2022 年臺灣國際科學展覽會
優勝作品專輯

作品編號	190038
參展科別	電腦科學與資訊工程
作品名稱	A Person Re-identification based Misidentification-proof Person Following Service Robot
得獎獎項	二等獎

國　　家	Macau
就讀學校	Macau Pui Ching Middle School
指導教師	Lam Kin Un
作者姓名	Leong Chi Io

關鍵詞

作者照片

1. Introduction

Abstract

Two years ago, I attended a robot contest, in which one of the missions required the robot to follow the pedestrian to complete the task. At that time, I used their demo program to complete the task. Not long after, I found two main issues:

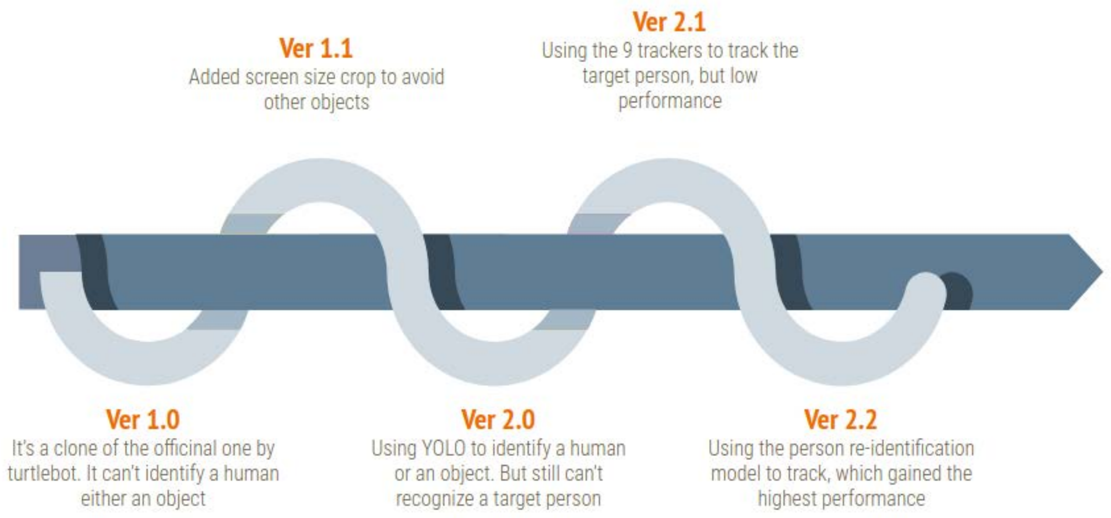
1. The program follows the closest point read by the depth camera, which if I walk close to a wall next to, the robot may likely 'follow' the wall.
2. Not to mention if another pedestrian crosses between the robot and the target.

Regarding these two issues, I decided to improve it.
I've started searching on the web to see if there are any similar projects that have already been done. Not unexpectedly, there's a suitcase named cowarobot (***fig 1.2***) that can follow the owner. Testes proved that it wouldn't follow the wall, which was solved by the first situation, but it may still follow another pedestrian when they cross between it.
Person following has a variety of applications in reality, for example an automatic luggage carrier in the airport, an automatically follows you supermarket car inside the supermarket, or an human-robot collaborating to explore unknown areas.
As the tesla bot (***fig 1.1***) revealed its appearance in Tesla's AI day on Aug 20, 2021, service robots will become much more popular in the future, even if the family has a domestic service robot. As this happens, many features must be implemented in order to complete daily tasks, and person following is one of them.
In this report, we demonstrate using person re-identification technology, which was mostly used at fixed-spot monitoring cameras to track pedestrians, to re-identify the target. Combined with object detection, this successfully solved both issues in the original following procedure.

Fig 1.1 & 1.2 Tesla bot and a Cowarobot

2. Solution

2.1 Developing History


Ver 1.1
Added screen size crop to avoid other objects
Ver 2.1
Using the 9 trackers to track the target person, but low performance
Ver 1.0
It's a clone of the officinal one by turtlebot. It can't identify a human either an object
Ver 2.0
Using YOLO to identify a human or an object. But still can't recognize a target person
Ver 2.2
Using the person re-identification model to track, which gained the highest performance

For 2 years, I've developed 5 versions of this person follower until today. The first 2 versions can't even classify a human or not, the second 2 versions could classify a human but not recognize it. The last version, with the optimization of this project, archives everything.

2.2 System Building

Our system is divided into 3 parts: the *Initializer*, the *Main Application,* and the *RobotHandler*. The following actions occur in order:

1. The system first registers an initial descriptor with the target person's front and back body image cropped out by human detection. At this point, the person must stand inside a blue bounding box in the center of the robot's view.
2. After registration is done, the main application continues to track people with YOLO and capture the descriptor of everybody detected with the image cropped out.
3. It compares them with the initial descriptor (the targets) with *cosine similarity*. The descriptor with the highest similarity would be recognized as our target.
4. The robot will follow the target person using the x-coordinate in the image and the distance from the target person by inputting them into the PD Controller, which is handled by the RobotHandler for the robot to move.

When moving, the robot will keep collecting the target person's coordinates on the map and save them as waypoints for trajectory trend prediction, and this part is handled by the *Waypoint Recorder Application*.

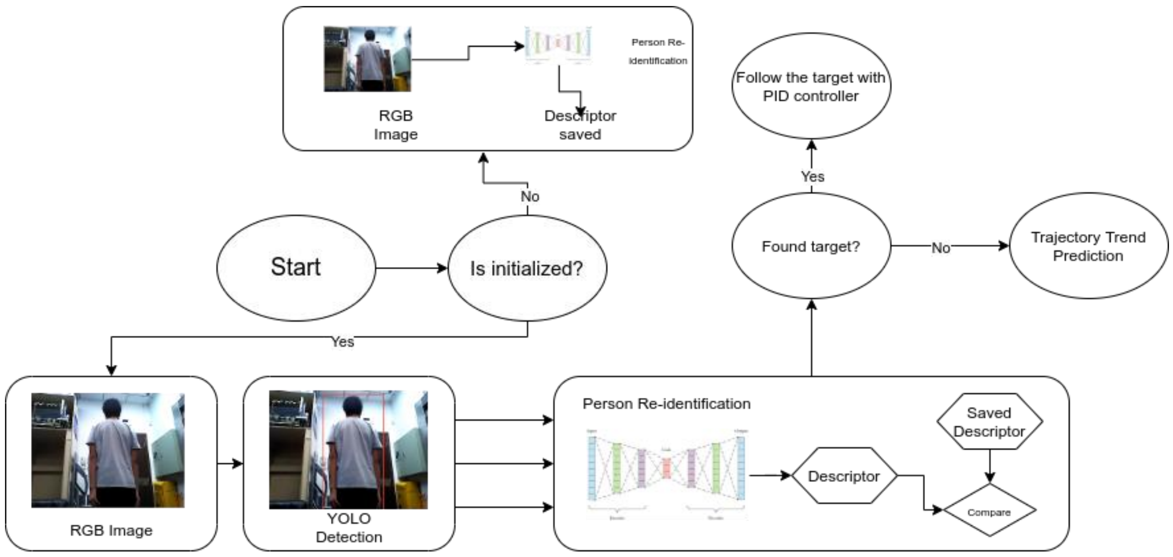


Fig 2.1.1 The brief graph of our system

In normal situations, the robot is in **NORMAL** mode. If the Main Application loses the target person from the camera view (for instance, another person suddenly crosses between the target and the robot), the robot will first enter **CONFIRM_LOST** mode. The robot will follow a temporary person which has intersections between the target's last existent box for 0.5s. The robot will also decrease its PD Controller's parameters to slow-down their reaction to changes. If the duration has passed, and the robot still could not spot the main target person from the view, then the robot will enter **LOST** mode. If the robot re-identified a target, it will enter **CONFIRM_REIDENTIFIED** mode, since this may only last for one frame and have to confirm for 0.5s.
Currently, entering **LOST** mode will only result in the robot stopped, which is not allowed in reality situations. So *Trajectory Trend Prediction* will and must be applied in the future, which it was highlighted in cyan in ***fig 2.1.2***

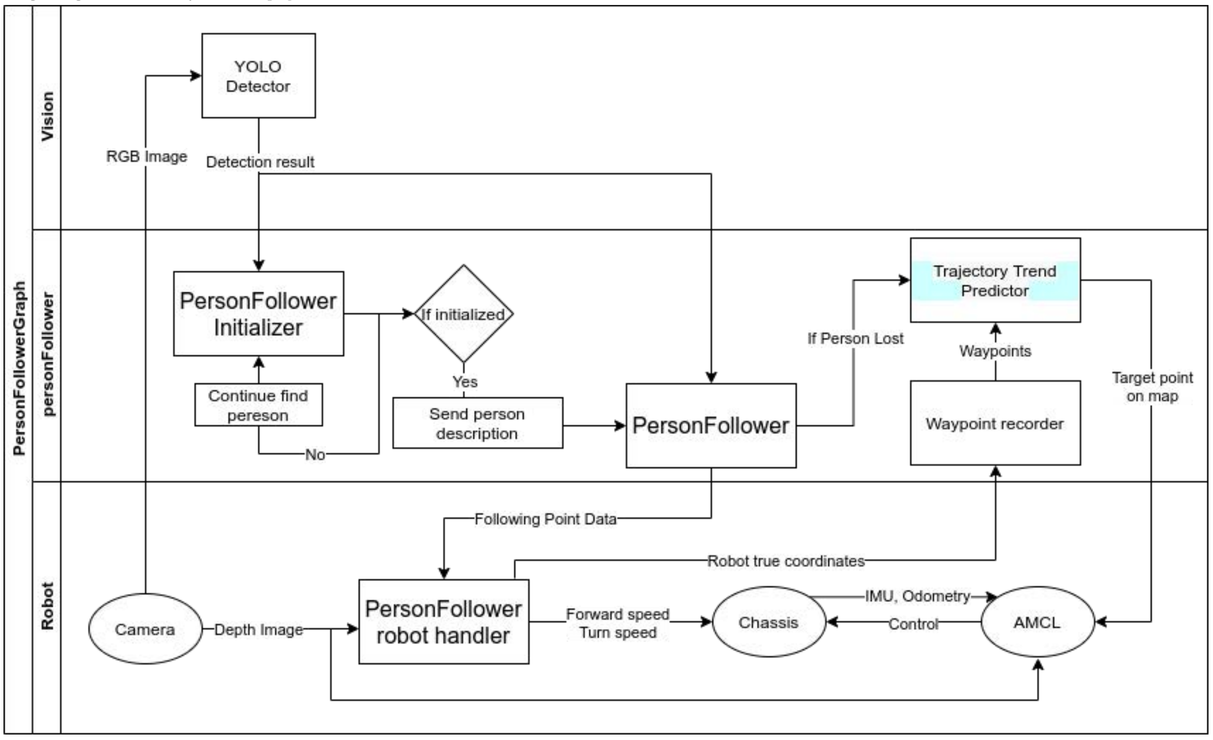


Fig 2.1.2 A system graph of the system

2.3 Method Proposing

2.3.1 Extracting Persons

Our robot relies heavily on its computer vision. It makes use of a camera to “look” at the outside world and passes this information to a series of image processing and (re)identification models so that it can follow the main target. We have used object detection to detect all individuals who appear in front of the robot.

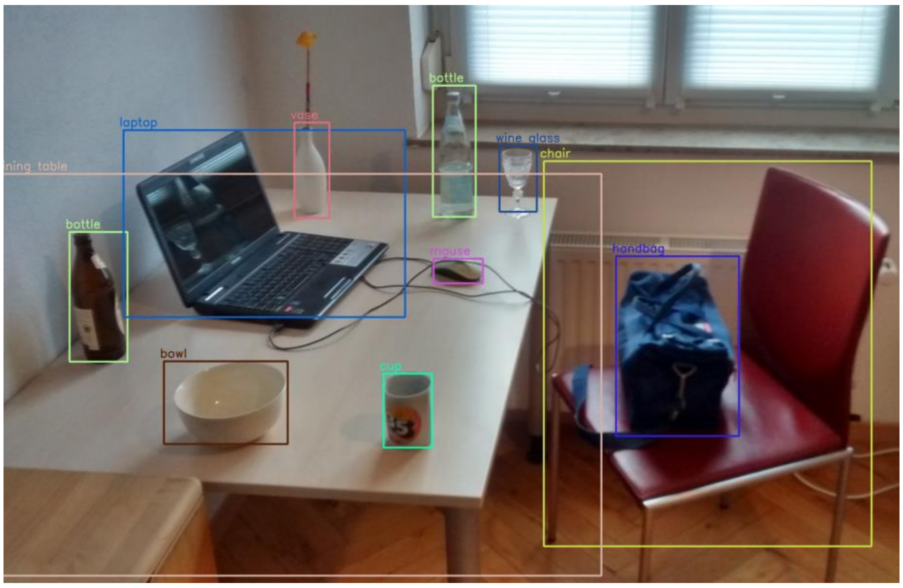


Fig 2.3.1.1 An example of object detection in a room

We have tested four methods: SSD300 & 500 [2], R-FCN [3] and YOLO [1] and compare each other's mAP and FPS. The results are shown in ***Fig 2.3.2***.

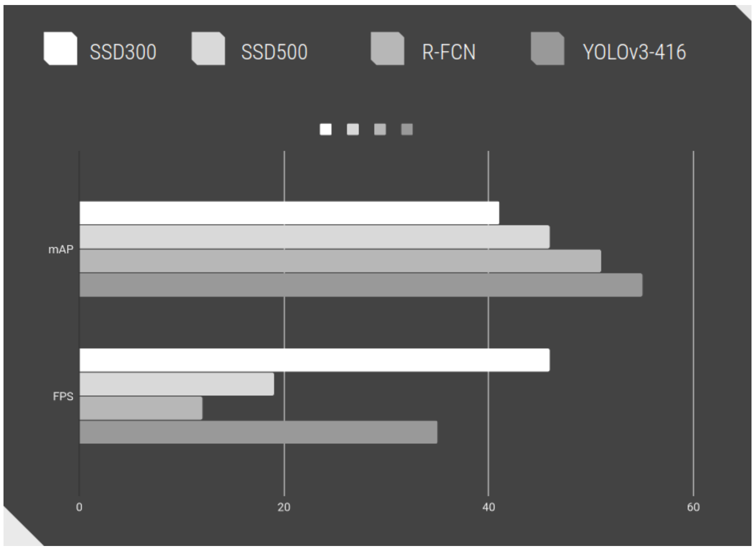


Fig 2.3.1.2 The comparison mAP (mean average precision) and FPS (frame per second) between tested methods

As we need to keep the video stream live to spot the target person, FPS is an important criterion. The results yield that SSD300 gains the highest speed in FPS, while YOLO has mediocre speed (about 30

~ 35 FPS, close to real-time), but YOLO has the highest mAP score which others could not achieve, so at last YOLO was selected to be our detection model.

2.3.2 Extract Features from Persons

After extracting people from the image, we would need to recognize the master target among all people in order to follow him/her.
Before person re-identification is used, we have tested using object trackers to archive the re-identification process. As the trackers yield high accuracy results.

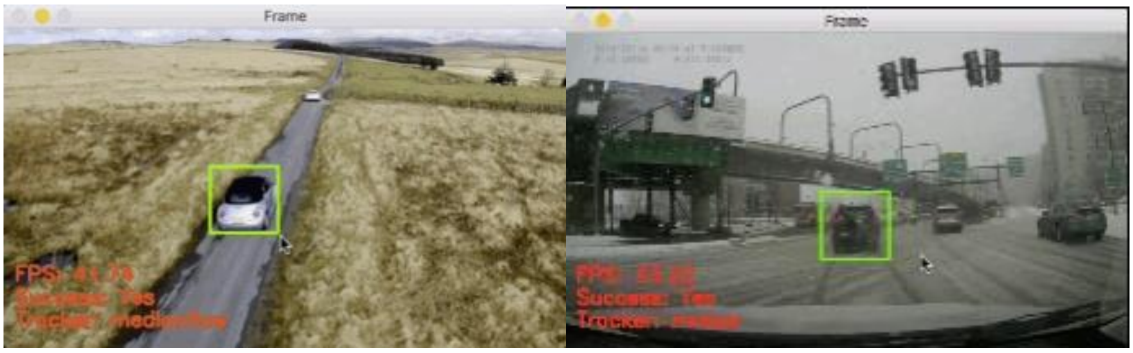


Fig 2.3.2.1 Two cars tracked by MedianFlow tracker

Here is a list of trackers we have tested, descriptions written by Dr.Adrian Rosebrock [4].

Backend: Adaboost	
Tracker	Brief Intro
BOOSTING Tracker [5]	It is over a decade old. This tracker is slow and does not work very well.
MIL Tracker	Better accuracy than BOOSTING tracker

Backend: Correlation Layers	
Tracker	Brief Intro
KCF Tracker [6]	Faster than BOOSTING and MIL. Similar to MIL and KCF, it does not handle full occlusion well.
CSRT Tracker [7]	Tends to be more accurate than KCF but slightly slower.
Dlib Correlation Layer Tracker	Better than everything above, but accuracy lowers the longer it runs

Backend: Itself	
Tracker	Brief Intro
MedianFlow Tracker	If there is too large of a jump in motion, such as fast-moving objects, or objects that change quickly in their appearance, the model will fail.
TLD Tracker	Is incredibly prone to false-positives

MOSSE Tracker	Very fast but not as accurate as CSRT or KCF

The trackers require a piece of the image of the target object to do the tracking operation, which means the object trackers could track any objects. It is not limited to humans, cars in the figure do work.

2.3.2.1 Testing the trackers

As for testing the listed trackers, we have set up a situation with the following figure to test if the trackers do behave as how they are in the example videos.
In the tests, there is a camera that stayed at a fixed location and a fixed angle, as in ***Fig 2.3.2.2***:

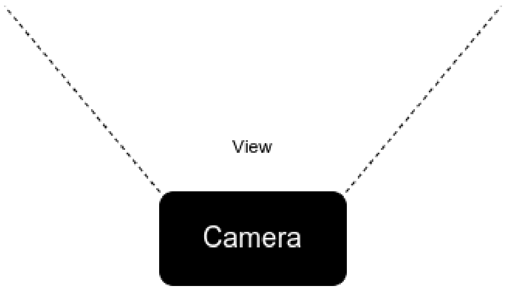


Fig 2.3.2.1 The camera position environment

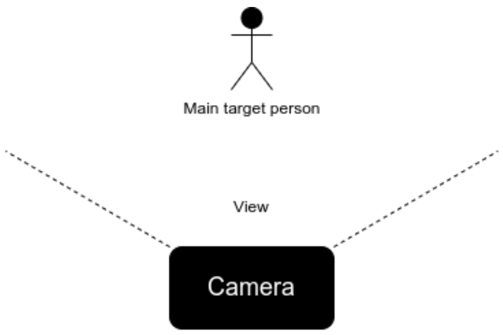


Fig 2.3.2.2 The main person

The environment will be fixed so the lighting conditions would never change throughout the tests, and the main target person that the tracker should recognize, will lay in the center.
There would be two levels of tests performed. Each level includes 2 scenarios: The single-person scenario and the multi-person scenario. In the single-person scenario of the first level, the main target person only slightly swings without others appearing in front of the robot. As pedestrians crossing was a common scenario in the following task, there will be two non-target people (a.k.a '*Distractors*') slowly following a specific trajectory, trying to steal the focus of the main target person from our robot in the multi-person scenario.

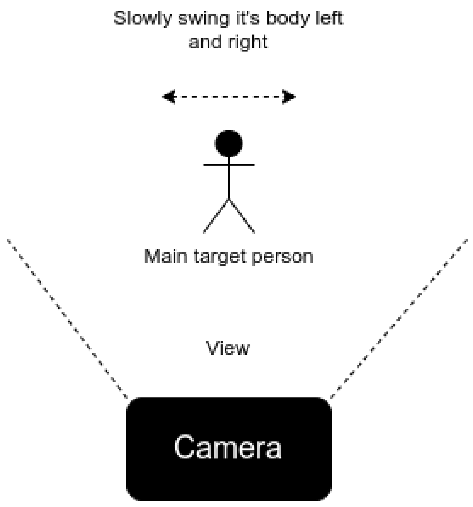
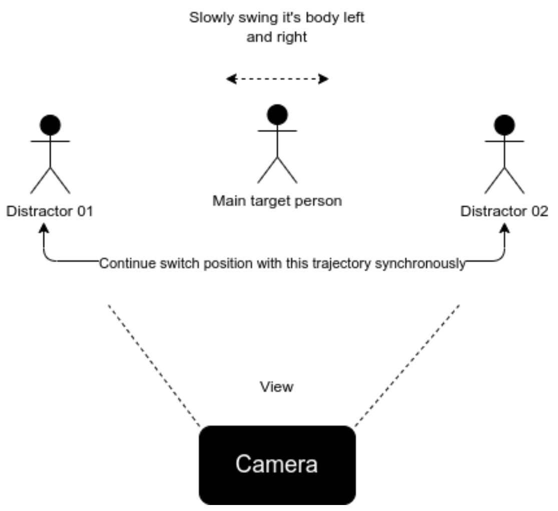


Fig 2.3.2.3 The single-person scenario in the first level


Slowly swing it's body left and right
Distractor 01
Main target person
Distractor 02
Continue switch position with this trajectory synchronously
View
Camera
Fig 2.3.2.4 The multi-person scenario in the first level

The second level will have the same layout as the first level, but this time the main target person would move drastically instead of slightly moving. This increases the difficulty for the tracker, especially when the 'distractors' are present in the multi-person scenario.

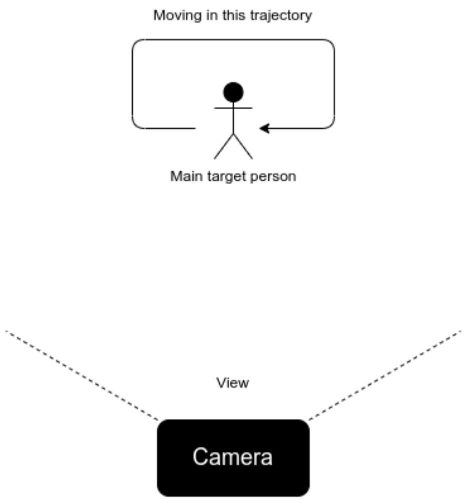


Fig 2.3.2.5 The single-person scenario in the second level

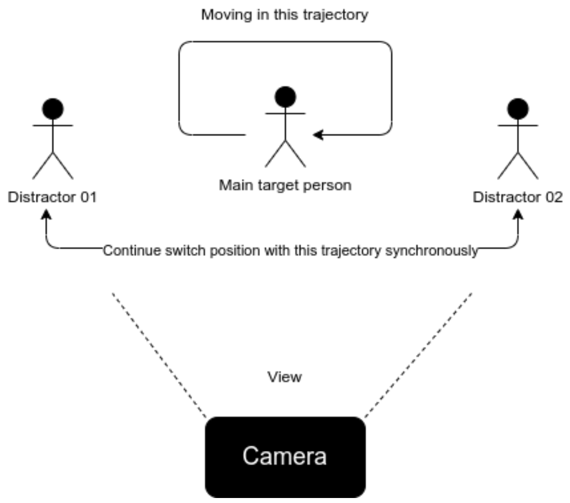
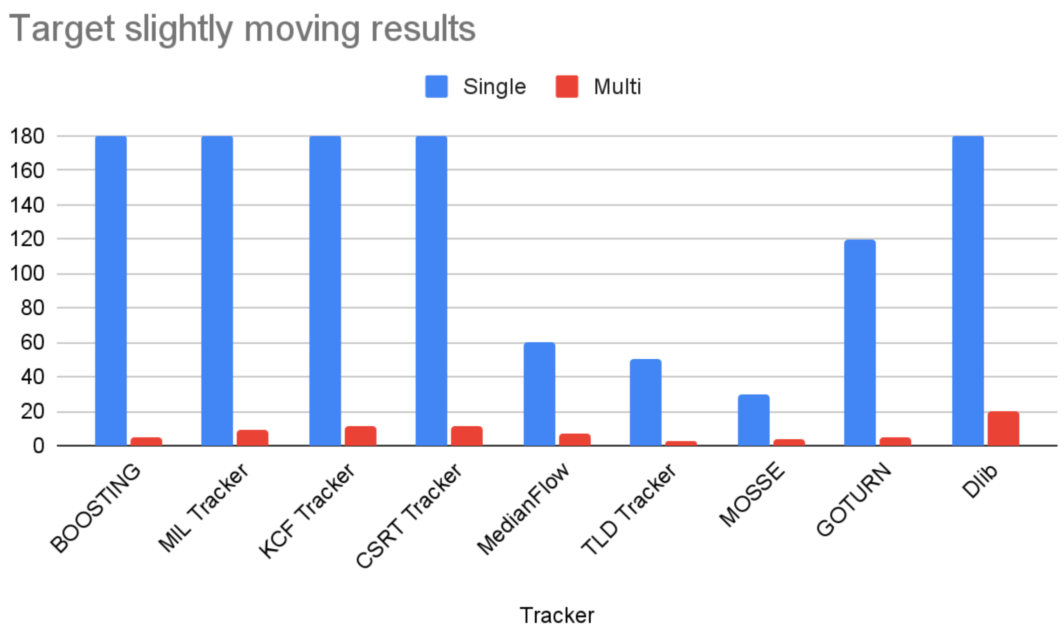

Moving in this trajectory
Distractor 01
Main target person
Distractor 02
Continue switch position with this trajectory synchronously
View
Camera

Fig 2.3.2.6 The multi-person scenario in the second level

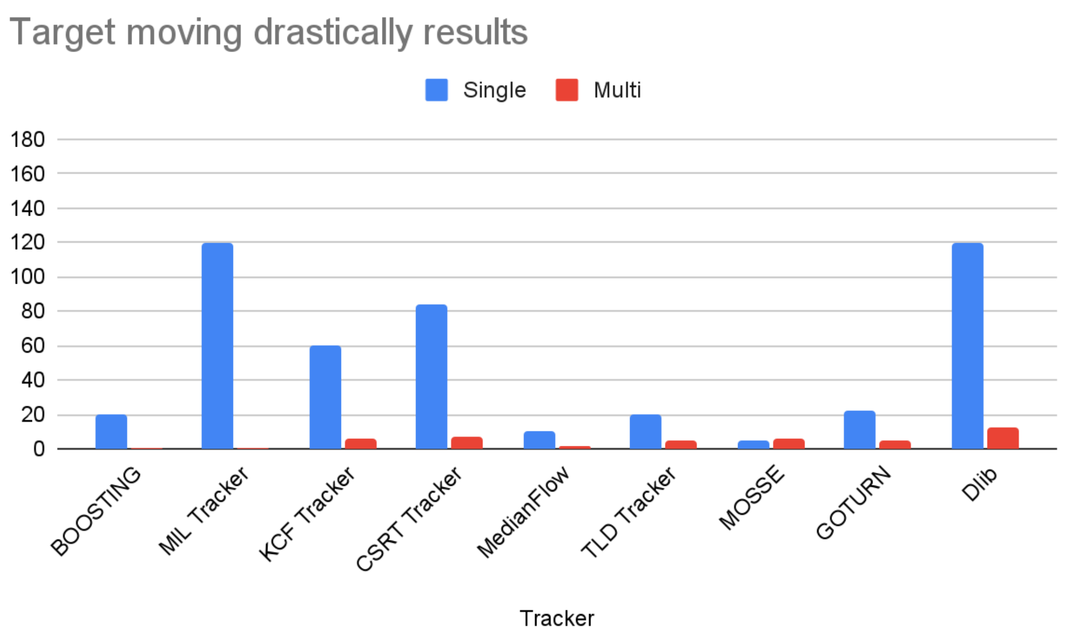
Each tracker is tested 3 times in a stage, and each stage lasts for 180 seconds (3 minutes). Due to the trackers will not lose the box when it missed the target (for instance, dlib correlation layer tracker's box will continue to increase its size until the box is as large as the image if it lost the target), If the centroid of the box was not on the main target person, the tracker will be estimated as it lost the target and the test will stop immediately.

Here are the test results in level 1:



As by the results, the object trackers performed well when the target was slightly moving without anybody. But it immediately descended dramatically into seconds when there were several pedestrians starting to annoy. Meaning it could not handle large motion jumps and blockings.

Here are the test results in level 2



As we could observe it obviously, the lastling time descended more dramatically even in the single-person only scenario, not to mention the multi-person scenario, which meaning these trackers couldn't handle large motion jumps also, as even the dlib correlation layer tracker failed in average of 13 seconds, and the phenomenon of the main target person's position being stolen becomes more often in the second level.
In conclusion, These trackers do archive what our target was, follow humans and identify the human. But following was a timed task, and multi person scenarios were the most common scenarios of following, including pedestrians crossing and blocking the view causing the target. With the poor result shown, These trackers do not fit our standards.

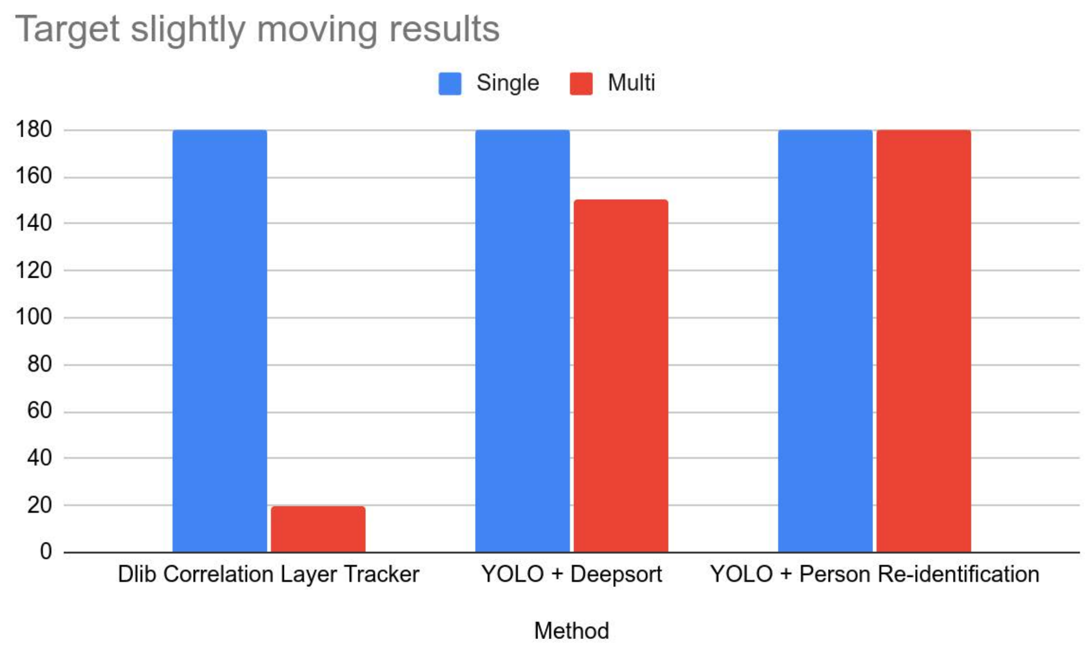
2.3.2.2 Re-identification method

Due to these consequences, we have decided to aim our eye on '*identifying*' the target instead of '*tracking*' it. We have found two methods that are following our concept:

Identify Methods	
Method	Brief Intro
YOLO + Deepsort [8]	Powered by kalman filter
YOLO + Person Re-Id	Re-identification system powered by AutoEncoders

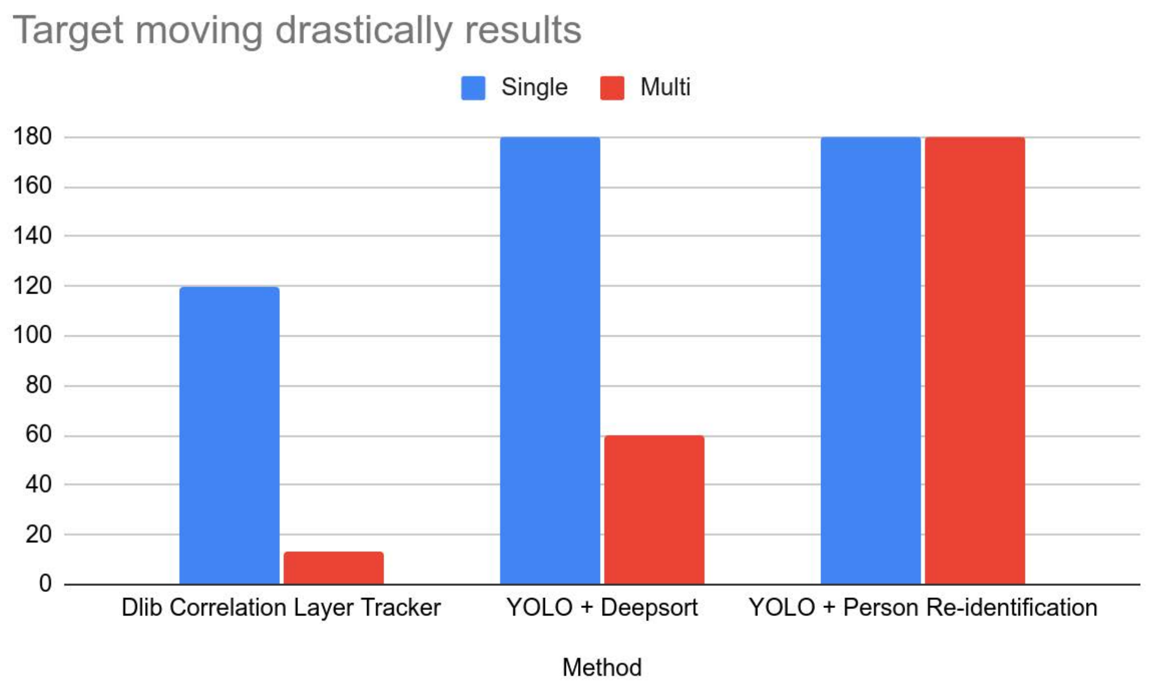
We applied these two methods into the level test too. On the right, you could see we also feat. the best object tracker

Here is Level 1



Compared to the best of the trackers, deepsort and person re-id method could hold on longer to multi-person scenarios, which person re-id method even successfully identified the target for a whole 180 secs period. This proves that methods using YOLO object detection have a large gap between object trackers.

Here is Level 2



At this point, we focus on comparison between Deepsort and Person Re-identification. Deepsort still does not lose the target though the target's a drastic movement. But it fails when distractors start to engage in it. YOLO + Person Re-identification could still identify the target even if the target was drastically moving and there were distractors engaging in. We could easily determine which one is better.

2.3.2.2 Gathering up

The test results promoted that person re-identification could pass all tests without losing the main target person in 3 minutes. Theoretically, if the test continues without a time limit, it could '*recognize*' the main target person forever. This method's accuracy wouldn't be decreased as time progresses due to the fact that person re-identification focuses on '*recognizing*' the target instead of '*tracking*' it. Here is an example: if a person re-identification model has recognized me, even another non-target person blocks the view, no matter how long time has passed (1 minute, 1 hour, 1 day, even a year), as long as the environment does not change substantially, once the other person unblocks the view, the system could still recognize the target. You can imagine as face recognition systems, will the person be lost on the screen because of blocking time? It won't. This is a thing other trackers could not do since they are not trying to '*recognize*' the target but to track. (Dlib Correlation Layer Tracker is an exception, though if the target keeps moving before another person approaches, it would still lose the target). With these characteristics, the person re-identification model would never follow the wrong person. Moreover, in reality, our robot will be moving, but object trackers were often designed for fixed spot cameras. As such, our robot performs comparatively better than its alternatives. YOLO + Deepsort was actually a method between '*recognizing*' and '*tracking*', and this is why it failed in the multi-person scenario, due to its inheritance of some features of the trackers.

The person re-identification model, pre-trained by Intel developers, uses the RMNet[9] backbone. By inputting the people detected with YOLO to the model, the model will return a 1×256 feature vector representing that person.
We could use these feature vectors and do a comparison by using cosine similarity.

$$\cos\langle \vec{v_i},\ \vec{v_0}\rangle = \frac{\vec{v_i} \cdot \vec{v_0}}{|\vec{v_i}||\vec{v_0}|}$$

We opted for *cosine similarity* to calculate how close these vectors are since compared to Euclidean distance, it has the advantage of already normalized to be in $[-1,1]$ and is thus easier to tackle.

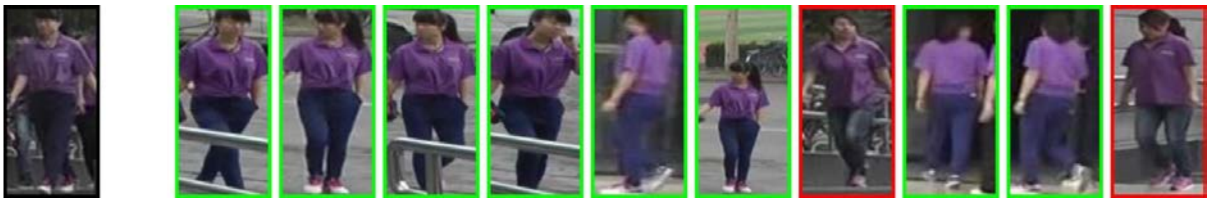


Fig 2.3.2.7 A testing image provided by the Intel development team. The black-bordered image was the initial image. The right images with green and red borders are images to be compared with. Images with red boxes are estimated that the person inside is not the same person as the initial one, and green has the opposite meaning.

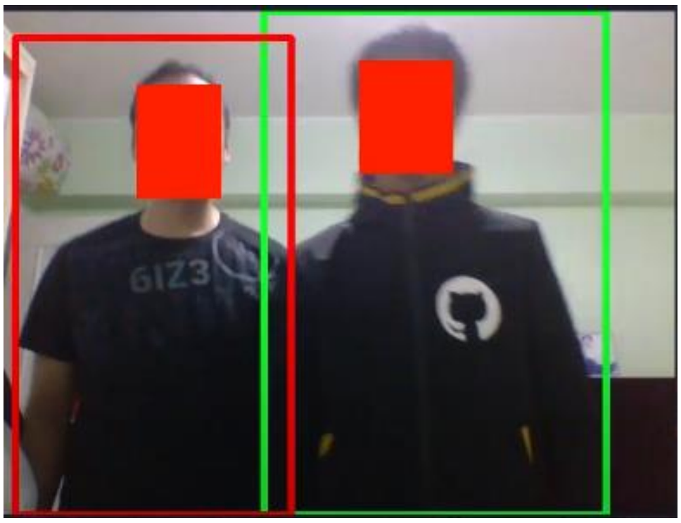
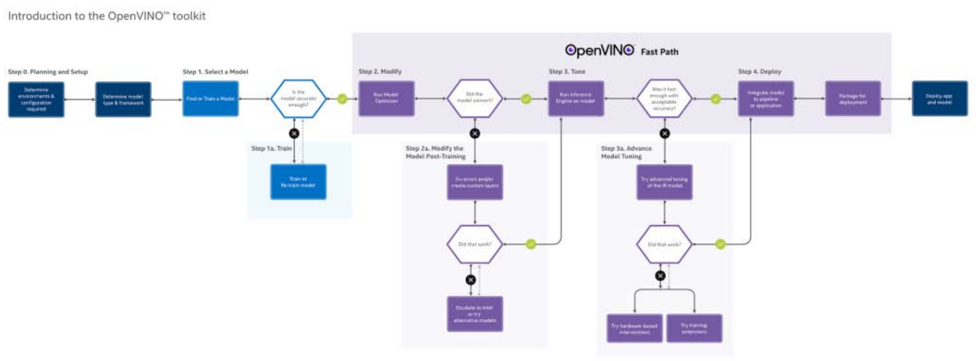


Fig 2.3.2.8 One of the images captured from the test
Green bounding box represents the main target person

2.4 OpenVINO (Intel Inc.)

In order to use the person re-identification model to recognize the target person, we need to use Intel's framework to evaluate it: OpenVINO. We often hear that robots require high computing performance to evaluate deep learning models, using units such as Graphics Processing Unit (GPU). OpenVINO aimed to be a model optimizer making models lighter - to evaluate on CPU. This is also a windfall to us, due to the fact that we can evaluate models without expensive devices, which can lower the project cost.


Introduction to the OpenVINO™ toolkit
OpenVINO Fast Path
Step 0. Planning and Setup
Determine environments & configuration required
Determine model type & framework
Step 1. Select a Model
Find or Train a Model
Is the model accurate enough?
Step 1a. Train
Train or Re-train model
Step 2. Modify
Run Model Optimizer
Did the model convert?
Step 2a. Modify the Model Post-Training
Fix errors and/or create custom layers
Did that work?
Escalate to Intel or try alternative models
Step 3. Tune
Run Inference Engine on model
Was it fast enough with acceptable accuracy?
Step 3a. Advance Model Tuning
Try advanced tuning of the IR model
Did that work?
Try hardware-based interventions
Try training extensions
Step 4. Deploy
Integrate model to pipeline or application
Package for deployment
Deploy app and model

The diagram of OpenVINO

2.5 Testing Videos

This video contains a fixed-spot test with two testers. The stream shown on the screen was the robot's camera view. In the video, the green bounding box represents the estimated target by person re-identification, while red boxes are pedestrians.

This video contains a robot test with two testers. The window showing in front of the view was the robot's camera view with the same box rules as the video on the left. As the person moves, you can see the robot was moving together with the target, archiving the following task.

2.6 ROS

ROS declares each program as '*nodes*' running asynchronously. Technically, multiprocessing is easy to achieve in most modern programming languages, for instance, Python has a library named *threading* that handles it. However, ROS introduced a communication platform for 'nodes' to communicate with each other through 'topics', on which each 'node' could 'publish' a message through a '*topic*' and another 'node' could '*subscribe*' that. This streaming communication tool provides ROS with the ability to 'glue' programs from different platforms, such as a 'node' written in C++ could easily communicate with a 'node' programmed in Python.
Another characteristic of ROS is the fact that it supports multiple-machine communications through the *TCP protocol*. Practically, a person can create a 'node' on a computer and communicate with another 'node' on another machine easily by ROS. This characteristic plays an important role in our system's structure. As you saw in the brief graph, our system involves a large amount of programs running asynchronously and requires communication between each other.

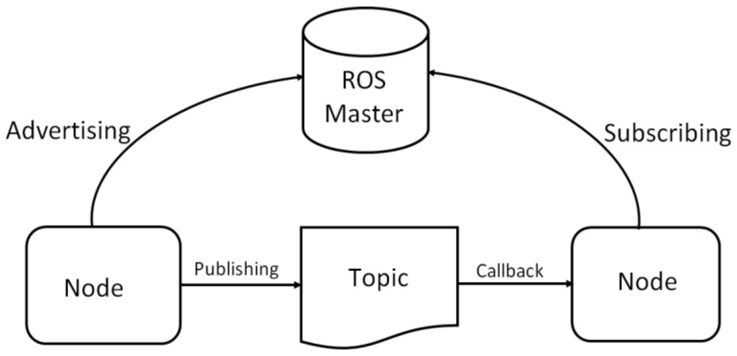


Fig 2.6.1 A typical ROS Model: System components.

2.7 Reality Testing

After the test in a fixed camera position, which was the scenario that person re-identification was designated for, we will test person re-identification in reality on a robot to confirm if reality scenarios will affect person re-identifications performances, such as luminous environment changes, background moving, or camera angle.
The robot we will be using to test is the same as the cover one.

2.7.1 Stage 1

We've designed two stages to test our robot. Each stage will be focusing on testing different subjects. The test arena will be divided into different areas, and each area will have some reality scenario simulations for the robot to face. We will test if the robot could still pass the area without misidentifying the target in 10 tests per stage.

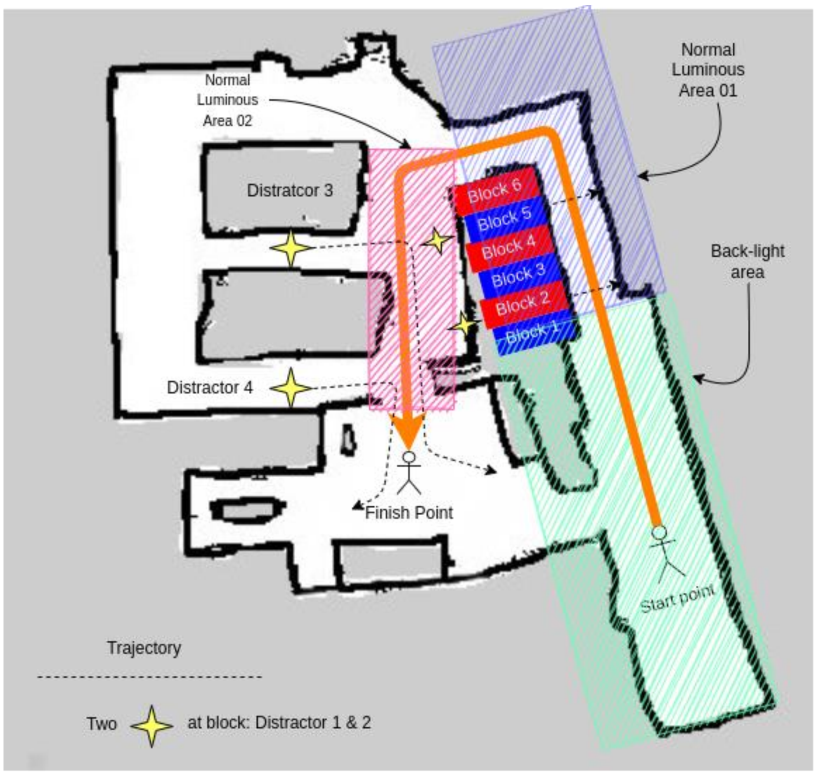


Fig 2.7.1.1 Stage 1 testing environment

As shown in the figure, Stage 1 is divided into 3 areas: Back-light area, Normal luminous area 01 and Normal luminous area 02. The robot is tested for its performance of following the target consistently while some non-target pedestrians occasionally pass by. Note that all lights are on throughout Stage 1. The test starts at the Back-light area, where the robot starts following the main target person at the start point. A large luminous object is placed at the upper front of the target, in order to see if the robot will be affected by that. The target person will then follow the orange trajectory to the end and passing areas. Before entering Normal luminous area 01, there will be no other people trying to appear before the robot, testing if its 'attention' would be drifted away; in other words, only the robot and the main target person being followed were present.

In Normal luminous area 01, two testers as ‘distractors’ do their job in this narrow lane by crossing between the target and the robot. The starting point of distractor 1 & 2 will be at block m & n, generated randomly at the beginning of the test.

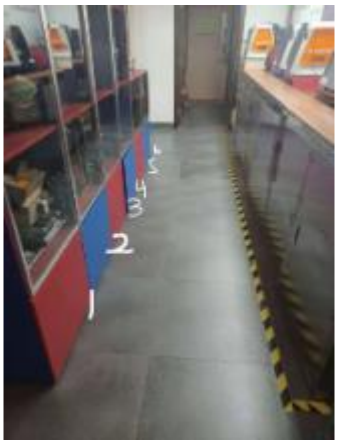


Fig 2.7.1.2 The blocks

When the main target experimenter passes by, the interference experimenters will walk between the robot and the main target experimenter. The interference experimenters will stop for 0.5 second while the main target experimenter keeps walking, blocking the view of the robot temporarily.
In Normal luminous area 02, the distractor 3 first comes in between the robot and the main target person and walks with the main target person on the left. Distractor 4 does the same shortly after and Stage 1 is finished.
In Stage 1, we have each test for approximately 1m08s. Technically, our method could recognize both sides of the main target person so they could watch the robot following them. But in this test, we hope that the main target person could let its back open for the robot to follow them. Moreover, the system gains more accuracy for the main target person to look at it when it is following. So, if the robot requires the main target person to turn back to let the robot find it back for more than once, that area will be estimated as failed.
However, there is still a special case that our main target person could turn back to let the robot follow them: the big U turn between Normal luminous area 01 and 02. As our robot currently does not have an obstacle avoiding system, the main target person could check on the robot when doing that U turn but not be considered as one.
Stage 1 will be focusing on testing normal pedestrian crossing situations, and a back-light situation to test may luminous affect person re-identification.

Here are the test results of Stage 1. As distractor 1 & 2’s stand point was random, the standing position displays each test’s stand point.

Normal Luminous Situation					
Test No.	standing position	Normal luminous area 02	Normal luminous area 01	Back-light area	remark
1	5 6	✓	✓	✓	
2	6 3	✓	✓	✓	
3	4 1	✓	✓	✓	
4	1 6	✓	✗	✓	The robot misses

					at last
5	5 2	✓	✓	✗	Poor light
6	3 5	✓	✓	✓	
7	3 4	✓	✓	✓	
8	2 1	✓	✓	✗	Poor light
9	5 4	✓	✓	✓	
10	4 6	✓	✓	✓	

Obviously, we could see that the robot could pass Normal luminous area 02 in all tests and 01 with only 1 failure. But we could see that there are two failures in the back light area. After checking the recorded testing video, we found out that a person re-identification missed in those rounds when it was passing that large luminous object. Regarding this, we redesigned stage 2 to focus on testing luminous environmental changes.

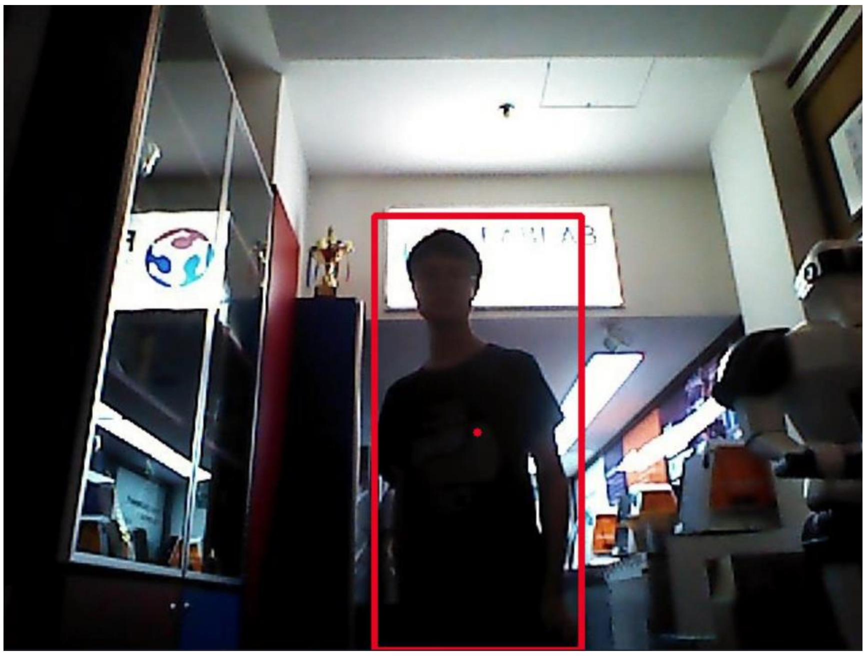


Fig 2.7.1.3 The Back-light area target lost capture

2.7.2 Redesigned Stage 2

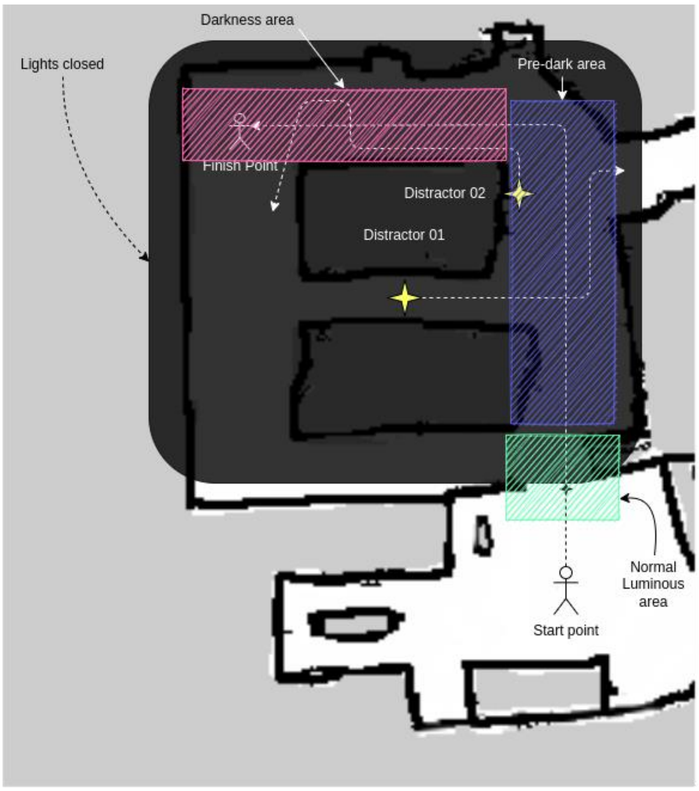


Fig 2.7.2.1 Stage 2 environment

After the redesignation, stage 2 will now be focusing on testing will luminous environmental changes affect the person re-identification system. We have 3 brand new areas two test our robot: Normal luminous area, pre-dark area and darkness area. The target will follow the yellow trajectory and there will still be distractors showing by the stars. But, The lights of the pre-dark area and Darkness area will be TOTALLY off. Pre-dark areas will still have a tiny bit of luminous due to light glowing at Normal luminous areas. But darkness will be completely dark.

Here are the test results of Stage 2

Luminous Changing Situation				
Test No.	Normal luminous area	pre-dark area	Darkness area	remark
1	✓	✓	✗	Can't recognize at last
2	✓	✗	✓	Dark
3	✓	✓	✓	
4	✓	✓	✓	Testers will stand for a chat at Area 03
5	✓	✓	✗	Can't recognize at last
6	✓	✓	✓	
7	✓	✓	✗	Can't recognize at last

8	✓	×	×	
9	✓	✓	☆✓	☆ : The tester has to turn around 1 time at last seconds
10	✓	✓	☆✓	☆ : The tester has to turn around 1 time at last seconds

We do not have to talk much about the performance of the Normal luminous area, so we will be focusing on the pre-dark and darkness area. As we could see, there are two failures in the pre-dark area due to the target being lost by person re-identification. But we could see that more failures were encountered in the Darkness area, causing more failures compared to stage 01.
In conclusion, these experiments claimed that our method still requires improvement on handling changing luminous conditions. But on the other hand, we could observe that these failures were caused by the target being *missed*, instead of the target being *misidentified*. Moreover, the object trackers can hardly *re-track* the target after the miss even if the target tries to proactively draw the tracker's attention. But person re-identification, as its name, could *re-identify* the target after the miss. As we do not allow drawing attention in our tests, the experiments proved that person re-identification is a superior solution in this task.

3. Applications

I developed this project mainly to improve the original contest's demo program and apply to the contest. The result was a huge success. Outside the competition, I tried to think of where I could apply this person follower into something else, and here are a few of them
One good application will be an automatic follow-you supermarket cart which carries your items when you're shopping in the supermarket (***Fig 3.1***).
Another application would be a luggage carrier inside an airport, as if you have a bunch of luggages, this could help you with that (***Fig 3.2***)
Some tasks require the robot and the human to do collab-work in unexplored areas. Obviously, this means map locating won't work in these unexplored areas. As if the task also indicates a human-being to join, person following will be crucial in this situation.

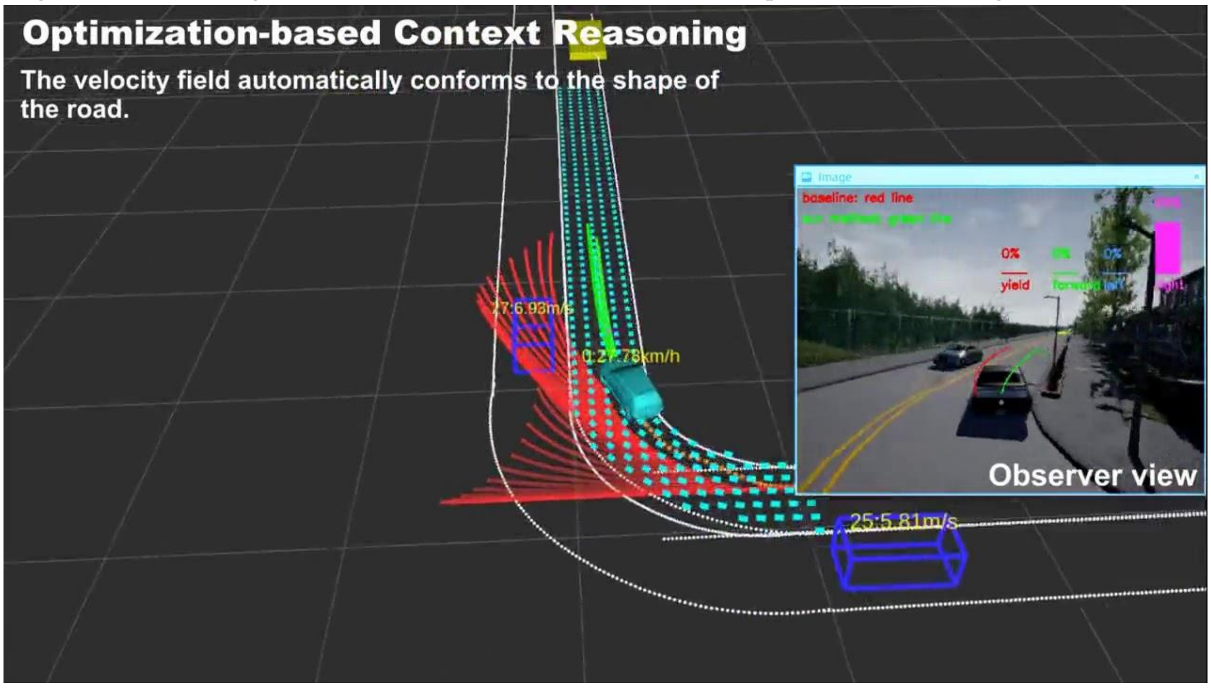
Fig 3.1 The automatic follow-me shopping cart example (Robot in image: Robovie-II)

Fig 3.2 An airport luggage carrier robot example (Robot in image: Care-E)

4. Future

4.1 Trajectory Prediction

As mentioned in system building, the follower will follow the person who is closest to the target's last existing point for 3 seconds. If the target misses, the robot will stop operating after 3 seconds losing the target until the target comes back into the robot's vision.
To solve this issue, we will be using SLAM to navigate through the known spaces in the area. If the target was lost, the system will use LSTM to perform trajectory prediction with pass trajectories the target has been through, after that, the robot will move to the endpoint to find the target


Optimization-based Context Reasoning
The velocity field automatically conforms to the shape of the road.
27:6.93m/s
0:27.78km/h
25:5.81m/s
baseline: red line
yield
Observer view

An example of vehicle trajectory prediction

4.2 GUI Friendly

In future days, we will let these robots operate in my school or other scenarios to service persons. For this scenario, we must have a GUI (Graphic User Interface) for users to use it easier

4.3 Security System

If anybody could register the robot to follow them, this would be dangerously insecure. So In order to let the robot re-register a target, the old user must deactivate the robot by using bionic features, such as fingerprint or face.

4.4 Brightness problem

The very main problem of our current system will be the poor night situation.
As the person was totally dark in this situation, it was hard for person re-identification to extract usable information from this blacked-out person.
We recently found a method called Zero-DCE (Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement)[10], which can increase the brightness of the image to normal.

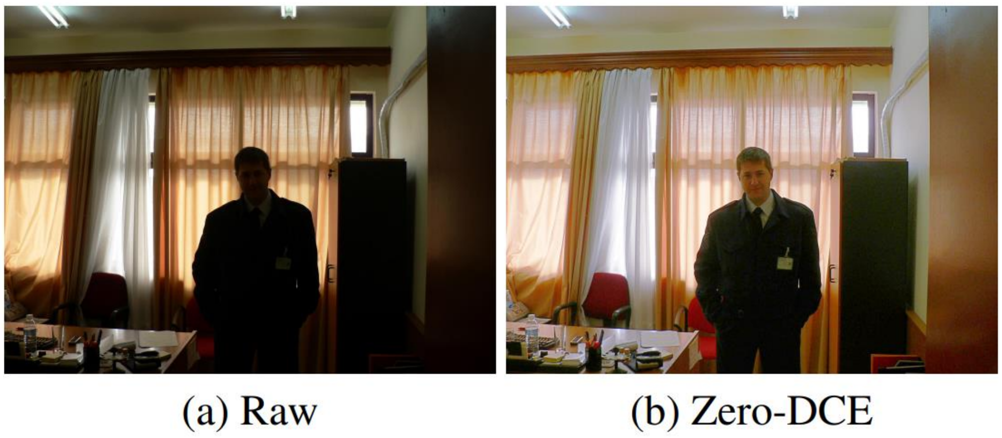


Fig 3.1 Zero-DCE used on the RAW image to increase brightness

Currently, we have tested this method and found it very compatible with our situation, and it has successfully enhanced one of our backlight situation image sample

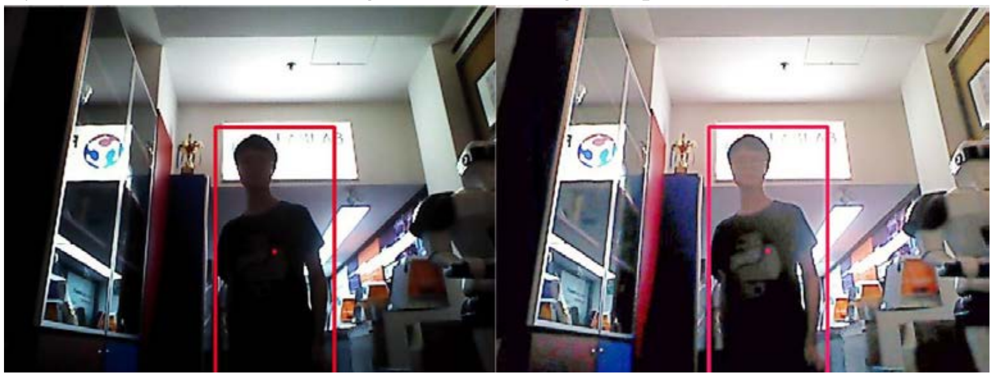


Fig 3.2 Left, sample image; Right, enhanced image

But there is only one problem with Zero-DCE: if we input a normal image to it, it will still increase the brightness and make it become worse.

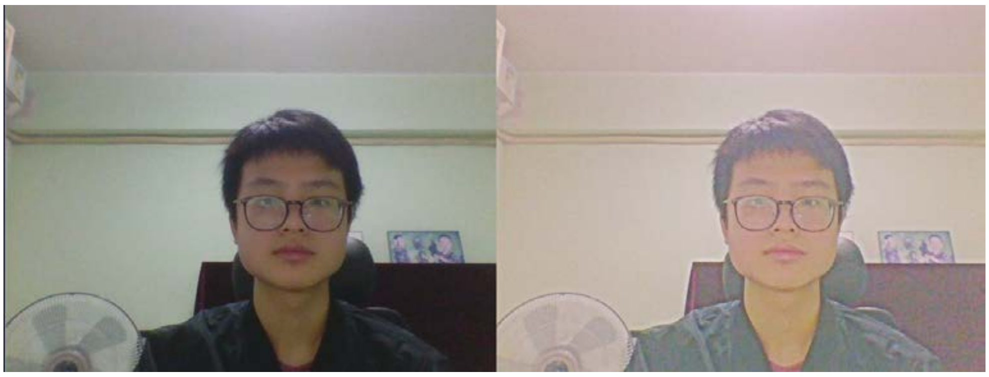


Fig 3.3 An normal image brightness improved by Zero-DCE

As with this characteristic, we have to find a way to determine whether the image was in a poor light situation.

4.5 The Robot Dog

A month ago, our lab has brought a robot dog called unittree v1

This dog has a powerful radar navigation system and flexible legs to walk in different scenarios. we may include our person follower into this robot dog, making it follow a person like a real dog.

Reference

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, May 9). You only look once: Unified, real-time object detection. arXiv.org. Retrieved April 12, 2021, from https://arxiv.org/abs/1506.02640.
[2] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016, December 29). *SSD: Single shot multibox detector*. arXiv.org. Retrieved November 25, 2021, from https://arxiv.org/abs/1512.02325.

[3] Dai, J., Li, Y., He, K., & Sun, J. (2016, June 21). *R-FCN: Object detection via region-based fully convolutional networks*. arXiv.org. Retrieved November 25, 2021, from https://arxiv.org/abs/1605.06409.
[4] Rosebrock, A. (2021, April 17). *OpenCV object tracking*. PyImageSearch. Retrieved November 25, 2021, from https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/.
[5] Freund, Y., & Schapire, R. E. (1999). *A short introduction to boosting*. CiteSeerX. Retrieved November 25, 2021, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.5846.
[6] Henriques, J.F., Caseiro, R., Martins, P., & Batista, J. (2015). High-Speed Tracking with Kernelized Correlation Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37, 583-596.
[7] Lukežič, A., Vojír, T., Zajc, L.Č., Matas, J., & Kristan, M. (2017). Discriminative Correlation Filter with Channel and Spatial Reliability. ArXiv, abs/1611.08461.
[8] Host, K., Pobar, M., & Ivašić-Kos, M. (2020, January). *Tracking handball players with the DeepSORT algorithm*. ResearchGate. Retrieved November 25, 2021, from https://www.researchgate.net/profile/Marina-Ivasic-Kos/publication/340364883_Tracking_Handball_Players_with_the_DeepSORT_Algorithm/links/5e8af789a6fdcca789f7f472/Tracking-Handball-Players-with-the-DeepSORT-Algorithm.pdf.
[9] Izutov, E. (2018, December 6). Fast and accurate person re-identification with rmnet. arXiv.org. Retrieved 2021, from https://arxiv.org/abs/1812.02465.
[10] Guo, C., Li, C., Guo, J., Loy, C. C., Hou, J., Kwong, S., & Cong, R. (2020, March 22). *Zero-reference deep curve estimation for low-light image enhancement*. arXiv.org. Retrieved November 25, 2021, from https://arxiv.org/abs/2001.06826.

【評語】190038

The project proposed a person following service robot that utilizes the state-of-the-art object detection method to achieve good performance of people re-identification. The core technique of the proposed work was implemented and evaluated. A proof-of-concept testbed was conducted to justify the feasibility of the proposed work. Overall, this is solid work and the result are very promising. Congratulation on the nice work done!