Experimental Results from On-orbit Edge-deployed AI Detection of Resident Space Objects Using Computer Vision

Paul Day
Booz Allen Hamilton
Jeremiah Crane, **Paul Cronk, Ph.D.**, **Carlos Jimenez, Bruce Johnson, Ph.D.,**
Rochelle Koeberle, **Darin Millard, Zackary Werner**
Booz Allen Hamilton

ABSTRACT SUMMARY

Space domain awareness (SDA) is crucial for ensuring the safety and sustainability of space operations, especially as the space domain transitions toward a contested, degraded, and operationally-limited environment. The number of resident space objects (RSOs) continues to grow, and traditional ground-based sensors face limitations of coverage and latency. These sensors are also susceptible to deception due to their predictable periods of observation. Maneuvers performed immediately before or after the RSO is observed can introduce sufficient error into orbit determination to cause track association issues. In addition, a paucity of angular diversity in observations exists due to most current high-accuracy SDA assets being ground-based. Relatedly, as the lunar and Martian orbital regimes become increasingly crowded, SDA's importance in those regions will grow, presenting significant challenges for ground-based SDA.

To address these shortcomings, we present the results of an experimental demonstration of on-orbit RSO object detection using convolutional neural networks (CNNs) deployed to an on-orbit edge compute device. The CNN is trained on a custom annotation set created using the continuous false alarm rate (CFAR) detection algorithm and a proprietary algorithm (patent pending) which includes elastic shape analysis to identify RSOs in imagery and generate synthetic training image datasets. The goal of the experiment is the detection of RSOs using artificial intelligence (AI), specifically a computer vision (CV) CNN model running on edge-compute hardware in orbit without requiring ground-based image processing. The initial set of candidate RSOs are bright, high-inclination rocket bodies with stable ephimerides selected for ease of detectability in the experimental demonstration.

Once mature, our onboard CV solution will represent a significant advancement in SDA by producing a scalable algorithm capable of downlinking captured SDA data with over 93% reduced throughput requirements than attempting to downlink whole images and isolate that data on the ground. By developing this technology, we are paving the way for the proliferation of small, inexpensive, but highly capable SDA sensors throughout multiple orbital regimes, vastly increasing the angular diversity of observations available for RSOs orbiting Earth and other celestial bodies. In this effort, we aim to contribute to safer operations in humanity's orbital commons.

1. INTRODUCTION

In increasingly crowded orbital environments, space domain awareness (SDA) data is critical to safety of flight and, in manned missions, safety of life. The limited supply and high cost of SDA sensors mean that SDA sensor attention is a coveted commodity. In short, SDA sensor coverage needs to scale, and quickly.

In support of this objective, this team is attempting to demonstrate an inexpensive, highly scalable SDA sensor solution that can go wherever satellites fly. Thus, at scale, this solution would not be limited in the amount of sky it can cover or suffer the fierce competition that afflicts requests for ground-based resources based on mission prioritization. When SDA sensors are mounted directly to spacecraft, the global space enterprise quickly brings SDA coverage wherever it goes. In this vein, dual-purpose systems which are both star trackers and surveillance sensors have previously been proposed. [1, 2]

The prototype system discussed in this paper is designed to produce resident space object (RSO) detection data that are useful to SDA systems. The remainder of this paper consists of a brief exploration of background work in this area and an overview of this mission design and method, followed by a presentation of the current results and

analysis of the neural network (NN) performance. It concludes with recommendations for further experimentation, testing, and system design.

2. BACKGROUND

Awareness of RSOs involves a number of steps. The first is detection, or the determination that an RSO is in the field of view of a sensor at a given moment. This is followed by measurement, or the extraction of a relative position vector, range, or other parameter(s) from the detection. A series of measurements can be used to perform object tracking, either through orbit determination directly or the use of Kalman or particle filters. Alternatively, some track-before-detect methods determine the presence of a target based on a signal-to-noise ratio (SNR) that only exceeds a detection value after a period of time over which, very low SNR values are integrated in a method which assumes some consistent motion. Finally, the resulting track can be associated with a catalog of known space objects or the determination made that an unknown object has been observed. This study is concerned with the first step in the chain—detection—or determining the presence or absence of an RSO in a given sensor collection.

Two primary methods for RSO detection using optical sensors have previously been proposed [3]. The first method, with which this paper is concerned, is the streak method [4], in which a sufficiently long real or synthetic exposure is used to allow relative motion between the stars and the target to be observed. Depending on the location and orientation of the observer, a long exposure causes RSOs or stars (or both) to form streaks in the image as their relative motion becomes apparent. These streaks can then be detected using computer vision (CV) or neural networks (NN) [5, 6, 7]. RSOs can be distinguished from stars by their different relative motion. Similarly, a ground-based observer with a constant camera angle notices minimal movement from a geostationary RSO while stars streak past due to the earth's rotation [5]. The same observer notices low earth orbit (LEO) RSOs streaking but at different angles and rates from the stars [1]. A space-based observer with a constant inertial camera angle notices minimal star movement while RSOs streak past at different rates and angles depending on their orbits.

The second family of methods, generally known as track-before-detect methods, use very low SNR sensor observations taken over a series of images along with reasonable assumptions about a target's movement (e.g., Keplerian orbital mechanics) to arrive at a maximum likelihood estimation. These techniques show promise, and their results are particularly impressive for very low SNR targets, but are often computationally expensive [8, 9, 10, 11, 12].

The streak method and track-before-detect methods can be augmented by using a star catalog to remove stars before performing streak detection [4, p. 19]. Star identification can be performed using NN [13, 14].

This study focused on the streak detection method and, more specifically, on the streak detection method in which the observer is on-orbit and in a constant attitude in an inertial reference frame.

3. METHOD

The approach to RSO detection is outlined in Fig. 1. The process involves on-orbit collection of images, on-orbit combination of images to enhance the SNR of any RSOs present in the collected images, and finally on-orbit inference of images to detect streaks indicative of RSOs.

A command to the spacecraft, originating from the ground, sets the parameters for the detection attempt. The specified number of images is obtained at the specified settings (exposure time, gain), after which, if more than one image was taken, the images are summed to produce a synthetic exposure time exceeding the camera's design capability. The image is then divided into appropriately sized subsections, or tiles, for inference before being converted into the appropriate format and passed to the inference hardware (HW). The inference is performed using a pruned version of YOLOv4-tiny, and the results are analyzed to determine if any detections exceed the minimum threshold for confidence in a detection.

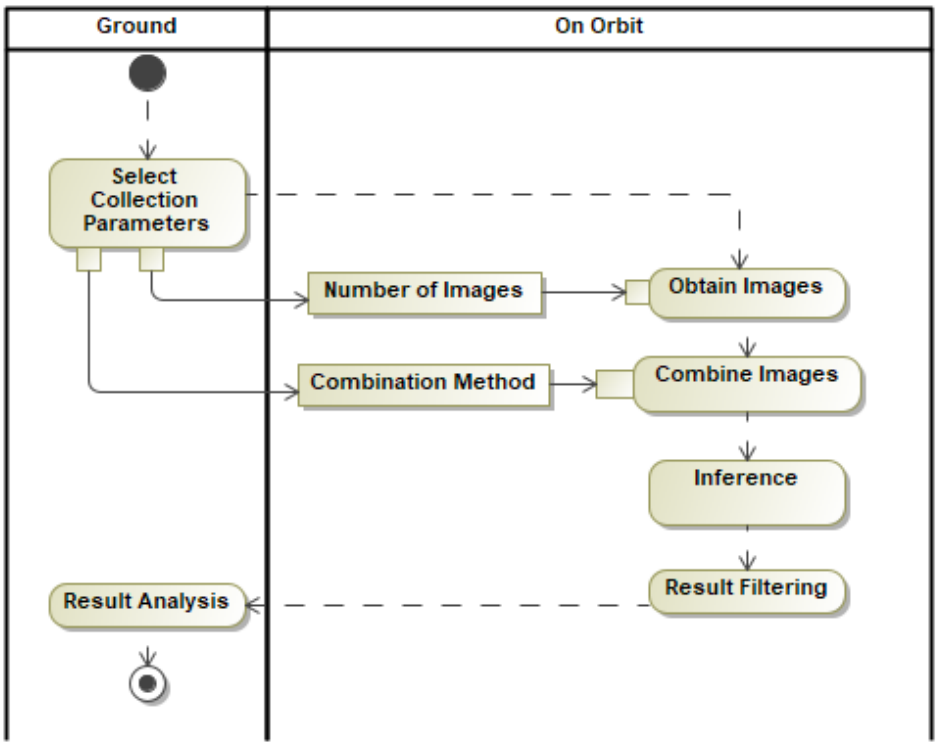

Fig. 1. On-Orbit RSO Detection Approach

The spacecraft used for this experiment was the Apex Aries SN1 (see Fig. 2). The imager used for this experiment, the Ubotica CogniSAT NEI AI-Enabled Non-Earth Imaging Space Camera System, is facing the viewer on the right.

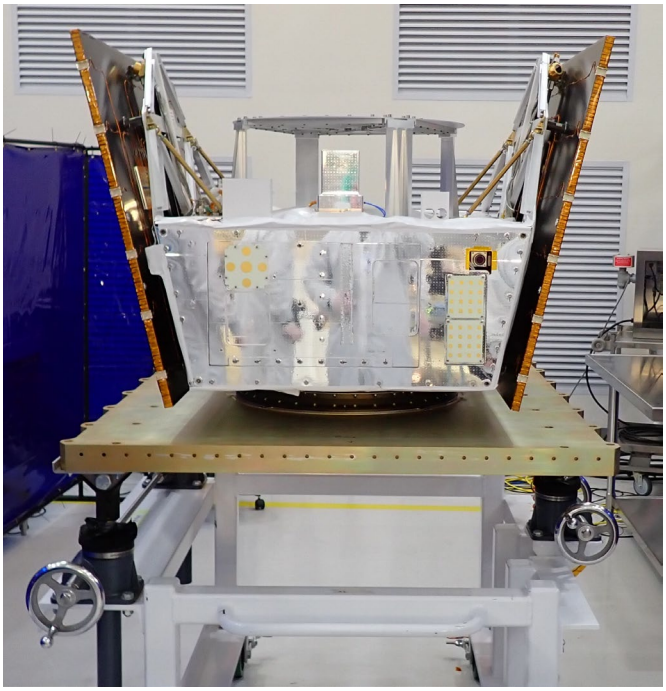

Fig. 2. The Apex Aries SN1 Spacecraft. Provided by Apex Space. Used with Permission.

3.1 Mission planning

To implement the RSO detection method in the on-orbit experiment, the team set out to aim the satellite's optics package at an orientation and time likely to yield a streak detectable by the experimental method. By identifying a set of likely bright objects (expended rocket bodies/debris with high-inclination orbits) with well-understood ephemerides, the team could plan to properly orient the host satellite's camera's field of view (FoV) during a predictable time interval such that targeted satellites would predictably cross the FoV, enabling execution of its RSO detection method at a time, place, and orientation likely to achieve positive results. Those results could be tested to confirm the expected bright objects were detected by the experimental method.

A simulation environment was used to model the experiment's host satellite, its pointing angles at key times, and the trajectories of potential observation targets. The simulation environment was also used to calculate optimal target observation time intervals. The simulation environment chosen was Ansys Systems Tool Kit (STK), which provides industry-standard rigor in physics-based modeling and sensor access calculation capabilities to generate the time intervals for mission planning. The time intervals were calculated as follows.

First, these components were built or imported into the Ansys STK scenario:

- A model of the host satellite, especially including a model of the on-board imaging system (as a sensor attached to the host satellite, including FoV and other characteristic parameters set based on design metrics)
- Host satellite ephemeris (based on most-current real-world ephemeris data)
- Two known bright rocket bodies with up-to-date ephemerides

Next, Ansys STK was used to plot the sun angle of the target debris. In this case, the sun angle was defined as the angle formed by two vectors: the first from the target to the sun and the second from the target to the host satellite, with the target debris as the vertex of this angle. On the same plot, but using a second vertical axis, the magnitude of the vector from the host satellite to the debris, or the slant range, was plotted. This plot was examined for periods when both the sun angle and the range were low, ideally below 40° and 800 km, respectively (see example in Fig. 3). These periods represent opportunities when the target debris would most likely be bright. Since the targets proposed for examination are long-abandoned debris, no modeling was done of their likely attitude or of any anisometric reflectivity, since they were assumed not to have solar arrays.

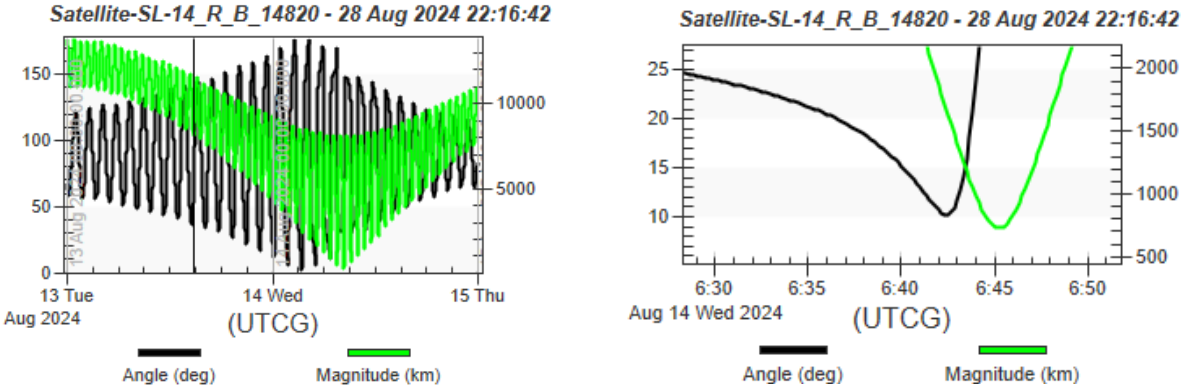

Fig. 3. An example of the sun angle and range plot of a selected rocket body relative to the host spacecraft for approximately 48 hours at left, with a specific 24-minute period of interest at right.

During these periods, the team selected observation opportunities when 1) the target was in sunlight; 2) the target could be held near the center of the FoV of the imager while maintaining the host spacecraft's star trackers the required distance off the sun for a conservative period before and after the observation; 3) the background of the resulting image would contain no sunlit portions of the earth, but only regions of earth in night or starfield; and 4) the relative motion between the two would result in a change of relative angle of 0.2° - 0.4° over the course of a 0.5s exposure. This latter objective is based on the 0.5s maximum exposure of the available imager (discussed in more detail below) and the observation that the NN performs well on synthetic streaks of 5 pixels or greater in length, which equates to 0.2° in the FoV. Opportunities that would exceed the 0.4° objective are rare, but the purpose is to avoid observations where the collected light is spread over too many pixels, resulting in a low SNR for the streak.

This could be repeated to generate access windows of opportunity for the other bright debris. Mission operators would then have the necessary quaternion-time pairs available to execute a mission using the proposed experimental RSO detection methodology and to verify that any detected bright objects were the objects intended and targeted for detection. This process was repeatable for objects of varying brightness, enabling further experimentation on iteratively dimmer objects and larger sun angles to determine the boundaries of the detection capability.

3.2 Image collection

Ideally, the collection approach should be agnostic (though some cameras will be better suited to this application than others) and the NN would sufficiently generalize the streaks to allow images from different cameras to be inferred with the same NN. For the work described in this paper, the Ubotica CogniSAT NEI AI-Enabled Non-Earth Imaging Space Camera System was used. This instance of the Ubotica camera has pixel dimensions of 1920x1200, a lens with an f-stop of 5.6, and a FoV of 99°. The output image has been de-bayered and is RGB. The imager is depicted in Fig. 4.

Fig. 4. The Ubotica CogniSAT NEI AI-Enabled Non-Earth Imaging Space Camera System. From Ubotica. Used with permission.

Depending on the exposure and brightness of the resulting images, multiple images can be combined to digitally extend the effective exposure duration. While this function may be used in future work, the images in this paper were obtained with a single 0.5s exposure. A portion of an image collected with this camera is shown in Fig. 5.

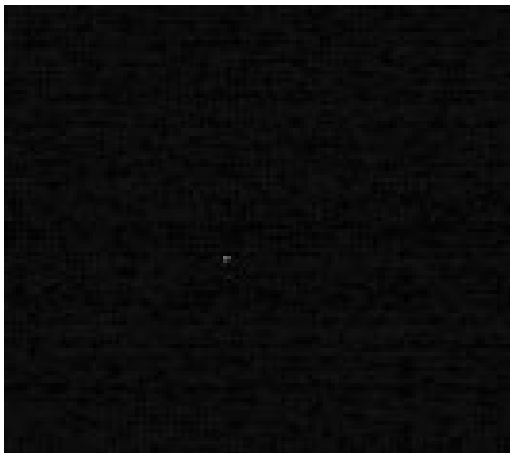

Fig. 5. One section of a 0.5s exposure obtained from the CogniSAT NEI camera of a starfield.

This image was collected with a 0.5s exposure and cropped to reduce downlink time without reducing the resolution of the section being downlinked. The image was converted to .png format to allow for lossless compression of the image, and to reduce downlink time. A second image (not shown) was collected of the same starfield with the same exposure time approximately 18 hours after the first image. The images are grayscale.

A few key characteristics to note about the image are the low diffraction, high spacecraft stability, and consistent and low noise level. The stars in the image are typically 2 pixels in diameter, indicating the low diffraction of this imager. This reduces the need for centroid determination to determine the star's location, but also precludes sub-pixelation techniques.

To explore the noise level, a histogram was produced of both images (see Fig. 6). Of note is the extent of the image noise. The image noise extends from a pixel value of zero to a pixel value of 20 on a uint8 scale (0 to 255). This is consistent across the image and neither image shows evidence of lens flare or other region-specific noise increases.

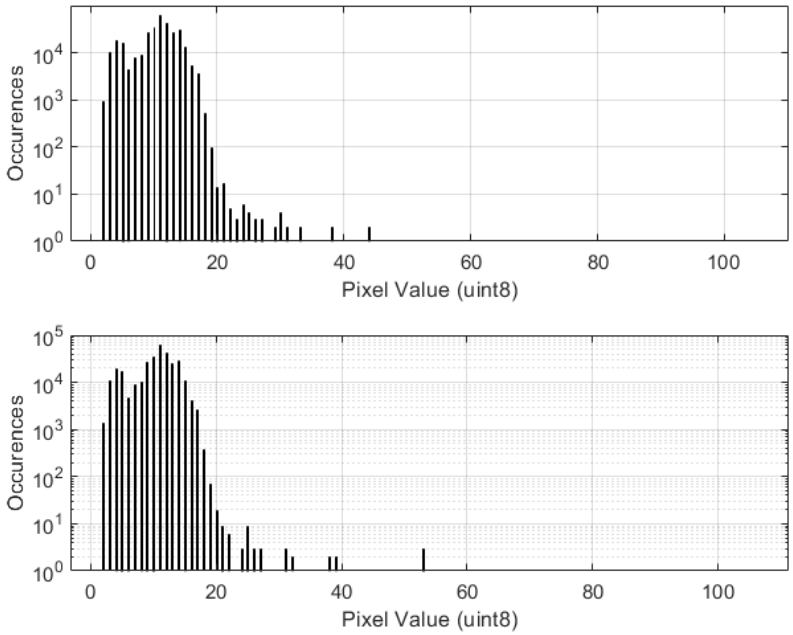

Fig. 6. The histograms of the two images

To characterize the impact of spacecraft stability on the images, the two images of the same starfield taken approximately 18 hours apart were compared (see Fig. 7). This figure was generated by binarizing both images about the value of 20, so that pixels above 20 (a value approximated from the histograms) were set to 255 and pixels below this value were set to zero. Then, a composite image was created in which pixels were white where both binarized images had pixel values of 255, black where both had pixel values of 0, and green or red where only one or the other binarized image had a pixel value of 255. This is done solely to illustrate the consistency of the star locations relative to pixel locations and the stability of the spacecraft attitude and imager, which is conducive to both RSO detection and future applications such as orbit determination. However, it also provides strong evidence that the detections are stars rather than shot noise or debris. The presence of the stars in the exact same pixels indicates the attitude repeatability or precision of the spacecraft to be <0.04 degrees which is the FoV pitch of a single pixel. This analysis also indicates that the stars are sufficiently stationary in the image such that nearby RSO should form streaks relative to this background starfield, assuming they have sufficient relative angular acceleration.

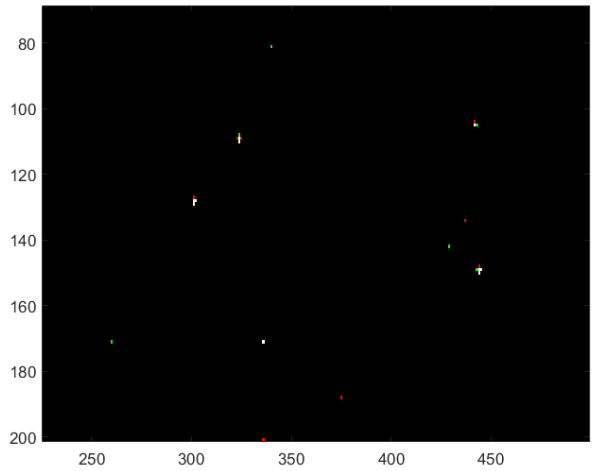

Fig. 7. A composite image showing where stars were detected in both images (white) or only one (red or green). A small portion of the image is displayed so that individual pixels are visible to the reader.

This analysis indicates the images from this camera should be conducive to RSO detection for sufficiently bright RSOs. Artificially increasing the exposure time through the summing of several images would be possible and increase the length of any RSO streaks but would also raise the noise floor for each image summed. See Fig. 8 for a histogram of the pixel-wise summation of the two previously discussed images. Alternatively, the images could be averaged, which would cause the motion of an RSO to generate a streak but would also reduce the overall pixel values (i.e., the brightness) of the streak.

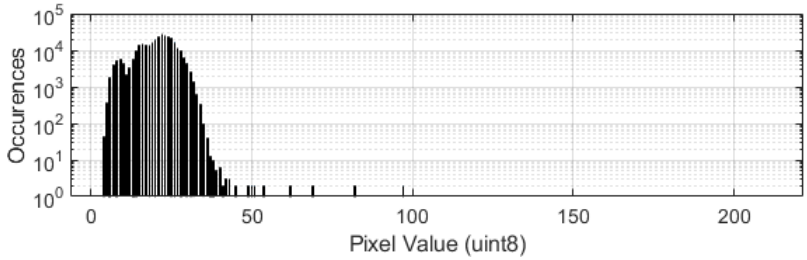

Fig. 8. A histogram of the pixel-wise sum of the two starfield images. Compare to Fig. 6.

3.3 Image pre-processing

We developed a pre-processing software suite which could be configured to provide a chain of image processing algorithms to be applied to the collected camera images before being input into the convolutional neural networks (CNN). The pre-processing software was designed to be flexible (i.e., to allow each of the image processing algorithms to be optional and configurable at runtime) allowing for a customized image pre-processing chain. The pre-processing software suite consists of a coordination script written in Python, capable of being configured at runtime through a JSON configuration file, and a suite of image processing applications written in C++. Fig. 9 shows the customizable image pre-processing chain.

Image Processing Applications:

- Grayscale - Converts the input RGB image into a grayscale image.
- Combine - Combines multiple images into a single RGB image using either a summing or averaging algorithm.
- Resample - Resamples the input image to the given dimensions.
- Tile - Converts the input image into multiple tiled images.
- Reformat - Reformats the image from an Int8 image to a floating-point 16 image.

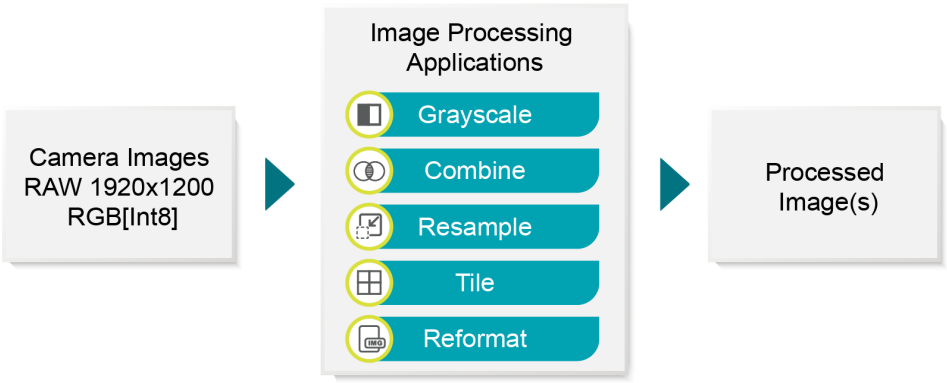

Fig. 9. Customizable image pre-processing chain

In testing, we found the CNN performed best with the pre-processing software configured to forego the grayscale and reformat steps, and to output 640x640 RGB (Int8) tiled images.

3.4 Inference

3.4.1 Details about the neural network structure

YOLOv4-tiny with a CSPDarknet29 backbone was selected to train, validate and test detecting RSOs with artificial intelligence [15]. YOLOv4-tiny is a member of the You Only Look Once (YOLO) suite of deep CNNs that are popular in computer vision because they excel in quick and accurate single pass detections that supply bounding boxes and class labels for objects in images and videos [16] [17] [18]. YOLOv4-tiny is a lightweight, compressed version of YOLOv4 that has less YOLO heads for prediction and a simpler network structure with only 29 convolutional layers [18]. YOLOv4-tiny has a relatively small model size of 23 megabytes (MB) and sacrifices accuracy for speed of detection; it achieves two-thirds the accuracy of YOLOv4 on the relatively complex Common Objects in Context (COCO) dataset [18]. As such, YOLOv4-tiny is an ideal candidate for deployment on edge devices with limited computing environments.

Other object detection models, for example YOLOv3 and EfficientNET, have been successfully used for RSO detection [5] [19]. However, neither model met our operational needs for a very small model weight file size. YOLOv3-Tiny and Tinier YOLO were also tested when determining our optimal lightweight object detection model, but these models were too large or not able to meet our accuracy expectations during training [20] [21]. The final version of YOLOv4-tiny we deployed contained an altered convolutional network structure to reduce the model size. The altered structure had two 512-sized convolutional layers removed from the neural network which reduced the model weight size to approximately 9 MB. Furthermore, during compilation on the Ubotica CogniSAT-XE2 AI co-processing system, the model layers underwent post-training quantization (PTQ) to make it deployable, further reducing the size to under 5 MB. The changes to the CNN structure were quantified by comparing accuracy metrics between the original model and a model with the altered structure. While the model weights were greatly reduced, there was only a minor drop in performance. The model size reduced from 23 MB to 9 MB, but the mean average precision (MAP) calculated from training decreased by only 5%. Based on our needs to deploy in a limited size, weight, and power (SWaP) environment on the edge, we decided the altered model structure with reduced size performed to our needs.

3.4.2 Initial bootstrap training data

Machine learning, when applied to object recognition, takes training data input, applies a model, and then makes distinctions between the classes of objects within the training set (which, in our case, are RSOs) and objects that are not members of the training set. A machine learning model's theory of knowledge produces justified beliefs about the class of a particular RSO using inference to generate a confidence interval deeming a candidate RSO to be a member of the training class. When this inference process works well, a true positive is generated where an RSO exists in the image under examination and its existence is detected and confirmed by the classifier. A training set must be generated to train a model so it can make true, positive inferences about the objects it examines in an image.

In our case, this training set was difficult to produce because the images under consideration are acquired in an arduous space-based environment. To overcome these challenging image acquisition circumstances, we utilized proprietary synthetic data generation process (patent pending) to bridge the gap in the difference between what our YOLO object classifier required and the number of genuine space images we had available. Our synthetic training images were generated using the streak generation process illustrated in Fig. 10.

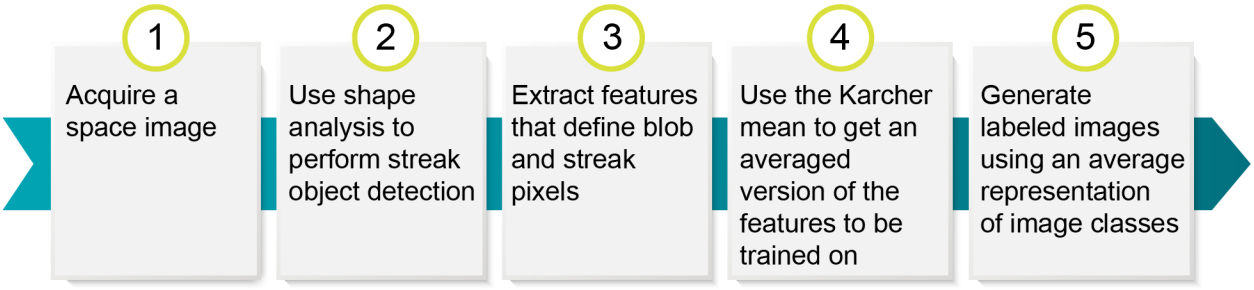

Fig. 10. Streak Generation Process

3.4.3 Using shape analysis for object detection

The problem of creating a synthetic dataset is that the beginning point must be based on an understanding of the features that constitute an object we are interested in classifying. Our foundation for extracting objects from an image did not rely on neural nets. After all, how could we use a neural net to generate a ground truth when no hand-labeled ground truth exists initially? We relied on mathematically rigorous techniques for extracting, classifying, and labeling objects in images. After performing these actions, we generated large synthetic datasets. To get a meaningful understanding of the objects that populated our synthetic dataset, we used elastic shape analysis (ESA) [22]. ESA allowed for a direct inquiry into an image to find objects that conformed to the shape of the object we wanted to find. The reference shape we used needed to only be a line 2 pixels wide and 20 pixels long.

3.4.4 Extracting features that define an object

Once we identified the streaks in our image, our next task was to discover the intrinsic qualities that defined the pixels comprising these streaks. The properties we were concerned with were the streak pixels' mean, median, and standard deviation within its grayscale color. Equipped with this knowledge, we were now able to add further realism to the streaks we intended to generate.

3.4.5 The Karcher Mean for object generation

The Karcher mean (KM) [22] refers to a method for finding how shapes can be averaged together. In our case, we were interested in how the KM allowed us to understand the average features that define a streak's feature-set. With this knowledge, the synthetic dataset's streaks are representative of what can be expected in a genuine space-based dataset.

3.4.6 Constructing a synthetic ground truth dataset

As we stated above, our challenge now was to make sure we could generate credible synthetic data for the purpose of training YOLO. With the above-named components in place, this challenge was addressed in a straightforward manner; for every training image to be produced, we randomly placed our synthetic RSOs in accordance with our generative parameters. Fig. 11 shows the results of our synthetic dataset generation.

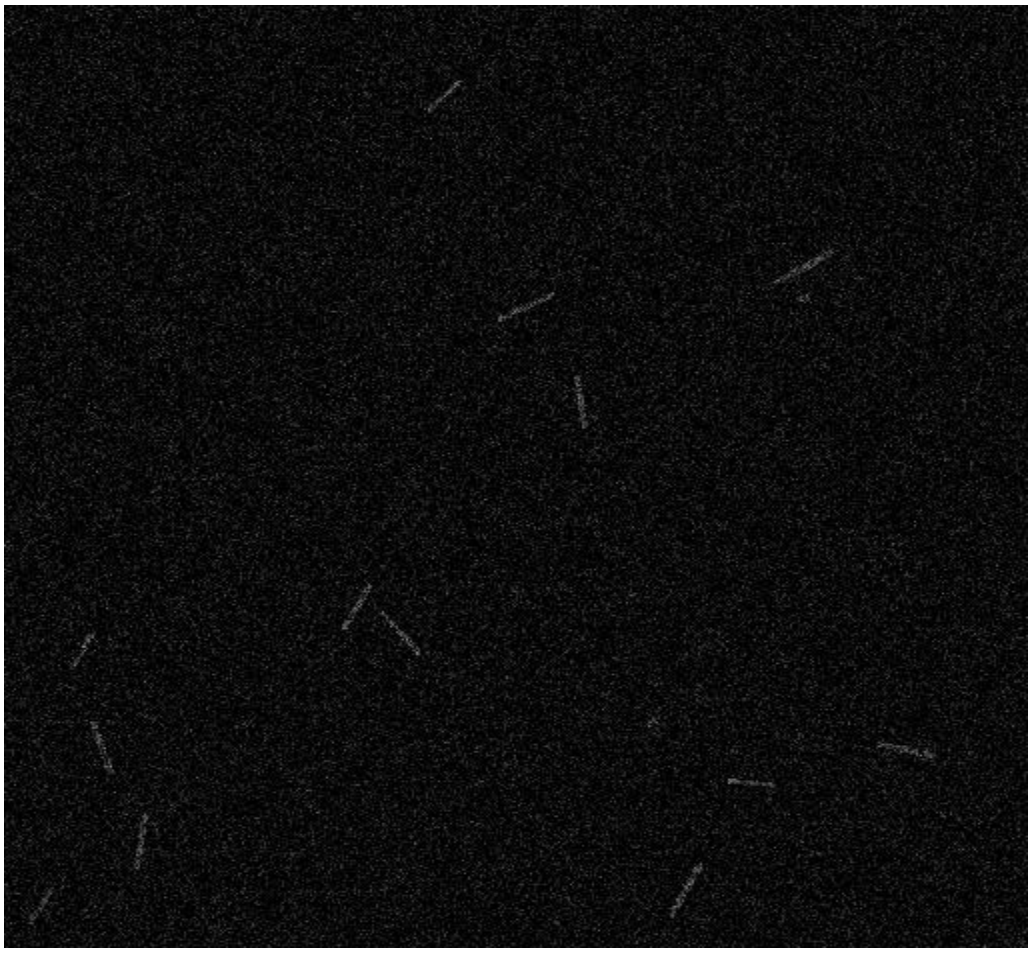

Fig. 11. Synthetic Image Example

3.4.7 Details of the training

To optimize model performance to detect satellite streaks and stars, we trained our model with a batch size of 64, a mini-batch size of 16 for 6,000 batches. The neural set was set to learn on 640x640 sized images with 3 bands. The learning rate during training was 0.00261 with momentum and decay values of 0.9 and 0.0005, respectively. We employed data augmentation techniques during training that randomly altered the following image properties: angle, saturation, exposure, and hue.

Training neural networks on hybrid image datasets, where some images are real and some are synthetic, have been shown to improve accuracy of CNNs, particularly in low-shot detection problems where a robust dataset of images may not exist [23]. As such, a hybrid image dataset that contained 3,292 total images, composed of 258 real SatNet images and 3,034 synthetic images, was used to train YOLOv4-tiny to detect RSOs as streaks in space imagery.

4. RESULTS

The goal of this research was to detect, with a neural network inference, an RSO in an image captured on orbit. However, as of the time of this writing, the research team encountered an unexpected regulatory roadblock that prevents on-orbit image capture of RSOs. To allow development to continue, an alternative plan was required in the interim.

Sections 3.4.2 through 3.4.6 describe how synthetic but feature-representative streaks can be generated to increase the chances that a NN trained on synthetic data will generalize sufficiently for real imagery. This technique was applied to the two test on-orbit images, which did not contain real RSO streaks. The result was an augmented image

consisting of a real on-orbit background image combined with feature-representative synthetic streaks. This represented the nearest representation possible before the receipt of real NEI after the regulatory roadblock is cleared.

To test the remainder of the previously described process, this image was then loaded onto the flat-sat for the Apex Aries SN1 spacecraft and inference was run using the Ubotica CogniSAT-XE2 AI Co-Processing Unit. Once the inference process was verified using orbit-rated hardware on the ground, the image was uploaded to the spacecraft and inference of the augmented image was performed on-orbit using the Ubotica XE2 Vision Processing Unit. The inference results were then downlinked from the spacecraft and overlaid on the augmented image. The results are shown in Fig. 12. Streak object detections are outlined in red and star object detections are outlined in green. The confidences of the inference are shown. The inference results from the flat-sat inference and the on-orbit inference were identical for the same image. Fig. 13 shows another inference example that was performed on the flat-sat, but not on-orbit.

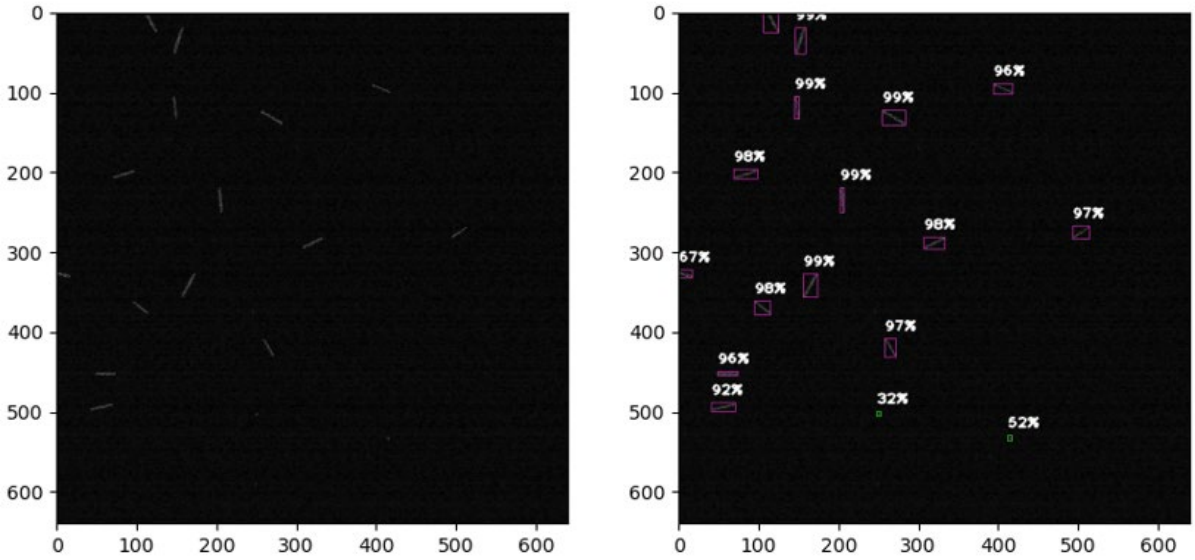

Fig. 12. Left: augmented image consisting of an on-orbit acquired starfield image overlayed with synthetic RSO streaks; Right: The same image with inference results obtained from an on-orbit inference overlayed.

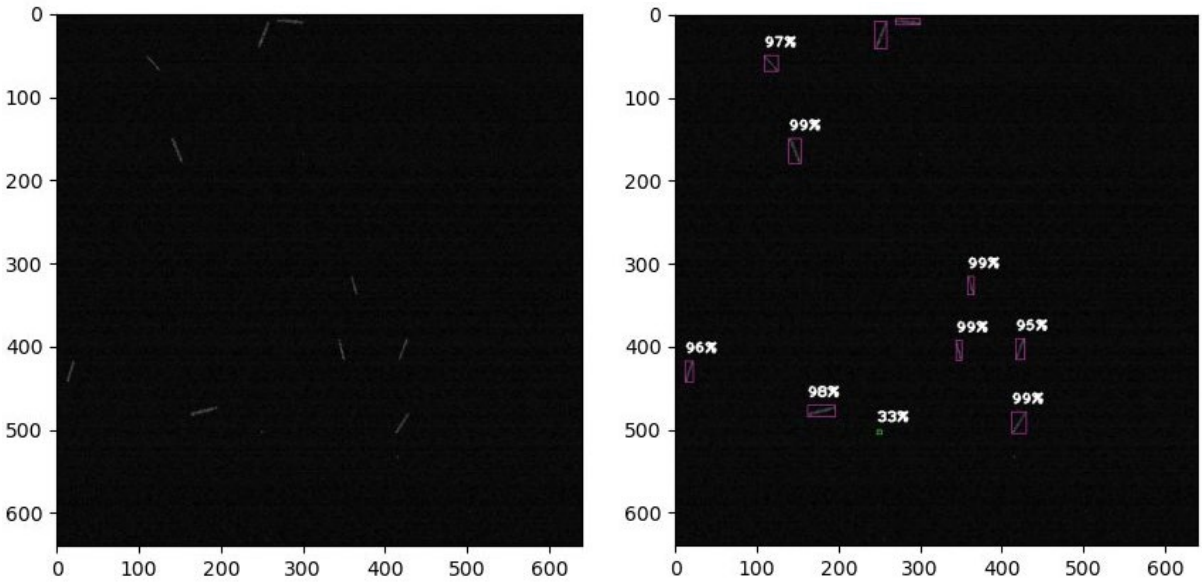

Fig. 13. Left: augmented image consisting of an on-orbit acquired starfield image overlayed with synthetic RSO streaks; Right: same image with interference results obtained from an interference on the flatsat overlayed

5. ANALYSIS

The downlinked inference results consist of a file of 70 kB for inference on a 640x640 section of collected image. It requires 6 inferences to completely process a 1920x1200 image. This means that rather than downlinking a 6.6 MB raw image, approximately 420 kB of inference results can be downloaded, reducing the downlink burden of an on-orbit SDA operation by approximately 16 times. If only positive inference results are downloaded, the communications requirements would be reduced even further.

The model was optimized during training by the MAP across all classes. The best model from training was selected where MAP@0.50 was 91.44% and precision, recall, and F1 were 0.91, 0.94, and 0.93, respectively. Streak objects were detected with a 98.74% accuracy in the validation dataset where there were 3,578 true positives and 34 false positives. Blob objects were detected at 84.15% accuracy in the validation dataset where there were 516 true positives and 51 false positives.

There are many reasons for YOLO to struggle in an extreme low-SWaP and space-located edge device including limited storage, relatively low bandwidth, unfamiliar lighting conditions, unforeseen space weather, and unfamiliar camera noise [23] [24]. However, removing layers from YOLOv4-tiny and training with hybrid imagery with a great variance in satellite streak brightness and image noise created a more robust model that suggested it could handle a domain shift for space applications.

5.1 Performance of YOLOv4-tiny on tasks that utilize on-orbit-captured NEI images

Domain shift is a common problem in the object detection field where a model may struggle to perform accurately when different lighting, camera angle, background, object appearance, or occlusion is present in imagery [24]. Synthetic imagery in a training set can be used as one tactic to help models improve generalization and robustness, because the objects themselves and the background can be manually altered to accommodate different or extreme conditions [23]. The combination of real and synthetic imagery in our training dataset allowed YOLO to overcome significant domain shift and, for example, correctly identify a satellite streaking across a telescope imagery with significant noise from a background nebula. See Fig. 14.

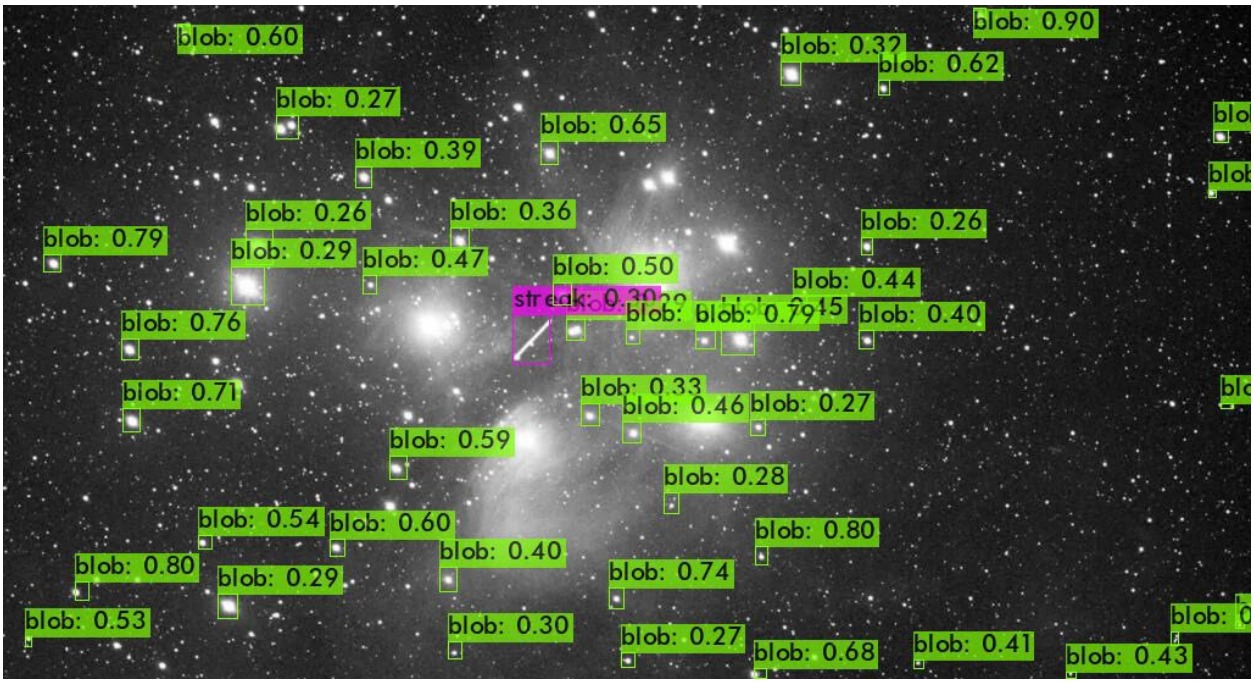

Fig. 14. Model inference conducted on a telescope image with a satellite streak correctly detected on purpose. Image collected by and used with permission from Aldoria Space (https://www.aldoria.com/sensors/)

6. CONCLUSION

6.1 Achievements

The goal of this project remains to detect RSOs using an on-orbit imager and a neural network trained for object detection running on on-orbit hardware. The use of the on-orbit imager to image RSOs is, at the time of this writing, legally prohibited pending clearance of an unexpected regulatory roadblock. However, the detection of feature-representative synthetic streaks in an augmented on-orbit starfield image was performed, and successfully demonstrated the feasibility of this technique for future RSO detection, and eventually tracking. Further it quantified the utility of relocating image-based RSO detection tasks from the ground to satellites. This is perhaps the most significant implication of the demonstrated technique: compared to link requirements for executing image-based RSO detection tasks on the ground, sensors that employ our onboard detection technique use over 93% less link throughput to relay valuable SDA data from on-orbit sensors to ground-based enterprise SDA data repositories. Once imaging of real RSOs is legally permissible, the team will begin exploring the detection limits of the technology, and engineering improvements to expand those limits towards less-detectable objects.

This research has so far successfully demonstrated a process for streak detection performed completely on-orbit. On the path to achieving this, the team constructed a mission planning process for selecting optimal times to observe known RSOs. The team selected particularly bright disused rocket bodies for the purpose of this experiment, but the mission planning process is easily transferrable to a wide variety of RSO types.

Additionally, this research involved the characterization of the Ubotica camera for this use-case. This is a novel application of this imager, but the image statistics suggest it will be adept at this application. In addition, the use of an "imager of opportunity," even one as capable as this, provides a test case for wider deployment of detection algorithms like the one described here onto other "imagers of opportunity" in support of expanding RSO detection efforts by SDA enterprises. These "imagers of opportunity" might even include hardware that is already on-orbit, as both public and private enterprises seek to cost-effectively increase the fidelity of their SDA catalogs.

Finally, this research developed a flexible image pre-processing software chain to support flexible tiling and synthetically-long exposures (summing or averaging multiple images), enabling further study on the optimal collection techniques for this application. This software chain will likely yield additional insights in the future into

optimal preparation of images for CV inferencing. A robust technique was presented here for developing synthetic datasets for CNN training which are specifically tailored to the RSO detection application. Finally, the use of an on-orbit edge Vision Processing Unit was demonstrated for the purpose of detecting image streaks caused by passing RSOs. In short, the hardware/software toolchain developed to carry out this experiment can continue to be leveraged to further expand the current state of the art in on-orbit CV-based detection.

6.2 Next Steps and Implications for Enterprises

Several steps remain to be addressed after this study. The most obvious is the collection and inference of images taken of known RSOs from orbit. Once that is accomplished and the process is demonstrated end-to-end, several other questions present themselves.

The mission planning section of this study illustrates how RSOs are most likely to become detectable in very specific and short time windows. A future system designed to detect RSOs from orbit whose presence is suspected or expected in a given region could make use of a tracking scheduler which specifically calculated these times. Previously untracked RSOs (such as debris and other previously-unnoticed objects) might be detected by simply looking away from the sun and into a dark background, but known RSOs associated with tracking information in extant catalogs requiring periodic updates would present the opportunity for optimal track update windows based on the closest approach and sun angles described here.

Further, once several images are obtained of RSOs from orbit, the characterization of the imager and a generalization of properties important for on-orbit imagers in this application should be conducted to inform future imager design. Finally, the extensibility of this application to other imagers, including other “imagers of opportunity” such as star trackers, should be pursued.

The > 93% reduction of link throughput requirements for RSO detection in space-based imaging, enabled by downlinking text-based detection location data rather than downlinking entire images and performing detection tasks on the ground, is the capstone achievement of this experiment so far. At scale, this bandwidth savings offers room to substantially grow sensor counts in existing SDA networks without incurring communications bottlenecks. This could enable improvements to sensor network angular diversity and sensor gap reduction through cheaply-scalable/proliferatable SDA sensors, which in turn could vastly improve the quality of SDA catalogs. In short, the potential SDA enterprise improvements unlocked by relocating image-based detection processing from the ground to edge-compute resources aboard the spacecraft are exciting indeed.

7. REFERENCES

[1] S. Dave, R. Clark and R. S. Lee, "RSOnet: An Image-Processing Framework for a Dual-Purpose Star Tracker as an Opportunistic Space Surveillance Sensor," *Sensors,* vol. 22, no. 5688, 2022.

[2] R. Clark, G. Chianelli and R. Lee, "Machine learning implementation for in-orbit RSO orbit estimation using star tracker cameras," in *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2020.

[3] S. Jahirabadkar, P. Pande and R. Aditya, "A survey on image processing based techniques for space debris detection," in *IEEE Bombay Section Signature Conference (IBSSC)*, Bombay, 2022.

[4] M. P. Levesque and S. Buteau, "Image processing technique for automatic detection of satellite streaks," Defense Research and Development Canada, Canada, 2007.

[5] J. Fletcher, I. McQuaid, P. Thomas, J. Sanders and G. Martin, "Feature-based satellite detection using convolutional neural networks.," in *Advanced Maui Optical and Space Surveillance Technologies*, 2019.

[6] C. Jeffries and R. Acuna, "Detection of streaks in astronomical images using machine learning," *Journal of Artificial Intelligence and Technology,* vol. 1, pp. 1-8, 2024.

[7] L. Valera, L. E. Boucheron, N. Malone and N. Spurlock, "Streak detection in wide field of view images using Convolutional Neural Networks (CNNs)," in *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2019.

[8] K. Fujimoto, M. Uetsuhara and T. Yanagisawa, "Statistical Track-Before-Detect Methods Applied to Faint Optical Observations of Resident Space Objects," in *Advanced Maui Optical and Space Surveillance Conference (AMOS)*, 2015.

[9] S. Virani and M. Holzinger, "Real-time Multi-target Detetction & Tracking of Space Objects using FiSSt Methods," in *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, 2020.

[10] M. Uetsuhara and N. Ikoma, "Faint Debris Detection by Particle based Track-before-detect Method," in *Advanced Maui Optical and Space Surveillance Conference (AMOS)*, 2015.

[11] P. Dao, R. Rast, W. Schlaegel, V. Schmidt, S. Gregory and A. Dentamaro, "Track-Before-Detect Algorithm for Faint Moving Objects based on Random Sampling and Consensus," in *Advanced Maui Optical and Space Surveillance Conference (AMOS)*, 2014.

[12] T. Delaite, J. Couetdic, E. Glemet and F. Cassaing, "Performance of an Optical COTS Station for the wide-field Detection of Resident Space Objects," in *The Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2023.

[13] B. Wang, H. Wang and Z. Jin, "An Efficient and Robust Star Identification Algorithm Based on Neural Networks," *Sensors,* vol. 21, no. 7686, 2021.

[14] D. Rijlaarsdam, H. Yous, J. Byrne, D. Oddenino, G. Furano and D. Moloney, "Efficient Star Identification Using a Neural Network," *Sensors,* vol. 20, no. 3684, 2020.

[15] A. Bochkovskiy, "Darknet YOLO," [Online]. Available: https://github.com/AlexeyAB/darknet. [Accessed 21 May 2024].

[16] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning.," *nature,* vol. 521, no. 7533, pp. 436-444, 2015.

[17] J. Redmon, S. Divvala , R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection.," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[18] A. Bochkovskiy, C.-Y. Wang and M. L. Hong-Yuan, "Yolov4: Optimal speed and accuracy of object detection.," *arXiv,* 2020.

[19] N. AlDahoul, H. A. Karim, A. De Castro and M. J. T. Tan, "Localization and classification of space objects using EfficientDet detector for space situational awareness.," *Scientific reports,* 2022.

[20] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement.," *arXiv,* 2018.

[21] W. Fang, L. Wang and P. Ren, "Tinier-YOLO: A real-time object detection method for constrained environments.," *Ieee Access,* vol. 8, pp. 1935-1944, 2019.

[22] A. Srivastava and E. Klassen, Functional and Shape Data Analysis, New York: Springer-Verlag, 2016.

[23] V. Seib, B. Lange and S. Wirtz, "Mixing Real and Synthetic Data to Enhance Neural Network Training - A Review of Current Approaches," *arXiv,* 2020.

[24] K. Saenko, B. Kulis, M. Fritz and T. Darrell, "Adapting visual category models to new domains.," 2010.

[25] A. Cabello and J. Fletcher, "SatSim: A synthetic data generaion engine for electro-optical imagery of resident space objects.," in *Sensors and Systems for Space Applications XV*, 2022.

[26] C. Patel, R. Gopalan, L. Ruonan and R. Chellappa, "Visual domain adaptation: A survey of recent advances.," *IEEE Signal Processing Magazine,* 2015.