

# Measuring Economic Activity From Space: A Case Study Using Flying Airplanes and COVID-19

Maurício Pamplona Segundo , *Member, IEEE*, Allan Pinto , *Member, IEEE*, Rodrigo Minetto , *Member, IEEE*, Ricardo da Silva Torres , *Member, IEEE*, and Sudeep Sarkar , *Fellow, IEEE*

**Abstract**—This work introduces a novel solution to measure economic activity through remote sensing for a wide range of spatial areas. We hypothesize that disturbances in human behavior caused by major life-changing events leave signatures in satellite imagery that allows devising relevant image-based indicators to estimate their impact and support decision-makers. We present a case study for the COVID-19 coronavirus outbreak, which imposed severe mobility restrictions and caused worldwide disruptions, using flying airplane detection around the 30 busiest airports in Europe to quantify and analyze the lockdown's effects and postlockdown recovery. Our solution won the rapid action coronavirus earth observation (RACE) upscaling challenge, sponsored by the European Space Agency and the European Commission, and now is integrated into the RACE dashboard. This platform combines satellite data and artificial intelligence to promote a progressive and safe reopening of essential activities. Code, trained model, and data are available at <https://github.com/maups/covid19-custom-script-contest>.

**Index Terms**—CNN-based object detection, COVID-19 pandemic, human and economic activity assessment, remote sensing.

## I. INTRODUCTION

OUR planet is experiencing an increase in disasters [1] and disease outbreaks [2] over the past decades. Therefore, developing methods and tools to provide meaningful information for assertive decision-making during emergencies, thus implementing safety and welfare measures, is of paramount

importance. Such a demand has led the research community to focus on finding indicators to support the different phases of emergency management [3]–[7]. One of the main challenges in designing an indicator, especially when its coverage area is beyond the country level, is collecting data that helps analyze a phenomenon and improves our understanding of its causes and symptoms. Information sources such as cell phone mobility data [8], airlines tracking information, social media, etc., have many limitations that hamper the process of integration, such as multiple providers, varying technologies, lack of infrastructure, sovereignty restrictions, and language mismatch, to list a few. We hypothesize that disruptions of this magnitude impact social behaviors and leave signatures in satellite imagery that can be automatically detected and quantified. And unlike other data sources, remote sensing stands out for its global coverage and versatility—it can integrate new indicators and new locations with little effort—while eliminating collection and format conversion complications. It is a single source for a variety of indicators across geographical boundaries. These characteristics are highly advantageous in fast response scenarios. Moreover, satellite imagery availability is growing fast, with several high-level application programming interfaces (API) offering numerous forms of access to different earth observation databases. Some examples are the Copernicus Open Access Hub,<sup>1</sup> Sinergise's Sentinel Hub,<sup>2</sup> Maxar's SecureWatch,<sup>3</sup> and Planet's Data API.<sup>4</sup>

The most recent global crisis, yet to be resolved, is the COVID-19 coronavirus pandemic. As of April 2021, this outbreak reached an unprecedented scale, with more than three million deaths and more than 140 million confirmed cases [9]. According to the United Nations' framework for the immediate socioeconomic response to COVID-19 [10], this critical period is far more than a health hazard. The socioeconomic impact is tremendous and will increase poverty and inequalities globally, jeopardizing lives and livelihoods, especially for vulnerable groups. Other issues include the lack of adequate social protection, losses in income and jobs, increased food insecurity, and a decline in global trades. These problems show the value of acting fast to mitigate adverse effects when a proper response is not timely possible and the recovery extension is unknown. To support such actions, we combine machine learning and satellite data to provide accessible COVID-19 information agilely.

Manuscript received April 26, 2021; revised June 9, 2021; accepted June 27, 2021. Date of publication July 2, 2021; date of current version July 28, 2021. This work was supported in part by the University of South Florida COVID-19 Rapid Response Grant Program and from the USF Institute for Artificial Intelligence (AI+X), in part by the US National Science Foundation under Grant CNS-1513126, in part by NVIDIA Corporation with the donation of the Titan Xp GPU used for this research, in part by the research Brazilian agencies CNPq, CAPES, and FAPESP under Grant 2014/12236-1, Grant 2015/24494-8, Grant 2016/50250-1, Grant 2017/20945-0, Grant 2019/16253-1, Grant 2019/17729-0, and Grant 2019/22262-3). (Corresponding author: Mauricio Pamplona Segundo.)

Maurício Pamplona Segundo and Sudeep Sarkar are with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33647 USA (e-mail: mauriciop@usf.edu; sarkar@cse.usf.edu).

Allan Pinto is with the University of Campinas, Campinas 13083-970, Brazil (e-mail: allan.pinto@ic.unicamp.br).

Rodrigo Minetto is with the Universidade Tecnológica Federal do Paraná, Curitiba 80230-901, Brazil (e-mail: rodrigo.minetto@gmail.com).

Ricardo da Silva Torres is with the Department of ICT and Natural Sciences, Norwegian University of Science and Technology, 6009 Alesund, Norway (e-mail: ricardo.torres@ntnu.no).

This article has supplementary downloadable multimedia material available at <http://10.1109/JSTARS.2021.3094053> provided by the authors.

Digital Object Identifier 10.1109/JSTARS.2021.3094053

<sup>1</sup>[Online]. Available: <https://scihub.copernicus.eu/>

<sup>2</sup>[Online]. Available: <https://www.sentinel-hub.com/>

<sup>3</sup>[Online]. Available: <https://www.maxar.com/products/securewatch>

<sup>4</sup>[Online]. Available: <https://developers.planet.com/>

Researchers explored different human signatures visible from space in the literature. Night-time lights disclose urbanization and population levels and can indicate wealthiness [11] and socioeconomic dynamics [12]. The tropospheric nitrogen dioxide concentration, primarily affected by fossil fuel consumption, directly correlates with economic activity variations [13]. The food supply chain is evaluable through land use classification of agricultural sites [14], [15] or transportation infrastructure monitoring [16], [17]. Along this line, aircraft are of particular interest to unveil human and economic activities (e.g., travel, tourism, freight) and track disease spread due to in-flight transmission [18]–[20].

For those reasons, aircraft detection is present in the most advanced aerial scene recognition benchmarks [21], [22], and there is extensive literature addressing airport operation, the great majority devoted to stationary aircraft [23]–[26]. However, the number of parked airplanes is not directly correlatable with airport traffic. For instance, Paris' Charles de Gaulle Airport and Rome's Leonardo da Vinci–Fiumicino Airport have, on average, approximately the same number of airplanes on the ground every day [27], even though the former has about 50% more flights in the same period [28]. Although flight traffic is available from sources other than satellite imagery, such as the OpenSky Network (OpenSky) [29] and Flightradar24<sup>5</sup>, Liu *et al.* [30] point out that no individual technology can reach global coverage or support all aircraft types. For instance, OpenSky tracks flight data through automatic dependent surveillance-broadcast (ADS-B) communications. For this reason, it cannot detect airplanes without an ADS-B transmitter or too far away from its ADS-B receivers (e.g., most of the ocean area and a large part of Asia and the South Hemisphere continents).

Traffic estimation using satellite images requires detecting flying aircraft, whose literature is not so developed. Zhao *et al.* [31] designed ingenious heuristics to perform this task using the water vapor absorption channel from LandSat-8 thanks to the observation that reflectance increases for high-altitude surfaces and generates bright spots on aircraft locations. Despite the high accuracy in ideal conditions, this method is affected by weather conditions (e.g., high clouds) and low altitudes (e.g., aircraft during takeoff and landing). Besides, Landsat-8's repeat cycle of 16 days [32] hinders the ability to use temporal data to cope with these difficulties. Heiselberg [33] and Liu *et al.* [30] utilized images captured by the Sentinel-2 satellites for the same job. These satellites' multispectral instrument design makes them observe the earth's surface at different times in each spectral band [34]. As the ground serves as a reference to merging bands, the resulting multispectral images (from MSI sensor on-board Sentinel-2) present interband measurement displacements due to parallax for objects at high altitudes and high-speed movement for objects at any elevation. Fig. 1 illustrates how these displacements create a colored pattern for flying airplanes in the three MSI bands of visible light. The works of Heiselberg [33] and Liu *et al.* [30] also relied on experts to handcraft heuristics for the airplane detection, which may prevent immediate analyses in outbreaks that require a fast response.

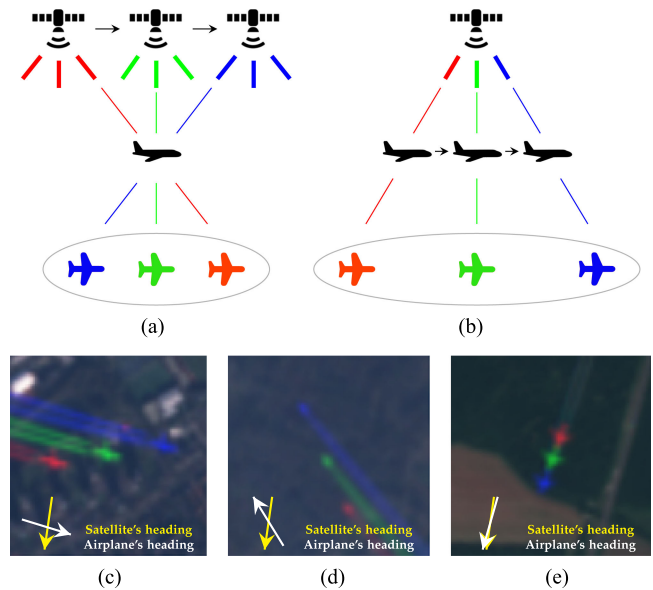


Fig. 1. Airplane color patterns in Sentinel-2 images: These satellites observe the ground surface at different times in each spectral band, creating (a) a parallax effect for airplanes at high altitudes and (b) a similar color separation effect for airplanes in high-speed at any elevation; both effects combined produce the colored patterns shown in figures (c)–(e). This figure contains modified Sentinel-2 data processed by Euro Data Cube.

In this work, we take advantage of the advances driven by deep learning algorithms [35]—bioinspired neural networks that learn representations with multiple abstraction levels and discover intricate patterns in massive data—to devise a dependable and adaptable detector. Our approach goes beyond the detection of flying airplanes as we use our airplane detector to build a time series of the number of landing or takeoff airplanes in airports. We then process those time series to estimate structural breaks (caused by lockdown restriction at the beginning of the pandemic) and to compute the recovery rate for the monitored airports. The presented framework supports decision-making during the COVID-19 pandemic by measuring how fast the airports are recovering toward getting into their normal operation and if such recovery is in accordance with opening and lockdown policies, which should be defined considering the number of cases and deaths of COVID-19 pandemic. Furthermore, our approach was designed to require a small amount of labeled data for training the proposed flying airplane detector, without loss of generality. We validate our approach by monitoring the 30 busiest airports in countries with some integration to the European Union. The results show the effectiveness of our solution to measure such activities and the recovery rate of such airports. Finally, we can adapt our approach to different global-scale applications, like tracking container ships in ports and cargo trucks on highways, which do not have other data sources not as readily available as air traffic data. One of the reasons for us to focus on airport traffic in this work is to have a ground truth data source to verify the accuracy of the proposed framework.

In summary, this study presents several contributions to the remote sensing research field and to aid society in facing the

<sup>5</sup>[Online]. Available: <https://www.flightradar24.com/>

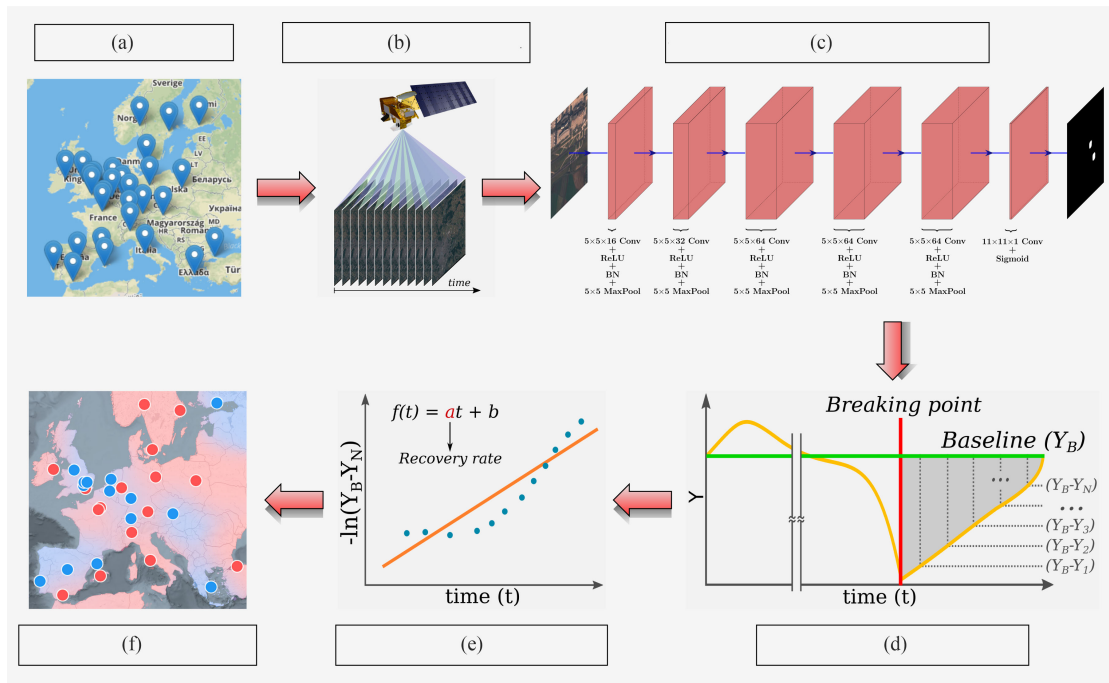


Fig. 2. Main steps of the proposed approach for detecting flying airplanes and to measure breakouts and recovery rates of airports activities. This figure contains modified Sentinel-2 data processed by Euro Data Cube. a) AOI selection. b) Image acquisition. c) Flying airplane detection. f) Activity indicator. e) Recovery estimation. d) Time series analysis.

negative aspects of the COVID-19 pandemic. First, this method is currently fully integrated into the *rapid action coronavirus earth* (RACE) observation dashboard [36], which is an open platform of the European Space Agency (ESA) that uses earth observation satellite data and artificial intelligence to measure the impact of the COVID-19 lockdown and to monitor postlockdown recovery. This study also shows how to design and train a shallow neural network to detect flying airplanes using remote sensing imagery and with a minimum amount of annotated images. The flying airplane detection task is still an open problem in the current literature, and this study contributes with the proposal of a new technique for this task that has been successfully used in a challenging and practical scenario related to combating the COVID-19 pandemic. The third contribution of this study relies on the construction of a new dataset, along with the ground-truth annotations of flying airplanes, to support future researches involving flying airplane detection through satellite images. The fourth contribution refers to the analysis of time series built with our flying airplanes detector to estimate breakouts and recovery rate, automatically, and to support decision-making to opening and closing airports taking into account the official numbers of COVID-19 cases and deaths. Finally, the source code of our solution, the trained models, and the annotated data are freely available to the scientific community, encouraging reproducibility of our results and the use of our solution in similar situations in the future.

## II. PROPOSED APPROACH

This study introduces a new approach for measuring airport activities using satellite to support decision-making during the

COVID-19 outbreak, as illustrated in Fig. 2. More precisely, we proposed a new method for detecting flying airplanes on Sentinel-2 satellite that enables the monitoring of airport activity in wide areas [Fig. 2(a) and (b)]. To do so, we use a fully convolutional network (FCN) [37], which relies solely on convolutional filters and has many successful applications in the literature, including semantic segmentation [37], object detection [38], and human pose estimation [39]. We devised a lightweight FCN architecture [Fig. 2(c)] containing few trainable parameters, and thus, requiring lower amounts of training data than other deep learning architectures available in the literature addressing the object detection problem [40]. We summarized the detected flying airplanes in a time series representation associated with the number of airplanes around each monitored airport over time (in days). Next, we performed a time series analysis to estimate breaking points associated with the first lockdown restriction related to the COVID-19 pandemic [Fig. 2(d)]. Then, we fitted a log-linear regression model to estimate the recovery rates for the airports considered in this study, which allowed us to correlate the estimated recovery rates with the cases and deaths of coronavirus disease in those locations [Fig. 2(e)]. Finally, we compiled this knowledge into an activity indicator [Fig. 2(f)] to support the conception, planning, implementation, and evaluation of disease-containment actions. These stages are detailed in the following sections.

### A. Data Collection

The first stage of this study consisted of creating a dataset to analyze human travel behavior in the European Union. Our analysis relies on detecting flying airplanes from satellite images

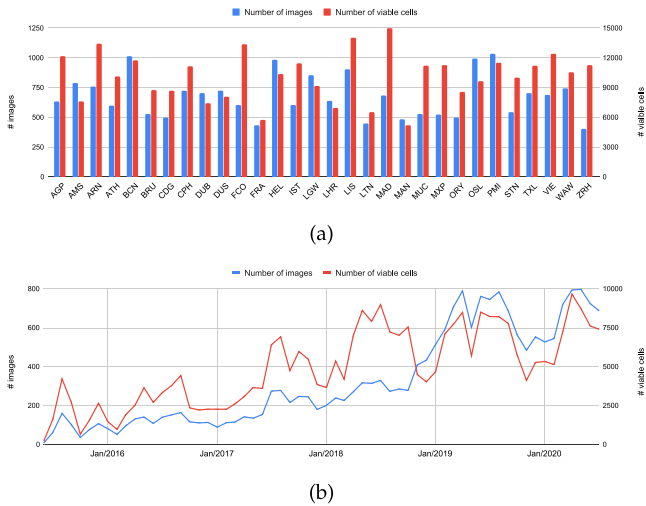


Fig. 3. Data collection statistics. (a) Total number of images and viable cells per airport. (b) Number of images and viable cells per month for all airports.

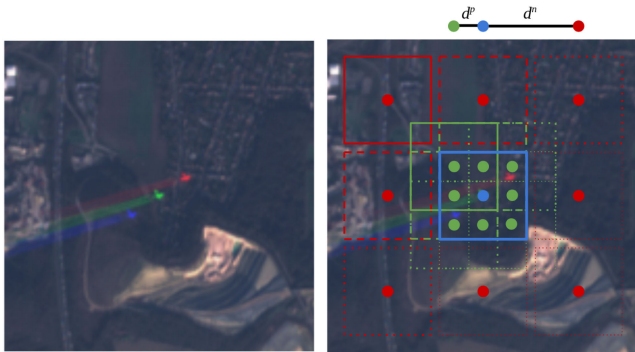


Fig. 4. Sampling strategy for training: for each annotation (point shown in blue), we extract positive image patches centered on blue and green points. Green points are  $d^p$  pixels away from the blue one in one or both axes ( $d^p = 3$ ). Negative image patches are centered on red points, which are  $d^n$  pixels away from the blue one in one or both axes ( $d^n = 25$ ). This figure contains modified Sentinel-2 data processed by Euro Data Cube.

and measuring the volume of flights over time. To capture traffic dynamics through remote sensing, the satellite must have a high revisit rate, it must perceive high-speed objects, and imaging resolution must be sufficiently high so that airplanes are visible. The Copernicus Sentinel-2 constellation meets all of these requirements [34]. The Sentinel-2 mission includes two identical satellites in the same sun-synchronous polar orbit,  $180^\circ$  apart from each other, that revisit any earth location every 2 to 5 days (higher frequency for areas close to the poles). They capture the visible bands of their multispectral images [red, green, and blue (RGB)] with a ground sampling distance (GSD) of 10 m and a time-lapse of approximately 0.5 s between consecutive bands (red–green and green–blue).

The proposed method operates on sequences of satellite images from areas of interest (AOI), which are rectangles with 1.05 longitude degrees in width and 0.7 latitude degrees in height, centered at the geographical coordinates of the 30 busiest airports (i.e., airports that had the highest number of passengers in 2019) in countries with some integration to the European

Union. Thus, the AOIs have an area of  $6000 \text{ km}^2$ , on average, and the selected airports cover 26 different cities and 18 different countries. Fig. 2(a) highlights their location, and the complete list of International Air Transport Association (IATA) airport codes is shown in Fig. 3(a). The specified AOI provides an observation window of approximately 20 min around each airport. More specifically, a satellite image captured at timestamp  $t$  shows flight arrivals in the range  $[t, t + 20 \text{ min}]$  and departures in the range  $[t - 20 \text{ min}, t]$  for the depicted airport.

We downloaded Sentinel-2 RGB images using the Sentinel-Hub engine [41]. We divided each AOI into a  $7 \times 7$  grid and evaluated each grid cell's viability considering the following criteria: 1) Cells cannot have more than 30% of cloud coverage; and 2) cells cannot have more than 10% of missing data. The average number of viable cells per image is 15. Fig. 3(a) shows the number of images and viable cells per airport, while Fig. 3(b) shows the total number of images and viable cells over time. These images were captured by Sentinel-2 satellites between June 26th, 2015 and July 30th, 2020.

### B. Shallow FCN for Detecting Flying Airplanes

Current state-of-the-art detectors either use region proposal or feature pyramid networks to estimate both bounding box coordinates and classes of objects in a scene [40], [42]. Such detectors take advantage of deep architectures [43] that contain hundreds of millions of trainable parameters, and thus, require large training data. With this in mind, we hypothesize that the use of shallow architectures is more adequate for modeling our problem due to the absence of a large training data with annotated ground truth. Shallow architectures have fewer trainable parameters, in comparison to deep architectures, and thus, require a reduced amount of labeled data during the training stage. It is important to notice that labeling data is a costly and a time-consuming process, which may become prohibitive when a rapid response is necessary. Furthermore, as we are only interested in counting airplanes, information like bounding boxes and airplane size, in terms of pixels, is not relevant to us. With this in mind, we can reduce the complexity of our architecture by modeling the problem of counting airplanes as a classification problem devised to classify each pixel of the image as being or not the center of a flying airplane [green dot in Fig. 1(c)–(f)].

To validate our hypothesis, we designed a shallow FCN [37] to produce a probability value for each pixel of an input image, as illustrated in Fig. 2(c). More precisely, the receptive field around each pixel was set to a region of  $51 \times 51$  pixels, which perceives airplanes traveling up to  $1800 \text{ km/h}$ , twice as much as the typical commercial cruise speed. This surplus handles the variability introduced by parallax and altitude changes. The proposed shallow FCN architecture consists of five consecutive  $5 \times 5$  convolutional layers, each followed by a rectified linear unit activation [44], batch normalization [45], and a  $5 \times 5$  max pooling. Then, we added an  $11 \times 11$  convolutional layer with sigmoid activation to output values between 0 and 1. All layers use unit strides so that the output resolution is the same as the

**Algorithm 1:** Pseudocode for Training Our FCN Model.

```

Input: Training images, Flying airplane annotations
Result: FCN model that detects flying airplanes on satellite images
1 Randomly initialize the learnable parameters of the FCN model;
2 Follow the strategy shown in Figure 4 to build an initial set of training samples by using airplane annotations to extract
  image patches with  $51 \times 51$  pixels of positive and negative samples from the input images; extract additional negative
  patches from random locations that are at least 25 pixels away from all annotations until a 1:2 ratio between the amount
  of positive and negative samples is reached;
3 Set  $score = 0$ ;
4 repeat
5   Run one training epoch (3,000 model update iterations) using: (1) mini-batches with 256 random samples, (2) random
     flips (horizontal and vertical) and 90-degree rotations over samples as data augmentation operations, and (3) the
     Adam algorithm with a learning rate of  $10^{-4}$  to optimize the binary cross-entropy loss function;
6   Run the current model over the training images and save the list of detected airplanes;
7   Initially mark all detections as false alarms;
8   foreach annotated airplane do
9     Find the closest detection to the current annotation and mark this detection as a true positive if the distance
     between them is less than or equal to 25 pixels (half of the patch size);
10  end
11  Compute the detection rate  $DR$  as the number of detections marked as true positives divided by the number of
     annotated airplanes;
12  Compute the false discovery rate  $FDR$  as the number of detections marked as false alarms divided by the total
     number of detections;
13  Set  $score = DR(1 - FDR)$ , and save the current model if  $score$  increases;
14  Update the set of training samples by randomly replacing up to half of the negative patches with  $51 \times 51$  patches
     extracted from detections annotated as false alarms in this iteration;
15 until  $score$  does not increase for  $N$  epochs or the maximum number of epochs  $M$  is reached;

```

input. This architecture has a total of 277 745 learnable parameters, which correspond to a model size of only  $\sim 1.1$  MB. Finally, nonmaximum suppression [46] returns unique detections, and a threshold of 0.5 selects the ones that most likely represent an airplane.

Due to the global nature of the COVID-19 pandemic, the generalization is an important aspect to consider. More precisely, it is essential to detect flying airplanes in unknown areas without the need for retraining our model. To evaluate this competence, we used the airports with the first 15 IATA codes in alphabetical order for training [see Fig. 3(a)], and remaining 15 airports were used for testing only. With this split we can measure our detector’s performance in airports that were not seen during training to validate its generalization capability. Besides, we only used images from January 1, 2020 to June 30, 2020 for training. Thus, we can also use images outside this time period to evaluate our detector’s behavior in unseen images from known areas (training airports).

We adopted a semiautomatic strategy to obtain enough annotations even when a rapid response is required. First, we manually annotate as many instances of the target object as possible in a small number of images (in this work, we annotated 190 flying airplanes in 18 images from the Charles de Gaulle Airport). Then, we gradually expand this set of annotations by alternating between training a model with the existing annotations and manually inspecting the detection results to update the annotation set. In our case, we trained an initial model using our 18 annotated images and used it to detect airplanes in all training images. We removed all false alarms from the set of detected airplanes through a visual inspection, and used the coordinates of the remaining detections as the new set of annotations. We repeated this process once more, but this time we also adjusted the detected coordinates to overlay the green dot of the flying

airplane pattern [see Fig. 1(c)–(f)]. We ended up with 1782 flying airplanes in our final annotation set, which was used to train our final detection model. Every training repetition was carried out by Algorithm 1, with  $N = 1$  and  $M = \infty$  for temporary models and  $N = 10$  and  $M = 50$  for the final detector.  $N$  is a patience parameter used to stop the training early if the model does not improve for  $N$  consecutive epochs, and  $M$  is the maximum number of epochs.

### C. Time Series Generation

As shown in Fig. 7, satellite images do not always cover the entire AOI due to cloud occlusions or to a misalignment between the satellite visible area and the AOI. Thus, estimating the airplane count within a day is not accurate enough for further calculations. We alleviate this problem by using a temporal window  $w$ . The set of satellite images  $I_w$  within this window are used to produce an average airplane count  $C_w$ , as

$$C_w = \sum_{i=1}^7 \sum_{j=1}^7 \frac{\sum_{k \in I_w} c_{ij}^k}{\max\{1, \sum_{k \in I_w} v_{ij}^k\}} \quad (1)$$

where  $c_{ij}^k$  is the airplane count for the cell in the  $i$ th row and  $j$ th column of the  $k$ th image, and  $v_{ij}^k$  is 1 if this cell is viable and 0 otherwise. As can be seen, we compute averages at cell level for our  $7 \times 7$  AOI grids and then sum all cell averages to obtain a count estimate at image level. We use a window size of 30 days with a step size of one day to create our time series [see examples in Fig. 9(c)].

In some cases, satellite images may present some artifacts caused by the misalignment between color bands or sun-glint [see Fig. (5)]. Although their occurrence is rare, these artifacts

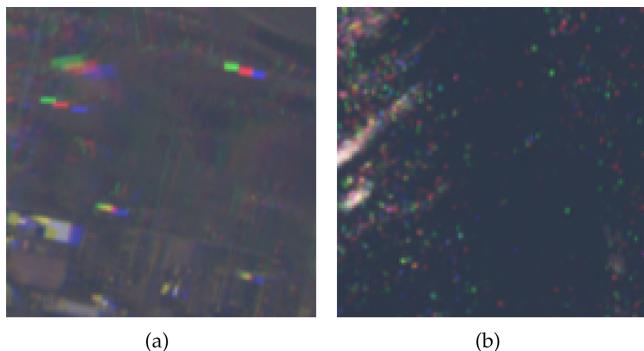


Fig. 5. RGB noise patterns in satellite images caused by (a) misalignment between color bands or by (b) sun-glint. This figure contains modified Sentinel-2 data processed by Euro Data Cube.

tend to produce several false positives close to each other. If ignored, this problem considerably affects individual cell averages and the final airplane count. To cope with these noisy regions, given that no grid cell in the training set has more than four annotated airplanes (considering the final annotation set used for training), we ignore cells that have more than five detected airplanes (set  $c_{ij}^k = 0$  and  $v_{ij}^k = 0$  if  $c_{ij}^k > 5$ ).

#### D. Time Series Analysis

To estimate the airports' activity in normal and exceptional periods, including the COVID-19 outbreak, we proposed a temporal analysis of time series built from the number of airplanes flying around airports considered in this work. We adopted the use of the concept of the structural breaks, which has been applied in other remote sensing-related problems [47]–[49]. Thus, we characterize the breaks as a position in the time series in which an abrupt shift (or decrease) is observed [47].

To detect a breaking point in the time series, we adopted two approaches: The simple moving-average (SMA) crossover [50] and Twitter's breakout detection [51]. The simple SMA crossover comprises two simple moving average to follow short-term and long-term tendencies. While the short-term moving average is more reactive to variations, the long-term moving average aggregate changes over a long time, and thus, produced a smoothed curve. When the curve resultant of these two moving average crosses, then we might have a change of tendency. In turn, the Twitter algorithm employs the E-divisive with medians (EDM) method [52] to automatically detect breakouts in time series. The authors employed the E-statistics to locate changes in mean without any assumption regarding the data distribution. According to authors, this approach was designed to work in presence of anomalies. A more in-depth discussion regarding the mathematics and statistical theories regarding this method can be found in the literature [51], [52]. We evaluate these two algorithms to detect a breaking point in our time series to find the instant  $t$  (in days) in which the airport starts recovering from a very low activity state.

After finding the breaking point in time series for each airport, the next stage of our analysis consists of computing the recovery rate of airports' activity [53]. First, we computed a baseline  $Y_B$ ,

which is the average of the short-term moving average. Then, we define the recovery rate using the following exponential model [54], as follows:

$$\frac{dy}{dt} = -\lambda(Y_B - Y_t) \quad (2)$$

where  $Y_t$  is the observed trend in the instant  $t$  and  $\lambda$  is the recovery rate computed by fitting a linear regression of  $-\ln(Y_B - Y_t)$  against time. Fig. 2(d) and (e) illustrates the methodology used to compute the recovery rate. In Fig. 2(d), for a given time series, the baseline  $Y_B$  is shown as a green line and the breaking point as a red line. Then, we compute the differences between the baseline and all  $Y$  values after the breaking point. These values are transformed into a logarithm scale and used to fit a linear regression [Fig. 2(e)]. Finally, the slope coefficient of the fitted line represents the recovery rate for that time series.

### III. RESULTS

We trained our neural network models on a single node of a GPU cluster, with four NVIDIA GTX 1080 Ti GPUs and 128 GB RAM. We ran our inference experiments in a single machine with an Intel Core i7-6800 K CPU, an NVIDIA Titan Xp GPU, and 128 GB RAM. We can process a month of data for the 30 AOIs considered in this work in less than 12 h with this configuration. This amount of time includes: 1) Downloading cloud masks, data masks, and RGB images; 2) running our detector on the RGB images; and 3) generating and analyzing time series.

#### A. FCN-Based Detection Results

To assess the effectiveness of our approach to detect flying airplanes, we designed two experiments in order to measure detection errors in practical scenarios: 1) Visual inspections around each detection to determine the proportion of false alarms and 2) comparison to existing publicly available flight records to quantify the occurrence of false negatives. In the context of this work, false alarms occur when our method classifies background patches as a flying airplane, while false negatives occur when the method classifies flying airplane patches as background.

1) *Visual Inspection to Quantify False Alarms*: We conducted an error analysis to determine the number of false alarms of our method in detecting flying airplanes by visually inspecting the detection results in images that were not seen during training. To this end, we first extracted patches around each detection in images captured between June 2015 and December 2019 from the 15 training AOIs. After manually classifying each patch as a true positive ( $TP$ ) or a false alarm ( $FA$ ), we compute the false discovery rate ( $FDR = \frac{FA}{TP+FA}$ ) for unseen data from AOIs seen during training. We adopted this strategy because it allows accurately estimating FDR in the absence of ground truth annotations. From the total number of 25 747 detections, 410 were false alarms, which corresponds to a 1.59% FDR. This results in 7.6 false alarms per month, i.e., less than one false alarm per airport each month. Knowing that each AOI has an average of 10 images per month and each image has up to

20 million airplane candidates, the incidence of false alarms in areas observed during training is minimal.

To evaluate the generalization capability of our detector operating in unknown areas, we repeated the previous analysis using the images captured between June 2015 and June 2020 from the AOIs not seen during training. There were 616 false alarms in the midst of 30 958 detections, a 1.99% FDR. The average number of false alarms per month was 10.1, which again resulted in less than one false alarm per airport each month. These values are similar to the ones reported for training AOIs and show that our detector is equally applicable to AOIs that were not seen during training but have a similar setup.

We show several examples of regions depicting correct detections and false alarms in Fig. 6. These examples show the robustness of our detector to a wide range of variations and illustrate the most common causes of misdetection.

2) *Analysis of Publicly Available of Flight Records to Quantify False Negatives:* Given the vast amount of data collected in this work, a comprehensive annotation of flying airplanes that allows a precise estimation of the number of false negatives is unfeasible. Thus, to quantitatively measure the detection accuracy, we used publicly available flight track records from the OpenSky [29] as a reference, as it provides extensive coverage of the European continent. To illustrate one advantage of this study, the steps to adapt our framework to the African continent are 1) listing new AOIs and 2) downloading satellite images for them. Meanwhile, OpenSky currently lacks coverage in this area and could not be used in such rearrangement.

In our experiments, we retrieved all OpenSky records whose portrayed airplanes are inside one AOI at the same moment that the area is imaged by a Sentinel-2 satellite from January 2020 to June 2020. As shown in Fig. 7, these records not necessarily include all airplanes that appear in one image, either due to the absence of tracks for some flights or to the registering of incomplete tracks. Also, not all airplanes whose tracks are available within a certain AOI can be detected by our approach. In most cases, this occurs because these airplanes appear over nonviable cells. In other cases, small airplanes are not visible in the satellite image.

For this analysis, we created two time series with monthly estimates of the number of flights for each AOI, one using our detection results and the other using the OpenSky records. The time series for training and testing AOIs are presented in Fig. 8(a) and (b), respectively. When comparing the two series of the same AOI, we obtained a root mean squared error (RMSE) of 2.9 and a mean signed deviation (MSD) of  $-0.8$  on average for training AOIs, and a 2.7 RMSE and a  $-1.4$  MSD on average for testing AOIs. Once again, the results for known and unknown AOIs are close and reinforce our detector's generalization power. The RMSE values show that, when looking into individual AOIs, our results are comparable to OpenSky records, exhibiting similar volume of flights and trends in airport activity. The MSD values reveal that our estimates are consistently lower than their respective OpenSky values. This negative bias approximates the average number of false negatives per image. These false negatives, however, can either be caused by detection failure [Fig. 7(a)] or data absence [Fig. 7(b)].

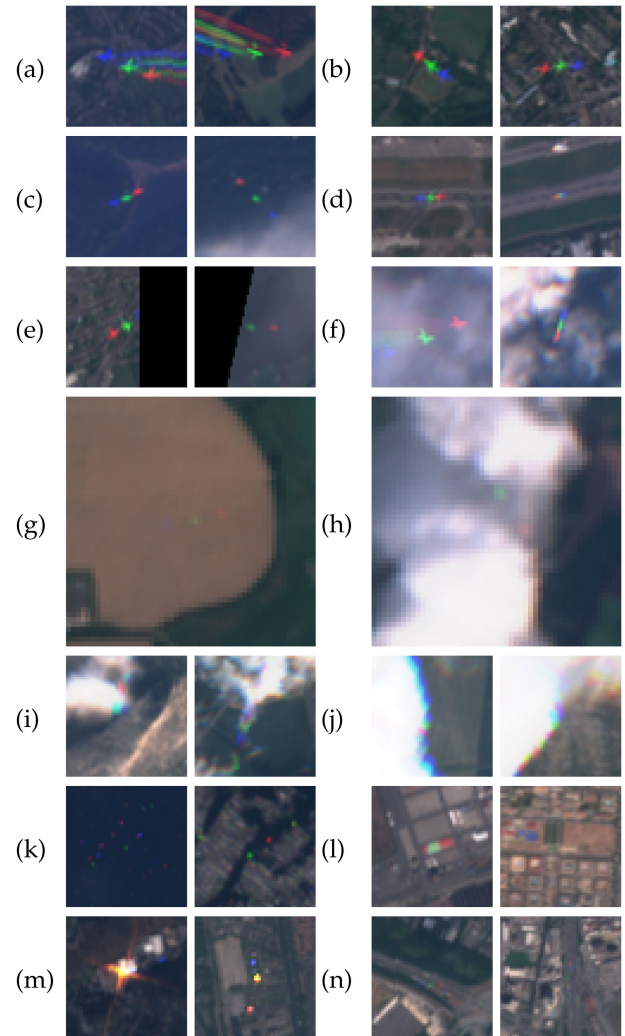


Fig. 6. Examples of satellite image patches surrounding automatic detections. Samples classified as **true positives**. (a), (b) Large airplanes, whose shape contours are visible, some of them with contrails. (c) Medium airplanes, which are still highly visible but have no shape cues. (d) Airplanes at low-speed (captured during landing or take off). (e) Incomplete airplanes due to missing data. (f) Moderate cloud occlusions. (g) Small airplanes, which are barely visible even after further magnification. (h) Severe cloud occlusions. Samples classified as **false alarms** caused by. (i) Image stitching in cloud areas. (j) Parallax in cloud edges. (k) Sun-glint over water surface. (l) Colored buildings. (m) Ground lights. (n) Movement patterns from vehicles in roadways. This figure contains modified Sentinel-2 data processed by Euro Data Cube.

If we consider the average time series from several AOIs [thick black lines in Fig. 8(a) and (b), the RMSE is reduced by two thirds for known airports (from 2.9 to 1.0) and by half for unknown airports (from 2.7 to 1.4). By averaging multiple AOIs from different countries, we not only obtain stabler series but also integrate airport activity at continental level. To illustrate how accurate those measurements can be, we recreate the real number of flights per month from 42 European countries (data from March to September of 2019 and 2020 made available by Eurocontrol [28]) using the monthly average of the 30 AOIs considered in this work (data from August to September of 2020 generated by the live version of our approach at the RACE dashboard [36]). As both series are in different scales,

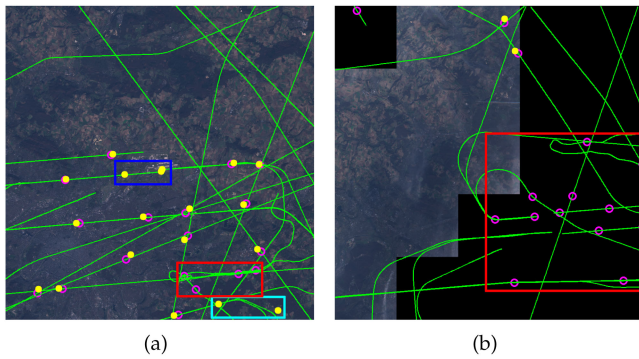


Fig. 7. Visual comparison between our detection results (filled yellow circles) and the OpenSky Network flight records (unfilled pink circles). The green lines represent the OpenSky tracks, from which we trace the location of the airplanes at the image acquisition timestamp. Many airplanes are depicted in both circle styles, and the displacement between yellow and pink circles is caused by different factors (e.g., airplane altitude and speed, satellite viewpoint). Sometimes detected airplanes do not have a corresponding OpenSky location due to incomplete tracks during take-off and landing [blue rectangle in (a)] or to missing track records [cyan rectangle in (a)]. In turn, some OpenSky locations were not detected by our method because airplanes were too small [red rectangle in (a)] or because they were located over nonviable cells [red rectangle in (b)]. This figure contains modified Sentinel-2 data processed by Euro Data Cube.

we normalized them by their maximum values in the considered period, and the outcome is presented in Fig. 8(c). This result shows that our monthly estimates are directly proportional to the real number of flights. Thus, despite noisy, our time series for individual AOIs fluctuate around their expected values and allow us to conduct AOI-specific analyses.

### B. Time Series Analysis Results

This section presents the time series analysis applied in this study to discover outbreaks in airports' activities caused by the COVID-19 pandemic and to estimate their recovery rate. In both analyses, we use the proposed flying airplane detection algorithm to build a time series associated with the number of flying airplanes in each AOI considered in this study.

1) *Structural Break Detection*: We assess the effectiveness of both SMA crossover [50] and Twitter's breakout detection [51] by analyzing their parameter space and then selected the best technique for our problem considering their best configuration. To measure the effectiveness of such methods to detect structural breaks, we considered the date May 1st as being the observed breaking point, which reflects the period in which the airports presented a significantly reduction in their operations. We measure the ability of both techniques in detecting these breaking points by computing the mean absolute error (MAE) and the RMSE between the observed and predicted breaking points.

For the SMA crossover method, we analyzed the short-term parameters considering values ranging from 7 to 49 days and the long-term parameters taking values ranging from 14 to 98 days. From these experiments, we could observe that SMA crossover method presented a better performance using a 14-day window size for the short-term parameter and a 49-day window size for the long-term. For the Twitter algorithm, we analyzed its two main parameters, *m*size and *beta* parameters. For the *m*size, we

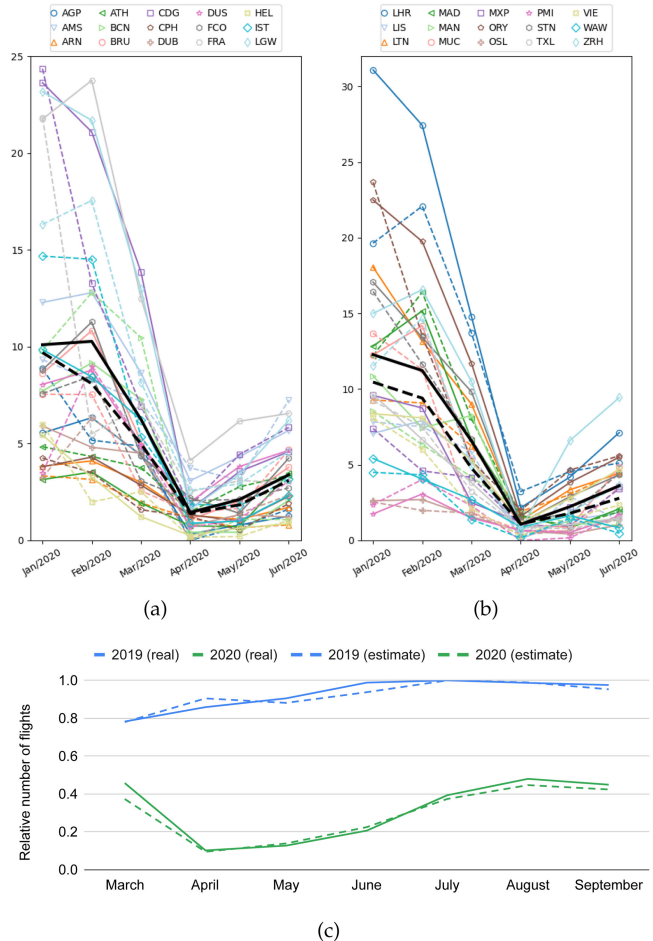


Fig. 8. Average number of flights per month computed using our detector (dashed lines) and OpenSky records (solid lines) for (a) known and (b) unknown airports; thick black lines show the average of all airports in the same chart. (c) Relative number of flights from March to September of 2019 and 2020 computed using Eurocontrol's data for 41 EU countries (solid lines) and our framework for 30 airports (dashed lines).

consider eight values ranging from 64 to 128 days, and for the *beta* parameter (penalization parameter), we consider ten values ranging from 0.1 to 1.0. Before applying the Twitter algorithm for detecting breaking points, we first use a smoothing technique to remove random variations, and thus, reveals underlying trends clearly. From these experiments, we could observe a better performance considering a *m*size and *beta* parameter values of 64 and 0.2, respectively. As a result, we observed a 19.7 MAE and a 23.4 RMSE for the SMA crossover method, and a 19.9 MAE and a 26.5 RMSE for the Twitter algorithm. Both metrics indicate the prediction quality, with errors ranging from 0 to infinity and lower values being better.

Once the algorithms returned quantitative values, we adopted the use of the Wilcoxon signed-rank (WSR) test statistic to verify if both algorithms are statistically different. The WSR test is a nonparametric test used to assess the null hypothesis that two related paired samples come from the same distribution [55]. More precisely, we computed the breaking points using both algorithms for the 30 airports considered in this study. Then, we converted the detected breaking point dates into the Julian



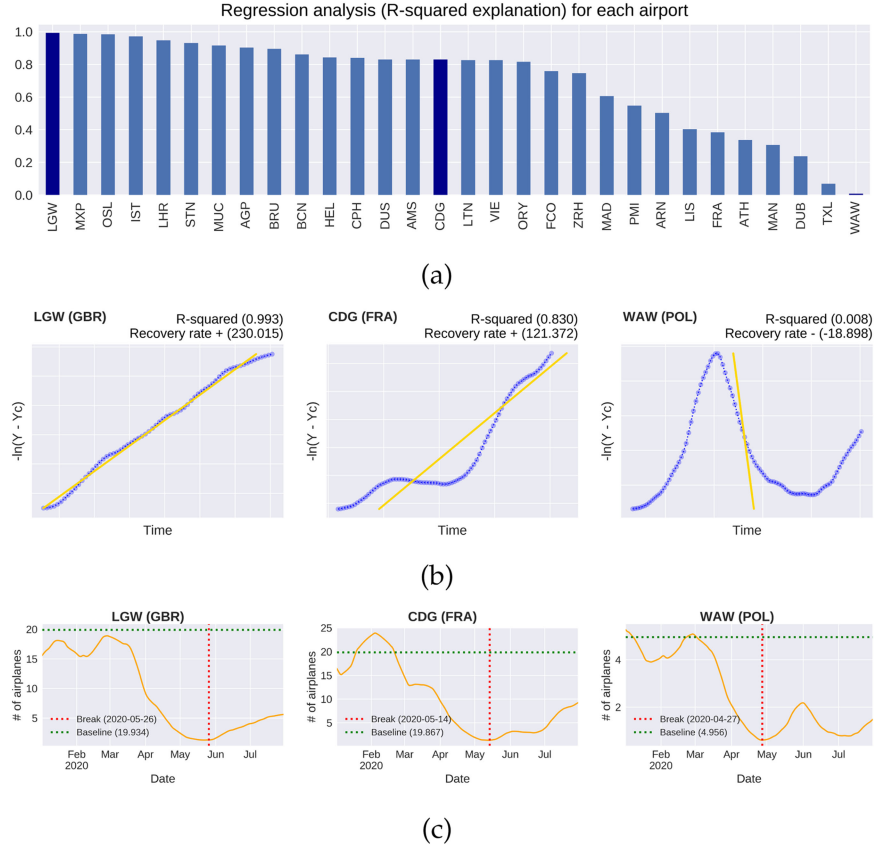


Fig. 9. Top figure shows (a) the performance results (in terms of R-squared) obtained after fitting a log-linear model to each airport data. The second row (b) shows examples in which we could fit a perfect log-linear regression model and thus compute the recovery rate (LGW and CDG), and an example in which we could not fit a log-linear regression model (WAW), which suggests that this airport does not present a clear recovery pattern. Finally, the third row (c) shows the time series for these three examples as well as the baseline and detected breaking point.

format, and then we applied the WSR test to check if both algorithms are statistically different. The obtained p-value confirmed that the differences between the two algorithms' results are statistically significant, considering a confidence level of 95%. Thus, from hereon we use the best configuration of the SMA crossover since this method presented a better performance.

2) *Recovery Rate and COVID-19 Analysis*: To compute the recovery rate, we fitted a log-linear regression model considering a linear regression algorithm with a mean squared error (mse) as a cost function. In short, we estimated the structural break for each time series and their respective baseline, which corresponds to the average number of flying airplanes. Then, we computed the log of the difference between the daily estimations of counted airplanes and the baseline value. The estimation of log-linear regression models considers such differences (in log scale) as a dependent variable and the data timestamp as an independent variable.

To measure the goodness of fit of regression models to estimate the recovery rate for the airports, we adopted the R-squared metric, also known as coefficient of determination, which ranges from 0.0 to 1.0 and measures the proportion of variance in the independent variable explained by dependent variables

$$R - \text{squared} = 1 - \frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{\sum_{t=1}^n (y_t - \bar{y})^2}, \quad \bar{y} = \frac{1}{n} \sum_{t=1}^n y_t \quad (3)$$

where  $y_t$  and  $\hat{y}_t$  are the observed and predicted values for the time  $t$ . In this context, an R-squared of 0.0 means that the fitted model does not explain any variation in the independent variable around its mean, while an R-squared of 1.0 means that the fitted model explains all variations in the independent variable around its mean, i.e., the obtained regression model fitted all data points [55].

Fig. 9(a) shows the performance of the fitted models in terms of R-squared, from which we could observe values higher than 0.8 for several airports. Also, Fig. 9(b) and (c) illustrate airports with positive (first two columns) and negative recovery rate (last column) and their respective time series from January 1st, 2020 to July 30th, 2020. Only two airports presented a negative recovery rate in the considered period (MAN and WAW), meaning that their activities was still decreasing after the breaking point.

Finally, we present an analysis of the COVID-19 situation and the necessary strictness for border control policies by correlating the recovery rates and the 14-day moving average of the daily number of new cases and deaths, as illustrated in Fig. 10. This study considered the official number of cases and deaths reported until July 30th, 2020 [56], [57]. When we correlated the recovery rate and the daily number of new cases of COVID-19, we observed a strong positive correlation between both variables for VIE, PMI, and ZRH. This suggests that these airports restarted their activities while the number of new cases

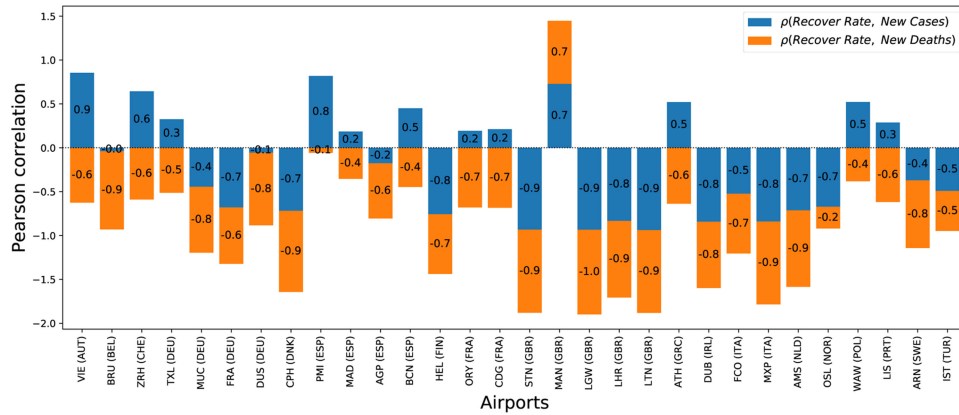


Fig. 10. Correlation analysis between the recovery rates and the total number of cases and deaths caused by the COVID-19 disease.

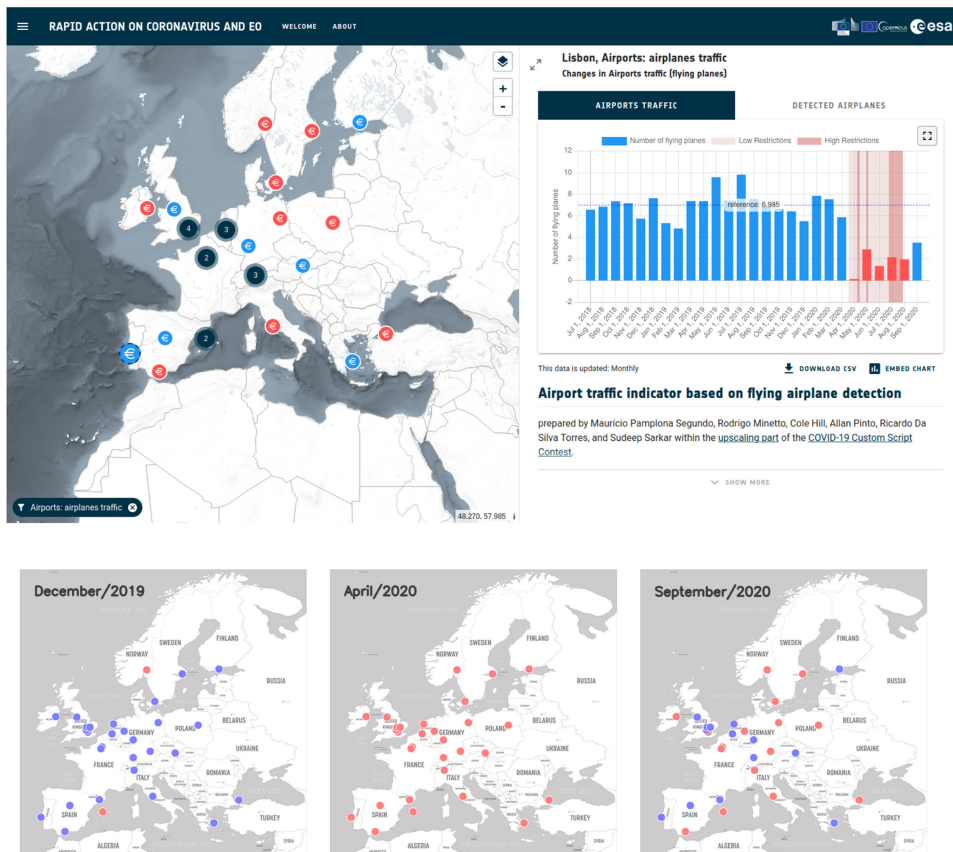


Fig. 11. RACE observation dashboard. The map shows the location of the 30 airports being monitored. Red and blue circles in the map indicate low and normal traffic levels, respectively, and dark blue circles show the numbers of airports in that region. The bar chart in the right shows the results of our method for a specific airport, which comprises the time series of the number of flying airplanes. Finally, the three maps in the bottom shows the traffic levels for three time periods: before the COVID-19 pandemic (Dec. 2019); early period after the pandemic, with mobility restrictions (Apr. 2020); and after a gradual reopening (Sep. 2020).

was still increasing. Conversely, we observed a strong negative correlation between the recovery rate and the daily new cases of COVID-19 for several airports such as MUC and DUS airports from German, and the SNT, LGW, LHR, and LTN airports from Great Britain.

Of course, the positive correlations found in our analysis do not imply the causation of the rise in the total number of cases and deaths. However, airports expanding their activities while the total number of cases and deaths caused by COVID-19

is increasing can negatively impact the pandemic's trajectory. Therefore, we can surely state that the indicator of activity presented in this work can reveal which airports and countries demand more attention from the authorities.

#### IV. CONCLUSION

The measurements of human activity are nowadays an essential task for planning actions to fight against huge outbreaks that

impact human daily activities as the COVID-19 pandemic. In this context, the proposed approach for measuring airport activities can serve society as a valuable and independent indicator of human activity without any political biases. Our solution now integrates the RACE observation dashboard [36], a platform from the ESA that uses Earth observation satellite data and artificial intelligence to measure the impact of the coronavirus lockdown and to monitor postlockdown recovery, as illustrated in Fig. 11.

Our detector was able to locate most airplanes appearing on satellite images with minimal false detections. On average, we get less than one false detection per month in each airport. Furthermore, our time series analysis can be used to identify abnormal behaviors in air traffic and correlate changes in the number of airplanes with COVID-19 statistics. Decision-makers can use this information to substantiate border control and lockdown measures.

Thanks to our semiautomatic annotation strategy, this approach can be straightforwardly adapted to detect other objects, such as ships [33] and transportation trucks [58], and use this outcome to devise other informative indicators.

#### REFERENCES

- [1] "The human cost of disasters: An overview of the last 20 years (2000–2019)," United Nations Office for Disaster Risk Reduction, Geneva, Switzerland, Tech. Rep., 2020.
- [2] K. F. Smith *et al.*, "Global rise in human infectious disease outbreaks," *J. Roy. Soc. Interface*, vol. 11, no. 101, 2014, Art. no. 20140950.
- [3] C. Raymond *et al.*, "Understanding and managing connected extreme events," *Nature Climate Change*, vol. 10, no. 7, pp. 611–621, Jul 2020.
- [4] R. Eyre, F. De Luca, and F. Simini, "Social media usage reveals recovery of small businesses after natural hazard events," *Nature Commun.*, vol. 11, no. 1, Apr 2020, Art. no. 1629.
- [5] T. Simon, A. Goldberg, and B. Adini, "Socializing in emergencies—A review of the use of social media in emergency situations," *Int. J. Inf. Manage.: J. Inf. Professionals*, vol. 35, no. 5, pp. 609–619, 2015.
- [6] C. Betsch, V. Colizza, S. del Valle, C. Ihekweazu, and C. Troncoso, "A vision for actionable science in a pandemic," *Nature Commun.*, vol. 11, no. 1, Sep. 2020, Art. no. 4960.
- [7] R. Minetto, M. P. Segundo, G. Rotich, and S. Sarkar, "Measuring human and economic activity from satellite imagery to support city-scale decision-making during COVID-19 pandemic," *IEEE Trans. Big Data*, vol. 7, no. 1, pp. 56–68, Mar. 2021.
- [8] S. Chang *et al.*, "Mobility network models of COVID-19 explain inequities and inform reopening," *Nature*, vol. 589, no. 1, pp. 82–87, Jan 2021.
- [9] "Worldometer—Coronavirus," Accessed: April 20, 2021. [Online]. Available: <https://www.worldometers.info/coronavirus/>
- [10] "A UN framework for the immediate socio-economic response to COVID-19," United Nations Sustain. Develop. Group, New York, NY, USA, Tech. Rep., 2020.
- [11] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty," *Science*, vol. 353, no. 6301, pp. 790–794, 2016.
- [12] M. M. Bennett and L. C. Smith, "Advances in using multitemporal night-time lights satellite imagery to detect, estimate, and monitor socioeconomic dynamics," *Remote Sens. Environ.*, vol. 192, pp. 176–197, 2017.
- [13] F. Liu *et al.*, "Abrupt declines in tropospheric nitrogen dioxide over China after the outbreak of COVID-19," *Sci. Advances*, vol. 6, no. 28, 2020, Art. no. eabc2992.
- [14] M. Weiss, F. Jacob, and G. Duveiller, "Remote sensing for agricultural applications: A meta-review," *Remote Sens. Environ.*, vol. 236, 2020, Art. no. 111402.
- [15] C. Alcantara, T. Kuemmerle, A. V. Prishchepov, and V. C. Radeloff, "Mapping abandoned agriculture with multi-temporal modis satellite data," *Remote Sens. Environ.*, vol. 124, pp. 334–347, 2012.
- [16] E. Hoppe *et al.*, "Transportation infrastructure monitoring using satellite remote sensing," in *Materials and Infrastructures*. Hoboken, NJ, USA: Wiley, 2016, ch. 14, pp. 185–198.
- [17] M. Tello, C. López-Martínez, and J. J. Mallorqui, "Automatic vessel monitoring with single and multidimensional SAR images in the wavelet domain," *ISPRS J. Photogrammetry Remote Sens.*, vol. 61, no. 3/4, pp. 260–278, 2006.
- [18] E. Choi *et al.*, "In-flight transmission of SARS-COV-2," *Emerg. Infectious Disease*, vol. 26, no. 11, pp. 2713–2716, 2020.
- [19] N. C. Khanh *et al.*, "Transmission of SARS-COV 2 during long-haul flight," *Emerg. Infectious Disease*, vol. 26, no. 11, pp. 2617–2624, 2020.
- [20] A. Pavli *et al.*, "In-flight transmission of COVID-19 on flights to Greece: An epidemiological analysis," *Travel Med. Infectious Disease*, vol. 38, 2020, Art. no. 101882.
- [21] D. Lam *et al.*, "xView: Objects in context in overhead imagery," 2018, *arXiv: vol.1802.07856*.
- [22] G.-S. Xia *et al.*, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3974–3983.
- [23] A. Zhao *et al.*, "An effective method based on ACF for aircraft detection in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 744–748, May 2017.
- [24] B. Kamgar-Parsi, A. K. Jain, and J. E. Dayhoff, "Aircraft detection: A case study in using human similarity measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 12, pp. 1404–1414, Dec. 2001.
- [25] F. Zhang, B. Du, L. Zhang, and M. Xu, "Weakly supervised learning based on coupled convolutional neural networks for aircraft detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5553–5563, Sep. 2016.
- [26] D. Grosgeorge, M. Arbelot, A. Goupilleau, T. Ceillier, and R. Allieux, "Concurrent segmentation and object detection CNNs for aircraft detection and identification in satellite images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 276–279.
- [27] "Rapid action on coronavirus and EO—Airports: Throughput," Accessed: Jan. 21, 2021. [Online]. Available: <https://race.esa.int/?indicator=E13b>
- [28] "EUROCONTROL—COVID-19 Dashboard," Accessed: Jan. 21, 2021. [Online]. Available: <https://www.eurocontrol.int/>
- [29] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up Opensky: A large-scale ADS-B sensor network for research," in *Proc. 13th Int. Symp. Inf. Process. Sensor Netw.*, 2014, pp. 83–94.
- [30] Y. Liu *et al.*, "Space eye on flying aircraft: from Sentinel-2 MSI parallax to hybrid computing," *Remote Sens. Environ.*, vol. 246, 2020, Art. no. 111867.
- [31] F. Zhao, L. Xia, A. Kylling, R. Li, H. Shang, and M. Xu, "Detection flying aircraft from Landsat 8 OLI data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 141, pp. 176–184, 2018.
- [32] "Landsat 8 (18) data users handbook," United States Geological Sur., US Dept. Interior, Reston, VA, USA, Tech. Rep., 2019.
- [33] H. Heiselberg, "Aircraft and ship velocity determination in Sentinel-2 multispectral images," *Sensors*, vol. 19, no. 13, 2019, Art. no. 2873.
- [34] "Sentinel user handbook and exploitation tools—Sentinel-2 user handbook," Eur. Space Agency, Paris, France, Tech. Rep., 2015.
- [35] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [36] "Rapid Action on Coronavirus and EO—Airports: Airplanes traffic," Accessed: Jan. 21, 2021. [Online]. Available: <https://race.esa.int/?indicator=E13d>
- [37] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
- [38] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
- [39] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.
- [40] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
- [41] "Sentinel-hub," Accessed: Jan. 01, 2021, [Online]. Available: <https://www.sentinel-hub.com/>
- [42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [43] R. Minetto, M. Pamplona Segundo, and S. Sarkar, "Hydra: An ensemble of convolutional neural networks for geospatial land classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6530–6541, Sep. 2019.

- [44] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, vol. 15, 2011, pp. 315–323.
- [45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Machine Learn.*, 2015, pp. 448–456.
- [46] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit.*, vol. 3, 2006, pp. 850–855.
- [47] N. Menini *et al.*, "Tucumã: A toolbox for spatiotemporal remote sensing image analysis [software and data sets]," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 3, pp. 110–122, Sep. 2019.
- [48] D. P. Roy *et al.*, "Landsat-8 and Sentinel-2 burned area mapping—A combined sensor multi-temporal change detection approach," *Remote Sens. Environ.*, vol. 231, 2019, Art. no. 111254.
- [49] E. L. Bullock, C. E. Woodcock, and C. E. Holden, "Improved change monitoring using an ensemble of time series algorithms," *Remote Sens. Environ.*, vol. 238, 2020, Art. no. 111165.
- [50] A. C. Szakmary, Q. Shen, and S. C. Sharma, "Trend-following trading strategies in commodity futures: A re-examination," *J. Banking Finance*, vol. 34, no. 2, pp. 409–426, 2010.
- [51] N. A. James, A. Kejariwal, and D. S. Matteson, "Leveraging cloud data to mitigate user experience from 'breaking bad'," in *Proc. IEEE Int. Conf. Big Data*, 2016, pp. 3499–3508.
- [52] G. J. Székely and M. L. Rizzo, "Energy statistics: A class of statistics based on distances," *J. Statist. Plan. Inference*, vol. 143, no. 8, pp. 1249–1272, 2013.
- [53] E. H. Van Nes and M. Scheffer, "Slow recovery from perturbations as a generic indicator of a nearby catastrophic shift," *Amer. Naturalist*, vol. 169, no. 6, pp. 738–747, 2007.
- [54] A. J. Veraart, E. J. Faassen, V. Dakos, E. H. van Nes, M. L. ÅErling, and M. Scheffer, "Recovery rates reflect distance to a tipping point in a living system," *Nature*, vol. 481, no. 7381, pp. 357–359, 2012.
- [55] J. Devore, *Probability and Statistics for Engineering and the Sciences*. Pacific Grove, CA, USA: Brooks/Cole, 1987.
- [56] J. Hasell *et al.*, "A cross-country database of COVID-19 testing," *Sci. Data*, vol. 7, no. 1, Oct 2020, Art. no. 345.
- [57] B. Xu *et al.*, "Open access epidemiological data from the COVID-19 outbreak," *Lancet Infectious Diseases*, vol. 20, no. 5, p. 534, 2020.
- [58] "Rapid Action on Coronavirus and EO—Number of Trucks," Accessed: Jan. 21, 2021. [Online]. Available: <https://race.esa.int/?indicator=E12c>



**Maurício Pamplona Segundo** (Member, IEEE) received the B.Sc., M.Sc., and D.Sc. degrees in computer science from the Federal University of Paraná (UFPR), Curitiba, Brazil, in 2007, 2010, and 2013, respectively.

He is a Postdoctoral Researcher with the Institute for Artificial Intelligence (AI+X), University of South Florida Tampa, FL, USA. His research interests include computer vision and pattern recognition, and biometrics, remote sensing, 3D reconstruction, accessibility tools, and vision-based automation.



**Allan Pinto** (Member, IEEE) received the B.Sc. degree from the University of São Paulo (USP), São Paulo, Brazil, in 2011, the M.Sc. and Ph.D. degrees from the University of Campinas (Unicamp), Campinas, Brazil, in 2013 and 2018, respectively, all in computer science.

He is a Postdoctoral Researcher with the University of Campinas (Unicamp), Brazil. A part of his doctoral research was accomplished with the University of Notre Dame, Notre Dame, IN, USA, in which he worked on different topics such as presentation attack

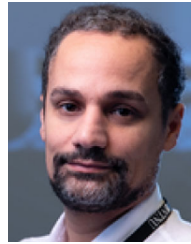
detection in biometric systems, content-based image retrieval, and multimedia forensics. His research interests include image and video analysis, pattern recognition, machine learning, computer vision, multimedia forensics, biometrics, content-based image retrieval, remote sensing, and sports science.

Dr. Allan is a Member of the editorial board of the *Forensic Science International: Reports*, and also a Member of IEEE Information Forensics and Security Technical Committee (IFS-TC) of the IEEE Signal Processing Society.



**Rodrigo Minetto** (Member, IEEE) received the Ph.D. degree in computer science from University of Campinas (UNICAMP), Campinas, Brazil and Sorbonne Université, France, in 2012.

He is an Associate Professor with Federal University of Technology—Paraná (UTFPR), Curitiba, Brazil. He was a Visiting Scholar with University of South Florida (USF), Tampa, FL, USA, in 2017. His research interests include image processing, computer vision, and machine learning.



**Ricardo da Silva Torres** (Member, IEEE) received the B.Sc. degree in computer engineering and the Ph.D. degree in computer science from the University of Campinas (Unicamp), Campinas, Brazil, in 2000 and 2004, respectively.

He was a Professor with the Unicamp, from 2005 to 2019. He is a Professor of Visual Computing with the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He has been developing multidisciplinary eScience research projects involving multimedia analysis, multimedia retrieval,

machine learning, databases, information visualization, and digital libraries. He has authored or coauthored more than 200 articles in refereed journals and conferences.

Dr. Torres serves as a PC Member of several international and national conferences. He has been serving as a Senior Associate Editor for the IEEE SIGNAL PROCESSING LETTERS and an Associate Editor for the *Pattern Recognition Letters*.



**Sudeep Sarkar** (Fellow, IEEE) received the M.S. and Ph.D. degrees in electrical engineering, on a University Presidential Fellowship, from The Ohio State University, Columbus, OH, USA, in 1990 and 1993.

He is a Professor of Computer Science and Engineering and Associate Vice President of Research & Innovation with the University of South Florida in Tampa, FL, USA. He has 25 years of a expertise in computer vision and pattern recognition algorithms and systems, and holds three U.S. patents and has

authored or coauthored high-impact journal and conference papers.

Dr. Sarkar is a Fellow of the American Association for the Advancement of Science (AAAS), Institute of Electrical and Electronics Engineers (IEEE), American Institute for Medical and Biological Engineering (AIMBE), and International Association for Pattern Recognition (IAPR); and a Charter Member and Member of the Board of Directors of the National Academy of Inventors (NAI). He was the recipient of the National Science Foundation CAREER award in 1994, the USF Teaching Incentive Program Award for Undergraduate Teaching Excellence in 1997, the Outstanding Undergraduate Teaching Award in 1998, and the Theodore and Venette Askounes—Ashford Distinguished Scholar Award in 2004.