Research paper

A transfer learning approach to space debris classification using observational light curve data

James Allworth [a,b,*], Lloyd Windrim [a], James Bennett [b], Mitch Bryson [a]

[a] *Australian Centre for Field Robotics, Rose Street Building, University of Sydney, NSW, 2006, Australia*
[b] *EOS Space Systems Pty Ltd, 18 Wormald St, Symonston, ACT, 2609, Australia*

ARTICLE INFO

ABSTRACT

This paper presents a data driven approach to space object characterisation through the application of machine learning techniques to observational light curve data. One-dimensional convolutional neural networks are shown to be effective at classifying the shape of objects from both simulated and real light curve data. To the best of the authors' knowledge this is the first generalised attempt to classify the shape of space objects using real observational light curve data.

It is also demonstrated that transfer learning is successful in improving the overall classification accuracy on real light curve datasets. The authors develop a simulated light curve dataset using a high fidelity three-dimensional ray-tracing software. The simulator takes in a textured geometric model of a Resident Space Object as well as its ephemeris and uses ray-tracing software to generate photo-realistic images of the object that are then processed to extract the light curve. Models that are pre-trained on the simulated dataset and then fine-tuned on the real datasets are shown to outperform models purely trained on the real datasets. This result indicates that transfer learning will allow organisations to effectively utilise deep learning techniques without the requirement to build up large real light curve datasets for training.

1. Introduction

Space debris poses a significant collision risk to current and future space missions due to the number of debris fragments and the difficulty in removing them from orbit. This issue is exacerbated by a collisional cascading effect, where collisions between objects create debris, each increasing the chance of more collisions and thus triggering an uncontrolled growth in fragments [1]. The risk to operational spacecraft is reduced through Space Situational Awareness (SSA), whereby organisations attempt to maintain orbital information about all objects in space and predict their future states, allowing satellite operators to make informed decisions [2]. However, a significant proportion of the space debris population is uncharacterised, which means that information about an object's size, shape, material and orientation are unknown or poorly defined [3]. These unknown characteristics limit the accuracy of orbital predictions, as there is a coupling between the object's characteristics and orbital perturbations caused by non-conservative forces such as drag and solar radiation pressure [4]. Resident Space Object (RSO) characterisation is the process of determining the characteristics of an object, commonly from ground-based observations. It is an important goal in SSA to improve the accuracy of orbital predictions to reduce the collision risk to satellites.

This paper presents an approach to RSO characterisation using light curve data. A light curve is a series of discrete measurements of an object's brightness, typically measured as an apparent magnitude over time. Light curves can be collected for any RSO that is illuminated by the sun and visible to an observer. They are generally extracted from earth-based visual observations using a camera attached to a telescope, where the spatial structure of the object is not resolvable due to the attenuation of the light through the atmosphere as well as the size and distance of the object. Previous research has demonstrated that light curves vary with, and hence potentially contain information about, an RSO's physical characteristics including size, shape, orientation and material composition [5–8]. Traditional methods for recovering the RSO's physical characteristics from the light curve data have required the estimation of a significant number of parameters using a light curve inversion process [9]. These methods have shown some success for simple shape and attitude estimation based on simulated light curve data, however these results have not been repeated on real light curve data.

This paper proposes a data driven approach to RSO characterisation using machine learning techniques applied to light curve observations. Machine learning has experienced a rapid increase in popularity over

---

the last decade, primarily due to the exceptional results that deep learning models have achieved on a range of perceptual problems [10]. Deep learning is a subset of machine learning that involves the use of "deep" models, such as multi-layered neural networks which have the capacity to learn complex tasks. A deep neural network typically comprises several layers of non-linear function approximators that are trained to learn representations of data useful for tasks such as classification [11]. Deep learning has proven most successful when the networks are provided with a large well labelled training dataset in order to learn the complex non-linear relationships within the data. This is expected to be the major limiting factor in the application of deep learning to RSO characterisation, as real light curve data is both challenging to obtain and label [12]. Thus it is difficult to build up a large well-balanced dataset for a range of different object classes. This idea is supported in the literature where research has shown Convolutional Neural Networks (CNNs) to perform well on large simulated light curve datasets, while not being able to achieve the same results on real light curve data [13].

In this paper we develop a novel approach to data-driven RSO characterisation that overcomes the limitations of small training datasets by using a process of "simulation transfer learning". Transfer learning is a process by which the performance of a model on a machine learning task is improved by utilising knowledge acquired from an initial model that has been previously trained on a related task. When this initial model has had access to a large and extensive training dataset, it can boost the performance in the new task for which the available training dataset is much smaller, even when the task is not exactly the same. The key contribution of our work is to demonstrate a framework for simulation transfer learning, in which the initial model is trained using a high fidelity simulation of the real task (i.e. simulated light curves of RSOs). We demonstrate an approach in which a neural network is first trained on a large simulated dataset, before the weights of this network are used to initialise a separate neural network which is trained on smaller real light curve dataset using a process of "fine-tuning". Our results illustrate that this approach out-performs models for RSO characterisation that were trained purely on either the real dataset, or simulation data alone.

1.1. Related work

Initial research into light curve based RSO characterisation was primarily conducted using simulated data as a result of the difficulty in both obtaining and determining ground truth values for observational data. Light curve inversion techniques, first used for asteroid characterisation [14,15], were applied to RSO characterisation with the aim of determining the shape and attitude of objects using simulated light curves [9,16]. Whilst this technique was successful in estimating the shape and state of non-resolvable asteroids [14,15], artificial RSOs have highly angular shapes with a variety of materials and a wide range of possible rotation states [9]. These factors significantly increase the complexity of applying this method to space debris characterisation and require a large number of non-linear parameters to be solved [9].

Data fusion was subsequently explored in the literature, through the combination of light curve and object position data, to take advantage of coupling between an object's characteristics and orbital dynamics [5–8]. This provided additional information into the light curve inversion process and was typically used in conjunction with the Unscented Kalman Filter (UKF) to account for the non-linearity in the orbital dynamics, object rotation state and measurement equations [5–8]. More recent research has suggested that these non-linearities and the possibility of multiple solutions invalidate the Gaussian assumption made by the UKF [17,18]. Consequently, Bayesian estimators have been investigated as possible method for solving the light curve inversion problem in the non-Gaussian space [17,18]. Whilst the outlined methods have a solid theoretical background, they have focused on simulated data and often been applied to relatively simplistic versions

of the full characterisation problem. They have yet to demonstrate effectiveness on more generalised characterisation cases using real light curve data.

As RSO characterisation has been shown to improve SSA, more optical tracking campaigns have focused on the collection and analysis of light curves [19–24]. These campaigns demonstrate that a significant portion of the debris population is rotating in the satellite body-fixed coordinate system and that these rotations often result in discernible patterns in the light curves. Light curves have been analysed to determine the rotation rate of debris and by tracking small groups of objects over a number of years, studies have shown that this rotation rate often changes over time [22,24]. Silha et al. examined light curves for over 400 individual objects during two separate tracking campaigns and observed some general trends to be present relating the light curve to the shape of the object [21,23]. Rocket bodies or cylindrically shaped payloads were typically found to have two distinct peaks per rotation, whilst four distinct peaks were often observed for box-winged payloads. Whilst there were outliers to these trends, these are promising results for the possibility of performing RSO shape classification based on patterns in real light curves.

In the last five years, the growing catalogue of space debris as well as the rise in prominence of machine learning techniques, has led to research focusing on data driven machine learning approaches to RSO characterisation. Initial work in this domain used traditional machine learning techniques such as Random Decision Forest (RDF) and Support Vector Machines (SVM) to perform shape classification on simulated light curve data [25–27]. These techniques have also been applied to real data showing successful classifications for simplified two or three class problems, where light curves have been used to distinguish between general object classes (i.e. payloads vs. rocket bodies vs. debris) as opposed to classifying objects based on their specific shape [28,29]. While this work shows promise, it should be noted that in both cases the datasets were highly skewed towards payloads, making it difficult to generalise these results.

The recent wave of success of deep neural networks to a variety of perceptual problems has inspired the application of these techniques to the problem of RSO classification using light curves [12,13,30–33]. Furfaro and Linares found that on shape classification tasks involving simulated data, there was only a small improvement in accuracy when using a 1 Dimensional Convolutional Neural Network (1D-CNN) compared with traditional machine learning techniques [13,31]. Similar to previous work, the authors applied this method to a three class classification on real light curve data, attempting to identify objects as rocket bodies, debris and other. It was found that the 1D-CNN significantly outperformed traditional machine learning techniques on the 3 class real data classification, achieving an accuracy of 75.4% [13]. The authors argued that the many layer configuration of the CNN enables it to learn more complex decision boundaries, which suggests that the simulated light curve data does not effectively encapsulate the complexities present in the real light curve data.

As a result of the large number of parameters in deep neural networks and in particular CNNs, significant amounts of data and computational time is required to achieve optimal performance. This was first demonstrated in 2012, when AlexNet, a CNN containing 60 million parameters, improved state-of-the-art results on an image classification task by more than 10% by training on a dataset of 1.2 million images [34]. Current state-of-the-art models contain significantly more parameters and are typically trained on much larger datasets. For classification tasks within the image domain, it has become standard practice to apply transfer learning in order to improve CNN performance and reduce training requirements [35]. Networks are pre-trained on very large standardised datasets, enabling the earlier layers to learn generic basic image features such as edges and corners [36,37]. These weights are then fine-tuned on the target/goal dataset to allow the parameters in the later layers to be trained to identify specific features that are unique to the target/goal dataset. This method has been shown to

be particularly effective where there is a limited amount of training data available for a specific task [38,39]. In addition to imagery data, transfer learning has proven successful when applied to a range of other data types including hyperspectral data [40], MRI data [35], synthetic aperture radar data [41] and time sequence data [42]. In cases where there is limited real data available within a domain, simulation data has been used successfully in the pre-training phase to improve model performance on the target/goal dataset containing real world data [41,43,44]. Transfer learning is most successful when the target/goal dataset is similar to the dataset used for pre-training the model as this reduces the amount of fine-tuning required.

Previous research into RSO characterisation using simulated light curves has typically applied simplified Flat Facet Models (FFM) of RSOs combined with various Bidirectional Reflection Distribution Functions (BRDFs) to model reflectance [6,13,17]. In the literature both the simplified anisotropic-Phong BRDF developed by Ashikhmin and Shirley [45] and the Cook–Torrance BRDF [46] have been commonly used to simulate the resulting reflected light from the sun based on the orientation of the flat facets representing the RSO's structure [47]. Whilst these methods produce realistic results when applied to simple shapes, there are a number of limitations of using a FFM with a BRDF. The FFM does not incorporate self-shadowing or any secondary reflections, both of which need to be considered for RSOs with concavities or extrusions from the main body like solar panels. Additionally, it is difficult to develop complex shapes with a number of different materials using the FFM and features such as curved surfaces are unable to be correctly implemented. Finally, the FFM does not consider the blurring effect caused by the rotation of the object during the period of camera exposure, which is an important factor for fast rotating objects. In order to be utilised in transfer learning on real light curve datasets, it is expected that simulated light curves will need encapsulate the features present in the real data and thus incorporate many of the effects outlined above.

It has been demonstrated in the research outlined that it is possible to classify the shape of objects using simulated light curves. Observations on trends in real light curve data have also identified that similarly-shaped objects often produce light curves with a similar pattern. However, attempts to apply classification methods to real data have proven to be less successful due to a lack of real data and differences between the complexity of the simulated and real light curves. Recent results suggest that deep learning is more suitable to dealing with classification of data with highly complex decision boundaries, however these methods rely on large and well labelled datasets to be most effective. Transfer learning is known to be an effective way of increasing the performance of neural networks, particularly in cases where the dataset is small. This research explores the idea of developing a high fidelity simulated light curve dataset for the purpose of pre-training a neural network in order to improve classification results on a real light curve dataset.

1.2. Contributions and paper outline

The key contributions of this paper are:

- Development and use of a high fidelity light curve simulation environment for providing realistic training data, which is used to boost the performance of neural network models for RSO characterisation based on the principles of transfer learning.
- Development of a CNN model for RSO shape classification that performs effectively on both simulated and real data.
- Demonstration that transfer learning improves classification performance on real light curve data and reduces the need for large well labelled real light curve datasets.

Section 2 presents the methodology for our approach, including the development of the light curve simulation environment, the extraction

of real light curves from Electro Optic Systems (EOS) Space Systems[1] observational data, the development of the classification model and finally the transfer learning process. Section 3, the experimental setup, introduces the four light curve datasets that are analysed in the remainder of the paper as well as providing an overview of the model setup and selected training parameters. The results are presented in Section 4, with a brief examination of some of the properties of the real EOS light curves that were collected. This is followed by the presentation of the object classification results for each of the datasets and the transfer learning results on the real light curve datasets. In Section 5 these results are discussed and compared to previous results in the literature. Finally, an overview of the key results and conclusions are provided in Section 6.

2. Methodology

2.1. Light curve simulation environment

A light curve simulator was developed with the aim of generating high fidelity simulated light curves. The simulator takes a textured geometric model of a given RSO, the RSO orbital parameters and ground-based sensor location. It uses this information to generate the simulated measurement of a light curve (apparent magnitude vs. time), based on the solar reflection from the object.

The simulator pipeline is depicted in Fig. 1 and is composed of three main steps:

1. Initialisation Step: Real world sensor and sun position are recorded as well as object ephemeris (position and velocity) data to build a profile of the object illumination conditions and imaging angles.
2. 3D Rendering Step: Takes in the ephemeris data generated in the initialisation step as well as textured geometric model of an RSO and generates a sequence of realistic images. Images were generated using the open source rendering software Blender[2] and its built in physically based probabilistic ray-tracing engine Cycles.[3]
3. Light Curve Extraction Step: Processes the sequence of rendered images to extract the apparent magnitude for the object at each timestep, which is then recorded in the form of a light curve.

An initial version of the light curve simulator developed here was originally presented in [48]. In this paper we summarise key aspects of the simulator for clarity to the reader.

2.1.1. Initialisation step

The simulation environment aims to accurately replicate the real world geometry of the three primary components: the RSO, the ground-based sensor and the sun. As such, the position and motion of these objects are calculated in Python using the Two Line Element (TLE) for the object, which is propagated using the Simplified General Perturbations 4 (SGP4) propagator [49]. This is done in conjunction with the Naval Observatory Vector Astrometry Software (NOVAS) library to output the position information for the three components at each time step for a specified sampling rate [50]. The final part of the initialisation step checks the object is visible to the sensor at each of these timesteps. This involves determining that the RSO is not in earth shadow, that the sensor is in darkness and that the RSO is more than 15° above the horizon from the perspective of the sensor. The recorded ephemeris information is then fed into the second phase of the simulation environment, the 3D rendering step.
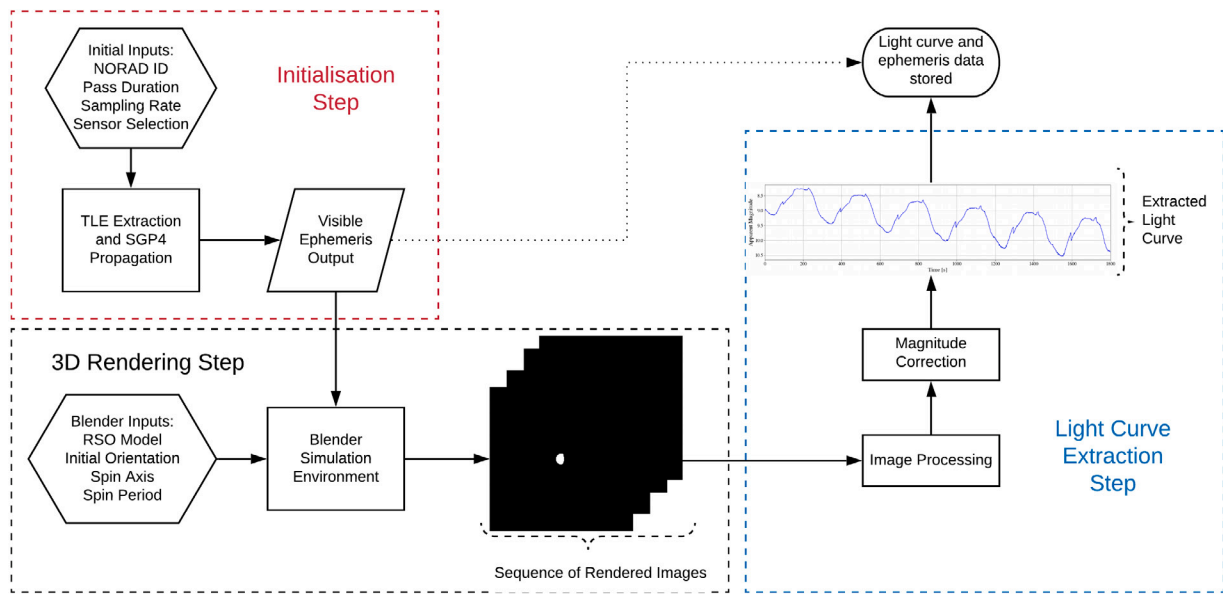

Fig. 1. Simulation Environment Set Up: The Initialisation Step uses real world data for the sun, sensor and specified object to determine visibility and viewing angles. The 3D Rendering Step uses this information to generate a sequence of realistic images which are then processed in the Light Curve Extraction Step.
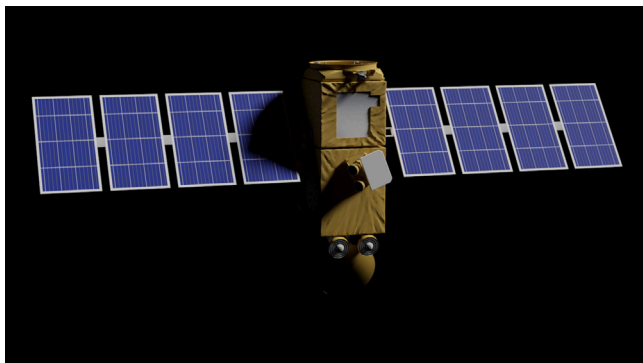

Fig. 2. Blender-based rendering of a model of the Jason 1 satellite. The ray tracing engine used with Blender allows for complex effects (such as self-shadowing) which improves the fidelity of simulated light curve profiles.

2.1.2. 3D rendering step

The 3D rendering step takes in the ephemeris data for the three primary components (the RSO, the ground-based sensor and the sun) as well as a textured geometric model of the RSO and produces a photo-realistic image of the object as would appear to the telescope/light curve sensor on the ground. A number of satellite models with a variety of different shapes and configurations have been downloaded from the NASA 3D models website [51] and used to simulate light curves. An example of one of these models rendered in Blender is depicted in Fig. 2. Self shadowing and secondary reflections, where light is reflected from one surface onto another, are captured by the Blender ray-tracing engine which enhance the realism of the simulated measurements.

There are a range of configurable options prior to the simulation of a pass, including the initial orientation and rotation period of the object as well as specific camera settings depending on sensor configuration. These parameters can be controlled by the user to model specific tracking scenarios or initialised randomly from a range of possible options during the generation of simulated dataset. Once initialised, Blender simulates images of the rotating object for the duration of the pass using the built-in physically based probabilistic ray-tracing engine called Cycles. This sequence of images is then passed into the light

curve extraction step, which is the third and final step in the simulation pipeline.

2.1.3. Light curve extraction step

This step processes the sequence of rendered images to determine the apparent magnitude of object in each image, which enables comparisons with real light curve data. As the RSO is the only object within the image, the brightness of the object is calculated through the integration of the pixel counts over the image. However, this brightness is not a physically standardised system and is only self consistent when compared to other Blender images.

In the real world, a similar issue occurs with the brightness, or instrumental magnitude, of an object in the image depending on a number of factors including the optical observing system and atmospheric conditions. In order to convert the instrumental magnitude of the object in the image to a standardised apparent magnitude, the objects instrumental magnitude is compared with the instrumental magnitude of stars within the same image. These stars have a catalogued apparent magnitude enabling the conversion of instrumental magnitude to apparent magnitude for the particular optical conditions present in the image using the following equation:

$$m_1 = m_2 + I_{m_1} - I_{m_2} \tag{1}$$

where m_1 is the apparent magnitude of the object being determined, m_2 is the known apparent magnitude of an object (usually a star) and I_{m_1} and I_{m_2} are the calculated instrumental magnitudes of the two objects in the image. It is important to note that in order to achieve high accuracy using this method, the two objects should be relatively close in angular distance. As the angular distance between the objects increases, it becomes more likely that the light from each object is passing through different atmospheric conditions and thus will reduce the accuracy of the calculated apparent magnitude. In particular, this can become an issue when using a sensor with a wide field of view.

Based on this real world solution, a similar approach was used to convert the brightness of objects in Blender rendered images to apparent magnitude through comparison with the Ashikhmin–Shirley BRDF applied to the FFM. Rather than producing imagery, the FFM takes the intensity of the sunlight as an input and calculates the fraction of light reflected towards the observer using the BRDF, after a correction is applied for distance. Since the output from the model is a fraction of the

intensity of the sunlight, this can be directly converted to an apparent magnitude using the apparent magnitude of the sun for calibration.

Using this relationship, a flat plate of the same size, material and viewing conditions was simulated using the FFM and the Blender simulation environment. This was then used as a calibration object in order to convert from instrumental magnitude in Blender to the apparent magnitude system using the following equation:

$$m_1 = m_{ff} - I_{m_b} - 2.5 \log_{10} \left(\frac{B}{t_{exp} \cdot (\frac{d}{d_r} \cdot scale)^2} \right) \qquad (2)$$

where m_{ff} is the apparent magnitude of the flat plate at 10 000 m determined using the flat facet model, I_{m_b} is the Blender instrumental magnitude of the flat plate at 10 000 m, B is the sum of the pixel intensity of the object extracted from the Blender rendered image during the camera exposure, expressed in counts (dimensionless), whilst t_{exp} is the exposure time in seconds. The parameter d is the distance between the object and the camera used in Blender, which is expressed in metres, whilst d_r is the distance used for the reference object which was 10 000 m. Finally, $scale$ is a dimensionless scale factor that was applied to the object, camera and sun positions when inputting them into the Blender simulation. This scale factor is typically set to 1000 in order to reduce the distances used in the simulation, as the accuracy of Blender simulations decreases at extreme distances. Further details on the derivation of this equation can be found in Ref. [48].

2.1.4. Post processing

The Blender simulated data was initially generated with a uniform length and 1 s sampling rate with no external noise. In contrast, real light curve tracks typically have a non-uniform length and sampling as well as missing observations due to signal interference from background stars and clouds. These effects were also recreated in the simulated light curves through post processing in order to capture their impact during model training and thus to allow them to be accounted for in the models used for processing real data.

Subsequently, the length of the simulated light curves was changed by using Gaussian distributions to randomly select the start and end indices, with the condition that each light curve must contain more than 100 observations. The sampling rate was randomly selected with approximately 60% set to 1 Hz, 30% set to 0.5 Hz and 10% set to 0.33 Hz which were the three most common sampling rates in the EOS dataset. Unlike the EOS dataset, the sampling rate does not change over the course of the pass. Gaussian distributions were also used to randomly select the number of observations to remove in order to simulate bright stars passing behind the object and clouds passing between the sensor and the object.

2.2. Extraction of real light curves

This research into RSO characterisation has been supported by EOS, which is an Australian based company who operate six optical telescopes at two sites in Australia [52]. Starting in June 2019, a tracking campaign has been conducted for the purpose of light curve analysis with EOS collecting imagery on a number of different RSOs. EOS telescopes are typically operated in rate tracking mode, where the telescope follows the apparent movement of the RSO. Consequently, the tracked object appears as an unresolved circular blob in the centre of the image. Background stars can either appear as streaks or as circular blobs depending on the speed of the telescope and the camera's frame rate. Fig. 3(a) depicts a typical raw telescope image that was captured using an EOS telescope. Fig. 3(b) shows the output mask of the same image highlighting the object in green with background stars appearing as white.

A data processing pipeline has been developed in order to extract light curves from a sequence of telescope imagery. Initially, the image is normalised using flat and dark frames to remove sensor specific noise in the image. The algorithm then uses thresholding to separate the background sky brightness from stars and objects in the image. Additionally, a Fast Fourier Transform (FFT) approach is used to enhance peaks in the image and assist in identifying dim objects. The open source Python library PhotUtils,[4] is utilised to produce a mask of objects or stars detected within the image and to provide statistics on each of these detections. This includes determining the size of the aperture for each identified object and the pixel counts within the aperture. Using this information, allows for the conversion to an instrumental magnitude using the following equation outlined in [53]:

$$I_m = -2.5 \log_{10} \left(\frac{N_{ap} - A_{ap} S_{sky}}{t_{exp}} \right) \qquad (3)$$

where I_m is the instrumental magnitude of the object, N_{ap} is the sum of the counts (intensities) of all pixels in the aperture, A_{ap} the area of the aperture (in pixels), S_{sky} is the background sky signal per pixel and t_{exp} is the exposure time of the image (in seconds).

Additional statistics are then collected on the identified objects, allowing them to be classified as either stars, RSOs or noise. The stars in the image are passed into a star mapping algorithm which uses the telescope pointing and the geometrical relationship between the stars to identify them to specific stars in the UCAC4 catalogue [54]. This star identification process enables both astrometric and photometric correction for the RSOs, which allows the conversion from sensor specific instrumental magnitude to apparent magnitude using Eq. (1). This process is repeated over the sequence of images captured in the track to record the apparent magnitude of the object in each image resulting in the final output of a light curve.

Between June 2019 and August 2020, more than 2000 light curves have been extracted from EOS imagery using this method. A non-rotating or very slowly rotating object will typically only have one face visible to the observer during a pass. This is often the case for active satellites which have operator controlled attitude. The resultant light curve will be relatively flat and not provide enough observable information in a single pass to classify the object as a specific shape or configuration. In this case, configuration refers to a specific type of a shape class (e.g. rocket body is the shape of the object and a Falcon 9 rocket body is a specific configuration of the rocket body shape class).

As an initial pre-processing step, Lomb–Scargle analysis was applied to the light curves, in conjunction with manual inspection, to identify if rotational features were present. The Lomb–Scargle periodogram is known to be effective at detecting and characterising periodicity in unevenly sampled time series data [55]. Thus, when applied to light curves, it is useful at identifying periodicities associated with the rotation of the object. To illustrate this point, an example of Lomb–Scargle periodogram applied to an EOS light curve is presented in Section 4.1. For classification purposes any objects that were determined to be non-rotating were removed from the dataset.

The remaining light curves were examined and then further filtered based on photometric quality, which is primarily affected by atmospheric conditions such as cloud cover. Any light curves with large gaps or significant portions of cloud cover which was not corrected by photometric correction processes were removed from the dataset. Additionally, a minimum threshold of 10 light curves was required for a shape/configuration class to be included in the final dataset used for classification as a number of objects were only tracked once or twice.

2.3. Classification model development

The network diagram of the 1D-CNN used in this work is displayed in Fig. 4. This diagram also provides information about various network spatial parameters, such as the number and type of layers utilised as well as the number of neurons in each layer. These parameters were

---

(a) Raw Telescope Image
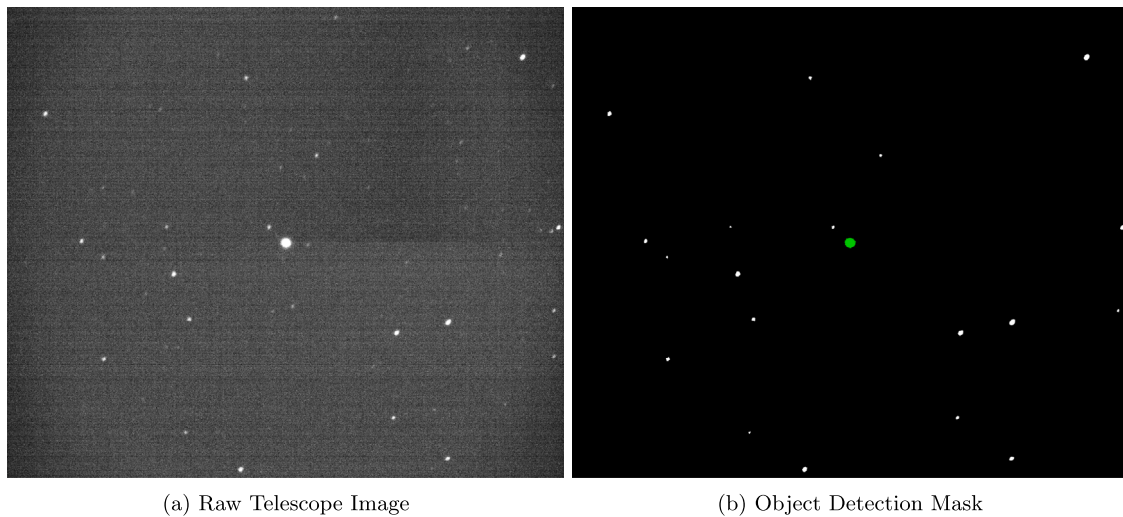
(b) Object Detection Mask

Fig. 3. Non Resolved RSO Identified by the Object Detection Algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
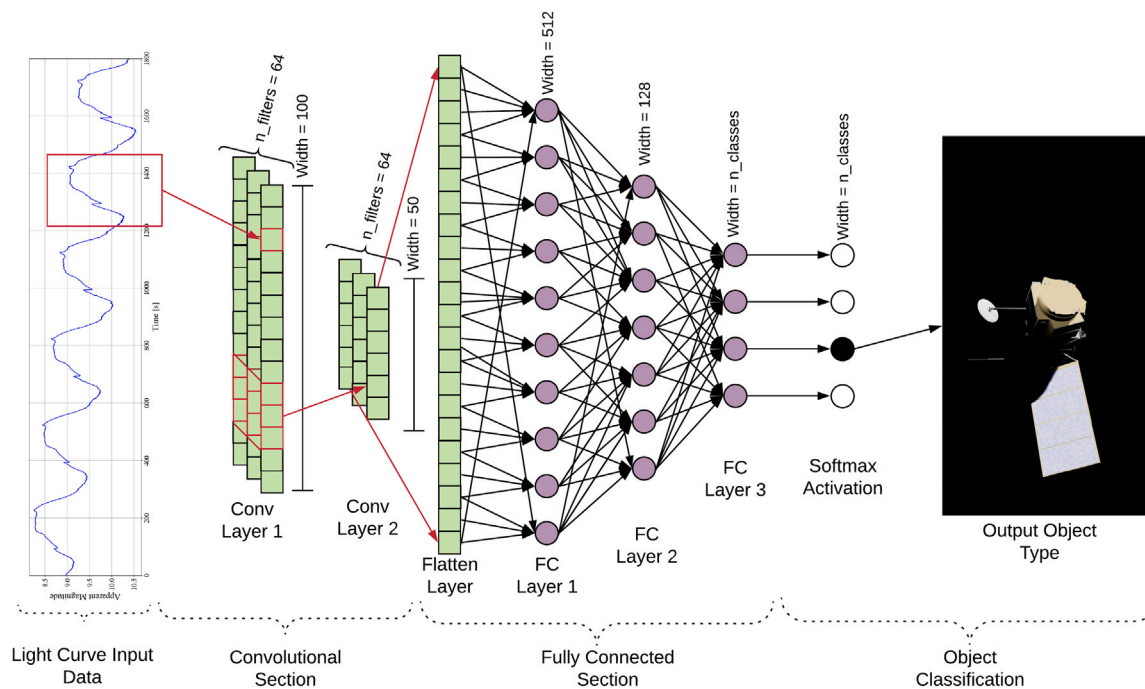

Fig. 4. 1D-CNN Architecture for Object Classification.

selected after an initial grid search was performed to determine the optimal number of convolutional and fully connected layers as well as the size and width of various layers. This grid search was performed for both the Blender simulated dataset and the EOS dataset before the final selection of the network parameters. While these parameters are not optimal for all the datasets used in this research it was decided that for simplicity and ease of comparison, a standardised network with fixed network parameters would be applied to all of the datasets. It is expected that small gains in performance could be achieved through a more in depth search of the parameter space for specific datasets presented in this paper.

The 1D-CNN configuration that was selected has two convolutional layers with each convolutional layer followed by a max pooling layer and a batch normalisation layer. For each convolutional layer a stride size of three was used for the convolutions and padding was not used. After the convolutional layers the parameters are flattened to be fed

into the fully connected section of the network. Three fully connected layers are then utilised with dropout and batch normalisation implemented between each of the fully connected layers. Rectified Linear Unit (ReLU) activation is used for both the convolutional and the fully connected layers, with the exception of the final output layer. A softmax activation is applied to the output of the final layer and it is used to determine the model's classification for a given input. Thus, the number of neurons in this final layer changed to match the number of classes for the specific dataset that was being evaluated. During training, cross-entropy loss is used to calculate a loss value that is then back-propagated through the network to update the network parameters.

In addition to the object's magnitude information, three additional features were selected to create a multidimensional time series input for the model. As discussed previously, the magnitude of the object is affected by the object's characteristics as well as its phase angle and

range. Consequently, the phase angle and range information has been included for every timestep that magnitude data was recorded. Finally, the time since the start of the track in seconds has also been included as the sampling rate was inconsistent for the real data. The CNN requires that each light curve be of uniform length so a standard length of 1200 observations was selected for each light curve. Input light curves are either truncated or zero filled as part of the pre-processing to ensure that they are the standard length. It should be noted that this standard length was primarily selected based on the average length of the EOS light curves, which were typically extracted from 20 min tracks of the object with an average sampling rate of 1 Hz resulting in approximately 1200 observations. Those interested in repeating of further developing this work may achieve better results utilising a different standard length depending on the specifics of their light curve dataset.

2.4. Transfer learning process

Transfer learning was applied to the standard 1D-CNN pipeline with the aim of improving classification results on the real light curve datasets. Fig. 5 depicts the transfer learning setup that is selected when the EOS dataset is the goal/target dataset and the Blender simulation dataset is used to pre-train the model. In the pre-training step, the standard 1D-CNN architecture is trained using a light curve dataset, in this case the Blender simulation data, with the learned model weights from this training process saved. The weights from certain layers can then be transferred to another standard 1D-CNN model and used to initialise the training of this model on the goal/target dataset. In this configuration, only the weights for the convolutional layers of the network are transferred from the model trained on the simulated data and used for initialisation of further training on the goal dataset. The weights for the fully connected layers are initialised randomly (no transference from another model).

The fine-tuning step involves training the model on the goal/target dataset and updating the weights for all layers which are set to be trainable. There is the option to freeze (set to not trainable) some or all of the layers which have been initialised with weights from the pre-trained model. Freezing weights reduces training time and assists in prevent over-fitting to a specific dataset. However, it also limits the ability of the model to adapt to any differences between the dataset used for pre-training and the goal/target dataset. In Fig. 5 the goal/target dataset is the EOS dataset and the fine-tuning is set to update the weights for all layers. As some of the weights have been initialised from the pre-trained model, the fine-tuning typically requires significantly less data and time than the pre-training.

In the transfer learning process, both the number of layers transferred from the first to the second network and the number of layers that are frozen (not trainable) during fine-tuning are configurable parameters. In our work we have selected different combinations for these parameters depending on the dataset used for pre-training and the goal/target dataset. The selection process for these parameters is outlined in Section 3.4 and the parameters that were selected for each configuration are presented in Table 2.

3. Experimental setup

3.1. Blender simulated dataset

The Blender simulation environment has been used to generate a dataset containing 4500 light curves for 13 different satellite models. The light curves are sampled at 1 s intervals for the duration of a 30 min track, resulting 1800 data points per light curve. A number of TLEs were used during the initialisation step to simulate different phase angle and range conditions. Additionally, the starting orientation of the model was generated randomly and the rotation period was randomly selected for a range of integer values between 10 s and 300 s. The axis of rotation is randomly selected from one of the three

primary body axis of the object and a stable rotation using the selected rotation period is applied to this axis. These rotation profiles were selected to provide a wide range of different conditions, as well as to approximately represent spin states that were found to be typical of rotating debris objects, which is the focus of this body of research. Currently, tumbling and very slowly rotating objects have not been included in this dataset but will be an area of focus in future research, along with the rotation profiles of active objects.

3.2. EOS light curve dataset

The EOS light curve dataset consists of more than 900 light curves for 9 different configuration classes and 22 unique objects. During the tracking campaign there was an initial focus in tracking rocket bodies as they are typically large rotating objects making them easy to track and analyse. This has resulted in a skewed dataset with approximately 60% of the extracted light curves obtained from the three main rocket body configuration classes. Additionally, there are only a small number of unique RSOs for each class with 3 classes only containing one unique RSO. This will limit the ability of the model to learn generalisable patterns in the light curve that apply to a configuration class and may enable easier classification, based on rotation periods for specific objects, rather than light curve features. Table 4 in the Results section depicts the composition of the EOS dataset for the 9 different configuration classes.

3.3. MMT light curve dataset

The Multichannel Monitoring Telescope (MMT) [56] provides publicly available light curve observations of satellites and space debris.[5] This dataset was used as part of our training and evaluation of the classification algorithms and has been used previously in the literature for real light curve classification [13]. In this case, Linares separated the light curves into 3 general classes (satellites, rocket bodies and debris) and demonstrated that a 1D-CNN was able to outperform other classification methods on this task [13].

In this paper, the aim is to classify objects based on shape or object configuration so a different approach has been taken to the MMT dataset. The MMT dataset automatically applies phase dispersion minimisation to extracted light curves and uses this analysis to determine the rotational state of the object. Only light curves which have a determined rotation period and exceed 500 individual observations were downloaded for object classification. This ensured that the majority of light curves were relatively long with a stable rotation, resulting in repeating patterns in the light curves to enable the neural network to differentiate between the classes.

Using this criteria, approximately 40,000 light curves were downloaded from the MMT database for 154 unique objects. These were manually labelled into 27 separate configuration classes based on online information about the objects. Similar to the EOS dataset, this is a very unbalanced dataset with the largest three classes containing approximately 75% of the light curves.

A sub dataset was also extracted from the downloaded MMT light curves to create a balanced dataset of 500 light curves each from the largest 8 configuration classes. This dataset was used to test model performance on a more balanced dataset and is referred to as MMT Balanced in the Results section. Of the 8 classes selected, 7 are various types of rocket bodies and the remaining class is a box wing configuration.
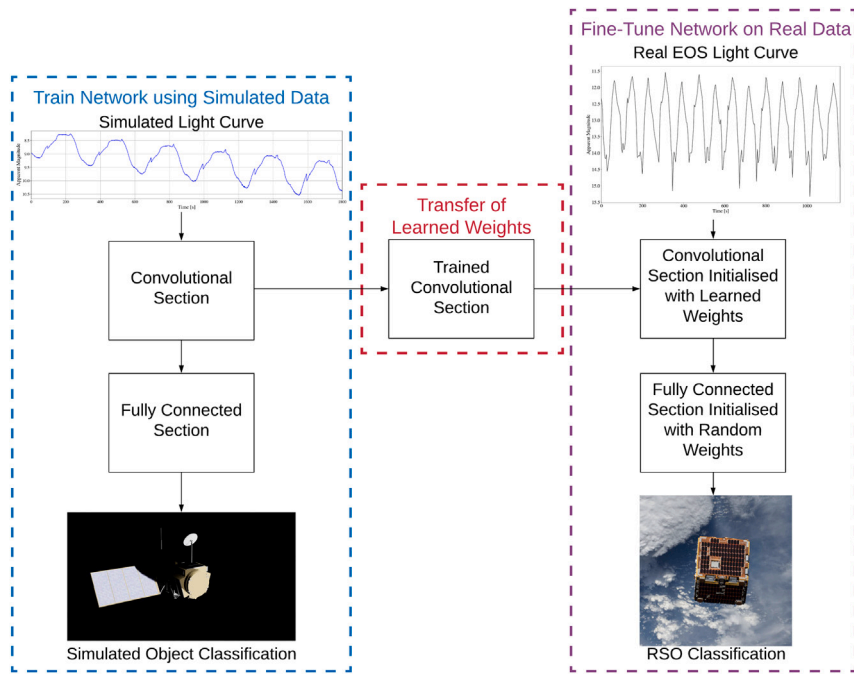
---

[5] https://mmt9.ru/satellites.

Fig. 5. Transfer Learning Process with EOS Dataset as Target/Goal Dataset and Blender Simulation Data used for Pre-Training.

Table 1
Training hyperparameters used during 1D-CNN training for each dataset.

Hyperparameter	Sim dataset	EOS dataset	MMT dataset	MMT balanced dataset
Training epochs	100	500	50	500
Batch size	256	128	256	64
Dropout	0.7	0.7	0.7	0.7
Starter learning rate	0.001	0.001	0.001	0.001
Exponential decay rate	0.6	0.8	0.8	0.9
Exponential decay steps	1000	300	50	100

3.4. Model setup and training hyperparameters

There are a number of training hyperparameters that are configurable during the model setup and training process. Table 1 displays the main hyperparameters that were selected for each dataset. These hyperparameters were mainly configured based on the size and number of classes in each dataset. The Blender simulated dataset and the MMT Dataset were both comparably large, relative to the other datasets, so they had a high batch size and a reduced amount of epochs required for training. During the training process, the goal was to finish training the model once it reached a stable loss value and prior to it over-fitting the training data.

K-fold cross fold validation was applied to effectively evaluate the trained models while utilising all of the available data. K was set to 5, with the model trained on 4 of the folds, whilst the remaining fold is held out as the validation set. This process is repeated until each fold has been used as the validation set resulting in 5 separately trained models. The overall evaluation is determined by taking the average of the recorded validation scores for each model on its validation set. This process results in a combined cross validation accuracy for the entire dataset.

As a result of the class imbalances in both the EOS and MMT datasets, class weights were incorporated into the loss function to increase the cost associated with the misclassification of the smaller classes. This was done on a per batch basis to improve the overall accuracy and was shown to be particularly effective for the EOS dataset.

The transfer learning process introduces additional hyperparameters into the model selection phase. The number of layers to transfer from the initial network to final network and the number of layers to freeze

Table 2
Layers transferred and fine-tunable for each pre-training and goal dataset combination.

Pre-train dataset	Fine-tune/Goal dataset	Layers transferred	Fine-tunable layers
Blender sim	EOS	2	5
MMT balanced	EOS	2	4
Blender sim	MMT balanced	4	1
EOS	MMT balanced	4	1

during the fine-tuning process on the goal dataset are both configurable parameters. A grid search was performed for each transfer learning combination (pre-training dataset and fine-tuning/goal dataset) that is presented in this paper. Based on this grid search the configuration that achieved the highest overall classification accuracy was selected and is recorded in Table 2.

4. Results

4.1. EOS dataset analysis

The EOS dataset was examined to gain a greater understanding of light curves and investigate some of the relationships found within the data. Fig. 6 depicts an example of a light curve extracted from EOS imagery using the outlined method. In Fig. 6, the left hand side of the figure depicts the extracted light curve with the apparent magnitude plotted on a time scale, while the right hand side depicts a Lomb–Scargle Periodogram of the light curve. This is the light curve of a Falcon 9 rocket body which is clearly rotating, resulting in an interesting pattern. The Lomb–Scargle periodogram displays a clear signal
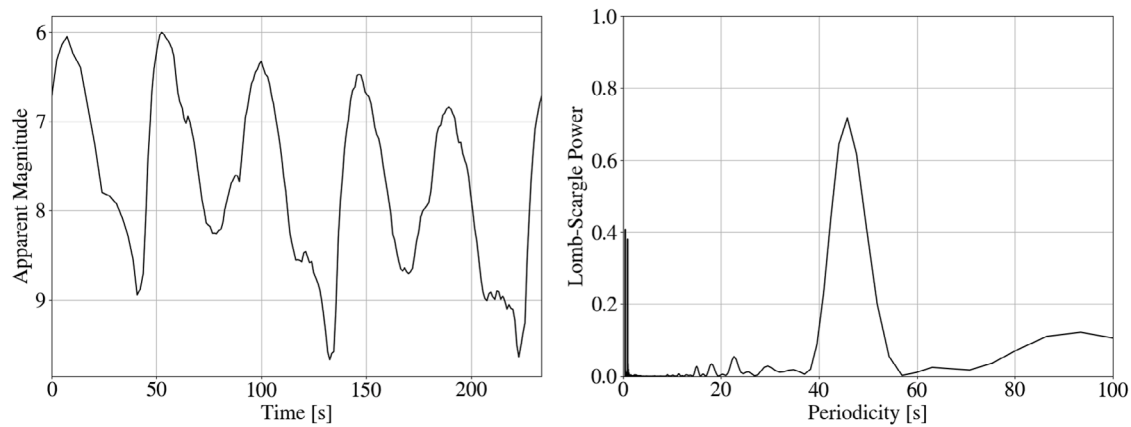
Fig. 6. Extracted Light Curve and Lomb–Scargle Analysis for a Falcon 9 Rocket Body; NORAD ID 40108. Rotation period is determined to be approximately 86 s with the peaks (decrease in apparent magnitude; object becomes brighter) in the brightness in the light curve corresponding to the side of the rocket body facing the observer and the troughs (increase in apparent magnitude; object becomes dimmer) when either the nozzle or the nose cone is facing the observer.

at approximately 43 s associated with a periodicity it has identified within the light curve.

Given the symmetrical nature of a rocket body, the signal identified by the Lomb–Scargle analysis can represent a half rotation of the rocket body rather than a full rotation. In this case, upon further inspection of the light curve, the signal is associated with a half rotation and subsequently, the rotation period is determined to be 86 s. This rotation period can be confirmed through the additional features that are present on the slightly dimmer first and third troughs of the light curve, which represents one full rotation, as opposed to the brighter second trough at approximately 80 s. The peaks in the apparent magnitude plot, when the object is brightest (low apparent magnitude), represent periods when the large side panels of the rocket body are reflecting light towards the observer. Meanwhile the troughs, where the object is becoming dimmer (increasing apparent magnitude), depict periods where either the nose cone or the nozzle of the rocket body are reflecting light towards the observer. Additionally, there is a general trend of decreasing magnitude in the light curve as time increase. This is a result is a combined effect of the both the object moving further away from the observer and an increase in the phase angle between the sun, the object and the observer.

The relationship between the phase angle and an object's magnitude is further examined in Fig. 7. The normalised apparent magnitude is depicted as a function of the phase angle for two different objects which were tracked on numerous occasions. The apparent magnitude has been normalised for the object's range in order to emphasise the effect of phase angle. The apparent magnitude of Etalon 1, a spherical calibration object with a number of retro reflectors, is shown to be highly dependent on the phase angle in Fig. 7(a). Although there are a few outlier tracks, there is a clear correlation between the increasing phase angle and the normalised magnitude, indicating the object is becoming dimmer.

For the more complicated box-wing shaped Optus B1 satellite, displayed in Fig. 7(b) there appears to be a similar general correlation. However, there is significantly more variability in the results due to the shape and rotation of the object. Even approaching a phase angle of 80°, large sections of the recorded measurements are brighter than the average brightness at a phase angle of 15°. These results demonstrate that the phase angle clearly influences the brightness of the object. However, for complex rotating RSOs with convex shapes and specular reflection, this relationship is difficult to accurately model without detailed information about the RSOs shape, material and attitude state.

RSOs that are in a stable rotation state typically produce light curves with a cyclic pattern, such as the light curve depicted in Fig. 6, which reduces the intraclass variability of their light curves. In contrast, objects that are tumbling or have rapidly changing rotation periods

Table 3
Comparison of different machine learning techniques: Mean 5-fold cross validation accuracy and standard deviation (each fold using 80% training and 20% holdout).

	SVM	FC	1D-CNN
Blender dataset	51.62 ± 0.49	60.40 ± 0.31	**84.40** ± 0.37
EOS dataset	44.07 ± 0.34	58.25 ± 0.86	**75.32** ± 1.11
MMT dataset	70.95 ± 0.02	75.29 ± 0.08	**90.71** ± 0.09
MMT balanced	51.41 ± 0.22	58.89 ± 0.32	**80.07** ± 0.33

Table 4
EOS dataset class analysis: Cross validation results for 1D-CNN model.

	F1 score	Precision	Recall	Unique RSOs	Light curves	% of dataset
Spherical	0.94	1.0	0.89	3	188	22.8
CZ-3B	0.75	0.77	0.73	3	166	20.6
ARIANE 5	0.88	0.80	0.97	4	213	20.1
Falcon 9	0.80	0.81	0.79	5	193	17.7
HS-601	0.42	0.36	0.5	2	78	8.3
GOES 8	0.44	0.67	0.33	2	58	5.6
DS-2000	0.5	0.33	1.0	1	19	2.0
DFH-3 Bus	0.8	1.0	0.67	1	15	1.6
Galileo 1	0.0	0.0	0.0	1	12	1.3

can produce a wide array of light curves making classification more difficult. Fig. 8 depicts the rotation period of 8 rocket bodies over a 10 month tracking interval between July 2019 and May 2020. The majority of rocket bodies appear to have a reasonably stable rotation period with small variations over the course of the tracking interval. The obvious outlier is object 39482, which showed large variations in rotation period, often over very short periods of time. This object is in a significantly lower elliptical orbit than the other rocket bodies and as such is more heavily affected by atmospheric drag. This result is promising for classification purposes as it suggests that a large portion or rocket bodies are in a stable rotation state.

4.2. Object classification

Three different types of supervised machine learning classification models were evaluated on the three light curve datasets. A Support Vector Machine (SVM) with a non-linear kernel was selected as an initial shallow learning baseline model. A dense four-layer Fully Connected (FC) neural network was also implemented to provide a comparison with the 1D-CNN. To enable testing on the whole dataset, k-fold cross validation was used (with 5 folds each having an 80/20 train/validation split). Given the stochastic nature of machine learning, this process was repeated 10 times. The resulting mean cross validation accuracy and standard deviation determined from the 10 evaluations are displayed in Table 3 for each model and dataset.
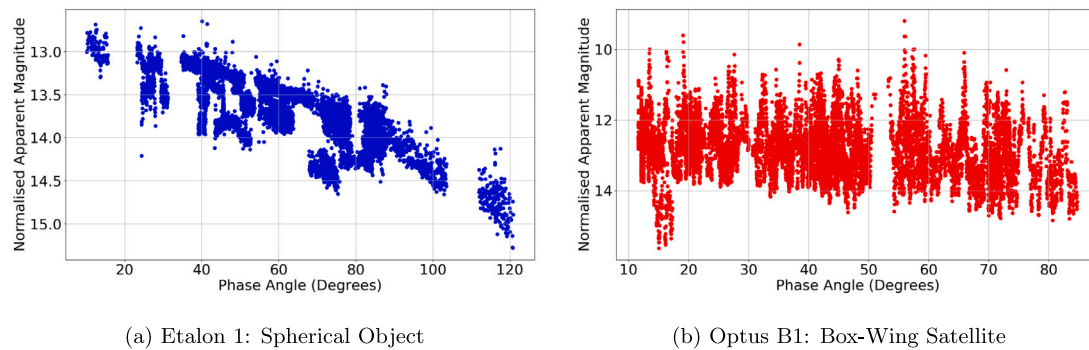
(a) Etalon 1: Spherical Object

(b) Optus B1: Box-Wing Satellite

Fig. 7. Comparison of all collected light curves for two different objects to examine magnitude dependence on phase angle. The apparent magnitude has been normalised for range to emphasise the effect of the phase angle on the object's magnitude.
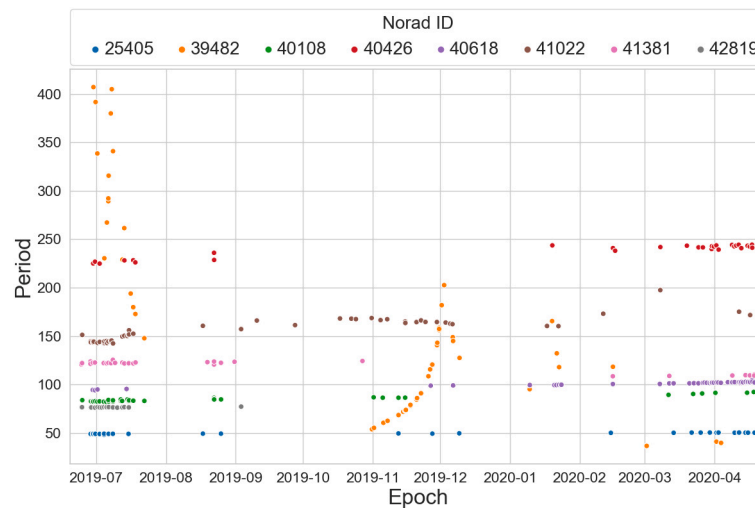

Fig. 8. Rotation Period Evolution of Rocket Bodies: Results are obtained from EOS light curve data collected over a period of approximately 10 months.

The SVM achieves the lowest classification accuracy on all three datasets, performing particularly poorly on the EOS dataset with a mean classification accuracy of 44.07%. The FC network improves on these results for each dataset with the largest improvement on the EOS dataset. The 1D-CNN significantly outperforms the FC network achieving an increase in classification accuracy of at least 15% on all three datasets.

The results of the 1D-CNN models on each of the datasets demonstrates that this is the best method for object shape classification on both the simulated Blender light curves and the real light curves extracted from telescope data. All models achieve their best performance on the MMT dataset and worst performance on the EOS dataset. The standard deviation is relatively low across all the model-dataset combinations indicating that the models are finding a relatively stable solution in the training process and increasing confidence in the results. Fig. 9(a) depicts the normalised confusion matrix for the 1D-CNN evaluated on the Blender simulation dataset. The model performs well with the majority of the 13 classes achieving an overall accuracy of 84.40%. The majority of misclassifications appear to be between classes that are similar.

The 1D-CNN achieved an overall cross validation accuracy of 75.3% on the 9 class configuration classification for the EOS real light curve dataset. Fig. 9(b) depicts the classification results in a normalised confusion matrix. The model achieves very high accuracy on the spherical class as these light curves are typically quite different from the remainder of the dataset. Similar to the results obtained from the simulated Blender dataset, there are a number of misclassifications between similarly shaped objects such as the three different rocket body classes (CZ-3B, ARIANE 5, FALCON 9).

Table 4 depicts the F1 score, precision and recall of the trained 1D-CNN model for each class in the EOS dataset. It also contains the number of light curves and unique objects per class. From this analysis it is clear that even with class weighting implemented, the model performance decreases significantly for smaller classes within the dataset. This is particularly noticeable for the 3 smallest classes which all have less than 20 instances present in the dataset.

The three different machine learning models all achieve a relatively high overall classification accuracy on the 25 class MMT dataset. As previously mentioned, this is a highly imbalanced dataset with 75% of the light curves part of the three largest classes. Thus, the models can achieve a relatively high classification accuracy by only performing well on the larger classes. Whilst the 1D-CNN significantly outperforms the other two models, achieving a cross validation classification accuracy of 90.71% it performs quite poorly on a number of the smaller classes. The class imbalance has an even greater impact on the results for classification using the FC network and the SVM. Both models demonstrate decent performance on the three largest classes, however they are only capable of achieving very poor results for the remainder of the classes.

On the MMT Balanced dataset the overall classification accuracy is significantly lower at 80.0% for the 1D-CNN model, when compared with the total MMT dataset. However, on the balanced dataset the model achieves similar performance on all classes as depicted in Fig. 10. The results for the MMT Balanced dataset are analysed in more detail in Table 5, which displays the F1 score, precision and recall as well as the number of unique RSOs for each class in this dataset. These results show that the model achieves the best performance on the LS-400 class,
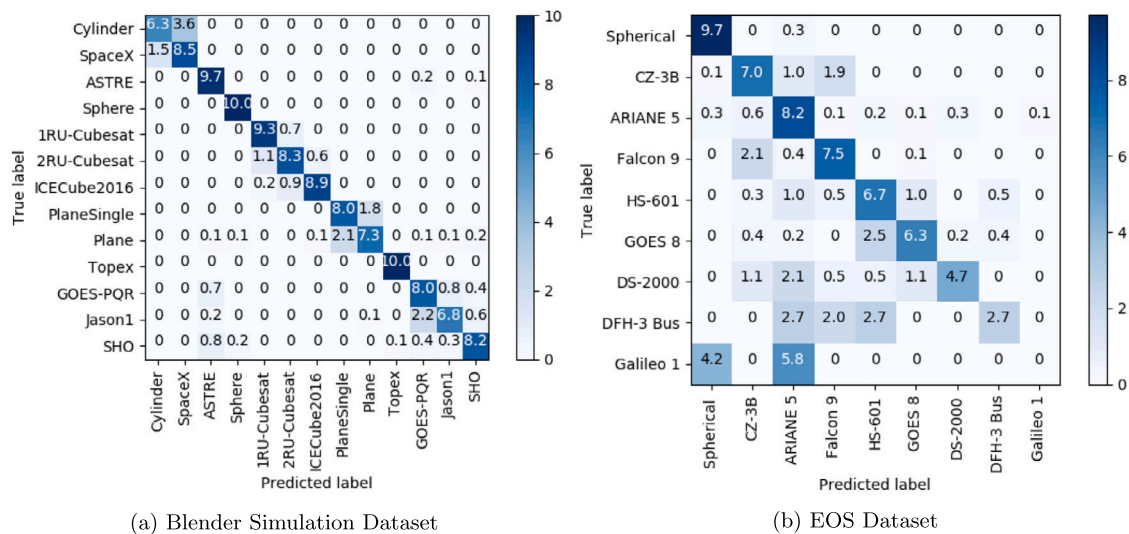
(a) Blender Simulation Dataset

(b) EOS Dataset

Fig. 9. Normalised Confusion Matrix Cross Validation Results for 1D-CNN: Results are normalised between 0 and 10 for each class, with 10 representing 100% of the light curves in a class, for view-ability and comparison purposes.
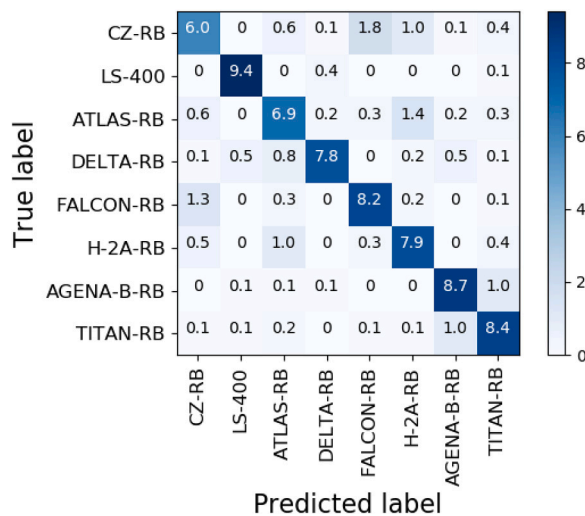

Fig. 10. MMT Balanced Normalised Confusion Matrix: 1D-CNN Cross Validation Results.

Fig. 11. Transfer Learning Results, Evaluation/Target Dataset: EOS Dataset.

Table 5
Class Analysis MMT Balanced: 1D-CNN Cross Validation Results.

	F1 Score	Precision	Recall	RSOs
CZ-RB	0.635	0.618	0.655	33
LS-400	0.943	0.941	0.945	32
ATLAS-RB	0.744	0.712	0.781	30
DELTA-RB	0.814	0.781	0.851	11
FALCON-RB	0.797	0.811	0.787	5
H-2A-RB	0.788	0.815	0.771	4
AGENA-B-RB	0.850	0.889	0.815	3
TITAN-RB	0.818	0.835	0.805	3

which is the only box-wing satellite class, with the remainder of classes being rocket body configurations. The model appears to achieve slightly lower performance on the two rocket body classes with the most unique RSOs (CZ-3B and ATLAS-RB).

4.3. Transfer learning

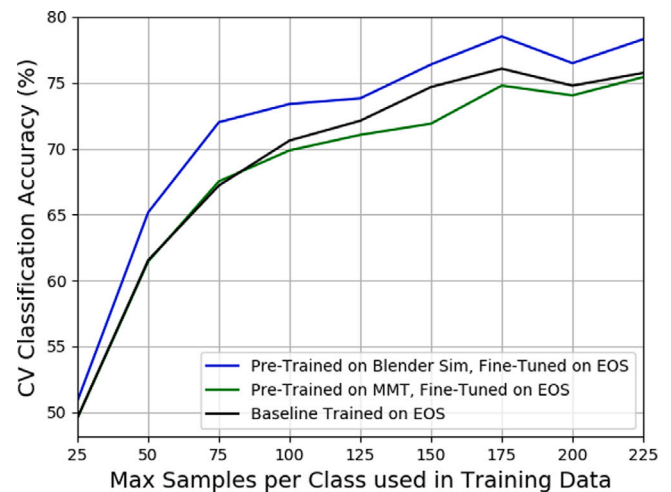Fig. 11 displays the classification accuracy of the transfer learning approach applied to the EOS dataset as a function of the training data size. The transfer learning models were pre-trained on either the Blender simulation dataset or the MMT dataset and then fine-tuned on the EOS dataset. For comparison, the baseline 1D-CNN model without transfer learning, labelled Baseline, is just trained on the EOS dataset. During the training/fine-tuning process on the EOS dataset, models were trained on a limited number of samples per class. For classes with less samples than the limit, all instances of the class were included. Each model is validated on the full EOS dataset to obtain the cross validation classification accuracy.

Transfer learning using a model previously trained on the Blender simulation data was found to be effective at increasing the overall classification accuracy on the EOS dataset for all dataset training sizes. The maximum improvement of 4.81%, compared to the baseline 1D-CNN model, occurs when the training data is restricted to 75 samples per class. When training on the full EOS dataset, transfer learning improves the classification accuracy by approximately 3% to 78.31%.

Interestingly, pre-training the model on the MMT dataset and then using transfer learning to apply this knowledge to the EOS dataset was found not to improve performance on the EOS dataset. In fact, as the class size of the training dataset increased it was found to reduce the overall classification accuracy of the model.
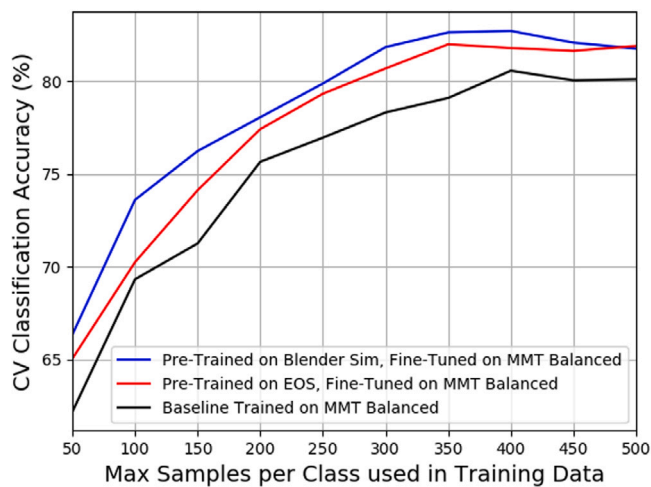
Fig. 12. Transfer Learning Results, Evaluation/Target Dataset: MMT Balanced Dataset.

Transfer learning was also applied to the MMT Balanced dataset with Fig. 12 depicting the results relative to training dataset size. Transfer learning improved the classification accuracy on the MMT Balanced dataset for both models pre-trained on the Blender simulated dataset and the EOS dataset. The model pre-trained on the Blender simulated dataset was found to have the best performance. Initial improvement for smaller training class sizes was 4 to 5% accuracy compared with the baseline model with no transfer learning. This improvement gradually decreased as the training data size increased to approximately 2% when the full dataset is used in training. Similar results were found for the model that was pre-trained on the EOS dataset, however the improvements relative to the baseline were not as significant for reduced training data size.

5. Discussion

The results presented in this research demonstrate that machine learning models are an effective tool for performing RSO classification from both simulated and real light curves. This paper provides both the framework for and a demonstration of a generalised method for object shape classification from real light curve data. Previous research by Linares and Furfaro [13] had shown traditional machine learning methods such as RDF and SVM to be effective for classifying objects from simulated light curves generated using the FFM. However, they found that a 1D-CNN significantly outperformed these methods when attempting to classify real light curves as either payloads, rocket bodies or debris. The authors theorised that this was due to the increase in complexity for the real data compared to the simulated data. They proposed that the many layer structure of the 1D-CNN enabled it to learn complex decision boundaries and distinguish between the 3 classes for the real data.

This theory is supported by the object classification results presented in this research, with the 1D-CNN achieving a significantly higher classification accuracy than both the FC neural network and the SVM on each of the real datasets. Similar results occurred for the object classification on the Blender simulated dataset. This suggests that the Blender simulated dataset presented in this research is more complex, with less delineation between class boundaries, than the FFM dataset that was used in previous research [13,57].

The normalised confusion matrix depicted in Fig. 9(a) for the Blender dataset demonstrates that the model has some difficulty in distinguishing between similar objects. In particular, approximately 35% of the Cylinder light curves are misclassified as the SpaceX rocket body and there is also a significant percentage of SpaceX light curves

being misclassified as the Cylinder. This is to be expected, as these models are identical, with the exception of the nose cone and nozzle on the SpaceX model. These features would not be visible to the observer for all rotation axis and orientation combinations so there would be a number of very similar light curves between the two classes. Similarly, there are some misclassifications between the different types of cubesats as well as the two different versions of the flat plane. However, for the majority of cases, the model is not misclassifying objects as a vastly different shape. This increases the confidence that the model is able to distinguish between generalised shapes as well as specific object configurations.

The results on the MMT Balanced dataset indicated that the model was generally able to distinguish between different types of rocket bodies. It is interesting to note that when comparing the different types of rocket body in Table 5, as the number of RSOs decreased for a class, the model performance increased. This suggests that the model may be learning specific features about individual RSOs, such as the rotation period, and using this information to assist in classification of the objects.

One of the major trends evident for both of the real datasets was the class imbalances and the clear correlation between number of training samples and model performance on a class. Both datasets were heavily weighted towards large pieces of debris, primarily rocket bodies, as they are more easily tracked and often rotating. Future tracking campaigns with EOS will focus on collecting data from a broader range of objects. Due to the time and cost of collecting the real data, the authors will also explore the idea of simulating specific objects for classes with limited real light curve data using the Blender simulation environment. These simulated light curves will then be incorporated in the training data in an attempt to improve results on the real light curves. The difficulty with this approach is developing an exact model for a specific object in Blender due to the limited amount of information available for most space debris objects.

Transfer learning was found to be an effective method to increase model performance and the overall classification accuracy. On both real light curve datasets, transfer learning appeared to be most effective when the size of the training dataset for the fine-tuning section of the model was restricted. As the amount of training data increased, the benefit of transfer learning was typically seen to diminish. It is expected that as more training data becomes available for a specific dataset this trend would continue until the baseline 1D-CNN model, without transfer learning, equalled or outperformed the 1D-CNN with transfer learning applied. However, as discussed previously, it is difficult to obtain and label real light curve data, particularly for a wide range of classes. This research has demonstrated that transfer learning can be used to improve the performance of small real world light curve datasets and reduces the requirement for long intensive tracking campaigns.

It is interesting to note that pre-training on the simulated Blender dataset was found to be more effective than pre-training on either of the real datasets. This is likely a result of the Blender dataset being significantly larger than either of the real datasets and having a wider range of object models, whilst the two real datasets are heavily skewed towards rocket body classes. It could also be due to the different data characteristics between the two real datasets, with the MMT dataset typically having a significantly higher sampling rate and typically shorter tracks than the EOS dataset.

It was found that when the target dataset was the MMT Balanced dataset, more of the layers were transferred across from the pre-trained networks. This is highlighted in Table 2, which depicts the number of layers that were transferred from the pre-trained network to the fine-tune network for each of the transfer learning configurations. Four layers transferred signifies that both convolutional layers and the first two fully connected layers were transferred across. This implies that as well as the general features in the lower layers, more specific features in the higher layers that are learnt during the pre-training are being

utilised in the final network on the goal dataset. Additionally, only the final output layer was set to be trainable during the fine-tuning process, which reduces fine-tuning time and computational requirements. In contrast, when the EOS dataset was the target dataset for the transfer learning, only the two convolutional layers were transferred across from the pre-trained networks and the majority of the layers were set to be trainable during the fine-tuning process. It is possible that this is required as a result of the unbalanced classes in the EOS dataset compared with the MMT Balanced and Blender simulation datasets.

Unlike previous results on simulated datasets in the literature, the classification results on all 4 datasets, even with the incorporation of transfer learning, did not achieve close to 100% accuracy. In terms of the simulated data, it is expected that this result occurs due to the increase in variability in the input parameters used to generate our simulated light curves. Previous simulations based on the FFM used the same initial epoch, initial quaternion and angular rate for all light curves resulting in a less complex classification task [13,57]. In the real world, it is expected that a certain percentage of light curves will not provide enough information for classification to be determined. This is as a result of the relationship between the axis of orientation and the complexity of the object as well as the limitations of the information observable in 1D light curve data. Additionally, for the proposed method to be effective, the real light curves must be accurate enough that the variations in magnitude as a result of the objects rotation are distinct from the variations caused by noise.

6. Conclusion

This paper has investigated a data driven approach to RSO characterisation through the application of machine learning techniques. On all four datasets presented in this paper, the 1D-CNN significantly outperformed both FC networks and SVMs. The results on both simulated and real data demonstrate that 1D-CNNs are able to learn features from the light curve data that enable shape classification of rotating RSOs. The 1D-CNN achieved a classification accuracy of 75.32% on a 9 class classification task on the EOS dataset and an accuracy of 80.07% on an 8 class classification task on the MMT Balanced dataset obtained from publicly available data. To the best of the authors' knowledge, these results represent the first attempts to classify objects as specific shapes based on real observational light curve data using machine learning.

Additionally, the transfer learning results presented in this paper illustrate that it can be used as an effective tool to improve the classification accuracy on real light curve datasets. Models initially trained on the Blender simulated light curve data and then fine-tuned on real light curve data outperformed models that were exclusively trained on the real light curve data. This implies that the Blender based simulation environment produces high fidelity light curve data which encapsulates many of the features found in real light curve data. Performance improvement on the full EOS dataset was approximately 3% whilst the improvement on the MMT Balanced dataset was approximately 2%.

Transfer learning was shown to improve results more significantly when the amount of training data available for the target dataset was small and to gradually decrease in effectiveness as training data increased. This result implies that organisations are not required to spend significant resources building up a large real light curve dataset to ensure that machine learning techniques are effective. Instead, they can use transfer learning to train a model on a simulated dataset and then fine-tune on a small real light curve dataset and still achieve comparable results to training on a larger real light curve dataset.

Future work will investigate methods to further improve model performance on an increased number of object classes, as well as examining more difficult cases when an object's rotation is not visible from a single sensor. This could be done by tracking an object simultaneously from multiple sensor locations, in order to gain multiple viewing angles of the object. Another possibility is incorporating different types of sensors, such as laser ranging or infrared imagery, which provide

different information about the object. For real time utilisation, a model certainty score could be associated with any object classification prediction, using a method such as MC-Dropout with follow up tracking or alternative sensor tracking required when the prediction certainty is below a certain value. Further research will also investigate different types of neural networks with both recurrent neural networks and temporal convolutional networks showing promise on sequential time series data.

Declaration of competing interest

Acknowledgements

References

[1] Donald J. Kessler, Burton G. Cour-Palais, Collision frequency of artificial satellites: The creation of a debris belt, J. Geophys. Res.: Space Phys. 83 (A6) (1978) 2637–2646, http://dx.doi.org/10.1029/JA083iA06p02637, URL http://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JA083iA06p02637.

[2] David A. Vallado, Wayne D. McClain, Fundamentals of Astrodynamics and Applications, fourth ed., Microcosm Press, Hawthorne, California, 2013.

[3] Thomas Schildknecht, Optical surveys for space debris, Astron. Astrophys. Rev. 14 (1) (2007) 41–111, http://dx.doi.org/10.1007/s00159-006-0003-9, URL http://link.springer.com/10.1007/s00159-006-0003-9.

[4] Andrew D. Dianetti, Ryan Weisman, John L. Crassidis, Observability analysis for improved space object characterization, J. Guid. Control Dyn. 41 (1) (2018) 137–148, http://dx.doi.org/10.2514/1.G002229, URL https://arc.aiaa.org/doi/10.2514/1.G002229.

[5] Moriba K Jah, Ronald A Madler, Satellite characterization: Angles and light curve data fusion for spacecraft state and parameter estimation, in: Proceedings of AMOS Conference, Maui, Hawaii, 2007, pp. 1–10.

[6] Charles J. Wetterer, Moriba K. Jah, Attitude determination from light curves, J. Guid. Control Dyn. 32 (5) (2009) 1648–1651, http://dx.doi.org/10.2514/1.44254, URL http://arc.aiaa.org/doi/10.2514/1.44254.

[7] Richard Linares, Moriba K Jah, John L Crassidis, Inactive space object shape estimation via astrometric and photometric data fusion, Adv. Astronaut. Sci. 143 (2012) 217–232.

[8] Richard Linares, Moriba K. Jah, John L. Crassidis, Christopher K. Nebelecky, Space object shape characterization and tracking using light curve and angles data, J. Guid. Control Dyn. 37 (1) (2014) 13–25, http://dx.doi.org/10.2514/1.62986, URL http://arc.aiaa.org/doi/10.2514/1.62986.

[9] Doyle Hall, Brandoch Calef, Keith Knox, Mark Bolden, Paul Kervin, Separating attitude and shape effects for non-resolved objects, in: Proceedings of AMOS Conference, Maui, Hawaii, 2007, pp. 1–12.

[10] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, first ed., in: Adaptive Computation and Machine Learning Series, MIT Press, Cambridge, MA, 2017.

[11] François Chollet, Deep Learning with Python, Manning, Shelter Island, NY, 2018, OCLC: 1019988472.

[12] Roberto Furfaro, Richard Linares, Vishnu Reddy, Space debris identification and characterization via deep meta-learning, 2019, p. 9.

[13] Richard Linares, Roberto Furfaro, Vishnu Reddy, Space objects classification via light-curve measurements using deep convolutional neural networks, J. Astronaut. Sci. (2020) http://dx.doi.org/10.1007/s40295-019-00208-w, URL http://link.springer.com/10.1007/s40295-019-00208-w.

[14] M Kaasalainen, Optimization methods for asteroid lightcurve inversion I. shape determination, Icarus 153 (1) (2001) 24–36, http://dx.doi.org/10.1006/icar.2001.6673, URL http://linkinghub.elsevier.com/retrieve/pii/S0019103501966734.

[15] M Kaasalainen, Optimization methods for asteroid lightcurve inversion II. the complete inverse problem, Icarus 153 (1) (2001) 37–51, http://dx.doi.org/10.1006/icar.2001.6674, URL http://linkinghub.elsevier.com/retrieve/pii/S0019103501966746.

[16] Brandoch Calef, John Africano, Brian Birge, Doyle Hall, Paul Kervin, in: Victor L. Gamiz, Paul S. Idell, Marija S. Strojnik (Eds.), Photometric Signature Inversion, San Diego, California, USA, 2006, p. 63070E, http://dx.doi.org/10.1117/12.683015, URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.683015.

[17] Marcus J. Holzinger, Kyle T. Alfriend, Charles J. Wetterer, K. Kim Luu, Chris Sabol, Kris Hamada, Photometric attitude estimation for agile space objects with shape uncertainty, J. Guid. Control Dyn. 37 (3) (2014) 921–932, http://dx.doi.org/10.2514/1.58002, URL http://arc.aiaa.org/doi/10.2514/1.58002.

[18] Richard Linares, John L. Crassidis, Space-object shape inversion via adaptive hamiltonian Markov chain Monte Carlo, J. Guid. Control Dyn. 41 (1) (2018) 47–58, http://dx.doi.org/10.2514/1.G002296, URL https://arc.aiaa.org/doi/10.2514/1.G002296.

[19] T Schildknecht, R Musci, M Ploner, G Beutler, W Flury, J Kuusela, J de Leon Cruz, L de Fatima Dominguez Palmero, Optical observations of space debris in GEO and in highly-eccentric orbits, Space Debris, Adv. Space Res. 34 (5) (2004) 901–911, http://dx.doi.org/10.1016/j.asr.2003.01.009, URL http://www.sciencedirect.com/science/article/pii/S0273117704000651.

[20] T Schildknecht, R Musci, C Früh, M Ploner, Color photometry and light curve observations of space debris in GEO, in: Proceedings of AMOS Conference, Maui, Hawaii, 2008, pp. 1–6.

[21] Jiří Šilha, Jean-Noël Pittet, Michal Hamara, Thomas Schildknecht, Apparent rotation properties of space debris extracted from photometric measurements, Adv. Space Res. 61 (3) (2018) 844–861, http://dx.doi.org/10.1016/j.asr.2017.10.048, URL https://linkinghub.elsevier.com/retrieve/pii/S027311771730786X.

[22] N. Koshkin, L. Shakun, E. Korobeynikova, S. Melikyants, S. Strakhova, V. Dragomiretsky, A. Ryabov, T. Golubovskaya, S. Terpan, Monitoring of space debris rotation based on photometry, Odessa Astron. Publ. 31 (2018) 179–185, http://dx.doi.org/10.18524/1810-4215.2018.31.147807, URL http://oap.onu.edu.ua/article/view/147807.

[23] Jiří Šilha, Stanislav Krajčovič, Matej Zigo, Juraj Tóth, Danica Žilková, Pavel Zigo, Leonard Kornoš, Jaroslav Šimon, Thomas Schildknecht, Emiliano Cordelli, Alessandro Vananti, Harleen Kaur Mann, Abdul Rachman, Christophe Paccolat, Tim Flohrer, Space debris observations with the slovak AGO70 telescope: Astrometry and light curves, Adv. Space Res. (2020) S0273117720300727, http://dx.doi.org/10.1016/j.asr.2020.01.038, URL https://linkinghub.elsevier.com/retrieve/pii/S0273117720300727.

[24] Conor J. Benson, Daniel J. Scheeres, William H. Ryan, Eileen V. Ryan, Nicholas A. Moskovitz, GOES spin state diversity and the implications for GEO debris mitigation, Acta Astronaut. 167 (2020) 212–221, http://dx.doi.org/10.1016/j.actaastro.2019.11.004, URL https://linkinghub.elsevier.com/retrieve/pii/S0094576519313815.

[25] Michael Howard, Bernie Klem, Joe Gorman, RSO characterization with photometric data using machine learning, in: Advanced Maui Optical and Space Surveillance Tech. Conf.(AMOS), Maui, Hawaii, 2015, p. 10.

[26] Navraj Singh, Joshua T Horwood, Jeffrey M Aristoff, Jeremy Murray-Krezan, Athena: A data-driven anomaly detection and space object classification tool for SSA, Adv. Astronaut. Sci. 158 (2016) 4097–4116.

[27] Bin Jia, Khanh Pham, Erik Blasch, Dan Shen, Zhonghai Wang, Genshe Chen, Space object classification using fused features of time series data, in: Advanced Maui Optical and Space Surveillance Tech. Conf.(AMOS), 2017, pp. 1–12.

[28] W.D. Bennette, K. Zeliff, J. Raquepas, Classification of objects in geosynchronous earth orbit via light curve analysis, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–6, http://dx.doi.org/10.1109/SSCI.2017.8280966.

[29] Mahmoud Khalil, Elena Fantino, Panos Liatsis, Evaluation of oversampling strategies in machine learning for space debris detection, in: 2019 IEEE International Conference on Imaging Systems and Techniques, IST, IEEE, Abu Dhabi, United Arab Emirates, 2019, pp. 1–6, http://dx.doi.org/10.1109/IST48021.2019.9010217, URL https://ieeexplore.ieee.org/document/9010217/.

[30] Roberto Furfaro, Richard Linares, David Gaylor, Moriba Jah, Ramona Walls, Resident space object characterization and behavior understanding via machine learning and ontology-based Bayesian networks, in: Advanced Maui Optical and Space Surveillance Tech. Conf., (AMOS), Maui, Hawaii, 2016.

[31] Roberto Furfaro, Richard Linares, Vishnu Reddy, Space objects classication via light-curve measurements: Deep convolutional neural networks and model-based transfer learning, in: Proceedings of AMOS Conference, Maui, Hawaii, 2018, pp. 1–17.

[32] B. Jia, K.D. Pham, E. Blasch, Z. Wang, D. Shen, G. Chen, Space object classification using deep neural networks, in: 2018 IEEE Aerospace Conference, 2018, pp. 1–8, http://dx.doi.org/10.1109/AERO.2018.8396567.

[33] Yurong Huo, Zhi Li, Yuqiang Fang, Feng Zhang, Classification for geosynchronous satellites with deep learning and multiple kernel learning, Appl. Opt. 58 (21) (2019) 5830, http://dx.doi.org/10.1364/AO.58.005830, URL https://www.osapublishing.org/abstract.cfm?URI=ao-58-21-5830.

[34] Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 25, Curran Associates, Inc., 2012, pp. 1097–1105, URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[35] Alexander Selvikvåg Lundervold, Arvid Lundervold, An overview of deep learning in medical imaging focusing on MRI, Z. Med. Phys. 29 (2) (2019) 102–127, http://dx.doi.org/10.1016/j.zemedi.2018.11.002, URL https://linkinghub.elsevier.com/retrieve/pii/S0939388918301181.

[36] Mingsheng Long, Yue Cao, Jianmin Wang, Michael I Jordan, Learning transferable features with deep adaptation networks, p. 9.

[37] Matthew D. Zeiler, Rob Fergus, Visualizing and understanding convolutional networks, in: David Fleet, Tomas Pajdla, Bernt Schiele, Tinne Tuytelaars (Eds.), Computer Vision, ECCV 2014, in: Series Title: Lecture Notes in Computer Science, vol. 8689, Springer International Publishing, Cham, 2014, pp. 818–833, http://dx.doi.org/10.1007/978-3-319-10590-1_53, URL http://link.springer.com/10.1007/978-3-319-10590-1_53.

[38] Dimitrios Marmanis, Mihai Datcu, Thomas Esch, Uwe Stilla, Deep learning earth observation classification using imagenet pretrained networks, IEEE Geosci. Remote Sens. Lett. 13 (1) (2016) 105–109, http://dx.doi.org/10.1109/LGRS.2015.2499239, URL http://ieeexplore.ieee.org/document/7342907/.

[39] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, Laurens van der Maaten, Exploring the limits of weakly supervised pretraining, in: Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, Yair Weiss (Eds.), Computer Vision, ECCV 2018, in: Series Title: Lecture Notes in Computer Science, vol. 11206, Springer International Publishing, Cham, 2018, pp. 185–201, http://dx.doi.org/10.1007/978-3-030-01216-8_12, URL http://link.springer.com/10.1007/978-3-030-01216-8_12.

[40] Lloyd Windrim, Arman Melkumyan, Richard J. Murphy, Anna Chlingaryan, Rishi Ramakrishnan, Pretraining for hyperspectral convolutional neural network classification, IEEE Trans. Geosci. Remote Sens. 56 (5) (2018) 2798–2810, http://dx.doi.org/10.1109/TGRS.2017.2783886, URL http://ieeexplore.ieee.org/document/8245897/.

[41] David Malmgren-Hansen, Anders Kusk, Jorgen Dall, Allan Aasbjerg Nielsen, Rasmus Engholm, Henning Skriver, Improving SAR automatic target recognition models with transfer learning from simulated data, IEEE Geosci. Remote Sens. Lett. 14 (9) (2017) 1484–1488, http://dx.doi.org/10.1109/LGRS.2017.2717486, URL http://ieeexplore.ieee.org/document/7968358/.

[42] Siyu Shao, Stephen McAleer, Ruqiang Yan, Pierre Baldi, Highly accurate machine fault diagnosis using deep transfer learning, IEEE Trans. Ind. Inf. 15 (4) (2019) 2446–2455, http://dx.doi.org/10.1109/TII.2018.2864759, URL https://ieeexplore.ieee.org/document/8432110/.

[43] David Isele, Akansel Cosgun, Transferring autonomous driving knowledge on simulated and real intersections, 2017, arXiv:1712.01106, arXiv:1712.01106 [cs], URL http://arxiv.org/abs/1712.01106.

[44] Bichen Wu, Alvin Wan, Xiangyu Yue, Kurt Keutzer, Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D lidar point cloud, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, Brisbane, QLD, 2018, pp. 1887–1893, http://dx.doi.org/10.1109/ICRA.2018.8462926, URL https://ieeexplore.ieee.org/document/8462926/.

[45] Michael Ashikhmin, Peter Shirley, An Anisotropic Phong Light Reflection Model, Technical Report, Journal of Graphics Tools, 2000.

[46] Robert L Cook, A reflectance model for computer graphics, ACM Trans. Graph. 1 (1) 18.

[47] Charles J. Wetterer, Richard Linares, John L. Crassidis, Thomas M. Kelecy, Marek K. Ziebart, Moriba K. Jah, Paul J. Cefola, Refining space object radiation pressure modeling with bidirectional reflectance distribution functions, J. Guid. Control Dyn. 37 (1) (2014) 185–196, http://dx.doi.org/10.2514/1.60577, URL http://arc.aiaa.org/doi/10.2514/1.60577.

[48] James Allworth, Lloyd Windrim, Jeffrey Wardman, Daniel Kucharski, James Bennett, Mitch Bryson, Development of a high fidelity simulator for generalised photometric based space object classication using machine learning, in: Proceedings of the International Astronautical Congress, Vol. 2019, Washington, p. 14.

[49] David Vallado, Paul Crawford, SGP4 orbit determination, in: AIAA/AAS Astrodynamics Specialist Conference and Exhibit, American Institute of Aeronautics and Astronautics, Honolulu, Hawaii, 2008, http://dx.doi.org/10.2514/6.2008-6770, URL http://arc.aiaa.org/doi/10.2514/6.2008-6770.

[50] John Bangert, Wendy Puatua, George Kaplan, Jennifer Bartlett, William Harris, Amy Fredericks, Alice Monet, User's guide to NOVAS version c3.1, p. 124.

[51] Kristen Erikson, NASA 3D Models, National Aeronautics and Space Admistration, URL https://nasa3d.arc.nasa.gov/models/sort_name.

[52] James Bennett, Michael Lachut, Daniel Kucharski, Sven Flegel, Marek Möckel, James Allworth, David Kooymans, Alex Pollard, Craig Smith, Joseph O'Leary, Hansani Kankanamalage, Richard Samuel, Progress in a new conjunction and threat warning service for Space Situational Awareness, in: Advanced Maui Optical and Space Surveillance Tech. Conf.(AMOS), Maui, Hawaii, 2018, pp. 1–12.

[53] W. Romanishin, An Introduction to Astronomical Photometry Using CCDs, University of Oklahoma, 2006.

[54] N. Zacharias, C.T. Finch, T.M. Girard, A. Henden, J.L. Bartlett, D.G. Monet, M.I. Zacharias, The fourth us naval observatory ccd astrograph catalog (UCAC4), Astron. J. 145 (2) (2013) 44, http://dx.doi.org/10.1088/0004-6256/145/2/44, URL https://iopscience.iop.org/article/10.1088/0004-6256/145/2/44.

[55] Jacob T. VanderPlas, Understanding the Lomb–Scargle Periodogram, Astrophys. J. Suppl. Ser. 236 (1) (2018) 16, http://dx.doi.org/10.3847/1538-4365/aab766, URL https://iopscience.iop.org/article/10.3847/1538-4365/aab766.

[56] G.M. Beskin, S.V. Karpov, A.V. Biryukov, S.F. Bondar, E.A. Ivanov, E.V. Katkova, N.V. Orekhova, A.V. Perkov, V.V. Sasyuk, Wide-field optical monitoring with Mini-MegaTORTORA (MMT-9) multichannel high temporal resolution telescope, Astrophys. Bull. 72 (1) (2017) 81–92, http://dx.doi.org/10.1134/S1990341317030105, URL http://link.springer.com/10.1134/S1990341317030105.

[57] Richard Linares, Roberto Furfaro, Space object classification using deep convolutional neural networks, in: Information Fusion (FUSION), 2016 19th International Conference on, IEEE, 2016, pp. 1140–1146.