Object Recognition to Support Navigation Systems for Blind in Uncontrolled Environments

Marlon Marcon
marlonmarcon@utfpr.edu.br
Federal University of Technology – Parana
Dois Vizinhos, Paraná, Brazil

André Roberto Ortoncelli
ortoncelli@utfpr.edu.br
Federal University of Technology – Parana
Dois Vizinhos, Paraná, Brazil

ABSTRACT

Efficient navigation is a challenge for visually impaired people. Several technologies combine sensors, cameras, or feedback channels to increase the autonomy and mobility of visually impaired people. Still, many existing systems are expensive and complex to a blind person's needs. This work presents a dataset for indoor navigation purposes with annotated ground-truth representing real-world situations. We also performed a study on the efficiency of deep-learning-based approaches on such dataset. These results represent initial efforts to develop a real-time navigation system for visually impaired people in uncontrolled indoor environments. We analyzed the use of video-based object recognition algorithms for the automatic detection of five groups of objects: i) fire extinguisher; ii) emergency sign; iii) attention sign; iv) internal sign, and v) other. We produced an experimental database with 20 minutes and 6 seconds of videos recorded by a person walking through the corridors of the largest building on campus. In addition to the testing database, other contributions of this work are the study on the efficiency of five state-of-the-art deep-learning-based models (YOLO-v3, YOLO-v3 tiny, YOLO-v4, YOLO-v4 tiny, and YOLO-v4 scaled), achieving results above 82% performance in uncontrolled environments, reaching up to 93% with YOLO-v4. It was possible to process between 62 and 371 Frames Per Second (FPS) concerning the speed, being the YOLO-v4 tiny architecture, the fastest one. Code and dataset available at: https://github.com/ICDI/navigation4blind.

KEYWORDS

Computer Vision, assistive technology, object detection, YOLO

1 INTRODUCTION

The World Health Organization (WHO) estimates over 285 million blind and visually impaired people globally, whose 39 million are blind [1]. In Brazil, around 0.75% of the population is blind [2]. Visual impairment can seriously impact people's quality of life, as they encounter many challenges in most daily activities [3]. One of the biggest challenges faced by such people is associated with secure and efficient navigation [4], such as obstacles, stairs, traffic corners, signposts on the pavement, and slippery paths [5, 6].

Over the past few years, several Computer-vision-based technologies have been proposed to increase the autonomy and mobility of visually impaired people [7]. Such proposals combine different types of sensors, cameras, or feedback channels [8] and process 2D [9] and 3D [10] data. Despite the availability of such systems, they still suffer from dynamic interactions and adaptability to changes from the internal to the external environment. There are few easy-to-use navigation aids, but these devices are expensive and inaccessible for most patients who need them [11], tending to be complex to a blind person's needs [5, 12].

This work presents the results of the initial efforts to develop a system based on computer vision to support the navigation of blind and visually impaired people in an indoor environment, in our case, some corridors of a building on a university campus.

The building that is the focus of the case study has an embossed tactile floor for the visually impaired. In this context, the first planned functionality for a navigation system is to recognize and provide feedback on boards and objects, divided into five classes. In Figure 1 we present samples of each adopted class, and in Table 1 we describe in details each of them.

Figure 1: Examples of the labeled classes: a) fire extinguisher; b) emergency sign; c) attention sign; d) internal sign; and e) other

We conducted the case study with a database with 20 minutes and 6 seconds of videos recorded by a camera positioned at the chest height of the person as they walked through all the building corridors. In these videos, we manually labeled each occurrence of objects associated with defined classes (Figure 1).

After labeling the database, experiments were conducted with different versions of the YOLO (You Only Look Once) [9, 13], which is one of the most used one-stage video-based object recognition algorithms. In the case study, we explore five YOLO's versions: i) YOLO-v3; ii) YOLO-v3 tiny; iii) YOLO-v4; iv) YOLO-v4 tiny; and v) YOLO-v4 scaled.

In summary, the main contributions of the paper include:

- A dataset for indoor navigation, with annotated ground truth bounding boxes of objects that represent common interactive signs in a real-world situation;

Table 1: Description of labeled classes

ID	Name	Description
A	Fire extinguisher	This type of object was selected for two reasons: i) it can be important in emergency situations; and ii) blind people may collide with them in the hallways while moving.
B	Emergency sign	Signs with recommendations that must be followed in emergency/evacuation situations.
C	Attention sign	Signs that indicate equipment/recommendations that should be used in emergency situations, such as signs that indicate the position of fire extinguishers.
D	Internal sign	Internal signs that identify the university rooms names.
E	Other	Boards/papers attached to the walls that do not fit into any of the previous categories.

- A study on the efficiency of state-of-the-art deep-learning-based models applied to object detection in indoor environments;
- An effective indoor object detection model, achieving over 93% performance on average in uncontrolled environments.

To present these contributions, the remainder of this work is organized as follows. In Section 2 we present the literature review; Details about the methodology are in Section 3; Results and discussions are in Section 4; Finally, Section 5 concludes the paper.

2 LITERATURE REVIEW

Several approaches have been proposed to improve social integration and support the development of daily tasks for blind and visually impaired people. Among them, we mention studies related to automated Braille text transcription [14], systems accessibility [15], approaches to notify blind people about obstacles, and navigation systems [5, 16].

This Section presents a literature review related to blind and visually impaired navigation support. Subsection 2.1 details state-of-the-art methods for different categories of this type of system. Subsection 2.2 presents state of the art related to Video-based Object Recognition, describing relevant techniques for navigation systems development.

2.1 Blind Navigation System

Kuriakose *et al* [5] categorized the tools and technologies for blind and visually impaired navigation support into five groups. Some methods can be in more than one group. Subsections 2.1.1 to 2.1.5 detail each of them and present examples of their approaches.

2.1.1 Visual Imagery Systems: It uses computer vision algorithms and optical sensors to detect obstacles and then guide the user to navigate safely by giving directions to avoid them. Recent methods fall under this group of systems [11, 17].

In [17] obstacles are detected with an RGB-D camera. Route planning is dynamically adapted to improve navigation safety based on this detection. They recalculate the path using previously modeled geometric information from the environment.

A navigation system for the blind was proposed in [11]. It is a real-time system that descriptively monitors the environment, providing audio that describes the objects and their location to the user. They perform object detection and classification using the trained Single Shot Detector (SSD) MobileNet v2 Convolutional Network model installed on the Raspberry Pi 3 Model B+.

2.1.2 Non-visual Data Systems: Non-visual data systems do not use vision algorithms or optical sensors as a primary resource. Methods in this group commonly use several kinds of sensors. An example of a method in this category, proposed by [18] uses the Internet of Things (IoT) to detect objects with wireless sensors. The system informs the user of its name and distance for each detected object by voice feedback.

2.1.3 Map-based Systems: Such kind of system consists in applying multimodal tactile maps to assist the navigation of blind and visually impaired people. These resources are an efficient way for spatial learning—methods in this class present the difficulty of updating maps' contents.

The approach of [19] falls into this category. The authors produced a map designed with a participatory design approach. The map used uses augmented reality, combining projection, audio output, and tactile tokens.

2.1.4 Systems with 3D Sound: In this group, it is possible to point out [20], which uses a wearable sensor that gives users tactile and audio feedback to provide an auditory and tactile representation of the surrounding environment.

It can also identify [21], which features a molded helmet with stereo cameras and headphones. This system emits specific musical sounds that correspond to information about characteristics of the obstacle that the user faces. A drawback of such a group needs previous train on the system.

2.1.5 Smartphone-based Solutions: This group involves solutions that users use on smartphones, offering portability and convenience for users. We can highlight recent works for this category [22, 23].

In [22] the authors propose a system called LineChaser to help blind people walk in lines in public spaces. The system uses an RGB-D camera to guide a blind user to the end of the line and continuously reports the distance and direction to the last person on the line, which must follow.

The method proposed in [23] the authors proposed the NavCog3 system, which is a navigation assistant that uses Bluetooth beacons installed in the environment and a user's smartphone. The system guides you based on nearby points of interest (e.g., entrances, stores) identified by wireless. This system can be scaled to large environments but requires the prior installation of the necessary equipment.

2.2 Video-based Object Recognition

The object recognition task demands determining the location and category of objects in an image. This research line has recently received attention because of its relationship to video analysis and image comprehension.

The development of Deep Learning Methods enabled the development of powerful tools, which have contributed to improving object recognition results concerning the traditional methods [24]. Deep Learning for object recognition can be divided into two categories [25]:

- Two-stage algorithms: as the name suggests, they perform two steps. In the first step, we identify possible target regions, and the second one completes the classification. This method has high accuracy but also limits the detection speed. We can highlight Convolutional Neural Network (CNN) based architectures in this group, we can highlight R-CNN, Fast R-CNN, and Faster R-CNN [26].
- One-stage algorithms: in a single step, use only one network to predict object classes and bounding boxes. This class of algorithms improves the detection speed, but the accuracy for small target detection is not as good as the two-stage algorithm. Architectures in this group, we can highlight RetinaNet [27], YOLO [9, 13], and SSD [28].

In this paper, we compare the results of one-stage video-based Object recognition algorithms. We selected versions of the YOLO architecture for the experiments, which is one of the most commonly explored.

YOLO [9] transforms the target detection into a regression problem. The whole framework only uses a relatively simple CNN structure to predict the bounding box's position and the candidate box's class. The third version of YOLO (YOLO-v3), proposed by [29] in 2018, uses the variant of Darknet composed of 53 layer network trained on ImageNet. For the detection task, 53 more layers are stacked onto it, totaling a set of 106 fully convolutional underlying layers.

In 2020 the fourth version of YOLO (YOLO-v4) was proposed [13], which adopts YOLO-v3 as a one-stage dense prediction in the head. Recently the fifth version of YOLO has been presented, and it has been explored by several works [30, 31], but there is still no paper about the YOLO-v5. By modifying the depth, width, resolution, and structure of the YOLO-v4, the YOLO-v4 scaled [32] improves the results and represents the state-of-the-art o such kind of architecture.

3 METHODOLOGY

This work presents the results of the initial efforts to develop a navigation system based on Computer Vision for blind and visually impaired people in a closed environment.

We conducted a case study to assess the accuracy of state-of-the-art Deep Learning algorithms for video-based object recognition in the classification of objects present in the main building of a university campus (Table 1).

This Section presents the case study methodology. The experimental database is described in Subsections 3.1 and 3.2, which present the process of video recording and object labeling , respectively. Subsection 3.3 defines the Deep Learning algorithms configurations and also the training process. Subsection 3.4 presents the metrics used to evaluate the experimental results.

3.1 Video recording

We produced a videos' database collected in the largest building on a university campus, which has four floors with around a thousand square meters each. Every building floor has a hallway that runs through it thoroughly. In the middle of each floor, there is a central ladder. We recorded videos walking in the halls on both sides of the building on each floor.

The videos were collected in two days with a difference of approximately three months to verify the algorithms results under different conditions. In this way, some objects were removed and added to the building. We captured videos at different times, changing the sunlight incidence.

Considering that each building floor has two sides recorded twice on different days, we captured 16 videos. The average time is about 75 seconds, totaling more than 20 minutes of recording time. Table 2 presents a list of information about each scene, including time, the recording day, hour, and split we used in our trials.

All videos were collected with a Go Pro Hero 7 camera, positioned at the chest height of a person who walked through the building hallway, making recordings with this camera facing him. When a key object is found, the person turns the camera towards it, turns it back to the front, and follows their path to the end of the hallway. Figure 2 shows frames of collected videos and also examples of labeled objects.

Table 2: Recording videos description

ID	Floor	Side	Day	Hour	Duration (seconds)	Split
1	1st	A		14:04	72	Train
2	1st	B		14:08	74	Test
3	2nd	A		14:13	84	Train
4	2nd	B	08/09/2021	14:16	57	Train
5	3rd	A		14:21	84	Validation
6	3rd	B		14:25	52	Train
7	4th	A		14:30	114	Train
8	4th	B		14:33	61	Test
9	1st	A		18:02	92	Train
10	1st	B		18:06	55	Validation
11	2nd	A		18:11	64	Test
12	2nd	B	06/12/2021	18:13	42	Train
13	3rd	A		13:18	71	Test
14	3rd	B		13:22	38	Train
15	4th	A		13:26	114	Train
16	4th	B		13:30	103	Train

3.2 Object Labelling

We recorded the videos at a rate of 30 frames per second (FPS). We selected only the first frame of each second of the video to compose the experimental database. For each frame, we labeled the objects of interest (Table 1) with a bounding box around it. In Figure 2 we present some examples of the annotations.

Figure 2: Samples of the labeling process.

We divided the videos into two sets, and then the authors did the labeling process. Each person labeled one of the videos sets and validated the labels assigned by the other. We show the number of objects tagged for each class and their distribution in the training, validation, and testing sets in Table 3.

Table 3: Number of objects labeled for each class

ID	Name	Train	Test	Validation	Total
A	Fire extinguisher	159	73	34	266
B	Emergency sign	232	115	55	402
C	Attention sign	265	110	53	428
D	Internal sign	388	145	59	592
E	Other	492	93	22	607
Total		1536	536	223	**2295**

3.3 Trained Models

To train, validate, and test the object detection algorithms, we split our set of 16 videos (described in Subsection 3.1) into three subsets:

- **Training set**: the images of this set were used as examples to train the models applied in the experiments;
- **Validation set**: this set provides an unbiased assessment of a model trained with the train set while adjusting its hyper-parameters. We adopt the early stop strategy to select the best model for the training process, i.e., the model with the highest value of AP on the validation set was selected; and
- **Test set**: this set was used to evaluate the results of each model trained with the evaluation metrics - presented in the Subsection 3.4.

We trained five different YOLO models and used them in the experiments:

- **YOLO-v3**: The third version of YOLO proposed by [29] in 2018;
- **YOLO-v3 tiny**: This version was proposed by Joseph Redmon [29], consisting of decreasing the depth of the convolutional layer of the YOLO-v3. This simplification makes the YOLO-v3 suitable for real-time applications;
- **YOLO-v4**: The fourth version of YOLO proposed by [13] in 2020;
- **YOLO-v4 tiny**: like YOLO-v3 tiny, this version simplifies YOLO-v4 making it suitable for real-time application [33]; and
- **YOLO-v4 scaled**: modifies the depth, width, resolution, and structure of the YOLO-v4. In a recent study with the MS

COCO dataset, the YOLO-v4 scaled showed better results in relation to the YOLO-v3, YOLO-v4, and others network architectures [32];

3.4 Evaluation Metrics

We compared the YOLO architectures by using six metrics, cited as follow:

- **Precision**: the percentage of correctly detected objects. Considering the total of True Positives (TP) and False Positives (FP), we compute the Precision with Equation 1.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

- **Recall**: the fraction of relevant instances retrieved. Considering the total of TP and False Positives (FP), we compute the Recall with Equation 2.

$$recall = \frac{TP}{TP + FN} \tag{2}$$

- **F1-score**: is a harmonic mean between Precision and Recall. We calculate the F1-score metric with Equation 3. ,

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \tag{3}$$

- **Intersection over union (IoU)**: defines how accurate a predicted bounding box was in comparison with the ground truth. The IoU is computed with Equation 3.

$$IoU = \frac{Area\ of\ overlap\ between\ bounding\ boxes}{Area\ of\ union\ between\ bounding\ boxes} \tag{4}$$

The IoU value can vary between 1 and 0, and the higher the value, the better the results. An IoU threshold value (confidence level) can be used to define whether a prediction is a TP or an FP, e.g., objects with IoU greater than 0.5 can be considered a TP case for many kinds of applications. In Figure 4 we present a comparison between different IoU values when considering the ground-truth and the estimated bounding box in an object detection system. As we can see, a threshold of 0.5 represents a good detection. When we deal with video object detection, a detection can vary over time, and detections in the following frames can confirm the first detection.

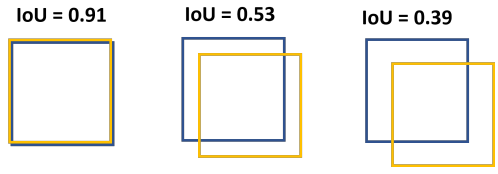

Figure 3: Examples of different IoU metric values

- **Average Precision (AP)**: is a popular evaluation metric used for object detection, which combines Recall and Precision obtained with different threshold values.

To calculate the value of AP it is necessary to define a set R of 11 equally spaced recall results ($R = \{0, 0.1, 0.2, ..., 0.9, 1\}$). For each $r_x \in R$, two equations are computed:

- precision(r_x): returns the precision of the prediction method when the recall value is r_x. When this function is computed to all values of $r_x \in R$, the result is a precision-recall curve; and
- max(precision(r_x)): selects the highest value obtained with the precision(r_y) function, for all values $r_y \in R$, let $r_x \geq r_x$.

Figure 4 presents examples of curves obtained with the functions precision(r_x) and max(precision(r_x)).

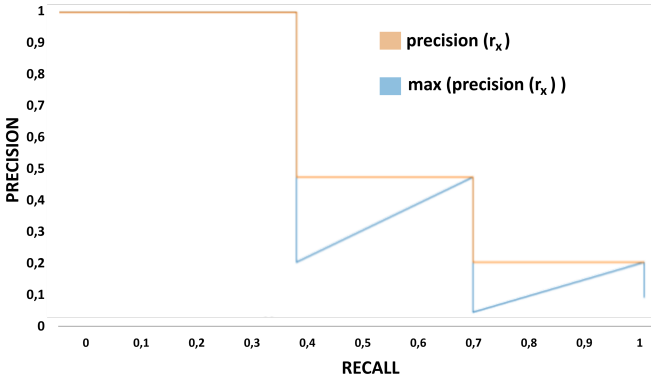

Figure 4: Example of precision(r_x) e max(precision(r_x)) curves

Based on the function max(precision(r_x)) the Average Precision (AP) is computed with the equation in Equation 5.

$$AP = \frac{1}{11} \sum_{x=1}^{11} max(precision(r_x)) \tag{5}$$

- **Frames Per Second (FPS)**: represents the number of frames that the algorithm was able to process per second. It is an important metric, as navigation systems demand real-time processing.

4 EXPERIMENTS

We train and tested our models on a PC with a CPU Ryzen 7 2700X, 32GB of RAM, and a GPU Geforce RTX 2070 Super. The source code, trained models, and an example of object detection video are available at https://github.com/... (the link will be shared if the work is approved - to not interfere in the double-blind review process).

4.1 Experimental Results

Table 4 presents the results obtained with each of the trained models applied to detect objects in the test set. Each table row refers to one of the trained models, and each column denotes one of the evaluation metrics - in the order they have presented in the Subsection 3.4. As we can observe, all the tested methods are likely to real-time processing, performing at a minimum frame rate of 62 in the slowest network (YOLO-v4).

Table 4: Experimental Results. th_α refers to the minimum value to perform detection and select a class by the network and th_β refers to the threshold to consider a detected bounding box good enough (IoU ≥ 0.5). FPS data extracted from [32] and estimated with a GPU GTX 1080ti.

Method	Threshold th_α	Threshold th_β	Recall	Precision	F1-score	IoU	AP	*FPS
YOLO-v3	0.25	0.50	0.91	0.88	**0.90**	72.76%	92.34%	73
YOLO-v3 *tiny*	0.25	0.50	0.80	0.81	0.80	65.16%	84.93%	368
YOLO-v4	0.25	0.50	0.88	**0.91**	0.89	73.70%	**93.08%**	62
YOLO-v4 *tiny*	0.25	0.50	0.78	0.90	0.84	**77.12%**	82.22%	**371**
YOLO-v4 *scaled*	0.25	0.50	**0.92**	0.87	**0.90**	73.76%	92.32%	97

Concerning the efficiency of the object detection task, we note that all the networks have an AP value varying from 82% (YOLO-v3 tiny) to 93% in the YOLO-v4 network. To improve a trade-off analysis to simplify the model selection, we plot in the Figure 5. This graph's horizontal and vertical axes represent the FPS and the % AP, respectively. We also plot the Pareto frontier of the tested methods. We can observe that every method represents a good selection in an application scenario with a highlight on the: YOLO-v4, as the highest AP value; the YOLO-v4 tiny as the fastest network; and the YOLO-v4 scaled as a good choice regarding efficiency and processing time.

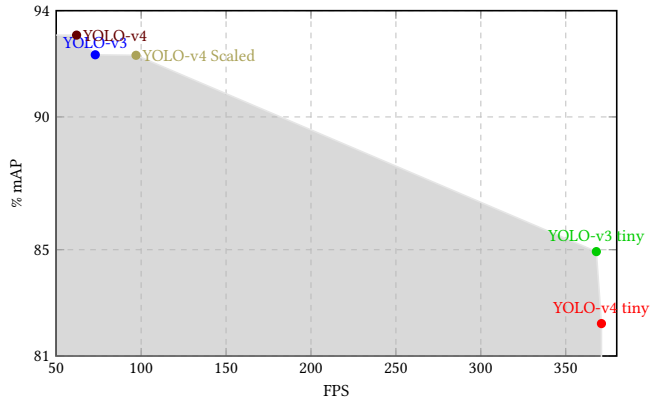

Figure 5: Scatter plot of the model selection experiment. The light gray area highlights the Pareto frontier.

4.2 Qualitative results

To show the effectiveness of the tested method, we present some examples of detections on our dataset of the YOLO architectures we use. In Figure 6 we present results of the tested networks for three video frames, of the scene with ID 11 on the Table 2. Despite the AP results previously presented, we observe that such detections do not differ as much in qualitative analysis.

5 CONCLUSION

This work presents experimental results for video-based detection of objects in an uncontrolled indoor environment (corridors of the largest building of a university campus). The case study was conducted with five state-of-the-art deep-learning-based models (YOLO-v3, YOLO-v3 tiny, YOLO-v4, YOLO-v4 tiny, and YOLO-v4 scaled). We achieved over 93% performance on average in our experimental database - which was created for this study and was freely available.

The results obtained represent the initial efforts to develop a system to assist the navigation of the visually impaired in uncontrolled environments. This paper also unlocks applications regarding robotic navigation in indoor environments. In future works, we intend to extend the amount of data detected, analyze the algorithms in outdoor environments, and develop an application that helps impaired people navigate our campus. We also aim to conduct tests with blind people and develop wearable strategies to improve our system.

As we provide our labeled database and the algorithms used in the experiments, we hope to contribute to related works.

REFERENCES

[1] World Health Organization. Blindness and vision impairment, 2020. URL https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.
[2] Mariana Almeida. As Condições da Saúde Ocular no Brasil. *Revista Universo Visual*, pages 6–8, sep 2019. Available: https://universovisual.com.br/secao/edicoes/pdfs/UV113.pdf. Accessed in: 31/12/2021.
[3] Wafa M Elmannai and Khaled M Elleithy. A highly accurate and reliable data fusion framework for guiding the visually impaired. *IEEE Access*, 6:33029–33054, 2018.
[4] Abdelsalam Helal, Mounir Mokhtari, and Bessam Abdulrazak. *The engineering handbook of smart technology for aging, disability, and independence*. John Wiley & Sons, 2008.
[5] Bineeth Kuriakose, Raju Shrestha, and Frode Eika Sandnes. Tools and technologies for blind and visually impaired navigation support: A review. *IETE Technical Review*, pages 1–16, 2020.
[6] Abbas Riazi, Fatemeh Riazi, Rezvan Yoosfi, and Fatemeh Bahmeei. Outdoor difficulties experienced by a group of visually impaired iranian people. *Journal of current Ophthalmology*, 28(2):85–90, 2016.
[7] Jumi Hwang, Kyung Hee Kim, Jong Gyu Hwang, Sungchan Jun, Jiwon Yu, and Chulung Lee. Technological opportunity analysis: Assistive technology for blind and visually impaired people. *Sustainability*, 12(20):8689, 2020.
[8] D Munteanu and R Ionel. Voice-controlled smart assistive device for visually impaired individuals. In *IEEE International Symposium on Electronics and Telecommunications*, pages 186–190. IEEE, 2016.
[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.
[10] Marlon Marcon., Olga Bellon., and Luciano Silva. Towards real-time object recognition and pose estimation in point clouds. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 164–174, 2021.
[11] Shambhavi Roy, Siddhi Gharge, Reha Jasoriya, Palak Agrawal, and Isha Jounjalkar. Orcap: Object recognition cap (a navigation system for the blind). In *IEEE International Conference for Innovation in Technology*, pages 1–5. IEEE, 2020.
[12] Lisa Ran, Sumi Helal, and Steve Moore. Drishti: an integrated indoor/outdoor blind navigation system and service. In *IEEE Conference on Pervasive Computing*

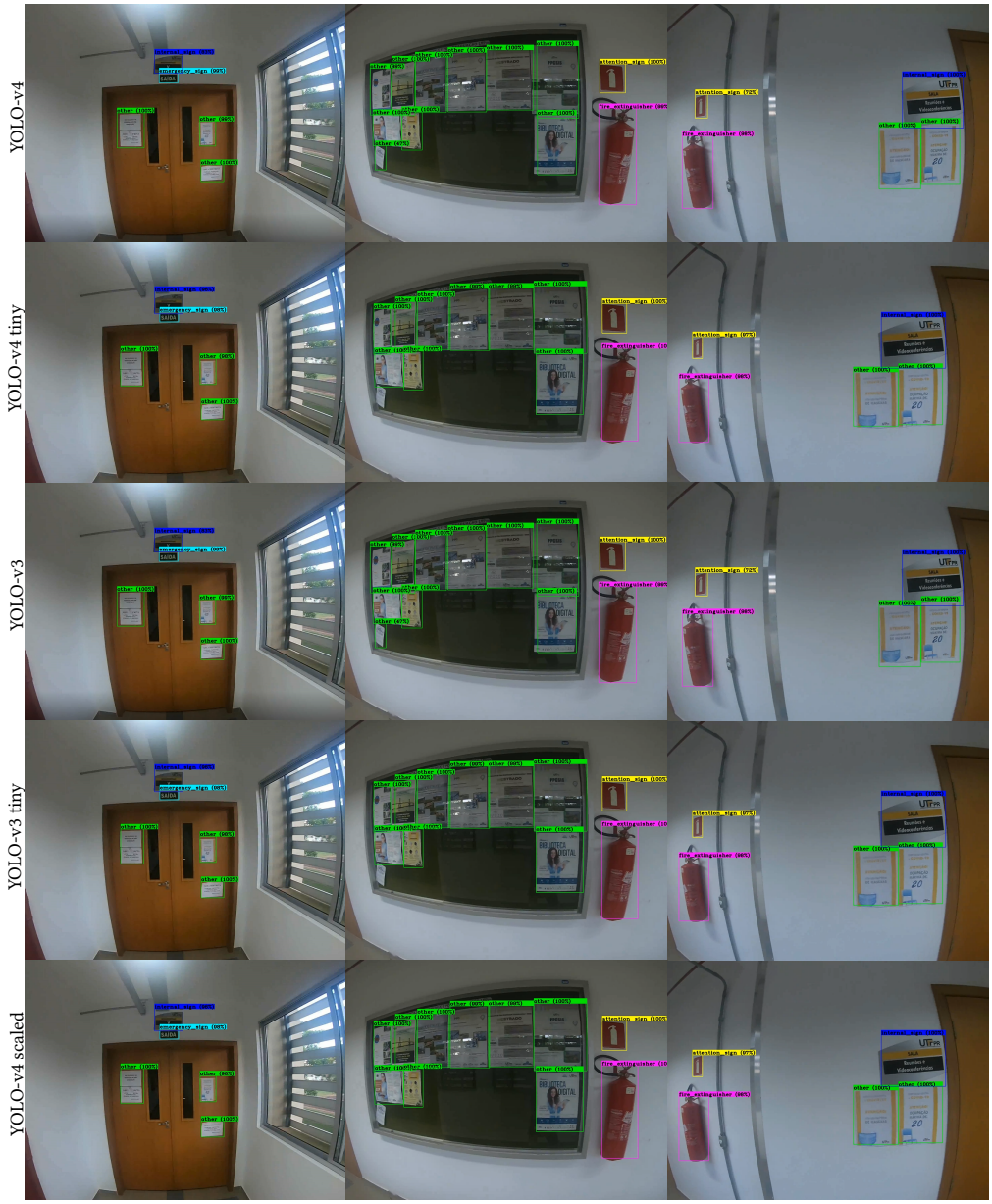

Figure 6: Detection example of tested networks.

and Communications, pages 23–30, 2004.

[13] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[14] André Roberto Ortoncelli, Marlon Marcon, and Franciele Beal. An automated approach to mitigate transcription errors in braille texts for the portuguese language. *Computer on the Beach*, 11(1):326–331, 2020.

[15] Davi Sardinha Pacheco and Amivaldo Batista Dos Santos. Padrões de navegabilidade w3c e acessibilidade na web sob a perspectiva pessoal de um cego. *Computer on the Beach*, 11(1):218–227, 2019.

[16] Luis Claudio Leite Pereira and Janine Kniess. Equipamento vestível para auxílio na mobilidade de pessoas com deficiência visual. *Computer on the Beach*, 11(1): 326–331, 2020.

[17] Bing Li, Juan Pablo Munoz, Xuejian Rong, Qingtian Chen, Jizhong Xiao, Yingli Tian, Aries Arditi, and Mohammed Yousuf. Vision-based mobile indoor assistive navigation aid for blind people. *IEEE Transactions on Mobile Computing*, 18(3): 702–714, 2018.

[18] Daniel Vera, Diego Marcillo, and Antonio Pereira. Blind guide: Anytime, anywhere solution for guiding blind people. In *World Conference on Information Systems and Technologies*, pages 353–363. Springer, 2017.

[19] Jérémy Albouys-Perrois, Jérémy Laviole, Carine Briant, and Anke M Brock. Towards a multisensory augmented reality map for blind and low vision people: A participatory design approach. In *Conference on Human Factors in Computing Systems*, pages 1–14, 2018.

[20] Simona Caraiman, Anca Morar, Mateusz Owczarek, Adrian Burlacu, Dariusz Rzeszotarski, Nicolae Botezatu, Paul Herghelegiu, Florica Moldoveanu, Pawel Strumillo, and Alin Moldoveanu. Computer vision for the visually impaired: the sound of vision system. In *IEEE International Conference on Computer Vision Workshops*, pages 1480–1489, 2017.

[21] G Balakrishnan, G Sainarayanan, R Nagarajan, and Sazali Yaacob. Wearable real-time stereo vision for the visually impaired. *Engineering Letters*, 14(2), 2007.

[22] Masaki Kuribayashi, Seita Kayukawa, Hironobu Takagi, Chieko Asakawa, and Shigeo Morishima. Linechaser: A smartphone-based navigation system for blind people to stand in lines. In *Conference on Human Factors in Computing Systems*, pages 1–13, 2021.

[23] Daisuke Sato, Uran Oh, João Guerreiro, Dragan Ahmetovic, Kakuya Naito, Hironobu Takagi, Kris M Kitani, and Chieko Asakawa. Navcog3 in the wild: Large-scale blind indoor navigation assistant with semantic features. *ACM Transactions on Accessible Computing*, 12(3):1–30, 2019.

[24] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019.

[25] Lu Tan, Tianran Huangfu, Liyao Wu, and Wenying Chen. Comparison of retinanet, ssd, and yolo v3 for real-time pill identification. *BMC Medical Informatics and Decision Making*, 21(1):1–11, 2021.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[29] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[30] Wentong Wu, Han Liu, Lingling Li, Yilin Long, Xiaodong Wang, Zhuohua Wang, Jinglun Li, and Yi Chang. Application of local fully convolutional neural network combined with yolo v5 algorithm in small target detection of remote sensing image. *PloS one*, 16(10):e0259283, 2021.

[31] Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao. Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. In *IEEE/CVF International Conference on Computer Vision*, pages 2778–2788, 2021.

[32] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13029–13038, 2021.

[33] Zicong Jiang, Liquan Zhao, Shuaiyang Li, and Yanfei Jia. Real-time object detection method based on improved yolov4-tiny. *arXiv preprint arXiv:2011.04244*, 2020.