

Survey Mode: A Review of Machine Learning in Resident Space Object Detection and Characterization

Konstantinos Tsapralis* and George Choumos†

National Observatory of Athens, IASARS, BEYOND CENTER, Athens, 11523, Greece

National Kapodistrian University of Athens, Athens, 10563, Greece

Vaios Lappas‡

National Kapodistrian University of Athens, Athens, 10563, Greece

Charalampos Kontoes§

National Observatory of Athens, IASARS, BEYOND CENTER, Athens, 11523, Greece

The number of Resident Space Objects (RSO) in Earth's orbit has been rising steadily since the first satellite was launched. The need to protect functioning space assets from collisions with space debris is becoming more urgent as the number of debris has increased considerably partly due to a number of anti-satellite weapons tests which have created large debris clouds. Operational satellites have also been increasing exponentially due to the planned and already launched constellations. The field of Space Surveillance and Tracking (SST) has been established to try to mitigate the risks of collisions, using sensors such as radars and optical telescopes to detect, characterize, and catalog RSOs. This paper aims to provide a survey of the Machine Learning based methods that are being used to detect and characterize RSOs while also analyzing some non-machine learning approaches.

I. Nomenclature

<i>RSO</i>	= Resident Space Object
<i>SST</i>	= Space Surveillance and Tracking
<i>SSA</i>	= Space Situational Awareness
<i>ML</i>	= Machine Learning
<i>DL</i>	= Deep Learning
<i>ANN</i>	= Artificial Neural Network

II. Introduction

"Space," [the Hitchhiker's Guide to the Galaxy] says, "is big. Really big. You just won't believe how vastly hugely mindbogglingly big it is. I mean you may think it's a long way down the road to the chemist, but that's just peanuts to space." [1]. This vastness is now increasingly occupied by a growing number of Resident Space Objects (RSOs), a consequence of the continuous increase in space launches and anti-satellite weapon tests. Such an increase in Earth's orbit occupancy not only highlights the enormity of space but also underscores the escalating risks both to operational space assets (due to the increased risk of collision with other assets or debris that would render them unusable) but also to the space industry as a whole by increasing the probability of the Kessler Syndrome, a theoretical scenario where each RSO collision leads to the creation of more debris which in turn increases the number of collisions until the probability of collision is so high that would render the launch and operation of satellites impossible.

This growing population of RSOs presents significant challenges for the field of Space Situational Awareness (SSA), which tries to monitor and catalog this large number of objects in order to timely manage potential conjunctions. Space

*Research Associate, National Observatory of Athens, IASARS, BEYOND CENTER, k.tsapralis@noa.gr and kostis.tsapralis@gmail.com

†Research Associate, National Observatory of Athens, IASARS, BEYOND CENTER, g.choumos@noa.gr and g.choumos@gmail.com

‡Professor in Aerospace Systems, Department of Aerospace Science & Technology, National Kapodistrian University of Athens, Greece, valappas@aerospace.uoa.gr, Senior Member AIAA.

§Research Director, National Observatory of Athens, IASARS, BEYOND CENTER, kontoes@noa.gr

Surveillance and Tracking is the field of SSA that deals with artificial objects and encompasses a number of applications such as RSO detection and classification, RSO characterization, orbit determination, conjunction analysis, reentry prediction and RSO catalog creation and maintenance.

In this paper we will focus on two applications: RSO detection and RSO characterization. RSO detection is the application of identifying and tracking objects in space, while RSO characterization is the application of obtaining detailed information about the objects' physical properties and potential functionality. Machine learning and deep learning are fields that have shown great promise in improving the accuracy of algorithms in many different domains. Given this widespread adoption, the goal is for the paper to serve as an overview on the use of machine learning based techniques for these two applications by surveying the available literature and grouping them by the algorithms and datasets used so that researchers in the field interested in applying ML and DL based techniques in their work have a starting point in what has already worked and how they can expand on that.

The paper first presents the methodology used to query for published papers on the subjects, filtered for the subset of interest. Then an analysis is performed of the methods used in each application. Due to the nature of the applications, the majority of papers retrieved were found to use data from optical telescopes, hence the field of computer vision is dominating the methods used to detect and characterize RSOs. Traditional techniques of image processing were being used almost exclusively up to the last 6-7 years, while recently ML-based and more specifically methods utilizing artificial neural networks are beginning to eclipse the rest.

III. Literature Survey Methodology

The survey of the literature was implemented by utilizing the advanced search options of the Google Scholar platform. Google Scholar is a search engine specifically for scholarly literature that allows the use of complex queries to retrieve very targeted results.

In our case we were interested in finding papers that were using machine learning or deep learning techniques and applications related to space surveillance and space situational awareness so we used broad terms to try to retrieve as many papers as possible. The actual terms and query that were used are the following:

("machine learning" OR "deep learning") AND ("satellite collision" OR "collision avoidance" OR "orbit prediction" OR "space situational awareness" OR "space surveillance" OR "kessler syndrome")

In order to improve the workflow an open source tool called sort-google-scholar(<https://github.com/WittmannF/sort-google-scholar>) was used which programmatically queries the Google Scholar API, fetches and sorts the results and saves them in comma separated values (csv) files. The tool also allows for a limit to the number of results that are returned, which we set to 300. The tool was run on a yearly basis starting in 2007 up until September 2023.

The results can be seen in figure 1, and we believe represent the evolution of interest in the field of SSA/SST over time. The number of papers before 2010 is very low while as the 2010s progress the number of papers steadily increase and after 2018 we have an almost exponential increase (true number of papers cannot be seen due to the 300 papers limit we have set in the sort-google-scholar tool).

The next review step was going over each of the paper and by analyzing it, classify it 5 categories from highly relevant/relevant (meaning that the paper is using machine learning methods in the context of an SSA/SST specific application) to review papers, a "maybe" category for papers that either don't use machine learning techniques or are using but it's not a relevant application for us (e.g. ML for predicting Space Weather) to finally an irrelevant category for papers that briefly mentioned one of the terms used in the query but are not related to the domain. This categorization and the evolution of the relevance of the papers can be seen in figure 2.

Finally once the relevance sorting had been completed we used the list of very relevant/relevant papers to further group them by application. The popularity of each application can be seen in figure 3 and highlight the different applications that have evolved over time in the field of SSA/SST. From this list RSO characterization and RSO detection were chosen for this paper for their closely related thematic.

IV. RSO detection

RSO detection is the process of identifying and locating objects in space, such as satellites, debris, and other celestial bodies. This involves using ground-based or space-based sensors, like radar and optical telescopes, to continuously monitor the space environment and track the positions and trajectories of RSOs. The primary goal of RSO detection is to locate and segment the part of the sensor output that contains space objects. This review focuses on the detection of

Artificial Intelligence for SST - Results per year (2007-2023)

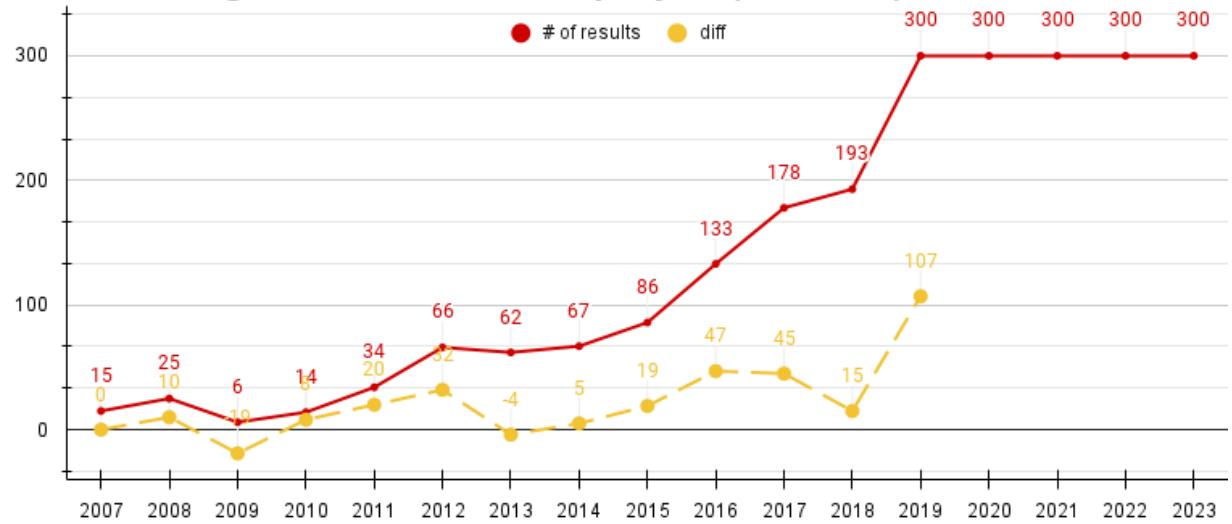


Fig. 1 Number of papers related to SSA/SST and ML/DL

space objects in optical telescope images and specifically when objects are non-resolved (meaning the satellite features cannot be distinguished from the image as they span only a few pixels).

Detecting the space objects is the first task in a series of processing steps in order to maintain an up-to-date catalog of space objects, which is essential for providing the necessary SST services such as collision avoidance, fragmentation and reentry analysis, satellite maneuvers and understanding the overall orbital congestion.

A. Non-Machine Learning Processing Methods

As a first step we will aggregate all the methods for detecting RSOs that don't use machine learning techniques. It should be noted that this should not be treated as an exhaustive list of all classical image processing methods since the query that was performed included specifically the terms machine learning or deep learning, so the papers that will be presented here are a by-product of this search.

Starting with the first paper [2] in 2005, Edith Stöveken et al. presented a few methods to specifically detect space debris in optical telescope images (vs stars and artifacts - hot/dead pixels and cosmic ray events). The first method is utilizing a star catalog to create a mask that removes the known stars in the captured image, then the objects of interest remain. A second method is using the object characteristics, to distinguish between point and streak like objects. The use of a series of images from the same region of the sky can be used to calculate the mean value of all images in each pixel position which can create a mask that only highlights stars. Subtracting this mask from the following images removes stars and moving objects of interest remain. A final method is presented which requires a sequence of at least three images, where the apparent velocity is calculated from the apparent shift in the object from the first to the second, then it is extrapolated to where it is expected in the third image. If that object is found it can be considered an object of interest.

Li, K. et al [3] in 2009 introduced the concept of using Mathematical Morphology (MM) [4] in space debris detection which is a set of four basic operators (erosion, dilation, opening and closing) that can be applied at images to process geometrical structures. The idea behind these operators is the use of a secondary matrix (called "probe") that is then applied in each pixel of the telescope image, with each operator performing a task in the specific domain. For example applying the erosion operator in the telescope images can remove object effects that are smaller than the selected probe size. On the other hand applying the dilation operator will augment objects in the image that are smaller than the probe. Finally the opening and closing operators are used to remove a specific type of noise called salt and pepper that is present on binary images.

In 2013 Sun, R.-Y. et al [5] capitalized on the Mathematical Morphology techniques by publishing a full pipeline of 5 steps to detect and localize space debris in optical telescope images: (1) Bias/dark/flat calibration (not usually needed

AI for SST - Results relevance over the years 2007-2023

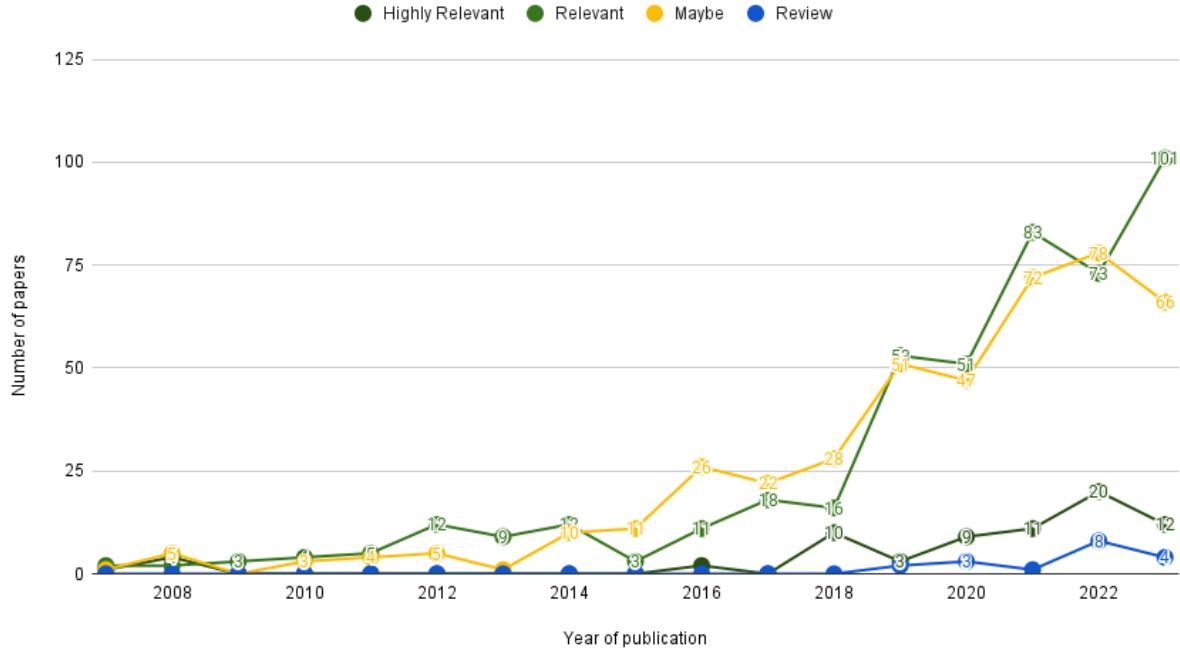


Fig. 2 Number of papers based on relevance to SSA/SST domains and ML/DL usage.

in space debris detection as mentioned by authors); (3) Mathematical morphology filtering and enhancement (this is the core of the process, using the top-hat transformation operator (Fig. 4) a non-linear high-pass filter is created which removes the smear noise and improves the image quality); (4) Global thresholding (a threshold value is chosen and a bit mask is produced); (5) Subtraction and segmentation (the mask is subtracted from the image and segments are calculated in the resulting image by identifying areas with at least 8 connected pixels using the two pass union-finding algorithm); (6) Determining the image positions in pixel coordinates (using a point spread function (PSF) fitting technique).

Progressing further on the concept of a multi-step processing system using Mathematical Morphology operators X_i , J. et al. proposed [6] an algorithm that first removes the image background by using a low pass filter template (of size 6×6), then performs adaptive threshold segmentation in two consecutive images which produces binary images. Then the first image is subtracted from the second which removes stars and what remains is debris and some extra noise points around where stars were. The next step is the “twice morphologic filtering” where two successive “open” operators are applied, with a logic “AND” operation between the output of the first open operation and the initial binary image. The final step is the application of a Frame Correlative Debris Detection Algorithm to the candidate debris regions, which tries to detect the debris in three consecutive images, by calculating a velocity from a candidate debris in the and the second images, then the looking for the debris in the expected region in the third image based on the calculated velocity. This whole pipeline is also used as a tracking algorithm by also employing Kalman filtering

The same scientific team followed up in 2015 with a new paper [7] that proposes a new multi-step process for detecting debris. The method is based on 5 consecutive images of the sky region of interest. As a first step a maximum projection image (Fig. 5) is calculated where each pixel value is the maximum value of the five images in each pixel position. Then the mean intensity of the five images is calculated and removed from the maximum projection image to remove the stars and noise. Twice morphological filtering is applied to connect different debris candidate regions and eliminate noise points. A Hough Transformation converts all detected streaks to points and thus the start and end points of the streak are detected which calculates the debris region. Finally a Particle Filter based on Bayesian inferences and Monte Carlo simulations is used which locates the debris in the initial frames and is then used to detect the debris in the latter images, reducing the calculation cost.

In [8] another team used a four step process based on Mathematical Morphology. First smear noise is eliminated

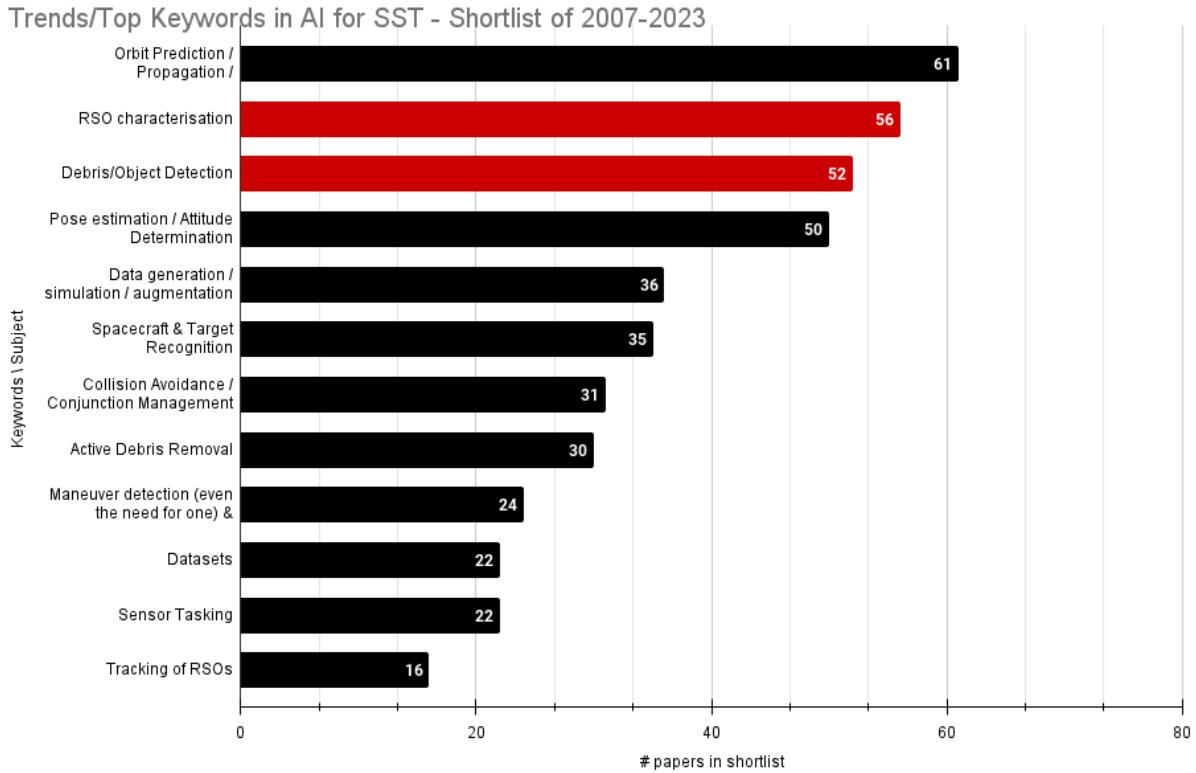


Fig. 3 A ranking of SSA/SST applications by the number of papers found.

using a TopHat operator, then artifacts - byproduct of the first operation - are removed by a second TopHat operation with a different orientation of the structural element (probe). The third step is the calculation of a median frame of consecutive images upon which a threshold (calculated as 3 background fluctuant levels above that of smoothed background) is used to highlight the candidate regions. Because the Mathematical Morphology processes produce a lot of noise which increases the number of false positives a specific binary morphological filter utilizing a-priori properties of object point shapes is used to reduce it. Finally groups with sets of 5 connected pixels are highlighted as candidates. The candidates objects are taken in the initial raw consecutive images, and false positives are further removed by comparing the candidate debris position vs the initial images and removing the ones who are in the neighborhood.

In the following work in [9] Núñez, J. et al. proposed using an image deconvolution technique, and specifically the iterative Richardson–Lucy (R–L) deconvolution method to clean up and enhance the stars and debris in optical images such as other debris detection algorithms can be applied more efficiently.

Feuge-Miller, B. et al [10] used a statistical method for detecting and tracking objects (they reference them as Anthropogenic Space Objects) in telescope star images. The methodology is based on a-contrario analysis, stating that features in the images that cannot be attributed to noise from hardware, atmosphere or the background celestial environment should be (anthropogenic) space objects. The approach involves several steps: hypothesis testing, detection pipeline, astrometry, and orbit determination. The hypothesis testing is based on defining measurable regions and applying significance thresholds to control false positives. The detection pipeline involves evaluating derivative vector fields, removing likely star features, applying track association procedures, and filtering steps. Astrometry involves transforming image-frame coordinates into right ascension and declination angles for potential space objects. Finally orbit determination compares the precision and accuracy of measurements against known state ephemerides. The methods is tested against real data.

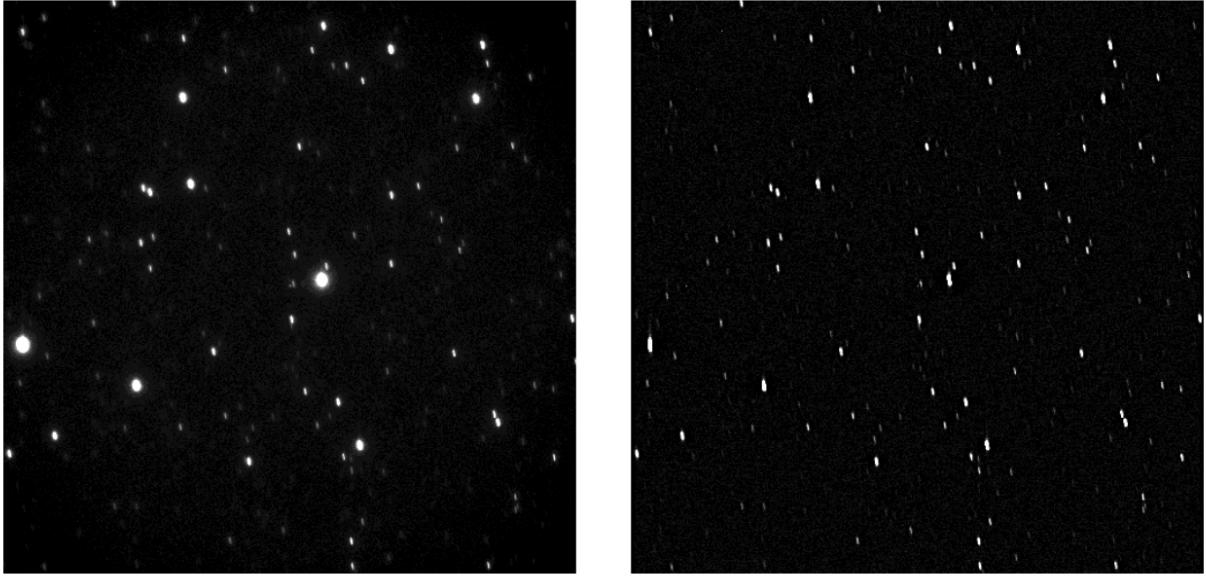


Fig. 4 The effect of the top-hat transformation which removes the smear noise and improves the image quality. Work by Sun, R.-Y. et al [5] Left: The original image. Right: Image after it has been transformed with a structure element with size 7×1 pixels.

B. Machine Learning based methods

Now let's see the machine learning methods used to detect space debris. The papers that have been analyzed were classified in the following categories based on the method used:

- 1) Custom feature generation with additional ML model
- 2) Custom Neural Networks either built from scratch, or with major architecture adaptations
- 3) Existing Neural Network architectures used either as is or with small adaptations
- 4) Dataset papers
- 5) Dataset re-labelling methods

1. Custom feature generation with ML model

The following list of papers contain methods which use a combination of methods for creating features from the raw images before feeding these features in an ML model. This is in contrast to the majority off papers using neural network based models where the feature extraction step is part of the model (especially when using convolutional neural networks).

The first paper that uses an ML method in our list was proposed in 2014 by Huang, T et al [11]. They are using telescope images, which they break up into blocks. Then each block is processed and the following features are produced: a) overall expectation: the mean value of the pixel intensities in the image block, b) overall variance: measures how much the pixel intensities vary from the overall expectation, c) overall skewness: describes the asymmetry of the distribution of pixel intensities in an image block and d) duty ratio: the concept is derived from digital circuits theory and is defined as the ratio of the event duration to the total period of a signal. These features are used to train a Naive Bayes Classifier which is used to classify each block into categories (target, smear, or background). As an extra step a joint decision process is run which re-evaluates these classifications by considering the classifications of adjacent blocks, enhancing the accuracy and reliability of the target detection. The full pipeline can be seen in figure 6.

Xi, J., at al. tried a different approach [12] of using image preprocessing techniques (such as maximum projection) for hot pixel and background removal, but also for star removal (using star detection and segmentation) and space debris candidate region proposal. Then a dedicated neural network called SDdecNet (based on the LeNet5 architecture) has been developed to detect and classify the candidate regions. Finally this SDdecNet model infers every candidate region on whether it's space debris or not.

Du, Y. et al. [13] tried another combined approach of using classical image processing techniques with machine

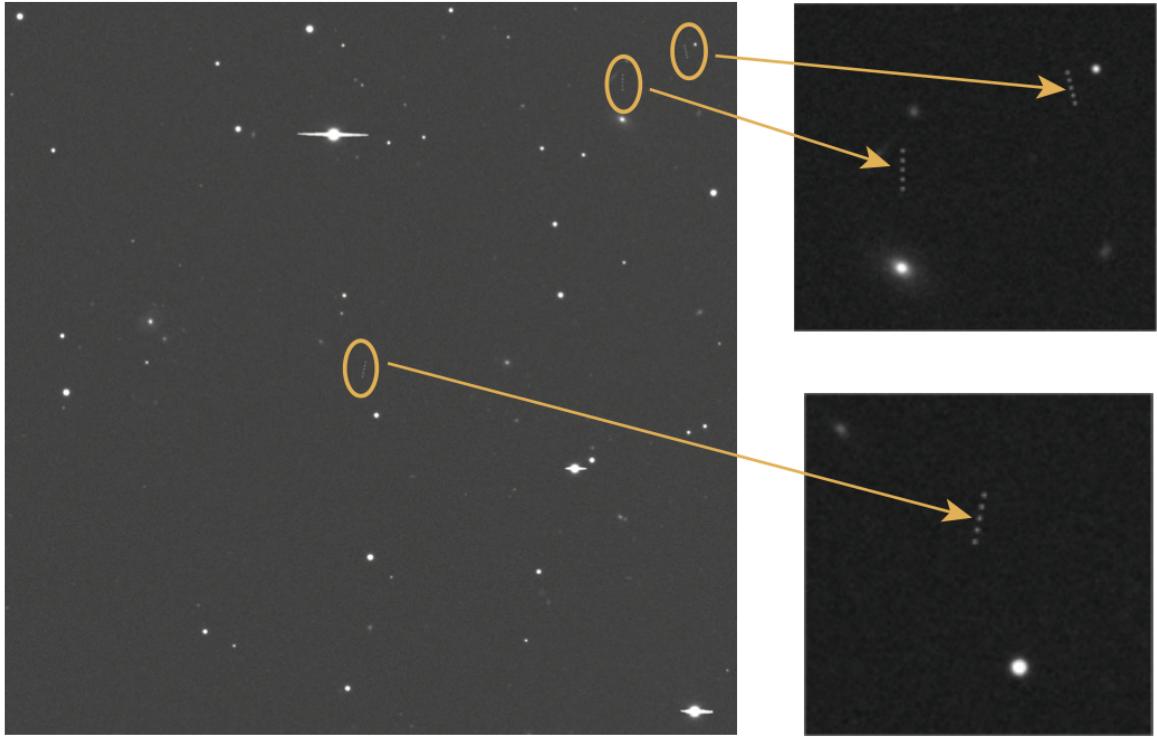


Fig. 5 The maximum projection technique. Each pixel value is the maximum value of the five images in each pixel position. Then the mean intensity of the five images is calculated and removed from the maximum projection image to remove the stars and noise. Work by Xi, J. et al. [7]

learning methods. The background and smear effect are removed with specific thresholds, then in the specific application the stars vs debris (appearing as streaks) are identified using a machine learning model based on Support Vector Machines (SVMs).

Montanaro A. et al [14] proposed a new object detection method that uses a stacking method before the images are processed by a simple neural network with three convolutional layers. Specifically, the stacking method improves the signal over background ratio (SBR) by overlapping many frames shifted by a specific number of pixels depending on the speed and direction of the observed object. The model was trained using simulated data generation by the EUSO Simulation and Analysis Framework (ESAF).

Zhang, X. et al [15] created a model that tries to detect weak debris targets against a background of dense stars with long tails resulting from long exposure times. They are using simulated data and processing them initially with a rectangular fitting technique to aggregate trailing images and reduce noise influence. This fitting is based on the predictable angle and length of the star trails. Once the star tails are aggregated, the next step involves recognizing semi-occluded targets. This is done by analyzing the occluded state of the target and using a fully connected network (FCN) to classify the semi-occluded scenes and extract the semi-occluded images. The final step integrates traditional multi-frame association with the processed single-frame images.

Lin, B. et al [16] created a mixed classic method and CNN based pipeline for small target detection against real star image data. The pipeline starts by preprocessing the images to remove the background by splitting the image into smaller grids. For each grid the histogram of pixel values is clipped using a custom method with the median and mean values. A binarization step is followed to separate the background from the foreground (stars and objects). Then a region proposal search is executed using the foreground segments to create candidate regions around the points of interest with variable sizes for further processing. Finally these candidate regions are input into a custom CNN (called SF-CNN) which performs the final classification. As a last step to remove redundant targets int the same proposal a non-maximum suppression operation is performed.

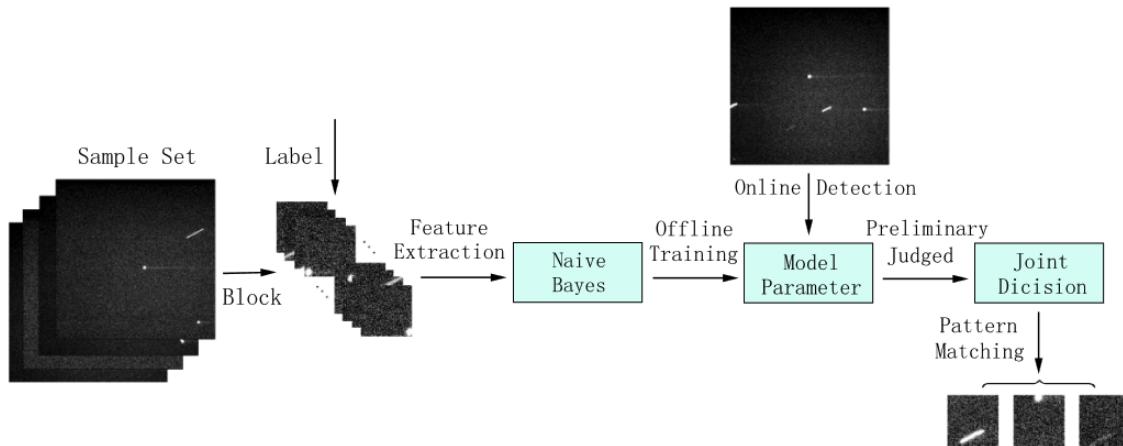


Fig. 6 Flow diagram of the joint decision and Naive Bayes algorithm (JD-NB) for object detection in [11].

2. Custom Neural Networks

The second category includes models that have been built from scratch or that are using an established architecture but is augmented with new modules and components to adapt them to the space object detection task.

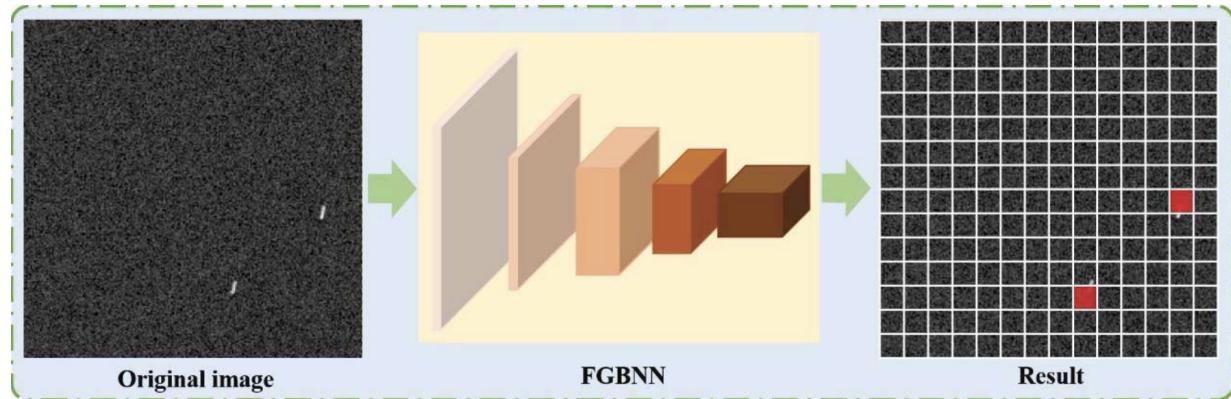


Fig. 7 The pipeline of the FGBNN. The image split into 14×14 grids by the FGBNN and for each grid predict the class probabilities. If the grid with space debris, the predicted value is one. The grids with space debris labeled with red. Work by Xiang, Y. et al. [17].

Xiang, Y. et al. [17] used a custom convolutional neural network architecture (Fig. 7) named FGBNN which processes images by splitting them into a 14×14 pixel grid, then detects whether there's debris in each grid cell. The authors highlight the fast training and inference speed of the model as the main advantage due to the smaller number of parameters compared to similar detection models.

Guo, X. et al [18] created a CNN model for space target segmentation based on an encoder-decoder structure incorporating channel and spatial attention modules (Fig. 8) to enhance the feature fusion and focus on the relevant parts of the image. The method also performs a post processing to calculate the centroid of the detected targets. The model has been trained in a simulated dataset that was created by the authors by adding target blocks to images, with a variety of signal-to-noise ratios (SNR).

Kyselica, D. et al [19] tackled the problem of trying to predict the current position of a space object in an image sequence, given the previous positions. The model is based on an Long Short-Term Memory (LSTM) neural network. It was trained on synthetic data generated by the authors and validated with real telescope images.

Fitzgerald, G. et al [20] created a custom model to detect and track GEO satellites in wide field of view (WFOV) imaging systems. The problem with images produced by these systems is the small footprint of the object in the image

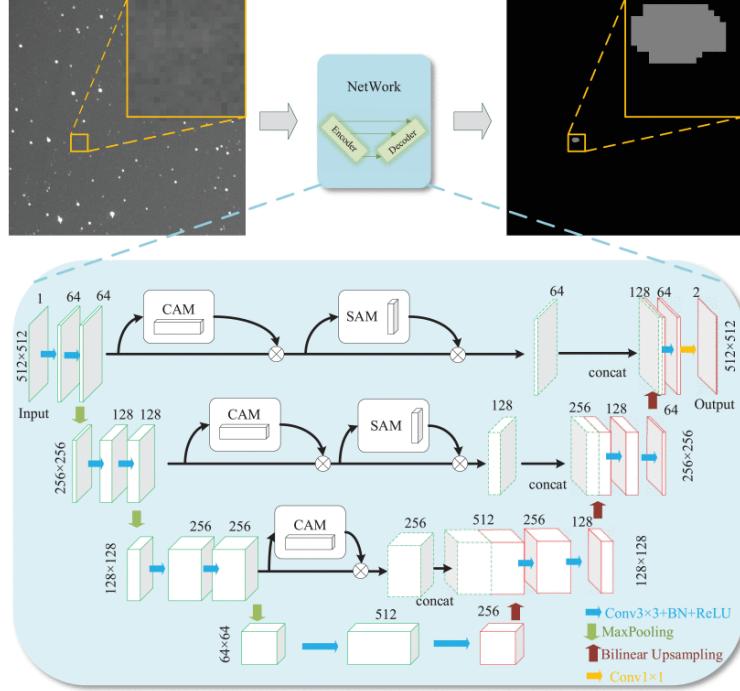


Fig. 8 The proposed encoder-decoder structure architecture with the segmentation networks in [18]. Each cube represents a multi-channel feature map. CAM and SAM are channel attention module and spatial attention module respectively.

as well as the inherent noise of the sensor. The system is composed of two models, the HARP-Net and NOC-Net. The High Area Region Proposal Network (HARP-Net) is the first stage, which uses a 2D convolutional neural network (CNN) with stacked temporal frames. HARP-Net generates a heatmap of Regions of Interest (ROIs) by leveraging the temporal features in the sequential frames. The LOC-Net, which incorporates elements of YOLOv5, uses the heatmap to predict bounding boxes of the RSOs (Fig. 9). It works on the cropped areas from the heatmap that contain high-value ROI pixels.

Ibele, A. et al [21] also created a method detect space objects in wide field of view (WFOV) imaging systems. The authors used a custom CNN based on the ResNet architecture with modifications to focus on the highest-resolution feature map in the feature pyramid, simplify the architecture, and optimize it for faster training and inference.

Tao, J. et al [22] created a custom CNN based model called SDebrisNet (Fig. 10) to detect space debris in telescope video sequences. The model is composed of several modules, a spatial and a temporal feature extraction module, a spatial feature enhancement module and finally a spatial-temporal feature fusion module. The Spatial Feature Extraction module is based on the MobileNetV3 architecture [23] and as the name suggests it extracts spatial features from the image. Then the Spatial Feature Enhancement module includes four spatial feature enhancement networks designed to strengthen the spatial details of small objects. Then after the spatial features have been enhanced the Temporal Feature Extraction module receives the high-level feature maps of each frame, and processes them with a self-attention method to learn spatiotemporal coherence from consecutive frames, outputting temporal salient high-level features. Finally the Spatial-Temporal Feature Fusion module fuses the enhanced spatial features and temporal features, resulting in saliency images for each frame in the video clip. The last step involves the centroid calculation of the detected object by calculating the weighted centroid coordinate of each pixel in the salient region based on its intensity. The model has been trained and test in 45 synthetic and 2 real telescope videos.

Yuan, Y. et al [24] proposed an enhanced version of the YOLOv5 architecture (Fig. 11) to detect small and weak space object in simulated telescope images. The proposed version has the following added modules: a) The Cross-Layer Context Fusion Module (CCFM) enhances the feature information and network identification ability. It operates through multiple parallel branches to learn context information at different scales, b) The Adaptive Weighting Module (AWM) is designed to map small-scale features sequentially to the features of the previous layer, assisting the CCFM

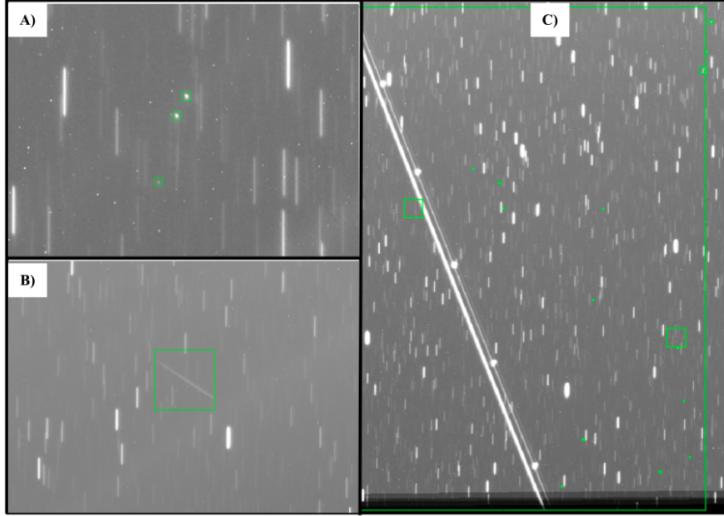


Fig. 9 Various RSOs of different orbital regimes and resulting morphology. A cluster of GEO RSOs are annotated in (A), a low-light MEO RSO is highlighted in (B), and a bright streaking LEO RSO is shown in (C). Taken from work by Fitzgerald, G et al.[20]

in further enhancing the expression of features and c) The Spatial Information Enhancement Module (SIEM) is key in capturing multi-directional spatial environmental information, particularly vital for small object detection where spatial information is easily lost. Finally they are also utilizing a new data augmentation technique (Contrast Mosaic) to enhance data complexity and diversity.

3. Existing Neural Network Architectures

In the third category, the authors of the papers have used the established architectures from computer vision models and either used them as-is or with minimal changes.

In 2019 Varela, L. et al. [25] explored the usage of convolutional neural networks (CNN), which are a type of neural network architecture that has shown great success in image processing tasks. The main advantage of the CNNs lies in the fact that handcrafting features are no longer necessary for creating an algorithm, but rather the training process itself produces these features which are usually detected by the first layers of the network. An even more useful type of CNNs are Region Based CNNs (R-CNNs) which also output the region in the image where the object of interest lies, by iteratively searching all parts of the image and feeding each part in a dedicated CNN that can apply a label to the frame-part. This architecture has the disadvantage of having to check the image versus multiple sizes which significantly increases the performance required. A third architecture is the YOLO architecture, which instead of looking at the image multiple times, instead tries to predict bounding boxes in the areas of the image that seem relevant. By looking at the image only once, it increases the performance of the system. The system proposed uses the v2 of the YOLO architecture which has a number of improvements compared to the first one.

Similarly Temple, D. [26] in 2019 used the v3 of the YOLO architecture for detecting RSOs in a vast network of sensors owned by the company. Additionally a Kalman filter is used to track the RSOs over time in different images.

Another team in [27] also used the YOLO v2 architecture with a dataset from the Pulkovo Observatory to detect objects in two types of images: a)fixed camera – constant background and b) moving camera – changing background.

De Vittori, A. et al [29] created a system that aims to predict and extract the position and shape of space object tracklets from telescope images, focusing on real-time processing with high accuracy. The system was trained on both synthetic images and real images. Synthetic images were created with a new Python application developed in-house. The app uses TLE (Two-Line Element) files of known Low Earth Orbit (LEO) space objects to simulate their transits across a simulated night sky. The output is dual-tone (black and white) images representing the night sky with white tracklets. In addition to the synthetic, real images were also used to test the system. The model at the heart of the system is a neural network based on the U-Net (Fig. 12) architecture [28] which performs the image segmentation to highlight the detected tracklets.

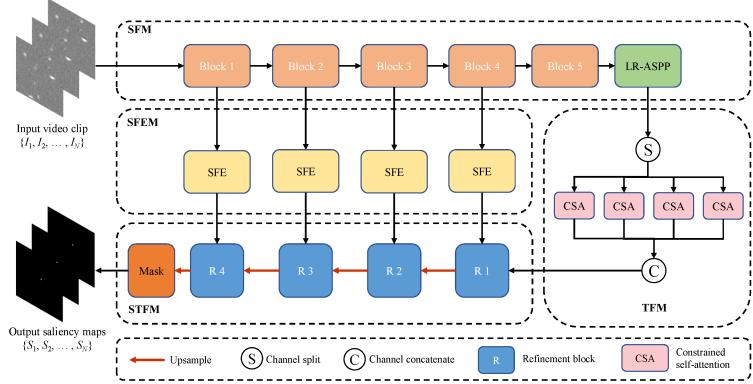


Fig. 10 Detailed illustration of the proposed saliency detection network (SDebrisNet), which consists of the spatial feature extraction module, spatial feature enhancement module, temporal feature extraction module, and saliency prediction module. Work by Tao, J. et al [22]

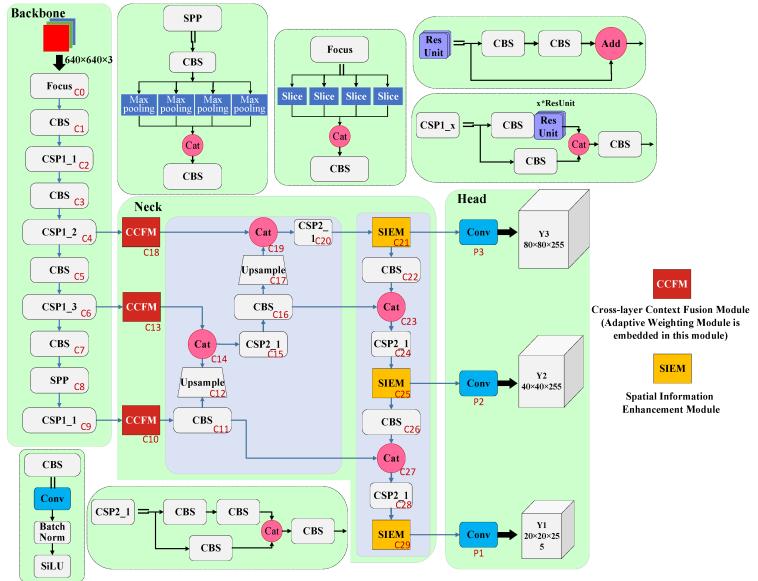


Fig. 11 The framework of the CS-YOLOv5 model for small and weak. Yuan, Y. et al [24]

Mastrofini, M. et al [30] created a new algorithm called Brightest Objects Sky Segmentation (BOSS) designed to focus on the brightest objects in optical sensor images, which can be stars for attitude determination or resident space objects (RSOs) for space surveillance tasks. The dataset was created from three different sources. Real night sky acquisition campaigns, simulated images using the Stellarium software and images obtained from a high fidelity star tracker image simulator which leads to strong model generalization abilities (Fig. 13). The BOSS algorithm is utilizing a U-Net (Fig. 12) based neural network which is trained to output the probability of each pixel in the image being a foreground object (star, RSO etc). Finally a custom clustering algorithm groups the pixels and prioritizes the most significant clusters based on their dimension, energy, and maximum energy value.

Su, S. et al [31] used a custom convolutional neural network inspired by the Faster RCNN architecture to detect dim and small space objects in star images, and a k-means clustering algorithm to locate the centroid of the detected objects. The clustering algorithm involves bilinear interpolation to increase the target's pixel count, segmentation of the target and background, and iterative target region window adjustment to minimize noise-induced errors. The model was trained on simulated data, semi-real data (simulated mixed with real) and fully real data.

Jordan, J. et al [32] created a CNN model based around the MobileNetV2 [33] architecture for efficient object detection. They created a custom synthetic dataset to train the model, and was tested on real and synthetic data. The

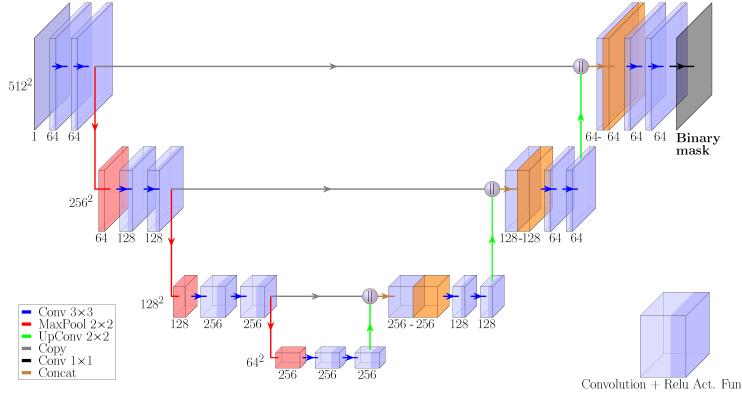


Fig. 12 The U-Net image segmentation architecture. [28]

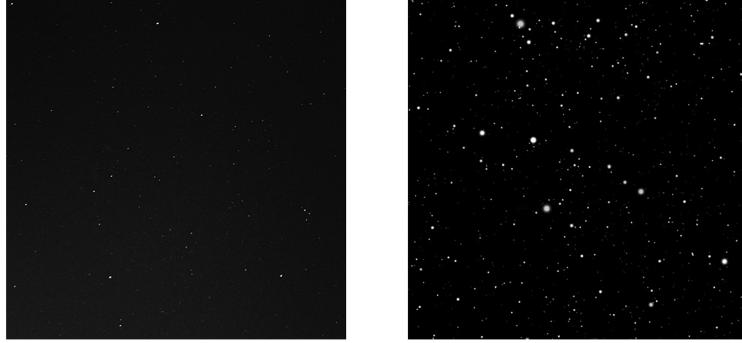


Fig. 13 Real vs synthetic data comparison: real night sky image (on the left) and Stellarium-generated night sky (on the right). Work by Mastrofini, M. et al [30]

focus of the method is to create a model that is efficient to be potentially used in mobile ground telescopes.

Dai, Y. et al [34] created an object detection model based on the YOLOv2 architecture specifically for GEO objects (Fig. 14) which tend to be fainter in optical images. The methods works on image sequences and leverages a post processing step called the CFS technique which utilizes the principle of moving continuity and trajectory consistency of space objects across frames. Specifically space objects are expected to move with uniform speed along a straight-line trajectory. The CFS method utilizes this characteristic to filter and supplement the detection results. The model was trained on data collected from the Kelvins SpotGEO Challenge[35].

4. Dataset Papers

During our search the following paper was found which created new datasets specifically for space object detection. Fletcher J. et al. [36] named the dataset SatNet which includes 104,100 annotated images captured in rate-track mode against GEO targets. The authors mention some of the shortcomings (especially due to human annotation biases such that results in a large distribution of brighter objects, objects that are shown better in a dimmer background, or objects that occur more often) and have created a new simulation software and with it a new dataset called SatSim which includes 50.000 images from each of 20 simulated sensors. The authors also trained a neural network based on the YOLO v3 model architecture and compared it with a popular open source software called SExtractor [37] that also supports object detection in sky images.

Chen, Z. et al [38] created a new framework called simulation-augmented benchmarking framework for RSO detection (SAB-RSOD). They used a simulation system to create data (stars and streaks of RSO) using real sensor parameters. Then the dataset was augmented with real work images from a star tracker (Swarm star tracker). The dataset is used to train a YOLO and a Faster RCNN model as well as their own model based around the U-Net (Fig. 12) architecture.

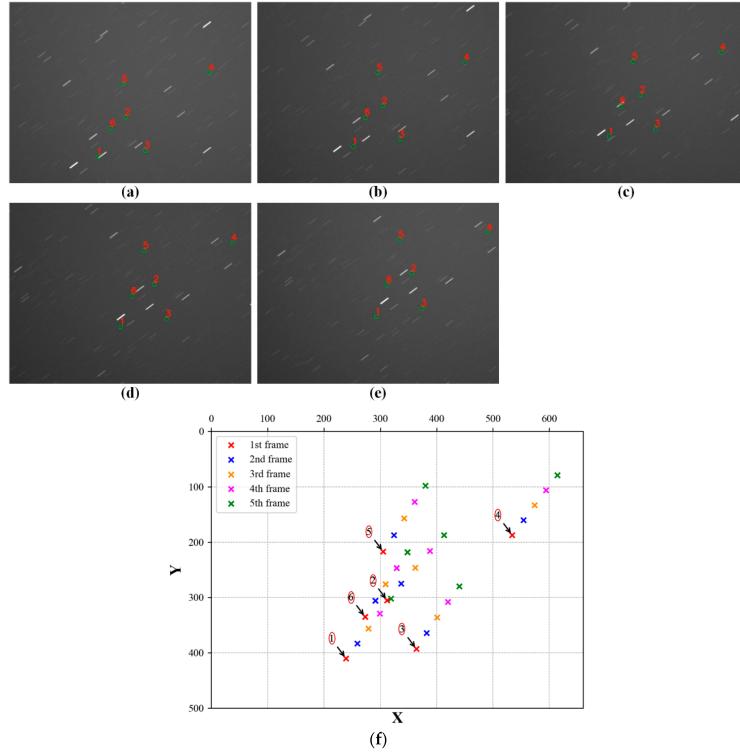


Fig. 14 An example of image sequence of GEO objects: (a–e) five frame images within a sequence that contain six objects; (f) the source positions are shown by the cross symbols in different colors. Dataset from the work by Dai, Y. et al [34]

5. Relabelling Methods

Finally the following two papers were found which include innovative methods to tackle the problem of unlabeled or noisy data. Data annotation requires time, money and expertise which makes it hard and expensive to create high quality datasets. With the following work the authors tried to find ways to overcome these obstacles.

Dimitrescu, F. et al [39] created a novel system (Fig. 15) which recursively re annotates incomplete astronomical data (i.e. not all visible objects are annotated). They used a two model system with a first object detection module based on the Faster R-CNN [40] architecture with a ResNet-18 [41] backbone and the second is a custom CNN classifier. The method works by parsing the image through the detection phase for candidate region generation, which are then fed into the classifier. Objects with a high classification confidence are added into the training dataset to enhance the model performance.

Li, H. et al [42] created a method to relabel noisy data in space debris detection datasets (such as stars miss-labeled as debris). The method begins by denoising and smoothing the background (removing hot pixels, flicker noise, and evening out uneven background). Then two identical CNNs are trained on the same dataset. They make their predictions on the same samples. Then each network's predictions are used as extra information to correct the noisy labels of the other network, thus refining it during training. The method was tested on a mix of real and simulated data.

V. RSO characterization

Resident Space Object (RSO) characterization is the process of obtaining detailed information about the physical properties, functional capabilities, and potential behaviors of objects orbiting Earth, such as satellites and space debris. This process helps maintain the safety and operational efficiency of space assets by determining an object's size, shape, mass, orientation, reflectivity, and material composition. RSO characterization may also assess the operational status of objects, such as differentiating between active satellites, non-functional satellites, or space debris.

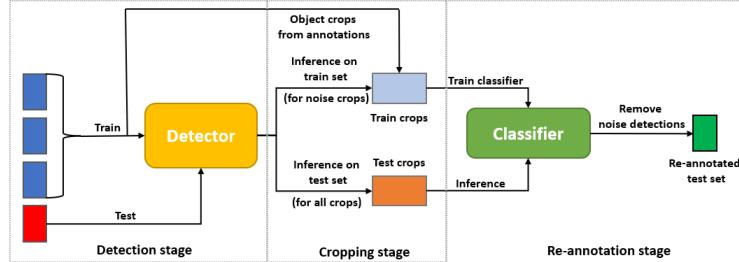


Fig. 15 Overview of the re-annotation model by Dimitrescu, F. et al [39] captioned from their work: "It consists of 3 stages: the detection stage, in which the object detection model is trained and the objects of interest are detected on the validation set; the cropping stage, in which crops for the classifier are extracted based on the detections from the previous stage; the re-annotations stage in which the classifier is trained on the extracted crops of the objects of interest from the train set, as well as noise, and then used to detect which objects found in the detection stage are valid. In the last stage, the valid detected objects are also saved to be used in the next re-annotation step."

A. Classical Image Processing Methods

In this part, similar to the classical image processing methods of the RSO detection application, will present general image processing methods used in the RSO characterization which do not user machine learning based techniques.

Starting in 2011, Chaudhary, A. et al [43] created a technique for characterizing RSOs called RSO fingerprinting using time series of brightness data in multiple wavelengths. It represents an RSO (specifically working with satellites) as a model being composed of two separate facets, the body and the solar panels. Then it uses the Lx technique, which works on the hyperspectral brightness data by trying to match features and extra information about the RSO in more detail in each level. For L0 it tries to match features such overall size, type, gross brightness, and the basic shape and position of principal specular glints (bright spots due to reflection). L1 features comprise the geometric shape of the signature brightness, color indices, and variations due to the angle of sunlight (sub-solar angle). The Level 2 level distinguishes the light contributions of the solar panel and the main body of the RSO. It looks at the sloping regions or bifurcations in the signature brightness and color, helping to separate and identify the contributions of different components of the RSO. Finally Level 3 tracking the temporal evolution of the fractional abundance of the solar panel and the body of the RSO. This level provides insights into the mechanical stability and attitude of the RSO.

In 2014, Linares, R. et al [44] used light curves and angle data (which are time series produced by optical telescope observations, the light curves representing the time series of the brightness and angles representing the time series of topocentric azimuth and elevation of the RSO) to identify the most probable shape of an unknown resident space object along with its associated rotational and translational states. They used two techniques, a multiple-model adaptive estimation (MMAE) and an Unscented Kalman Filter (UKF). MMAE was used to estimate the most probable shape of the space object (from a bank of possible shapes, but can choose the best model that fits the actual shape when the true shape is not in the bank). The Unscented Kalman Filter was used to estimate the rotational and translational states based on fusing angles and light curve data, along with their associated models. Simulated data were used to verify the method.

B. Machine Learning Based Methods

Now let's proceed with analysing the papers using ML based techniques. Similarly to the RSO detection section we have classified the papers into the following categories this time based on the type of characterization performed:

- 1) Behaviour and intent classification
- 2) Shape classification
- 3) Spin/state/attitude/pose estimation and classification
- 4) Space object type classification / target recognition
- 5) Material identification
- 6) Future lightcurve prediction

1. Behaviour / Intent Classification

The first characterization application is trying to predict the behaviour or event of the target space object. This involves the classification of the maneuvers that the space object performs to classes such as expected space keeping ones, or anomalous (which may indicate a malfunction or malicious intent).

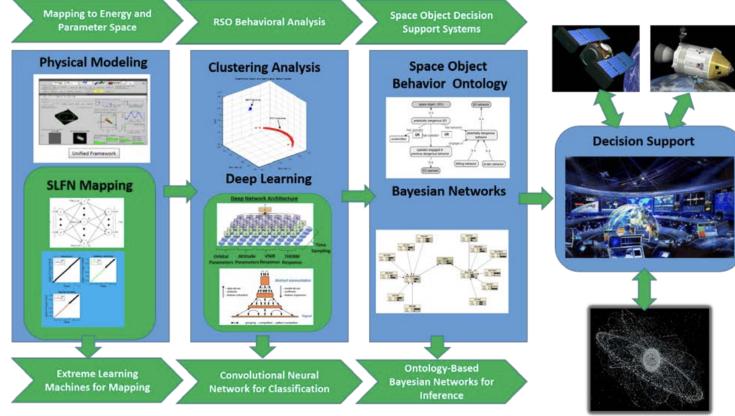


Fig. 16 The End-to-End Machine Learning and Ontology-based Bayesian Networks for Space Object Decision Support System developed by Furfaro, R. et al [45] for behavior classification.

The first paper in 2016 by Furfaro, R. et al [45] aims to predict and understand the behavior of Resident Space Objects (RSOs) with a goal of developing a decision support system (Fig. 16) for space domain awareness. Specifically, by using Extreme Learning Machines (ELM) and convolutional neural networks (CNN), the aim is to infer critical parameters of RSOs from sensor measurements. These parameters include aspects like the object's energy state, momentum, and other physical characteristics. Then the characteristics are used for clustering RSOs in the feature space, which helps in identifying different types of behaviors exhibited by these space objects, such as standard operational patterns, anomalous behaviors, or patterns indicative of potential threats or malfunctions. By integrating Space Object Behavioral Ontologies (SOBO) with Bayesian Networks (BN), the study provides a probabilistic framework for predicting the behavior of RSOs. This prediction is based not only on the learned patterns from machine learning models but also incorporates expert knowledge and domain-specific information from the ontologies. The study used simulated data generated from physical models simulating the behavior of RSOs, including various energy and state parameters.

2. Shape Characterization

The second characterization application one of the most popular ones (in number of papers involving it) is the space object shape classification. Defining the shape of an RSO can be useful for SST applications such as orbit determination (because knowing the shape of the object can aid in calculating the effect of the solar wind or the atmospheric drag).

In 2019 Linares, R. et al [46] presented a method for characterizing the shape of a space object using light curves. They generated simulated light curves using a physically-based forward model. Then they used the data to train a convolutional neural network to directly predict the object shape. The study focused around 4 classes of rocket bodies such as a) A cylinder with round top, b) a simple cylinder, c) a shape representative of an Atlas Upper Stage; and d) a shape representative of a Falcon 9 Upper Stage (Fig. 17). Additionally the team also trained a variational - the variational version of the autoencoder tries to learn a probability distribution (mean and variance) for each latent attribute - autoencoder on the data in order to understand and visualize the latent distribution of light curve data (Fig. 18).

Yao, L. et al [47] used lightcurve data and a 1-D CNN to characterize the space debris basic shape (specular cylinders, diffuse cylinders, box-wing satellites, and cube satellites are used as primary shapes). The model was trained on simulated data (generated using the Cook-Torrance Bidirectional Reflectance Distribution Function model[48], considering various shapes and materials for space debris) and compared against other ML models such as k-nearest-neighbors, Support Vector Machines, Random Forests and Decision Trees all of which it outperformed in accuracy. Additionally another dataset was created from observations extracted from the Mini-Mega TORTORA (MMT) database [49] with two shapes (rocket body vs defunct satellite) training both the custom CNN model and an Random Forest model.

van Rooij, S. et al. [50] used radar measurements (specifically Inverse Synthetic Aperture Radar (ISAR) images)

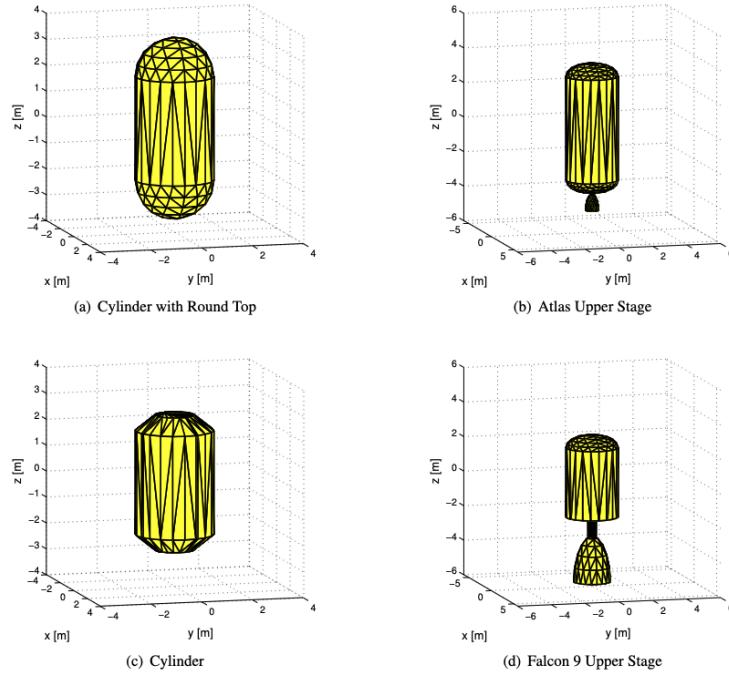


Fig. 17 Models for Rocket Bodies used in the shape characterization work by Linares, R. et al [46]

to determine the size and shape of satellite solar panels and buses, using a custom CNN model based on the VGG19 architecture [51]. The study used simulated radar images for training. The simulation covers five different satellites (Envisat, Grace, ICESat, Landsat7, and Landsat8) with various geometries, including different azimuth and elevation angles, and noise levels. The model works by segmenting the image into the different components (e.g. solar panel or bus) and thus their shape and size can be estimated. The results of the model can be seen in figure 19.

Balachandran, K. et al [52] used an LSTM based neural network model in conjunction with a hidden Markov model to classify the shape (ellipsoidal asteroid, rocket upper-stage, CubeSat, torus, probe, and box-wing satellite) and spin rate (tumbling vs stabilized) of LEO RSOs using lightcurve data. They used simulated data to train and validate their model. They run the lightcurve data against statistical tests for periodicity (Fisher's Exact Test for Periodicity) and aliasing (Augmented Dickey Fuller test) and implemented feature selection via wavelet decomposition.

3. Attitude/Pose Estimation and Classification

The task of attitude or pose estimation is also useful for use in SST applications such as orbit determination (because knowing the RSO attitude can help calculate the solar wind and atmospheric drag influence in the orbit evolution) but also to future missions aimed at active debris removal.

Phan, D. et al in 2019 [53] developed a pipeline to characterize GEO objects into two categories: a) three-axis stabilized GEOs (TAGs) (Fig. 20) and b) tumbling objects (TOs) (Fig. 21), further sub-classifying TOs into rocket bodies or tumbling satellites. Also for TAGs, a sub-classification into signature types is implemented, aiding in assessing physical structures on the satellites. Using lightcurve data collected from partners the pipeline developed uses a number of statistical and ML methods to achieve the desired classification, namely: 1) a random forest (RF) classifier (utilizes statistical features derived from light curves for initial classification) 2) periodicity test (assesses the statistical significance of each period found in the light curve, distinguishing between TAGs and TOs) 3) aliasing test (evaluates whether residuals from a spline fit to the light curve are normally distributed, helping detect aliasing due to under-sampling) 4) quality of fit test (compares the coefficient of determination - R^2 - of the light curve's spline fit against empirical distributions, aiding in stability mode assessment) 5) hidden Markov model (HMM): Integrates the assessments from the above tests into a dynamic Bayesian network for the final classification.

Zhang, H. et al [54] used Gaussian Process Regression on simulated satellite images (BUAA-SID dataset [55]) to

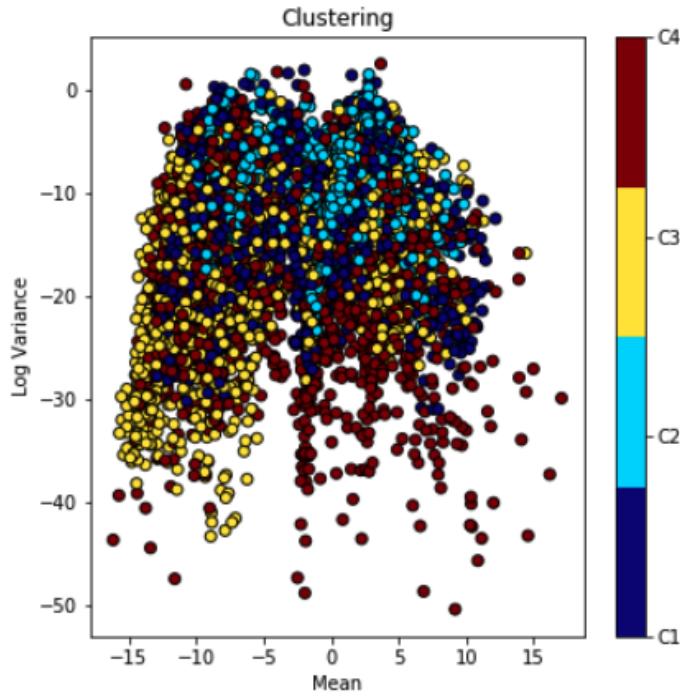


Fig. 18 VAE-learned light curves in the latent variable space work by Linares, R. et al [46]

perform a) satellite recognition (the type of satellite observed) b) satellite pose/attitude estimation. Specifically for pose estimation a manifold constraint is introduced. It uses a manifold representation (e.g., normalized n-sphere in a $n+1$ -dimensional space) to represent the degrees of freedom in pose variation, so that the model regression output is constrained in the possible satellite poses. Finally the model also provides an uncertainty measurement, by giving a confidence score with each prediction.

Afshar, R. et al [56] also used the (BUAA-SID dataset [55]) to create models for both target recognition and pose estimation. Their innovation lies in the use of transfer learning and data augmentation techniques, enhancing the classification accuracy under various space conditions, including noise and different illumination. A two-stage CNN architecture is used. The first stage involves a pre-trained network (Inception CNN [57]) on the ImageNet[58] dataset for feature extraction, a transfer learning technique. The second stage trains the network for classification and regression (for the tasks of satellite recognition and pose estimation respectively) on images of the dataset (which has been augmented by applying transformations like rotating, shifting, re-scaling, and zooming) to enhance classification accuracy making it more robust against different noise and lightning conditions.

Badura, G. [59] in 2020 applied a CNN for the task of inferring an RSO's attitude from lightcurves. They tried to improve upon the issue of bidirectional reflectance distribution function (BRDF) mixing, sensor noise, and atmospheric turbulence, by generating a synthetic dataset that realistically incorporates BRDF signatures and environmental parameters. They generated the dataset by using the Beard-Maxwell model of reflectance [60], but adapted it for materials like silicon solar panels, glossy paint, and aluminum. Finally they used a CNN model based on the LeNet [61] architecture to estimate four different maneuvers: a) tumbling, b) accelerating in rotational rate, c) stabilizing, and d) inactive.

Paulete, C. et al [62] used data from radar measurements called Radar Cross Section (RCS). It is analogous to a lightcurve in the sense that it represents the radar reflectivity of an object (from the point of view of the radar installation) over a time span. This dataset was used to train two ML models to perform a) Attitude stability classification (in four classes, Earth pointing, inertia pointing, geodetic and tumbling) and b) Direct Object Identification (44 classes representing various satellites and a class for unknown object). Additionally the azimuth and elevation in the ITRF and GCRF reference frames is provided for each radar measurement. The model is a neural network using a Long Short

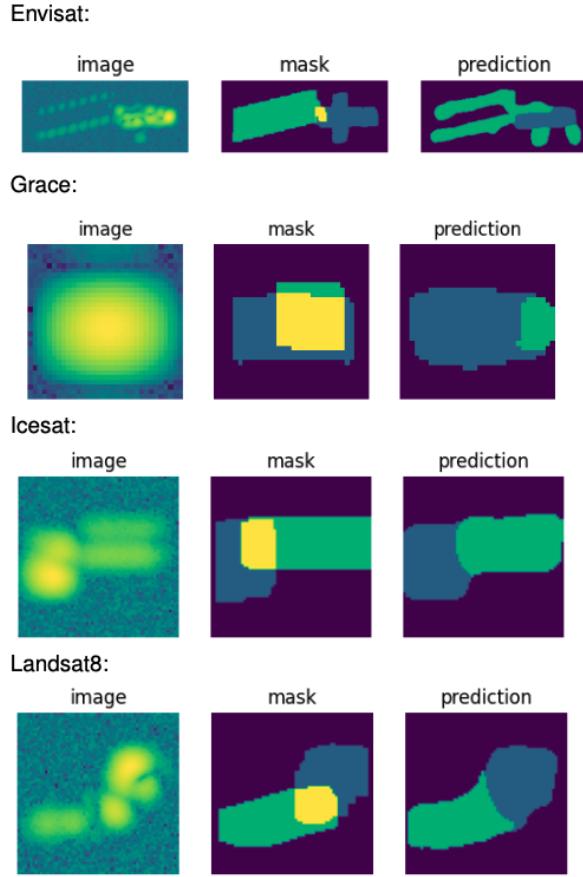


Fig. 19 Captioned from work by van Rooij, S. et al. [50] on shape characterization. "Examples of normalized ISAR images (not same scale, small satellites are zoomed), ground truth masks and model predictions for Envisat, Grace, IceSAT and Landsat8 satellites, using the Leave-one-out approach. In these images the purple pixels indicate the background class, the blue pixels indicate the bus, the green pixels indicate the solar panels and the yellow pixels indicate the overlap of bus and solar."

Term Memory (LSTM) layer (which can work with timeseries data) which is connected to a feedforward layer.

McNally, K. et al [63] created a lightcurve simulator to produce generated lightcurve data due to the lack of open source datasets. Then they also used real lightcurves to both verify the validity of the generator but also test the model trained on simulated data. They predicted two different variables: a) attitude (2 classes stable or spinning satellite but additionally where the satellite antenna is pointing at) and b) satellite (6 different satellites are used for identification). The model was a feedforward neural network which is using the following features generated from the raw lightcurves: a) the maximum and minimum values of the apparent magnitude b) the coefficients of a third order polynomial regression fitting the lightcurve c) the first two most representative frequency amplitudes of the fourier transform of the lightcurve.

Qashoa, R. et al [64] worked to classify LEO RSOs by object type and spin rate (stable satellite, tumbling satellite, rocket body) using light curves. Using a dataset of real light curves, features were extract using a Wavelet Scattering Transformation (WST), which were then used as input to an SVM and an LSTM-based model to compared the accuracy of the two models. The authors also worked on visualizing the features from the WST coefficients in a 3D space with an example shown in figure 22.

Also the work by Balachandran, K. et al [52] which was described in the Shape Characterization subsection also includes the task of spin rate (tumbling vs stabilized) classification.

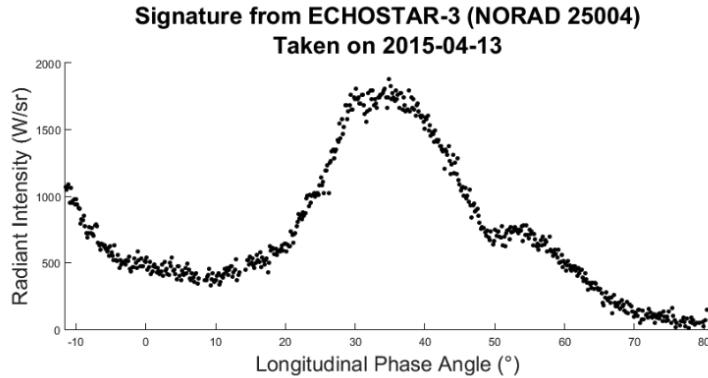


Fig. 20 Example of a typical signature from a stable TAG satellite, work by Phan, D. et al in 2019 [53].

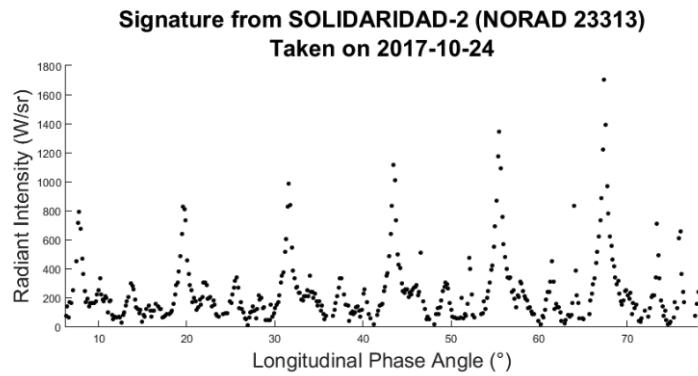


Fig. 21 Example of a sinusoidal signature from a tumbling TAG satellite, work by Phan, D. et al in 2019 [53].

4. Space Object Type Classification / Target Recognition

The space object type or target recognition is one of the most basic tasks in characterizing RSO as it tries to match unknown observed objects to a list of known targets either in general categories such as debris vs satellite, or more detailed by trying to detect the specific type of satellite or satellite component (e.g. satellite body, solar panel etc).

Chaudhary, S. et. al [65] used the SPARK dataset [66] (which contains 150,000 RGB images of 10 spacecraft and 1 debris class with bounding boxes highlighting them, along with the corresponding depth images) to create a new classification pipeline. In their work first the depth images are de-noised using the morphological opening operation. Then using the de-noised depth image a mask is created highlighting the object of interest. Then the mask is applied to the RGB image, which helps keep only the object of interest. Finally this masked image is fed into a neural network model which is using the EfficientNet B4 architecture [67] which performs the space object classification. The result of each processing step can be seen in figure 23.

Krantz, H. et al [68] created a model to classify the type (three types of Starlink satellites and one type of OneWeb satellites, 4 classes in total) of LEO satellites based on their photometric signatures. They use the concept of effective albedo of a satellite (a theoretic construct calculated by comparing the observed satellite to an ideal uniform diffuse sphere) as input feature to a gradient boosted decision trees classifier along with the elevation and azimuth angles. The images are collected with an all sky camera performing photometric survey.

Walker, L. et al [69] generated hyperspectral data with the goal of predicting the proportions of materials in satellites and infer the presence of large components (like solar panels and antennas), and finally perform satellite classification. The dataset was generated using physics and sensors simulations of 3D models of satellites, their orbits and rotation state vectors. An artificial neural network (ANN) was created to predict the fraction of light due to each material at different time points, using simulated data. The output is then converted into statistical features (mean, standard deviation, correlation coefficients, etc.) to produce new features. These features are used in a gradient boosted decision

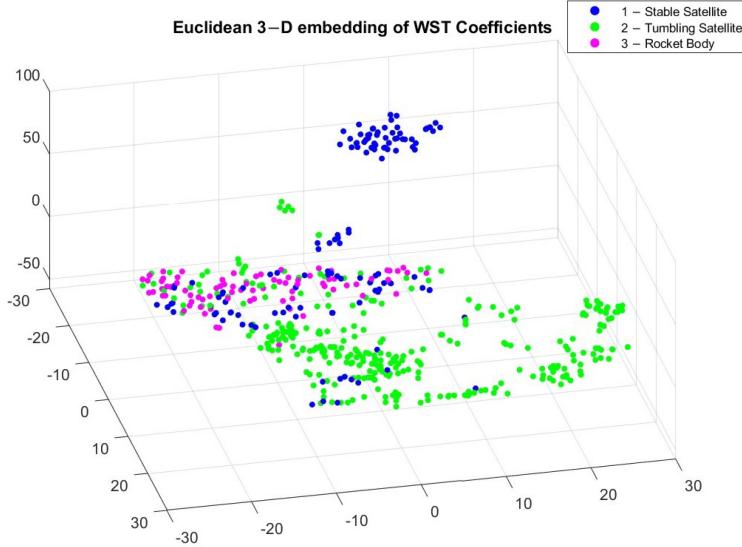


Fig. 22 Example of 3D embedding of Wavelet Scattering Transformation coefficients using Euclidean distance in work by Qashoa, R. et al [64].

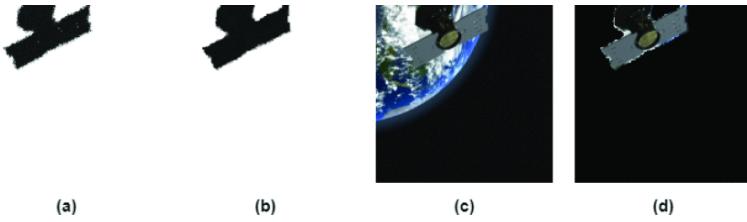


Fig. 23 The processing steps for the space object characterization work by Chaudhary, S. et. al [65]: (a) Depth Image (b) Depth image after applying opening operation (c) RGB image (d) Masked image.

tree model (XGBoost) which predicts the presence of large satellite components (solar arrays, antennas, etc.). Finally a k-Nearest Neighbours (kNN) algorithm performs the final classification into different satellite types (GNSS, Earth Observation, Communications, Rocket Bodies and CubeSats).

Additionally the works by Zhang, H. et al [54], Afshar, R. et al [56], Paulete, C. et al [62], McNally, K. et al [63] and Qashoa, R. et al [64] which were already described in the Attitude/Pose Estimation subsection have also contribution to the type/target classification task.

5. Material identification

The material identification task is used for extracting RSO information that can help in space target recognition. To this end Li, N. et al [70] created a new pipeline by combining a Graph Neural Network (GNN) with a convolutional neural network (CNN) on hyperspectral images. As a first step a superpixel segmentation is performed to group pixels with similar characteristics. Then global structure topological graphs are generated from these segmented superpixels. These graphs augment the original hyperspectral images and the combined data are fed into the joint GNN CNN model. The model outputs the probability for each pixel and each type of material. The architecture of the system can be seen in figure 24, while a visualization of the output generated in figure 25.

6. Future lightcurve prediction

Future lightcurve prediction was one of the niche applications found in the review, as the prediction of the future lightcurve value of an RSO could aid in other characterization tasks which use lightcurve data. Durpee, W. et al [71] used two neural network based models (a Feed Forward and an LSTM based) with lightcurve data trying to predict the future lightcurve, meaning the future magnitude one would measure while observing the object. They performed feature

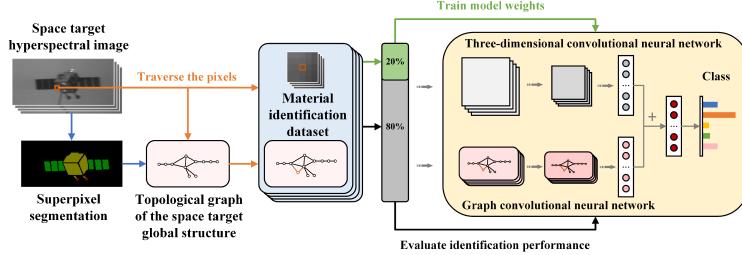


Fig. 24 The architecture of the system designed by work by Li, N. et al [70] for the task of material identification.

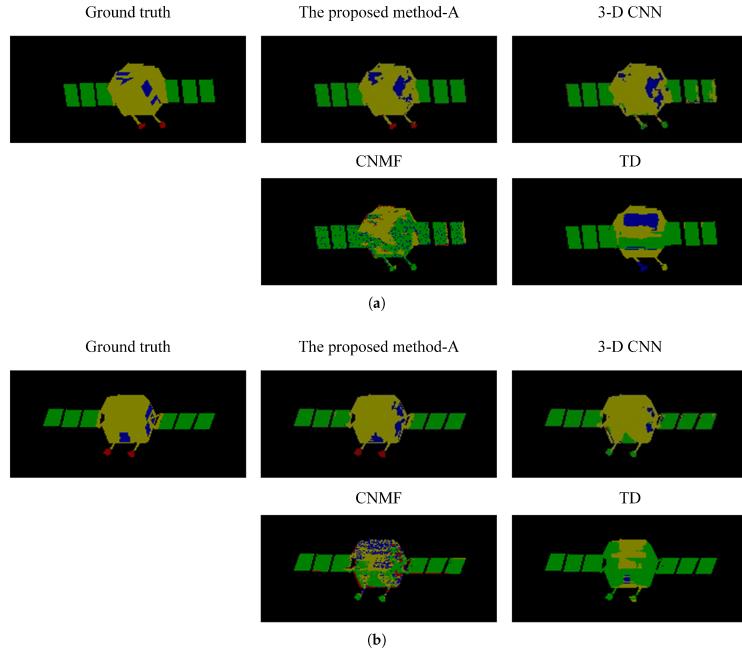


Fig. 25 Examples of material identification results, work by Li, N. et al [70]

engineering to handle missing values in the data, created rolling window samples and created new features to handle the irregular time steps by tracking the difference between measurement values.

VI. Data

Finally we will perform a small analysis on the types of data used in each application. We have summarized the data in two tables, table 1 for RSO detection and table 2 for RSO characterization.

Looking at the summarized data a few interesting points stand out. The first is the fragmented landscape of datasets. Most work studied is being done in datasets created by the authors, either from real collected data (13 papers for RSO detection, 7 for RSO characterization) or data generated by the authors to fit the demands of their work (9 papers for RSO detection, 9 for RSO characterization). Researchers are also using hybrid methods by incorporating real data with simulated ones or using a real data as an extra validation set to measure the transferability of their methods.

With regards to the data type used, for RSO detection optical telescope images are the only type studied but that can also be attributed to our literature search criteria. For RSO characterization the type used depends more on the task. Optical, hyperspectral, lightcurve and angles (declination and right ascension coordinates) have all been used in different tasks showcasing the different approaches followed by various teams. The majority of the data types are (Optical, hyperspectral) or can be converted (lightcurve) to images so as to be exploited by Convolutional and other Neural Network architectures.

VII. Conclusion

With this paper we aimed to provide a thorough survey on the use of Machine Learning and Deep Learning methods for the applications of RSO detection and characterization.

It has been shown that the field of optical telescope image processing has been served by classical image processing techniques up to a few years ago when computer vision models based on artificial neural networks have started to dominate the field. Researchers are using established neural network architectures (such as YOLO, Faster R-CNN and U-Net) either as-is or by adapting them for efficiency or augmenting them with additional modules to better perform on the RSO detection application. With regards to the datasets used, the lack of standardised big datasets seems to be contributing to a fragmented landscape where researchers are using either smaller amounts of real data, or larger simulated datasets, with some also exploring the fusion of the two.

For the RSO characterization application there exists a number of different tasks where research is being undertaken. Attitude/pose estimation and classification, shape characterization, material identification and future lightcurve prediction are some of the main ones that were retrieved from the literature. A sample of the papers that have been published for these tasks have been analyzed showing a strong emphasis on the use of machine learning based techniques, and specifically deep learning ones, with convolutional neural networks being the most common method. With regards to the datasets, similarly to the RSO detection application most of the work is being performed in individual ones, emphasize the need for creating extensive, open datasets to serve as benchmarks, thereby enhancing characterization methods.

Acknowledgments

The work presented is co funded by the European Office of the United States Air Force through grant FA8655-23-1-7218.

References

- [1] Adams, D., *The hitchhiker's guide to the galaxy*, Hitchhiker's Guide to the Galaxy, Random House, New York, NY, 2007.
- [2] Stövèken, E., and Schildknecht, T., *Algorithms for the Optical Detection of Space Debris Objects*, 2005.
- [3] Li, K., Zhang, Q., Guo, Z., Yuan, J., Zhao, C., and Xu, K., “Space Debris Detection Algorithm Based on Mathematical Morphology,” *2009 International Conference on Artificial Intelligence and Computational Intelligence*, IEEE, 2009. <https://doi.org/10.1109/aici.2009.350>, URL <https://doi.org/10.1109/aici.2009.350>.
- [4] Vicent, L., and Soille, P., “Watersheds in digital spaces: An efficient algorithm based on immersion simulations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, pp. 583–593.
- [5] Sun, R.-Y., and Zhao, C.-Y., “A new source extraction algorithm for optical space debris observation,” *Research in Astronomy and Astrophysics*, Vol. 13, No. 5, 2013, pp. 604–614. <https://doi.org/10.1088/1674-4527/13/5/011>, URL <https://doi.org/10.1088/1674-4527/13/5/011>.
- [6] Xi, J., “A Frame Correlation and Kalman Filtering Algorithm for Space Debris Detection,” *Journal of Information and Computational Science*, Vol. 11, No. 15, 2014, pp. 5317–5325. <https://doi.org/10.12733/jics20104688>, URL <https://doi.org/10.12733/jics20104688>.
- [7] Xi, J., “A Maximum Projection and Particle Filtering Algorithm for Space Debris Detection,” *Journal of Information and Computational Science*, Vol. 12, No. 1, 2015, pp. 161–169. <https://doi.org/10.12733/jics20105176>, URL <https://doi.org/10.12733/jics20105176>.
- [8] yu Sun, R., wei Zhan, J., yin Zhao, C., and xiang Zhang, X., “Algorithms and applications for detecting faint space debris in GEO,” *Acta Astronautica*, Vol. 110, 2015, pp. 9–17. <https://doi.org/10.1016/j.actaastro.2015.01.001>, URL <https://doi.org/10.1016/j.actaastro.2015.01.001>.
- [9] Núñez, J., Núñez, A., Montojo, F. J., and Condominas, M., “Improving space debris detection in GEO ring using image deconvolution,” *Advances in Space Research*, Vol. 56, No. 2, 2015, pp. 218–228. <https://doi.org/10.1016/j.asr.2015.04.006>, URL <https://doi.org/10.1016/j.asr.2015.04.006>.
- [10] Feuge-Miller, B. G., Jah, M. K., Karra, A. T., Iyer, S., and Kucharski, D., “A-contrario detection and tracking from optical telescope data,” *Acta Astronautica*, Vol. 210, 2023, pp. 129–140.
- [11] Huang, T., Li, Z., Zhou, Y., Xiong, Y., and Zhang, H., “Joint decision and Naive Bayes learning for detection of space multi-target,” Jul. 2014. <https://doi.org/10.1007/s10043-014-0067-0>, URL <http://dx.doi.org/10.1007/s10043-014-0067-0>.

- [12] Xi, J., Xiang, Y., Ersoy, O. K., Cong, M., Wei, X., and Gu, J., “Space Debris Detection Using Feature Learning of Candidate Regions in Optical Image Sequences,” *IEEE Access*, Vol. 8, 2020, pp. 150864–150877. <https://doi.org/10.1109/access.2020.3016761>, URL <https://doi.org/10.1109/access.2020.3016761>.
- [13] Du, Y., Wen, D., Liu, G., and Qiu, S., “Small Moving Target Recognition in Star Image with TRM,” *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 35, No. 02, 2020, p. 2150004. <https://doi.org/10.1142/s021800142150004x>, URL <https://doi.org/10.1142/s021800142150004x>.
- [14] Montanaro, A., Ebisuzaki, T., and Bertaina, M., “Stack-CNN algorithm: A new approach for the detection of space objects,” *Journal of Space Safety Engineering*, Vol. 9, No. 1, 2022, pp. 72–82.
- [15] Zhang, X., Liu, Y., Duan, H., and Zhang, E., “Weak Spatial Target Extraction Based on Small-Field Optical System,” 2023.
- [16] Lin, B., Wang, J., Wang, H., Zhong, L., Yang, X., and Zhang, X., “Small Space Target Detection Based on a Convolutional Neural Network and Guidance Information,” *Aerospace*, Vol. 10, No. 5, 2023, p. 426.
- [17] Xiang, Y., Xi, J., Cong, M., Yang, Y., Ren, C., and Han, L., “Space debris detection with fast grid-based learning,” *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, IEEE, 2020. <https://doi.org/10.1109/iicspi51290.2020.9332372>, URL <https://doi.org/10.1109/iicspi51290.2020.9332372>.
- [18] Guo, X., Chen, T., Liu, J., Liu, Y., and An, Q., “Dim Space Target Detection via Convolutional Neural Network in Single Optical Image,” *IEEE Access*, Vol. 10, 2022, pp. 52306–52318.
- [19] Kyselica, D., Krajčovič, S., Šilha, J., and Ďuríkovič, R., “Advanced Algorithms for Segmentation of Space Debris Astronomical Images,” *2022 26th International Conference Information Visualisation (IV)*, IEEE, 2022, pp. 336–342.
- [20] Fitzgerald, G., Liu, R., and Asari, V., “Geosynchronous satellite detection and tracking with WFOV camera arrays using spatiotemporal neural networks (GEO-SPANN),” *Pattern Recognition and Tracking XXXIII*, Vol. 12101, SPIE, 2022, pp. 25–40.
- [21] Ibele, A., Romo, S., Williams, B., Kramer, S., and Noster, T., “Convolutional neural network approaches for deep-space object detection in wide field of view camera arrays,” 2022.
- [22] Tao, J., Cao, Y., and Ding, M., “SDebrisNet: A Spatial–Temporal Saliency Network for Space Debris Detection,” *Applied Sciences*, Vol. 13, No. 8, 2023, p. 4955.
- [23] Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al., “Searching for mobilenetv3,” *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.
- [24] Yuan, Y., Bai, H., Wu, P., Guo, H., Deng, T., and Qin, W., “An Intelligent Detection Method for Small and Weak Objects in Space,” *Remote Sensing*, Vol. 15, No. 12, 2023, p. 3169.
- [25] Varela, L., Boucheron, L., Malone, N., and Spurlock, N., “Streak detection in wide field of view images using Convolutional Neural Networks (CNNs),” *Advanced Maui Optical and Space Surveillance Technologies Conference*, edited by S. Ryan, 2019, p. 89.
- [26] Temple, D., “Conditionally Augmented Temporal Anomaly Reasoner And Convolutional Tracking System,” *Advanced Maui Optical and Space Surveillance Technologies Conference*, edited by S. Ryan, 2019, p. 87.
- [27] Bobrovsky, A. I., Galeeva, M. A., Morozov, A. V., Pavlov, V. A., and Tsytulin, A. K., “Automatic detection of objects on star sky images by using the convolutional neural network,” *Journal of Physics: Conference Series*, Vol. 1236, No. 1, 2019, p. 012066. <https://doi.org/10.1088/1742-6596/1236/1/012066>, URL <https://doi.org/10.1088/1742-6596/1236/1/012066>.
- [28] Ronneberger, O., Fischer, P., and Brox, T., “U-net: Convolutional networks for biomedical image segmentation,” *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, Springer, 2015, pp. 234–241.
- [29] De Vittori, A., Cipollone, R., Di Lizia, P., and Massari, M., “Real-time space object tracklet extraction from telescope survey images with machine learning,” *Astroynamics*, Vol. 6, No. 2, 2022, pp. 205–218.
- [30] Mastrofini, M., Agostinelli, I., and Curti, F., “Design and Validation of a U-Net-Based Algorithm for Star Sensor Image Segmentation,” *Applied Sciences*, Vol. 13, No. 3, 2023, p. 1947.
- [31] Su, S., Niu, W., Li, Y., Ren, C., Peng, X., Zheng, W., and Yang, Z., “Dim and Small Space-Target Detection and Centroid Positioning Based on Motion Feature Learning,” *Remote Sensing*, Vol. 15, No. 9, 2023, p. 2455.

- [32] Jordan, J., Posada, D., Zuehlke, D., Radulovic, A., Malik, A., and Henderson, T., “Satellite detection in unresolved space imagery for space domain awareness using neural networks,” *arXiv preprint arXiv:2207.11412*, 2022.
- [33] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., “Mobilenetv2: Inverted residuals and linear bottlenecks,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [34] Dai, Y., Zheng, T., Xue, C., and Zhou, L., “Effective Multi-Frame Optical Detection Algorithm for GEO Space Objects,” *Applied Sciences*, Vol. 12, No. 9, 2022, p. 4610.
- [35] Chen, B., Liu, D., Chin, T.-J., Rutten, M., Derksen, D., Martens, M., von Looz, M., Lecuyer, G., and Izzo, D., “Spot the GEO Satellites: From Dataset to Kelvins SpotGEO Challenge,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2086–2094.
- [36] Fletcher, J., McQuaid, I., Thomas, P., Sanders, J., and Martin, G., “Feature-Based Satellite Detection Using Convolutional Neural Networks,” *Advanced Maui Optical and Space Surveillance Technologies Conference*, edited by S. Ryan, 2019, p. 16.
- [37] Bertin, E., and Arnouts, S., “SExtractor: Software for source extraction,” *Astronomy and Astrophysics Supplement Series*, Vol. 117, No. 2, 1996, pp. 393–404. <https://doi.org/10.1051/aas:1996164>, URL <https://doi.org/10.1051/aas:1996164>.
- [38] Chen, Z., Yang, Y., Bettens, A., Eun, Y., and Wu, X., “A Simulation-Augmented Benchmarking Framework for Automatic RSO Streak Detection in Single-Frame Space Images,” *arXiv preprint arXiv:2305.00412*, 2023.
- [39] Dumitrescu, F., Ceachi, B., Truică, C.-O., Trăscău, M., and Florea, A. M., “A Novel Deep Learning-Based Relabeling Architecture for Space Objects Detection from Partially Annotated Astronomical Images,” *Aerospace*, Vol. 9, No. 9, 2022, p. 520.
- [40] Faster, R., “Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, Vol. 9199, No. 10.5555, 2015, pp. 2969239–2969250.
- [41] He, K., Zhang, X., Ren, S., and Sun, J., “Deep residual learning for image recognition,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [42] Li, H., Niu, Z., Sun, Q., and Li, Y., “Co-Correcting: Combat Noisy Labels in Space Debris Detection,” *Remote Sensing*, Vol. 14, No. 20, 2022, p. 5261.
- [43] Chaudhary, A., Payne, T., Gregory, S., and Dao, P., “Fingerprinting of Non-resolved Three-axis Stabilized Space Objects Using a Two-Facet Analytical Model,” 2011.
- [44] Linares, R., Jah, M. K., Crassidis, J. L., and Nebelecky, C. K., “Space Object Shape Characterization and Tracking Using Light Curve and Angles Data,” , Jan. 2014. <https://doi.org/10.2514/1.62986>, URL <http://dx.doi.org/10.2514/1.62986>.
- [45] Furfaro, R., Linares, R., Gaylor, D., Jah, M., and Walls, R., “Resident Space Object Characterization and Behavior Understanding via Machine Learning and Ontology-based Bayesian Networks,” 2016.
- [46] Furfaro, R., Linares, R., and Reddy, V., “Shape Identification of Space Objects via Light Curve Inversion Using Deep Learning Models,” *Advanced Maui Optical and Space Surveillance Technologies Conference*, edited by S. Ryan, 2019, p. 17.
- [47] Yao, L., and Chang-yin, Z., “The Basic Shape Classification of Space Debris with Light Curves,” , Apr. 2021. <https://doi.org/10.1016/j.chinastron.2021.05.005>, URL <http://dx.doi.org/10.1016/j.chinastron.2021.05.005>.
- [48] Cook, R. L., and Torrance, K. E., “A Reflectance Model for Computer Graphics,” , Jan. 1982. <https://doi.org/10.1145/357290.357293>, URL <http://dx.doi.org/10.1145/357290.357293>.
- [49] Karpov, S., Beskin, G., Biryukov, A., Bondar, S., Ivanov, E., Katkova, E., Perkov, A., and Sasyuk, V., “Mini-Mega-TORTORA wide-field monitoring system with sub-second temporal resolution: first year of operation,” *Revista Mexicana de Astronomia y Astrofisica Conference Series*, Revista Mexicana de Astronomia y Astrofisica Conference Series, Vol. 48, 2016, pp. 91–96.
- [50] van Rooij, S. B., Uysal, F., Papari, G., Kruithof, M. C., and Cuenca, M. C., “IAA-ICSSA-04-04-2022 DETERMINATION OF SATELLITE SOLAR PANEL AND BUS SIZE FROM RADAR IMAGES WITH DEEP LEARNING,” ????
- [51] Simonyan, K., and Zisserman, A., “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

- [52] Balachandran, K., and Subbarao, D., “Classification of Resident Space Objects by shape and spin motion using neural networks and photometric light curves,” *Proceedings of the 8th European Conference on Space Debris, Darmstadt, Germany*, 2021, pp. 20–23.
- [53] Dao, P., Haynes, K., Gregory, S., Hollon, J., Payne, T., and Kinateder, K., “Machine Classification and Sub-classification Pipeline for GEO light curves,” *Advanced Maui Optical and Space Surveillance Technologies Conference*, edited by S. Ryan, 2019, p. 53.
- [54] Zhang, H., Zhang, C., Jiang, Z., Yao, Y., and Meng, G., “Vision-Based Satellite Recognition and Pose Estimation Using Gaussian Process Regression,” , Jun. 2019. <https://doi.org/10.1155/2019/5921246>, URL <http://dx.doi.org/10.1155/2019/5921246>.
- [55] Zhang, H., and Jiang, Z., “Multi-view space object recognition and pose estimation based on kernel regression,” , Oct. 2014. <https://doi.org/10.1016/j.cja.2014.03.021>, URL <http://dx.doi.org/10.1016/j.cja.2014.03.021>.
- [56] Afshar, R., and Lu, S., “Classification and Recognition of Space Debris and Its Pose Estimation Based on Deep Learning of CNNs,” , 2020. https://doi.org/10.1007/978-3-030-50726-8_79, URL http://dx.doi.org/10.1007/978-3-030-50726-8_79.
- [57] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., “Going Deeper with Convolutions,” 2014. <https://doi.org/10.48550/ARXIV.1409.4842>, URL <https://arxiv.org/abs/1409.4842>.
- [58] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., “ImageNet: A large-scale hierarchical image database,” , Jun. 2009. <https://doi.org/10.1109/cvpr.2009.5206848>, URL <http://dx.doi.org/10.1109/CVPR.2009.5206848>.
- [59] Badura, G., Valenta, C., and Gunter, B., “Convolutional Neural Networks for Inference of Space Object Attitude Status,” 2020.
- [60] Maxwell, J. R., Beard, J., Weiner, S., Ladd, D. E., and Ladd, S., “Bidirectional Reflectance Model Validation and Utilization.” 1973. URL <https://api.semanticscholar.org/CorpusID:59800301>.
- [61] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P., “Gradient-based learning applied to document recognition,” , 1998. <https://doi.org/10.1109/5.726791>, URL <http://dx.doi.org/10.1109/5.726791>.
- [62] Paulete, C., Cano, D., Siminski, J., Pérez, C., Escobar, D., and Tirado, J., “AIMLRCS: A Machine Learning approach to spacecraft attitude and object identification based on RCS from the S3TSR,” *8th European Conference on Space Debris*, 2021, p. 167.
- [63] McNally, K., Ramirez, D., Anton, A. M., Smith, D., and Dick, J., “Artificial Intelligence for Space Resident Objects Characterisation with Lightcurves,” *8th European Conference on Space Debris*, 2021, p. 85.
- [64] Qashoa, R., and Lee, R., “Classification of Low Earth Orbit (LEO) Resident Space Objects’(RSO) Light Curves Using a Support Vector Machine (SVM) and Long Short-Term Memory (LSTM),” *Sensors*, Vol. 23, No. 14, 2023, p. 6539.
- [65] Chaudhary, S., Jain, P., Jakhetiya, V., Guntuku, S. C., and Subudhi, B. N., “Localizing Features with Masking for Satellite and Debris Classification,” , Sep. 2021. <https://doi.org/10.1109/icipc53495.2021.9620178>, URL <http://dx.doi.org/10.1109/ICIPC53495.2021.9620178>.
- [66] Musallam, M. A., Ismaeil, K. A., Oyedotun, O., Perez, M. D., Poucet, M., and Aouada, D., “SPARK: SPAcecraft Recognition leveraging Knowledge of Space Environment,” 2021. <https://doi.org/10.48550/ARXIV.2104.05978>, URL <https://arxiv.org/abs/2104.05978>.
- [67] Tan, M., and Le, Q. V., “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *arXiv*, 2019. <https://doi.org/10.48550/ARXIV.1905.11946>, URL <https://arxiv.org/abs/1905.11946>.
- [68] Krantz, H., Pearce, E. C., and Block, A., “Characterization of LEO satellites with all-sky photometric signatures,” *arXiv preprint arXiv:2210.03215*, 2022.
- [69] Walker, L., Vasile, M., Campbell, A., Murray, P., Marshall, S., and Savitski, V., “Intelligent Characterisation of Satellites from Hyperspectral Time Series Analysis,” ????
- [70] Li, N., Gong, C., Zhao, H., and Ma, Y., “Space target material identification based on graph convolutional neural network,” *Remote Sensing*, Vol. 15, No. 7, 2023, p. 1937.
- [71] Dupree, W., Penafiel, L., and Gemmer, T., “Time forecasting satellite light curve patterns using Neural Networks,” *22nd Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), Maui, HI*, 2021.

Paper	Data Type	Real vs Simulated Data
Edith Stöveken et al [2]	Optical Telescope Images	Real data
Li, K. et al [3]	Optical Telescope Images	Unclear
Sun, R.-Y. et al [5]	Optical Telescope Images	Real data
Xi, J. et al. [6]	Optical Telescope Images	Simulated data
Xi, J. et al. [7]	Optical Telescope Images	Simulated data
Sun, R. et al. [8]	Optical Telescope Images	Real data
Núñez, J. et al [9]	Optical Telescope Images	Real data
Feuge-Miller, B. et al [10]	Optical Telescope Images	Real data
Huang, T et al [11]	Optical Telescope Images	Real data
Xi, J., at al. [12]	Optical Telescope Images	Simulated data
Du, Y. et al. [13]	Optical Telescope Images	Simulated data
Montanaro A. et al [14]	Optical Telescope Images	Simulated data
Zhang, X. et al [15]	Optical Telescope Images	Real data
Lin, B. et al [16]	Optical Telescope Images	Real data
Xiang, Y. et al. [17]	Optical Telescope Images	Simulated data
Guo, X. et al [18]	Optical Telescope Images	Simulated data
Kyselica, D. et al [19]	Optical Telescope Images	Simulated data
Fitzgerald, G. et al [20]	Optical Telescope Images	Real data
Ibele, A. et al [21]	Optical Telescope Images	Both
Tao, J. et al [22]	Optical Telescope Images	Both
Yuan, Y. et al [24]	Optical Telescope Images	Simulated data
Varela, L. et al. [25]	Optical Telescope Images	Real data
Temple, D. [26]	Optical Telescope Images	Real data
Bobrovsky, A. et al [27]	Optical Telescope Images	Real data
De Vittori, A. et al [29]	Optical Telescope Images	Both
Mastrofini, M. et al [30]	Optical Telescope Images	Both
Su, S. et al [31]	Optical Telescope Images	Both
Jordan, J. et al [32]	Optical Telescope Images	Both
Dai, Y. et al [34]	Optical Telescope Images	Real data
Fletcher J. et al. [36]	Optical Telescope Images	Both
Chen, Z. et al [38]	Optical Telescope Images	Both
Dimitrescu, F. et al [39]	Optical Telescope Images	Real
Li, H. et al [42]	Optical Telescope Images	Both

Table 1 Types of datasets used in the RSO detection papers.

Paper	Data Type	Real vs Simulated Data	Task
Chaudhary, A. et al [43]	Hyperspectral images	Real data	Attitude/Pose Estimation and Classification
Linares, R. et al [44]	Lightcurve (Magnitude) and Angles data	Simulated data	Attitude/Pose Estimation and Classification
Furfaro, R. et al [45]	Angles data	Real data	Behaviour / Intent Classification
Linares, R. et al [46]	Lightcurve (Magnitude) data	Simulated data	Shape Characterization
Yao, L. et al [47]	Lightcurve (Magnitude) data	Both	Shape Characterization
van Rooij, S. et al. [50]	Inverse Synthetic Aperture Radar (ISAR) images	Simulated data	Shape Characterization
Balachandran, K. et al [52]	Lightcurve (Magnitude) data	Simulated data	Shape Characterization
Phan, D. et al in 2019 [53]	Lightcurve (Magnitude) data	Real data	Attitude/Pose Estimation and Classification
Zhang, H. et al [54]	Optical images	Simulated data	Attitude/Pose Estimation and Classification
Afshar, R. et al [56]	Optical images	Simulated data	Attitude/Pose Estimation and Classification
Badura, G. [59]	Lightcurve (Magnitude) data	Simulated data	Attitude/Pose Estimation and Classification
Paulete, C. et al [62]	Radar Cross Section (RCS) data	Real data	Attitude/Pose Estimation and Classification
McNally, K. et al [63]	Lightcurve (Magnitude) data	Both	Attitude/Pose Estimation and Classification
Qashoa, R. et al [64]	Lightcurve (Magnitude) data	Real data	Attitude/Pose Estimation and Classification
Chaudhary, S. et. al [65]	Optical images	Simulated data	Space Object Type Classification / Target Recognition
Krantz, H. et al [68]	Lightcurve (Magnitude) and Angles data	Real data	Space Object Type Classification / Target Recognition
Walker, L. et al [69]	Hyperspectral images	Simulated data	Space Object Type Classification / Target Recognition
Li, N. et al [70]	Hyperspectral images	Both	Material Identification
Durpee, W. et al [71]	Lightcurve (Magnitude) data	Real data	Future lightcurve prediction

Table 2 Types of datasets used in the RSO Characterization papers.