






# Identification of problematic epochs in astronomical time series through transfer learning<sup>★,★★</sup>

Stefano Cavuoti<sup>1,2,★★</sup>, Demetra De Cicco<sup>3,1,4,★★</sup>, Lars Doorenbos<sup>5</sup>, Massimo Brescia<sup>3,1,2</sup>,  
Olena Torbaniuk<sup>6,7</sup>, Giuseppe Longo<sup>3</sup>, and Maurizio Paolillo<sup>3,1,2</sup>

<sup>1</sup> INAF – Osservatorio Astronomico di Capodimonte, via Moiariello 16, 80131 Napoli, Italy  
e-mail: [stefano.cavuoti@inaf.it](mailto:stefano.cavuoti@inaf.it)

<sup>2</sup> INFN – Section of Naples, via Cinthia 9, 80126 Napoli, Italy

<sup>3</sup> Department of Physics, University of Napoli “Federico II”, via Cinthia 9, 80126 Napoli, Italy  
e-mail: [demetra.decicco@unina.it](mailto:demetra.decicco@unina.it)

<sup>4</sup> Millennium Institute of Astrophysics (MAS), Nuncio Monseñor Sotero Sanz 100, Providencia, Santiago, Chile

<sup>5</sup> AIMI, ARTORG Center, University of Bern, Murtenstrasse 50, Bern 3008, Switzerland

<sup>6</sup> Department of Physics and Astronomy ‘Augusto Righi’, University of Bologna, via Piero Gobetti 93/2, 40129 Bologna, Italy

<sup>7</sup> INAF – Osservatorio di Astrofisica e Scienza dello Spazio di Bologna, via Piero Gobetti 101, 40129 Bologna, Italy

Received 28 March 2024 / Accepted 6 May 2024

## ABSTRACT

**Aims.** We present a novel method for detecting outliers in astronomical time series based on the combination of a deep neural network and a k-nearest neighbor algorithm with the aim of identifying and removing problematic epochs in the light curves of astronomical objects.

**Methods.** We used an EfficientNet network pretrained on ImageNet as a feature extractor and performed a k-nearest neighbor search in the resulting feature space to measure the distance from the first neighbor for each image. If the distance was above the one obtained for a stacked image, we flagged the image as a potential outlier.

**Results.** We applied our method to a time series obtained from the VLT Survey Telescope monitoring campaign of the Deep Drilling Fields of the *Vera C. Rubin* Legacy Survey of Space and Time. We show that our method can effectively identify and remove artifacts from the VST time series and improve the quality and reliability of the data. This approach may prove very useful in light of the amount of data that will be provided by the LSST, which will prevent the inspection of individual light curves. We also discuss the advantages and limitations of our method and suggest possible directions for future work.

**Key words.** methods: data analysis – techniques: image processing – galaxies: active

## 1. Introduction

Astronomical time series are sequences of data that represent the variation of a physical quantity over time in the observation of celestial objects or phenomena. They are essential for studying and discovering the properties and dynamics of various astrophysical sources, such as stars, planets, supernovae, black holes, and galaxies (Scargle 1997; Aigrain & Foreman-Mackey 2023).

However, an astronomical time series analysis poses several challenges, such as the presence of noise, gaps, outliers, and artifacts in the data. Artifacts are unwanted signals that do not reflect the true nature of the observed object and are caused by external factors, such as instrumental errors, atmospheric conditions, peculiar events (e.g., cosmic rays, satellite tracks), and contaminant sources (Li et al. 2018; Malz et al. 2019). Artifacts can severely affect the quality and reliability of data and may lead to false or misleading conclusions. Therefore, it is crucial to detect and remove artifacts from astronomical time series before performing any further analysis.

\* Observations were provided by the ESO programs 088.D-4013, 092.D-0370, and 094.D-0417 (PI: G. Pignata).

\*\* The code is available at <https://github.com/cavuoti/AnomalyInTimeSeries>.

\*\*\* The first two authors share the first authorship and are in alphabetical order.

One specific case of astronomical time series are those obtained from the VLT Survey Telescope (VST), a 2.6-meter optical telescope located at the Paranal Observatory in Chile. The VST is designed to perform wide-field imaging surveys of the southern sky, covering a field of view (FoV) of one square degree with a pixel scale of 0.214" (Capaccioli & Schipani 2011). The VST time series are useful for studying various topics, such as variable stars, transient events, Solar System objects, and cosmology (Cappellaro et al. 2015; Falocco et al. 2015; De Cicco et al. 2015, 2019, 2021; Botticella et al. 2017; Fu et al. 2018; Liu et al. 2020; Poulain et al. 2020).

The VST time series, as all astronomical data, are affected by artifacts, such as bad pixels, saturation, cosmic rays, ghost images, and background fluctuations. These artifacts can hamper the detection and characterization of faint or fast-varying sources and decrease the scientific value of the data. Data reduction and calibration methods (see, e.g., Grado et al. 2012) are expected to correct and mask such artifacts or, when this proves difficult, flag them to allow easy filtering in the data analysis phase. However, it is not unusual to find residual defects in the reduced data, especially in time series, where we cannot use time averaging to remove many of the aforementioned problems. Therefore, it is important to develop improved methods for identifying and flagging these artifacts in order to limit their impact on scientific results.

One possible way to tackle this problem is to use transfer learning, a technique in machine learning that allows one to leverage the knowledge learned from a source domain to improve the performance on a target domain. Transfer learning can be useful when the target domain has limited or scarce data or when the source domain has some similarities or relevance to the target domain, as happens in astronomy (at least in terms of labels; Awang Iskandar et al. 2020; Martinazzo et al. 2021), malware classification (Prima & Bouhorma 2020), Earth science (Zou & Zhong 2018), and medicine (Ding et al. 2019; Esteva et al. 2017; Kim et al. 2021; Menegola et al. 2017).

In this work, we propose the use of an EfficientNet network trained on ImageNet as an anomaly detector for astronomical time series. An EfficientNet network (Tan & Le 2019) is a convolutional neural network (CNN) architecture that scales up the network depth, width, and resolution in a balanced and efficient way using a compound coefficient. ImageNet is a large-scale dataset often used for such a purpose (Deng et al. 2009), and it contains around 1.3 million images, where the original task was to classify each image into one of 1000 classes.

The idea is to use the EfficientNet network as a feature extractor, as proposed in Doorenbos et al. (2022), in the detection of active galactic nuclei (AGN) and to then use an algorithm for the detection of sources deviating from the standard behavior.

This approach can improve and automatize the identification and removal of problematic epochs in the light curves without the need for visual inspection. This process can therefore improve the reliability of the light curves and the results obtained with their analysis. We expect this to be highly relevant in light of the *Vera C. Rubin* Legacy Survey of Space and Time (LSST; Ivezić et al. 2019).

The paper is organized as follows: in Sect. 2, we describe the data used for this work, while in Sect. 3, we present the method. In Sect. 4, we discuss the results, and finally in Sect. 5 we draw our conclusions.

## 2. Data

As a benchmark, we used the same dataset as De Cicco et al. (2019, 2021, 2022). The dataset consists of *r*-band observations of the COSMOS field performed with the VST spanning three observing seasons (hereafter, seasons) from December 2011 to March 2015 and are part of a long-term effort to monitor the LSST Deep Drilling Fields before the start of the *Vera Rubin* Telescope operations. The VST is a 2.6-meter optical telescope that covers a FoV of  $1^\circ \times 1^\circ$  with a single pointing (the pixel scale is  $0.214'' \text{ pixel}^{-1}$ ). The three seasons include 54 visits in total and with two gaps; more details about the dataset are given in Table 1 (adapted from Table 1 of De Cicco et al. 2019). As explained in De Cicco et al. (2019), they excluded 11 of the 65 visits that constitute the full dataset.

The observations in the *r*-band were designed to have a three-day observing cadence, although the exact final cadence depended on the observational constraints. The single-visit depth is  $r \lesssim 24.6$  mag for point sources, at a  $\sim 5\sigma$  confidence level. This makes our dataset particularly interesting for studies aimed at forecasting the performance of the LSST, as its single-visit depth is expected to be similar to our stacked COSMOS images. We refer the reader to De Cicco et al. (2015) for information on the reduction and combination of the exposures performed using the VST-Tube pipeline (Grado et al. 2012) as well as the source extraction and sample assembly. The VST-Tube magnitudes are in the AB system.

The sample contains 22 927 sources detected in at least 50% of the visits in the dataset (i.e., they have at least 27 points in

their light curves) with an average magnitude of  $r \leq 23.5$  mag within a  $1''$ -radius aperture. We started with the same sample for our analysis, but while De Cicco et al. (2019) focused on the sub-sample of sources selected by either variability or multiwavelength properties, we used the entire sample, as done in De Cicco et al. (2021). In fact, our aim is to efficiently identify the unreliable measurements in the full dataset before engaging in any classification effort that could be biased by photometric or aesthetic artifacts. These problematic sources are mainly satellite tracks, aesthetic defects, cosmic rays, saturation, and bad seeing together with objects blended with a neighbor in at least some of the visits (those with poor seeing), making it substantially harder to measure fluxes correctly. We decided to analyze and compare this sample, as it comes from the pipeline and dataset already presented in De Cicco et al. (2021).

As the reference for each object, we used the stacked image, that is, the median of all the exposures in the first season – which covers a five-month baseline – with a seeing full width at half maximum (FWHM) of less than  $0.80''$ . This choice ensured that the reference image is of higher quality than each individual epoch and that it does not include all the defects present in the whole dataset. A visual representation of the resulting image from single epochs is shown in Fig. 1.

## 3. Method

We propose a novel method for detecting outliers in astronomical time series based on combining a deep neural network and a *k*-nearest neighbor (*k*-NN) algorithm. The preliminary phase is similar to what has been done in Doorenbos et al. (2022). The main idea is to use an EfficientNet, a CNN (Tan & Le 2019), pretrained on ImageNet as a feature extractor and to perform a *k*-NN search in the resulting feature space to measure the distance to the first neighbor for each image. If the distance is above a certain threshold, we flag the image as a potential outlier.

Convolutional neural networks (Lecun et al. 1990; LeCun et al. 2015) are a class of deep neural networks designed for processing structured grid data, such as images. They have been proven to be highly effective in computer vision tasks, including image classification, object detection, and segmentation. A typical CNN consists of multiple layers, including:

- Convolutional layers: these layers apply convolution operations to their input, using learnable filters to extract features. Convolutional layers help capture hierarchical patterns in the input.
- Pooling layers: pooling layers reduce the spatial dimensions of the input by downsampling. Common pooling operations include max pooling, which retains the maximum value in a region, and average pooling, which computes the average.
- Fully connected layers: these layers connect every neuron to every neuron in the previous and subsequent layers, and they help make predictions based on the extracted features.
- Activation functions: non-linear activation functions, such as a rectified linear unit (ReLU), introduce non-linearity, allowing the network to learn complex relationships.

An example of the typical architecture of a CNN is shown in Fig. 2.

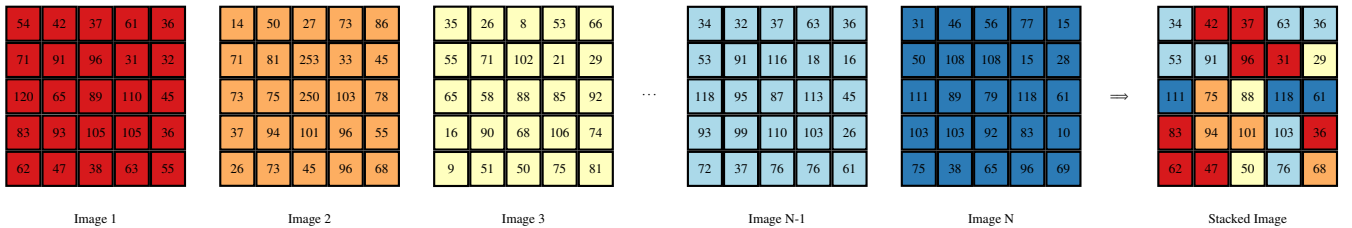
Scaling in CNNs involves adjusting various architectural dimensions to control the model complexity and computational efficiency. In particular, classical CNNs perform one of the following kinds of scaling:

- Depth scaling: increasing the depth of a CNN involves adding more layers. Deeper networks can capture more complex features but require more computational resources.

**Table 1.** VST-COSMOS dataset used in this work.

Visit	OB-ID	Obs. date	Seeing ( <i>FWHM</i> ) (arcsec)	Visit	OB-ID	Obs. date	Seeing ( <i>FWHM</i> ) (arcsec)
1	611279	2011-Dec.-18	0.64	28	986614	2013-Dec.-30	1.00
2	611283	2011-Dec.-22	0.94	29	986617	2014-Jan.-03	0.86
3	611287	2011-Dec.-27	1.04	30	986620	2014-Jan.-05	0.81
4	611291	2011-Dec.-31	1.15	31	986626	2014-Jan.-12	0.73
5	611295	2012-Jan.-02	0.67	32	986630	2014-Jan.-21	1.18
6	611299	2012-Jan.-06	0.58	33	986633	2014-Jan.-24	0.80
7	611311	2012-Jan.-18	0.62	34	986648	2014-Feb.-09	1.28
8	611315	2012-Jan.-20	0.88	35	986652	2014-Feb.-19	0.89
9	611319	2012-Jan.-22	0.81	36	986655	2014-Feb.-21	0.93
10	611323	2012-Jan.-24	0.67	37	986658	2014-Feb.-23	0.81
11	611327	2012-Jan.-27	0.98	38	986661	2014-Feb.-26	0.81
12	611331	2012-Jan.-29	0.86	39	986664	2014-Feb.-28	0.77
13	611335	2012-Feb.-02	0.86	40	986670	2014-Mar.-08	0.91
14	611351	2012-Feb.-16	0.50	41	986674	2014-Mar.-21	0.96
15	611355	2012-Feb.-19	0.99	42	986677	2014-Mar.-23	0.92
16	611359	2012-Feb.-21	0.79	43	986680	2014-Mar.-25	0.66
17	611363	2012-Feb.-23	0.73	44	1095777	2014-Mar.-29	0.89
18	611367	2012-Feb.-26	0.83	45	1095783	2014-Apr.-04	0.58
19	611371	2012-Feb.-29	0.90	46	986683	2014-Apr.-07	0.61
20	611375	2012-Mar.-03	0.97	47	1136410	2014-Dec.-03	1.00
21	611387	2012-Mar.-13	0.70	48	1136457	2015-Jan.-10	0.71
22	611391	2012-Mar.-15	1.08	49	1136481	2015-Jan.-28	0.90
23	611395	2012-Mar.-17	0.91	50	1136490	2015-Jan.-31	0.73
24	768813	2012-May-08	0.74	51	1136503	2015-Feb.-15	0.70
25	768817	2012-May-11	0.85	52	1136531	2015-Mar.-10	0.80
26	768820	2012-May-17	0.77	53	1136540	2015-Mar.-14	0.84
27	986611	2013-Dec.-27	0.72	54	1136543	2015-Mar.-19	1.00
				stacked	–	–	0.67

**Notes.** The four columns report the visit number, OB identification number, observing date, and seeing FWHM, respectively, for the 54 visits used for the present analysis; visits are listed in chronological order. All the visits were obtained by combining five exposures, for a total exposure time of 1800 s. However, visit 53 was obtained by combining ten exposures, for a total exposure time of 3600 s. Adapted from Table 1 of [De Cicco et al. \(2019\)](#).



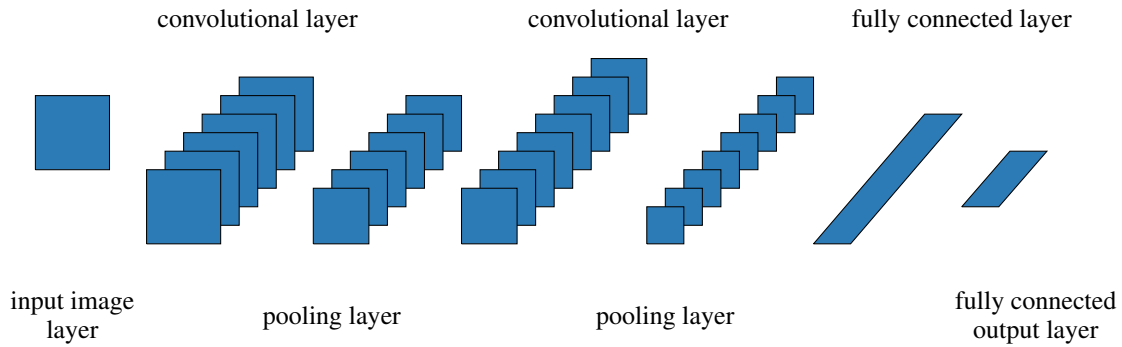
**Fig. 1.** Example derivation of our stacked image starting from individual images. The individual images are represented by simple  $5 \times 5$  matrices. The median value for each cell is selected in order to derive the stacked image. This process is suitable to avoiding wrong values in single epochs. For example, the values above 250 in Image 2 have been removed completely, as when using the mean value, the presence of such values, although mitigated by the number of independent images, would affect the final result much more.

- Width scaling: wider networks increase the number of filters in each layer. This helps in capturing more diverse features but also leads to higher computational requirements.
- Resolution scaling: resolution scaling involves adjusting the input image size. Lower resolutions reduce computational demands but may result in loss of information.

The EfficientNet network is a CNN architecture that scales up the network depth, width, and resolution in a balanced and efficient way. This method is called compound scaling, and it combines depth, width, and resolution scaling in a balanced manner, maintaining a constant ratio. It achieved state-of-the-art accuracy on

ImageNet, a large-scale image dataset comprising more than 1.3 million images belonging to more than 1000 categories. By using a pretrained EfficientNet network, we can leverage the general image processing skills learned from the ImageNet dataset and adapt them to the specific characteristics of astronomical time series. For this work, we made use only of the first part of the network that extracts the features until the last pooling layer, while we completely discarded the final fully connected layer that in EfficientNet works as classifier (see Fig. 2).

The k-NN ([Cover & Hart 1967](#)) algorithm is a non-parametric supervised learning method that classifies an object



**Fig. 2.** Architecture of a traditional CNN in which there is a sequence of convolutional and pooling layers, and after a final subsampling layer, there is a fully connected one that performs the classification or regression task. In this specific work, we removed the fully connected part, and we used only the feature extraction, which is performed up to the last pooling layer.

based on the label of its  $k$  closest neighbors through a majority vote. In our case, we represented images by their features extracted from the EfficientNet network and measured the Euclidean distance of an image to its closest neighbor in this feature space ( $k = 1$ ). This distance reflects the similarity to the other images, and a large distance indicates a possible anomaly.

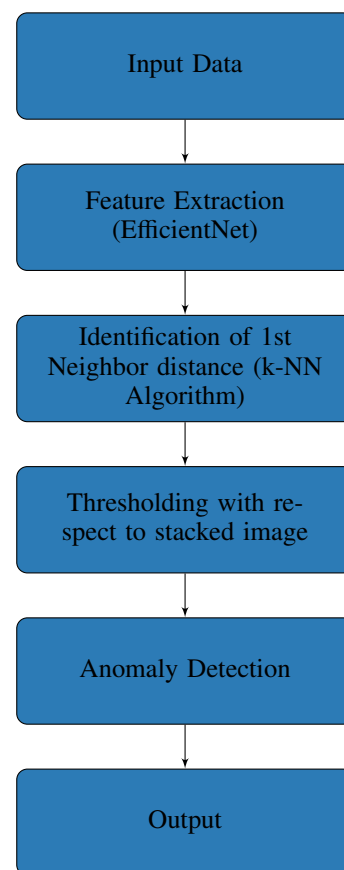
Since we needed a threshold to distinguish a real anomaly from something that is simply less similar, we used a stacked image (see Sect. 2) as a reference for the distance calculation. The stacked image is one that minimizes the noise and the presence of defects or spurious features with respect to the individual epochs, and it results in a sharper and cleaner reference template. We could thus safely assume that the stacked image represents the normal state of the scene and that any image that deviates significantly from it is an outlier.

In practice, as depicted in Fig. 3, our method consists of the following steps:

- Input data preparation: time series data from astronomical observations consisting of multiple images of the same celestial object and its stacked image are collected.
- Feature extraction: a pre-trained EfficientNet network on ImageNet is used. EfficientNet is a CNN architecture that efficiently scales depth, width, and resolution. The features are collected from input images using the EfficientNet as a feature extractor.
- Identification of first neighbor: the  $k$ -NN algorithm ( $k=1$ ) is applied to measure the Euclidean distance among features extracted by the EfficientNet network.
- Thresholding: the threshold for anomaly detection is identified. We use a stacked image as a reference for distance calculation.
- Anomaly detection: if the distance from the first neighbor is above the predefined threshold, the image is flagged as a potential outlier. We identify images deviating significantly from the normal state represented by the stacked image.
- Output: a list of potential outlier images identified during the anomaly detection process is produced.

## 4. Experiments and results

We decided to perform two testing campaigns, one on a larger dataset in order to have a wider case study and one on a smaller dataset, as we could then manually verify the results in a more

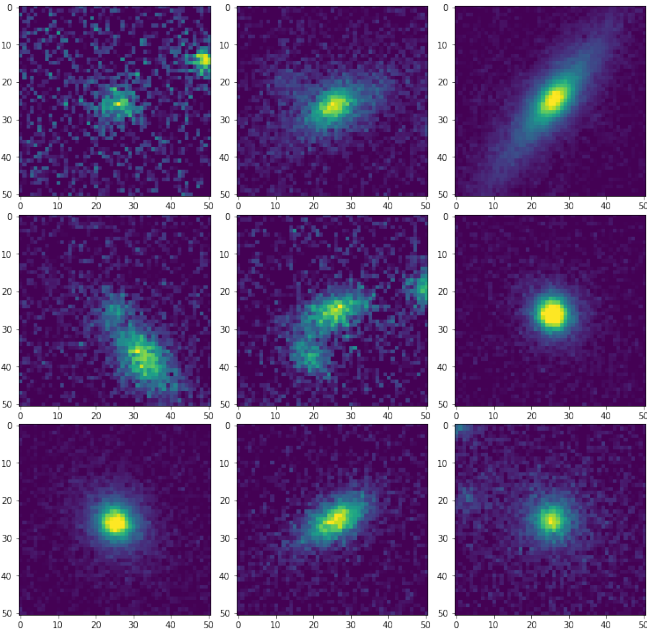


**Fig. 3.** Schematic description of the algorithm.

detailed way. We decided to use as the larger dataset the Unlabel Set and as the smaller dataset the Label Set, both derived from De Cicco et al. (2021).

### 4.1. Large dataset

We conducted a thorough analysis of the results obtained from applying our algorithm to the Unlabel Set dataset by De Cicco et al. (2021). The dataset comprises 17 995 sources observed multiple times (Table 1), yielding a total of 989 725 individual observations.



**Fig. 4.** Examples of problematic objects from epoch 32. The full set of sources flagged as problematic in epoch 32 is available in Fig. A.1.

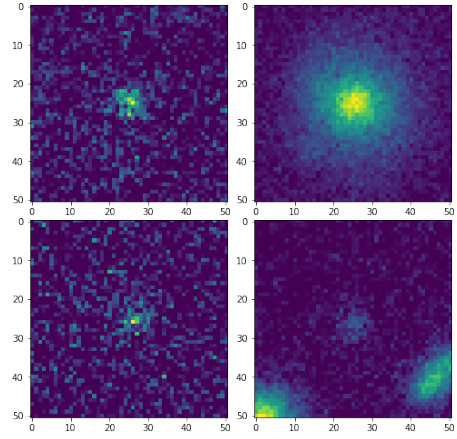
As a first step, we used the same procedure of [De Cicco et al. \(2019\)](#) to expunge bad epochs from the light curves, applying a sigma-clipping algorithm with a threshold of five times the value of the normalized median absolute deviation (NMAD) from the median magnitude of each source. This conservative threshold enabled us to exclude extreme outliers, preserving the most real variable events of astrophysical interest, such as AGN, supernovae, and other transients, provided the transient was not an extremely bright event in an otherwise quiescent galaxy (of which we have found no evidence in our data so far). This process refined the dataset to 914 156 epochs. Despite this initial cleaning step, the final light curves can still be affected by many photometric problems and artifacts, as discussed in Sect. 1, that may require additional ad hoc refinements to obtain fully reliable light curves. In our case, we directly applied the algorithm described in Sect. 3, finding 336 additional anomalous epochs.

Upon initial analysis, we observed that a specific epoch, namely epoch 32, according to the nomenclature provided in Table 1, exhibited 44 problematic images. A swift inspection revealed them to be images with a low S/N, as this is one of the epochs with the worst seeing values ( $1.18''$ , see Table 1, Figs. 4 and A.1).

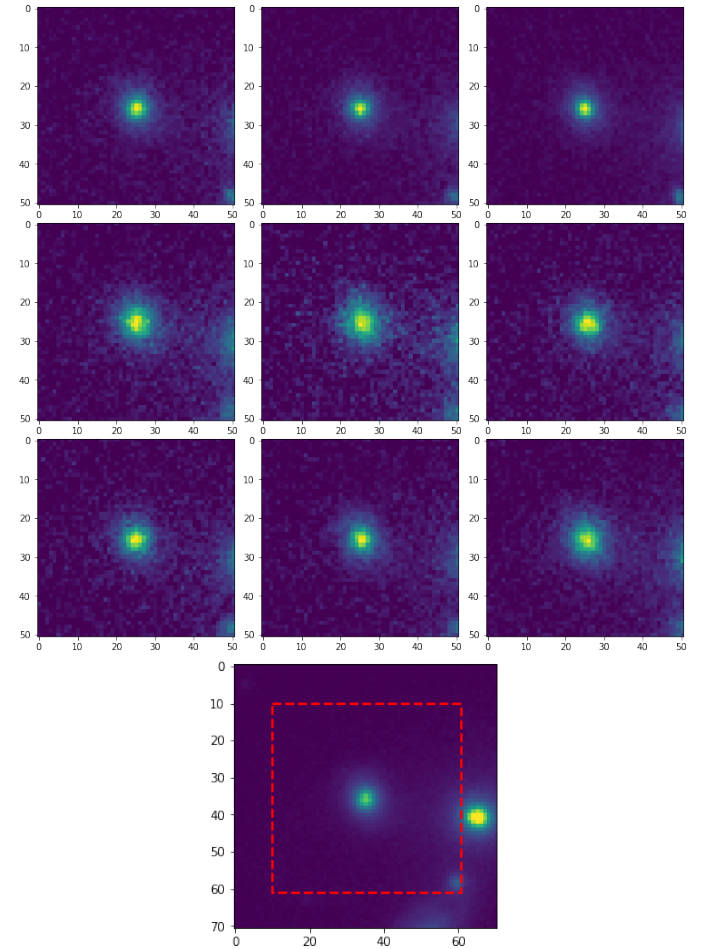
On four occasions, epoch 34 produced outliers linked to a low S/N (see Fig. 5). Epoch 34 actually has a worse seeing (i.e.,  $1.28''$ ) than Epoch 32.

Another noteworthy case concerns source 23 560, where nine out of 54 epochs were identified as anomalous, likely due to the contamination from a bright neighbor (see Fig. 6). In fact, source 23 560 has a bright companion that can be seen in the bottom panel of Fig. 6, so we are in the presence of a configuration that, due also to the fluctuation in the point spread function, affects the companion outskirts that contaminate this source and is spotted by our algorithm.

Twelve additional epochs showed contamination because of defects from electronics or saturated neighbors. Even when the central object flux does not seem directly affected, the algorithm background estimation might be distorted (Fig. 7).



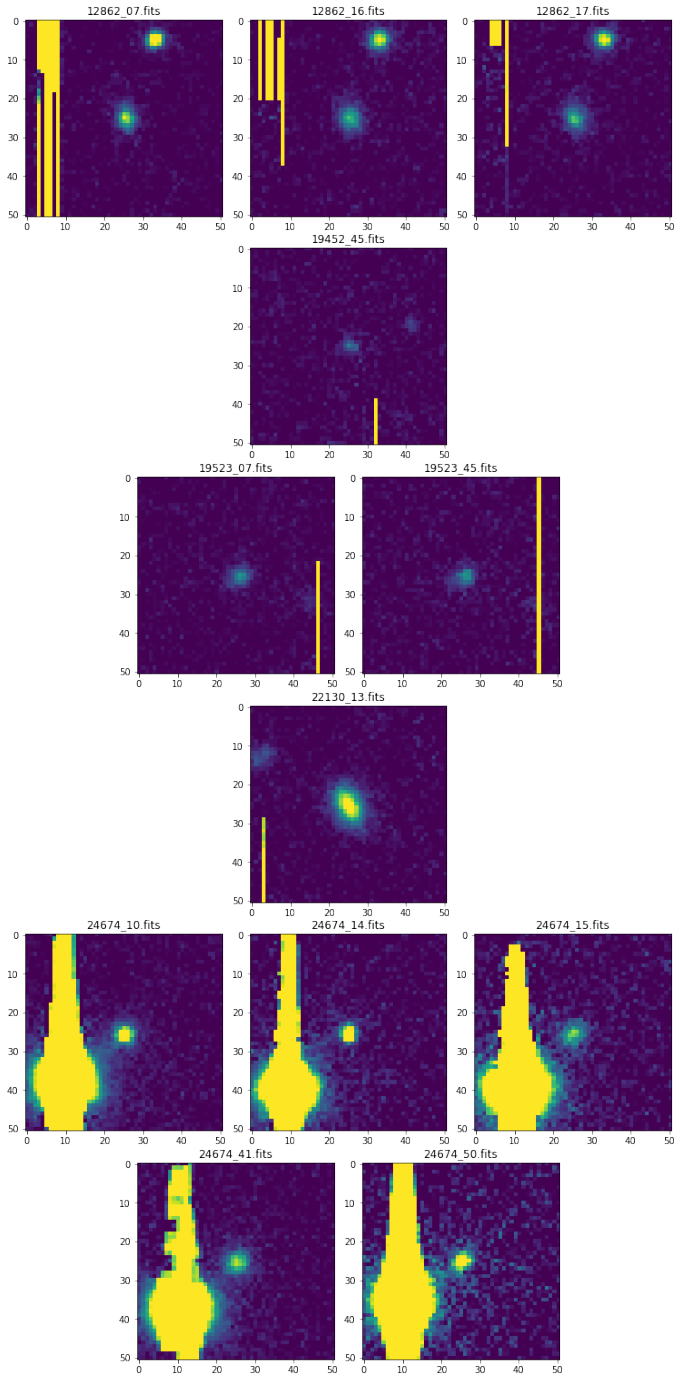
**Fig. 5.** Sources flagged as outliers in epoch 34 due to low S/N with regard to the reference epoch.



**Fig. 6.** Thumbnails of the flagged epochs for source 23 560 (including epoch 32, which was excluded from Fig. A.1). The last panel is a broader view of the stacked image showing the nearby contaminant; the size of the previous thumbnails is shown by the red dashed square.

Additionally, we identified 185 instances of astronomical tracks (asteroids, satellites, etc.) that, while not directly influencing the central source flux, could distort background estimation and thus the magnitude determination (Figs. 8 and A.2).

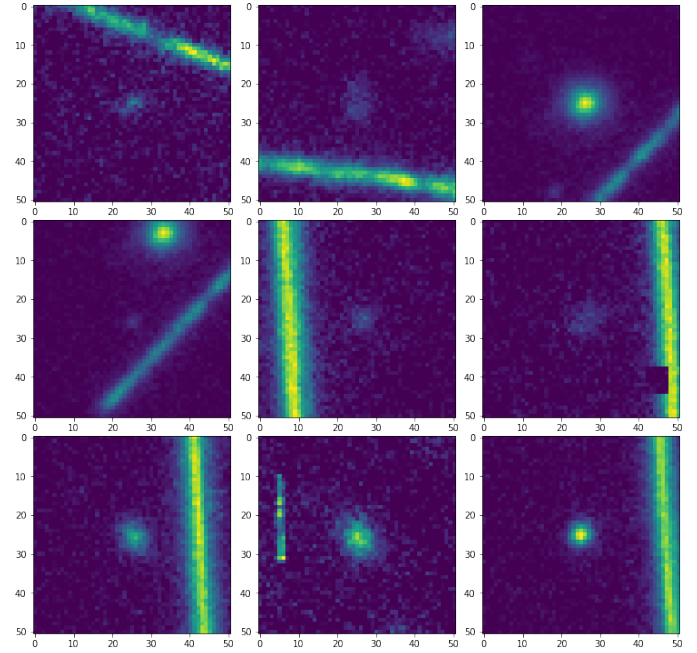
In three cases, an epoch was flagged due to the appearance of a secondary object in a single epoch (Fig. 9). While this is



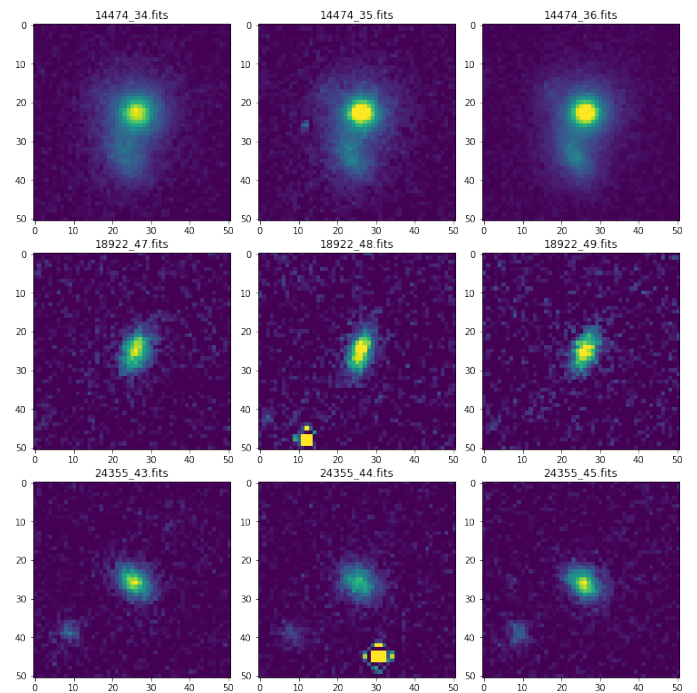
**Fig. 7.** Objects with a bright source nearby. Each row corresponds to a different object with some flagged epochs, with the exception of the last two lines, as they correspond to the same object.

an intriguing collateral study, it is unlikely to impact variability measurements; nonetheless, the algorithm correctly identified them as outliers.

The most significant finding pertains to 52 events where source measurements were undoubtedly influenced to some extent by tracks or other issues (Figs. 10, A.3, and A.4). These epochs should be excluded from any further study involving these objects. In particular, we would like to emphasize that, with some exceptions, those epochs could not be spotted with a different threshold of the sigma clip since they lie in the core of the distribution of the light curve despite being spoiled. There are

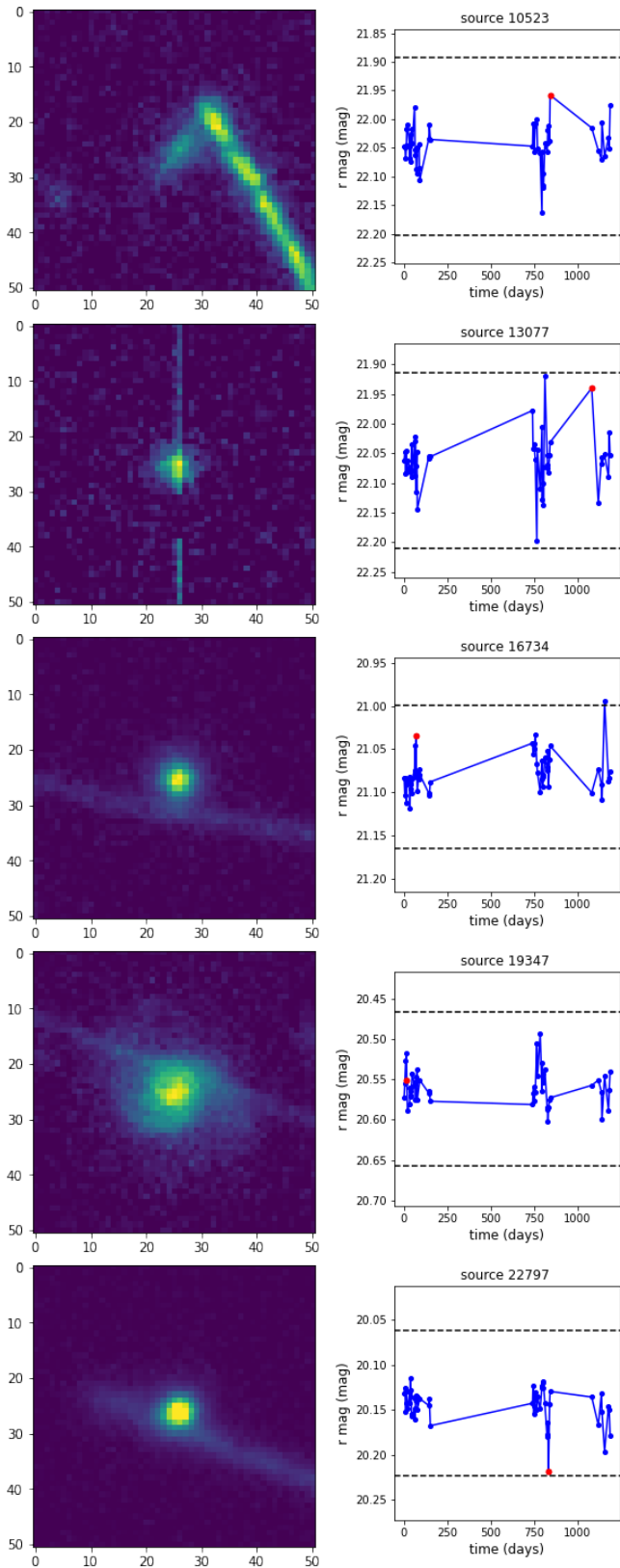


**Fig. 8.** Some examples of objects with tracks. (See Fig. A.2 for the whole sample.)

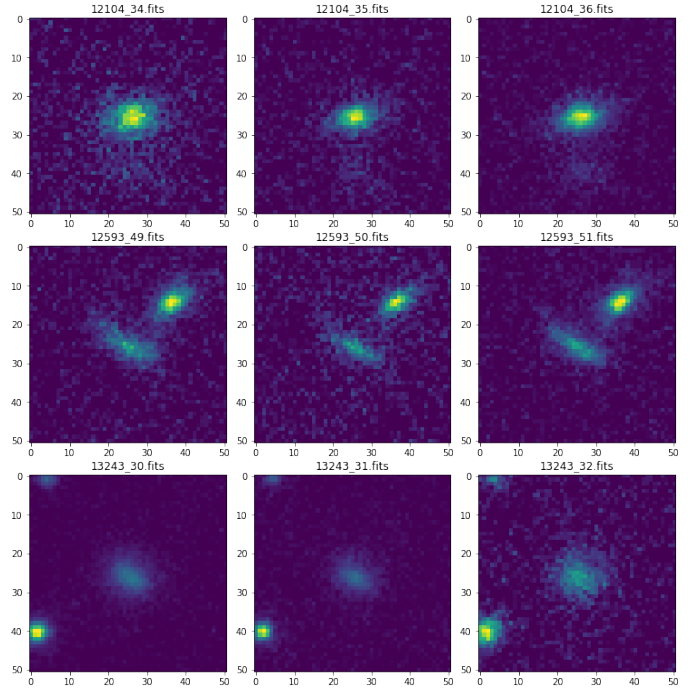


**Fig. 9.** Sources with an additional object that appears nearby. For each object, the epoch flagged by the method is shown in the central column. The epoch before and the one after are also shown in the first and last column, respectively. For the first object, the new object detection corresponds to the faint object located approximately at (10,25).

some cases, such as the last line of Fig. 10, where although there is an additional flux coming from a track, the measured flux from the pipeline is lower. In this specific case, we observed that due to the track, the pipeline provided a wrong estimate of the centroid of the object, and therefore the aperture magnitude does not contain all the flux of the actual source. Since our method does



**Fig. 10.** Some of the problematic objects identified by our method. In most cases, if a track appears, it is closer to the object with respect to the aperture photometry, so it slightly affects the flux measurement. In the left column, one can see the image as identified by our algorithm, while in the right we report the light curve with the flagged epoch marked with red. The horizontal lines represent the threshold of the sigma clip. (See Figs. A.3 and A.4 for the whole sample.)



**Fig. 11.** Three of the objects flagged as anomalies for which we do not have a proper motivation. Some of them can still be low S/N objects. For each source, the epoch flagged is the central one, while on the left and on the right, we have the previous and the next epoch, respectively. (See Figs. A.5 and A.6 for the whole sample.)

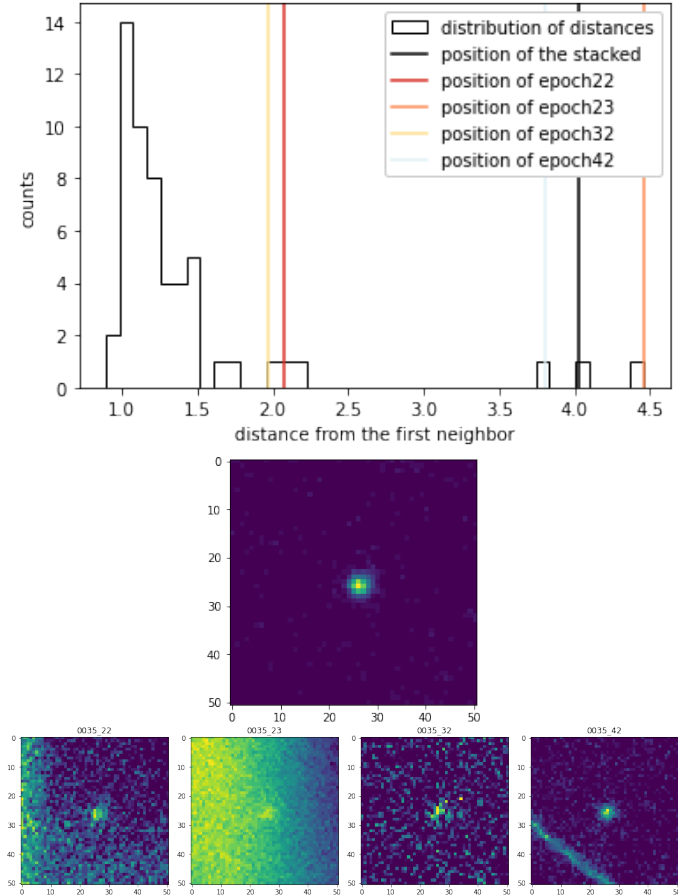
not rely on centroid measurements but makes use of the original images, it is able to identify these outliers anyway.

Finally there remain 27 epochs for which, despite being flagged as outliers, a clear interpretation could not be found. They represent approximately 8% of the flagged sources (that corresponds to 0.003% of the whole set of images presented to the algorithm) and could be considered as false positives identified by our algorithm (Figs. 11, A.5, and A.6). However, we observed that a significant portion (approximately 22% of these cases) belonged to one of the two epochs with the best seeing conditions (epochs 14 and 6, with seeing of 0.5'' and 0.58'' respectively; see Table 1). This suggests that given the exceptionally good seeing conditions, these objects exhibit even greater dissimilarities than others when compared to the stacked image. However, considering the minimal information loss (0.003% relative to the original dataset), we find this result to be more than acceptable.

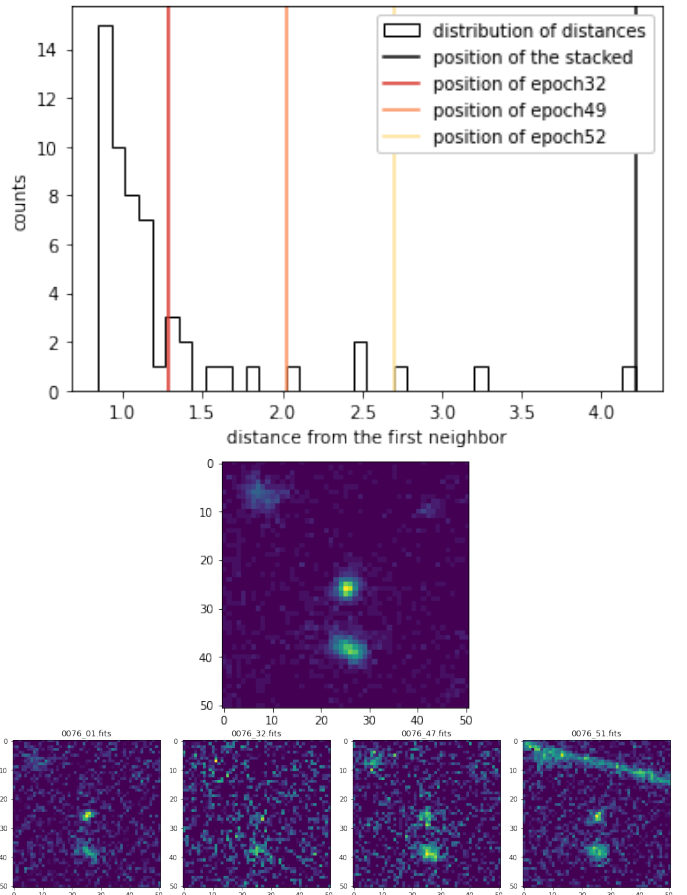
Finally, we performed a test on all 989 725 individual observations, including epochs rejected by the sigma-clipping criterion, which resulted in 1715 flagged images. A large part of these are already included in the ~85 000 objects discarded by the sigma clipping, while the majority of the remaining objects are the same ones selected by our approach after applying the sigma clipping. It is clear, however, that our algorithm is not capable of reproducing the sigma-clipping results, and therefore it is better to use it after the preliminary cleaning procedure as a further refinement in order to flag and eliminate most of the remaining outliers.

#### 4.2. Small dataset

We then transitioned to utilizing the Label Set dataset. This smaller dataset enabled us to visually explore data filtered by



**Fig. 12.** Source 0035. First row: histogram of the distances from the stacked image. Second row: stacked image. Last row: epochs 22, 23, 32, and 42.



**Fig. 13.** Source 0076. First row: histogram of the distances. Second row: stacked image. Last row: epochs 1, 32, 47, and 51.

sigma clipping. It was derived by De Cicco et al. (2021) and consists of all the objects in the field for which a classification of the sources, in terms of star, galaxy, and AGN, is available. The size of this dataset made it an ideal case for testing for any possible issue with our new algorithm. Our method did not identify all the anomalies in this dataset, comprising 2675 sources and a total of 147 125 individual observations. Post sigma clipping, this number decreased to 140 920, with 6205 epochs eliminated by the sigma clip.

By executing our algorithm on the complete Label Set dataset, we identified 289 anomalies, of which 202 were already flagged by the sigma clip. For the anomalies not identified by the sigma clip, a significant number was found to be tracks, in some cases overlapping the source.

Our focus then shifted to the 6003 sources that were clipped but not identified by our algorithm. Notably, these sources did not exhibit images entirely distorted by artifacts that we know are present based on the unfiltered dataset, suggesting both methods effectively detect and filter such image-related issues.

However, there are some cases where something was overlooked, although we expected our algorithm to easily spot the epoch as anomalous. We explore two examples of the few cases where our algorithm did not identify the problem.

For source 0035, by manually inspecting the light curve, we identified the following issue: In epochs 22 and 23, there is a large contamination coming from a defect of the image on the left, while in epoch 42 there is a track. In addition, there are

epochs with a lower S/N (for instance epoch 32). The algorithm spotted epoch 23, where the contamination is bigger, but completely missed the two other occurrences. By looking at the histogram of the distances (see Fig. 12) as computed by our algorithm, we noticed that all the problematic epochs were in fact detected as outliers of the distribution but are simply not different enough to meet our exclusion criteria.

Source 0076 exhibits a track in epoch 51. In Fig. 13, we show two more epochs with epochs 51 and 32 for comparison. From the histogram, it appears that the track, although being anomalous, is considered more similar to the stacked image; we spotted a similar behavior in other sources. In all of these cases, the source with the track is outside the distribution of standard epochs, but the stacked image is still above it. This effect could be mitigated by considering a “weirdness” threshold. Regardless of the position of the stacked image, the image was flagged as to be rejected, and clearly this, on one hand, reduces the number of wrong epochs not flagged, but, on the other hand, it increases the number of good epochs rejected. Although we added this possibility to the code, we consider that performing the sigma clip before implementing our algorithm provides the best results.

## 5. Discussion and conclusion

In this study, we propose a novel method for detecting outliers in astronomical time series, using VST monitoring campaigns of the COSMOS field as a test bed. Astronomical time series,



crucial for understanding celestial phenomena, present challenges such as noise, gaps, contaminants, and artifacts. Our approach, which falls under so-called transfer learning methods, combines a deep neural network, specifically an EfficientNet network trained on ImageNet, with a k-NN algorithm. The EfficientNet network serves as a feature extractor, and the k-NN algorithm measures the distance in the feature space to identify potential outliers. Anomalies are flagged based on the distance from the first neighbor and comparison with the same distance obtained using the stacked image, representing the normal state of the scene that has, by construction, a higher S/N and hence represents an outlier by definition.

Our experiments were separated into two cases extracted from the same dataset, one involving a larger dataset (Unlabel Set) and one involving a smaller dataset (Label Set), in order to verify the effectiveness of our method in detecting various anomalies, including a low S/N, contamination from neighboring sources, data reduction errors, bright objects affecting background estimation, and tracks. The performance of the algorithm was evaluated in comparison to a classic sigma-clipping procedure commonly used in astronomical data analysis.

In the case of the larger dataset (Unlabel Set), our method identified anomalies that were missed by the sigma-clipping algorithm, including instances of a low S/N, contamination, and tracks. Additionally, it successfully flagged epochs affected by bright saturated objects, data reduction errors, and off-center tracks (regarding the latter aspects, the method seems to be very efficient). Although there were some epochs that were flagged by the method for which we do not have a proper explanation, even when considering all of them as false positives, they are negligible in number with respect to the whole dataset.

In the case of the smaller dataset (Label Set), we did the opposite by looking at the anomalies identified by the sigma clip that were not flagged by our algorithm. In most cases, the distribution of the distances showed that the epochs that should clearly be removed visually, although deviating from the distribution, are below the threshold identified by the stacked image. This suggests that both of the analyses need to be performed, that is, the sigma clipping first and then a refinement with the method proposed in this work.

One key aspect of our method is that unlike traditional approaches, such as sigma clipping, that focus on identifying outliers in a limited parameter space (e.g., extreme magnitude variations in the light curve), our method examines the overall structure of the images across the time series, considering the relationships and patterns in the feature space. This is particularly evident in the light curves presented in the appendix (see Fig. A.4), where anomalies flagged by our method go beyond the typical peak-detection scenarios. Peaks alone do not capture the diverse range of anomalies present in astronomical time series, and our method provides a more comprehensive approach to outlier detection. In substance, a simple adjustment of the sigma-clipping parameters would not suffice to capture the nuanced anomalies identified by our method. This highlights the complementary nature of our approach, offering a distinct and valuable perspective in the realm of artifact detection in astronomical time series.

The presence of defected epochs in astronomical time series can significantly impact statistical indicators commonly derived from light curves. One notable example is the pair slope, a metric used for instance in De Cicco et al. (2021). The pair slope measures the trend of variability between consecutive data points in a light curve, providing insights into the underlying astrophysical processes. However, the accuracy of such indicators

is compromised when defected epochs are present, and these epochs may hamper the ability to detect and characterize a variety of important astrophysical events, such as tidal disruption events, supernovae, AGN, and blazars. Our method, by effectively identifying and flagging these anomalies, contributes to the preservation of the integrity of statistical measures derived from light curves, ensuring that the results are more reliable and reflective of the true astrophysical phenomena.

As a further development in the future, we plan to explore different pretrained networks, such as vision transformers Dosovitskiy et al. 2020. Additionally, a possible improvement could be found by exploring adaptive thresholding methods based on the distribution of the distance meant to work with the threshold determined by the stacked image in order to make the algorithm more robust to variations in data characteristics.

In addition to the effectiveness of the proposed method in detecting outliers in astronomical time series, it is noteworthy to discuss the computational efficiency and parallelization potential. The nature of the algorithm lends itself to an “embarrassingly” parallel paradigm. Each time series is entirely independent from the others, allowing for parallel execution on separate computational nodes. This parallelization capability makes the method well suited for deployment on high-performance computing clusters or distributed computing environments.

During our testing phase, the algorithm demonstrated impressive efficiency even on a machine without a GPU – we utilized an Intel® Core™ i9-10980XE CPU @ 3.00GHz<sup>1</sup>. On average, processing each time series consisting of a maximum of 54 epochs took less than 1 second. It is important to highlight that this timing was achieved on a CPU, and the use of a GPU is expected to further accelerate the processing time.

The scalability and adaptability of the algorithm to parallel processing environments make it a promising solution to efficiently handle large-scale astronomical datasets, such as the ones that will be obtained from the LSST. Future implementations may leverage GPU resources to achieve even faster processing times, which is particularly beneficial when dealing with extensive datasets generated by modern astronomical surveys.

*Acknowledgements.* D.D. acknowledges PON R&I 2021, CUP E65F21002880003. D.D. and M.P. also acknowledge the financial contribution from PRIN-MIUR 2022 and from the Timedomes grant within the “INAF 2023 Finanziamento della Ricerca Fondamentale”. M.B. and S.C. acknowledge the ASI-INAF TI agreement, 2018-23-HH.0 “Attività scientifica per la missione Euclid - fase D” Topcat (Taylor 2005) and STILTS (Taylor 2006) have been used for this work. Some of the resources from Stutz (2022) has been used for this work. The k-NN used in this work is part of the Scikit package (Pedregosa et al. 2011).

## References

- Aigrain, S., & Foreman-Mackey, D. 2023, *ARA&A*, 61, 329
- Awang Iskandar, D. N. F., Zijlstra, A. A., McDonald, I., et al. 2020, *Galaxies*, 8, 88
- Botticella, M. T., Cappellaro, E., Greggio, L., et al. 2017, *A&A*, 598, A50
- Capaccioli, M., & Schipani, P. 2011, *The Messenger*, 146, 2
- Cappellaro, E., Botticella, M. T., Pignata, G., et al. 2015, *A&A*, 584, A62
- Cover, T., & Hart, P. 1967, *IEEE Trans. Inform. Theory*, 13, 21
- De Cicco, D., Paolillo, M., Covone, G., et al. 2015, *A&A*, 574, A112
- De Cicco, D., Paolillo, M., Falocco, S., et al. 2019, *A&A*, 627, A33
- De Cicco, D., Bauer, F. E., Paolillo, M., et al. 2021, *A&A*, 645, A103
- De Cicco, D., Bauer, F. E., Paolillo, M., et al. 2022, *A&A*, 664, A117
- Deng, J., Dong, W., Socher, R., et al. 2009, in 2009 IEEE conference on computer vision and pattern recognition, IEEE, 248

<sup>1</sup> <https://ark.intel.com/content/www/us/en/ark/products/198017/intel-core-i9-10980xe-extreme-edition-processor-24-75m-cache-3-00-ghz.html>

- Ding, Y., Sohn, J. H., Kawczynski, M. G., et al. 2019, *Radiology*, **290**, 456
- Doorenbos, L., Torbaniuk, O., Cavuoti, S., et al. 2022, *A&A*, **666**, A171
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2020, arXiv e-prints [arXiv:2010.11929]
- Esteva, A., Kuprel, B., Novoa, R. A., et al. 2017, *Nature*, **542**, 115
- Falocco, S., Paolillo, M., Covone, G., et al. 2015, *A&A*, **579**, A115
- Fu, L., Liu, D., Radovich, M., et al. 2018, *MNRAS*, **479**, 3858
- Grado, A., Capaccioli, M., Limatola, L., & Getman, F. 2012, *Mem. Soc. Astron. Ital. Suppl.*, **19**, 362
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, *ApJ*, **873**, 111
- Kim, Y. J., Bae, J. P., Chung, J.-W., et al. 2021, *Sci. Rep.*, **11**, 3605
- LeCun, Y., Bengio, Y., & Hinton, G. 2015, *Nature*, **521**, 436
- Lecun, Y., Boser, B., Denker, J. S., et al. 1990, in *Advances in Neural Information Processing Systems 2*, ed. D. S. Touretzky (USA: Morgan Kaufmann), 396
- Li, B., Yu, C., Hu, X., et al. 2018, in *Algorithms and Architectures for Parallel Processing*, ed. J. Vaidya & J. Li (Cham: Springer International Publishing), 284
- Liu, D., Deng, W., Fan, Z., et al. 2020, *MNRAS*, **493**, 3825
- Malz, A. I., Hložek, R., Allam, T., et al. 2019, *AJ*, **158**, 171
- Martinazzo, A., Espadoto, M., & Hirata, N. S. 2021, in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 4176
- Menegola, A., Fornaciali, M., Pires, R., et al. 2017, in *2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017)*, IEEE, 297
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, *J. Mach. Learn. Res.*, **12**, 2825
- Poulain, M., Paolillo, M., De Cicco, D., et al. 2020, *A&A*, **634**, A50
- Prima, B., & Bouhorma, M. 2020, *Remote Sensing Spatial Inform. Sci.*, **4443**, 343
- Scargle, J. D. 1997, in *Astronomical Time Series*, eds. D. Maoz, A. Sternberg, & E. M. Leibowitz (Dordrecht: Springer Netherlands), 1
- Stutz, D. 2022, Collection of LaTeX resources and examples, <https://github.com/davidstutz/latex-resources>.
- Tan, M., & Le, Q. V. 2019, arXiv e-prints [arXiv:1905.11946]
- Taylor, M. B. 2005, *ASP Conf. Ser.*, **347**, 29
- Taylor, M. B. 2006, *ASP Conf. Ser.*, **351**, 666
- Zou, M., & Zhong, Y. 2018, *Sensing Imaging*, **19**, 6

Appendix A: Complete figures

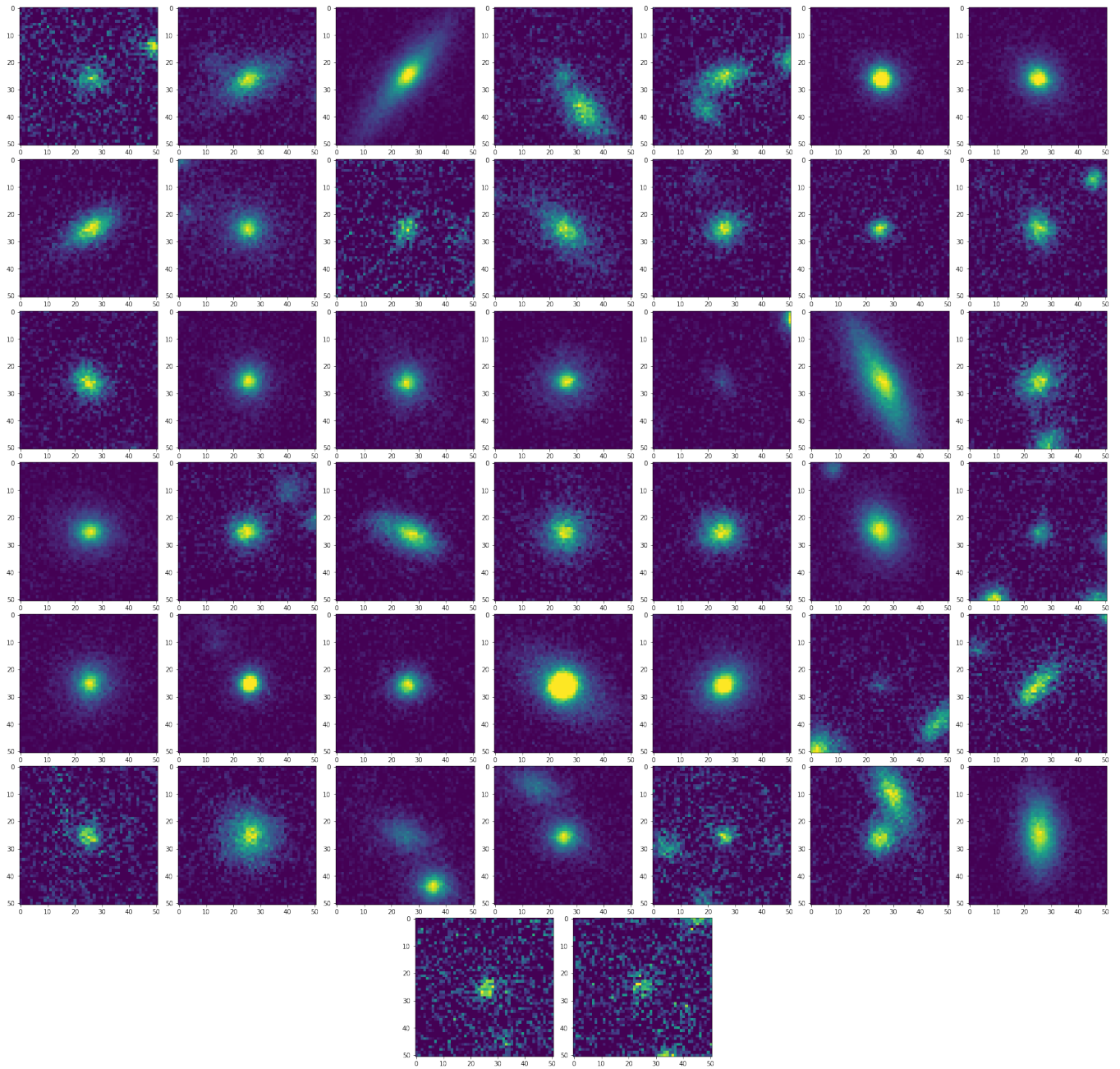


Fig. A.1: Full set of sources flagged as problematic in epoch 32.

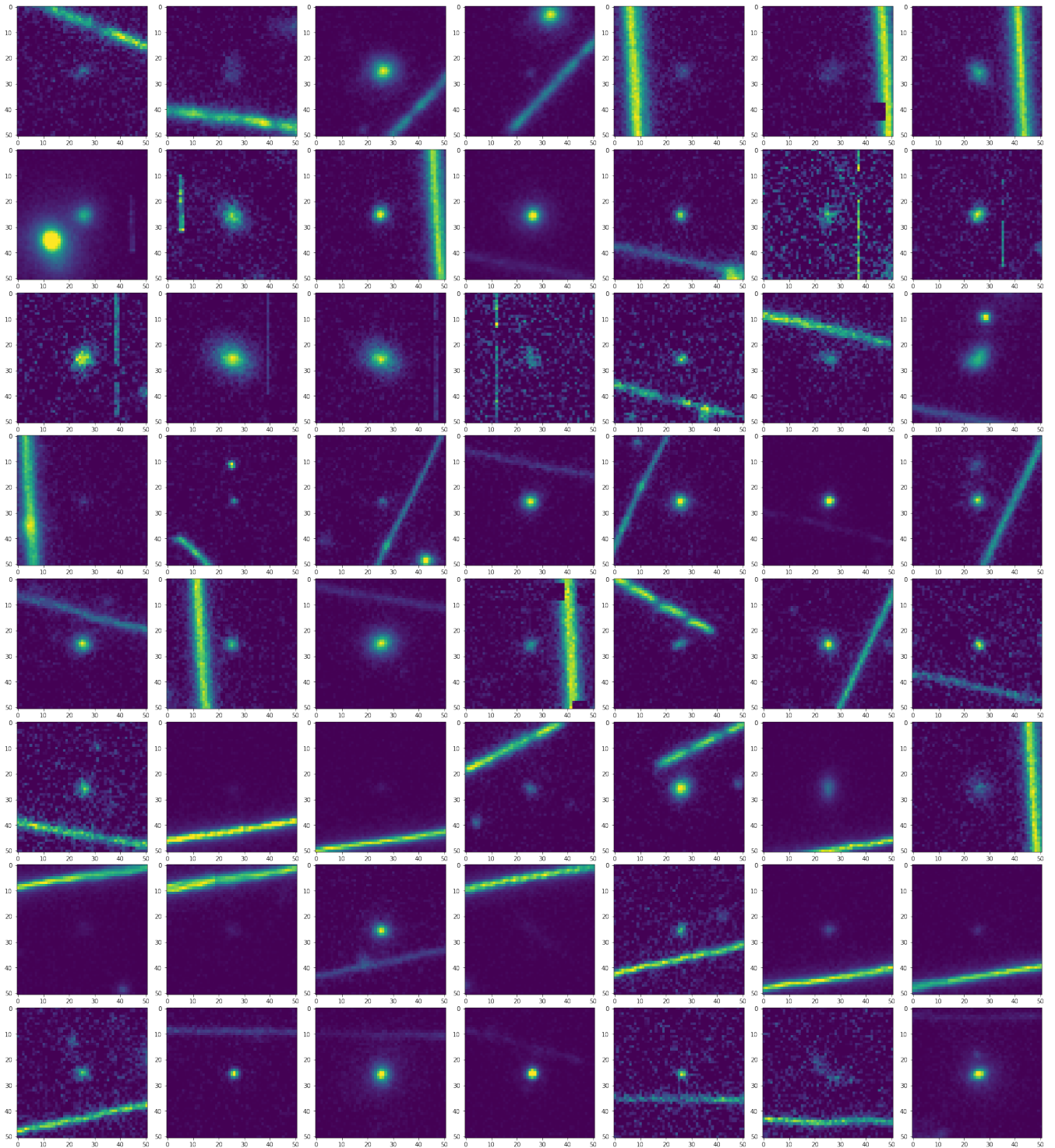


Fig. A.2: Objects with track.

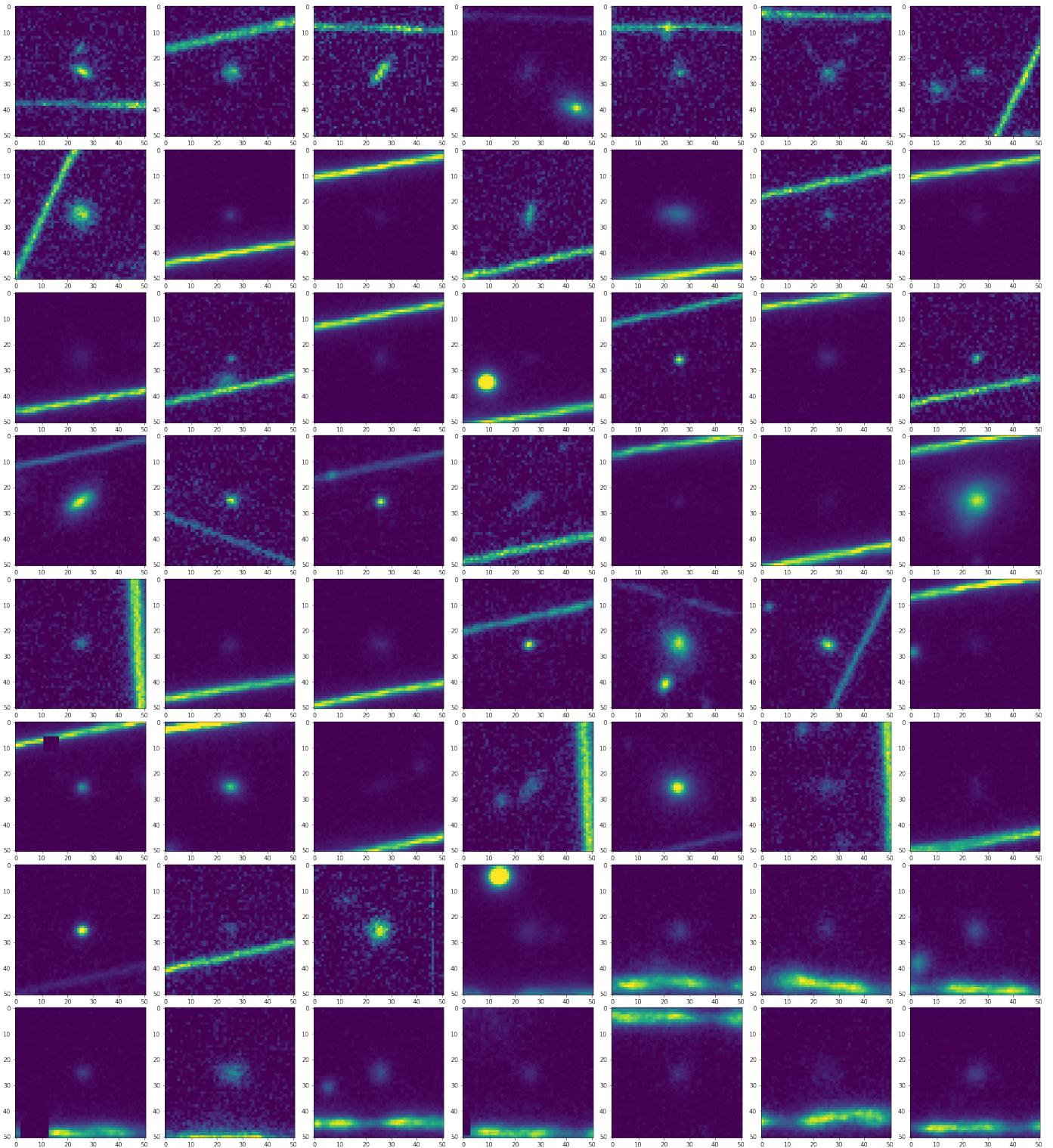


Fig. A.2: continued

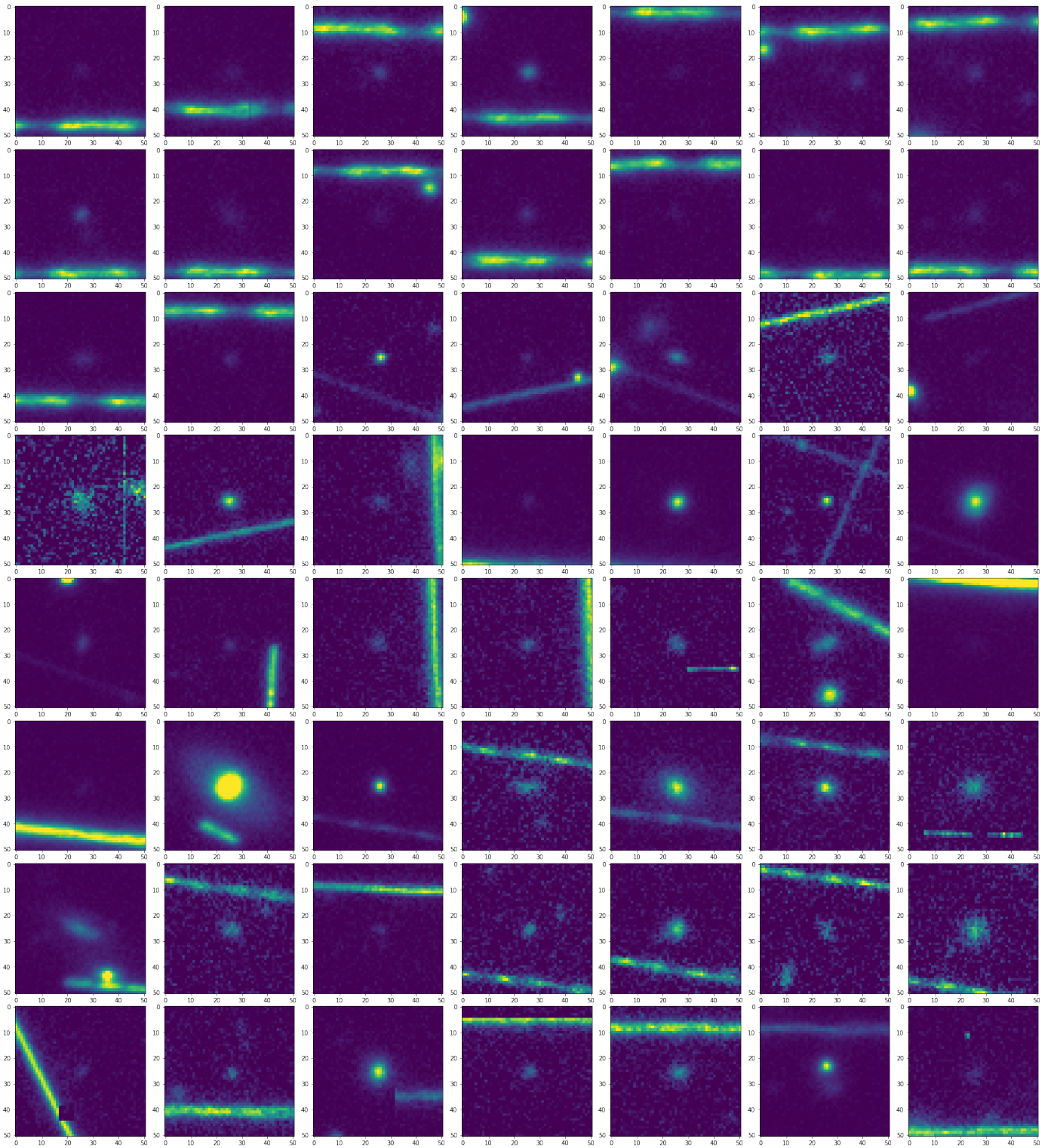


Fig. A.2: continued

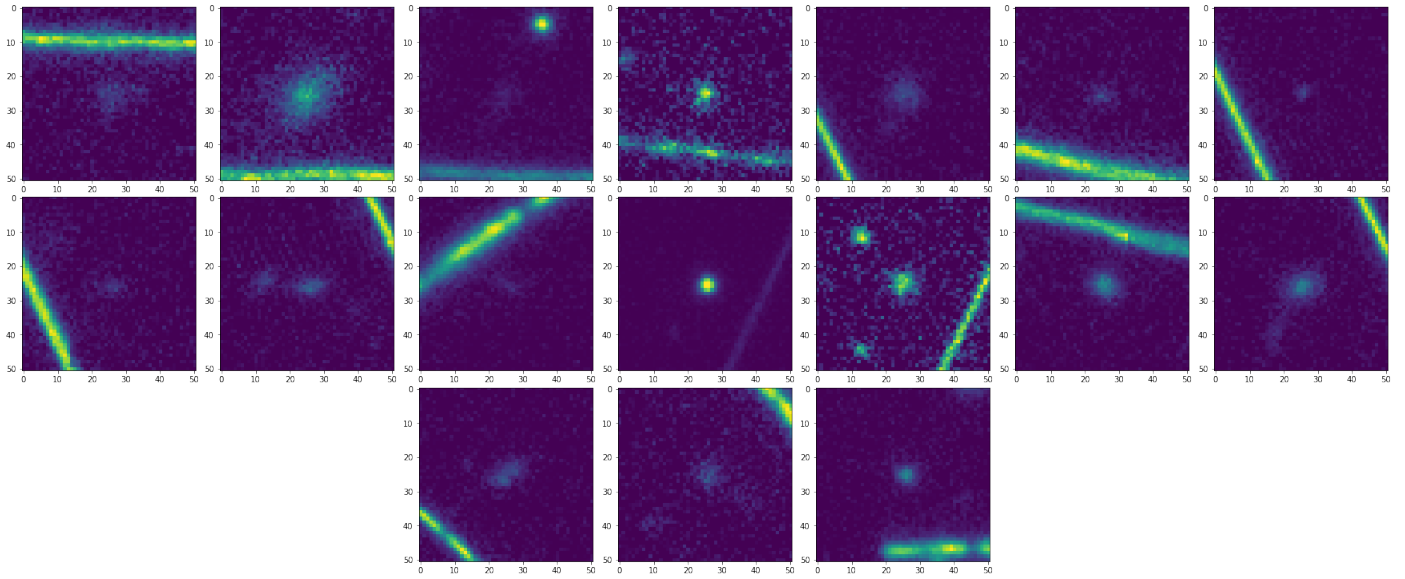


Fig. A.2: continued

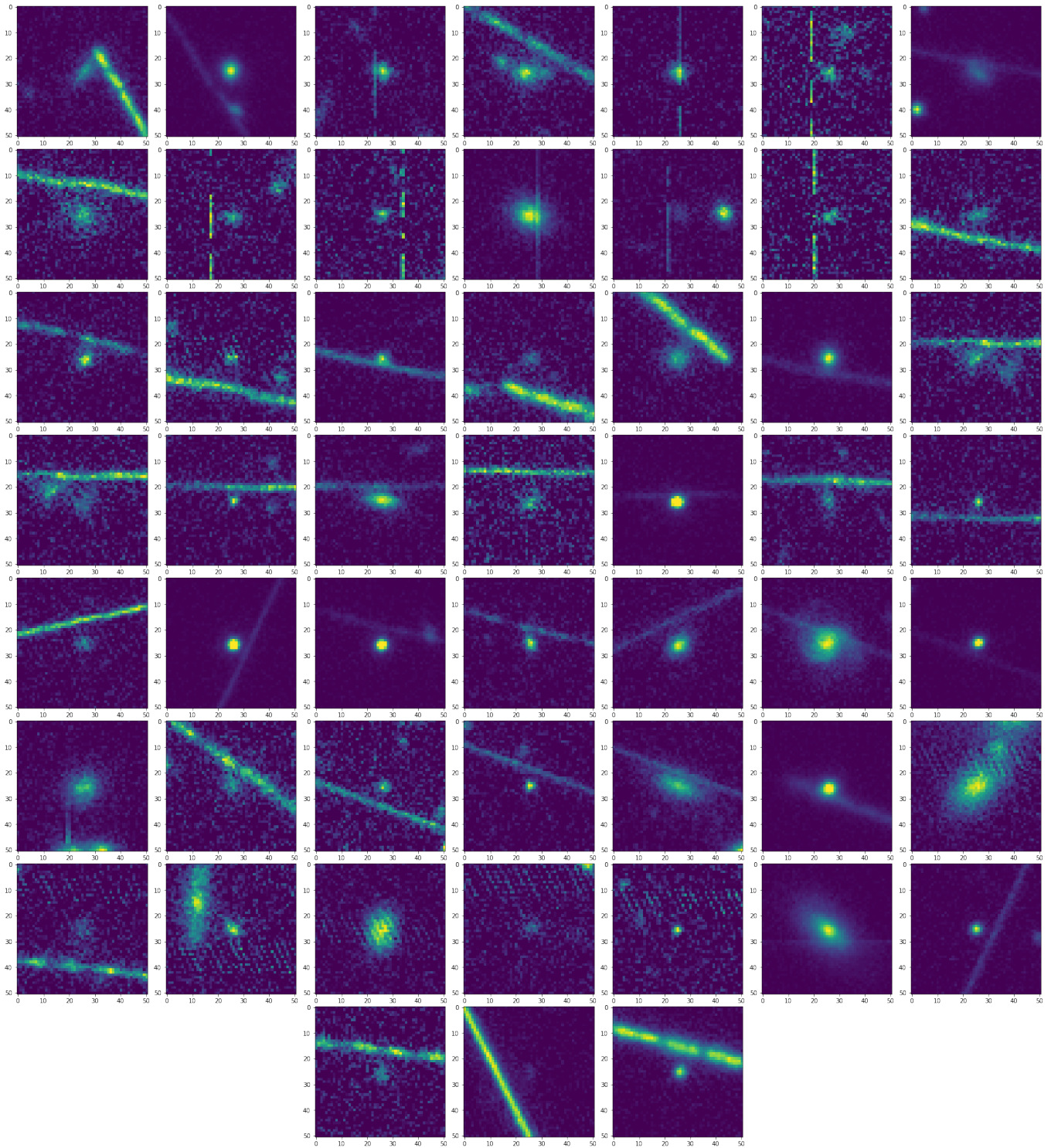


Fig. A.3: Problematic objects. In most of the cases, if a track appears, it is closer to the object with respect to the aperture photometry, so it slightly affects the flux measurement.



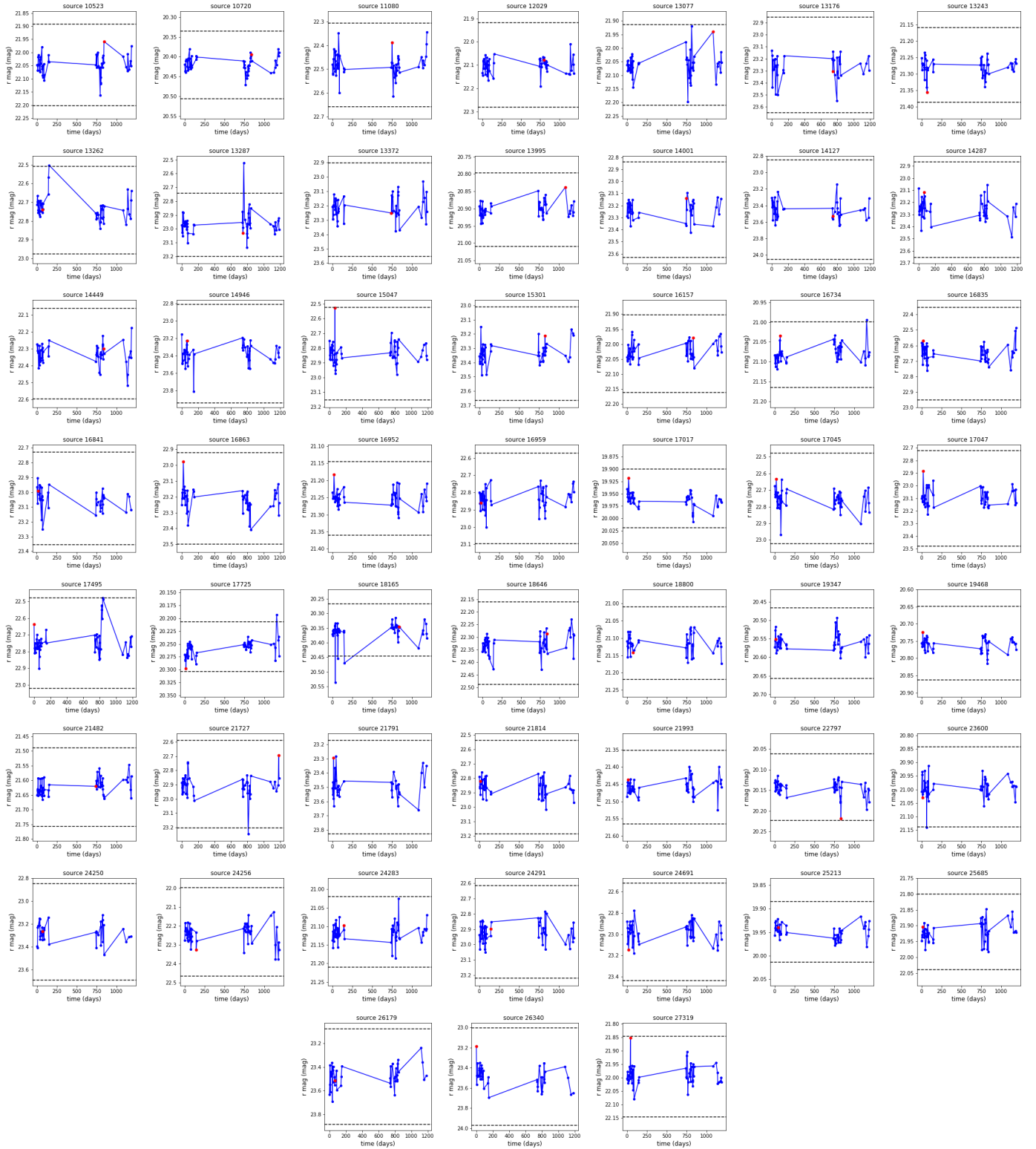


Fig. A.4: Light curves for the 52 problematic objects shown in Fig. A.3. The flagged epoch is marked in red. The horizontal lines represent the threshold of the sigma clip.

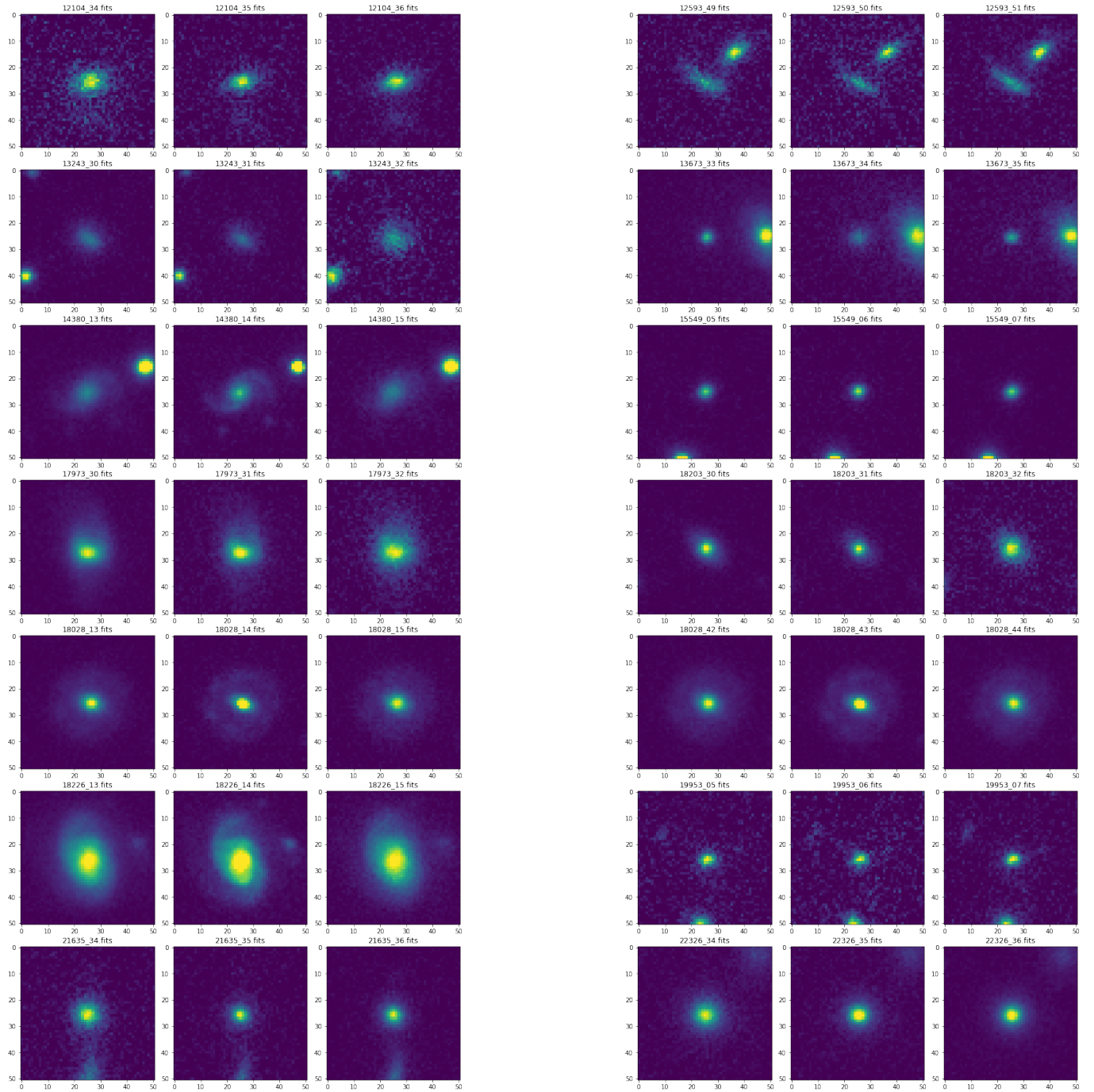


Fig. A.5: Objects flagged as anomalies for which we do not have a proper motivation. Some of the anomalies can still be low S/N objects. For each epoch flagged, we added the one before and the one after for comparison. In two cases (second line, left column and fourth line, right column), epoch 32 is selected as the “after” epoch.

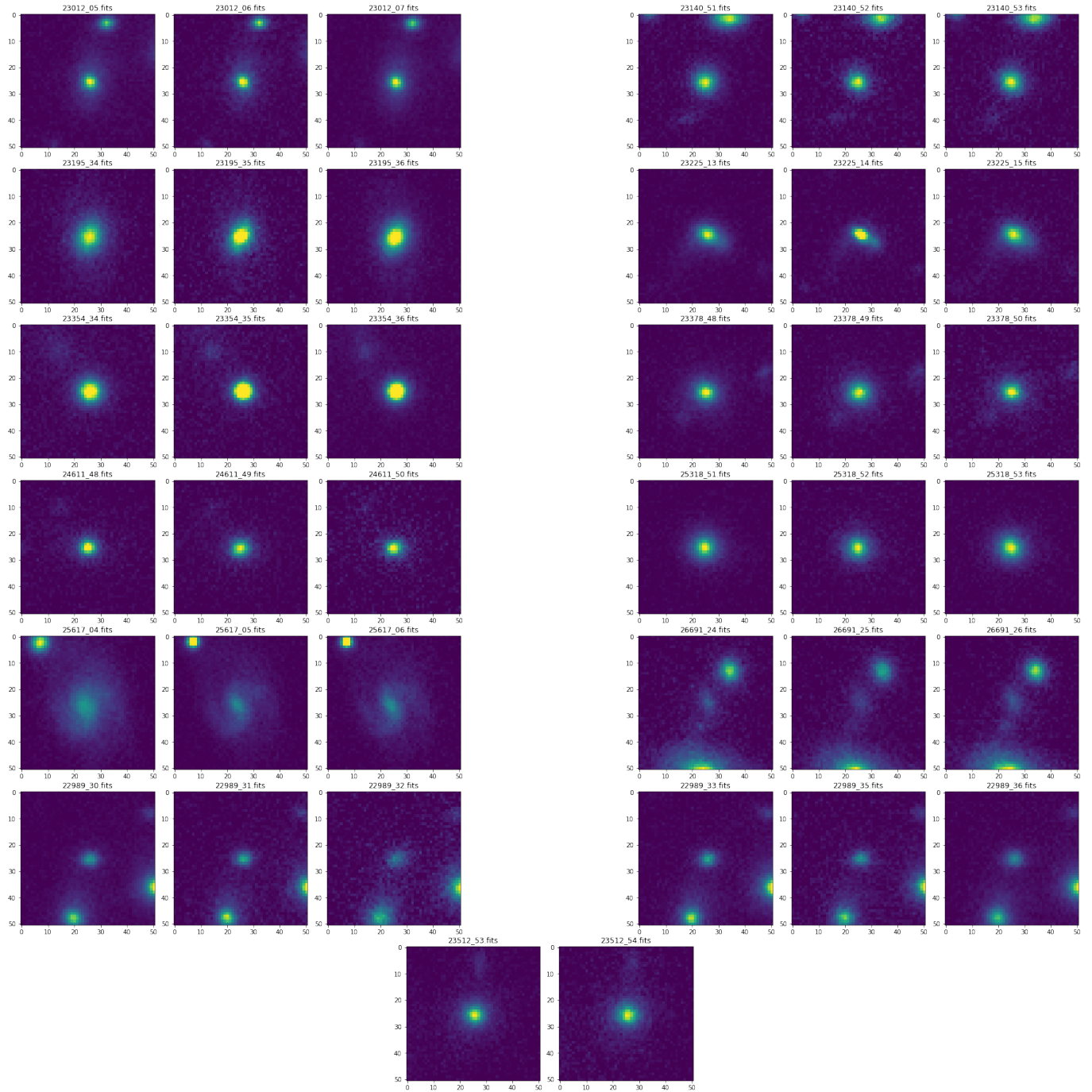


Fig. A.6: Objects flagged as anomalies for which we do not have a proper motivation. Some of the anomalies can still be low S/N objects. For each epoch flagged, we added the one before and the one after for comparison. In the last two lines, six epochs appear since the method selected more than one epoch (epochs 31 and 35; 34 was removed by the sigma clipping), and this causes a series (from epoch 30 to 36). In the last line, there are only two epochs since the selected one is the last.