

DETECTION & IDENTIFICATION OF ON-ORBIT OBJECTS USING MACHINE LEARNING

M. Perez⁽¹⁾, M. A. Musallam⁽²⁾, P. L. Henaff⁽³⁾, A. Garcia⁽²⁾, E. Ghorbel⁽²⁾, K. A. Imsaeil⁽²⁾, and D. Aouada⁽²⁾

⁽¹⁾LMO, United Kingdom, Email: m.perez@lmo.space

⁽²⁾Interdisciplinary Center for Security, Reliability and Trust (SnT) - University of Luxembourg, Luxembourg, Email: mohamed.ali@uni.lu, albert.garcia@uni.lu, enjie.ghorbel@uni.lu, kassem.alismaeil@uni.lu and djamila.aouada@uni.lu

⁽³⁾MDA, United Kingdom, Email: paul.lehenaff@mda.space

ABSTRACT

Space Debris Surveillance and Tracking has become a necessity to ensure the safety of current and future space missions. In this work we present techniques for automatic Detection and Identification of Space Resident Objects (SRO) using Machine Learning models in the frame of Space Surveillance and Tracking (SST). A test facility was built in order to generate representative images for training the algorithms. We also present a description of the experimental setup used for the generation of training and validation images, the selected test cases, the used algorithms and the performance achieved. It is found that Machine Learning, including Deep Learning algorithms have the potential to be used for identification and classification tasks in space.

Keywords: Orbiting objects, Detection, Identification, Machine Learning, Deep Learning, Computer Vision.

1. INTRODUCTION

The ability of Machine Learning (ML), including Deep Learning (DL) methods to identify and classify satellites (and therefore distinguish them from non-satellite objects or anomalous behaving man-made objects) based on sky observations and catalogue information has been demonstrated in several studies, as in [1, 2, 3]. Recently, there has been a growing interest in the possibility of implementing a space-based surveillance and tracking system for monitoring objects directly in space [4]. Of special interest is the possibility of creating an observation network based on different commercial missions that have sensors available in space. Such systems should be able to extract as much insight as possible from low resolution images taken under very varying illumination conditions, and hence there is a motivation to develop techniques that can make this processing in space using the existing (or soon to be available) computing capacity. One candidate technique that promises to greatly improve the performance of these algorithms is the characterisation of the

satellite image using ML, and specifically DL techniques, in low-resolution images (as it happens when satellites are observed from a very far distance).

LMO was awarded a grant from the UKSA to develop techniques for Identification and Classification based on low-resolution satellite images taken in space. As part of the project, LMO in collaboration with the *Computer Vision, Imaging, and Machine Intelligence (CVI²)* research group at the SnT, University of Luxembourg, implemented several object detection and classification algorithms based on ML, including DL techniques, trained, and validated these models using experimental data generated using cameras in a representative setup. Technical inputs for the camera's implementation was provided by MDA-UK together with support in the implementation of the experimental test setup.

The goal of this work is to explore the feasibility of ML as well as DL approaches through off-the-shelf models to identify and classify satellites and space debris. To this end, an experimental setup has been built to generate a dataset of images covering a range of spacecrafts and space debris.

The following sections of this paper cover: algorithm selection in Section 2, experimental setup in Section 3, experimental images dataset in Section 4, synthetic images dataset in Section 5, algorithms and evaluations in Section 6, and finally results and discussion in Section 7.

2. ALGORITHM SELECTION

In order to develop a robust Space Surveillance and Tracking (SST) system capable of detecting anomalous behaviour of Space Resident Objects (SRO) and raising warnings to operators, a selection of three algorithms has been performed. Note that the proposed algorithms can be combined or used independently.

1. Satellite Identification & Classification algorithm with two core functionalities. Firstly, the algorithm

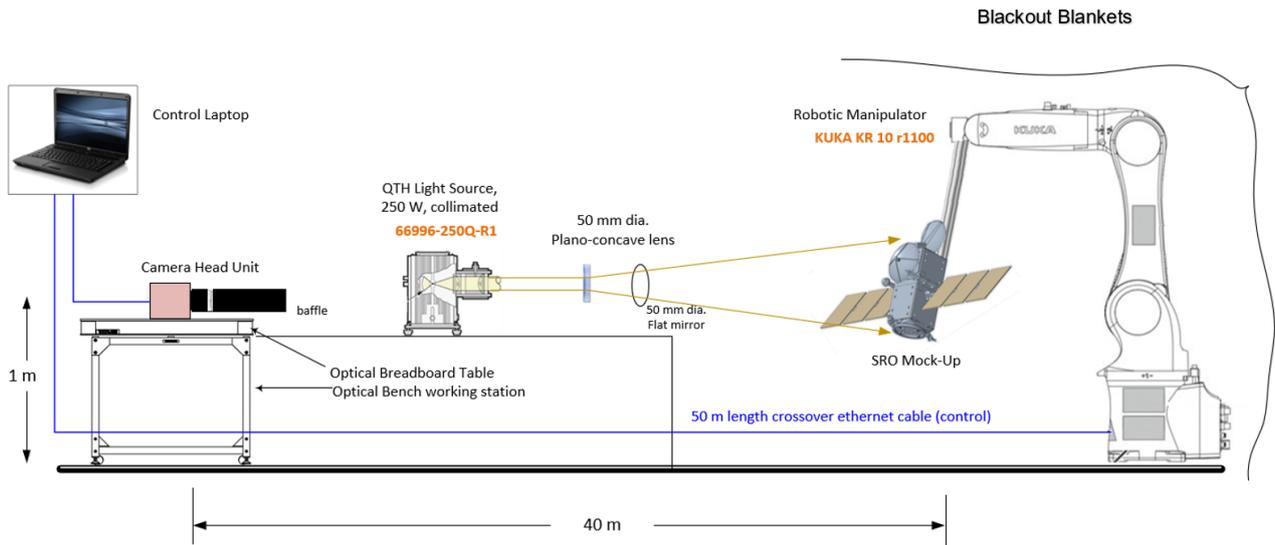


Figure 1. Schematic of the experimental setup used.

should be able to differentiate between satellites and non-satellites. In other words, the algorithm should be able to perform binary classification on whether the orbiting object is a satellite or not. Secondly, based on a predefined catalogue, the algorithm should be capable of classifying the specific model of the satellite being targeted. This means that the algorithm should label the satellite with one out of a total of N satellite classes defined in the catalogue. This task is also known as multi-class classification.

2. Satellite Stability Diagnosis algorithm. By estimating several attitude-related metrics, such as tumbling rate, pose variation, pointing of solar panels and more, an assessment on the stability of the satellite can be performed by means of comparison with historical data. In this sense, a change in pose or a change in rotation rate that significantly differs from historical data can point to a malfunction of the satellite. This scenario would be followed by the raise of a warning to the operators in order to take any required actions.
3. Material Identification algorithm. A grouping of satellite colour signatures, i.e., the ratio of Red (R), Green (G) and Blue (B) components, can be used with unsupervised learning techniques and/or classifiers to identify the satellite being imaged and derive further information. It can be said that such algorithms would be relying on the material composition of the satellites.

This work centers its attention into an exploration of Machine Learning (ML) approaches in order to develop algorithm number 1, this is, Satellite Identification & Classification. To this end, an exploration process from *simple to complex* methods is followed in order to test and understand how different Machine Learning classification

models perform. The final outcomes of this exploration process is an in-depth analysis on why one model is better than another. The explored methods can be categorized into two main groups: (1) classical ML algorithms and (2) Deep Learning (DL) algorithms based on Deep Neural Networks architectures. A brief summary of the selected approaches is presented here.

- Support Vector Machine (SVM) [5]: uses raw pixel information in the form of a flattened image array for both binary and multi-class classification.
- Multinomial Softmax Regression (MSR) [6]: uses raw pixel information in the form of a flattened image array for multi-class classification by means of regression optimization.
- Principal Component Analysis (PCA) [7]: performs dimensionality reduction of the flattened image (not including colour signature maps), down to 10 features, which are further used for classification.
- Cubic Feature Vectors: uses a specific set of features based on PCA to further classify the satellite.
- Deep Neural Networks (ResNet) [8]: a specialized Deep Neural Network (DNN) architecture, which is a state of art very deep Convolutional Neural Network (CNN), is used to extract class-related features that are further used to classify the satellite.

SVM, MSR, PCA and Cubic Feature Vectors belong to the first category, i.e., *classical ML* algorithms. Meanwhile, the latter approach, ResNet, belongs to the second category of DL. It is common to confuse the two main categories previously presented due to the fact that DL algorithms are regarded as a sub-category of ML algorithms. For this reason, it is worth noting that the first category (Classical ML algorithms) specifically refers to

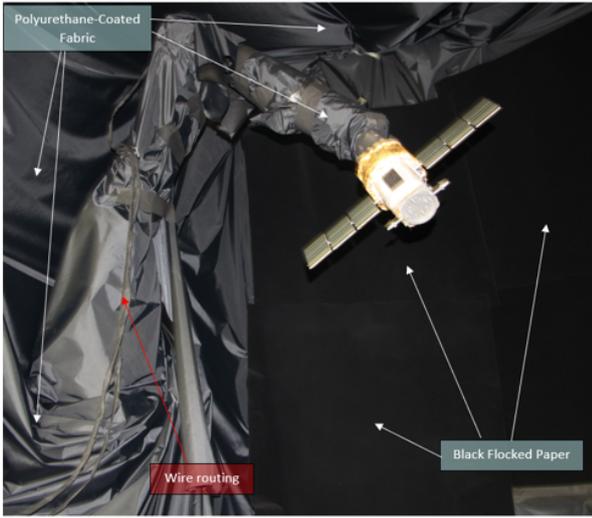


Figure 2. Final implementation of the mock-up stand connected to the 6-degrees of freedom robotic arm.

approaches that are based on fitting and optimization processes without the use of Neural Network architectures. A detailed description of each one of these classification approaches is offered in Section 6.

3. EXPERIMENTAL SETUP

To validate the proof-of-concept and verify by test the initial performances of the selected algorithms, an experimental setup was designed and implemented in collaboration with MDA-UK with inputs from CVI² related to image datasets for training. The experimental setup design (shown in Figure 1) was implemented in a mid-range underground tunnel that allowed for testing at different ranges in order to get images at different resolutions and under varying illumination conditions. 8 different ranges starting from 35m distance and down to approximately 2m distance were used for the images. Different targets were designed and built representing mock-ups of different known satellite buses (Calypso, Cloudsat and Jason-3) as well as debris objects (separation ring and fairing-like objects), sized such as to achieve the desired small pixel resolution, and using different combinations of known materials. Combining the cameras with the tunnel range this allowed to simulate images of medium sized objects at different distances in the order of 50 km. These mock-ups were installed on a 6-degrees of freedom robotic arm provided by MDA-UK to control their orientations. The whole setup was covered on light absorbing material to minimize undesirable light sources, and the mock-ups had heaters installed to simulate different thermal configurations. An incoherent, collimated, power-regulated light source was used to simulate lighting with an intensity similar to that produced by the sun in space (in the visible range). The final implementation is shown in Figure 2 and Figure 3.

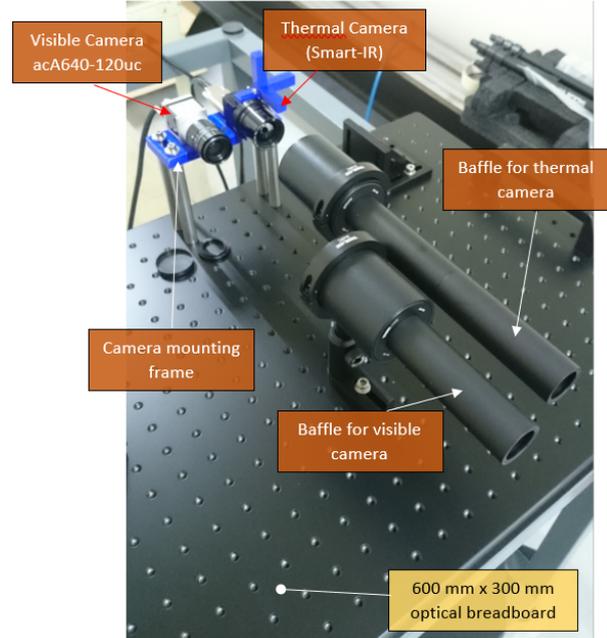


Figure 3. Final implementation of the camera setup with visible and infrared cameras.

4. EXPERIMENTAL IMAGES DATASET

A set of 60.460 experimental images for training algorithms in the Identification and Classification of objects in space was generated in both the visible range and thermal infrared range. This dataset contains 4 classes: three types of satellites (Calypso, Cloudsat and Jason-3) and a separate class *debris*. The dataset was generated using varying illumination angles and a variety of orientations within a limited range of $\pm 30^\circ$ rotation around each axis, as well as 8 different ranges. The limiting factor for the range of rotations was the robotic arm capacity to move the satellite around a fixed point in the constrained space available in the tunnel (height and width). The data has the following labels: unique index, relative positions (x, y and z) from the camera's reference frame to the satellite body reference frame, relative rotations (around x, y and z) of the satellite body reference frame with respect to the cameras reference frame, expressed as a result of consecutive rotations, a flag indicating if it is a satellite or a debris object and a label indicating which satellite bus it is. A set of 9.114 images for training algorithms in anomaly detection based on satellite attitude was generated. This dataset simulates historical observations of a target over half a year including changes in the sun phase angle. This dataset (collected in visual and thermal images) will allow for the development of Algorithms 2 and 3 in the future. Examples of the resulting images are shown in Figure 4 (close range and far range images).

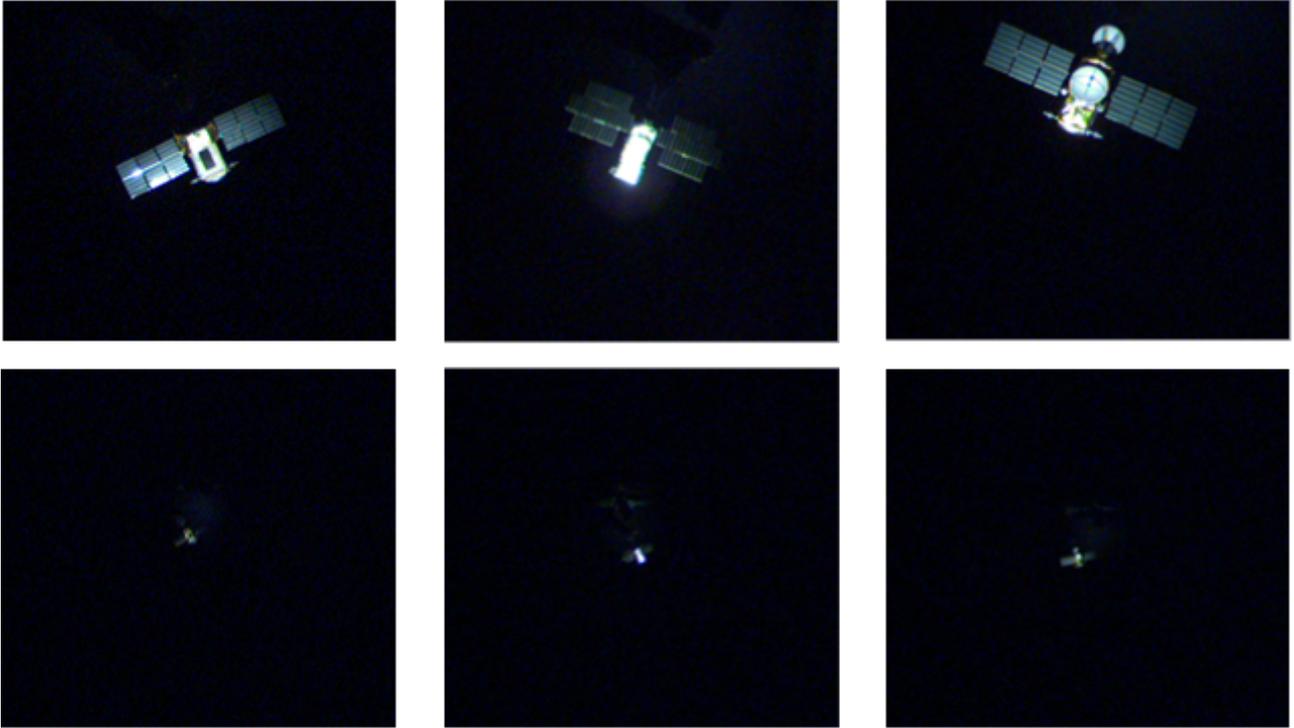


Figure 4. Sample images captured with in the experimental laboratory setup. From left to right columns: Calipso, CloudSat and Jason-3 spacecrafts. From top to bottom rows: close range and far range images.

5. SYNTHETIC IMAGES DATASET

In addition to the laboratory dataset generated through the setup described in Section 3 and presented in Section 4, a synthetic dataset has been generated. The synthetic dataset has been developed thanks to the graphics engine Unity in combination with 3D CAD models of the same space objects presented in the laboratory dataset, namely, Calipso, CloudSat and Jason-3 satellites as well as 2 different debris objects (all of them regarded as belonging to the same class *debris*). The main advantages of generating synthetic data are: (1) full control in the data variability, meaning that all generation-related parameters can be fully regulated on-demand such as lightning conditions and earth on the background, (2) an *unlimited* amount of data can be generated (until all data variability is covered) and (3) additional imagery techniques can be added to the generation process such as blurring, saturation variation, brightness variation and many more. In relation to the latter advantage, our synthetic dataset presents Gaussian blurring which intends to mimic the noise introduced by the camera used during the generation of the laboratory dataset. One critical disadvantage of solely relying on synthetic data is the effect known as *domain gap*. A significant accuracy drop generally appears when a model is trained on purely synthetic data and then evaluated on real data (thus the name). This is a well-known and well-studied behaviour among the learning-based academia, and thus several solutions tackling this accuracy drop have been proposed. Since developing a *domain gap* solution for space data falls out of

the range of this work, we make use of one of the simplest yet effective *domain gap* generic solutions, that is, combining the synthetic data with the laboratory data in hopes of adapting the trained model to the latter type of data.

Due to the nature of the task, i.e., orbiting space objects classification, a wide range of data variability appears introduced by the object's pose (position and orientation), the lightning conditions and the earth on the background. Following the conditions of the laboratory dataset, the here generated dataset does not present earth on the background. Moreover, the dataset generated and used for this work is a reduced version of the one presented in the 2021 SPARK challenge standing for *SPAcecraft Recognition leveraging Knowledge of Space Environment* [9]. The dataset generated presents roughly 3.500 images for each satellite class while the *debris* class presents a total of 7.000 images. All the images have been generated in colour (with RGB channels) and with a size of 224×224 pixels. Several samples of the generated images can be seen in Figure 5.

6. ALGORITHMS AND EVALUATIONS

Having generated the laboratory as well as the synthetic datasets, all the proposed approaches for space objects classification are optimized and evaluated. The classical ML-based algorithms (SVM, MSR, PCA and Cubic Feature Vectors) have all been implemented by LMO and op-

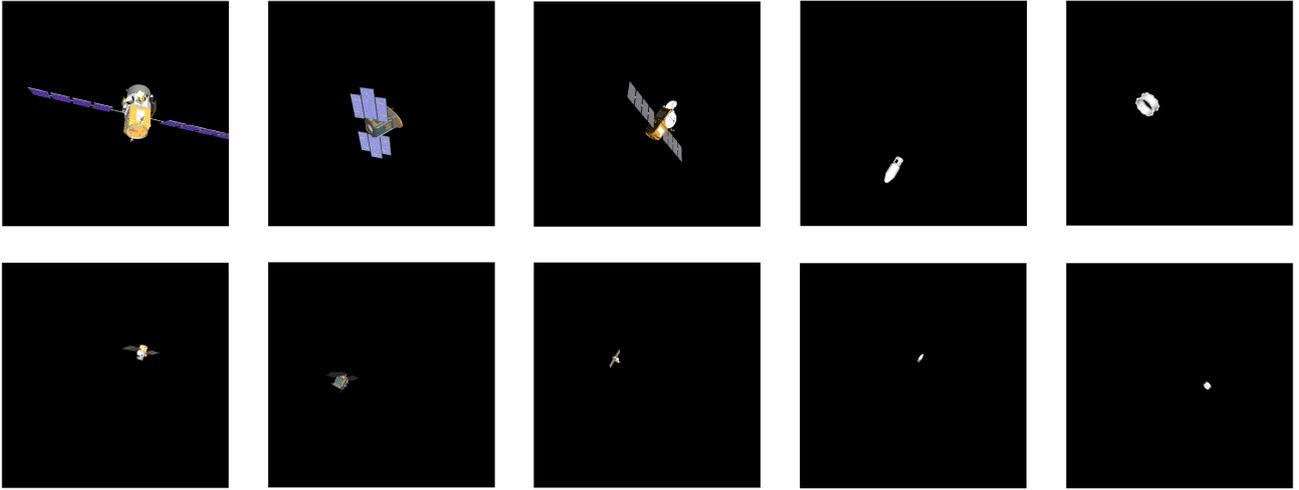


Figure 5. Sample images from the generated synthetic dataset. From left to right columns: Calipso, CloudSat, Jason-3, space debris (1) and space debris (2). From top to bottom rows: close range and long range.

timized using a greyscale version of long range-only laboratory images. This has been done in order to be able to train, validate and test datasets using the minimum computing power possible. Additionally, the laboratory images have been cropped to a size of 50×50 at the center of the space object depicted in each image. This cropping step mimics a detection stage that needs to be performed when relying on purely ML-based solutions since these are in general position-sensitive with respect to the pixels in the image. Experimental exploration has shown that a split of 10% training data and 90% test data is sufficient and challenging enough for achieving high classification results. This split has been performed at random based on a fixed seed in order to ensure the same test set over all classical ML-based approaches.

On the other hand, the DL-based approach, this is through DNNs and more specifically through CNNs, has been implemented by CVI² within SnT (University of Luxembourg). CNN models have two significant advantages by nature with respect to classical ML-based approaches; (1) such models are position-insensitive over the pixels meaning that for classification purposes there is no need for a prior detection step and (2) using synthetic data as a complement to real data has been experimentally found to enhance the final results. For these reasons, both synthetic and laboratory datasets have been used on their full resolution as well as on their RGB format. Despite these great advantages, DL-based models are known to require large amounts of data to be properly trained. Thus, a split of 50% training data and 50% test data with a fixed random seed is used throughout all the experiments with CNNs.

Following subsections provide explanations on each one of the explored classification models as well as on their performances over their corresponding laboratory test splits.

6.1. Support Vector Machines

Support Vector Machines can be used to perform binary classification through fitting a hyperplane, which splits the two classes of the training set, over the input space \mathbb{R}^d with d the number of dimensions of the input vector. In our case we need to tackle a multi-class classification problem. In order to handle this with SVM, a one-vs-all (also known as one-vs-rest) classification strategy has been followed. Since our dataset presents a total of 4 different classes, a total of 4 different binary SVM classifiers are needed. The one-vs-all strategy works as follows:

1. For each class c in the dataset train an SVM binary classifier considering class c as the positive class (label 1) and all the other classes as the negative class (label 0). This training stage yields a total of 4 SVM classifiers that are able to identify if an input contains a specific space object or not.
2. During evaluation, the input flattened image is processed by all 4 SVM classifiers and the overall winner (based on the 4 yielded scores) decides the class to be predicted.

The specific SVM model selected has been a Linear SVM classification model with a regularization parameter C of 0.1. A total of 5 runs were performed in order to measure the classification accuracy. The average classification accuracy has been found to be **98.9 %**.

6.2. Multinomial Softmax Regression

For each input array x of dimension d , a Multinomial Softmax Regression model computes the probability that x belongs to the class c , for $c = 0, \dots, k - 1$ (in our case $k = 4$), using Eq. 1.

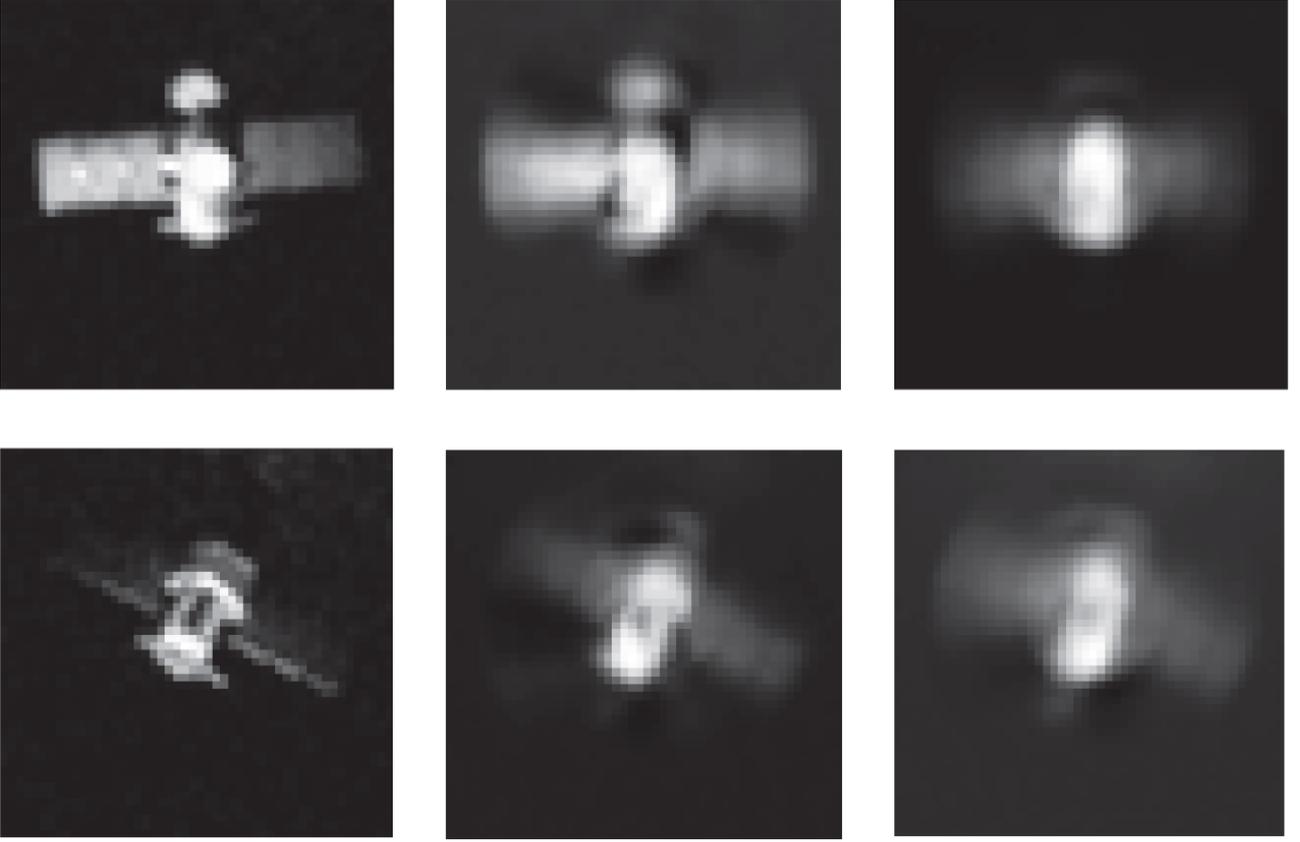


Figure 6. From left to right columns: original image, reconstruction from PCA with $N = 10$ and reconstruction from PCA with $N = 2$. From top to bottom rows: Jason-3 sample and Calipso sample.

$$h(x) = \frac{1}{\sum_{c=0}^{k-1} e^{\theta_c \cdot x / \tau}} \begin{bmatrix} e^{\theta_0 \cdot x / \tau} \\ e^{\theta_1 \cdot x / \tau} \\ \vdots \\ e^{\theta_{k-1} \cdot x / \tau} \end{bmatrix}. \quad (1)$$

For each c , θ_c is an array of dimension d with the classifier parameters corresponding to the class c . Moreover, in Eq. 1, \cdot represents the dot product and τ is the temperature parameter (a positive scalar) where small values lead to less variance. In order to learn the parameters θ_c the following loss function (or cost function) presented in Eq. 2 is minimized.

$$L(X, \theta) = -\frac{1}{n} \left[\sum_{i=1}^n \sum_{c=0}^{k-1} \mathbb{1}_{y^{(i)}=c} \ln \frac{e^{\theta_c \cdot X^{(i)} / \tau}}{\sum_{j=0}^{k-1} e^{\theta_j \cdot X^{(i)} / \tau}} \right] + \frac{\lambda}{2} \sum_{c=0}^{k-1} \sum_{l=0}^{d-1} \theta_{cl}^2, \quad (2)$$

where n is the total number of samples of the training set, k is the number of classes (4 in our case), d is the number

of dimensions of the input arrays (can also be seen as the number of input features) which in our case is equal to $50 \times 50 = 2,500$, $\mathbb{1}$ is the indicator function, X is the training set, y is the list of ground-truth class labels, and λ is the regularization parameter for the parameters θ . This loss function has been minimized by the Stochastic Gradient Descent (SGD) method as presented in Eq. 3.

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} L(X, \theta), \quad (3)$$

where α is the learning step parameter, ∇_{θ} is the gradient operator with respect to θ and \leftarrow is the update function. The final MSR model has been trained using $\alpha = 0.3$, $\lambda = 1e-4$ and for a total of 150 iterations (epochs). The trained model has achieved a classification test accuracy of **98.4 %**.

6.3. Principal Component Analysis

The two previously described models take as input the flattened greyscale cropped image. Taking this approach means processing as many dimensions as features in the input image. Moreover, due to the design of these ML-based models, the number of parameters grows with the

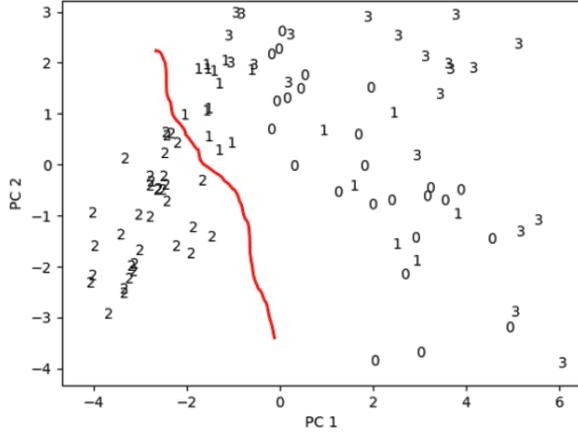


Figure 7. Plot of the first 100 2-dimensional PCA representations of the laboratory dataset. Note how class number 2, corresponding to the space debris class, can easily be separated from the rest of the classes (red separation has been manually plotted to show this).

number of input features. In hopes of reducing the number of parameters needed as well as using more compact and meaningful features PCA is used. After transforming the input arrays through PCA the MSR model is retrained and re-evaluated to see the performance impact.

PCA is an extremely popular technique among the data processing community which performs dimensionality reduction while encoding significant features. This method finds orthogonal directions of maximal variation in a dataset. Once these directions have been found then the dataset can be projected onto a subset of these directions thus yielding a reduced representation. The sorted directions, from higher to lower data variation, correspond to the eigenvectors sorted by the eigenvalues of covariance matrix $\hat{X}\hat{X}^T$, where \hat{X} is the zero-mean dataset X and \cdot^T is the matrix transpose operation. It should be noted that the orthogonal directions are found over the training set and then these are used over the test set. The reason for this is that the test data should not influence in any way the PCA process since it is considered to be completely unseen data.

PCA lets the developer choose the number of dimensions to be reduced by choosing how many dimensions should be output have. This is done by selecting and projecting all the data to the top N most variant directions, where N is a parameter to be fixed by the developer. We have explored two different scenarios, N equal to 2 and equal to 10. When $N = 2$ the yielded representations can be depicted in a 2-dimensional scatter plot. Figure 7 presents these 2-dimensional PCA representations for the first 100 images of the laboratory dataset. The retrained MSR model using PCA with $N = 2$ yields a classification accuracy of 68.8%. On the other hand, the retrained MSR model using PCA with $N = 10$ yields an accuracy of 97.3%. This behaviour is to be expected as with $N = 2$ only two features are being used to rep-

resent the space object in comparison with 10 features. To better understand this difference in the amount of information compressed by PCA, a reconstruction process is performed from the yielded representations. Figure 6 shows a comparison of two sample images, their corresponding reconstructions from PCA with $N = 10$ as well as their reconstructions from PCA with $N = 2$.

6.4. Cubic Feature Vectors

An extension to the PCA dimensionality reduction process has been also explored by performing a cubic feature mapping. This mapping can be defined as a function ϕ that maps an input feature vector x with d dimensions into a new feature vector $\phi(x)$ such that, for any other d -dimensional array x' , Eq. 4 holds.

$$\phi(x)^T \phi(x') = (x^T x' + 1)^3. \quad (4)$$

Such cubic mapping represents the new input space of the dot product defined in Eq. 4. This dot product is a specific case of a degree-3 polynomial kernel which can be used in combination with PCA. Thus, combining PCA with the here defined kernel, new cubic feature vectors are generated. The main advantage of using a polynomial kernel is that it enables a classification model, such as Linear SVM or MSR, to perform non-linear classification. Finally, an MSR model has been trained with these cubic feature vectors yielding an outstanding classification accuracy of 99.8%.

6.5. Deep Neural Networks

A series of trainings and evaluations have been performed in order to shed light on the usage of CNN, a specialized category of Neural Network architectures for image processing, on the task of space objects classification. To this end, an extremely popular and widely used off-the-shelf CNN has been selected named ResNet [8]. Such architecture has a track proven record over a vast range of different computer vision tasks such as classification. In our series of experiments we have trained and evaluated different sizes of the ResNet architecture, in particular ResNet18, ResNet34 and ResNet50. Additionally, different combinations of ranges from the camera to the space object have been considered on the train and test datasets. Regarding the synthetically generated dataset, ResNet18 has been used to perform an exploration and comparison when using such data as a complement to the laboratory dataset. Note that the test sets are fully composed by laboratory data, meaning that synthetic data is only used during training. The approach followed to exploit the synthetic data is as follows:

1. First, a training step is performed over pure synthetic data. A train/test split is also used over the synthetic

Table 1. Classification accuracy results for different CNN architectures under different training conditions.

CNN Architecture	Training data	Ranges trained - tested	Train accuracy (%)	Test accuracy (%)
ResNet18	Synthetic	Mixed - Mixed	99.13	99.43
ResNet18	Synthetic + Lab	Mixed - Mixed	99.85	99.96
ResNet18	Synthetic + Lab	Long - Long	99.44	99.75
ResNet34	Lab	Long - Mixed	99.60	86.20
ResNet34	Lab	Mixed - Long	99.02	98.00
ResNet50	Lab	Mixed - Mixed	99.80	99.50
ResNet50	Lab	Long - Long	99.20	99.33



Figure 8. Integrated Gradients qualitative analysis. The top row presents a spacecraft sample image (on the left) with its corresponding pixel attributions (on the right). The bottom row presents the same scenario for a space debris image.

dataset in order to keep track of the best performing iteration.

2. Once the CNN has been optimized with the synthetic dataset it is used as the starting model for further training. The second, and final, training stage only involves laboratory data. In doing this, all the information and knowledge learned during the synthetic training stage is transferred.

Such technique is expected to enhance the final classification results over the laboratory test set. Table 1 shows all the results obtained from all the experiments performed with the ResNet architecture. From the here obtained results several insights can be extracted. First of all, adding

synthetic data into the training process has been proven to enhance the final classification results as expected. Secondly, it can be derived that the task of classifying 3 spacecrafts and a generic space debris class appears to be well-behaved and fair to learn using CNN architectures. Nonetheless, it is worth noting that the trained models only show such high performance when the evaluated images are visually consistent with the images used during training. As found on the obtained results with the ResNet34 architecture, a model trained on long-range images only will drop its performance when evaluating over mixed-ranges images, and vice versa. Lastly, ResNet50, which corresponds to the biggest CNN architecture and the one which should presumably outperform its smaller versions, does not surpass ResNet18 when this one combines synthetic data with laboratory data. This remarkable finding shows how synthetic data can be extremely beneficial for the development of such DL-based models. Thus, it can be concluded that combining synthetic and laboratory data for space imagery tasks appears to be a greatly profitable technique.

In addition to the here presented classification results, a fast model interpretation assessment has also been carried. The model interpretation analysis performed has used the technique known as *Integrated Gradients* [10]. Such interpretation technique assigns an importance score to each pixel of the input image based on its impact on the final prediction. A qualitative interpretation analysis has been done with two different models, one trained on far range only and another trained on mixed ranges. Figure 8 shows two samples with their corresponding pixel attributions using the far range model. Visual inspection shows how the model generally focuses on the main body of the spacecrafts while for space debris their geometric shape is what is used for recognition. Solar panels appear not to be highly relevant for classification except in the case of CloudSat in which their solar panels have a different geometric shape. Furthermore, the model trained on long range was also tested on a close range image of a Jason-3 spacecraft. Figure 9 presents this image along its pixel attributes and the yielded classes probabilities. It can be noted how, in this specific Jason-3 image, the model was not able to find the same visual features to the ones with which it was

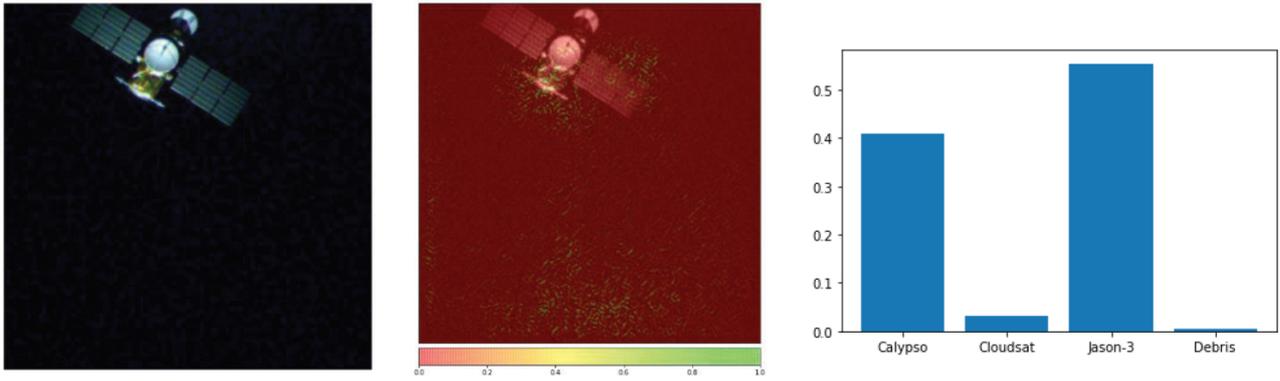


Figure 9. Cross-range Integrated Gradients qualitative analysis on a Jason-3 spacecraft image. From left to right: the input image, its corresponding pixel attributions and the predicted probabilities.

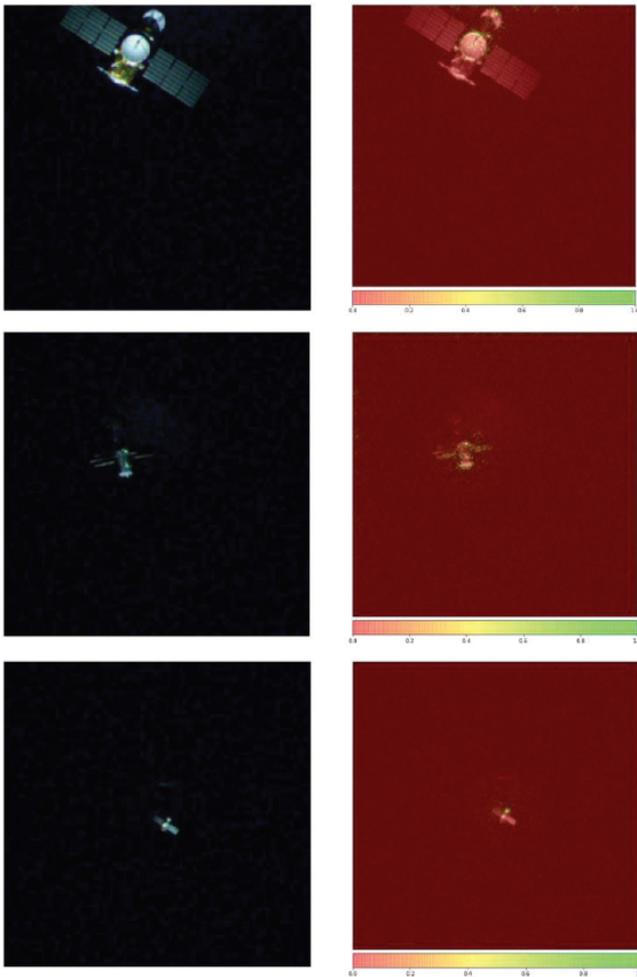


Figure 10. Integrated Gradients qualitative analysis over mixed ranges on a Jason-3 spacecraft. From top to bottom rows: close range, mid range and close range images. From left to right columns: input image and its corresponding pixel attributions.

Table 2. Summary of the best results achieved by each methodology over far range only images.

Methodology	Accuracy achieved (%)
SVM	98.9
MSR	98.40
PCA	97.3
Cubic Feature Vectors	99.8
Deep Neural Networks	99.3

trained. Also note how the focused attributes (green colored pixels) are perceptibly more scattered than in previous cases as seen in 8. For these reasons, the model confuses this Jason-3 spacecraft with a Calypso spacecraft as seen in the predicted probabilities. To conclude, Figure 10 presents one far, one mid and one close range image of a Jason-3 spacecraft using the mixed ranges model. Note how in this case the attributed pixels are better distributed over the body of the spacecraft in all ranges.

7. RESULTS AND DISCUSSION

All the best results, grouped by each methodology and obtained on far range only images, are summarized in Table 2. Classical Machine Learning-based methods have proven to consistently achieve high accuracy, even when using a split of 10%/90% between training and testing samples. Despite their great performances, ML-based methods, such as SVM and MSR, are position-sensitive with respect to pixel values. This means that when translating a set of pixels to another position of the image then the performance will greatly be affected by this. It is worth noting that, when performing dimensionality reduction, through PCA for instance, pixel values are combined into new features which can encode geometric information. It can be of interest, as future work, to evaluate the training and testing outcomes when using mixed ranges with ML-based algorithms. A solution to this

position-related drawback is to include a prior algorithm that detects and localizes the object in the image. On the other hand, Convolutional Neural Networks do not present this limitation as they are, by design, position-insensitive (due to the nature of the convolution operation). CNNs are the most robust option for the here studied application. The reason for this lies in the great visual processing capabilities that they have for extracting distinctive visual features of the depicted space objects. Moreover, CNNs are capable of handling a wide range of visual conditions, if properly trained, such as lighting variations and different distances to the object.

All algorithms have performed comfortably over the dataset generated from the experimental setup. It is to be noted that this dataset, even if representative of illumination conditions in space, has some limitations: (1) all space objects are imaged against a dark uniform background and (2) there is low variability on the data in terms of size (range) and orientation. Despite this, these conditions are likely to be representative of many surveillance scenarios when satellites may be observed in orbits such that the earth does not appear in the background, and at the same time the nominal operation of the satellites usually imply that there is a limited set of orientations expected at a certain point in their orbit (e.g., slow slew rate pointing to earth for most Earth Observation missions). Based on this, it is concluded that these results are successful as a first proof-of-concept demonstrating the potential of ML and DL algorithms to perform recognition and classification in space.

In hopes of further motivating the research community towards exploring ML methods including DL for space applications, CVI² in collaboration with LMO has organized and launched a challenge titled *SPAcecraft Recognition leveraging Knowledge of Space Environment* (SPARK) [9].

REFERENCES

1. Lane B., Poole M., Murray-Krezan J., Using Machine Learning for Advanced Anomaly Detection and Classification, AMOS Technical Conference, 2016.
2. Murray Krezan J., Poole M., Autonomous Object Characterization with Large Datasets, AMOS Technical Conference, 2015.
3. Muratov L., Perkins T., Fox M., Use of AI for Satellite Model Determination from Low Resolution 2D Images, AMOS Technical Conference, 2019.
4. Olmos D., Roda F., Middleton K., Naudet J., Space-Based Space Surveillance Operational and Demonstration Missions.
5. Cortes C., Vapnik V., Support-vector networks. *Machine learning*. 1995 Sep;20(3):273-97.
6. Böhning, D., Multinomial logistic regression algorithm. *Ann Inst Stat Math* 44, 197–200 (1992).
7. Pearson K. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh,*
- and *Dublin Philosophical Magazine and Journal of Science*. 1901 Nov 1;2(11):559-72.
8. He K., Zhang X., Ren S., Sun J., Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition 2016* (pp. 770-778).
9. IEEE ICIP 2021 SPARK challenge. <https://cvi2.uni.lu/spark-2021/> Accessed: 2021-04-07.
10. Sundararajan M., Taly A., Yan Q., Axiomatic attribution for deep networks. In *International Conference on Machine Learning 2017 Jul 17* (pp. 3319-3328). PMLR.