IntelliCane: An Agentic Approach to Real-Time Obstacle Avoidance and Intelligent Decision-Making for the Visually Impaired Through a Monocular Servo-Guided Cane Using Deep Learning-Based Environmental Mapping

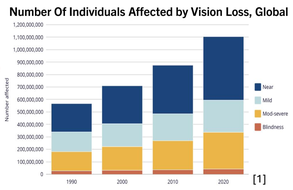
Tanay Chitlur & Akash Ragam — Portland OR, USA

ROBO-0066T

Background

Over 1 billion people world-wide are visually impaired with more than 300 million people having severe or complete blindness

- Worldwide there are only ~**28,000 guide dogs for 300+ million people** that are visually impaired
  - Guide dogs can cost upwards of **$60,000 per year**
- White canes and guide dogs aid but **do not actively help avoid obstacles and prevent collisions**
- **Collision avoidance**, **wayfinding**, and **object localization** are key difficulties for the visually impaired
- Sensors (LiDAR and GPS) alone only **minimally aid navigation and slow users down**
- Most electronic travel aids (ETAs) **reduce speed**, affecting mobility and independence



Current Solutions are not adequate!

- **No intelligent navigation based on situation analysis or user goals**
- **WeWalk Smart Stick©, the current most technologically advanced cane, only warns users of front obstacles and does not provide guidance around them**

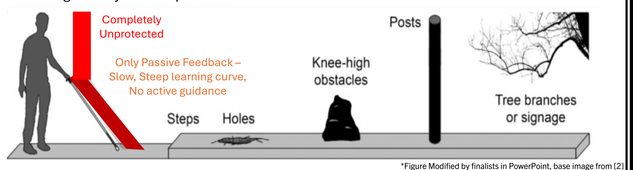


*Figure Modified by finalists in PowerPoint, base image from [2]

Problem Statement

Problem	Unmet Needs	Proposed Solution
Limited navigation tools for visually impaired individuals result in reduced mobility, independence, and reduced safety	Existing assistive devices do not provide active guidance, limit independence, have a steep learning curve, and significantly reduce walking speed	IntelliCane – Goal-conditioned smart cane using real-time vision, object detection, and LLM-based reasoning for intelligent, independent navigation with active path planning and obstacle avoidance

Figure created by finalists in PowerPoint

Claim

- Develop a compact and intuitive navigation aid to accurately perform SLAM to use with a servo-driven guidance system
- Implement intelligent decision making for the user to navigate safely, independently, and contextually through environments

Impact

- IntelliCane provides real-time ML-based navigation using a camera and servo, increasing mobility and independence for the visually impaired
- The system will be able to actively guide the user, detect hazards, follow paths and instruct the visually impaired user

Goal

- Empowers visually impaired individuals with safe, quick and more independent navigation
- Gives the visually impaired spatial awareness allowing them to confidently interpret their surroundings and make informed decisions

Hardware Overview

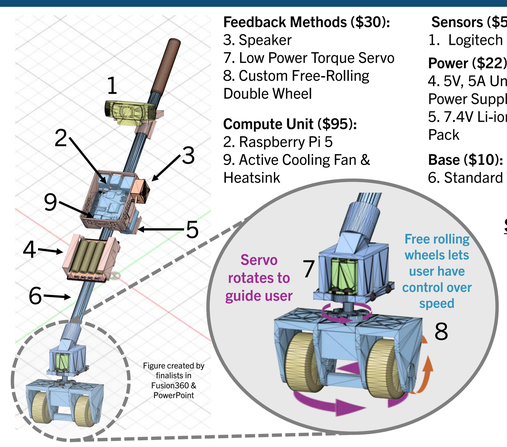


Figure created by finalists in Fusion360 & PowerPoint

Feedback Methods ($30):
3. Speaker
7. Low Power Torque Servo
8. Custom Free-Rolling Double Wheel

Compute Unit ($95):
2. Raspberry Pi 5
9. Active Cooling Fan & Heatsink

Sensors ($50):
1. Logitech c920 Camera

Power ($22):
4. 5V, 5A Uninterruptable Power Supply (UPS)
5. 7.4V Li-ion Servo Battery Pack

Base ($10):
6. Standard White Cane

Specifications:

Total Cost: $207
Weight: 2.8lbs
Battery Life: 2hrs 6mins
Processor Unit: Quad-core ARM Cortex-A76

IntelliCane Capabilities

Obstacle Avoidance

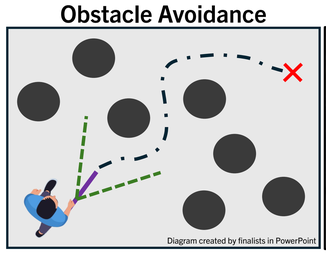


Diagram created by finalists in PowerPoint

Outdoor Keep In Path

Diagram created by finalists in PowerPoint

Object Detection & Action

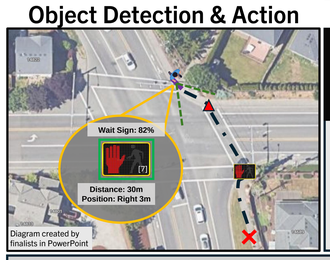


Diagram created by finalists in PowerPoint

Indoor Keep In Path

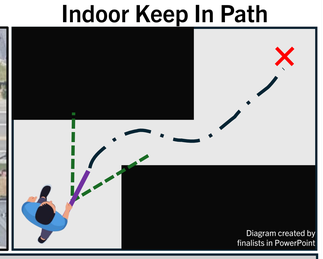


Diagram created by finalists in PowerPoint

- Goal
- Obstacles
- IntelliCane
- Generated Path
- Person
- Wait & Decide
- Camera FOV

Top view of pictures of person from [7], map backgrounds from [8], all other aspect of diagram created by finalists

Obstacle Course Diagrams

Figure created by finalists

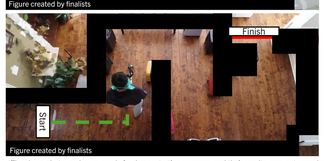


Figure created by finalists

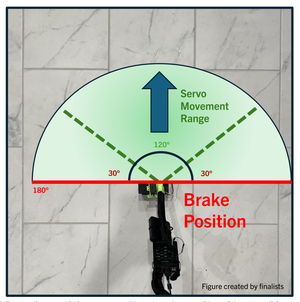


Figure created by finalists

*The above photographs are purely for demonstration purposes and their results were not recorded. Due to privacy restrictions, we are unable to show photos or videos of the test participants.

Research Focus

Current Assistance Devices

[3] Guide Dogs

[4] Electronic Travel Aids

[5] WeWalk ©

White Cane [6]

Research Problem

[7]

Current methods do not facilitate independent navigation

- Slow
- Steep learning curve
- Frequent Environmental Contacts
- **No active guidance**

Research Focus

[7]

IntelliCane

Deep-learning based solution utilizing a monocular depth estimation for intelligent navigation

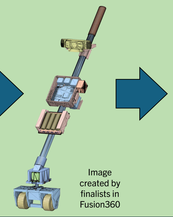


Image created by finalists in Fusion360

Impact

Image created by finalists

Safe, Quick, Intelligent and Independent navigation for the visually impaired

Figure created by finalists in PowerPoint

Over 250 million visually impaired individuals face significant challenges in navigation, limiting mobility and independence. Traditional aids like white canes and guide dogs offer minimal obstacle detection and require extensive training. This project, IntelliCane revolutionizes mobility by leveraging a monocular depth estimation system with deep learning to provide real-time, intelligent navigation—increasing walking speed, safety, and independence for the visually impaired.

IntelliCane Agents and Tools

Visual Perception Tool	Depth Map Creation Agent	Mapping + Path Agent	Object Semantics Agent	Path Affordance Agent	Servo & Speech Executor Tool
Retrieves 1080p real-time camera feed from Logitech c920 camera. Lowers resolution, normalizes pixels, and sets framerate **Input** – Raw RGB camera feed **Output** – Preprocessed RGB camera feed	Create depth maps from camera feed using custom built & quantized depth estimation model **Input** – Preprocessed 364x364 RGB camera feed **Output** – Depth map 364x364	Use depth map and A* path planning to determine optimal routes around obstacles **Input** – Depth map 364x364 **Output** – Obstacle avoidance servo position	Uses two transfer-trained object detection models (indoor/outdoor) to identify objects and obstacles. **Input** – Preprocessed 364x364 RGB camera feed **Output** – Objects detected (in-text), distance from user & position on frame	Uses two custom OpenCV edge detectors and camera angle to guide servo movement and keep user on path **Input** – Preprocessed 364x364 RGB camera feed **Output** – Path affordance servo position	Custom servo library for Raspberry Pi 5 to send correct pulse width modulations to servo and text to speech tool **Input** – Final servo position, verbal commands **Output** – Pulse widths to physically move the servo, text to speech output

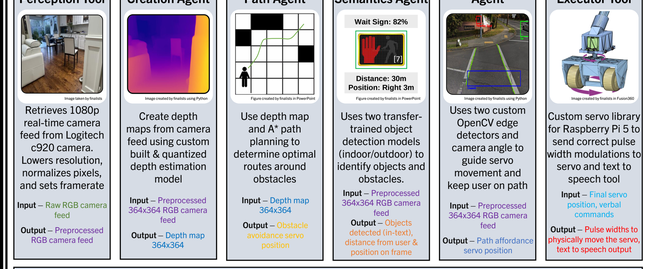


Figure created by finalists in PowerPoint

Contextual Reasoning Model

Receives:
- Objects detected
  - Object Distances
  - Position on Frame
- Path affordance servo position
- Obstacle avoidance servo position

Reasoning:
- Incorporates various servo potions from programs
- Applies reasoning and "sense"

Output:
- Final servo position
- Decision (Stop, Go)
- Key objects detected
- Verbal commands for user ("Open door")

Figure created by finalists in PowerPoint

IntelliCane Agentic Architecture

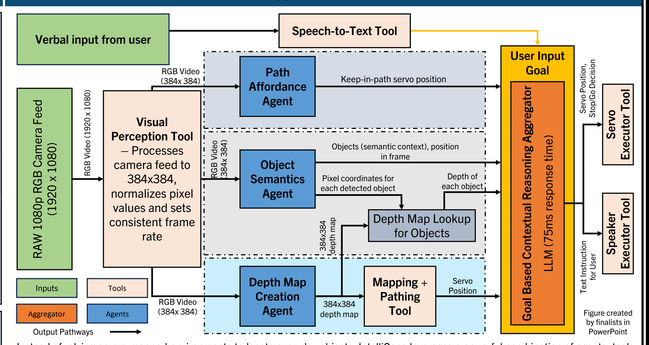
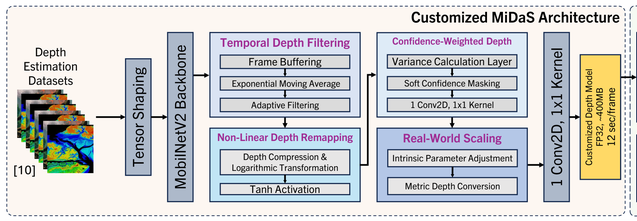


Figure created by finalists in PowerPoint

Instead of relying on pre-mapped environments to locate everyday objects, IntelliCane leverages a powerful combination of agents, tools and LLM-based reasoning to create real word movement directions and instructions. A user can say "go to a water fountain," and our framework will first attempt to directly detect a fountain using its onboard camera. If a fountain is not found, IntelliCane uses contextual reasoning to infer nearby alternatives—such as restrooms, which are commonly located near fountains—and navigates intelligently toward them.

Depth Map Creation Agent

Customized MiDaS Architecture



Quantized MiDaS

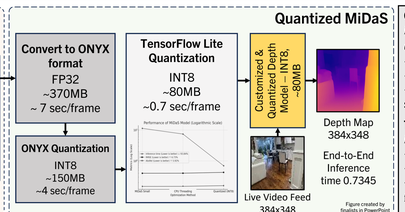


Figure created by finalists in PowerPoint

Confidence-Weighted Depth Maps:

$D_{conf}(x,y) = C(x,y) * D(x,y)$

$C(x,y) = e^{-\sigma \cdot V(x,y)}$

$- D(x,y)$ is the raw depth at pixel (x,y)

$- C(x,y)$ is the confidence weight

$- V(x,y)$ is the local variance of depth in a small window W *around* (x,y)

Temporal Depth Filtering:

$D_t(x,y) = \alpha D_t'(x,y) + (1-\alpha) D_{t-1}(x,y)$

$- D_t(x,y)$ is the smoothed depth at time t

$- D_t'(x,y)$ is the new raw depth estimate

$- D_{(t-1)}(x,y)$ is the previous frame's filtered depth

Non-Linear Depth Remapping:

$D_{re-map}(x,y) = \log(1 + kD(x,y))$

$- k$ is a scaling constant

Calibration Layer for Real-World Depth Scaling:

$D_{metric}(x,y) = \frac{f \cdot B}{D_{raw}(x,y)}$

$- f$ *is camera focal length*

$- B$ *is baseline distance*

Temporal Depth Filtering:
Frame Buffering Layer - Stores previous 3-5 frames of depth estimates
Exponential Moving Average (EMA) Smoothing - Adjusts current depth map based on past values Reduces flickering and instability in depth estimates
Adaptive Filtering - Uses neighbor-based filtering to reduce depth noise from fast movements

Non-Linear Depth Remapping
Depth Compression Function - Applies logarithmic or piecewise scaling:
- 0.3m - 5m range → stretched for higher resolution
- 5m+ range → compressed to reduce unnecessary details
Tanh Activation - Emphasizes nearby object depth details while compressing distant objects

Confidence-Weighted Depth Maps
Variance Calculation Layer:
- Assigns lower confidence to highly reflective or uniform surfaces based on local variance
Soft Confidence Masking - Multiplies depth values by confidence weights filtering out unreliable depth readings
Final Depth Adjustment (Convolutional Layer) - Smooths depth predictions by applying a confidence threshold

Calibration Layer for Real-World Depth Scaling
Intrinsic Parameter Adjustment - Uses camera intrinsic parameters (focal length, FOV, etc..) to convert raw depth into real-world distances
Metric Depth Conversion - Applies a conversion factor based on camera's intrinsic parameters

Mapping + Pathing Tool

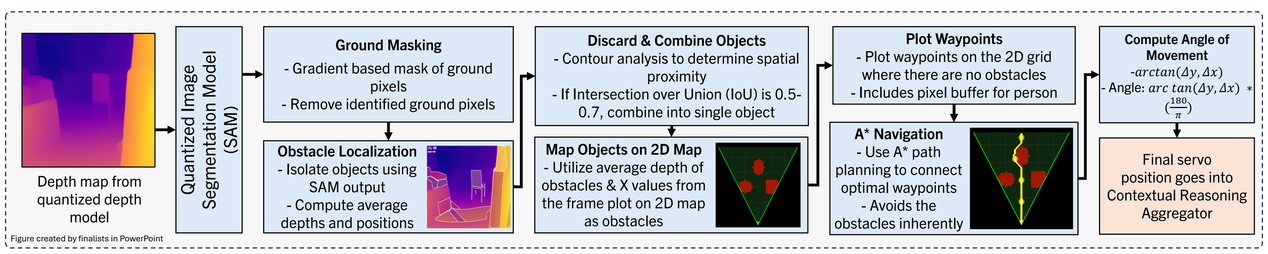


Figure created by finalists in PowerPoint

Object Semantics Agent

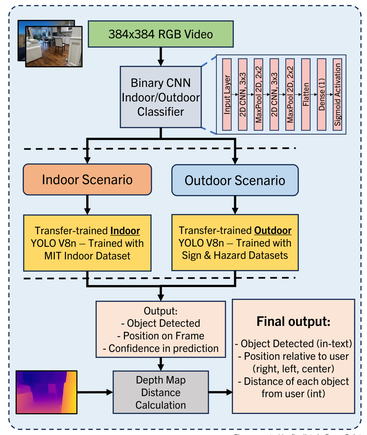
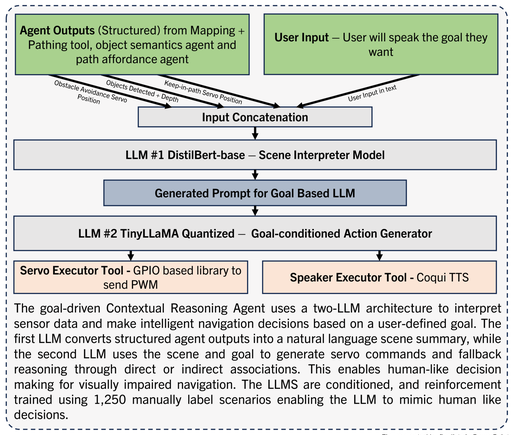


Figure created by finalists in PowerPoint

Contextual Reasoning Aggregator



The goal-driven Contextual Reasoning Agent uses a two-LLM architecture to interpret sensor data and make intelligent navigation decisions based on a user-defined goal. The first LLM converts structured agent outputs into a natural language scene summary, while the second LLM uses the scene and goal to generate servo commands and fallback reasoning through direct or indirect associations. This enables human-like decision making for visually impaired navigation. The LLMS are conditioned, and reinforcement trained using 1,250 manually label scenarios enabling the LLM to mimic human like decisions.

Figure created by finalists in PowerPoint

Path Affordance Agent

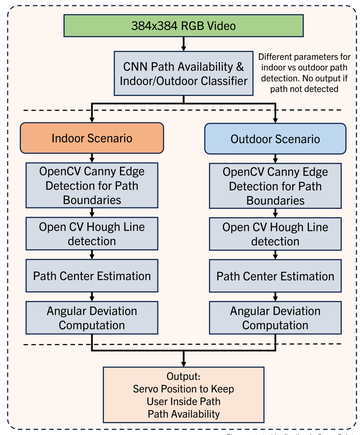


Figure created by finalists in PowerPoint

Servo Executor Tool

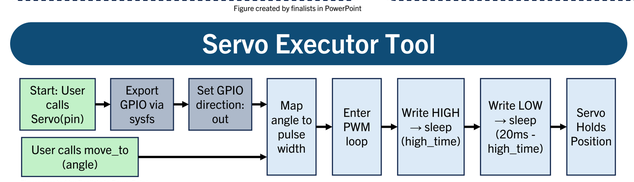


Figure created by finalists in PowerPoint

Speech-to-Text Tool

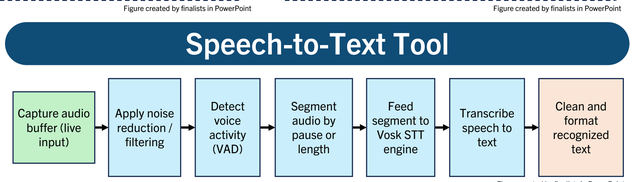
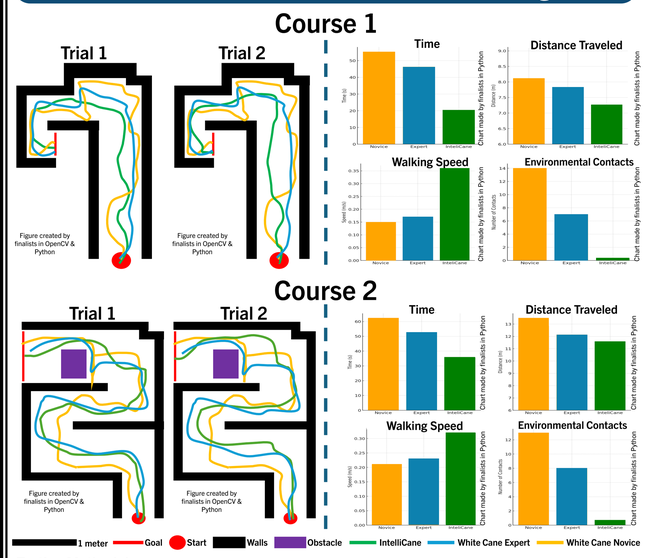


Figure created by finalists in PowerPoint

Results

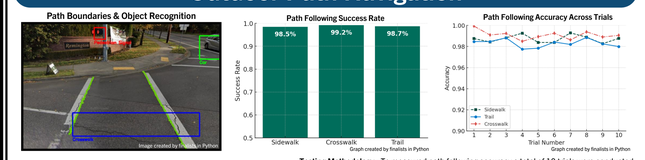
Indoor Obstacle Course & Path Following



Testing Methodology

- **Experimental Setup:** Indoor and outdoor navigation experiments were conducted using a drone positioned nearly perpendicular to the participant to track movement in real-time.
- **Data Processing:** Footage from the drone was processed using OpenCV, with distances scaled to real-world measurements for accuracy.
- **Participant Setup:** The course was undisclosed to participants, and all used a white cane identical to the base of the IntelliCane.
- **Novice vs. Expert Participants:** Novice participants were introduced to both canes without training, while expert participants were trained in two common white cane techniques: two-point touch and constant contact.
- **Mobility Metrics:** Key metrics recorded included distance traveled, duration, walking speed, and environmental contacts.
- **Environmental Contacts:** Contacts with surroundings were counted based on the number of times the cane touched anything other than the floor.

Outdoor Path Navigation

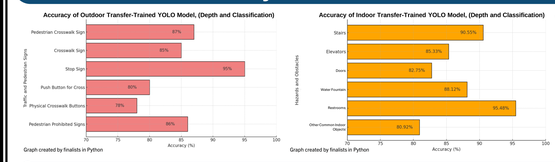


$$Keep-in-Path\ Accuracy = \frac{Total\ Time\ in\ Path}{Total\ Trial\ Time} * 100\%$$

Total Time in Path = Duration (seconds) the user remains inside the detected sidewalk/trail boundaries
Total Trial Time = Entire duration (seconds) of the walking trial

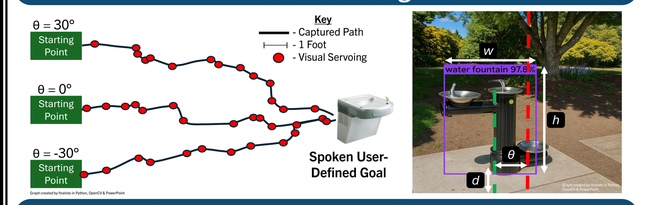
Testing Methodology - To measured path following accuracy a total of 10 trials were conducted per path type (sidewalk, crosswalk, trail). Each trial consisted of walking down a path or sidewalk in various locations for 2 minutes. The accuracy was measured on the number of times the user deviated from the trial. Testers were used multiple times for different trials in different locations. No blindfolds were used to ensure safety of the tester on the open road. The accuracy was not compared to a standard white cane because white canes lack this capability entirely. 5 trials were conducted to detect the accuracy of uneven paths.

Object Detection

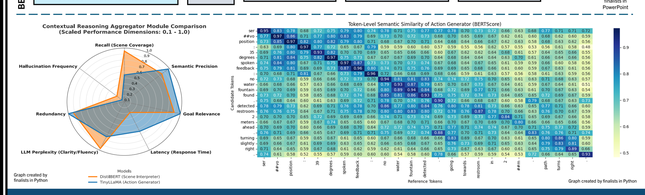


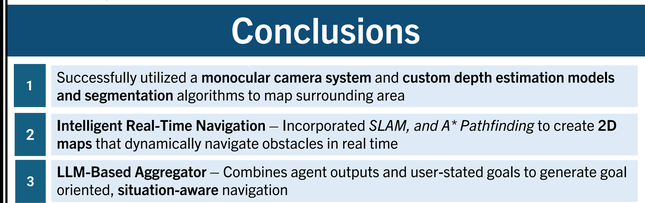
Our custom transferred learned You Only Look Once (YOLO) v8-nano models were trained and evaluated using various public & online datasets including the MIT indoor dataset and sign datasets. The accuracy shown is the average depth and classification accuracy when tested with 500 samples from each class.

Visual Servoing



Aggregator Statistical Evaluation





Conclusions

1. Successfully utilized a **monocular camera system** and **custom depth estimation models and segmentation** algorithms to map surrounding area
2. **Intelligent Real-Time Navigation** – Incorporated *SLAM, and A* Pathfinding* to create **2D maps** that dynamically navigate obstacles in real time
3. **LLM-Based Aggregator** – Combines agent outputs and user-stated goals to generate goal oriented, **situation-aware** navigation
4. **Optimization for Raspberry Pi** – Agents were quantized for **efficient, low-power processing**, enabling real-time navigation on a compact & affordable platform
5. **Extensible Framework** – Can easily support additional features by adding expert agents (e.g. waypoint navigation)

Future Work

1. Scale out solution making it viable for **clinical trials** and use in the real world
2. **Integrate GPS** to enhance waypoint navigation and outdoor localization
3. Use a reasoning **LLM to enhance "Train of Thought"** to support more complex goals with the ability to support more situational awareness

Impact

1. Enables the visually impaired to **navigate independently, safely, and quickly,** significantly improving quality of life.
2. **Low cost** and simplicity of hardware design **enables wide accessibility**
3. Enhances the situational awareness of the visual impaired user while providing **real-time active guidance** that adapts to the **environment**

FAVORITE MOMENT WITH THE PROJECT!

Our favorite moment was watching visually impaired individuals at the Oregon Blind Commission confidently navigate with IntelliCane. Seeing their independence and gratitude was very fulfilling and gave us confidence that we are on our way to bringing this solution to the real world!

References

[1] Global estimates of vision loss. (2021, April 9). The International Agency for the Prevention of Blindness. https://www.iapb.org/learn/vision-atlas/magnitude-and-projections/global/
[2] Bleau, M., Paré, S., Djerounou, I., Chebat, D. R., Kupers, R., & Ptito, M. (2021). Blindness and the reliability of downwards sensors to avoid obstacles: A study with the eyecane. Sensors, 21(8). https://doi.org/10.3390/s21082700
[3] Nevada31. (n.d.). Guide dog sign. iStock. Retrieved April 17, 2025, from https://www.istockphoto.com/illustrations/guide-dog
[4] Media, P. (n.d.). Headbeams, bursary awards and more. Powerd Media. Retrieved April 17, 2025, from https://powerd.media/audio/headbeams-bursary-awards-and-more
[5] About us – WeWALK smart cane. (n.d.). Retrieved April 17, 2025, from https://wewalk.io/en/about/
[6] Medline white cane for blind/vision impaired 1ct. (n.d.). Medline At Home. Retrieved April 17, 2025, from https://athome.medline.com/en/medline-white-cane-for-blind-vision-impaired-1ct-mdswhtcane
[7] Homepage. (2023, September 27). Creative Commons. https://creativecommons.org/
[8] Person top down view stock illustrations – 728 person top down view stock illustrations, vectors & clipart. (n.d.). Dreamstime. Retrieved April 17, 2025, from https://www.dreamstime.com/illustration/person-top-down-view.html
[9] Google maps. (n.d.). Google Maps. Retrieved April 17, 2025, from https://www.google.com/maps?entry=wc
[10] MiDaS. (n.d.). PyTorch. Retrieved April 17, 2025, from https://pytorch.org/hub/intelisl_midas_v2/
[11] Hematteja, A. V. N. M. (n.d.). Traffic sign dataset - Classification. Kaggle. Retrieved March 5, 2025, from https://www.kaggle.com/datasets/ahmetejal19bec1025/traffic-sign-dataset-classification
[12] Labs, D. (n.d.). Stains image dataset. Kaggle. Retrieved March 5, 2025, from https://www.kaggle.com/datasets/datacluster/stains-image-dataset
[13] Larxel. (n.d.). Road sign detection. Kaggle. Retrieved March 5, 2025, from https://www.kaggle.com/datasets/andrewmvd/road-sign-detection
[14] Mahesa, M. R. (n.d.). Image obstacle in public spaces. Kaggle. Retrieved March 5, 2025, from https://www.kaggle.com/datasets/muffinextumahesa/image
[15] Tuan, Ali. (n.d.). Traffic-Signs-Dataset. Kaggle. Retrieved March 5, 2025, from https://www.kaggle.com/datasets/tuanai/traffic-signs-dataset
[16] COCO Dataset. (n.d.). COCO Common Objects in Context. Retrieved March 5, 2025, from https://cocodataset.org/
[17] Google Open Images Dataset. (n.d.). Google Research. Retrieved March 5, 2025, from https://storage.googleapis.com/openimages/web/index.html
[18] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2021). Vision transformers for dense prediction. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 12179-12188.
[19] Ranftl, R., Bochkovskiy, A., & Koltun, V. (2020). MiDaS: Robust monocular depth estimation in diverse environments. *arXiv preprint arXiv:2006.06802*.
[20] Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems (NeurIPS)*.
[21] Godard, C., Aodha, O. M., Firman, M., & Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
[22] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
[23] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
[24] Jocher, G., Chaurasia, A., Qiu, J., & Stoken, A. (2023). YOLOv8. Ultralytics YOLO for real-time object detection. Ultralytics Technical Report.
[25] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100-107.
[26] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1, 269–271.
[27] Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. MIT Press.