Flying Guide Dog: Walkable Path Discovery for the Visually Impaired Utilizing Drones and Transformer-based Semantic Segmentation

Haobin Tan[1], Chang Chen[1], Xinyu Luo[1],
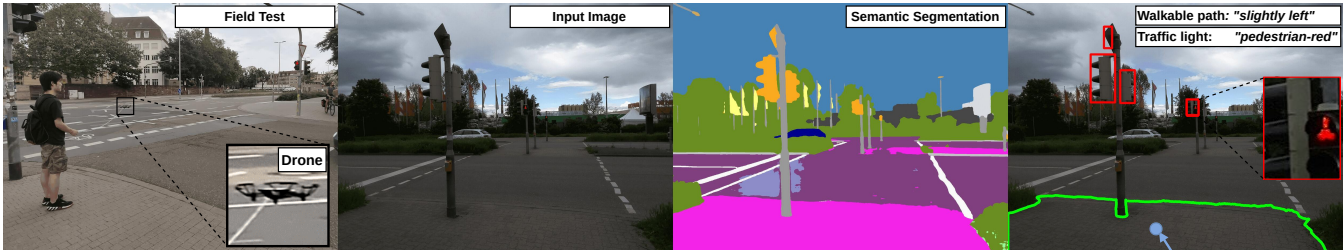Jiaming Zhang[2], Constantin Seibold[2], Kailun Yang[2] and Rainer Stiefelhagen[2]

Fig. 1: Flying guide dog prototype. Images from left to right are: field test in front of an intersection, input image from the drone's perspective, semantic segmentation result, interpreted result with centroid, boundary and bounding boxes for the drone control algorithm.

Abstract—**Lacking the ability to sense ambient environments effectively, blind and visually impaired people (BVIP) face difficulty in walking outdoors, especially in urban areas. Therefore, tools for assisting BVIP are of great importance. In this paper, we propose a novel "flying guide dog" prototype for BVIP assistance using drone and street view semantic segmentation. Based on the walkable areas extracted from the segmentation prediction, the drone can adjust its movement automatically and thus lead the user to walk along the walkable path. By recognizing the color of pedestrian traffic lights, our prototype can help the user to cross a street safely. Furthermore, we introduce a new dataset named Pedestrian and Vehicle Traffic Lights (PVTL), which is dedicated to traffic light recognition. The result of our user study in real-world scenarios shows that our prototype is effective and easy to use, providing new insight into BVIP assistance.**

I. INTRODUCTION

It has been predicted that the number of blind and visually impaired people (BVIP) will increase to more than 115 million by 2050 [1]. Lacking the ability to sense ambient environments effectively, they face difficulty in walking outdoors, especially in urban areas. Therefore, assistive tools are indispensable. Previous electronic assistance systems focused on providing guiding information via audio [2] or tactile feedback with wearable equipment [3]. Although users can approach the destination based on these feedbacks, they need to repeatedly finetune their orientation during the navigation. To tackle this problem, we build a novel "flying guide dog" prototype, exploring the combination of drone and street view semantic segmentation. According to the user study conducted in [4], drone navigation is more accurate and faster as it gives a continuous feedback in the direction of travel. Performing semantic segmentation on frames captured by the

drone's camera, a variety of ambient visual information can be extracted, such as sidewalks, crosswalks, and traffic lights. Based on its perception of the environment, the drone adjusts itself and leads the user to walk safely. To follow the drone, the user holds a string attached to the drone. Besides, the user receives voice prompts via a Bluetooth bone conduction headphone.

Discovering the walkable path is one of the major functions of our prototype. Based on the segmentation prediction, walkable area (*e.g.* sidewalks, crosswalks) can be discovered by their corresponding colors. In order to make the drone keep flying along the walkable path safely, we develop a control algorithm so that the drone can automatically adjust its direction and velocity according to the estimated centroid of the sidewalk. Another function is assisting the user to pass the pedestrian traffic light, *i.e.* street crossing. Our prototype not only distinguishes pedestrian crossing lights from other types of traffic lights but also recognizes their color. Since there is currently no dedicated traffic light dataset containing both pedestrian and vehicle traffic lights, we introduce a new dataset called *Pedestrian and Vehicle Traffic Lights (PVTL)*.

To verify the effectiveness of our system, we further conduct a user study in real-world scenarios. The result indicates that our prototype is effective for visually impaired assistance and easy to use.

In summary, our contributions are summarized as follows:

- We propose a novel "flying guide dog" prototype utilizing the drone and semantic segmentation, which can effectively help BVIP to walk in urban scenarios safely.
- We develop a control algorithm to enable the drone to fly along the walkable path automatically and to interact with users via voice feedbacks in crossing streets.
- We introduce a new traffic light dataset named *Pedestrian and Vehicle Traffic Lights (PVTL)*, which focuses on recognizing the category and the color of traffic lights simultaneously.
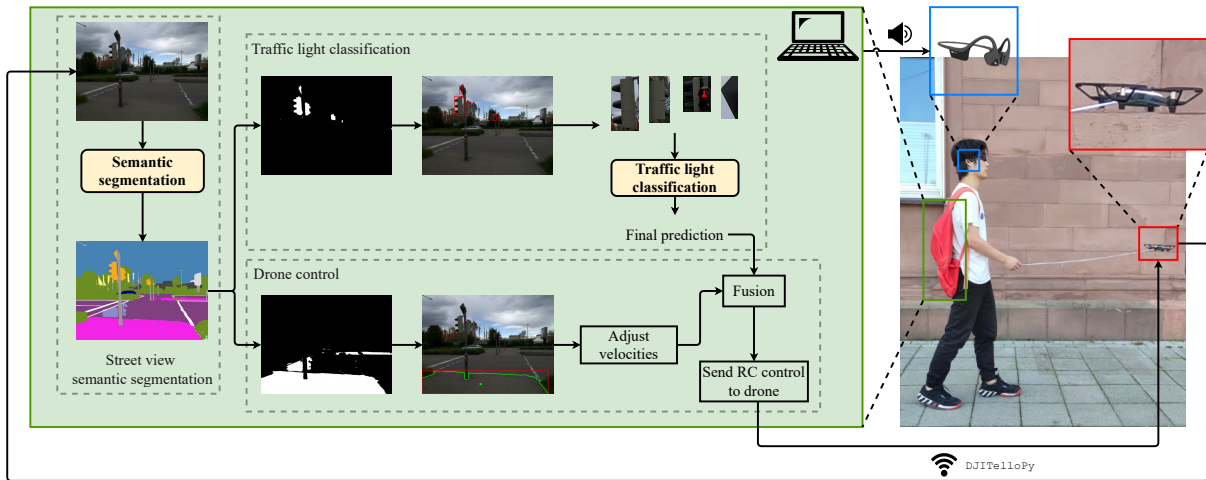
Fig. 2: System overview. There are three components: street view semantic segmentation, traffic light classification, and drone control.

II. RELATED WORK

Assistive systems. A variety of assistive devices dedicated for BVIP have been developed. A skeleton-pose-based framework was built upon the drone for Human-Drone Interaction [5]. A guide-dog robot system was designed based on template maching for assisting the BVIP in self-walking [6]. In order to help BVIP cross the street, Cheng *et al.* [7] proposed a real-time pedestrian crossing light detection algorithm for the BVIP using histogram of oriented gradients (HOG) and support vector machine (SVM) and integrated it into a wearable navigation system. Yang *et al.* [8] unified intersection-centered perception tasks by utilizing real-time semantic segmentation. While there were robotic guide dogs [9], [10] and drone-based assistive systems [4], [11] for BVIP, we pioneer to realize path discovery and intersection navigation with semantic segmentation on drone videos.

Semantic segmentation. As a seminal work, FCN [12] performs dense pixel-to-pixel classification in an end-to-end manner. Subsequent networks follow the per-pixel recognition paradigm and significantly improve scene segmentation performance. Introducing the global pyramid pooling feature, PSPNet [13] embeds multi-scale scenery context features in an FCN-based pixel prediction framework. Recently, non-local attention layers and Multilayer Perceptrons (MLPs) are frequently used for dense prediction tasks like image semantic segmentation [14], [15]. In contrast to FCN-based models, SegFormer [16] combines a hierarchical Transformer [17] encoder and a lightweight All-MLP decoder. It yields a powerful representation without complex and computationally demanding modules. Compared to existing systems that use FCN-based semantic segmentation for assisting BVIP, we build our flying guide dog system with SegFormer, which can yield an efficient and robust segmentation.

III. APPROACH

Our goal is to implement a "flying guide dog" prototype, which is able to automatically discover walkable areas and avoid obstacles so that it can guide the BVIP person to walk safely. The drone is connected to the computer using

`DJITelloPy` [1] library via Wifi. For each frame captured by the drone's camera, the semantic segmentor outputs a colorized prediction. To make the drone fly along the walkable path, the largest walkable area is extracted and its centroid is then estimated. On the basis of centroid estimation, velocity adjustment is computed. Meanwhile, if traffic lights are detected, we crop them out and input them into the traffic light classification model. Fusing the classification prediction and velocity adjustment, a Radio Control (RC) command is sent to the drone. Additionally, the user gets voice prompts via a Bluetooth-connected bone conduction headphone.

As illustrated in Fig. 2, our system is composed of three main modules: (a) street view semantic segmentation (Sec. III-A), (b) traffic light classification (Sec. III-B), and (c) drone control (Sec. III-C).

A. Semantic segmentation

Robustness and effeciency. A vital requirement for the segmentation model is real-time performance. According to the comparison of real-time semantic segmentation models, SegFormer-B0 [16] has significant advantages on speed and accuracy. Futhermore, it is more robust to common corruptions and perturbations.

Datasets. While Cityscapes [18] are recorded in a unified setting, Mapillary Vistas [19] are globally taken by diverse devices from different viewpoints. Moreover, comprising 25,000 densely annotated street level images into 66 categories, Mapillary Vistas is $5\times$ larger than Cityscapes in terms of fine-grained annotations. Therefore, Mapillary Vistas is more diverse and promising for yielding robust models, thus suitable for our prototype.

Implementation details. We train SegFormer-B0 [16] on Mapillary Vistas [19] using `mmsegmentation` [2] codebase on a single GTX 1080 Ti GPU. During training, we apply data augmentation techniques, including random resizing with ratio 0.5-2.0, random horizontal flipping, and random cropping to 768×768. We train the model for 160K iterations with AdamW optimizer. Initial learning rate is set to $6E-5$

---

[1] https://github.com/damiafuentes/DJITelloPy
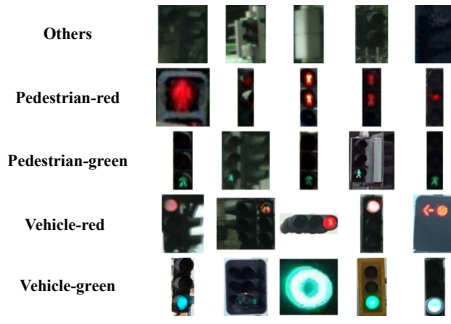[2] https://github.com/open-mmlab/mmsegmentation

Fig. 3: PVTL dataset with 5 categories. The "others" refers to side or back of traffic lights and unilluminated traffic lights.

and a polynomial learning rate decay scheduler with factor 1.0 is employed, following the training strategy in [16].

B. Traffic light classification

After semantic segmentation, a prediction mask regarding traffic lights can be obtained. Based on the mask, the traffic light patch can be cropped from its full-resolution image. Performing image classification in a cropped patch is more computationally efficient than in a full-scale image. To simplify and unify the recognition of the color and category of traffic lights, we train a light CNN, which is expected to be more accurate and efficient. More importantly, it should eliminate the influence of vehicle traffic lights. Yet, currently there is not a dataset directly suitable for our task.

PVTL dataset. To facilitate traffic light classification, we introduce a new dataset called *Pedestrian and Vehicle Traffic Lights (PVTL)* with the goal of distinguishing pedestrian traffic lights from vehicle traffic lights. We perform four steps to collect this dataset. Firstly, we crop the traffic lights from Cityscapes [18], Mapillary Vistas [19], and Pedestrian lights [20] based on their annotations. Secondly, we clean up those image patches with resolution smaller than 8×8. Then, we manually annotate the images into 5 categories: {*pedestrian-red, pedestrian-green, vehicle-red, vehicle green, others*}. Last but not least, a class balancing is performed to maintain 300 images for each class. We visualize some representative examples in Fig. 3. The category *others* refers to the side and back of the traffic lights or traffic lights that are not illuminated. All data and annotations will be made publicly available.

Pedestrian light prediction. As shown in Fig. 4, the input of the model is the traffic light patch cropped from every single frame when crossing traffic light intersections. In order to eliminate the influence of vehicle traffic lights, we keep only the pedestrian light prediction and discard others. Each prediction will be appended to the buffer list, whose length is 7 frames. Based on the accumulated predictions, the most frequent class is output as the final traffic light prediction, which can robustify the system and ensure the safety.

C. Drone control

After achieving semantic segmentation and traffic light classification, four types of velocity are calculated to control the drone, as depicted in Fig. 5. Among them, the up/down velocity v_{ud} is obtained by the Tello's vision positioning
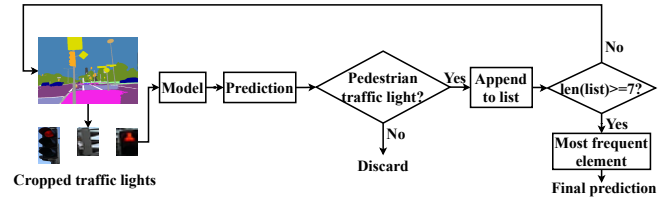

Fig. 4: Multi-frame based traffic light classification.

system to maintain the flying height h_{target}, which is preset as $1.2m$ to ease the user interaction. Finally, RC commands are sent to the drone using `DJITelloPy`. A detailed description of the control strategy is presented in Algorithm 1.

Fly along walkable path. Based on the mask of the walkable path (*e.g. sidewalks* and *crosswalks*), the largest walkable area $L_{walkable}$ is cropped from the colorized prediction C. Through the image moment of $L_{walkable}$, the centroid of the walkable area is computed. Smoothing the position of the contour centroid with high-pass and low-pass filters makes the movement more stable. Horizontal difference between the filtered centroid and the center of the whole image is used as the reference of the left/right velocity v_{lr}. The horizontal adjustment is important to prevent most of collisions and assist BVIP to walk safely. After adjusting the left/right velocity, the drone can constantly fly in the middle of the walkable path.

In order to fly towards the direction of a walkable path, the yaw velocity v_{yaw} is required to veer. Motivated by [21], the image with the largest walkable area is further divided into three partitions horizontally. Then the confidence $conf$ for each part, the mean of the pixel values, is compared with a pre-adjusted threshold θ_{conf} to obtain the final binary control code $list_{codes}$. Each control code is assigned to separate yaw speed v_{yaw} in $list_{yaws}$.

Fusion with traffic light classification. When encountering a traffic light during walking, the drone control with a preset forward velocity $v_{f,0}$ will be updated and integrated with the traffic light recognition. If the pedestrian traffic light turns green, the drone accelerates forward. If it is red, the forward velocity is set to 0 and the drone will hover in place until the light turns green. If the light changes from green to red during crossing, the drone will maintain the adjusted velocity.

Voice feedback. At each step of traversing the pedestrian traffic lights, voice prompts (*i.e.* speech) will be sent to the user through the bone conduction headset, such as "stop" at the red light and "go" at the green light. Using voice

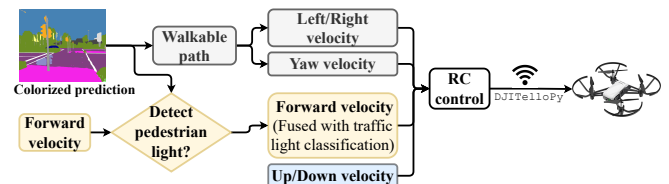Fig. 5: Drone control overview. Four types of velocity are updated based on the walkable path and the traffic light prediction.

Algorithm 1: Drone control

---

input : colorized prediction $C \in \mathbb{R}^{H \times W \times 3}$, pedestrian crossing light $color \in \{red, green, None\}$, current altitude h

output: up/down velocity v_{ud}, yaw velocity v_{yaw}, left/right velocity v_{lr}, forward velocity v_f

---

1 Initialize parameters: pixel threshold θ_{conf}, target altitude h_{target}, speed up $speedup$, up/down velocity v_{ud}, forward velocity $v_{f,0}$; lists: preset yaw velocities $list_{yaws}$, binary control code $list_{codes}$;

2 $start\_crossing \leftarrow$ false;

3 **while** *drone is flying and video stream is on* **do**

    // Maintain target altitude

4   **if** $h \neq h_{target}$ **then**

5 | $v_{ud} \leftarrow (h_{target} - h)/h_{target} * v_{ud}$;

6   **end**

    // Fly along walkable path

7   Extract largest walkable area $L_{walkable}$ from C;

8   $(x_{centroid}, y_{centroid}) \leftarrow$ estimate_centroid$(L_{walkable})$;

9   $v_{lr} \leftarrow x_{centroid} - \frac{W}{2}$;

10   $(R_l, R_m, R_r) \leftarrow$ partition$(L_{walkable})$;

11   **for** $p \in (R_l, R_m, R_r)$ **do**

12 | $conf \leftarrow$ mean(p);

13 | $code \leftarrow$ binary_conversion$(conf, \theta_{conf})$;

14 | Append $code$ to $list_{codes}$;

15   **end**

16   $v_{yaw} \leftarrow$ get_yaw_vel$(list_{yaws}, list_{codes})$;

    // Fusion with traffic light classification

17   **if** $color = None$ **then**

18 | $v_f \leftarrow v_{f,0}$;

19   **else if** $color = green$ or $start\_crossing$ **then**

20 | $v_f \leftarrow v_{f,0} + speedup$;

21 | $start\_crossing \leftarrow$ true;

22   **else**

23 | $v_f \leftarrow 0$;

24   **end**

25 **end**

---

feedback can always keep a certain safe distance between the user and the drone.

IV. EXPERIMENTS

A. Semantic segmentation results

Overall result. To investigate the training procedure of SegFormer-B0 [16] on Mapillary Vistas [19], we present the logging of training loss in Fig. 6a and the one of accuracy in Fig. 6b, which are measured in every 50 iterations. In the first few hundred iterations, the loss drops dramatically. After 25,000 iterations, the loss and accuracy start to converge and remain relatively steady. After training for 160K iterations, SegFormer-B0 yields 41.83% mean Intersection over Union (mIoU) on validation set of Mapillary Vistas, with the shorter side of input image being 1024.

Per-class result. Intersection over Unions (IoUs) of some categories of interest for BVIP and related to our task are reported in Table I. SegFormer-B0 [16] achieves compelling results on categories that are significant for walkable area discovery, such as *Road* (85.20%), *Sidewalk* (62.32%), and *Land Marking - Crosswalk* (63.80%), which guarantee the accuracy and effectivness of our system. Moreover, the
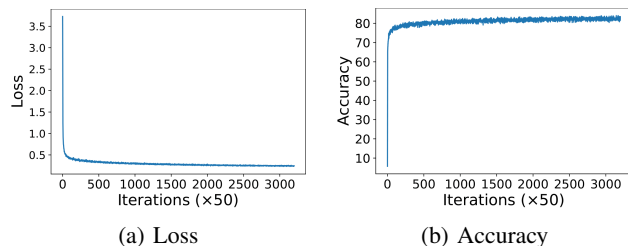
(a) Loss   (b) Accuracy

Fig. 6: Training Loss (left) and accuracy (right) of SegFormer-B0 [16] on Mapillary Vistas [19].

TABLE I: Per-class IoU of SegFormer-B0 [16] on Mapillary Vistas [19] validation set in 1024×2048 resolution.

Category	IoU
Bike Lane	31.87
Crosswalk - Plain	21.09
Curb	53.20
Pedestrian Area	31.24
Road	85.20
Sidewalk	62.32
Lane Marking - Crosswalk	63.80
Traffic Light	61.06
Traffic Sign (Front)	68.67
Car	86.84

model obtains 61.06% accuracy of the *Traffic Light*, thus it can generate a sufficient patch for the downstream traffic light classification.

Comparison to state-of-the-art methods. Based on the evaluation on Mapillary Vistas [19], Table II summarizes results including frame per second (FPS), mIoU, and parameters. Compared to other real-time approaches (*e.g.* RGPNet [22]), SegFormer-B0 [16] achieves comparable accuracy with significantly fewer parameters, which shows great potential for deployment on edge devices and mobile devices. Note that to propose a novel network is not our scope in this work, instead to introduce the feasibility of deploying a Transformer-based semantic segmentation model on the assistive system. Besides, our system can be constructed by arbitrary efficient backbones.

TABLE II: Comparison of state-of-the-art methods on Mapillary Vistas [19] validation set. Inference speed (FPS) is calculated on 1024×2048 image resolution.

	Method	FPS ↑	mIoU (%) ↑	Params (M) ↓
Non Real-Time	BiSeNet (R101) [23]	9.3	20.4	50.1
	TASCNet (R50) [24]	**11.9**	46.4	**32.8**
	TASCNet (R101) [24]	8.8	48.8	51.8
	ShelfNet (R101) [25]	9.1	49.2	57.7
	RGPNet (R101) [22]	10.8	50.2	52.2
	RGPNetB(WRN38) [22]	3.4	**53.1**	215.0
Real-Time	RGPNet (HarDNet39D) [22]	34.7	**42.5**	9.4
	RGPNet (R18) [22]	**35.7**	41.7	17.8
	SegFormer-B0 (ours)	15.2	41.8	**3.8**

B. Qualitative comparison

Fig. 7 depicts the qualitative results. For images taken from the pedestrian's point-of-view (POV), models trained on Cityscapes is almost unable to segment the sidewalk from the road. This result may be explained by the fact that images of Cityscapes [18] are taken by a camera mounted behind the vehicle's windshield, *i.e.*, from the vehicle's POV. Accordingly, the models lack the ability of generalization to segment images from the viewpoints of the pedestrian.
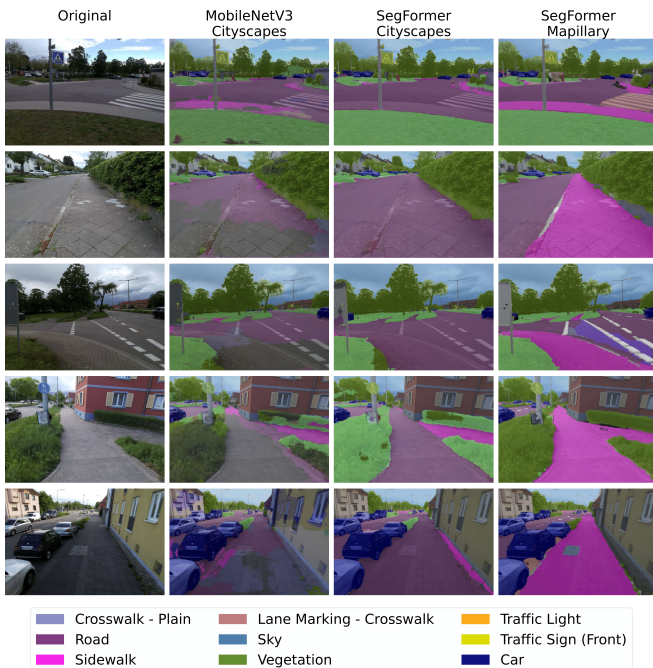
Fig. 7: Qualitative results of predictions of different models trained on different datasets. Compared to models trained on Cityscapes [18], SegFormer-B0 [16] trained on Mapillary Vistas [19] segments sidewalk and crosswalk much better.

In contrast, SegFormer-B0 trained on Mapillary Vistas [19] outputs a clearly better segmentation between sidewalks, crosswalks, and roads, thanks to diverse viewing perspectives in the training samples.

C. Traffic light classification results

As shown in Table III, we first construct our light CNN model which contains 5 convolutional layers and 3 fully connected layers. In addition, we fine-tune ResNet-18 [26] as our alternative. Both models are trained on PVTL dataset for 25 epochs. Compared with ResNet-18, our light CNN model is slightly less accurate but more lightweight and computationally efficient.

TABLE III: Comparison of traffic light classification models.

Models	ResNet-18 [26]	Light CNN
Weight Initialization	Pretrained on ImageNet	Default
Accuracy ↑	**90%**	83%
FPS ↑	167	**670**
#Params (M) ↓	11.6	**2.4**
Flops (GFLOPs) ↓	7.2	**0.35**

D. Runtime analysis

The drone we adopt for runtime measurement is DJI Tello. Equipped with a RGB camera, it supports 720p HD transmission. To handle image processing, model inference, and further drone control, we utilize a laptop with an Intel i7 CPU and an Nvidia GeForce GTX 1050 Ti GPU. After warming up for 200 frames, we measure and average the runtime of the subsequent 800 frames. As summarised in Table IV, the mean processing time of a single frame at the resolution of 320×240 is 85ms, mostly on semantic segmentation (71ms). In other words, the speed of our prototype is 11.7 FPS.

TABLE IV: Running time analysis. "Others" refers to the time that costs for frame transmission, velocity adjustment, and sending RC command.

Models	SegFormer [16] + ResNet-18 [26]	SegFormer [16] + Light CNN
Segmentation (ms) ↓	**70**	71
Classification (ms) ↓	18	**14**
Others (ms) ↓	1	1
All (ms) ↓	89	**85**
FPS ↑	11.274	**11.713**

E. User study

To evaluate the assistance functions of our system, an user study in real world scenarios is conducted using NASA Task Load Index (NASA-TLX) method [27].

User study setup. During the user study, we use the same hardware setup as in Section IV-D. The laptop is placed in a backpack, as shown in Fig. 8a. Six participants aged between 24 and 35, including 5 males and 1 female, are sighted but blindfolded during the test to simulate BVIP. The drone flies in front of the participants to simulate the real guide dog and to get an unobstructed view of the environment. The participant follows the drone by feeling the traction from a string attached to the drone.

Our prototype is evaluated in two open-space scenarios, as exhibited in Fig. 8. The first scenario focuses on walking along the sidewalk in 20 meters length. Some obstacles, such as *bicycles* and *pedestrians*, were randomly appeared in the sidewalk to test the obstacle avoiding function. The second scenario, an intersection including vehcile and pedestrian traffic lights, is selected to test the street crossing functionality according to the pedestrian light prediction. After testing, participants fill multi-aspected questionnaires based on the six categories listed in Table V. They score each aspect from 1 to 10, among which 1 and 10 mean *strongly disagree* and *strongly agree*, respectively.

Evaluation. A radar graph assessing multiple aspects is shown in Fig. 9. For features of walking along the walkable path, *Orientation* and *Position*, participants find it reliable for guiding, citing that they can even avoid obstacles using this system. In terms of *Physically easy to use* and *Learnability*, they also make positive comments. After a brief introduction the participants are able to use the system alone. However,

(a) Sidewalk   (b) Street crossing

Fig. 8: User study consists of two scenarios: (a) sidewalk (b) street crossing. To simulate visually inpaired people, the participant wears a blindfold during the user study.

TABLE V: Six aspects of the user study questionnaire.

| Aspects	Statements
Orientation	I can walk in the proper direction.
Position	I can walk in the middle of the path.
Traffic Light	I can cross the road at the right time according to the pedestrian lights.
Learnability	I can get familiar with this system easily.
Mentally easy to use	I don't need much time to think to follow the drone.
Physically easy to use	I don't need to put in a lot of physical effort when using the system.
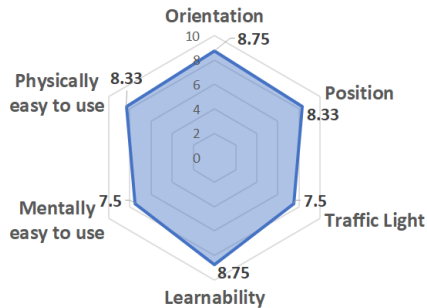
Fig. 9: Evaluation results of the user study. Average scores are given by participants

the *Traffic light* feature still needs to be improved. It could be harder to cross the street with a series of pedestrian lights for wrongly guiding users to the pedestrian zone in the middle. To tackle this issue, the speed of drone can be adjusted during crossing via the aforementioned drone control algorithm. The participants also rate the feature *Mentally easy to use* lower than others, as sometimes it is insufficient to feel the traction from the connected string. This issue is subsequently improved with additional voice feedback using a wireless bone conduction headphone, which aims to alert users the walking direction and the color of the pedestrian light. Despite a limited number of participants, most of them comment that the function of our prototype is indeed very helpful for people with visual impairment, which proves the usability and effectiveness of our flying guide dog prototype.

V. CONCLUSION

In this work, we propose a novel "flying guide dog" prototype. Combining street view semantic segmentation, traffic light classification, and drone control algorithm, our prototype is capable to automatically discover the walkable path, avoid obstacles, and thus guide the user walk safely. Moreover, to better distinguish pedestrian traffic lights from other types of traffic lights, we set up a new dataset called *PVTL*. Results of the user study suggest that our approach is effective for BVIP assistance.

Nonetheless, the prototype is limited by the drone's battery capacity, which can only support maximal 13 minutes flight time. Another major limitation is that the drone is too light to resist the wind. A reasonable approach to tackle these issues could be to use a more powerful drone with larger battery capacity.

REFERENCES

[1] R. R. A. Bourne *et al.*, "Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis," *The Lancet Global Health*, 2017.

[2] A. Fiannaca, I. Apostolopoulos, and E. Folmer, "Headlock: A wearable navigation aid that help blind cane user traverse large open spaces," *ASSETS*, 2014.

[3] H. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarré, and D. Rus, "Enabling independent navigation for visually impaired people through a wearable vision-based feedback system," in *ICRA*, 2017.

[4] M. Avila Soto, M. Funk, M. Hoppe, R. Boldt, K. Wolf, and N. Henze, "DroneNavigator: Using leashed and free-floating quadcopters to navigate visually impaired travelers," in *ASSETS*, 2017.

[5] Z. Marinov, S. Vasileva, Q. Wang, C. Seibold, J. Zhang, and R. Stiefel-hagen, "Pose2drone: A skeleton-pose-based framework for human-drone interaction," *arXiv*, 2021.

[6] Y. Wei and M. Lee, "A guide-dog robot system research for the visually impaired," in *ICIT*, 2014.

[7] R. Cheng, K. Wang, K. Yang, N. Long, J. Bai, and D. Liu, "Real-time pedestrian crossing lights detection algorithm for the visually impaired," *Multimedia Tools and Applications*, 2018.

[8] K. Yang, R. Cheng, L. M. Bergasa, E. Romera, K. Wang, and N. Long, "Intersection perception through real-time semantic segmentation to assist navigation of visually impaired pedestrians," in *ROBIO*, 2018.

[9] T.-K. Chuang *et al.*, "Deep trail-following robotic guide dog in pedestrian environments for people who are blind and visually impaired-learning from virtual and real worlds," in *ICRA*, 2018.

[10] A. Xiao, W. Tong, L. Yang, J. Zeng, Z. Li, and K. Sreenath, "Robotic guide dog: Leading a human with leash-guided hybrid physical interaction," *arXiv*, 2021.

[11] F. Huppert, G. Hoelzl, and M. Kranz, "GuideCopter-A precise drone-based haptic guidance interface for blind or visually impaired people," in *CHI*, 2021.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017.

[14] S. Chen, E. Xie, C. Ge, D. Liang, and P. Luo, "CycleMLP: A MLP-like architecture for dense prediction," *arXiv*, 2021.

[15] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen, "Trans4Trans: Efficient transformer for transparent object segmentation to help visually impaired people navigate in the real world," in *ICCVW*, 2021.

[16] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," *arxiv*, 2021.

[17] A. Vaswani *et al.*, "Attention is all you need," in *NeurIPS*, 2017.

[18] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[19] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *ICCV*, 2017.

[20] J. Roters, X. Jiang, and K. Rothaus, "Recognition of traffic lights in live video streams on mobile devices," *TCSVT*, 2011.

[21] I.-H. Hsieh, H.-C. Cheng, H.-H. Ke, H.-C. Chen, and W.-J. Wang, "Outdoor walking guide for the visually-impaired people based on semantic segmentation and depth map," in *ICPAI*, 2020.

[22] E. Arani, S. Marzban, A. Pata, and B. Zonooz, "RGPNet: A real-time general purpose semantic segmentation," in *WACV*, 2021.

[23] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *ECCV*, 2018.

[24] J. Li, A. Raventos, A. Bhargava, T. Tagawa, and A. Gaidon, "Learning to fuse things and stuff," *arXiv*, 2018.

[25] J. Zhuang, J. Yang, L. Gu, and N. Dvornek, "ShelfNet for fast semantic segmentation," in *ICCVW*, 2019.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[27] S. G. Hart, "NASA-task load index (NASA-TLX); 20 years later," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2006.