SolarPACES 2022

Placeholder Session title

https://doi.or/10........ DOI placeholder (WILL BE FILLED IN BY TIB Open Publishing)

Published: (WILL BE FILLED IN BY TIB Open Publishing)

Hel-IoT web Server: A Smart Heliostat Development Platform

Jose A. Carballo[1,3][https://orcid.org/0000-0003-0529-5672], Javier Bonilla [2,3][https://orcid.org/0000-0002-2322-2867], Jesus Fernández-Reche [1][https://orcid.org/0000-0003-1967-7823], Antonio Ávila-Marín [1][https://orcid.org/0000-0002-9523-9705], Diego-Cesar Alarcón-Padilla[2][https://orcid.org/0000-0002-8843-8511]

[1] CIEMAT-PSA. Point Focus Solar Thermal Technologies. P.O. Box 22, E-04200 Tabernas, Almería (SPAIN)

[2] CIEMAT-PSA. Solar Thermal Applications. P.O. Box 22, E-04200 Tabernas, Almería (SPAIN)

[3] CIESOL, Solar Energy Research Center, Joint Institute University of Almería - CIEMAT, Almería (SPAIN)

Abstract. The smart heliostat is a concept that arises from applying artificial intelligence and computer vision techniques, mainly the machine learning technique called object detection, to the traditional heliostat to increase its efficiency and reduce costs [1,2]. Continuing with previous work at CIEMAT-PSA focused on the development of a smart heliostat and especially the solar tracking subsystem [3], a powerful tool (Hel-IoT web server) has been developed helping us with the laborious task of creating the dataset and training the model employed for object detection in solar tracking. The main goal of the Hel-IoT web server is to enhance the image data set employed to train new models and, at the same time, to improve the model that the web server is continuously running. The first conclusions that we can extract from the use of the Hel-IoT web Server for more than 4 months is that it is a very powerful tool for the creation of labeled data sets with meteorological information, especially for the creation of data sets for smart heliostats. Furthermore, thanks to the joint use of CETA cluster (Centro Extremeño de Tecnologías Avanzadas), the Hel-IoT web server can improve the model automatically by periodically retraining it.

Keywords: Artificial intelligence, Solar tracking, Object detection

Smart heliostat

The smart heliostat is a concept based on the application of techniques derived from industry 5.0, mainly artificial intelligent and computer vision techniques. This work focuses on the development of solar trackers, which is a fundamental subsystem of heliostats. The Smart Heliostat Tracker (Hel-IoT) is based on machine learning. The main hardware elements are a low-cost low-power consumption embedded computer plus a wide-angle camera. Software includes a neural network able to detect the Sun, receiver, clouds and other heliostats. The general idea is to calculate the heliostat aiming position from detentions of the Sun and receiver in the camera video frames.

It is assumed that Hel-IoT is attached to the heliostat surface mirror's center point (O') and moves with it; other positions are also possible. This example considers the target as the element where the concentrated flux is reflected; the same applies for the receiver. The camera provides a plane view (CP) of the scene. The neural network detects the Sun's (S') and target's (T') center points. The middle point (A") between S' and T' is the desired heliostat aiming point. The current heliostat aiming point (A') is the center point in the plane view (CP).

The heliostat is then moved to place the current aiming point (A’) at the desired aiming point (A”). Figure 1a shows a camera view for an out-of-focus heliostat, whereas Figures 1b shows the same view for a focused heliostat.

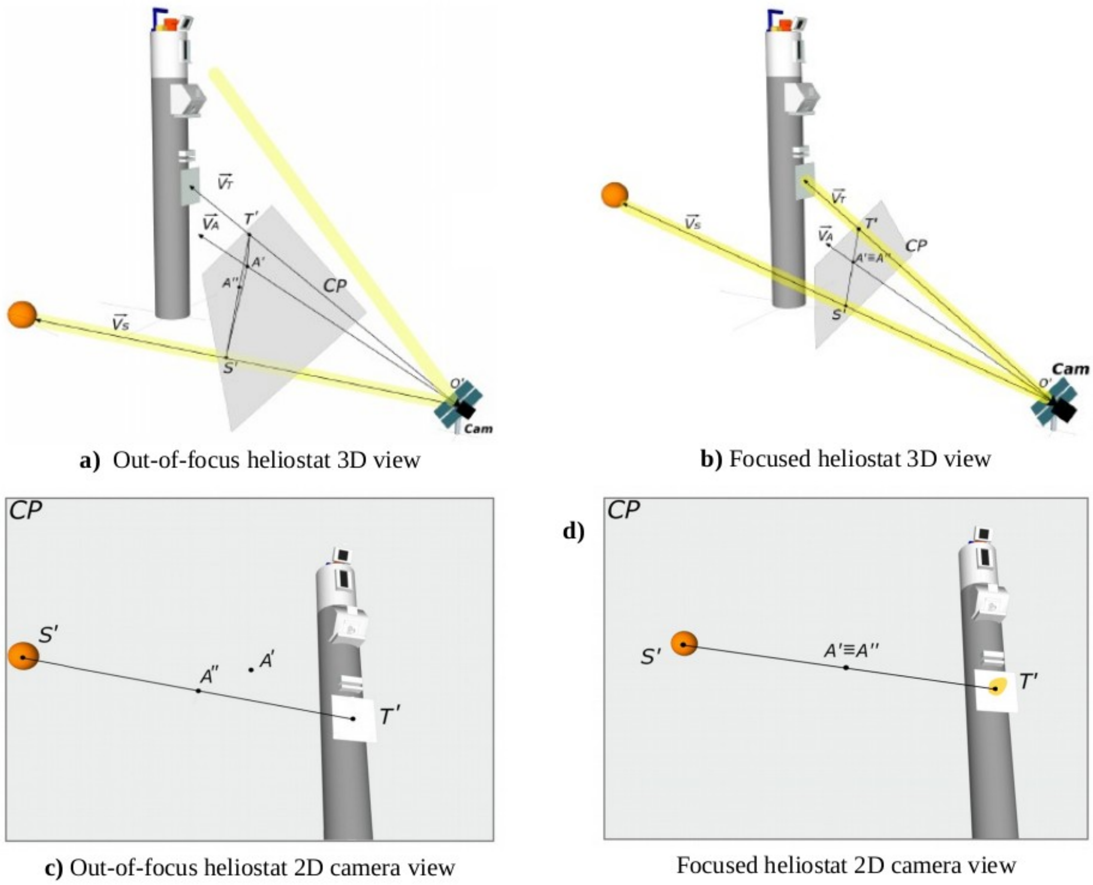

Figure 1. Hel-IoT Sun tracker approach

Hel-IoT hardware is just composed by a low-cost wide-angle camera and a low-cost low-power consumption computer with wireless (Wi-Fi to send and receive data and commands) and local (GPIO port to control the heliostat actuators) communication capabilities.

The tracking software has been programmed in Python [4], which is an open-source object-oriented programming language. Python is widely used for scientific computing. Tensorflow [5] can be used to add machine learning capabilities to the software; it integrates seamlessly with Python. Tensorflow is an open-source machine learning framework developed by Google. Tensorflow provides flexibility to design, implement and test different kinds of neural networks. Nevertheless, Hel-IoT is not limited to any particular hardware and software configuration. Machine learning applies statistical techniques to create algorithms and software that can learn from data (learning patterns) without being explicitly programmed to do it. Deep learning is a class of machine learning algorithms that learns data representation. Deep learning is commonly applied in computer vision, where in some cases outperforms humans on image and object recognition tasks [6].

Traditional solar tracking systems require strict and labour-intensive installation, expensive hardware, and periodic re calibration. Hel-IoT reduces the installation and maintenance cost. The hardware is low cost (< 135 €) and it is cheaper than current control system hardware in industrial heliostat trackers (> 780 $ ≈ 675 €) [7,8]. Furthermore, there is room for further cost reduction, for instance bulk purchases or using less powerful custom-tailored hardware. Hel-IoT does not require an accurate installation process like traditional sun tracker systems for heliostats. The installation cost could be further reduced by employing less expensive heliostat equipment. Fixed cost reduction could also lead to a reduction in the optimal heliostat area from an economic point of view. This will also translate into a cost reduction in the heliostat structure [8]. It decreases the operation cost. It is autonomous and does not require calibration which reduces the spillage, thus increasing the overall plant performance. Hel-IoT hardware is equipped with Wi-Fi technology, which further reduces installation costs by not needing communication wiring. Wireless communications are only needed to receive supervision and to send local additional information.

Hel-IoT also provides additional information to detect plant design issues, i.e. blocking and shadowing due to surrounding heliostats. Moreover, cloud detection provide data to optimally control the solar plant. The detection of clouds provides useful information to make predictions about the near future collected solar radiation and the overall plant production. This information enables to optimally control the solar plant and could help to keep the energy production stable, anticipating startups and shutdowns of the system. This is an active research area.

Hel-IoT web Server

As commented before, Hel-IoT is based on artificial intelligent, concretely a technique called object detection in where an artificial neural network is able to detect objects in images. For that, it is necessary to train a neural network with many previously labeled images. The tasks of creating the set of labeled images and then training the model with the image set can be very tedious and requires tools that help us. There are different tools developed for labeling, but all of them have a more general approach and are not adapted to Hel-IoT's requirements. For such reasons, Hel-IoT web server has been developed. It facilitates to create the labeled image set and automatically train the neural network model. Hel-IoT web server provides a friendly graphical environment that makes it possible for anyone without computer science knowledge be able to label images and retrain the model to improve its performance.

Server Architecture

In addition to the aforementioned software: Python and Tensorflow, Hel-IoT web Sever uses additional free and open source software such as Docker [9] for the automatization of the deployment of applications within software containers, providing an additional layer of abstraction and automation of application virtualization across multiple operating systems. NGINX [10] web server/reverse proxy, Flutter [11] for the web app development, CloudBeaver [12] database client for data management, analysis and administration and PostgreSQL object-relational database system [13], see Figure 2. Hel-IoT web Sever has access to the Plataforma Solar de Almería (PSA) servers where it stores the data and creates backup copies. It also has access to the meteorological information available on the PSA network.

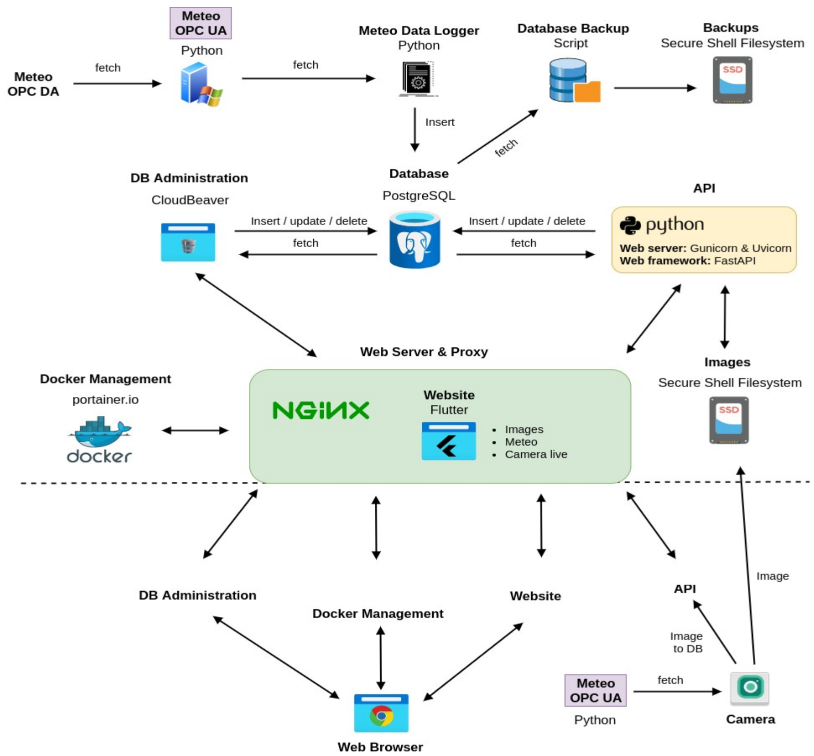

Figure 2. Hel-IoT web Server Architecture

Server Operation

On the one hand, Hel-IoT the server constantly stores meteorological data in the database, these data are available through API queries. On the other hand, a set of Hel-IoT systems, deployed in the CESA field in PSA, are taking images with low-cost cameras. Immediately after, the meteorological information obtained through the API is added to each image in its metadata. Then, the neural network of the Hel-IoT system processes the image to detect objects, sends the image to the store system and sends the object detection information to the server. Hel-IoT web server is responsible for managing the graphical user interface to inspect meteorological data and manage the set of images with the information on the detection of historical objects, see Figure 3.

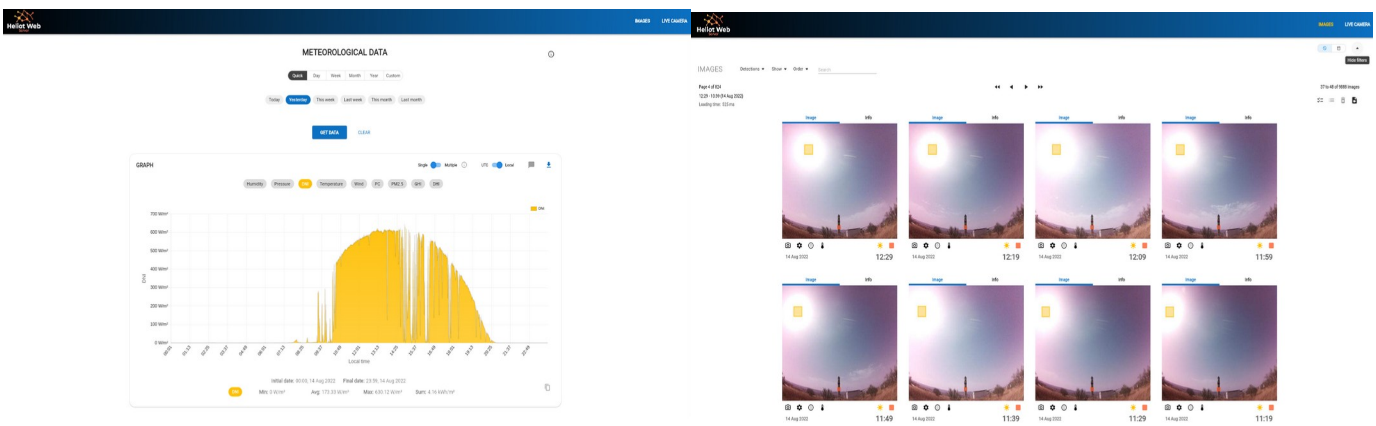

Figure 3. Hel-IoT web Server graphical user interface

Labeling and training

Hel-IoT websever provides an image labeling tool developed for Hel-IoT. Whit this tool we can label a new image or correct an image taken by one of the heliot systems deployed in the CESA field and available on the server, see Figure 4.

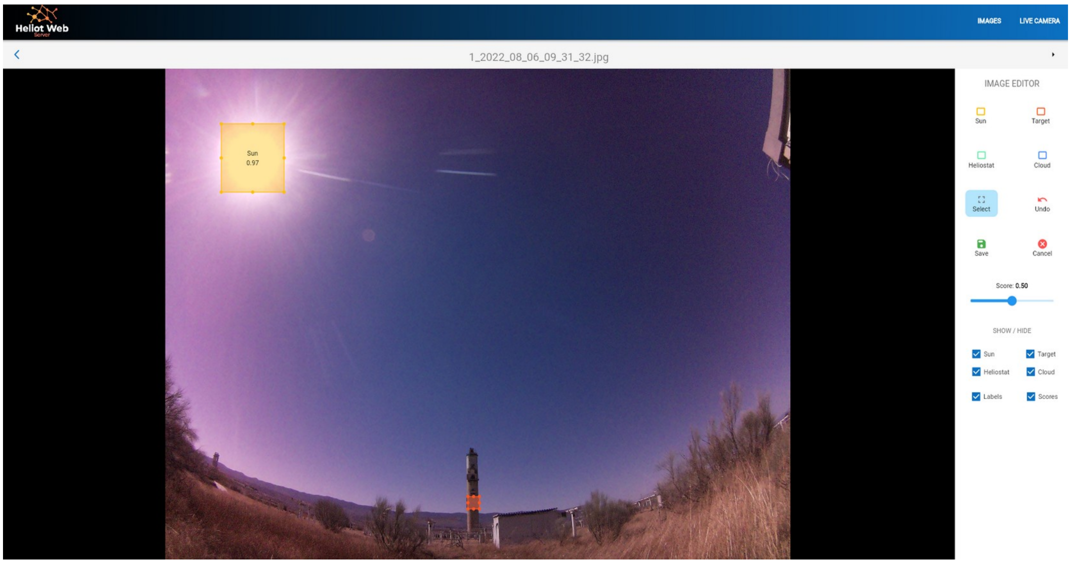

Figure 4. Hel-IoT labeling tool

From time to time, object detection predictions have to be corrected on certain images Images that are then manually corrected or re-labelled, and sent to the CETA cluster. When the cluster receives a new set of relabeled images, it retrains the model in order to improve object detection accuracy. If the model retrained with the new set of corrected images shows a better performance, according to established metrics in an evaluation images set, it is deployed to the Hel-IoT systemsin the CESA field.

Results

The system has been continuously running since April 2022, but we are still making some adjustments. Regarding the meteorological and image database, the system is robust and completely functional. All the data coming from different sensors or systems are available through the Hel-IoT web server.

Regarding online retraining, the full automation level of the process saves a lot of time and, although it is too early to obtain conclusions about the model evolution with such periodically retraining, it has been possible to appreciate an improvement in the performance over time. For example, in the first image from 23rd of April, we can see that the neuronal network model did not detect the Sun and the target detection was imprecise. On the contrary, in the second image on Figure 5 from 14th of August, the Sun and the target detection were accurate. Also in this image, the model detected three heliostats and a -low density cloud. This shows a substantial improvement thanks to the automatic successive retraining.

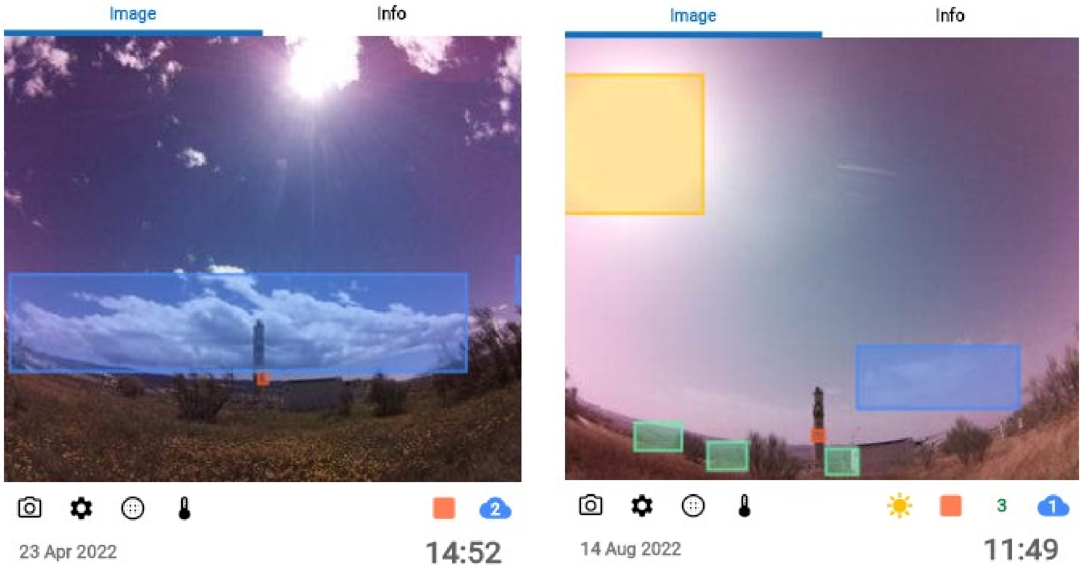

Figure 5. Neural network object detection evolution

Conclusions and future work

Hel-IoT web server has shown that it is a powerful tool to manage image and meteorological data for smart heliostat developing tasks. The labeling tool developed in this work adjusts perfectly to the required demands and reduces labeling time by allowing inaccurate or erroneous detections to be modified, in this way we don't need to label the whole image from scratch. Periodic retraining with corrected images can improve the model and reduce retraining time by only using images where the model did not perform well.

Future work includes that Hel-IoT web server automatically evaluate the performance of different models with a set of evaluation images, store and compare additional metrics to analyze in terms of precision and accuracy different models, cameras and embedded devices to improve the development of Hel-IoT.

Author contributions

Jose A. Carballo: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft

Javier Bonilla: Supervision, Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – review & editing

Jesus Fernández-Reche: Formal Analysis, Investigation, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing

Antonio Ávila-Marín: Formal Analysis, Funding acquisition, Investigation, Project administration, Supervision, Writing – review & editing

Diego-Cesar Alarcón-Padilla: Funding acquisition, Project administration, Writing – review & editing

Competing interests

The authors declare no competing interests.

Funding

This work has been funded by the National R+D+i Plan Project PID2021-126805OB-I00 (HELIOSUN project) of the Spanish Ministry of Science and Innovation funds and Plan Andaluz de Investigación, Desarrollo e Innovación (PAIDI 2020) of Consejería de Transformación Económica, Industria, Conocimiento y Universidades de la Junta de Andalucía funds. This work was also partially supported by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETA-CIEMAT), funded by the European Regional Development Fund (ERDF). CETA belongs to CIEMAT and the Government of Spain.

References

1. J.A. Carballo, J. Bonilla, M. Berenguel, J. Fernández-Reche & G. García, G. . New approach for solar tracking systems based on computer vision, low cost hardware and deep learning. Renewable energy, 133, 1158-1166, 2019,
2. J.A. Carballo, J. Bonilla, M. Berenguel, J. Fernández-Reche & G. García, G. Machine learning for solar trackers. In AIP Conference Proceedings (Vol. 2126, No. 1, p. 030012). AIP Publishing LLC.2019.
3. J. Bonilla, J.A. Carballo, J. Fernández-Recha, M. Berenguel, L. Valenzuela. Machine learning perspectives in concentrating solar thermal technology. 10th EUROSIM Congress, Logroño (Spain), 2019.
4. G. Rossum, Python reference manual, CWI (Centre for Mathematics and Computer Science),Amsterdam, The Netherlands, The Netherlands, 1995.
5. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, Corrado. G. S., A. Davis, J. Dean, M.Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, (2015). https://www.tensorflow.org/.
6. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252. doi:10.1007/s11263-015-0816-y.
7. F. Téllez, M. Burisch, C. Villasante, M. Sánchez, C. Sansom, P. Kirby, C. Caliot, A. Ferriere, C.A. Bonanos, C. Papanicolas, A. Montenon, R. Monterreal, J. Fernández, State of the art in heliostats and definition of specifications (Survey for a low cost heliostat development), 2017. doi:10.5281/zenodo.834887.
8. J.B. Blackmon, Parametric determination of heliostat minimum cost per unit area, Sol. Energy. 97 (2013) 342–349. doi:10.1016/j.solener.2013.08.032.
9. Docker. Docker, Inc. https://www.docker.com/. (08/2022)
10. NGINX. https://www.nginx.com/resources/wiki/ (08/2022)
11. Flutter. https://flutter.dev/ (08/2022)
12. Cloudbeaver . https://cloudbeaver.io/ (08/2022)
13. Postgresql. https://www.postgresql.org/ (08/2022)