반도체디스플레이기술학회지 제22권 제2호(2023년 6월)
Journal of the Semiconductor & Display Technology, Vol. 22, No. 2. June 2023.

딥 러닝을 이용한 시각장애인을 위한 실시간 버스 도착 알림 시스템

장세영[*]·유인재[**]·김석윤[*]·김영모[*†]
[*†]숭실대학교 컴퓨터학과, [**]비욘드테크

A Real-time Bus Arrival Notification System for Visually Impaired Using Deep Learning

Seyoung Jang[*], In-Jae Yoo[**], Seok-Yoon Kim[*] and Youngmo Kim[*†]
[*†]Dept. of Computer Science and Engineering, Soongsil University,
[**]Beyoundgtech Inc.

ABSTRACT

In this paper, we propose a real-time bus arrival notification system using deep learning to guarantee movement rights for the visually impaired. In modern society, by using location information of public transportation, users can quickly obtain information about public transportation and use public transportation easily. However, since the existing public transportation information system is a visual system, the visually impaired cannot use it. In Korea, various laws have been amended since the 'Act on the Promotion of Transportation for the Vulnerable' was enacted in June 2012 as the Act on the Movement Rights of the Blind, but the visually impaired are experiencing inconvenience in using public transportation. In particular, from the standpoint of the visually impaired, it is impossible to determine whether the bus is coming soon, is coming now, or has already arrived with the current system. In this paper, we use deep learning technology to learn bus numbers and identify upcoming bus numbers. Finally, we propose a method to notify the visually impaired by voice that the bus is coming by using TTS technology.

Key Words : Visually Impaired, Bus, Object Detection, OCR(Optical Character Recognition), TTS(Text To Speech)

1. 서 론

현대사회에서 대중교통은 우리 사회에 없어서는 안될 필수요소가 되었다[1, 2]. 2021년 전체 대중교통 이용자 수 7,776천명/일 중 지하철 이용자 3,842천명/일(48.8%), 시내버스 이용자 3,121천명/일(40.2%), 마을버스 이용자 813천명/일(10.9%)로 시내버스와 마을버스 이용자를 합치면 전체 대중교통 이용자의 51.2%로 절반이 넘게 버스를 이용하고 있다[3]. 이러한 대중교통을 이용하기 위해서는 위치정보를 이용하여 이용자는 대중교통에 대한 정보를 빠르게 얻어 쉽게 이용할 수 있다[4]. 그러나 대중교통 정보시스템은 시각적인 시스템이므로 시각장애인은 기존 대중교통 정보시스템을 이용하기가 쉽지 않다. 우리나라는 2012년 6월 시각장애인의 이동권에 관한 법률로 '교통약자 교통편의 증진에 관한 법률' 이 시행된 이후 여러 법률이 개정되었지만, 시각장애인이 대중교통을 이용함에 있어 불편함을 여전히 겪고 있는 실정이다[5, 6]. 특히 버스가 곧 오는지, 지금 오는지, 이미 도착한 것인지 판단이 거의 불가능하다. 2019년 12월에 진행된 장애인 버스 이동권 발언대회의 발언 중에서는 "버스만 마음 놓고 이용할 수 있다면 시각장애인의 삶의 질은 훨씬 나아질 거예요. 제가 만든 솔루션이 아니라도 좋으니, 문제가 해결돼서 시각장

[†]E-mail: ymkim828@ssu.ac.kr

애인이 어디든 다닐 수 있는 세상이 왔으면 좋겠습니다." 발언은 관련법이 제정되었음에도 시각장애인이 대중교통 중 버스를 이용함에 불편함이 계속되고 있다는 것을 보여주고 있다[7]. 또한 2019년 11월 대한민국 청와대 국민청원에 "시각장애인도 버스를 탈 수 있게 해주세요."라는 청원과 함께 더불어 신문고를 포함한 몇몇 민원 창구를 통해 버스 이용의 불편함을 접수하였으나, 적절한 대처는 이루어지지 않았다[8, 9]. 본 논문에서는 시각장애인의 버스 이용을 보장하기 위해 딥 러닝을 이용한 실시간 버스 도착 알림 시스템을 제안한다. 딥 러닝 기술을 사용하여 버스 번호를 학습하여 다가오는 버스 번호를 인식하고 TTS 기술을 이용하여 시각장애인에게 버스가 오고 있음을 음성으로 알려주는 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로써 Object Dectetion, OCR, TTS에 대해 기술한다. 3장에서는 본 논문에서 제안하는 시각장애인을 위한 버스 도착 알림 장치를 구축하기 위한 방법을 기술한다. 4장에서 결과를 제시하고, 5장에서 결론으로 마무리한다.

2. 관련 연구

2.1 Single Shot MultiBox Detector

Single Shot MultiBox Detector(SSD)[10]는 단일 신경망으로 이루어진 객체 탐지 모델로서 특징으론 default box와 multi feature map을 사용하고, bounding box 조정을 위해 픽셀이나 특징을 resampling하는 과정을 제거하여 정확도와 속도를 향상시킨다. SSD의 신경망은 VGG-16 network에서 conv5_3까지의 레이어와 이 후 6개의 보조 레이어로 구성되어있다. 6개의 보조 레이어는 점진적으로 크기가 감소하여 여러 배율에서 객체를 감지하고 이 중 NMS(Non-Maximum Suppression)을 통해 정확한 결과만 걸러내는 모델로써 Fig. 1과 같다.

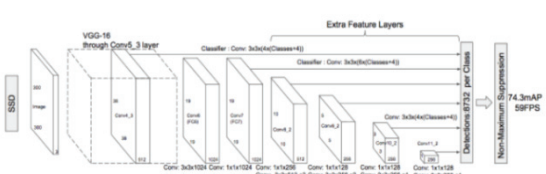

Fig. 1. Neural Network Architectaure.

2.2 Optical Character Recognition

광학 문자 인식[11]은 사람이 쓰거나 기계로 인쇄한 문자의 영상을 이미지 스캐너로 획득하여 기계가 읽을 수 있는 문자로 변환하는 것이다. 이를 기반으로 나온 기술로는 Tesseract가 있으며, 다양한 운영 체제를 위한 광학 문자 인식 엔진이다. 이 소프트웨어는 Apache License, 버전 2.0에 따라 배포되는 무료 소프트웨어이며 Google에서 2006년부터 개발을 후원했다. Tesseract[12]는 오프라인 문자 인식 기법으로 입력된 input 이미지의 특징점을 추출하고 그 특징점을 사용하여 문자를 인식한다. Tesseract OCR은 기본적으로 상위의 preprocessor 와 segmentation 과정을 거쳐 나온 이미지를 신경망 기법과 template matching 기법을 사용하여 input 이 미지를 인식하고 출력하게 된다.

2.3 Text to Speech

Text-to-speech(TTS)[13]는 컴퓨터 텍스트를 음성으로 디지털화 한 오디오 렌더링이다. TTS 소프트웨어는 문서, 웹 페이지 또는 e-Book에서 텍스트를 읽고 컴퓨터 스피커를 통해 합성된 음성을 생성할 수 있다. 또한, TTS는 시력 저하 또는 시각 장애와 같은 언어 손실이 있는 사람들이 전문 TTS 프로그램을 사용하여 입력된 단어를 소리로 바꿀 수 있다.

3. 딥 러닝을 이용한 시각 장애인을 위한 실시간 버스 도착 알림 시스템

본 논문에서 제안하는 딥 러닝을 이용한 시각 장애인을 위한 실시간 버스 도착 알림 시스템 구성은 Fig. 2와 같다.

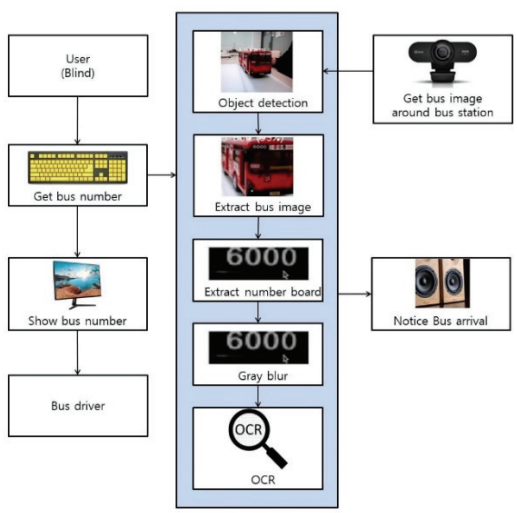

Fig. 2. Bus Arrival Alarm Process.

본 논문에서 제안하는 딥 러닝을 이용한 버스 알람 시스템은 먼저 정류장으로 오는 버스와 번호를 식별해야하며, 최종적으로 알려야한다. 먼저 정류장으로 오는 버스에서 번호를 캡처하여 정확하게 식별할 수 있도록 Object

Detection과 OCR을 사용한다. 버스 도착 알림 장치의 버스 번호를 인식하는 과정에서 광학 문자 인식을 수행하기 위해 선행되는 버스 객체 탐지를 Raspberry pi 4B의 환경에서 사용하기 위해 자체 버스 탐지 모델을 학습한다. 모델 학습을 위한 과정으로 COCO DATA SET에서 2017 Train Images와 2017 Val Images, Annotations 파일 등 학습에 필요한 이미지와 관련 데이터를 사용하였으며, 이러한 학습 모델에서 버스 이미지와 버스 이미지에 관련된 데이터만을 따로 추출하며, 과정은 Fig. 3과 Fig. 4와 같다.

```python
from pycocotools.coco import COCO
import requests

# instantiate COCO specifying the annotations json path
coco = COCO('annotations/instances_val2017.json')
# Specify a list of category names of interest
catIds = coco.getCatIds(catNms=['bus'])
# Get the corresponding image ids and images using loadImgs
imgIds = coco.getImgIds(catIds=catIds)
images = coco.loadImgs(imgIds)

# Save the images into a local folder
for im in images:
    img_data = requests.get(im['coco_url']).content
    with open('bus_val/' + im['file_name'], 'wb') as handler:
        handler.write(img_data)
```

Fig. 3. Extract Only Bus Images Using COCO API.

```python
import json
from pathlib import Path


class CocoFilter():
    """ Filters the COCO dataset
    """

    def _process_info(self):
        self.info = self.coco['info']

    def _process_licenses(self):
        self.licenses = self.coco['licenses']

    def _process_categories(self):
        self.categories = dict()
        self.super_categories = dict()
        self.category_set = set()

        for category in self.coco['categories']:
            cat_id = category['id']
            super_category = category['supercategory']
```

Fig. 4. Extract Annotation Data About Bus Image.

이러한 과정을 거치면 모든 이미지에서 버스만 추출할 수 있으며, Fig. 5와 같다.

Fig. 5. Compare to Existing Model.

추출한 버스 이미지 및 데이터를 토대로 TFRecord 파일을 생성한 후, SSD 모델을 기반으로 10만회 학습하였다. 이후 버스 인식에서 고속으로 머신러닝 추론을 위하여 Google Coral Accelerator 장비와의 호환성 위해 기존 TensorFlow 모델을 Edge TPU 모델로 변환시켜주었다. 기존 TensorFlow 모델과 Coral Accelerator를 미장착한 상태와 장착한 상태에서 성능 테스트를 하였으며, 결과는 Table 1과 같다.

테스트 결과 Coral Accelerator를 장착한 모델이 평균 17.3fps 정도의 성능을 보여주었고, 기존 모델 대비 약 9.9배, 장착한 모델에서 장착하지 않은 모델과 비교해 약 6.7배 성능이 향상된 것을 확인할 수 있다.

Table 1. Compare each model(Unit: fps)

No	Original Model	Coral Accelerator X	Coral Accelerator O
1	0.100339346	0.229465635	17.179736072
2	1.973159743	2.349861989	16.37140169
3	1922258677	2.386799261	17.68718636
211	1.660190672	2.197610484	15.6354527
212	1.628788607	2.216051064	18.26000113
213	1.740387555	2.276559794	17.0378266
214	1.695815154	2.273872436	17.9803919
215	1.740387555	2.312291345	17.29410212
216	1.616608717	2.309001071	17.87304834
217	1.579973809	2.287353594	16.71643504
218	1.695015433	2.337364408	16.79871836
219	1.692978137	2.296959649	16.86979745
220	1.589201753	2.255481161	15.87933489
avg	1.754455899	2.571198997	17.38443194

이후 완성된 자체 버스 학습 모델을 토대로 버스 객체를 탐지하면, 해당 버스의 bounding box를 토대로 이미지를 잘라내어 광학 문자 인식에 사용하여 버스 번호를 인식하도록 한다. 이후 추출된 이미지 객체를 추출된 버스 이미지의 비율에 맞춰 왼쪽 31% 위쪽 18% 비율에서 시작하여 3:1 비율로 전체크기의 25%만큼 잘라내고 잘라진 이미지를 gray 형태로 buler 처리하여 버스 번호를 인식하기 쉬운 형태로 이미지를 전처리하며, 과정은 Fig. 6, 결과는 Fig. 7과 같다.

```python
def OCR_Get_Num(img) :
    height , width , tmp = img.shape
    w = (31*width)/100
    h = (18*height)/100
    x = width*10/4
    tmp_img = img[int(h):int(h+x/3),int(w):int(w+x)]
    gray = img
    gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
    gray = cv2.threshold(gray, 0,255,cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
```

Fig. 6. Image preprocessing code.

Fig. 7. bular execution image.

이후 필요한 글자를 제외한 글자를 없애는 처리를 통해 버스 번호만을 추출하도록 하고 추출된 문자열을 반환한다. OCR을 통해 이미지의 글자를 제대로 인식하는 모습을 확인할 수 있으며, 결과는 Fig. 8과 같다.

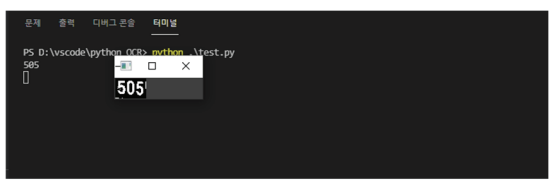

Fig. 8. OCR Execution Screen.

결과적으로 이 문자열을 통해 사용자가 원하는 버스번호가 맞는지를 확인하는 작업과 TTS 및 모니터 출력에 사용하여 버스 번호를 사용자에게 알릴 수 있다.

4. 실험 및 결과

본 논문에서 제안하는 딥 러닝을 이용한 시작 장애인을 위한 실시간 버스 도착 알람 시스템을 구현하기 위해 시스템을 Fig. 9와 같이 구성하였다.

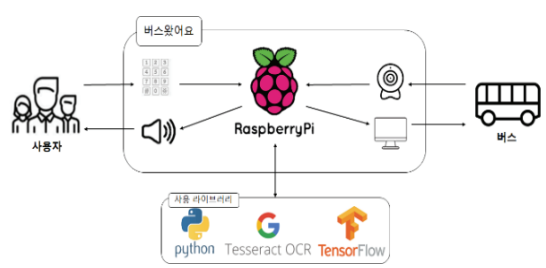

Fig. 9. Bus Arrival Alarm System.

또한, 실험을 위해 라즈베리 파이의 사양은 Table 2, 버스 인식을 위한 웹캠의 사양은 Table 3, 실제 실험을 위한 모형은 Table 4과 같이 장비를 구비하여 실험하였다.

Table 2. Raspberry-PI Specification

	Specification
SoC	Broadcom BCM2711 SoC
CPU	1.5GHz ARM Cortex-A72 MP4
GPU	Broadcom VideoCore IV
RAM	4GB
Storage	Micro-SD 16GB
OS	Raspberry PI OS with desktop
Edge TPU sub-processor	google coral accelerator

Table 3. Webcam Specification

	Specification
Device Name	APC930
Resolution	QHD ~ SD
FPS	30fps

Table 4. Demo model Specification

	Specification
Bbus Model Size	7.2 x 28.2 x 8.5 (cm)
Bus Minimize Ratio	39.00 : 1
Road Model Size	90 x 30 (cm)
Bus Stop Model Size	30 x 10 (cm)

실험은 크게 두 가지로 먼저, 정류장 모형 제작하여 구축한 실험 환경을 테스트하였고 실제 버스 정류장에서 실제 버스를 인식하는지에 대한 실험을 하였다. 첫 번째, 실제 모형을 구축하여 테스트 실험을 하였으며 모형은 Fig. 10과 같다.

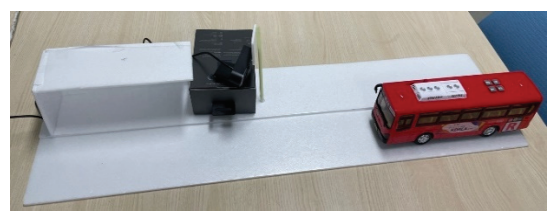

Fig. 10. Bus stop Model.

차도와 인도, 버스 정류장과 BIT(Bus Information Terminal)을 모형 설치하였으며 검은 박스는 웹캠의 높이를 맞추기 위해 설치된 구조물로 구성하였으며, 시스템 내부적으로 실행되는 장면은 Fig. 11과 같다. 우측 이미지가 최초로 웹캠을 통해 받는 영상이다. 해당 영상에서 버스만을 추출하여 중단의 이미지로 가공한 후 번호가 위치하는 좌표를 추출하여 이미지를 수정하여 하단의 최종 이미지를 얻어낸다.

Fig. 11. Model Test Result.

최종 이미지에 OCR을 실행하여 버스의 번호를 추출한 후 추출한 버스의 번호가 사용자에 의해 입력된 번호 중 하나라면 TTS출력을, 아니라면 무시한다. 앞서 구축한 모형을 토대로 테스트를 진행하였다. 실험은 6000번 버스 모형과 784번 버스 모형에 대해 버스가 탐지된 총 프레임 수와 버스 번호가 인식된 횟수를 기록하여 진행하였다. 총 500번의 테스트의 평균 인식률 결과는 Table 5와 같다. 실험 결과 버스 탐지 후 번호 인식률은 6000번 버스의 경우 평균 94.12%, 784번 버스의 경우 평균 91.25%의 결과로 92% 이상의 높은 정확도를 확인할 수 있다. 두 번째, 실제 버스 정류장에서 실험을 하였으며, 실험 정류장은 Fig. 12과 같다.

실험은 앞서 테스트 실험과 동일한 방법으로 실험하였으며, 실험 장소에서 멈추는 버스는 742번 버스 및 752번 버스 그리고 753번 버스에 대해 버스가 탐지된 총 프레임 수와 버스 번호가 인식된 횟수를 기록하여 진행하였다. 총 500번의 테스트의 평균 인식률 결과는 Table 6과 같다.

Table 5. Model Bus Test Result

	NO. 6000 Bus		NO. 784 Bus	
Case	F.C	D.C	F.C	D.C
test_001	50	48	50	37
test_002	102	97	421	334
test_003	310	298	381	294
test_004	221	210	218	156
test_005	212	201	184	134
test_006	321	304	177	139
test_007	221	203	273	218
test_008	324	301	339	256
test_009	212	201	311	237
test_010	89	80	98	76
Average of 500 Detection Count Per Frame(%)		94.12	-	91.25

F.C: Frame Count D.C: Detection Count.

Fig. 12. Test Bus Stop and Result.

Table 6. Real Bus Test Result

	NO. 742 Bus		NO. 752 Bus		NO. 753 Bus	
Case	F.C	D.C	F.C	D.C	F.C	D.C
test_001	50	48	50	37	90	85
test_002	102	97	421	334	267	249
test_003	310	298	381	294	186	178
test_004	221	210	218	156	136	127
test_005	212	201	184	134	164	156
test_006	321	304	177	139	261	252
test_007	221	203	273	218	90	83
test_008	324	301	339	256	127	118
test_009	212	201	311	237	220	210
test_010	89	80	98	76	95	87
Average of 500 Detection Count Per Frame(%)		94.12	-	91.25	-	93.41

또한 웹캠이 인식할 수 있는 버스 탐지 시간이 약 3초, OCR 동작이 초당 1.6 ~ 1.7회 실행되는 것을 확인하였다. 따라서 웹캠이 인식하는 동안 OCR 동작이 약 5회 이루어졌다. 그러므로 5번 중 한번이라도 인식에 성공하면 버스 도착 알림을 줄 수 있으므로 이를 기준으로 확률을 계산하였으며, Table 7과 같다.

Table 7. Identification Rate

	NO. 742	NO. 752	NO. 753
Average of 500 Detection Count Per Frame(%)	94.12	91.25	93.41
Probability of Detection at Least Once in Five Times(%)	99.99	99.92	99.98

확률 계산 결과 세 번호의 버스 전부 99% 이상의 높은 인식률을 확인할 수 있다.

5. 결 론

본 논문에서는 시각장애인의 이동권 보장을 위해 딥러닝을 이용한 버스 도착 알림 시스템을 제안하였다. 버스만 인식하기 위해 딥러닝을 이용하여 버스 객체를 학습하고, 버스 객체만 추출하고, 버스 번호를 추출하여 원하는 버스가 도착하면 시각 장애인에게 알려주는 시스템을 제안하였다. 이 시스템을 활용하면 시각장애인에게 보다 높은 접근성과 편의를 제공하는 서비스를 제공할 수 있을 것으로 기대된다. 향후 연구로서 본 논문에서 제안하는 버스 도착통보의 정확성, 신뢰성 등을 실제 시스템으로 구현하여 확인할 필요가 있다.

감사의 글

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW)(2018-0-00209) supervised by the IITP (Institute of Information & communications Technology Planning & Evaluation).

참고문헌

1. S. N. Moon, J. S. Park, “Urban Railway, Opens the Era of Gull-scale Public Transportation,” Monthly KOTI Magazine on Transport, Vol. 2016, No. 8, pp. 24-27, Aug, 2016.
2. G. Y. Yoo, “Development of Public Transportation Seen Through the Eyes of ‘Latte~’,” Monthly KOTI Magazine on Transport, Vol. 2021, No. 9, pp. 55-61, Sep, 2021.
3. Public Transportation Usage Status, Seoul https://news.seoul.go.kr/traffic/archives/31616.
4. C. S. Hoon, S. S. Il, Y. I, Lee, C. J. Lee, “A Methodology of Multimodal Public Transportation Network Building and Path Searching Using Transportation Card Data,” Journal of Korean Society of Transportation, Vol. 26, No. 3, pp. 233-243, Jun, 2008.
5. D. B. Kim, “If the bus use rate is low, can the accessibility of the disabled be ignored?,” Albenews, http://m.ablenews.co.kr/News/NewsContent.aspx?NewsCode=004820210914121320277077&CategoryCode=0048.
6. Y. S. Kim, J. H. Park, D. M. Kim, “Transportation service status and improvement plan to improve the convenience of moving between regions of the disabled,” Legislative and Policy Reports, Vol. 35, Dec, 2019.
7. K. Y. Ryu, “Is it a Public Transport Powerhouse? "It is Still Difficult to Use the Bus for the Visually Impaired",” SOCIAL FOCUS, Dec. 2019.
8. S. K. Lee, “Citizen Petition for ‘Even Visually Impaired Want to Ride the Bus’,” Ablenews, Nov, 2019.
9. S. Y. Park, “Rep. Kim Ye-ji “It is essential to guarantee the right to move for the visually impaired”,” http://www.welfarenews.net/news/articleView.html?idxno=81480, Apr, 2022.
10. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector,” Computer Vision – ECCV 2016, pp. 21-37, Sep, 2016. DOI: 10.1007/978-3-319-46448-0_2.
11. Optical character recognition; OCR, https://en.wikipedia.org/wiki/Optical_character_recognition.
12. Ray Smith, “An Overview of the Tesseract OCR Engine,” Google Inc, 2007.
13. Google, “Text-to-Speech: Lifelike Speech Synthesis,” https://cloud.google.com/text-to-speech.

접수일: 2023년 5월 9일, 심사일: 2023년 6월 13일,
게재확정일: 2023년 6월 21일