This project is co-financed by the European Union

Grant Agreement No.: 824603
Call: H2020-SwafS-2018-1
Type of action: RIA
Starting date: 1/02/2019

App: final version of one simple device to characterise street lamps spectra

Coordinator: [Jaime Zamorano]
Quality reviewer: [Kat Austen]

Deliverable nature	Report
Dissemination level	Public
Work package and Task	D2.2
Contractual delivery date	07/2021
Actual delivery date	10/2021

Authors

Author name	Organization	E-Mail
Jaime Zamorano	**UCM**	**jzamorano@fis.ucm.es**
Rafael González	**UCM**	**rafael08@ucm.es**
Lucia García	**UCM**	**lucia_garcia@ucm.es**

Abstract	StreetSpectra is an ACTION pilot citizen science project whose smartphone app is designed mainly for mapping the location and nature of street lights. This report describes the final solution implemented by the StreetSpectra pilot, based on mobile applications, data gathering platforms, classification platforms and OpenData platforms for publishing. It also describes the sister CS project named AZOTEA, born during the 2020 confinement period.
Keywords	Public lighting - Light Pollution - Spectrum - Smartphone - DSLR - Citizen Science - Photometry

Disclaimer

The information, documentation and figures available in this deliverable, is written by the ACTION project consortium under EC grant agreement 824603 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

How to quote this document

J. Zamorano, R. González & L. García (2021), ACTION App: final version of one simple device to characterise street lamps spectra

TABLE OF CONTENTS

1. Introduction

This document describes the evolution of the StreetSpectra project from our initial version of a single mobile app to a more comprehensive project. Over the course of 4 months we have developed an extended architecture for StreetSpectra, which gathers data through the use of a robust third-party app (Epicollect 5) and leverages existing platforms and workflows to facilitate data analysis by citizens. In this document we explain the basics of the StreetSpectra project (an ACTION pilot citizen science project). We then describe the components of the final StreetSpectra solution. Finally we describe the current state of AZOTEA, a citizen science project born during the three months confinement period. For further reference and context:

- ACTION Deliverable D2.1 introduces StreetSpectra and describes the procedure to take images as well as a tutorial for image capture and spectra classification.
- ACTION Deliverable D2.10 describes the StreetSpectra final architecture and also introduces AZOTEA, a small CS project born during the confinement period.

2. Motivation for the StreetSpectra project

Street lamps are a significant contributor to light pollution. Read, for instance, the report 'An Investigation of LED Street Lighting's Impact on Sky Glow' (U.S. Department of Energy, 2017) or 'Street lighting has detrimental impacts on local insect populations' (Boyes et al. 2021 Science Advances DOI: 10.1126/sciadv.abi8322)

However the diversity of street lamp technology means that gaining a comprehensive overview of the problem is challenging. Read for instance: 'Evolution of Brightness and Color of the Night Sky in Madrid' (Robles et al. Remote Sens. 2021, 13(8), 1511; https://doi.org/10.3390/rs13081511). The StreetSpectra project provides easy-to-use methods based on accessible technology to enable citizens to map and classify street light technology.

Besides using expensive professional spectrophotometers, some useful information about the light emitted by street lamps can be obtained by using smartphone cameras, and other simple and inexpensive portable devices. However, few simple workflows for gathering this data exist. Our aim was to turn the smartphones into scientific instruments to analyze lamps colors and their spectra[1]. We will define a process and tools so that any citizen will be able to determine the kind of lamp installed on lampposts and its main characteristics. We have developed a very simple method to obtain the spectra of the lamps (StreetSpectra). This is a citizen science pilot study of the H2020-SwafS-2018-1-824603 EU project. More info about the project at https://actionproject.eu/

2.1 Street Lights as Pollution

Artificial light at night (ALAN) can cause light pollution that is a significant contributor to biodiversity loss and health problems, and also has consequences for climate change, among other unwanted impacts. The current transition to LEDs can sometimes lead to energy savings, but the blue excess of the light emitted by white LEDs is more likely to produce skyglow – the atmosphere scatters and reflects light back towards the Earth's surface. The blue component is also more damaging to plants and animals than old-style warm yellowish lights. The light that escapes to the atmosphere, due to an excess of illumination or a bad design of luminaires, is dispersed and some of this light is directed back to us. Thus, one of the undesired effects of light pollution is the brightening of the night sky and the loss of the starry skies in our cities. The artificial illumination of our streets is one of the major contributors to light pollution.

Besides a big amount of research papers dealing with the unwanted effects of Light Pollution, a comprehensive scenario of the problem is described (in plain lay citizens words) at the Science for Environmental Policy DG Environmental News Alert Service of the European Commission. For instance:

- The 'Dark Ecological Network': strategically tackling light pollution for biodiversity and people (2021)
- Nocturnal use of LEDs negatively affects freshwater microorganisms, (2019)
- Moth behaviour disrupted by street lighting, may affect pollination (2016)
- Artificial light at night—the impact on plants and ecology (2016)
- Switching to LED street lighting could alter urban bat behaviour (2016)
- Street lighting affects insect biodiversity (2012)

We are experiencing a major change in the technology used for street lighting and most of the high-pressure sodium (HPS), metal halide (MH) and mercury vapor (MV) lamps

[1] Spectra is used as a plural of spectrum.

are being replaced by light-emitting diode lamps (LEDs). Light pollution experts working across different fields have warned about the impact of blue artificial light at night both on human health, flora and fauna. Some of the LEDs used on the street are too white and contain a potential hazardous blue component. Read for example 'Long-Term Comparison of Attraction of Flying Insects to Streetlights after the Transition from Traditional Light Sources to Light-Emitting Diodes in Urban and Peri-Urban Settings' by Roy H. A. van Grunsven, Julia Becker, Stephanie Peter, Stefan Heller & Franz Hölker (https://www.mdpi.com/2071-1050/11/22/6198).

In several US cities, where the retrofit has been performed using too white LEDs, citizens are asking the authorities to re-introduce the old warm lights.

Figure 1. Image of Berlin at night taken by astronauts aboard the International Space Station (ISS) showing different illumination colors of the streets: white LEDs and orange HPS lamps. The North direction is on the right of the picture.

The replacement of street lights to LED technology is already taking place in Europe. Figure 1 shows a picture that has been taken by astronauts aboard the International Space Station (ISS035-E-17210, taken at 2013.04.06), and shows clearly the differences in lightning between areas in Berlin where such changes have occurred. The west of the city (up in the picture) is experiencing the change from gas lights, fluorescent and mercury vapor lights to LEDs while the east of Berlin lighting is mainly composed of sodium lamps (although also is switching to LEDs) (you can browse more examples at http://citiesatnight.org/). There is a vast diversity of lamps used as they belong to different companies and models, ranging from very white LEDs with CCT 6000K to warm ones with 2700K and even amber lights with lower CCT.

There is scarce information on the fraction of lighting technologies employed in our cities and also of the schedule of the retrofit changes. Location, type, number of sources and light pollution brightness are needed to feed into scientific models that describe the scattering in the atmosphere and the impact of light pollution at medium and long distances from the sources.

2.2 Citizens shining a light on street lighting

StreetSpectra invites citizens to join a project to help map some of these factors, to shine a light on the details of street lighting.

How could the citizens participate? In StreetSpectra, there are two different ways. They can take the *observers role*, participants take pictures using their own smartphone camera with a cheap diffraction grating and can determine and analyse the lamps' spectra. Usually all the lamps in a typical street are similar and it is not necessary to take a spectrum of every lamppost. Taking pictures of both ends of the street will do.

The other way to participate is to act as the *analyst role.* In this way, other citizens may examine - from the comfort of their homes - a collection of images taken by the *observers* and submit their classification results through the Internet.

The observations & classifications archive will contain pictures with the location of the lamps) as well as their type. The contributor, or any interested citizen or researcher, can browse this open data.

Besides gathering this important data, and the implications for scientific research into ALAN, the project is well suited for educational purposes and to raise awareness of light pollution.

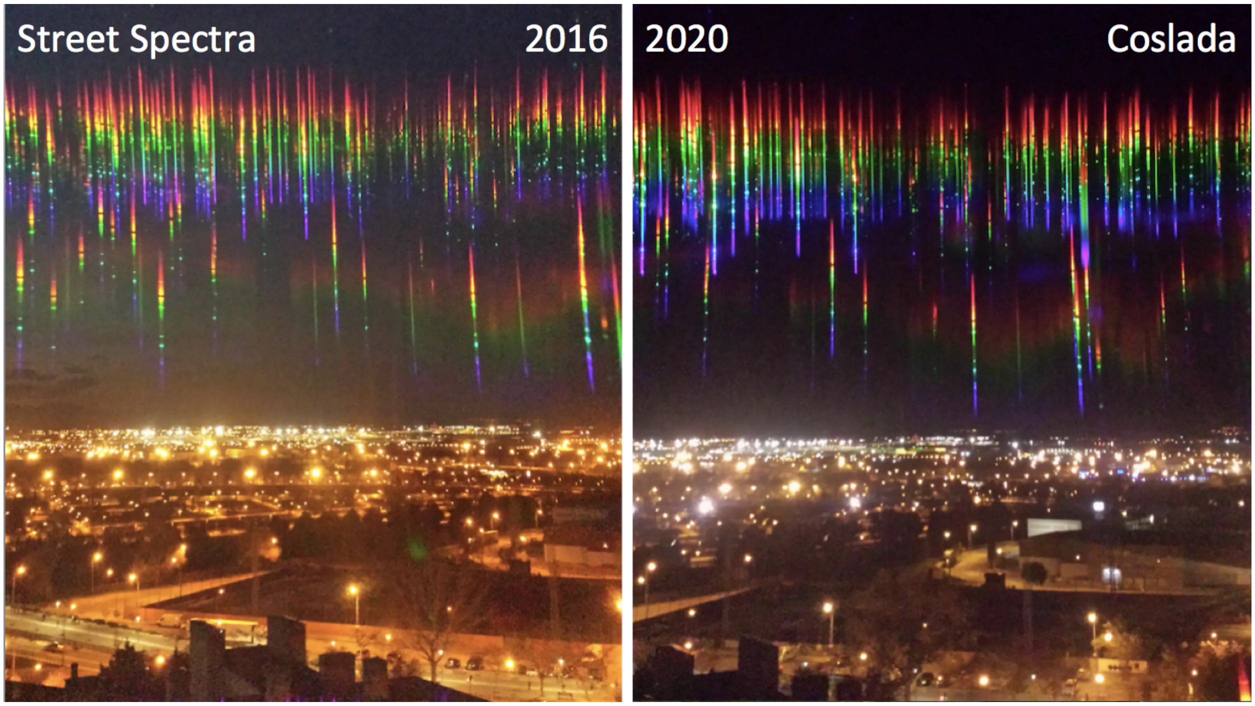


Figure 2. Wide field image of the street lamps spectra obtained in Coslada (near Madrid). The lighting retrofit has changed the orange HPS lamps (2016) to mostly white LEDs.

3. StreetSpectra

3.1. Introduction

StreetSpectra project is based on the use of simple and cheap transmission grating on top of the smartphone lens of the camera. This simple setup allows recording both the lamp or light source and the spectrum.

The StreetSpectra webpage is the online entry point for those who want to contribute to our project. The gallery section has some examples of spectra recorded by collaborators with their smartphones. We show in Figure 2 an striking example of the changes in street illumination (lighting retrofit) in Coslada (near Madrid) with mainly orange HPS lamps being changed to white LEDs.

A comprehensive tutorial has been developed and published as part of the ACTION project that will give the citizen scientist the necessary background to be involved in this project.

Also, our purpose is to educate younger citizens in this relatively unknown sub-field of environmental awareness. So, we have developed and published suitable teaching materials for high schools.

To gather pictures of street light spectra, we have designed the simplest possible spectrograph: a cheap transmission grating held on top of the smartphone camera. These are some of the diffraction grating suppliers. We recommend buying a diffraction grating of **500** lines/mm:

- Edmund Optics
- Jeulin
- AliExpress
- Amazon Spain
- Amazon UK
- Amazon DE

ACTION Deliverable D2.1 introduces StreetSpectra and describes the procedure to take images as well as a tutorial for image capture and spectra classification. The UCM-LICA database of street lamps spectra has been updated with images of lamps spectra as seen using this simple setup.

3.2. Relation with the ACTION toolkit

StreetSpectra is one of the citizen science projects that were included as a pilot to test the developments of the ACTION project. The ACTION toolkit is a resources collection created by the ACTION consortium to help citizen science projects to develop. According to the classification given in the ACTION Toolkit, StreetSpectra falls into the **Investigation Projects category**. An exhaustive description of the ACTION toolkit can be found at The ACTION project website. In StreetSpectrat, Citizen Scientists participate in Data Collection via a mobile app and data gathering platform, and in Data Analysis using a crowdsourcing platform.

3.3. Use by Citizen Scientists

From the citizen scientist perspective, StreetSpectra provides two cooperation options according to his/her/their desired role in this initiative (which are non exclusive):

- As a *data collector* using a mobile application (and supported by a data gathering platform invisible to him/her/them)
- As a *data analyst* using a web browser and connecting to a platform that supports data classification (crowdsourcing platform).

Finally, both citizens and researchers will have the chance of getting the classification results as datasets by pointing their web browsers to a publishing platform and also see a graphical depiction of the mapping efforts in another website (geographical information system).

3.4. Final Architecture

It is the inclusion of the data analyst role that has made us evolve our initial architecture. As part of participating in ACTION as a pilot, we have received several inputs on how to make this possible. From the vision of a simple application that collected pictures of spectra, we have evolved into an architecture that includes several components:

- **Data Gathering Platform (Epicollect 5)**. This is the component that is in charge of collecting observations. It includes both a mobile application as well as a backend server.
- **Observations Database**. Part of the ACTION WP4 work is to provide pilots with an observations database. We have implemented a Mongo (non relational) database with an HTTP API to insert/download generic observations from other ACTION pilots, and have used this for StreetSpectra.
- **Crowdsourcing Platform (Zooniverse)**. In order to increase the reliability of observations and overcome the limitations of the mobile application, we have developed a platform where citizens may examine and analyse the photos and spectra taken and stored in the Zooniverse Data gathering Platform and submit their classifications.
- **Publication Platform (Zenodo)**. Both observations and classifications are scientifically valuable data that should be published as Open Data and should be able to be cited as such in other papers/studies.
- **Workflow Platform (Airflow)**. This platform orchestrates the data flows among the above platforms, providing support for what is commonly referred to as Extraction, Transformation & Loading (ETL) processes. This has also been deployed by ACTION WP4.

Deliverable D2.10 described in detail the platform components listed above and defined a conceptual workflow of tasks to execute (figure 3). As expected, the actual workflow and tasks implementation slightly differs from our conceptual view and offers in practice three workflows, which will be described below.

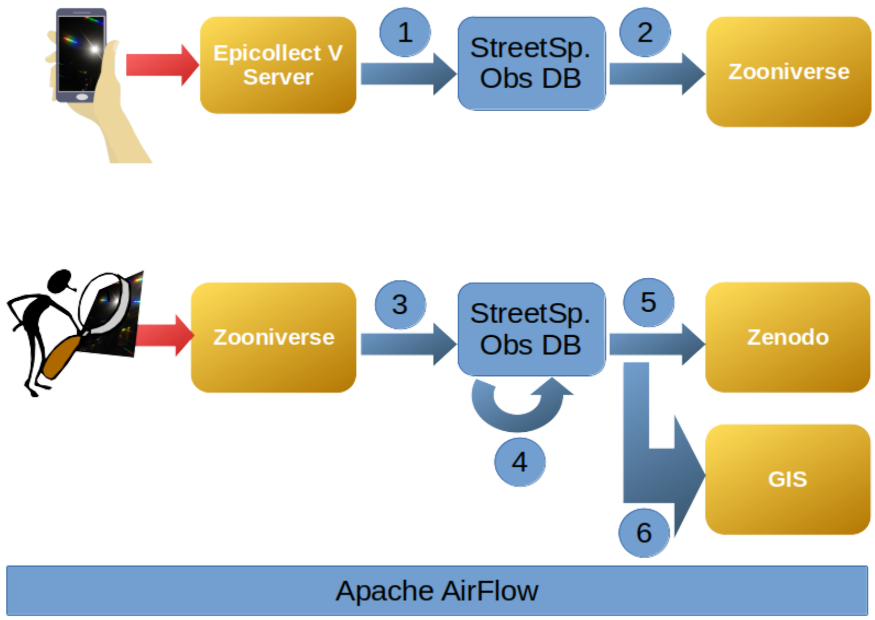


Figure 3. StreetSpectra overall solution architecture
for observers (top) and data analysers (bottom)

3.5. StreetSpectra collecting workflow.

This workflow's goal is to collect observations from Epicollect 5 and to feed them to the ACTION database. As a reminder, Epicollect 5 is an open, mobile centric, citizen science platform developed and managed by the *Imperial College of London*. The ACTION Database is a non-relational observations database (MongoDB) that can be interacted with using an HTTP API.

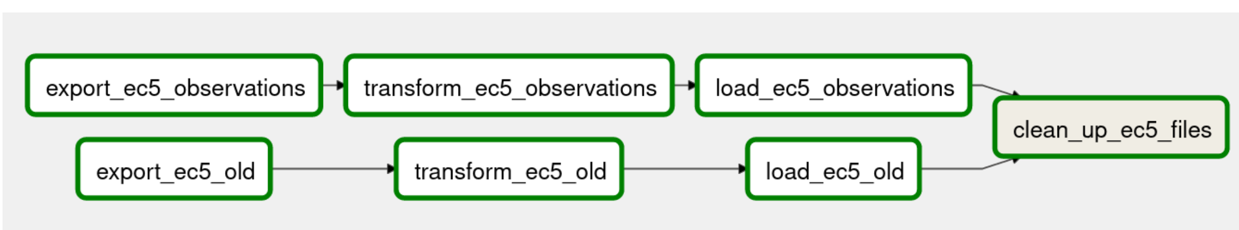


Figure 4. StreetSpectra Collecting workflow. This diagram is automatically generated by Airflow from the DAG (workflow) specification file.

The following tasks comprise this workflow:

- **`export_ec5_observations`**. This task is responsible for downloading the current month's observations from Epicollect 5 and storing them as text lines in a monthly JSON file whose format is defined by Epicollect 5, which in turn depends on the Epicollect 5 StreetSpectra project form.
- **`transform_ec5_observations`**. This task takes as input the generated JSON file from the former task and generates a new JSON with the information needed by StreetSpectra, and adding some new metadata (observation type and observation source platform)
- **`load_ec5_observations`**. This task takes the file generated by the previous task and loads it into the ACTION database. All the tedious work is done by the ACTION hook, a custom Airflow component developed for ACTION.
- **`clean_up_ec5_files`**. When the observations are successfully loaded into the ACTION database, temporary files are deleted so as no to fill up disk space.

Figure 4 also shows three additional parallel tasks (*_old tasks) which we introduced to collect additional images in a prototype Epeivcollect % project that we discontinued later on, showing how easy and flexible it is to alter each workflow to add additional tasks.

3.6. Zooniverse feeding workflow

This workflow's goal is to periodically feed the Zooniverse StreetSpectra project with enough images so that the classifier citizens always have something to examine. Uploading is automatically done in batches (a SubjectSet). There is a tradeoff of batch size between having classification results early, which promotes using low batch sizes and the Zooniverse system imposed maximum limit. The Zooniverse team recommends between 100-500, with 500 being an upper bound. For the time being, we have set this size to 100 images by SubjectSet, but this can be changed later on. Unlike the other two workflows' periods, this one is set on a daily basis so that we can detect early if the current SubjectSet has been already classified by all participants. If there are not enough classified images yet.

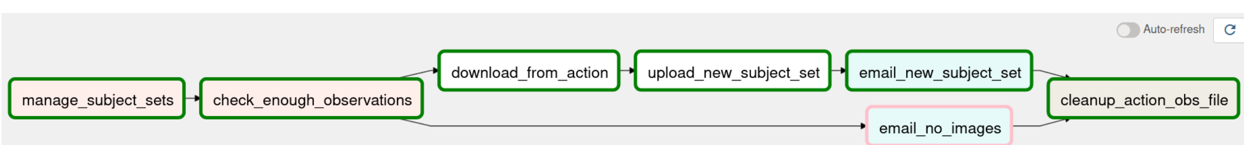


Figure 5. StreetSpectra Zooniverse feeding workflow. A branch is shown depending on whether we have new images to be uploaded or not.

The following tasks comprise this workflow:

- **`mange_subject_sets`**. This task is a special “short-circuit” operator. It detects if the current SubjectSet has reached a certain classification level threshold (75%) . If not, it makes this workflow end silently. On the other hand, If the classification level is 100%, it removes the current SubjectSet from Zooniverse.
- **`check_enough_observations`**. This task is a special “branch” operator and is executed when the previous task has asserted that we need to create a new SubjectSet. This task checks that there are enough images to create a complete SubjectSet of 100 images. If so, the workflow branches execution to the upper set of tasks (see figure 5). If not, the lower branch is executed.
- **`download_from_action`**. This task downloads a series of images & metadata from the ACTION database, creating a JSON file. The issue with this task is that we have to record somewhere where we left off downloading images, as we do not wish to generate duplicates for image classification in Zooniverse. This is solved by using an ***Airflow variable*** that is updated by this task with the timestamp of the next StreetSpectra image to be downloaded in the following run.
- **`upload_new_subject_set`**. This task creates a new SubjectSet from the JSON metadata downloaded from the previous task. It is worth noting that the images themselves are really URLs to the real images stored in Epicollect 5, so we minimize data copies and storage.
- **`email_new_subject_set`**. This task notifies the StreetStectra operation & maintenance staff (an email address) about the creation of a new SubjectSet.
- **`email_no_images`**. This task is executed when there are not enough StreetSpectra images to complete a SubjectSet. It notifies the StreetStectra operation & maintenance staff (an email address) of this fact.
- **`cleanup_action_obs_file`**. This taks clear all temporary files created by the current daily workflow run.

3.7. Aggregation and publication workflow

This workflow’s goal is to collect all individual classifications from the StreetSpectra Zooniverse project, aggregate the classifications and publish both the individual and aggregated results in Zenodo (StreetSpectra & ACTION communities) as a CSV dataset. The resulting CSV datasets file format is defined in the enclosed appendix. The complexity of this workflow stems from two facts:

- Zooniverse classification exports are always made from the beginning of the project and for efficiency reasons we wish to perform data insertions[2] and aggregations only from new data since the previous workflow last run. We had to introduce an auxiliary relational database (SQLite) to easily detect the data increments. The database ignores duplicated records and keeps track of the last classification timestamp.
- The images contain more than one light source, i.e, lampposts and other sources., and we don't know in advance which light source the classifying citizen will choose. So we have to perform aggregations per classified light source per image.

This workflow is executed monthly.

[2] The ACTION database does not avoid data duplicates when writing.

Figure 6. Aggregation and publication workflow.
Upper figure shows the initial tasks up to the check_new_spectra branching task.
Lower figure shows the following tasks till the end of the workflow.

The following tasks comprise this workflow (see figure 6):

- **`export_classifications`**. This task executes the Zooniverse raw data export and saves the result in a JSON file. This operation imposes a heavy load on the Zooniverse servers, so it is allowed only once per day. Since the export is made from the beginning of the project, resulting files can be large.
- **`only_new_classifications`**. This task reads the JSON file generated by the previous task, decodes its content and it is inserted into an auxiliary database. After the insertion, a subset JSON file is created with the new entries since the last export took place.
- **`transform_classifications`.** This task adds additional ACTION-specific metadata to the file generated by the former task before it can be inserted into the ACTION database. This is done so that the previous two tasks can be reused for other Zooniverse projects other than ACTION.
- **`preprocess classifications`.** Zooniverse data exports contain a large amount of metadata in deeply nested JSON structures. To aid further processing of the StreetSpectra tasks, this task picks only the relevant metadata and stores

it in a flat table inside the auxiliary database, so that the next tasks can handle a simplified data model.

- **`check_new_spectra`**. This task detects if there are new classifications in the database left from the previous task. This is a branching task and branches to the aggregation task or the dummy end task depending on the test.
- **`aggregate_classifications`**.
- **`export_aggregated_csv`**. This task exports all aggregated classifications from the beginning of the project to a dataset (CSV file).
- **`publish_aggregated_csv`**. This task publises to Zenodo (StreetSpectra & ACTION communities) a versioned dataset with the aggregated classifications generated in the previous task.
- **`export_individual_csv`**. This task exports all individual classifications from the beginning of the project to a dataset (CSV file).
- **`publish_individual_csv`**. This task publishes to Zenodo (StreetSpectra & ACTION communities) a versioned dataset with the individual classifications generated in the previous task.
- **`join_published`**. This is a dummy task whose mission is to synchronize the parallel execution of exporting and publishing both the individual and aggregated results.
- **`skip_to_end`**. This is a dummy task whose mission is to be a placeholder of the other branch when there are no new classifications detected.
- **`clean_up_classif_files`**. This taks clear all temporary files created by the current monthly workflow run

3.8. Airflow reusable components

We have developed a collection of Airflow components within the StreetSpectra project. Some of them are not specific to StreetSpectra and reusable to other contexts. We have published these components in a public Python GitHub repository. It is a Python library that, once installed, provides all components described below.

3.8.1. Hooks

Hooks are components that encapsulate the often complex interaction between Airflow and an external system. A hook needs an “Airflow connection” which encapsulates the actual system protocol endpoint and credentials (often, simply username and password). By using custom hooks, the Airflow operators (aka tasks) are greatly simplified. The following new hooks have been developed.

- Epicollect5 Hook. This hook allows Airflow operators to download observations from the Epicollect5 Observing Platform. The added value of this hook is that it hides the usage details such as the protocol used: HTTPS protocol with paging hiding.
- Zooniverse hook. Zooniverse already provides a rich and complex PANOPTES Python bindings to Zooniverse. This hook allows the easy management of the Zooniverse entities like SubjectSets (creation, expiration, detection of new subject sets to create), along with the associated subjects. Another special Zooniverse hook was also subclassed for StreetSpectra in order to insert project specific metadata into Zooniverse.
- Zenodo Hook. This hook allows easy publication of Zenodo datasets within Airflow, hiding the complexity of Zenodo REST API.
- Action Database Hook. This hook allows inserting and querying observations from any ACTION project pilot, not just StreetSpectra, hiding the details of HTTP and query pages.

3.8.2. Operators

Operators are the basic building blocks of Airflow. Each operator provides a single, specific task to execute within a workflow. While it is possible to develop a workflow using the generic `BashOperator` and `PythonOperator` operators, we have made extensive usage of operator customization (subclassing) to help develop more compact workflows, easier to understand and maintain.

The following reusable operators have been developed:

- EC5 Export Entries.
- Zenodo Publish Dataset.
- Zooniverse Export.
- Zooniverse Delta.
- Zooniverse Transform.
- Action Upload
- Action Download. A series of ACTION Download operators have been developed, with variations in when to download observations and how to manage observations in chunks of fixed size, starting from a date.

3.8.3. Callables

Airflow provides two operators for explicit branching between tasks (`BranchOperator` & `ShortCircuitOperator`). However, the real test the branch is based upon must be provided by the application. This piece of code is named a "callable" is simply a Python function which must return `True` or `False` (`ShortCircuitOperator`) or the task identifier of the next call to execute (`BranchOperator`). The following callables have been developed:

- Manage Zooniverse subject sets. Queries for the % classification status in the current subject set and returns true or false if a certain threshold is reached. Also removes the current subject set if it is 100%. This callable is reusable to other projects dealing with Zooniverse.
- Check the number of entries in the ACTION database for a given project against a given threshold. It returns the next task id to execute depending on the test. This callable is reusable for any other ACTION project.
- Check new subjects. This callable checks for new classifications in the Zooniverse export file. Valid only for StreetSpectra.

3.8.4. Airflow: Conclusions after usage

Airflow is really a powerful environment to coordinate a workflow (or workflows) across several computer systems as it has been our case in StreetSpectra: Epicollect 5, ACTION database, Zooniverse, Zenodo). All this would have been more tedious, error-prone and would probably require more maintenance work to modify the workflows to evolving needs if it had not been possible by Airflow. We have been able to decompose our big problem into smaller, well defined tasks and some of them are even reusable to other projects (ACTION or other CS projects needing heavier IT support). However, Airflow has a steep learning curve at first. Both the tool deployment and initial training on Airflow have been possible only through ACTION WP4 support. The tool itself allows for big scale deployments in other contexts, but for the time being we have remained to keep it as simple as possible.

3.9. Conclusions and further work

From the early start of the StreetSpectra project, paused by the lockdown period, to the final deliverable, we have evolved our final vision of this CS project. The ability to introduce citizen scientists as data analyzers opens the project to more citizen cooperation and we have developed a coherent ecosystem of data gathering, processing and analysis for StreetSpectra so indeed this can be considered a first, soft release. Once we have tested all tasks and workflows more thoroughly, we will begin promoting StreetSpectra more widely.

Further work in StreetSpectra (we are already working on them) includes:

- Add maps. Instead of adding yet another platform (GIS platform) to maintain, we will probably generate dynamic HTML maps with all the mapping efforts and classifications done so far. This will be possible thanks to Airflow by either adding new tasks to the aggregation and publishing workflow or by designing a new Airflow mapping workflow.
- Localize Zooinverse StreetSpectra in Spanish, our main target for the time being. This can only happen with cooperation from the Zooniverse team and after having passed their revision process.

4. AZOTEA: Citizen Science during the confinement period

AZOTEA is another “investigation type” of Citizen Science projects as described in the ACTION Toolkit in which the citizens were involved in the data collection.

During mid-March to the end of June 2020, Spain suffered a strict lockdown due to the anti-covid-19 measures. However, these exceptional circumstances opened a door to research the consequences of a decrease of human activities during this period in terms of light pollution impacts.

The idea was that citizen scientists monitored the brightness and color of the night sky using pictures taken with consumer grade cameras to evaluate variations in light pollution that may happen during the exceptional period of self-isolation. We expected a decrease of light pollution since the decrease of human activity is normally associated with darker skies. Anybody with a DSLR (digital single lens reflex) camera who was interested in the project was welcomed to participate.

The objectives were:

1. To study the impact of human activity on light pollution.
2. The evaluation of the brightness and color of the sky of our cities and towns.
3. The introduction to photometric measurement techniques with digital cameras.
4. To establish a network of digital cameras to monitor the brightness and color of the night sky.

An in-depth description of AZOTEA has already been included in deliverable D2.10.

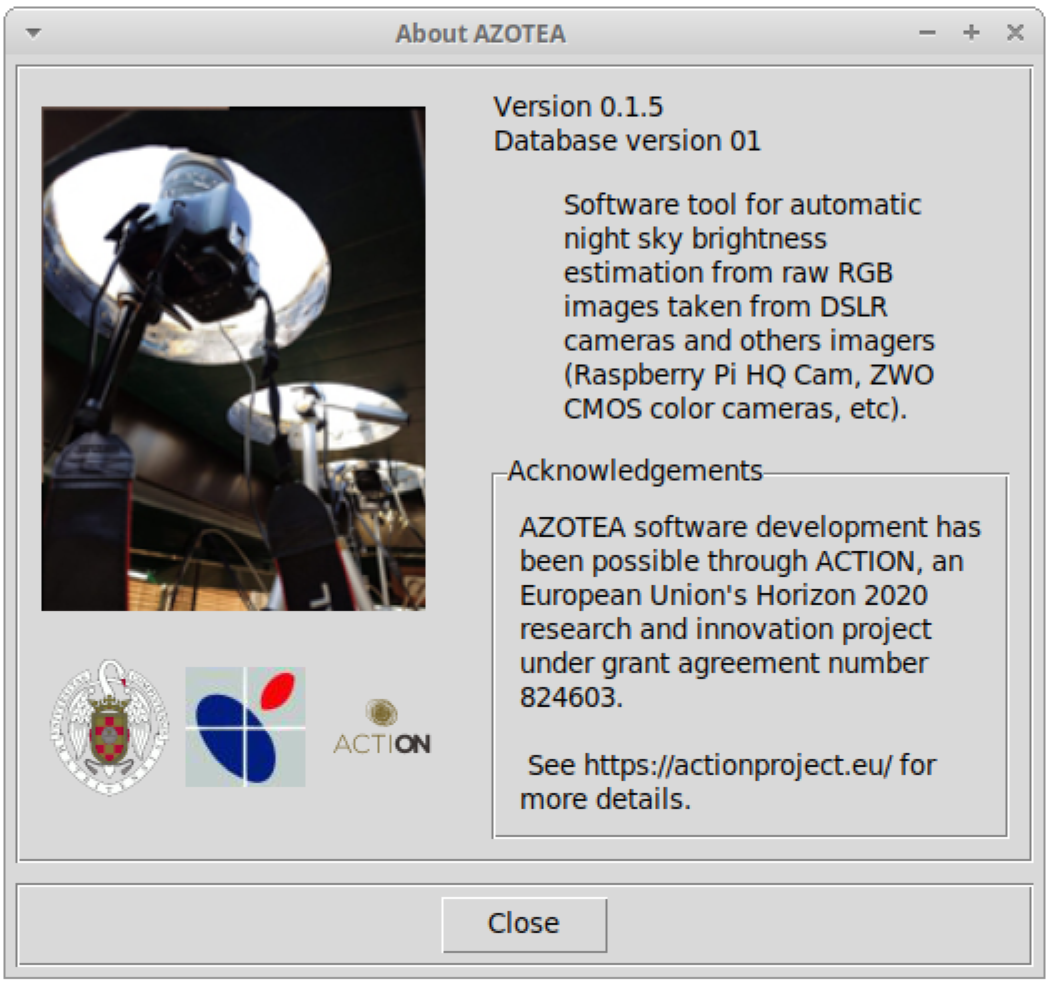
4.1. Results.

Deliverable D2.10 presented the results of this project and are not repeated here. These are the main results we have achieved with AZOTEA during this period are:

- The AZOTEA client software to process the DSLR raw images to yield night sky brightness is also available as open software in GitHub. This is presented in detail below.
- New monthly dataset versions are being uploaded to Zenodo.

- A poster "**:** [AZOTEA: (Zenithal astronomy during & after the lockdown)] showing this project has been presented for the [eALAN2021 conference] AZOTEA GUI Application

Since publication of D2.10, we have been proactively working during this period trying to solve the storage space problem that AZOTEA creates when trying to set up a centralized image repository. The result is azotea-client, a cross-platform rich client developed with the Python built-in GUI capabilities. These are based on the well-known Tk toolkit that enables development of basic but fully functional graphical applications[3].



[3] *SAOImage DS9 is another well known example in the astronomical community of a rich client developed with Tk.*

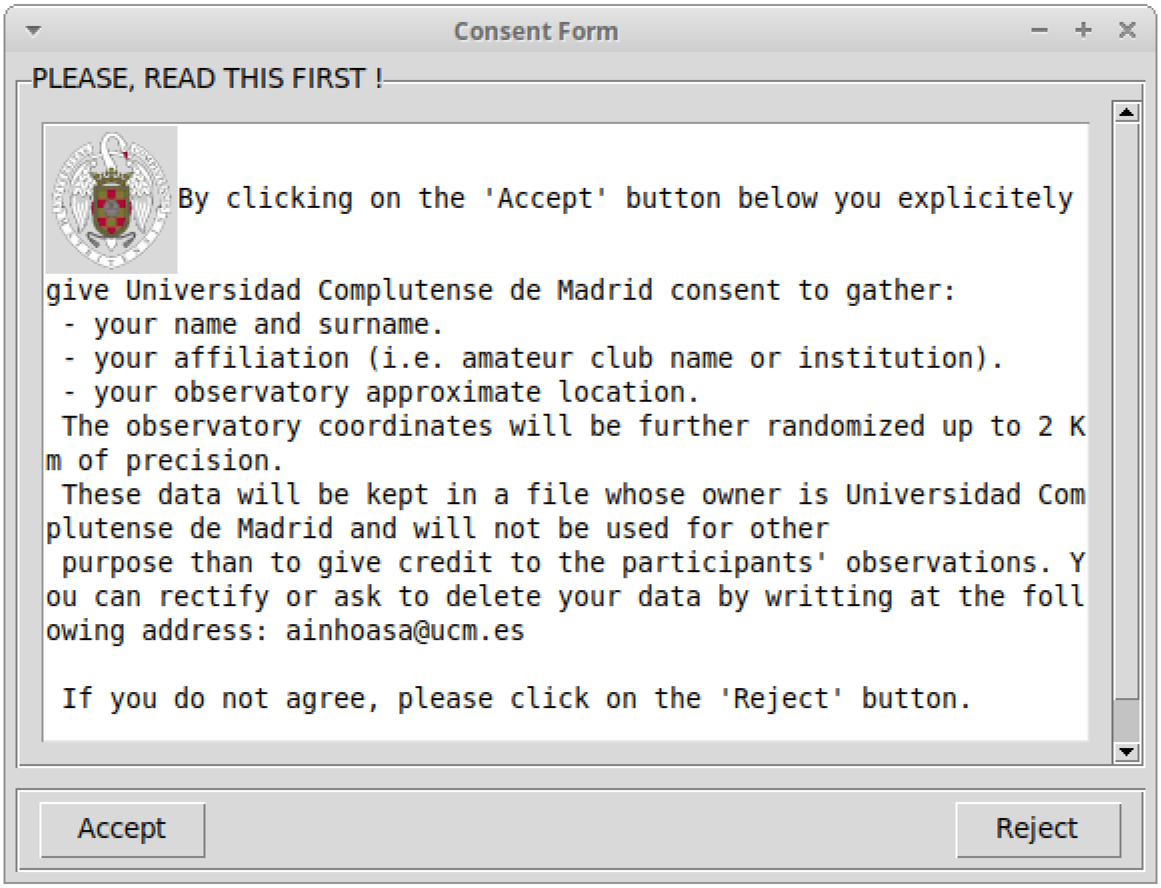


Figure 7. Upper image: AZOTEA client credits. Lower image: Consent form.

The AZOTEA client software is easy to operate. We need first to give our consent on capturing personal data (see Figure 7), set up preferences for the camera, observer, location, region of interest, publishing & miscelanea options and we are ready to go.

4.1.1. AZOTEA Preferences

The AZOTEA preferences dialog is opened under the menu Edit > Preferences. Figure 8 shows some of these preferences

As a general remark, AZOTEA observations are made by a single observer using a single camera in a certain location. But we have made it possible to add several observers, cameras, locations and regions of interest. This makes it possible for us too to re-process images from different observers and their setups.

- The `Camera` tab allows us to select the default camera metadata we will be using during the reduction process. If the camera is not available in the drop-down list of cameras, the user has the choice to edit a new one. The recommended way to do it is simply to select an image belonging to the new camera model. AZOTEA will read all the relevant camera EXIF metadata and will present it to the user.
- The `Location` tab allows us to select the default Location where the observations are being made. If the location is not available in the drop-down list of locations, the user has the choice to edit a new one. Location may include coordinates. However, the location coordinates being published will be randomized with a precision of 1 Km from the real location coordinates, which will be kept local to the user PC.
- The `Observer` tab allows us to select the default Observer who made the observations. If the observer is not available in the drop-down list of observers, the user has the choice to edit a new one, which includes name and affiliation both in long and short form.
- The `ROI` tab allows us to define a rectangular region of interest in which the software will compute the average brightens in each of the four channels (R1, G2, G3, B4). Aun automatic centering button allows the user to select an image taken with the chosen camera and automatically calculates the rectangle coordinates in the image center.
- The `Publishing` tab allows us to define the server URL where the reduced data will be published. This needs authentication data (username/password for the time being).
- The `Miscelanea` tab allows us to define other needed metadata that were not part of the former categories (such as default optics configuration just in case there is no EXIF optics metadata in the image)

Figure 8. From left to right, up to down: Camera, Region of Interest, Observer and Location preferences for the AZOTEA rich client. Location coordinates are internally randomized with 2km error.

4.1.2. AZOTEA main screen and operation

Once the user has spent some time adjusting the preferences, he/she is now ready to process the images. This is a two step process:

- ☐ First, by selecting `Files > Load images` … the software registers a directory with images in an internal database, making sure that there are no image duplicates with previously registered images. The duplication test is made by calculating an md5 hash with the file contents itself and not by image name nor file paths, because they may change and/or overlap. This makes the registration process a bit slow for directories up to 600 images per directory, but still quite manageable. Progress is shown in the upper table showing relevant metadata as well as the status bar (see figure 9).

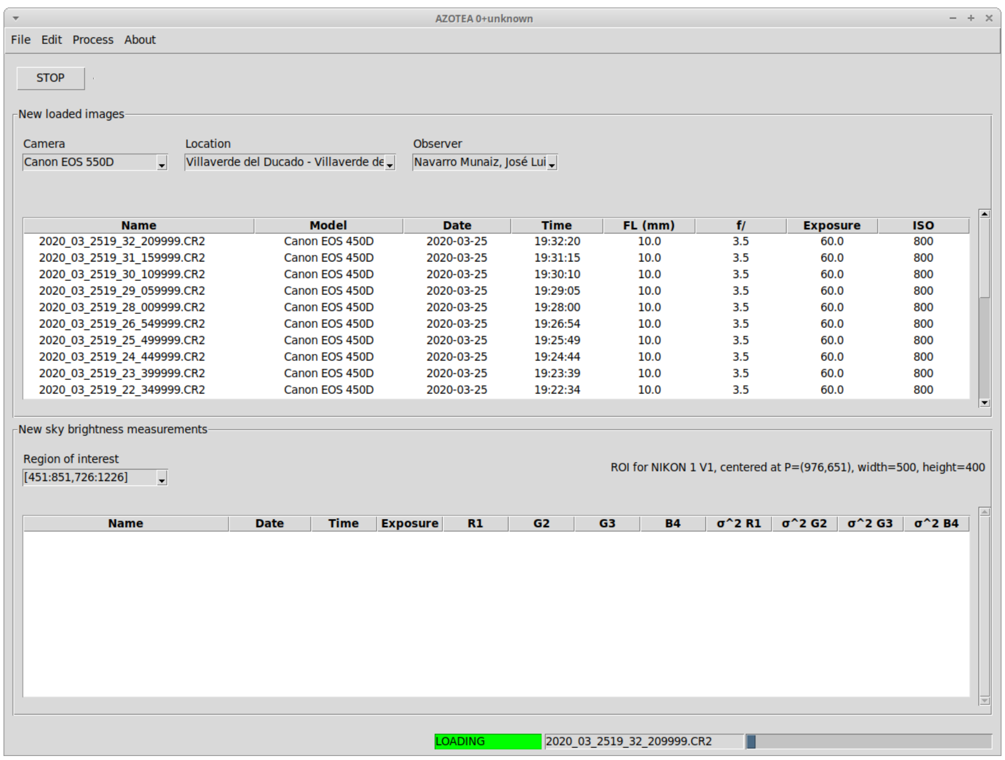


Figure 9. Image registration under way. Some metadata are displayed such as camera model, date & time,exposure time and ISO.

- ☐ Second, by selecting `Process > Sky Brightness` … the actual sky brightness measurements in the given ROI are being made. Progress is shown in the lower table showing averages and variances in each channel as well as the status bar (see figure 10).

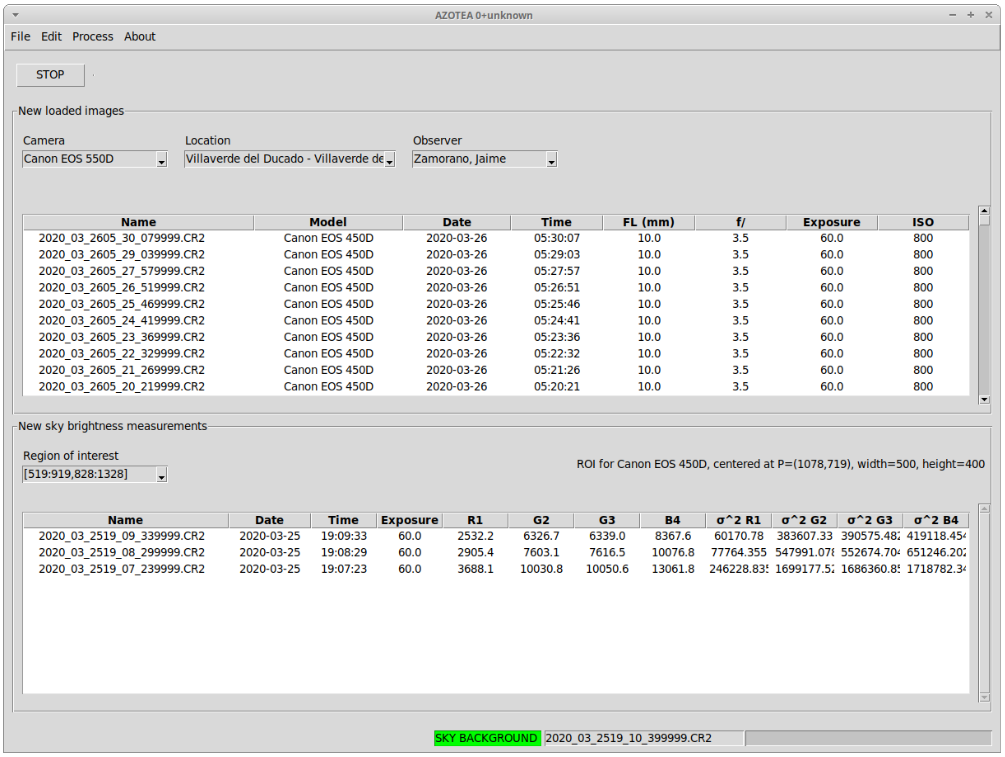


Figure 10. Sky brightness statistics under way.

Note that the Region of interest can be changed after the image processing is done, but this doesn't mean registering the images again, only computing the sky brightness again.

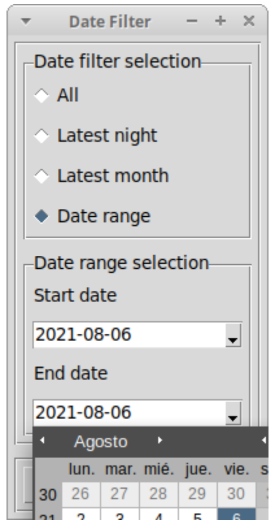


Figure 11. File export options dialog.

Once the statistics are completed, we can export the results in CSV format. A dialog window opens with several time options available (figure 11). Note that the other dimensions like observer, location are fixed once they have been set up as defaults. The CSV file format is defined in a separate document and available in Zenodo.

4.2. Conclusions and further work

Beyond the scope of ACTION, we will continue to work in AZOTEA. Besides StreetSpectra and the growing network of TESS-W photometers from [the EU funded STARS4ALL project,] this is another way in which citizen scientists can cooperate with us in measuring and characterizing the light pollution.

We have two paths available:

- Distribute, maintain and evolve our rich client application to future user needs. This include:
    - Include other image taking devices other than DSLRs. (i.e specialized astrophotography color CMOS devices or even Raspberry Pi HQ camera)
    - Setting up a new AZOTEA server that picks up reduced results online in a similar way as a CSV exported file.
    - Include other graphical utilities in AZOTEA Client as we see fit.

- Absolute calibration of the results.
  We are publishing a series of papers developing a new photometric system for observation obtained with RGB cameras. (More information and the related papers can be found at UCM RGB webpage This important path, explained in detail in Deliverable D2.10, would allow us to measure light pollution in traditional calibrated photometric magnitudes, rather than in counts and allows us to compare results from different observers in nearby large areas like Madrid.

6. APPENDIX

6.1 StreetSpectra CSV File Format

This document contains the description of the columns in the datasets exported to Zenodo.

StreetSpectra dataset is versioned and periodically published in the Zenodo open data platform contains users' classifications and has passed through two stages:

- Data collection stage where observers take spectra using a low cost diffraction grating and Epicollect 5 mobile APP, uploading the images and metadata into Epicollect 5 Data gathering platform and its mobile APP. Optionally they can provide its own guess of the spectrum type. This may be an important datum if the observer is familiarized with the different illumination technologies.
- Data classification stage where other citizens examine individually each of the images uploaded by observers. This classification takes place in the Zooniverse classification platform. For a given subject (image), M users must provide their own guesses of the spectrum type examining the photo, and the spectrum. A user can only classify one light source per subject. This M is configured per classification project with typical values ranging from 5 to 15.

6.1.1 Raw classifications file format

This file contains a useful subset of the data/metadata exported by the StreetSpectra Classification Zooniverse project. Each line corresponds to an individual classification made by a user in a given subject (image)

File containing 1 header line and N data lines. Separator is semicolon, Header labels and description follows:

csv_version	string	CSV file format version. Helps processing software cope with CSV version changes. Current value is string `1.0`.
subject_id	integer	unique Zooinverse subject identifier. One subject is one image to be examined and classified.

classification_id	integer	unique Zooinverse classification identifier. A single classification result made by a user.
started_at	string	Classification UTC start timestamp, IS08601 format
finished_at	string	Classification UTC end timestamp, IS08601 format
width	integer	Subject (image) width
height	integer	Subject (image) height
source_x	float	light source x coordinate within the image
source_y	float	light source y coordinate within the image
spectrum_x	float	spectrum box corner point, x coordinate
spectrum_y	float	spectrum box corner point, y coordinate
spectrum_width	float	spectrum box width
spectrum_height	float	spectrum box height
spectrum_angle	float	spectrum box angle (degrees) (respect to X axis?)
spectrum_type	string	spectrum type One of `('LED', 'HPS', 'LPS', 'MH', 'MV')`.
image_url	string	Image URL stored in the observing platform. You can retrieve the image using this URL. Metadata coming from the observing platform.
image_long	float	Image longitude (decimal degrees). Strictly speaking this is the observer's position (serving as a proxy). Metadata coming from the observing platform.
image_lat	float	Image latitude (decimal degrees). Strictly speaking this is the observer's position (serving as a proxy). Metadata coming from the observing platform.
image_observer	string	Observer's nickname. May be empty. Metadata coming from the observing platform.

image_comment	string	Observer's comment. May be empty. Metadata coming from the observing platform.
image_source	string	Observing platform name (currently `'Epicollect 5'`)
image_created_at	string	Image creation timestamp, in UTC ISO-8601 format. Metadata coming from the observing platform.
image_spectrum	string	Observer own guess for the spectrum. May be empty. Metadata coming from the observing platform.

6.1.2 Processed file description

The processed file contains the aggregated information from M users about each light source. Each line corresponds to a given light source aggregated classification made by users.

Since a given image contains more than one light source with its spectrum and we cannot avoid each user choosing a different light source for classification, the final classification is made on a 'per light source' basis.

File containing 1 header line and N data lines. Separator is semicolon, Header labels and description follows:

csv_version	string	CSV file format version. Helps processing software cope with CSV version changes. Current value is string `1.0`..
source_label	string	A tag identifying the light source within a subject (image)
source_x	float	Light source average x coordinate taken from each users' selection. Empty if no classification was made at all (rejection tag = 'Not classified).
source_y	float	Light source average y coordinate taken from each users' selection. Empty if no classification was made at all (rejection tag = 'Not classified).
spectrum_type	string	Aggregated spectrum type inferred from each users' selection. One of `('LED', 'HPS', 'LPS', 'MH', 'MV')`. Empty if no aggregated spectrum type could be inferred.

spectrum_dist	string	Spectrum classification distribution made by the individual users classifying this particular light source. Syntax is Python-like i.e. `[('LED', 2), ('HPS', 1)]`
agreement	string	Percentage agreement made by the individual users classifying this particular light source. Given as a fraction, i.e `'2/3'` => 2 users out of 3 chose LED in the example above.
rejection_tag	string	If the spectrum type is empty, this field gives a tag with the reason. Values contemplated so far are `('Ambiguous','Not classified')`
kappa	float	Inter-rater consistency agreement statistics when classifying light sources in the same subject. The current statistics used is Flesiss' Kappa.
users_count	integer	Number of different users when computing kappa for a given subject
image_url	string	Image URL stored in the observing platform. You can retrieve the image using this URL. Metadata coming from the observing platform.
image_long	float	Image longitude (decimal degrees). Strictly speaking this is the observer's position (serving as a proxy). Metadata coming from the observing platform.
image_lat	float	Image latitude (decimal degrees). Strictly speaking this is the observer's position (serving as a proxy).
image_observer	string	Observer's nickname. May be empty. Metadata coming from the observing platform.
image_comment	string	Observer's comment. May be empty. Metadata coming from Observing platform.
image_source	string	Observing platform name (currently `'Epicollect 5'`)
image_created_at	string	Image creation timestamp, in UTC ISO-8601 format. Metadata coming from the observing platform.
image_spectrum	string	Observer own guess for the spectrum. May be empty. Metadata coming from the observing platform.

6.2 AZOTEA CSV File Format

This document contains the description of the columns in the exported AZOTEA dataset. AZOTEA dataset is versioned and periodically published in the Zenodo open data platform contains reduced users' images.

6.2.1 Sky background measurement file format

File containing 1 header line and N data lines. Separator is semicolon, Header labels and description follows:

session	string	An identifier of the reduction session. All images processed in the same session have the same identifier. The identifier has the format of a timestamp YYYYMMDDHHMMSS
observer	string	Observer's name and surname.
organization	string	Observer's Affiliation. Preferable in acronym form for brevity (i.e AAM instead of Agrupación Astronómica de Madrid). The AZOTEA client software chooses the acronym in the preferences if available.
location	string	Observer's location name. In the AZOTEA client software, If the site name (i.e. M30 Este) is different from the location name (i.e. Madrid), both are placed here separated by a dash. Otherwise, just the location name is placed.
type	string	Image type. Either LIGHT or DARK. Currently only LIGHT images are being published.
tstamp	string	ISO 8601 timestamp format. This is extracted from the camera and should be UTC.
name	string	Image file name being processed
model	string	Camera model as extracted by the EXIF header.
iso	string	Image ISO, the camera signal gain. Extracted from EXIF metadata

roi	string	Image Region of Interest. Format is Python-like: [x1:x2,y1:y2]
dark_roi	string	Region of interest to measure dark signal. It may be a dark zone not exposed to light within the image itself. Not being used.
exptime	float	Image exposure time. Extracted from the EXIF metadata
aver_signal_R1	float	Image average signal in ADUs for the Red channel and the given region of interest. No dark nor bias subtracted, these are raw counts.
std_signal_R1	float	standard deviation in ADUs for the Red channel.
aver_signal_G2	float	Image average signal in ADUs for one Green channel and the given region of interest. No dark nor bias subtracted, these are raw counts.
std_signal_G2	float	standard deviation in ADUs for the Red channel.
aver_signal_G3	float	Image average signal in ADUs for the other Green channel and the given region of interest. No dark nor bias subtracted, these are raw counts.
std_signal_G3	float	standard deviation in ADUs for the Red channel.
aver_signal_B4	float	Image average signal in ADUs for the Blue channel and the given region of interest. No dark nor bias subtracted, these are raw counts, tehse are raw counts.
std_signal_B4	float	standard deviation in ADUs for the Red channel.
aver_dark_R1	float	Image average dark signal in ADUs for the Red channel and the given dark region of interest. Not bias subtracted, these are raw counts. Not being used, default to 0.
std_dark_R1	float	standard deviation in ADUs for the dark region in the Red channel.s. Not being used, default to 0.

aver_dark_G2	float	Image average dark signal in ADUs for one Green channel and the given dark region of interest. Not bias subtracted, these are raw counts. Not being used, default to 0.
std_dark_G2	float	standard deviation in ADUs for the dark region in one Green channel.. Not being used, default to 0.
aver_dark_G3	float	Image average dark signal in ADUs for the other Green channel and the given dark region of interest. Not bias subtracted, these are raw counts. Not being used, default to 0.
std_dark_G3	float	standard deviation in ADUs for the dark region for the other Green channel.. Not being used, default to 0
aver_dark_B4	float	Image average dark signal in ADUs for the Blue channel and the given dark region of interest. Not bias subtracted, these are raw counts. Not being used, default to 0.
std_dark_B4	float	standard deviation in ADUs for the dark region in the Blue channel.s. Not being used, default to 0.
bias	integer	Camera global bias. This is an integer and usually a power of two.