PROCEEDINGS OF SCIENCE

Performance estimation of deep learning methods for change detection on satellite images with a low-power GPU

Antonio Di Pilato,[a,*] **Nicolò Taggio**[b] **and Michele Iacobellis**[b]

[a] *University of Bari and National Institute for Nuclear Physics,*
Via Giovanni Amendola, 173, Bari, Italy
[b] *Planetek Italia,*
Via Massaua, 12, Bari, Italy
E-mail: antonio.dipilato@ba.infn.it, taggio@planetek.it, iacobellis@planetek.it

Change detection is an interesting task in the field of remote sensing, thanks to many useful applications that range from land cover studies to anomalies' observation (landslips, snowslides, wide firewoods, floods, etc.). Satellites like Sentinel-2 provide a full coverage of our planet every few days, but transmitting multispectral images of the same region multiple times within a small time interval is not always an efficient operation. At the same time, the analysis of each image on ground requires a considerable amount of time and efforts that might be reduced if knowing in advance that a portion of the new data do not contain any additional information with respect to data acquired in a previous time. Therefore, the idea of comparing onboard a new image with an older one of the same region represents a powerful tool that can help to both reduce the bottleneck effect occurring during the transmission of data to the ground stations and organize the post-processing analysis in a more efficient way.

In this study, deep learning methods are used to perform the change detection task with Sentinel-2 multispectral images. A pre-existing dataset focused on urban changes is exploited for training and validation purposes, while adopting two different approaches (semantic segmentation and classification). In addition, a benchmark test is conducted on a low-power consumption GPU, the NVIDIA Jetson AGX Xavier, to investigate throughput and speed performance with two different inference frameworks, TensorFlow and NVIDIA TensorRT, as this energy-efficient platform is suitable for the installation onboard the satellites in future missions.

International Symposium on Grids & Clouds 2021, ISGC2021 22-26 March 2021
Academia Sinica, Taipei, Taiwan (online)

*Speaker

PoS(ISGC2021)013

1. Introduction

Earth observation (EO) missions have known an unprecedented growth in the last few years. The increased number of satellites launched into orbit and the augmented data production capabilities of the advanced sensor technologies installed onboard the payloads have raised a complex challenge: the data management and transmission. Processing the huge amount of data acquired everyday in a more efficient way is becoming crucial indeed, as it requires innovative solutions both onboard, where the mass memory storage is limited, and on the ground stations, where data are usually available after a considerable time delay due to the bottleneck effect resulting from the limited transmission bandwidth. The so-called "Big Data problem" has thus become a relevant feature of the remote sensing related tasks.

The research of modern solutions to the challenging problem of Earth observation missions has converged to two main approaches. First, the deep learning (DL) approach has the objective of identifying data, in a more automated way, that carry important information, as it would be useful to receive them on ground in shorter times. This strategy implies the introduction of a pre-processing analysis onboard the payloads that involves relevant computational load. Indeed, the execution time and the memory occupancy of DL algorithms depend on their complexity, and become prohibitive when exploiting traditional CPUs only. In addition, the CPUs hosted onboard the spacecrafts must handle important control operations, and most of the time they cannot be employed for complex scientific analyses that would require considerable amounts of time. Therefore, the second approach involves the usage of Graphics Processing Units (GPUs) as more specific dedicated hardware. These devices have proven to be extremely efficient for DL tasks, as they can accelerate the inference process by several orders of magnitude.

However, in-depth studies must be conducted to investigate how these new technologies and methodologies can be exploited within the space scenario, where constraints in terms of energy efficiency and memory management are particularly challenging.

2. Change detection for Earth observation

An interesting application that benefits from the combination of the DL approach with the usage of heterogeneous computing in the field of remote sensing is the change detection (CD) task. Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times [1]. Three main classes of changes can be distinguished:

- *urban changes*: new houses and infrastructures, roads, bridges, and all those events related to human intervention;
- *natural changes*: vegetation growth, snow melting, and all those events related to climate and seasonal variations;
- *anomalies*: floods, wildfires, earthquake disasters, and all those "occasional" events that might be caused from both men and nature.

The large variety of applications, from land usage and surface monitoring to anomaly detection, makes the CD studies a valuable resource of information. The change detection task applied to

the development of urban areas [2, 3] is a very useful case study, aiming at monitoring, in a more automated way, the land usage changes due to urban expansion and urban spread over the years. It is important to remark that clouds and variations of Earth's surface brightness condition are not elements of interest for change detection studies and are usually targeted as "fake" changes; therefore, any DL model trained for this task should not detect them as changes, which is not trivial.

One of the most common strategies adopted to perform the change detection task in EO is to compare two satellite images of the same scene but acquired at different subsequent times. Such comparison can be operated with different methodologies, including DL algorithms. In the onboard-service approach, two different actions would be operated, depending on the amount of changes detected by the algorithms. In the first scenario, a few changes are detected and the onboard system would discard the new acquired image or put it in a "low-priority" queue for delayed download. This action is motivated by the fact that new data do not provide additional information with respect to data of the same scene acquired at a previous time. In the second scenario, instead, a significant number of changes are detected and the onboard system would transmit new data to the ground stations as fast as possible, as they might contain important information and should be submitted to an in-depth analysis workflow. The objective of this study is twofold:

1. testing some DL models developed for change detection studies;
2. evaluating the performance of a low-power consumption GPU that could fit the space scenario and its constraints.

The first task was performed by adopting two different approaches: semantic segmentation and classification. The semantic segmentation approach is more suitable for in-depth analyses on the ground stations, as the output of the model developed for this method is a binary map that labels each pixel as changed or unchanged (pixel-wise classification). In order to obtain such a high-level output format, the complexity of the algorithm is not negligible in the onboard scenario. The classification approach, instead, aims to provide fast results at the cost of losing part of the information that could be later recovered with post-processing; therefore, the output of the model is a single score indicating if the image pair contains a significant number of changes or not (thus being suitable for the execution onboard).

3. Dataset overview

In this work, data acquired by the Copernicus Sentinel-2 mission, developed by ESA, were used. Sentinel-2 consists of a constellation of two twin satellites flying in the same Sun synchronous polar orbit but phased at 180° at a mean altitude of 786 km, designed to give a high revisit frequency of 5 days at the Equator with a field of view of 290 km. It aims at monitoring variability in land surface conditions through the acquisition of high-resolution multi-spectral images that can be exploited for land cover/change classification, atmospheric correction and cloud/snow separation.

Each satellite carries a Multispectral Instrument (MSI) for data acquisition. The MSI measures the Earth's reflected radiance in 13 spectral bands: the visible and near-infrared (VNIR) and the short-wave infrared (SWIR) ranges, with three different spatial resolutions (10 m, 20 m and 60 m). A schematic view of Sentinel-2 MSI is shown in Figure 1.

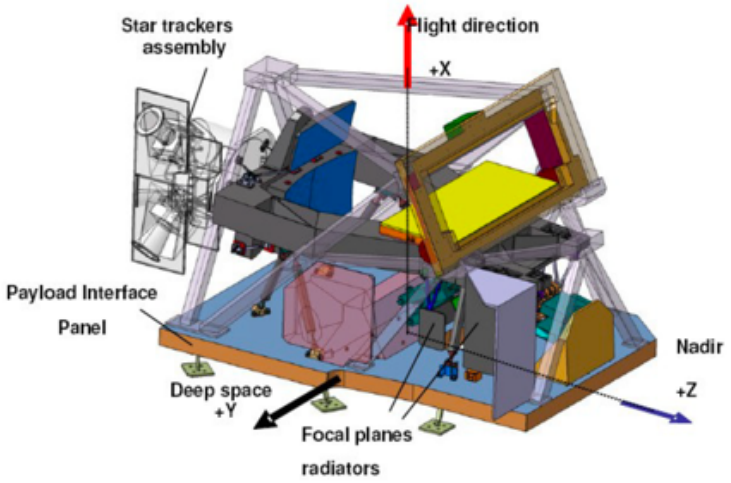

Figure 1: Sentinel-2 MSI internal configuration [4].

Even though large amounts of data are acquired by the Sentinel-2 mission and available for free, most of the studies require that ground truths (or labels) must be generated with the help of human eye, in a less automated way. Therefore, since the creation of a new dataset was not intended among the objectives of this study and it represents a difficult and time-consuming operation, an existing dataset was used for the purpose. The Onera Satellite Change Detection (OSCD) dataset [5] consists of 24 Sentinel-2 L1C image pairs, covering regions of approximately 600×600 pixels at 10 m resolution ($2 - 3$ years time difference) with no (or few) clouds. Bands at 20 m and 60 m resolution were upsampled to the resolution of 10 m such that all the channels have aligned pixels. Pixel-wise ground truths are available for 14 samples only, in the form of binary "change maps" (CMs). The dataset is focused on urban areas; thus, only urban changes are labeled, while natural changes are ignored. A sample image pair (RGB-only) of the Pisa area is shown in Figure 2.

Figure 2: Sample Sentinel-2 image pair of the Pisa area. Left is the image "pre" (before changes), and right is the image "post" (after changes) [5].

4. Training strategy

The OSCD dataset is characterized by few images having large sizes; therefore, it is important to operate a pre-processing step in order to exploit such data for training purposes. For each of the 14 regions, patches of size 128×128 were generated by pointing to random pixel locations (x, y) and cropping the images $[x, x + a, y, y + a]$, where $a = 128$. All the 13 bands were used in this work (thus each image of a pair has size $128 \times 128 \times 13$). Every selection was operated such that the resulting crop was totally included within the region borders. In addition, one or more (depending on the changes content) random transformations were applied among the following ones: rotation of 90°, 180°, 270° or 360°, vertical or horizontal flip. If the ratio between the number of changed pixels and the total number of pixels was less than 10%, only one transformation among the available ones was applied and the resulting patch was stored in the training dataset; otherwise, all the six transformations were operated, one at time, and the six resulting images saved. The same selections and transformations were applied to both the pre-changes and the post-changes images, as well as to the associated ground truth. The random selection of the pixel locations and the applied transformation(s) helps to reduce the effect of pixel correlation between patches generated from the same initial image. A number of 500 patches were selected in each region, for a total of 13919 input pairs.

Another important aspect of this study was the choice of the loss function that should be minimized during the learning process. Using a classical binary cross-entropy loss, that generally provides good results for classification tasks (note that semantic segmentation is nothing but a pixel-by-pixel classification), would not perform as well as expected. Indeed, the OSCD dataset (and consequently the training dataset generated with the process described above) is unbalanced with respect to the classes' population. This is a typical feature of change detection datasets, due to the fact that changes can be considered "rare events" as the number of changed pixels only cover a small fraction of the total number of pixels in each image (about 1%). Therefore, a *weighted binary cross-entropy* loss represented a natural choice for this task:

$$\mathcal{L}^{bce} = -\sum_i w_i t_i \log s_i \tag{1}$$

where $i = 0, 1$ (binary classification), w_i is the weight assigned to class i (inversely proportional to class population, such that samples of the most populated class have a lower weight during the calculation of the loss function), t_i and s_i are respectively the ground truth and the output with respect to class i, both expressed in the form of one-hot vectors.

In addition, k-fold cross validation was adopted, representing a very useful method to evaluate the performance of the deep learning models, especially when the dataset is small. Specifically, the dataset was split into k=5 equal parts, with four of them used for training and the last part reserved for validation. This procedure was repeated five times, each time using a different partition of the dataset for training and the remaining samples for validation. In the end, five models were trained and validated for each architecture, and average scores were evaluated in order to measure the performance of the model architectures independently of the chosen partition.

4.1 Metrics of interest

In order to evaluate the performance of the trained models, the following metrics have been considered:

$$balanced\ accuracy = \frac{1}{2}\left(\frac{T_P}{T_P + F_N} + \frac{T_N}{T_N + F_P}\right) \tag{2}$$

$$precision = \frac{T_P}{T_P + F_P} \tag{3}$$

$$recall = \frac{T_P}{T_P + F_N} \tag{4}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

where T_P are *true positives*, T_N *true negatives*, F_P *false positives* and F_N *false negatives*. In this study, positives are changed pixels.

The *balanced accuracy* represents the mean value of two *accuracy* scores, evaluated on the two classes separately (especially efficient for unbalanced datasets); the *precision* is the fraction of positive predictions that are actually correct, while the *recall* quantifies the fraction of actual positives that are correctly identified. In this work, *recall* was chosen as the most important score to evaluate and to be maximized. Indeed, the general idea is that it is preferable to have more F_P than losing real changes (resulting in a higher number of F_N); however, *F1-score* was also considered, as it combines both *precision* and *recall* to provide an extremely useful value that defines the model performance.

5. Semantic segmentation approach

As stated above, the semantic segmentation approach aims to produce a binary change map where each pixel is either 0 (unchanged) or 1 (changed), depending on the aligned pixels of the two images acquired on the same region at different times. A UNet-like [6] architecture (fully-convolutional neural network) is developed for the semantic segmentation approach and shown in Figure 3. First, the pre-changes and the post-changes images are concatenated along the channel axis; then, the CNN encoder operates feature extraction on the new image of size $128 \times 128 \times 26$. It consists of several convolutional layers and max pooling layers, which partially drop spatial information that is later recovered by the CNN decoder through the skip-connections method (concatenation of layers in the decoding part of the network with corresponding layers, of the same size, from the encoding part). The CNN decoder reconstructs the binary change map by exploiting features extracted by the encoding part of the network, providing the final output of the algorithm.

An important preliminary step is the selection of the change threshold. Indeed, in the output change map each pixel has a value that ranges from 0 to 1; however, marking as "changes" those pixels whole value is above 0.5 is not necessarily the best choice according to the objective of this work. In order to maximize the *recall* value while keeping high the value of the *F1-score* too, the average values (across the five models trained with cross validation) of the metrics were calculated for different thresholds and shown in Figure 4. Selecting a threshold equal to 0.3 (which means

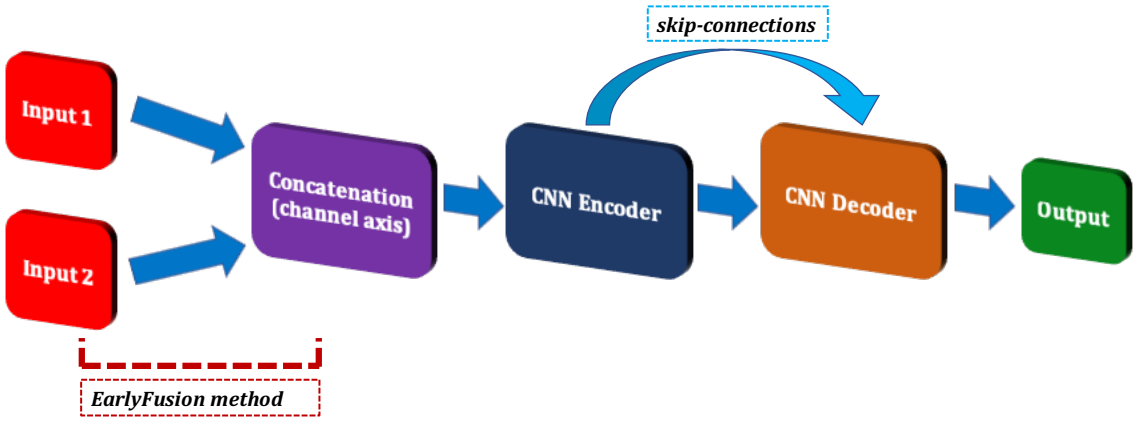

Figure 3: Model architecture developed for the semantic segmentation approach. The two input images are concatenated along the channel axis (*EarlyFusion* method); then, the CNN encoder extracts features at different abstraction levels, while the CNN decoder reconstructs the output change map, recovering spatial information through the skip-connection technique.

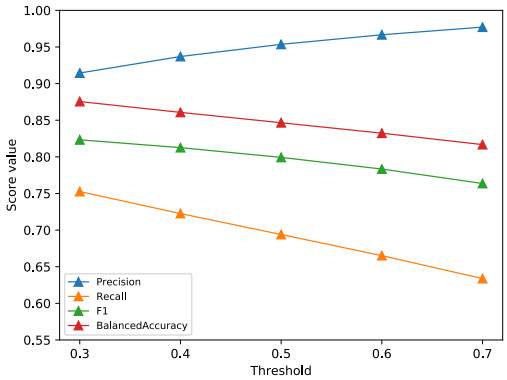

Figure 4: Trends of the average values of the metrics of interest as a function of the change threshold, for the semantic segmentation approach.

predicting as changes all those pixels, in the output change map, having values of 0.3 or larger) maximizes the *recall*; furthermore, the *precision* value is larger than 90%, which indicates that the fraction of false positives is still small and ensures the goodness of the model performance.

Figure 5 shows the validation scores obtained by the five models trained with cross validation (*left*) and their average values (*right*) for the semantic segmentation approach. Error bars are calculated as the standard deviation across the five models trained with cross validation. The results are comparable with the state of the art [7]. The large error bar on the *recall* score is caused by anomalous performance of one of the five models during the training process; despite regularization techniques were used, small effects of vanishing gradient (loss function stuck in a local minimum) might always occur.

5.1 Sample change maps

Two sample change maps reconstructed by the semantic segmentation model are here presented. Figure 6 shows the comparison between the ground truth and the predicted change map on the Pisa

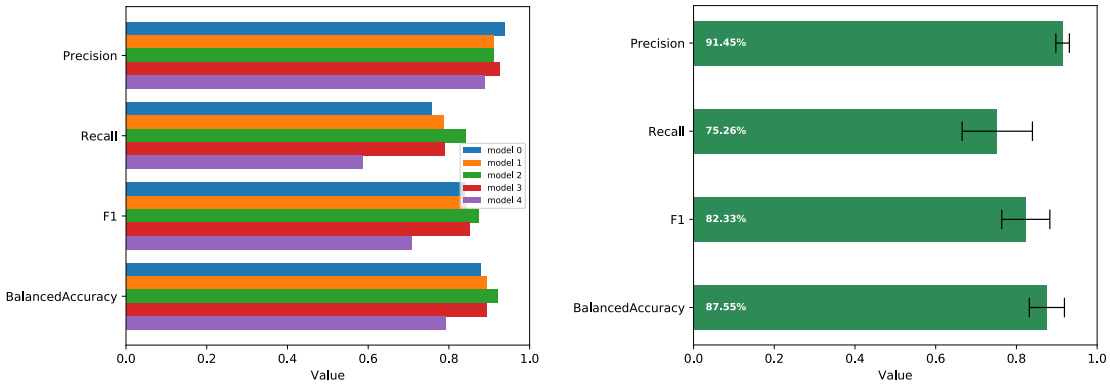

Figure 5: (*Left*) Validation scores of the five models trained with cross validation. (*Right*) Average values of the validation scores obtained by the five models. Error bars represent the standard deviation calculated across the five models. Plots represents results obtained with the semantic segmentation approach.

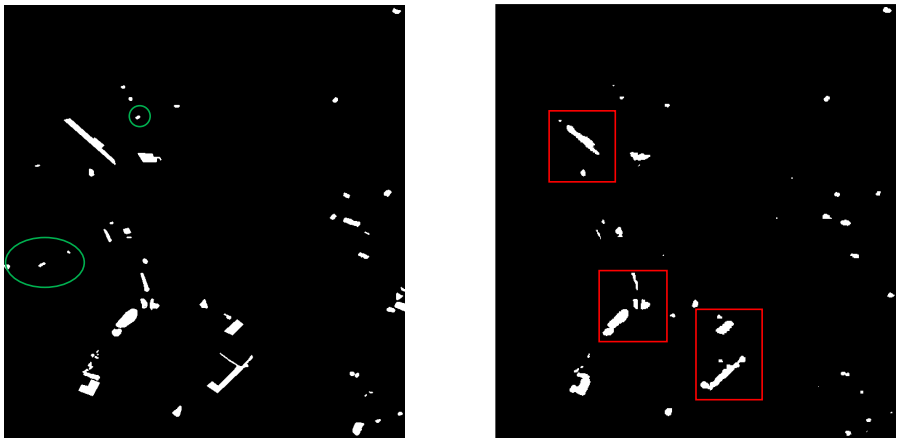

Figure 6: Comparison between the ground truth (*left*) and the predicted change map (*right*) of the Pisa area. In green are outlined some of the small changes that are not detected by the algorithm, while in red are located some large changed structures correctly identified.

area (the pre-changes and post changes images have been shown in Figure 2). Most of the changes are detected at the correct locations, but not perfectly outlined. This result indicates that a possible overtraining effect is mitigated by regularization techniques, dropout and batch normalization layers included within the network's architecture. In addition, a few small changes are not detected, while the reconstruction of large changed clusters can be improved by using proper post-processing algorithms.

Figure 7 shows the image pair of the Hong Kong area included in the OSCD dataset, and the comparison between the ground truth and the predicted change map. Also in this case, some small changes are not detected while large structures are correctly outlined, suggesting that network capacity of detecting changed regions is limited to clusters of a certain size. In addition, a region of false positives is located in the bottom left part of the change map, which represents a "drawback" effect of having selected a change threshold equal to 0.3. The choice of a higher change threshold might exclude such structure from predictions at the cost of a lower recall value.

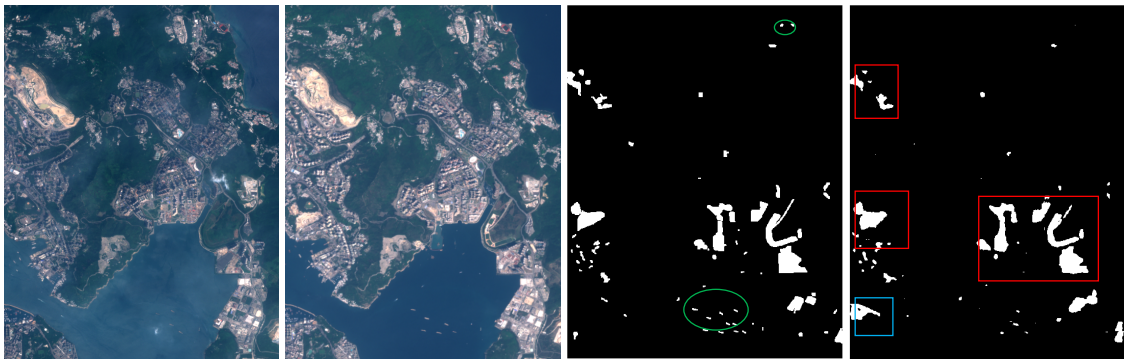

Figure 7: From left to right: sample Sentinel-2 image pair of the Hong Kong area (pre-changes and post-changes images) [5] and comparison between the ground truth and the predicted change map. In green are outlined some of the small changes that are not detected by the algorithm, in red are located some large changed structures correctly identified, while the blue square outlines a region of false positives.

6. Classification approach

The classification model was designed with speed and efficiency in mind; therefore, a smaller architecture was chosen to perform change detection onboard future satellite missions. A classical CNN was used for the purpose, and the architecture is shown in Figure 8. The two input images are concatenated along the channel axis and features are extracted by the CNN encoder as in the semantic segmentation approach. Then, a flattening operation is performed to achieve dimensionality reduction, followed by a multi-layer perceptron part consisting of a few dense layers and a single output neuron providing the final score.

Ground truth was defined based on the change map provided in the OSCD dataset. The ratio between the number of changed pixels and the total number of pixels was again calculated; if its value was larger than 0.15% (that, for images of size 128×128, corresponds to ~ 25 pixels), the image pair was labeled as containing changes of interest (numerically 1, 0 otherwise). In addition,

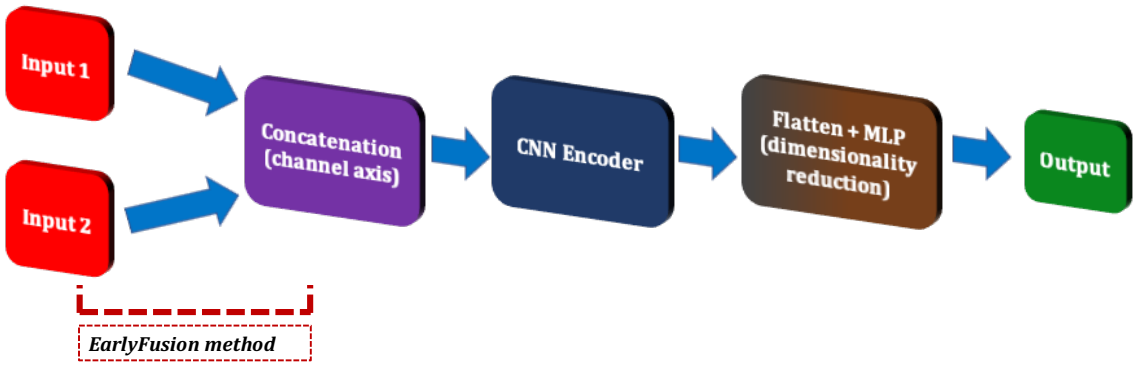

Figure 8: Model architecture developed for the classification approach. The two input images are concatenated along the channel axis (*EarlyFusion* method), and the CNN encoder extracts features at different abstraction levels. Dimensionality reduction is then performed, and a few dense layers provide the final score (single neuron output).

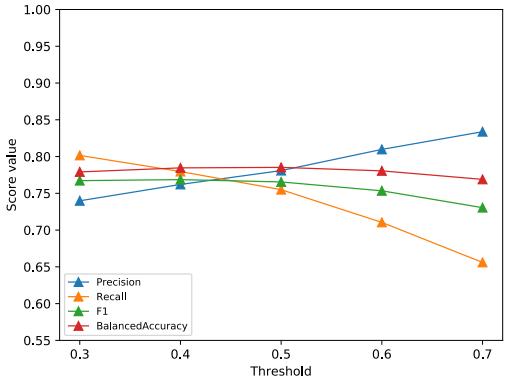

Figure 9: Trends of the average values of the metrics of interest as a function of the change threshold, for the classification approach.

since the output of the network is a single score ranging from 0 to 1, the change threshold represents here the minimum value above which such output is considered to be equal to 1. The threshold was set to 0.3 as in the semantic segmentation approach, as it provides the highest values for both *recall* and *F1-score* (the latter being almost constant in the range $0.3 - 0.5$). The average values of the scores as a function of the change threshold are shown in Figure 9.

Figure 10 shows the validation scores obtained by the five models trained with cross validation (*left*) and their average values (*right*) for the classification approach. Error bars are calculated as the standard deviation across the five models trained with cross validation. High values of *recall* and *F1-score* are achieved, even though they strongly depend on the ground truth definition. Indeed, in this work a simple choice was operated, without restrictive requirements on the changed pixels. However, better choices could be adopted, such as requiring the presence of a large cluster of changed pixels in the change map, etc. The results obtained with this simple approach are though very promising, and future improvements might be considered for the execution of deep learning algorithms onboard the satellites before downloading new data to the ground stations for further

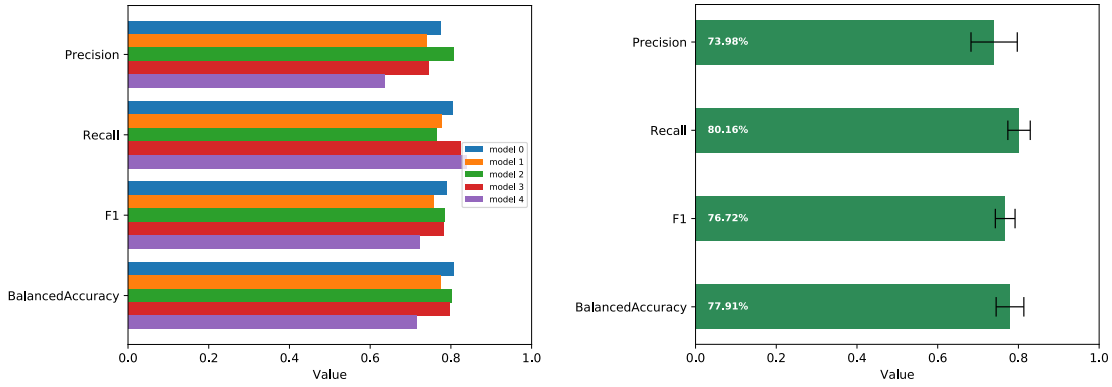

Figure 10: (*Left*) Validation scores of the five models trained with cross validation. (*Right*) Average values of the validation scores obtained by the five models. Error bars represent the standard deviation calculated across the five models. Plots represents results obtained with the classification approach.

analyses. It is important to note that, due to its smaller architecture, the classification model can be executed faster than the fully-convolutional network used for semantic segmentation, and requires less memory occupancy. These two features make the CNN ideal for the execution onboard satellites to allow a fast decision making process.

7. Benchmark test

The semantic segmentation and classification models were both tested with a low-power consumption GPU: the NVIDIA Jetson AGX Xavier. This device delivers the performance of a GPU workstation in an embedded module that features a 512-core Volta GPU with 64 Tensor Cores (accelerating dense linear algebra computations), a 8-core ARM v8.2 64-bit CPU and 2 Deep Learning Accelerator (DLA) cores (accelerating neural network execution efficiently). Seven power modes are available, differing from each other for power consumption, number of active CPU cores, GPU maximal frequency and activation status of DLA cores. Two power modes were selected in this test: the 15 W default mode, optimal compromise for desktop-usage and low power consumption, and the 30 W mode, which delivers high performance for scientific purpose.

The benchmark test was set with the following procedure. First, 50 warmup runs were performed, to ensure that measurements were made with the same GPU's initial status. Afterwards, 200 valid runs were operated, each of them consisting of performing inference with the selected model on the entire batch of 42 image pairs, corresponding to the Pisa area included in the OSCD dataset. Inference time was measured for each run and, finally, throughput was calculated with the following formula:

$$Throughput = \frac{nRuns \times nImagePairs}{elapsedTime} \tag{6}$$

The benchmark test involved two inference frameworks: TensorFlow, also exploited during the training process of the models, and NVIDIA TensorRT, a software development kit for high-performance deep learning inference on NVIDIA GPUs, delivering low latency and high throughput. Furthermore, a comparison was made with the performance of TensorFlow on a NVIDIA Tesla K40, a GPU commonly employed in computing data centers.

Throughput results are shown in Figure 11. NVIDIA TensorRT provides the highest throughput values, providing speedup factors of 32x and 11x with respect to TensorFlow for the semantic segmentation and the classification models respectively. The different speedup factors is related to the design of the software, which can optimize complex architectures better than smaller ones (due to the larger number of layers that can be optimized and accelerated at inference time). NVIDIA TensorRT was only used with the GPU Jetson Xavier (in half-precision mode, FP16), since Tesla K40 is designed for high performance and the throughput delivered by the inference accelerator on this device was not interesting for the study conducted. For the semantic segmentation model, TensorRT allows to perform inference on 233 and 292 images per second with power modes 15 W and 30 W respectively, while for the classification model these values increase to 303 and 361 images per second. In both cases, TensorRT on Jetson Xavier outperforms TensorFlow on Tesla K40, which has about 10 times higher power consumption.

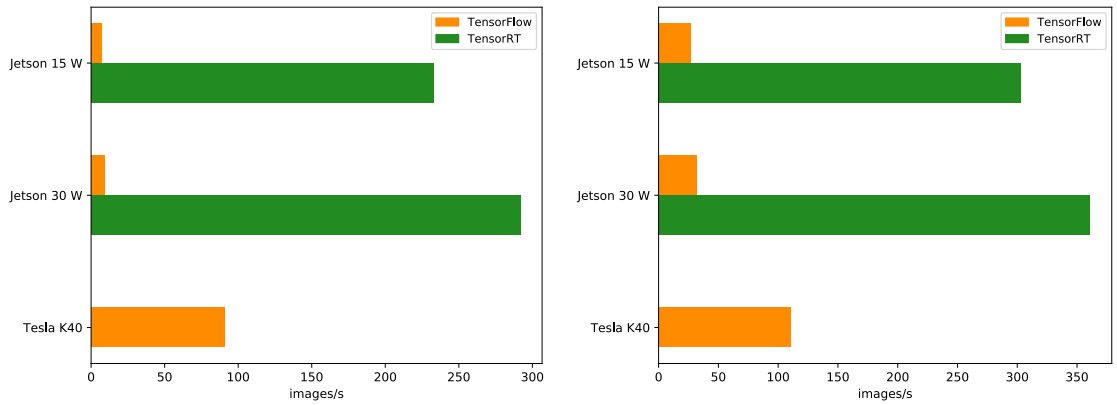

Figure 11: Throughput results for the semantic segmentation (*left*) and the classification (*right*) models.

8. Conclusions

Artificial intelligence brings a powerful set of tools in the field of remote sensing, as it allows the satellites to perform operations and make decisions in an autonomous way. Furthermore, the interest towards change detection is increasing in last few years, thanks to its useful applications, such as climate and urban growth studies, emergency management and anomaly detection. However, efforts are required to integrate heterogeneous platforms in the onboard system, as they can accelerate the decision making process by several orders of magnitude. Low-power consumption devices must be considered to fit the space scenario and its constraints, while improving the memory and energy management onboard the future satellite missions.

9. Acknowledgements

The authors would like to thank the organization team of the online conference "International Symposium on Grids and Clouds 2021" for the great opportunity and the possibility to discuss the results of this work with the scientific community. Also, this work was supported by Planetek Italia and the ReCaS data center management team, as they provided access to powerful and high-performance devices.

References

[1] Ashbindu Singh, *Review Article Digital change detection techniques using remotely-sensed data*, International Journal of Remote Sensing, **10**, https://doi.org/10.1080/01431168908903939 (1989)

[2] Hayet Si Salah et al., *Change detection in urban areas from remote sensing data: a multidimensional classification scheme*, International Journal of Remote Sensing, **40**, https://doi.org/10.1080/01431161.2019.1583394 (2019)

[3] Maria Papadomanolaki et al., *Detecting urban changes with recurrent neural networks from multitemporal Sentinel-2 data*, IEEE International Geoscience and Remote Sensing Symposium, https://hal.inria.fr/hal-02266094 (2019)

[4] V. Cazaubiel et al., *The multispectral instrument of the Sentinel-2 program*, International Conference on Space Optics, https://doi.org/10.1117/12.2308278 (2017)

[5] R. C. Daudt et al., *Urban Change Detection for Multispectral Earth Observation Using Convolutional Neural Networks*, arXiv:1810.08468 [cs.CV] (2018)

[6] O. Ronneberger et al., *U-Net: Convolutional Networks for Biomedical Image Segmentation*, arXiv:1505.04597 [cs.CV] (2015)

[7] R. C. Daudt et al., *Fully Convolutional Siamese Networks for Change Detection*, arXiv:1810.08462 [cs.CV] (2018)