# The Application of SemaFORR Architecture to practical indoor Navigation Problems

Author: Jialu Huang

Supervisor: Elizabeth Sklar

# Outline for Section 1

1. Introduction
    - 1.1 Project Overview
    - 1.2 Background
2. Contribution
    - 2.1 System Design
    - 2.2 System Implementation
3. Experiment and Evaluation
    - 3.1 Unit tests
    - 3.2 physical Experiment
4. Conclusions

# Problem Definition

*Cognitive Indoor Navigation Technique*

Robots will break down during the navigation due to dynamic obstacles (or even dynamic environments with a changing map), and error from sensors. (Laser, odometry, etc.)

In some situation, robots may recover from a confused state (pp. 35, Fig. 26), which might take a long time due to lack of information about the environment or reasonable action decisions.

# Aims and Objectives

*Apply the SemaFORR theory to the ROS Navigation system*

## What do we want to achieve?

A cognitive navigation system which can not only handle the simple navigation tasks but also works functionally with dynamic obstacles and sensor noise.

## How? – Technique Salad!

- A basic Navigation system – **ROS Navigation system**
- A cognitive decision-making architecture – **SemaFORR**
- Information Collector – **Image based detector**
- Control Center – **Control & Communication techniques**

# Difficulties

*The Hard Easy*

- High complexity of the ROS
- Linux is new to me 😕
- Design of the layout of advisors in the SemaFORR architecture
- Multiple Object detection methods employed
- Real-time problems (running time, concurrent problems, etc.)

# Background

*An overview of the literature review*

Detailed background to the subject of indoor navigation including localisation and robotic mapping methods in the past two decades. (pp. 7, Sec.2.2)

Robotic perception based on laser sensor and vision information as well as the object detection focusing on the door detection and obstacle avoidance. (pp. 10, Sec. 2.3)

Introduction of the decision-making strategies: FORR (For the Right Reason) Theory and its advance version SemaFORR (Shared Experience Multi Agent) Theory. (pp. 12, Sec. 2.4)

# Outline for Section 2

1. Introduction
    1.1 Project Overview
    1.2 Background
2. Contribution
    2.1 System Design
    2.2 System Implementation
3. Experiment and Evaluation
    3.1 Unit tests
    3.2 physical Experiment
4. Conclusions

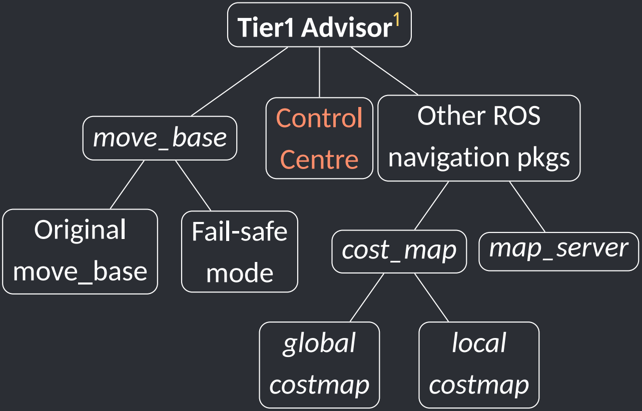
# Structure of the SemaFORR Architecture

**Algorithm 1:** SemaFoRR Architecture

```
Initialization;
while not Arrive_Destination do
    if Tier1 not Failed then
        Advice ← Tier1_Advisor($perception^t$);
        Execute Advice;
    else if Plans in Tier2_Advisor then
        while (Tier1Failed) && (Size(Plans) > 0) do
            Execute Plans[0];
            Plans.pop(0);
            Check Tier1;
        end
    else
        Choice ← Tier3_Advisor($perception^t$);
        if size(Choice) > 1 then
            Execute Choice[0];
            Choice.pop(0);
            Plans ← Choice;
        else
            Execute Choice;
        end
    end
end
```

# Structure of the System

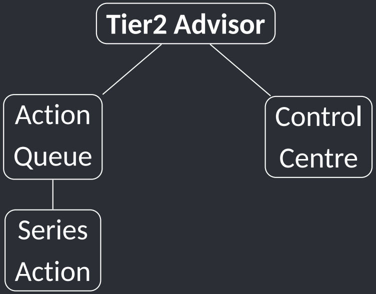
*The SemaFORR based Navigation System – Tier1 Advisor*



[1] We assume that the Tier1 Advisor is always 'right'

# Structure of the System

*The SemaFORR based Navigation System – Tier2 Advisor*



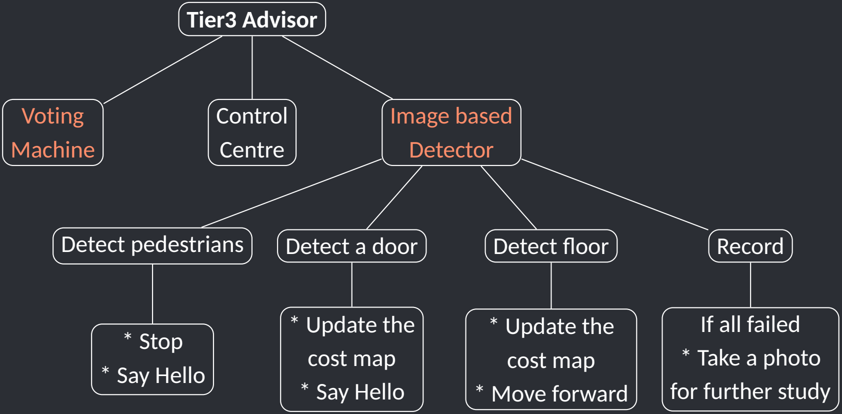
In the Tier2 Advisor, there is an Action queue and a control centre similar to that one inside the Tier1 advisor.

The Tier2 Advisor stores and executes existing plans.

The Action queue simply stores the serial action plans received from the Tier3 advisor

# Structure of the System

*The SemaFORR based Navigation System – Tier3 Advisor*



Tier3 Advisor

Voting Machine

Control Centre

Image based Detector

Detect pedestrians

* Stop
* Say Hello

Detect a door

* Update the cost map
* Say Hello

Detect floor

* Update the cost map
* Move forward

Record

If all failed
* Take a photo for further study

## Structure of the System

*Actions and sub-advisors in the Voting Machine*

Each sub-advisor will vote for actions based on their own biased detection results, all the votes will be added up to figure out the final decision.

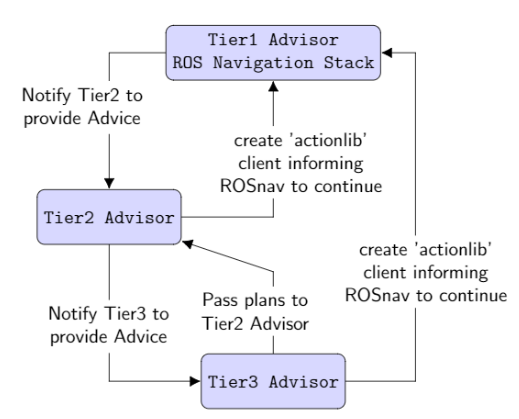
| | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| | Update map | Stop | Forward | Say Hello | series Actions |
| DM | ✓ | ✓ | – | ✓ | ✓ |
| BG | – | ✓ | – | ✓ | ✓ |
| PF | ✓ | – | ✓ | – | – |

Table: Actions and sub-advisors in the Voting Machine[2]

[2] DM: Door Man; BG: Body Guard; PF: Path Finder

# Communication in the System

*Signals and Messages among the Advisors*



Tier1 Advisor
ROS Navigation Stack
Notify Tier2 to provide Advice
create 'actionlib' client informing ROSnav to continue
Tier2 Advisor
create 'actionlib' client informing ROSnav to continue
Pass plans to Tier2 Advisor
Notify Tier3 to provide Advice
Tier3 Advisor

The messages can be divided into three classes, the request, the status and the goal.

# System Implementation

*The SemaFORR based Navigation System – Tier1 Advisor*

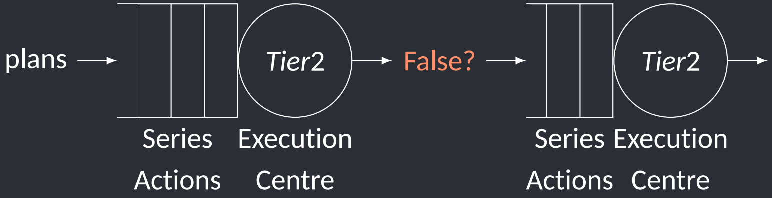
## Tier1 Advisor: Control centre + Revised ROS navigation system

```
if(!planner_->makePlan(start, req.goal, global_plan) ||
    global_plan.empty()){
      ROS_DEBUG_NAMED("move_base","Failed to find a plan,
      try to ask for advice from Tier 2 Advisor");
      //The ROS action that searching nearby goals were deleted
           ...
      //Ask for advice from the Tier 2 Advisor
      move_base::chooseAdvisor signal;
      signal.State = 2;
      switch_advisor_pub_.publish(signal);
      ROS_INFO("Waiting for Advice from Tier 2 Advisor...");
      r.sleep();}
```

# System Implementation

*The SemaFORR based Navigation System – Tier2 Advisor*

Complex serial actions will be passed to the Tier2 Advisor and stored in the action queue. These actions will be executed step by step. After each execution the Tier2 Advisor will check the status of the Tier1 Advisor.



# System Implementation

*The SemaFORR based Navigation System – Tier3 Advisor*

## Image based Detector

- Haar feature-based Detector
  - Pedestrian detector
  - Door Detector
- Line Segmentation based floor detection

## Voting Machine

- Voting machine Server

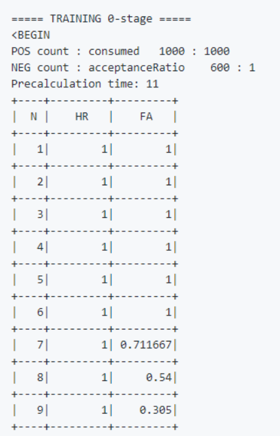
# System Implementation

*Image based detector in the Tier3 Advisor*

## Haar Cascade Classifier

- Pedestrian Classifier: Opencv lower_body classifier
- Door Classifier: Trained by 120 positive images and 50 negative images.

A variant Adaboost algorithm are used in the training processes. The final classifier is a weighted sum of all weak classifiers.



```
===== TRAINING 0-stage =====
<BEGIN
POS count : consumed   1000 : 1000
NEG count : acceptanceRatio    600 : 1
Precalculation time: 11
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3|        1|        1|
+----+---------+---------+
|   4|        1|        1|
+----+---------+---------+
|   5|        1|        1|
+----+---------+---------+
|   6|        1|        1|
+----+---------+---------+
|   7|        1| 0.711667|
+----+---------+---------+
|   8|        1|     0.54|
+----+---------+---------+
|   9|        1|    0.305|
+----+---------+---------+
```

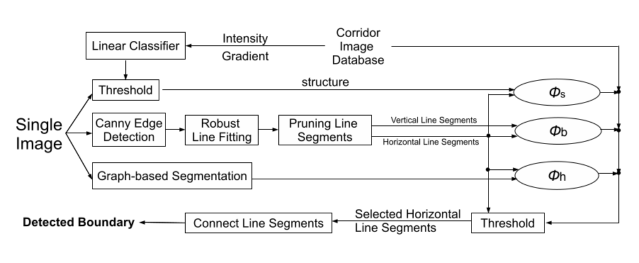
# System Implementation

*Image based detector in the Tier3 Advisor*

## Line Segmentation based floor detection

The score model contains three core elements, namely Structure Score, Bottom Score and the Homogeneous Score:

$$\Phi_{total}(l_h) = \omega_s \bar{\phi}_s(l_h) + \omega_b \bar{\phi}_b(l_h) + \omega_h \bar{\phi}_h(l_h)$$



# System Implementation

## *Voting Machine in Tier3 Advisor*

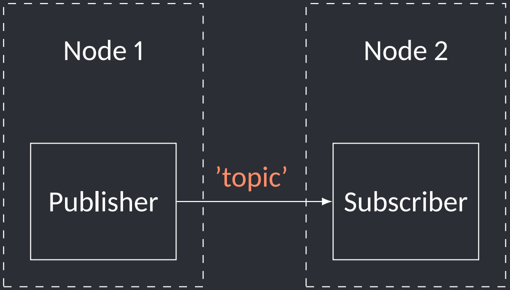
**Algorithm 2:** Voting Machine

```
Initialization(Advisor, Perception, Action, weight);
for i = 0 to Advisor.num() do
    $Advisor[i].comment \leftarrow Advisor[i].vote(Perception^t)$;
end
Sum_comment($i, weight[i] \times Advisor[i].comment$);
for j = 0 to Action.num() do
    $Action[j] \leftarrow Advisor.comment[j]$;
end
$index = arcmax(Action)$;
if $index.size() > 1$ then
    $index \leftarrow$ Random_pick($index$);
$Final\_Action \leftarrow Action[index]$;
```

# System Implementation

*Communications and Connections: Publisher/Subscriber*

Information will be stored in the message files (pp. 26-28, Fig.12-15) and published through specific topics, the related nodes will subscribe to the topics.



Message files:

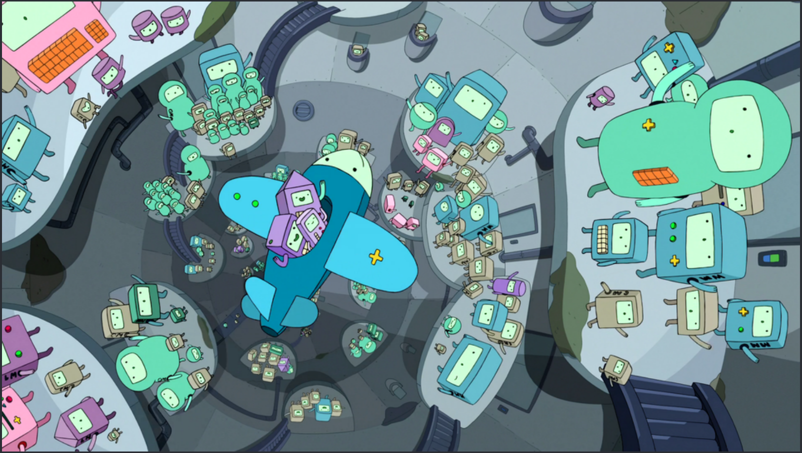
- ChooseAdvisor
- Detection_result
- Action_result
- PoseStamped
- SoundRequest
- Image

...

# System Implementation

## *Communication and Connections: Server/Client*

Call a server, pay the ’input’ and get your results back!



# Outline for Section 3

1. Introduction
   - 1.1 Project Overview
   - 1.2 Background
2. Contribution
   - 2.1 System Design
   - 2.2 System Implementation
3. Experiment and Evaluation
   - 3.1 Unit tests
   - 3.2 physical Experiment
4. Conclusions

# Experiment and Evaluation

*Unit test for the door, pedestrian and floor detectors*

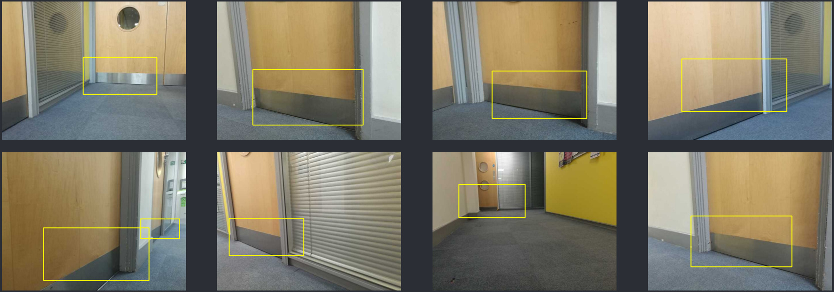


Figure: Correctly detected door images

# Experiment and Evaluation

*Unit test for the door, pedestrian and floor detectors*

Table: Test results of the Haar Cascade classifier for door detection

| | Accuracy | Error Rate | Recall | FP[3] Rate | Specificity | Precision |
|---|---|---|---|---|---|---|
| **value** | 0.65 | 0.35 | 0.80 | 0.50 | 0.50 | 0.80 |

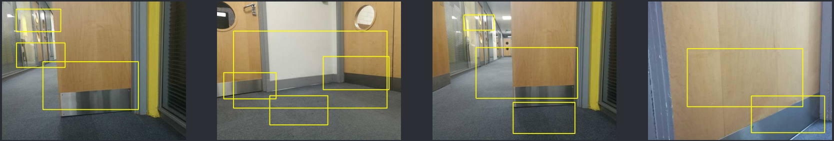


Figure: False detected door images

[3] False Positive

# Experiment and Evaluation

*Unit test for the door, pedestrian and floor detectors*

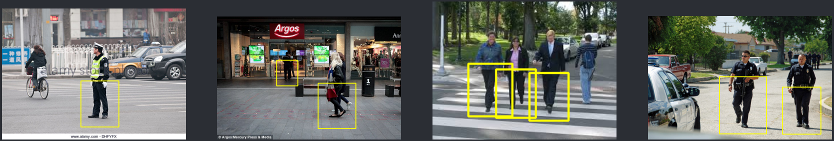


Figure: Correctly detected pedestrian images

# Experiment and Evaluation

## *Unit test for the door, pedestrian and floor detectors*

Table: Test results of Haar Cascade classifier for pedestrian detection

| | Accuracy | Error Rate | Recall | FP[4] Rate | Specificity | Precision |
|---|---|---|---|---|---|---|
| **value** | 0.57 | 0.42 | 0.50 | 0.20 | 0.80 | 0.50 |

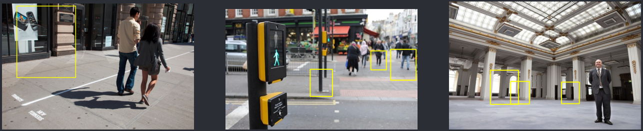


Figure: False detected pedestrian images

[4] False Positive

# Experiment and Evaluation

*Unit test for the door, pedestrian and floor detectors*

80 images(60 positive 20 negative) are tested, result examples and the ROC curve with an AUC area of 0.71 are shown below .

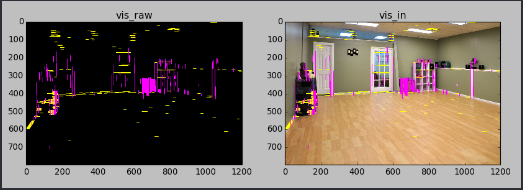


Figure: An example of the floor detection test

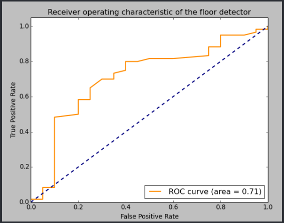


Figure: ROC curve

# Experiment and Evaluation

*Design of the Physical tests*

Purple circle: Start point Orange circle: Destination.

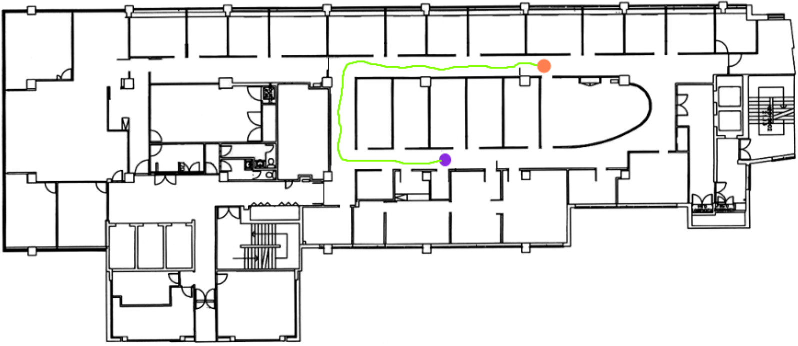


Figure: Test map used in this project

# Experiment and Evaluation

## *Design of the Physical tests*

- Scenario 1 Lovely Walk

  The corridor door is open without pedestrian.
- Scenario 2 Desperate Journey

  Only closed corridor doors will be set in this test.
- Scenario 3 Little Challenge

  An ajar door will be given in the environment.
- Scenario 4 Aliens

  The Pedestrians test will be set in this scenario.
- Scenario 5 Great Adventure

  Both ajar doors and the pedestrians will be considered.
- Scenario 6 Oops

  a random test for unexpected break down .

# Experiment and Evaluation

## *Results and Evaluation: Average Recovery Time*

Table: Average Recovery time of the ROS and SemaFORR based navigation system

| Recovery Time (s) | Lovely Walk (Oops) | Desperate Journey | Little Challenge | Aliens | Great Adventure |
|---|---|---|---|---|---|
| **ROS** | 21.66 | – | 158.76 | 35.31 | 159.99 |
| **SemaFORR** | 12.08 | 8.03 | 11.61 | 9.46 | 17.09 |

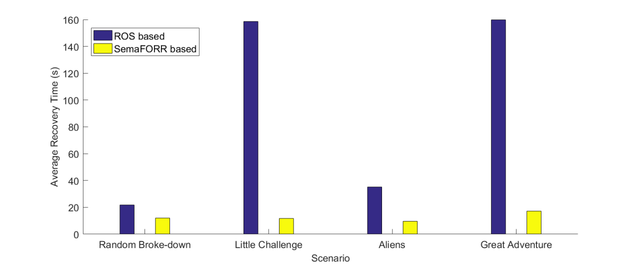


Figure: average recovery time in different tests

# Experiment and Evaluation

*Results and Evaluation: Distribution of Recovery Time*

Recovery time of the SemaFORR system is more stable.

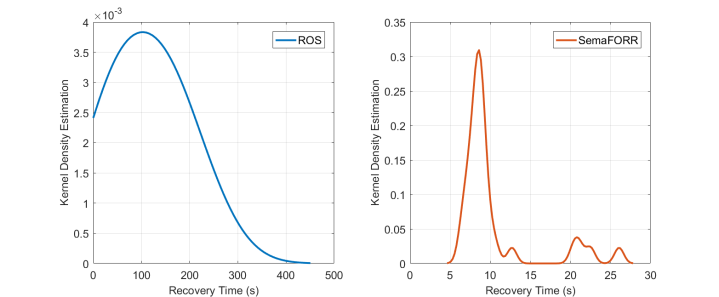


Figure: Kernel Density Estimation of the recovery time of the proposed systems

# Experiment and Evaluation

## *Results and Evaluation: Success Rate & Comparison*

Table: Success Rate test results

| Success Rate (%) | Lovely Walk (Oops) | Desperate Journey | Little Challenge | Aliens | Great Adventure |
|---|---|---|---|---|---|
| **ROS** | 100.00 | 0.00 | 60.00 | 90.00 | 50.00 |
| **SemaFORR** | 60.00 | 40.00 | 50.00 | 80.00 | 50.00 |

Table: Comparison between the two studied systems

| Change Rate (%) | Lovely Walk (Oops) | Desperate Journey | Little Challenge | Aliens | Great Adventure |
|---|---|---|---|---|---|
| **Recovery Time** | -44.22 | – | -92.69 | -73.21 | -89.32 |
| **Success Rate** | -40.00 | +40.00 | -16.67 | -11.11 | 0.00 |
| **Lethality**[5] | 0 | 1 | 1 | 0 | 1 |

[5] Times of serious failure during the test

# Experiment and Evaluation

## *Results and Evaluation: Detection and Voting Analysis*

Table: False test results versus false detection results

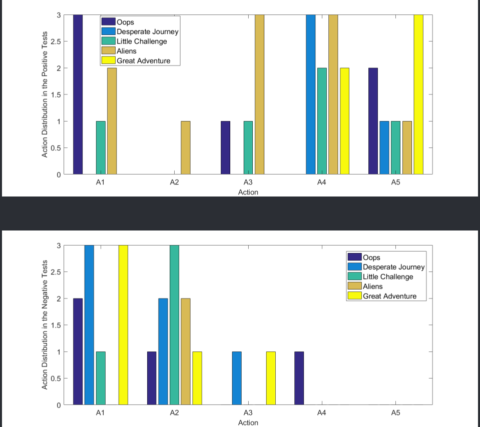
| Selection | Oops | Desperate Journey | Little Challenge | Aliens | Great Adventure | Total |
|---|---|---|---|---|---|---|
| **Door** | 4/9 | 1/1 | 0/0 | 1/5 | 2/2 | 8/17 |
| **Pedestrian** | 1/3 | 0/1 | 0/0 | 0/1 | 3/3 | 4/8 |
| **Floor** | 1/1 | 4/4 | 1/3 | 1/3 | 4/5 | 11/16 |

Table: Action distribution in the positive tests

| Selection | Oops | Desperate Journey | Little Challenge | Aliens | Great Adventure | Total |
|---|---|---|---|---|---|---|
| **A1** | **3** | 0 | 1 | 2 | 0 | 6 |
| **A2** | 0 | 0 | 0 | 1 | 0 | 2 |
| **A3** | 1 | 0 | 1 | 3 | 0 | 5 |
| **A4** | 0 | **3** | **2** | **3** | 2 | **10** |
| **A5** | 2 | 1 | 1 | 1 | **3** | 8 |

# Experiment and Evaluation

## *Results and Evaluation: Detection and Voting Analysis*



A4 and A5 are effective actions that can produce positive results even in some complicated environments.

Both A1 and A2 should be improved in further study.

Figure: Action distribution in positive and negative tests

# Outline for Section 4

1. Introduction
   1.1 Project Overview
   1.2 Background
2. Contribution
   2.1 System Design
   2.2 System Implementation
3. Experiment and Evaluation
   3.1 Unit tests
   3.2 physical Experiment
4. Conclusions

# Conclusion

*Advantages and Contributions*

- The SemaFORR theory has been studied and applied to the indoor navigation system;
- Implementing the SemaFORR based system by creating three tiers advisor.
- In the Tier3 advisor, two core modules, the image based detector and the voting machine, are designed and implemented.
- Tests with six different scenarios are designed and implemented.
- SemaFORR based system has faster recovery performance.

## Conclusion

*Limitations and Further Work*

- The accuracy of the image based detector can barely meets the requirement, which should be improved.
- For now, the amount of detectable dynamic obstacles and available actions are too small.
- The voting process should be improved by using weights derived from observation of the robot's performance and the recorded tests' data in different scenarios. Advanced machine learning techniques can also be applied to train the robot to use related optima strategy in specific environments.
- Other perceptions like sound, temperature, etc.

## Finale

*Thank you for your time!*