Solution Brief

Medical and Life Sciences

Gwell Medical (mRobot) Optimizes Hospital Logistics Robotic Solution Based on Intel® Architecture

Authors:

Zhen (Fiona) Zhao
Deep learning technical consulting engineer, Intel Corporation

Lei Zhang
Software algorithm engineer, Gwell Medical

Xuesong Shi
Research scientist, Intel Labs China, Intel Corporation

Dmitry Kurtaev
Deep learning software engineer, Intel Corporation

Gan Zhang
Navigation control development manager, Gwell Medical

Hengkang Liang
Hardware engineering director, Gwell Medical

Yiqun (Charlie) Hu
Account manager, China Intelligent Platform Sales, Intel Corporation

Qi (Scott) Wang
Solution architect, Healthcare and Life Science, Intel Corporation

The Noah robot is an excellent substitute for human labor. A logistics worker can usually handle up to 30 kg at a time, while the Noah robot can carry up to 300 kg. As labor costs increase and hospitals require more stringent logistics management, a traditional human labor approach will make it hard to meet these challenges. In the future, hospitals will have trends similar to factories: more robots will gradually assist with these changing demands.

– Wei Lü
Cofounder and CTO
Gwell Medical Technology Co., Ltd.

Gwell's Noah hospital robot uses Intel® technologies to perform many human tasks

Hospital business operations have significant logistics demands for the distribution of medications, consumables, food and beverages, and for waste removal. Technologies including robotics, computer vision, artificial intelligence (AI), and deep learning (DL) are automating logistics operations in hospitals to help improve efficiency and streamline human workloads.

To address these rapidly evolving needs, Shanghai Gwell Medical (formerly known as mRobot) Technology Co., Ltd. developed the Noah hospital logistics robot based on an Intel® Core™ processor, Intel® Movidius™ vision processing unit (VPU), and Intel® Distribution of OpenVINO™ toolkit. The Noah robot incorporates cutting-edge technologies such as autonomous navigation, elevator access, loading/unloading, and multirobot scheduling. It can support transportation tasks for operating rooms, pharmacy intravenous admixture services (PIVAS), specimen gathering, and other distribution links in a hospital.

Background: Autonomous robot for hospital logistics

Technologies used in autonomous driving are key for the deployment of logistics robots in a hospital. To explore the surrounding environment, a robot uses lidar, ultrasound, radar, and optical sensors to collect volumes of data. The robot crunches sensor data and large amounts of parallel and sequential computing data to develop a comprehensive and accurate understanding of the environment and evaluate possible scenarios, prioritize tasks, and then execute them.

In actual usages, a logistics robot needs to identify objects such as elevators, doors, and pedestrians and perform functions such as elevator entry and exit, door access, freight collection and delivery, and obstacle circumventing. For these autonomous driving functions, the robot needs the capabilities of perception, decision-making, and control and must overcome challenges such as target detection, motion planning, and simultaneous localization and mapping (SLAM).

Object detection is an essential and challenging task in computer vision. It requires the algorithm to predict a bounding box with a class label for each instance in the image.[1] For this purpose, a feature image pyramid is the basis of the solution chosen by many robotic object-detection designs. This algorithm allows the model to detect a wider range of objects by scanning at the position and on the pyramid hierarchy.[2] However, the low accuracy of this object-detection method is a problem for field deployments and may cause blockage and erroneous motion—especially in the case of multiple object tracking (MOT). To handle these complex cases in MOT scenarios, a unified MOT framework learning method is proposed to make full use of long-term and short-term cues.[3]

For indoor real-time logistics robotic positioning and mapping, GPS signals are often weak and inertial measurement units (IMUs) may drift as well. Therefore, a new robotic self-localization method is proposed to prerecord the scene, encode "landmark" visual features through visual recognition, and then perform localization by matching the perceived and stored landmarks.[4] Efficient lidar-based semantic SLAM is an effective method to facilitate the mapping process, along with semantic information integration and a 3D laser for range scanning.[5]

Obstacle avoidance and most-efficient route planning are important motion-planning requirements of autonomous logistics robots. As the output of the object detection and tracking module can be reasonably assumed to follow a multivariate Gaussian distribution, there is a new proposal for a safe path planning paradigm with a Gaussian process regulated risk map.[6] With a neural network–based heterogeneous sensor, depth imaging, and roll-and-pitch measurements, the traversability and collision avoidance of a robot can be calculated in real time.[7] Moreover, a potential collision can be detected in the trajectory by calculating the multidimensional trajectory from the start to an arbitrary target state and exploiting the piecewise polynomial formulation.[8]

Challenge: The transformation to automated hospital logistics

The daily diagnosis and treatment activities in a typical hospital require significant logistics. According to Gwell Medical's estimation, a hospital with 1,200 beds will consume or generate the following daily: 200 surgical instrument sterilization bags; 2,000 boxes of orally administered drugs; 2,514 blood samples; 5,800 large infusion bags; 1,200 hospital gowns; and approximately 2,200 other consumables. This equates to more than 20 tons of supplies and 3,000 deliveries, requiring 60 logistics workers.

The traditional logistics arrangement of a nurses' aide and trolley can present problems for staffers like dispatching errors and personal safety issues that come with heavy physical workloads, risk of radiation, and exposure to epidemic infections. These can increase operational costs. Automated robots can reduce these issues and are becoming a deployment priority in hospitals with heavy logistics workloads.

Today, a number of mechanical logistics systems are used in hospitals, including pneumatic tube systems (PTSs), track vehicle systems (TVSs), box-type conveyor sorting systems (BCSSs), and vertical logistics transportation systems. These systems can reduce logistics workloads and improve efficiency to some extent. However, implementation challenges can be numerous, including limitations on the size and weight of transported goods, significant building renovation requirements, long project cycles, and high costs for system upgrades.

With large loading, flexible usage scenarios, and quick rollout, autonomous robot deployment is a new trend in hospital logistics systems. A third-party survey showed that the percentage of hospitals in China conducting trial or real deployment of logistics robots is still small, with plenty of room for future development.[9] Logistics robots are expected to play a bigger role in hospitals in the future.[10]

Solution: Noah hospital logistics robot based on Intel® architecture

The Noah robot from Gwell Medical utilizes cutting-edge technologies, such as autonomous navigation, automatic elevator access, loading/unloading, and multirobot scheduling to quickly and securely perform transportation tasks for operating rooms, PIVAS, specimen gathering, and other distribution duties in hospitals.

Figure 1. The Noah hospital logistics robot. Image courtesy of Gwell Medical.

For robotic perception based on computer vision, Gwell Medical uses heterogeneous compute hardware including Intel Core processors, Intel Movidius VPUs, and the Intel Distribution of OpenVINO toolkit to simplify product deployment and increase deep learning inference performance with less power and low control latency.

Intel Core processors with Intel® HD Graphics provide a strong foundation for the Noah robot, with powerful general-purpose compute performance, availability, scalability, and security. Meanwhile, Intel Movidius VPUs provide a dedicated deep learning inference architecture with high performance at extremely low power. With high-quality image processing, computer vision, and deep neural network inference, the object detection in hospital logistics scenarios is seamlessly enabled at an optimal performance-to-power-and-cost ratio.

The Noah robot uses a Faster R-CNN network for object detection. In Figure 2, Faster R-CNN, derived from Fast R-CNN, consists of four parts: convolution layers, region proposal network (RPN), region of interest pooling (ROI pooling), and classifier. By replacing the original selective search method with RPN, Faster R-CNN automatically extracts more accurate candidate regions (region proposal). The number of candidate regions is reduced from 2,000 to 300 after using RPN, which improves the efficiency of object detection while ensuring accuracy.[11]

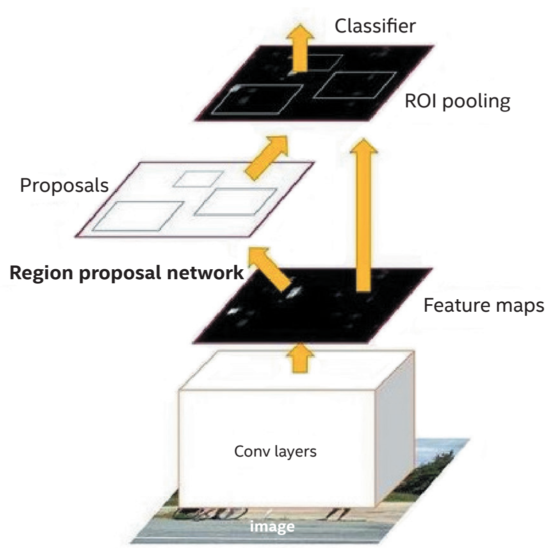

Figure 2. Basic structure of Faster R-CNN[12]

The OpenVINO toolkit offers highly optimized neural network computing performance and accelerates deep learning inference on Intel® hardware platforms including CPUs, iGPUs, VPUs, and FPGAs. Intel teams helped Gwell Medical optimize their Faster R-CNN network models generated from deep learning training, reduce conversion/deployment efforts from training to deployment environment, execute static model analysis, and adjust deep learning models for optimal performance on target devices.

With the Intel Distribution of OpenVINO toolkit, the Noah robot achieves optimized efficiency with improved accuracy for environmental perception, positioning, and tracking. The robotic solution can identify various types of human bodies, such as pregnant women, children, elderly and disabled people, and so on, as well as objects such as operating beds, instrument stands, wheelchairs, plants, and more in the hospital environment.

Solution optimization: Inference acceleration with the Intel Distribution of OpenVINO toolkit

The Intel Distribution of OpenVINO toolkit provides highly optimized deep learning inference on Intel hardware, including CPUs, iGPUs, VPUs, and FPGAs. The Noah hospital robot deployed the Intel® Deep Learning Deployment Toolkit to accelerate

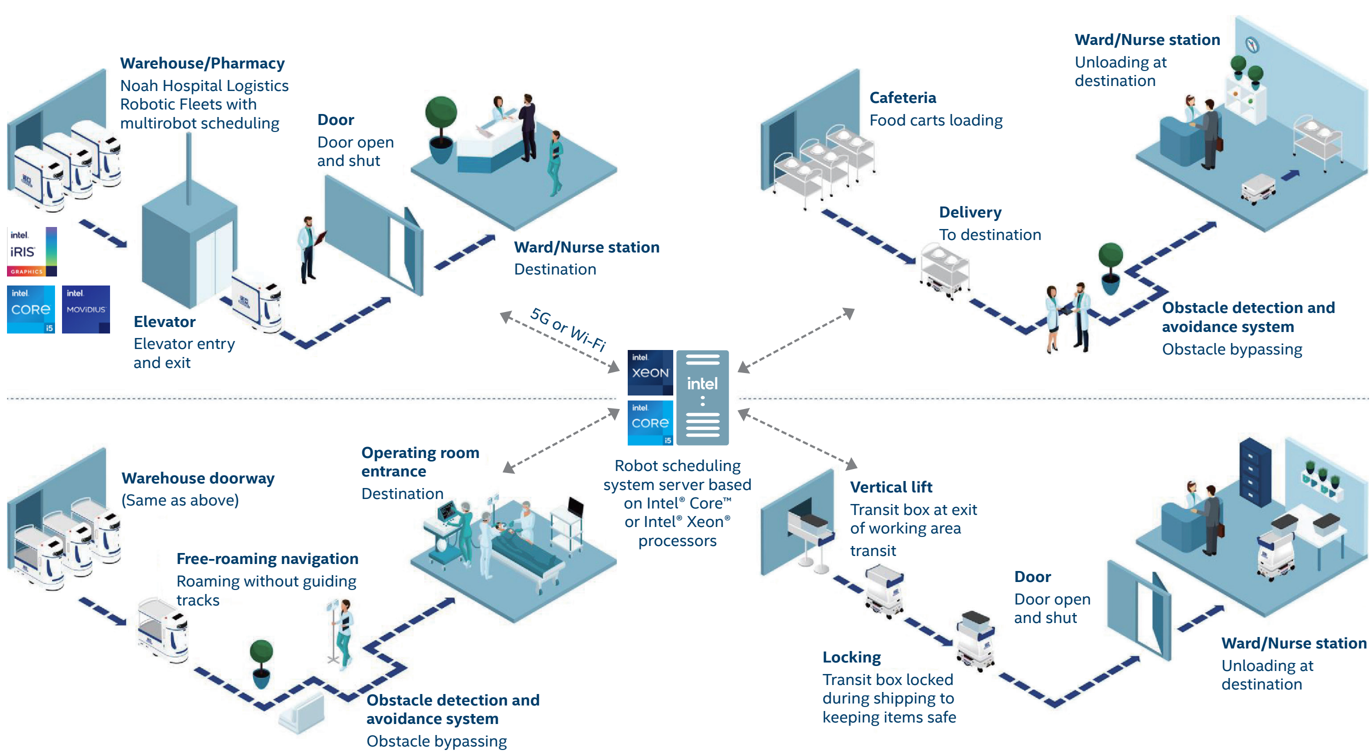

Figure 3. Hospital logistics usage scenarios for the Noah robot. Image courtesy of Gwell Medical.

inference performance. Shown in Figure 4, the Intel Deep Learning Deployment Toolkit consists of two modules: the model optimizer (MO) and the inference engine (IE). The MO, a cross-platform command-line tool, facilitates the transition from training (popular DL frameworks such as TensorFlow) to deployment, performs static model analysis, and provides optimal inference performance on the target devices.[13]

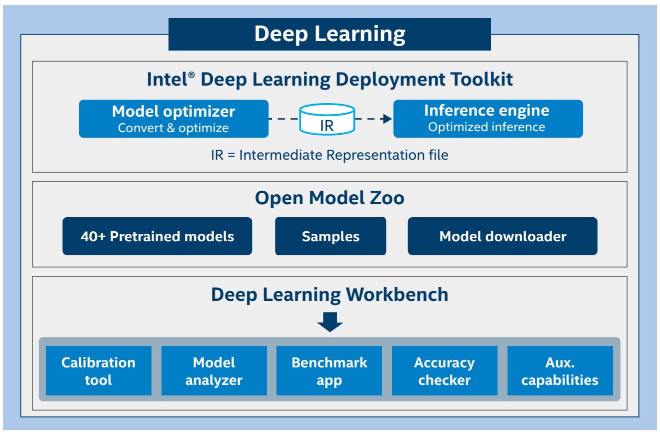

Figure 4. The deep learning inference components in the Intel® Distribution of OpenVINO™ toolkit

For deep learning networks, the intermediate representation (IR) file uses XML to describe topology and binary files to store network weight and bias values. The IE loads and executes a network model on edge devices based on the Intel® CPU and VPU. The IE provides unified, cross-platform C/C++ and Python APIs for inference. For the Intel CPU, iGPU, and VPU, the IE will load plugins from Intel® oneAPI's Deep Neural Network (oneDNN) Library, Intel® clDNN, and Intel® VPU runtime, respectively. As a dynamic function library, the CPU plugin detects the instruction set architecture (ISA) in the runtime and deploys just-in-time (JIT) code generation and ensures the optimization for the latest ISA.[14] The iGPU plugin implements the various types of deep learning inference computing layers commonly used in CNNs through a C/C ++ interface with highly optimized building blocks.[15] The VPU plug-in uses a combination of highly parallel programmable computing with workload-specific hardware acceleration, and, by placing these components together on a general-purpose intelligent storage structure,[16] speeds up the operation of each layer in the neural network.

In Figure 5, Gwell Medical verified inference performance of object detection models running on TensorFlow and the OpenVINO toolkit on Intel® processors. TensorFlow was run only on Intel Core processors; the toolkit was run on an Intel Core processor, its processor graphics (iGPU), and an Intel Movidius VPU. Results indicated that OpenVINO provided improved inference performance compared with TensorFlow.

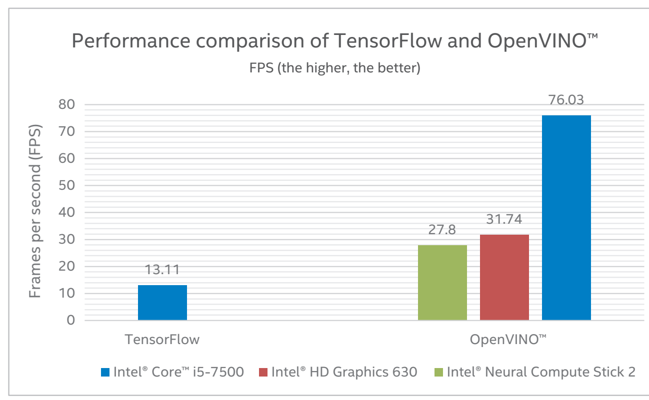

Figure 5. Inference performance of TensorFlow and OpenVINO™. Data source: Gwell Medical, March 2020.[17]

Optimization method 1: OpenVINO Asynchronous mode

The Intel Distribution of OpenVINO toolkit provides two modes of inference: Synchronous (SYNC) and Asynchronous (ASYNC) mode. For SYNC mode, workflow is processed with latency as priority: the inference time of each input is shorter than the time interval between each input or the preprocessing time for each of them. Therefore, a separate inference request is set up for each input. ASYNC mode tries to optimize the whole inference process and reduces data transfer costs and is more suited for throughput-oriented workflows. In ASYNC mode, several inference requests are set up simultaneously, and each request is on a child thread, with no data dependency with each other. The inference requests of an input batch are all computed at the same time. Once all inference requests of this batch are completed, results will be ready and recorded. Then the next batch will be pushed to the pool for a new round of inference. As there are usually multiple cores or compute units on CPUs and VPUs, an application suggests using ASYNC API to set up multiple inference requests and run them on different cores in parallel.

Optimization method 2: OpenVINO model-loading optimization

In addition to inference acceleration, model-loading time optimization is also crucial. The user application may access multiple types of computing hardware. For example, the CPU is utilized for system maintenance, data postprocessing, and I/O operations, while the iGPU or VPU is accessed for deep learning inference. Model-loading processes can consume many CPU resources and cause delays. When several DL models are used simultaneously, memory usage and overhead time to load models will increase.

The OpenVINO toolkit's MO converts a trained model into two files: the XML file with the deep learning network topology and the binary BIN file with network weights and biases. The IE then loads the files into the system memory, which compiles topology dynamically and creates an intermediate program of CNN, including the bottom layers for later execution.[18] The compilation usually happens on the host and creates some temporary data during the process. After the programming references to these temporary objects are destructed, these resources will be recycled by destructors.

When using an Intel Core processor iGPU to inference, the Gwell Medical application encountered a memory bottleneck when loading model into memory. The Intel team recommends a solution using "cl_cache" for offline compilation of deep learning graphs/topologies and saving the result to a local binary file. cl_cache is a binary representation mechanism to cache the OpenCL kernels in text form (.clcache is the file extension).[1] It can be applied by setting environmental variables to specify the storage path for "cl_cache" files. On Linux, this is the path: cl_cache_dir=/home/user/Documents/mrobot/cl_cache_dir/.

When the IE loads the model for the first time, the graphics driver will use the path for the binary files compiled by OpenCL kernels. When invoked for the second time, the IE checks the binary file at the path above. If found, the driver extracts the file and reduces the topology compilation overhead on the host. If not, the IE creates the file again.

On Linux, the reduction in runtime system resources by cl_cache can be profiled by the "top" command[20] or the VTune™ Performance Analyzer. In Table 1, if cl_cache is disabled when loading the model in runtime, resident memory (RES) and shared memory (SHR) usage is 201 MB and 115.3 MB, CPU usage is up to 100 percent, the total CPU time is 7.92 seconds, and the total program run duration (model load, data read, inference, and result output) 9.9 seconds. When cl_cache is enabled, RES and SHR are reduced to 188 MB and 62.5 MB, CPU usage to 15.3 percent, and the total runtime to 1.42 seconds.

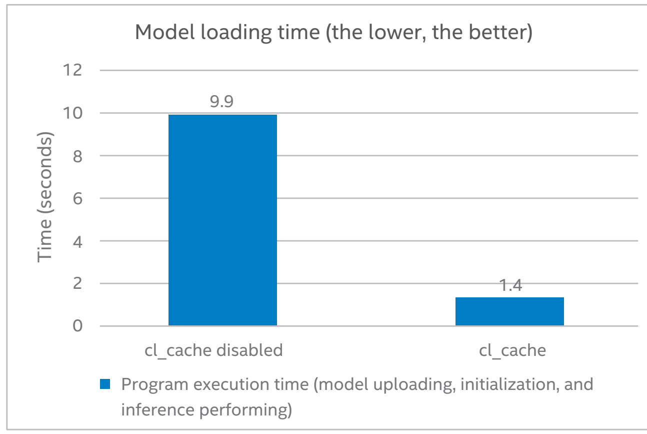

Figure 6. The time of model uploading before and after the use of cl_cache.

As the top command provides only instantaneous memory usage, the VTune Performance Analyzer can be used for a more comprehensive evaluation on program memory overhead. In Table 2, without cl_cache, total program memory allocation is 9 GB (5 GB of which is for model dynamic compilation by the CPU and iGPU [clDNN]) and only freed when the program ends. With cl_cache, only model copying is needed without runtime compiling, and memory allocation is lowered to 4 GB and with reduced elapsed time from 97.05 s to 7.97 s.

cl_cache when loading mode	"top" command output			
	Resident memory (RES)	Shared memory (SHR)	% CPU	TIME+
Disabled	201 MB	115 MB	100%	7.92 s
Enabled	188 MB	62.5 MB	15.3%	1.42 s

Table 1. Runtime system resource consumption changes when using cl_cache.[21]

Data from VTune™ Performance Analyzer	Compiling mode	
	Compiling in the runtime	Using cl_cache
Elapsed time	97.05 s	7.97 s
Allocation size	9 GB	4 GB
Deallocation size	5 GB	451 MB
Allocations	7.5 GB	6.7 GB
Total thread count	15	25

Table 2. VTune™ data regarding the memory usage of the same model under different compiling modes.

In addition to this cl_cache optimization technique for the iGPU, the OpenVINO toolkit provides ExportNetwork as one of IE's APIs for Intel VPU inference accelerating. It can generate a compiled binary (blob) and store the model topology as cache for myriadPlugin. When needed, the models can be directly loaded through the ImportNetwork function to reduce compilation overhead.

With Intel's hardware accelerators and Intel® software toolkits, Gwell Medical implemented application functions, improved AI inference performance, and consolidated the robotic computing resources and workflows.

Result: Efficient autonomous hospital logistics

With the usual requirements, hospital logistics workers can be at risk of exposure to infectious diseases and radiology radiation. With Intel technologies utilizing deep learning, the Noah robotic solution uses positioning and tracking to successfully perceive its environment within acceptable power and form factor limitations. This enables a number of hospital logistical improvements:

- **Workflow efficiency:** Walking distances can be decreased and medical supplies can be delivered faster, reducing the need for nurses' aides to be on call 24/7 at surgery centers or wards.
- **Medical supplies tracking:** Enable whole process monitoring and backtracking with identity authentication.
- **Analysis and alert:** Get data visualization dashboards and reports, with consumption statistics and low-stock alerts.
- **Logistics IT application:** Integrate with a supply-chain warehouse ordering system and use with intelligent ATM counter or kiosk deployment.
- **Logistics Security:** Comes with smart lock, identity authentication, bar code scanning, end-to-end monitoring, and an anomaly alert.

Noah robots are currently deployed in hospitals in China. In Guangzhou Women and Children's Medical Center, the robots are used in three delivery scenarios: surgical instrument sets for the surgical center, infusion bags for PIVAS, and gathering specimens for the laboratory. Data shows that two robots in the surgery center in one year can reduce the nurses' walking distance up to 1,944 km (more than the road distance between Guangzhou and Beijing).[22]

In February 2020, the Noah robots were able to take the place of medical staff and perform deliveries in areas of the hospital with high risk of exposure to the virus that causes COVID-19. The robots undertook pharmaceuticals delivery work with minimal human intervention in the process, reducing person-to-person contact and infection risks. The hospitals in China that deployed the Noah robots include Shanghai Ruijin Hospital, Wuhan Asia General Hospital, Wuhan Tongji Hospital, Central Hospital of Wuhan City, and First Affiliated Hospital of Harbin Medical University. The robots also helped speed up COVID-19 testing in some of these hospitals' ICUs. The robots enabled 24/7 distribution and alleviated the issues of heavy workloads and unpredictable service times due to limited medical personnel.[23]

Figure 7. Application scene samples seen by Noah robots in a hospital environment. Image courtesy of Gwell Medical.

Conclusion: The next step for hospital logistics robots

In June 2019, Intel, together with commercial robot partners in China, released *Robot 4.0: Edge-enhanced Cloud Robotics System and Infrastructure.*[24] This white paper explores the possibility of tens or even hundreds of robots working collaboratively in the robot 4.0 era with the support of a cloud-to-edge-device integrated system architecture. With more powerful intelligence for each robot, there will be a qualitative leap in context understanding, autonomous movement, continuous learning, knowledge construction, human-robot interaction, and more. These improvements are expected to further improve the efficiency, reliability, and usability of hospital logistics robots.

Key technology 1: 5G communications

Thanks to 5G's low latency of control signaling and data transmission, hospital logistics distribution robot fleets of up to 50 robots can safely and simultaneously work in a hospital. In April 2019, Gwell Medical and their partners successfully enabled a 5G function on the Noah robots to make it one of the first logistics robotic solutions confirmed to have all key technical indicators of China's design standards.

Key technology 2: Collaborative perception and localization

Due to the limitations of sensors and field of view (FOV), a robot may misjudge or lose track of its own position. With a joint decision-making strategy, the collaboration between robots can significantly improve their positioning and system reliability. In addition, external sensors such as surveillance cameras can be used to enhance robotic FOV and enable the whole system to maneuver better. For example, when an obstacle like a group of people standing in the path is detected, the system can guide the robots to new paths in real time to avoid congestion.

Key technology 3: Constant learning and adaptation

If the robots can adapt to scene changes (like room equipment rearrangements) and task assignment changes (such as a new object category for identification) without human interventions, system maintenance and upgrade costs can be greatly reduced. To work toward this goal, the Lifelong Robotic Vision research project from Intel Labs China and academic partners[25] has emphasized improving adaptability to scene changes[26] and the incremental learning ability of perception systems for new tasks.[27] The project provides open data sets and joint research to accelerate the evolution of intelligent autonomous robotic design.

Key technology 4: Extensible computing platform

Due to workload complexity and variety, heterogeneous hardware is a better design option for logistics robots. For example, state estimation and path planning are better handled by a powerful general-purpose CPU. Vision-related tasks like feature extraction and object identification are accelerated by VPUs, iGPUs, and FPGAs that have multiple small parallel cores. Performance requirements for different configurations of the same product may also vary. For robotic products with the best performance and cost balance, Intel offers a range of software and hardware solutions that support extensible computing performance. The solutions include Heterogeneous Extensible Robot Open (HERO) Platform for the flexible configuration of processors and accelerators[28,29] and the Intel Distribution of OpenVINO toolkit for DL inference.

With high-performance hardware, software toolkits, and DL technologies, Intel will continue to work closely with Gwell Medical and other partners to develop logistics robotic solutions in the future.

About Gwell Medical

With a focus on hospital logistics and intelligent medical treatment, Shanghai Gwell Medical (formerly known as mRobot) has developed AI technologies for its core products for indoor positioning and navigation, robot kinematics and dynamics, speech interaction, and computer vision. The Noah hospital logistics robot, smart nursing screen, and other IoT products from Gwell Medical are listed in the product catalog for epidemic prevention and control by the General Group of Standardization Administration of China (SAC). Gwell Medical now owns more than 350 patents and is a member of the drafting committee of the AI Standardization General Working Group of SAC.[30]

About Intel

You may know us for our processors. But we do so much more. Intel invents at the boundaries of technology to make amazing experiences possible for business and society, and for every person on Earth. Harnessing the capacity of cloud computing, the ubiquity of the Internet of Things, the latest advances in memory and programmable solutions, a portfolio of artificial intelligence technologies, and the promise of always-on connectivity, Intel is disrupting medical treatment and life science industries and helping them address challenges.

References

1. Tian, Zhi, et al. "Fcos: Fully convolutional one-stage object detection." *Proceedings of the IEEE International Conference on Computer Vision*. 2019.
2. Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
3. Feng, Weitao, et al. "Multi-object tracking with multiple cues and switcher-aware classification." *arXiv preprint arXiv:1901.06129* (2019).
4. Ma, Wei-Chiu, et al. "Exploiting Sparse Semantic HD Maps for Self-Driving Vehicle Localization." *arXiv preprint arXiv:1908.03274* (2019).
5. Chen, Xieyuanli, et al. "SuMa++: Efficient LiDAR-based semantic SLAM." *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.
6. Guo, Hongliang, et al. "Safe Path Planning with Gaussian Process Regulated Risk Map." *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.
7. Sebastian, Bijo, Hailin Ren, and Pinhas Ben-Tzvi. "Neural Network Based Heterogeneous Sensor Fusion for Robot Motion Planning." *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.
8. Beul, Marius, and Sven Behnke. "Fast Time-optimal Avoidance of Moving Obstacles for High-Speed MAV Flight." *arXiv preprint arXiv:1908.02028* (2019).
9. "The Application of AGC in China's Hospitals" [Online]. Available: https://mp.weixin.qq.com/s/sFMJI5lbxmQYrZ0T3feMqw [Accessed: 20-11-2019].
10. "China Medical Robot Industry Map: nearly" [Online]. Available: https://www.iyiou.com/p/44490.html [Accessed: 03-05-2017].
11. https://arxiv.org/pdf/1506.01497.pdf
12. https://jhui.github.io/2017/03/15/Fast-R-CNN-and-Faster-R-CNN/
13. "Model Optimizer Developer Guide - OpenVINO toolkit." [Online]. Available: https://docs.openvinoToolkit.org/latest/_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html. [Accessed: 11-Feb-2019].
14. https://github.com/intel/mkl-dnn
15. https://github.com/intel/clDNN
16. https://www.movidius.com/
17. FPS: Frames per second, the size of each frame is (384 x 384). The higher the number is, the better the performance it delivers. The test system configuration: Intel® Core® i5 processor 7500, Ubuntu 16.04 operating system, Intel® OpenVINO™ Toolkit development version (2020.1), TensorFlow (1.14).
18. https://intel.github.io/clDNN/index.html
19. "What is cl_cache?" [Online]. Available: https://github.com/intel/compute-runtime/blob/master/documentation/FAQ.md [11-Feb-2020].
20. https://en.wikipedia.org/wiki/Top_(software)
21. Gwell Medical test configuration: Intel Core i7-8700K CPU, model acceleration using integrated graphics Intel® UHD Graphics 630, Ubuntu 16.04 operating system , memory: 32 GB, OpenVINO™ Toolkit development version 2019R3.
22. http://mini.eastday.com/bdmip/190428162914016.html
23. http://ihealth.dxy.cn/article/679761
24. https://blog.csdn.net/gotouchtech/article/details/99089648
25. https://lifelong-robotic-vision.github.io/
26. Shi, Xuesong, et al. "Are We Ready for Service Robots? The OpenLORIS-Scene Datasets for Lifelong SLAM." *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
27. She, Qi, et al. "OpenLORIS-Object: A Robotic Vision Dataset and Benchmark for Lifelong Deep Learning." *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
28. http://www.robotplaces.com/info/3704.jhtml
29. Shi, Xuesong, et al. "HERO: Accelerating Autonomous Robotic Tasks with FPGA." *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.
30. https://mp.weixin.qq.com/s/yzEDXDI4Y2SGNMAqBVquKg