

Multi-Frame Imaging to Detect Orbital Debris

Tim Vrakas

Abstract—Debris in Earth orbit poses a significant risk for high-density satellite constellations. One promising approach to improve the tracking of debris is to use the existing star tracker cameras present on most satellites. Conventional Star Tracker imagers use static image stacking to improve SNR, which reduces the SNR of RSOs moving across the frame. We propose a filtering approach with Convolutional Neural Networks to process sequences of star tracker images. This approach removes static background features and noise, and extracts moving features, allowing more robust detection of RSOs.

Index Terms—Computational Photography, Space Safety, Orbital Debris

1 INTRODUCTION

ORBITAL debris poses a major threat to the operation of LEO satellite constellations, the density of which are increasing rapidly. The Starlink constellation now contains over 5,000 operational space vehicles. While these satellites can maneuver to avoid collisions with each other and with cooperative vehicles, there exist thousands of non-tracked, non-maneuverable, or non-cooperative objects on potentially intersecting orbits. Maintaining space safety requires active tracking of thousands of objects, with a revisit rate frequent enough to track changing orbits. Existing ground radar systems cannot reliably track small objects, and do not provide a target revisit rate frequent enough to track maneuvering objects or rapidly decaying orbits.

One proposed approach to improve the tracking of Resident Space Objects (RSOs) is to utilize imagers already on orbit [1]. Satellites utilize “Star Trackers” to determine their orientation in space, by comparing the position of visible stars against a known almanac of star patterns. These imagers are specifically designed to precisely identify the angular position of dim stars in noisy conditions. These imagers could also be used to track the RSOs as they pass near the observing satellite and are illuminated by the sun.

Many existing star tracker systems perform image stacking to improve the SNR of the captured image and improve the detectability of dim stars. When a vehicle has angular velocity, it may perform alignment on the images before stacking. However, this alignment is only effective for background objects which maintain constant relative position between frames. Objects passing near the vehicle will have significant relative velocity, and will move across the star field background between frames. When these frames are stacked with static alignment, the SNR of the moving object will be worse in the final image, and it will be difficult to identify its position at a specific moment in time.

2 RELATED WORK

In [2], the authors propose the use of a camera system on one satellite to determine the attitude and health of a



Fig. 1. A stack of 60 Starlink satellites prepared for launch. A Star Tracker camera, with red lens cover, is visible on each.

nearby vehicle, based on the pattern reflected light over time, known as the “light curve”. They propose a convolutional neural network with the goal of object classification, consisting of three convolutional layers, followed by three fully connected layers. This network is trained on synthetic data generated by a simulation, based on ground measurements. This network demonstrates $> 80\%$ effectiveness at classifying the tumbling status of a nearby spacecraft.

In [1], the authors assess the feasibility of using a Commercial Off the Shelf (COTS) star tracker for RSO Detection.

- T. Vrakas is a Masters Student in the Electrical Engineering department at Stanford University
E-mail: timv@stanford.edu

They predict a star tracker detection system should be capable of detecting between 1 and 10 RSOs per day, as a rough estimate. They also propose the use of a CNN to process individual star tracker frames and extract objects with signatures likely to be RSOs.

In [3], the authors propose the dual-use of star tracker imagers for both attitude determination, and tracking of RSOs. They propose the use of a CNN to classify object types, and characterize the health and activity of objects. They also develop an approach to compute orbital parameters for detected objects based on observed image position.

In [4], the authors utilize the large scale of the Starlink constellation to measure the Earth’s radiation environment with very high spatial resolution. They demonstrate the value of a “big data” approach enabled by constellations of thousands of identical vehicles, with effectively unlimited downlink bandwidth for the purposes scientific data collection.

3 PROPOSED METHOD

We propose a method improve the image processing pipeline of star tracker imaging systems, with the goal of improving the sensitivity of the system to the signatures of RSOs. Instead of performing image stacking with a fixed offset to improve SNR, we may use a 3D convolutional neural network, which processes a series of image frames as a single 3D dataset, and can provide filtering specifically trained to improve the detection of RSOs. This approach is able to remove noise from the image easily, as well as removing fixed stars in the background, while extracting features that move in a continuous path across the image.

3.1 Characterizing available star tracker images

A dataset of the sequences of star tracker images does not yet exist, so we needed to first develop a synthetic dataset to demonstrate it’s feasibility. I modeled the characteristics of the synthetic data off single image frames publicly available in [4]. In the format available these images are 8-bit greyscale, though the specific format of the raw data captured will be dependent on the star tracker sensor hardware. Figure 2 shows the a histogram of the intensity values in the image. From this histogram, we can approximate the noise characteristics of the imager. The signature is dominated by Poisson noise, with several prominent bins of intensity spaced with a period of 12.75 integer values. The mean of this scaled Poisson distribution is approximately 3.2 of these intervals. Figure 2 also shows a Poisson distribution with $\lambda = 3.2$, scaled by a factor of 12.75. In addition to this Poisson noise, there is Gaussian noise, measurable as the width of the data in each bin of the reference distribution. For the Gaussian noise, $\sigma = 2$.

3.2 Generating synthetic star field images

For each image in our synthetic dataset, we sample a Poisson distribution to pick the number of stars in the image. The coordinates of each star is generated from a uniform distribution over the image. The intensity of each star point is a function of the distance (uniformly distributed) and the stellar mass, which is sampled from a Salpeter Stellar Mass

Function (SMF). We iterate over all the simulated stars, and for each add a Gaussian Point Spread Function (PSF) to the image, with scaled intensity. The resulting ideal starfield is shown in Figure 4, upper left. Next, we can apply noise to the image. We sample the image according to the parameters of our Poisson noise, and then add Gaussian noise to the image. In Figure 3, we compare the original reference image to a synthetic image, with intensity histograms of each. This synthesis approach very closely models the noise profile of the image, and the images are nearly identical in visual characteristic.

3.3 Generating Training Data

The training data consists of 1200 independent events. Each event consists of 10 frames, each frame is 200x200px in size, 8-bit grayscale. To generate the data for each event, we start with a static background star field. Then, we choose a motion vector and brightness for the RSO, and overlay it at the correct position in each frame. Then, we apply the our noise profile to each frame. Of the 1200 events in the dataset, 1000 are used for model training, and 200 are used for validation.

4 CNN DESIGN

Due to the size of each entry in the dataset, the Convolutional Neural Network was very memory-intensive to train. The model was trained on a CUDA-enabled GPU with only 4GB of RAM, and so the structure of the neural network was kept relatively simple. Table 1 details the structure of the neural network, which consists of 4 convolutional layers. The result image has the same dimension as the input image. We use both 2D and 3D convolutional layers, to provide robust denoising on each individual frame, and to allow the network to understand trends spanning multiple frames, that is, in the 3rd dimension of the input data. In all cases we used a kernel of size 5. With more memory available for training, a larger kernel may be more effective at tracking objects that move rapidly between frames. The first two convolutional layers expand the data into 240 channels with standard Convolutional layers, and the last two layers collapse the data back to the original shape using Transposed Convolutional operators [5]. Following each of these four layers, we apply a ReLU activation function, with a Hyperbolic tangent activation function after the final layer, to help produce an image with well defined regions of interest.

4.1 CNN Training

The CNN was trained for 3 Epochs, with all 1000 training events used in each Epoch. The model is trained in batch sizes of 22 events. Training takes approximately 10 minutes to complete with GPU CUDA acceleration enabled. Figure 5 shows the progression of training loss over the three Epochs performed. More Epochs did not yield significant improvement in final loss, so changes to the structure of the network or use of an adaptive learning rate may be needed to achieve greater performance.

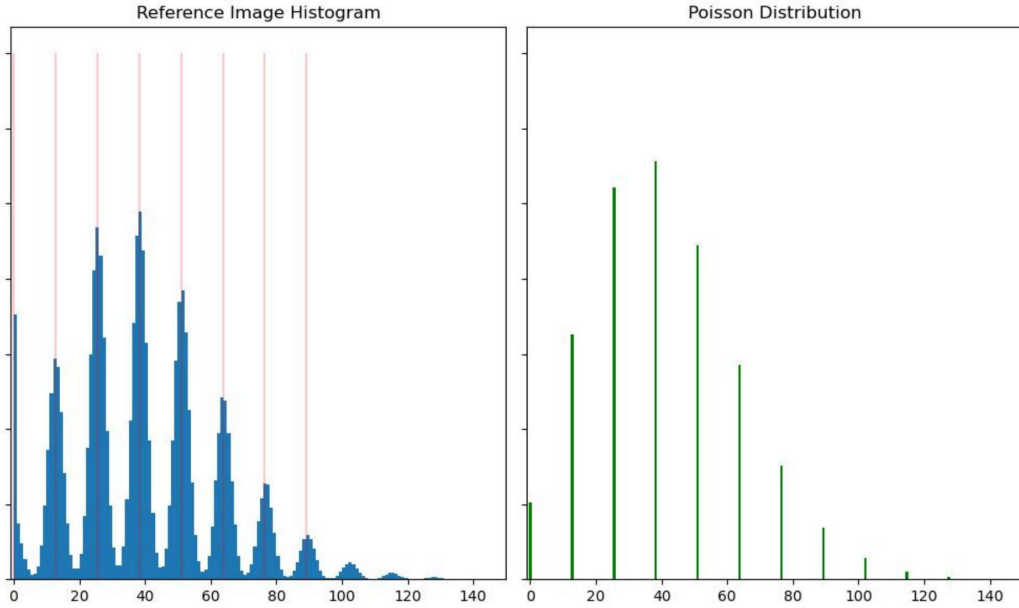


Fig. 2. Noise characteristics of an available star tracker image, with a reference Poisson distribution for comparison

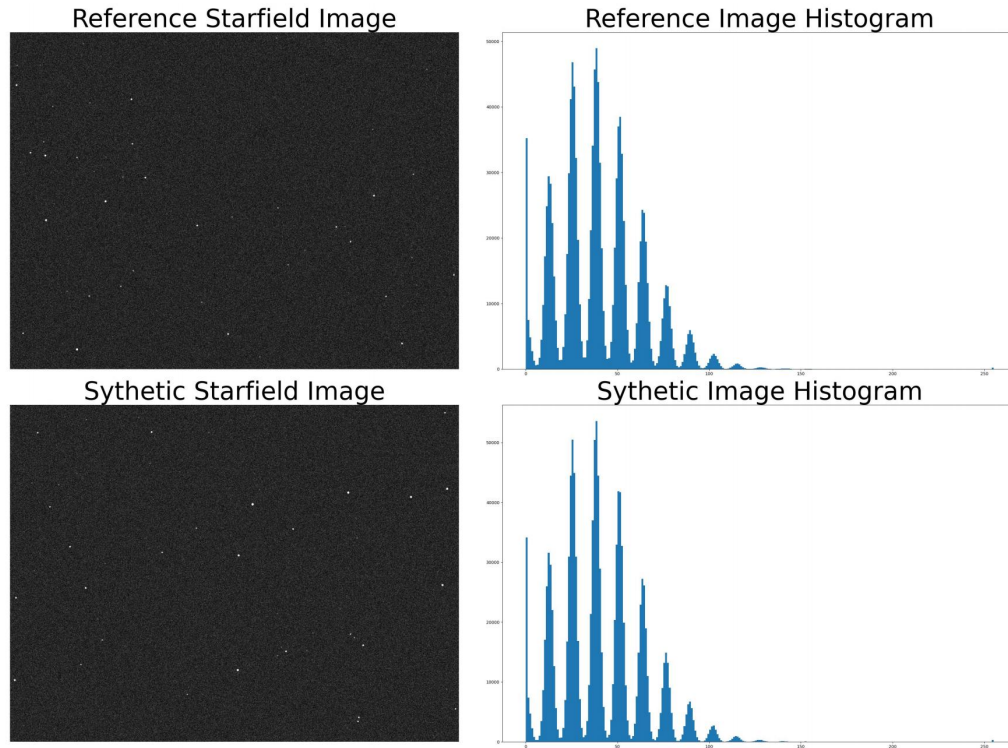


Fig. 3. The reference starfield image, and histogram of intensity, alongside a synthetic starfield image, and a corresponding histogram.

5 RESULTS

The CNN is effective at extracting the RSO feature from images in the validation set. Figure 6 shows a image from the validation set, where the model has extracted the RSO

location in the image, and has removed the static stars and noise from the data. Once training is complete, the model achieves a loss of 8.21×10^{-6} on the validation dataset. Included in attached code is an animated GIF of a sequence

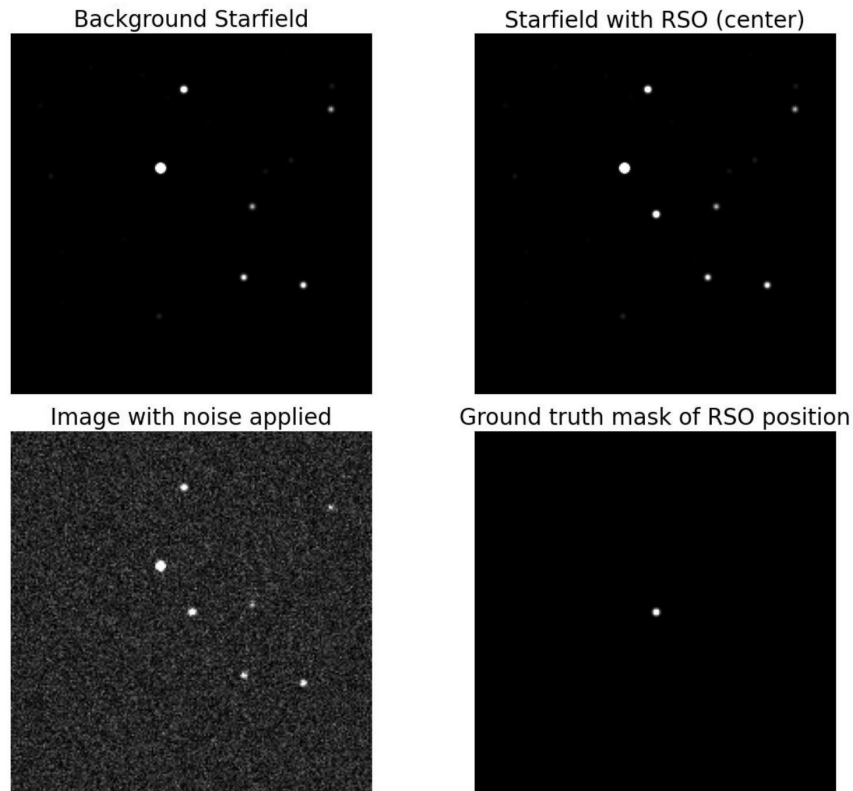


Fig. 4. The steps required to produce the synthetic training images. First, we create an ideal starfield image (top left). Then, we add an RSO to the image at some defined position (top right). Then, we apply the noise profile to this image (bottom left). We also save a ground-truth image containing just the RSO (bottom right).

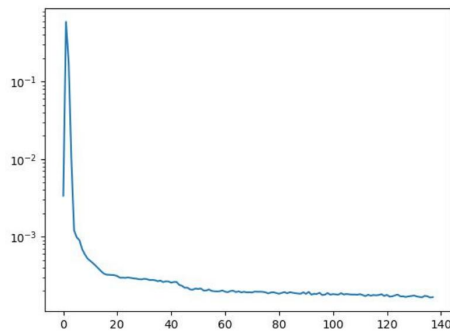


Fig. 5. Loss progression of training process, for three Epochs of training

of images from the validation set, demonstrating the ability of the CNN to track the moving RSO as it moves across the image. In the sample results, the output images track the RSO position across each frame, and generally have a well-defined region of interest. However, the centroid of the result image may not track the centroid of the RSO perfectly. To implement a precise centroiding step, we would use the output of the CNN as a mask to expose the regions of interest we suspect contain RSOs, and would then apply this

TABLE 1
Details of the layers of the CNN

CNN Layer	Layer Type	Input Shape	Output Shape
Layer 1	3D Conv.	10x200x200	10x12x200x200
Layer 2	2D Conv.	120x200x200	240x200x200
Layer 3	2D Transposed Conv.	240x200x200	120x200x200
Layer 4	3D Transposed Conv.	10x12x200x200	10x200x200

mask to the original frames, allowing us to centroid the true location of the object.

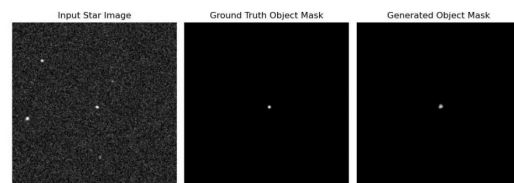


Fig. 6. Trained CNN applied to validation data

6 CONCLUSION

This approach to processing multi-frame sequences of star tracker images shows significant promise, and is worth further exploration. The results demonstrate that the approach of processing multiple frames together as a single capture is effective both to extract RSOs moving across the frame, and to remove the noise introduced by the imager into each frame.

There are several ways we may explore further development of this approach in the future.

First, in order to properly validate the model, we should build dataset of real star tracker images, to supplement the synthetic data. For best effect, we should capture sequences of images from star trackers that contain RSOs moving on known trajectories visible in the frame. This should be feasible using Starlink satellites, because conjunctions with debris happen many times per day.

Second, we can improve the fidelity of the synthetic dataset by taking into account the non-ideal geometry of the lens, and fixed pattern noise of the sensor. The synthetic data currently assumes an ideal imaging geometry, with no distortion. This is not realistic, and the model could be improved to capture these effects in the training data. Similarly, the model only Gaussian and Poisson noise to the synthetic data, and does not include fixed pattern noise in the sensor, or noise caused by high-energy particle radiation. These features could all be included to develop a more accurate training dataset.

Third, we should explore using a CNN with different structures. With a more capable training system, we could explore CNNs with more layers, or more complex structures like U-Nets. More work is needed to understand the optimal structure of a CNN designed for this task.

Fourth, additional work will be needed to post-process the extracted RSO signal, perform centroiding to determine object position and velocity, and determine if a detection system built around this approach is viable for use in space.

Further exploration of advanced computational imaging strategies, like the use of CNNs for image processing onboard satellites, potentially very valuable. In addition to the attitude determination and object tracking we have discussed, space vehicles use imaging systems for diagnostic purposes, and many carry payloads of optical and multi-spectral imagers, as well using optical data links for communications. The industry tends to avoid more "complex" solutions, perhaps due to a hard-learned respect for the challenges of the space environment. However, as modern computing technology becomes more prevalent in space, there are many opportunities to bring the performance of modern computational imaging to bear.

REFERENCES

- [1] S. Clemens, R. Lee, P. Harrison, and W. Soh, "Feasibility of Using Commercial Star Trackers for On-Orbit Resident Space Object Detection," in *The Advanced Maui Optical and Space Surveillance Technologies Conference*, S. Ryan, Ed., Sep. 2018, p. 23.
- [2] G. Badura, C. Valenta, and B. Gunter, "Convolutional neural networks for inference of space object attitude status," 09 2020.
- [3] S. Dave, R. Clark, and R. S. K. Lee, "Rsonet: An image-processing framework for a dual-purpose star tracker as an opportunistic space surveillance sensor," *Sensors*, vol. 22, no. 15, 2022. [Online]. Available: <https://www.mdpi.com/1424-8220/22/15/5688>
- [4] H. Shah, R. Cleave, W. McAlpine, Z. Fleetwood, K.-T. Pham, and S. Shermer, "Global ionizing radiation environment mapping using starlink satellite data," *IEEE Transactions on Nuclear Science*, vol. PP, pp. 1–1, 01 2023.
- [5] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2018.



Tim Vrakas Tim Vrakas is a Masters Student in Electrical Engineering at Stanford University. In his spare time, he works as an Electrical Engineer at Space Exploration Technologies Corp, on the Starlink satellite constellation. At Stanford he developed electronics for the Stanford Space Initiative (SSI). He now lives in Redmond, WA.