Monocular Deep Learning Multimodal with Object Relevance Estimation for Real-Time Navigation of Visually Impaired individuals

Enrique García Varela[1]
enriquegv001@gmail.com

Rafael Espinosa Castañeda[1]
rafael.espinosa.castaneda@tec.mx

[1] *Instituto Tecnológico de Monterrey, Campus Querétaro, Querétaro, México*

Abstract

This study proposes a real time assistant for visually impaired people. The assistant helps in the navigation of visually impaired by unifying task of panoptic segmentation and monocular depth estimation, through Panoptic FPN and Midas, respectively. The outcome is a video captured on a mobile device, generating spoken descriptions of user's environment to facilitate navigation, applying a heuristic algorithm for adapting prediction to user environment expectation. The model has been tested on members from Asociación Cultural y Recreativa para la Proyección del Invidente Puebla, A.C. (ACRIP) organization, and result effective for user experience analysis.

1 Introduction

From a global perspective, the World Health Organization (WHO) has reported that at least 2.2 billion people worldwide experience some form of distance vision impairment. The primary causes of these impairments are refractive errors and cataracts, affecting just 36% and 17% of the population with appropriate treatment, respectively. These conditions can significantly impact children's developmental skills in areas such as motor functions, language acquisition, emotional well-being, social interactions, and cognitive abilities. Similarly, adults may experience depression, anxiety, increased isolation, and a heightened risk of injuries due to falls while navigating their surroundings. WHO (2023b)

In Latin America and the Caribbean, there are approximately 50,000 blind individuals and 20,000 visually impaired individuals per million people. These individuals can be categorized by their specific deficits in distance and near vision. Such circumstances are notably prevalent in marginalized locations. WHO (2023c)

The WHO defines Assistive Technology as products designed to aid individuals in functioning independently and enhancing their well-being. However, ensuring inclusively in creating assistive technology poses several challenges. These challenges stem from product characteristics, including orientation towards affluent markets, inadequacies in delivery systems, and the need for contextual specialization of the product. WHO (2023a)

As is described in the review Bineeth Kuriakose and Sandnes (2022), various assistive travel aids offer enhanced navigation support to users, serving as complementary tool alongside their cane. These aids are categorized into Electronic Orientation Aids (EOAs), Position Locator Devices (PLDs), Electronic Travel Aids (ETAs). EOAs utilize sensors to detect paths and obstacles, however, their high computational requirements limit their real-time applicability. PLDs integrate Global and Geographic Information Systems to guide individuals based to reach their objective base on their current location, nonetheless, they do not detect obstacles and are ineffective indoors. ETAs rely on mobile phone cameras, ultrasonic sensor, Radio Frequency Identification, or other similar technologies. Except for mobile phones, often come with high costs, inability to predict object classes, weak signals, or limited portability.

Hence, utilizing smartphones as a complementary tool alongside canes has emerged as one of the most effective solutionsKuriakose et al. (2023). However, a notable limitation lies in the passive methods employed by monocular cameras to predict depth estimation, rather than applying signal reception for precise distance measurement. Kuriakose et al. (2022)

The motivation for this research is underscored by its impact on the United Nation Goals. Specifically addressing the third goal: to ensure healthy lives and promote well-being for all at all ages. By assisting a population with visual disabilities, the aim is to mitigate psychological second effects they may experience and reduce daily accidents. Additionally, it aligns with the tenth goal: to reduce inequality within among countries. By enhancing the quality of life for individuals facing visual impairments, particularly considering the correlation in Latin America between inequality and marginalized locations Centeno (2018), whereas visual impairment cases are more prevalent. WHO (2023c).

2 Related Work

2.1 Object Detection

One of the primary architecture types aimed at enhancing accuracy in object detection is the Two Stage Detectors. These models employ backbone structures to get region proposals and construct a classifier along with bounding boxes model on top of this base. In contrast, One Stage Detectors outputs object classification from a neural network architecture and it don't require additional backbone structures, by this it enhance speed processing.

Noticing the capabilities of Two Stage Detectors in enabling real-time predictions, several noteworthy architectures are described hereafter. R-CNN Girshick et al. (2014) warps each object by fine-tuning AlexNet with an IoU threshold greater than 0.5, applying SVM Classifier and Greedy Non-maximum Suppression to select the most accurate prediction. SPP-Net He et al. (2014) acquires a feature map for each object detected through convolutions blocks. These maps are subsequently divided into multiple gird bins, encoding features via a spatial pyramid using a pooling layer. They are then appended to input into a fully connected layer that returns the object labels.

Faster R-CNN Ren et al. (2016b) starts with region proposal, extracting a region of interest (RoI) for each object's dimensions. Following this, it applies a max-pooling and fully connected layers to generate an RoI feature vector. This vector has two heads: one utilizes a Fully Connected and softmax activation for class prediction, while the other employs Fully Connected and regression layers for bounding box output. Faster R-CNN uses Singular Value Decomposition for storing the dominant wights.

RPN Ren et al. (2016a) employ Faster R-CNN as backbone. It uses a selective search via Fully Convolutional Net (FCN) to get from RoI and convolution feature map anchor boxes, which are more accurate than bounding boxes. The selective search method determines whether these anchor boxes contain an object and performs predictions only on those that do. Furthermore, its optimization process involves sharing model parameters to combine stages in the fine-tuning process.

R-FCN Dai et al. (2016) addresses the translation-invariance issue encountered in detecting the same object in various positions. It accomplishes this by dividing bins based on position-sensitive score maps. In simpler terms, pixels corresponding to each label are grouped according to their location (e.g., top-left, right-bottom categories). FPN Lin et al. (2017) aims to predict objects of the same class but with different scales. It achieves this by employing a two-step process involving a down-sampling pyramid followed by an up-sampling pyramid. The second pyramid inherits semantic features from the same-level layer of the first pyramid, and each resolution modification is derived from a residual block.

Deformable CNN Dai et al. (2017) aims to achieve geometric transformation for pixel-wise classification, considering factors such as object scale, pose, viewpoint, and part deformation. It employs a deformable convolution kernel to adapt the Region of Interest (RoI) structure. This kernel's deformation is adjusted by an offset obtained from another convolution feature map from the RoI. Mask R-CNN He et al. (2018) fine-tunes Faster R-CNN, utilizing backbones from ResNet or FPN. It incorporates the RoI Align method that utilizes bilinear interpolation before the pooling layer. Additionally, a convolutional structure is implemented on the object detector to facilitate instance segmentation.

CenterNet Zhou et al. (2019) stands out as one of the top deep learning detectors in terms of Accuracy Precision. It operates using a backbone consisting of Fully Convolutional Encoder-Decoder networks like ResNet He et al. (2015), employing residual learning blocks that incorporate skip connections to optimize training for deeper networks. Additionally, Hourglass Newell et al. (2016) utilizes an encoder-decoder structure through residual blocks, while DLA Yu et al. (2019) focuses on hierarchical fusion, incorporating additional scales and semantic features. Its second stage comprises three main components, these are a head for predicting object classes, an offset head for correcting democratization error and head dimension for specifying bounding box edges. Following this, a peak extraction filter is applied to identify true categories. The object classification head, which utilizes Gaussian kernels 1 and a focal loss function to emphasize the importance of object points.

$$Y_{xyc} = exp(-\frac{(x-\hat{p}_x)^2 + (y-\hat{p}_y)^2}{2\sigma_p^2}) \qquad (1)$$

In which, p is all the key points and $\hat{p}$ the lower resolution equivalent

$$L_k = -\frac{1}{N}\Sigma_{xyc}\begin{cases}(1-\hat{Y}_{xyc})^a log(\hat{Y}_{xyc}) \ \ if \ Y_{xyc}=1\\ \\ (1-Y_{xyc})^b(\hat{Y}_{xyc})log(1-\hat{Y}_{xyc}) \ \ other\end{cases}$$

(2)

Where $a = 2, b = 4$. And the way it works is with the difference from Gaussian kernel center pixel vs true keypoints to increase Loss function. While for each other pixel different from keypoints, there is a discount rate to increase the weight of the loss as they get closer to each center keypoint. Subsequently additional stages are used to get 3D detections and Human pose estimations. On COCO Lin et al. (2015) dataset for objects instance segmentation, the model achieved good performance with results from Hourglass-104 with 40.3 AP and 14 FPS, for speed ResNet-18 with 28.1 AP and 142 FPS and for trade-off DLA-34 with 37.4 AP and 52 FPS.

Panoptic FPN Wu et al. (2019) model integrates instance and semantic segmentation computations, delivering accurate predictions characterized by high-resolution outputs, semantic richness, and multi-scale features. This model facilitates the classification of instance segmentation for each pixel, detecting *Stuff Classes* and assigning *Thing Classes*. The architecture involves the initial acquisition of multi-scale features through the FPN, featuring 256-channel outputs and a ResNet backbone trained on ImageNet. This pyramid is used in instance branch, for scaling features across resolutions from 1/32 to 1/4. Subsequently, a region-based branch is employed, for instance segmentation using pooling regions of interest (RoI) via Faster R-CNN. For semantic segmentation, the model utilizes a FPN with three up-sampling stages consisting of layers such as 3x3 convolutions, normalization, and 2x bilinear up-sampling, using 2-pixels interpolation to increase image size. These stages operate at scales of 1/16, 1/8, and 1/4, followed by a 1x1 convolution, 4x bilinear up-sampling (using 4-pixels function interpolation), and softmax activation. The integration of Mask R-CNN enhances the output by combining Faster R-CNN instance and FCN semantic branches.

During training, the Loss Function $L = \lambda_i(L_c + L_b + L_m) + \lambda_s L_s$ is employed, where classification loss is L_c, mask loss L_m, bounding box loss L_b, and semantic loss L_s. Here, λ acts as a tuning parameter optimizing instance (i) or semantic (s) focus. Post-processing to balance losses from branches involved adjusting mini-batches, learning rates, and data augmentation. Training was conducted on COCO and Cityscapes Cordts et al. (2016) datasets, which contains semantic segmentation from urban street images.

2.2 Depth estimation

The MiDaS Ranftl et al. (2022) architecture encodes images using ResNet as a backbone and employs convolution blocks for decoding, creating disparity feature maps that result in a depth map. The encoder relies on extensive training datasets, leveraging a combination of 3D movie frames. It is notably disruptive due to its accuracy in monocular frames without necessitating stereo pairs.

The latest iteration, MiDaS v3.1 Birkl et al. (2023), experiment on PyTorch timm repository. The backbone with best performance is BEiT with 512 resolution Bao et al. (2022), this is a self-supervised vision Transformer (ViT), useful for a proposed pre-training task named *mask image modeling (MIM)*. This process copies BERT model but with a focus on visual tokenization, transforming image pixels into discrete tokens. Through a decoder the image is reconstructed, thereby incorporating embedding information obtained from image patches representing spatial value dimensions.

Due to the model's diverse training datasets, various error metrics were employed for evaluation. The DIW dataset utilized the Weighted Human Disagreement Rate, while the ETH3D and Sintel datasets assessed the mean absolute relative error in depth calculations $\frac{1}{M}\Sigma_{i=1}^{M}|d_i - d_i^*|/d_i^*$, comparing the relative depth d_i against ground truth values d_i^*. The KITTI, NYU Depth v2, and TUM datasets employed the percentage of *bad depth pixels* δ_1, mapping pixels where $max(d_i/d_i^*, d_i^*/d_i) > 1.25$. Additionally, the root mean square error for disparity D_i was utilized for evaluation $(\frac{1}{M}\Sigma_{i=1}^{M}|D_1 - D_i|^2)^{\frac{1}{2}}$.

Roboust-CVD Kopf et al. (2021) is an algorithm designed for monocular dense depth map estimation and the integration of object Optical flow. It starts creating a dense optical flow between frames and a R-CNN classifier selects consistent or dynamic pixels. Afterwards, CNN generates a depth map, serving an input for the next layer which computes 3D reprojection for the current pixel using depth parameters (scalar coefficient) and for next pixel using camera parameters (intrinsic and pose estimation). Then optimization stage involves iterative switching between camera and depth parameters, followed by post-process that facilitates flexible deformation and geometry aware-depth filter.

2.3 Real-time smartphones

DeepNavi Kuriakose et al. (2023) is a smartphone application with monocular depth estimation capabilities, used for real-time navigation and Scene identification. By using detector EfficientDet-Lite4 a BiFPN for backbone trained on ImageNet, alongside the Rule of 57 for distance estimation. It generates outputs employing a text-to-speech model using Pyttsx-TFLite python's module. Users have acknowledged the model's overall performance, despite occasional incorrect classifications, achieving an average accuracy object detection score of 87.7%. Suggested enhancements include expanding the range of detectable classes, given the training was conducted on a limited dataset of 20 labels. Additionally, referencing other models could augment its capabilities.

3 Methodology

Our proposal is a Multimodal Deep Learning architecture designed to convert video into speech, aiding users in real-time navigation by describing relevant objects. The architecture comprises two branches: the first is to obtain object detection and the second to get depth estimation. The initial branch utilizes the Panoptic FPN transformer, followed by a heuristic algorithm for classifying object hierarchies for each pixel, named *Hierarchical Indexation.* This process generates the panoptic segmented tensor and a mask with class hierarchies. Meanwhile, the second branch employs the MiDaS v3.1 transformer, followed by an algorithm to determine the object's proximity to the camera, referred to as *Depth Awareness.* This generates an image mask indicating the proximity scale value for each pixel. Upon processing an initial frame through both branches, the output involves the fusion of the tensors for detected objects, the class hierarchy mask and the proximity scale mask, see architecture in figure 1.

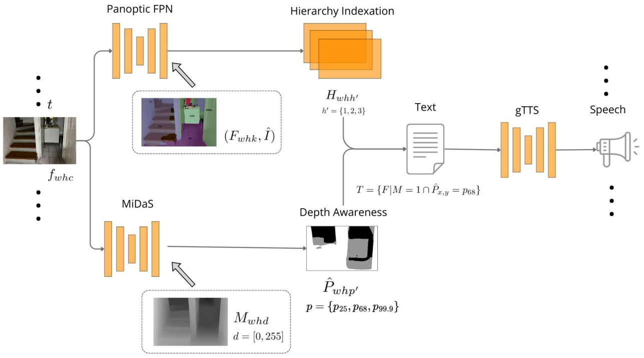

Figure 1: Our propose

Let f_{whc} represent the frame at time t, where w, h, and c denote the width, height, and color channels, respectively. Which will iterate over time to get speech.

In the branch for object detection, let $(F_{whk}, \hat{I})$ be the output tuple obtained from Panoptic FPN. Respectively, the tuple contains a tensor with k values for each class id at position $\{x, y\}$ in f, as well as a dictionary with key k and its values are the array α with information from predicted classes. Additional, $H_{whh'}$ is the class hierarchy mask obtained from *Hierarchy Indexation* algorithm, this is an tensor which $h' = \{1, 2, 3\}$ dimension are hierarchy values at position $\{x, y\}$ in f, respectively this dimension values represent *very relevant, relevant, not relevant.*

The steps in *Hierarchy Indexation* algorithm are: let I be a dictionary with the key i for instance segmentation classes and S another dictionary with key s for semantic segmentation classes, with h' values for both dictionaries. First map from I and S to $\hat{I}$ to append h' in α, next map F with $\hat{I}$ to get H.

In the branch for depth estimation, let M_{whd} be the MiDaS output depth map an array with relative depth $d = [0, 255]$ (scale: closer-far) at position x, y in f. And $\hat{P}_{whp'}$ the proximity scale mask obtained from *Depth Awareness* algorithm, which is an tensor with $p' = p_{25}, p_{68}, p_{99.9}$ dimension as the corresponding percentile value from M_d distribution, at position $\{x, y\}$ in f. The p' dimension values depict proximity layers, respectively, *very close, close, far.*

The steps in *Depth Awareness* algorithm are: let β be a threshold parameter for comparing σ_M, which is the standard deviation from M_{whd}. If $\beta < \sigma_M$ then remove from M all values smaller than p_{90}, where p is the percentile from M. Map M to function $\hat{P} : d \to d$.

$$\hat{P}(M) = \begin{cases} p_{25} & \text{if } d_{x,y} <= p_{68} \\ p_{68} & \text{if } p_{68} < d_{x,y} <= p_{99.9} \\ p_{99.9} & \text{if } d_{x,y} < p_{99.9} \end{cases} \tag{3}$$

The condition for removing pixels from M is necessary since proximity mask is obtained from percentile values. Thus, when pixels are removed means that its distribution is from a f with large depth, therefore proximity layers are obtained just from closer pixels and not the whole image. Aiming to find the statistical metric to classify between large and small depth images, a descriptive analysis was experimented over depth map pixel's distribution, for more information see **Results** section.

Let T be the very relevant close objects to the user, which are obtained fusion tensors in function $T = \{F|M = 1 \cap \hat{P}_{x,y} = p_{68}\}$. These objects are appended to a string, which then is computed to count the number of repeated objects. Finally, this string is the input to module google Translate's Text-to-Speech (gTTS) and plays the audio in Jupiter notebook, with *IPython.diplay Audio* method.

The algorithm script was developed using Python and JavaScirpt to be executed on Google Colab, leveraging T4 GPU capabilities. Deep Learning models were implemented through Transfer Learning techniques using Python3, PyTorch and Detectron2 Model Zoo baseline for Panoptic FPN. Camera visualization functionalities were employed via OpenCV within Colab. Additionally, DroidCam v6.5.2 Client, a cross-platform software, enabled smartphone usage as a webcam through LAN connection. The infrastructure components utilized during testing included the Windows 10 operating system and Microsoft Edge version 119.0.2151.58.

Regarding the test, the equipment used included a Pavilion Laptop and an iPhone 13. Furthermore, a custom smartphone case was designed to be carried on the user's chest, enhancing accessibility. Bluetooth AirPods earphones were also employed to improve sound quality.

Additionally, a qualitative analysis was performed through interviews with eleven visually impaired and blind members from ACRIP. The test was conducted in an indoor scenario illustrated in 8. The user experience evaluation employed a three-section interview questionnaire format, as shown in 1, to identify the user needs and perspectives regarding accessibility and prototype performance. Furthermore, thematic and sentiment analyses were applied to evaluate the results.

4 Results

First experiments to select which model to apply, where comparing rule 57 from DeepNAVI Kuriakose et al. (2023) and YOLOv4 Bochkovskiy et al. (2020), by following AiPhile (2021) article. But neither of their performances were accurate. Next, we experimented through depth maps and searched to get a proximity scale on it, although this was not a real distance estimation it was useful for our purpose of estimating only relevant objects. See tests in figure 6. Additionally, neither YOLOv4 nor CenterNet could classify all pixels to detect different objects. Therefore, we decided instead to fusion Pantoptic FPN and proximity layers obtain from our depth estimation proposal.

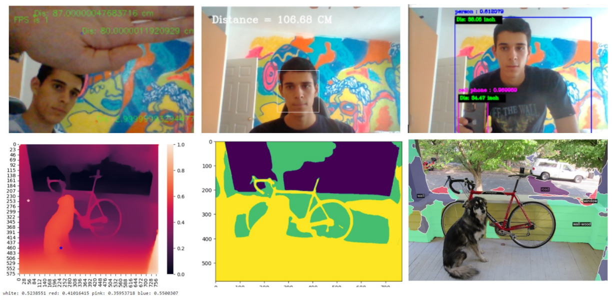

Figure 2: Algorithm evaluation

Related to the Depth Awareness threshold, experiments have been tested by comparing percentile score and standard distribution, between small and large depth frames, on a supervised dataset that we have created.

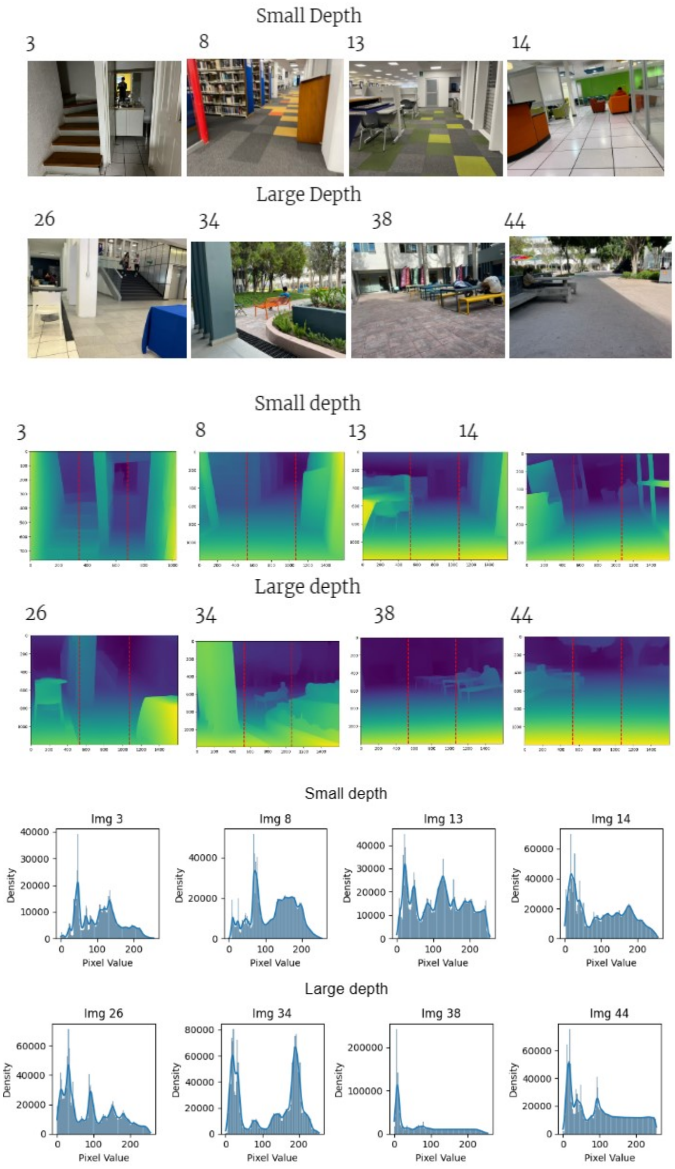

Figure 3: Images depth, division

In figure 7, first displays intial classification frames from our dataset, next depth maps output from MiDaS and third pixels distribution from the depth maps. These present a right-skewed shape and there is a possible difference from shapes, between small and large depth frames. Notably, *small depth* frames tend to approximate a normal distribution, while *large depth* exhibit more stochastic shapes.

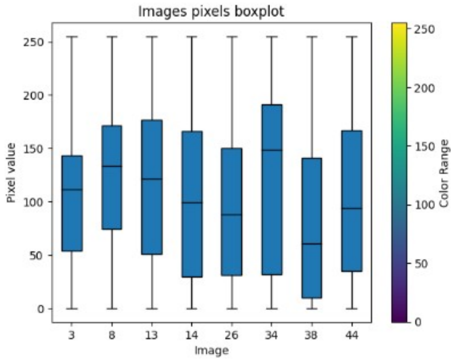

Figure 4: Standard deviation

The box-plot was utilized to compare pixel distributions, revealing that the Q3 value shows no significant difference in depth classification between values ranging from 130 to 180. However, a distinct pattern emerges where the Q1 value for *small depth* is notably higher than the majority of *large depth* values. Similarly, the mean pattern indicates a narrower range for *small depth* compared to the range observed in *large depth* values.

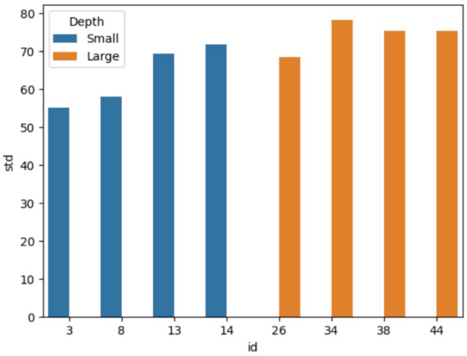

Figure 5: Standard deviation

Upon comparing pixel standard deviations, it was observed that three out of four *small depth* images display values under 70, whereas three out of four images with *large depth* exhibit values exceeding 70.

Given the differences observed in distribution shape and standard deviation between depth classes, the standard deviation metric was employed for comparing Depth Awareness thresholds.

Furthermore, see 6 for tests with ACRIP members. The thematic qualitative analysis graph 7 was generated through the application of specific data transformation and encoding steps from their answers. The subsequent steps include:

1. Select Needed Objects mentioned in question from common issues and accident anecdotes.
2. Create Hierarchy Classification variable and check the relevance for each object or if they were *missing objects* for object not trained in the models.
3. Create variable E = technical failures, in three categories and assign a value for importance: TP (0) = No error, FP_det (1) = error in detection incorrect classification label, FP_mid (1) = error in incorrect depth layer assignation, FN (2) = error didn't detect object.
4. Create variable P = pass test, where if test was passed equals to 1, if not 0.
5. Create classifier variable S = type of feeling: 1 = positive, 0 = neutral, -1 = negative.
6. Q = Quantity of extra objects detected, that shouldn't appear.
7. M = Message effectiveness: clear or concise = 1, both = 2, none = 0.
8. T = Amount of tasks where the model is useful, more than one = 2, one = 1, none = 0.
9. Expand all objects per user.
10. Create variable for frequency that equals 1 for each user.
11. Compute Likeability indicator per category in 4, where N = Amount of users tested.
12. Group all objects, using sum as aggregation function to all the numeric variables.
13. Add a variable for theme to group objects by category that where not detected.
14. Group categories from same objects, using sum as aggregation function for all numeric variables.

$$L = \frac{\sum_{i=1}^{N}(P + S + M + T - Q - E)}{N} \tag{4}$$

Figure 6: Users test

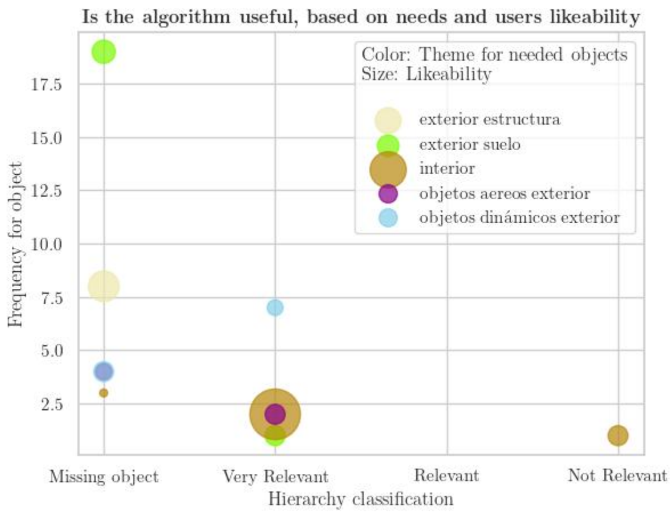

Figure 7: frequency of thematic objects category by trained hierarchy classification, with dot shape size scale on likability

Although, figure 7 describes Very Relevant objects to have a higher frequency than the other two less relevant hierarchical classes, an even likeability average is balanced between objects that are Very Relevant and other object which are missing for models training. Most frequent object themes are not present on the model's learned labels. Is also worth identifying that objects corresponding to *exterior suelo* are really important, both for a moderated appreciation from the user and for its huge impact on the needs. Whilst for highest satisfaction is from users presented that their needs at the interior.

Although, figure 7 indicates a higher frequency of *Very Relevant* objects compared to the other two less relevant hierarchical classes, an even likeability average is observed between objects categorized as *Very Relevant* and those absent from the models during training. The most frequently occurring object themes are not included in the labels learned by the models. Notably, objects corresponding to *exterior suelo* hold significant importance, marked by a moderate appreciation from users and a substantial impact on their needs. Whilst, the highest satisfaction is reported by users who express their needs within interior settings.

5 Conclusions

Throughout this paper, we introduced a multi-modal neural network designed to aid visually impaired individuals in indoor and outdoor navigation through heuristic automation, employing standard deviation from monocular depth maps. The user experience has been generally positive, despite limitations are mainly in lack for relevant objects for them.

The findings, broader implication is obtained mainly in the algorithms presented for depth estimation in real-time through monocular cameras, via a way for delimiting proximity areas. Same it has relevance in the user experience, for visually impaired individuals.

Thus, further research can search for improvements in the model by applying Panoptic FPN feature extraction on broader range of classes for exterior ground objects. Also, to change the heuristic algorithms in each branch, to compute deep learning layers to make hierarchy and proximity classification and to evaluate the model on mean Average Precision metric. Finally, test the algorithm on an outdoor environment.

6 Acknowledgments

I, Enrique García, would like to thank Rafael Espinosa for his valuable support and guidance throughout the dissertation. Additionally, I am deeply grateful for the unwavering support of my parents.

References

AiPhile. Distance(webcam) estimation with single-camera opencv-python. *Medium*, 2021. URL `https://medium.com/mlearning-ai/distance-estimation-with-single-camera-opencv-python-2`

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022.

Raju Shrestha Bineeth Kuriakose and Frode Eika Sandnes. Tools and technologies for blind and visually impaired navigation support: A review. *IETE Technical Review*, 39(1):3–18, 2022. doi: 10.1080/02564602.2020.1819893. URL `https://doi.org/10.1080/02564602.2020.1819893`.

Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

Lajous A. Centeno, M.A. Challenges for latin america in the 21st century. *OpenMind BBVA*, 2018.

URL https://www.bbvaopenmind.com/en/articles/challenges-for-latin-america-in-the-21-st-century/.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016.

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks, 2017.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, page 346–361. Springer International Publishing, 2014. ISBN 9783319105789. doi: 10.1007/978-3-319-10578-9_23. URL http://dx.doi.org/10.1007/978-3-319-10578-9_23.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation, 2021.

Bineeth Kuriakose, Raju Shrestha, and Frode Eika Sandnes. Distance estimation methods for smartphone-based navigation support systems. In Kohei Arai, editor, *Intelligent Systems and Applications*, pages 658–673, Cham, 2022. Springer International Publishing.

Bineeth Kuriakose, Raju Shrestha, and Frode Eika Sandnes. Deepnavi: A deep learning based smartphone navigation assistant for people with visual impairments. *Expert Systems with Applications*, 212:118720, 2023. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.118720. URL https://www.sciencedirect.com/science/article/pii/S0957417422017432.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation, 2016.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016a.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016b.

WHO. Assistive technology. *World Health Organization*, 2023a. URL https://www.who.int/news-room/fact-sheets/detail/assistive-technology.

WHO. Blindness and vision impairment. *World Health Organization*, 2023b. URL https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.

WHO. Visual health. *Pan American Health Organization*, 2023c. URL https://www.paho.org/en/topics/visual-health.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation, 2019.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019.

7 Appendix

7.1 GitHub repository

https://github.com/enriquegv001/depth_and_det_visual_impair

7.2 Users tests

Evaluation section	Questions
Needs Pain Points (1) Expectations (2) Motivations (3)	Problemas más comunes en navegación diaria (1) Anécdotas de accidentes (1) ¿Si hubiera un método de asistencia en la navegación con tecnología que esperarías que este tuviera? (2) ¿Si hubiera un método de asistencia con su trabajo con tecnología que esperarías que este tuviera? (2) ¿Consideras que habría alguna característica en particular que te fuera de agrado? (E.g. Poder adaptar la voz) ¿Hay algo en particular que te agrade de la navegación en tu día diario? (3) ¿Conoce alguna tecnología para poder asistirle con alguna tarea?
Accessible guidelines	¿Qué sentimiento tiene al navegar con la aplicación? ¿Habría algún objeto que no requiere que te avisen? ¿El mensaje es claro y conciso? ¿Te gustaría que vibraran los lentes, cuando se avise de algún obstáculo? ¿Cuánto está dispuesto a pagar por una tecnología como esta? ¿Para qué tareas crees que te sería más útil este aparato? ¿En base a las pruebas realizadas que tan útil considerar, poder usar estos lentes con estas características? ¿Consideras que habría alguna característica en particular que te fuera de agrado, como poder adaptar la voz? ¿Hay algo en particular que te agrade de la navegación en tu día diario? ¿Considera alguna mejora para esta aplicación tecnológica?
Real user test	¿En que entorno necesita utilizar más este aparato, ya sea interior o exterior? ¿El algoritmo presenta alguna falla técnica durante las encuestas? ¿Logró pasar la prueba o no?

Table 1: Questions Qualitative Analysis

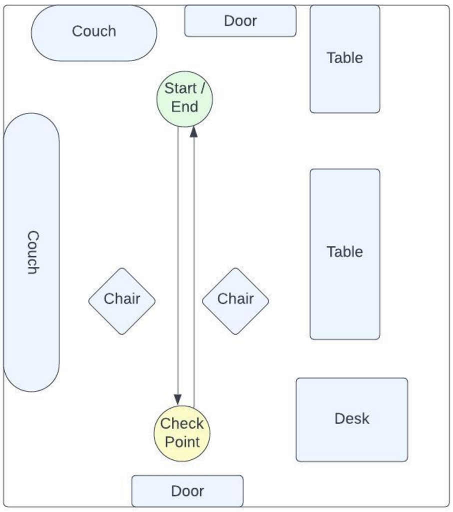

Figure 8: Test Scenario