

PEDESTRIAN VISION

PJ GRANIERI

RICKY ZHAO

MANELIN JOHN RAJAN

ANTHONY SPADAFORE

ZACHARY MCPHERSON

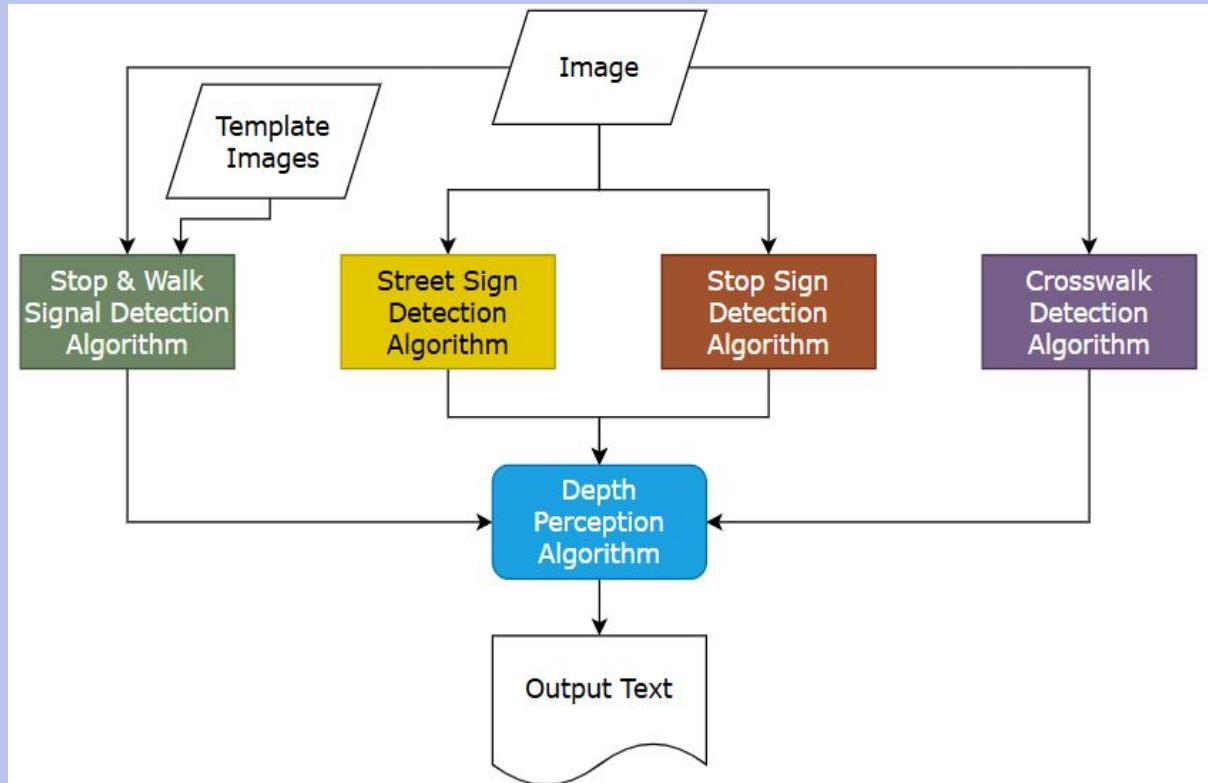
FALL 2025

PROJECT OVERVIEW

The purpose of this program is to aid visually impaired individuals as they traverse through urban environments in their daily lives.

Using an image of an individual's forward visual field, our program...

- 1) detects crucial information for safe and accurate transportation
- 2) computes the relative distance to each detected object
- 3) narrates the overall physical scenario



WALK SIGNAL DETECTION



ALGORITHM OVERVIEW & RESULTS

Detects the presence of a walk or stop signal.

1. Read image (HSV) and predefined templates (Binary).
2. Perform "red" thresholding, isolating "stop."
3. Perform "white" thresholding, isolating "walk."
4. Execute multi-scale template matching.
5. If the largest found correlation coefficient (r) is greater than the set threshold, the object was successfully detected.
6. Return a mask of the location of found signal.



STREET SIGN DETECTION



ALGORITHM OVERVIEW & RESULTS

Detects the presence of street signs and their contents.

1. Read image (HSV).
2. Detect blue regions using adaptive thresholding.
3. Apply morphological operations to clean mask.
4. Extract contours and filter candidates by size and aspect ratio.
5. Perform OCR on each blue region using EasyOCR.
 - a. Apply CLAHE enhancement if needed.
6. Clean and validate text using street suffix dict.
7. Merge nearby fragments into street names.
 - a. Combine complementary pairs
(name + suffix).



STOP SIGN DETECTION



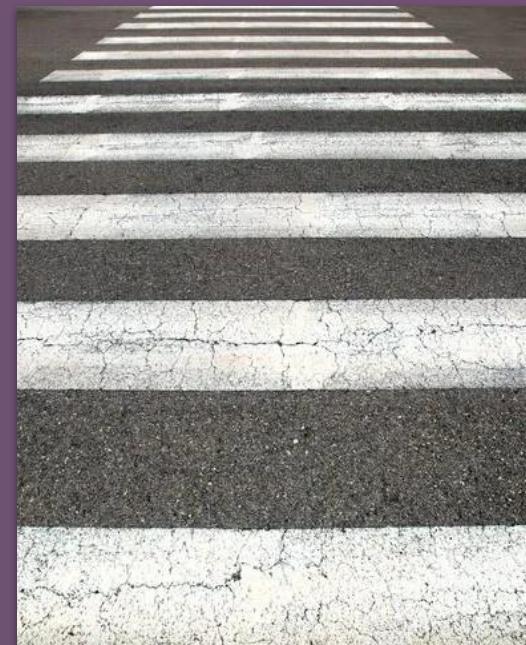
ALGORITHM OVERVIEW & RESULTS

Detects the presence of stop signs.

1. Read image (HSV).
2. Detect all red to create a binary mask.
3. Search for contours that match an octagon via various filters.
4. Define a region of interest (ROI) for each octagon that is found.
5. Uses EasyOCR to search for the text "STOP" within each ROI.
6. Returns a mask of the bounding box to be passed into the depth perception module.



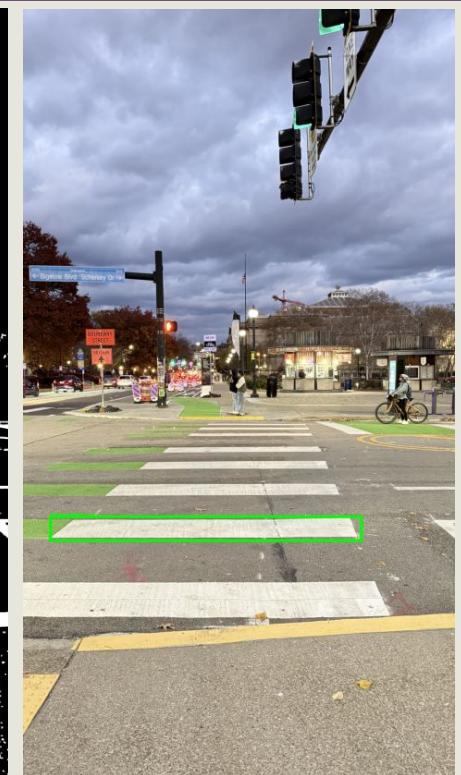
CROSSWALK DETECTION



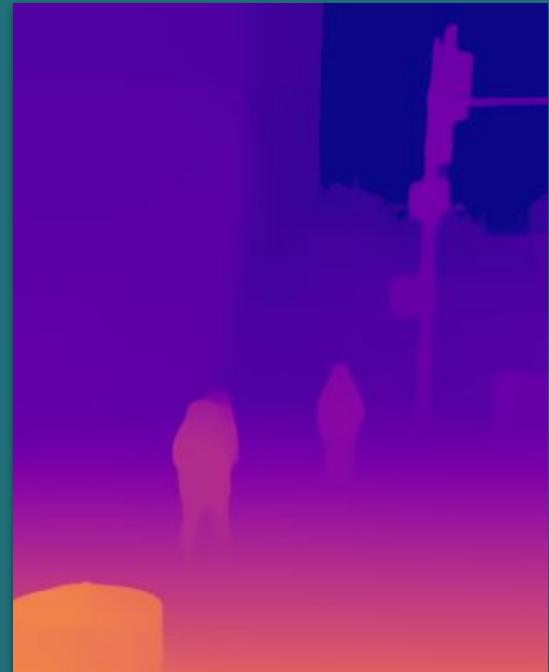
ALGORITHM OVERVIEW & RESULTS

Detects the presence of a crosswalk.

1. Read image (Grayscale).
2. Perform thresholding to isolate all bright white road markings.
3. Find all rectangles through contour analysis.
4. Filter rectangles by their horizontal aspect ratio and length.
5. Locate the rectangle with the largest area.
6. Return a mask highlighting the location of found rectangle.



DEPTH PERCEPTION



ALGORITHM OVERVIEW & RESULTS

Computes the depth estimation across the image using MiDaS.

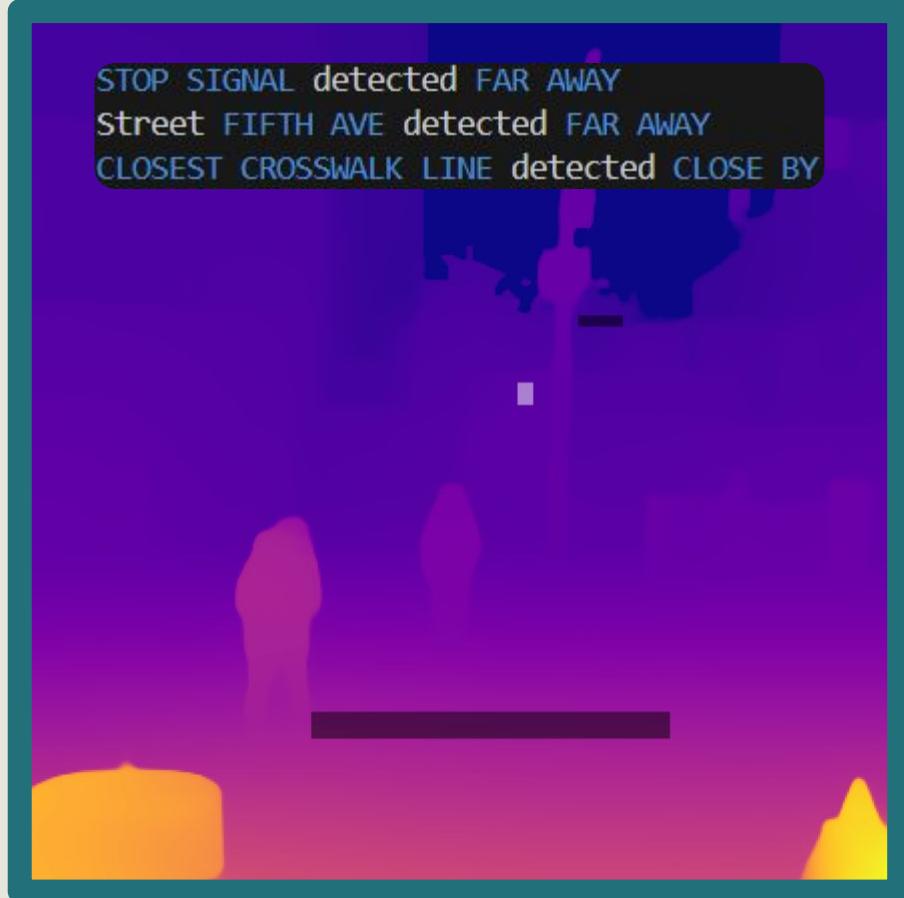
1. Read Image (RGB) and masks from the other modules.
2. Find depth estimation across the image.
3. Calculate the average depth estimation within each mask.
 - a. Returns a value between [0, 1].
 - b. Not a representation of the real-world distance.
4. Using a set threshold value, the average depth estimation is determined to be CLOSE BY or FAR AWAY.



WALK SIGNAL detected CLOSE BY (0.38797)
STREET SIGN detected FAR AWAY (0.24842)

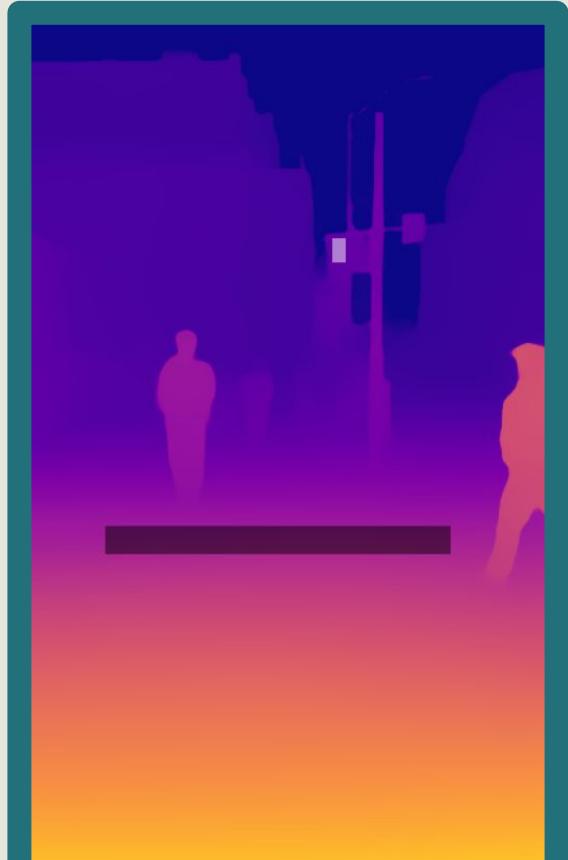
TEST #1





TEST #2

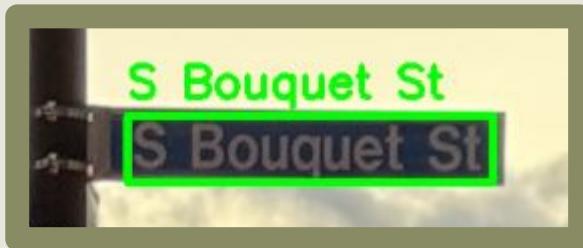




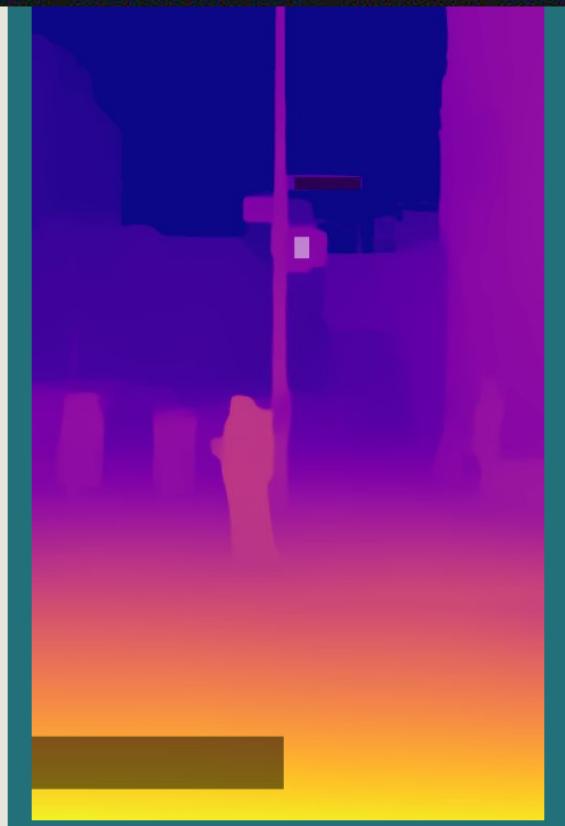
WALK SIGNAL detected FAR AWAY
CLOSEST CROSSWALK LINE detected CLOSE BY

TEST #3



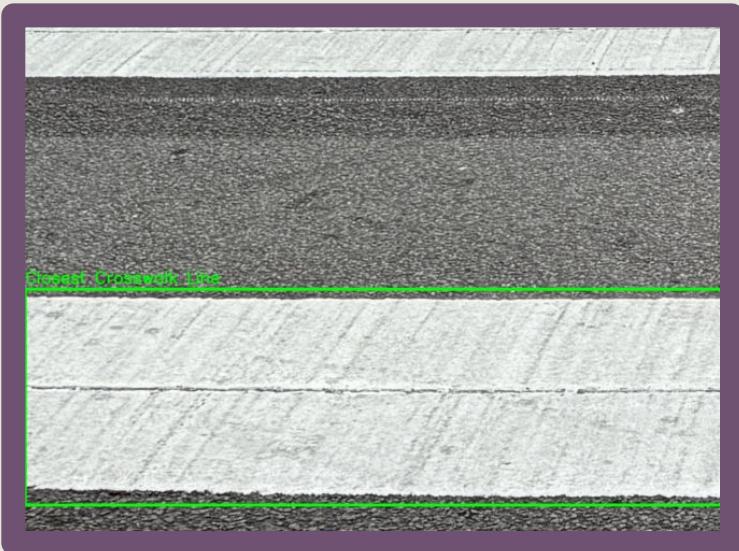
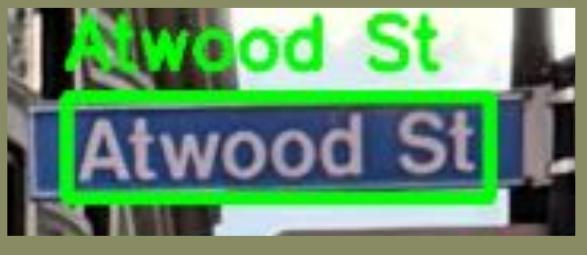


STOP SIGNAL detected CLOSE BY
Street S BOUQUET ST detected FAR AWAY
CLOSEST CROSSWALK LINE detected CLOSE BY

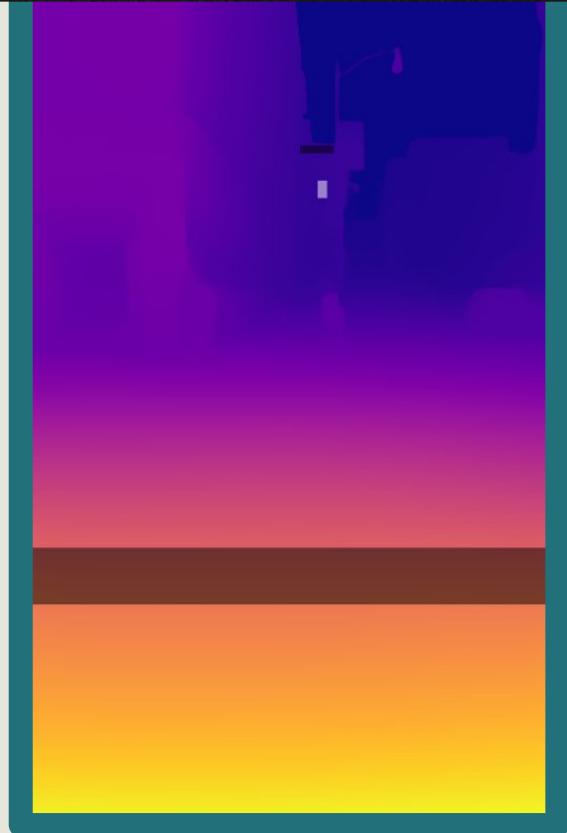


TEST #4





WALK SIGNAL detected FAR AWAY
Street ATWOOD ST detected FAR AWAY
CLOSEST CROSSWALK LINE detected CLOSE BY



TEST #5





STOP SIGN detected CLOSE BY
CLOSEST CROSSWALK LINE detected FAR AWAY

CHALLENGES AND FUTURE PROGRESSION

While our program was an overall success, we still ran into some issues...

ISSUES	POTENTIAL SOLUTIONS
Color thresholding methods varied in performance based on the time of day.	Adaptive thresholding using EXIF metadata and predefined color thresholds (needs a larger dataset).
MiDaS does not compute real-world depth information, just a relative estimation of how far away objects in the frame are from the camera.	Use a video feed with structure-from-motion techniques or a stereo camera setup to get real-world depth information.
Test cases with too much noise affected the accuracy of the detection. Some results are not entirely ideal.	Fine tune thresholding and filtering; use other detected objects to crop initial image.
Template matching algorithm took a relatively long time to execute, effectively limiting framerate.	Develop more efficient filtering methods, reducing reliability on template matching.

**THANK
YOU**