Smart Glasses Application System for Visually Impaired People Based on Deep Learning

Jyun-You Lin
Department of Computer Science and Information Engineering
National Taitung University
Taitung County, Taiwan
a0226asdf@gmail.com

Chi-Lin Chiang
Department of Computer Science and Information Engineering
National Taitung University
Taitung County, Taiwan
s11016200@gmail.com

Meng-Jin Wu
Department of Computer Science and Information Engineering
National Taitung University
Taitung County, Taiwan
z087850780@gmail.com

Chih-Chiung Yao
Department of Computer Science and Information Engineering
National Taitung University
Taitung County, Taiwan
brianyiao0105@gmail.com

Ming-Chiao Chen
Department of Computer Science and Information Engineering
National Taitung University
Taitung County, Taiwan
joechen@nttu.edu.tw

***Abstract*— The recent development of deep learning has promoted object detection to make rapid progress. In recent years, smart wearable technology is rapidly becoming part of everyday life, including watches, glasses and many other wearable items have been enriched with technology. In this study, we propose a smart glasses application system for visually impaired people based on deep learning. The system can use voice response to visually impaired people about the objects in front of them by uploading the photos to our backend object detection system through the camera function of smart glasses, and then download the text descriptions of the result and then use the text-to-speech function. According to the experimental results, the proposed system takes 3.788 seconds from uploading photos to making voice results, and the success rate of object detection is 96.3%. We hope that this application system can assistant visually impaired people to understand the surrounding environment and interact more closely with the people around them to improve the quality of life of them.**

Keywords— smart glasses, visually impaired people, object detection, text-to-speech

I. INTRODUCTION

Inspired by the video that a grandfather who born colorblind, and for his birthday, received glasses that have color correcting lenses that help him see color [1]. Furthermore, the Microsoft's Seeing AI research project [2], which brings the cloud and AI together to design an intelligent app to help the blind navigate his day. As space design and disability facilities for the visually impaired are not yet widespread in Taiwan, they have caused many restrictions. Therefore, we combine the smart glasses and object detection deep learning system, can inform visually impaired people around the situation, giving them self-care convenience.

Based on the above point, this work uses EPSON BT-300 smart glasses as a medium for input and output. It is based on Android system to perform image recognition from the perspective of the visually impaired people. The visually impaired people can take pictures with built-in camera and upload it to the back-end object detection deep learning system. Then the system downloads detection results and use Android built-in TTS function to convert the text to voice to inform the visually impaired people so that they can clearly understand the current environment.

II. RELATED WORKS

In 2007, Google released the Android operating system platform [3], it's based on Linux OS, using Java as its main development language. Android offers the free cross-platform software development kit and provides Android studio integrated development environment for the programmer. Android is an OS with strong market potential, it has many hardware support vendors who let those apps on Android platform grow quickly.

Text-to-Speech (TTS) function can convert text into voice, user just need to inputs text and it will automatically generate correspond voice. Android already have built-in TTS API, so we can use the Chinese version to implement Text-to-Speech function.

In the past few years, the field of wearable devices technology has been in its infancy, and various related industries have developed smart glasses. We choose EPSON BT-300 smart glasses as our wearable development device. The EPSON BT-300 is an Intel Atom X5 1.44GHZ CPU and Android OS 5.1. It supports still image format, audio format and even 3D image presentation, and it can use Wi-Fi for wireless Internet access and upload information. Its lens with 5 million pixels, and a handheld controller to assist users in application. Furthermore, EPSON BT-300 also provides the standard Android and EPSON APIs, such as display control, UI control, camera control, sensor control and Bluetooth.

In recent years, deep learning has made great progress. Among the object detection, YOLO (You Only Look Once) v3 [4–6] is a new and fast object detection method. Its architecture is shown in Figure 1. It treats object detection as a Single Regression Problem, directly puts the entire image into the neural network to predict the position of Bounding Box and its corresponding category probability.

The advantages of YOLO v3 are as follows:

- YOLO v3 detects objects very fast.
- YOLO v3 can avoid background errors and generate False Positives.

978-1-7281-4999-8/20/$31.00 ©2020 IEEE

	Type	Filters	Size	Output
	Convolutioal	32	3*3	256*256
	Convolutioal	64	3*3/2	128*128
1*	Convolutioal	32	1*1	
	Convolutioal	64	3*3	
	Residual			128*128
	Convolutioal	128	3*3/2	64*64
2*	Convolutioal	64	1*1	
	Convolutioal	128	3*3	
	Residual			64*64
	Convolutioal	256	3*3/2	32*32
8*	Convolutioal	128	1*1	
	Convolutioal	256	3*3	
	Residual			32*32
	Convolutioal	512	3*3/2	16*16
8*	Convolutioal	256	1*1	
	Convolutioal	512	3*3	
	Residual			16*16
	Convolutioal	1024	3*3/2	8*8
4*	Convolutioal	512	1*1	
	Convolutioal	1024	3*3	
	Residual			8*8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 1. YOLO v3 model

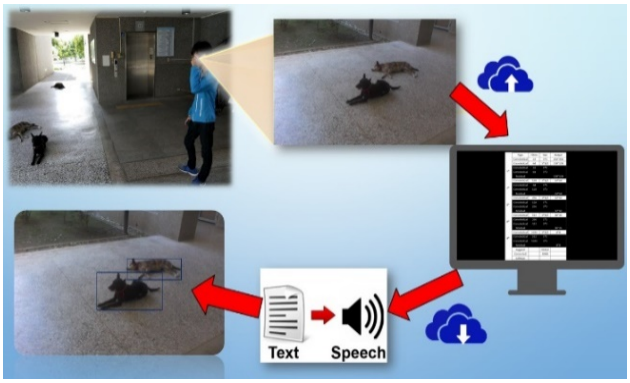


Fig. 2. System flow diagram

- YOLO v3 can learn the generalization features of object.

III. METHOD

Our system flow diagram is shown in figure 2. The visually impaired people use smart glasses to take pictures of the scene, and the photos are immediately stored in SD card, and uploaded to the back-end server. The YOLO v3 deep learning model will perform object detection, and then output the text format of the result that can be read by TTS. TTS then outputs the detection result in a voice manner, so that the visually impaired people can clearly know the objects in front of them and their position.

A. Smart Glasses System

This paper uses EPSON BT-300 smart glasses as our wearable development device. The following will introduce the camera module and voice output module of the smart glasses system respectively.

Camera Module

The detailed steps of the camera module [7, 8] are described as follows:

- Obtain camera permissions and data read/write permissions
- Set and initialize the UI components that display the preview screen, and then start camera and open the preview screen.
- Set the buttons on the screen and controller of the EPSON BT-300 to take pictures.
- Save the captured photos to the SD card.
- The module will record this Capture Session, the storage space will be deleted and released when taking next photos.

B. Upload and Download System

The system will upload the photos taken by smart glasses to the back-end server, and call YOLO v3 deep learning module to perform object detection, and then download the results to the smart glasses to output the final voice results for the visually impaired people. The following will be explained in detail for the Client and Server module respectively.

Client side module [9]

1. Upload module:

- Use Socket to establish a TCP connection with the server.
- Read the captured photo file into the photo buffer of the Client.
- Transfer the contents of the photo buffer of the client to the server.

2. Download module:

- Receive the data downloaded from the server and store it in text buffer of the client.
- Write data in the text buffer of the client the text file (txt).
- Close the connection.

Server side module [10]

1. Upload module:

- Wait for the client to establish a connection.
- Receive the data uploaded by the client and store it in the photo buffer of the server.
- Write the contents of the photo buffer on the server side into the photo file.

Call the YOLO v3 object detection system.

2. Download module:

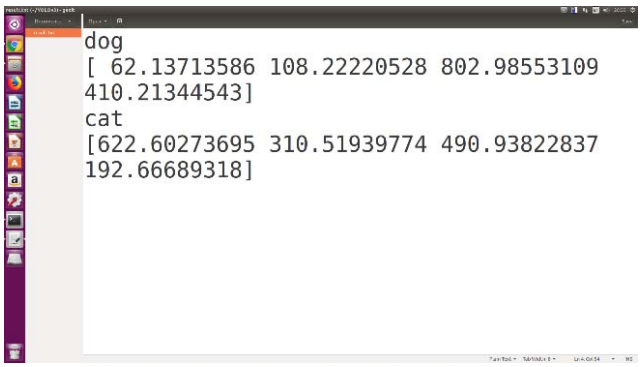


Fig. 3. Text file output of object detection result.

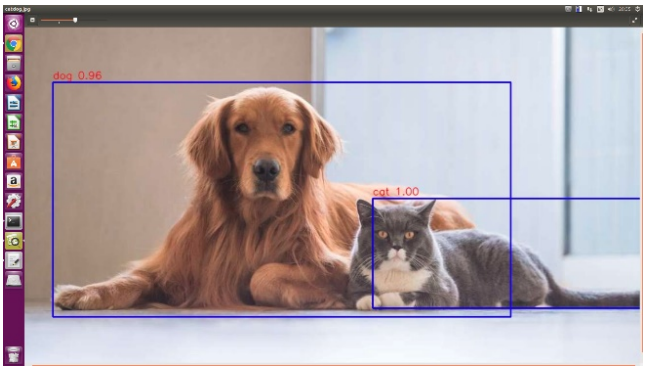


Fig. 4. Result of object detection.

- Read the text file generated by the YOLO v3 and read it into the text buffer on the server side.
- Download the data in the text buffer of the server to the client.
- Close the connection.

C. *YOLO v3 Deep Learning Object Detection System*

This work uses YOLO v3 based on Tensorflow and Keras versions to implement object detection.

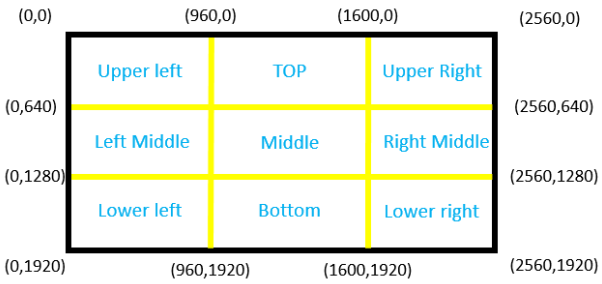
Detection Steps

1. Download the weight file (yolov3.weights) that the authors have trained using the COCO dataset. This weight file can recognize 80 objects (such as cats, dogs, horses, birds, bicycles, cars, trucks, traffic lights, chairs, and beds).
2. Convert the network structure and weight files of Darknet to the h5 files supported by Keras.
3. Modify the execution file to enable the model to additionally generate text file containing the object classes shown in Fig. 3. The upper left coordinate (x, y) of bounding box of the object, and the length and width (w, h) of the bounding box of the object.
4. The object detection deep learning system outputs the results, as shown in Fig. 4. It also shows the bounding box.

D. *Text-To-Speech (TTS)*

This section describes the detailed process of the TTS system to convert text into speech.

1. Check if TTS is installed and available, install it if it is not installed, and set the language to Chinese.



	range of coordinate
upper left	(0,0)~(960,640)
top	(960,0)~(1600,640)
upper right	(1600,0)~(2560,640)
left middle	(0,640)~(960,1280)
middle	(960,640)~(1600,1280)
right middle	(1600,640)~(2560,1280)
lower left	(0,1280)~(960,1920)
bottom	(960,1280)~(1600,1920)
lower right	(1600,1280)~(2560,1920)

Fig. 5. Object orientation description.

TABLE I. HARDWARE SPECIFICATION

CPU	Intel Core i7-8700K 3.40GHz
GPU	NVIDIA GTX 1080Ti 11G
RAM	16GB
HDD	2TB
OS	Ubuntu 16.04

2. Organize the recognition results into the text required for speech output.

- Use switch/case command to distinguish different objects into different units. The format is: “have” + size + unit + object + orientation
- Orientation discrimination: As shown in Fig. 5, the X-coordinate of the object is added with W/2, and the Y-coordinate of the object is added with H/2 is the center coordinate of the object to determine the corresponding orientation of the object.

3. Voice output the sentence, using engine.speak command to output the above organized text (for example: “have” + 1 + dog + on the left). When size>1, then we add “respectively” before the orientation and add “and” between the two orientations (for example: “have” + 2 + dogs + respectively + on the left and right).

IV. EXPERIMENT

This section demonstrates the experimental results of our system. The hardware specification of the object detection deep learning system is shown in Table 1.

In our experiments, we will use three kinds of objects: (1) life goods (e.g. backpacks, umbrellas, clocks, and so on), (2) transportation vehicles (e.g. cars, trains, airplanes, and so on), (3) animals (e.g. cats, dogs, person, and so on). The relevant results are shown in Tables 2, 3, 4, 5. (In the following tables, Object*n: n is the number of the object in a photo)

1. Life goods: As shown in Table 2, the uploading average time is 2.449 seconds, and the detection

TABLE II. THE DETECTION RESULT OF LIFE GOODS CLASS

No	input photo	test result	total spend time (sec)	detection rate
1	book*5	book*4	3.182	80%
2	chair*3	chair*3	3.431	100%
3	refrigerator*1	refrigerator*1	3.409	100%
4	spoon*4	spoon*3	3.476	75%
5	suitcase*3	suitcase*3	3.304	100%
6	umbrella*1	umbrella*1	3.366	100%
7	book*1	book*1	3.51	100%
8	kite*1	kite*1	5.119	100%
9	clock*1	clock*1	3.951	100%
10	fire hydrant*1	fire hydrant*1	4.225	100%
11	dog*1 frisbee*1	dog*1 frisbee*1	3.372	100%
12	potted plant*1	potted plant*1	4.793	100%
13	book*1	book*1	3.233	100%
14	surfboard*1	surfboard*1	3.762	100%
15	teddy bear*2	teddy bear*2	3.338	100%
16	tie*1	tie*1	3.261	100%
17	wine glass*2	wind glass*2	3.495	100%
18	stop signal*1	stop signal*1	3.413	100%
19	keyboard*1	keyboard*1	3.584	100%
20	smart phone*1	smart phone*1	3.752	100%
Average time			3.649	97.80%

TABLE III. THE DETECTION RESULT OF TRANSPORTATION VEHICLE CLASS

No	input photo	test result	total spend time (sec)	detection rate
1	bear*2	bear*2	3.66	100%
2	bird*5	bird*5	3.638	100%
3	cat*1 man*1	cat*1 man*1	4.383	100%
4	bicycle*1 man*1	bicycle*1 man*1	4.79	100%
5	bicycle*1	bicycle*1	3.32	100%
6	bicycle*2 man*2	bicycle*2 man*2	3.97	100%
7	bus*1 man*4	bus*1 man*3	3.405	80%
8	bus*4 man*3	bus*4 man*2	3.488	86%
9	bus*1 car*1	bus*1 car*1	3.428	100%
10	car*1	car*1	3.795	100%
11	car*1	car*1	3.397	100%
12	car*1	car*1	5.392	100%
13	motorcycle*1 man*1	motorcycle*1 man*1	3.565	100%
14	motorcycle*1 man*1	motorcycle*1 man*1	4.666	100%
15	motorcycle*1 man*1	motorcycle*1 man*1	3.48	100%
16	train*1	train*1	3.898	100%
17	train*1	train*1	3.421	100%
18	train*1	train*1	3.807	100%
19	motorcycle*2	motorcycle*2	3.591	100%
20	car*1	car*1	3.674	100%
Average time			3.838	98.30%

time is 0.124 seconds, and the download average time is 1.068 seconds. The average time taken from the smart glasses to the voice output is 3.649 seconds. The recognition rate is 97.8%.

2. Transportation vehicles: As shown in Table 3, the average upload time is 2.506 seconds, and the detection time is 0.127 seconds, and the average download time is 1.195 seconds. The average time taken from the smart glasses to the voice output is 3.838 seconds. The recognition rate is 98.3%.

3. Animals: As shown in Table 4, the average upload time is 2.401 seconds, the detection time is 0.126 seconds, and the average download time is 1.078 seconds. The average time taken from the smart glasses to the voice output is 3.614 seconds. The recognition rate is 90.7%

TABLE IV. THE DETECTION RESULT OF ANIMAL CLASS

No	input photo	test result	total spend time (sec)	detection rate
1	bear*2	bear*2	3.399	100%
2	bird*5	bird*5	3.343	100%
3	cat*1 man*1	cat*1 man*1	3.66	100%
4	cattle*3	cattle*3	3.898	100%
5	dog*1	dog*1	3.486	100%
6	elephant*2	elephant*2	3.721	100%
7	giraffe*4	giraffe*3	3.574	75%
8	horse*1	horse*1	3.937	100%
9	man*1	man*1	4.259	100%
10	sheep*4	sheep*4	3.489	100%
11	zebra*3	zebra*3	3.48	100%
12	bear*3	bear*1	3.523	33%
13	cat*1 dog*1	cat*1 dog*1	3.88	100%
14	cattle*4	cattle*3 dog*1	3.62	75%
15	giraffe*2	giraffe*2	3.31	100%
16	man*1 horse*1	man*1 horse*1	3.601	100%
17	zebra*2	zebra*1	3.287	50%
18	dog*1	dog*1	3.585	100%
Average time			3.614	90.70%

TABLE V. THE DETECTION RESULT OF FOOD CLASS

No	input photo	test result	total spend time (sec)	detection rate
1	apple*3	apple*3	4.632	100%
2	apple*1	apple*1	4.376	100%
3	apple*2	apple*2	3.392	100%
4	banana*1	banana*1	3.483	100%
5	banana*1	banana*1	3.93	100%
6	banana*1	banana*1	3.29	100%
7	cake*1	cake*1	3.752	100%
8	cake*1	cake*1	4.187	100%
9	cake*1	cake*1	3.698	100%
10	carrot*5	carrot*5	5.206	100%
11	carrot*1	carrot*1	3.508	100%
12	carrot*3	carrot*3	5.793	100%
13	orange*2	orange*2	3.859	100%
14	orange*3	orange*3	3.475	100%
15	orange*1	orange*1	3.315	100%
16	donut*6	donut*6	3.61	100%
17	donut*3	donut*2	5.894	67%
18	donut*1	donut*1	4.005	100%
19	orange*1	orange*1	3.591	100%
Average time			4.052	98.20%

4. Food: As shown in Table 5, the average upload time is 2.631 seconds, the detection time is 0.124 seconds, and the average download time is 1.288 seconds. The average time taken from the smart glasses to the voice output is 4.052 seconds. The recognition rate is 98.2%

Based on the above results, the average recognition rate is 96.3%; among them, those with larger objects (e.g. The transportation vehicles class) have higher recognition rate; in contrast, those with smaller objects (e.g. The life goods, animals and food), the recognition rate is relatively low; it

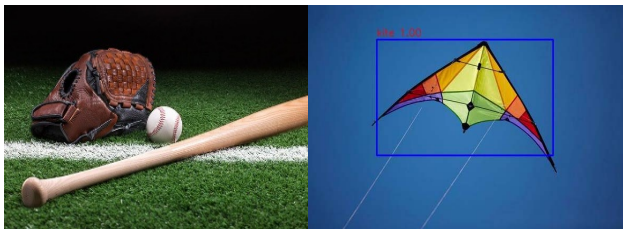


Fig. 6. The test data #1 and #2 of life goods class

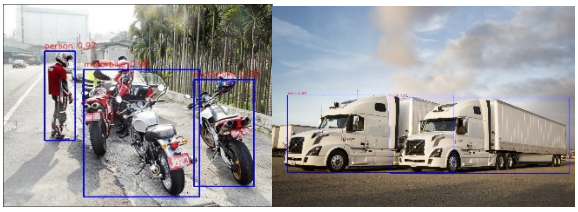


Fig. 7. The test data #3 and #4 of transportation vehicle class.

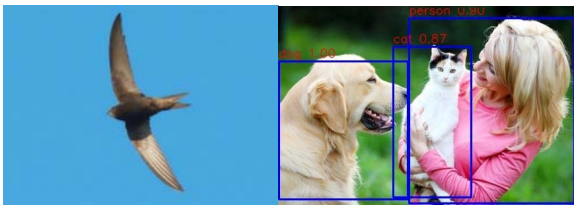


Fig. 8. The test data #5 and# 6 of animal class

may be because some objects are too similar or difficult to detect and cause recognition errors. For example, the black dogs and the black cats are more likely to recognize errors. Moreover, some dogs and cats have variety ears are similar and black is less distinguishable.

In addition, the average upload time is 2.496 seconds, the average detection time is 0.125 seconds, and the average download time is 1.195 seconds. The average time taken from the smart glasses to the voice output is 3.788 seconds.

The test photos used in this experiment are: The life goods class (Fig. 6), the transportation vehicles class (Fig. 7), the animals class (Fig. 8).

V. Conclusion

This work uses EPSON BT-300 smart glasses, combined with deep learning and TTS, to develop an application system suitable for visually impaired people to adapt environment. According to the experimental results, the average time taken from the smart glasses to the voice output is 3.788 seconds, the overall recognition rate is 96.3%. We hope the application system can help the visually impaired people obtain more living resources to communicate with the outside world, and can better understand the environment unknown to them. Let the visually impaired people can interact more closely with the people around them, without fear of being blurred and uncertain.

Acknowledgment

This work was partly supported by Ministry of Science and Technology in Taiwan, under grants MOST 107-2813-C-143-025-E and 108-2813-C-143-002-E.

References

[1] JustineReed97, *66 year old Bodybuilder William Reed sees color for first time*. LeRoy, New York, 2017.

[2] Microsoft, *Seeing AI 2016 Prototype - A Microsoft research project*. 2016.

[3] "Android Developers," *Android Developers*, Aug. 03, 2018. https://developer.android.com/ [Accessed Sep. 01, 2019].

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779–788, Jun. 2016. doi: 10.1109/CVPR.2016.91.

[5] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *ArXiv180402767 Cs*, Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[6] Larry, "YOLOv3." Nov. 30, 2018, [Online]. Available: https://github.com/xiaochus/YOLOv3 [Accessed: Sep. 21, 2019]

[7] P. J. Deitel and H. M. Deitel, *Adroid how to program with an introduction to Java*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2017.

[8] M. Basta, "Widgetgrid-ar." Mar. 14, 2015, [Online]. Available: https://github.com/mattbasta/widgetgrid-ar [Accessed: May 18, 2019].

[9] E. R. Harold, *Java network programming*, 4th edition. Beijing: O'Reilly, 2014.

[10] M. Summerfield, *Programming in Python 3: a complete introduction to the Python language*, 2nd ed., Fully rev. ed. Upper Saddle River, NJ: Addison-Wesley, 2010.