AN EMPIRICAL STUDY OF APPLYING AI ON THE EDGE FOR EARTH OBSERVATION

Gabriel García[1], Sergio Alonso[1], Katalin Takáts[1], Josep Maria Gonfaus[1], Pau Gallés[1], Jaume Gibert[1], David Vilaseca[1], Gerardo Richarte[*1], and Javier Marín[1]

[1]***Satellogic***
Carrer d'Bailén 3, 1er piso, Barcelona 08010, Spain
[*]*gera@satellogic.com*

1 INTRODUCTION

Artificial Intelligence (AI) has still largely untapped potential for Earth Observation (EO) technology. State-of-the-art (SoA) deep-learning models rely on the use of powerful devices meant to be run on the ground. Latest edge computing devices can run deep-learning models that yield SoA results [1]. EO could gain significant benefits from increasing satellite operability, improving capabilities and performance, and reducing latencies and costs by utilizing edge computing right where the data is generated. Especially, as more and more data is captured on a daily basis, with the current and the upcoming generation of satellite constellations, there is a clear need to reduce the overall overhead of sending the data to the ground for later processing. Drastic reduction of latencies would allow near real-time insights, which in return permits taking actions autonomously on the fly as well as setting up early alerts (e.g. for disaster management). Hence, bringing more computational capabilities to where the data/images are captured/generated is of paramount importance. Following this direction, companies such as Xailient, Palantir [2], SpiralBlue or OrbitsEdge are already providing their services by running their own technology on board of satellites.

One clear example where edge computing can provide significant gains is that of detecting objects on images. The classical workflow involves first transferring the images from the satellite to computers on the ground and only then processing the raw data and doing the model inference. However, on board the satellite, a model trained using raw or low-level product imagery can directly run inference on the edge device. In this scenario, the amount of data transferred is minimal after obtaining the present objects and their locations.

In this work we evaluate the benefits of using an image product processed on board with respect to an on-ground solution. The experiments are carried out using IQUAFLOW [3] (https://github.com/satellogic/iquaflow), our open-source framework originally created to provide a set of tools to assess image quality by means of conducting learning-based tasks, such as detection, segmentation, or classification. A common example consists of evaluating a deep-learning model performance using the same data compressed with different algorithms and compare it with the original data. This type of experiment may permit to adequately choose the best trade-off when compressing images onboard. In this scenario, and other possible ones, IQUAFLOW is intended to provide the guarantees and the environmental setup to test how each of the variables can affect the overall performance while being fair and rigorously judged with the same experimental setup.

For the described goal, we conduct a segmentation task. In particular, we focus on segmenting building footprints in very high-resolution images. We have assembled a novel dataset, which consists of two different data products available at Satellogic: L3 and RapidResponse. The latter is a result of an efficient and optimized processing pipeline that we run on board our satellites. The main goal of our experiments is to evaluate the performance of a deep-learning model that is able to run inference on board our satellites, using the RapidResponse images, and see how much its performance differs from using L3 images on the ground. Results turn in favor of the edge computing platform, unveiling the potential of applying AI on board.

2 DATASET

2.1 Satellite imagery

Satellite images from Satellogic's constellation of low orbit satellites are used in this study. Founded in 2010, Satellogic specializes in Earth observation data collection and analytical imagery solutions. Satellogic designs, builds and operates its own fleet of Earth observation satellites to frequently collect affordable high-resolution imagery for decision-making in a broad range of industrial, environmental and government applications. The Satellogic satellite constellation consists of individual small satellites, named NewSats. Each of the NewSat satellites has a multispectral and a hyperspectral sensor.

The multispectral camera has four bands, covering a wavelength range from 450 nm to 900 nm (blue: 450–510 nm, green: 510–580 nm, red: 590–690 nm and near-infrared: 750–900 nm). Pixel resolution (GSD) at nadir is 0.99 m (native resolution) which is delivered as 1 m. This camera has a swath width of 5 km. Satellogic offers multispectral imagery at different processing levels:

- **L1 analytics product** is a top of atmosphere reflectance product [4]. The four-band images are results of a pipeline that incorporates detailed, careful steps of image processing, image alignment, geometric and radiometric calibrations, and orthorectification. This product includes image data in Geotiff (uint16) format, with a geographical accuracy of 10 m CE90.
- **L3 visual product** is a color-corrected product, also known as true color image (red, blue and green). This product includes image data in Geotiff (uint8) format.
- **RapidResponse product** is a result of a lightweight and fast pipeline designed to be run on board of our satellites. In particular, this pipeline produces four-band Geotiff images, with a worse geographical accuracy than L1 products ($30 - 500$ m CE90). The images are not orthorectified on board. The pipeline applies dark frame and flat field corrections to every image. After that, the pipeline convolves each image with the point spread function (PSF) of the sensor to undo the original blurring that comes with it. Finally, the four bands are stitched and stacked together.

2.2 Data collection

The dataset was constructed using captures from six different locations, three from Europe and three from the USA. Captures were taken between April and July 2022. There are two main reasons behind selecting these specific areas. The first and most important reason is that we found that the coverage and quality of the Microsoft building footprints [5, 6], used to create our ground-truth (Sec. 2.3), were more accurate in Europe and USA than other areas around the world. In Figure 1, we show some examples with missing and inaccurate building footprints. The second reason is that the selected captures represent urban areas with a great variation of building characteristics, see Fig.2.

(a) Brazil suburb area. (b) Chile suburb area. (c) Afghanistan rural area.

Figure 1: Examples of poor quality building footprints in different areas of the world.

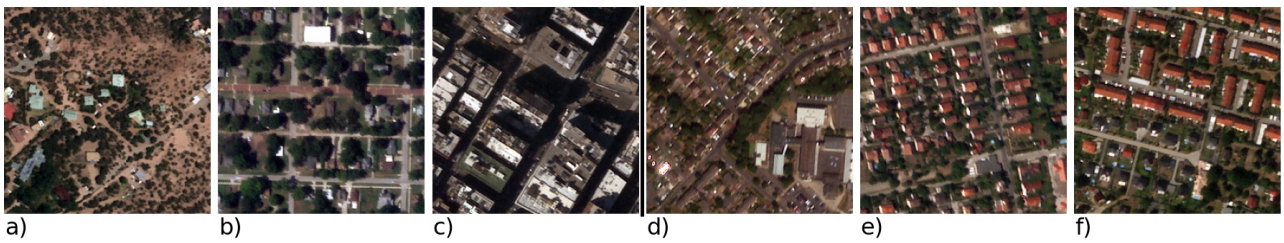


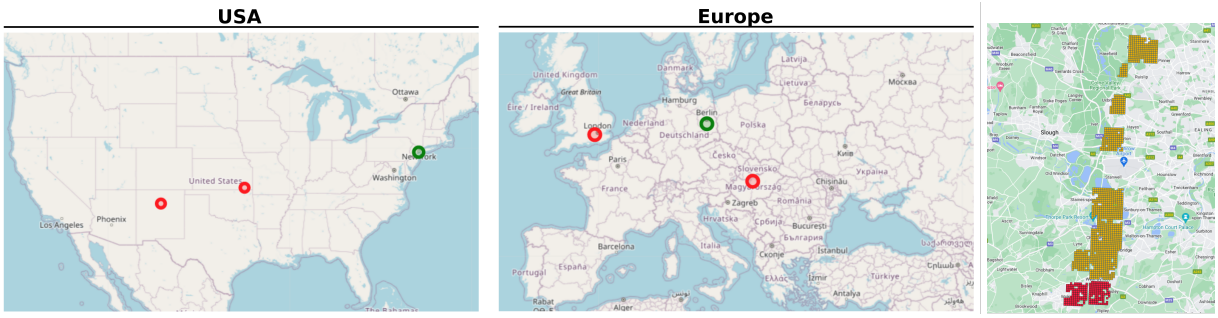
Figure 2: Images around the USA and Europe showing different types of urban areas. a) New Mexico, b) Kansas, c) New York, d) London, e) Budapest, and f) Berlin

Figure 3: Examples of the annotations of L3 (left) and RapidResponse (right) images at different locations showing the input images (left), and the annotations overlaid on them (right). This figure demonstrates the quality of the annotations on the L3 images, as well as the accuracy with which we were able to transform the labels to the RapidResponse images.

2.3 Annotations

In order to obtain building footprint annotations, we evaluated different solutions and data sources, and decided to use the datasets published by Microsoft, [5, 6]. These data cover large areas of the Earth, and were created using high resolution imagery. They are results of an image segmentation model, hence the accuracy of the annotations is variable, and the images used were taken in a period of several years. Since our dataset contains images only from 2022, there are differences due to buildings appearing and disappearing. Many of these differences and errors were corrected manually (some of them by simply removing bad areas within the captures), taking special care of the locations intended as testing partition (see Sec. 2.4). Although we used imperfect labels for training purposes, deep-learning models are known for their robustness against noisy labels [7]. Hence, we let the models deal with them.

As mentioned above, the RapidResponse (RR) images have less accurate georeferencing than those of the L3 images, therefore the annotations could not be used by default. We needed to adjust the footprint polygons to fit the RapidResponse images by calculating the transformations necessary for registering one set of images to the other. We used the Advanced Normalization Tools (ANTs) library [8] to calculate these transformations. ANTs uses the Insight ToolKit image registration framework [9] to calculate both the affine transform and the pixel-level deformations using the symmetric image



(a) (b) (c)

Figure 4: Partitioning strategy followed within the experimental setup. a) Training (red) and testing (green) geographical split in USA; b) Training (red) and testing (green) geographical split in Europe; c) Training (yellow) and validation (red) crops over London capture.

Table 1: Dataset summary

		# of images									
		L1					RapidResponse				
Region	Location	Train	Val	Test	Total	Total Km²	Train	Val	Test	Total	Total Km²
US	Cherrivale, KS	978	254	-	1232	80.74	1002	253	-	1258	82.44
	Santa Fe, NM	2035	372	-	2407	157.74	2044	406	-	2450	160.56
	New York City, NY	-	-	532	532	34.86	-	-	527	527	34.54
EU	Berlin, GE	-	-	754	754	49.36	-	-	755	755	49.47
	Heathrow, UK	1140	217	-	1357	88.93	1073	222	-	1295	84.86
	Budapest, HU	750	236	-	986	64.62	727	239	-	966	63.31
	Total	4903	1079	1286	7268	473.25	4846	1123	1282	7251	475.18

normalization method (SyN) [10]. First, we calculated the transformations necessary to register the RapidResponse images to the L3 ones, then we applied the inverse of the transformations to the corner points of the footprint polygons. Following these steps we were able to achieve a very accurate registration for the majority of the images, and as a result, the labels fit the RapidResponse images just as well as the L3 ones. Figure 3 shows examples demonstrating the quality of the annotations of the RapidResponse images vs the annotations of the L3 imagery.

2.4 Dataset partitioning

As mentioned in Section 2.2, the dataset was built using captures from six locations, three in the USA and three in Europe. The dataset was split into training, validation and test in a geographical manner, in order to assure the correct evaluation of the model. The images of the locations of New York City (NY) and Berlin (Germany) were set aside as test sets, whereas the remaining four locations, Cherrivale (Kansas), Santa Fe (New Mexico), Heathrow (UK) and Budapest (Hungary), were split geographically as train and validation sets. Figure 4 shows the exact location on the map and how the training and validation were split within each capture.

Out of these captures we cropped, for each partition, images of the size of 256×256 pixels with no overlapping in a sliding window fashion, and converted the building footprint polygons into binary masks (1 representing building and 0 background), keeping the original pixel resolution. The summary of the final dataset can be found in Table 1. In total, the dataset covers an area of approximately 475 km^2. We did not balance the amount of images in terms of pixels per class.

Regarding the different partitions, due to the corrections we mention earlier in Section 2.3, training and validation might slightly differ between L3 and RR datasets. Nonetheless, for the test partition, we manually extract the exact same subareas within the test captures, to assure both models are fairly compared.

3 MODELS/EXPERIMENTS

3.1 Semantic Segmentation Model for Building Extraction

3.1.1 Experiment management

The model training and parameter tuning experiments are carried out using IQUAFLOW [3], our open-source framework designed to assess the performance of AI systems under different scenarios. The framework is intended to provide the guarantees and the environmental setup to test how each of the variables can affect the overall performance while being fair and rigorously judged with the same experimental setup.

3.1.2 Model architecture

The neural network design we use in this work follows the UNet architecture [11]. UNet is a convolutional neural network architecture developed for biomedical image semantic segmentation and has been used in several satellite image segmentation studies.

This network contains two modules, the first module is the contraction (also called as the encoder) which is used to capture the context in the image. The encoder is a stack of convolutional and max pooling layers. The second module is the symmetric expanding module (also called as the decoder) which is used to enable precise localization using transposed convolutions. Thus it is an end-to-end Fully Convolutional Network (FCN) with skip connections between the encoder blocks and their symmetric decoder blocks. The encoder consists of the repeated application of two 3×3 convolutions, followed by an Exponential Linear Unit (ELU) and batch normalization. Then a 2×2 max pooling operation is applied to reduce the spatial dimensions. Again, at each downsampling step, we double the number of feature channels, while we cut in half the spatial dimensions. Meanwhile, the decoder module consists of an upsampling of the feature map followed by a 2×2 transpose convolution, which halves the number of feature channels. These are followed by a concatenation step with the corresponding feature map from the contracting module, and a 3×3 convolution with ELU. At the final layer, a 1×1 convolution is used to map the channels to the desired number of classes.

3.1.3 Loss Function and Evaluation Metrics

As a loss function we used soft Dice loss, which is commonly used for image segmentation problems:

$$\mathscr{L} = 1 - \frac{2 \cdot \sum\limits_{\text{pixels}} y_{true} \cdot y_{pred}}{\sum\limits_{\text{pixels}} y_{true} + \sum\limits_{\text{pixels}} y_{pred}} \tag{1}$$

Four commonly used metrics for image segmentation, including precision, recall, F1-score and Jaccard index or intersection over Union (IoU), were used for quantitative evaluation in this study.

The IoU we used consists in dividing the total number of true positives (TP) by the sum of all the positives, TP + false negatives (FN), and false positives (FP):

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{2}$$

The precision score (3) measures the rate of TP among all the detections, while the recall (4) measures the percentage of detected ground truth annotations. Finally, the F1-score is defined as the harmonic mean of precision and recall (5).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{F1} - \text{score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{(\text{Precision} + \text{Recall})} \tag{5}$$

3.2 Model Training

In order to train the network, we used the Adam optimizer with an initial learning rate set to 0.001. The maximum number of epochs was set to 100. The model was monitored using the validation loss to avoid the potential problem of over-fitting. The training phase was, in all the experiments, stopped before reaching the maximum number of epochs. In addition to the Adam optimizer, we used a learning rate reduction scheme based on the ReduceLROnPlateau technique, decreasing the learning rate by a factor of 10 when the loss metric had not improved for nine consecutive epochs. The stopping criterion consisted in ceasing the training after 25 consecutive epochs without improvement of the validation loss. Although significantly high, the model was still able to improve after several epochs being stuck.

3.2.1 Data Augmentation

Data augmentation was proven to be an effective technique to increase the diversity of the training set by applying image transformations avoiding potential problems as over-fitting. This technique increases the ability of the model to generalize and make better, more accurate predictions. In our experiments, we used the Albumentation library [12] to apply either a horizontal or a vertical flip or a random rotation on each image in our training partition, hence duplicating the volume of our training data.

4 RESULTS

We evaluated the performance of our models on the test partition. Table 2 compares the resulting metrics of the L3 and RapidResponse datasets and models. Model training was repeated 5 times for each dataset, and each model was evaluated separately. Table 2 reports the mean and standard deviation of the metrics of the 5 evaluations. The resulting numbers show that the IoU and the F1-score are somewhat higher in the case of the RR model, but the difference is within the standard deviation of the different model runs. Hence, we can conclude that there is no significant difference between the

Table 2: Model test results - each model training was repeated five times, and evaluated separately. The table shows the mean and standard deviation of the five results for each model and test set. Other than the overall metrics, the two locations of the test sets were also evaluated separately.

		Loss	IoU	F1-score	Precision	Recall
Level L3	NY	0.372 (0.020)	0.609 (0.013)	0.629 (0.020)	0.813 (0.009)	0.515 (0.027)
	Berlin	0.372 (0.022)	0.682 (0.012)	0.523 (0.020)	0.674 (0.015)	0.433 (0.028)
	Overall	0.387 (0.015)	0.658 (0.010)	0.613 (0.015)	**0.787** (0.007)	0.513 (0.021)
RapidResponse	NY	0.362 (0.032)	0.611 (0.018)	0.638 (0.032)	0.782 (0.028)	0.543 (0.052)
	Berlin	0.413 (0.013)	0.709 (0.004)	0.565 (0.004)	0.604 (0.040)	0.543 (0.037)
	Overall	**0.360** (0.012)	**0.668** (0.012)	**0.644** (0.012)	0.733 (0.032)	**0.599** (0.043)

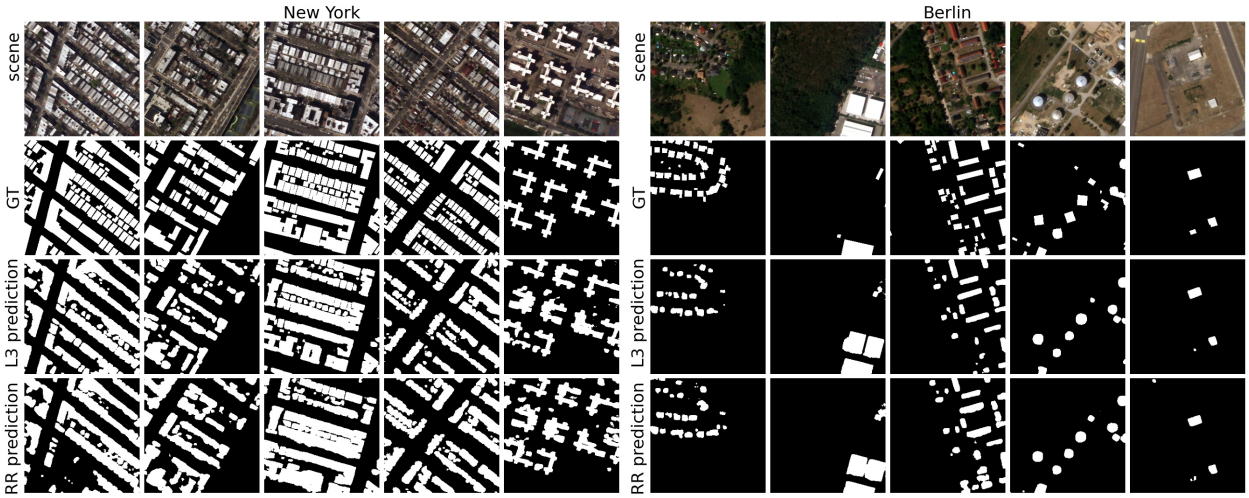


Figure 5: Examples of model predictions at the two locations and by the two models. The first row shows the RGB scene, the second row the ground truth pixel mask, the third row demonstrates the predicted building footprints using the L3 image and the corresponding model, while the last row shows the predictions using the RR image and its corresponding model.

results of the two models, and we were able to achieve the same level of performance using our imagery processed on board as with the level L3 images processed on the ground.

Table 2 also compares the metrics for each of the two different locations in our test partition, while Figure 5 shows examples of predicted building footprints for both regions. Both the IoU and the F1-score values differ for the two areas, the former is higher for the images of Berlin, while the F1-score is better for those of New York. The difference can be explained by the building characteristics and densities of the two locations, which result in a different distribution of building coverage ratios.

We ran the network on three different NVIDIA® devices: 1) a Jetson™ TX2i; 2) a Jetson Orin™; and 3) a GeForce RTX® 3090. For each run, we used the same batch of six images, using the best power mode (Max-N for the Jetsons). In Table 3, we compare the prediction times when using the GPU. We only report the CPU times for the TX2i.

Table 3: Inference times using different NVIDIA® devices. Inference was run on batches of six, and the reported values are averages of five repetitions each.

TX2i		Orin	3090 RTX
GPU	CPU	GPU	GPU
349 *ms*	2,089 *ms*	99 *ms*	45 *ms*

5 CONCLUSIONS

In this work, we used IQUAFLOW to asses the benefits of using edge computing devices on board. More concretely, we tackled the same problem, building footprint segmentation, using L3 imagery, processed on the ground, and our RapidResponse product, created on board. The experiment results show that both models, on board and on the ground, achieved a similar IoU and F1-score, indicating that running the same model on board does not decrease the performance. Therefore, the overall overhead of sending the data to the ground for later processing can be discarded, reducing latencies and costs.

ACKNOWLEDGEMENTS

This work was supported by the European Regional Development Fund (ERDF) and the Spanish Government, Ministerio de Ciencia, Innovación y Universidades—Agencia Estatal de Investigación—RTC2019-007434-7.

REFERENCES

[1] Z. Li, B. Hou, Z. Wu, L. Jiao, B. Ren, and C. Yang, "FCOSR: A Simple Anchor-free Rotated Detector for Aerial Object Detection," 2021. arXiv:2111.10780, unpublished.

[2] Xailient, "Palantir POC - Case Study." `https://xailient.com/casestudies/palantir-poc/`. Accessed: 2022-09-23.

[3] P. Gallés, K. Takáts, M. Hernández-Cabronero, D. Berga, L. Pega, G. Becker, J. Serra-Sagristà, D. Vilaseca, and J. Marín, "IQUAFLOW: A new framework to measure image quality," 2022. arXiv, in prep., see also `https://github.com/satellogic/iquaflow`.

[4] J. C. Vrabel, P. Bresnahan, G. L. Stensaas, C. Anderson, J. Christopherson, M. Kim, and S. Park, "System characterization report on the Satellogic NewSat multispectral sensor." (ver. 1.1, April 2022), chap. L of Ramaseri Chandra, S.N., comp., System characterization of Earth observation sensors: U.S. Geological Survey Open-File Report 2021–1030, 28 p., https://doi.org/10.3133/ofr20211030L.

[5] Microsoft, "Global ML building footprints." `https://github.com/microsoft/GlobalMLBuildingFootprints`. Accessed: 2022-08-30.

[6] Microsoft, "US building footprints." `https://github.com/microsoft/USBuildingFootprints`. Accessed: 2022-08-30.

[7] J. A. Gütter, A. Kruspe, X. X. Zhu, and J. Niebling, "Impact of training set size on the ability of deep neural networks to deal with omission noise," *Frontiers in Remote Sensing*, 2022.

[8] B. B. Avants, N. Tustison, and G. Song, "Advanced normalization tools (ANTS)," *Insight j*, vol. 2, no. 365, pp. 1–35, 2009.

[9] B. B. Avants, N. J. Tustison, M. Stauffer, G. Song, B. Wu, and J. C. Gee, "The Insight ToolKit image registration framework," *Frontiers in neuroinformatics*, vol. 8, p. 44, 2014.

[10] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain," *Medical image analysis*, vol. 12, no. 1, pp. 26–41, 2008.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[12] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020.