영어 및 숫자의 학습 데이터를 수집하였고, 그중 문자 별 사용 빈도가 높은 폰트 일부를 선정해 사용하였다.

2.1.1 한글

한글은 초성 19개, 중성 21개, 종성 28개가 사용되고, 11,172개의 조합이 가능하다. 그중 KS X 1001[3]에 수록된 2,350자의 완성형 한글을 선정하였다. 이를 이미지로 만든 완성형 글자 데이터베이스인 PHD-08[4]을 사용하였고, 총 1,713,150개의 한글 데이터가 사용되었다. 그림 1은 한글 학습 데이터의 예시이다.

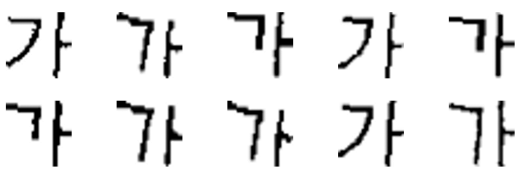


그림 1. 한글 학습 데이터 예시
Fig. 1. Example of training data for Hangul

2.1.2 영어 및 숫자

영어와 숫자는 한 음운이 한 글자로 구성되어 단순하고 글자 수가 적다[5]. 영어는 A부터 Z까지 대소문자를 구분하지 않아 26가지가 사용되었다. 숫자는 0부터 9까

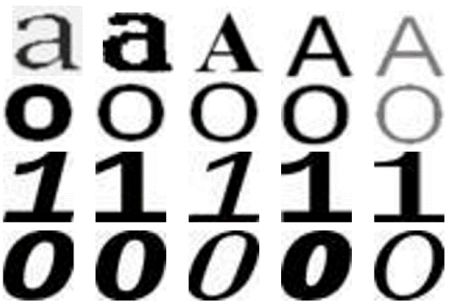


그림 2. 영어와 숫자 학습 데이터 예시
Fig. 2. Example of training data for English and Numbers

지 10가지가 사용되었다. 영어 및 숫자 데이터는 MIT License의 데이터를 사용하였고, 총 1,560개의 영어 데이터와 960개의 숫자 데이터가 사용되었다. 그림 2는 영어와 숫자 학습 데이터의 예시이다.

2.2 CNN 구조

본 논문에서는 한글, 영어 및 숫자를 인식하기 위해 VGG-16[6]을 참조한 CNN 구조를 사용하였다. 이때, learning rate는 0.1, momentum은 0.9, mini-batch size는 256 그리고 epoch는 5로 설정하였다. 그림 3은 본 논문에서 제안하는 CNN 구조이다.

표 1을 통해 VGG-16과 본 논문에서 제안하는 CNN 구조를 비교하였다. VGG-16은 이미지 인식에 적합한 구조로 입력 크기가 크고 글자를 인식하기에는 복잡한 구조를 가지고 있다. 글자 인식에 적합한 구조를 선정하기 위해 다양한 방식으로 시도한 결과 중 인식 정확도가 가장 높은 구조를 채택하였다.

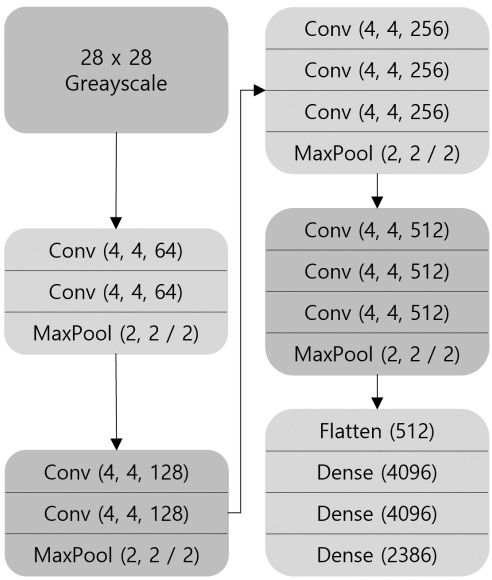


그림 3. 제안된 CNN 구조
Fig. 3. Proposed CNN architecture

표 1. VGG-16과 제안된 CNN구조의 비교
Table 1. Comparison between VGG-16 and the proposed CNN architecture

	VGG-16	Proposed CNN
Input layer	Size: 224x224	Size: 28x28
Convolution layer	Size: 3x3 Layers: 13	Size: 4x4 Layers: 10
Pooling layer	Size: 2x2 / 2 Layers: 5	Size: 2x2 / 2 Layers: 4
Dense layer	Size: 4096 Layers: 3	Size: 4096 Layers: 3

2.3 글자 인식 과정

본 장에서는 한글, 영어 및 숫자가 복합적으로 구성된 단어 중 한 글자를 인식하고, 텍스트로 출력하는 과정에 대해 설명한다. 글자 인식 과정은 Segmentation, Recognition, Post-Processing의 3단계로 구성되어 있다.

2.3.1 Segmentation

Segmentation 과정은 한 단어를 글자 단위로 구분하는 과정이다. 본 논문에서는 단어를 글자 단위로 분리하기 위해 일반적으로 글자 사이에 존재하는 빈 열 공간(Blank row)을 이용하였다.

그림 4의 '애플망고치즈설빙'을 인식하기 위해 original data에서 한 글자 단위와 두 글자 단위로 segmentation을 진행하여 결과를 single segmentation과 double segmentation으로 각각 저장한다. '애플'을 예시로 들면 single segmentation에는 'ㅇ', 'ㅐ', 'ㅍ'이 저장되고, double segmentation에는 '애', 'ㅐㅍ'이 저장된다.

그림 4. Segmentation 과정 예시
Fig. 4. Example of Segmentation process

2.3.2 Recognition

앞서 기술하였듯 한글은 영어 및 숫자와 달리 한 음운이 여러 개의 글자로 구성된다. 따라서 별도의 처리 과정을 거치지 않으면 그림 4의 '애'의 경우, 'ㅇ'과 'ㅐ'로 구분되고, '치'의 경우 'ㅊ'와 'ㅣ'로 구분된다. 이렇게 글자를 구분하게 되면 한글의 완성형 글자로 인식하지 않고 'ㅇ', 'ㅐ'그리고 'ㅣ'가 영어나 숫자로 인식될

가능성이 있다. 위 문제를 해결하기 위해 Segmentation 과정에서 저장한 single segmentation과 double segmentation 결과를 비교한다. 그 과정은 다음과 같다.

1. Single segmentation 결과와 double segmentation 결과를 인식하여, 해당 결과에 대한 confidence와 텍스트를 도출한다.
2. Single segmentation과 double segmentation의 confidence 값을 Algorithm 1을 통해 임계값(Threshold)과 비교하여, single segmentation 결과나 double segmentation 결과를 출력한다. 본 논문에서는 임계값을 0.7로 설정하였고, 이를 통해 각 segmentation에 대한 민감도를 조절할 수 있다.
3. 2의 결과를 통해 한 글자에 대한 출력 결과를 도출한다.

Algorithm 1. Recognition process code

```
Input the confidence of single segmentation 'S', double segmentation 'D' and threshold 'T'
if S > T and D < T
  result = the output of single segmentation;
else if S > T and D > T
  result = the output of double segmentation;
else if S < T and D > T
  result = the output of double segmentation;
else result = the output of single segmentation;
```

2.3.3 Post-Processing

Post-Processing 과정은 Recognition 과정에서 도출된 출력값의 정확도를 향상시키기 위한 과정이다. 이를 위해 검색엔진을 사용했으며, 그 과정은 다음과 같다.

1. 각 검색엔진 URL에 Recognition 과정의 결과를 결합해 해당 단어를 검색하기 위한 URL을 구성한다.
2. 각 검색엔진에 URL로 검색을 요청했을 때 수신되는 HTML 파일을 수신한다.
3. 수신된 HTML 파일에서 수정된 검색어를 나타내는 HTML tag와 class 이름을 이용해 검색엔진별 수정된 검색어가 존재하는지 확인한다.

4-1. 수정된 검색어가 존재하지 않는다면 기존 Recognition 과정의 결과를 최종 출력값으로 사용한다.

4-2. 수정된 검색어가 존재한다면 수정된 검색어를 최종 출력값으로 사용한다.

그림 5는 Recognition 과정의 결과가 정상적으로 인식된 상황의 예시이다. 그림 5의 단어를 인식하는 절차는 다음과 같다.

1. '팥인절미설빙' 글자를 정상적으로 인식
2. '팥인절미설빙' 검색을 위한 URL 구성
3. 검색엔진에 수정된 검색어 존재하지 않음

4-1. 기존 검색 결과인 '팥인절미설빙' 출력

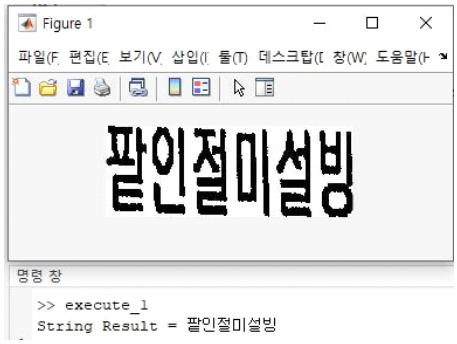


그림 5. 단어가 올바르게 인식된 상황
Fig. 5. The situation that the word is recognized correctly

그림 6은 Recognition 과정의 결과가 정상적으로 인식되지 못한 상황의 예시이다. 그림 6의 단어를 인식하는 절차는 다음과 같다.

1. '인절미떡' 글자를 정상적으로 인식하지 못해 '민절미떡'으로 인식
2. '민절미떡' 검색을 위한 URL 구성
3. 검색엔진에 수정된 검색어(인절미떡) 존재

4.2 출력값으로 수정된 '인절미떡' 출력

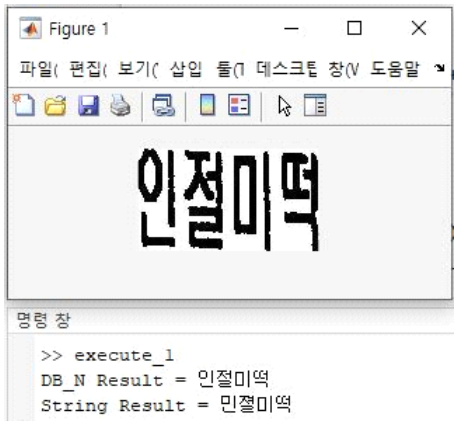


그림 6. 단어가 올바르게 인식되지 않은 상황
Fig. 6. The situation that the word is not recognized correctly

Ⅲ. 성능평가

본 장에서는 영수증을 이용해 글자 인식 시스템의 성능 평가를 진행하였다. 영수증은 6개가 사용되었고, 영수증에서 시스템의 입력 데이터로 필요한 단어를 추출하기 위해 CRAFT[7]를 이용하였다. 그림 7은 성능 평가를 위해 사용된 영수증 일부이다.

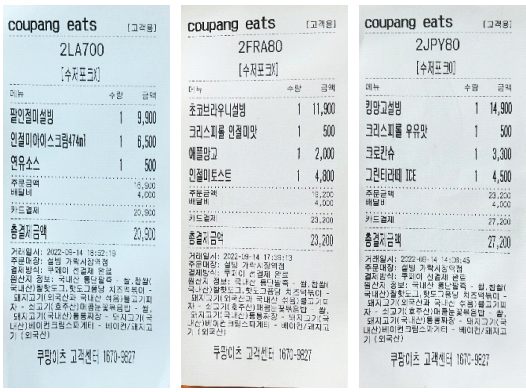


그림 7. 성능 평가를 위해 사용된 영수증 일부
Fig. 7. Some receipts used for performance evaluation

그림 8은 글자 인식 시스템의 인식 정확도를 나타낸다. Recognition 과정에서는 평균 85.2%, 최대 88%의 인식 정확도를 보여주고, Post-Processing 과정에서는 평균 87.8%, 최대 90.1%의 인식 정확도를 보여준다. 이를 통해 한글, 영어 및 숫자가 복합적으로 구성된 단어에서 글자별 특성에 제한받지 않고 인식하는 것을 확인하였고, Post-Processing 과정이 인식 정확도를 향상시키는 것이 확인되었다.

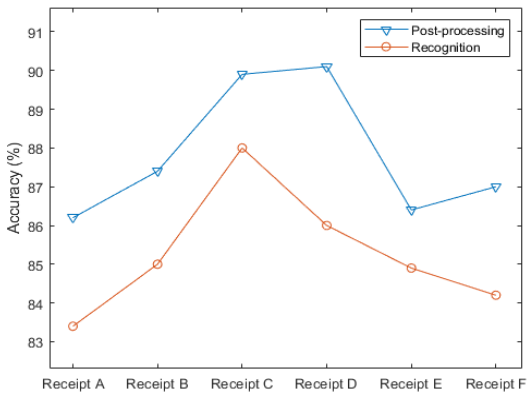


그림 8. 인식 정확도 측정 결과
Fig. 8. Measurement results for recognition accuracy

그림 9는 영수증별 인식 결과를 나타낸다. 그래프의 순서대로 영수증별 단어(Total words), 인식 오류가 발생한 단어(Recognition errors) 그리고 검색엔진을 통해 수정된 단어(Modified words)를 의미한다. Modified words는 recognition errors를 기반으로 단어를 검색하고 수정하기 때문에 recognition errors는 실제 단어와 유사하게 인식되어야 한다. 이는 학습 데이터의 추가 확보 및 CNN 구조 개선을 통해 Recognition 과정의 인식 정확도를 높인다면 Post-Processing 과정의 인식 정확도가 높아져 시스템의 전체적인 인식 정확도가 향상될 수 있음을 의미한다.

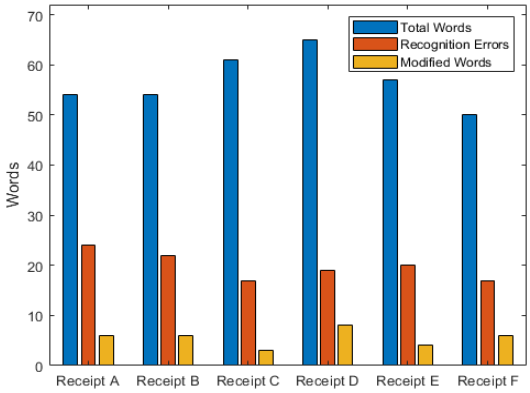


그림 9. 영수증별 인식 결과
Fig. 9. Recognition results for each receipt

Ⅳ. 결 론

본 논문에서는 CNN을 통해 한글, 영어 및 숫자를 복합적으로 학습시켰다. 그리고 학습 결과를 기반으로 한 단어에서 각 글자를 구분해 인식하고, 인식된 단어를 검색엔진을 이용해 인식 정확도를 높이는 시스템을 구현했다.

구현한 시스템의 성능 평가를 위해 영수증을 사용하여 글자 인식 성능을 평가했다. 성능 평가 과정을 통해 본 논문에서 구현한 시스템이 한글, 영어 및 숫자가 복합적으로 구성된 단어에서 각 글자를 구분해 인식하는 것을 확인하였다. 이후 검색엔진을 통해 인식된 결과를 개선하여 인식 정확도가 향상된 것을 확인하였다. 추후 훈련 데이터를 보완하여 진행한다면 인식 정확도가 더욱 높아질 수 있으리라 기대한다.

References

[1] AWS, *What is OCR (optical character*

recognition)?(2022), Retrieved Feb. 24, 2023, from https://aws.amazon.com/what-is/ocr/?nc1 = h_ls.

[2] A. Baldominos, Y. Saez, and P. Isasi, "A survey of handwritten character recognition with MNIST and EMNIST," *MDPI Applied Sci.*, vol. 9, no. 15, 3169, Aug. 2004. (https://doi.org/10.3390/app9153169)

[3] Korean Standards Association, "*Code for information interchange (hangeul and hanja)*," Sep. 1974.

[4] D. S. Ham, D. R. Lee, I. S. Jung, and I. S. Oh, "Construction of printed hangul character database PHD08," *The J. Korea Contents Assoc.*, vol. 8, no. 11, pp. 33-40, Nov. 2008. (https://doi.org/10.5392/JKCA.2008.8.11.033)

[5] G. H. Kang, J. H. Ko, Y. J. Kwon, N. Y. Kwon, and S. J. Koh, "A study on improvement of Korean OCR accuracy using deep learning," in *Proc. KIICE Info. and Commun. Conf. 2018*, pp. 693-695, Jeonju, Korea, May 2018.

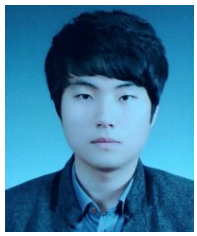
[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv cs.CV, arXiv: 1409.1556*, Sep. 2014. (https://doi.org/10.48550/arXiv.1409.1556)

[7] Y. M. Baek, B. D. Lee, D. Y. Han, S. D. Yun, and H. S. Lee, "Character region awareness for text detection," *arXiv cs.CV, arXiv: 1904.01941*, Apr. 2019. (https://doi.org/10.48550/arXiv.1904.01941)

장 혁 수 (Hyuksoo Jang)

2022년 2월 : 강남대학교 소프트웨어응용학부 졸업
2022년 3월~현재 : 한양대학교 융합전자공학부 석사과정
<관심분야> 차량 내부 네트워크, Time-Sensitive Network
[ORCID:0000-0002-9784-7931]

고 상 호 (Sangho Goh)



2022년 2월 : 한양대학교 융합전자공학부 졸업
2022년 3월~현재 : 한양대학교 융합전자공학부 석사과정
<관심분야> 차량 내부 네트워크, 로봇 내부 네트워크, Time-Sensitive Network, AI

이 재 현 (Jaehyun Lee)

2021년 8월 : 한서대학교 항공전자공학과 졸업
2021년 9월~현재 : 한양대학교 융합전자공학부 석사과정
<관심분야> 로봇 내부 네트워크, 유무선 통신

박 승 권 (Sungkwon Park)

1982년 2월 : 한양대학교 공학사
1983년 8월 : Stevens Institute of Technology, Hoboken, NJ 공학석사
1987년 12월 : Rensselaer Polytechnic Institute, Troy, NY 공학박사
1987년 8월~1992년 8월 : Tennessee Technological University 조교수
1992년 9월~1993년 1월 : Tennessee Technological University 부교수
1993년 3월~현재 : 한양대학교 융합전자공학부 정교수
2000년 12월~2001년 6월 : 방송통신위원회 방송 정책기획 위원회 위원
2001년 4월~2005년 12월 : 정보통신부 디지털 케이블 TV 추진위원회 위원장
2008년 2월~2011년 7월 : 한양대학교 정보통신처 처장
2008년 3월~2010년 3월 : PG-803, Telecommunication Technology Association 의장
2008년 4월~2020년 2월 : 한국 미래 케이블 포럼 의장
2013년 10월~2014년 9월 : 방송통신위원회 자문위원
2017년 6월~2021년 12월 : Rapporteur, ITU-T SG.9 Q.8
2016년 1월~현재 : ISO TC22 SC31 Korean Delegate
2019년 7월~2020년 6월 : 한양대학교 공대 학장
<관심분야> 로봇 내부 네트워크, 차량 내부 네트워크, 통신이론, AI
[ORCID:0000-0001-6144-7488]