DEVELOPMENT OF A LOW-COST ROBUST AUTONOMOUS APPROACH TO MODERN CAMPUS SECURITY

Isaac Tan Jia Le[1], Kuai Yi Zhuang, Ezann[1], Neo Hao Jun[1], Tan Yong Leng Kelvin[1], Darius New[2]

[1] Hwa Chong Institution (College), 661 Bukit Timah Road, Singapore 269734
[2] DSO National Laboratories, 12 Science Park Drive, Singapore 118225

ABSTRACT

Unmanned Ground Vehicles (UGVs) provide a novel alternative to conventional campus surveillance methods, including manual patrol and camera surveillance. This project aims to develop a low-cost, robust, and modular UGV for campus surveillance, allowing campus security to remotely monitor different areas efficiently. The UGV, through the usage of a Simultaneous Localisation and Mapping algorithm, is able to map and self-navigate. Through testing, we have determined optimal planning and locomotion controllers for the UGV to operate in urban environments. Afterwards, the UGV conducts computer vision (CV) tasks locally, in order to detect any potential intruders in its line of sight, through the usage of a pose estimation model. Images of detected intruders would then be passed through another pose classification model, to determine any further suspicious movement.

INTRODUCTION

Due to the large size of modern campuses, the advent of advanced autonomous monitoring technologies presents an opportunity to realise enhanced efficiency in the management of campus grounds. As opposed to traditional surveillance methods such as closed-circuit television systems, Unmanned Ground Vehicles (UGVs) take a more proactive approach against campus intruders, trained to recognise and follow unidentified trespassers on the fly, rather than merely being used for static surveillance.

Commercial off-the-shelf (COTS) autonomous surveillance systems do already exist in the market, yet such systems can cost upwards of 70,000 USD (93,160 SGD) to rent per year, which is too high a price for these systems to become commonplace. This project thus seeks to develop a low-cost and robust autonomous land vehicle which can be easily adapted for a range of campus surveillance purposes, boosting current monitoring efforts through the use of a novel trespasser detection architecture and feedback system. This can help organisations to accurately monitor their compounds and quickly notify campus security of anomalous situations so that the situation can be resolved more promptly.

METHODS

Hardware

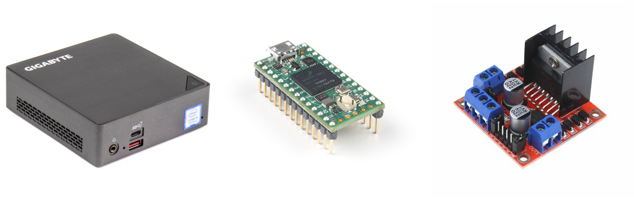


Figure 1. *GIGABYTE BRIX | PJRC Teensy 4.0 | L298N Dual H-bridge Motor Driver*

A custom cost-effective four-wheeled autonomous robot was developed for the purpose of this project, powered by the compact GIGABYTE BRIX PC and a Teensy 4.0 Development Board running on Robot Operating System 2 (ROS2). The chassis is powered by four Pololu 34:1 6V Metal Gear Motors which are controlled by two L298N motor driver modules.

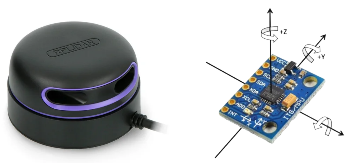


Figure 2. *SLAMTEC RPLIDAR A2M12 | MPU 6050 Accelerometer and Gyroscope*

In order for the UGV to localise and map itself in foreign environments, a SLAMTEC RPLIDAR A2M12 is used to generate a 2D point cloud of the UGV's surroundings that is passed through a Simultaneous Localisation and Mapping (SLAM) algorithm. Data from the motor encoders and the MPU6050 Inertial Measurement Unit (IMU) are also fused to provide odometry data that aids the robot's navigation capabilities.

Figure 3. *Luxonis OAK-D Pro Camera.*

The Luxonis OAK-D Pro camera, pictured above, is used to provide the UGV with the ability to detect potential intruders and provide data on the position and distance of any detected intruders, through the usage of a high-resolution colour camera with on-device Neural Network inferencing and CV capabilities. The OAK-D Pro also features night vision using its flood mode setting, which can increase perception during low-light and no-light environments, enhancing the robot's visual capabilities during night surveillance missions.

Localisation

In order for the UGV to adapt effectively to a wide variety of modern campus environments, the preexisting line-tracking system is replaced with a more robust Simultaneous Localisation and Mapping (SLAM) SLAM solution. This enables the UGV to self-localise and map out its surroundings with ease, meaning that it can be easily deployed across different environments with minimal setup. We decided to utilise the ROS2 SLAM Toolbox which uses a sparse graph optimization with loop closure detection. The graph-based SLAM solution is chosen as it is less computationally intensive as compared to other methods and is better at loop closure, allowing for errors in odometry data to be easily corrected. Patrolling is also a repetitive task, and the maps generated can be reused over multiple runs.

Navigation
As the UGV is intended for night-time surveillance where surrounding conditions are relatively still and little variation in the landscape is expected, Dijkstra's algorithm was selected to plot paths for the robot to patrol around any desired areas. As a relatively simple path planning algorithm, it is largely unable to account for any significant changes in the current generated map, which we will account for by using collision detection functions in our locomotion controllers. Ultimately, the simplicity of Dijkstra's algorithm frees up computational resources for other more mission-critical tasks, which is found to deliver more overall utility.

Human Detection & Classification
A human detection and pose estimation algorithm was selected to identify intruders and allow the robot to chase after them. After this, the image is piped through a classification model that differentiates between a walking and running human to set the speed at which the patrol robot should pursue the intruder at. For this, we chose to use MediaPipe BlazePose to do simultaneous human detection and pose estimation, leveraging the Oak-D Pro's ability to detect depth. BlazePose lite had lower latency and better frame rates than MoveNet Lightning (used by the previous project). For classification, we utilised a Multi-layer Perceptron pose classifier network, with BlazePose's pose landmarks as inputs. Using the running and jogging videos from the KTH action database, each frame of the videos is extracted and run using BlazePose to obtain the pose landmarks. The entire dataset contains around 41,000 images, with an 8-8-9 train-validation-test split with respect to the subject in the video. Additionally, a custom dataset containing 1,340 images was created to simulate a real-life environment.The network was trained for 66 epochs, with early stopping enabled after validation accuracy did not improve for 20 epochs. To ensure we did not overfit, the validation accuracy trend was observed (Appendix B).

RESULTS AND DISCUSSION
Hardware (Chassis)

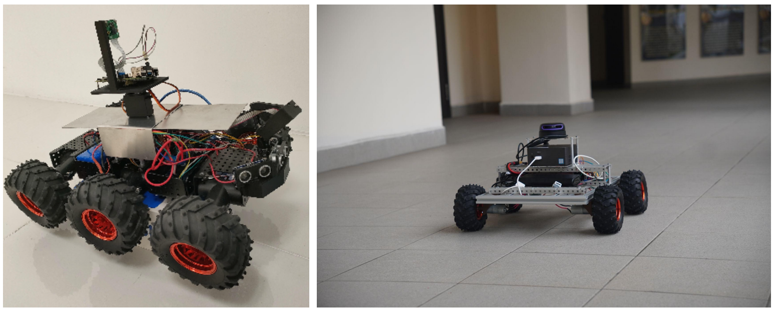


Figure 4. *Original chassis from previous iterations of the project | Revamped Chassis*

As pictured above, the original chassis inherited by this project is a Dagu Wild Thumper 6WD All-Terrain Chassis priced at 311 USD (410 SGD). It had a large camber due to a faulty suspension and lacked favourable mounting points for peripherals.

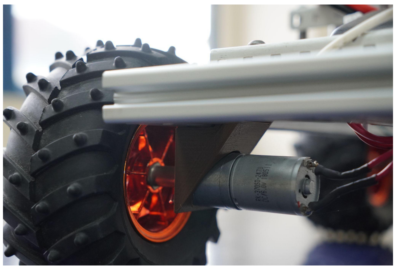


Figure 5. *3D-printed motor mount attached to the 2020 T-slot aluminium extrusion.*

The revamped chassis consists of a frame made of six 2020 T-slot aluminium extrusions connected by 90-degree angle brackets, upon which all payloads are mounted. The high strength-to-weight ratio and excellent corrosion resistance of the aluminium extrusions make for a sturdy, yet light, build that performs well in outdoor environments. As seen above, the drive system is attached to the frame via a custom 3D-printed motor mount, incorporating triangular supports and a dual-point mount to achieve excellent load handling in a compact package. Through our experiments, such a chassis design is found to be able to support all the necessary peripherals and move smoothly under load, making it a cost-effective and more readily available alternative to the original COTS system.

Hardware (Components)

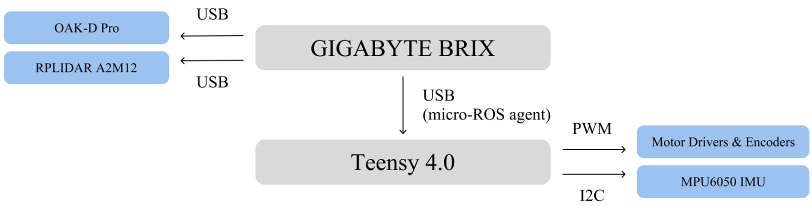


Figure 6. *UGV's system overview diagram.*

The UGV runs on ROS2, taking advantage of the sizable computing power of the BRIX to run the necessary high-level algorithms for autonomous area surveillance with low latency. ROS2 allows the transfer of data via system-wide ROS2 topics, enabling the integration of the various subsystems, namely localisation, navigation, and CV. For lower-level control, the BRIX communicates with an external Teensy 4.0 through the micro-ROS interface with the help of a Micro USB cable. The Teensy then interprets the velocity commands it receives into pulse-width modulation (PWM) commands so as to control the motors through the motor drivers and outputs odometry data to the BRIX through the fusion of the data it receives from the encoders and IMU on the digital and inter-integrated circuit (I2C) lines respectively.

In total, the cost of the system is kept at an estimated 1,800 SGD, significantly lower than that of commercially available alternatives. Moreover, as this particular system largely reused pre-existing components, the total cost of development is further reduced below 1,000 SGD. A more comprehensive breakdown of the costs can be found in Table 1 in Appendix A.

Localisation

Various tests were conducted within the school campus, which are areas that are initially unknown to the robot. The figures below showcase the actual environment and the maps generated by the SLAM algorithm.

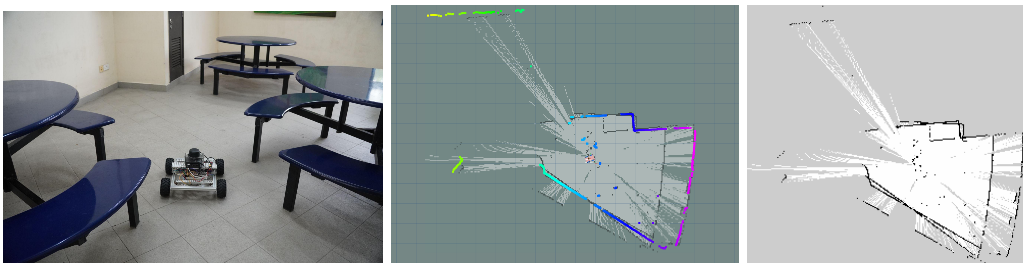


Figure 7. *UGV in school campus with obstacles | Generated 2D point cloud | Final generated map*

Through the various tests conducted, data supplied by the A2M12's 12-metre range and maximum scan frequency of 15Hz is found to be more than adequate in enabling the UGV to conduct SLAM and generate a good approximation of the environment. After a single run dedicated to generating a map of the area, the map was loaded back into the localisation algorithm and the robot was able to know its pose within the map through loop closure. This proved to be good enough for online navigation for the robot during the patrolling task.

Navigation

Using the generated map from SLAM, we were able to use Dijkstra's algorithm to rapidly find a path for the robot to follow that was obstacle-free, see figure below.

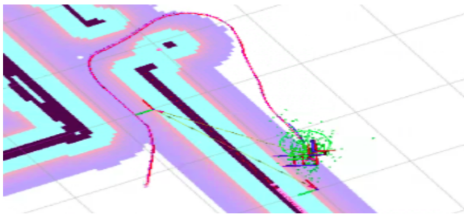


Figure 8. *Example of a path generated by Dijkstra's algorithm on ROS2.*

However, Dijkstra's algorithm assumes a well-controlled environment in path generation and does not account for inaccuracies. Since the UGV will be operating in outdoor conditions on a variety of terrain, the error probability is high and inconsistencies between the projected path set by the path planner and the UGV's actual movement are likely to accumulate to a sizable error over time. It is therefore necessary to define a pre-computed path and direct the UGV's movement onto that path via a movement controller.

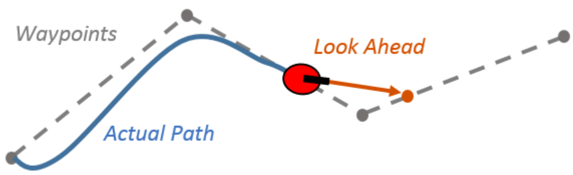


Figure 9. *Example of a pure pursuit algorithm tracking a set of pre-defined waypoints.*

The controller selected is a pure pursuit algorithm, a tracking algorithm that works by calculating the required velocities that will move the UGV from its current position to a goal position while staying on pre-computed paths. The algorithm constantly looks for a point (the look-ahead point) on the generated path a set distance away from itself, generates linear and angular velocities to pursue the look-ahead point, and constantly updates the look-ahead point as it moves along the path. To further improve performance, the Regulated Pure Pursuit controller is used to include the ability to predict potential collisions ahead of time and ensure that the robot can better regulate its velocity as it moves turns of sharp curvature. This allows the UGV to better navigate the tight spaces in modern campus compounds.

Human Detection & Classification

BlazePose Lite was able to accurately identify humans and their landmarks, and we were able to merge that data with the Oak-D Pro depth sensing capabilities.

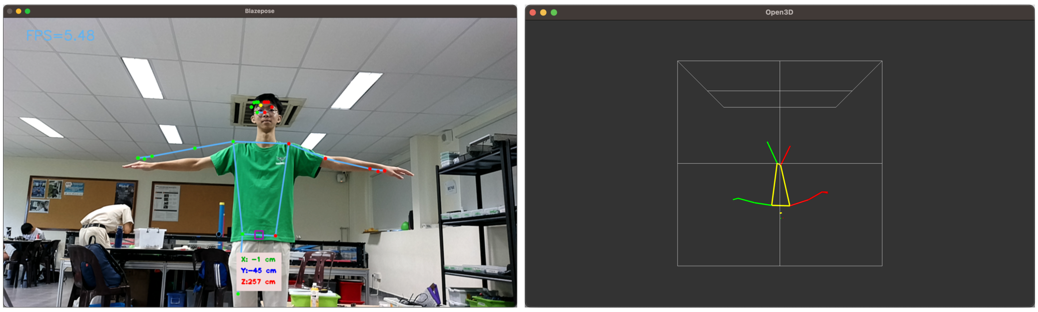


Figure 10. *Annotated example of pose detection by the model | The detected pose rendered in a 3D space*

Generally, BlazePose identifies landmarks largely accurately. Even under more unideal conditions such as with occlusions, BlazePose is still able to perform well.

Figure 11. *Examples of accurate landmark estimation in non-ideal conditions, with partial occlusion (left), the subject holding an object (middle), and a blurry image (right).*

However, a challenge the model faced was face occlusion. This is due to the nature of Blazepose's architecture - a face is used as a proxy for the pose detector. While this allowed the model to be more efficient and robust against false positives, this sometimes led to misidentification of landmarks or simply having false negatives, with no landmarks being detected at times.

Figure 12. *Accurate landmark detection with face occluded (left) | Misidentification of landmarks with face occluded (middle) | No landmarks detected with face occlusion (right)*

On the KTH test dataset, the model had a mean 90.9% accuracy. The model generally over-classified images to the jogging class, with the walking class having a greater precision than the jogging class and the jogging class having a higher recall. On the custom dataset, the accuracy fell to a mean of 74.0%, with the over-classification of the walking class as jogging continued to a stronger extent. (Appendix C)

Analysis of the misclassified images suggests that the model relies heavily on arm position to classify an image, with the model classifying an image as “jogging” when the subject’s arms are raised and bent. Conversely, the model classifies an image as “walking” when the subject’s arms are lowered and straight. However, this contributed to the overclassification of images to the “jogging” class, especially in the case where the subject was holding up objects. This can be seen with the model only attaining 28.4% accuracy on images in the custom dataset where the subject was walking while holding up an object. Further samples of classification based on the subject’s arm position can be found in Appendix D.

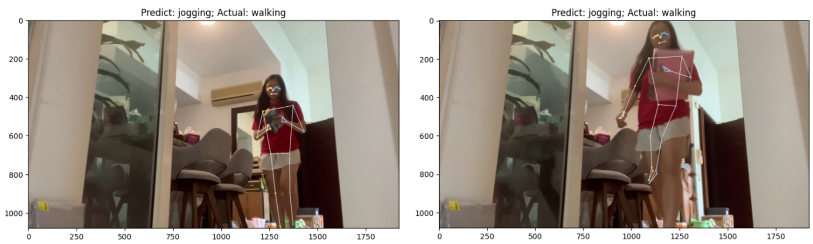


Figure 28. *Examples of the model misclassifying images where the subject was walking while holding up an object*

This is likely due to how the model was trained on the KTH action dataset, which had no videos of subjects holding up objects. To overcome that, a larger and more varied dataset could be created to train the model on, including images where the subject is holding objects.

CONCLUSION

As the adoption of mobile UGVs for surveillance continues to grow, this project serves as a proof of concept that the use of such vehicles need not come at a large financial expense. A working prototype has been designed and created at less than 2 percent of the cost required to rent an average surveillance vehicle for a year, lowering access barriers to the benefits of novel surveillance technology.

Moreover, equipped with the ability to autonomously navigate in foreign environments and a novel, reliable, and accurate trespass detection system, the UGV requires minimal time investment to begin patrolling new areas and can efficiently survey large areas thoroughly, leveraging technological advancements to reduce manpower costs and the risk of human error in modern campus security.

In addition to our prototype, the ability to include human-in-the-loop interfaces to allow the security officers to designate patrol routes, be notified of anomalies detected during patrol and for them to decide if the robot should follow and chase down the intruder or not would greatly benefit them.

ACKNOWLEDGEMENTS

We would like to thank and express our deepest gratitude to our mentors, Mr Darius New and Dr Tan Yong Leng Kelvin, for their invaluable support and guidance for the entire duration of this project. Without their mentorship, this paper would not have come to fruition.

It would be remiss if we did not thank Ms Yao Sihan Joyce, Mr Phoon Zhi Jun Joey, and Ms Sim Yu Jin Emeline for their assistance in the administrative portion of this project, as well as DSO National Laboratories for providing us with the opportunity to participate in Research@YDSP 2023.

REFERENCES

Bazarevsky, V., & Grishchenko, I. (2020, August 13). On-device, real-time body pose tracking with MediaPipe Blazepose. https://ai.googleblog.com/2020/08/on-device-real-time-body-pose-tracking.html

Cohen, J. (2022, November 2). *Computer Vision for Multi-Object Tracking.* https://www.thinkautonomous.ai/blog/computer-vision-for-tracking/

Coulter, R. (1992). Implementation of the Pure Pursuit Path Tracking Algorithm. Technical Report CMU-RI-TR-92-01, Carnegie Mellon University, 13. https://www.ri.cmu.edu/pub_files/pub3/coulter_r_craig_1992_1/coulter_r_craig_1992_1.pdf

Farivar, C. (2021, June 27). Security robots expand across U.S., with few tangible results. *NBC News.* https://www.nbcnews.com/business/business-news/security-robots-expand-across-u-s-few-tangible-results-n1272421

Khalid, A. (2013, June). Autonomous Patrol and Surveillance System (APSS)–A Student Project to Help Aid the Campus Police. In *2013 ASEE Annual Conference & Exposition* (pp. 23-237).

Luxonis. (n.d.). *OAK-D Pro.* https://docs.luxonis.com/projects/hardware/en/latest/pages/DM9098pro/#dm9098pro

S. Koenig and M. Likhachev (2002). D* Lite. In *AAAI Conference of Artificial Intelligence (AAAI)*, 476-483. http://idm-lab.org/bib/abstracts/papers/aaai02b.pdf

S. Macenski, S. Singh, F. Martin, J. Gines (2023, May 31). Regulated Pure Pursuit for Robot Path Tracking. Autonomous Robots, 2023. https://doi.org/10.48550/arXiv.2305.20026

Macenski, S., Jambrecic I., "SLAM Toolbox: SLAM for the dynamic world", Journal of Open Source Software, 6(61), 2783, 2021.

Chung, Jen-Li, Lee-Yeng Ong, and Meng-Chew Leow. 2022. "Comparative Analysis of Skeleton-Based Human Pose Estimation" Future Internet 14, no. 12: 380. https://doi.org/10.3390/fi14120380

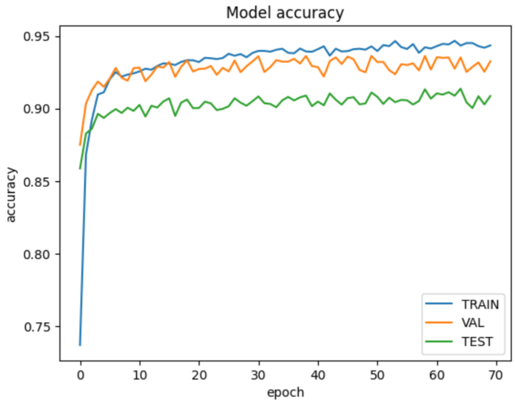
APPENDICES

Appendix A

Table 1. *Breakdown of UGV component costs.*

Item	**Cost (SGD)**
GIGABYTE BRIX GB-BKi7HA-7500 (rev. 1.0)	800.00
Teensy 4.0 Development Board	32.00
Luxonis OAK-D Pro	460.00
SLAMTEC RPLIDAR A2M12	300.00
L298N Motor Driver Module (2 sets)	4.00
MPU6050 Inertial Measurement Unit	4.00
Aukey PB-Y37 20000mAh 65W PD Powerbank	72.00
Elements 7200mAh 100C 7.4V Lipo Battery (Hardcase)	70.00
Type C to DC Cable	22.00
Chassis Materials	20.00
Miscellaneous Parts	16.00
Total Cost	**1,800.00**

Appendix B



Model accuracy trend of training, validation and test sets.

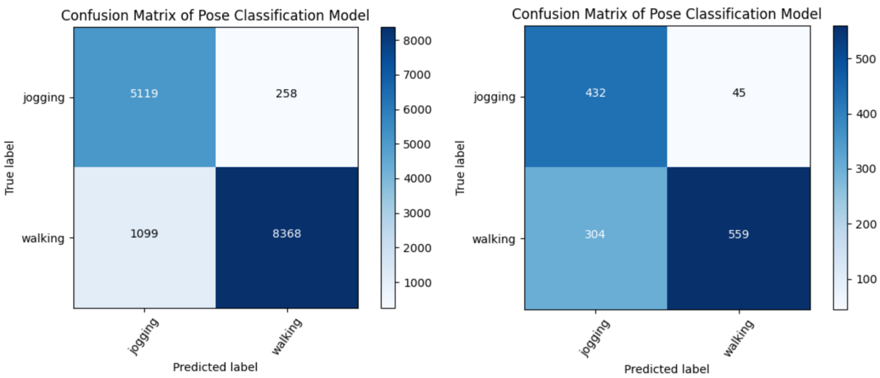
Appendix C

Table 2. *Precision, Recall, and F1 Score of KTH Test Dataset.*

	Precision	**Recall**	**F1 Score**
Jogging	0.82	0.95	0.88
Walking	0.97	0.88	0.92

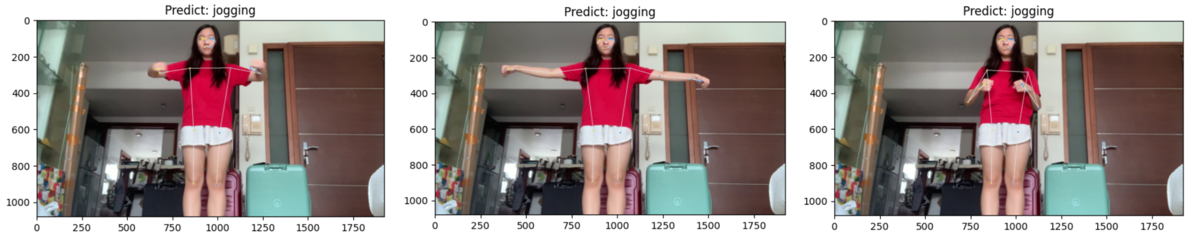
Table 3. *Precision, Recall, and F1 Score of custom dataset.*

	Precision	**Recall**	**F1 Score**
Jogging	0.59	0.91	0.71
Walking	0.93	0.65	0.76

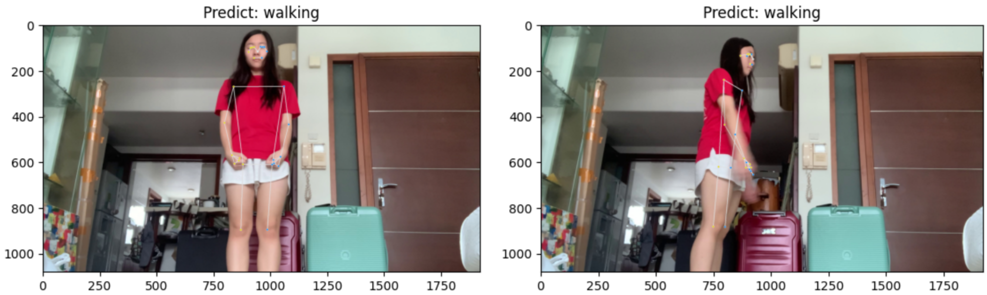


Confusion matrix on test dataset (left) | Confusion matrix on custom dataset (right)

Appendix D



Further examples of the model classifying images as "jogging" based on having raised and bent arms.



Further examples of the model classifying images as "walking" based on having lowered and straight arms.