Artificial Intelligence, Machine Learning and Robotics to Support International Nuclear Safeguards In-Field Inspection Activities

Heidi Smartt, Sydney Dorawa, Shiloh Elliott, Philip Honnold, Zahi Kakish, Adithya Ramakrishnan, Tania Rivas, Nathan Shoman, Kyle A. Williams
Sandia National Laboratories, Albuquerque, NM, United States

ABSTRACT

Sandia National Laboratories (SNL) is in the process of creating Inspecta (International Nuclear Safeguards Personal Examination and Containment Tracking Assistant), an Artificial Intelligence (AI)-powered smart digital assistant (SDA) with robotic capabilities, aimed at enhancing the effectiveness, efficiency, and safety of international nuclear safeguards inspections. This innovative tool is designed to assist inspectors on-site by supporting or automating tasks that are typically mundane, hazardous, or susceptible to errors. In 2021, the development team established the specifications for Inspecta by analyzing International Atomic Energy Agency (IAEA) documents and consulting with former IAEA inspectors and subject matter experts. This process involved aligning in-field inspection tasks with existing commercial or open-source technologies to outline a roadmap for the initial prototype of Inspecta, while also identifying areas needing further research and development. From 2022 – 2024, the focus has shifted to integrating a critical inspection activity, the examination of seals, into an early version of Inspecta. This has involved developing both the software and hardware capabilities necessary for this task. This document outlines the ongoing advancements in Inspecta's functionalities, specifically those supporting the seal examination process.

INTRODUCTION

The IAEA Department of Safeguards is responsible for verifying international nuclear safeguards agreements. The mission of international safeguards is "to deter the spread of nuclear weapons by the early detection of the misuse of nuclear material or technology. This provides credible assurances that States are honouring their legal obligations that nuclear material is being used only for peaceful purposes" [1]. The implementation of international safeguards is unique for different states, as they are based on sovereign agreements between a State and the IAEA, as well as from facility-to-facility as determined through a safeguards agreement's facility attachment. Safeguards activities at a nuclear facility are also based on state factors and the IAEA's technical objectives as defined in the Annual Implementation Plan.

Despite the variability in their application, inspectors consistently perform a range of common tasks, including auditing facility records, inspecting and maintaining safeguards equipment, taking measurements and samples, examining and verifying seals, counting items, reviewing surveillance

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2024-07923C

footage, confirming design information, and monitoring for any discrepancies. These tasks can be both mentally and physically demanding, making them prone to human error; further, some tasks involve activities in hazardous environments. Moreover, the workload of international safeguards inspectors is on the rise due to several factors: the increasing number and variety of nuclear facilities under safeguards, the growing global inventory of special nuclear materials due to the longevity of safeguards for waste products and spent fuel, and a shift in the inspectors' role from auditing to more investigative duties. Despite these growing responsibilities, inspectors have limited time at facilities and must perform their duties as efficiently, effectively, and safely as possible. Efforts to enforce obligations associated with the Additional Protocol (AP) further compounds their workload.

AI and Machine Learning (ML) are becoming increasingly integrated into daily life, as seen in technologies like automated driver-assistance systems, personalized online shopping recommendations, voice-controlled smart home devices, and AI-powered vacuum cleaners, as well as SDAs like Amazon's Alexa[1]. Applying these advanced technologies to the field of international nuclear safeguards could significantly enhance the efficiency, effectiveness, and safety of inspection activities, particularly those that are repetitive, hazardous, and error-prone.

To address the evolving needs of safeguards inspectors, we are developing a prototype AI-enabled SDA with robotic support named Inspecta. Inspecta is designed to assist with general tasks, such as note taking, and more specialized tasks, such as reading seal numbers through optical character recognition (OCR). It will be housed in a compact, portable, and wearable device (with the current version residing on a Google Pixel 6 smart phone), primarily interacting with inspectors through voice commands and, when necessary, displaying information visually. We have assumed that there will be no communication to the Internet or cellular network from within the facility, and thus all software and algorithms run on-device. It is important to note that Inspecta is intended to complement human inspectors, not replace them.

In this paper, we will share details on Inspecta technical capabilities ("skills") that primarily assist inspectors in a seal examination task, selected to demonstrate progress towards a relevant high-impact task. Skills required for the seal examination task include speech recognition, speech synthesis, wake word, OCR (of both seal/container numbers and documents), information recall, robotic autonomy, object detection, segmentation, pose estimation, and mapping. We note that the seal examination task was selected early in the project after a literature review of IAEA inspection tasks and IAEA publications identifying challenges [2, 3, 4, 5], and from interviews with former IAEA inspectors and subject matter experts.

TASK DESCRIPTION AND INSPECTA ASSISTANCE

The seal examination task is important but tedious for IAEA inspectors. In one application, tamper-indicating metal CAP seals (Figure 1) with a numeric identifier are attached to containers after the

[1] https://developer.amazon.com/en-US/alexa

contents have been verified. An inspector will be escorted by facility personnel to the material holding location, find seals to examine, compare the seal number on the item and its container with the seal number and container number on a paper list, and mark that the seal has been examined and confirmed. The inspector also physically inspects the seal and seal wire for signs of tampering and pulls on the wire and seal to ensure proper connection to the container. A small set of seals may be selected for removal and verification at the IAEA headquarters; this selection process is performed using a statistical algorithm that informs how many and which seals should be removed and replaced.

Figure 1: CAPS metal seal, image: IAEA, 2020.

Examining seals with the Inspecta app on the Google smart phone includes the following steps and skills (in parentheses):

- Ingest seal, container list and documents prior to inspection (OCR)
- Inspector escorted to seal/s location
- Inspector opens Inspecta app and asks to start seal examination (speech recognition and synthesis) or manually selects the task on the smart phone screen
- Inspecta uses OCR to capture seal and container numbers and they are automatically marked as examined in Inspecta app (task tracking)
- Inspector can ask Inspecta for information, i.e., “Inspecta, what is a significant quantity of nuclear material?” (information recall)
- Inspector can add a label to the Spot-generated map if desired (to pinpoint seal location, for example) (mapping)

Spot can also separately perform seal examination and steps and skills required are:

- Inspector asks Spot to perform seal examination (speech recognition, autonomy)
- Spot locates a container (object detection)
- Spot traverses room to container and locates a seal on the container (autonomy, object detection)
- Spot estimates the pose required for the manipulator arm camera to best align with seal number and acquires images of the seal (segmentation, pose estimation)
- Spot sends seal numbers examined to Inspecta (Spot/Inspecta interface)

INSPECTA APPLICATION

Before developing the Inspecta architecture, key considerations such as security, privacy, usability, and the adaptability of existing algorithms, libraries, and software components were prioritized. The selection of hardware with necessary input/output capabilities and sensors was also crucial and led to the selection on the Google Pixel 6 phone. Initial considerations included using existing platforms like Alexa, Siri, or the open-source Mycroft[2] as a foundation. However, due to the proprietary nature and cloud-based operation of Alexa and Siri, along with the lack of reliable wireless connections in nuclear facilities, these options were deemed unsuitable. Mycroft's limited documentation and available skills further led to the decision to develop the SDA platform from scratch.

A significant aspect of Inspecta's development is the emphasis in on-device learning due to potential lack of communications available within a nuclear facility. The Open Neural Network Exchange (ONNX) API is utilized for developing models compatible with Inspecta's code base, allowing models built in PyTorch and TensorFlow to be converted to ONNX format for deployment across various platforms. Moreover, model quantization is employed to enhance performance on embedded and mobile devices by reducing the precision of the machine learning algorithms' mathematical operations.

Code for the Inspecta app was originally developed with Xamarin to allow multiple end-user platforms (iOS, Android, Windows, etc.) since the team was uncertain about which platform might ultimately be deployed. Xamarin is no longer supported, and all code has been migrated to .NET MAUI; however, the capability to deploy to multiple platforms remains. The Inspecta app has been built using a commercial UI kit such that project resources could be spent on skill development rather than app development. Once modules are developed offline, they are integrated into the phone UI. Figure 2 shows example screen shots of the note taking and seal examination modules.

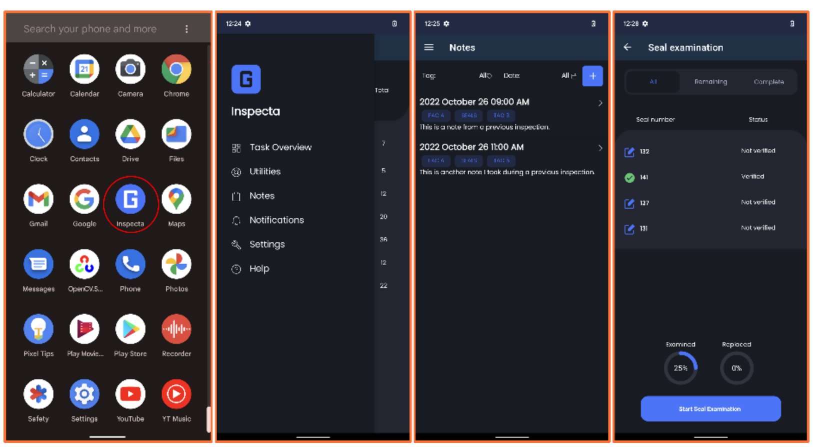

Figure 2: Inspecta app, built in a UI kit. (Left) Google Pixel 6 screen with Inspecta app in center right, (Left Center) Inspecta's main screen, (Right Center) Note taking interface, (Right) Seal examination module.

[2] https://mycroft.ai/about-mycroft/

The main capabilities to-date within the Inspecta app include speech synthesis (also called text-to-speech), speech recognition (also called speech-to-text), wake word, seal examination, note taking, document OCR, information recall and maps. Speech synthesis uses native Android services, while the other modules have required development and are described next.

Speech recognition utilizes Meta's Wav2Vec2 [6] algorithm, where the waveform is initially captured through an on-device microphone. This waveform is then converted and preprocessed before being fed into Wav2Vec2, which outputs the estimated text. The predicted text is subsequently compared against a set of possible commands using the Levenshtein distance to evaluate similarity. If the distance is too high for any of the given commands, indicating a significant discrepancy, it is inferred that a question has been posed, triggering the information recall module.

The wake word capability ("Hey Inspecta" – triggering the system to begin listening) has gone through many iterations to improve robustness. The current approach uses a custom classification model based on openWakeWord[3] that provides a Softmax score for the presence of "Hey Inspecta" in an audio clip. The model was trained on 100k+ synthetic samples mixed with noise and background sounds.

When an inspector says "Hey Inspecta, start seal examination," the Inspecta app opens the phone's video camera and begins a two-stage OCR process (text detection and recognition based on CRAFT [7] and a CRNN [8]) to eventually acquire the seal and container numbers (Figure 3) and correlate them with a list of seals/containers previously ingested into the app. The app keeps track of which seals and containers have been examined (Figure 2, Right).

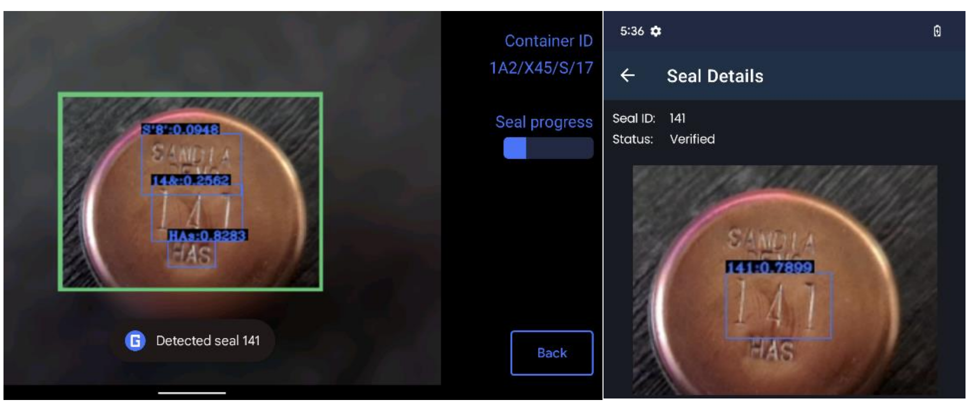

Figure 3: OCR process for acquiring seal and container numbers.

Document scanning, using OCR, is a process that captures an image of a document and transcribes any text it contains. This function is particularly valuable for tasks such as scanning lists of seals and corresponding containers for later seal examination or capturing documents for the information recall

[3] https://github.com/dscripka/openWakeWord

module. However, document scanning introduces challenges not encountered in the seal examination task, including the need for higher resolution images to discern numerous words on a page and the issue of dealing with warped or skewed pages, which can affect performance and text ordering. Unlike seal examination, ensuring the correct sequence of words from left to right and top to bottom is crucial for document scanning. Document scanning is not a real-time task; it requires capturing a high-resolution image for accuracy, with the user toggling settings to determine if bounding box ordering should be performed. Document scanning uses a more computationally intensive algorithm, DeepSolo [9].

With information recall, an inspector can ingest needed documents into Inspecta prior to an inspection, ask natural language questions, and receive spoken information back. As an example, we ingested the 2022 Safeguards Glossary into Inspecta and then asked "Hey Inspecta, what is the definition of a significant quantity?" Inspecta then verbally responds, "the approximate amount of nuclear material for which the possibility of manufacturing a nuclear explosive device cannot be excluded." The challenge with information recall for international nuclear safeguards is the lack of nuclear influenced datasets. We have collected the most prevalent nuclear related keywords using 1,000 Office of Scientific and Technical Information (OSTI) abstracts related to nuclear safeguards to replace words at random for both the training and testing of Stanford Question Answer Dataset (SQuAD) datasets [10]. This method was inspired by the "Salt and Pepper" technique [11] used by AJAX (Artificial Judgement Assistance from teXt) to generate a nuclear domain specific corpus. Currently, SentencePiece, a text tokenizer, is being used to create new vocabulary files to best match the new nuclear training corpus.

Note taking is straight forward and is shown in Figure 2. We won't detail much of maps in this section, as maps are acquired via Spot robot – however, the concept is that an image of the Spot-acquired 2D map is transmitted to the Inspecta app, where an inspector can open and view the map (currently) and eventually drop labels at locations of interest.

ROBOTIC AUTONOMY

Boston Dynamics' Spot robot was acquired in September 2022 to provide a physical implementation of Inspecta, especially for tasks that are tedious or hazardous. The benefits of incorporating robotic support are (1) reduced radiation exposure to inspectors, (2) ability to image difficult-to-access locations via the camera within Spot's manipulator arm, and (3) ability to perform tedious and repetitive tasks. While the commercially available Spot robot comes with a significant level of capability, additional skill development is required for our application, namely, autonomy, object detection, segmentation, pose estimation for seal OCR and gripping, and map acquisition.

The Spot robot can natively be controlled using (1) a joystick-like tablet with an operator selecting functions like walking, crawling, sitting, and object manipulation or (2) "Autowalk" which enables the operator to record a manually-controlled mission and replay it. Neither of these are ideal for safeguards inspections as we don't wish to add tasks for inspectors (operating a robot) and Autowalk requires a static environment (robot performs the same activity for items in the same place repeatedly). Therefore, we are developing a robotic autonomy stack such that an inspector can command Spot to perform a task, such as seal examination, with Spot able to autonomously perform the task. High level tasks, like seal examination, are broken down into sub-behaviors and sub-trees for additional modularity and reactivity, e.g., having Spot undock from its power source, stand, look around a room, traverse the room, perform an action like taking a picture and return to the dock. An informed search behavior using object detection and LiDAR will allow us to explore an unknown area and locate a container after being escorted by an inspector to a starting location. The modular nature of the behavior trees allows for greater *reactivity* to changes occurring around the robot. For example, if a container is moved while Spot is attempting to navigate towards it, this will result in Spot replanning to move towards the container using a new trajectory.

More specifically, the team has created multiple action primitives for the Spot robot using the Spot software development kit (SDK), with plans to expand this suite throughout the project. These primitives are being tested in a simulated inspection environment within the AutonomyNM Testbed[4], using 55-gallon drums with metal cup seals as targets for Spot to identify and process using OCR technology. Additionally, capabilities are being developed for Spot to follow inspectors wearing fiducial markers, with ongoing research into how to securely integrate these markers into clothing to prevent unauthorized control. The introduction of behavior trees has been well-received, leading to the open-sourcing of the "spot_bt[5]" package, fostering transparency and community involvement. Furthermore, in collaboration with Boston Dynamics' efforts to enhance Spot's operability through ROS 2 [12], a widely used robotics middleware, the team has migrated our work to ROS 2 under the name "spot_bt_ros[6]." Both packages are open-source and available for any user to download and develop around their autonomy needs.

OBJECT DETECTION

Object detection for this project employs the YOLOv5 algorithm from ultralytics[7], running on a Jetson graphics processing unit (GPU) mounted on Spot. Initially, the model was pretrained using the COCO[8] dataset, which comprises a wide range of everyday objects, to establish a broad visual understanding. To enhance the system's utility for our needs, Limbo[9] synthetic data is then utilized for transfer learning. The Limbo dataset consists of over 1 million synthetic labeled images of various containers, including UF_6 cylinders (30B, 48X, 48Y, 48G types), wine barrels, and 55-gallon drums, among other objects. Finally, to improve the model's performance further, additional datasets specifically focused on 55-gallon drums and metal cup seals are being compiled and labeled in our lab. This final dataset is continually trained and tested on to ensure reliability of the object detection

[4] https://autonomy.sandia.gov/autonomynm
[5] https://github.com/sandialabs/spot_bt
[6] https://github.com/sandialabs/spot_bt_ros
[7] https://github.com/ultralytics/yolov5
[8] https://cocodataset.org/#home
[9] https://limbo-ml.readthedocs.io/

skill. This progression in training data from COCO, to Limbo, to in house is shown in Figure 4. Results from the drum and seal object detection system on test data are shown in Figure 5.

Figure 4: Progression of object detection training. (Left) COCO dataset for common objects, (Center) Limbo dataset for UF_6 and other cylinders, (Right) 55-gallon drums and metal cup seals at the AutonomyNM testbed.

Figure 5: Test results for object detection of 55-gallon drums and metal cup seals, with confidence level displayed.

Our team has leveraged several annotation tools while compiling object detection datasets. LabelMe[10] lets the user draw bounding boxes on individual photos. This approach is simple but time consuming. LabelStudio[11] lets the user draw bounding boxes for key frames in a video and interpolate between them, which is much more time efficient than labeling individual photos. However, LabelStudio does not output video data in a YOLOv5 compatible format, so we had to write a conversion script. Lastly, RoboFlow[12] lets the user automatically annotate images with a foundation model such as Grounding

[10] https://pypi.org/project/labelme/
[11] https://labelstud.io/
[12] https://blog.roboflow.com/grounding-dino-zero-shot-object-detection/

DINO[13]. Grounding DINO is a vision-language model that can perform open set object detection of arbitrary text queries zero shot, i.e., without any query specific training. While traditional object detection systems are trained to identify a specific set of objects, Grounding DINO attempts to identify any object associated with a text query. Grounding DINO generated training data is shown in Figure 6. Note this model is too slow to run in real time, too computationally expensive to run on an edge device, and fails to draw good bounding boxes in many circumstances. Despite these flaws it can powerfully accelerate dataset generation.

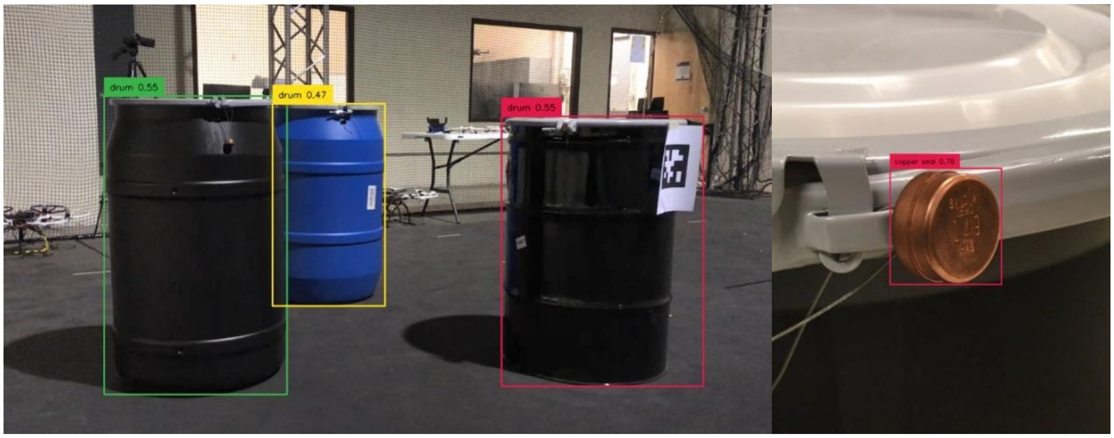

Figure 6. Grounding DINO auto labelled training data.

Finally, RoboFlow allows users to upload their own models and use them to auto label additional data. With this technique we can progressively bootstrap a more and more thoroughly trained object detection system.

OBJECT SEGMENTATION

Another visual skill generally useful in robotics is segmentation. Rather than drawing bounding boxes around objects, segmentation seeks to categorize individual pixels into groups or instances. This is useful when trying to visually understand parts of a scene that cannot easily be captured by a bounding box, such as which areas of the visual field are walkable, and which are hazardous. It is also useful for isolating relevant point cloud data from RGB-Depth cameras, which will be discussed next.

A common ML algorithm for segmentation is Mask-RCNN[14]. A more recent foundation model for segmentation is Meta's Segment Anything Model[15] (SAM). Some results from these models are shown in Figure 7.

[13] https://arxiv.org/abs/2303.05499
[14] https://arxiv.org/abs/1703.06870
[15] https://arxiv.org/abs/2304.02643

Figure 7. Pretrained segmentation results. Mask-RCNN in the middle. Segment Anything Model on the right.

Pretrained Mask-RCNN successfully segments out the humans and a few odd objects in the scene but struggles to identify Spot as a single object. This is likely because robotic dogs were not in the Mask-RCNN training data. However, Mask-RCNN's performance could certainly be improved by collecting task specific training data and fine tuning. Meta's SAM does very well on this example with no fine tuning. Foundation models, by definition, have been trained on such vast data that they generalize well to new situations, which explains SAM's strong performance here.

Figure 8. SAM segmentation from bounding boxes for drums and seals.

SAM can also segment out an object given only a bounding box around it, such as the bounding boxes produced by YOLO. This is shown in Figure 8. While SAM performs incredibly well, it is too computationally intensive to run locally onboard Spot. However, this YOLO-SAM pipeline can be used to auto generate segmentation training data for smaller models such as Mask-RCNN. Generating segmentation datasets is labor intensive without this auto labeling pipeline, as object perimeters are harder to label than simple bounding boxes.

POSE ESTIMATION

As can be seen in Figure 4, Figure 5, and Figure 6, metal cup seals on containers are not always at convenient angles for reading the inscription on their surface. Spot needs to be able to estimate the orientation of the seal and position its manipulator arm perpendicular to the face of the seal. This will enable clear photography of the inscription as well as grasping. Grasping may be implemented in future work to ensure metal cup seals are properly attached to their containers. The team is currently working on “pose estimation” for Spot to perform seal examination most effectively.

Pose estimation falls into the category of geometric or 3D computer vision, which is typically performed with depth sensor data in addition to traditional red-green-blue (RGB) images. RGB plus Depth (RGBD) sensors based on projected infrared stereo vision are common in this domain. Spot has five such sensors built into its body, and the team is considering mounting an additional RGBD sensor to its manipulator arm.

An example of a depth channel image is shown in Figure 9 on the left. This depth scan contains a 55-gallon test drum with a stuffed animal on top in the foreground in green, a human engineer in the middle-ground in blue, and patches of floor and ceiling in the background in red. It is difficult to do anything useful on the full point cloud. Conveniently, the segmentation techniques presented in the previous section can isolate the point clouds of specific objects. Segmented out sections of the point cloud can be useful for estimating the exact spatial location of drums and nearby humans, or for estimating the orientation of seals. Once the seal point cloud is isolated from the background, algorithms such as Iterative Closest Point (ICP) can be used to match the point cloud data to a known model for the seal, thus inferring its orientation. An initial implementation of this ICP[16] approach is shown on simulated sensor data in Figure 9 on the right. The blue ‘model’ point cloud is generated from a CAD file of the seal. The orange ‘scan’ point cloud is a simulated sensor scan. The green ‘scan aligned’ point cloud is the sensor scan after running ICP and solving for seal orientation.

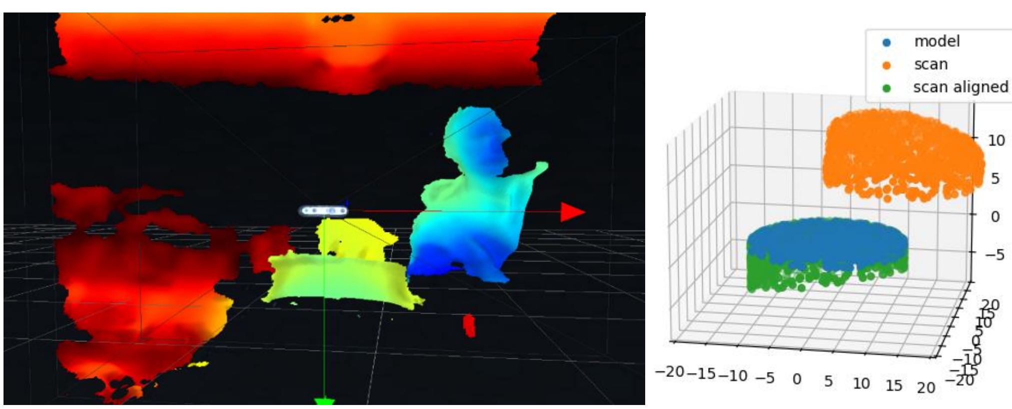

Figure 9. (Left) Depth channel data from an RGBD sensor. (Right) Initial implementation of ICP for seal pose estimation on simulated sensor data.

[16] https://pypi.org/project/simpleicp/

As an alternative to mounting an additional RGBD sensor, the team considered structure from motion algorithms using only the RGB camera built into Spot's manipulator arm. While theoretically possible, as humans estimate depth with our built in RGB sensors, it proved difficult to get good results with this method using existing open-source software such as OpenSfM[17]. Segmenting out individual object point clouds would also be more challenging with this method, as there is no one-to-one mapping from already segmented RGB space, to point cloud space.

Lastly, the team is considering modern deep learning alternatives to the scan, segment, ICP pipeline described above. PoseCNN[18] is a powerful and popular deep learning-based approach for six degree of freedom (position and orientation) object tracking based only on RGB data. It was published in 2017. However, this method only works on a closed set of objects used in training, and it is difficult to generate additional inspection specific data in six degrees of freedom. FoundationPose[19], published in 2024, is designed to work on objects it was not explicitly trained on, making it an open set 6D object tracker. We have not yet tested this approach for our applications.

MAPPING

Finally, the team is developing the ability for Spot to acquire point clouds using an attached Ouster OS1 LiDAR payload (Figure 10) for generation of 2D and 3D facility maps. Once Spot's LiDAR collects point cloud data, the data is processed using mapping algorithms on the Jetson GPU. An image of a 2D map is generated and sent to the Inspecta app. The team is progressing with the implementation of the Simultaneous Localization and Mapping (SLAM) algorithm to compile and produce 3D point cloud environments that can be used to either reconstruct 3D mesh environments (like virtual environments) or be used for design information verification with subsequent data collections. Additionally, the SLAM algorithm will provide enhanced navigation information for Spot to make informed decisions while attempting to locate a container in an unknown environment.

Figure 10: Spot with Ouster OS1 LiDAR payload.

[17] https://github.com/mapillary/OpenSfM
[18] https://arxiv.org/abs/1711.00199
[19] https://nvlabs.github.io/FoundationPose/

SUMMARY AND NEXT STEPS

An AI-enabled SDA with Spot robotic support can be integrated into the process of international nuclear safeguards inspections to assist with mentally and physically challenging tasks and those prone to human error to increase the effectiveness, efficiency, and safety of inspections. In this work, we have identified and down-selected a safeguards task, seal examination, that is tedious and prone to error to demonstrate Inspecta's capabilities and build skills that can be used for completing additional tasks in the future. We have developed various skills including speech synthesis/recognition, OCR, task tracking, information recall, wake word, document scanning using OCR, note taking, Spot autonomy, object detection, object segmentation, pose estimation, and mapping. Possible future work includes (1) improving reliability of previously developed skills like OCR and information recall, (2) advancing Spot's autonomy stack through further integration of the other skills (object detection, segmentation, pose estimation, mapping), and (3) adding a new task, spent fuel verification, that will utilize the existing skills and require development of new skills. The robotics team intends to extend our work to deep imitation learning [13] for generalizable control when attempting to tug on a seal given how they are rarely uniform in their installation on a container.

ACKNOWLEDGEMENTS

The authors would like to acknowledge and thank the U.S. National Nuclear Security Administration (NNSA) Office of Defense Nuclear Nonproliferation R&D Safeguards portfolio for funding and supporting this research.

REFERENCES

[1] International Atomic Energy Agency. https://www.iaea.org/topics/basics-of-iaea-safeguards. Accessed 3/17/2021

[2] IAEA Safeguards, "Emerging Technologies Workshop: Trends and Implications for Safeguards Workshop Report." February 2017, Available at: https://www.iaea.org/sites/default/files/18/09/emerging-technologies-130217.pdf

[3] IAEA Safeguards, "Emerging Technologies Workshop: Insights and Actionable Ideas for Key Safeguards Challenges Workshop Report." STR-397, January 2020. Available at: https://www.iaea.org/sites/default/files/20/06/emerging-tehnologies-workshop-290120.pdf

[4] IAEA Safeguards, "Research and Development Plan: Enhancing Capabilities for Nuclear Verification." STR-385, January 2018. Available at: https://www.iaea.org/sites/default/files/20/06/rd-plan-2018.pdf

[5] IAEA Safeguards, "Development and Implementation Support Programme for Nuclear Verification 2020-2021." STR-393, January 2020. Available at: https://www.iaea.org/sites/default/files/20/06/dis-plan-2020-2021.pdf

[6] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33, 12449-12460.

[7] Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character region awareness for text detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9365-9374).

[8] Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence, 39(11), 2298-2304.

[9] Ye, M., Zhang, J., Zhao, S., Liu, J., Liu, T., Du, B., & Tao, D. (2023). Deepsolo: Let transformer decoder with explicit points solo for text spotting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 19348-19357).

[10] P. Rajpurkar, R. Jia, P. Liang, Know what you don’t know: Unanswerable questions for squad, arXiv preprint arXiv:1806.03822 (2018).

[11] B. Wilson, K. Duskin, M. Subramanian, R. Goychayev, A. M. Zuniga, Artificial judgement assistance from text (ajax): Applying open domain question answering to nuclear non-proliferation 1190 analysis, ESARDA Bulletin (2021).

[12] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics vol. 7, May 2022.

[13] T. Zhao, et al., “Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware,” arXiv, 2023. Available at: https://arxiv.org/abs/2304.13705