

딥러닝 기반 OCR 인식 엔진의 정확도 향상을 위한 전/후처리 기술 구현

장창복¹, 김기봉^{2*}

¹(주)알투스소프트 연구소장, ²대전보건대학교 컴퓨터정보과 교수

Implementation of Pre-Post Process for Accuracy Improvement of OCR Recognition Engine Based on Deep-Learning Technology

Chang-Bok Jang¹, Ki-Bong Kim^{2*}

¹Head of Rearsach Institute, R2SOFT Co., Ltd.

²Professor, Dept. of Computer Information, Daejeon Health Institute of Technology

요약 4차산업 혁명이 도래함에 따라 AI 기술을 적용하는 솔루션 개발이 활발하게 이루어지고 있다. 2017년도부터 금융권, 보험사를 중심으로 AI 기반 RPA(Robotic Process Automation)을 이용한 업무 자동화 솔루션 도입이 이루어지기 시작했으며, 최근에는 RPA 솔루션 도입 단계를 지나 확산하는 시기로 진입하고 있다. 이러한 RPA 솔루션을 이용한 업무 자동화 중에서 각 종 문서들을 이용한 업무 자동화에는 문서내의 문자 정보를 얼마나 정확하게 인식하는지가 매우 중요하다. 이러한 문자 인식은 최근 딥러닝 기술을 도입함으로써 그 정확도가 많이 높아졌지만, 여전히 완벽한 인식 정확도 갖는 인식 모델은 존재하지 않는다. 따라서, 본 논문에서는 딥러닝 기반 문자 인식 엔진에 전/후 처리 기술을 적용할 경우 얼마나 정확도가 향상되는지를 확인하고 RPA 인식 엔진과 연계 기술을 구현하였다.

주제어 : 인공지능, RPA, 딥러닝, 융합, 4차 산업, OCR

Abstract With the advent of the 4th Industrial Revolution, solutions that apply AI technology are being actively developed. Since 2017, the introduction of business automation solutions using AI-based Robotic Process Automation (RPA) has begun in the financial sector and insurance companies, and recently, it is entering a time when it spreads past the stage of introducing RPA solutions. Among the business automation using these RPA solutions, it is very important how accurately textual information in the document is recognized for business automation using various documents. Such character recognition has recently increased its accuracy by introducing deep learning technology, but there is still no recognition model with perfect recognition accuracy. Therefore, in this paper, we checked how much accuracy is improved when pre- and post-processor technologies are applied to deep learning-based character recognition engines, and implemented RPA recognition engines and linkage technologies.

Key Words : Artificial Intelligence, Robotic Process Automation, Deep Learning, Convergence, the fourth industrial revolution, Optical Character Recognition,

*This research was supported through the Korea Industrial Technology Association(KOITA) funded by the Ministry of Science and ICT(MSIT)

*Corresponding Author : Ki-Bong Kim(kbkim@hit.ac.kr)

Received August 30, 2021

Revised December 24, 2021

Accepted January 20, 2022

Published January 28, 2022

1. 서론

최근 4차산업 혁명이 도래함에 따라 AI 기술을 적용하는 솔루션 개발이 활발하게 이루어지고 있다. 특히, 2017년도부터 금융권, 보험사를 중심으로 AI 기반 RPA를 이용한 업무 자동화 솔루션 도입이 이루어지기 시작했으며, 최근에는 RPA 솔루션 도입 단계를 지나 확산하는 시기로 진입하고 있다. RPA(Robotic Process Automation) 기술은 일반적으로 업무 효율성을 높이기 위해 업무프로세스를 자동화하는 기술을 의미한다. 이러한 기술은 2000년대 중반에 처음으로 도입되기 시작했으나, 외부 API 연계 문제, 적용할 수 있는 업무 프로세스 부족 등으로 실제 업무 자동화에 적용하기에는 다소 미흡한 점이 있었다. 하지만 최근 AI 기술의 발전으로 인해 2018년도부터 딥러닝 기술 기반 인식 엔진을 탑재한 RPA 솔루션이 개발됨에 따라 다양한 업무를 자동화 할 수 있는 환경이 조성되었으며, 이에 따라 해당 솔루션을 도입하는 기업 및 기관들이 증가하고 있다. 최근 RPA 솔루션에서 도입된 기술로는 사용자가 업무를 어떠한 형태로 수행하고 있는지를 파악하기 위해서, 업무 수행시 필요한 문자 및 이미지의 정보를 인식하기 위한 기술과 사용자의 업무환경에서 접근해야 하는 UI 객체 인식 기술이 있다[1-3]. 이러한 기술들은 앞에서 언급하였듯이, 일반적으로 Deep Learning 기술을 이용하여 인식 엔진을 구현하고 있다[4]. RPA 도입단계에서는 업무 수행을 위해 사용되는 ERP, 그룹웨어, 메일시스템 등을 이용하여 업무를 자동화 하는 사례가 많았으며, 최근 RPA 업무 도입 확산과 더불어 점차적으로 인식 처리가 필요한 업무를 자동화하기 위한 시도와 사례가 증가하고 있다. 이러한 인식 처리 업무는 이미지 또는 PDF 형태로 제공하는 문서를 인식하고 이를 처리하기 위한 내용이 대부분이며, 특히 문서내 전체 문자를 인식하기보다 문서 내 제목 및 특정 영역의 문자 정보를 인식하여 이를 판단하는 업무가 많다. 하지만, 인식을 위해 제공되는 문서 파일의 형태(해상도, 문서 양식의 표준화, 글자크기, 문서 내 노이즈 여부, 문서의 기울기)에 따라, 문자 인식 정확도의 차이가 많이 발생하고 있다. 이러한 정확도가 낮은 RPA 솔루션은 인식처리가 필요한 업무 자동화에 적용하기가 어려운 실정이므로, 많은 RPA 솔루션 개발 업체들은 인식 정확도를 높이기 위한 연구를 진행하고 있다. 인식 정보의 정확도 향상은 좀 더 다양한 업무에

자동화가 적용될 수 있도록 해주며, 단순·반복적인 업무 수행에 따른 생산성 저하와 수작업에 의한 휴먼 에러의 발생, 민감한 데이터의 보안 취약성 등을 개선하여 업무 절차 표준화, 업무 편의성 제고, 데이터 신뢰성 확보, 디지털 노동력의 효율적인 활용 및 비용절감, 고생산성 업무환경을 구축할 수 있도록 해준다. 이러한 인식 정확도를 향상시키기 위한 방법은 딥러닝 기술을 활용한 인식 AI 모델 구축과 전, 후처리를 통한 방법이 있다. 일반적으로 잘 구축된 모델은 가장 효과적인 문자 인식 정확도 향상 기술이나, 서로 다른 다양한 업무환경에 적합한 AI 모델을 학습을 통해 재구축하기에는 학습용 데이터 수집, 정제, 가공, 모델 개발 과정 등이 필요하기 때문에, 특정 기간 동안 업무자동화를 적용하고 개발하기 위해서는 한계가 존재한다. 그에 반해, 전/후처리 기술은 문서내 모든 문자를 인식하고 처리하는 업무가 아닌 특정 영역의 문자 인식 업무에서 비교적 빠르게 해당 업무 환경에 맞는 인식기반 업무 자동화 환경을 구축할 수 있는 장점을 가지고 있다. 이에 따라, 본 논문에서는 딥러닝 기반 문자 인식 엔진에 전/후 처리기 기술을 적용할 경우 얼마나 정확도가 향상되는지를 확인하고 RPA 인식 엔진과 연계 기술을 구현하였다.

2. 관련 연구

OCR 기술은 패턴 인식의 한 분야로, 사람이 쓰거나 인쇄한 문자의 형태를 인식하여 기계가 읽을 수 있는 문자로 변환하는 것을 의미한다[5-7]. 이러한 문자를 인식하기 위한 방안들이 다양하게 연구되고 있으며, 인식율을 높이기 위해 전처리, 그리고 후처리 과정으로 위치시키기도 한다[8-12]. 다음 Fig. 1은 딥러닝 기반 OCR 파이프라인을 도식화한 것이다[5]. 전통적인 OCR 절차는 인식 문서에 대한 전처리 과정을 거친 이미지에서 글자를 검출하며, 검출된 문자에 대한 인식, 인식된 문자의 후처리 과정으로 되어 있다. 최근 OCR 기술에서도 딥러닝 기술이 적용된 연구가 많이 일어나고 있으며, 특히 검출(Detection), 인식(Recognition) 과정에서 CNN, RNN과 같은 AI 모델을 많이 이용하고 있다.

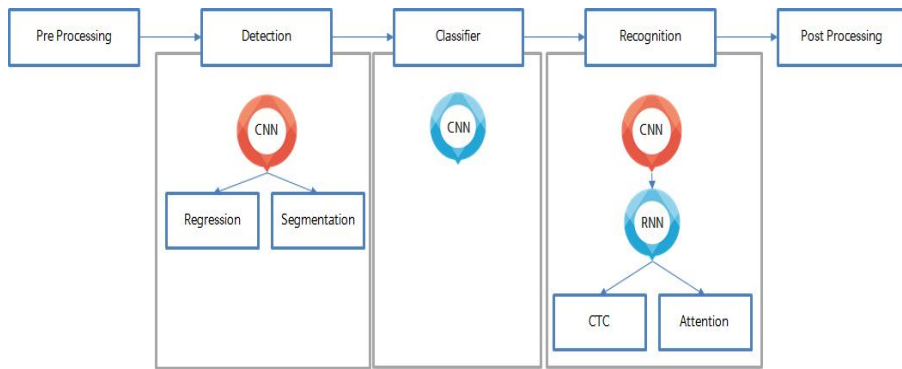


Fig. 1. Pipeline based on Deep Learning

전처리 과정은 글자 인식 성능을 향상시키기 위하여 입력된 이미지를 보정하는 단계로써 이진화, 이미지 대비 보정, 이미지 기하학적 변환(기울임 보정, 키스톤 보정), 노이즈 제거와 같은 기술이 존재한다. 이진화 기술은 임계값을 이용하여 입력되는 이미지의 컬러를 이진화된 흑백 이미지로 변환하는 기술이다. 일반적으로 전역적(Global Thresholding) 이진화 보다 적응적 이진화(Adaptive Thresholding) 기법을 많이 사용한다. 이미지 대비 보정은 문자와 문서내 배경을 분리하기 쉽게 문자의 경계를 선명하게 해주는 기법으로써, 일반적으로 CLAHE(Contrast Limited Adaptive Histogram Equalization)을 많이 사용한다.

기울임 보정 기술은 기울어진 상태로 되어 있는 이미지를 인식을 향상을 위해 원래 모습으로 복원하는 기술이다. 키스톤 보정은 스캔한 문서나 캡처된 이미지가 사각형 형태가 아닌 사다리꼴 형태를 가질 때 아핀

변환 등을 수행하여 사각형 모습의 문서형태로 변환하고 유용한 데이터가 없는 영역을 제거하는 기술이다. 노이즈 제거 기술은 이미지 이진화와 대비 보정을 통해 사라지지 않은 노이즈를 추가적으로 제거하는 기술로써, 가우시안 블러(Gaussian Blur), 중앙 흐림, 양방향 필터링 같은 다양한 필터를 이용한다.

후처리 기술으로써는 인식된 문자 정보에 대한 보정 작업으로써 사전을 활용하여 잘못된 인식된 문자를 유사한 단어로 대체하여 정확도를 높이는 기술이다.

3. 기능 구현

3.1 시스템 구성

본 논문에서 제안하는 시스템은 Fig. 2와 같이

PDF, 이미지로 되어 있는 문서에 대한 전처리를 담당하는 전처리기, 전처리 이후 문자 인식을 처리하는 OCR 인식 엔진, 인식된 문자에 대한 후처리를 담당하는 후처리기로 구성되어 있다. 전처리에서는 이미지 흑백화 및 이진화, 이미지 기울기 보정, 이미지 RGB 추출 기능을 포함하여 업무에서 사용되는 문서들을 처리하는 모듈이며, OCR 인식엔진은 입력된 문서 이미지내에 문자 정보를 추출하고 인식하는 부분이다. OCR 인식엔진은 이미 구현된 AutomateOne2 ibizBot 제품을 이용하였다. 후처리 부분은 업무 환경에서 사용되는 문서의 사전 데이터를 구축하고 잘못 인식된 문자 정보의 수정 및 유사 단어 제시 기능을 포함하고 있다. 아울러 전후처리기와 OCR 인식엔진과 연계는 내부 API를 이용하는 방법과 외부 스크립트를 통해 연계하는 방법으로 구분될 수 있으며, 본 논문에서는 ibizbot에서 동작가능한 자동화 스크립트를 구현하였다. 내부 API를 이용하는 방법은 구현된 전/후처리기 API를 OCR 엔진에서 직접 호출하여 처리하는 방법이며, 자동화 스크립트를 이용하는 방법은 OCR 엔진에서 동작하는 스크립트에서 전/후처리기를 직접 실행하고 처리 결과를 이용하는 방법이다.

3.2 OCR 전/후처리기 기능 개발

o 전/후처리 절차

본 논문에서 전/후처리를 수행하는 절차는 Fig. 3과 같다. 기업에서 사용하는 PDF 또는 이미지 문서를 전처리기를 통해 이미지로 변환하고 전처리를 수행하여 인식엔진을 통해 문자로 인식한다.

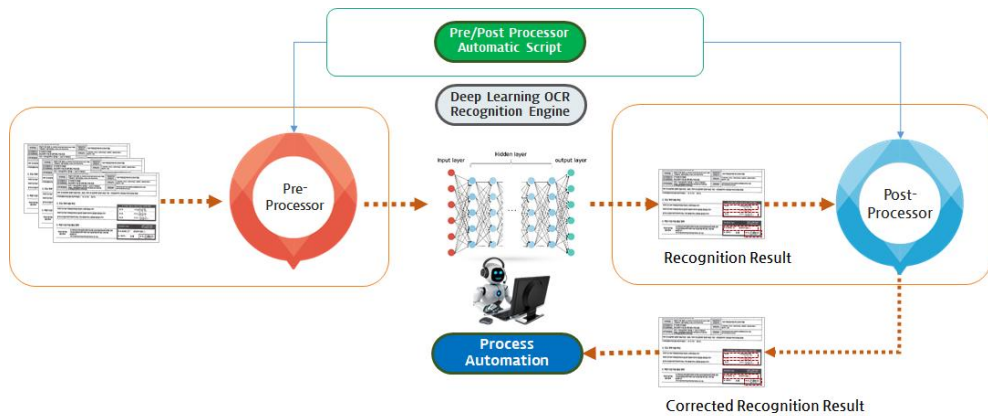


Fig. 2. System Component

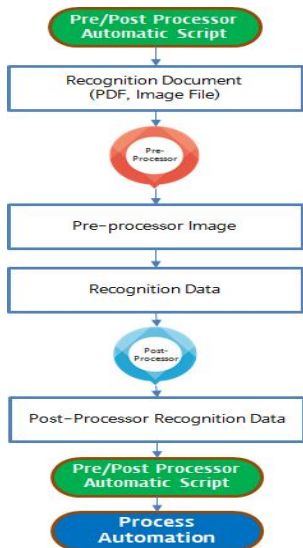
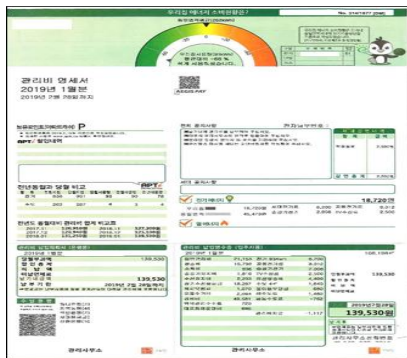


Fig. 3. Processing Procedure

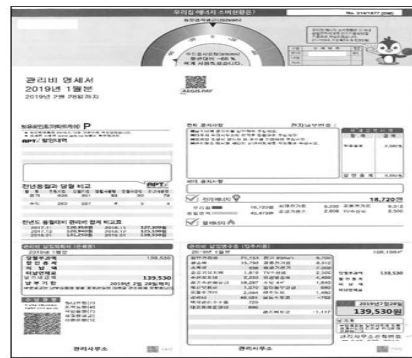
이러한 전/후처리 기능은 RPA 로봇에서 API를 호출하여 이용할 수 있으며, 또는 업무 프로세스 스크립트를 통해 동작시킬 수 있다. 본 논문에서는 RPA 로봇에서 사용되는 스크립트를 통해 전/후처리를 동작시키고 그 결과를 이용할 수 있는 구조로 개발하였다.

o 전처리 기능

본 논문에서는 전처리 기능으로 이미지 흑백화 및 이진화, 이미지 기울기 보정, 이미지 RGB 추출 기능을 적용하였다. 이미지 흑백화 및 이진화 기능은 문자 인식시 처리율 향상과 인식율을 높이기 위해 사용되는 기능이다. Fig. 4와 Fig. 5는 흑백화 및 이진화 처리 결과와 이를 구현한 소스코드를 보여준다.



GrayScale and Binarization Processing(Before)



GrayScale and Binarization Processing(After)

Fig. 4. Black/White and Binary processing result

```

import cv2
import os
import numpy as np

def main():
    imagePath = r"C:\WRPA\OCR_Automateone\Receive\Sample_Image.jpg"
    img_array = np.fromfile(imagePath, np.uint8)
    img = cv2.imdecode(img_array, cv2.IMREAD_COLOR)
    grayscale_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    ret, dst = cv2.threshold(grayscale_img, 0, 255, cv2.THRESH_OTSU)

    result, encoded_img = cv2.imencode('.jpg', grayscale_img)

    if result:
        with open("C:\WRPA\OCR_Automateone\Template\템플릿" +
                + os.path.basename(imagePath), mode='w+b') as f:
            encoded_img.tofile(f)

if __name__ == '__main__':
    main()
    
```

Fig. 5. Black/White and Binary processing Source Code

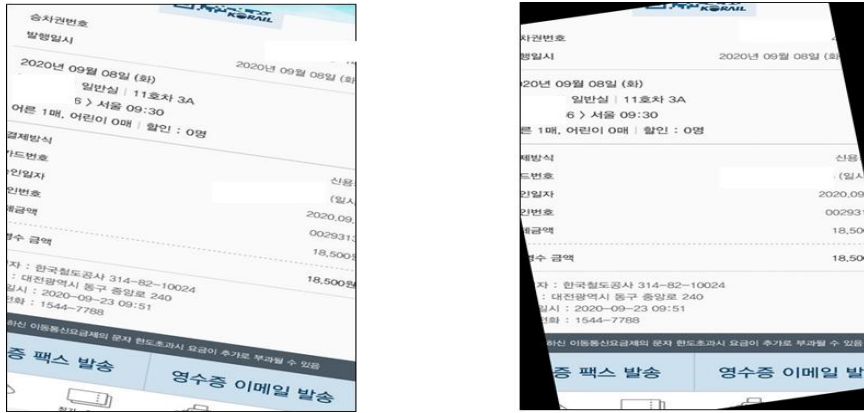


Fig. 6. Tilt correction processing result

```

참조 0개
static void Main(string[] args)
{
    Program program = new Program();
    Bitmap bitmap = ProcessBitmap(@"C:\WRPA\OCR_Automateone\Receive\Sample_Img.jpg");
    if (bitmap != null) { bitmap.Save(@"C:\WRPA\OCR_Automateone\Template\Sample_result.jpg",
        System.Drawing.Imaging.ImageFormat.Jpeg); }
}

private static int $STRIP_COUNT = 10;
참조 1개
private static Bitmap ProcessBitmap(string filePath)
{
    Bitmap sourceBitmap = default;
    if (File.Exists(filePath))
    {
        sourceBitmap = Image.FromFile(filePath) as Bitmap;
        StripInfo stripInfo = new StripInfo(sourceBitmap, STRIP_COUNT);

        int stripX1 = 2;
        int stripX2 = 6;

        double angle = SkewCalculator.GetRotateAngle(stripInfo, stripX1, stripX2);
        angle = (angle + 180 / Math.PI);
        Bitmap targetBitmap = BitmapHelper.RotateBitmap(sourceBitmap, angle);

        return targetBitmap;
    }

    return sourceBitmap;
}
    
```

Fig. 7. Tilt correction processing Source Code

특히 이미지의 RGB 값의 분포에 따라 이진화시 오차
 려 문자정보 소실, 불명확한 형태로 변환될 여지가 있다.
 이에 따라 RGB 분포 값에 따라 적절한 이진화 방법을
 이용하여 이미지의 특정 RGB값 추출 기능을 구현하였다.

이미지 기울기 보정 기능은 기울어진 상태로 문자를
 인식할 경우 오류가 발생할 확률이 높기 때문에 이를
 보정하여 인식율을 높이기 위한 기능이며, Fig. 6과
 Fig. 7은 실행 결과 및 소스코드를 보여준다.

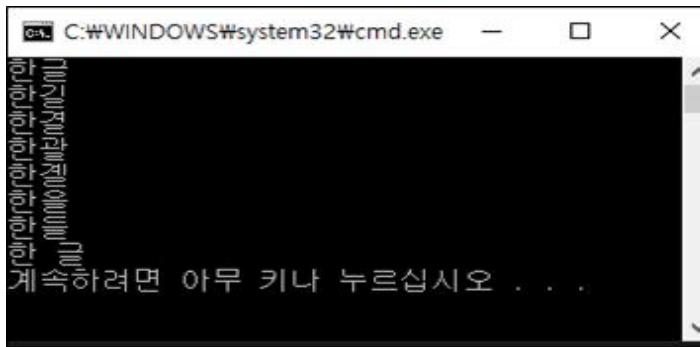


Fig. 8. Similar word processing results

```

참조 0개
static void Main(string[] args)
{
    string Sample_Spell = "한글";

    string File_Ko_aff = @"C:\RPAA\OCR_Automateone\Base\ko.aff";
    string File_KO_dic = @"C:\RPAA\OCR_Automateone\Base\ko.dic";

    using (Hunspell hunspell = new Hunspell(File_Ko_aff, File_KO_dic))
    {
        List<string> Spell_List = hunspell.Suggest(Sample_Spell);
        foreach (string text in Spell_List)
        {
            Console.WriteLine(text);
        }
    }
}
    
```

Fig. 9. Similar word processing Source Code

○ 후처리기 기능

후처리 기능으로는 문자 인식 이후 오류 단어를 수정하기 위한 사전(Dictionary) 데이터를 구축하였으며, 잘못 인식된 문자 정보에 대한 유사 단어를 제시하는 기능을 개발하였다. 또한 특정 업무 분야에서 사용하는 단어를 추가할 수 있도록 기능을 구현하였다. Fig. 8과 Fig. 9는 “한글”이라는 문자에 대해 유사 단어를 제시하는 기능의 실행 결과 화면 및 소스코드이다.

최종적으로 처리된 결과는 인식된 문자와 공백 등이 포함되어 Fig. 10과 같은 형태로 제공된다. Fig. 10의 인식 결과에서 보듯이 전체적인 문자 인식 정확도는 매우 높은 편이나 일부 문자(π, ₩) 등과 같이 대체할 수 없는 단어로 인식되는 경우도 있음을 확인 할 수 있었다.

4. 제안 기법의 평가

4.1 실험 환경 및 방법

본 논문에서 개발한 기능과 인식정확도를 측정하기 위하여 아래와 같은 운영환경을 구성하고 테스트 하였다.

가. 운영 환경

- 1) 운영체제 : 윈도우 10, 인텔 i5-4670, 16GB
- 2) 테스트 소프트웨어 : 전처리기 및 후처리기 모듈, 전/후처리 스크립트 모듈
- 3) 인식 엔진 : AutomateOne2 iBizbot 솔루션
- 4) 테스트 대상 문서 자료 : 1,769건(기차 영수증, 대중교통 영수증, 카드 영수증, 관리비 고지서, 보험 관련 인보이스 등)

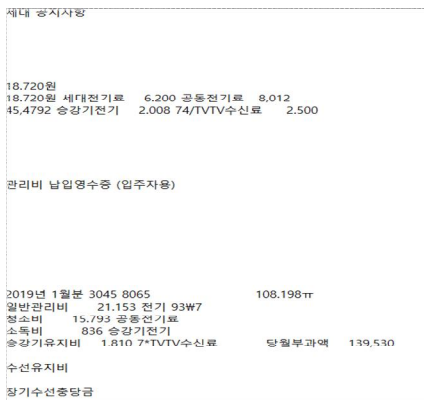


Fig. 10. Character Recognition results

나. 시험 방법

- 본 논문에서는 인식정확도를 측정하기 위해 대상 문서로부터 추출하고자 하는 문자 정보의 정답을 미리 정의하고 인식된 문자와 비교

- 인식 대상 문서를 수집하고 해당 문서의 문자 정보를 사전에 파악하여 Table 1. 인보이스의 인식 문자 정보와 같은 정답을 준비하였으며, 인식엔진(AutomateOne2 Solution)을 이용하여 문자를 인식

Table 1. Recognition Characters of Invoice

Answer Information
MASTER
Policy holder(MCIS)
Policy holder(Invoice)
Insured person(MCIS)
Insured person(Invoice)
Items(MCIS)
Items(Invoice)
HS CODE
Basic Clause
Special Clause

- 인식된 문자와 사전 준비된 정답 문자를 비교하여 정확도 계산

- 전/후처리를 적용한 경우와 적용하지 않은 경우를 구분하여 정확도를 측정하고 비교하였으며, 아래의 Table 2는 시험에 사용된 기술을 말함

Table 2. Using Techniques of Pre/Post Processor in Test

Classification	Pre-processor	OCR recognition engine	Post-processor
Case 1 No prep/post-processor	- Back/White and Binary processing	Deep learning-based engine	- NO
Case 2 Using Prep/post-processor	- Back/White and Binary processing - Tilt correction processing - Denoising	Deep learning-based engine	- Construct Dictionary - Similar words

4.2 실험 결과

o 인식정확도 테스트 결과

테스트 결과 전/후처리를 적용하지 않은 경우는 82.44% 정도의 정확도를 가졌으나 적용 후에는

95.15%로 12.71%의 정확도 향상을 가져왔다. 또한 일반적으로 종이로 된 문서를 스캔할 경우 해상도(DPI)가 낮은 경우도 있으며, 이러한 이미지의 인식정확도는 현저하게 낮아졌다. 본 논문의 실험을 통해 해상도가 낮은 이미지의 경우 31.65% 정도의 인식율을 보였으나 전/후처리를 적용할 경우 45%정도로 인식율이 향상되는 것을 확인하였다. 결국 문서 내 문자 정보를 이용하는 업무를 자동화하기 위해서는 디지털화된 인식 문서의 해상도가 최소 200 DPI 이상이 되어야 적용 가능하다. 본 논문을 통해 개발된 전후처리 기능을 활용할 경우 RPA 솔루션을 이용한 업무자동화시 문서 인식 정확도를 높일 수 있고, 업무 분야별 사전 데이터를 손쉽게 구축할 수 있기 때문에, 현재 업무 자동화에 많이 도입되고 있지 않은 문서 인식을 활용한 업무가 자동화될 수 있는 기능을 제공함을 확인하였다.

Table 3. Result of Test

Data Type	Recognition Rate	
	No Pre/Post Processor	Pre/Post Processor
DataSet_Ticket	88.54%	95.90%
DataSet_Invoice	86.18%	98.24%
DataSet_etc(200DPI Resolution)	72.60%	91.30%
DataSet_etc(100dpi Resolution)	31.65%	45%
Average(except DataSet_Etc(100DPI and 200DPI Resolution)	82.44%	95.15%

5. 결론

최근 4차 산업혁명 시대가 도래함에 따라 인공지능 기술을 이용한 다양한 솔루션들이 시장에 소개되고 있다. 특히 업무 자동화를 수행하기 위해 AI 기술이 접목된 RPA 솔루션은 최근 공공기관, 기업들에서 도입을 서두르고 있는 실정이다. 이러한 RPA 솔루션을 이용하여 업무를 자동화할 경우, 문자 인식을 통해 수행해하는 업무는 RPA 솔루션의 인식 정확도가 높지 않으면 제외될 수밖에 없다. 하지만 다양한 문서를 판독하고 이러한 정보를 입력해야 하는 업무는 매우 반복적이거나 단순할 수 있어, RPA 업무에 적합한 형태를 갖는 경우가 많다. 따라서 본 논문에서 개발한 기능을 통해 RPA 솔루션을 도입하고자 하는 기관 및 기업이 문서 인식을 이용한 업무를 자동화하는 경우에도 자동화가

가능해질 수 있다. 또한, 특정 업무 영역에서의 이미지 문서의 인식 정확도 향상을 위한 기술이므로, 개발완료 초기에는 특정 업무 영역 자동화에 활용될 수 있으며, 추후 사용자 사전 정보들이 누적될 경우 다양한 업무영역, 범용적 업무 영역에서도 활용 할 수 있다.

REFERENCES

- [1] K. B. Kim. (2019). A Study of Convergence Technology in Robotic Process Automation for Task Automation. *Journal of Convergence for Information Technology*, 9(7), 8-13
DOI : /10.22156/CS4SMB.2019.9.7.008
- [2] C. S. Lee. (2018). *RPA, Changes to the Office, Current Status and Implications of RPA at Home and Abroad*, DIGIECO. KT Economic Management Institute (Online).
<http://www.smallake.kr/wp-content/uploads/2018/08/rpa-office%EC%97%90-%EA%B0%80%EC%A0%B8%EB%8B%A4-%EC%A4%84-%EB%B3%80%ED%99%94201807191531982430925.pdf>
- [3] C. Kroll, A. Bujak, V. Darius, W. Enders & M. Esser. (2016). Robotic Process Automation-Robots conquer business processes in back offices. *Capgemini Consulting*, 1-48.
- [4] Y. G. Hyun & J. Y. Lee. (2018). Trends Analysis and Future Direction of Business Process Automation, RPA (Robotic Process Automation) in the Times of Convergence. *Journal of Digital Convergence*, 16(11), 313-327.
DOI : 10.14400/JDC.2018.16.11.313
- [5] S. K. Lee & S. H. Hong. (2018). *Can Analog weather data be digitized with OCR?*. Kakao AI Report (Online). <https://brunch.co.kr/@kakao-it/319>
- [6] J. W. Kim, S. T. Kim, J. Y. Yoon & Y. I. Joo. (2015). A Personal Prescription Management System Employing Optical Character Recognition Technique. *Journal of the Korea Institute of Information and Communication Engineering*, 19(10), 2423-2428.
DOI : 10.6109/jkiice.2015.19.10.2423
- [7] J. H. Roh & D. S. Choi, (2014), Character recognition using a target string, Proceedings of Symposium of the Korean Institute of communications and Information Sciences,
- [8] S. G. Hong, S. S Hwang & S. D. Kim. (2012). A license plate recognition system robust to vehicle location and viewing angle. *Journal of the Institute of Electronics and Information Engineers*, 49(12), 113-123.

DOI : 10.5573/ieek.2012.49.12.113

- [9] J. H. Ju & J. S. Oh, (2012). An adaptive binarization algorithm for degraded document images. *Journal of Korean Institute of Communications and Information Sciences*, 37(7), 581-585.
DOI : 10.7840/KICS.2012.37.7A.581
- [10] C. H. Son & H. M. Park. (2012). Fast multiple-image-based deblurring method. *Journal of the Institute of Electronics Engineers of Korea SP*, 49(4), 49-57.
- [11] J. W. Song, N. R. Jung & H. S. Kang. (2015), Container BIC-code region extraction and recognition method using multiple thresholding. *Journal of the Korea Institute of Information and Communication Engineering*, 19(6), 1462-1470.
DOI : 10.6109/jkiice.2015.19.6.1462
- [12] G. C. Lee & J. S. Yoo. (2017), Development an Android based OCR Application for Hangeul Food Menu. *Journal of the Korea Institute of Information and Communication Engineering*, 21 (5), 951-959.
DOI : 10.6109/jkiice.2017.21.5.951

장 창 북(Chang-Bok Jang)

[정회원]



- 2003년 2월 : 한남대학교 공과대학원 컴퓨터공학과 (공학석사)
- 2007년 2월 : 한남대학교 공과대학원 컴퓨터공학과 (공학박사)
- 2008년 2월 ~ 2011년 8월 : 한남대학교 BK21 연구교수

- 2012년 3월 ~ 현재 : (주)알투스소프트 연구소장
- 관심분야:유비쿼터스, 인공지능, 딥러닝, 업무자동화
- E-mail : chbjang@r2soft.co.kr

김 기 봉(Ki-Bong Kim)

[종신회원]



- 1993년 2월 : 충남대학교 대학원 전산학과 (이학석사)
- 1998년 8월 : 충남대학교 대학원 전산학과 (이학박사)
- 1995년 9월 ~ 1997년 2월 : 해전대학교 전임강사

- 1997년 3월 ~ 현재 : 대전보건대학교 컴퓨터정보과 교수
- 관심분야 : 데이터베이스, 의료정보시스템
- E-mail : kbkim@hit.ac.kr