THE CONTRIBUTION OF OBJECT DETECTORS AND VLMS IN EXPLAINABLE SOCIAL NAVIGATION

AAU Robotics mini Project November 2025

Our Team

AAU Robotics
Msc University
Student

Daniel Dharampal

AAU Robotics
Msc University
Student

Endri Dibra

AAU Robotics
Msc University
Student

Nicklas Deding

CONTENT

1 General Overview

2 Object Detectors

3 Visual Language Models

4 Path Planning

5 Implementation

CONTENT

6 Research topic : SmolVLA

7 Challenges

8 Limits

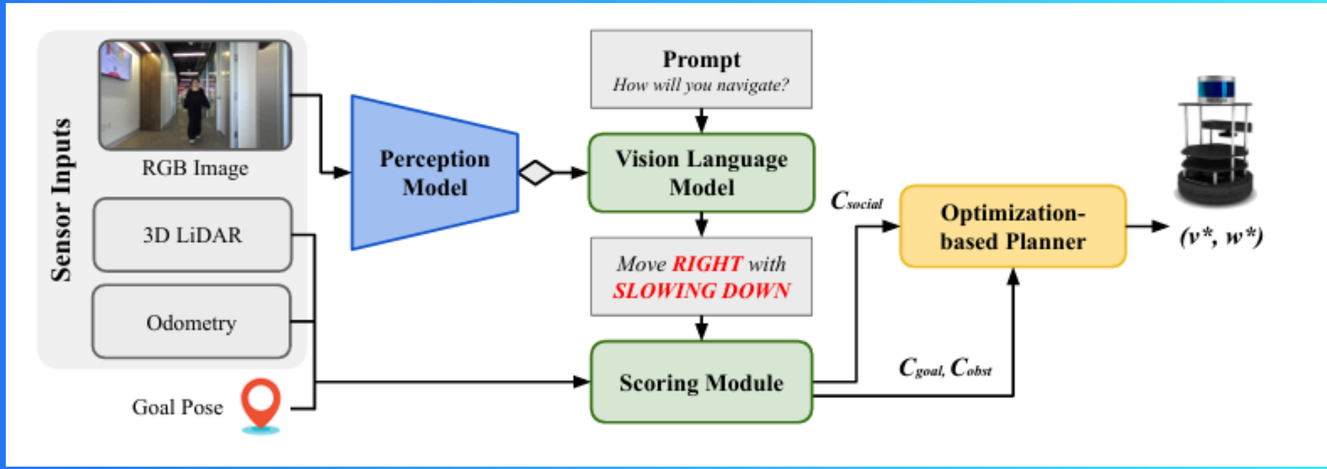
9 Conclusion

10 Future Scope

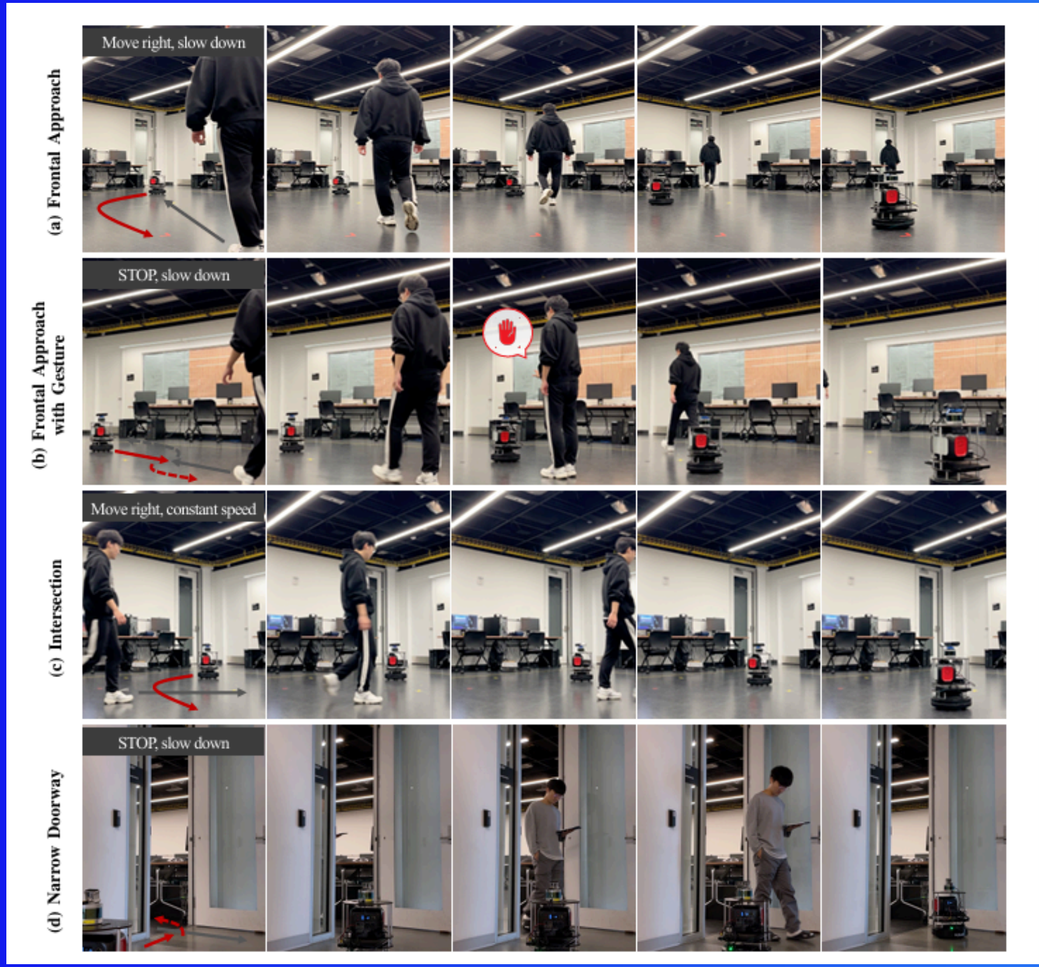
GENERAL OVERVIEW

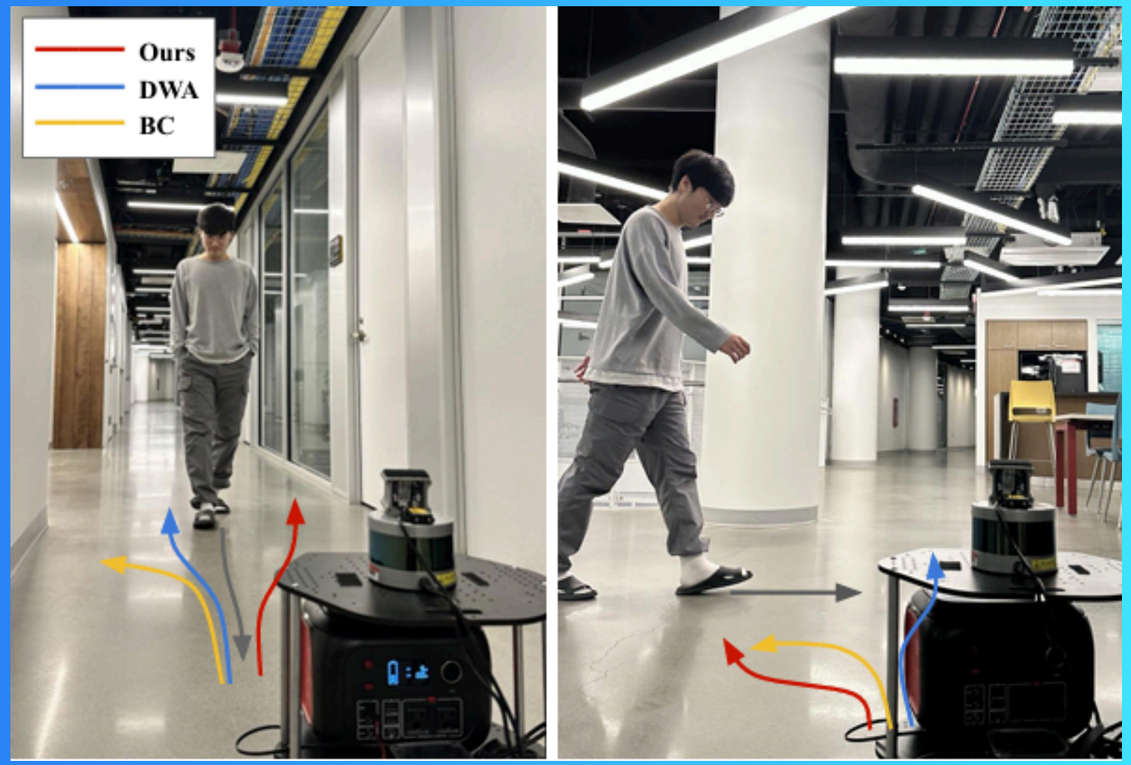
GENERAL OVERVIEW [1]

- Our assigned paper to read and implement was: VLM-Social-Nav paper
- They use a real mobile robot [TurtleBot] in dynamic real environments
- They use Object Detectors to detect social entities such as people, obstacles, possible gestures, doors and groups
- They treat the VLM [GPT-4V] as a scoring module that evaluates scene elements or candidate actions and outputs social cost information that is integrated into the path planner
- They construct a social cost term and fuses it with geometric costs [distance, collision] before planning



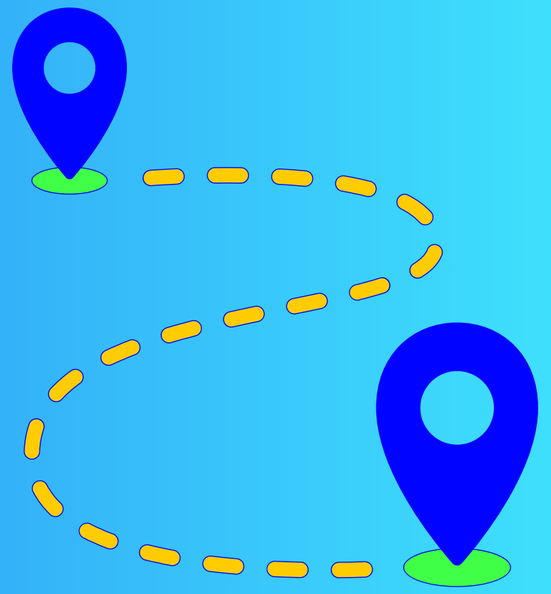
GENERAL OVERVIEW [2]





GENERAL OVERVIEW [3]

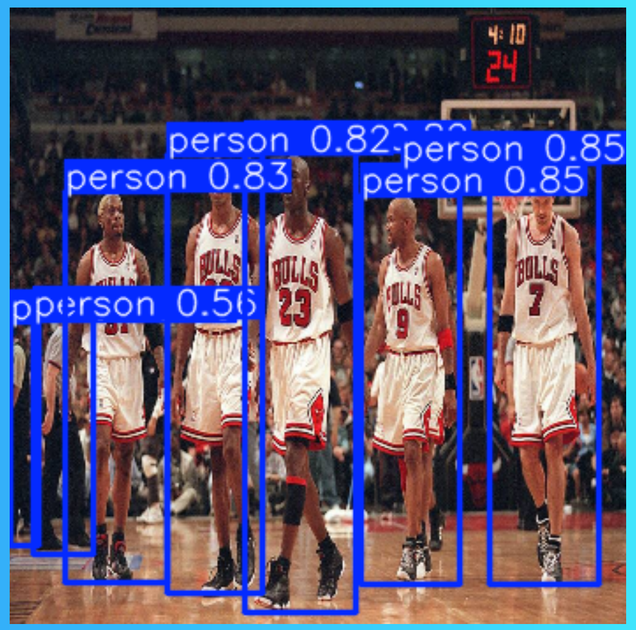
- We use 2D static images for testing our algorithm. We run our object detector to detect people and obstacles and place them into bounding boxes, crop each bounding box, then send each crop to a VLM to request a per-object social threat score
- the VLM is used in two ways: 1st The whole image is prompted to return four quadrant scores, and 2nd each detected object crop is prompted for a threat score. We add the social scores into the geometric cost map
- We create a cost map from distance transform (inverse distance to obstacle) and add socialWeight * socialCost, [where social cost come from blended quadrant + per-object Gaussian], to that cost map. We also have curvature and turn penalties in A* and smoothing

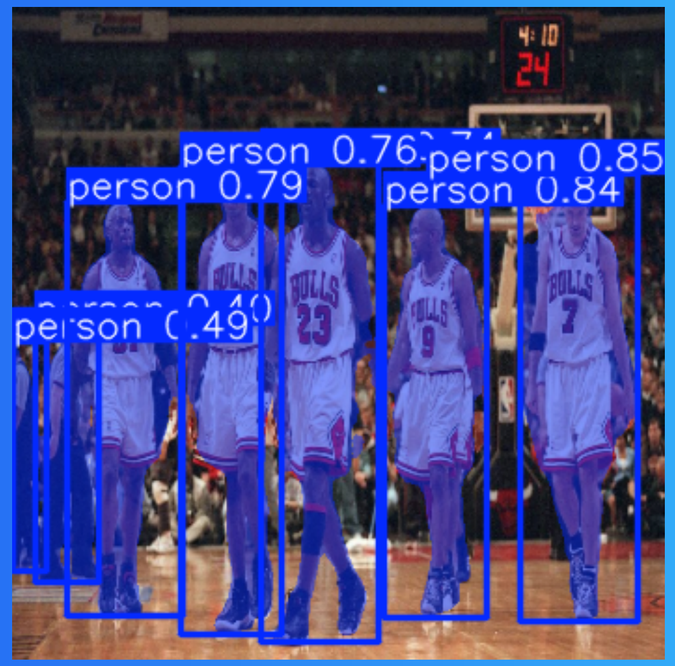


ROBOTIC PERCEPTION

OBJECT DETECTORS

- We make use of YOLOv11n developed by Ultralytics, for object detection and recognition purposes
- It is lightweight
- It performs in real-time cases
- It is mostly correct on its predictions regarding people and objects
- It can be used for segmentation purposes as well



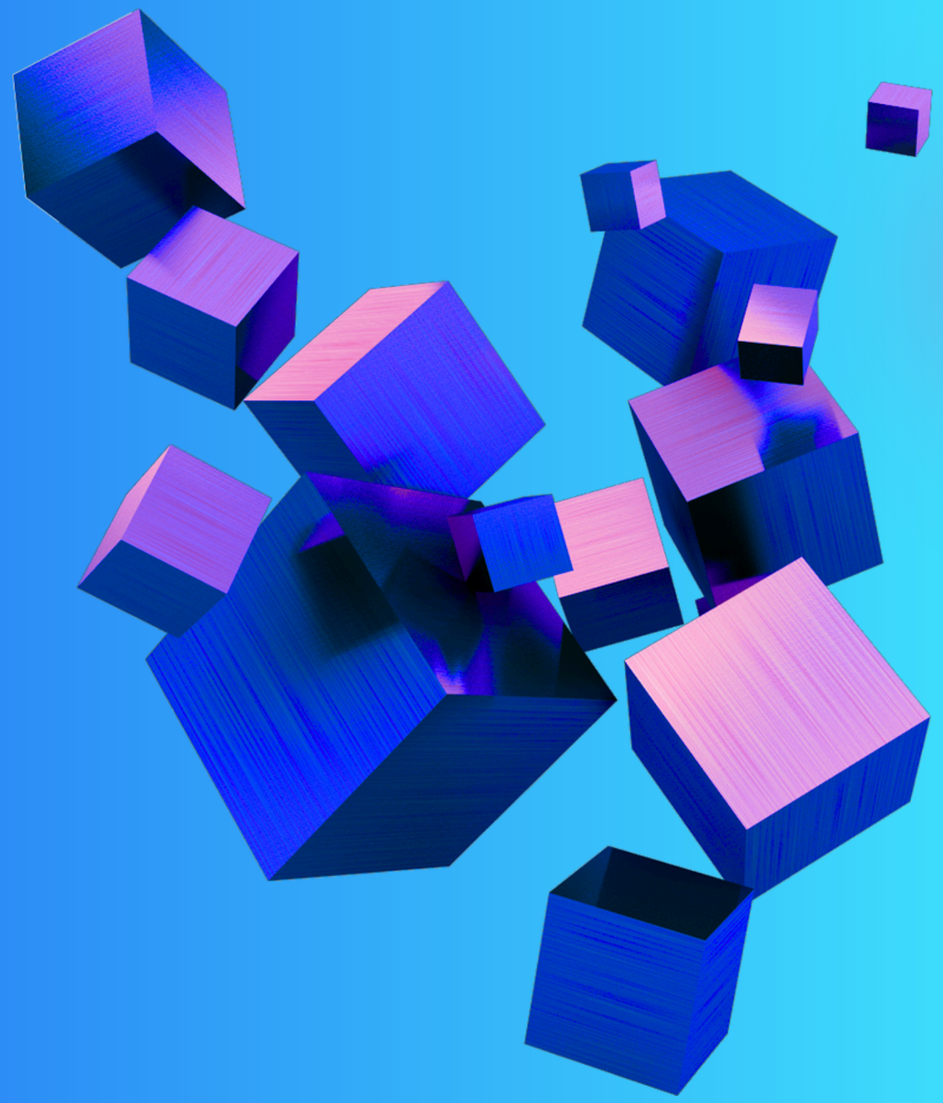


VISUAL LANGUAGE MODELS

- The VLM used for this project is smolVLM - Instruct provided and developed by Hugging face
- It has a size of 1.7 Billion parameters
- It can be used for multimodal (image + text) tasks where the input comprises text queries along with one or more images
- It is able to answer questions about images and describe visual content
- For better performance it loads and runs the model in half-precision [torch.float16 or torch.bfloat16]

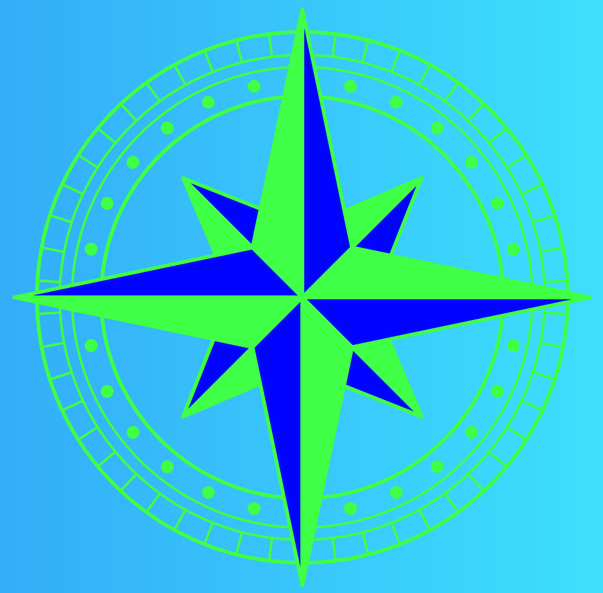
PATH

PLANNING



PATH PLANNING

- We approach path planning using A* [A Star] algorithm, for the reason that it is a decent choice in cases like 2D static images, where the Start and Goal positions are known and there are static obstacles as well, not dynamic ones
- It behaves like a global planner
- Guarantees optimality, because it is able to find the optimal [least-cost] path, when its heuristic never overestimates the true cost
- **f(n) = g(n) + h(n)**
- **f(n)**, Estimates total cost of the cheapest solution path passing through n [nodes]
- **g(n)**, Cost from the start node to n [node]
- **h(n)**, Heuristic estimate of the cost from n [node] to the goal



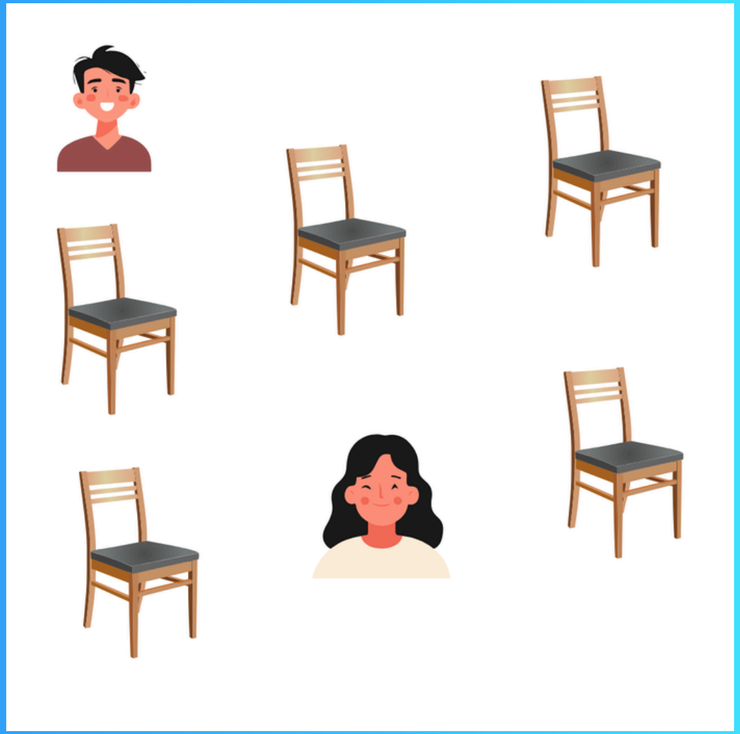
IMPLEMENTATION

IMPLEMENTATION [1]

- We will present the implementation of our work in a flow of steps, explaining in a simple way our process and approach

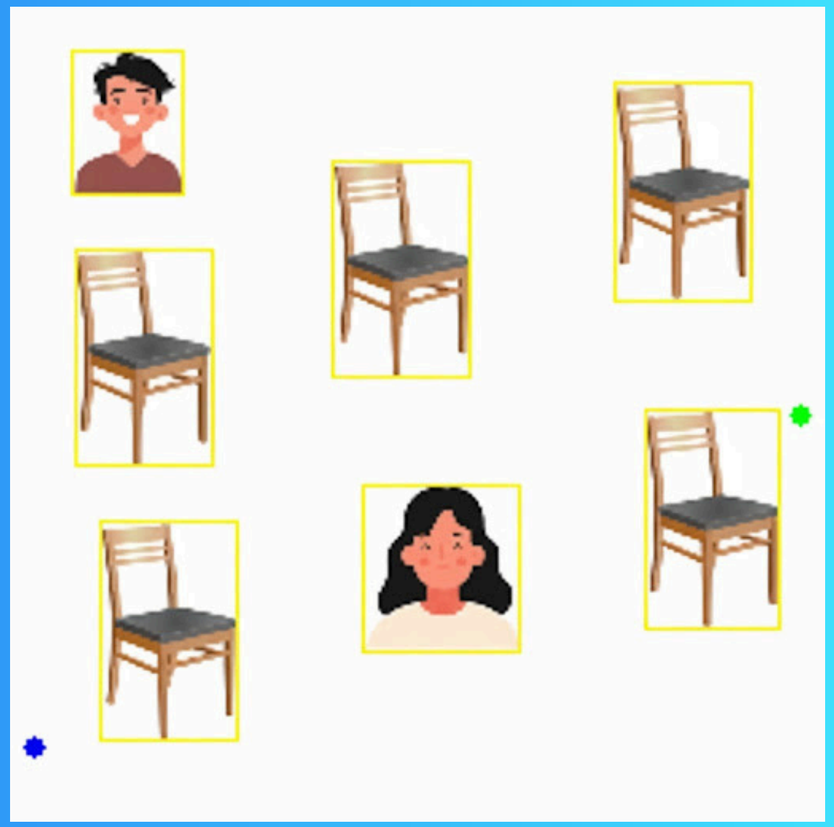
- First we have our original image, with no background noise, consisted of people and obstacles, to make it difficult for the A* algorithm to find an optimal path

Original Image



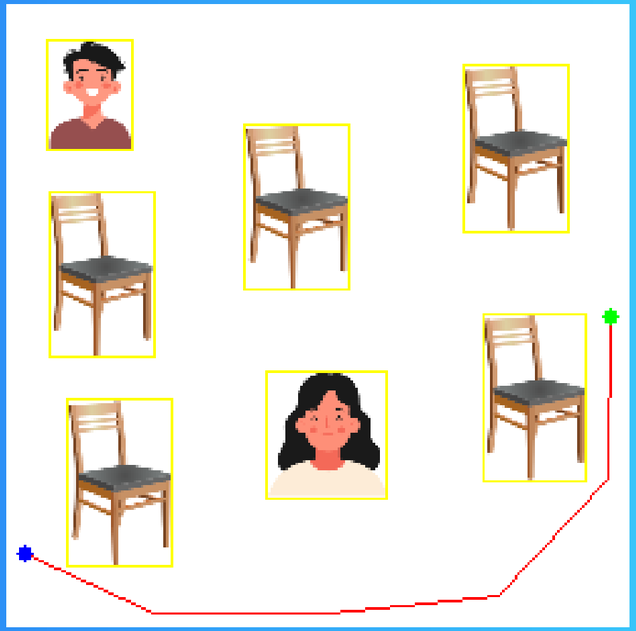
IMPLEMENTATION [2]

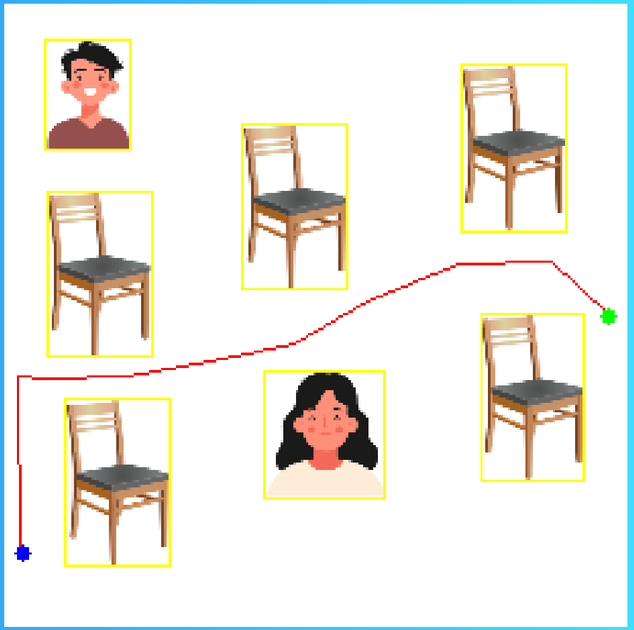
- After that, we set up the Start [Green Dot] and Goal [Blue Dot] positions
- Then, we apply people and obstacle detection through YOLOv11n model, placing them into yellow bounding boxes
- The 4 walls, people and obstacles are inflated with high cost [a restrictive radius], so that the path will not touch them. People have higher cost for safety
- People are multiplied with 1.0 while other obstacles with 0.3
- A* has penalties for curve and turn cases, to produce a smoother path, which is pruned as well



IMPLEMENTATION [3]

- smolVLM steps in to provide a risk score for each feature on the image from 0 .0 to 1.0 as well as evaluating for safety the four quadrants of the image: top-left, top-right, bottom-left, bottom-right, from 0 to 1 too. This is our social cost
- Cost Map = 0.6 * Social Cost + 0.4 Geometric Costs
- Now is where A* algorithms and its cost map take place and start their application and sanctions
- A* should find an optimal path without colliding with the features on the image, respecting social norms and choosing the shortest path to the goal

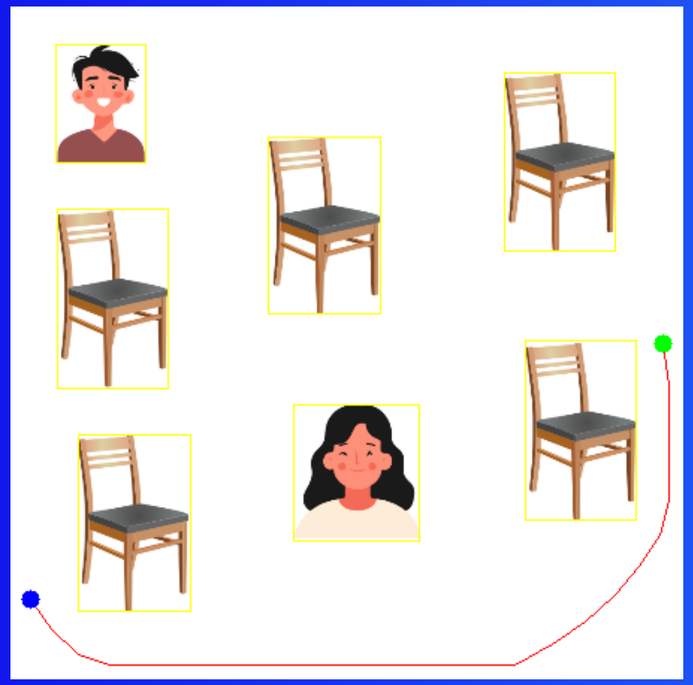




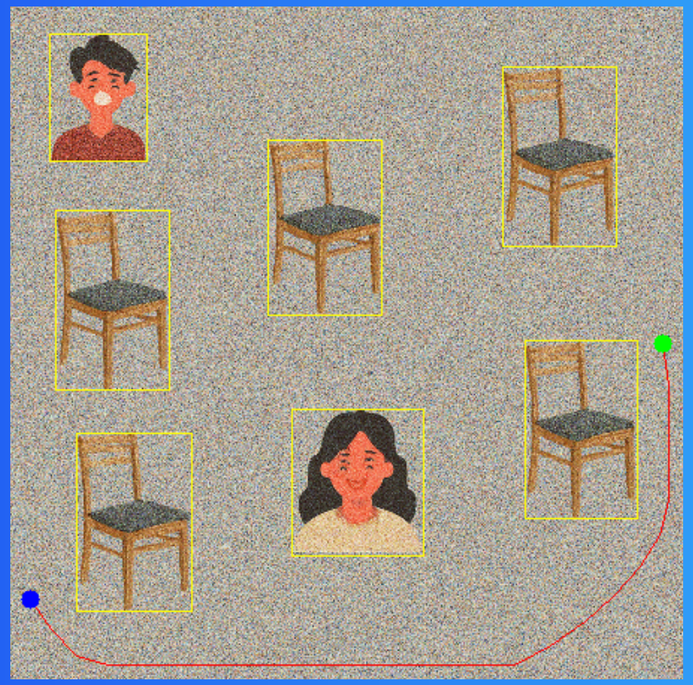
IMPLEMENTATION [4]

- Testing the algorithms in light and heavy background noises, with **450 x 450** pixels resolution

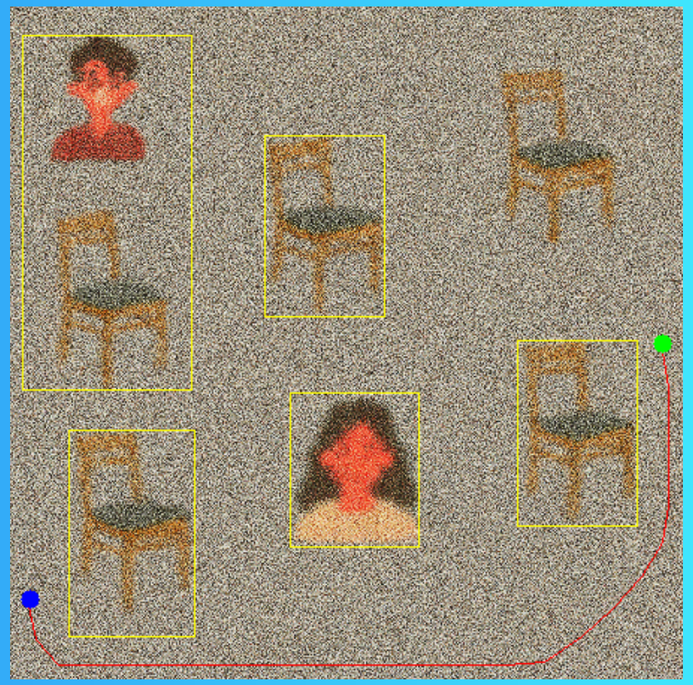
Original Image



Lighter Noise



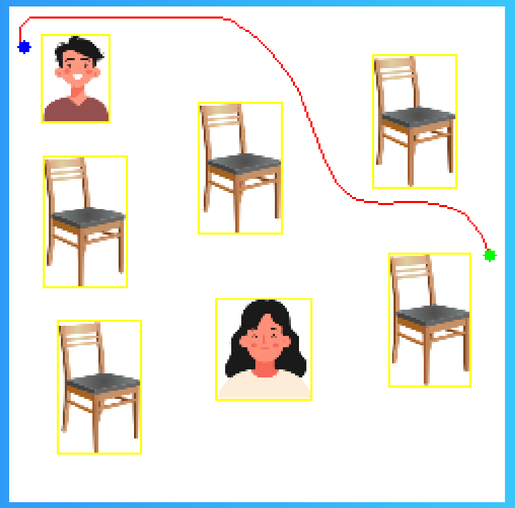
Heavier Noise



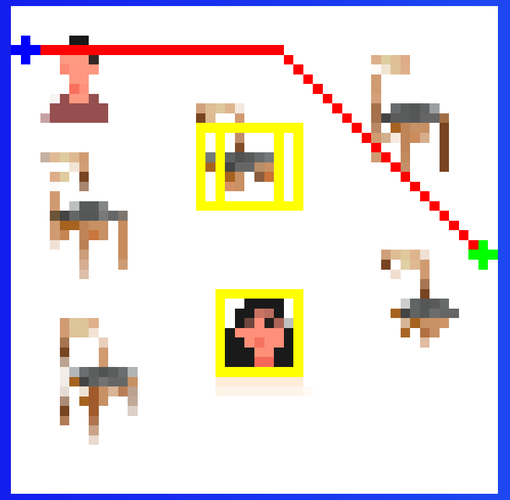
IMPLEMENTATION [5]

- Testing the algorithms in original, light and heavy background noises, but with different size resolutions as well from 50 x 50 pixels to 250 x 250 pixels

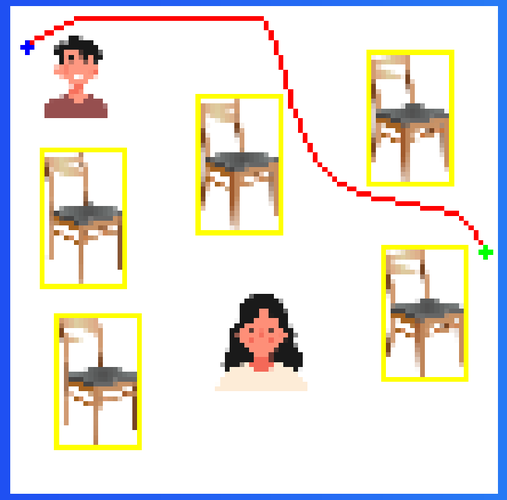
250 x 250



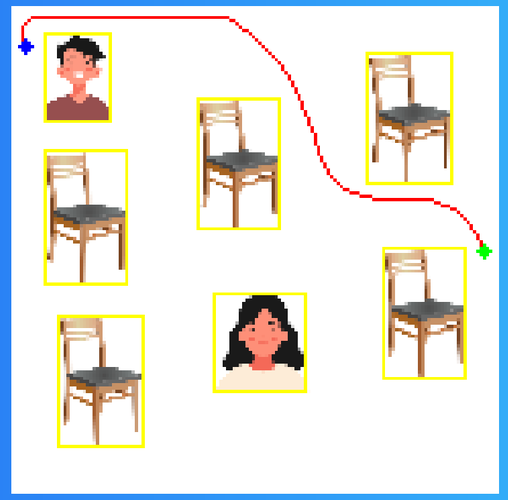
50 x 50



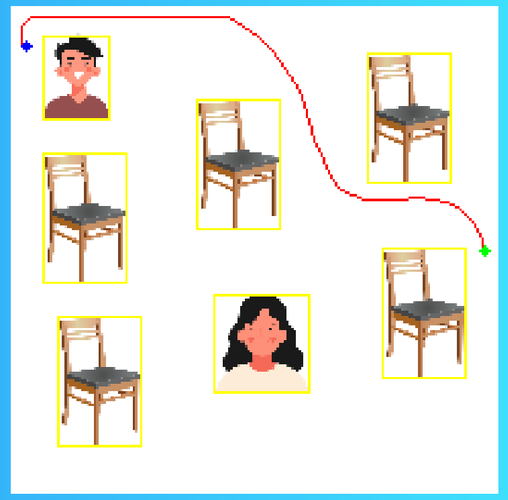
100 x 100



150 x 150

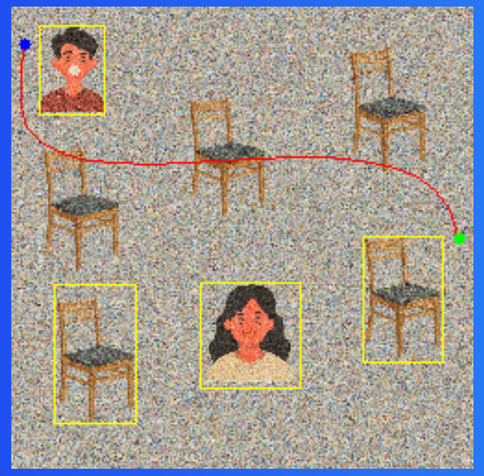


200 x 200

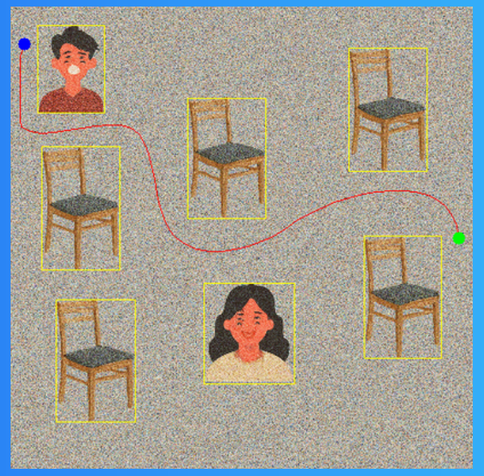


IMPLEMENTATION [6]

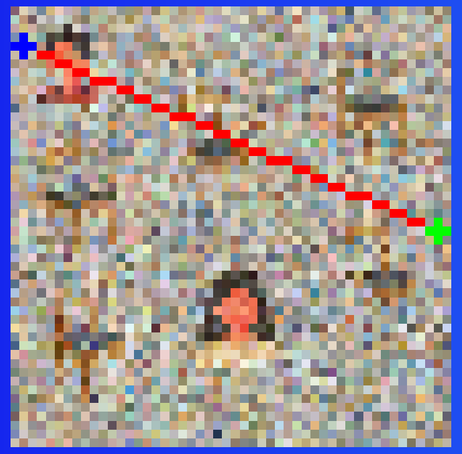
250 x 250



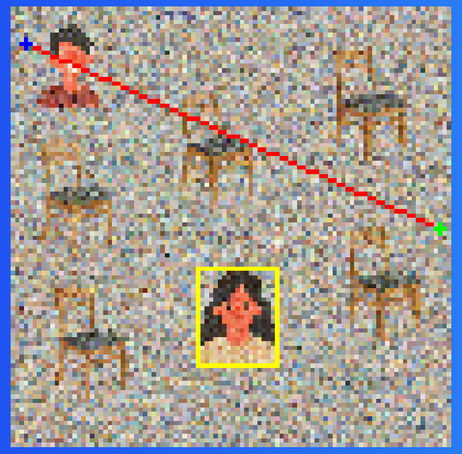
450 x 450



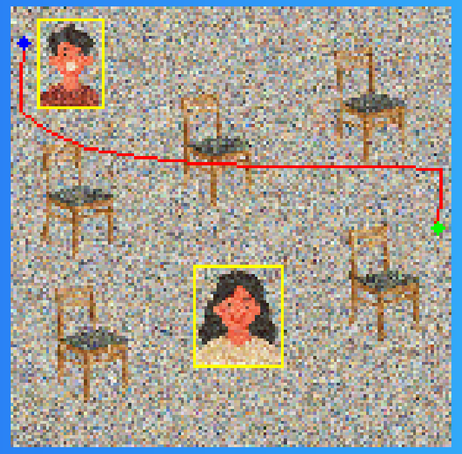
50 x 50



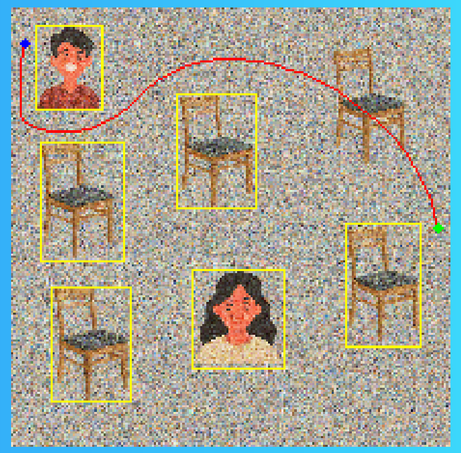
100 x 100



150 x 150



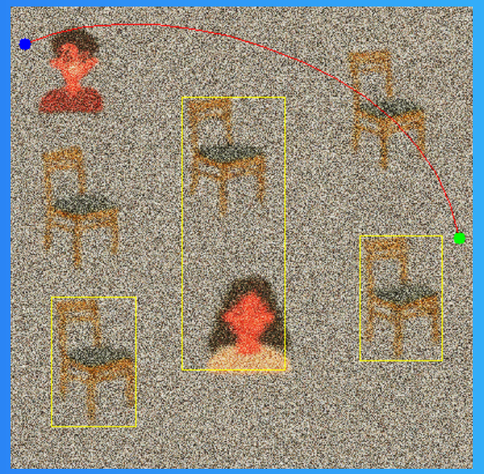
200 x 200



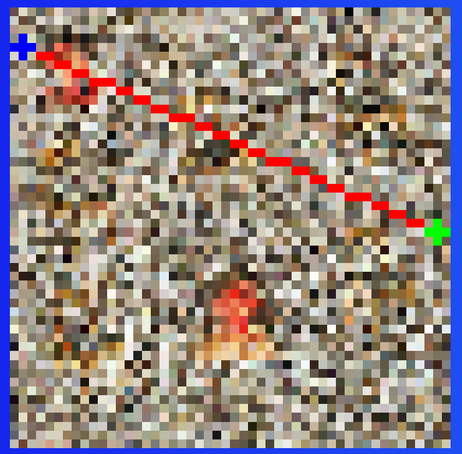
IMPLEMENTATION [7]

250 x 250

450 x 450

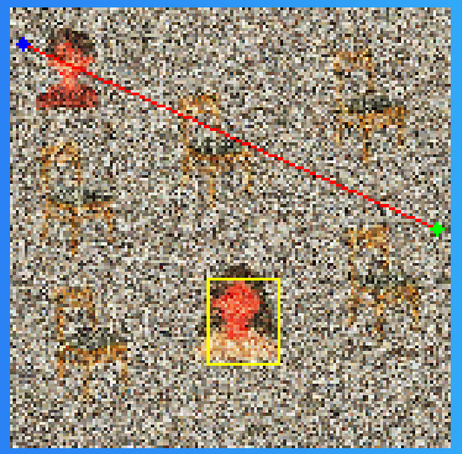


50 x 50

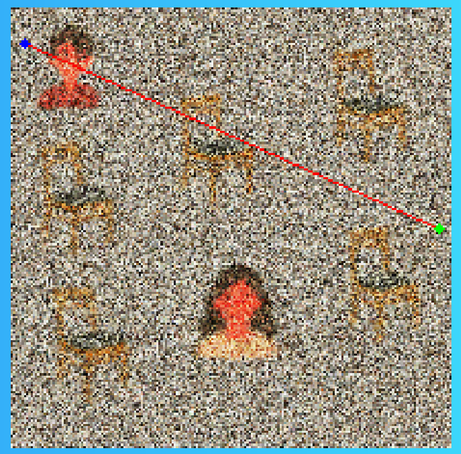


100 x 100

150 x 150



200 x 200

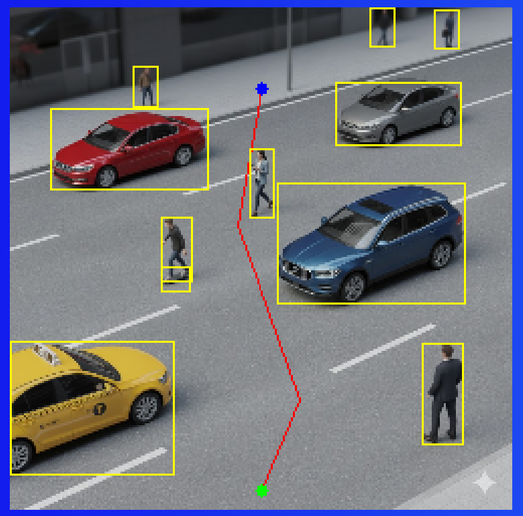


IMPLEMENTATION [8]

- Testing the algorithms in different background scenes,

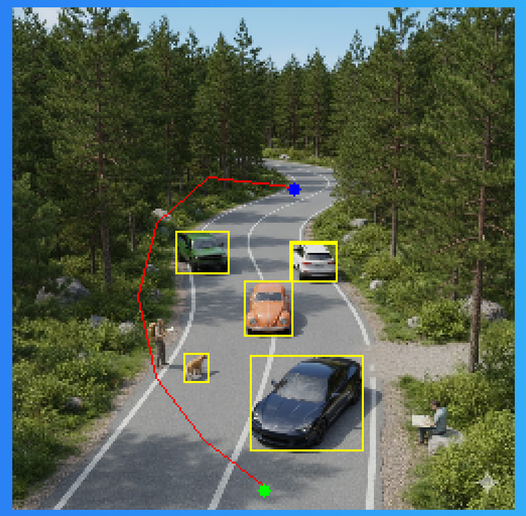
  to evaluate their robustness and generalization

Urban Road

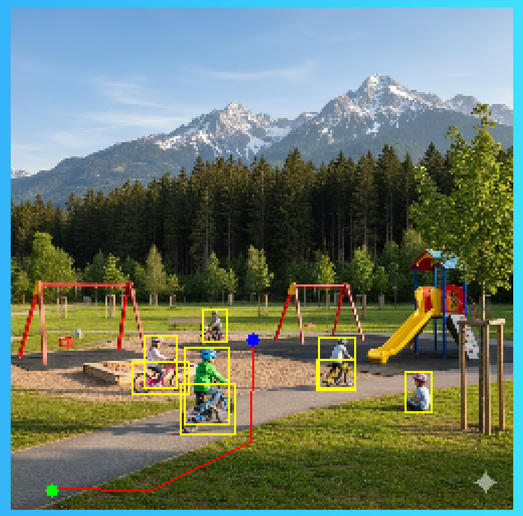


Laboratory

Nature



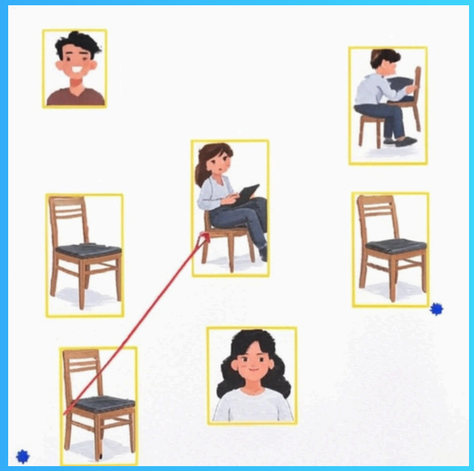
Playground



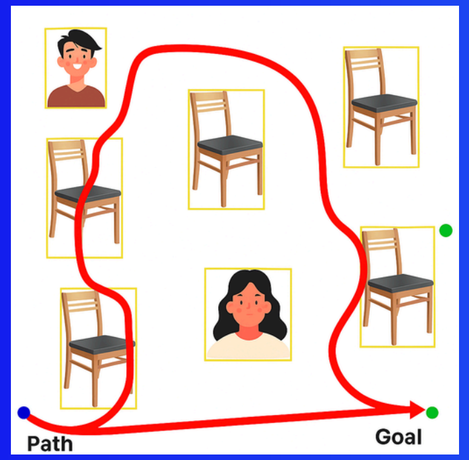
IMPLEMENTATION [9]

- Finally we compare our results with other LLMs like ChatGPT-5, Gemini-2.5-Flash, Grok-4, Claude-Sonnet-4, to evaluate our design and development with strong and rich in data LLMs/VLMs

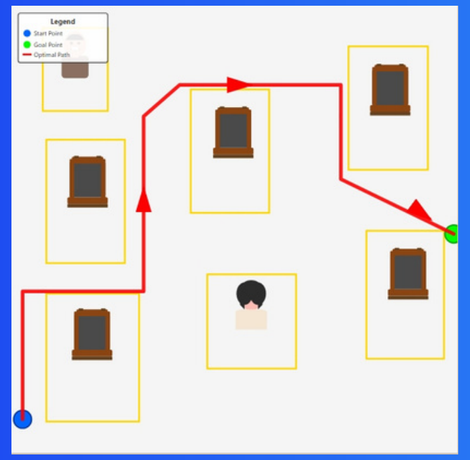
Grok



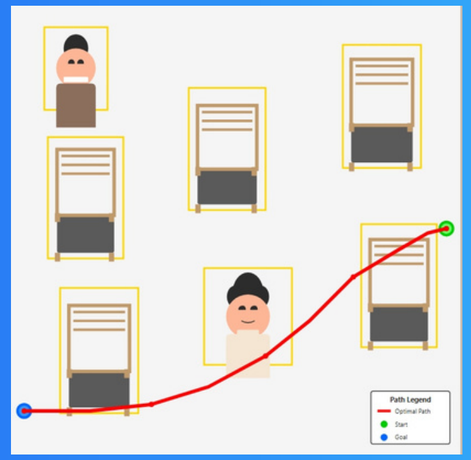
ChatGPT



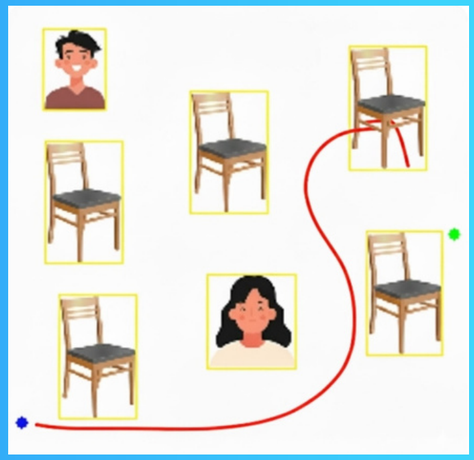
Claude



Claude



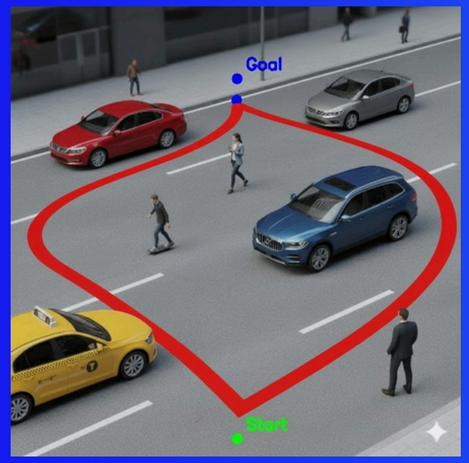
Gemini



IMPLEMENTATION [10]

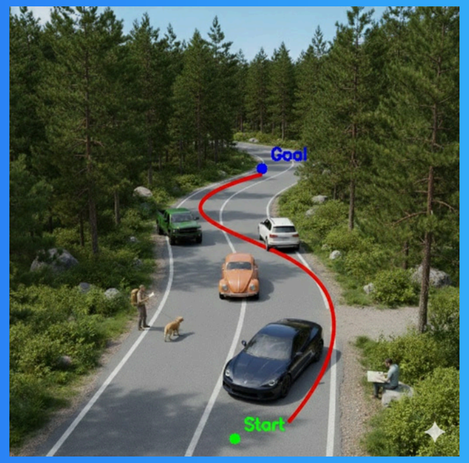
- And comparing the images with different scenes only with Gemini

Gemini



Gemini

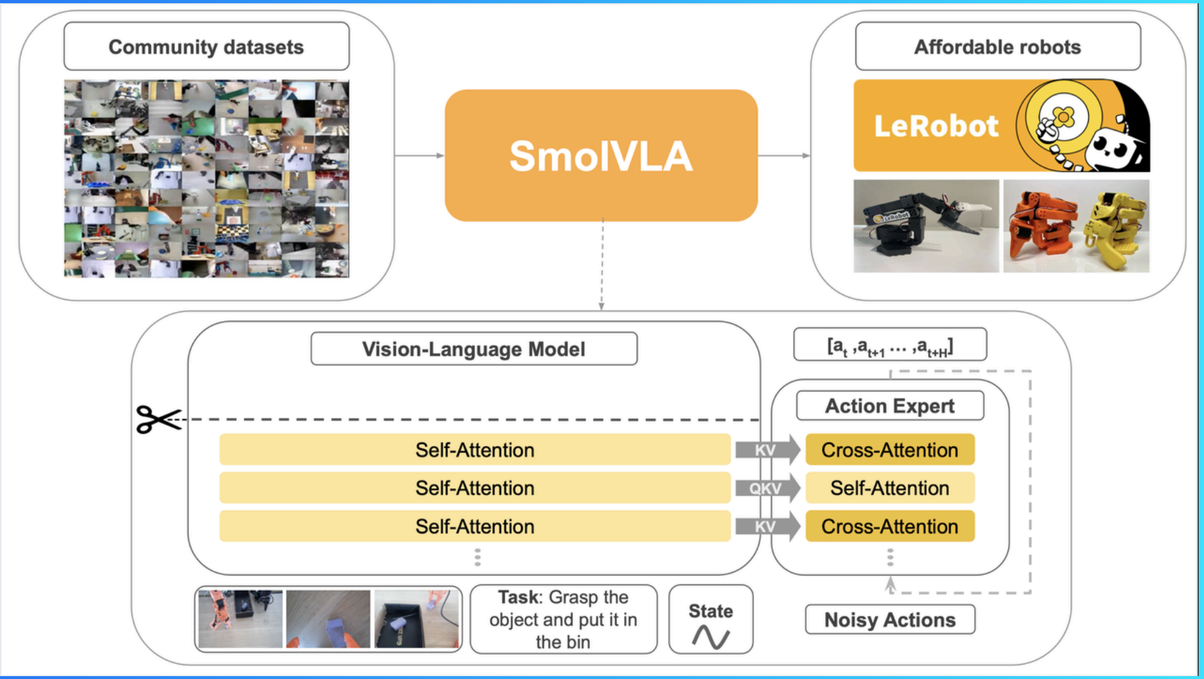
Gemini



Gemini

SMOLVLA [1]

- smolVLA, is a Visual Language Action model
- smolVLA core Architecture

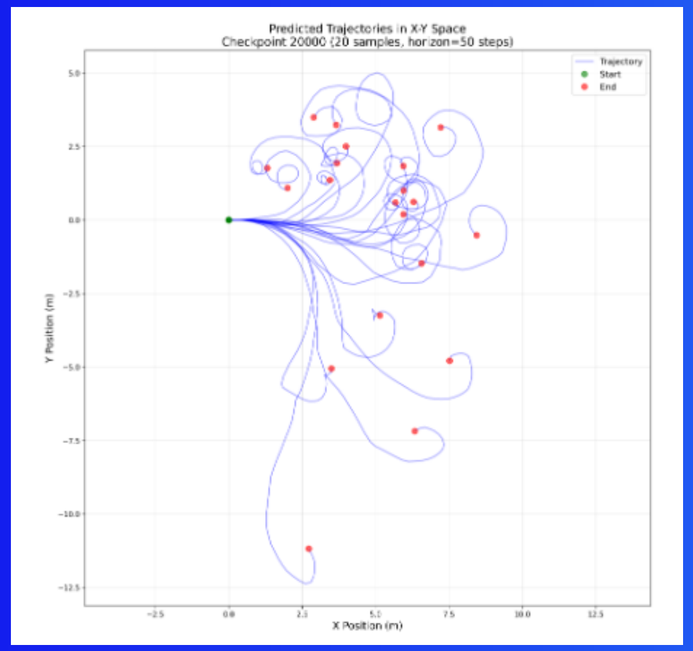


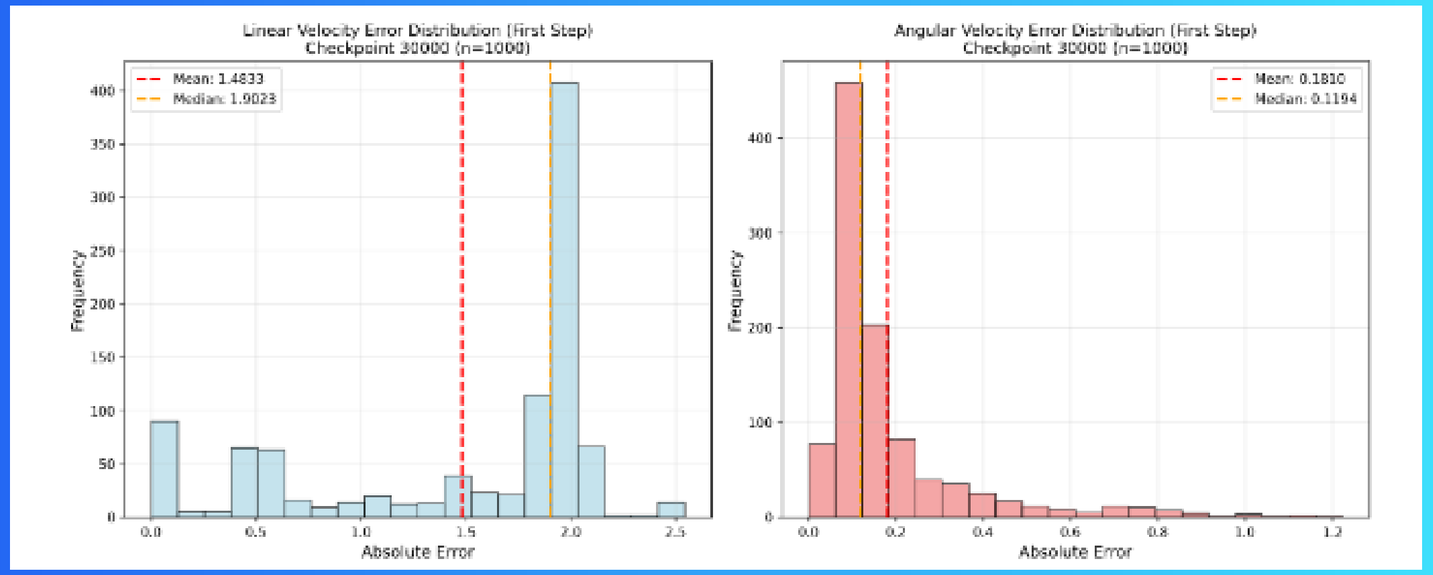
SMOLVLA [2]

- Socially CompliAnt Navigation Dataset [SCAND]: Is a Large-Scale Dataset of Demonstrations for Social Navigation, used to train the higher neural layers of smolVLA

SMOLVLA [3]

- smolVLA Tuning to generated the desired path with proper linear and angular velocities





SMOLVLA [4]

- Resulted paths from training and tuning phase of smolVLA
- Testing phase and results

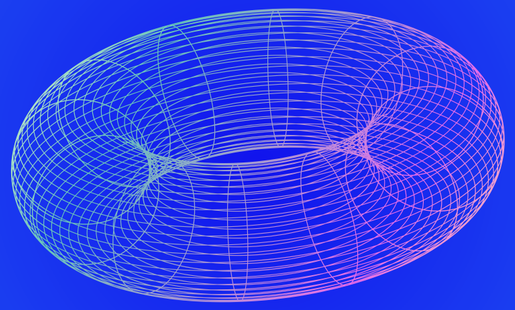
Figure 3.12: 3 random samples showing the ground truth and predicted movement vector for the model trained and evaluated on Jackal data.

Figure 3.13: 3 random samples showing the ground truth and predicted movement vector for the model trained on only Spot data and evaluated on Jackal data.

CHALLENGES

Parameters Tuning

Proper Prompts



Latency Duration

Different Start - Goal Positions and Scenes/Backgrounds

LIMITATIONS

01 Static 2D Images Only

02 Image Scaling to different sizes, ignores depth

03 smolVLM limited output, because of its size

04 Real-time process, takes a couple of minutes to provide the final generated path

CONCLUSION

Ultimately, for our team this period and project have been stepping stones to understand better these technologies and their importance for a proper socially aware navigation, as well as to continue further alongside with a more robust skillset, building and executing our main group project.

FUTURE SCOPE

TIAGO PRO

The next step is to implement the algorithm in a real-life robot, Tiago Pro from Pal, in a 3D world scenario

THANK YOU !

FOR YOUR ATTENTION