A Machine Learning Approach to Space Debris Characterisation and Classification

James Allworth BE (Hons 1), BCom

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

THE UNIVERSITY OF
SYDNEY

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

Submitted January 2022; revised June 2022

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

James Allworth

30 June 2022

Authorship Attribution Statement

This thesis contains material that has previously been published in [3] and [4]. For both of these publications, I set up the experiments, extracted the data, analysed the data and wrote the initial drafts of the papers. Coauthors assisted with the initial experimental design and the editing process before submission for publication. Specifically, Section 3.2 and Section 3.3 contain material that was published in both [3] and [4]. Section 3.4 contains material that was originally published in [3]. Section 4.2 and Section 4.3 contain material that was originally published in [4].

James Allworth

30 June 2022

Abstract

Space debris is becoming an increasingly prevalent issue through a combination of the recent rise in the accessibility of space and the current difficulty in actively removing space debris from orbit. As a result of the high relative velocity between orbital objects and the difficulty in accurately maintaining the state of all objects, space debris poses a significant collision risk to active satellites. Current risk mitigation strategies rely on space situational awareness, which focuses on tracking space objects and then using orbital mechanics to predict their future states. Satellite operators can then be informed of any potential future conjunctions between debris and active satellites. For this method to be effective, the information provided to satellite operators must be both accurate and timely. The accuracy of these predictions is limited by a lack of knowledge about the physical characteristics of a large proportion of space debris, such as shape, size, orientation, mass and material properties.

This thesis outlines a data-driven approach to space object characterisation through the application of neural networks to light curves extracted from non-resolved ground based optical observations. A light curve is a temporal history of an object's brightness and has been shown to contain information about the object's physical characteristics. Neural networks are known to be more effective when they are trained on a well-labelled dataset with a large number of examples in order for the complex non-linear relationships within the data to be learned. This has previously been a limiting factor when applying deep learning to light curve based object classifications as light curves are difficult to obtain and label, so real world datasets remain relatively small.

This thesis presents simulation-based transfer learning as a method for overcoming this limitation and improving shape classification performance on real world light curve datasets. A high-fidelity light curve simulation environment has been developed in order to generate a large, well-labelled simulated light curve dataset that encapsulates the features present in real world light curves. The simulated dataset is used for the initial training of a one dimensional convolutional neural network. This pre-trained model is then transferred to the smaller real world light curve dataset where it is fine-tuned. This process is shown to boost the performance of the model on the real world dataset, compared to models that are trained only on the real world dataset. To further improve performance on challenging cases, a framework for effectively combining multiple light curve observations of a single object is also developed. Finally, in order to utilise this framework efficiently, a targeted scheduling process has been developed, using uncertainty quantification of the neural network output, to selectively prioritise the re-observation of challenging cases and thus reduce misclassifications.

Acknowledgements

I would like to start by sincerely thanking my supervisors Mitch Bryson, James Bennett and Lloyd Windrim for the time and effort that they have put in to this thesis. The guidance and insight that you have all provided over the last four years has been invaluable.

Thank you to Electro Optic Systems for providing both financial support and operational resources to facilitate this research. It has been very enjoyable working with the team in Canberra and the access to real world data has resulted in more interesting and impactful research.

While almost every member of EOS Space Systems has contributed in some manner, I would like to thank a few people specifically. In particular, to Michael Lachut, thank you for helping to sanity check a lot of my ideas and always being willing to provide feedback. To Jeffrey Wardman, thanks for being willing to share your machine learning experience during the early stages of my thesis as well as your appreciation for my questionable music choices. To Lexa, thank you for providing some much needed humour in the office and for helping to ensure that everything ran smoothly. To James Webb, thanks for helping to get the development network over the line and for your words of wisdom along the way. To the software team, Alex, Robert, Harris, Chun, Matt, Visa and David, thanks for your assistance in the various projects that we worked on together as well as your hard work in helping to collect and enable access to the telescope imagery.

To my friends and colleagues at the Australian Centre for Field Robotics, thanks for making the past few years an enjoyable period. A special thank you to my family, who are my biggest supporters and have always been there for me. Finally, I would like to thank my incredible partner Ella, for her constant support as well as patience and understanding. This thesis would not have been possible without you.

"You'll never get bored when you try something new.
There's really no limit to what you can do."

Dr. Seuss

Contents

List of Figures

List of Tables

Nomenclature

List of Symbols

A_{ap}	area of the aperture
B	sum of the pixel intensity of an object extracted from a Blender rendered image
b	bias parameter
dt	difference in time between two observations
C	the number of unique shape classes for a given dataset
d	distance
el_{min}	minimum elevation angle of space object from an observing sensor
f	a predictive function (model) that maps an input to an output
h_{min}	minimum observable altitude of space object for an observing sensor
I_m	instrumental magnitude
l	length of convolutional filter
m	apparent magnitude
N_{ap}	sum of the pixel counts in the aperture
n	number of distinct light curves in a group
n_c	number of unique classes
n_f	number of filters
r	range of the space object from an observing sensor
R_e	radius of the Earth
s	softmax score vector output from neural network
S_{sky}	background sky signal
t	time step where observations are recorded
T	total number of unique time steps for a specified time series
t_{exp}	the exposure time of the image
v	an output feature map from a neural network
$\mathbf{x}$	single light curve to be input into model
$\mathbf{x}_G$	group of light curves to be input into a model
y	output class predicted by the model
z	an observation containing four separate measurements (apparent magnitude, time, phase angle and range) recorded at a single time step

σ	standard deviation
ϕ_i	phase angle
ϕ_q	photon flux
ω	convolutional filter

List of Acronyms

AMR	area-to-mass ratio
AS	Ashihmin and Shirley
BRDF	bidirectional reflectance distribution function
CCD	charge-coupled device
CCR	corner cube reflector
CNN	convolutional neural network
1D-CNN	one-dimensional convolutional neural network
C-RNN	convolutional-recurrent neural network
EOS	Electro Optic Systems
FC	fully connected
FFM	flat facet model
GAP	global average pooling
GEO	geosynchronous Earth orbit
GTO	geostationary transfer orbit
GUI	graphical user interface
HAMR	high area-to-mass ratio
ISS	International Space Station
LEO	low Earth orbit
LSTM	long-short term memory
MC	Monte Carlo
MEO	medium Earth orbit
MMT	Mini-MegaTORTORA
MLI	multi-layer insulation
MMAE	multiple-model adaptive estimation
MKL	multiple kernel learning
NORAD ID	North American Aerospace Defense Command identification number
NOVAS	US Naval Observatory vector astronomy software
PCA	principal component analysis
RDF	random decision forest
ReLU	rectified linear unit
RNN	recurrent neural network
RSO	resident space object
SGP4	simplified general perturbations 4
SLR	satellite laser ranging

SRP	solar radiation pressure
SSA	space situational awareness
SVM	support vector machine
TEME	true equator mean equinox
TLE	two line element
TSC	time series classification
UKF	unscented Kalman filter
ZIMLAT	Zimmerwald laser and astrometry telescope

Chapter 1

Introduction

Space debris poses a significant collision risk to both current and future space missions due to the increasing number of debris objects and the difficulty in actively removing debris from orbit. Currently, the risk to operational spacecraft is mitigated by 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 regarding avoiding potential conjunctions. However, the accuracy of these orbital predictions is limited by the unknown physical characteristics of a large proportion of debris objects. Knowledge of these physical characteristics, such as shape, size, orientation, mass and material composition, is an important factor in accurate orbital prediction. This is because they are required to correctly determine the orbital perturbations caused by non-conservative forces, such as drag and solar radiation pressure.

Resident space object (RSO) characterisation is the process of determining these physical characteristics for an object on-orbit and RSO characterisation from light curve observations is an active area of research. A light curve is a sequence of measurements of the RSO's brightness and contains information about its physical characteristics (i.e. shape, size, orientation and material composition). However, the direct determination of this information from the light curve is an ill-posed problem as a result of the coupling between the different characteristics and subsequently light curve-based

RSO characterisation remains an unsolved problem.

Data-driven approaches, utilising deep neural networks, have shown promising results on RSO classification tasks based on simulated light curve data [79, 38, 83, 57, 51]. However, these results have not been replicated on real world light curve data [38, 83]. Deep neural networks are a subset of machine learning algorithms that are typically comprised of multiple layers with a large number of parameters and non-linear components enabling them to learn complex non-linear relationships in the data [42]. Due to the requirement of learning a large number of parameters, in general, the larger the number of examples used during training (given sufficient balancing of classes) the better the resultant model will perform on the given task. In the case of RSO shape classification, light curves are difficult to obtain and label, so real world datasets remain relatively small and are often heavily unbalanced, restricting generalisation. It has been theorised that a lack of well-labelled training data is limiting the performance of deep neural networks on real world RSO characterisation tasks [38, 83].

Transfer learning is known to be an effective method of boosting model performance when training data is limited [89, 94, 90]. This process involves pre-training a deep neural network on a similar dataset that contains a large amount of well-labelled data, before fine-tuning the model weights on the final smaller dataset. Transfer learning is most successful when the dataset used for pre-training is similar to the final dataset as this reduces the amount of fine-tuning required on the final dataset [53].

The approach developed in this thesis uses high-fidelity simulated light curve data to perform simulation-based transfer learning in order to more effectively train deep neural networks and subsequently demonstrate improved shape classification performance on real light curve datasets. It is also demonstrated that the effective combination of multiple light curve observations for a single object can be used to further increase performance and assist with classification of challenging objects. Finally, an outline is provided of a suggested method for utilising the developed models in the context of real world live operations, including the incorporation of an uncertainty metric to reduce misclassifications.

1.1 Background and Motivation

1.1.1 Space Debris Overview

Artificial RSOs can be broadly grouped into two categories:

- Active payloads: RSOs that serve a purpose in orbit and may be controlled or uncontrolled.
- Space debris: all human-made objects in Earth orbit or re-entering the atmosphere that are non functional [68]. They are composed of defunct payloads, rocket bodies and fragments of space hardware produced by operational activities, deterioration (including explosions) and collisions.

Although there exists many proposed approaches to the problem [2, 8, 15, 19, 32, 34, 44], there are currently no cost-effective methods to actively remove space debris from orbit [34]. RSOs that are located in the lower altitude range of low Earth orbit (LEO) will experience orbital decay due to atmospheric drag, which acts as a natural sink helping to mitigate the problem of space debris [65]. However, even for an object in mid-LEO (600-900km) this process takes decades to hundreds of years and for higher orbits, such as medium Earth orbit (MEO) or geosynchronous Earth orbit (GEO), orbital decay does not occur [85]. Thus, as space becomes increasingly accessible, the number of RSOs in orbit will continue to rise.

The issue of space debris was first identified in 1978 when Kessler predicted that collisions between objects would create more debris, each increasing the chance of more collisions and thus triggering an uncontrolled growth in fragments [64]. As a result of the high relative velocities between objects on-orbit, RSOs with a diameter greater than 1cm are considered to pose a significant threat to operational spacecraft in the event of a collision [44]. At current levels, space debris is already a considerable risk to operational satellites with the first identified catastrophic conjunction event occurring in 2009, when a defunct Russian satellite collided with an active Iridium satellite [99]. This conjunction resulted in approximately 2200 pieces of catalogued debris [99], increasing the risk to other active payloads at a similar altitude and highlighting the danger of an uncontrolled growth in fragments.

Of even more concern is the threat of space debris to human space flights/missions. In May 2021, the Canadian robotic arm attached to the outside of the International Space Station (ISS) was struck by a small untrackable debris fragment resulting in a 5mm hole [132]. While outer shielding can provide satellites with some protection against objects that are smaller than 1cm in diameter [44], as of November 2021 it is estimated by the European Space Agency that there are more than 1,000,000 objects on-orbit larger than 1cm in diameter [31]. These have resulted from the estimated 630 fragmentation events including break-ups, collisions or anomalous events which have occurred on-orbit in the last 60 years [31].

The recent rise in accessibility of space through both national and commercial launch providers has led to a significant increase in on-orbit objects within the last decade. As more organisations attempt to utilise the benefits of space, and space operations become increasingly integrated into society, it is expected that this trend will continue.

1.1.2 Space Situational Awareness

To mitigate the risk that space debris poses to operational satellites, various organisations maintain catalogues of on-orbit RSOs containing information about their orbital state and operational status [108]. The collection and use of this information to make informed decisions in the space environment is part of a process known as SSA [47]. While some satellite operators share precise orbital state information about their satellite based on the tracking, telemetry and control systems, this information is not provided publicly for the majority of active payloads and is not available for debris objects. Thus these catalogues are primarily developed and updated through observations of RSOs [108].

SSA is particularly important due to the fact that events are sometimes not reported by other operators in the space environment, highlighting the need for continuous observation and monitoring. This concept can be effectively illustrated through two recent incidents. Firstly, a collision between a piece of space debris and an active Chinese military satellite resulted in a fragmentation event which was unreported by the Chinese government [5]. This was the fifth confirmed accidental collision between two catalogued RSOs and it resulted in 37 catalogued fragments [25]. Secondly, a recent Russian anti-satellite test has resulted in more than 1,000 additional pieces of space debris on-orbit at around 480km in altitude [71]. The altitude of the satellite prior to the missile strike means that the debris will remain on-orbit for a number of years and is expected to pose a threat to satellites in lower altitude LEO environment, including the ISS [71]. In both these cases, no warning was provided to other satellite operators and the incidents were first reported by ground-based observers through the detection of debris.

1.1.3 Observation and Tracking of Resident Space Objects

Although there have been recent developments of space based sensors [135], observations of RSOs typically come from ground-based sensors and can be grouped into either passive or active observations. Passive observations include optical and infrared observations, which are recorded through observing reflected sunlight or emitted thermal radiation from the RSO [108, 122, 116]. Active observations, such as radar and laser ranging, require that the observer targets the RSO with a radar or laser and then records the reflected signal from the satellite [74].

Position and velocity information can be extracted from a sequence of RSO observations and input into an orbit determination process to determine or update an RSO's orbital state. The orbital state is then propagated to predict possible conjunctions with other RSOs within a specified time period. As well as considering conjunctions involving active payloads, it is also important to predict and observe potential conjunctions between debris objects. While they currently cannot be prevented, a conjunction typically results in a significant fragmentation event, leading to a highly dynamic space environment and increasing the risk of future collisions [65].

Conjunction information is valuable to satellite operators if it is accurate, precise and timely. In the event of a predicted conjunction threat to an active satellite, the operators can assess the threat level and manoeuvre the operational spacecraft. This prevents damage to the operational spacecraft and avoids the creation of additional debris through a conjunction event, thus helping to mitigate the collisional cascading effect. However, these manoeuvres use propellant, which limits the operational life of the spacecraft, thus it is also damaging to provide satellite operators with a false positive conjunction warning.

The accuracy of ground-based observations varies both with different sensor types and with the quality of specific sensors. With the use of astrometric correction, sub-arcsecond accuracy is achievable for optical observations with standard commercial sensors [122], while a ranging accuracy of 1.5m has been demonstrated for laser ranging to objects without retro-reflectors [10]. Therefore, while the current state of an RSO can be accurately determined through observation, uncertainty in orbital perturbations caused by non-conservative forces, such as drag and solar radiation pressure reduce the accuracy of orbital predictions. These non-conservative forces are dependent on an object's size, shape, mass, attitude and drag/reflectivity properties [28]. Thus high-fidelity orbit propagation requires detailed knowledge of the characteristics of the RSO [129]. However, this information is unknown for the vast majority of RSOs, which can result in large orbital prediction errors after short propagation spans. This limits the effectiveness of SSA for protecting active satellites as it reduces both the amount of time that satellite operators have to make a decision as well as the certainty of the manoeuvre required.

1.1.4 Resident Space Object Characterisation

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 [108]. RSO characterisation is the process of determining these characteristics for an object, commonly from ground-based observations. It is an important goal in SSA in order to improve the accuracy of orbital predictions to reduce the collision risk to active satellites [36, 96]. Additionally, RSO characterisation can be used to assist in identifying targets for potential future debris removal missions as well as analysing payloads from non-cooperative nations or organisations.

A light curve is the temporal history of an object's apparent magnitude [128], which can be extracted from a series of non-resolved images of an object captured using a ground-based telescope. The apparent magnitude of an RSO is a function of its physical characteristics, including size, shape, orientation and surface material properties, as well as the distance to the observer and the phase angle relative to the sun and the observer. The phase angle and range of the object are both determinable parameters from the object's orbital state. Thus one or more of the characteristics should be recoverable from the light curve data [80]. The extraction and analysis of a number of light curves for different objects [112, 84, 35], has demonstrated that for rotating objects, a repeating pattern can often be observed in the light curve.

Light curve based RSO characterisation has been found to be a particularly challenging problem due to the limited amount of information that is present in the light curve data and the fact that it is difficult to separate the different characteristics in order to solve for them individually. Subsequently, it has been found to be a highly underdetermined problem [33]. While significant steps have been taken towards solving this problem in the literature using both theoretical [128, 81, 48, 33] and data-driven approaches [49, 115, 58, 37, 83, 38, 51], typically these approaches have been applied to constrained versions of the problem using simulated data. Currently, a significant portion of catalogue space debris remains uncharacterised with RSO characterisation remaining both an active and an unsolved problem [113].

1.1.5 Deep Learning and Data-Driven Approaches to RSO Characterisation

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 [45]. 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. Recent research has focused more on data-driven approaches, with the idea that machine learning models could be used to enable quick determination of RSO classes made directly from observational data [79]. One of the advantages of this method is that it will be scalable to the expected increase in catalogued objects as sensors improve and more RSOs are injected into orbit. While results on simulated datasets have shown success, application to real light curve data has proven to be more difficult due to the smaller dataset size and the differences between the simulated and real light curve data [79, 38, 83].

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 [42]. 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 [20].

Deep learning models typically require access to a large well-labelled dataset in order to effectively train the parameters within the network to perform a specified task. The lack of appropriate quantities of training data 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 [39]. Thus it is difficult to build up a large well-balanced dataset for a range of different object classes. Consequently, the size and quality of the training dataset limits the achievable performance and robustness of the neural network. 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 [83].

A high-fidelity light curve simulation environment enables the possibility of creating a large, well-labelled dataset that could be used to supplement observational RSO light curve datasets for the purposes of training neural networks. Provided the simulated light curves encapsulate similar features to those found in a real light curve dataset, models pre-trained on the simulated dataset could then be fine-tuned on a small real light curve dataset in a process known as transfer learning. Transfer learning is known to be an effective way of increasing the performance of neural networks, particularly in cases where the dataset is small [91].

1.2 Contributions

This thesis utilises simulation-based transfer learning to improve the shape classification performance of deep neural networks when applied to real light curve datasets. This approach requires the development of a high-fidelity simulator in order to generate simulated light curves that encapsulate the features which are present in the real light curve data. Additionally, for challenging cases, a framework is also developed where multiple light curve observations for a single RSO can be effectively combined in order to further improve performance. In order to efficiently use this method in the real word, we incorporate an uncertainty metric into the final framework to allow the user to best allocate observational resources while also reducing misclassfications.

The primary contributions of this thesis are as follows:

1. Development and use of a high-fidelity 3D rendering simulation-based environment for providing realistic light curve data as well as validation of this simulation environment through comparison with real data.

2. Demonstration that simulation-based transfer learning can be utilised to improve the classification performance of a 1D-CNN on real light curve data, through the use of the simulation environment to generate a high fidelity simulated dataset. This reduces the requirement for large well-labelled real light curve datasets in order to effectively train deep neural networks.

3. Development of a neural network frameworks for RSO shape classification that are capable of efficiently combining a variable number of light curve observations

of a single target object in order to support both multi-temporal and multi-site observations. Evaluation of these frameworks under different observing strategies, with the incorporation of uncertainty into the model framework to predict the utility of collecting further observations on a specific object.

1.3 Publications

The following works have been published during the undertaking of this thesis:

1. James Allworth, Lloyd Windrim, Jeffrey Wardman, Daniel Kucharski, James Bennett, Mitch Bryson, Development of a high fidelity simulator for generalised photometric based space object classification using machine learning, *Proceedings of the 70th International Astronautical Congress*, Washington DC, 2019.
2. James Allworth, Lloyd Windrim, James Bennett, Mitch Bryson, A transfer learning approach to space debris classification using observational light curve data, *Acta Astronautica*, Volume 181, April 2021, Pages 301-315, ISSN 0094-5765.

1.4 Thesis Structure

The remainder of this thesis is outlined as follows. Chapter 2 provides a survey of the literature on RSO characterisation, with a focus on light curve based characterisation using both traditional methods and data driven approaches. Chapter 3 presents the development of a high-fidelity 3D rendering based light curve simulation environment, which will be used to generate realistic simulated light curve datasets. In Chapter 4 a 1D-CNN model is developed to perform RSO shape classification using light curve datasets and it is demonstrated that simulation-based transfer learning can be used to improve classification results on the real light curve datasets. Chapter 5 extends on earlier work by presenting a method for effectively combining multiple light curve observations of a single RSO within the model architecture in order to further improve shape classification accuracy for challenging targets. A targeted scheduling process is also developed, through the use of an uncertainty based threshold, in order to identify cases where the collection of additional data is beneficial. Finally, Chapter 6 provides a summary of the research presented in this thesis as a whole, discusses the implications of the key findings and suggests future work.

Chapter 2

Literature Review

This chapter presents an overview of key concepts and a review of the literature that is most relevant to this thesis. Section 2.1 provides an overview of the coupling between orbital perturbations and RSO characteristics, demonstrating the importance of RSO characterisation to accurate orbit predictions. A summary of the common RSO observation methods is provided in Section 2.2, while Section 2.3 reviews optical surveys that have been performed to collect and analyse real world light curves. Section 2.4 reviews the development of light curve inversion based techniques for RSO characterisation as well as providing an overview of the light curve simulation environment that is commonly used in the literature. An analysis of data-driven approaches to RSO characterisation from light curve data is provided in Section 2.5, while Section 2.6 provides an overview of key deep learning techniques that have been utilised in other domains but have not yet been applied to RSO characterisation. Finally, a brief summary of the literature review will be provided at the end of this chapter in Section 2.7, which will highlight the key findings that have been identified.

2.1 Coupling between Orbital Perturbations and RSO Characteristics

As outlined in the introduction, one of the key goals of SSA is to maintain accurate orbital state information about all RSOs on-orbit, in order to reduce the risk of collisions between RSOs. While these RSOs can be observed using the methods presented in Section 2.2, in order to obtain their current state information, non-conservative forces acting on the RSO reduce the accuracy of orbital predictions [96]. In particular, atmospheric drag is the dominant non-conservative perturbing force for RSOs in LEO, while solar radiation pressure is the most significant factor for RSOs in higher orbits. The orbital perturbations as a result of these forces are dependent on the physical characteristics of the RSO.

Initial attempts to incorporate these forces into orbital predictions used the "cannonball model", which assumes that the RSO is a spherical object with a constant area-to-mass ratio (AMR) and drag/reflectivity properties [28]. A parameter known as the ballistic coefficient could then be determined for a specific object based on a comparison between its predicted and observed state [103]. This model was determined to result in significant errors for RSOs with a high area-to-mass ratio (HAMR) [36], however it was initially thought they provide reasonable solutions for RSOs with a low AMR, including operational satellites and large debris objects (e.g. rocket bodies). More recent research has concluded that the cannonball model results in large errors for non-spherical objects with a low AMR [96].

While more sophisticated models have been developed to incorporate forces into the orbital predictions [96, 30], they rely on information about the RSO's characteristics. However, these parameters are typically unknown for space debris objects as well as classified payloads [108, 113]. Subsequently, RSO characterisation remains an important and unsolved issue for improving the accuracy of orbital predictions, as well as other SSA related tasks

2.2 Observation Methods for RSOs

A number of different observations methods have been developed in order to observe space debris, both to determine the position and velocity of the RSO as well as to infer information about its characteristics. These can be broadly characterised into either active or passive observations. Active observations involve the observer targeting the object with a signal (e.g. radar or laser) and then observing the reflected signal, passive observations involve observing natural reflections (i.e. from the sun) or emissions from the object (e.g. optical observations, which may range from visible wavelength to infrared).

2.2.1 Active Observation of RSOs

Radar systems have proven to be very effective at detecting and observing RSOs in LEO [108]. Characterisation techniques from radar observations were drawn from the field of planetary radar astronomy where they were developed to estimate the shape and spin of natural satellites. These radar-based methods included cross sectional estimation and range Doppler interferometry [126]. Radar methods are limited to RSOs that are larger than the radar's wavelength and are subject to signal loss in proportion to distance to the fourth with reference to the distance between the observing sensor and the RSO, making it expensive and difficult to observe small objects in non-LEO orbits [80].

Satellite laser ranging (SLR) has also been shown to be an effective method for RSO characterisation, in particular for determining the attitude of known objects [73, 74]. SLR is able to measure very precise distances to satellites through laser pulses. Spinning arrays of corner cube reflectors (CCRs) cause millimeter-scale modulation of range measurements, which enable the spin rate of the RSO to be accurately determined [73]. However, this method relies on a precise knowledge of the shape of an RSO as well as the RSO having multiple CCRs. CCRs are uncommon on debris objects and in their absence the returned signal is significantly weaker resulting in the requirement for more powerful lasers [43]. Additionally, similar to radar, laser signal loss is in proportion to distance to the fourth and is subsequently not currently suitable for debris observations in non-LEO regimes. Laser ranging is a likely to become an increasingly useful tool for RSO characterisation in the future as more advanced laser systems are developed and more precise measurements of attitude are required.

2.2.2 Passive Observation of RSOs

Optical observations are passive observations of the RSO using a telescope with a sensor attached to it, which records the sunlight reflected from an RSO in the direction of the observer. As such, this method requires that both the object is illuminated by the sun and that the illuminated face of the object is visible to the observer. Additionally, optical observations are primarily undertaken when the sun is below the horizon to reduce the level of noise in the image.

Some ground-based telescopes are able to resolve features of very large RSOs (e.g. the ISS), however atmospheric distortion of the reflected light prevents the overwhelming majority of RSOs from being resolved in telescope imagery [23]. Subsequently, research in the field of RSO characterisation from optical telescopes has focused on non-resolved imagery [108]. Non-resolved optical observations can be used to obtain observation angles data, which provide information on the angular position and angular velocity, as well as to obtain photometric information [35]. Photometry is a measure of the brightness of the RSO, which is usually performed at visible wavelengths (known as broadband photometry) [63]. When an RSO's photometric information is recorded across a sequence of images it can be used to generate a light curve, which is the temporal history of the RSO's brightness, typically measured in apparent magnitudes [128].

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 [56, 128, 80, 82]. In addition, to the RSO's physical characteristics, the light curve is also dependent on the range between the observing sensor and the RSO as well as the phase angle between the sun, the RSO and the observer. In this context, phase angle is defined as the angle between the unit vector from the RSO to the observer and the unit vector from the RSO towards the sun [92]. Both the range and phase angle can be determined from the RSO's orbital state and thus knowledge about one of more of the RSO's characteristics should be determinable from the light curve data [80]. As light curves are obtained through passive observation, the loss in intensity is proportional to distance squared and the are subsequently more sensitive to object size at large distances compared with active observation methods [108]. This makes them suitable for RSO characterisation for RSOs at all orbital bands including geostationary transfer orbit (GTO) and GEO.

In addition to the broadband photometric observations used to obtain light curves, different sensors can be attached to optical telescopes in order to obtain characteristic information about RSOs. Observations at other wavelengths have been investigated including the thermal infrared and short/medium wave infrared bands [116, 117]. These measurements can provide additional information about the material characteristics and orientation dynamics of an RSO [95]. While these wavelengths relax the requirement that the object must be sun-illuminated, as thermal emissions occur throughout the orbit, the observed signal is typically reduced in these bands compared with the visible observations. Additionally, the use of spectrometry to analyse the reflectance spectra of RSOs has been found to be beneficial when attempting to determine information about the material properties of the RSO [1, 63, 123]. However, this technique has not been widely used in attempts to classify the shape and attitude of RSOs as observations are typically sparsely sampled temporally.

2.2.3 Emerging Technologies for the Observation of RSOs

As well as the established methods that have been outlined above, there are a number emerging technologies that are expected to play an important role in future RSO tracking and characterisation. Of particular note, is the recent development of event-based sensors [21, 59], as well as single-photon avalanche diode detectors [75]. Both of these sensors, enable the collection of very high temporal resolution observations, compared with current observation methods, which integrate the incoming signal over the exposure period. This may enable more precise measurements of specific features, like spectral flashes, in the light curve allowing for more detailed characteristics to be determined.

Furthermore, recent research in the field of adaptive optics using deformable mirrors to correct for atmospheric distortion has shown that RSO features greater than 50cm are able to be resolved in corrected imagery [23, 44]. There is also potential to use space-based sensors [135, 131], which would alleviate the signal distortion caused by the atmosphere, enabling resolved imagery of space debris. The advancement and implementation of these methods may allow for more detailed RSO characterisation in the future with features observable directly in the captured imagery.

2.3 RSO Characterisation using Observational Light Curve Data

In this thesis, the focus is primarily on RSO characterisation from light curve data. Light curves were selected as compared with active observation types they are more suitable for non-LEO orbits [105, 108]. Additionally, light curves are known to be dependent on the RSO's characteristics [56, 128] and are more widely available than other passive observation types. The following subsections provide a review of the optical surveys that have been performed to collect light curves of RSO as well as the analysis of this data to determine characteristics about the RSOs.

2.3.1 Initial Optical Surveys and Identification of High Area-to-Mass Ratio Objects

Prior to 2000, space debris had been identified and analysed in the LEO region using both radar and optical observations, however due to the limitations of radar in higher orbital regimes, very little was known about debris in the GEO and the GTO regions [105]. Schildknecht used non-resolvable ground-based optical observations to survey these regions and identified a substantial population of small and previously uncatalogued RSOs [105, 106, 108]. This research also used orbital analysis to determine that many of the identified RSOs had HAMRs, resulting in more variable orbits due to the non-conservative force applied by solar radiation pressure (SRP) compared with objects with low AMRs [106, 108]. These findings highlighted the inaccuracies associated with the cannon-ball assumption that was commonly used in orbit propagation and demonstrated that the characteristics of the object needed to be taken into consideration in order to accurately predict the future state of the object. They also emphasised the growing issue of space debris and the threat that it poses to operational missions.

In order to further analyse the identified HAMR objects, Schildknecht extracted light curves from the optical observations of selected HAMR objects [108]. Using these light curves he was able to determine that the HAMR objects had a highly variable apparent magnitude, although distinct periodic signatures were able to be identified over short time spans, indicating a tumbling motion and a complex shape [108]. Observations of a small group of low AMR GEO debris objects appeared to have flat or slowly varying light curves whilst all HAMR objects showed variations with periods of a few minutes [107]. However, he was unable to identify a correlation between brightness variation and the AMR, nor to classify objects into specific classes or determine shape information from the light curves [108, 107]. This body of research, led by Schildknecht, were the first significant attempts at classifying RSOs using real light curve data. It demonstrated that light curves could be collected and analysed to determine characteristic information about an RSO in orbit as well as highlighting the challenges of dealing with real world data.

In 2010, Früh and Schildknecht investigated the validity of light curve simulations for various simple shapes in comparison to the real RSO data that they were collecting through the 1-meter Zimmerwald laser and astrometry telescope (ZIMLAT) telescope [35]. A flat facet-based model was used, similar to the approach taken by other research at the time [128] (outlined in more detail in Section 2.4.2), however the simplifying assumption of pure diffuse (Lambertian) reflection was also applied. This assumption removes the ability to simulate glints caused by specular reflection which can provide valuable information about the RSOs attitude and material properties. The authors simulated various rotations and phase angles for three simple shapes: a cube, a cylinder and the shape of a thin multi-layer insulation (MLI) structure, before comparing the results with real light curve data for three RSOs [35]. A simple pattern recognition algorithm was developed and the patterns detected, although not identical for different light curves of the same object, were clearly different for two different objects when compared [35]. This result shows that it may be possible to classify objects based on their light curves, however due to the limited amount of real data analysed, it is unclear how generally applicable these results are. Additionally, the comparison between simulated and observed light curves indicate that a significantly higher fidelity simulation is required in order to accurately model the observed light curves.

2.3.2 Combination of Multiple Observations to Provide Additional Information

Based on earlier research, Skinner et al. observed that there were a large number of parameters involved in RSO characterisation and that observational light curve data provides relatively sparse information [116]. Subsequently, the combination of multiple observation types was investigated to determine if that would provide additional information, enabling more precise characterisation. This was done by performing simultaneous visible and infrared observations on selected RSOs, with a continued focus on HAMR objects [116, 117]. During the observational campaign, 12 different

HAMR objects were observed, with the visible light curve analysis indicating the objects were rotating, which supported the previous results published by Schildknecht [108, 107]. Based on the infrared observations, the objects were generally determined to be a few square metres in size with temperature between 200K and 320K and a low thermal mass [116]. The authors found that by incorporating this information into more sophisticated SRP models they were able to improve the fidelity of the orbit propagation for these objects [117]. These results emphasise the importance of RSO characterisation for improving the accuracy of orbital predictions as well as highlighting the benefits of combining multiple observation types in order to increase knowledge of the RSO.

Research has also demonstrated that optical observations of a single RSO can be simultaneously collected from multiple sensors separated spatially in order to obtain light curves from multiple viewing angles [41, 70]. It was theorised that this would enable the determination of the RSOs rotation parameters including the inertial rotation period and the orientation of the rotation axis in space after promising results on simulated data [70]. While, these initial studies were unable to determine the exact rotation parameters for the observed RSOs, the simultaneous observations did enable a more detailed analysis [41, 69] and this is an ongoing area of research.

2.3.3 Analysis of Recent Optical Surveys

In the last five years, there has been an increase in awareness of the threat of space debris and subsequently the importance of RSO characterisation, combined with an improvement in the accessibility of automated telescope technology. This has led to a recent growth in optical tracking campaigns focused on the collection and analysis of light curves for RSO characterisation using the techniques developed by Schildknecht [112, 69, 113, 13, 114, 16]. 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 [69, 13].

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 [112, 113]. 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. In a more recent study, the authors extracted two parameters, maximum signal amplitude and complexity, from a database of 512 real light curves once the rotation phase had been fitted [114]. By plotting the obtained parameters for various objects, the authors were able to identify clusters related to different object shapes [114]. Whilst there were outliers to the identified trends, these results indicate that real light curves can be used for RSO shape classification and are a significant step towards a more generalised solution.

2.4 Light Curve Inversion Techniques

Through Schildknecht's observations in the early to mid 2000s, it had become clear that space debris was a significant issue and that the characterisation of debris objects was an unsolved and challenging problem. At this time, it was difficult to obtain high quality light curve data due to cost of telescope equipment as well as the limited time available for hiring operational sensors. Additionally, the algorithms for automatically detecting space debris in telescope imagery and extracting calibrated light curves from these images were not widely developed or available. Subsequently, with limited access to real light curve data, a separate branch of research developed in the mid to late 2000s, with a light curve inversion based approach to solving RSO characterisation using simulated light curve data. The following sections review the development of this method and the incorporation of astrometric data to provide additional information as well as the light curve simulation environment that has commonly been used in the literature.

2.4.1 Initial Application of Light Curve Inversion

Similar to the problem of RSO characterisation, a method for characterising asteroids from non-resolvable optical observations had been developed by solving the light curve inversion problem [60, 61]. In its simplest form, the convex light curve inversion problem is defined in [60] as:

$$\mathbf{L} = A\mathbf{g}, \tag{2.1}$$

where $\mathbf{L}$ is the vector of the observed brightness of the object (the light curve), A is the reflection matrix, mapping the albedo-areas to a given normal direction with the reflection response that is received and $\mathbf{g}$ is the parameter to be solved for, describing the areas of the facets of a polyhedron.

While the full solution to this problem is ill-posed, asteroids typically have a primary rotation axis with smooth surfaces made of a single material enabling assumptions of Lambertian reflection, convex hull and single axis rotation [33]. Using these assumptions, Kaasalainen and Torppa were able to determine the shape, rotation periods, pole directions and other parameters of non-resolvable asteroids [60, 61].

Inspired by these results, light curve inversion was applied to the problem of RSO characterisation with the aim of determining the shape and attitude of objects using simulated light curves [45, 18]. However, in contrast to asteroids, artificial RSOs have highly angular shapes made from a variety of materials and a wide range of possible rotation states [45]. Additionally, asteroid inversion problems typically require large datasets of light curves with varied viewing geometry in order to be solved [78]. This data might not be obtainable for specific RSOs based on their viewing constraints and is difficult to obtain for large numbers of RSOs. Thus the assumptions made for asteroids no longer apply in the case of artificial RSOs, resulting in the coupled shape, attitude and surface inversion problem becoming heavily under-determined in a highly non-linear domain [33].

To constrain the problem, researchers separated the shape and attitude determination as well as applying a number of simplifying assumptions to the generation of the simulated light curves including diffuse only reflection and convex shape [45, 18]. By assuming the orientation of the objects were known, Calef was able to use simulated light curve and thermal emission data to reconstruct objects with simple shapes by adapting Minkowski's formula [98, 18] . This idea was developed further by Hall who used light curve data to perform shape independent attitude analysis and attitude independent shape analysis in order to constrain either the shape or attitude for simple rotating objects [45]. These results showed that it was possible to use light curves to recover information about both the shape and the attitude of an RSO in the constrained simulation environment. However, the assumptions made by both Calef and Hall to constrain the problem and to simplify the simulation, limit the application of this method to real data.

2.4.2 Development of Light Curve Simulation Environments

In order to gain a better understanding of the properties of light curves as well as the relationship between RSOs and their light curves, various simulation environments have been developed in the literature. This section reviews physical light curve simulation environments as well as outlining the development of the flat facet model (FFM) with the Ashihmin and Shirley (AS)-bidirectional reflectance distribution function (BRDF), which has been commonly used for light curve simulations in the literature.

Physical Simulation Environment

Cowardin et al. developed a physical simulation environment in a laboratory to investigate the effects of various materials on light curves [26] and later to analyse the light curves of rocket bodies in further detail [24]. Physical models were illuminated by an arc lamp, with their rotation state controlled by a robotic arm and charge-coupled device (CCD) camera with standard Johnson/Bessel filters used to capture the resultant light curves [26]. Results reported from the investigation of various materials highlighted the significance that changes in material properties, such as colour and glossiness, can have on the light curve [26].

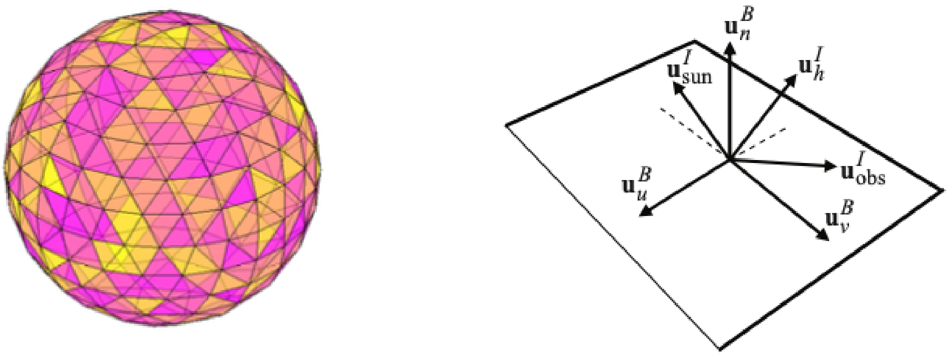
To analyse SL-8 rocket bodies, three probable rotation states for the scaled models were selected to be tested in the experimental laboratory set up and compared against real light curves of SL-8 rocket bodies: end-over-end, 10° wobble and spin-stabilized [24]. The authors concluded that light curves of rocket bodies are highly dependent on the rotation state and less so on the solar phase angle. The physical simulation environment developed in this research was very useful for gaining a greater understanding about the complex relationship between objects and their light curves. However, it also showed how difficult it is to accurately simulate RSOs on-orbit even when their shape is known [24]. Additionally, in a physical simulation environment it is difficult to rapidly test the wide variety of parameters that can affect an object's light curve due to the need to make physical alterations to models. This has led to subsequent research focusing on computer-based light curve simulations.

Flat Facet Model for Light Curve Simulation

The FFM is a commonly used light curve simulation method which has been developed and further refined in the literature [56, 128, 80, 129, 82, 97]. The key premise is that flat facets can be combined to create more complex shapes as depicted in Figure 2.1 (from [97]). Figure 2.1a presents an example of a convex shape created using flat facets, while Figure 2.1b displays the reflection geometry for a single facet. The selected reflectance model is applied to each facet individually with the light curve generated by summing over the reflected intensity contribution of all illuminated and observer-visible facets [80]. The fraction of reflected light in the direction of the observer from a single facet can be calculated using Equation 2.2 [97].

$$F_{obs}(i) = \frac{F_{sun}(i)\beta_{total}(i)A(i)(\mathbf{u}_n^I(i) \cdot \mathbf{u}_{obs}^I(i))}{||\mathbf{d}||^2} \tag{2.2}$$

where $F_{sun}(i)$ is the fraction of sunlight that strikes the facet and is not absorbed, $\beta_{total}(i)$ is the fraction of light reflected from the facet as determined by the BRDF, $A(i)$ is the area of the facet, $\mathbf{u}_n^I(i)$ is the unit vector of the facet pointing in the direction of the outward normal to the face, $\mathbf{u}_{obs}^I(i)$ is the unit vector pointing from



(a) Convex Shape Generated using a Combination of Flat Facets **(b)** Reflection Geometry of a Single Flat Facet

Figure 2.1 – (a): Example of a space object shape model generated using the flat facet method. (b): Reflection geometry of a single flat facet with vectors depicting the body frame (B) of the facet as well as the direction of the sun, the observer and the half vector (h) in the inertial reference frame (I). Image sourced from [97]

the RSO to the observer and **d** is the distance between the RSO and the observer.

Whilst there are a number of different reflectance models that can be used to determine β_{total} in Equation 2.2, the BRDF most commonly used in the literature is the simplified version of the AS-BRDF developed for flat surfaces [6]. The AS-BRDF incorporates both diffuse and specular reflection, which is a significant improvement on previous simulation environments where only diffuse reflection was considered. The FFM also enables the facets to be combined in order to create more realistic shape models. However, in comparison to real world scenarios, the FFM does not account for self shadowing or secondary reflections and does not model curved surfaces effectively.

2.4.3 Astrometric and Photometric Data Fusion

Building on the light curve inversion results determined by Hall [45] and Calef [18], a body of research has been developed using data fusion between angular measurement of the RSO's position and light curve data [56, 128, 80, 81]. The data fusion provides additional information into the light curve inversion process by taking advantage of the coupling between an object's characteristics and its orbital dynamics through the influence of solar radiation pressure [56]. This enables filtering methods to perform simultaneous estimation of the RSO characteristics and orbital state. This research has typically employed the use of simulated data, using the FFM outlined in Section 2.4.2 to simulate light curves.

Initial research decoupled the attitude and shape determination by focusing on the use of a unscented Kalman filter (UKF) to estimate the attitude of the RSO provided with a known shape input [56, 128]. Light curves and angles data was simulated for simple shape objects, including a flat plate [56] and a cylinder with hemispherical caps, in order to simulate a rocket body [128]. Even after being initialised with perturbations the UKF was able to successfully recover the true orbit and partial RSO characterisation was achieved in terms of its size and reflectance properties [56]. However, it was found that the method was highly dependent on the known shape parameters, with any changes to these parameters resulting in systematic error in the predicted values as well as cases where the UKF would not converge [128].

Data fusion of the angles and light curve data was further explored in the literature with the application of a UKF to perform shape estimation [80], followed by mass and area estimation [81]. This research was further extended to incorporate the use of multiple-model adaptive estimation (MMAE) in conjunction with the UKF to determine the most probable shape of an RSO from a bank of candidate shape models [82]. In all cases the initial results were demonstrated on simulated light curves from the FFM with relatively simple object shapes as well as other constraints in order to simplify the problem.

More recent research has suggested that the non-linearities present in the coupled shape/attitude determination and the possibility of multiple solutions invalidate the Gaussian assumption made by the UKF [48, 78], which had previously been heavily relied on in the literature. Consequently, Bayesian estimators have been investigated as possible method for solving the light curve inversion problem in the non-Gaussian space [48, 78]. 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 cases using real light curve data.

2.5 Data-Driven Approaches to RSO Characterisation

The growing catalogue of space debris combined with easier access to real world light curve data and the rise in prominence of machine learning techniques, has led to contemporary research investigating data-driven machine learning approaches to RSO characterisation. Initial work in this domain used traditional feature-based machine learning classifiers to perform shape classification on features extracted from light curve data [49, 115, 58, 12, 66]. More recently, the exceptional results that deep neural networks have achieved in other domains has inspired research into their ability to perform RSO shape classification directly from light curve data [37–39, 51, 57, 79, 83, 86]. The following sections provide a review of the relevant literature for the application of both traditional machine learning models, using manually extracted features, and deep learning models to the problem of light curve-based RSO classification.

2.5.1 Feature-Based Machine Learning Approaches to RSO Classification

Traditional machine learning approaches apply classifiers, such as random decision forests (RDFs) and support vector machines (SVMs), to feature vectors extracted from light curve data, rather than applying them directly to the raw light curve data. These features can either be manually selected to incorporate the features that the authors determine to be most important, such as dominant frequency, maximum glint and Gabor features [49, 58] or selected automatically through techniques such as principal component analysis (PCA) [115]. In order for the classifiers to be able to effectively differentiate between the different shape classes, the features must encapsulate the distinguishing characteristics of each RSO shape in the light curves.

Application to Simulated Light Curve Data

In 2015, Howard demonstrated that feature vectors and traditional machine learning classifiers could be effectively applied to simulated light curve data to classify the shape, stability and material of various objects [49]. After testing a number of different classifiers, he determined that RDF was the most successful classifier once noise had been added to the simulated data with an overall classification accuracy of 86.04% on dataset achieved. Similar techniques used in subsequent research on simulated light curves supported these results [115, 58]. It should be noted that in all cases, the simulation environment used to generate light curves were very simplistic, even compared with other light curve simulations, such as the FFM, which was commonly used in research at the time [81]. This significantly limits the validity of these results when considering their application to real world light curve data.

Application to Real Light Curve Data

Subsequent research has attempted to overcome these limitations by applying traditional machine learning methods to real light curve data showing successful classifications for simple two or three-class problems [12, 66]. In both these papers, the aim was to classify the light curves into general object classes (i.e. payloads vs. rocket bodies vs. debris) as opposed to classifying objects based on their specific shape. Whilst this was done successfully, it should also be noted that both light curve datasets contained many more examples of payloads than other object types. To illustrate the extent of this imbalance, in the dataset used in [12], 97.4% of the light curves were labelled as payloads, 2.5% were labelled as rocket bodies and 0.1% were labelled as debris. With this kind of imbalance in the dataset it is impossible to generalise the results of these studies to the wider RSO classification problem. This research depicts a step forward in the literature as it was the first time that machine learning methods had been applied to real world data. However, the imbalance in the light curve datasets and the fact that only general object classes were considered, rather than the actual shape of the object, restricts the impact of these results.

2.5.2 Deep Learning Approaches to RSO Classification

While the fundamental concepts of deep neural networks had previously been developed [76, 77], computational advances and increases in available data in the last decade have resulted in the rapid widespread implementation of deep learning models due to their extraordinary success on a range of perceptual problems [42]. Deep learning was first applied to light curve-based object characterisation in 2016 by Linares and Furfaro who identified that the rapidly growing catalogue of uncharacterised space debris required a new approach compared to the methods that were currently being explored in the literature [79]. They hypothesised that a CNN would enable a new way of processing RSO observations, where in contrast to traditional methods using light curve inversion methods, quick determinations of RSO classes would be made possible directly from observational data [79].

A CNN is a specific type of deep neural network that has been found to be very effective at processing spatial data (e.g. image classification tasks). Similar to the traditional machine learning classifiers presented in previous work [49], a CNN is a supervised machine learning technique which maps inputs (light curve measurements) to outputs (classes of RSOs). Unlike the traditional classifiers, a CNN does not require a manually created feature vector, instead it has the capacity to learn the most important features of the light curve through the supervised training process.

Application to Simulated Light Curve Data

Similarly to previous approaches, Linares and Furfaro first tested the proposed CNN model on simulated light curve data generated using the FFM [79, 38]. For the simulation, the RSO shape and surface parameter models are randomly generated and

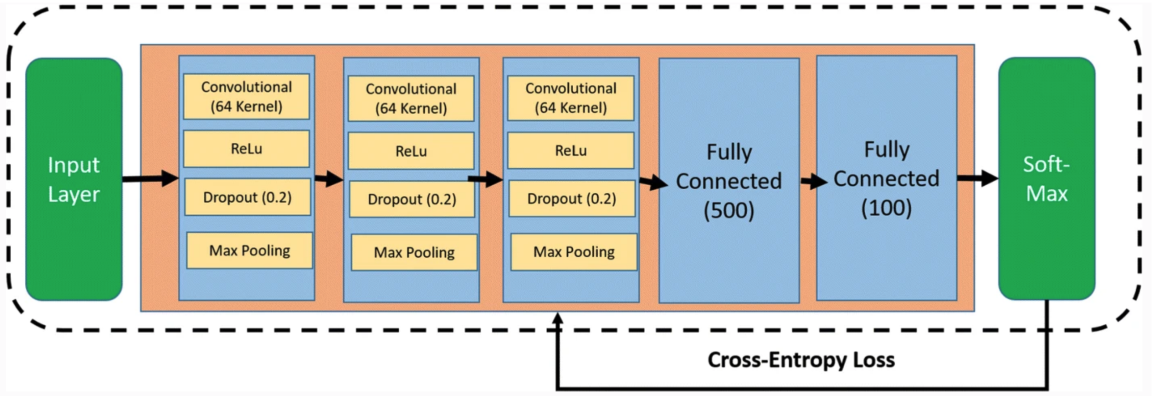


Figure 2.2 – 1D-CNN Network Architecture developed by Furfaro and Linares, image sourced from [83]. The model takes in a light curve input and outputs a predicted RSO class for the input. The network contains three convolutional layers utilising rectified linear unit (ReLU) activation, dropout and max-pooling as a well as a fully connected layer with dropout and ReLU activation.

can be generically grouped into four categories: fragment, regular polygon prisms, rocket bodies and rectangular cuboids. The light curves were simulated using the FFM with AS-BRDF model and quaternion parameterisation that is employed during previous work by Linares [80, 82]. A 1D-CNN model was selected as the classifier, given that the light curve data is one-dimensional time-series data. The developed network architecture is displayed in Figure 2.2, with the network utilising a three 1D convolutional layers followed by two fully connected layers. The output layer used a softmax function to map the network outputs to the classification states (the RSO shape models). The 1D-CNN was trained with 45,000 randomly generated scenarios and then tested on 5,000 scenarios not used in the training set. All scenarios have the same orbital elements as well as initial quaternion and angular rate of the RSOs.

Initial results demonstrated that the 1D-CNN could effectively classify the objects in different classes based on the simulated light curve dataset with an accuracy of 99.6% [79]. While this appeared to be a significant result, as the authors continued to build on this research they tested the performance of two traditional machine learning methods, using bagged decision trees and SVM as a basis for comparison [38, 83]. After adding more shape classes to increase the complexity of the simulated dataset, they found that for the simulated light curves all three methods achieved an overall classification accuracy over 95% with only a marginal improvement using the 1D-CNN compared with the traditional approaches [38]. Subsequently, the authors found that the on the simulated light curve data, the accuracy between traditional machine learning and deep learning methods was generally comparable [83].

In 2018, Jia expanded on the previous research by examining the use of a recurrent neural network (RNN) for light curve characterisation [57]. RNNs, unlike other neural networks, have a concept of memory and as such are generally applied to time sequence data such as text or audio-based problems where the order of the data is important [20]. Comparable results were achieved between the RNN model and the 1D-CNN model on a dataset containing phase-folded star light curves with three different classes of stars with both models achieving an accuracy over 96% [57]. The authors justified their use of the star light curve dataset as opposed to an RSO dataset by arguing that model-based simulation is computationally expensive with the parameters of space objects often unknown and that real RSO data is difficult to acquire.

Recent research has explored neural networks architectures that include a combination of recurrent and convolutional layers in order to create a network that is capable of processing both spatial and temporal data. In 2019, Huo developed a convolutional-recurrent neural network (C-RNN) architecture to extract the features of 5 different GEO RSOs from simulated light curves [51]. Interestingly, this model was only used to maximize the feature distance of the different categories and not classify the light curves. A separate multiple kernel learning (MKL) classifier based on an SVM was used to simultaneously classify the RSO's shape and attitude class from the feature vector, achieving an accuracy of 99.6% [51]. This hybrid method was shown to outperform using the C-RNN as an end-to-end classifier as well as a number of different architectures and traditional machine learning approaches. However, direct comparisons between the network with and without the RNN component (i.e. comparison between C-RNN and CNN) showed only a 0.4% improvement from including the RNN component [51].

Application to RSO Light Curve Data

Furfaro and Linares were also the first to investigate the effectiveness of deep learning models on real light curve data classification tasks [38, 83]. They applied their developed 1D-CNN model, as depicted in Figure 2.2, to a three class classification on real light curve data, attempting to identify objects as rocket bodies, debris and other. They obtained the light curve data from the Mini-MegaTORTORA (MMT) [14], which has recently made a set of light curves publicly available. In contrast to the result on the simulated dataset, 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% [38, 83]. The authors argued that the many layer configuration of the 1D-CNN enables it to learn more complex decision boundaries that are present in the real data, which are unable to be identified by the traditional machine learning techniques [38]. This suggests that the simulated light curve data generated by the authors does not effectively encapsulate the complexities present in the real light curve data. Additionally, while this research does demonstrate that the 1D-CNN is able to identify patterns in the light curves of the objects, once again the real light curve dataset is grouped into general object classes as opposed to shape or model classes. Subsequently, this result is inconclusive for determining if shape classification is possible on real light curve datasets.

Furfaro and Linares also identified during their research that the inability to obtain large, high quality and well-labelled datasets of real light curve data may be a limitation in achieving high accuracy levels as the CNN tends to overfit the data on smaller datasets [38]. Although the MMT dataset has been made available to the public, it is difficult to label this dataset into object shapes as the specific shapes of many debris objects are unknown. Thus when using this dataset, researchers have typically separated the objects into general classes as opposed to specific shape classes. Additionally, the quality of the dataset is mixed and it consequently requires significant pre-processing in order to remove poor quality light curves. Subsequently, Furfaro and Linares propose the idea of transfer learning as an option for increasing the classification accuracy on real light curve datasets without the necessity of obtaining a real world large, high quality and well labeled dataset [38]. Transfer learning refers to the situation where what has been learned in one setting (a CNN trained on a dataset) can be exploited to improve generalisation in another setting (a new smaller dataset) [42].

In 2021, Liu further expanded on the research presented in [51] by creating a neural network with combined convolutional and recurrent layers, which was then applied to real light curve data [86]. As in previous research, the MMT dataset was used with the objects were classified into four separate classes; non-rotating objects, rotating rocket bodies, rotating satellites and rotating debris. The distinction between non-rotating objects and other RSOs was made as their research determined that it was difficult to distinguish between non-rotating rocket bodies and actively three-axis stabilised objects [86]. The trained network achieved an overall classification accuracy of 86.07% on the test dataset without requiring any a priori information about the objects.

This research demonstrates that the developed model is able to separate rotating and non-rotating objects as well as being able to classify the rotating objects into three classes [86]. However, as there is no comparison made by the authors between the developed neural network and other models (i.e. traditional machine learning methods or a standard 1D-CNN) it is unclear how beneficial the combination of the convolutional and recurrent layers are in terms of model classification performance. In comparison with other research, the authors did report a notable improvement in accuracy on the real world MMT light curve dataset compared with the results reported by Furfaro and Linares [83]. Although it is difficult to make direct comparisons between the two methods, as it is unclear on the exact light curves that were selected from the MMT dataset and the fact that the selected light curves were organised into a different number of classes. Once again, this research classifies the objects into general categories rather than attempting to determine the shape of the object.

2.6 Deep Learning Approaches used in Other Domains

In the context of deep learning, the discussion of the literature presented so far has centred around deep learning techniques that have been applied in the domain of RSO characterisation. The following sections provide a brief review of key deep learning results and concepts that have been presented in other domains and are potentially relevant to RSO classification.

2.6.1 Time Series Classification

RSO shape classification from light curve data is categorised as a time series classification (TSC) task, with the aim to develop a model that is capable of taking an input light curve (time series data) and outputting a single predicted shape class. In 2019, a comprehensive review of deep learning for TSC was provided in [54]. This study found that while RNNs and in particular long-short term memory (LSTM)-RNNs have been effective in time series forecasting applications and language translation, they have been less widely used in classification tasks [54]. Although, RNNs are designed for sequence data, they are primarily used to predict an output at each step in the time series rather than a single classification result. In subsequent research, 1D-CNNs models have been demonstrated to achieve state-of-the-art performance on a number of TSC tasks [55, 100, 17].

2.6.2 Transfer Learning

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

[72]. 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 [89]. Networks are pre-trained on very large standardised datasets, enabling the earlier layers to learn generic basic image features such as edges and corners [88, 137]. 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 [94, 90]. In addition to imagery data, transfer learning has proven successful when applied to a range of other data types including hyperspectral data [130], MRI data [89], synthetic aperture radar data [93] and time sequence data [111]. 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 [93, 52, 133]. However, the effectiveness of transfer learning is known to be dependent on the similarity between the target/goal dataset and the dataset used for pre-training the model, with more similarity resulting in increased success [53].

2.6.3 Multi-Input Combination Methods

Typically, neural networks trained for classification tasks process a single input and output a predicted class for that input, with input in this case referring to a single example from the dataset (e.g. a light curve or an image). However, the fusion of multiple inputs within the network architecture, either inputs with different data types (e.g. image and text) related to the same object or inputs with different views of the same object (e.g. flower and leaf of a plant) has also been explored in the literature and found to significantly increase performance in a number of cases [120, 110, 134, 109, 119]. It is important to note that the multiple inputs still relate to a single output, with the idea to increase knowledge within the network about a specific example by providing additional information in order to improve model performance.

Seeland and Mader performed a systematic investigation into different model architectures and combination methods for three different multi-view image classification datasets [109]. The datasets were collected and organised so that for each example, there were the same amount of views and the views were collected from approximately the same perspective (e.g. the car dataset contained five separate views of each example; front, rear, side, front-side, front-rear). The authors found that a multi-branch approach with late combination of the feature vectors was the most effective approach, with the multi-view models outperforming single view models on all three datasets [109]. The authors note that the combination of multiple views can be particularly beneficial for challenging classification tasks where there is high inter-class similarity and low intra-class similarity [109].

2.7 Summary of Literature Review

RSO characterisation remains a relevant and unresolved problem in the field of SSA. A recent paper published in February 2018 estimated that attitude surveys through light curve inversion had only been performed for 400 hundred of the more than 17,000 objects catalogued by Space-Track [112]. The increasing amount of debris being observed and catalogued requires a solution that is able to scale well to the expected increase in the amount of data being generated. Deep learning techniques have shown remarkable results on a number of perceptual problems, particularly when provided with large amounts of labelled data. Recently, researchers in the RSO characterisation community have identified deep learning techniques as a potential method for assisting in RSO characterisation [79, 38, 83, 86, 51].

Research, thus far has primarily focused on the application of CNNs to simulated light curves [79, 38] however the potential of RNNs has also been investigated [57] and a combination of these two models [86, 51] as well as application of these techniques to real light curve data [38, 83, 86]. Although classification accuracy for basic shapes using simulated data has been high using a CNN, when applied to real data, the accuracy has significantly decreased [38, 83, 86]. It has been suggested that this is due to a lack of labelled real data as well as the fact that the simulations are less complex than the real data. This is supported by the fact that traditional machine learning techniques have shown comparable results to deep learning methods when applied to simulated light curve data, yet achieved significantly lower performance on the more complicated real light curve data [83].

As Furfaro and Linares identify, transfer learning may provide a potential method for increasing the classification accuracy on real light curve data, currently this has not been investigated thoroughly. Transfer learning has been shown to boost performance in other domains where training data associated with a specific task is limited [94, 90, 130]. For simulation-based transfer learning to be successful, a high-fidelity simulation is required that encapsulates the light curve features that are found in the real light curves. Limited research has been performed into the validity of current simulation techniques in comparison with observed light curve data. Subsequently, it may be necessary to implement a more realistic simulation environment in order to ensure that the neural network trained on the simulation data is able to learn features that are present in the real data.

Additionally, the combination of multiple observations (both multiple observation types [117, 116, 56] and multiple views of the same RSO [41, 70]) has been shown to provide additional information in the context of RSO characterisation. However, currently the fusion of multiple observations within deep learning models in order to improve model performance on RSO characterisation tasks has not been considered in the literature. In other domains, the combination of multiple inputs within the neural network has been shown to significantly increase performance for challenging tasks [120, 110, 134, 109, 119].

Chapter 3

Development of a High-Fidelity Light Curve Simulator

3.1 Introduction

The primary aim of this chapter is to develop a simulation environment that is able to generate light curves that effectively encapsulate the features that are present in real light curve data. This will enable the development of a large, well-labelled dataset that can be used for simulation-based transfer learning. Subsequently, the more realistic the simulation environment is, the more effective the transfer learning is expected to be. Additionally, the simulation environment will allow for controlled exploration and investigation into how changes to various properties effect the light curve. Subsequently, it should be developed in such a way that allows generated light curves to be easily compared with real world light curves.

In this chapter, the methodology for extracting light curves from ground-based optical telescope imagery data is also presented. This will enable validation of the developed simulation environment through comparisons between simulated and real light curve data. It is also important in the wider context of the thesis, as this method will allow for the development of a real world light curve dataset that will used during the transfer learning research presented in Chapter 4.

As outlined in the literature review, previous research into RSO characterisation using simulated light curves has typically applied a simplified FFM of RSOs combined with various BRDFs to model reflectance [128, 48, 83]. In the literature both the simplified anisotropic-Phong BRDF developed by Ashikhmin and Shirley [6] and the Cook-Torrance BRDF [22] have been used to simulate the resulting reflected light from the sun based on the orientation of the flat facets representing the RSO's structure [129].

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 or long camera exposures. To be effectively utilised for transfer learning on real light curve datasets, it is expected that the simulated light curves will need to encapsulate the features present in the real data and thus incorporate many of the effects outlined above.

By creating a more realistic simulation environment using recent advancements in rendering and computer graphics, it is hypothesised that the simulated light curves will encapsulate more of the features observed in real light curve data. This will enable the training of more accurate and robust RSO characterisation models using transfer learning. The specific contributions of this chapter are:

1. Development and use of a high fidelity 3D rendering based simulation environment for providing realistic light curve data.
2. Validation of the simulation environment through comparison with real data.

Section 3.2 presents an overview of the development of the simulation environment as well as providing details about the three main steps that are required for the simulation of a light curve. Section 3.3 outlines the process that was developed to automatically extract real light curves from optical imagery containing observations of RSOs (captured using Electro Optic Systems (EOS) telescopes). The results of the light curve extraction algorithm are presented in Section 3.4 and are compared with simulated light curves to validate the simulation environment. Section 3.5 provides a discussion of the work presented and Section 3.6 contains a brief summary of this chapter as well as explaining how this research fits into the wider context of the thesis.

3.2 Development of Light Curve Simulation Environment

A light curve simulator has been 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 Figure 3.1 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[1] and its built in physically based probabilistic ray-tracing engine Cycles[2].

---

[1] https://www.blender.org/

[2] https://www.cycles-renderer.org/

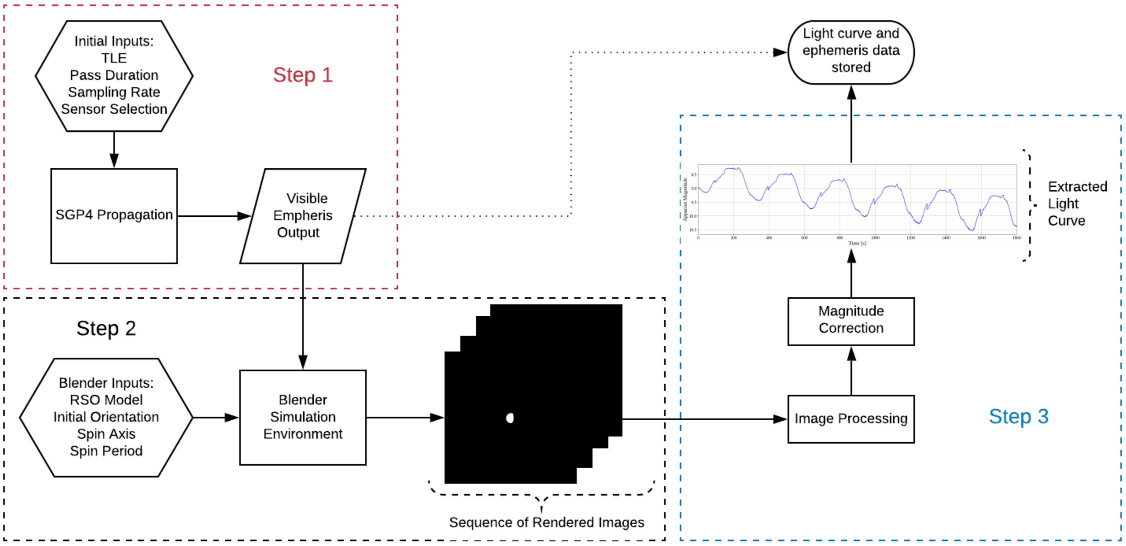


Figure 3.1 – Simulation Environment Setup: The Initialisation Step uses real world data for the sun, sensor and specified object to determine visibility and viewing angles. The 3D Rendering Step uses this information to generate a sequence of realistic images which are then processed in the Light Curve Extraction Step.

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.

3.2.1 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. This is both in order to increase the fidelity of the simulation as well as to reduce the difficulty in comparing real and simulated light curves. Subsequently, the position and motion of these three primary components are pre-calculated during the initialisation step and then input into Blender.

This initialisation step is performed in Python where a graphical user interface (GUI) allows the user to select the North American Aerospace Defense Command identification number (NORAD ID) of the object as well as the start epoch, pass duration, sampling rate and observing sensor. The two line element (TLE) for the object is automatically extracted from the Space-Track website[3] with the closest TLE prior to the start of the simulated pass selected. Simplified general perturbations 4 (SGP4) [121] is used in conjunction with the US Naval Observatory vector astronomy software (NOVAS) library [9] to propagate the TLE in order to obtain the position information for the sun, the observer and the RSO for a specified sampling rate. To simplify coordinate transformations, the origin of the Blender coordinate system was considered to be the centre of mass of the Earth, which is the origin of the true equator mean equinox (TEME) coordinate system. TEME was selected as it is the coordinate system used by both TLEs and SGP4.

At each timestep, a visibility check is performed to ensure that the RSO is both illuminated by the sun and visible to the observing sensor. The conditions used to determine visibility for a ground-based sensor are as follows:

- RSO elevation $> 15°$ above horizon from the perspective of the observing sensor. At lower elevations the reflected light travels through more of the atmosphere making it difficult to acquire the RSO and obtain accurate measurements.
- Sun elevation $< -6°$ below horizon from the perspective of the observing sensor. For the optical observations to be recorded the observing sensor should not be illuminated by the sun.
- RSO not in Earth shadow. The RSO must be illuminated by the sun for it to be visible to the observer. This was calculated through implementation of the conic shadow model provided by Hubaux [50].

When the object is determined to be visible to the observer the vectorised position information of the sun, RSO and sensor are recorded in the TEME coordinate system. If the RSO is not visible no information is recorded and the simulation is propagated forward in time until it becomes visible or reaches the end of the pass specified by the user. The recorded ephemeris information is used as input to initialise the second phase of the simulation environment, the 3D rendering step.

---

[3] https://www.space-track.org/

3.2.2 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 it would appear to the telescope/light curve sensor on the ground. Blender was selected as the rendering software as previous research has shown it to be an effective tool for simulating realistic light curves [62, 84]. Blender is an open source 3D rendering software, which has a built-in physically based probabilistic ray-tracing engine called Cycles. During the development of the simulation environment, Blender 2.80 was the most updated stable release, so it was the version used for the following simulations. More recent work has utilised updated versions of Blender, specifically versions 2.90 and 2.92.

Prior to the simulation of a pass in Blender, there are a number of configurable options 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 Cycles ray-tracing engine. This sequence of images is then passed into the light curve extraction step, which is the third and final step in the simulation pipeline.

The following subsections outline key settings that were selected in Blender in order to simulate the primary components.

Sun

To simulate the sun, a default Blender sun lamp was used, which provides light of a constant intensity emitted in a single direction from infinitely far away. Consideration was included for the colour of the sunlight by setting the black body shader node in Blender to a temperature of 5778K, which controls the colour of the light from the sun node. The intensity of the sunlight impacting the model was set to be 1062W/m^2. This was determined to be the intensity of the sunlight in the visible spectrum impacting on objects in Earth orbit [97]. No consideration is made for small changes in intensity at different altitudes and all incoming light rays are considered parallel.

Camera

Blender provides a default camera object with a number of options for customisation. This allowed multiple cameras to be created with the aim of simulating the various sensors that were used to collect the real data. These camera objects are then saved into the Blender file and can be selected during the simulation process. All cameras are set to record imagery in grayscale as is common for astronomical cameras used for tracking (e.g. EOS camera data used in experiments). Additionally, the OpenEXR format was used to output the simulated imagery with the gamma correction set to 1.0 and a linear transform used to prevent the corrections that Blender normally applies to image data for visual effect. The cameras were also simulated to have the same ground-based position as the telescopes used in our real-data acquisition (see Section 3.3 below), in order to facilitate comparison across the datasets.

Object Models

The GUI in Blender allows the modelling and visualisation of complex objects and is developed to handle highly detailed models with a range of materials and textures. In a similar manner to the cameras, object models stored in the Blender file format can be selected to be used during the simulation. The materials used for the models are also controllable during the simulation run time as opposed to having to store the same object multiple times to test the effect of different materials on the light curve.

Additionally, Blender supports a wide range of common 3D formats, allowing object models developed in other programs to be imported into the simulation environment. This has been useful for initial simulation testing as a number of models have been

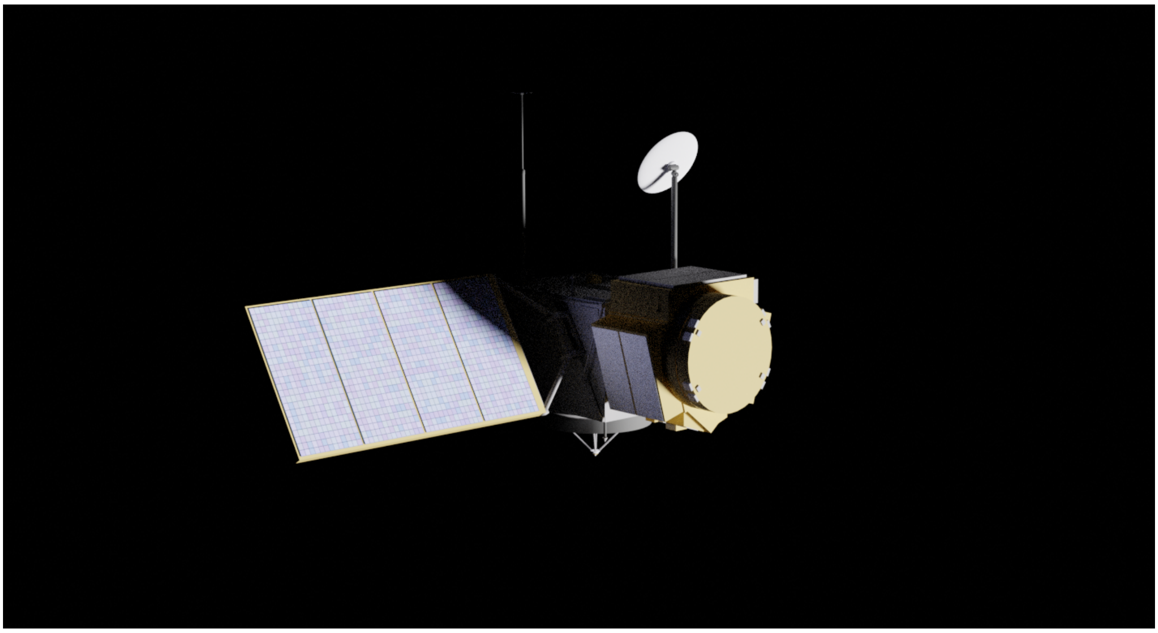


Figure 3.2 – Blender-based rendering of a model of the Topex satellite. The ray tracing engine used with Blender allows for complex effects (such as self-shadowing) which improves the fidelity of simulated light curve profiles.

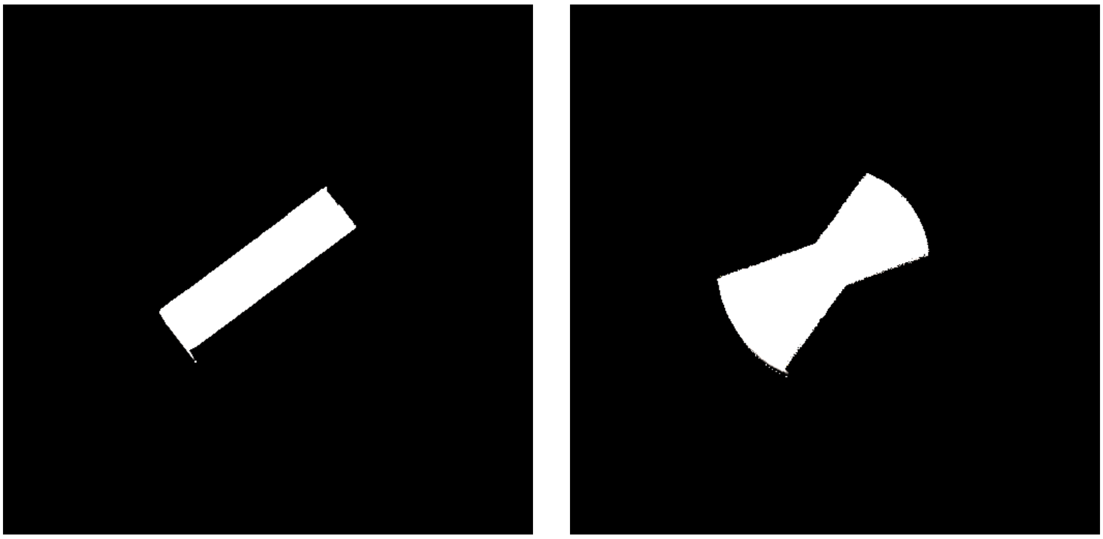
downloaded from the NASA 3D model website[4]. An example of one of these models rendered in Blender is depicted in Figure 3.2. Self shadowing and secondary reflections, where light is reflected from one surface onto another, are captured by the Blender ray-tracing engine which enhance the realism of the simulated measurements.

Motion Blur Implementation

During initial testing of the simulation it was determined that to emulate real light curves the object must continue to rotate during the image exposure as it does in the real world. Previous simulations using the FFM rotate the object to a certain orientation and use the normal vectors of the facets at this orientation to calculate the light reflected in the direction of the observer. However, for objects that are rapidly rotating or for simulations involving long exposures, the change in orientation of the normal vectors during exposure affects the intensity of the light reflected towards the

---

[4] https://nasa3d.arc.nasa.gov/models



(a) No Motion Blur **(b)** Motion Blur Implemented

Figure 3.3 – Example of a rotating cubesat with and without motion blur implemented in Blender. Both examples are imaged using a 1 second exposure and the same spin parameters however motion blur is only implemented in (b). Motion blur is an important consideration for fast rotating objects or when using a long exposure during image capture.

observer.

This effect can be simulated in Blender by applying animation to the object and motion blur to the scene. Figure 3.3 depicts a comparison between the rendering of an object with and without motion blur. The simulated object is a 3U CubeSat rotating around its x axis with spin period of 10 seconds and imaged using a 1 second exposure. For this comparison the distance in the simulation environment has been scaled to make the object resolvable in the image. Further analysis on the impact that this implementation has on the extracted light curve is presented in the results section.

3.2.3 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.

The apparent magnitude is a measure of the relative brightness of an object as seen by an observer on Earth above the atmosphere, based on an inverse logarithmic relationship defined by the following equation:

$$m = -2.5\,\log_{10}(\phi_q) \tag{3.1}$$

where m is the apparent magnitude and ϕ_q is the photon flux of the object received at the observer.

Consequently, very bright objects have lower magnitudes, such as the sun with a magnitude of -26.7, whilst dimmer objects have higher magnitudes.

As the RSO is the only object within the image, its brightness can be 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. Thus it can not be directly converted to apparent magnitude using Equation 3.1.

Calculation of Apparent Magnitude in Real World Telescope Imagery

In the real world, a similar issue occurs with the brightness of an object in the image depending on a number of factors including the optical observing system, background noise and atmospheric conditions. Due to calibration requirements, it is difficult to determine the exact photon flux captured by the observing sensor. However, the instrumental magnitude for an object can be calculated using the pixel counts in an image as outlined in [102]:

$$I_m = -2.5\,\log_{10}\left(\frac{N_{ap} - A_{ap}S_{sky}}{t_{exp}}\right) \tag{3.2}$$

where N_{ap} is the sum of the pixels in the aperture, A_{ap} the area of the aperture (in pixels), S_{sky} is the background sky signal and t_{exp} is the exposure time of the image (in seconds).

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:

$$m_1 = m_2 + I_{m_1} - I_{m_2} \tag{3.3}$$

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.

Thus in the real world the apparent magnitude of an RSO is determined by comparing its instrument magnitude with that of a star with a known apparent magnitude through the applications of Equation 3.2 and Equation 3.3. This process is known as photometric correction and it also has the benefit of correcting the brightness of the object for atmospheric attenuation as the apparent magnitude system is defined to be the brightness of the object as appearing to observer on Earth above the atmosphere. Without photometric correction, the measured brightness of the observed object would be dependent on the elevation as at lower elevations the reflected light would be travelling through more of the atmosphere to reach the observer. However, as light from the stars observed in the image with object is also travelling through the same section of the atmosphere, it will be equally affected and therefore the photometric correction process will remove the atmospheric attenuation effect.

Calculation of Apparent Magnitude in Blender Rendered Imagery

Based on the 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 AS BRDF applied to the FFM. Rather than producing imagery, the FFM takes the intensity of the sunlight as an input in W/m^2 and calculates the fraction of light reflected towards the observer using the AS 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 apparent magnitude by using the known value of the apparent magnitude of the sun for calibration. See [38] for further detail on this method where the authors state that the apparent magnitude of an object in the FFM simulation can be calculated as follows:

$$m = -26.7 - 2.5 \log_{10}\left(\frac{F}{C_{sun,vis}}\right) \tag{3.4}$$

where -26.7 is the value for the apparent magnitude of the sun, F is the calculated fraction of the reflected sunlight from the object that is visible to the observer, as determined by the BRDF and $C_{sun,vis}$ is the intensity of the sunlight in Earth orbit.

Subsequently, the FFM was used to create a reference point for converting the extracted sum of pixel intensities from the Blender-generated images into a corrected apparent magnitude that could be compared to real data. A simple 1×1 m flat plate with the same material settings, was simulated at a distance of 10,000 m with both Blender and the FFM using the same object, sun and camera geometry to ensure that reflective angles were the same.

Using the method outlined in [38] and through application of Equation 3.4, the apparent magnitude of the flat plate from the FFM was determined. The instrument magnitude of the object in the Blender simulation was then determined through Equation 3.2, which was simplified as the background sky signal (S_{sky}) is 0 in the Blender simulation. Similar to the method applied when determining the apparent magnitude of RSOs in real imagery, this object can then be used as a calibration object to determine the apparent magnitude of objects in Blender generated imagery.

However, in the real world, stars have a fixed apparent magnitude for observers on Earth, whilst the reference object created using the FFM is only accurate for the specific distance of 10,000 m as the intensity of the light is subject to the inverse square law. Thus, objects rendered in Blender at other distances will need to be corrected. This is done by dividing the extracted intensity by the square of the ratio of actual distance used in Blender for a specific object to the 10,000 m reference point.

A final correction must also be applied to account for the scale factor applied in Blender to the positions of the camera, sun and object. This scale factor was found to be required due to accuracy limitations in Blender when rendering dim objects at extreme distances. Thus, to prevent this from occurring a scale reduction of 1,000 was applied equally to the position information for all three components. This scale reduction is corrected for by dividing the measured Blender intensity by the square of the scale factor so that an accurate comparison can be made with real world data.

These corrections are applied to Equation 3.3, resulting in the final equation that was used to convert the sum of the Blender pixel intensities in an image to the apparent magnitude:

$$m_1 = m_{ff} - I_{m_b} - 2.5\log_{10}\left(\frac{B}{t_{exp}\cdot(\frac{d}{d_r}\cdot scale)^2}\right) \tag{3.5}$$

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 the 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 1,000 in order to reduce the distances used in the simulation, as the accuracy of Blender simulations decreases at extreme distances.

Using Equation 3.5, the calibrated apparent magnitude of the Blender rendered object

Figure 3.4 – EOS Site Locations, Site A is located at Mt Stromlo near Canberra on the east coast of Australia, while Site B is located at Learmonth on the west coast of Australia. Photographs provided by Ian Ritchie and Dr Francis Bennet

can be extracted from each image to produce a light curve from the sequence of images, allowing for direct comparison with real light curve data. As measurements are recorded in the apparent magnitude system, atmospheric extinction does not need to be incorporated into the simulation environment provided comparisons are made with light curves that are also recorded in the apparent magnitude system. The determined apparent magnitude was recorded along with the ephemeris information at each time step and the other parameters used in the pass. After processing, the Blender rendered images are deleted to reduce data storage requirements.

3.3 Extraction of Real Light Curves

As part of the validation process for the light curve simulation environment, an automated data processing pipeline has been developed to extract light curves from a sequence of EOS telescopes images. EOS is an Australian based company who operate six optical telescopes at two sites in Australia [11] and have provided the real observational data that is presented in this chapter. Figure 3.4 depicts the location of these two sites on a map of Australia as well as providing an image of both sites.

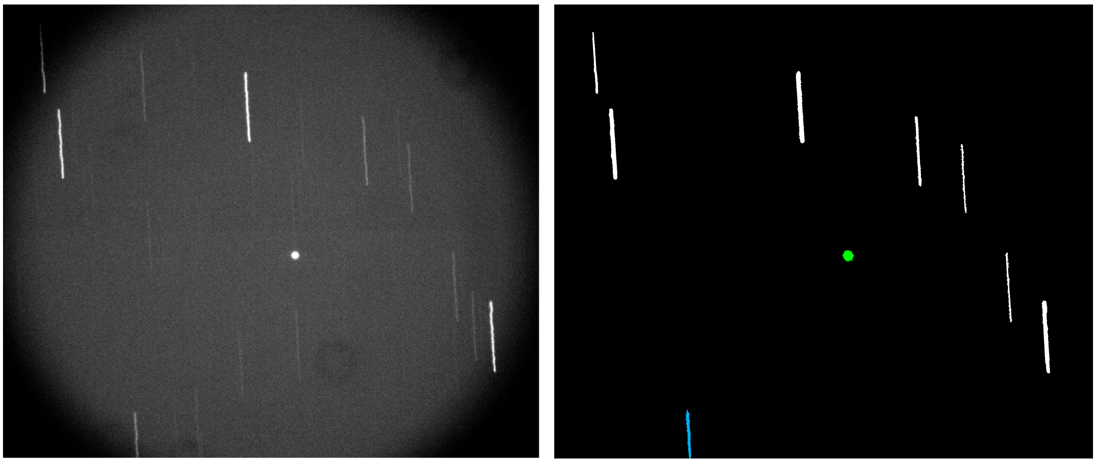
EOS telescopes typically operate in rate tracking mode, meaning that the telescope follows the apparent movement of a specific RSO. Consequently, the tracked RSO appears as an unresolved circular blob in the centre of the image. Other RSOs and stars can appear as either streaks or circular blobs depending on their relative in-frame velocity to the object being tracked, the movement of the telescope and the camera's frame rate. Figure 3.5 displays two examples of typical telescope imagery produced by EOS during object tracking as well as the output object detection masks which are produced as part of the light curve extraction process.

In Figure 3.5a the telescope is tracking a MEO object, which is clearly visible in the centre of the image. The movement of the telescope tracking the object causes the background stars to appear as elongated streaks in the imagery, enabling simple classification between the RSO and the background stars. A more challenging example is presented in Figure 3.5c, where the telescope is tracking a GEO object, which is again visible at the centre of the image. Additionally, in this image there is a second RSO that is located in the top right quadrant of the image. As the telescope is relatively stationary during the image exposure, the stars appear less elliptical and it is more difficult to determine if the objects in the image are stars or RSOs.

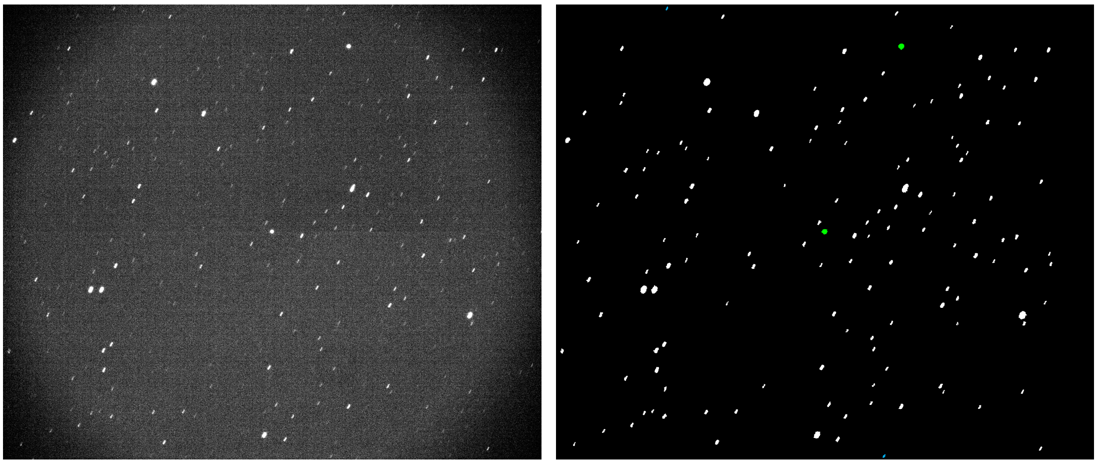
Upon examination of the EOS telescope imagery, initial consideration was given to utilising the orbital state of the tracked object to assist with the object detection. However, this approach would require an accurate orbital state and potentially restrict the algorithm to detecting RSOs that had previously been catalogued. Subsequently, it was determined that the algorithm should be able to operate without any provided positional knowledge of the object. While this makes object detection more challenging it enables the algorithm to detect uncatalogued RSOs.

The developed algorithm processes a sequence of EOS telescope imagery and extracts the apparent magnitude for the RSO in each image which is then recorded to generate the light curve for the pass. When multiple RSOs are detected within a sequence of images, a separate light curve is generated for each RSO.

The initial step in the algorithm is to detect the objects of interest (both stars and RSOs) within the image and calculate the instrument magnitude for these objects.



(a) Example 1: Raw Telescope Image **(b)** Example 1: Object Detection Mask



(c) Example 2: Raw Telescope Image **(d)** Example 2: Object Detection Mask

Figure 3.5 – Examples of non resolved RSO imagery collected by an EOS sensor and the object detection masks output but the light curve extraction algorithm. Example 1: Simple detection case with object in the centre of the image and stars visible as clear streaks. Example 2: More challenging case with stars appearing less elliptical and multiple RSOs present in the image.

As part of pre-processing, sensor specific noise is removed from the image using Equation 3.6 as outlined in [102].

$$\textit{Reduced Frame} = \frac{\textit{Raw Object Frame} - \textit{Bias Frame} - \textit{Dark Frame}}{\textit{Normalised Flat Frame}} \tag{3.6}$$

The open source Python library PhotUtils[5], is then utilised to produce a mask of all bright objects detected within the image and to provide statistics on each of these detections. This includes determining the area of the aperture for each identified object and the pixel counts within the aperture. With the bright objects identified in the image through the masking process, PhotUtils is then used to calculate a 2D gridded background sky brightness for the image. Using this information and the recorded image exposure time, allows for the conversion from pixel counts to an instrumental magnitude using Equation 3.2.

Using the provided telescope pointing information at the start and end of image exposure, the movement of the telescope is modelled to estimate both the expected star streak length and expected movement of stars between frames. This information combined with the movement of the detected objects within the image is used to classify objects as either RSOs, stars or noise. This process enables not only the detection of RSOs that appear as circular blobs but also the detection of RSOs that are streaking through the image. Once a star or RSO is initially detected in a frame, the position and brightness information are recorded enabling tracking of a specific star/RSO through a sequence of frames and assisting with subsequent detections.

Figure 3.5b and Figure 3.5d depict examples of the output detection mask after noise has been removed and the remaining objects have been classified as RSOs or stars. In these masks, RSOs are depicted as green while complete stars streaks with both endpoints visible are white. The blue streak in the bottom section of Figure 3.5b is identified as a partial star streak with only one endpoint visible. A secondary RSO in Figure 3.5d is correctly detected in the top right quadrant of the image.

---

[5] https://doi.org/10.5281/zenodo.4049061

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 as specific stars in the UCAC4 catalogue [136]. While the star mapping is typically performed on each image, it requires a minimum of four complete or partial star streaks with a visible endpoint to ensure accurate mapping. During LEO tracking, star mapping often occurs over multiple images before reaching convergence, as star streaks are significantly longer resulting in fewer stars being detected.

Once the star identification process is complete both astrometric and photometric correction is applied to the RSOs. The astrometric correction removes any telescope pointing error, while the photometric correction enables the conversion from sensor specific instrumental magnitude to apparent magnitude using Equation 3.3. Systematic photometry errors in the UCAC4 catalogue are estimated to be approximately 0.05 to 0.1 magnitudes [136]. Subsequently, to reduce errors caused by individual stars, a linear regression is performed on the magnitudes of all the identified stars in the image to improve the photometric correction performance. 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.

To reduce noise associated with background stars in the output light curve, the astrometrically corrected RSO location was cross-referenced with the UCAC4 catalogue. If any background stars brighter than magnitude 15 were within a 10” threshold of the RSO, the apparent magnitude of the RSO was not recorded for that image. This step was found to be necessary to ensure that sudden rapid increases in an RSO’s detected magnitude were actually caused by a change in the brightness of the RSO rather than the influence of a nearby star. Note that the determined threshold distance of 10” is specific to the telescope and camera parameters that were used to collect the real world observations. Subsequently, this value should be determined by the user when implementing this method on other systems.

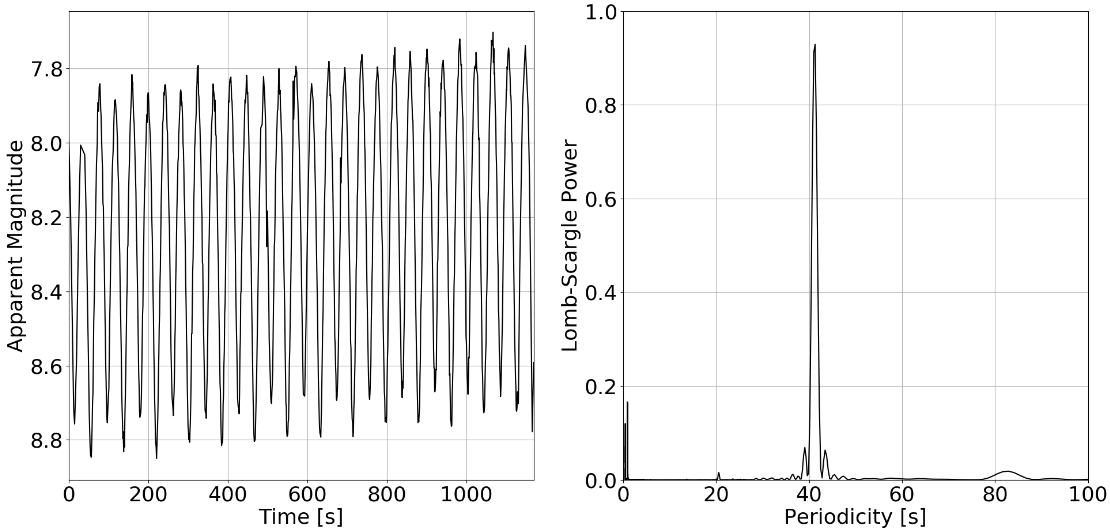


Figure 3.6 – Extracted Light Curve and Spin Analysis for a SpaceX Falcon 9 Stage 2 Rocket Body, NORAD ID: 40108. The light curve depicts a cyclical pattern, indicating it is rotating, with a variation in amplitude of approximately 1 magnitude between peaks and troughs. Lomb-Scargle analysis shows a strong peak at 41.3 seconds. This represents a half rotation of the rocket body due to its symmetry, so the spin period was determined to be 82.6 seconds.

3.4 Results

3.4.1 Extracted Real Light Curve

The left hand side of Figure 3.6 depicts a light curve that was extracted, using the algorithm outlined in Section 3.3, from observations of a SpaceX Falcon 9 stage 2 rocket body (NORAD ID 40108). The imagery was captured at an average sampling rate of 1Hz for approximately 20 minutes on the 7th of July 2019 using the 0.7m telescope at Mt Stromlo.

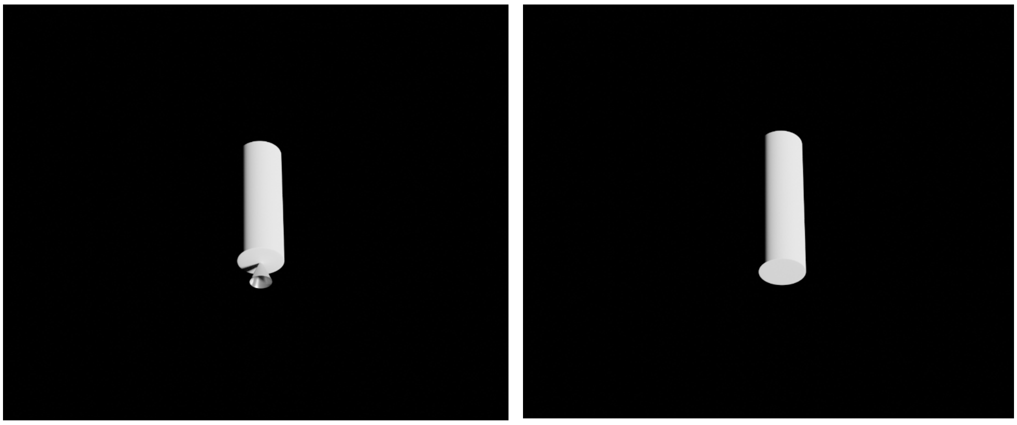
The cyclical pattern present in the light curve indicates that object 40108 is rotating. This was confirmed by applying the Lomb-Scargle analysis [87, 104] to the light curve, with a strong peak evident in the Lomb-Scargle power plot on the right hand side of Figure 3.6 at a period of 41.3 seconds. This period actually represents a half rotation of the rocket body, which, given its symmetrical nature, experiences a bright peak followed by a dim trough twice per rotation and thus has a spin period of 82.6 seconds. This can also be observed through the pattern in the dim troughs in the light curve visible in the lower section of the graph (higher magnitude) where every second trough is slightly dimmer. The consistent difference between these lower troughs indicates that the object is not exactly symmetrical and increases confidence in the extracted spin period of 82.6 seconds.

A further trend visible in the light curve plot is the general increase in magnitude throughout the duration of the pass. It is thought that this trend is a result of the gradual decrease in the phase angle measured during the pass. However, it could also be a function of a slow precession of the rocket body around a secondary axis. The phase angle is the angle measured between the direction of the incoming flux (light from the sun) and the direction of the observer.

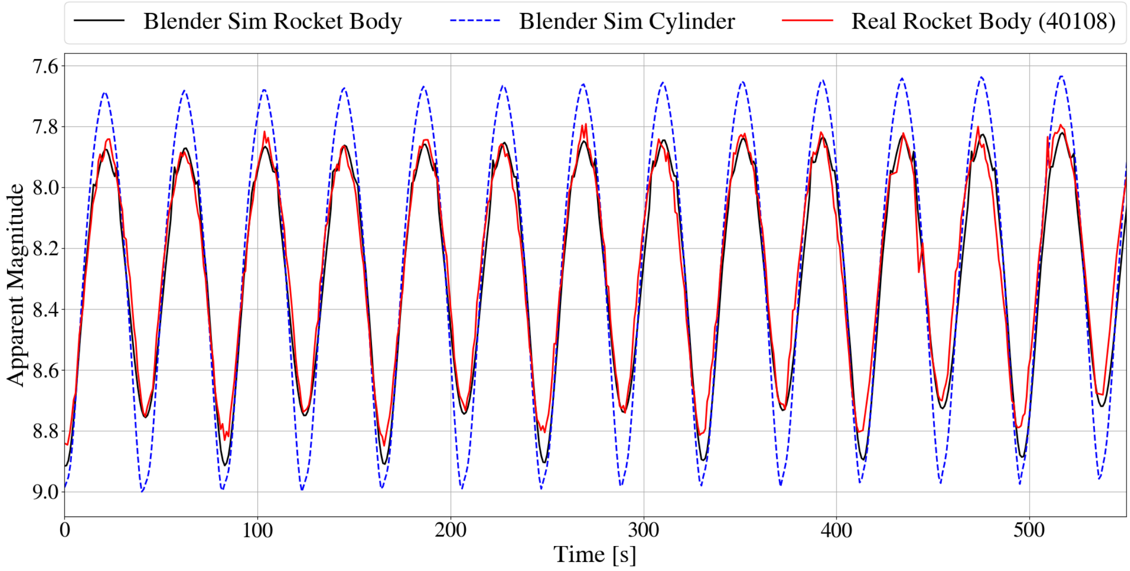
3.4.2 Comparison between Simulated and Real Data

To compare a simulated light curve with the extracted real light curve presented in Figure 3.6, a simple rocket body was modelled in Blender based on approximate dimensions of the second stage of the SpaceX Falcon 9 rocket. It was modelled as a cylinder with a nozzle extruding at the base and a small nose cone at the tip. A rendered version of the model used for the simulation is visible in Figure 3.7a. Additionally, a simple cylinder without the nose cone and the nozzle, depicted in Figure 3.7b, was simulated to be the same diameter and length as the SpaceX rocket body as a source of comparison.

The TLE for object 40108 was propagated in Python using SGP4 to the start of the observation period and positions for the rocket body, the observing telescope at Mt Stromlo and the sun were recorded at 1 second intervals for the duration of the observed pass. This information was then input into the Blender simulation environment and the SpaceX model was rotated on both its x and y axes separately using the spin period of 82.6 seconds determined by the Lomb-Scargle analysis presented in Figure 3.6.



(a) SpaceX Stage 2 Falcon R/B Model **(b)** Simple Cylinder Model



(c) Comparison between Blender Simulated Data and Real Light Curve

Figure 3.7 – Initial Validation of Simulation Environment. A good convergence was found between the simulated light curve for the rocket body (black) and the real light curve of the SpaceX rocket body extracted from EOS imagery (red). The simulated light curve of the simple cylinder model is included for comparison (blue). The models used in Blender for the rocket body and the simple cylinder are displayed in (a) and (b) respectively.

A good convergence was found with the real light curve for the case when the simulated SpaceX model was rotated about the x axis. This light curve is presented in Figure 3.7c along with the extracted light curve and the simulated light curve from the cylinder rotated around its x axis. In particular, note that the pattern identified earlier where every second trough is dimmer is present in the simulated light curve for the rocket body. Based on the simulation these troughs were found to correspond with the nozzle of the rocket facing the telescope, whilst the slightly brighter troughs (first evident at an approximate apparent magnitude of 8.75 at 40 seconds) occur when the nose cone of the rocket is facing the observer. The peaks on the graph correspond to when the full face of the cylindrical part of the rocket body was observable to the telescope.

Whilst the light curve for the simulated simple cylinder has a similar oscillation to the other light curves, the pattern present in every second dip is not evident. The cylinder has a higher apparent magnitude at the peaks as the length of the cylinder used was the total length of the rocket body including the nozzle and nose cone, so the face of the cylinder has a larger area than the face of the rocket body. The nozzle and nose cone for the rocket body were also found to be more reflective than the ends of the cylinder when facing the camera resulting in lower dips in the light curve of the cylinder. Additionally, the peaks in both the rocket bodies light curves appear to have small features and disturbances whilst the cylinder has smooth peaks. This is hypothesised to be as a result of side-on reflections from the nozzle which are present for the rocket body but absent from the cylinder simulation.

Furthermore, all three light curves also feature the general trend of increased brightness that was identified previously. This indicates that this trend is more likely to be a result of the decreasing phase angle than a complex spin effect, as both the simulated objects are only rotated around their x-axis.

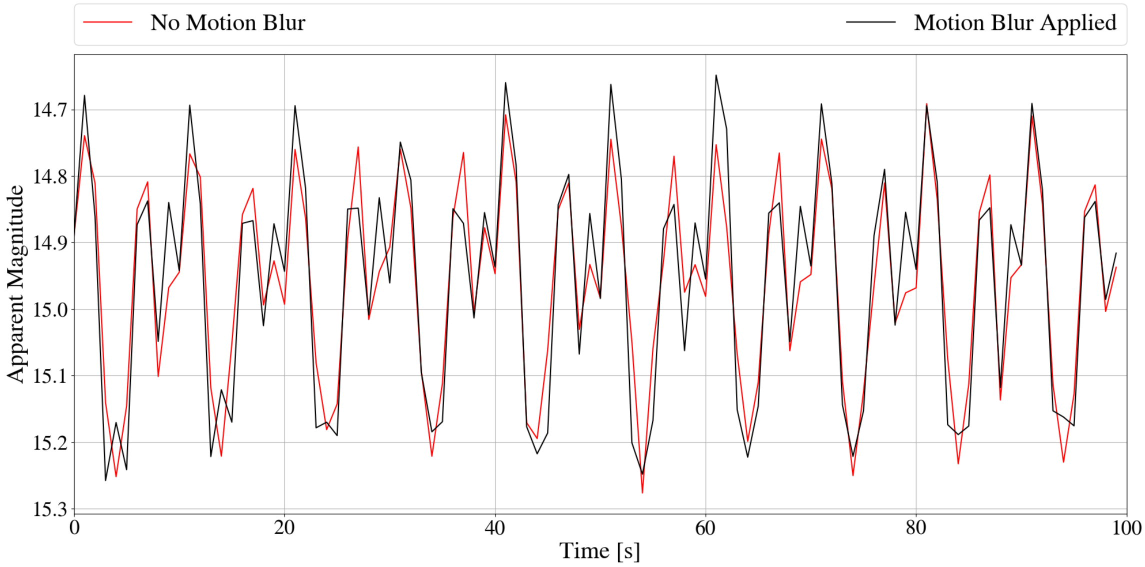


Figure 3.8 – Light Curve Comparison of Motion Blur Implementation (3U CubeSat). While the same general shape is present in both light curves, small differences are observable between the two cases. Of note is the peak that is apparent at approximately 10 seconds in the black light curve (motion blur applied) but not in the red light curve (no motion blur).

3.4.3 Motion Blur

Figure 3.8 depicts a comparison between two light curves generated using the Blender simulation environment for the case when motion blur is applied compared with no motion blur. The light curves shown are extracted from a simulation of the 3U CubeSat depicted in Figure 3.3, rotating around its x axis with spin period of 10 seconds.

While the general shape and pattern present in both light curves is quite similar, small differences are clearly observable in a number of the light curve features. This is particularly evident at approximately 10 seconds into the pass where a distinct black peak is visible in the motion blur applied light curve that is not captured by the light curve generated without motion blur. This peak is clearly visible in both light curves at other points during the pass in both light curves indicating that it is not a spurious peak. Additionally, further differences are notable between the two light curves in the magnitudes of the brighter peaks and the shape of the dimmest troughs.

3.5 Discussion

Although additional work is required to validate the Blender simulation environment, initial comparisons with rocket body light curves show good similarity between the real extracted data and the simulated data. This close similarity, particularly when contrasted with the differences to the simulated light curve of the cylinder, emphasises the importance of using more complex models to generate realistic light curves. Although the general shape of the cylinder is the same as the rocket body, the addition of the nozzle and nose cone has a significant effect on the resultant light curve.

It should be noted that the troughs corresponding to the nozzle are slightly dimmer in the simulation than in the real light curve. This potentially indicates that the size of the nozzle should be increased or a more reflective material should be used in the simulation in order to better represent the real object. Future development of the simulation environment will include creating realistic models of other space debris and performing further validation with extracted light curves to ensure that realistic light curves are being generated for a variety of shapes and models.

The inclusion of self-shadowing in the Blender modelling will allow for the simulation of more realistic light curves for convex objects and objects with large extrusions (e.g. solar panels). For these kinds of objects significant portions of the satellite may be shadowed by the solar panel or the body of the satellite may prevent reflections from the solar panel from reaching the observer. Additionally, motion blur has been shown to have an effect on the generated light curve, particularly for objects that are rapidly rotating or when longer exposures are used. Whilst these may seem like small details, it is expected that these are the kind of features that will allow a machine learning algorithm to differentiate between light curves associated with different shapes and models.

The visualisation aspect of the GUI built into Blender has also proved useful in analysing how small changes in the model affect the light curve. This was evident in the work with the rocket body where analysis of the simulation allowed the orientation of the rocket body to be determined when specific features in the light curve were generated.

In terms of current differences between the simulated light curves and extracted light curves, images are currently simulated with no gap in exposure between images. This differentiates them from EOS sensors which have a slight gap between images primarily to allow for readout time. Subsequently, the real light curves may be missing information compared with the simulated data, particularly for objects with a high spin rate. Furthermore, the EOS sensors currently use an adaptive algorithm to determine the exposure time to use during a pass depending on the brightness of the object and the mode of the telescope. This means that the exposure time sometimes changes during the course of a pass. Future analysis will be done to determine whether either of these differences have a significant effect on the simulated light curve and if so they will be replicated in the simulation environment.

Finally, comparative to the FFM, one of the detriments of the Blender simulation environment is that it is significantly more computationally expensive. Tests on a CPU with 12 cores simulating a 30 minute pass resulted in the generation of approximately 1 light curve per hour, while a NVIDIA GeForce 1080 Ti graphics cards was able to generate a light curve in 15 minutes. The simulation environment has now been modified to be run in parallel on the GPU, enabling an improvement to 10 light curves per hour per GPU.

3.6 Summary

A high-fidelity Blender based simulation environment has been developed for the generation of light curves of rotating space objects. The simulator allows complex geometric models as input and includes self-shadowing as well as motion blur. Initial validation through comparison with a real light curve of a SpaceX stage 2 rocket body show a close similarity between the simulated and real light curve. Further validation will be performed in the future comparing simulated light curves with real light curves for other objects.

In the wider context of the thesis, the simulation environment developed in this chapter enables the generation of a large, well-labelled dataset of realistic light curves. This dataset will be used in order to train a complex neural network model with the aim of performing simulation-based transfer learning to improve classification performance on real world light curve datasets. Increasing the similarity between the simulated and the real data reduces the requirement of extensive retraining during the transfer learning process when applying the network to real data. The simulation environment also provides an avenue for exploring the relationship between RSOs and their light curves in a controlled environment and without the requirement of obtaining real world telescope time.

Chapter 4

Simulation Transfer Learning for Space Object Characterisation

4.1 Introduction

It has been demonstrated in the research reviewed in Chapter 2 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.

In this chapter, a novel approach to data-driven RSO characterisation is developed 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.

This chapter builds on the research presented in Chapter 3, by using the developed simulation environment to generate a high-fidelity simulated light curve dataset. This simulated dataset is then used for the purpose of pre-training a neural network in order to improve classification results on a real light curve dataset.

The key contribution of this work is to demonstrate a framework for simulation transfer learning from light curve data, in which the initial model is trained using a high fidelity simulation of the real task (i.e. simulated light curves of RSOs). In the developed approach, 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". Results illustrate that this approach out-performs models for RSO characterisation that were trained purely on either the real dataset, or simulation data alone. The specific contributions of this chapter are:

1. Development of a CNN model for RSO shape classification that performs effectively on both simulated and real data.
2. Demonstration that transfer learning improves classification performance on real light curve data and reduces the need for large well-labelled real light curve datasets.

The methodology for this approach is provided in Section 4.2 including the development of the classification model, an overview of the relevant datasets and an explanation of the transfer learning process. The results are presented in Section 4.3, 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 4.4 these results are discussed and compared to previous results in the literature. Finally, an overview of the key findings and conclusions are provided in Section 4.5.

4.2 Methodology

4.2.1 Problem Setup

Light curve data is univariate time series data, with observations of the RSO's apparent magnitude recorded at specific time steps. When time series data has a uniform sampling rate, the measured data can be input directly into a neural network with no additional pre-processing steps required. However, many real sensing systems use an adaptive sampling rate based on the brightness of the object relative to the background noise in order to improve RSO detection and tracking, which results in non-uniform sampling of the light curve data. Even in cases where the collecting sensor has a fixed sampling rate, the extracted light curve will generally have missing observations resulting in non-uniform sampling due to star overlaps, intermittent cloud and rapid dimming of the RSO.

To overcome this issue, interpolation was considered in order to convert the data to a uniform sampling rate during post-processing. However, as a result of the combination of the uneven sampling rate and the missing data, it was determined that interpolation would require too many assumptions about the structure of the data, which could result in the introduction of errors and biases that were not properties of the original data.

Another option that was investigated was the transformation of the data into the frequency domain, using the Lomb-Scargle Periodogram [87, 104] to obtain the power spectrum as a function of the frequency. While this resolves the uneven sampling issue, it was found during initial testing that as a result in the high variance in spin periods between objects, information was condensed at different regions of the frequency domain. This required either selective sampling of the frequency domain to obtain an appropriate frequency window for each light curve or inputting data that was sparsely sampled, with only small sections containing useful information.

In order to simplify the data pipeline and reduce the risk of human selection adversely impacting the results, it was determined that the time data should be input directly into the neural network along with the magnitude data. This enables the network to learn features that will identify when there are gaps in the data or changes in the sampling rate and take this into account. In addition to the apparent magnitude and time, it was decided that the phase angle and object range information should also be input into the network at each time step. This is because aside from the RSO's physical characteristics, these are the two parameters that affect the brightness of the object.

Subsequently, at a given time step t, an observation z is recorded containing four separate measurements:

$$z_t = \{m_t, dt_t, \phi_t, r_t\} \tag{4.1}$$

where m is the apparent magnitude of the object, dt is the time in seconds since the previous observation, ϕ the phase angle and r is the range of the RSO from the observing sensor.

The input light curve to the network, $\mathbf{x}$ is created from a series of observations collected sequentially:

$$\mathbf{x} = \{z_1, z_2, ..., z_T\} \in \ \mathbb{R}^{4\times T} \tag{4.2}$$

where T is the total number of unique time steps where observations were collected.

The task of light curve shape classification is to find the predictive function f that maps in an input light curve $\mathbf{x}$ to a corresponding unique shape class label $y \in \{1, ..., C\}$ such that:

$$f : \mathbf{x} \rightarrow \{1, ..., C\}, \tag{4.3}$$

where C is the number of possible unique shape classes.

Thus, as there are four measurements at each observation timestep, the input to the network is multivariate time series data as opposed to univariate time series data.

4.2.2 Classification Model Development

In the literature presented in Section 2.6, it was found that there are two deep learning model types, 1D-CNNs and RNNs, that have commonly been used as the predictive functions for RSO shape classification, as well as various combinations of these model types. However, the overview of the deep learning on the general domain of TSC in Section 2.6.1 suggests that RNNs are more applicable for time series forecasting applications, while 1D-CNNs are better suited for TSC. Subsequently, in this chapter, a 1D-CNN architecture was selected as the primary model architecture.

One-Dimensional Convolutional Neural Networks

1D-CNNs are typically comprised of a series of convolutional layers, followed by fully connected layers. The convolutional layers contain a specified number of filters of a certain width, which are convolved with the input to extract features. As the filters are generally shorter than the input, they act as a sliding window, commencing at the start of the input and convolving with each section as they move towards the end of the input. Equation 4.4 depicts the general form of a single convolution with one filter for a section of univariate time series input with a centered timestamp t as provided in [54]:

$$Conv_t = f(\omega * X_{t-l/2:t+l/2} + b) \mid \forall \; t \in [1, T] \tag{4.4}$$

where $Conv$ is the result of a convolution applied on a univariate time series X of length T with a filter ω of length l, a bias parameter b and a final non-linear function f (e.g. ReLU), with $*$ denoting the convolution operator.

In the case of the univariate time series data, the filter is one-dimensional (time). This formulation can be extended to multivariate time series data by adding a second dimension to the filter that is equal to the number of measurements at each timestep (i.e. four for the input light curve data).

Both the weights in the filter ω and the bias b are trainable parameters, which are updated using backpropagation and gradient descent-based optimisation in order to extract the most informative features from the input. It should be noted that when multiple filters are used for a specific layer, each filter learns its own set of weights and subsequently extract different features from the input. As the light curve is passed through the series of network layers, first convolutional and later fully connected layers, it is transformed into a tensor of extracted features, initially related to small local regions of the light curve. The further the light curve progresses through the network, the larger the receptive field becomes and subsequently the features encode more information.

For a classification task, the final output layer of the network is a fully connected layer with the same amount of neurons as the number of unique classes C for a specific dataset. Softmax activation is applied to the output of this layer to convert the values into a normalised probability distribution and determine the predicted class y for a specific input light curve $\mathbf{x}$. For further discussion and detail on the operations of convolutional layers in neural network architectures, the reader is referred to [42].

Overview of Selected Network Architecture

The network diagram of the 1D-CNN that has been selected for this chapter is displayed in Figure 4.1. This diagram also provides information about various network spatial parameters, such as the number and type of layers utilised as well as the size of specific layers. These parameters were 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 real and simulated light curve datasets before the final selection of the network parameters, which are provided in Table 4.1 along with the parameters that were considered during the grid search. While these parameters are not optimal for all

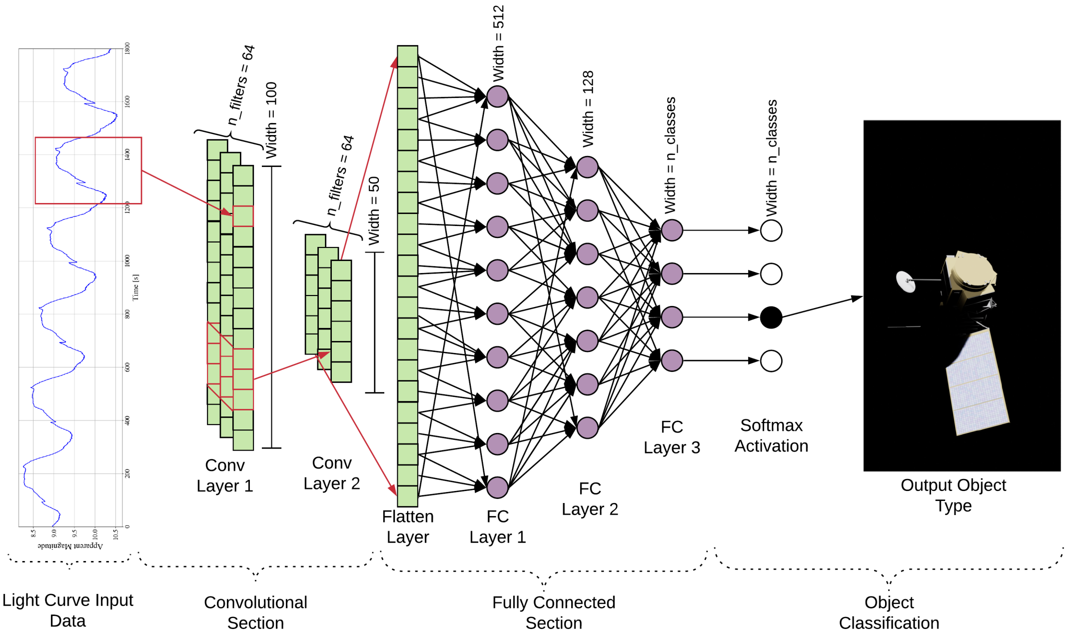


Figure 4.1 – 1D-CNN Architecture for RSO Shape Classification from Light Curve Data: The first section of the network consists of two 1D-convolutional layers, with each convolutional layer followed by a max pooling and batch normalisation layer. A flatten layer is then used to enable the feature vector to be input in to the fully connected section of the network. Three fully connected layers are used with softmax activation implemented on the final output layer of the network to output the model prediction. ReLU activation is used in all other layers

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

Table 4.1 – Selected Network Parameters for the 1D-CNN Model

	Selected Parameters	Parameters Considered in Grid Search
Convolutional Layers	2	[1, 2, 3, 4, 5]
Fully Connected Layers	3	[1, 2, 3, 4, 5]
Conv Layer 1: Filter Width	100	[20, 50, 100, 150, 200]
Conv Layer 2: Filter Width	50	[20, 50, 100, 150, 200]
Number of Filters	64	[10, 25, 50, 75, 100]
FC Layer 1: Width	512	[64, 128, 256, 512, 1024]
FC Layer 2: Width	128	[32, 64, 128, 256, 512]
Max Pooling Size	4	[1, 2, 3, 4, 5]

the fully connected layers.

Dropout is a regularisation tool that is commonly used during the training of a deep neural network to help limit overfitting [118]. For each new batch, neurons are randomly switched off at a user specified rate, preventing the model from relying too heavily on specific features and promoting generalisation. In inference mode, dropout is not used as it would increase the variability in the output classifications, as each input would be processed by a slightly different model.

ReLU activation is used for both the convolutional and the fully connected layers, with the exception of the final output layer. 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.

4.2.3 Overview of Light Curve Datasets

In this chapter, three light curve datasets have been used to evaluate the developed shape classification model and determine the effectiveness of simulation transfer learning:

- **Blender Simulated Dataset**: This is a simulated dataset that was generated using the simulation environment developed in Chapter 3.
- **EOS Dataset**: This a dataset of real light curves that have been extracted from EOS telescope imagery using the method presented in Section 3.3.
- **MMT Dataset**: This dataset has been sampled from a publicly available database[1] of real light curve observations of satellites and space debris collected using the MMT automated telescope system [14], which is located in the Caucasus mountain range in southern Russia.

The following sections provide a brief overview of each of these datasets.

Blender Simulated Dataset

The Blender based simulation environment has been used to generate a dataset containing 4,500 light curves for 13 different satellite models. These models were selected based on the expected object shapes of RSOs, with groups of similar shaped objects (e.g. cylinder and rocket body) included in order to make the classification task more challenging. Rendered images of these models are depicted in Section A.1. Typically, models contain a variety of materials with different reflectivity properties. These reflectivity properties were set to fixed specific values in Blender for each material based on real world values for similar materials and the effect of aging is not considered. A brief examination of the effect of variations in the material properties for specific object components on the output light curve is presented in Section A.2.

---

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

The light curves were sampled at 1 second intervals for the duration of a 30 minute track, resulting in 1800 observations per light curve. A number of TLEs for various RSOs with different orbital parameters were used during the initialisation step in order to simulate passes with a variety of phase angles and range conditions. Additionally, the starting orientation of the model was generated randomly, while the rotation period was sampled from a range of five possible integer values between 10 seconds and 300 seconds.

In this research, all rotation is simulated to be around a single axis which is randomly selected from one of the three primary body axis of the object and a stable rotation using the selected rotation period is then applied to the selected 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 in observations of rotating debris objects. Currently, the more challenging rotation profiles of tumbling and very slowly rotating objects has not been included in this dataset, however these will be an area of focus in future research as well as the rotation profiles of active objects.

The Blender simulated data was initially generated with a uniform length and 1 second sampling rate. 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 1Hz, 30% set to 0.5Hz and 10% set to 0.33Hz 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.

EOS Dataset

Commencing in June 2019, a tracking campaign was conducted for the purpose of light curve analysis with EOS sensors collecting imagery on a number of different RSOs. Using the light curve extraction algorithm presented in Section 3.3, more than 2,000 light curves were extracted from EOS imagery collected between June 2019 and August 2020.

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 [124]. 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 4.3.1.

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). 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 ten light curves was required for a shape/configuration class to be included in the final dataset used for classification as a number of RSOs were only tracked once or twice during the collection period.

The resultant 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 4.5 in the Section 4.3.2 depicts the composition of the EOS dataset for the 9 different configuration classes.

MMT Dataset

The MMT is an automated telescope system located in Russia that has been in operation since 2014 [14]. Light curve observations of satellites and space debris collected from this system are publicly available and have been previously used in the literature for real light curve classification [83]. In this case, the authors 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 [83].

In this thesis, 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.

MMT Balanced Dataset

As a result of the high class imbalance in the MMT dataset, 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 eight configuration classes. This dataset was used to test model classification performance on a more balanced dataset and is referred to as MMT Balanced for the remainder of this thesis. Similar to the EOS dataset, the MMT dataset is also skewed towards the observation and analysis of rocket bodies. Subsequently, of the eight largest classes in the MMT Balanced dataset, seven are various types of rocket bodies and the remaining class is a box wing configuration.

4.2.4 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. Figure 4.2 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

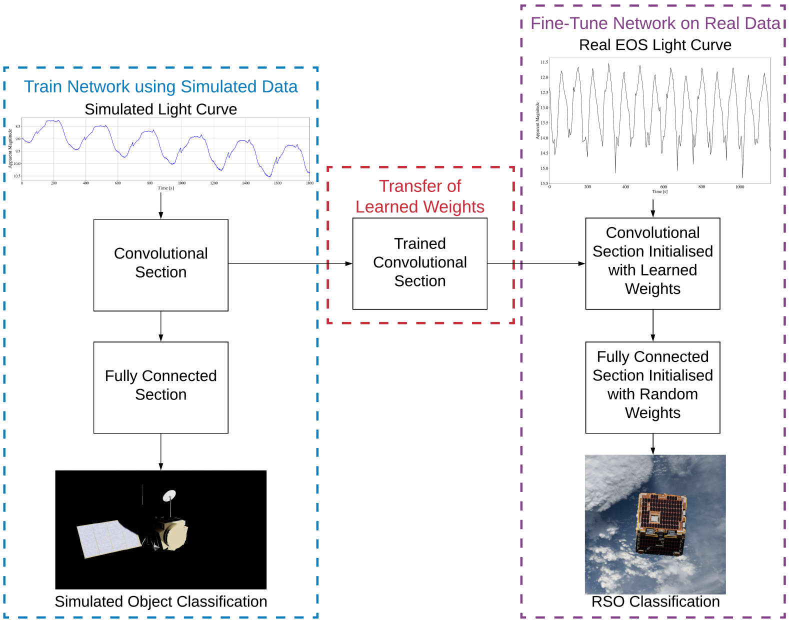


Figure 4.2 – Transfer Learning Process with EOS Dataset as Target/Goal Dataset and Blender Simulation Data used for Pre-Training

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 Figure 4.2 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 4.2.5 and the parameters that were selected for each configuration are presented in Table 4.3.

4.2.5 Model Setup and Training Hyperparameters

There are a number of training hyperparameters that are configurable during the model setup and training process. During training, the Adam optimiser [67] was used, with Table 4.2 displaying the main hyperparameters that were selected for each dataset. These hyperparameters were primarily 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.

Table 4.2 – 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 Ratio	0.3	0.3	0.3	0.3
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

K-fold cross 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 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 4.3.

Table 4.3 – Layers Transferred and Fine-Tuneable for each Pre-Training and Goal Dataset Combination

Pre-Training Dataset	Fine-Tune/Goal Dataset	Layers Transferred	Fine-Tuneable Layers
Blender Sim	EOS	2	5
MMT Balanced	EOS	2	4
Blender Sim	MMT Balanced	4	1
EOS	MMT Balanced	4	1

4.3 Results

4.3.1 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. This section includes an analysis of a light curve extracted from EOS imagery as well as an examination into

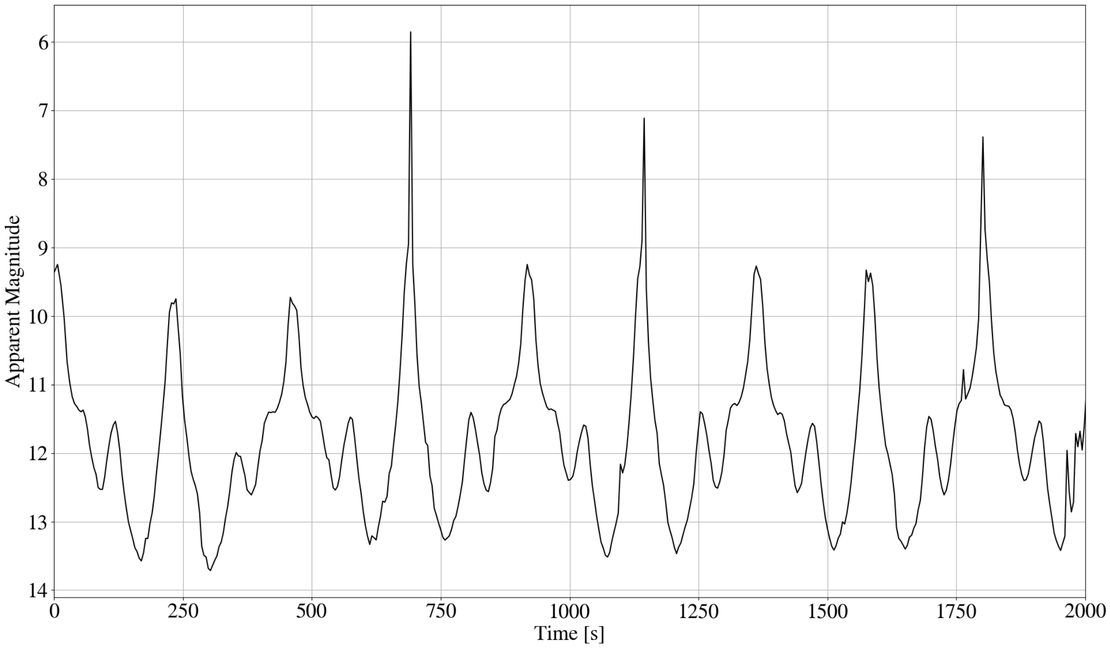


Figure 4.3 – Extracted Light Curve for MEASAT-3; NORAD ID: 29648, collected on the 28th of July 2021. Rotation period is determined to be approximately 428 seconds.

the dependence of apparent magnitude on phase angle and finally a brief investigation into the spin period stability of rotating rocket bodies.

Analysis of Collected Light Curve

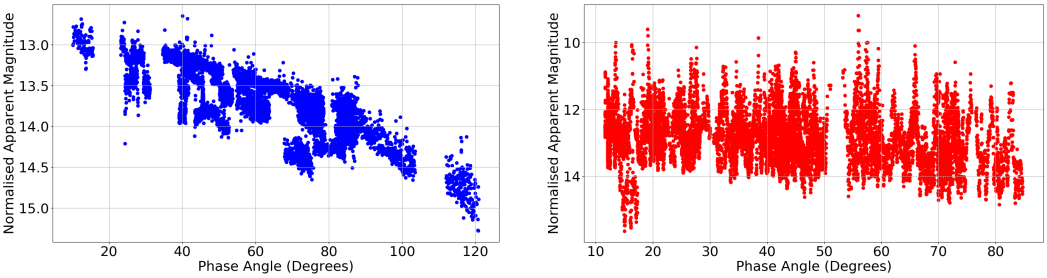
Figure 4.3 depicts an example of a light curve extracted from EOS imagery using method outlined in Section 3.3. This is the light curve of MEASAT-3, a non-operational communications satellite that is clearly rotating resulting in a repeating pattern in the light curve. It can be difficult to determine the rotation profile of a complex object from a light curve as there are often multiple possible solutions due to the number of unknown parameters including solar angle orientation, axis of rotation, orientation of the object, spin period and material reflectivity properties. In this case, additional light curves were collected for the object and further analysis was performed in the Blender simulation environment using a MEASAT-3 model. Subsequently, it was determined that MEASAT-3 was most likely in single axis rotation, with a rotation period of 428 seconds indicated by the Lomb-Scargle analysis.

MEASAT-3 has a standard box-wing configuration, with a rectangular solar panel on either side of the main satellite bus and a number of large circular antennas attached to the bus. One full rotation of the object can be observed between the strong peaks at approximately 700 seconds and 1130 seconds. The peaks in the apparent magnitude plot occur when the object is brightest (low apparent magnitude) and represent periods where both the solar panels and the satellite bus are reflecting sunlight towards the observer. These peaks can be observed twice per rotation, with one at the start/end of the pattern and one in the middle. An example of this middle peak is visible at approximately 900 seconds.

It is interesting to note that the peaks at the start/end of the rotation (700 seconds and 900 seconds) are relatively short while the middle peak is wider with two additional smaller peaks on either side. These are believed to be a result of the light reflected from the large circular antennas that are attached to the satellite bus. It is expected that a neural network will be able to learn distinguishable features within light curves such as these in order to differentiate between different shape classes and object configurations.

Relationship between Phase Angle and Apparent Magnitude

Figure 4.4 examines the relationship between the phase angle and an object's magnitude based on real world observations of two RSOs. 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 Figure 4.4a. 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.



(a) Etalon 1: Spherical Object **(b)** Optus B1: Box-Wing Satellite

Figure 4.4 – 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.

For the more complicated box-wing shaped Optus B1 satellite, displayed in Figure 4.4b 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.

Spin Period Evolution of Rotating Rocket Bodies

RSOs that are in a stable rotation state typically produce light curves with a cyclic pattern, such as the light curve depicted in Figure 4.3, which reduces the intra-class variability of their light curves. In contrast, objects that are tumbling or have rapidly changing rotation periods can produce a wide array of light curves making classification more difficult. Figure 4.5 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

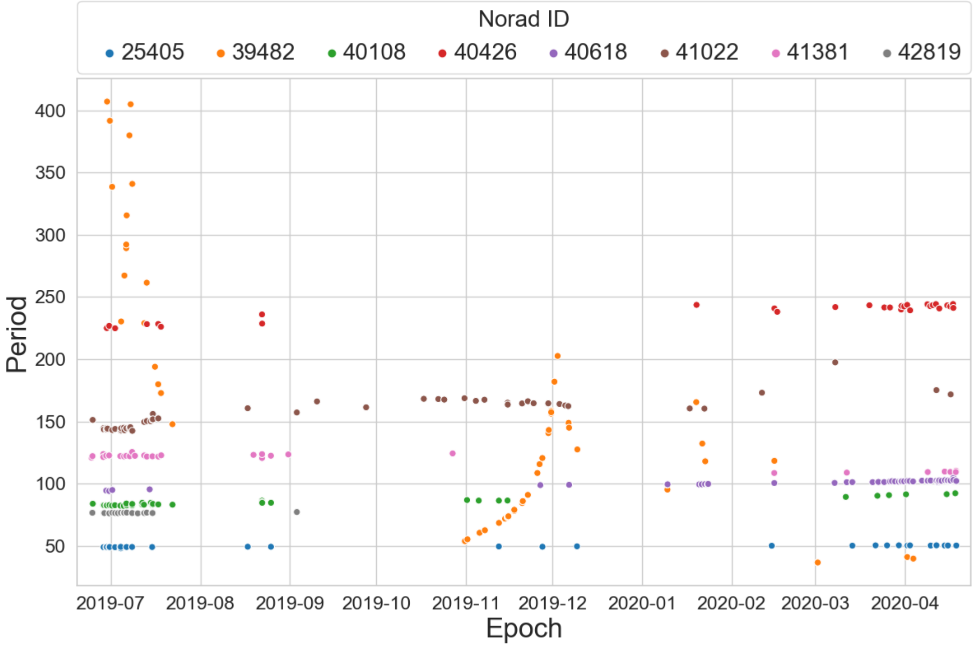


Figure 4.5 – Rotation Period Evolution of Rocket Bodies: Results are obtained from EOS light curve data collected over a period of approximately 10 months

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 of rocket bodies are in a stable rotation state.

4.3.2 Object Classification

Three different types of supervised machine learning classification models were evaluated on the three light curve datasets. A 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 4.4 for each model and dataset.

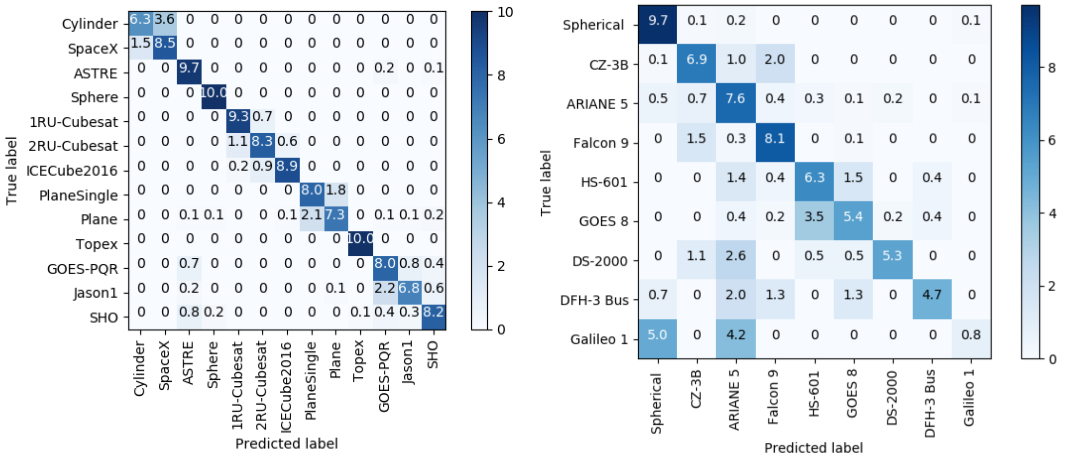
Table 4.4 – Comparison of Different Machine Learning Techniques: Mean 5-Fold Cross Validation Accuracy and Standard Deviation (each fold using 80% training and 20% holdout)

	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

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.

Figure 4.6a 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.



(a) Blender Simulation Dataset **(b)** EOS Dataset

Figure 4.6 – 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

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. Figure 4.6b 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 4.5 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

Table 4.5 – EOS Dataset Class Analysis: Cross Validation Results for 1D-CNN Model

Class	F1 Score	Precision	Recall	Unique RSOs	Light Curves	% of Dataset
Spherical	0.94	0.97	0.91	3	188	20.1
CZ-3B	0.70	0.69	0.71	3	166	17.7
ARIANE 5	0.76	0.76	0.76	4	213	22.8
Falcon 9	0.79	0.81	0.77	5	193	20.6
HS-601	0.64	0.63	0.65	2	78	8.3
GOES 8	0.56	0.54	0.58	2	58	5.6
DS-2000	0.57	0.53	0.63	1	19	2.0
DFH-3 Bus	0.5	0.47	0.54	1	15	1.6
Galileo 1	0.12	0.08	0.2	1	12	1.3

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 Figure 4.7. The results for the MMT Balanced dataset are analysed in more detail in Table 4.6, 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, 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

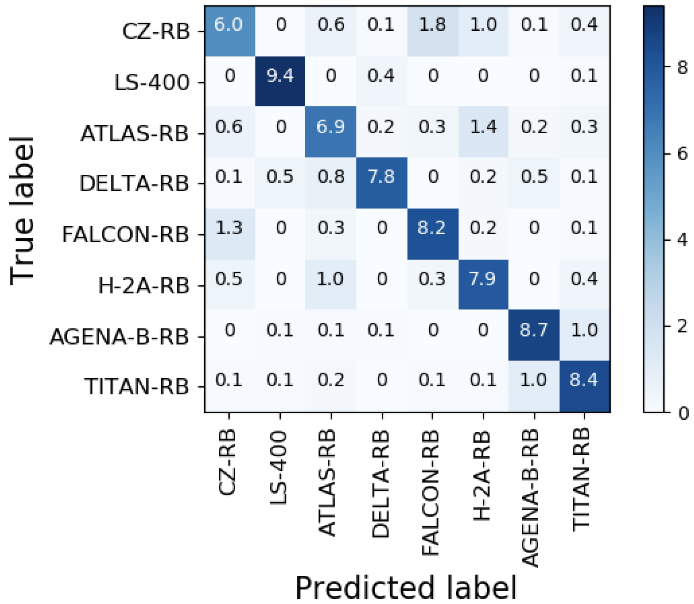


Figure 4.7 – MMT Balanced Normalised Confusion Matrix: 1D-CNN Cross Validation Results

rocket body classes with the most unique RSOs (CZ-3B and ATLAS-RB).

4.3.3 Transfer Learning

Figure 4.8 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 specified 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

Table 4.6 – Class Analysis MMT Balanced: 1D-CNN Cross Validation Results

Class	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

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.

Transfer learning was also applied to the MMT Balanced dataset with Figure 4.9 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.

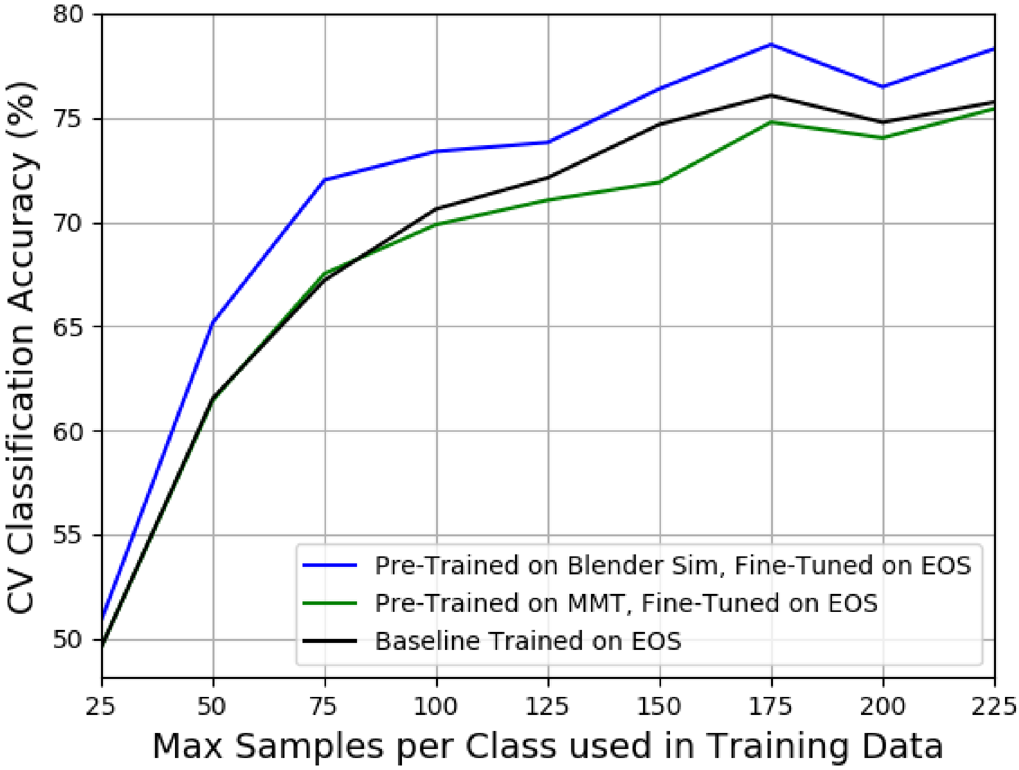


Figure 4.8 – Transfer Learning Results, Evaluation/Target Dataset: EOS Dataset

4.4 Discussion

The results presented in this chapter demonstrate that machine learning models are an effective tool for performing RSO classification from both simulated and real light curves. This chapter 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 [83] 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

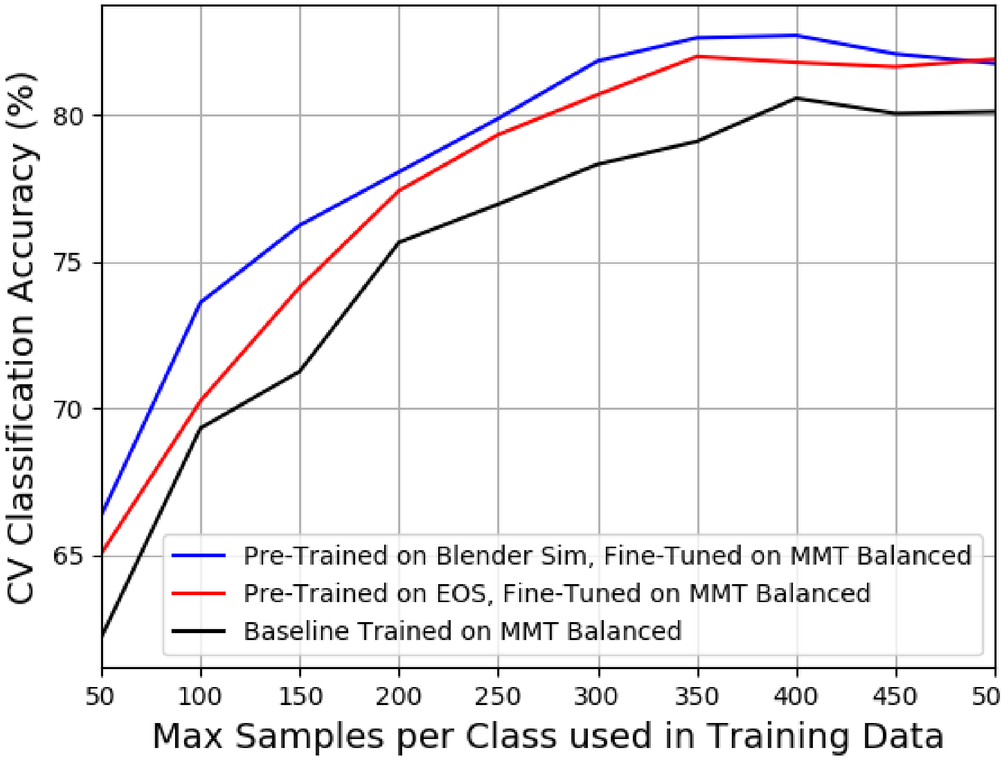


Figure 4.9 – Transfer Learning Results, Evaluation/Target Dataset: MMT Balanced Dataset

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 [83, 79].

The normalised confusion matrix depicted in Figure 4.6a 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 4.6, 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, it would also be interesting to investigate the benefit of simulating specific objects for classes with limited real light curve data using the Blender simulation environment. These simulated light curves would then be incorporated in the real world dataset in an attempt to improve results on the real light curves for the selected object type. The difficulty with this approach is developing an exact model of 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. The results in this chapter demonstrate 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 4.3, which depicts the number of layers that were transferred from the pre-trained network to the fine-tuned 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. This results in increased intra-class variability and inter-class similarity within the simulated dataset compared with previous simulations based on the FFM, which used the same initial epoch, initial quaternion and angular rate for all light curves [79, 83]. Subsequently, the classification task becomes more challenging and is a better representation of the difficulties faced in classifying RSOs from observational light curve data, increasing the effectiveness of the transfer learning.

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. 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.

Finally, it should be noted that this classification method relies upon the rotation of the observed acRSO making it more suitable for debris classification as opposed to active objects. While this method may be used for the classification of active objects in a controlled rotation, it is unclear whether the short duration light curves used in this research could be used to classify an actively stabilised object as the lack of rotation would result in limited observability of the object's shape. An alternative approach may be to attempt classification using long duration light curves (i.e continuous observation over the course of a night) with the apparent magnitude measured as a function of the solar phase angle rather than time.

4.5 Summary

In this chapter a data-driven approach to RSO characterisation has been investigated through the application of machine learning techniques. On all four datasets presented, 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.

Additionally, the transfer learning results presented in this chapter 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%.

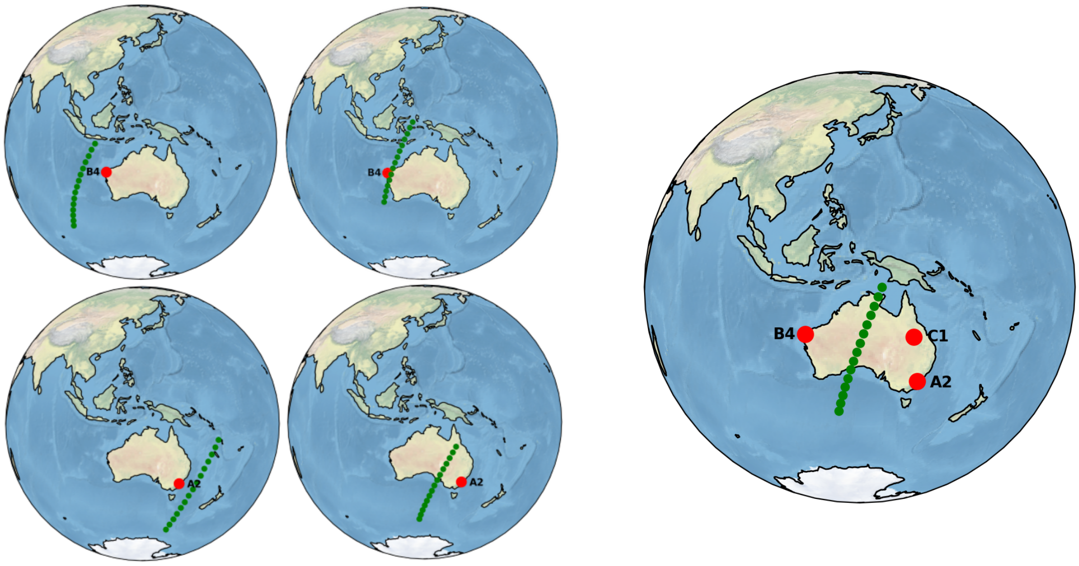
Chapter 5

Space Object Characterisation using Multiple Observations

5.1 Introduction

The previous chapter presented a method for improving RSO shape classification performance on real world light curve datasets through a process of simulation-based transfer learning. Whilst this method demonstrates improvements on previous solutions, the model was still unable to correctly classify a significant portion of the dataset. This is to be expected given the complexity of the real world RSO classification problem and the relative sparsity of information contained in the light curve data extracted from non-resolvable imagery. The problem is made more challenging by a high inter-class similarity between certain classes, as well as intra-class variability for objects undergoing different types of rotation. In particular, specific objects may be incorrectly classified from a single light curve observation if the combination of their rotation and viewing direction relative to the ground station does not result in perceivable variations in the measured light curve signal.

As a result of these ambiguities, it is common for human experts to collect multiple tracks of the same object over an extended period of time. This allows them to



(a) Repeated Tracking Scenario **(b)** Simultaneous Tracking Scenario

Figure 5.1 – Multi-Observation Tracking Scenarios: RSO track depicted in green, active sensor in red (a) Example of repeated tracking of a single RSO, with separate tracks of the RSO observed from a single sensor at a time. In this example four separate tracks are observed of the same RSO resulting in a group of four light curves. (b) Example of a simultaneously observed track where all three sensors (A2, B4 and C1) collect observations on a single RSO at the same time resulting in a group of three light curves.

build knowledge about the object over time, as a variety of phase and viewing angles provide different information for an object, increasing their confidence in the final characterisation of the object. Thus it is expected that it would be advantageous for a model to also be able to combine multiple light curve observations for the same object in a similar fashion.

This chapter investigates the combination of groups of light curves obtained for the same object in order to determine if they can be used to improve classification performance for challenging cases. In particular, two potential scenarios are considered for obtaining groups of light curves:

1. **Repeated tracking of a single object across multiple separate passes in order to generate multiple light curve observations**. Figure 5.1a depicts

an example of repeated tracking of a single RSO, with separate tracks of the RSO observed from a single sensor at a time. In this example four separate tracks are observed of the same RSO resulting in a group of four light curves. Note that these observations can be collected from different sensors but they are not collected simultaneously.

2. **Simultaneous light curve observations of a single object from different ground stations**. Figure 5.1b depicts an example of a simultaneously observed track where all three sensors (A2, B4 and C1) collect observations on a single RSO at the same time resulting in a group of three light curves.

The repeated tracking scenario was selected as multiple tracks of a single object will provide multiple viewpoints as the object rotates, with different starting orientations and viewing angles. For this scenario, it is assumed that the light curves collected across multiple passes have been correlated to the same object through comparison of the angular position with the orbital elements. This scenario was also selected because it is typical of the manner in which rate tracking telescopes operate. Telescope time is a finite resource and when characterisation is not considered, more information can be gained by performing state updates on as many RSOs as possible over the course of a tracking session. This results in short observation passes (typically three minutes or less) in order record sufficient observations to update the state on a semi-regular basis depending on the RSO's visibility. Subsequently, it is beneficial to be able to combine multiple shorter light curves for a specific RSO in order to characterise the object without the requirement for a single light curve of extended length.

The simultaneous observation scenario was selected as observing the same object from multiple sites, enables light curves to be collected from multiple viewing angles, with different sections of the object reflecting sun light towards each sensor. It is expected that with a larger difference in viewing angle there will be more information to be gained through the combination of the simultaneous observations. Previous research has demonstrated that simultaneous light curve observations can be used to obtain additional information about an RSO's characteristics [41] and proposed methods for obtaining the attitude of an RSO with a known shape [101, 70]. Subsequently, while simultaneous observations do require multiple telescopes, it is expected that they might provide an avenue for improving classification performance on RSOs that are difficult to characterise.

In order to identify challenging classification cases, where repeated tracking or simultaneous observations may provide additional information and subsequently improve classification performance, a targeted scheduling process has been developed based on model certainty. Model certainty is calculated through the use of [40], with a threshold approach used to determine RSOs that the model is uncertain about in order to prioritise them for additional collection of light curves. This process is shown to reduce misclassifications, through the combination of multiple light curves for difficult cases, as well as improve sensor tasking efficiency by, reducing tracking requirements for RSOs where the model is determined to have a high confidence in its predicted shape class. This process is seen as an important development for the utilisation of the model framework on real world classification tasks.

The current light curve classification model that was developed in Chapter 4 does not allow for the combination of multiple light curve observations for the same object. Subsequently, in this chapter a multi-observation framework is developed for performing RSO classification from groups of light curve data. The key contributions of this chapter are as follows:

1. The development of a neural network architecture for object classification that is capable of efficiently combining a variable number of light curve observations of a single target object.
2. Demonstration that the resulting model achieves increasing classification performance with the addition of multiple observations of a single target, compared to a single observation-only model.
3. Proposal of a targeted scheduling process, through the incorporation of uncertainty into the framework, in order to enable the determination of more difficult

> cases where collection of additional data is required. This target scheduling process both reduces misclassifications and increases sensor tasking efficiency.

The remainder of the chapter is organised as follows. Section 5.2 presents the development of the various light curve group combination methods that are investigated in this chapter as well as a process for targeted scheduling utilising an uncertainty threshold. Section 5.3 provides an overview of the two different simulated light curve datasets and explains the training and evaluation processes for the different group combination methods. The results of the different light curve combination methods applied to the different tracking scenarios are presented in Section 5.4. Section 5.5 contains a discussion of these results, while Section 5.6 provides a brief summary of the research presented in this chapter.

5.2 Methodology

5.2.1 Problem Definition and Notations

As previously established in Chapter 4, the task of light curve shape classification, for a single light curve is to find the predictive function f that maps in an input light curve $\mathbf{x}$ to a corresponding unique shape class label $y \in \{1, ..., C\}$ such that:

$$f : \mathbf{x} \rightarrow \{1, ..., C\}, \tag{5.1}$$

where C is the number of possible unique shape classes.

When considering multiple light curve observations of a single RSO, the collected light curves are organised into groups $\mathbf{x}_G$, with each group containing a number n of distinct light curves for the same RSO. Thus, when considering light curve groups, the redefined classification task becomes to find the predictive function f that maps a group of input light curves $\mathbf{x}_G$ to a corresponding shape class label $y \in \{1, ..., C\}$ such that:

$$f : \mathbf{x}_G \rightarrow \{1, ..., C\} \mid \mathbf{x}_G = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}\}, \tag{5.2}$$

where $\mathbf{x}^{(i)}$ represents a single light curve.

It should be noted here that the goal is to develop a flexible and generalised model that is capable of processing any number of light curves, without the need to train a separate model.

5.2.2 Model Development and Selection

Initial Model Selection

The first step in the model development process was to implement a model that was capable of performing well on the single light curve classification task, prior to the consideration of combining multiple light curves. Based on the success of the 1D-CNN presented in Chapter 4 and recent results presented in the literature on similar TSC problems [17, 55], 1D convolutions have once again been selected as the basic building blocks of the network.

One of the limitations with the 1D-CNN implemented in Chapter 4 is that the FC section of the network required a uniform length input. Consequently, the light curves needed to be either truncated or zero filled as part of pre-processing to ensure that they were all of uniform length. This was not a significant issue for the Blender simulated dataset, where all light curves were simulated to be a similar length. However, it was noted that the large variations in the length of light curves in the real world EOS and MMT datasets resulted in light curves with significant portions of null values.

In this chapter, a global average pooling (GAP) layer is implemented after the convolutional section of the network, as opposed to the Flatten layer that was used in Chapter 4. The pooling is implemented over the filter dimension, resulting in a single output for each of the filters used during the last convolutional layer. The advantage of using the GAP layer is that its output is always equal to the number of filters used, regardless of the length of the input time series. Thus the time series data is not required to be pre-processed to a specific length through either truncating or padding. As the GAP layer also reduces the number of parameters in comparison

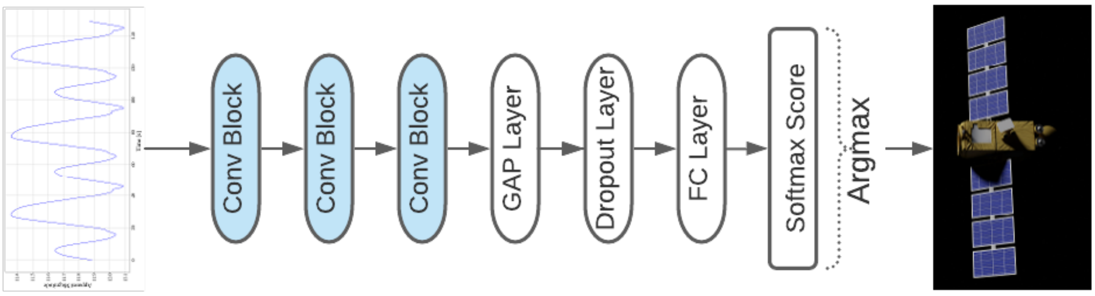


Figure 5.2 – Baseline 1D-CNN Model: Each Conv Block contains a 1D-convolutional layer followed by max pooling and batch normalisation layers.

with the Flatten layer, only a single FC classification layer is implemented after the GAP layer. Softmax activation is applied to the output of this layer in order to determine the model's classification for a given input.

A grid search was performed over the network parameter space in order to determine the optimal network parameters to use. Figure 5.2 depicts the structure of the network that was implemented as the baseline model, while Table 5.1 displays the final network parameters that were selected as well as the various parameters that were considered in the grid search. Each convolutional block consists of a 1D-convolutional layer followed by a max pooling layer and a batch normalisation layer. A dropout layer was used between the GAP layer and the final output layer in order to regularise the model during training.

The developed 1D-CNN model depicted in Figure 5.2 will be used as the baseline model for the multi-observation combination approaches that are presented in the following subsections. The light curve group combination approaches that have been investigated in this research can be split into three main approaches:

1. **Group Max Softmax Score**: This approach takes the overall maximum softmax score prediction from a group of output softmax scores where each score corresponds to an individual light curve in the group.

2. **Softmax Score Combination**: This approach combines the softmax scores for individual light curves through element-wise aggregation in order to obtain

Table 5.1 – Final Network Parameters Selected for the Baseline Model

	Selected Baseline Parameters	Parameters Considered in Grid Search
Convolutional Layers	3	[1, 2, 3, 4, 5]
Convolution Width	10	[5, 10, 15, 20, 40]
Convolution Strides	2	[1, 2, 3]
Max Pooling Size	4	[1, 2, 3, 4, 5]
Max Pooling Strides	1	[1, 2, 3]
Filters	64	[16, 32, 64, 128]
Fully Connected Layers	1	[1, 2, 3]
Dropout	0.3	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]

a single prediction for the group.

3. **Multi-Branch Network Architecture**: This approach uses multiple branches to allow the input of multiple light curves into a single model and subsequently combine them to output a single prediction for the group.

Initial Approach: Group Max Softmax Score

When processing a single light curve input, the final layer of a network trained for multi-class classification, outputs a vector of length C, corresponding to the number of unique classes. As this is a mutually exclusive classification problem, a softmax function is applied to the output vector to convert the values into a normalized probability distribution between 0 and 1. The output is denoted as the softmax score s, with the value for each class indicating the likelihood that an input light curve $\mathbf{x}$ belongs to that class. Thus the class with the maximum likelihood in the softmax score is selected as the output class y predicted by the model.

As established in Equation 5.2, the grouped classification task requires a model to process multiple light curve inputs for an RSO and output a single predicted shape class for the group. An initial approach to the problem is to separate the group of light curves denoted by $\mathbf{x}_G$ and input them individually into a single input/single output model. This will result in an output softmax score $s^{(i)}$ for each individual light curve $\mathbf{x}^{(i)}$. The group max softmax score approach simply selects the overall maximum likelihood from this output group of softmax scores as the output shape class y_G predicted by the model for the group of input light curves $\mathbf{x}_G$. This approach requires no changes to the model architecture, however it also doesn't combine any information from the different light curves in the group. In future comparisons this approach will be referred to as the 'Group Max'.

Softmax Score Combination

Softmax score combination methods apply a similar approach to the Group Max method, in that the group of light curves are again input individually into a trained single input/single output model. However, these methods combine information from the multiple light curves in the group through element-wise aggregation of the group of output softmax scores as outlined in [29]. This aggregation can be performed through either element-wise addition or element-wise multiplication of the output scores to produce a combined group softmax score. The class with the highest value in the combined group score is then selected as the predicted output class y_G for the input light curve group $\mathbf{x}_G$. The functions used for softmax score sum and product are depicted below in Equation 5.3 and Equation 5.4 respectively while the overall approach is depicted in Figure 5.3.

$$y_G = \operatorname{argmax} \sum_{i=1}^{n} s^{(i)} \tag{5.3}$$

$$y_G = \operatorname{argmax} \prod_{i=1}^{n} s^{(i)} \tag{5.4}$$

Similar to the group max method, the softmax score combination does not require any changes to either the architecture or the training of the model in order to evaluate the multiple light curve inputs, as each light curve in the group is input and processed

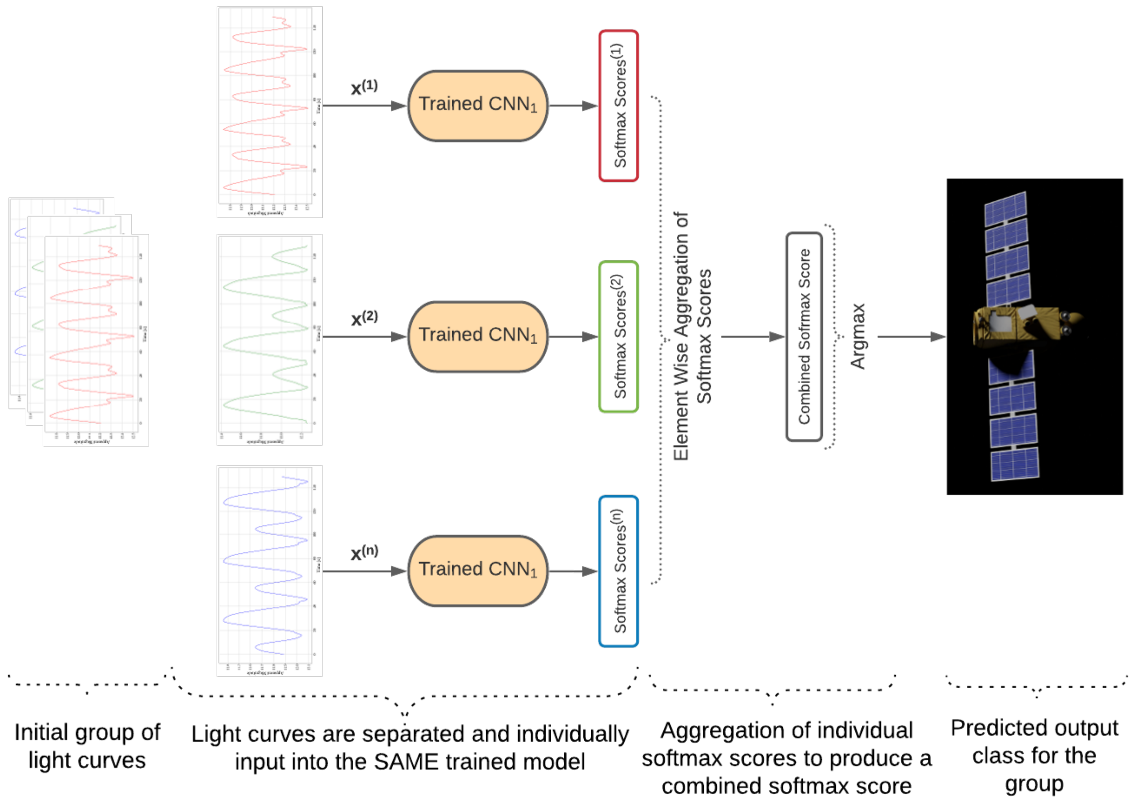


Figure 5.3 – Softmax Score Combination Method: Each light curve in the group in individually evaluated by the same trained model. The output individual softmax scores are then combined through element-wise aggregation to produce a combined softmax score for the group. The maximum value of this combined score is selected as the shape class prediction for the group of light curves.

individually by the trained model. Subsequently, it can be implemented for any model that has been trained on a single light curve dataset. However, the advantage of using this method compared with the group max method is that the information from multiple light curves is combined through the softmax score aggregation process. In future comparisons, the softmax score sum is referred to as 'Scores Sum', while the softmax score product is referred to as 'Scores Product'.

Multi-Branch Architecture

While the softmax score combination approach does combine information from the multiple light curves in the group, the information is combined outside of the model rather than within the model. Previous research has shown that when multiple input data is combined within the model architecture, the model is capable of learning an efficient way to combine the different inputs and thus improve classification performance [127, 7, 109].

Initially, a stacking approach was investigated, with multiple light curves stacked together in a depth-wise manner in order to feed the group of light curves into the network as a single input. However, it was determined that this method would result in models that were unable to process groups with different numbers of light curves and would also require all light curves to be of equal length. Subsequently, this approach was considered too restrictive and instead a multi-branch architecture has been used, where each branch is capable of processing a separate input and producing a feature vector. The feature vectors from the individual branches are then combined within the network, with the combined feature vector being processed by the remainder of the network to output a single classification for the group of inputs.

Using the main components from the baseline model, a multi-input network has been developed that is capable of processing a group of light curves through the use of multiple branches, displayed in Figure 5.4. As with the baseline model, each branch of the network contains three convolutional blocks followed by a GAP layer. The branch takes a single light curve input $\mathbf{x}^{(i)}$ and outputs feature map v:

$$v^{(i)} = NN_{branch}(\mathbf{x}^{(i)}) \in \mathbb{R}^{1 \times n_f} \tag{5.5}$$

where n_f is the number of filters used in the final convolutional block prior to the GAP layer.

The output feature vectors from each branch can then be concatenated together to form a single combined feature vector, v_G, for the group of light curves. In this

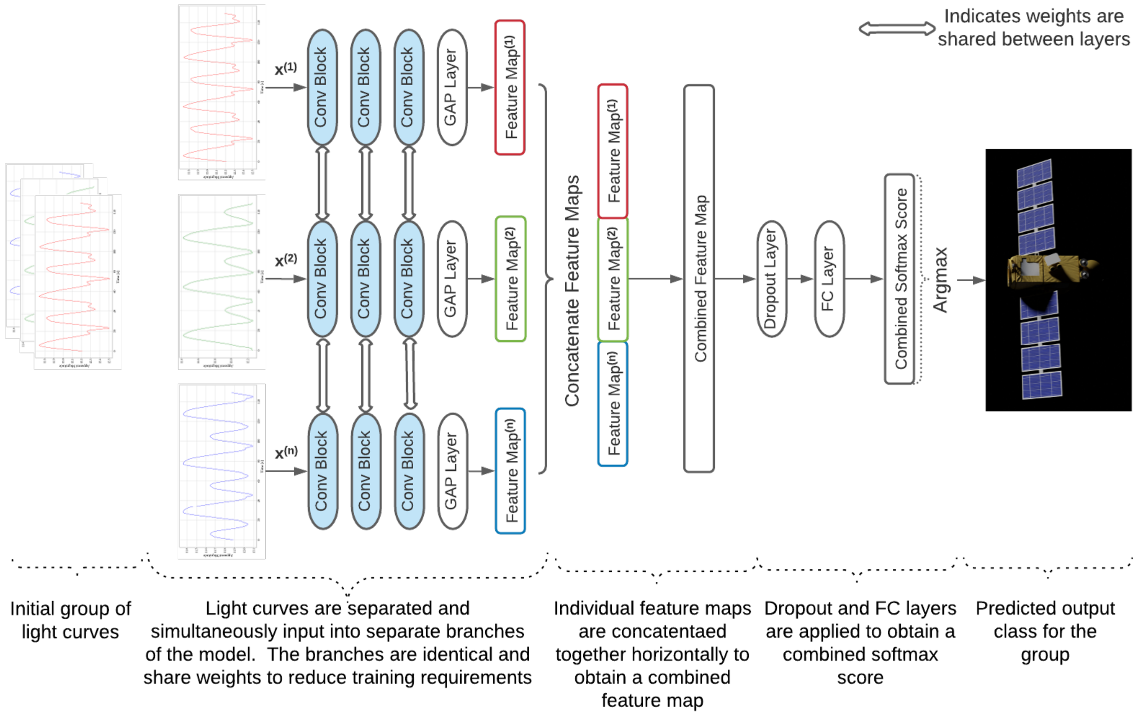


Figure 5.4 – Multi-Branch Network Architecture: Light curves from the same group are input simultaneously into separate convolutional branches of the network. The individual feature maps output from each convolutional branch and are then combined together through horizontal concatenation to produce a combined feature map for the multiple inputs. Model weights are shared between the different input branches to reduce model complexity and training requirements.

work, two different concatenation strategies have been investigated. The first method uses horizontal concatenation of the extracted feature vectors to obtain the combined feature vector as presented in Equation 5.6 and Figure 5.4:

$$v_G = [v^{(1)}, \ldots, v^{(n)}] \in \mathbb{R}^{1 \times n \cdot n_f} \tag{5.6}$$

Dropout is applied to the combined feature map to regularise the network and a final FC layer is applied with softmax activation to determine the predicted output shape class for the group of light curves. It is possible to add additional dropout and FC layers after the combination of the feature maps, however it was found that in practice a single dropout and FC layer provided the best performance for the datasets that were evaluated in this research. In future comparisons, this model is referred to as 'Multi-Branch'.

For multi-input classification tasks where different types of input data are being processed (e.g. audio and imagery) or specified different views are being processed (e.g. flower and leaf of a plant), each branch will be trained to extract features for these specific cases. Thus, a different set of weights must be learnt for each branch and the model is typically only capable of producing an output if the correct number and type of inputs are present. In the case of the grouped light curves classification task, all of the inputs are of the same type (e.g. light curve data) and are from a non-specific viewpoint. Subsequently, the weights that the model learns in one branch for extracting a set of features from one light curve will also be effective at extracting features from another light curve. Therefore, the model branch weights can be shared between the different input branches as depicted in Figure 5.4. As a result, only one set of branch weights is required to be trained for the model, rather than a new set for each branch, which reduces both the complexity and the training requirements of the model.

As a result of the shared weights between the branches, the branch section of the 'Multi-Branch' network is capable of processing groups of light curves that have a variable number of light curves without the requirement for training multiple models. However, as shown in Equation 5.6, when using the horizontal concatenation method for combining the individual feature maps, the length of the combined feature map is dependent on n, the number of light curves in the group. Thus if a Multi-Branch model has been trained to classify groups of light curves containing five individual light curves, it will only be able to evaluate groups of light curves which also contain five light curves. The evaluation of a group of light curves that only contained four light curves, would require the training of a separate model on a dataset containing groups of four light curves. This is a limitation of the Multi-Branch approach with the horizontal concatenation method, particularly for the repeated tracking scenario where it is likely that groups would contain different number of light curves.

To provide a more flexible multi-branch architecture, a second feature map combi-

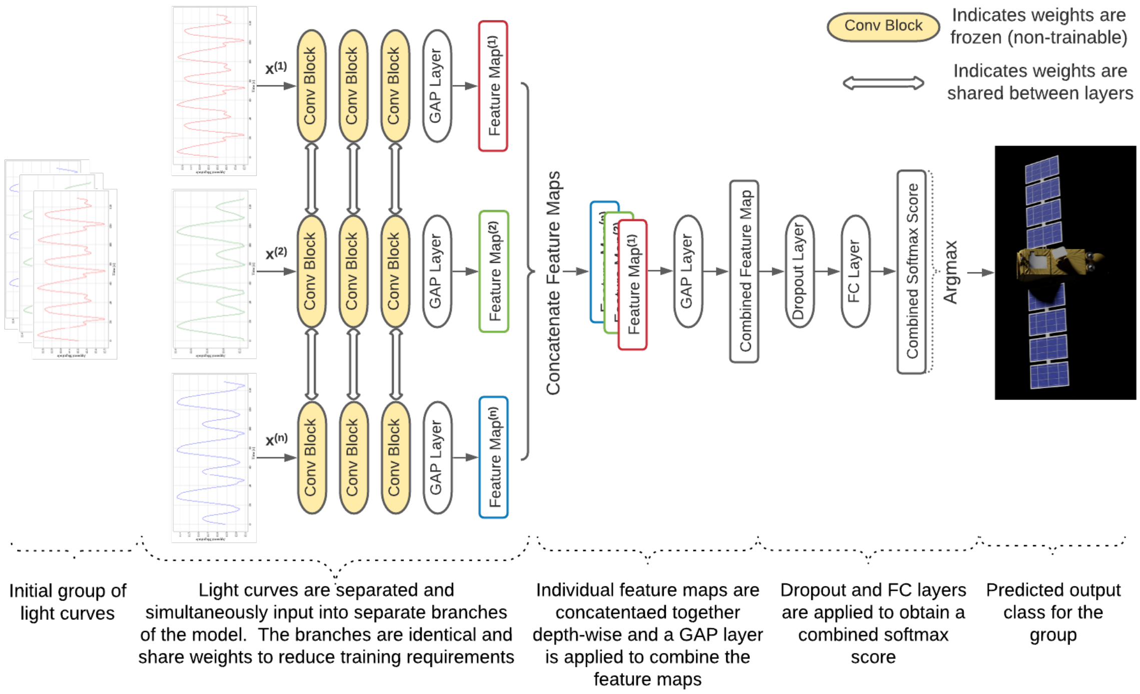


Figure 5.5 – Multi-Branch GAP Network Architecture: Individual feature maps are combined through depth-wise concatenation followed by a GAP Layer averaging across the different feature maps. This results in the length of the combined feature map being invariant to the number of light curves in the group.

nation method has been proposed, which utilises concatenation in the depth-wise direction followed by 1D GAP layer implemented in the same direction. This method averages the extracted feature maps for each of the input light curves resulting in a combined feature map, which has a fixed length, equal to the number of features in the final convolutional layer. Subsequently, the length of the combined feature map v_G is independent of the number of light curves in the group as depicted in Equation 5.7:

$$v_G = \frac{1}{n}\sum_{i=1}^{n} v^{(i)} \in \mathbb{R}^{1\times n_f} \tag{5.7}$$

The network architecture for this approach is displayed in Figure 5.5. Aside from the change in combination method of the feature vectors, this approach, which is referred to as ‘Multi-Branch GAP’ in future comparisons, is exactly the same as the

Multi-Branch approach.

As a result of the independence of the combined feature map to the number of light curves in the group, a single Multi-Branch GAP model is capable of processing light curve groups with any number of light curves. This increase in generalisation comes at a cost of potential loss of information due to the averaging of the extracted feature maps, rather than allowing the remaining network layers to learn the best method to combine them. For the repeated tracking case, as there is no obvious correlation between tracks to make order important, this is unlikely to be an issue. However, for the simultaneous observation case, it is possible that the model could learn to efficiently combine the feature maps from different sensors provided the input order remained consist. Subsequently, results from both the Multi-Branch and the Multi-Branch GAP approaches will be presented in the results section.

A final consideration for both of these network architectures is the training of the model weights. Two different training methods will be introduced here and discussed further in Section 5.3.2. The first method, is to train the entire model (both the branch weights and the combined section) using groups of light curves. The second approach is to load in pre-trained weights, from the baseline model, for the network branches and freeze this section of the network (set the weights to be non-trainable), prior to training the remainder of the model (the combined section) using groups of light curves.

This second approach is possible as the network architecture for an individual branch is the same as the network architecture that was used for the baseline model. As the model weights for the individual branches are shared between branches, only one set of branch weights is required. The weights that would be loaded into the model and frozen are depicted as yellow in Figure 5.5. The idea behind this approach is that the convolutional section of the baseline model will learn an efficient way of extracting features from individual light curves, which is the job required of each branch in the multi-branch networks. When these weights are loaded in from the baseline model and frozen it then allows the remainder of the network (the combined section) to learn to effectively combine the groups of individual feature maps, without the additional requirement of also learning the branch weights.

Both the Multi-Branch and the Multi-Branch GAP models will be evaluated using both training approaches. In future comparisons, Multi-Branch models trained using the second approach are referred to as MB Frozen Branch, while Multi-Branch GAP models trained using the second approach are referred to as MB GAP Frozen Branch.

5.2.3 Use of Predicted Model Certainty for Targeted Scheduling of Challenging RSOs

Model certainty is an important consideration, both for identifying challenging classification cases and for reducing misclassifications during real world operations. Monte Carlo (MC)-dropout [40] is a technique that can be used to quantify model uncertainty. This is done by applying dropout at inference time and then performing multiple stochastic forward passes through the network. This approach results in a number of sample predictions for a single input, each produced from a slightly different model as the dropout randomly switches neurons off. In this work, each group of light curves $\mathbf{x}_G$ is input into the model 100 times with dropout activated in order to obtain 100 softmax score predictions, each from a different model.

The certainty of the model for a particular input can be estimated through determination of the mean prediction and the sum of the per class variance for recorded softmax scores of the Monte Carlo samples. The calculation of the mean prediction from the Monte Carlo samples is depicted in Equation 5.8, while Equation 5.9 displays the determination of the sum of the per class variance. Model certainty for a specific input is high when the output softmax scores sampled using MC-dropout have a high mean and a low variance.

$$y_{avg} = \text{argmax} \, \frac{1}{K} \sum_{k=1}^{J} s_k \tag{5.8}$$

$$\sigma^2 = \sum_{c=1}^{C} \frac{1}{K} \sum_{k=1}^{K} (s_{c,k} - s_{c,avg})^2 \tag{5.9}$$

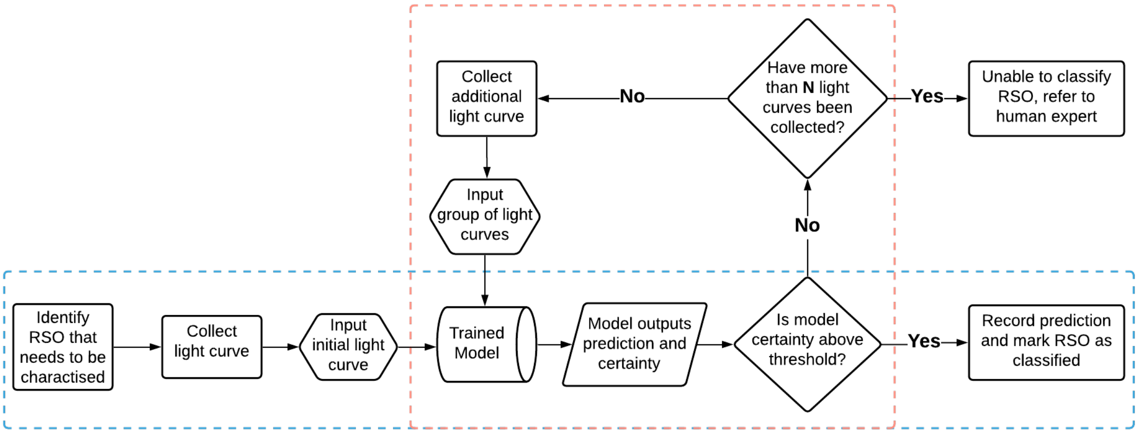


Figure 5.6 – Proposed process for using model certainty to enable targeted scheduling of challenging targets. Process starts on the left hand side of the figure with no knowledge of the RSO's characteristics. Initially the steps in the blue box are followed and if the model certainty about the prediction for a single light curve is above the specified threshold then the object is marked as classified and the process is complete. However, if the model certainty about the prediction is below the threshold, the steps in the red box are repeated until either the RSO is classified or it is deemed too difficult to classify and referred to a human expert

where s_c is the softmax output for a particular class c, C is the number of possible shape classes and K is the number of Monte Carlo samples of the softmax predictions for a specific input $\mathbf{x}$.

Using the model certainty, a threshold based approach is proposed, depicted in Figure 5.6, to determine if an RSO should be marked as classified or if additional light curve data is required. It has been designed in order to identify RSOs that are difficult to classify and would likely result in a misclassification as well as to limit unnecessary tracking of objects that the model can correctly classify. This approach requires the specification of a model certainty threshold in order to determine if a specific RSO should be marked as classified or if additional light curve observations are required. When determining the metric to use as a threshold, it was found that both mean softmax certainty and the sum of the softmax class variance produced very similar results, with no noticeable benefit obtained from a combination of the two metrics. Subsequently, mean softmax certainty was selected to be used as the threshold metric.

This process is based around a typical real world scenario where the task is to classify a specific object, with the aim being to both reduce misclassifications as well as to optimise telescope tasking. The scenario starts on the left hand side of Figure 5.6 with no prior knowledge of the RSO's characteristics and no previously collected light curves for the RSO.

Initially, the steps in the blue box are followed, with the collection of a single light curve for the RSO, which is then input into a trained model that outputs the predicted shape class for the RSO and a certainty score for the prediction. If the model certainty score for the prediction is above a predefined threshold set by the user, then the shape class prediction is recorded and the RSO is marked as classified, ending the process with no further characterisation tasking required.

However, if the model certainty is below the threshold, then the process follows the steps in the red box. This involves re-tasking a sensor to collect an additional light curve for the RSO. This light curve is then combined with the original light curve to form a group of two light curves, which are then input into the trained model to obtain a new predicted shape class and certainty score. The key idea is that the additional information that the model obtains from the combining the information contained within both light curves may improve the certainty of the prediction enabling the RSO to be classified with an acceptable level of confidence.

The steps in the red box are repeated until either the model certainty is above the specified threshold or the process has been repeated a maximum of N times. In this chapter the limit on N is set to be 5 as this is the maximum number of light curves that have been simulated for a single object in the repeated tracking dataset. In practice, this value should be determined based on the additional information gained as more light curves are added to the group. Once the maximum number of light curves has been collected for a specified RSO, if the model certainty is still below the certainty threshold then the RSO should be referred to a human expert for additional follow up.

While the process outlined here is based around the repeated tracking dataset, the same logic could be used to schedule simultaneous observations for challenging classification cases or potentially task alternative sensor types such as laser or radar.

5.3 Experimental Setup

5.3.1 Simulation Environment Setup

In order to evaluate and compare the effectiveness of the proposed models for RSO shape classification using light curve groups, two separate light curve datasets have been simulated using the 3D simulation environment developed in Chapter 3. The first dataset is referred to as the *repeated tracking dataset* and it contains groups of light curve observations of a single object obtained from multiple passes. The second dataset is referred to as the *simultaneous observations dataset* and is comprised of groups of light curve observations collected from multiple sensors on a single RSO target simultaneously. An overview of the specific details of these two datasets is provided in the following sections.

The simulation environment was initially set up to mirror the real world EOS telescope set up, with one site located at Mt Stromlo (Site A) and a second site located at Learmonth (Site B). In order to investigate the potential benefits of obtaining simultaneous multi-site observations from three sites, a third site, labelled as Site C, was added in the simulation environment. The location of Site C, was selected to be in central Queensland with a similar longitude to the Mt Stromlo site and a similar latitude to the Learmonth site. Figure 5.8 depicts the three site locations on a map of Australia. This three site sensor configuration was used to generate both simulated datasets.

Repeated Tracking Dataset Generation

As outlined in Section 5.1, the information contained within the light curve for a rotating object depends on a number of factors including the orientation of the object and the viewing plane relative to the object's axis of rotation. Thus, some objects may be difficult to classify based on a single light curve. Additionally, due to the regular operation mode of rate tracking telescopes, it is beneficial to be able to characterise objects from shorter light curves. Based on these considerations, a simulated dataset was generated to investigate the utility of combining groups of light curves from repeated tracking of the same RSO. It is important to note that these are not simultaneous observations and ideally the tracks should be spaced out over the course of a night or multiple nights. Thus if the simulated object was visible from multiple sites at once, one site was randomly selected to collect the observations.

As for the previously simulated dataset outlined in Section 4.2.3, 13 different object models were simulated, with the initial orientation of the object model generated randomly. A range of different TLEs were used during the initialisation step to simulate different orbit types as well as enabling different phase angle, range and viewing conditions. The object was simulated to have a stable rotation period around a single primary axis with the spin axis randomly selected from one of the three primary body axis of the object. In Section 4.2.3, the rotation period was randomly selected from a list of five possible options. To increase the reality of the simulation, as well as the difficulty of the shape classification task, for this dataset the spin period was randomly sampled from a truncated normal distribution using a mean spin period of 50 seconds and a variance of 100 seconds. The lower and upper bounds were set to be 20 seconds and 500 seconds respectively.

Once the simulation environment had been initialised, observations were collected at one second intervals for a duration of 150 seconds from a single telescope to collect the light curve for a track. After this the simulator was propagated forward for one hour before the next track could be collected. This was to ensure that if a GEO object was continuously visible for an extended period of time that light curves were not collected continuously. Following the one hour time period, another light curve was collected when the object next became visible for any of the active sensors. This process is repeated until five separate light curves each with a 150 observations have been collected for the object. Subsequently, over 6,000 groups of light curves were simulated for the 13 different object models with light curve group containing five

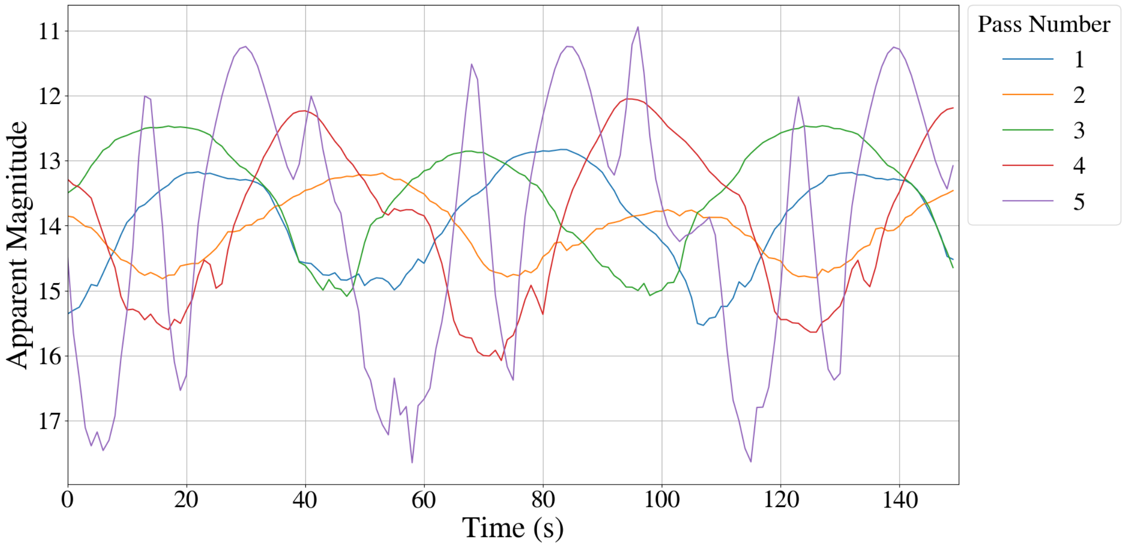


Figure 5.7 – Repeated observations light curve example: Jason 1 satellite model. This figure depicts an example of a group of light curves for a single object from the repeated tracking dataset. Five separate passes are observed of the same simulated object resulting in five light curves. The passes are have a minimum temporal separation of one hour. Variation in viewing angles relative to the model orientation and rotation for separate passes result in light curves with different features observable.

light curves collected for the same object over the course of five separate passes.

Additionally, in order to investigate the affect that the number of observations (light curve length) had on the combination methods, the repeated tracking dataset was sub-sampled to create two additional datasets, with max light curve lengths of 50 and 100 observations respectively. The sub-sampling was performed by using the observations recorded in the first 50 and 100 seconds of the track, respectively, rather than randomly selecting a section of the track. This was to ensure that all combination methods were trained and evaluated on the same data.

To enable the reader to gain a better understanding of the repeated tracking dataset, Figure 5.7 depicts an example of a light curve group, that has been generated for the Jason 1 satellite model. While a rotational pattern is visible in all of the extracted light curves, the light curves extracted from passes 1, 2 and 3 appear to show a relatively smooth oscillating pattern with limited defining features visible. In pass 4, a clearer pattern emerges with several defining features present, while in pass 5 there are a number of strong, clearly defined features present in the light curve. This example demonstrates the difference in information contained within a light curve depending on the viewing angle of the observer relative to the rotational axis of the object and initial orientation of the object at the start of the observed pass. Subsequently, the combination of a group of light curves obtained from repeated tracking of an object, may enable improved classification performance.

Simultaneous Observation Dataset Generation

The second simulated dataset has been generated to investigate if simultaneous observations from multiple sites can be combined to improve shape classification performance. The advantage of using simultaneous observations is that although the orientation of the object is the same, it is being observed from multiple viewing angles. As a result, different sections of the object are reflecting light towards the different observing sensors.

It is expected that more information will be gained from simultaneous observations when their is a larger difference in viewing angle between the sites. The difference in viewing angle between two sites simultaneously observing the same object can be calculated by obtaining the position vectors from the sites to the object and then using the cosine rule to calculate the angle as provided in Equation 5.10.

$$\theta_{\mathbf{ab}} = \arccos\left(\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\right) \tag{5.10}$$

where $\theta_{\mathbf{ab}}$ is the difference in viewing angle between the two sites, $\mathbf{a}$ is the position vector from the first site to the object and $\mathbf{b}$ is the position vector from the second site to the object.

The simulator setup for the generation of this dataset was very similar to the set up used for the repeated tracking dataset, with the same senors and object models used. The main difference was that it was stipulated that observations were only recorded

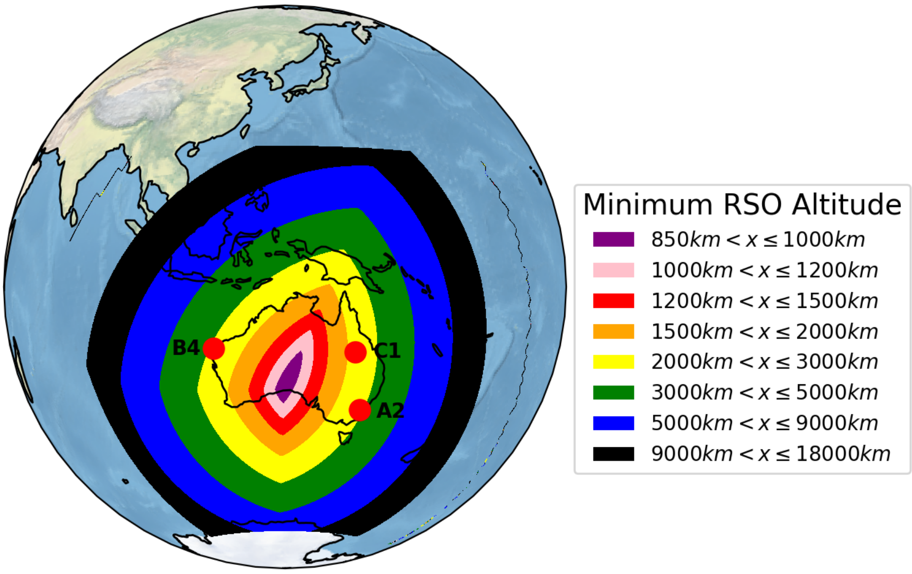


Figure 5.8 – Minimum RSO Altitude Contour Plot for Three Sites Simultaneous Observations. There is a significantly larger baseline between the sensor combinations of A2/B4 and C1/B4, compared with the sensor combination of A2/C1, which are located much closer together.

when the object was visible from all three sites simultaneously. This condition enabled consistency within the dataset as all examples contained light curves collected from all three sites. However, it also restricted the difference in viewing angle between the sites as the visibility conditions had to be met at all three sites.

Figure 5.8 depicts the location of the three sites as well as the minimum RSO altitude that is observable simultaneously from all three sites. This was calculated using the equation for minimum observable object altitude from a single sensor displayed in Equation 5.11 and then taking the maximum values from the three sensors.

$$h_{min} = R_e \left[\frac{\cos(el_{min})}{\cos(el_{min} + \frac{d}{R_e})} - 1 \right] \tag{5.11}$$

where h_{min} is the minimum observable altitude, R_e is the radius of the Earth (assuming spherical Earth), el_{min} is the minimum RSO elevation angle that a sensor is capable of observing at (set to 15° in this work) and d is distance from the observing sensor.

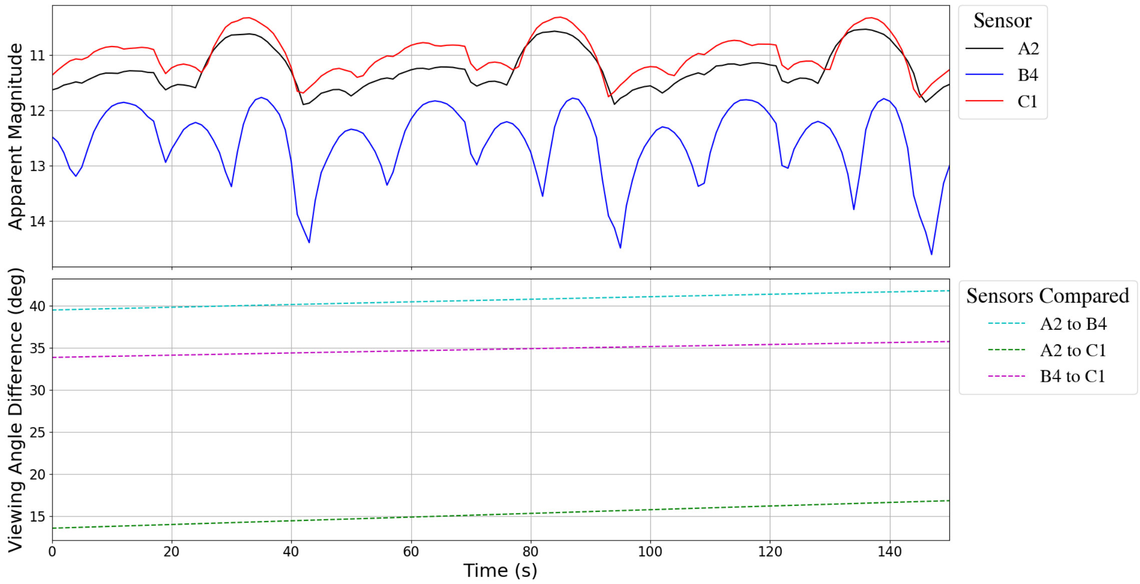


Figure 5.9 – Example of simultaneous light curves collected from three sensors (A2, B4 and C1): ICECube (3U CubeSat) satellite model. The TLE used to simulate this pass was for NORAD ID: 02403 (Low-MEO). The light curves (upper section) collected simultaneously from A2 and C1 are very similar while the light curve collected from B4 depicts a different pattern. When comparing the viewing angle difference (lower section) between the various sensors, A2 and C1 have a similar viewing angle (less than 20 degrees difference) while the viewing angle difference between B4 and the other two sensors is much larger.

Based on the minimum observable altitude, a number of TLEs in the upper-LEO and low-MEO orbital ranges were selected to initialise the simulation. These were selected in order to generate a variety of passes that were visible to all three sensors simultaneously, whilst also having a large viewing angle between different sites. Subsequently, a dataset containing approximately 4500 groups of three simultaneously observed light curves from three different sensors were generated. Similar to the repeated tracking dataset, the light curves were sampled at 1 second intervals for 150 seconds and 13 different satellite models were used in the simulation.

An example of a group of light curves from the simultaneous observation dataset collected using the three different sensors is presented in the upper section of Figure 5.9, for ICECube2016 (3U Cubesat). The bottom section of the figure depicts the difference in viewing angle between the various combinations of sensor pairs. As expected, there is a larger difference between the sensor pairs of A2/B4 as well as B4/C1, while there is a less of a viewing angle difference A2/C1 as these sensors are located significantly closer together in angular terms. When comparing the light curves in the upper section of the figure, the light curves for A2 and C1 are very similar, with only a slight shift in the magnitude and small time bias observed. In contrast, the light curve for B4 is quite different as a result of the large difference in viewing angle, and clearly depicts four separate peaks per rotation while only one clear peak is observable in the light curves for A2 and C1.

5.3.2 Model Training and Evaluation Setup for Different Combination Methods

As outlined in Section 5.3.1, both datasets contain groups of light curves for a single object. In order to fairly evaluate the different combinations methods, it was determined that the models should be trained on the same number of light curves where possible, as neural networks typically perform better when trained on larger datasets. This presents an issue as some of the methods use a single input/single output model, while the multi-branch approaches are capable of processing multiple light curves. Thus, if a single input/single output model is only trained on one light curve per group, it will be trained on a fifth of the light curve data compared with a multi-input light curve model that is trained on all five light curves per group.

Table 5.2 provides an overview of the different combination methods that are being compared and the different training methods that were used. The same training approaches were used for both the repeated tracking dataset and the simultaneous observation dataset.

Group Max Score, Softmax Scores Sum and Softmax Scores Product

These three approaches all involve the combination of multiple outputs from a single input/single output model, which is trained using a dataset of single light curves

Table 5.2 – Overview of the Training Methods used to compare the Different Combination Methods

Combination Method	Initial Training	Retraining Final Layers	Compatible with Variable Number of LCs in a Group
Group Max	Single LCs	-	Yes
Scores Sum	Single LCs	-	Yes
Scores Product	Single LCs	-	Yes
Multi-Branch	Groups of LCs	-	No
Multi-Branch GAP	Groups of LCs	-	Yes
MB Frozen Branch	Single LCs	Groups of LCs	No
MB GAP Frozen Branch	Single LCs	Groups of LCs	Yes

*LC is ‘Light Curve’ abbreviated

rather than groups of light curves. Thus the selected dataset (repeated tracking or simultaneous observations) containing groups of light curve examples is separated into single light curves examples with the order randomised so that the groups no longer exist. To further clarify this concept, taking the repeated tracking dataset as an example, the 6,000 groups of five light curves are separated into 30,000 single light curve examples. The baseline model is then trained on this single light curve dataset with the weights of the trained model saved. During training, there is no attempt made to combine multiple light curves.

During evaluation, the light curves are evaluated as groups, with a group split into individual light curves, which are then input into the trained model before the outputs are combined using the selected method to obtain an overall shape classification prediction for the group. As any number of light curves can be combined using this approach there is no need to train separate models when evaluating performance on groups with different numbers of light curves.

Multi-Branch

In contrast, the Multi-Branch (see Figure 5.4) approach is trained on groups of light curves. As discussed in Section 5.2.2 the Multi-Branch case, which uses horizontal concatenation to combine the output from the multiple branches, is only capable of processing a fixed number of inputs. Subsequently, in order to evaluate the effect that combining five light curves instead of four has on the classification accuracy, a separate model must be trained for both cases.

Therefore, during the training process, five separate Multi-Branch models are trained, each on a different number of light curves. While the model that is trained on the groups of five light curves has access to the full dataset during training, the model that is trained on the groups of one light curve only has access to a fifth of the data during training. During evaluation, each of the models are evaluated separately on the light curve groups that contain the same number of light curves as the training data.

Multi-Branch GAP

Unlike the Multi-Branch approach, models utilising the Multi-Branch GAP architecture (see Figure 5.5) are capable of processing groups containing different numbers of light curves. Thus, like the score combination methods, only a single model is required to be trained for the Multi-Branch GAP architecture. The single model is trained on groups of light curves containing the maximum number of light curves in the dataset (five for the repeated tracking dataset and three for the simultaneous observation dataset). This enables a fair comparison with the score combination methods as the number of individual light curves that each model is trained on is the same.

During evaluation, the trained model is evaluated on groups of light curves, with the same model used to evaluate groups with different numbers of light curves.

MB Frozen Branch and MB GAP Frozen Branch

As outlined in Section 5.2.2 MB Frozen Branch and MB GAP Frozen Branch have the exact same network architecture as their counterparts (Multi-Branch and Multi-Branch GAP). However, there is a difference in the way that the models are trained, with models containing 'Frozen' in the title using pre-trained weights for the convolutional layers in the network branches. These pre-trained weights are loaded from the convolutional section of the same baseline model that is used to evaluate the score combination methods. It is important to note that the saved weights are only loaded for the branch section of the models and that these weights are frozen so they are not changed during any subsequent training of the model. The remaining weights in the fully connected section of the models are randomly initialised and then trained and evaluated using the same approaches taken to train the Multi-Branch and Multi-Branch GAP models respectively.

5.3.3 Model Training Parameters

The following parameters were used in order to train the different models required for comparing the combination methods outlined in Section 5.3.2. During training, K-fold cross validation was applied, as described in Section 4.2.5, with the value of K set to 5, in order to evaluate the trained models while utilising all the available data. As the classes are approximately balanced through the randomised simulation process, there is no requirement for incorporating the class weights into the loss function and as such the standard categorical cross entropy loss function was used during model training.

During training, Adam optimisation [67] was used with a learning rate initialised at 0.001 and an exponential decay schedule implemented after 10 epochs. Models were trained for a maximum of 500 epochs with early stopping implemented if the validation loss had not improved by more than 0.0005 in the past 20 epochs to help prevent overfitting.

5.3.4 State-of-the-Art Comparison (InceptionTime)

There has recently been significant advancement in the state-of-the-art models that have been developed for TSC tasks. The 'InceptionTime' model, developed by Fawaz

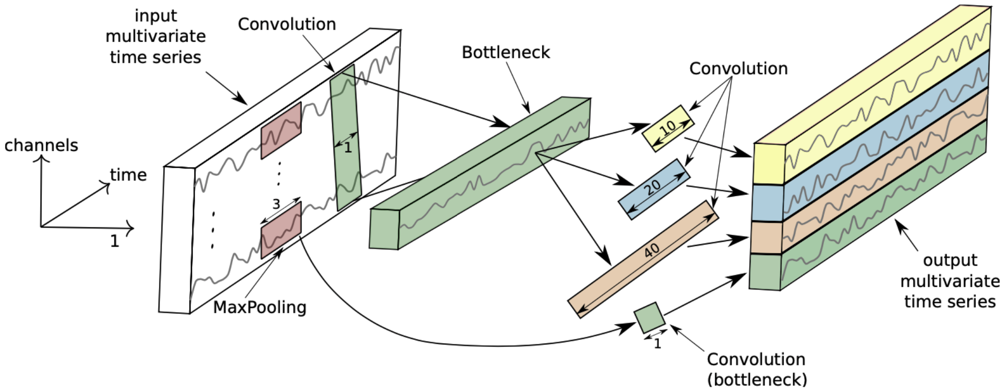


Figure 5.10 – Inception Module for TSC as presented in [55]. The Inception Module uses a bottleneck layer to reduce dimensionality of multivariate time series data before applying three different length 1D-convolutions using 32 filters each. The results of these convolutions are concatenated with a max-pooling operation on the original data followed by a pointwise convolution again with 32 filters. 'SAME' padding is used in all operations so that the length of the features output from the Inception Module is the same as the length of the input time series.

et al. [55], has been selected in order to compare the developed baseline model with a state-of-the-art model. InceptionTime was specifically designed for TSC and the authors demonstrated that it achieved state-of-the-art performance when evaluated on a number of benchmark TSC classification tasks. In this research, the InceptionTime model has been implemented using its default parameters as outlined by the authors of the paper. While this model is also based around 1D-convolutions, there are 2 key differences to baseline model that was outlined in Section 5.2.2.

Firstly, InceptionTime uses 'Inception Modules' as the primary building blocks for the network, which combines multiple 1D-convolutions with different convolution lengths. The diagram of the Inception Module presented in [55] is included as Figure 5.10, with a brief description of the key components, see [55] for a more detailed explanation. The InceptionTime model consists of a total of six Inception Modules, grouped into two residual blocks, each containing three modules. It is also important to note that both a stride of 1 and 'SAME' padding is used within these modules for both convolutional and pooling operations. This means that the output feature vector remains the same size as input length throughout the convolutional section of the network.

Secondly, the InceptionTime network has shortcut linear connections, which connect the two residual blocks. This process is designed to minimise the vanishing gradient problem by allowing a direct flow of gradient [46] and has been shown to improve the convergence rate of the network.

As the Group Max, Scores Sum and Scores Product combination methods all do not require any changes to the model architecture, the InceptionTime model will be compared with the baseline 1D-CNN for these three approaches. To ensure a fair comparison, the InceptionTime model will be trained on the same dataset as the Baseline 1D-CNN.

5.4 Results

5.4.1 Classification Performance on Simulated Repeated Tracking Dataset

The different light curve group combination methods outlined in Section 5.2.2 were evaluated on the repeated tracking dataset as well as the two sub-sampled datasets with shorter max light curve lengths. Table 5.3 depicts the change in cross validation accuracy for each of these combination methods as progressively more light curves per group are used during evaluation. When only one light curve is used per group, the trained models are evaluated on the first light curve in the group. When two light curves per group are used in evaluation, the trained models are evaluated on the first two light curves in the group. This evaluation process was continued up to the maximum of number of light curves in each group, which was set to be five light curves for the repeated tracking dataset.

Based on the results presented in Table 5.3, it can be seen that all of the combination methods result in improved classification accuracy as more light curves are added.

Table 5.3 – Comparison of Different Group Combination Methods for the Repeated Tracking Dataset: Mean 5-Fold Cross Validation Accuracy and Standard Deviation (each fold using 80% training and 20% holdout)

Max LC Length	Group Combination Method	Classification Accuracy and Standard Deviation (%) Light Curves Per Group used in Evaluation				
		1	2	3	4	5
50						
	Group Max	71.38 ± 0.35	74.81 ± 0.59	75.87 ± 0.60	76.21 ± 0.69	76.34 ± 0.80
	Scores Sum	71.38 ± 0.35	75.85 ± 0.53	77.36 ± 0.38	78.31 ± 0.57	79.17 ± 0.62
	Scores Product	71.38 ± 0.35	76.95 ± 0.67	78.78 ± 0.58	79.95 ± 0.64	81.07 ± 0.63
	Multi-Branch	65.78 ± 1.00	71.62 ± 0.06	72.77 ± 0.15	74.53 ± 0.89	74.99 ± 0.16
	Multi-Branch GAP	67.60 ± 0.17	73.32 ± 0.35	76.21 ± 0.38	77.76 ± 0.31	78.96 ± 0.37
	MB Frozen Branch	**71.88 ± 0.37**	**77.77 ± 0.56**	79.55 ± 0.40	**80.88 ± 0.31**	**82.21 ± 0.27**
	MB GAP Frozen Branch	71.38 ± 0.20	77.45 ± 0.39	**79.61 ± 0.23**	80.82 ± 0.34	82.01 ± 0.30
100						
	Group Max	77.15 ± 0.22	80.32 ± 0.33	81.33 ± 0.30	81.75 ± 0.29	81.85 ± 0.25
	Scores Sum	77.15 ± 0.22	80.95 ± 0.31	82.12 ± 0.20	83.03 ± 0.32	83.63 ± 0.41
	Scores Product	77.15 ± 0.22	81.67 ± 0.39	83.32 ± 0.25	84.41 ± 0.31	85.19 ± 0.37
	Multi-Branch	72.55 ± 0.40	76.80 ± 0.36	78.57 ± 0.10	79.76 ± 0.31	80.73 ± 0.57
	Multi-Branch GAP	72.96 ± 0.16	77.72 ± 0.36	80.20 ± 0.43	81.47 ± 0.39	82.31 ± 0.31
	MB Frozen Branch	**77.41 ± 0.33**	**82.19 ± 0.44**	**83.79 ± 0.18**	**85.04 ± 0.13**	**85.69 ± 0.15**
	MB GAP Frozen Branch	77.18 ± 0.10	81.74 ± 0.35	83.64 ± 0.18	84.82 ± 0.18	85.48 ± 0.33
150						
	Group Max	78.61 ± 0.46	81.45 ± 0.27	82.43 ± 0.23	82.96 ± 0.17	83.02 ± 0.15
	Scores Sum	78.61 ± 0.46	81.92 ± 0.28	82.98 ± 0.21	83.84 ± 0.26	84.25 ± 0.32
	Scores Product	78.61 ± 0.46	82.60 ± 0.32	84.17 ± 0.22	85.19 ± 0.26	85.76 ± 0.27
	Multi-Branch	73.14 ± 0.19	76.91 ± 0.04	78.59 ± 0.14	79.29 ± 0.07	79.77 ± 0.29
	Multi-Branch GAP	73.80 ± 0.19	78.57 ± 0.43	80.55 ± 0.22	81.78 ± 0.26	82.86 ± 0.20
	MB Frozen Branch	**78.78 ± 0.28**	**82.90 ± 0.27**	**84.72 ± 0.31**	**85.68 ± 0.08**	**86.24 ± 0.24**
	MB GAP Frozen Branch	78.32 ± 0.31	82.79 ± 0.26	84.42 ± 0.20	85.61 ± 0.13	86.04 ± 0.27

There is also a trend where longer light curves have a higher classification accuracy. The baseline model performance for a single light curve is 71.38% for light curves with a max length of 50 observations compared with 78.61% for a single light curve with a max length 150 observations. Key results from Table 5.3 are presented in Figure 5.11 and Figure 5.12

Figure 5.11 displays the cross validation accuracy results for the 150 observations case

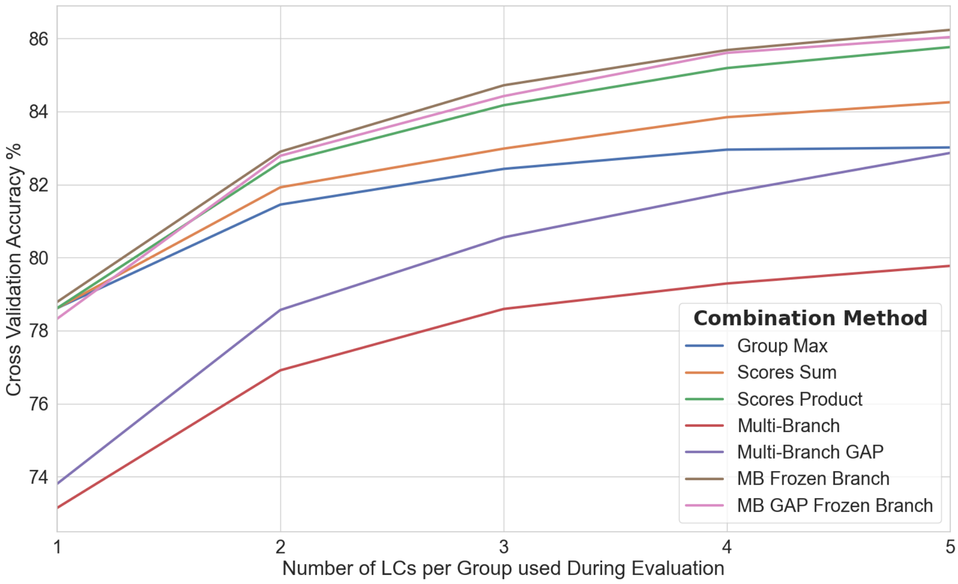


Figure 5.11 – Comparison between the Different Group Combination Methods for the 150 Observations Case (no sub-sampling). All combination methods result in improvement in classification accuracy as additional light curves are used during evaluation. The MB Frozen Branch approach achieves the highest performance, although very similar performance is achieved by the MB GAP Frozen Branch and Scores Product approaches. The Multi-Branch and Multi-Branch GAP approaches were significantly outperformed by other methods.

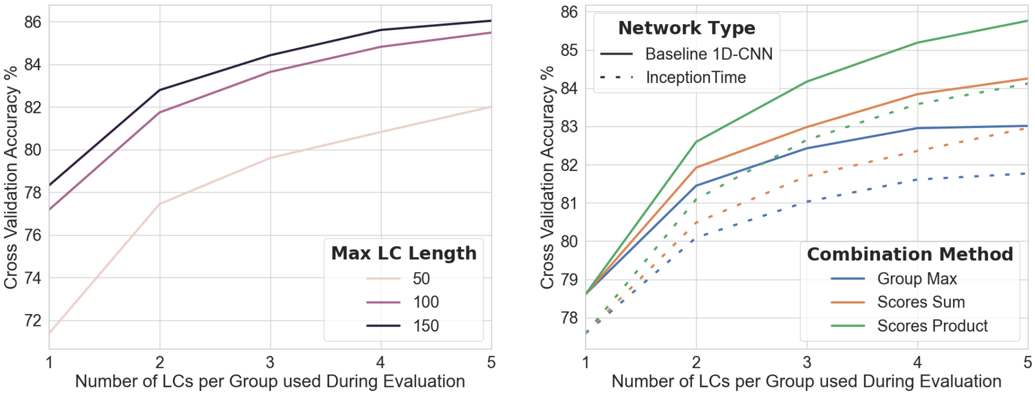
(no sub-sampling) plotted against the number of tracks in the group that were used during evaluation. There is a visible trend of increasing classification accuracy with increasing number of light curves used during evaluation. There does appear to be diminishing returns, in terms of the improvement in classification performance as the number of light curves used during evaluation increases, with the gradient of the lines decreasing. However, for the majority of classification methods it does not appear that a plateau has been reached and as such it is expected that further increases in performance could be achieved by adding additional light curves.

In terms of the combination methods, the MB Frozen Branch achieved the highest classification accuracy of 86.24% when five light curves were used during evaluation, compared with a classification of 78.78% when a single light curve was used during evaluation. This results in an improvement in classification performance of 7.46%.

Very similar performance was observed for the MB GAP Frozen Branch and the Scores Product methods. The Scores Product method outperformed both the Scores Sum and Scores Max methods with the difference in performance increasing as more light curves were used per group. For the Scores Max case, there is very little improvement in performance from the four light curve to the five light curve case and it appears that it has hit a plateau in terms of increases in classification accuracy.

It was noticeable that the Multi-Branch and Multi-Branch GAP models that were solely trained on the groups of light curves were significantly outperformed by the other methods, with the Multi-Branch case in particular performing poorly. This appears to predominately be an issue with the model weights that have been learned for the branch section of the network. This is highlighted by the fact that for the single light curve case, where there is no combination of light curves so all methods essentially revert to the baseline model architecture, the classification accuracy for the Multi-Branch and Multi-Branch GAP models is approximately 5% lower than the other methods. This is further emphasised by the significant improvement in performance of MB Frozen Branch and the MB GAP Frozen Branch models relative to the Multi-Branch and Multi-Branch GAP models.

Figure 5.12a depicts the results for MB GAP Frozen Branch combination method applied to the three different max light curve lengths presented in Table 5.3. As the max length of the light curve increases, the cross validation accuracy is observed to increase. It should be noted that there appears to be a strong correspondence between the amount of observations present in a group and the classification accuracy. This can be seen by comparing the cross validation accuracy for the 50 observations case with two light curves in a group (maximum of 100 observations available for the group), which is 77.45%, with the 100 observations case with only one light curve available, which is 77.18%. This relationship can also be observed through comparison of the 50 observations case with three light curves (max 150 observations) and the 150 observations case with one light curve as well as the 100 case with three light curves (max 300 observations) and the 150 observations case with two light curves (max 300 observations). However, there is a discrepancy in this pattern with the 50



(a) Light Curve Length Analysis **(b)** Comparison with InceptionTime

Figure 5.12 – **(a)** Effect of Light Curve Length on Classification Accuracy for MB GAP Frozen Branch Method. Cross validation accuracy is observed to increase with the max light curve length and even after combination there appears to be a correlation between the accuracy and the number of observations used. **(b)** Comparison between Baseline 1D-CNN model and InceptionTime model. Similar trends are observable for the different combination methods applied to both models with Scores Product found to be the most effective. The Baseline 1D-CNN was found to slightly outperform the InceptionTime model with the difference in performance increasing as more light curves were used during evaluation.

observations case with four light curves (max 200 observations) being outperformed by 100 observations case with two light curves max (200 observations).

It is also interesting to note in Figure 5.12a, that the difference between the 150 observations case and 50 observations case decreases as more light curves are added during evaluation. This may be because the 150 observations case is approaching the upper limit of the classification accuracy for this dataset, while improvements in performance can still be achieved in the 50 observations case.

Figure 5.12b provides a comparison between the Baseline 1D-CNN and the state-of-the-art InceptionTime model for the Group Max and softmax score combination methods. While the same trends are visible for both models, for all three combination methods, the Baseline 1D-CNN outperforms the InceptionTime model. This difference in performance also appears to increase as more light curves are used during evaluation. This can be seen through the comparison of the difference in classification

Table 5.4 – Repeated tracking dataset class analysis results for Multi-Branch GAP model with Pre-Trained Branch weights when provided with a single light curve (1 LC) compared with a group of 5 light curves (5 LC), max light curve length 150

Class	F1 Score		Precision		Recall	
	1 LC	5 LC	1 LC	5 LC	1 LC	5 LC
Cylinder	0.61	0.70	0.58	0.66	0.65	0.74
SpaceX	0.62	0.71	0.64	0.74	0.61	0.69
ASTRE	0.76	0.88	0.82	0.94	0.71	0.83
Sphere	0.96	0.98	1.00	1.00	0.92	0.96
1RU-Cubesat	0.81	0.90	0.87	0.93	0.76	0.87
2RU-Cubesat	0.64	0.80	0.64	0.81	0.65	0.79
ICECube2016	0.79	0.88	0.75	0.84	0.83	0.92
PlaneSingle	0.88	0.91	0.85	0.88	0.91	0.94
Plane	0.86	0.90	0.86	0.92	0.86	0.89
Topex	0.95	0.99	0.96	1.00	0.94	0.98
GOES-PQR	0.66	0.81	0.65	0.80	0.67	0.83
Jason1	0.78	0.90	0.75	0.88	0.82	0.91
SHO	0.75	0.87	0.72	0.84	0.78	0.90

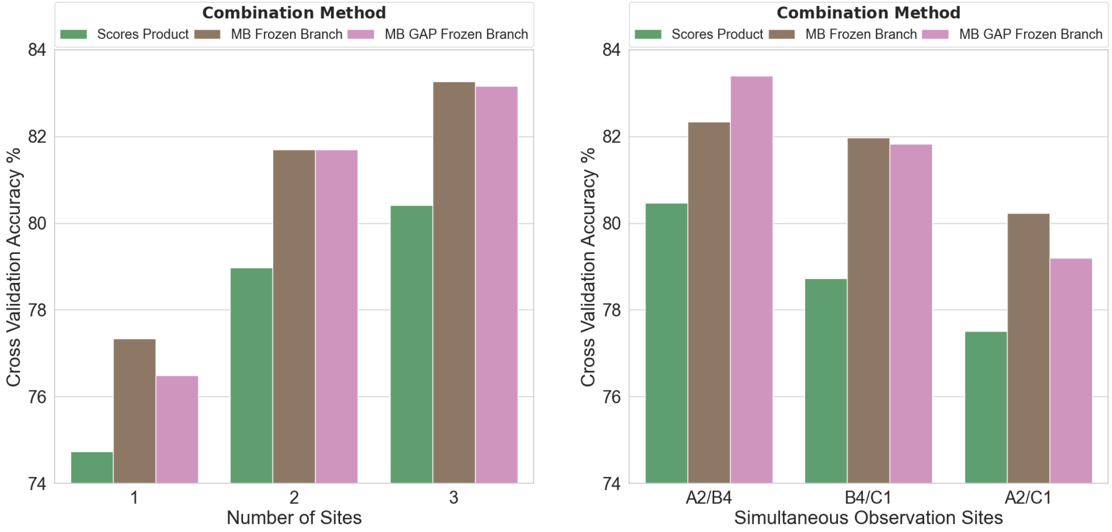
accuracy between the two models for the one light curve case, which is approximately 1%, increasing to a difference of 1.6% when all five light curves are used during evaluation with Scores Product combination method.

Table 5.4 presents a class analysis with F1 Score, precision and recall to evaluate the difference in performance when using one light curve during evaluation, compared with using a group of five light curves combined using the MB GAP Frozen Weights method. The MB GAP Frozen Weights method was selected as it was the best performing model that only required one model to evaluate groups containing different numbers of light curves. The results from the five light curve case show significant improvement for all classes across all three metrics compared with the one light curve case. For the one light curve case there are initially four classes with an F1 score below 0.7 that appear more challenging than other classes to classify; Cylinder, SpaceX, 2RU-Cubesat and GOES-PQR. The 2RU-Cubesat and GOES-PQR classes show more improvement in the five light curve case than the Cylinder and SpaceX class. This indicates that the Cylinder and SpaceX classes are difficult to classify and that improvement in classification from combinations of repeated tracks may be limited for similar objects.

5.4.2 Classification Performance on Simultaneous Multi-Site Dataset

The different light curve group combination methods have also been applied to the simultaneous observations dataset. Figure 5.13 displays the comparison between the three best performing methods; Scores Product, MB GAP Frozen Branch and MB Frozen Branch. Figure 5.13a depicts the results based on the number of sensors used for simultaneous observations. For the two site case in Figure 5.13a, the two sites are picked at random for a specific example using a repeatable process so that the same light curves are used for the different combination methods. The classification accuracy is observed to increase for all combination methods as more sites are used to simultaneously observe the object. In comparison, with the repeated tracking dataset, there is a greater increase in performance for the MB GAP Frozen Branch and MB Frozen Branch methods in comparison with Scores Product method.

Figure 5.13b displays the results for the three different two sensor combinations. In this case, for a specific sensor combination, the light curves are input into the same branches in order for the MB Frozen Branch method to potentially learn the best way to combine output vectors. The sensor combination with the largest baseline distance between the sensors, A2/B4, achieved the highest classification accuracy, whilst the A2/C1 combination, which has the smallest baseline distance between the sensors, had the lowest classification accuracy. It is also interesting to note that the MB GAP Frozen Branch model outperformed the MB Frozen Branch model by about 1% for



(a) Specified Number of Sites **(b)** Specified Sensor Combinations

Figure 5.13 – Comparison between Different Light Curve Combination Methods for Simultaneous Observation Data (a) Classification accuracy is found to increase as more sites are used during simultaneous observations with similar results observed for the MB Frozen Branch and MB GAP Frozen Branch models. (b) Comparison between simultaneous observations for various two sensor combinations. Simultaneous observations obtained using both A2 and B4, which have the largest baseline distance between the two sensors, was found to have the highest classification accuracy when combined. Observations using both A2 and C1, which have the lowest baseline distance between two sensors, was found to have the lowest classification accuracy when combined.

the A2/B4 combination, while the opposite occurred for the A2/C1 combination.

When directly comparing the results from the repeated tracking dataset and the simultaneous observations dataset it is important to consider the differences between the two datasets. In particular, there is a smaller number of both light curve groups as well as total light curves in the simultaneous observations dataset and different orbital elements were used to generate the passes. Subsequently, the lower classification accuracy of 74.73% for the baseline model (Scores Product with only one site used during evaluation), compared with 78.61% for the repeated tracking dataset (Scores Product with only one track used during evaluation) is not unexpected. However, it is important to note that the improvement in performance from adding an additional

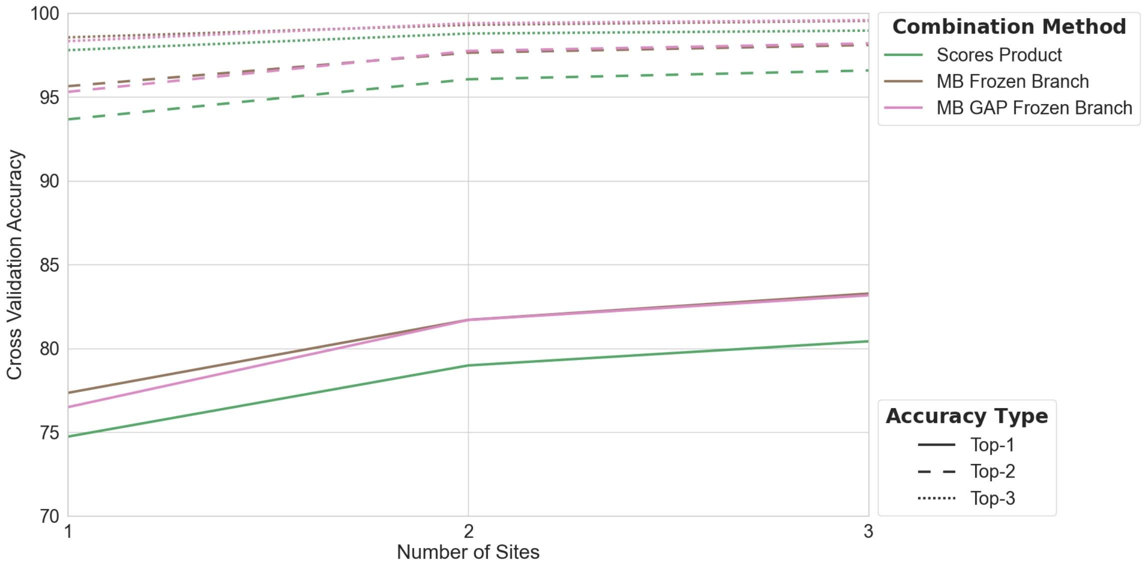


Figure 5.14 – Top-K Classification Accuracy for Different Combination Methods. All three methods achieve a very high Top-2 and Top-3 classification accuracy, with the Top-3 accuracy for the MB GAP Frozen Method Branch using three sites 99.6%.

track/site is similar in both cases and that approximately 15% of light curve groups are unable to be classified correctly in both datasets.

In order to further investigate the reason behind these misclassifications, Figure 5.14 displays the Top-K classification results for the simultaneous tracking dataset, whilst Figure 5.15 depicts the confusion matrix. In Figure 5.14, as well as the Top-1 classification accuracy results, the Top-2 and Top-2 results are depicted for the three different combination methods. For the three site simultaneous observation case, the Top-2 accuracy is 98.2% for the MB GAP Frozen Branch method, while Top-3 accuracy was recorded as 99.6%. These results show that even when an object is not identified as the correct class, the correct class is still in the top two or three predictions.

Figure 5.15 displays the confusion matrix for the three site simultaneous observation case with the light curves combined using the MB GAP Frozen Branch. When taken in context with the top-K results presented in Figure 5.14, the confusion matrix shows that the model is capable of differentiating between general shapes but can have

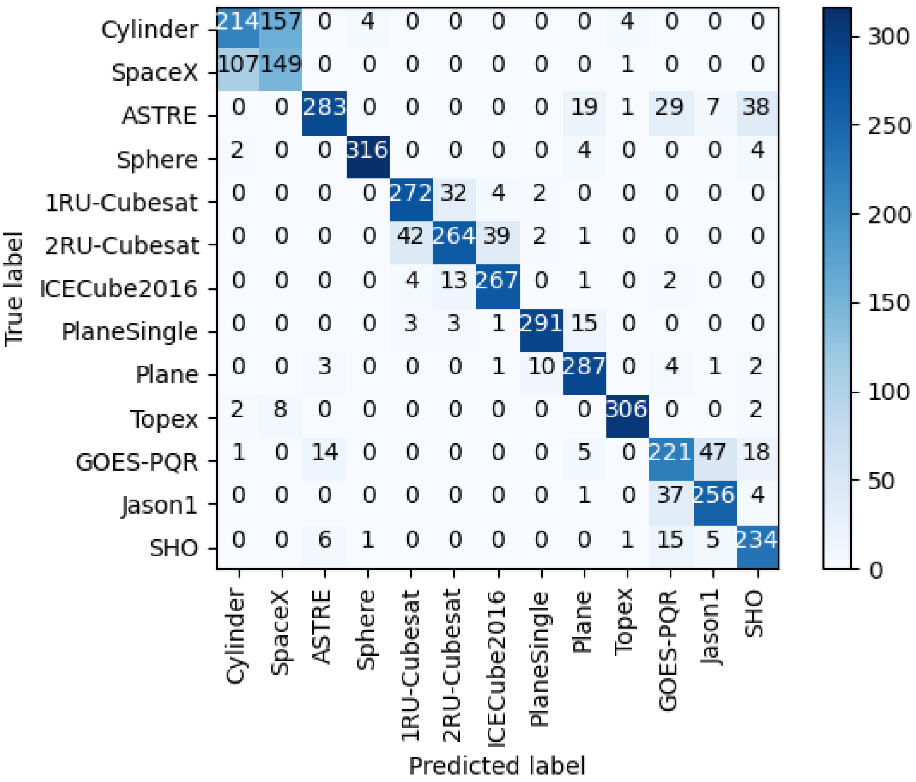
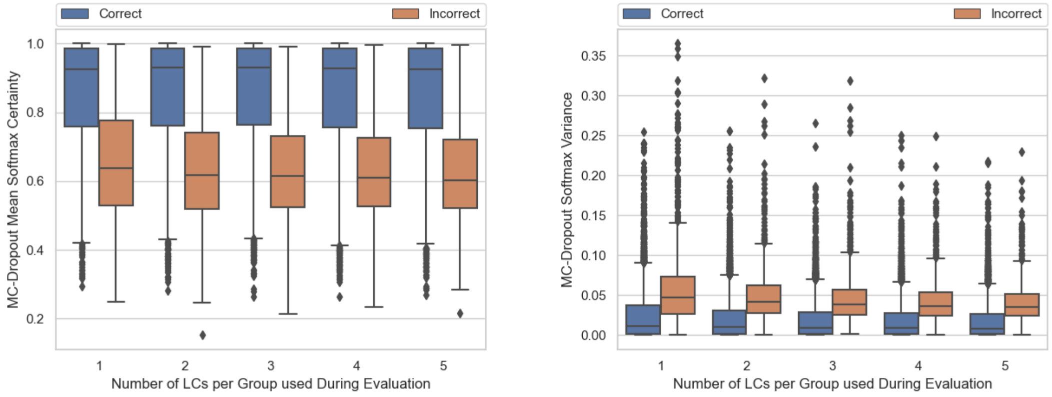


Figure 5.15 – Confusion Matrix for Simultaneous Tracking. Light curves were collected from three sites and the MB GAP Frozen Branch method was used to combine them.

difficulty differentiating between classes with a high inter-class similarity. This is most evident in the confusion matrix through the misclassifications that occur between the Cylinder and SpaceX classes. Similarly, there are a number of misclassifictions between the 1RU-Cubesat and the 2RU-Cubesat as well as the 2RU-Cubesat and the ICECube2016 (3U Cubesat). It is interesting to note that there are relatively few misclassifications between the 1RU-Cubesat and the ICECube2016 models.

To further investigate the ability to differentiate the SpaceX and Cylinder classes, the other shape classes were removed from the dataset and a model was trained on a two class classification problem. Even with simultaneous observations from all three sites and light curve group combination using the MB GAP Frozen Branch method, the model was only able to achieve a classification accuracy of 62.3%. This highlights the similarity between the two classes and the subsequent difficulty in differentiating



(a) Model Certainty **(b)** Softmax Variance

Figure 5.16 – Average model-reported uncertainty in prediction accuracy (using MC-Dropout) vs. number of light curves used during evaluation for Repeated Tracking Dataset. There is a correlation between higher model reported certainty and correct predictions as well as lower variance and correct predictions. A greater separation between the box plots for correct and incorrect predictions is observed as more light curves per group are used during evaluation.

them based on light curve observations.

5.4.3 Use of Model Certainty for Targeted Scheduling

Initial Analysis of Certainty Metrics

Figure 5.16 displays the softmax mean and variance results from the application of MC-Dropout with 100 samples per light curve group to the repeated tracking dataset. The MB GAP Frozen Weights combination method was selected to combine the groups of light curves as it was the best performing method that required only a single model to be trained for a variable number of light curves. There is a strong correlation between both higher mean softmax certainty and correct predictions, depicted in Figure 5.16a, as well as lower softmax variance and correct predictions shown in Figure 5.16b. This indicates that in general the model is more uncertain about examples that it is incorrectly predicting.

While the mean softmax certainty for the correct predictions does not significantly increase as more tracks are added per group during evaluation, the mean softmax certainty does decrease for the incorrect predictions. This results in a greater separation between the lower quartile of the box plot for the correct predictions and the upper quartile of the box plot for the incorrect predictions as more light curves are used during evaluation. A similar trend is observed for the variance results, although it is noted that the variance for both correct and incorrect results decreases as more light curves are observed during evaluation.

Targeted Scheduling Example

As outlined in Section 5.2.3, the certainty of the model can be used in conjunction with the proposed targeted scheduling process. It is expected that the threshold value selected by the user will vary depending on individual requirements, such as the importance of classification accuracy for a given object or dataset. Table 5.5 provides an example of how the proposed targeted scheduling process can be used with the threshold certainty to reduce both misclassifications and sensor tasking requirements. In this example a threshold value of 80% model certainty has been used, with a trained MB GAP Frozen Weights model applied to the repeated tracking dataset.

The scenario commences with no knowledge of any of the objects characteristics and no light curves for any of the objects. As depicted in Figure 5.6, an initial light curve is collected for all objects in the dataset, in this case there are 6301 objects in the repeated tracking dataset. For each object the trained MB GAP Frozen Weights model outputs a predicted shape classification and MC-dropout with 100 samples is used to determine the model certainty. If the model certainty is above the threshold certainty value (i.e. 80% in this example) for a specific object then that object is marked as classified and no further light curves are collected for that object. However, if the model certainty is below the threshold certainty value then an additional light curve is collected. This new light curve is combined with the initial light curve to make a group which is input into the MB GAP Frozen Weights model. This process is repeated until either the model certainty for the shape classification is above the threshold or the maximum of five light curves have been collected.

Table 5.5 – Example of Using the Targeted Scheduling Process with an 80% Certainty Threshold

Tracks	Always Correct	Always Incorrect	Change Correct	Change Incorrect	Step Accuracy	Combined Accuracy	% Objects Classified
1	3458	299	N/A	N/A	92.04	92.04	59.63
2	388	53	90	4	89.35	91.71	68.12
3	130	16	16	0	91.59	91.70	71.51
4	52	12	3	0	87.39	91.60	73.27
5	28	3	1	0	89.39	91.56	74.32

Table 5.5 contains information about how many objects pass the threshold certainty value when a new light curve is added as well as the objects that change to a correct on incorrect classification based on the addition of a light curve. Once a particular object has been marked as classified, it is removed from the dataset and no more light curves are collected for it. The 'Step Accuracy' column indicates the classification accuracy of the objects that were classified at that particular step. The 'Combined Accuracy' column indicates the total classification accuracy of all objects that have been classified. Finally, the '% Objects Classified' column indicates percentage of objects out of repeated tracking dataset, which contains 6301 objects, that have been marked as classified and are subsequently removed from further tracking.

The results presented in Table 5.5 show that with the 80% threshold, 74.32% of the dataset has been classified with a classification accuracy of 91.56%. It is also important to note that 59.63% of the dataset passed the classification certainty threshold with only a single light curve used during evaluation. This reduces the telescope requirements for repeated tracking of objects that are relatively easy to classify and allows the information that is gained from the shape classification to be used for other purposes without waiting for additional light curve collection.

The combination of multiple light curves through the repeated tracking scenario enabled an additional 14.69% of the examples to pass the specified certainty threshold in order to be classified. The biggest relative increase was seen when two light curves were used with a decreasing number of additional examples passing the certainty threshold as more light curves were used during evaluation. It is also notable that

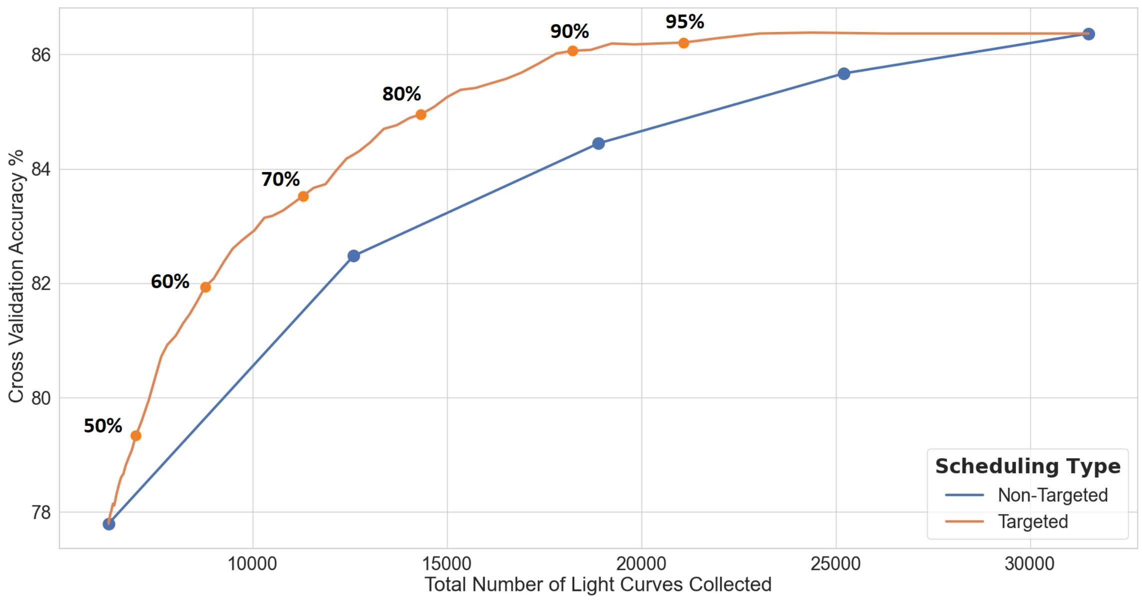


Figure 5.17 – Comparison of the model classification accuracy vs the total number of light curves collected for the targeted scheduling process compared with collecting the same number of light curves for each group. The five data points on the Non-Targeted line (blue) correspond to where a certain number of light curves were used to classify all the objects in the dataset. The Targeted line (orange) is generated by varying the certainty threshold that is used during the targeted scheduling process. The annotated points on this line corresponds to the certainty threshold that was used with the targeted scheduling process to generate that point.

the combination of light curves enabled 109 examples that were initially misclassified to be correctly classified, while only 4 examples that were initially classified correctly became misclassified. In terms of the remaining 25.68% of the dataset, where the model certainty did not pass the threshold value even after the full group of five light curves were used during evaluation, a classification accuracy of 65.82% was recorded.

Comparison between Targeted and Non-Targeted Approach

Figure 5.17 provides a comparison between the targeted scheduling process and the regular approach where certainty is not taken into account (non-targeted). It shows the classification accuracy for all objects in the repeated tracking dataset plotted against the total number of light curves that was collected to achieve that classification accuracy.

The target scheduling process is designed to identify challenging cases and to not classify them if the threshold certainty is too low, thus as was shown in Table 5.5, not all of the dataset is classified. However, in the non-targeted approach, the entire dataset is classified, since there is no concept of a certainty threshold. In order to make an equitable comparison between the two approaches, it was determined that the targeted scheduling process should also classify the entire dataset. Therefore, to generate the results displayed in Figure 5.17 the targeted scheduling approach was set up so that any objects that remained unclassified after five light curves had been collected, were classified using the five light curves even if the prediction certainty was below the threshold.

In Figure 5.17, there are only five data points for the non-targeted approach. These correspond to where a certain number of light curves were used to classify all objects in the dataset (i.e. the first data point corresponds to one light curve being used for each object and the last data point corresponds to five light curves being used in a group for each object). For the targeted approach, the line has been generated by varying the threshold certainty value, which is why the lowest and highest values are the same for both approaches. When the threshold certainty value is set to 0%, all objects are classified using just one light curve. In contrast, when the threshold certainty value is set to 100% none of the objects get marked as classified, so five light curves are collected and used for classification for all objects. The annotated markers on the targeted approach line (orange) indicate where a threshold certainty value was equal to the annotated value.

Figure 5.17 show that the targeted scheduling approach is more efficient at classifying objects compared with the non-targeted approach. This is seen by the fact that less light curves are required to be collected in the targeted scheduling approach to achieve the same level of accuracy.

5.5 Discussion

In this chapter it has been demonstrated that multiple light curves collected for the same RSO can be combined to improve the shape classification performance when compared with classification performance on a single light curve. This results presented in Table 5.3 and Figure 5.12a show that the improvement was most significant for comparatively shorter light curves, where the combination of an additional light curve provided the greatest information gain.

Comparison of Multi-Observation Combination Method Performance

When comparing the different multi-observation combination methods that have been evaluated in this research, the Scores Product method was found to be the best performing of the methods that did not require model architecture changes. This supported results that had previously been reported in the literature on similar view combination datasets [109, 29]. Overall, the MB Frozen Branch and MB GAP Frozen Branch methods were found to have a slight increase in performance when compared to the Scores Product method on the repeated tracking dataset and a more significant increase for the simultaneous tracking dataset. It was notable in Figure 5.11 that the Multi-Branch and Multi-Branch GAP methods had significantly lower performance than any of the other combination methods.

The poor performance of the models solely trained on combined group data (i.e. Multi-Branch and Multi-Branch GAP), demonstrates that this is not an effective method for learning the branch weights responsible for extracting the feature vectors from single light curve inputs. This is likely a result of the fact that models trained on single light curve data get feedback from all light curves, some of which may be difficult to classify, forcing the model to learn weights related to small details and features in the light curves. In contrast, during grouped training there is only a single piece of feedback for the group, which may contain up to five light curves. The strongest features present in the group will enable the model to identify the class of the light curve so that it will learn to identify these features rather than being able to detect the nuances for specific individual light curves. Subsequently, models that used the pre-trained frozen branch weights were able to extract more informative features prior to the feature map combination in comparison with models that were fully trained on group light curve data.

In addition to the classification performance, the model training requirements and ease of use must be considered when evaluating the various multi-observation combination methods. In this regard, the MB Frozen Branch method is impractical to use as it is unable to process groups that contain different numbers of light curves. It was hypothesised that this model might outperform the MB GAP Frozen Branch model when applied to the simultaneous observations dataset, as the GAP method averages the extracted feature maps, which could result in information loss. However, no significant difference was found in performance between these two methods on either dataset. Comparing the Scores Product method with the MB GAP Frozen Branch model, while there is a slight decrease in classification performance for the Scores Product method, it requires no architectural changes to the model and as such can be directly applied to any trained network.

The flexibility of the score combination approaches was was demonstrated in Figure 5.12b, through a comparison between the Baseline 1D-CNN model and the InceptionTime model that has shown state-of-the-art performance on a number of TSC tasks [55]. It was found that on the repeated tracking dataset, the Baseline 1D-CNN outperforms the InceptionTime model, however similar improvements were shown in both models through the combination of additional light curves. In both cases, Scores Product was found to be the best performing of the three score combination approaches. These results, demonstrate the applicability of this combination approach to other neural network types.

Utility of Simultaneous Observations

For this research, the simultaneous observation dataset was generated using TLEs in the high-LEO to low-MEO range to ensure that there would be a large viewing angle between the different sensors. This was done deliberately in order to determine if there was a significant advantage to collecting such observations when attempting to classify challenging RSOs. As orbital altitude increases the difference in viewing angle between two sites decreases (3° to 5° for GEO tracks from A2 and B4), resulting in more similarity between simultaneous tracks and less information to be gained in terms of shape classification.

The improvement in classification performance through the combination of multiple light curves was found to be similar for both the repeated tracking dataset and the simultaneous observation dataset. This suggests that there is no significant advantage to performing simultaneous observations over multiple sites (as opposed to repeated observations from a single site) in terms of RSO shape classification when considering a small number of ground observation sites (i.e. 2-3 sites as explored here). However, it is expected that this type of tracking may be beneficial for analysis of the spin state and orientation of RSOs, due to the measurable time difference between observable features in the simultaneously collected light curves.

Targeted Scheduling Process for Challenging Cases

Although the combination of light curves was found to improve shape classification performance, for both datasets, there remained approximately 15% of examples that were unable to be correctly classified. Similar to the results on the simulated dataset presented in Chapter 4, the misclassifications were primarily due to the high inter-class similarity between objects, particularly the SpaceX rocket body and the cylinder class as well as the different types of Cubesats. These objects were included in the dataset in order to provide challenging cases and the results presented in Figure 5.14 and Figure 5.15 demonstrate that the model is able to successfully perform general shape classification. However, highly detailed classification of specific features from one-dimensional light curve data remains a difficult challenge, even when multiple light curves are combined together to provide the model with additional information. This result suggests that data fusion from alternate observation sources with light curve data may be required in order to attain fine-grained shape classification.

This difficultly in classifying certain objects, also emphasises the importance of using the targeted scheduling approach. The results presented in Section 5.4.3, show that an uncertainty threshold can be effectively used with the targeted scheduling process outlined in Section 5.2.3 to identify RSOs that are challenging to classify. This process enables more efficient use of sensors resources, as it reduces additional scheduling of RSOs that the model is able to easily classify. It also reduces misclassfications by collecting additional light curves for challenging objects as well as identifying cases that the model is uncertain about. Using a certainty threshold of 80%, 74.32% of objects in the dataset had a predicted certainty value higher than the threshold and were subsequently classified, with a classification accuracy of 91.56%. Finally another benefit of this process is that it increases model interpretability for users as well as enabling greater control through the selection of different threshold values depending on the specific task.

5.6 Summary

In this chapter, multi-observation frameworks have been investigated for combining groups of light curves collected for the same RSO, in order to improve classification performance. While all frameworks were demonstrated to increase performance as additional light curves were used per group during evaluation, the Scores Product and the MB GAP Frozen Branch combination methods were found to be the most effective approaches. Both of these combination methods are flexible to a variable number of light curves being combined during evaluation as well as processing light curves of different lengths. Additionally, the Scores Product method was also shown improve shape classification performance when used with the InceptionTime model. This demonstrates that improved performance from the combination of multiple observations is not unique to the specific Baseline 1D-CNN model that was outlined in this chapter and that is also likely to be successful with other neural network models. The targeted scheduling process, based around a model certainty threshold, was shown to be an effective method for utilising the benefits of multiple observations for challenging targets while not significantly increasing sensor tasking demands.

Chapter 6

Conclusions and Future Work

This chapter provides a summary of the content that has been presented in this thesis, as well as a brief outline of the implications of this research in the context of RSO characterisation for SSA. Finally, a discussion of potential future work is also provided to highlight specific areas that could be lead to further improvements in this field of research.

6.1 Summary

This thesis has developed a data-driven approach to RSO characterisation through the utilisation of simulation-based transfer learning in order to overcome the lack of well-labelled training data that had previously restricted the performance of deep learning models on real light curve datasets. Additionally, a framework has been introduced for combining multiple light curves to improve model performance on RSOs that are difficult to classify. An uncertainty-based target scheduling process has also been implemented with this framework, which has been shown to increase the overall classification accuracy achievable for a group of objects while using a fixed number of re-observations.

Chapter 3 focused on the development of a high-fidelity simulation environment for generating realistic light curve sensor measurements of various space objects. The simulator takes in a textured geometric model of an RSO as well as an object's orbital ephemeris and uses a 3D graphics rendering engine to generate photo-realistic images of the RSO that are then processed to extract the light curve. This simulation environment was validated through comparison with real light curve data and has been effective in generating large, well-labelled light curve datasets that have been utilised to perform transfer learning on real light curve datasets.

Chapter 4 presented 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. It is also demonstrated that transfer learning is successful in improving the overall classification accuracy on real light curve datasets. 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.

In Chapter 5 a framework has been introduced for efficiently combining a variable number of light curve observations of a single RSO, in order to improve model performance for challenging RSOs. This framework was shown to be effective for both repeated tracking and simultaneous observation scenarios, with improved classification accuracy compared with a model evaluated using a single light curve. Model certainty has also been implemented with this framework in order to use a threshold based process for targeted scheduling. This process allows the identification of RSOs that are challenging to classify, enabling the collection and combination of additional light curves, resulting in both a reduction in misclassifications as well as an improvement in the efficiency of sensor utilisation.

6.2 Implications to RSO Characterisation for SSA

The primary motivation behind this body of research was to improve RSO characterisation methods in order to enable high-fidelity orbital propagation, thus reducing the space debris risk to operational satellites and human space flights/missions. Currently, detailed RSO characteristics are not typically used in real world operations to perform orbital propagation due to the difficulty in performing RSO characterisation. The contributions that have been developed in this thesis provide a significant step towards the utilisation of an automated data-driven approach for RSO characterisation from observational light curve data. This section briefly highlights key findings in this research that reduce the difficulty of implementing such an approach in the real world operations.

The demonstration of the utility of simulation-based transfer learning for RSO shape classification, reduces the barriers to entry that may have previously prevented organisations from using a deep learning approach. In particular, 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.

Additionally, the proposed framework for multi-observation combination also reduces requirements for the collection of long light curve tracks for characterisation purposes. It was initially assumed that the observation of multiple rotations in succession, would be required in order to obtain enough information to correctly classify the object. However, the results presented in Chapter 5 demonstrate that multiple shorter light curve tracks can be combined, with the variation in sensor viewing angle relative to the orientation and spin axis of the RSO enabling comparable performance. This result enables organisations to classify RSOs using light curves collected during regular state update observations without the requirement for specialised characterisation observations.

Finally, the implementation of uncertainty into the model framework is viewed as an important initiative for the real world utilisation of a deep learning model applied to RSO characterisation. Not only does this allow for the identification of challenging cases to improve sensor efficiency and reduce misclassifications but it also provides the user with a better understanding of model limitations. Models trained to perform mutually exclusive classification for certain shape classes are only capable of classifying objects into classes that they have been presented with during training. In real world operations, there are a range of possible shape classes that will need to be considered and if the model has not been presented with a certain class during training it will not be able to correctly classify it. In this case, the model certainty threshold should indicate that the model is uncertain about the class of the object and it can be referred to a human expert for further evaluation.

6.3 Future Work

Combined Shape Classification and Attitude Prediction

The research presented in this thesis has focused on shape classification from light curve data. However, as outlined in Chapter 2, there are a number of other physical characteristics, including attitude, size, mass and material properties. In particular, determination of the attitude of the RSO is seen as important both for improving orbital predictions [36, 128] and potential active debris removal missions [34, 32]. Consequently, it would be beneficial to investigate the utility of a network architecture that is able to perform both shape classification and attitude prediction. The research into simultaneous multi-site light curve observations presented in Chapter 5 suggests that the incorporation of simultaneous observations will be useful when determining the attitude of the RSO.

Higher Sampling Rates for Fine-Grained Classification

The development of new sensor types, including event-based sensors [21, 59] and single-photon avalanche diodes [75], raises the possibility of optical observations with a very high sampling rate. It is expected that this kind of data would allow more features to be observed in the extracted light curve, particularly spectral reflections, providing more information to a classifier model. This is likely to enable more fine-grained classification and assist with the differentiation between similar classes. Subsequently, as these sensor types become more widely available, this data should be considered to determine if the additional information does in fact lead to improved model performance, compared to the sensor sampling rates considered in this thesis.

Combination of Multiple Observation Types

It has been demonstrated in this thesis that multiple light curve observation can be combined to improve classification performance, however it was found that even using this method there was still challenging cases that could not be classified correctly. As outlined in Section 2.2 there are a number of different RSO observation methods, each providing unique characteristic information about the object. By combining different sensor modalities using a multi-observation framework similar to the one outlined in Chapter 5, information from different observation types could be input into the model in order to improve performance. Specifically, spectroscopy data or thermal observations could be used provide additional information about the material characteristics or laser ranging data to determine precise information shape of the object as it rotates. This is seen as a promising avenue of research that is likely to be required in order to fully characterise an RSO from ground-based observations.

Utilisation of Contemporary Neural Network Architectures and Techniques

The field of deep learning is a rapidly developing and dynamic research area, with a number of different models for TSC being proposed in the last few years as well as improvement in model selection and training techniques. Automated hyperparameter optimisation is now commonly used to assist in the selection of model structure and training hyperparameters as opposed to the grid search method that was used in this research. The use of such techniques may result in small improvements in the classification results in future work. Additionally, future research into RSO classification from observational data, should keep up to date with advances in deep learning models applied to TSC tasks in other domains and investigate the utility of contemporary models. In particular, recent advancements in transformer architectures using a self-attention mechanism [125], has resulted in them becoming the state-of-the-art network architecture for sequence-to-sequence tasks [27]. While the application of transformers to TSC tasks is an ongoing area of research, initial results appear promising and these types of networks may offer an improvement on the current convolutional based models.

Generation of Shape Models based on Observational Data

While the research outlined in this thesis has focused on classification models, from a SSA perspective, the ultimate goal would be to use the neural network to generate a 3D rendered shape model of the observed RSO, similar to the those used in the simulation environment. This would enable the generated shape model to be used directly for precise orbital propagation. While this is a more challenging task than shape classification, it is expected that the methods that have been developed in this thesis would also be applicable. As the neural network developed for such a task would require significantly more parameters than for the TSC task, simulation-based transfer learning will be important for efficiently training the network. Additionally, the combination of multiple observations, either light curves or other observation types, is likely required to provide such a model with the required information to generate a complex shape model.

List of References

[1] J. Africano, E. Stansbery, and P. Kervin. The optical orbital debris measurement program at NASA and AMOS. *Advances in Space Research*, 34 (5):892–900, January 2004. ISSN 02731177.

[2] G. Aglietti, B. Taylor, S. Fellowes, T. Salmon, I. Retat, A. Hall, T. Chabot, A. Pisseloup, C. Cox, A. Zarkesh, A. Mafficini, N. Vinkoff, K. Bashford, C. Bernal, F. Chaumette, A. Pollini, and W. Steyn. The active space debris removal mission RemoveDebris. Part 2: In orbit operations. *Acta Astronautica*, 168:310–322, March 2020. ISSN 00945765.

[3] J. Allworth, L. Windrim, J. Wardman, D. Kucharski, J. Bennett, and M. Bryson. Development of a High Fidelity Simulator for Generalised Photometric Based Space Object Classification using Machine Learning. In *Proceedings of the International Astronautical Congress*, volume 2019, page 14, Washington, 2019.

[4] J. Allworth, L. Windrim, J. Bennett, and M. Bryson. A transfer learning approach to space debris classification using observational light curve data. *Acta Astronautica*, 181:301–315, April 2021. ISSN 00945765.

[5] P. Anz-Meador. Orbital Debris Quarterly News. *NASA Orbital Debris Program Office*, 25(2):12, June 2021.

[6] M. Ashikhmin and P. Shirley. An Anisotropic Phong Light Reflection Model. Technical report, Journal of Graphics Tools, 2000.

[7] S. Aslani, M. Dayan, L. Storelli, M. Filippi, V. Murino, M. Rocca, and D. Sona. Multi-branch convolutional neural network for multiple sclerosis lesion segmentation. *NeuroImage*, 196:1–15, August 2019. ISSN 10538119.

[8] V. Aslanov and V. Yudintsev. Dynamics of large space debris removal using tethered space tug. *Acta Astronautica*, 91:149–156, October 2013. ISSN 00945765.

[9] J. Bangert, W. Puatua, G. Kaplan, J. Bartlett, W. Harris, A. Fredericks, and A. Monet. User's Guide to NOVAS Version C3.1. Technical report, United States Naval Observatory, Washington, DC, 2011.

[10] F. Bennet, R. Conan, C. D'Orgeville, M. Dawson, N. Paulin, I. Price, F. Rigaut, I. Ritchie, C. Smith, and K. Uhlendorf. Adaptive optics for laser space debris removal. In B. Ellerbroek, E. Marchetti, and J.-P. Véran, editors, *SPIE Astronomical Telescopes + Instrumentation*, pages 844744–844744–6, Amsterdam, , Netherlands, September 2012.

[11] J. Bennett, M. Lachut, D. Kucharski, S. Flegel, M. Möckel, J. Allworth, D. Kooymans, A. Pollard, C. Smith, J. O'Leary, H. Kankanamalage, and R. Samuel. Progress in a new conjunction and threat warning service for Space Situational Awareness. In *Advanced Maui Optical and Space Surveillance Tech. Conf.(AMOS)*, pages 1–12, Maui, Hawaii, September 2018.

[12] W. Bennette, K. Zeliff, and J. Raquepas. Classification of objects in geosynchronous earth orbit via light curve analysis. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6, November 2017.

[13] C. Benson, D. Scheeres, W. Ryan, E. Ryan, and N. Moskovitz. GOES spin state diversity and the implications for GEO debris mitigation. *Acta Astronautica*, 167:212–221, February 2020. ISSN 00945765.

[14] G. Beskin, S. Karpov, A. Biryukov, S. Bondar, E. Ivanov, E. Katkova, N. Orekhova, A. Perkov, and V. Sasyuk. Wide-field optical monitoring with Mini-MegaTORTORA (MMT-9) multichannel high temporal resolution telescope. *Astrophysical Bulletin*, 72(1):81–92, January 2017. ISSN 1990-3413, 1990-3421.

[15] C. Blackerby, A. Okamota, S. Iizuka, Y. Kobayashi, K. Fujimoto, Y. Seto, S. Fujita, T. Iwai, N. Okada, J. Forshaw, and A. Bradford. The ELSA-d End-of-life Debris Removal Mission: Preparing for Launch. In *Proceedings of the International Astronautical Congress*, Washington, DC, October 2019.

[16] J. Blake, P. Chote, D. Pollacco, W. Feline, G. Privett, A. Ash, S. Eves, A. Greenwood, N. Harwood, T. Marsh, D. Veras, and C. Watson. DebrisWatch I: A survey of faint geosynchronous debris. *Advances in Space Research*, 67(1): 360–370, January 2021. ISSN 02731177.

[17] A. Brunel, J. Pasquet, J. Pasquet, N. Rodriguez, F. Comby, D. Fouchez, and M. Chaumont. A CNN adapted to time series for the classification of Supernovae. *arXiv:1901.00461 [cs, stat]*, January 2019. arXiv: 1901.00461.

[18] B. Calef, J. Africano, B. Birge, D. Hall, and P. Kervin. Photometric signature inversion. In V. Gamiz, P. Idell, and M. Strojnik, editors, *Proceedings of SPIE*, page 63070E, San Diego, California, USA, August 2006.

[19] M. Castronuovo. Active space debris removal—A preliminary mission analysis and design. *Acta Astronautica*, 69(9-10):848–859, November 2011. ISSN 00945765.

[20] F. Chollet. *Deep learning with Python.* Manning, Shelter Island, NY, 2018. ISBN 978-1-61729-443-3. OCLC: 1019988472.

[21] G. Cohen, S. Afshar, B. Morreale, T. Bessell, A. Wabnitz, M. Rutten, and A. van Schaik. Event-based Sensing for Space Situational Awareness. *The Journal of the Astronautical Sciences*, 66(2):125–141, June 2019. ISSN 0021-9142, 2195-0571.

[22] R. Cook and K. Torrance. A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1):18, 1982.

[23] M. Copeland, F. Bennet, F. Rigaut, V. A. Korkiakoski, C. d'Orgeville, and C. Smith. Adaptive optics corrected imaging for satellite and debris characterisation. In D. Schmidt, L. Schreiber, and L. Close, editors, *Adaptive Optics Systems VI*, page 113, Austin, United States, July 2018. SPIE. ISBN 978-1-5106-1959-3 978-1-5106-1960-9.

[24] H. Cowardin, G. Ojakangas, M. Mulrooney, and S. Lederer. Optical signature analysis of tumbling rocket bodies via laboratory measurements. In *Proceedings of AMOS 2012 Technical Conference*, page 12, Maui, Hawaii, September 2012.

[25] H. Cowardin. Orbital Debris Quarterly News. *NASA Orbital Debris Program Office*, 25(4):14, December 2021.

[26] H. Cowardin, K. Abercromby, E. Barker, P. Seitzer, M. Mulrooney, and T. Schildknecht. An Assessment of GEO Orbital Debris Photometric Properties Derived from Laboratory-Based Measurements. In *Proceedings of AMOS 2009 Technical Conference*, Maui, Hawaii, September 2009.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

[28] A. Dianetti, R. Weisman, and J. Crassidis. Observability Analysis for Improved Space Object Characterization. *Journal of Guidance, Control, and Dynamics*, 41(1):137–148, January 2018. ISSN 0731-5090, 1533-3884.

[29] T.-B. Do, H.-H. Nguyen, T.-T.-N. Nguyen, H. Vu, T.-T.-H. Tran, and T.-L. Le. Plant identification using score-based fusion of multi-organ images. In *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pages 191–196, Hue, October 2017. IEEE. ISBN 978-1-5386-3576-6.

[30] R. Eapen and C. Frueh. Averaged solar radiation pressure modeling for high area-to-mass ratio objects in geosynchronous orbits. *Advances in Space Research*, 62(1):127–141, July 2018. ISSN 02731177.

[31] ESA. Space Debris by the Numbers. September 2021. URL https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers.

[32] S. Estable, C. Pruvost, E. Ferreira, J. Telaar, M. Fruhnert, C. Imhof, T. Rybus, G. Peckover, R. Lucas, R. Ahmed, T. Oki, M. Wygachiewicz, P. Kicman, A. Lukasik, N. Santos, T. Milhano, P. Arroz, R. Biesbroek, and A. Wolahan. Capturing and deorbiting Envisat with an Airbus Spacetug. Results from the ESA e.Deorbit consolidation phase study. *Journal of Space Safety Engineering*, 7(1):52–66, March 2020. ISSN 24688967.

[33] S. Fan and C. Frueh. A Direct Light Curve Inversion Scheme in the Presence of Measurement Noise. *The Journal of the Astronautical Sciences*, 67(2): 740–761, June 2020. ISSN 0021-9142, 2195-0571.

[34] J. Forshaw, G. Aglietti, S. Fellowes, T. Salmon, I. Retat, A. Hall, T. Chabot, A. Pisseloup, D. Tye, C. Bernal, F. Chaumette, A. Pollini, and W. Steyn. The active space debris removal mission RemoveDebris. Part 1: From concept to launch. *Acta Astronautica*, 168:293–309, March 2020. ISSN 00945765.

[35] C. Frueh and T. Schildknecht. Analysis of observed and simulated light curves of space debris. In *Proceedings of 61st International Astronautical Congress*, volume 1, pages 194–203, Prague, Czech Republic, September 2010.

[36] C. Frueh, M. Jah, and T. Kelecy. Coupled Orbit-Attitude Dynamics of High Area-to-Mass Ratio (HAMR) Objects: Influence of Solar Radiation Pressure, Earth's Shadow and the Visibility in Light Curves. *Celestial Mechanics and Dynamical Astronomy*, 117(4):385–404, December 2013. ISSN 0923-2958, 1572-9478. arXiv: 1312.0067.

[37] R. Furfaro, R. Linares, D. Gaylor, M. Jah, and R. 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, September 2016.

[38] R. Furfaro, R. Linares, and V. Reddy. Space Objects Classification via Light-Curve Measurements: Deep Convolutional Neural Networks and Model-based Transfer Learning. In *Proceedings of AMOS Conference*, pages 1–17, Maui, Hawaii, September 2018.

[39] R. Furfaro, R. Linares, and V. Reddy. Space Debris Identification and Characterization via Deep Meta-Learning. In *First Inernational Orbital Debris Conference*, page 9, 2019.

[40] Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of Machine Learning Research*, volume 48, pages 1050–1059, New York, New York, USA, 2016. PMLR.

[41] F. Gasdia, A. Barjatya, and S. Bilardi. Multi-Site Simultaneous Time-Resolved Photometry with a Low Cost Electro-Optics System. *Sensors*, 17(6):1239, May 2017. ISSN 1424-8220.

[42] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning Series. MIT Press, Cambridge, MA, 1st edition, 2017.

[43] B. Greene. Laser Tracking of Space Debris. In *13 International Workshop on Laser Ranging Instrumentation*, page 27, Washington, DC, 2002.

[44] D. Grosse, F. Bennet, F. Rigaut, C. d'Orgeville, V. Korkiakoski, C. Smith, M. Copeland, I. Price, M. Blundell, A. Chan, M. Ellis, A. Galla, L. Gers, J. Hart, M. Lingham, Y. Gao, E. Houston, E. Rees, Y. Wang, I. Ritchie, T. Travouillon, A. Vaccarella, and J. Webb. Adaptive optics tracking and pushing system for space debris manoeuvre. In D. Schmidt, L. Schreiber, and L. Close, editors, *Adaptive Optics Systems VI*, page 24, Austin, United States, July 2018. SPIE. ISBN 978-1-5106-1959-3 978-1-5106-1960-9.

[45] D. Hall, B. Calef, K. Knox, M. Bolden, and P. Kervin. Separating Attitude and Shape Effects for Non-resolved Objects. In *Proceedings of AMOS Conference*, pages 1–12, Maui, Hawaii, September 2007.

[46] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1.

[47] M. Holzinger and M. Jah. Challenges and Potential in Space Domain Awareness. *Journal of Guidance, Control, and Dynamics*, 41(1):15–18, January 2018. ISSN 0731-5090, 1533-3884.

[48] M. Holzinger, K. Alfriend, C. Wetterer, K. Luu, C. Sabol, and K. Hamada. Photometric Attitude Estimation for Agile Space Objects with Shape Uncertainty. *Journal of Guidance, Control, and Dynamics*, 37(3):921–932, May 2014. ISSN 0731-5090, 1533-3884.

[49] M. Howard, B. Klem, and J. Gorman. RSO Characterization with Photometric Data Using Machine Learning. In *Advanced Maui Optical and Space Surveillance Tech. Conf.(AMOS)*, page 10, Maui, Hawaii, 2015.

[50] C. Hubaux, A. Lemaître, N. Delsate, and T. Carletti. Symplectic integration of space debris motion considering several Earth's shadowing models. *Advances in Space Research*, 49(10):1472–1486, May 2012. ISSN 02731177.

[51] Y. Huo, Z. Li, Y. Fang, and F. Zhang. Classification for geosynchronous satellites with deep learning and multiple kernel learning. *Applied Optics*, 58 (21):5830, July 2019. ISSN 1559-128X, 2155-3165.

[52] D. Isele and A. Cosgun. Transferring Autonomous Driving Knowledge on Simulated and Real Intersections. *arXiv:1712.01106 [cs]*, November 2017. arXiv: 1712.01106.

[53] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Transfer learning for time series classification. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1367–1376, Seattle, WA, USA, December 2018. IEEE. ISBN 978-1-5386-5035-6.

[54] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, July 2019. ISSN 1384-5810, 1573-756X.

[55] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. Schmidt, J. Weber, G. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean. InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, November 2020. ISSN 1384-5810, 1573-756X.

[56] M. Jah and R. Madler. Satellite Characterization: Angles and Light Curve Data Fusion for Spacecraft State and Parameter Estimation. In *Proceedings of AMOS Conference*, pages 1–10, Maui, Hawaii, September 2007.

[57] B. Jia, K. Pham, E. Blasch, Z. Wang, D. Shen, and G. Chen. Space Object Classification using Deep Neural Networks. In *2018 IEEE Aerospace Conference*, pages 1–8, March 2018.

[58] B. Jia, K. Pham, E. Blasch, D. Shen, Z. Wang, and G. Chen. Space Object Classification using Fused Features of Time Series Data. In *Advanced Maui Optical and Space Surveillance Tech. Conf.(AMOS)*, pages 1–12, September 2017.

[59] A. Jolley, G. Cohen, D. Joubert, and A. Lambert. Evaluation of Event-Based Sensors for Satellite Material Characterization. *Journal of Spacecraft and Rockets*, pages 1–10, October 2021. ISSN 0022-4650, 1533-6794.

[60] M. Kaasalainen. Optimization Methods for Asteroid Lightcurve Inversion I. Shape Determination. *Icarus*, 153(1):24–36, September 2001. ISSN 00191035.

[61] M. Kaasalainen. Optimization Methods for Asteroid Lightcurve Inversion II. The Complete Inverse Problem. *Icarus*, 153(1):37–51, September 2001. ISSN 00191035.

[62] R. Kanzlar, J. Silha, T. Schildknecht, B. Fritsche, T. Lips, and H. Krag. Space Debris Attitude Simulation - iOTA (In-Orbit Tumbling Analysis). In *Advanced Maui Optical and Space Surveillance Tech. Conf.(AMOS)*, Hawaii, 2015.

[63] P. Kervin, J. Africano, P. Sydney, and D. Hall. Small satellite characterization technologies applied to orbital debris. *Advances in Space Research*, 35(7): 1214–1225, January 2005. ISSN 02731177.

[64] D. Kessler and B. Cour-Palais. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83 (A6):2637–2646, June 1978. ISSN 2156-2202.

[65] D. Kessler and N. Johnson. The Kessler Syndrome: Implications to Future Space operations. *Advances in Astronautical Sciences*, 137(8):47–61, 2010.

[66] M. Khalil, E. Fantino, and P. Liatsis. Evaluation of Oversampling Strategies in Machine Learning for Space Debris Detection. In *2019 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6, Abu Dhabi, United Arab Emirates, December 2019. IEEE. ISBN 978-1-72813-868-8.

[67] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference for Learning Representations*, San Diego, California, USA, 2015. arXiv: 1412.6980.

[68] H. Klinkrad. *Space Debris - Models and Risk Analysis.* Springer, Chichester, UK, 2016.

[69] N. Koshkin, L. Shakun, E. Korobeynikova, S. Melikyants, S. Strakhova, V. Dragomiretsky, A. Ryabov, T. Golubovskaya, and S. Terpan. Monitoring of Space Debris Rotation Based on Photometry. *Odessa Astronomical Publications*, 31(0):179–185, November 2018. ISSN 1810-4215.

[70] N. Koshkin, L. Shakun, O. Kozhukhov, D. Kozhukhov, V. Mamarev, V. Prysiaznyi, A. Ozeryan, V. Kudak, and I. Neubauer. Simultaneous Multi-Site Photometry of LEO Satellites for Rotation Characterization. In *Proceedings 8th European Conference on Space Debris*, page 8, 2021.

[71] A. Kramer. Russia Acknowledges Antisatellite Missile Test that Created a Mess in Space. *New York Times*, November 2021.

[72] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017. ISSN 0001-0782, 1557-7317.

[73] D. Kucharski, G. Kirchner, F. Koidl, C. Fan, R. Carman, C. Moore, A. Dmytrotsa, M. Ploner, G. Bianco, M. Medvedskij, A. Makeyev, G. Appleby,

M. Suzuki, J. Torre, Z. Zhongping, L. Grunwaldt, and Q. Feng. Attitude and Spin Period of Space Debris Envisat Measured by Satellite Laser Ranging. *IEEE Transactions on Geoscience and Remote Sensing*, 52(12):7651–7657, December 2014. ISSN 0196-2892.

[74] D. Kucharski, G. Kirchner, T. Otsubo, H.-C. Lim, J. Bennett, F. Koidl, Y.-R. Kim, and J.-Y. Hwang. Confirmation of gravitationally induced attitude drift of spinning satellite Ajisai with Graz high repetition rate SLR data. *Advances in Space Research*, 57(4):983–990, February 2016. ISSN 02731177.

[75] D. Kucharski, G. Kirchner, M. Jah, J. Bennett, F. Koidl, M. Steindorfer, and P. Wang. Full attitude state reconstruction of tumbling space debris TOPEX/Poseidon via light-curve inversion with Quanta Photogrammetry. *Acta Astronautica*, 187:115–122, October 2021. ISSN 00945765.

[76] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, December 1989. ISSN 0899-7667, 1530-888X.

[77] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 00189219.

[78] R. Linares and J. Crassidis. Space-Object Shape Inversion via Adaptive Hamiltonian Markov Chain Monte Carlo. *Journal of Guidance, Control, and Dynamics*, 41(1):47–58, January 2018. ISSN 0731-5090, 1533-3884.

[79] R. Linares and R. Furfaro. Space Object Classification Using Deep Convolutional Neural Networks. In *Information Fusion (FUSION), 2016 19th International Conference on*, pages 1140–1146. IEEE, August 2016.

[80] R. Linares, M. Jah, and J. Crassidis. Inactive Space Object Shape Estimation Via Astrometric And Photometric Data Fusion. *Advances in Astronautical Sciences*, 143:217–232, 2012.

[81] R. Linares, M. Jah, J. Crassidis, F. Leve, and T. Kelecy. Astrometric and photometric data fusion for inactive space object mass and area estimation. *Acta Astronautica*, 99:1–15, June 2014. ISSN 00945765.

[82] R. Linares, M. Jah, J. Crassidis, and C. Nebelecky. Space Object Shape Characterization and Tracking Using Light Curve and Angles Data. *Journal of Guidance, Control, and Dynamics*, 37(1):13–25, January 2014. ISSN 0731-5090, 1533-3884.

[83] R. Linares, R. Furfaro, and V. Reddy. Space Objects Classification via Light-Curve Measurements Using Deep Convolutional Neural Networks. *The Journal of the Astronautical Sciences*, March 2020. ISSN 0021-9142, 2195-0571.

[84] E. Linder, J. Silha, T. Schildknecht, and M. Hager. Extraction of Spin Periods of Space Debris from Optical Light Curves. In *International Astronautical Conference*, 2015.

[85] J.-C. Liou and N. Johnson. Risks in Space from Orbiting Debris. *Science*, 311 (5759):340–341, January 2006. ISSN 0036-8075, 1095-9203.

[86] T. Liu and U. Schreiber. Photometric space object classification via deep learning algorithms. *Acta Astronautica*, 185:161–169, August 2021. ISSN 00945765.

[87] N. Lomb. Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space Science*, 39(2):447–462, February 1976. ISSN 0004-640X, 1572-946X.

[88] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning Transferable Features with Deep Adaptation Networks. *Journal of Machine Learning Research*, 37:9, 2015.

[89] A. Lundervold and A. Lundervold. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*, 29(2):102–127, May 2019. ISSN 09393889.

[90] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the Limits of Weakly Supervised Pretraining. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, volume 11206, pages 185–201. Springer International Publishing, Cham, 2018. ISBN 978-3-030-01215-1 978-3-030-01216-8. Series Title: Lecture Notes in Computer Science.

[91] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the Limits of Weakly Supervised Pretraining. *arXiv:1805.00932 [cs]*, May 2018. arXiv: 1805.00932.

[92] V. Mallik and M. Jah. Reconciling space object observed and solar pressure albedo-areas via astrometric and photometric data fusion. *Advances in Space Research*, 63(1):404–416, January 2019. ISSN 02731177.

[93] D. Malmgren-Hansen, A. Kusk, J. Dall, A. Nielsen, R. Engholm, and H. Skriver. Improving SAR Automatic Target Recognition Models With Transfer Learning From Simulated Data. *IEEE Geoscience and Remote Sensing Letters*, 14(9):1484–1488, September 2017. ISSN 1545-598X, 1558-0571.

[94] D. Marmanis, M. Datcu, T. Esch, and U. Stilla. Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1):105–109, January 2016. ISSN 1545-598X, 1558-0571.

[95] P. McCall, M. Naudeau, and M. Adjouadi. Debris characterization techniques via unresolved long-wave infrared imaging from a space-based platform. *Journal of Applied Remote Sensing*, 8(1):084989, May 2014. ISSN 1931-3195.

[96] J. McMahon and D. Scheeres. Improving Space Object Catalog Maintenance Through Advances in Solar Radiation Pressure Modeling. *Journal of Guidance, Control, and Dynamics*, 38(8):1366–1381, August 2015. ISSN 0731-5090, 1533-3884.

[97] P. Mehta, R. Linares, and A. Walker. Photometric Data from Nonresolved Objects for Improved Drag and Reentry Prediction. *Journal of Spacecraft and Rockets*, 55(4):959–970, July 2018. ISSN 0022-4650, 1533-6794.

[98] H. Minkowski. Volumen and Oberflache. *Mathematische Annalen*, 57:447–495, 1903.

[99] C. Pardini and L. Anselmo. Review of past on-orbit collisions among cataloged objects and examination of the catastrophic fragmentation concept. *Acta Astronautica*, 100:30–39, July 2014. ISSN 00945765.

[100] C. Pelletier, G. Webb, and F. Petitjean. Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series. *Remote Sensing*, 11(5):523, March 2019. ISSN 2072-4292.

[101] F. Piergentili, G. Zarcone, L. Parisi, L. Mariani, S. H. Hossein, and F. Santoni. LEO Object's Light-Curve Acquisition System and Their Inversion for Attitude Reconstruction. *Aerospace*, 8(1):4, December 2020. ISSN 2226-4310.

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

[103] J. Sang, J. Bennett, and C. Smith. Estimation of ballistic coefficients of low altitude debris objects from historical two line elements. *Advances in Space Research*, 52(1):117–124, July 2013. ISSN 02731177.

[104] J. Scargle. Studies in astronomical time series analysis. II-Statistical apsects of spectral analysis on unevenly spaced data. *The Astrophysical Journal*, 263: 835–853, 1982.

[105] T. Schildknecht, M. Ploner, and U. Hugentobler. The search for debris in GEO. *Advances in Space Research*, 28(9):1291–1299, 2001.

[106] T. Schildknecht, R. Musci, M. Ploner, G. Beutler, W. Flury, J. Kuusela, J. de Leon Cruz, and L. de Fatima Dominguez Palmero. Optical observations of space debris in GEO and in highly-eccentric orbits. *Advances in Space Research*, 34(5):901–911, January 2004. ISSN 0273-1177.

[107] T. Schildknecht, R. Musci, C. Früh, and M. Ploner. Color Photometry And Light Curve Observations Of Space Debris In GEO. In *Proceedings of AMOS Conference*, pages 1–6, Maui, Hawaii, September 2008.

[108] T. Schildknecht. Optical surveys for space debris. *The Astronomy and Astrophysics Review*, 14(1):41–111, January 2007. ISSN 0935-4956, 1432-0754.

[109] M. Seeland and P. Mäder. Multi-view classification with convolutional neural networks. *PLOS ONE*, 16(1):e0245230, January 2021. ISSN 1932-6203.

[110] A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. van Riel, M. Wille, M. Naqibullah, C. Sanchez, and B. van Ginneken. Pulmonary Nodule Detection in CT Images: False Positive Reduction Using Multi-View Convolutional Networks. *IEEE Transactions on Medical Imaging*, 35(5):1160–1169, May 2016. ISSN 0278-0062, 1558-254X.

[111] S. Shao, S. McAleer, R. Yan, and P. Baldi. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Transactions on Industrial Informatics*, 15(4):2446–2455, April 2019. ISSN 1551-3203, 1941-0050.

[112] J. Šilha, J.-N. Pittet, M. Hamara, and T. Schildknecht. Apparent rotation properties of space debris extracted from photometric measurements. *Advances in Space Research*, 61(3):844–861, February 2018. ISSN 02731177.

[113] J. Šilha, S. Krajčovič, M. Zigo, J. Tóth, D. Žilková, P. Zigo, L. Kornoš, J. Šimon, T. Schildknecht, E. Cordelli, A. Vananti, H. K. Mann, A. Rachman, C. Paccolat, and T. Flohrer. Space debris observations with the Slovak AGO70 telescope: Astrometry and light curves. *Advances in Space Research*, page S0273117720300727, February 2020. ISSN 02731177.

[114] J. Šilha, M. Zigo, T. Hrobár, P. Jevčák, and M. Verešvárska. LIGHT CURVES APPLICATION TO SPACE DEBRIS CHARACTERIZATION AND CLASSIFICATION. In *Proceedings of the 8th European Conference on Space Debris*, page 8, Darmstadt, Germany, April 2021. ESA Space Debris Office.

[115] N. Singh, J. Horwood, J. Aristoff, and J. Murray-Krezan. Athena: A data-driven anomaly detection and space object classification tool for SSA. *Advances in Astronautical Sciences*, 158:4097–4116, February 2016.

[116] M. Skinner, R. Russell, R. Rudy, D. Gutierrez, D. Kim, K. Crawford, S. Gregory, and T. Kelecy. Time-resolved infrared spectrophotometric

observations of high area to mass ratio (HAMR) objects in GEO. *Acta Astronautica*, 69(11-12):1007–1018, December 2011. ISSN 00945765.

[117] M. Skinner, R. Russell, T. Kelecy, S. Gregory, R. Rudy, D. Gutierrez, D. Kim, and K. Crawford. Further analysis of infrared spectrophotometric observations of high area to mass ratio (HAMR) objects in GEO. *Acta Astronautica*, 80: 154–165, November 2012. ISSN 00945765.

[118] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[119] V. Strijbis, C. de Bloeme, R. Jansen, H. Kebiri, H.-G. Nguyen, M. de Jong, A. Moll, M. Bach-Cuadra, P. de Graaf, and M. Steenwijk. Multi-view convolutional neural networks for automated ocular structure and tumor segmentation in retinoblastoma. *Scientific Reports*, 11(1):14590, December 2021. ISSN 2045-2322.

[120] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, Santiago, Chile, December 2015. IEEE. ISBN 978-1-4673-8391-2.

[121] D. Vallado and P. Crawford. SGP4 Orbit Determination. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, August 2008. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-001-7.

[122] D. Vallado and W. McClain. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, Hawthorne, California, 4 edition, March 2013. ISBN 978-1-881883-18-0.

[123] A. Vananti, T. Schildknecht, and H. Krag. Reflectance spectroscopy characterization of space debris. *Advances in Space Research*, 59(10): 2488–2500, May 2017. ISSN 02731177.

[124] J. VanderPlas. Understanding the Lomb–Scargle Periodogram. *The Astrophysical Journal Supplement Series*, 236(1):16, May 2018. ISSN 1538-4365.

[125] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is All you Need. In *Advances in neural information processing systems*, volume 30, page 11, California, 2017.

[126] J. Walker. Range-Doppler Imaging of Rotating Objects. *IEEE Transactions on Aerospace and Electronic Systems*, AES-16(1):23–52, January 1980. ISSN 0018-9251.

[127] J. Wei, Y. Xia, and Y. Zhang. M3Net: A multi-model, multi-size, and multi-view deep neural network for brain magnetic resonance image segmentation. *Pattern Recognition*, 91:366–378, July 2019. ISSN 00313203.

[128] C. Wetterer and M. Jah. Attitude Determination from Light Curves. *Journal of Guidance, Control, and Dynamics*, 32(5):1648–1651, September 2009. ISSN 0731-5090, 1533-3884.

[129] C. Wetterer, R. Linares, J. Crassidis, T. Kelecy, M. Ziebart, M. Jah, and P. Cefola. Refining Space Object Radiation Pressure Modeling with Bidirectional Reflectance Distribution Functions. *Journal of Guidance, Control, and Dynamics*, 37(1):185–196, January 2014. ISSN 0731-5090, 1533-3884.

[130] L. Windrim, A. Melkumyan, R. Murphy, A. Chlingaryan, and R. Ramakrishnan. Pretraining for Hyperspectral Convolutional Neural Network Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(5): 2798–2810, May 2018. ISSN 0196-2892, 1558-0644.

[131] F. Wokke, A. Kramer, R. van Benthem, R. A. Annes, T. Flohrer, T. Schildknecht, E. Stöveken, E. Valtonen, J. Peltonen, E. Riihonen, T. Eronen, and J. Kuusela. AN INSTRUMENT DESIGN FOR SPACE-BASED OPTICAL OBSERVATION OF SPACE DEBRIS. In *Proceedings of the International Academy of Astronautics Space Debris and Space Traffice Management Symposium*, page 8, October 2005.

[132] A. Woodward. A piece of space debris punched a tiny hole in the International Space Station, damaging a robotic arm. June 2021. URL `https://www.businessinsider.com.au/space-debris-hit-international-space-station-damaged-robotic-arm-2021-6?r=US&IR=T`.

[133] B. Wu, A. Wan, X. Yue, and K. 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)*, pages 1887–1893, Brisbane, QLD, May 2018. IEEE. ISBN 978-1-5386-3081-5.

[134] N. Wu, J. Phang, J. Park, Y. Shen, Z. Huang, M. Zorin, S. Jastrzebski, T. Fevry, J. Katsnelson, E. Kim, S. Wolfson, U. Parikh, S. Gaddam, L. Lin, K. Ho, J. Weinstein, B. Reig, Y. Gao, H. Toth, K. Pysarenko, A. Lewin, J. Lee, K. Airola, E. Mema, S. Chung, E. Hwang, N. Samreen, S. Kim, L. Heacock, L. Moy, K. Cho, and K. Geras. Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening. *IEEE Transactions on Medical Imaging*, 39(4):1184–1194, April 2020. ISSN 0278-0062, 1558-254X.

[135] X. Yang, Y. Pi, T. Liu, and H. Wang. Three-Dimensional Imaging of Space Debris With Space-Based Terahertz Radar. *IEEE Sensors Journal*, 18(3): 1063–1072, February 2018. ISSN 1530-437X, 1558-1748, 2379-9153.

[136] N. Zacharias, C. Finch, T. Girard, A. Henden, J. Bartlett, D. Monet, and M. Zacharias. THE FOURTH US NAVAL OBSERVATORY CCD ASTROGRAPH CATALOG (UCAC4). *The Astronomical Journal*, 145(2):44, January 2013. ISSN 0004-6256, 1538-3881.

[137] M. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8689, pages 818–833. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10589-5 978-3-319-10590-1. Series Title: Lecture Notes in Computer Science.

Appendix A

Additional Information for Blender Simulated Dataset

A.1 Rendered Imagery of Shape Models

This section contains a rendered image of the each of the shape models that was used to generated simulated light curve data for both Chapter 4 and Chapter 5. Note that these images are not to scale and objects appear in the same order as in the confusion matrix presented in Figure 4.6a.

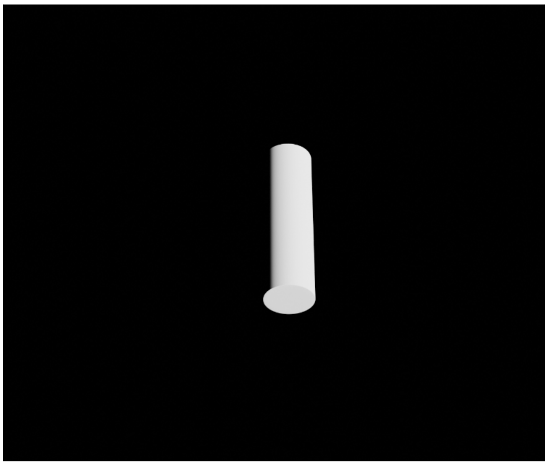


Figure A.1 – Cylinder Render
Same dimensions as the SpaceX stage 2 rocket body but no nozzle or nose cone

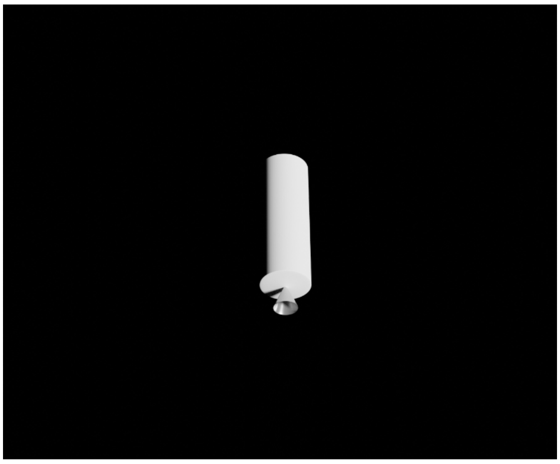


Figure A.2 – SpaceX Render
Designed based on the dimensions of SpaceX stage 2 rocket body

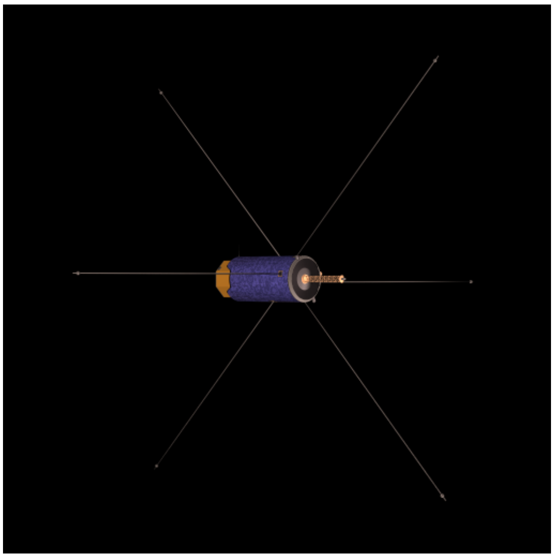


Figure A.3 – ASTRE Render

Cylindrical shaped object with antennas

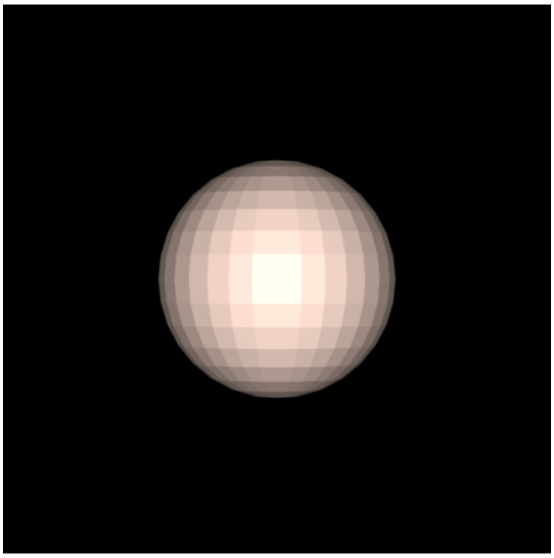


Figure A.4 – Sphere Render

Sphere with uniform material properties

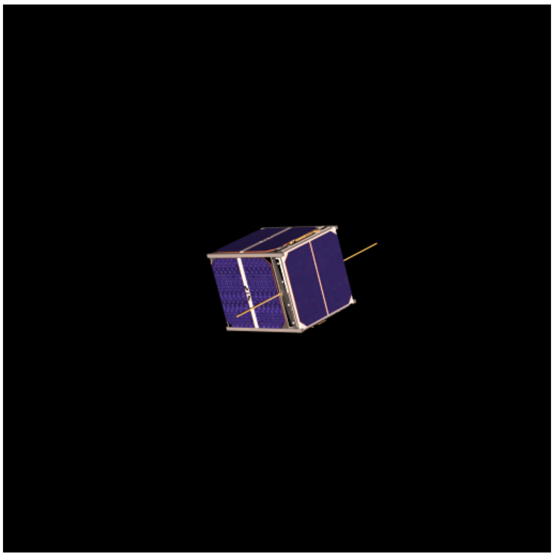


Figure A.5 – 1RU-CubeSat Render

1U-CubeSat (10cm x 10cm x 10cm)

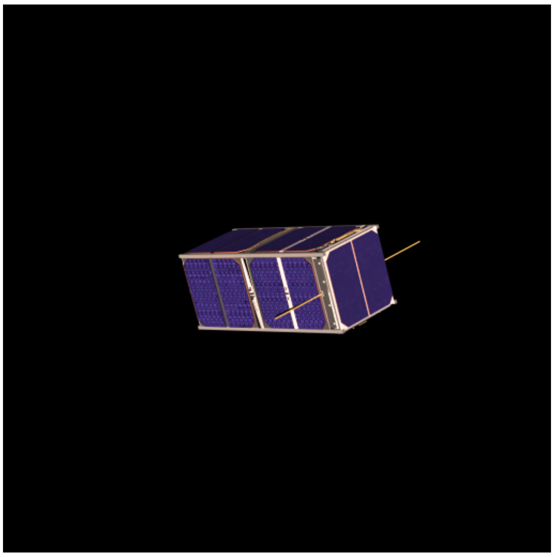


Figure A.6 – 2RU-CubeSat Render

2U-CubeSat (10cm x 10cm x 20cm)

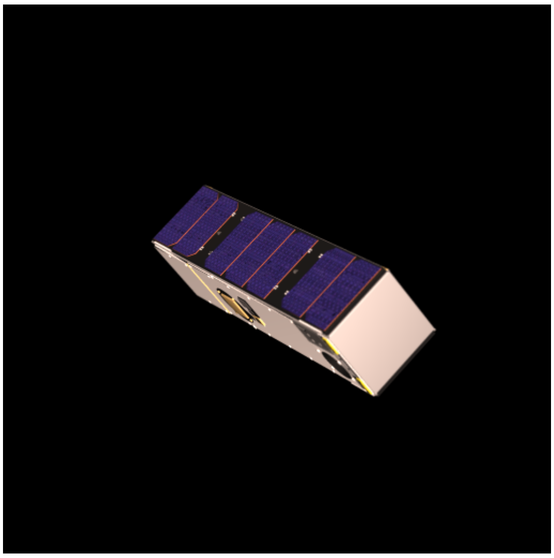


Figure A.7 – ICECube2016 Render

3U-CubeSat (10cm x 10cm x 30cm)

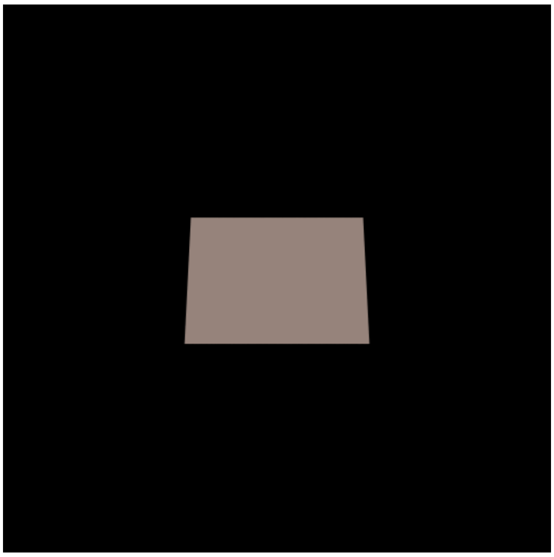


Figure A.8 – PlaneSingle Render

Flat plate with a single reflective side

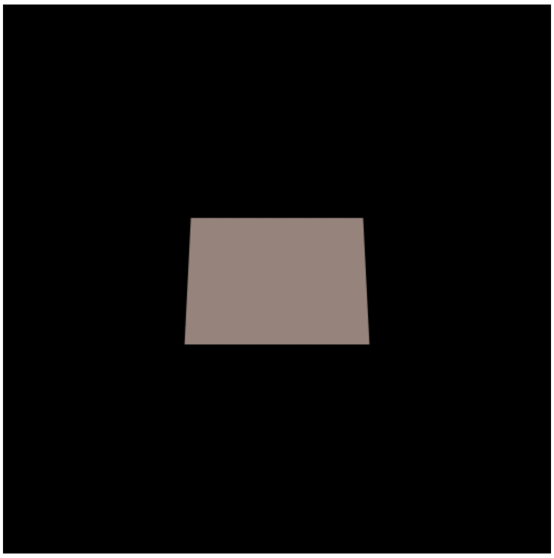


Figure A.9 – Plane Render

Flat plate identical to PlaneSingle however both sides are reflective

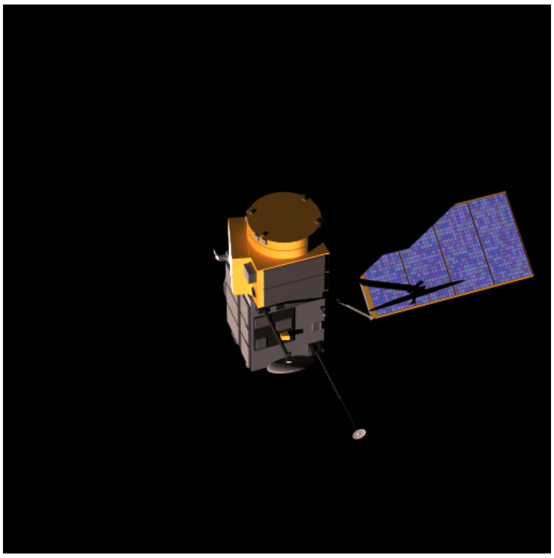


Figure A.10 – Topex Render

Single solar panel with antennas and a rectangular bus

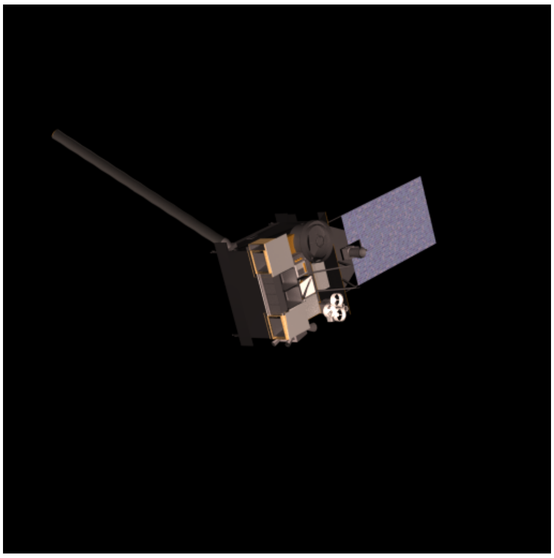


Figure A.11 – GOES-PQR Render
Single solar panel with rod-like antenna
and a rectangular bus

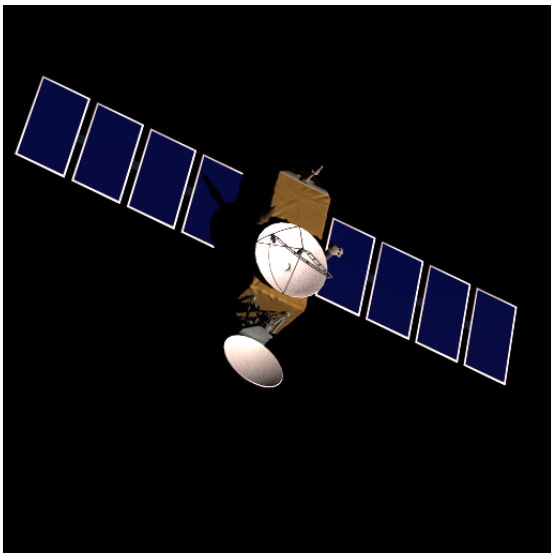


Figure A.12 – Jason1 Render
Dual wing satellite with a central bus
and multiple circular antennas

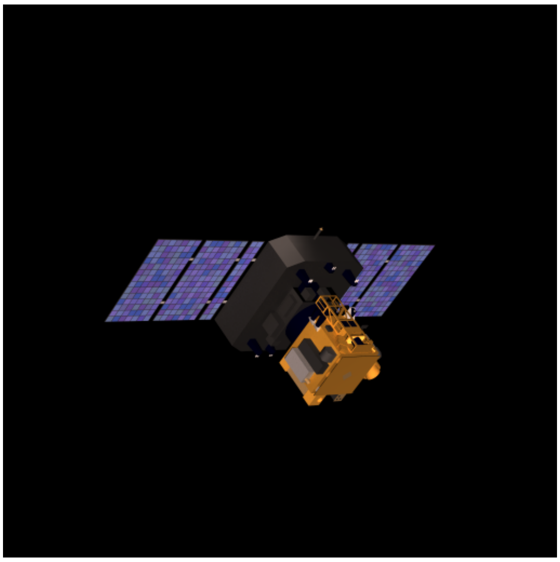
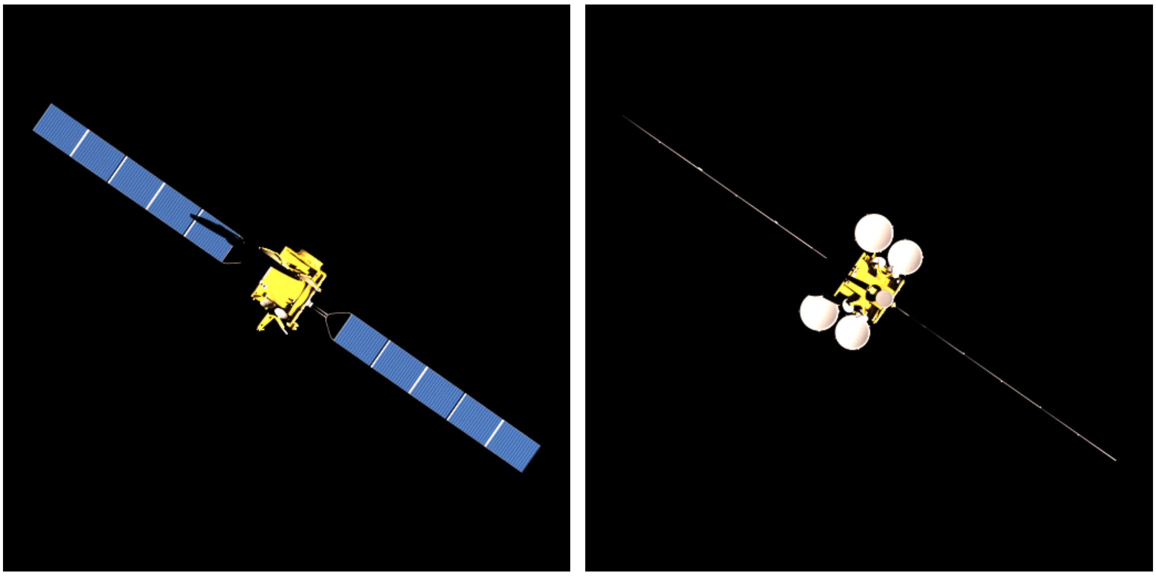


Figure A.13 – SHO Render
Dual wing satellite with central bus

A.2 Effect of Variations in Material Properties

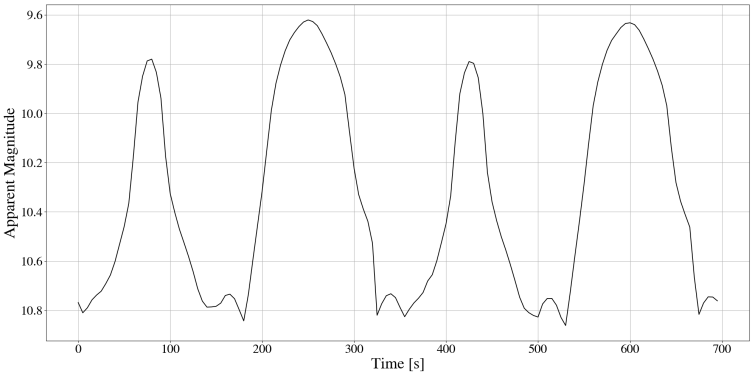
This section contains some initial results depicting the effect of changing various material properties in Blender on the shape of the output light curve. The experiments were performed using a model of the MEASAT 3 satellite, which has a box wing configuration with four large circular antennae. For these experiments, the angle of the solar panels were set to be perpendicular to the angle of the antennae. This can be seen in the rendered imagery presented in Figure A.14.



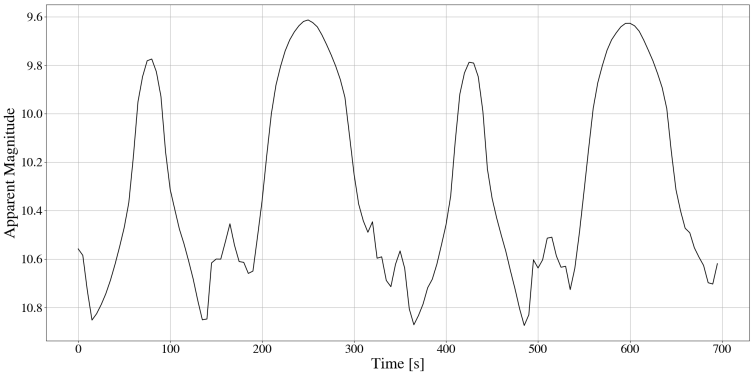
(a) Solar Panels Visible **(b)** Antennae Visible

Figure A.14 – Rendered imagery depicting the model of MEASAT-3 used in Blender

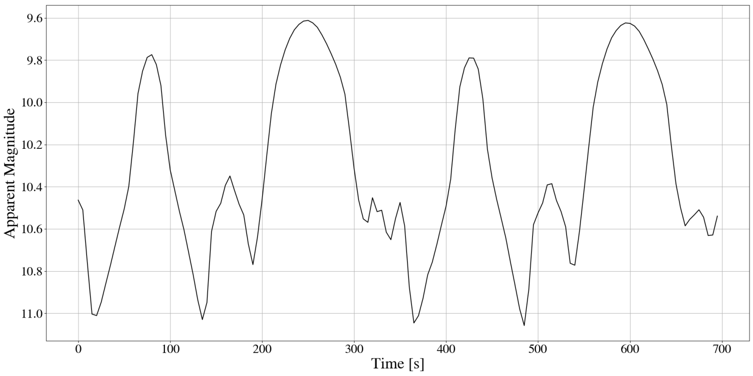
For the light curves presented in Figure A.15, only the reflectivity properties of the four antennae is varied. The sections of the light curve heavily influenced by the reflection from these antennae can be seen at approximately, 170s, 340s and 510s, with large variations seen at these points in the light curves. It should be noted that Figure A.15a and Figure A.15d depict the results for extreme cases, which are unlikely to correspond to the material properties for RSOs.



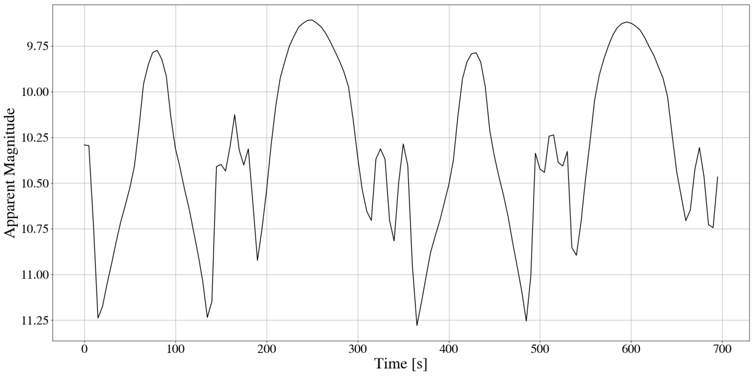
(a) Metallic: 0.0, Specular: 0.0, Roughness: 0.2



(b) Metallic: 0.25, Specular: 1.0, Roughness: 0.2



(c) Metallic: 0.75, Specular: 0.5, Roughness: 0.3



(d) Metallic: 1.0, Specular: 2.0, Roughness: 0.2

Figure A.15 – Comparison between extracted light curves for MEASAT 3 with variations in the antennae material properties