17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID #320

Space debris streak classification: a transparent deep learning approach to reduce false positive detections

Evridiki Ntagiou[a]*, Luigi Palladino[c], Jan Siminski[a], Gianluca Furano[b]

[a] *European Space Agency, ESOC, Robert-Bosch-Str. 5, 64293, Darmstadt, Germany, evridiki.ntagiou@esa.int, jan.siminski@esa.int*
[b] *European Space Agency, ESTEC, Keplerlaan, 1, 2201AZ, Noordwijk, The Netherlands, gianluca.furano@esa.int*
[c] *Department of Computer Science and Engineering, Università degli studi di Verona, Str. le Grazie, 15, 37134 Verona, Italy, palladino.luigi@outlook.com*
* Corresponding Author

Abstract

Space debris is observed from ground-based and space-based telescopes. Typically, faint streak-like features are generated in the resulting images due to the small size of the objects and relative observation geometry. The detection of low-SNR streaks poses a major challenge to state-of-the-art tools. In case of space-based telescopes on-board pre-processing can greatly reduce the data needs, but requires efficient algorithms to allow real-time selection of relevant images. Depending on the configured sensitivity in processing tools, they can generate a large number of false positive detections. The false streaks are caused by artifacts generated by different factors like oversaturation caused by bright stars, bad pixels, or cosmic rays. In this paper, we propose a method to greatly reduce the number of detected false positives. This method is based on exploiting data collected in the ESA's STREET database using modern data-driven techniques. We proposed a lightweight Deep Neural Network (DNN) approach based on the employment of a classical Multi-Layer Perceptron (MLP) on categorical features extracted from the STREET database. The proposed DNN uses manually obtained ground truth from STREET database to learn how to classify detected pixels between real streaks and false positives (artifacts). The aim of this work is to integrate the false positive rejection feature in the already existing streaks detection pipeline while maintaining the same level of transparency of the non-data-driven legacy system, despite the inclusion of a black-box element. This was possible by employing explainability techniques. The model achieves an accuracy of 92% in false positive detection, and the trustworthiness of these results is verified using both classical cross-validation techniques and by proposing modern criteria based on AI-explainability models such as Kernel Explainer from SHAP library. Through interpretation of Shapley values obtained by simulations, we were able to quantify the magnitude of the contribution of each feature in the model. Shapley values are interpreted both qualitatively and quantitatively with comparison to both manual procedure and statistical relevance of every feature used in the model. The results are very promising, hence this solution could be an example to increase the level of trust of using machine learning models in critical scenarios like space safety.

Keywords: Machine Learning, Explainable-AI, Space Debris, Streak detection

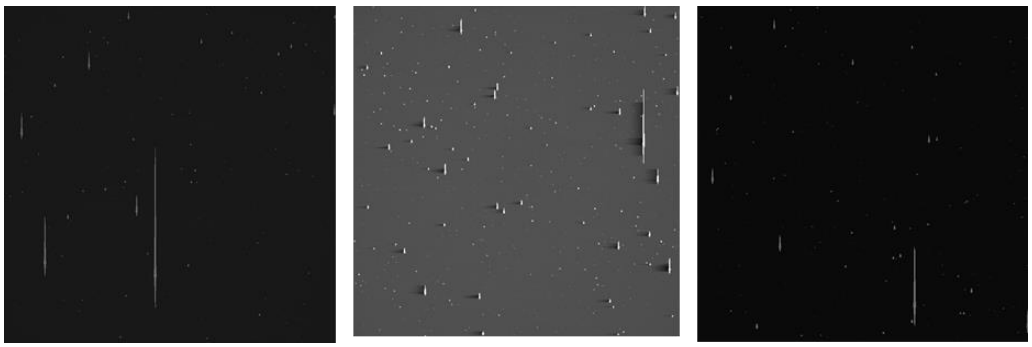

Fig. 1- Example of optical observations containing space debris streaks of ANTSY1 observatory from STREET Database.

1. Introduction

Streaks' detection is a common task in Space Surveillance and Tracking (SST) field when observing very small objects like space debris or when using very large Field of View (FOV) sensors (Fig.1). Modern systems like the one that generate images for ESA STREaks Evaluation Tool (STREET) [1] can handle this task very well, automating the processing of this elongated list of pixels. Still, many observatories produce images that generate up to 70% of false positive detections when used as inputs for this type of automatic tools. The false streaks are caused by artifacts generated by different factors like bright stars creating oversaturation disturbs, bad pixels or cosmic rays. In this paper, we propose a method to overcome the problem of detection of false positives. This method is based on exploiting data collected by ESA in STREET Database using modern data-driven techniques. We explore different solutions based on combination of multilayer perceptron (MLP), convolutional neural network (CNN) and vision transformers (ViT). The first approach is based on a multi-head deep neural network (DNN) architecture based on the ensemble of CoAtNet convolutional-attention-based model with a classical MLP. The CoAtNet[2] model manages the image data while the MLP manages categorical features, both part of STREET database. The proposed DNN uses manually obtained ground truth from the STREET Database [1] to learn how to classify detected pixels between real streaks and false positives (artifacts). An ablation test of single components (MLP and CoAtNet) is performed to quantify the importance of each component in an approach that can ensemble information both from images and categorical features. The usage of a lightweight deep learning model on categorical features extracted from streak images bring some advantages like efficient computation that does not require a GPU and transparency given by the explainability techniques. We achieve model accuracy up of 92% on false positives detection using both the configurations: the ensemble and also using MLP only. This result was confirmed through k-fold cross-validation. We added this new DL module at the end of the output of the pipeline of the existing legacy system improving the overall performance while maintaining the same level of transparency.

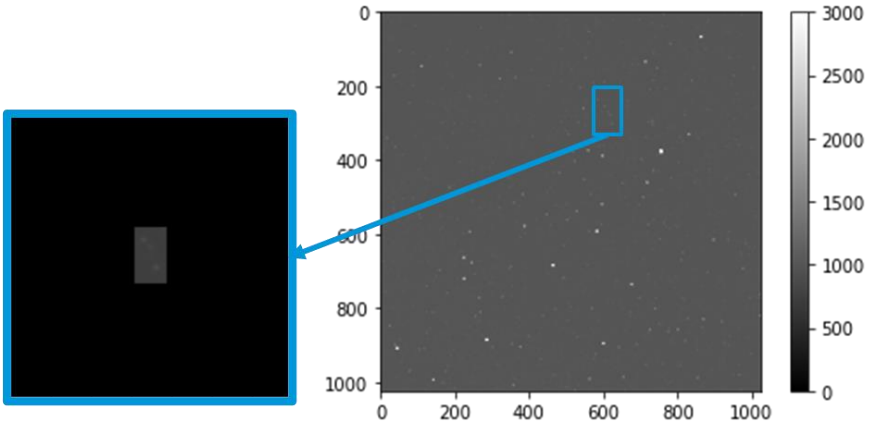

Fig. 2- Example of preprocessed image extracted from entire observed image. Being the bunding box smaller that 256x256 pixel, the image is resized applying zero-padding.

This was possible by employing explainability techniques. Through the interpretation of Shapley values obtained by simulations, we were able to interpret the magnitude of the contribution of each feature in the model. The explainability process was performed both qualitatively with comparison to the manual procedure of detection and quantitatively through statistical evidence which confirmed the shapely results. The union of classical validation techniques and explainable output of each prediction of the machine learning model is found to improve the level of trust and transparency even when employing a NN architecture with deep layers. We believe that this solution could be an example on how to increase the level of trust of using machine learning models in critical scenarios like space safety and how to integrate these types of artificial intelligence application in order to improve the already existing systems.

The structure of the paper is the following: streaks detection, the legacy system and the integration with the machine learning filter is described in Section 2. The DNNs used and dataset preprocessing methods are described in Section 3. In Section 4 experiments, testing and result performed are explained. Section 5 is dedicated to validation of the results including an analysis of explainability of the machine learning model. Section 6 is about discussion and future works before the concluding Section 7.

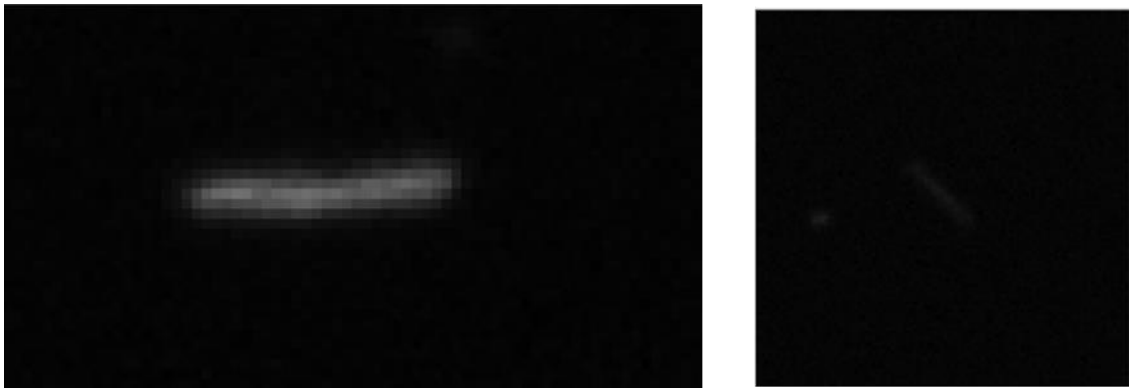

Fig. 3 - Example of faint streaks extracted from STREET database.

2. Streaks Detection and Legacy System

Streaks are low-SNR pixels corresponding to debris parts visible on images from ground-based/space-based telescopes. Streak detection is an important aspect of space debris analysis as it helps identify objects that are moving across the sky at high speeds.

This information is crucial for tracking the movement of satellites and predicting the potential collision risk with other objects in orbit. Streak detection is also used to monitor the behavior of space debris and identify any changes in its trajectory that may indicate a collision has occurred. There are several techniques used for streak detection, mostly based on the photometric reduction of such streak-like features, including the application of image processing algorithms on images acquired using ground-based telescopes and satellite-borne cameras[3]. Some recent works are based on the application of cross-correlation filtered images where radial base function is applied [4].

Being moving objects, streaks are often observed in images acquired using the technique of Sidereal Tracking. Sidereal tracking is the process of continuously adjusting the pointing direction of a telescope or other observing instrument to follow the apparent motion of celestial objects across the sky. This is necessary because the Earth's rotation causes celestial objects to appear to move relative to the observer, and tracking is required to keep the instrument pointed at the desired target. In sidereal tracking, the instrument is adjusted to follow the apparent motion of celestial objects with respect to the fixed stars. This is accomplished by continuously adjusting the pointing direction of the instrument to compensate for the Earth's rotation and is typically done using a motorized mount or other tracking mechanism.

Space debris streak detection from optical observations is a challenging problem due to the small size and high speed of debris objects, as well as the difficulties of tracking them against a cluttered and variable background. Despite these challenges, significant progress has been made in recent years in the development of methods for detecting and tracking space debris using optical observations. One approach that is promising is the use of advanced image processing and machine learning techniques to extract debris streaks from images and video sequences. These techniques can be used to detect and classify debris objects based on their appearance, size, and motion characteristics, and can be combined with traditional tracking algorithms to improve the accuracy and reliability of debris detection[1]. Another area of active research is the use of multi-sensor fusion and data association techniques to combine information from multiple sources, such as optical, radar, and lidar observations, to improve the accuracy and completeness of debris detection and tracking[5]. Overall, the state of the art in space debris streak detection from optical observations is rapidly evolving, and there is ongoing research to develop new and more effective methods for detecting and tracking these objects in order to improve space situational awareness and mitigate the risks posed by space debris.

Optical telescope images in the Space Surveillance and Tracking (SST) field may contain streaks as a list of pixels with higher local salient pixel density and more elongated shape than background noise and other artefacts. If the objective is to detect faint streaks (Fig.3), which have thinner, faint line of pixels, existing systems can produce a lot of false positives, with ranges that can go from 50% to 70% depending on the image coming from different observatories [3]. Artifacts can be in different forms and aspects, major causes can be multiples and simultaneously concurrent: bright stars that cause a state of oversaturation which can disturb readout amplifiers, bad pixels formed due to the degradation of observation instruments, cosmic rays that can interfere with digitalization of the image.

This effect can also happen simultaneously like the overlapping of a bright star with a real streak making this use case very challenging to existing techniques (Fig. 4).

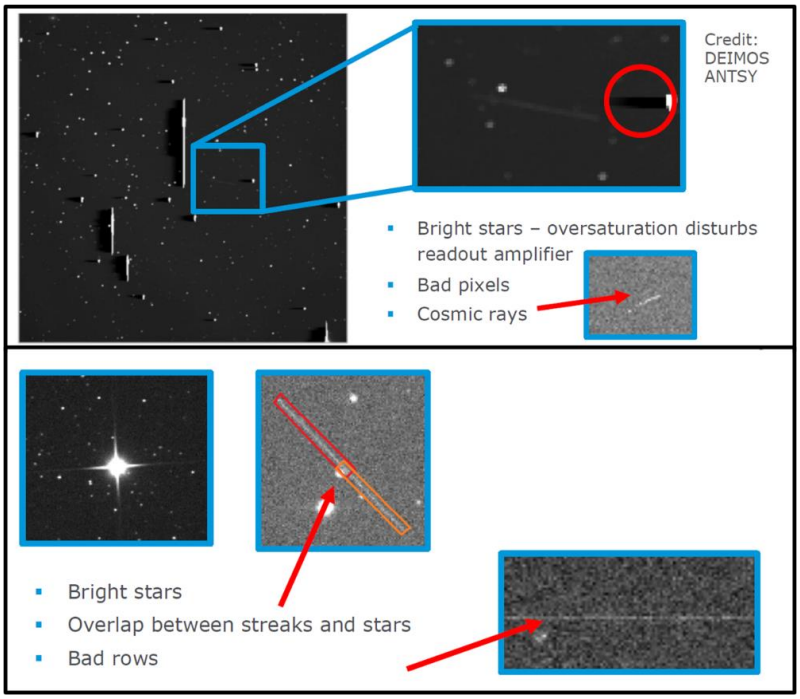

Fig. 4 - Example of artefacts that could be identified as false positives.

A typical example of detection system based on image processing is the one used also in Maric et. al. [1]. This is the "Legacy System" we are going to extend and is based on a processing chain that terminate with a manual verification of the detected streak. The AI module we introduce has the aim to remove the manual step thus going to automatize the whole pipeline. The proposed processing chain for astronomical image analysis includes the following steps:

1. Segmentation of the image to separate the background and features of interest.
2. Identification of stars in the image using a star-catalog matching algorithm.
3. Identification of objects of interest, such as features with certain lengths.
4. Fitting of a point spread function or line to the identified features.
5. Determination of the plate solution, mapping the X-Y coordinates to RA/DEC coordinates.
6. If multiple images are available, differencing of the images to detect any changes, such as moving objects.

Additional features can be extracted from images using tools like SExtractor [6]. An activity already performed by ESA with DEIMOS UK, has built a large database of these streak features to allow easy and seamless access to the acquired data. It is based on two independent tools (like StreakDet [1]) and is called STREET (STREak Evaluation Tool) Database (see Table 1). STREET comprehends ~120000 raw images. Between the feature present there are also X-Y coordinates (or RA/DEC) of streak centers, length, orientation. Every information is manually confirmed (for true positives, no false negatives information).

Table 1. The average percentage of detected streaks by Deimos Processing Tool and the number of streaks in total, including the estimated number of images with streak extrapolated from a manually checked set of images.

Sensor	Num. of images	Num. of observations with at least one streak	% of images with streaks	Estimated number of streak images
TRACKER 2	53532	26395	49.12	35220
ANTSY1	33717	25456	75.50	23253
PIRATE	23406	15897	67.92	20113
COAST	13141	7511	57.16	7999
PANOPTES-MAM	5400	3914	72.49	5074

3. Methods

Based on STREET database data described in section 2 and pre-existing system[1] we propose to add a filter to the end of the pipeline which can solve the problem of false positive detections. The filter consists of a small deep neural network which, based on features extracted from the streak images can perform the binary classification task that can really tell if the visualized pixels are a streak or an artifact. This addition requires a small preprocessing phase of features forming the new pipeline:

1. Full pipeline for detection based on [1]: Segmentation and extraction of features (using SExtractor)
2. Preprocessing of extracted features from legacy system
3. Real Streak vs Artifact classification with deep neural network

Our solution produces a reduction of false positives by 92%, potentially solving the problem of detection of artifacts and real streaks.

In the initial phase of this work, we include also images to evaluate the ensemble of both the contribution of images and features extracted by legacy system. Image pre-processing consists in the extraction of the bounding box containing the streak from the full sky observatory image (Fig.2). Pixel of the image are normalized, between the range 0-255. The extracted bounding box image dimension is resized to a fixed width and height. A statistical evaluation of the bounding box a reasonable dimension for each image excluding the lesser data possible (Fig.5, excluding data after ~256-dimension minimize data-loss).

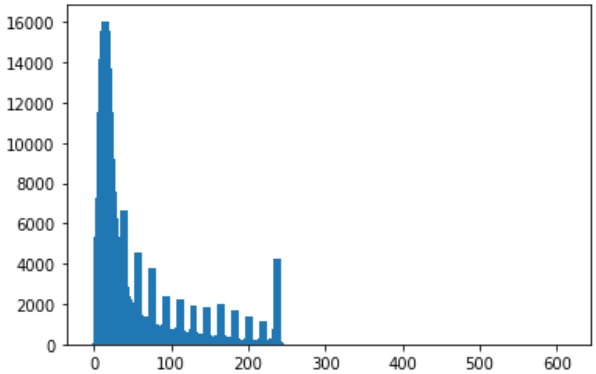

Fig. 5 - On x axis the sizes of major side of each bounding box, on y-axis its occourences.

If a bounding box image is lesser than 256x256 a zero-padding operation is performed. If a streak is larger than that dimension, the image is resized to 256x256 using linear pixel interpolation.

Subsequent from image pre-processing, there is processing of extracted features from legacy tool. A feature analysis was performed and a selection from the total is used for training. Partial reference to total parameters is to be found in SExtractor documentation[7]. Not all observatories yield the same features and for the same kind of data (real streak or artifact). In *italic* the ones that are *in common* between all the observatories and that will be considered for use in *training*. All features are of type numeric, apart from underlined ones. Only 'trail morphology' is a non-numeric feature and a feature "in common" between the different observatories. Trail morphology won't be considered for the training since is already a sort of "label" indicating possible classes: *'noise', 'artifact', 'vanishing_ends', 'chained', 'possibly_false_detected_trail', 'constant', 'intermitent'*. See Table 2.

Table 2. Each feature of the STREET database in correspondence with standard SExtractor output and brief description.

Streaks dataset feature	Used in training	Mapped feature (Sextractor)	Brief Description
barycenter x	Yes	XWIN_IMAGE	x coordinate of windowed image centroid
barycenter y	Yes	YWIN_IMAGE	y coordinate of windowed image centroid
beginning x	Yes	XMIN_IMAGE	Minimum x coordinate among detected pixels
end x	Yes	XMAX_IMAGE	Maximum x coordinate among detected pixels
beginning y	Yes	YMIN_IMAGE	Minimum y coordinate among detected pixels
end y	Yes	YMAX_IMAGE	Maximum y coordinate among detected pixels
flux auto	Yes	FLUX_AUTO	Kron-like automated aperture flux

flags	Yes	FLAGS	Source extraction flags
alpha J2000	Yes	ALPHA_J2000	J2000 right ascension of the isophotal image centroid
delta J2000	Yes	DELTA_J2000	J2000 declination of the isophotal image centroid
A world	Yes	No correspondence	Like A_image but in"world coordinates", converted from pixel coordinates using the local Jacobian of the transformation between both systems.
B world	Yes	No correspondence	Like B_image but in"world coordinates", converted from pixel coordinates using the local Jacobian of the transformation between both systems.
A image	Yes	A_IMAGE	Isophotal image major axis
B image	Yes	B_IMAGE	Isophotal image minor axis
elongation	Yes	ELONGATION	A_IMAGE / B_IMAGE
theta J2000	Yes	No correspondence	Like THETA_IMAGE but in J2000 measurement system. Isophotal image position angle
theta image	Yes	THETA_IMAGE	Isophotal image position angle
mag auto	Yes	MAG_AUTO	Kron-like automated aperture magnitude
xpeak image	Yes	XPEAK_IMAGE	Pixel x coordinate of the brightest pixel
ypeak image	Yes	YPEAK_IMAGE	Pixel y coordinate of the brightest pixel
background	Yes	BACKGROUND	Background level at the position of the centroid
flux best	Yes	FLUX_AUTO	Kron-like automated aperture flux
magnitude	No	MAG_AUTO	Kron-like automated aperture magnitude
real length	No	No correspondence	Length of the streak in real measurement unit
saturation	No	No correspondence	Saturation describes the intensity of the color.
length to FOV	No	No correspondence	Length to Field of View
length	No	No correspondence	Length in pixel
length arcsec	No	No correspondence	Length in arcsec
length arspeed	No	No correspondence	Length in arspeed
out of FOV	No	No correspondence	Boolean describing the state ofout of field of view
distance	No	No correspondence	Probably distance in world coordinates from observed object.
image center x	No	X_IMAGE	Pixel x coordinate of the isophotal image centroid
image center y	No	Y_IMAGE	Pixel y coordinate of the isophotal image centroid
angle of motion	No	No correspondence	Angle of motion
trail morphology	No	No correspondence	Already a label to classify streaks (not manually validated).

Only 21 features are selected and are the one in common between the output from all observatories and that are not already a form of label (like "trail morphology"). Principal Component Analysis (PCA) is performed on the 21 selected features and cumulative summed variance is shown to be able to explain the overall variance of the dataset (see Fig. 6).

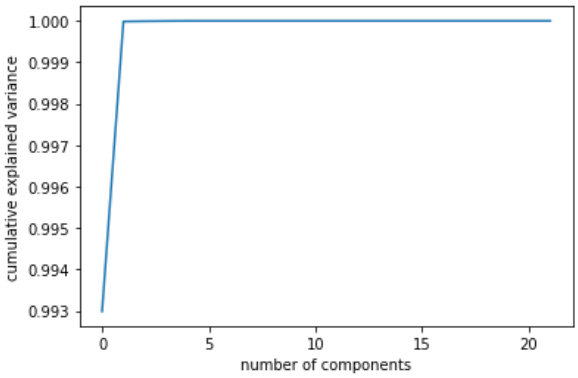

Fig. 6 - PCA Analysis on 21 selected features: The features are largely enough to cover all the variance of the dataset. With 8 features, 100% cumulative variance is reached.

The proposed neural network architecture for false positive detection has three possible main configurations: CoAtNet with only images, MLP for extracted features and a hybrid composition of the two. CoAtNet is implemented as the architecture described in the original paper[2] as configuration CoAtNet0 with only 25M parameters. The MLP has the architecture described in Fig.7 with only 201 parameters.

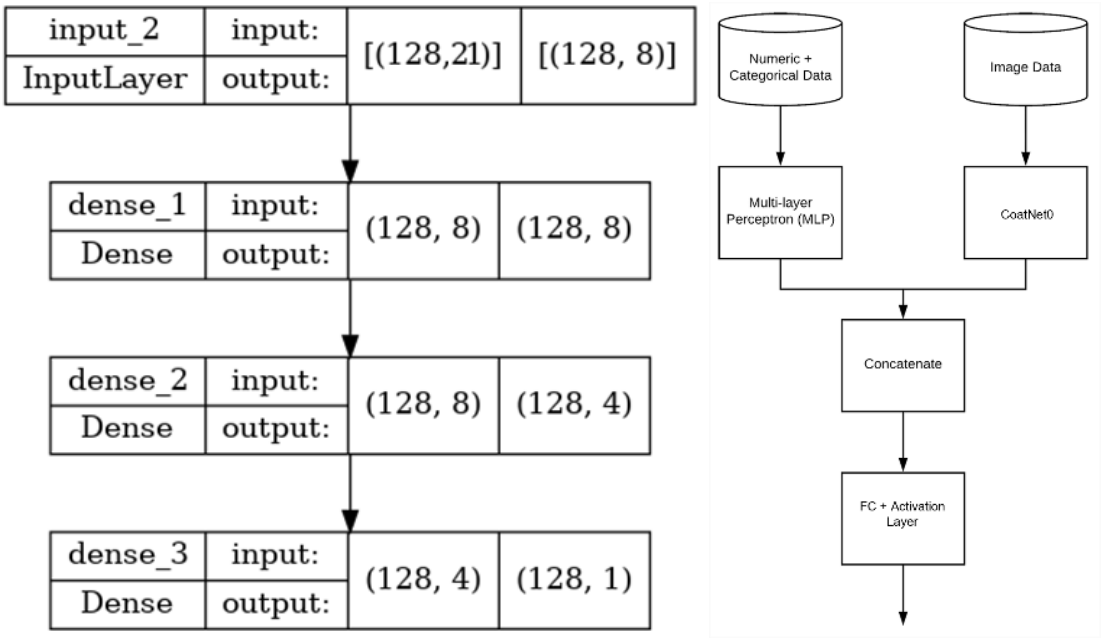

Fig.7 - Proposed MLP architecture for inference from features on the left and hybrid architecture on the right.

Hybrid architecture (Fig.7) combines the one but last layer of the CoAtNet0 and concatenate it on the second axis with the one but last layer of the MLP. Another fully connected layer is added at the end along with the output layer. The objective for the hybrid structure is to exploit information from both images and extracted features. The binary classification is performed with two classes labelled: "real streak" and "artifact".

Training is performed on the overall 298240 streak observation of the dataset with the following split for testing, validation and training set: 20% of the samples are kept for testing, 10% of the remaining 80% is used as validation set to keep track of overfitting and to permit the use of techniques like early stopping. The remaining data is used for training, the exact number are in Table 3.

Table 3. Dataset splitting

Dataset (total 298240)	Number of samples
Testing (20% of total)	59648
Validation (10% of 80% of total)	23859
Training (90% of 80% of total)	214733

Composition of Training and Validation set is varied applying K-fold cross-validation. Testing set remain constant. This is to exclude the possibility of "a fluke", where a lucky/unlucky split of data produce better or worse results. Results of multiple experiments will be then averaged to produce final results.

Supervised training is performed employing classical techniques like early stopping to prevent overfitting. The optimizer employed is Adam with an initial learning rate of $5x10^{-4}$ and learning rate scheduler with 100000 decay steps and a decay rate of 0.96. Loss function is binary cross entropy and real-time metrics is binary accuracy. Everything is implemented in python using Keras for architecture definition. Tensorflow is used to write a custom Data Generator which yield the hybrid data and feed it to the multiple input architecture. All the training is performed on a machine with two Nvidia RTX 3090 GPUs in SLI. The architecture needs to be configured in Keras to perform training in parallel on both GPUs.

4. Preliminary Results

Results obtained for the first configuration that uses only images are close to the random-choice performance (60% accuracy) stabilizing around epoch 1000 and performing poorly (Table 5).. Training in the other two configurations converge before Epoch 100. Results are promising and summarized in Table 4 and Fig. 8.

Table 4. Results obtained with 10-fold cross validation on testing set with hybrid configuration.

	Precision	Recall	F1-score	Support
Artifact	0.91	0.89	0.90	197840
Real streak	0.93	0.94	0.93	279344
Average	0.92	0.92	0.92	477184
Accuracy		0.92		477184

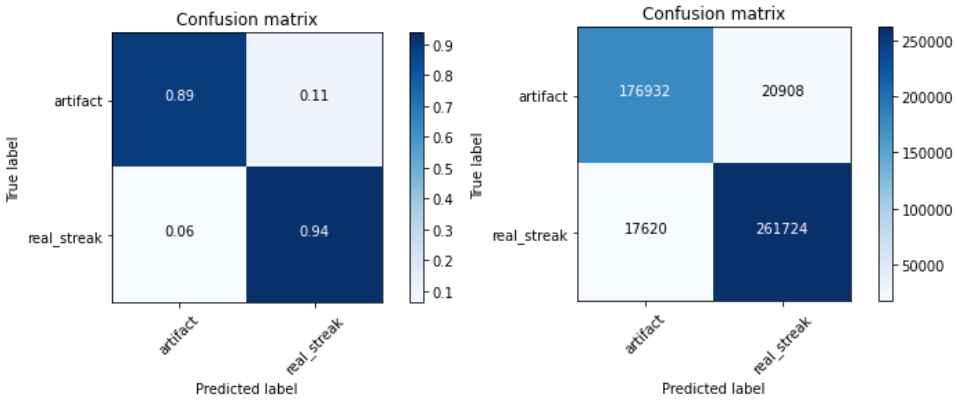

Fig. 8 - Confusion Matrix of training results of MLP

Table 5 - Results obtained with 10-fold cross validation on testing set with CoAtNet0 image-only configuration.

	Precision	Recall	F1-score	Support
Artifact	0.43	0.14	0.13	197840
Real streak	0.63	0.93	0.73	279344
Average	0.53	0.54	0.43	477184
Accuracy	0.60			477184

An ablation test is performed, each configuration is tested separately, and results are compared, to understand the contribution of the component to the overall system (Table 5). MLP configuration produced the same result as the MLP-only configuration suggesting that the almost full impact on the final result is given from the MLP trained on the features extracted by the legacy system. This is a first form of validation that led the next phase of this work which is the validation and transparency of the model which is described in section 5.

Table 6 - Results of ablation test: MLP seems to be the only contribution to final result.

Model	Average Accuracy from K-fold
Hybrid (CoatNet0 + MLP)	92%
Only CoatNet0	65%
Only MLP	92%

5. Model Validation and Explainability

Validation of the model is a crucial aspect of development of a machine learning application, especially in a critical scenario like space safety. To maintain the level of transparency while inserting a black box model such as a DNN at the end of the existing pipeline, explainability of the model is employed. Explainability techniques we propose to apply is based on Shapley values[8]. Shapley values are a method for explaining the contribution of each feature (or feature value) to the prediction of a model. They were originally developed in game theory to explain the contribution of each player to the overall game and have been adapted for use in machine learning to understand the relative importance of different features in a model's predictions. In our case, the game is the prediction of the model, while the players are the features included in the model.

To calculate Shapley values, we first compute the expected prediction of the model when all the feature values are randomly permuted. This gives us a baseline prediction that represents the average prediction of the model when all the features are scrambled. We then compute the change in the model's prediction that results from adding in each feature one at a time, and this change is the Shapley value for that feature.

In this way, Shapley values can provide insight into which features are most important for making a prediction and can help us to understand how the model is using different features to make its predictions. They can be particularly useful for understanding the behaviour of complex, black-box models, where it can be difficult to understand how the model is using the input features to make its predictions.

We apply Shapley values using the library SHAP (SHapley Additive exPlanations)[9] which is compatible with our python implementation in Keras. We employ SHAP's kernel explainer to MLP model to quantify the importance of each feature extracted by the legacy system to explain the result of the final prediction of the MLP model.

First Shapley Values Analysis (SVA) highlighted 8 features as the most impactful ones in the model: 'end x', 'beginning x', 'beginning y', 'end y', 'xpeak image', 'barycenter y', 'ypeak image', 'barycenter x', (Fig. 9). This is no surprise and add confidence in the predictions of the system since the combination of this feature "hides" streak length and orientation, which is valuable information. This cannot seem useful for a dedicated tracking scenario (i.e., when following a known object), the location of the streak is close to the centre. In all the other cases instead, for unknown objects, this is even more meaningful when trying to identify them.

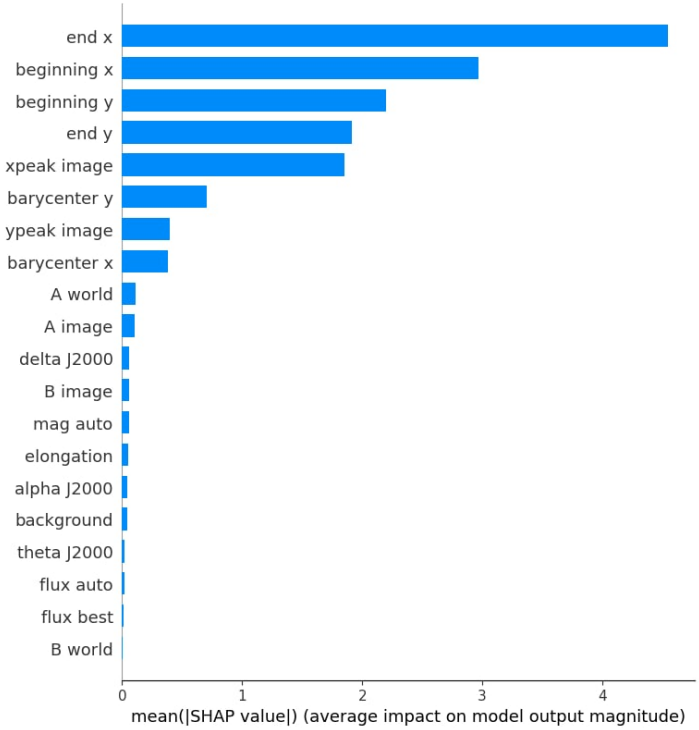

Fig. 9 - Most impactful features in MLP model

These results are promising since they reflect the characteristic feature that are normally looked upon when performing the procedure manually, also remanding to definition of streak: "a list of pixels with higher local salient pixel density and more elongated shape than background noise and other artefacts"[1]. Other than a qualitative evaluation SVA can motivate a quantitative evaluation based on correlation. There is a strong correlation between the most impactful features and the features "xpeak" and "ypeak", this confirming furthermore how the learned parameters are in line with the "higher salient pixels" phrase the from definition of streak. This correlation has been generated and studied using the tool LUX[10] and is visualized in Fig. 10.

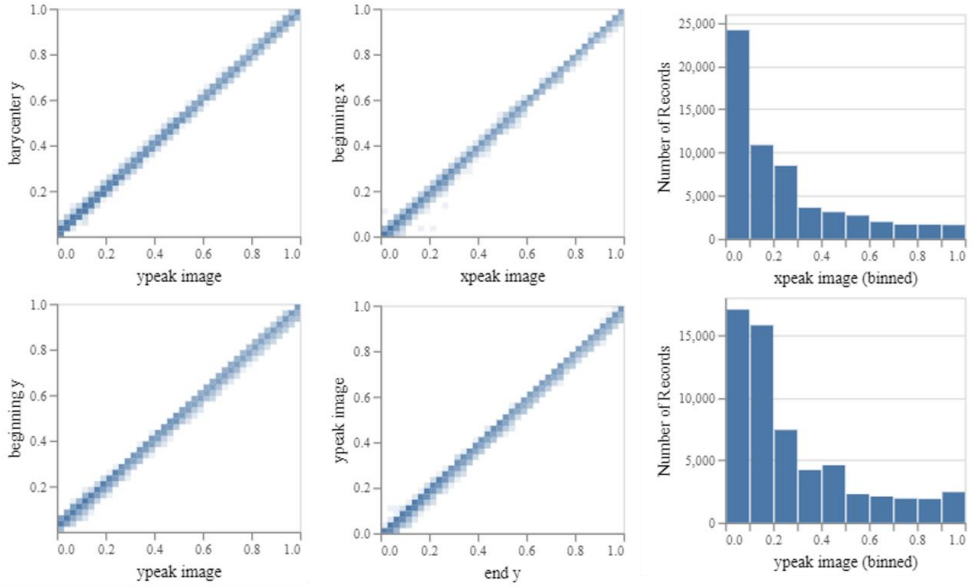

Fig. 10 - Correlations and distributions of values for features xpeak and ypeak.

A second training of the MLP network considering only the most impactful 8 features is then made following the qualitative and quantitative evaluation already presented. This training has been performed with K-fold procedure described before and yield the results describe in Table 7 and Fig.11.

Table 7 - Results of training with only 8 most impactful features.

	Precision	Recall	F1-score	Support
Artifact	0.81	0.83	0.82	197840
Real streak	0.99	0.86	0.97	279344
Average	0.84	0.85	0.85	477184
Accuracy	0.85			477184

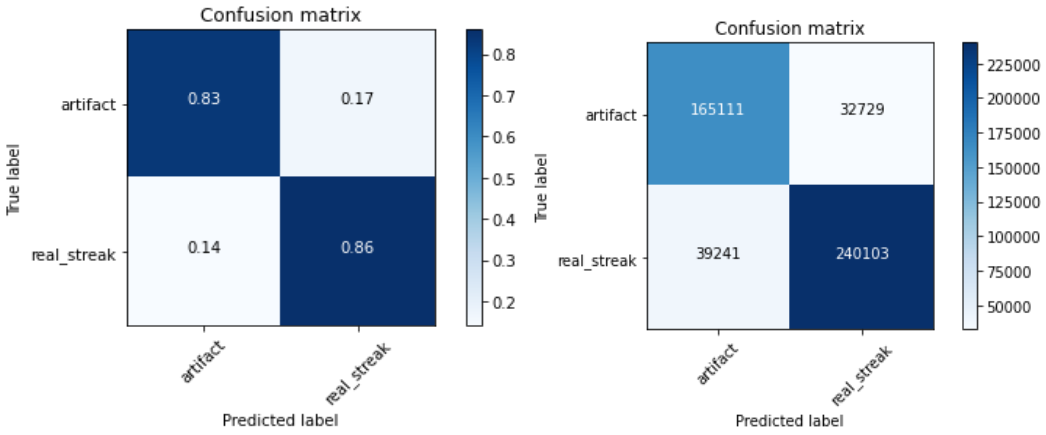

Fig. 11 - Confusion matrix for training of MLP with only 8 most impactful features.

These results confirm the SVA results and with 85% accuracy achieved still explain the small, but still not negligible impact of the other features that can make up for that 7% accuracy gap missing. A new SVA performed on the reduced model with only 8 features re-confirmed the importance of this feature and is visualized in Fig. 12.

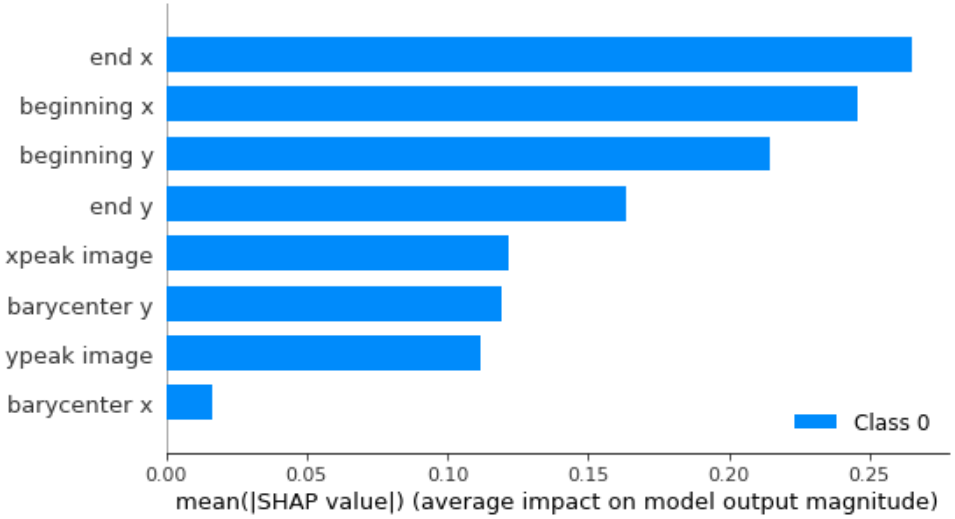

Fig. 12 - Impact of the selected 8 features on the reduced MLP model trained.

A further analysis of correlations between the remaining features revealed minor correlations between the remaining features that could explain the meaningfulness they have in the model to permit to reach 92% accuracy (Fig. 13).

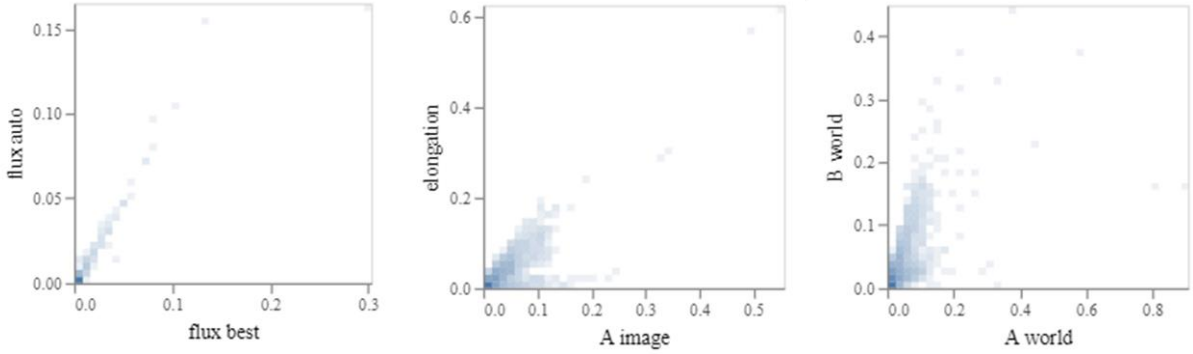

Fig. 13 - Correlation analysis of the remaining features.

6. Discussion and future works

Our machine learning solution was able to significantly improve the performance of the overall system by exploiting the huge amount of data present in STREET database. False positive detection has been reduced by 92% making it feasible in detecting faint streaks and removing a manual step from the pre-existing processing pipeline (Fig. 14). This result has been thoroughly validated using classical validation techniques and explainability techniques based on Shapley Values. By automatically identifying false detected streaks, the model was able to make more accurate and reliable detection, leading to more efficient and effective decision-making. At the same time, we were able to maintain the same level of trustworthiness and transparency by applying rigorous validation techniques like cross-validation and using explainability methods to provide insight into the model's decision-making processes. This gave the possibility to understand how the model was arriving at its predictions and provided confidence in the results. The whole explainability SVA has been supported by both qualitative and quantitative interpretations by making a comparison with parameters considered when performing the streak identification manually and considering relevant features statistical intra correlations. Overall, the implementation of our machine learning solution resulted in a noticeable improvement in the performance of the system while maintaining a high level of trust and transparency.

Future works could include a more in-depth analysis considering single results for each single observatory/telescope and try to train a machine learning model for each type of observation from single observatory. Also, a test of the

overall pipeline on space-based observations to test its generalizability would be an interesting experimentation. A retrain and test of the overall machine learning model adding noise to bounding box coordinates could lead to a deeper understanding of the elements considered by the AI-module in the decision making. In the end an AI-based segmentation of detected faint-streaks based on unsupervised algorithms like self-organizing maps would be a great addition to the actual system.

By implementing machine learning algorithms on the spacecraft ("on the edge" as is customary now), we can reduce the data transmission demands from the spacecraft to the ground. This is achieved because the algorithms are able to process the data while in flight and only send the critical information back to the ground. This strategy could prove valuable in detecting space debris, as the algorithm can work in parallel with existing general-purpose imagers such as star-trackers to provide the necessary data to verify and supplement ground-based measurements. The feasibility and use of similar machine learning techniques for star trackers has been demonstrated by these paper authors in [11] while the power and processing benchmarking on several flight processors has been discussed in [12].

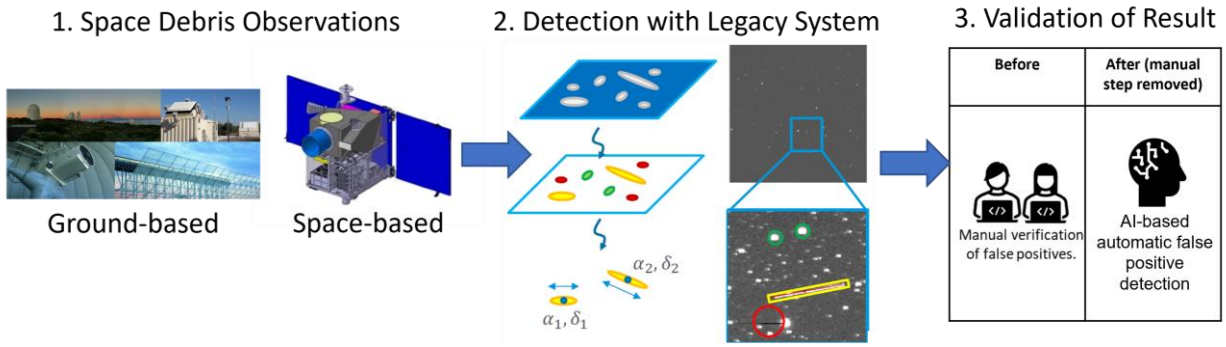

Fig. 14 – Overall pipeline from space debris observation to validation of the detected streak. Is clear as the insertion of an AI module improved the usability of the system removing a manual step.

7. Conclusions

The results of our work have been very promising, with the machine learning model consistently demonstrating strong performance and accuracy. This has led us to believe that our solution could serve as an example for increasing the level of trust in using machine learning models in critical scenarios, such as space safety. By applying rigorous validation techniques and using explainability methods, we have been able to demonstrate the reliability and robustness of the model, which is crucial in high-stakes situations. In addition, the ability to understand and interpret the model's decision-making processes helps to build confidence in the results and can help to mitigate concerns about the potential risks of relying on automated decision-making. Overall, we believe that our solution could serve as a model for using machine learning in critical scenarios and help to increase the trust in these types of systems.

References

[1] N. Maric *et al.*, "Streak Evaluation Tool," presented at the 8th European Conference on Space Debris, 2021. Accessed: Dec. 27, 2022. [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/148

[2] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoAtNet: Marrying Convolution and Attention for All Data Sizes." arXiv, Sep. 15, 2021. doi: 10.48550/arXiv.2106.04803.

[3] "Hunting streaks in space." https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Shaping_the_Future/Hunting_streaks_in_space (accessed Jan. 07, 2023).

[4] G. Nir, B. Zackay, and E. O. Ofek, "Optimal and Efficient Streak Detection in Astronomical Images," *AJ*, vol. 156, no. 5, p. 229, Oct. 2018, doi: 10.3847/1538-3881/aaddff.

[5] B. Wei and B. D. Nener, "Multi-Sensor Space Debris Tracking for Space Situational Awareness With Labeled Random Finite Sets," *IEEE Access*, vol. 7, pp. 36991–37003, 2019, doi: 10.1109/ACCESS.2019.2904545.

[6] "SExtractor – Astromatic.net." https://www.astromatic.net/software/sextractor/ (accessed Jan. 07, 2023).

[7] "The measurement (or catalog) parameter file — SExtractor 2.24.2 documentation." https://sextractor.readthedocs.io/en/latest/Param.html (accessed Jan. 08, 2023).

[8] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, 2017, vol. 30. Accessed: Jan. 08, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[9] "Welcome to the SHAP documentation — SHAP latest documentation." https://shap.readthedocs.io/en/latest/ (accessed Oct. 06, 2021).

[10] "Lux: A Python API for Intelligent Visual Discovery — Lux 0.1.2 documentation." https://lux-api.readthedocs.io/en/latest/ (accessed Jan. 09, 2023).

[11] Rijlaarsdam D, Yous H, Byrne J, Oddenino D, Furano G, Moloney D. Efficient Star Identification Using a Neural Network. *Sensors*. 2020; 20(13):3684. https://doi.org/10.3390/s20133684

[12] Lentaris, George; Maragos, Konstantinos; Stratakos, Ioannis; Papadopoulos, Lazaros; Papanikolaou, Odysseas; Soudris, Dimitrios; Lourakis, Manolis; Zabulis, Xenophon; Gonzalez-Arjona, David; Furano, Gianluca - High-performance embedded computing in space: Evaluation of platforms for vision-based navigation Journal of Aerospace Information Systems, 15 4 178-192 (2018) - American Institute of Aeronautics and Astronautics