Detecting Moving Vehicles on Sentinel-2 Imagery Using Semi-Automatic Labeling From S2A/S2C Tandem Phase

Guillaume Buthmann[1], Florentin Poucin[1], Jérémy Anger[1, 2]

[1] Kayrros SAS, 75009 Paris, France
[2] Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, 91190, Gif-sur-Yvette, France

Keywords: Vehicle detection, Sentinel-2 tandem, automatic annotations

Abstract

During the commissioning phase of ESA’s Sentinel-2C, tandem images with Sentinel-2A were acquired with a delay of 30 seconds. We present a novel, automated method for labeling moving vehicles in Sentinel-2 images, leveraging the temporal offset between these tandem acquisitions. We propose a filtering process that isolates pixels corresponding to vehicles that moved between the two acquisitions. We generate a training dataset based on this process, removing the need for a large manual labeling phase. The dataset is used to train a standard deep-learning-based vehicle detection model. Experimental results, as well as a validation study using ground-truth data from California, highlight the quality of the proposed labeling method, and show that a vehicle detection model can be successfully trained from quasi-simultaneous acquisitions.

1. Introduction

Road traffic is a significant contributor to CO_2 emissions globally. While the oil demand related to road transport remained stable between 2023 and 2024 (IEA, 2025), the need for accurate traffic measurement is crucial. Indeed, having a global vision of the traffic trends in all countries can help evaluate the impact of regulatory policies or analyze regional traffic trends. Traffic monitoring is typically performed thanks to vehicle counting stations, placed on strategic roads. Relevant traffic indices can be extrapolated from the vehicle counts (FHWA, 2022), which can be further linked to CO_2 emissions. Each country may have different strategies regarding the implementation of such systems, or some countries may not have a monitoring solution. In addition, the monitoring stations can be associated with important maintenance costs and often provide only a sparse cover of the country where they are implemented.

In order to reach a more global and unified coverage of traffic trends, we propose to work with publicly available satellite images. In this study, we focus on ESA’s Sentinel-2 constellation, which provides optical images covering most of the Earth’s mainland at 10 meters per pixel. Despite their relatively low resolution, Sentinel-2 images can reveal the presence of vehicles thanks to an optical effect induced by the MultiSpectral Instrument (MSI) used by the Sentinel-2 satellites (Drusch et al., 2012). Since each spectral band is captured with a slight temporal offset, including the red, green and blue bands, a moving object will be visible at different positions for the different bands. This phenomenon results in a ”rainbow” effect, visible in Fig. 1. This rainbow effect has been extensively studied and can be used to perform vehicle detection, given that the moving vehicles exceed a speed of around 50 km/h, allowing them to span multiple pixels (Fisser et al., 2022). Thus, by focusing on high-speed motorways, it is possible to identify vehicle pixels and count them effectively.

Sentinel-2C (S2C) was launched in September 2024 and entered nominal operations in January 2025. During the commissioning period, a tandem phase with Sentinel-2A (S2A) was performed for around two months. In particular, from the 9th to the 19th of December 2024, both satellites followed a full scenario acquisition plan, S2C acquiring the same scenes as S2A with a 30-second delay. This tandem phase was designed to perform radiometric calibration and validation of S2C with the existing constellation. The same procedure was used to calibrate the Sentinel-3 A/B satellites (Lamquin et al., 2020). These preliminary S2C products were disseminated to the community under the 99.05 and 99.06 processing baselines, indicating the uncalibrated nature of this data. They are not part of the Sentinel-2 Collection 1.[1]

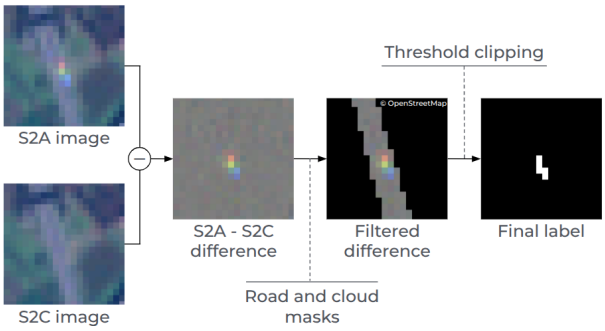


Figure 1. Summary of the proposed dataset creation method. The vehicles, visible thanks to the rainbow effect, are identified using a difference between tandem S2A and S2C images. The difference is filtered using road masks and region-optimized thresholds, as presented in subsection 3.3.

Besides the radiometric and geometric discrepancies between S2A and uncalibrated S2C imagery, the tandem images reveal moving parts of the scenes, such as natural phenomenons (clouds, waves) and human-made objects (boats, planes, vehicles). In this study, we propose a method that leverages the S2A/S2C tandem imagery to develop a moving vehicle detector operating from a single image. A summary of the proposed dataset creation method is shown in Fig. 1.

[1] At the time of writing, the reprocessing of the S2C imagery of December 2024 using a nominal, calibrated, baseline is planned, which will simplify the use of this tandem data.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-3-2026-43-2026

2. Related works

Due to the large potential of applications related to vehicle detection on satellite images, a notable research effort can be observed in this field. One of the first works focusing on vehicle detection on Sentinel-2 images leverages band filters to identify the blue pixels of the vehicles (Fisser, 2020). The method uses preprocessing filters, based on various spectral bands and band ratios, to build road masks and filter irrelevant areas. Then, the method identifies blue pixels inside rainbow trails by computing two different ratios involving the blue, green, and red channels of the image and keeping pixels satisfying empirical thresholds. Ultimately, groups presenting 1 to 3 blue pixels are identified as trucks by the method. Other work focused on demonstrating the applicability of spectral offset analysis for velocity determination in aircraft and ships (Heiselberg, 2019). Based on band thresholds, this method also provides valuable insights into the underlying physics of moving object detection across different vehicle types.

These methods present the interest of not requiring any labeled dataset, making them easy to use. However, they are limited as they use deterministic thresholds: vehicles that diverge from the predefined heuristic will hardly be detected. In order to obtain a global vehicle detection method, machine learning models have emerged as relevant candidates, thanks to their improved robustness. Building on top of the previous vehicle detection method for Sentinel-2 images (Fisser, 2020), a detection method featuring a random forest algorithm was developed (Fisser et al., 2022). The random forest algorithm is used to classify each pixel into one of 4 classes: background, or one of the 3 target classes of a vehicle (blue, green, and red pixels). For every blue pixel identified, the neighboring pixels are scanned to identify a green and a red pixel in order, which characterizes a vehicle. The creation of the training dataset required the manual labeling of around 3,000 bounding boxes.

The development of convolutional neural networks (CNNs) has also enabled progress in the field of small vehicle detection, along with many other fields of computer vision. Thanks to their versatility, CNNs have also been applied to vehicle detection on Sentinel-2 images (Blattner et al., 2021). The method uses a two-stage approach combining Faster R-CNN with a ResNet-50 backbone. The model was trained on a manually labeled dataset containing 4,686 bounding boxes. The results showed promising results when applied to Sentinel-2's 10-meter resolution data, showing that the model was able to specifically target the characteristic rainbow effect produced by the moving vehicles.

While vehicle detection in high-resolution imagery is a widely explored topic (Kaack et al., 2019, Zhou et al., 2020), the detection on low-resolution images is still being actively investigated. Previous methods based on machine learning and deep learning models show promising results, but extensive manual labeling is often required, and improving the performance with more complex models will require larger datasets. Due to the complex nature of the labeling task and the induced labeling time, we propose a novel method to generate semi-automatic vehicle labels on 10-meter resolution Sentinel-2 imagery.

3. Label generation

The tandem phase of S2A and S2C provides a unique opportunity to automatically obtain labels of moving vehicles. Indeed, considering two images of the same location taken with a 30-second delay, the main noticeable changes are due to moving objects such as vehicles or clouds. By computing a pixel-to-pixel difference between the S2A and the S2C image, moving objects result in a high difference, as illustrated in Fig. 2. In the following, we propose a filtering method that enables vehicle detection and minimizes false positive detections.

3.1 Tandem imagery preparation

In this section, we describe how the imagery was retrieved and processed in order to minimize false positives when computing the differences between tandem images.

Product pair identification. The first step of the dataset creation is to identify the pairs of S2A/S2C products. Listing every product for each satellite platform between 2024-12-10 and 2024-12-19 included, around 42793 product IDs and their metadata are retrieved. Then, using the day of acquisition and MGRS tile ID, unique pairs can be identified. For this study, only the B02, B03 and B04 bands were used, which correspond to the blue, green and red channels of the image, using the L2A Bottom-of-Atmosphere processing level. The Copernicus Data Space Ecosystem (CDSE) was used to download the bands and metadata of interest, using the S3 protocol.

S2C calibration. We then perform spatial and radiometric adjustments to ensure that the S2A and S2C images are as similar as possible. Indeed, as Sentinel-2C was undergoing a calibration phase, the S2C products were generated using a processing baseline that did not include fine calibration of the instrument, which can result in unwanted differences. To minimize the differences with the S2A images, we applied two corrections: spatial registration and radiometric adjustments. Because each detector of the MSI instrument requires a specific calibration, we consider 600×600 crops that are extracted randomly, while ensuring that they originate from a single detector for both S2A and S2C, using the detector footprint assets.

When considering a small crop, geometric inaccuracies can be modeled by a translation. Such translation can be robustly estimated using the phase correlation algorithm (Kuglin and Hines, 1975). We perform the estimation using the B04 band. The correlation score of the main and second peak is used to determine whether the estimation is accurate, and a sinc sub-pixel refinement is used (Foroosh et al., 2002, Hessel et al., 2021). Once the shift is estimated, the S2C imagery is resampled using a third-order spline interpolation.

Radiometric calibration is performed using a simple but robust approach. The median of each band of the S2A and S2C rasters is computed. Let M_b^A and M_b^C be the median for the band b of S2A and S2C respectively. Then, the band b of S2C is multiplied by a linear factor M_b^A / M_b^C.

Cloud filtering. In order to further prevent false positive detections, we filter cloudy areas. Indeed, clouds and cloud shadows can induce strong differences between two tandem images. To perform cloud and shadow filtering, we use the deep-learning-based cloud detector provided by the CloudSEN12 project (Aybar et al., 2022). For the following steps of the workflow, we mask any pixel belonging to the cloud or cloud shadow class, from both the S2A and S2C images.

(a) S2A image (b) S2C image (30 seconds later) (c) Difference between S2A and S2C

Figure 2. An example of difference: two vehicles are visible in the S2A image, but not in the S2C image, since they moved. The difference highlights the vehicles that moved between the two acquisitions.

Road masking. Finally, we apply road masks to the images and ignore any difference outside of these masks, as we are interested in detecting moving vehicles on roads only. We use OpenStreetMap to obtain the road masks associated with the images. This masking process focuses on motorways and trunk highways, which are the two largest road types according to the OpenStreetMap classification. Due to the low resolution of Sentinel-2 images, detecting vehicles on smaller roads hardly seems achievable and results in false positive detections. Additionally, we perform a 1-pixel dilation on the road mask to ensure that we do not miss vehicles due to an imprecise or overly strict road mask.

3.2 Channel difference process

After framing the problem using spatial and radiometric registration, as well as cloud and road masks, we focus on the method that enables vehicle detection. Let us consider a pair of images of size 600×600 pixels, taken from a tandem pair of S2A and S2C tiles, for which we performed spatial and radiometric adjustments beforehand. We present the labeling technique used to identify vehicles on the S2A image only, but the process can be reversed to label the vehicles on the S2C image.

Due to the rainbow aspect of vehicles, we process the red, green and blue channels separately, which allows us to use tailored difference thresholds that focus on vehicle pixels. Using a separate difference process for each channel also prevents differences on separate channels from averaging out, which would result in more false negatives. In the following, we detail the difference process for one of the channels only, the final labeling mask being obtainable by computing the union of the three resulting masks. The automatic labeling can be decomposed into three steps. First, we subtract the S2C image from the S2A image. Then, we compute the road and cloud masks, and ignore any difference outside the road mask or inside the cloud mask. Finally, we label as vehicles the pixels with a difference higher than a certain threshold. The computation of the threshold is detailed in subsection 3.3.

As this process alone is not sufficient to detect all vehicles from the S2A image, we propose a disambiguation method to make the difference between road and vehicle pixels. Indeed, by computing the difference between the S2A and S2C images and keeping pixels with a difference greater than a certain threshold, we make the assumption that vehicle pixels are brighter than road pixels. However, visible vehicles present strong variations in brightness and color scheme, as illustrated in Fig. 3. For example, darker vehicles would result in false positive and false negative detections using the previous method. Thus, we propose to improve the method by computing both S2A-S2C and S2C-S2A differences and, for each difference flagged as a potential S2A vehicle, determining which image actually contains the vehicle. This process allows us to keep only the actual S2A vehicles and label them accurately.

We propose the following method to distinguish between vehicle and road pixels. For an individual pixel (for which none of the surrounding pixels are flagged as potential vehicles), we compute the standard deviation across channels of the pixel. Out of the S2A and the S2C corresponding pixels, the pixel with the highest standard deviation is classified as the vehicle, while the other one likely corresponds to an empty road pixel. This method is based on the consideration that road pixels usually appear gray, resulting in a lower standard deviation when compared to vehicle pixels, which usually have a channel presenting a significantly different value. For groups of pixels, we instead consider the intra-channel standard deviation to avoid biases in cases where the road does not appear gray. With $\left((r_i^A, g_i^A, b_i^A)\right)_{i \in [\![1,N]\!]}$ the values of the N pixels flagged as a potential vehicle in the S2A image, we compute the standard deviation per channel as $\sigma^A = \left(\sigma_r^A, \sigma_g^A, \sigma_b^A\right)$. We finally compute the deviation $d^A = \sigma_r^A + \sigma_g^A + \sigma_b^A$. By repeating the same computations on S2C to obtain d^C, we determine that the image with the highest deviation for the selected pixels is the one featuring the vehicle. Using this method, the road will have a low deviation even if it does not appear gray, because road pixels within the detected patch are similar to each other. In comparison, the actual vehicle will present strong color variations within the patch, resulting in a higher deviation.

3.3 Threshold optimization

In order to obtain a robust vehicle detection model, we intend to include several regions of the world in the training dataset, as the intensity range and distribution of the pixels may strongly vary from one region to another. To this effect, we propose a method to compute three optimal difference thresholds (one per channel) for each region. To account for varying atmospheric conditions between datatakes, we optimize the threshold for each datatake independently.

For a given datatake that contains tandem S2A and S2C images, we select N pairs of images of size 600×600 suitable for vehicle detection. We select a small subset N_A (usually around 10) among these pairs, which we use to optimize the thresholds. For the purpose of designing a scalable automatic labeling process, and as manually labeled images can be used as ground-truth data for evaluating the performance of the computer vision model, we perform a phase of manual labeling on the selected subset of N_A images. For each pair of images, we

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-3-2026-43-2026

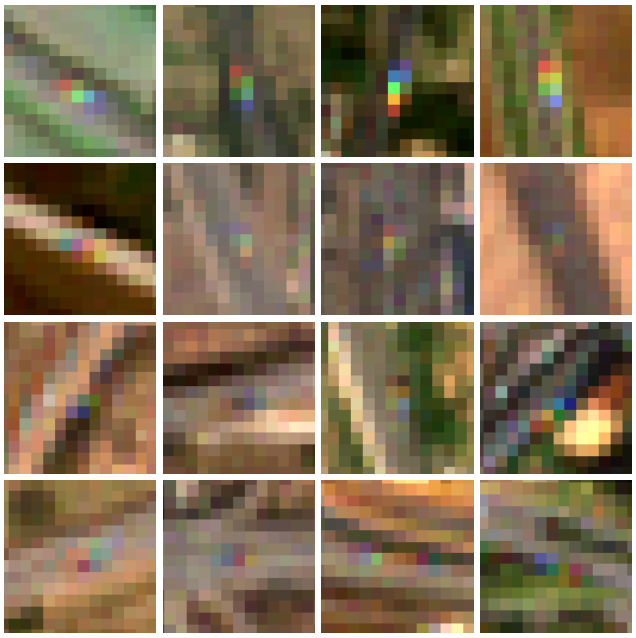


Figure 3. Examples of visible vehicles, varying in color and brightness. Vehicles may appear brighter or darker than the road.

manually label the S2A image, viewing both S2A and S2C images to ease the labeling process. Note that N_A is deliberately kept small and that the manually labeled images will not be part of the training or validation datasets.

For each channel, we compute the $(D_k)_{k\in[\![1,N_A]\!]}$ masks obtained from the process described in subsection 3.2. We then perform the difference clipping for a range of thresholds corresponding to various percentiles of the differences. For example, we discard the pixels with a difference below the top 1% of differences recorded for the considered channel. Thus, for each threshold, we obtain N_A automatically labeled images that can be compared to their manually labeled counterparts. We pick the threshold that maximizes an F1-score weighted towards precision between the manually and automatically labeled images. More precisely, for a precision P and a recall R, the weighted F1-score is:

$$F_1^w = \frac{P \times R}{\frac{1}{3}P + \frac{2}{3}R}$$

This metric helps reducing false positives, which are more critical in this application: it is preferable to miss a few vehicles rather than having numerous aliased pixels be labeled as vehicles. We illustrate in Fig. 4 the use of F_1^w. While the regular F1-score would have led to choosing 98.9 as the difference percentile used to compute the threshold for this datatake, we rely on F_1^w and choose 99.1 instead.

We select the triplet of thresholds, one per band, that maximizes the weighted F1-score and use these thresholds in the automatic labeling process for the rest of the datatake, namely the $N_B = N - N_A$ pairs of images that were set aside at the beginning of the process. In Fig. 5, we present tandem S2A/S2C images, along with the manually labeled ground truth. On the second row, we illustrate the effect of various thresholds on the automatic labeling.

3.4 Final dataset

Using the automatic labeling process detailed in the previous subsections, we create a training and validation dataset on which

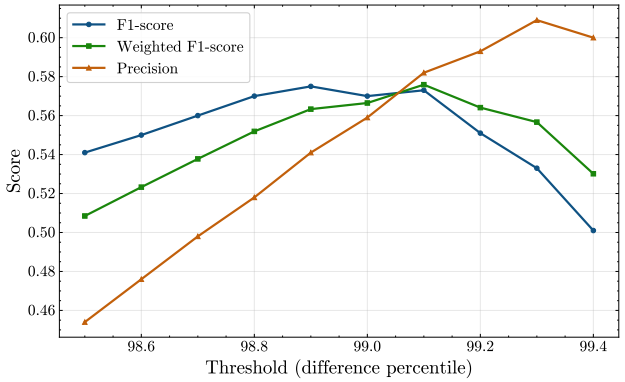


Figure 4. Evolution of F_1 and F_1^w, along with the precision, depending on the threshold selected, for a datatake in Italy.

a vehicle detection model can be trained. To this effect, we downloaded the RGB bands of 61 L2A-processed S2A and S2C tiles with original size $10,980 \times 10,980$ spread across various regions of the world, including Africa, Asia, North and South America, and Europe. The tiles were captured between the 9th and the 19th of December 2024, which is the period when both satellites followed the same acquisition plan with a 30-second delay.

The tiles were divided into 2,322 smaller 600×600 tiles randomly sampled while meeting specific criteria. The tiles were selected based on the detector used as presented in 3.1, but also based on remote sensing criteria relevant to vehicle detection: tiles containing too much water are not suitable for the task and were removed. This selection was performed by computing the NDWI (Normalized Difference Water Index) of the tiles and discarding a tile if its median NDWI is above 0.3. The NDWI can be computed as

$$\text{NDWI} = \frac{B_{03} - B_{08}}{B_{03} + B_{08}}$$

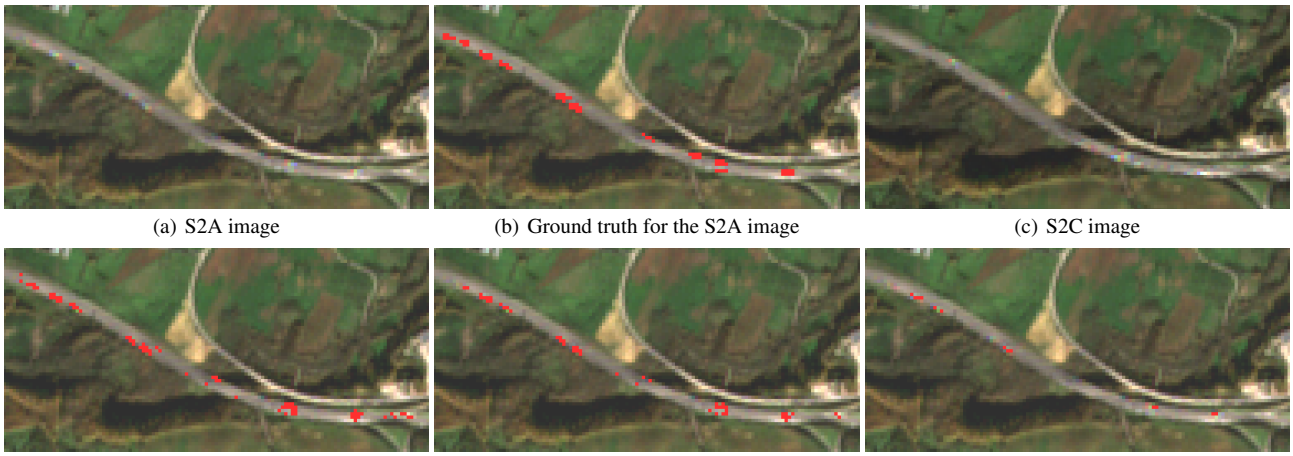
where B_{03} and B_{08} are respectively the green and near infra-red bands of a Sentinel-2 image. Finally, we applied road masks to the 2,322 tiles and kept the tiles containing a motorway or trunk road, according to the OpenStreetMap classification. After performing the automatic labeling on this smaller subset, we obtained 522 labeled images of size 600×600.

For the purpose of computing efficiency during training, the 600×600 tiles were cut into 36 100×100 tiles, and only tiles containing labeled vehicles were kept, removing many patches without roads. The final dataset used contains 2,703 images of size 100×100 with 26,371 positive pixels, corresponding to 10,186 objects. The dataset has a strong class imbalance, with vehicle pixels representing only 2% of the road pixels and 0.1% of the total pixels.

4. Detection Model

To evaluate the performance of the automatic labeling method, we train a vision model to detect moving vehicles, with a dataset generated using the method described in the previous section. The model takes a single 3-band L2A image as input and is supervised using the labels derived from the tandem imagery.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-3-2026-43-2026 | © Author(s) 2026. CC BY 4.0 License.



(a) S2A image (b) Ground truth for the S2A image (c) S2C image

(d) Automatic labeling (98.5-th percentile) (e) Automatic labeling (99.1-th percentile) (f) Automatic labeling (99.7-th percentile)

Figure 5. Effect of different thresholds. The displayed image is used to optimize the thresholds but is not part of the training dataset.

4.1 Architecture and training

Inspired by the UNet architecture (Ronneberger et al., 2015), which is widely used in computer vision, we create a smaller model that will be trained from scratch. The model features an encoder and decoder block, each composed of two blocks of layers (2D convolution, batch normalization, and ReLU activation). We also implement a skip connection between the encoder's first layer and the decoder's last layer. More precisely, we concatenate the output of the encoder's first layer to the output of the decoder's first layer, before using this concatenated vector as input data for the decoder's last layer. Since the vehicles span a few pixels, we design the model towards spatial resolution preservation. The skip connection helps preserve high spatial resolution, and the model only has two stages of convolution, so as not to reach tensors with too many channels and too low spatial dimensions. The final model has 14,513 parameters.

The last layer of the model outputs a feature map with higher values for pixels that are likely to be vehicles. During post-processing, this feature map is converted into a probability map using a sigmoid function, where the value of each pixel approximates the probability that it represents a vehicle. Thus, it is possible to choose a confidence threshold to perform the binary classification: for instance, pixels with more than 0.3 confidence could be classified as vehicles. This threshold is optimized when evaluating the model by picking the confidence threshold that maximizes the F1-score on the manually labeled test set.

The model is trained on 80% of the dataset presented in Section 3.4 and validated using the remaining 20%. The training is performed for 250 epochs, using a standard learning rate of 0.001 along with the RAdam optimizer. Due to the imbalanced nature of the task, we train the model using a focal loss (Lin et al., 2017). With p_t being the distance between the model's prediction p and the actual label y, the focal loss can be expressed as

$$\mathrm{FL}(p_t) = -\alpha(1-p_t)^{\gamma}\log(p_t)$$

The focusing term, influenced by γ, prevents easy examples (such as empty road pixels) from weighing too much on the loss, allowing the model to focus on more challenging examples. We use $\alpha = 0.25$ and $\gamma = 2$. To improve the robustness of the model and the diversity of the dataset, we use data augmentation techniques. To simulate variations of the atmospheric conditions, we implement random changes in contrast and brightness within a 50% range of the original values. We also add a 50% chance of flipping the images.

4.2 Performances

We run the trained model on the manually labeled test dataset to evaluate its performance accurately. The predictions and F1-score computations are restricted to pixels within road masks extracted from OpenStreetMap only. Using an optimized confidence threshold of 0.26 to classify the vehicles, we record a F1-score of 0.53 for the vehicle detection model. We present in Fig. 6 examples of predictions from the model, compared with the ground-truth labels, on crops taken from the manually labeled test dataset. Despite being trained on an automatically labeled dataset, the model was able to learn relevant features. It is able to detect vehicles in diverse settings and regions of the world. In the given examples, the model successfully detects vehicles even though the segmentation is not perfect: the predicted pixels are not perfectly aligned with the ground-truth pixels. While this is not an issue in most cases, since we consider a group of adjacent pixels to be a single vehicle, it can affect vehicle counts in the case of aliased areas. As illustrated in Fig. 7, the model confuses aliased pixels and vehicles due to the peculiar aspect of the road. Such individual detections of aliased pixels can result in imprecise vehicle counts. A continuous flow of vehicles may also be detected as one vehicle, but counting individual pixels does not improve the performance significantly.

To evaluate the importance of the size of the dataset, we train the same model on a fraction (around 50%) of the manually labeled dataset. Using 312 random manually labeled 100×100 images as a training dataset and the remaining manually labeled images as a test dataset (62 images of size 600×600), the model achieves an F1-score of 0.45. In comparison, the model trained on the automatically labeled dataset (containing 2703 images) achieves an F1-score of 0.54 on this smaller testing dataset. The evaluation thresholds of both models were optimized to achieve the best performance. This experiment shows that the larger amount of labeled images obtainable using the automatic labeling method can counterbalance the slight decrease in quality, compared with manually labeled images. The model trained on the automatically labeled data achieves higher performance, highlighting the potential of our method to generate large-scale datasets.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-3-2026-43-2026

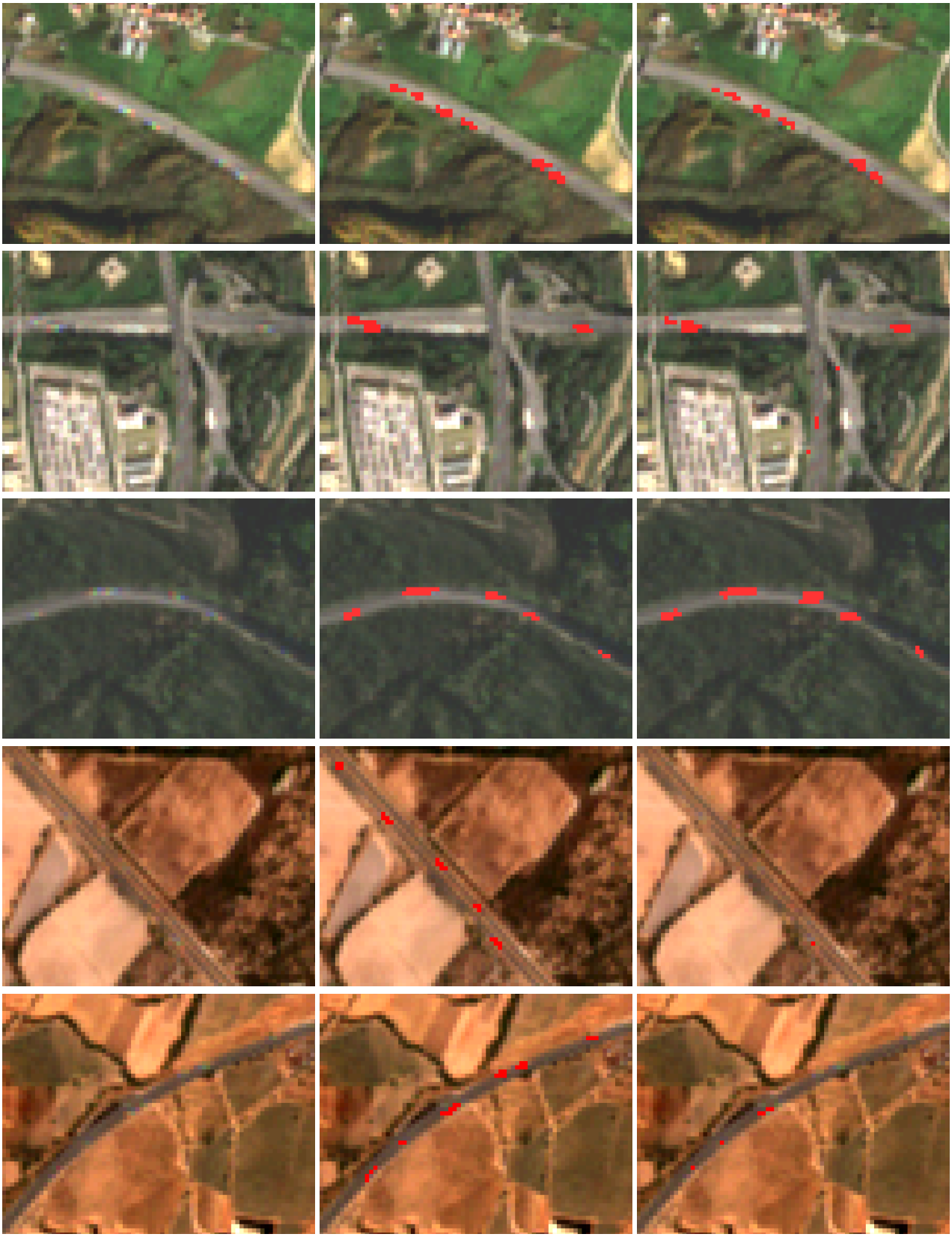


Figure 6. Examples of predictions from our model trained on the proposed dataset. From left to right: S2A input from the test dataset, manually labeled ground-truth masks, model predictions.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-3-2026-43-2026

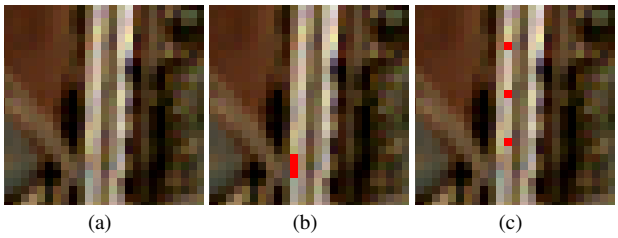


Figure 7. (a) Crop containing aliased pixels from the test dataset (b) Manually labeled ground-truth mask (c) Model prediction.

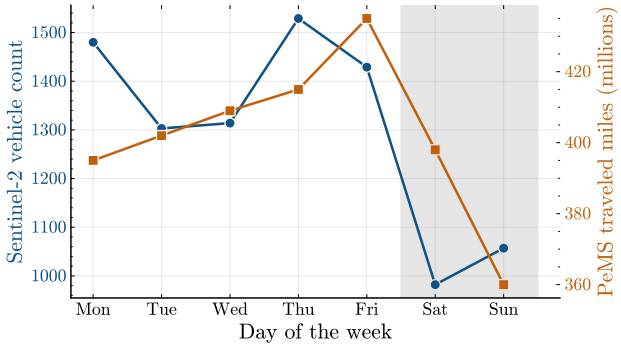


Figure 8. Comparison between the average number of vehicles counted in California per Sentinel-2 tile per day of the week and the average number of miles traveled per day of the week as reported by PeMS, between September 2023 and August 2024.

4.3 Validation using Caltrans data

We propose a study of traffic data in California to investigate further the quality of the predictions of the model and the relevance of the automatic labeling technique for real-world applications. The California Department of Transportation (Caltrans) offers various services related to traffic flow in California, including PeMS (Performance Measurement System), a website delivering data from more than 40,000 vehicle counting stations across the California freeway system. The data is publicly available, the sensors are placed on most of the roads that the model can analyze (motorways and trunk highways), and most sensors offer a 5-minute precision when counting vehicles. Since historical data is available for both Sentinel-2 images and PeMS data, we consider a timespan of one year, from September 2023 to August 2024.

Since PeMS covers the whole state of California, we select any Sentinel-2 tile that contains roads in California. Using OpenStreetMap, we apply road masks to the tiles and run the vehicle detection model on the images. For each tile, the results of the detection model are then converted into vehicle counts, by considering adjacent pixels as part of one vehicle. In Fig. 8, we present a comparison between the average vehicle count per Sentinel-2 tile per day of the week and the average miles traveled per day of the week as reported by PeMS during the same period. The objective of this aggregated comparison is to evaluate the ability of the model to bring out traffic trends visible in the ground-truth data. We observe that the two curves follow similar trends, with stable values on weekdays and a noticeable drop during the weekend. Observing the different units for each of the curves, we note that the decrease ratios are not immediately correlated, which is explained by the different nature of the measures considered. For example, there may be three times fewer vehicles on the roads on a Saturday, but these

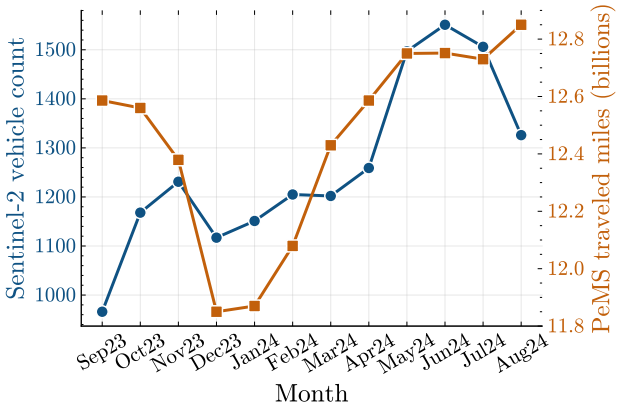


Figure 9. Comparison between the average number of vehicles counted in California per Sentinel-2 tile per month and the average number of miles traveled per month as reported by PeMS, between September 2023 and August 2024.

vehicles may travel two times more on average than on a regular weekday. Furthermore, due to the resolution of Sentinel-2, smaller vehicles are probably not detected by our approach, as noted in previous work (Fisser et al., 2022).

We present the same analysis with monthly averages in Fig. 9. The traveled miles reported by PeMS have been normalized to account for the different number of days in each month. We observe seasonal trends in both curves, with less activity during the winter months and a traffic peak during the summer. We note that September 2023 has an abnormally low Sentinel-2 vehicle count, which is due to an under-representation of dense tiles (such as Los Angeles) during this month, as clouds were obstructing the urban areas of these tiles. An analysis spanning several years would be necessary to counter such effects.

The analysis of California traffic reveals that a lightweight model trained on automatically labeled images achieves promising performance. The aggregated Sentinel-2 vehicle counts exhibit specific seasonal and weekly trends, showing that the model efficiently learned on the vehicle detection task.

5. Conclusion

We presented a novel method to automatically label moving vehicles in Sentinel-2 satellite images, leveraging the commissioning phase of Sentinel-2C, namely the tandem phase with S2A. While the tandem phase was designed to enable cross-calibration between the two satellites, we show that, besides calibration activities, these acquisitions are useful for downstream applications. Automatic labeling is performed by computing differences between S2A and S2C imagery with a 30-second delay, and applying a filtering that exploits the ”rainbow” effect of moving objects. To evaluate the quality of the labels, we used the automatically labeled images to train a simple semantic segmentation model from scratch. We highlight the quality of the labeling method by showing that the vehicle detection model achieves satisfying performance, by validating its results against ground-truth data from vehicle counting stations in California. This demonstrates the method’s potential for large-scale automated traffic monitoring, both to help train and validate the performance of vehicle detection models.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-3-2026-43-2026 | © Author(s) 2026. CC BY 4.0 License.

Acknowledgments

This project is based on a preliminary study by Maëlle Fontaine and Chloé Habasque. We thank ESA for making the tandem data available to the research community. All satellite images presented in this paper are courtesy of Copernicus Sentinel-2 data, 2024. PeMS data: Copyright © 2025 State of California.

References

Aybar, C., Ysuhuaylas, L., Loja, J., Gonzales, K., Herrera, F., Bautista, L., Yali, R., Flores, A., Diaz, L., Cuenca, N. et al., 2022. CloudSEN12, a global dataset for semantic understanding of cloud and cloud shadow in Sentinel-2. *Scientific data*, 9(1), 782.

Blattner, M., Mommert, M., Borth, D., 2021. Commercial vehicle traffic detection from satellite imagery with deep learning. ICML 2021 Workshop on Tackling Climate Change with Machine Learning Workshop.

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P. et al., 2012. Sentinel-2: ESA's optical high-resolution mission for GMES operational services. *Remote sensing of Environment*, 120, 25–36.

FHWA, 2022. Traffic monitoring guide [2022]. `https://rosap.ntl.bts.gov/view/dot/74643`.

Fisser, H., 2020. Truck detection with Sentinel-2 during COVID-19 crisis. `https://github.com/hfisser/Truck_Detection_Sentinel2_COVID19`.

Fisser, H., Khorsandi, E., Wegmann, M., Baier, F., 2022. Detecting moving trucks on roads using Sentinel-2 data. *Remote Sensing*, 14(7), 1595.

Foroosh, H., Zerubia, J., Berthod, M., 2002. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing*, 11(3), 188-200.

Heiselberg, H., 2019. Aircraft and ship velocity determination in Sentinel-2 multispectral images. *Sensors*, 19(13), 2873.

Hessel, C., De Franchis, C., Facciolo, G., Morel, J.-M., 2021. A global registration method for satellite image series. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, IEEE, 3121–3124.

IEA, 2025. Global energy review 2025. `https://www.iea.org/reports/global-energy-review-2025`. Licence: CC BY 4.0.

Kaack, L. H., Chen, G. H., Morgan, M. G., 2019. Truck traffic monitoring with satellite images. *Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies*, 155–164.

Kuglin, C. D., Hines, D. C., 1975. The phase correlation image alignment method. *Proceedings of the IEEE International Conference on Cybernetics and Society*, New York, NY, USA, 163–165.

Lamquin, N., Clerc, S., Bourg, L., Donlon, C., 2020. OLCI A/B tandem phase analysis, part 1: Level 1 homogenisation and harmonisation. *Remote Sensing*, 12(11), 1804.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.

Zhou, L., Liu, J., Chen, L., 2020. Vehicle detection based on remote sensing image of Yolov3. *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 1, IEEE, 468–472.