

Monitoring of Urban Changes With Multimodal Sentinel 1 and 2 Data in Mariupol, Ukraine, in 2022/23

Georg Zitzlsberger^{1b} and Michal Podhoranyi^{1b}

Abstract—The ability to constantly monitor urban changes is of significant socio-economic interest, such as detecting trends in urban expansion or tracking the vitality of urban areas. Especially in present conflict zones or disaster areas, such insights provide valuable information to keep track of the current situation. However, they are often subject to limited data availability in space and time. We built on our previous work, which used a transferred deep neural network operating on multimodal Sentinel 1 and 2 data. In the current study, we have demonstrated and discussed its applicability in monitoring the present conflict zone of Mariupol, Ukraine, with high-temporal resolution Sentinel time series for the years 2022/23. A transfer to that conflict zone was challenging due to the limited availability of recent very high resolution (VHR) data. The current work had two objectives. First, transfer learning with older and publicly available VHR data was shown to be sufficient. That guaranteed the availability of more and less expensive data as time constraints were relaxed. Second, in an ablation study, we analyzed the effects of loss of observations to demonstrate the resiliency of our method. That was of particular interest due to the malfunctioning of Sentinel 1B shortly before the selected conflict. Our study demonstrated that urban change monitoring is possible for present conflict zones after transferring with older VHR data. It also indicated that, despite the multimodal input, our method was more dependent on optical multispectral than synthetic aperture radar observations but resilient to loss of observations.

Index Terms—Deep neural network (DNN), multimodal, remote sensing, transfer learning, urban change monitoring.

I. INTRODUCTION

THE detection of changes with the use of satellite based remote sensing data has a history of almost six decades, with the first mentioning of a *change detector device* by [1]. Since then, many methods have been developed to detect changes [2], [3]. While many methods have been proposed to detect changes,

Manuscript received 20 October 2023; revised 22 December 2023 and 15 January 2024; accepted 29 January 2024. Date of publication 6 February 2024; date of current version 28 February 2024. This work was supported in part by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPS II) project “IT4Innovations excellence in science - LQ1602” and by the IT4Innovations Infrastructure, in part by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140), in part by the Open Access Grant Competition (OPEN-25-24 and OPEN-27-1), and in part by the ESA Network of Resources Initiative (ID:2923ca) to provide access to Sentinel Hub, and Airbus Pléiades. (Corresponding author: Georg Zitzlsberger.)

The authors are with the IT4Innovations, VSB – Technical University, 70833 Ostrava, Czech Republic (e-mail: georg@zitzlsberger.com).

Digital Object Identifier 10.1109/JSTARS.2024.3362688

in the last decade the focus moved more toward using neural networks [4], [5], [6], [7], [8], [9], [10] with the advent of DNNs in that time frame. The types of changes across those works vary drastically. Some works detect changes in general, including vegetation and water bodies, others consider only buildings but ignore other infrastructure. Only a subset considers urban structure types (UST) as defined by [11]. In addition, the majority of works operate on observation pairs that are required to be of sufficient quality. The result was a large occurrence of so-called siamese network architectures, which replicate the network on the input side for each image as a pair. Overall, they are limited for broad use due to requiring high-quality very high resolution (VHR) data, and they reduce the temporal resolution to detect and monitor urban changes.

In our previous works, summarized in Fig. 1, we have addressed these problems by introducing a new approach. First, we have introduced a method that leveraged an ensemble of neural networks for Level 1 Sentinel 1 and 2 multimodal remote sensing data [i.e., synthetic aperture radar (SAR) and optical multispectral]. Its design was tailored for the objective to continuously monitor urban changes [12]. This method operated on time series observations, partitioned into half-year windows, to provide enough context for allowing low quality Level 1 data and to localize changes over time. It pretrained a model, called *ensemble of recurrent convolutional neural networks for deep remote sensing* (ERCNN-DRS), with synthetic but noisy labels to avoid manual labor. In a follow-up study [13], a further optimization with transfer learning was demonstrated to fine-tune the pretrained network toward a specific area of interest (AoI) to increase the detection quality and allow more control of the UST changes to detect. For practical feasibility, the transfer learning used a set of windows simultaneously in order to simplify the manual ground truth generation guided by public VHR data, i.e., Google Earth historic imagery, and spanned a larger time frame of multiple years. Both works were trained and verified only for a fixed time frame, i.e., 2017–2020. It, therefore, raised the question whether a model, transferred to one time frame, would also be applicable for a different time frame.

In this work, we reused the pretrained ERCNN-DRS model and applied the same transfer learning method but to the AoI of Mariupol for the time frame 2017–2020. We subsequently evaluated the performance of the transferred model for the years 2022/23. This was done starting three months before the Russian

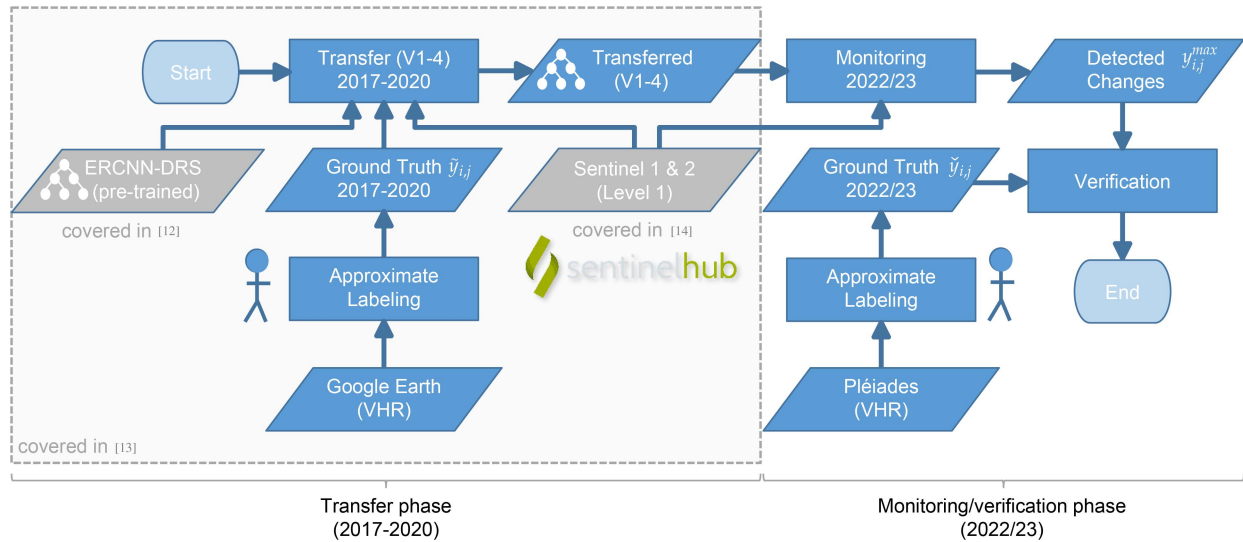


Fig. 1. Flowchart of the transfer learning and monitoring process for the AoI of Mariupol. In blue are data and processing steps of our current work; gray denotes previous work.

invasion on 24 February 2022, and until mid-2023. Due to being an active war zone, VHR remote sensing data was limited, even if commercially accessible.¹ However, medium resolution data, such as Sentinel 1 and 2 were still available for that region without limitations and allowed the monitoring of ongoing urban changes. We analyzed the applicability of transferring to the AoI of Mariupol for the time frame 2017–2020 and its use for 2022/23. Because of the lack of public VHR data for that time frame, commercial Airbus Pléiades observations were used for validation purposes. These would be expensive for transfer learning due to the required amount but we used them only for evaluation, which does not require a larger area and kept costs low.² As we demonstrate, the transfer with an earlier time frame using public data can be sufficient, which helps to keep costs low.

Furthermore, due to the outage of Sentinel 1B on December 23, 2021, we also addressed the question of the impact of loss of observations to the overall solution. As we will show in an ablation study, the chosen method is resilient and does not instantly break down if final observation patterns diverge from the ones seen during training.

This work was subject to two objectives. First, we analyzed whether a transferred model for a new AoI for a specific time frame can be used for a later time frame, even though observation patterns change. Second, the resiliency of our method to a decaying number of observations for the different observation modes, SAR and optical multispectral, were studied. We addressed both objectives with the case study to monitor urban changes of the, at the time of writing, besieged and occupied city of Mariupol in Ukraine where limited data is available despite the increased need to monitor changes.

¹Higher resolution Maxar WorldView data (0.15–0.3 m/pixel) for Ukraine was under embargo at the time of writing.

²ESA NoR sponsorship worth 600 € for the Airbus Pléiades data enabled the verification of our work.

The rest of this article is organized as follows. Section II describes our approach from the transfer of an existing pretrained ERCNN-DRS to verification. Section III provides details on the selected study area, observation data, and data processing. The training procedure is explained in Section IV. Training and verification results are discussed in Section V with both quantitative and qualitative analysis. This section also contains the ablation study to understand the resiliency of our method. In Section VI, we summarize the shortcomings of our approach that require further work. Section VII contains a discussion on the peculiarities and tradeoffs of our methods to give guidance to adapting it to other scenarios. Finally, Section VIII concludes this article and summarizes key results and areas of improvement.

II. METHODOLOGY

We built on top of three existing works to enable urban change monitoring and combined these in the current work for the different AoI of Mariupol. Fig. 1 summarizes our two-step approach from transfer to validation. Reused data from other works is in gray, and items covered in this work are in blue. Most of the steps were automatized with only the labeling procedures carried out manually. Our applied methodology is described in the following, separated by reuse of existing methods and their extensions needed by this work.

A. Existing Methods

The pretrained baseline ERCNN-DRS model stems from [12]. The model architecture is shown in Fig. 18 in Appendix A for completeness. That work laid the foundation of utilizing large observation counts, so-called deep-temporal windows $w_{i,j}^t$ of a fixed duration, with multimodal remote sensing data for identifying urban changes. These windows start at time t and are tiled with i and j being tile coordinates. Furthermore, these

TABLE I
IMPORTANT REMOTE SENSING DATA, TILE, AND WINDOW PARAMETERS

	Parameter	Mnemonic	Value
data	$b_{SAR}^{[asc dsc]}$	SAR bands	2 (VV+VH)
	b_{OPT}	optical bands	13
tiles	x	x -dimension	32 pixel
	y	y -dimension	32 pixel
windows	ρ	stride	1 (# obs.)
	δ	step ($\frac{\text{observation}}{\text{observation}}$)	2 days
	Δ	window period	6 months
	ω	min. window size	35 (# obs.)
	Ω	max. window size	92 (# obs.)

These were unchanged from the pretrained network stage and used identically in our current work.

windows were applied in a sliding window approach, allowing for a longer observation period than the fixed window duration.

To create the windowed time series the *rsdtlib* library [14] was utilized. It retrieves Sentinel 1 and 2 remote sensing Level 1 observations from *Sentinel Hub* and preprocesses them. The preprocessing involves temporal stacking, assembling, tiling, and windowing to retrieve the final multimodal windows $w_{i,j}^t$. Important parameters of the data pipeline stages are shown in Table I, characterizing the data, tiles, and windows. In the following we briefly describe these steps.

The multimodal data was defined by the number of bands, which were the two polarizations vertical–vertical (VV) and vertical–horizontal (VH) for SAR, and spectral channels for optical observations (13 bands in the range of ca. 440–2200 nm). For SAR, we considered ascending and descending orbit directions as individual observation modes. These have different observation directions and, hence, cannot be directly compared where larger elevation differences are present. Each observation mode was temporally stacked to only update pixels in the observations that were not masked due to clouds or out-of-swath. If masked, the value of the pixel in the previous observation was carried forward.

Due to memory constraints, the entire AoI was not considered at once. Instead, it was tiled into nonoverlapping 32×32 pixels patches. This was a configuration used for the pretraining, and so was used in this work as well.³

As shown in Fig. 2, windows were constructed from these multimode observations by assembling them into observations with a sampling step of δ . Using two days showed a good compromise of avoiding redundant observations (e.g., due to swath overlaps) and retaining high temporal resolution for the purpose of urban change monitoring. In the following, we use $\delta = 2$ days, unless otherwise noted. Windows were of a fixed period Δ of half a year and windows with fewer than $\omega = 35$ observations were discarded to ensure enough data points were available. Due to the window period and step size, there is a natural upper bound of observations per window $\Omega = 92 \approx \Delta/\delta$. A unit-stride ($\rho = 1$) was used for the sliding windows, which derived windows starting at every next sampled observation. Hence, every δ -sampled observation defined the start of a new

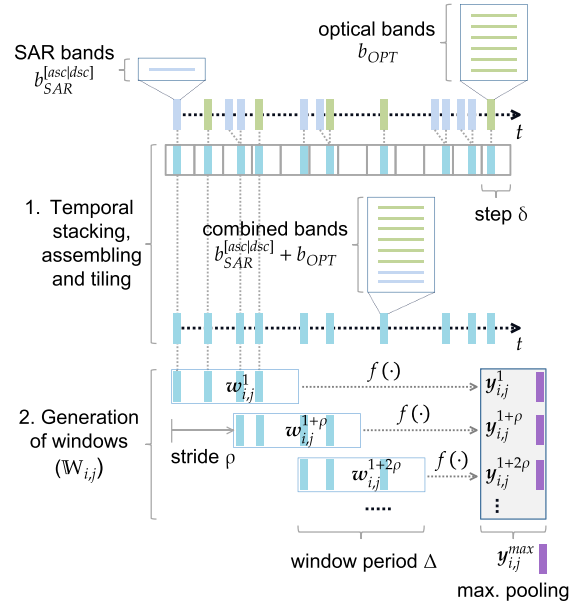


Fig. 2. Two steps of generating the set of windows $\mathbb{W}_{i,j}$ for each tile with coordinates i and j . The window predictions were used in combination with *maximum pooling over time* to retrieve a combined prediction $y_{i,j}^{max}$ during the transfer phase.

window, i.e., every two days or longer, depending on the amount of observations over time. As a result, a set of windows $w_{i,j}^t \in \mathbb{W}_{i,j}$ with the same parameters was retrieved that can be used for transfer learning or inference.

In the previous work [13], the transfer and optimization of the pretrained ERCNN-DRS model was already demonstrated. It used a different AoI (Liège in Belgium) compared with the pretraining AoIs for the time frame 2017–2020. The transfer was realized with a small amount of manually labeled tiles. It was demonstrated that, even with an already low effort ground truth guided by Google Earth historic imagery, the transfer showed an improved performance. Instead of labeling each window, a set of windows spanning a larger time frame of four years, was considered for labeling. To be able to train with a set of windows at each step, a maximum pooling method was applied. The maximum pooling followed the principle of *maximum pooling over time* [15]. Fig. 2 shows the maximum pooling of the individual window predictions $y_{i,j}^t$ to retrieve $y_{i,j}^{max}$.

To leverage the limited dataset size, bagging (bootstrap aggregating) [16] was utilized, using a nonexhaustive cross validation. Three model variants were trained with this cross-validation approach that used disjunct validation data from the overall dataset for every transfer variant. The bagging resulted in an ensemble of weak learners from the variants, which provided a better performance and precision/recall balance than each individual variant. However, this was only executed and validated for the same time frame as the pretraining.

B. Extension and Modification of Methods

In this work, we applied transfer learning to the pretrained ERCNN-DRS and fine-tuned it for the AoI of Mariupol. ERCNN-DRS is transferred four times, receiving the trained model variants V1-4. Similar as to the previous work [13], the

³Since ERCNN-DRS is fully convolutional, changes of tile sizes are possible for transfer or inference.

transfer is done with a nonexhaustive cross-validation approach where disjunct validation data are used. All four model variants were used for monitoring the urban changes in 2022/23 for the AoI of Mariupol. In this work, the performance of each variant, as well as the bagging ensemble of all four variants, is analyzed.

Since no public VHR data were available during the time frame 2022/23,⁴ we applied the transfer learning to the period of the beginning of 2017 until the end of 2020. The practical feasibility due to easy labeling with Google Earth historic VHR imagery has already been demonstrated in the aforementioned work. However, in this work, we analyzed the quality of predictions when the transferred model was applied to years outside the transfer period, in order to be able to monitor the urban changes during the recent years when public data was not yet available.

For the verification of changes in that time frame, we used commercial Airbus Pléiades VHR observations from the beginning of 2022 to early 2023. Opposed to Maxar WorldView, which was under embargo for Ukraine at the time of writing, Airbus Pléiades data were still commercially available. However, this restricted the best resolution for verification to 0.5 m/pixel (panchromatic). These observations were used identically to the earlier ground truth generation ($\tilde{y}_{i,j}$) for the transfer but with labeling the changes in the monitoring time frame ($\tilde{y}_{i,j}$). A final verification step compared the model predictions against the manually identified changes. This comparison is covered in Section V, which gives a quantitative and qualitative analysis.

The monitoring past the transfer time frame imposed additional challenges due to changing observation patterns over time. This was especially impacted by the failure of Sentinel 1B on December 23, 2021.⁵ The unavailability of Sentinel 1B led to a significant reduction of SAR observations with only Sentinel 1A left in operation. Both Sentinel 1 satellites were placed on the same orbit plane with a difference of 180° in orbit phase. As a result, the cycle time for Europe increased from six days to twelve days after the malfunction. Since the transferred models were trained with both Sentinel 1A and 1B satellite observations, this change could have had a significant impact on the operation of the transferred models. As we will show, our method did not break down by the change of observation patterns due to the reduction of SAR observations as a result of the loss of Sentinel 1B. We studied the resiliency and scalability of changes in observation frequency for each mode and the combination of modes with a simulated SAR and optical data loss in an ablation study (see Section V).

Overall, following three different observation data sources were used:

- 1) *Sentinel Hub* for Sentinel 1 and 2 data;
- 2) Google Earth historic VHR imagery; and
- 3) *Sentinel Hub* for Airbus Pléiades VHR observations.

The Sentinel 1 and 2 data were the primary data used for the transfer of the four model variants and their inference (monitoring). Their processing is described in detail in Section III-A.

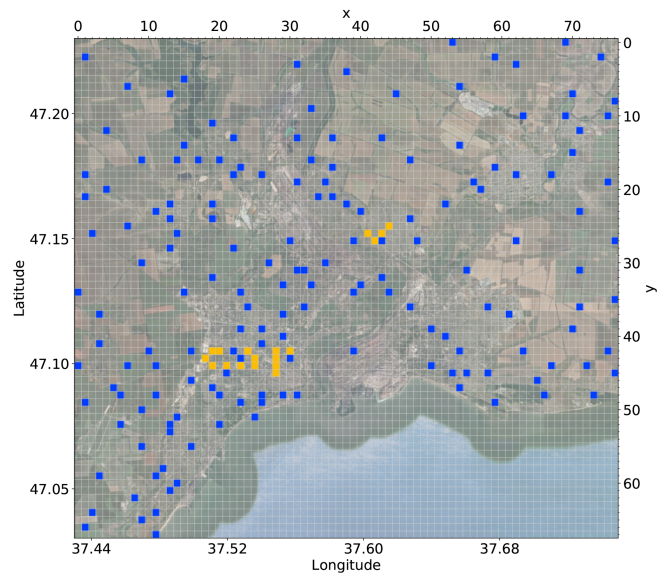


Fig. 3. Tiles for the AoI of Mariupol. The blue tiles covering 2017–2020 were used for training and validation, referred to as *trainval* dataset (164 in total). The 18 tiles in orange for 2022/23 were used for verification purposes, referred to as *testing* dataset. Geographic coordinates are in EPSG:4326 and tile coordinates are in (y, x) dimensions. Background image ©2019/20 Google Earth, for reference only.

The other two data sources were used for generating ground truth maps for transfer and verification, and are covered in Sections III-B and III-C, respectively.

III. STUDY AREA AND REMOTE SENSING DATA

We applied our methods to the area of Mariupol (Ukraine) to monitor urban-related changes and activities with the Russian invasion that began on 24 February 2022. During the first months of the siege, approximately up to 95% of the city and its infrastructure were damaged or destroyed.⁶ Up to the writing of this work, Mariupol was still under Russian occupation, and heavy reconstruction of buildings and infrastructure was observed. We monitored not only the city of Mariupol but also the surrounding area with over 536 km², covering suburbs, rural areas, the sea, mines, and farming regions. The area is also subject to frequent overcast due to its location by the Black Sea and winters with snow and ice.

Fig. 3 shows the tiled AoI, with training tiles for the transfer and verification tiles for the monitoring phase. To avoid spatial bias we used only disjunct sets of transfer and verification tiles, even though both covered different time frames. The processing of the three different data sources and types used in this work is described in the following.

A. Primary Data


The core data of our method comprised multimodal SAR and optical multispectral observations from Sentinel 1 and 2,

⁴Google Earth historic imagery ended early 2021 at the time of writing.

⁵Accessed: 16 Oct. 2023. [Online]. Available: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-1/Mission_ends_for_Copernicus_Sentinel-1B_satellite

⁶Accessed: 16 Oct. 2023. [Online]. Available: <https://web.archive.org/web/20221031202808/https://www.abc.net.au/news/2022-05-26/damage-data-reveals-extent-of-vicious-russian-tactics/101070918>

TABLE II
USED AOIS, THE COVERED AREAS, AND THE NUMBER OF THEIR AVAILABLE OBSERVATIONS, WITH REMOVED ONES IN PARENTHESES

	Site	SAR observations (ascending and descending)	Optical multispectral observations	Area (km ²)	Stage
Sentinel 1 and 2	Rotterdam	1,603 (−4)	278 (−10)	523.6	} pre training
	Limassol	468 (−0)	407 (−35)	576.2	
	Mariupol	648 (−0)	431 (−131)	536.2	} transfer
2022/23	Mariupol	155 (−0)	232 (−84)	536.2	} monitoring
Sources	SAR:	10 m/pixel, Sentinel 1, <i>SENTINEL1_IW_ASC/DSC</i>			} 
	Optical:	10 m/pixel, Sentinel 2, <i>LIC</i>			

Only level 1 products were used. Aois in gray were used for pretraining only, which is outside the scope of this work.

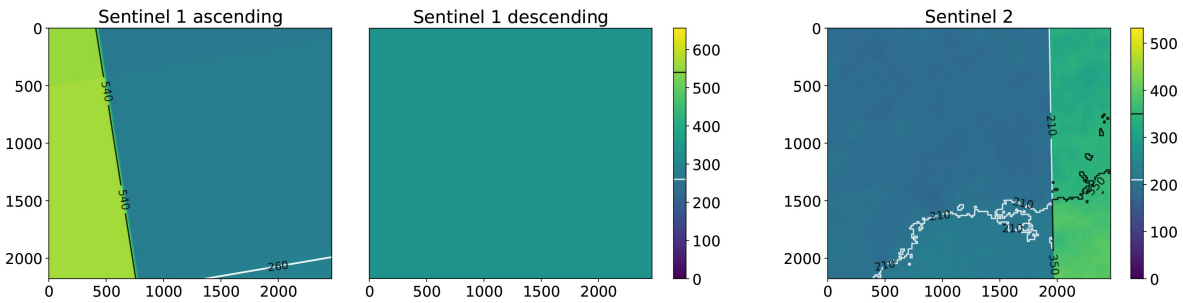


Fig. 4. Number of observations for each pixel within the AoI of Mariupol, separate for Sentinel 1 in ascending and descending orbit direction (left), and Sentinel 2 (right). Sentinel 1 has a range of [220, 546] and [324, 324] (no variation) for ascending and descending orbit directions, respectively. Sentinel 2 observations are within [166, 383]. Contours are shown for selected observation numbers.

respectively. They were used as input to the selected DNN and span the time frames of 2017–2020 for transfer, as well as the time frame November 24, 2021, up to mid-2023 for the actual monitoring phase. All data were retrieved from *Sentinel Hub* as *Level 1* products. Table II summarizes the primary data products used.

The *Level 1* products provided by *Sentinel Hub* were already orthorectified and coregistered. In our method, all available bands and polarizations were used, i.e., no dimensionality reduction was applied; that is both polarizations (VV+VH) for Sentinel 1 observations and 13 spectral bands for Sentinel 2 observations were available. The preprocessing followed the same way as used for the pretraining of ERCNN-DRS and is executed with the *rsdtlib* library as described earlier.

The available Sentinel observations for the period from 2017 until mid-2023 vary over time and by location within the selected AoI. Fig. 4 shows the available observations for each pixel within the AoI for $\delta = 1$ s. For Sentinel 1 in ascending orbit direction, different and overlapping swaths were visible. These result in more observations where swaths overlapped, and less where the surface was scanned less frequently or irregularly. Sentinel 1 in descending orbit direction did not show any variance due to full coverage of the AoI by the swaths. For Sentinel 2, in addition to swath patterns, cloud masking (as provided by *Sentinel Hub*) added to the irregularity of the observations. Also the coast of the Black Sea became visible, which was a result of the applied cloud detection method to overestimate clouds over land surfaces.

In Fig. 5, the available observations per each six-month window (Δ) are plotted, using $\delta = 2$ days. The loss of Sentinel 1B and the decrease of available SAR observations is clearly visible. It should be noted that due to the windowed time series used, the sudden drop of Sentinel 1B observations led to a gradual reduction of the observations over a half-year period prior to the day of malfunction.

B. Ground Truth for Transfer Period

To generate the labels $\tilde{y}_{i,j}$, as used for the transfer learning process, we utilized VHR historic satellite and aerial imagery from Google Earth. These were used for a selection of 164 random tiles (referred to as *trainval* dataset) in the AoI of Mariupol to approximately identify past urban changes in the time frame 2017–2020. Using a longer time frame, such as four years, simplifies the labeling process due to increased chances of more historic VHR observations being available. A higher number aids in identifying and localizing changes over time. Depending on the exact location within the selected AoI, between 10–20 historic observations were available.

Tiles that contained changes that were too large and homogeneous (e.g., open pit mines, destruction of large storage buildings, and steel factories), were removed during the random selection process. This avoided an undesired bias toward specific change patterns and instead balanced the transferred network toward a more diverse set of changes. In addition, this simplified the labeling process since areas that have constantly

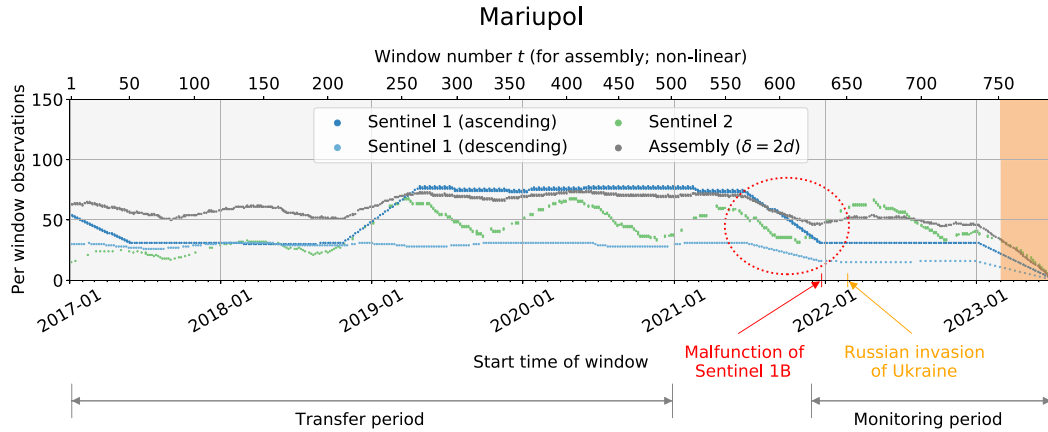


Fig. 5. Windows and their observations. In our work, we combined Sentinel 1 (blue) and 2 (green) observations in a two-day ($\delta = 2$ days) interval (gray). Highlighted in orange and discarded were windows with less than 35 (ω) observations. A malfunction of Sentinel 1B occurred on 23 December 2021 and the Russian invasion on 24 February 2022, as indicated on the timeline.

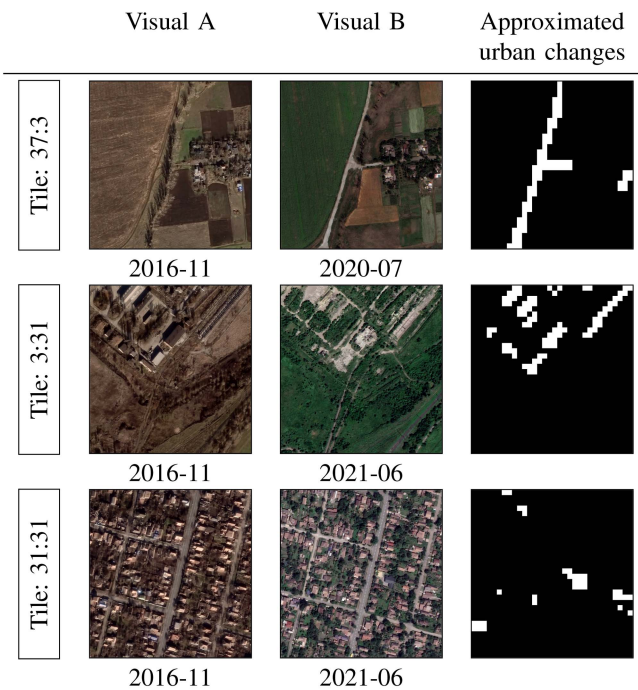


Fig. 6. At least two historic Google Earth VHR images near the beginning of 2017 (left) and the end of 2020 (middle) were used to approximate the ground truth (32×32 pixel) for urban changes $\tilde{y}_{i,j}$ (right).

been a subject of change were hard to label. Rapid and spatially constraint changes were easier to notice and mark.

The created ground truth $\tilde{y}_{i,j}$ is binary with an assigned value of 1.0 if a change was related to man-made UST objects, indicated by visual inspection of at least two VHR images. The value of 0.0 was assigned otherwise. Also, side effects of constructions were treated as changes, such as modified soil around construction sites or paved roads. Any change below the sensor resolution (< 10 m/pixel) was ignored. Fig. 6 shows three tiles as examples with the closest observations at the beginning of 2017 and end of 2020. The manually created binary ground truth is shown next to the visual samples. Due to the low amount

of available VHR observations, it is not possible to label the full extent of the changes. Hence, we refer to the labels as approximate. The three examples show the construction of a road (tile 37:3), destruction of factory buildings (tile 3:31), and (re)construction of buildings in a suburban area (tile 31:31).

C. Ground Truth for Monitoring Period

To verify the urban changes during the years 2022/23, recent Airbus Pléiades data were leveraged. Since these data were involved with significant costs, we selected 18 tiles (referred to as *testing* dataset) in two locations: 1) The city center (14 tiles) and 2) a suburb to the north (4 tiles). Pléiades data were used in panchromatic mode, resulting in 0.5 m/pixel resolution. This was sufficient to identify changes and evaluate the predictions. Depending on the location, six to seven observations at different times from March 2022 until January 2023 were screened.

The generation of the verification reference for the monitoring period $\tilde{y}_{i,j}$ followed the same rule as with the ground truth $\tilde{y}_{i,j}$. Both were binary, with values decided upon visual inspection of the available observations. However, $\tilde{y}_{i,j}$ only spans the monitoring period of 2022/23.

IV. TRAINING

The transfer phase was executed on the Karolina GPU cluster⁷ at IT4Innovations. One compute node with eight NVIDIA A100 GPUs, each with 40 GB of memory, was used for training. The training environment comprised Tensorflow 2.7, including Keras, and Horovod [17] 0.23.0 for leveraging multiple GPUs. We used synchronous SGD [18], [19] with a momentum of 0.8 and set the learning rate α to 0.008. The loss function \mathcal{L} was identical to the pretraining, that is the *Tanimoto loss with complement* [20]. It compared the maximum pooled prediction $\mathbf{y}_{i,j}^{\max}$ against the ground truth $\tilde{y}_{i,j}$, expressed as $\mathcal{L}(\tilde{y}_{i,j}, \mathbf{y}_{i,j}^{\max})$. For every tile, pixels at the border (dead area) were ignored,

⁷Accessed 16 Oct. 2023. [Online]. Available: <https://docs.it4i.cz/karolina/hardware-overview/>

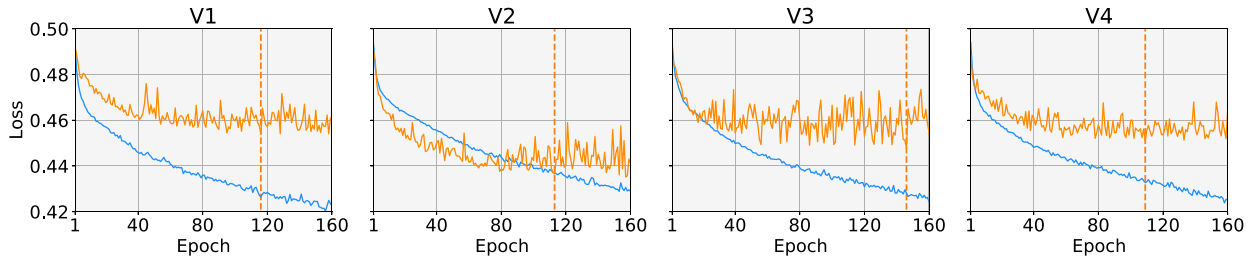


Fig. 7. Loss values over epochs for all four transferred variants. The transfer training losses are represented by the blue curve, and the validation losses by the orange curve. The orange dashed lines show the best epochs based on validation data.

and only the center 30×30 pixels of each tile were considered. This is due to the general problem of tiled data and convolutional networks, which increase errors toward the borders [21], [22], [23], [24]. We did not mitigate this in our current work, but an overlapping of tiles can easily be applied to remove these errors. Empirically, we found that removing only the direct border pixels is sufficient. Removing a larger border would result in reduction of label data. Hence, the use of only the center 30×30 pixels was a compromise between the avoidance of a higher loss of label data and increased computational and storage needs with the use of overlapping tiles. We, however, applied inference with 8 pixel overlapping tiles of size 93×93 for the final monitoring to ensure a complete coverage of the entire AoI and more efficient inference due to larger tiles (the result is shown in Fig. 23 in Appendix D).

For the transfer phase, the pretrained model was used, with no applied layer freezing. The shallow structure of the pretrained model architecture did not develop patterns found in deeper networks. In deeper networks, more general features are extracted in layers closer to the input and more specialized features toward the output [25]. Freezing layers closer to the input is a common practice for transfer learning so that only more specialized layers are transferred and the general ones are only reused. This results in lower resource needs and faster training. However, for our shallow network architecture such generalization and specialization patterns are less likely to develop. Instead, in our case we benefited from transfer learning by using a more specific definition of urban changes with coarse temporal information. Since the pretrained model was trained with more temporal information, i.e., one label per window, transfer learning can build on top of this.

Furthermore, a batch size of eight for each GPU was used, totalling an effective batch size of 64 (8×8 GPUs). Due to limited memory, for every tile (training sample) only ten partially overlapping windows were randomly selected to span the (almost) entire four years of 2017–2020. This follows the previously proposed approach to avoid bias of the network under training to specific windows and their observation patterns. In this work, the first window started at $t = 21$ and the following nine windows were selected based on uniform random relative offsets within the range of $[40, 49]$. The initial offset was needed; as with the temporal stacking not every pixel had a value at $t \in [1, 20]$. This originated from the cloud masks and out-of-swath where no previous value was available that leaves a zeroed

gap. Since we started the training time frame at the beginning of 2017, during winter, the window with $t = 21$ started in March 2017 (see Fig. 5). This was acceptable as urban changes are less likely during winter. Conversely, the last window does not always end with 2020. With the selected range, the time frame covered reaches into the second half of 2020. Again, the likeliness of changes is lower toward the end of the year in autumn or winter, which we considered a compromise between providing a variability of windows and full coverage. Since the ground truth was approximate and we focused mostly on changes within the four years, rather than changes toward the beginning or end, this was an acceptable tradeoff. The overlapping of windows ensured that all observations were visible by the network under training, except for observations at the beginning of 2017 and toward the end of 2020. Ultimately, the variance of selected windows adds augmentation to the training samples, which reduces variance and improves generalization of the trained model.

Further augmentation was carried out on each sample with random horizontal flipping (factor two), rotation in 90° increments (factor four), and binary application of a *temporal comb filter* [13] (factor two). Altogether, these augmentations increased the dataset size *trainval* by a factor of 16.

We would like to clarify that the chosen method for transferring was practically simple but had challenging memory requirements. The preparation of the ground truth was simplified by aggregating a set of windows and labeling changes visible over a longer time frame, such as multiple years. However, each prediction of a window needed to be maximum pooled for every training step. In turn, more windows over a longer transfer time frame needed to be concurrently trained with shared weights. In our case of using four years and ten randomly chosen windows, ca. 250 GB of memory was needed. We, hence, used multiple GPUs in a distributed data parallel training setting to increase the available memory.

Four different transfers were carried out on the same pretrained model but with different training/validation splits. We used a nonexhaustive cross-validation approach with all validation sets being disjoint. The loss curves of the transfers of the variants V1–4 are shown in Fig. 7. The best validation loss is highlighted with an orange dashed vertical line. For V1, epoch 116 showed the best validation loss. For V2, V3, and V4, the best epochs were 113, 146, and 109, respectively. We, hence, define the per-tile predictions of the four individual transfer models

TABLE III
COMPUTING SYSTEMS USED TO VERIFY THE TRANSFER PHASE

Infrastructure	Cluster	Hardware	Transfer (h)	Environment
IT4Innovations docs.it4i.cz	1 Karolina GPU node	2 AMD EPYC 7H12 (64 cores) 8 NVIDIA A100 (40 GB)	29:13h	TF 2.7, Horovod 0.23.0
LUMI lumi-supercomputer.eu	1 node of LUMI-G	1 AMD EPYC 7A53 (64 cores) 4 AMD MI250X* (128 GB)	38:30h	TF 2.10, Horovod 0.26.1
CESNET MetaCentrum metavo.metacentrum.cz	DGX H100	2 Intel Xeon 8480C (56 cores) 8 NVIDIA H100 (80 GB)	29:26h	TF 2.12, Horovod 0.27.0

* Multi chip module with two GPUs each—effectively eight GPUs.
The transfer time was for one variant up to epoch 160.

used in the monitoring phase as

$$\mathbf{y}_{i,j}^{V1} := \max \{ f_{V1}(\mathbf{w}_{i,j}^t) : \mathbf{w}_{i,j}^t \in \mathbb{W}_{i,j} \}$$

with the elementwise maximum operation over the 2-D predictions of each window. The trained parameters of V1 are used by the forward propagation $f_{V1}(\cdot)$. Similarly, the predictions $\mathbf{y}_{i,j}^{V2}$, $\mathbf{y}_{i,j}^{V3}$, and $\mathbf{y}_{i,j}^{V4}$ are defined for V2–4.

The predictions of a combination of V1–4 are defined as

$$\mathbf{y}_{i,j}^C := \sqrt[4]{\mathbf{y}_{i,j}^{V1} \cdot \mathbf{y}_{i,j}^{V2} \cdot \mathbf{y}_{i,j}^{V3} \cdot \mathbf{y}_{i,j}^{V4}}$$

with an elementwise maximum, the Hadamard multiplication and fourth root. This follows the bagging methodology to create an ensemble of weak learners. The combined predictions $\mathbf{y}_{i,j}^C$ were constructed with all windows starting in the monitoring period (unit stride $\rho = 1$). This is different to the transfer phase where only ten partially overlapping random windows were considered.

While the main development system was one node of the IT4Innovations' Karolina GPU cluster, other deep learning systems have also been used to confirm the transfer phase. Table III shows the verified systems with their hardware and software environment used. Transfer times were similar, except for the LUMI-G node. The reason was the I/O bound workload of our method. While the network only contained ca. 69 k parameters, the data samples we worked with were comparatively more complex. Since LUMI-G contained only a single CPU socket, I/O was limited compared with the other systems.

V. RESULTS

The transferred model variants V1–4, as well as their bagged ensemble, were analyzed quantitatively with commonly used metrics in the machine learning and remote sensing domains. In addition, qualitative analysis was carried out with selected examples that also demonstrated the temporal localization of changes. Finally, an ablation study, which varies the number of observations, gave further insights into the transferred model's performance and resiliency.

A. Quantitative Analysis

For quantitative analysis, we used three different metrics as follows:

- 1) Receiver operating characteristic (ROC) curve;

- 2) precision recall (PR) curve; and

- 3) Cohen's Kappa κ with varying thresholds.

ROC curves are a very common metric for binary classifiers [26]. Since we received predictions that are of continuous values and not binary, ROC is a useful choice. They do not require a certain threshold value to be defined, but apply different thresholds at once. The area under the curve (AUC) forms a metric that allows for comparison of different models.

ROC curves do not work well for unbalanced classes, which was also true for our case. The amount of no-changes were dominant to the amount of changes since most of the pixels did not change. The choice of PR curves [27] are more suitable for such skewed datasets as they can provide more insight [28]. Similarly to ROC curves, PR curves consider varying thresholds and the AUC is a well-suited metric for model comparison.

Cohen's Kappa [29] κ , provides the agreement of two raters. In this work it was used for the two classes of change and no-change. Since it does not apply varying thresholds itself and expects binary raters, it required the selection of a threshold value upfront. We studied κ with thresholds in the range of 0.0 to 1.0. Thresholds with the highest κ values would finally be used since they represent the best agreement of the prediction output and the compared ground truth.

Fig. 8 shows the ROC and PR curves, as well as κ for the *trainval* dataset used for the transfer variants. Whilst all variants show a similar performance, variant V3 scores lower for all three metrics. This likely stems from a worse representation of training samples that were randomly drawn for the nonexhaustive cross validation. Nevertheless, all variants are performing significantly better than the pretrained ERCNN-DRS (baseline). The combination of all methods showed the highest scores in all three metrics. Noticeable is also the higher κ for larger thresholds for the variants and their combination.

In comparison, Fig. 9 shows the metrics for the *testing*⁻ dataset. This dataset is the *testing* dataset with tile 43:18 removed. Since the *testing* dataset was characterized by only 18 different tiles (samples), outliers caused by tiles with larger changes skewed the results. In particular, tile 43:18 (see Fig. 11) significantly increased the AUCs due to large and intense construction activities. We, therefore, decided to remove it for our analysis and analyze smaller and heterogeneous changes. The curves for the full *testing* dataset can be found in Fig. 19 in Appendix B. For the *testing*⁻ dataset, all variants produce similar

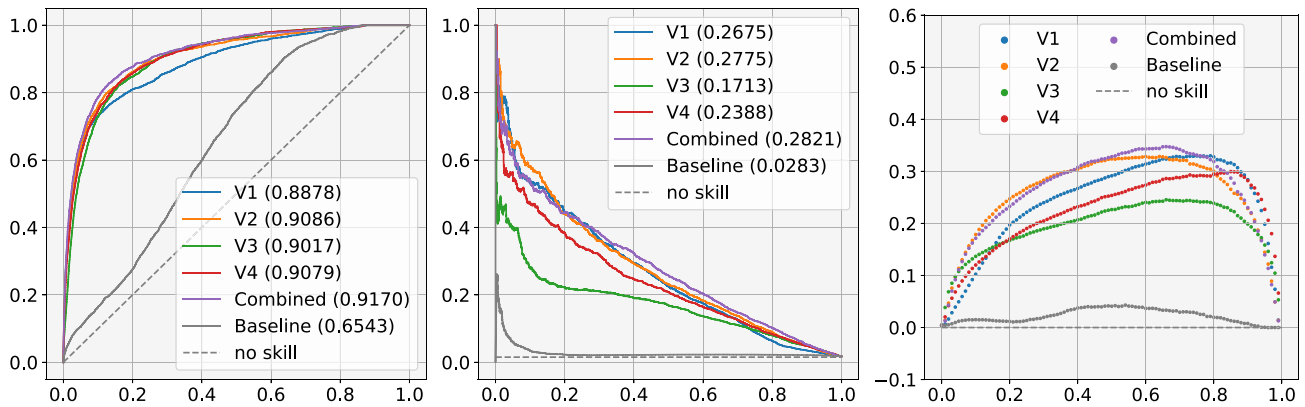


Fig. 8. For the *trainval* set: ROC (left) and PR (middle) curves; Cohen's Kappa is shown for different thresholds (right). Area under the ROC/PR curves are in parenthesis.

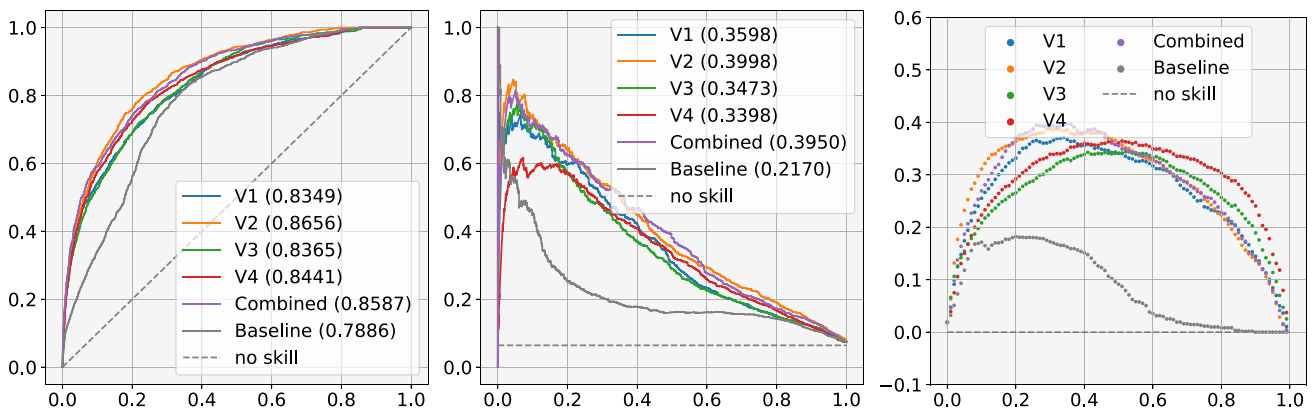


Fig. 9. For the *testing⁻* dataset set: ROC (left) and PR (middle) curves; Cohen's Kappa is shown for different thresholds (right). Area under the ROC/PR curves are in parenthesis.

results, with V4 showing a slightly different behavior for the PR curves and κ . The ERCNN-DRS baseline scores better here but still lacks behind its transferred variants. Their combinations do not score highest in the ROC and PR curves, but are close to the best variant (V2). The combined variants, however, score highest for the κ . What is noticeable here are the different thresholds that result in the highest κ between the *trainval* and *testing⁻* datasets. The former suggests thresholds of over 0.7, whereas the latter produces higher κ scores around 0.3. Nevertheless, the κ values are around 0.3 for thresholds of 0.7 in both cases, suggesting that outliers caused higher κ values for lower thresholds in the much smaller *testing⁻* dataset. After all, the time frame 2022/23 was subject to large and highly frequent urban changes with a deviation from the regular urban development in 2017–2020.

B. Qualitative Analysis

We selected four different tiles from the monitoring period that showed representative and diverse change patterns. The quality of the predictions were analyzed. Figs. 10 and 11 show two tiles that detected different urban changes and activities. For all four examples, we also depict six VHR Airbus Pléiades and Sentinel 2 (true color) observations over 2022/23. The

VHR observations were used for creating the ground truth for verification and only the latter were used for the predictions. The differences in spatial resolution are clearly noticeable. As a result, the only changes that were detectable by our method are larger than the sensor resolution of 10 m/pixel. The time stamp on the top corresponds to the Airbus Pléiades observation with the closest Sentinel 2 counterparts ± 3 days apart. For each tile, one pixel of four different change regimes was selected to show how their prediction values changed over time. We selected the regions to provide a diverse and balanced set of change patterns with clear indications given by the VHR data. Tracing the predictions over time enabled the localization of when changes happened. This also shows how the transferred model variants identified changes differently, based on different learned patterns (i.e., *trainval* splits). Nevertheless, the detected changes were coherent and directly attributable to change events.

Tile 26:43 in Fig. 10 i) shows smaller changes and a large one (bottom center). Whereas the smaller changes were harder to analyze—see charts a) and b)—the larger one shows the destruction of a large building. Depending on the selected pixels, changes took place at different times. This building was severely damaged between March and June 2022 according to the VHR Airbus Pléiades observations. The effects of this damage were

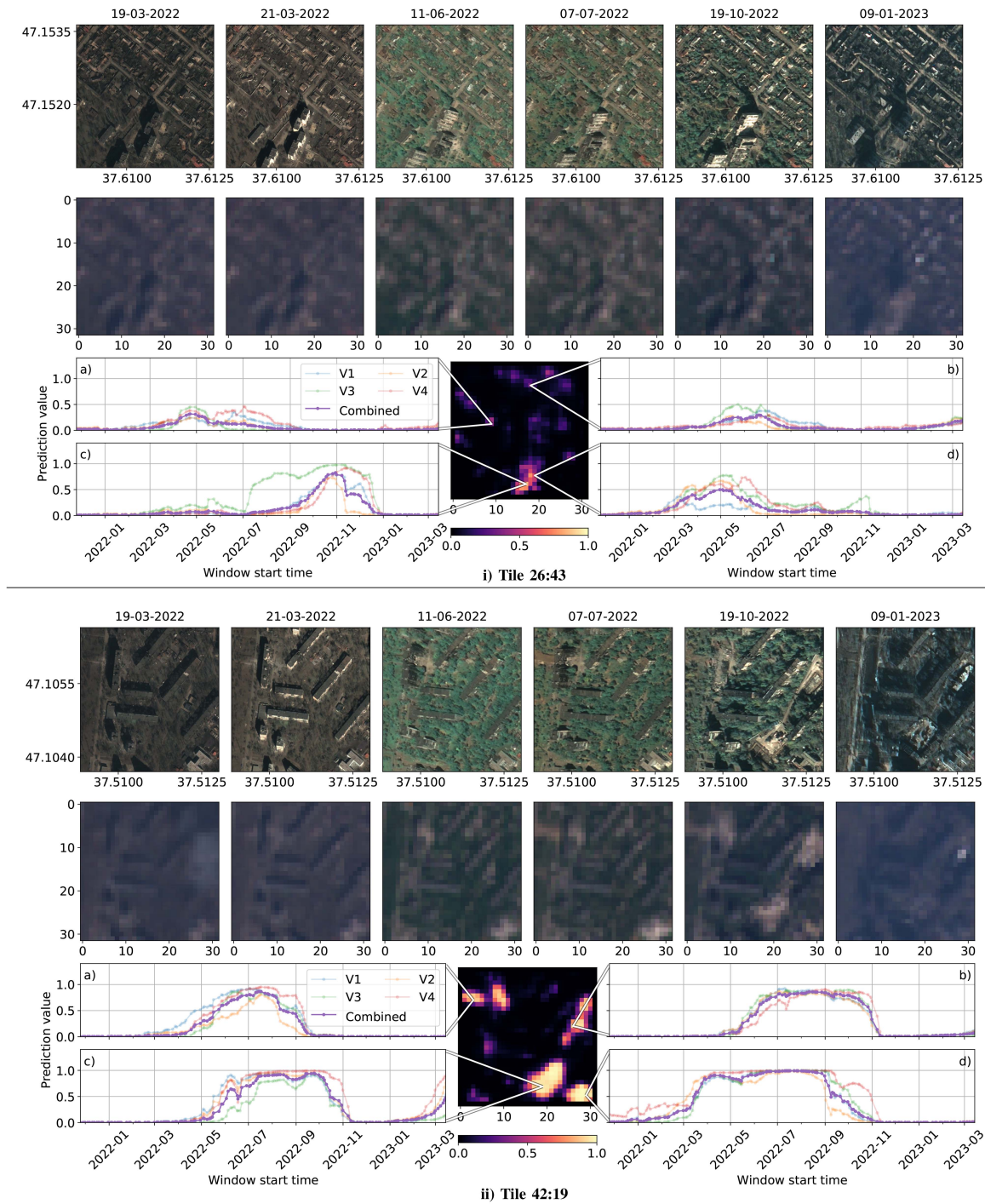


Fig. 10. Two examples for the verification of changes 2022/23 with a limited number of Airbus Pléiades observations (top rows). Second rows show Sentinel 2 true color data at similar observation times of the Pléiades counterparts (± 3 days). Bottom rows show the prediction $y_{i,j}^C$ with prediction value time series of four selected pixels.

also detected by the transferred models with a pixel selected in front of the building as shown in d). The building itself was torn down between the middle of October 2022 and early January 2023. Again, this was detected by the model variants, with V3 being overly sensitive to these changes in charts c).

The tile 42:19 in Fig. 10 ii) shows many large changes. Construction of building in b) and c) took place in the second half

of 2022. The selected pixel d) shows a reconstruction of a large building in the middle of 2022. In a), the nearby building was torn down in the middle of 2022. The selected pixel belonged to an area that was temporarily used for the destruction process.

The example i) in Fig. 11 addressing the tile 42:30 shows other urban-related changes with different patterns. In a) and b) the same building is monitored where the roof was damaged and

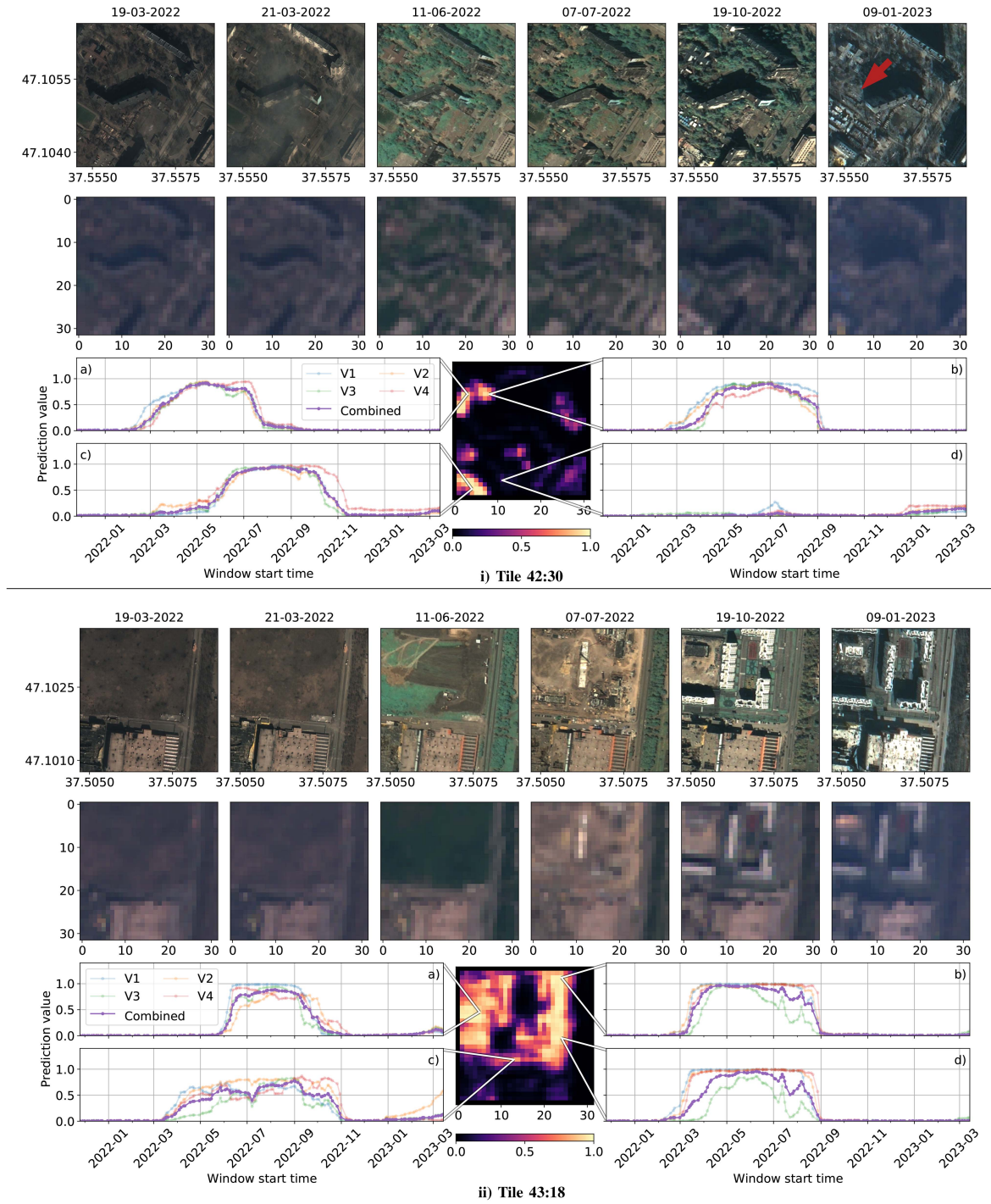


Fig. 11. Two examples for the verification of changes 2022/23 with a limited number of Airbus Pléiades observations (top rows). Second rows show Sentinel 2 true color data at similar observation times of the Pléiades counterparts (± 3 days). Bottom rows show the prediction $y_{i,j}^C$ with prediction value time series of four selected pixels.

reconstructed throughout 2022. A larger building was erected of which a quarter is visible in the lower left that led to changes in the second half of 2022, as shown in c). This scene also contains the destruction of two larger buildings. The one shown in d) only results in low prediction values; the other one is not detected at all (see the red arrow at rightmost Airbus Pléiades observation).

The last example ii) in Fig. 11 shows the excluded tile 43:18 from the *testing*⁻ dataset. This contains an exceptionally large set of changes with the construction of a building complex. These constructions happened swiftly after the Russian occupation of Mariupol at the end of May 2022. The charts a), b), and d) show concurrent constructions of different buildings of the complex.

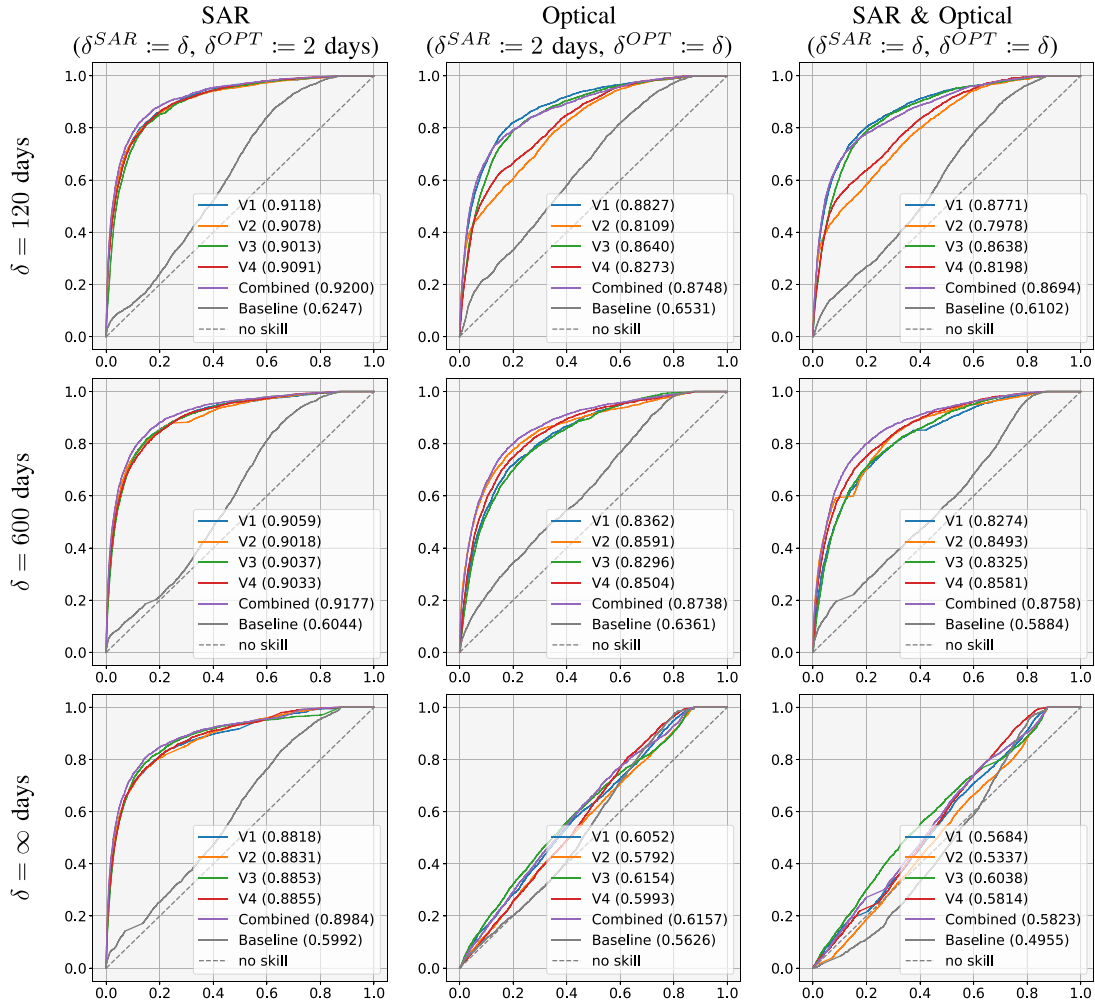


Fig. 12. ROC curves of the different models applied on *trainval* with different sampling steps.

The chart c) shows a supply corridor, which was later converted into a road.

C. Ablation Study

The unavailability of Sentinel 1B gave rise to the question as to how resilient our methods were to a change in the number of observations in relation to the overall prediction performance. In an ablation study, we selectively reduced the number of observations by increasing the step size δ . This effectively samples fewer observations and simulates scenarios where fewer real observations are available due to atmospheric disturbance or mission outages. We applied this approach separately for each mode to also analyze the impact of the modes to the predictions. The ablation study was carried out with both the *trainval* and the *testing⁻* dataset and, hence, to two different time frames. To avoid an influence by a shrunken number of observations per window (i.e., to fall below ω), we retained the same number of observations but only updated them at a step size interval. For example, moving from the default $\delta = 2$ days to $\delta = 120$ days resulted in windows with the same number of observations as for $\delta = 2$ days, but until 120 days had been reached, the observation values remained unchanged.

For the *trainval* dataset, which spans the years starting from early 2017 until the end of 2020, δ was changed from the default of 2 days to 120, 600, and ∞ days. The latter only contained one invariant observational state as no update has been done (infinite sampling step). Figs. 12, 13, and 14 show the resulting ROC, PR curves, as well as the κ for different thresholds, respectively.

A similar ablation study was done for the *testing⁻* dataset, covering the 2022/23 monitoring period. Due to the shorter time frame, only 2, 120, and ∞ days were evaluated as sampling step sizes. The ROC, PR, and κ curves are shown in Figs. 15, 16, and 17 accordingly.

The analyses showed three different effects. First, optical observations had a larger impact on the prediction performance. On the other hand, SAR observations tended to increase the threshold for which the highest κ value was received. This was due to a larger spread of values between changes and no-changes. Hence, SAR observations added more confidence on changes, but they added less in terms of identifying the right changes.

Second, with less or no optical changes, the predictions broke down quicker, which suggested that SAR observations could not compensate. This might have been caused by the lack of differentiation of change of materials.

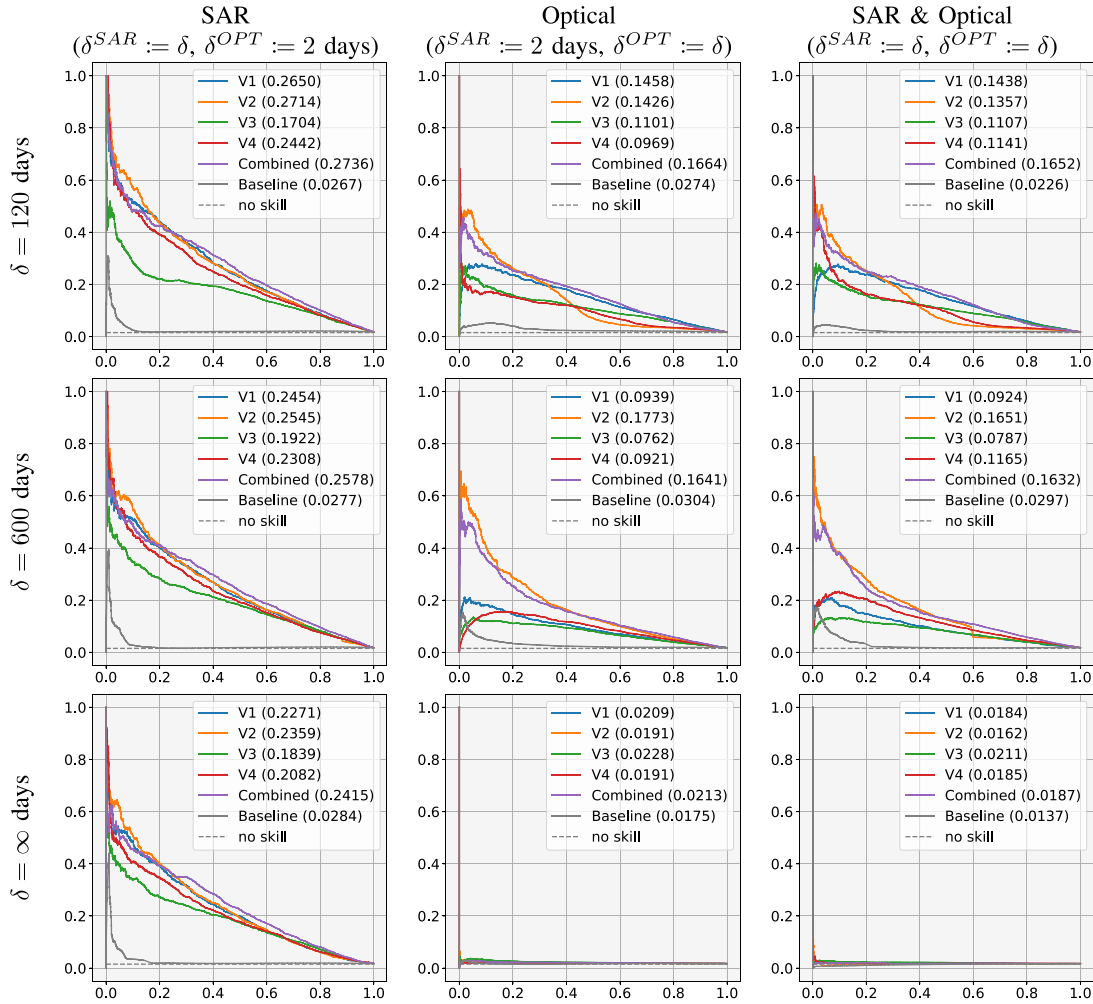


Fig. 13. PR curves of the different models applied on *trainval* with different sampling steps.

Third, the reduction of observations and the drop in performance was not linear. Even with few observations, predictions were possible. This was a result of the binary ground truth, which did not have or require information on the frequency or intensity of changes. It was possible for the model to still classify changes with a significantly reduced set of observations. However, fewer observations reduced the temporal resolution and had an impact on localizing a change over time.

The ablation results with the *testing⁻* set showed differences to the ones from *trainval*. With reaching $\delta = \infty$ for optical observations, the scores of the ROC and PR curves were not as close to a no-skill model as they were for *trainval*. Similarly, the best threshold for the κ values moved toward zero yet were not as low. Also, the baseline performed noticeably better for the *testing⁻* dataset. These effects were related to the smaller dataset size and the comparably larger changes that took multiple months. The pretrained ERCNN-DRS worked well for cases with spatio-temporal large scale changes but underestimated shorter and smaller changes (cf. [13]). Nevertheless, the transferred variants and their combination still provide the best performance for the 2022/23 monitoring period. The ablation study results for the full *testing* dataset can be found in Figs. 20, 21, and 22 in Appendix C.

VI. LIMITATIONS

In this study, we observed the following three limitations. First, building damages were only recognized if multiple pixels were involved and there was a significant change in the surrounding area (e.g., piles of debris and cleanup efforts). With the used resolution of, at best, 10 m/pixel, this translates to approximately an area of 30 m \times 30 m and upward.

Furthermore, changes were only detected if learned during the transfer phase. Since the changes in 2022/23 were of a different origin, we might not have been able to detect all changes, such as the ones shown in the Example i) in Fig. 11 (tile 42:30). Similarly, the selection of samples for the transfer had a direct impact on the detected change patterns. Changes of similar types and patterns biased the trained models and reduced sensitivity to other change types and patterns. This is why large and homogeneous changes were removed from the random tile selection process when creating the *trainval* dataset. Ideally, a better balance of changes and no-changes would help. This, however, is constrained by the available real changes in the transfer period. Therefore, our method works best for cases where urban changes constantly happen.

Lastly, changes were temporally localizable but not down to one month. The reason is the use of six-month windows

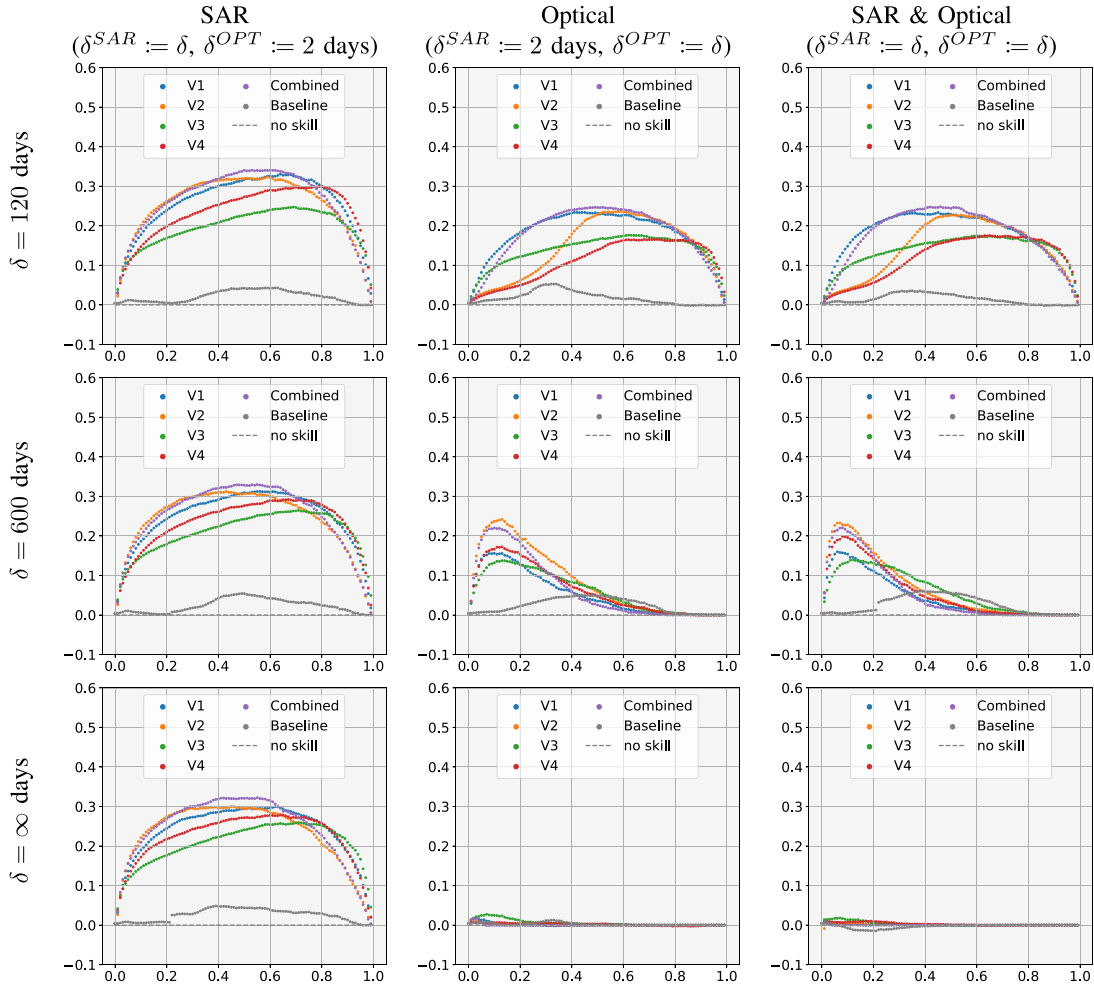


Fig. 14. Cohen's Kappa of the different models applied on *trainval* with different sampling steps and a changing threshold.

and the unclear response to changes over time. In our previous study [12], we addressed this partially. However, more work is needed to help narrow down changes to the granularity of one month or even less.

VII. DISCUSSION

To give more insight into the selected methods and obtained results we elaborate on five different aspects. These are the right choice of a threshold value, how changes are detectable, the effects of lowering the sampling rate, the amount of transfer samples in relation to overfitting, and the size of our network, which requires a large dropout rate. With the discussion topics we also aim to help with adopting our methods for other areas and scenarios.

A. Choosing a Threshold

The quantitative analysis in Section V-A was carried out with ROC and PR curves. These do not require to select a specific threshold when comparing the binary ground truth with the network's continuous and probabilistic output. They rather describe the network's performance irrespective of a threshold value, which gives a more detailed insight. For practical use,

however, a threshold needs to be chosen at which an urban change is considered as such. The earlier description of κ with a sweeping threshold aids in selecting the best value. The κ values in Fig. 8 for the *trainval* set indicate that a threshold value in the range of [0.6, 0.9] delivers the best performance, depending on the model variant. For the *testing*⁻ dataset, the best κ values are suggested in the range of [0.3, 0.6] in Fig. 9. Due to the small number of samples in that set, these values are subject to bias. A conservative joint threshold, hence, would be closer to the value of 0.6 for the combined models.

B. Detectable Changes

The use of a window in our method provides context to the neural network to identify changes resiliently, while working with error prone Level 1 observations [30]. As a result, changes are only detected as they appear within that window. Theoretically, changes can occur that barely are detectable within this limited context. If that happens continuously, i.e., a very slow change over decades, even with a sliding window approach, such a change will likely not be clearly detectable. The window duration of six months was chosen to be a good compromise of being short for localizing changes within a year,

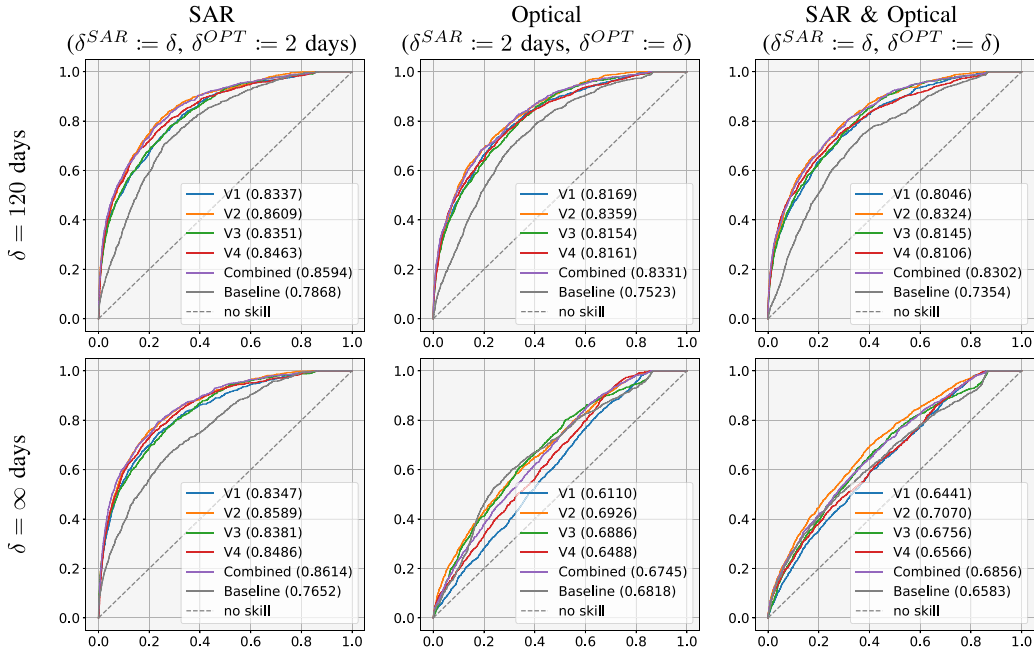


Fig. 15. ROC curves of the different models applied on $testing^-$ with different sampling steps.

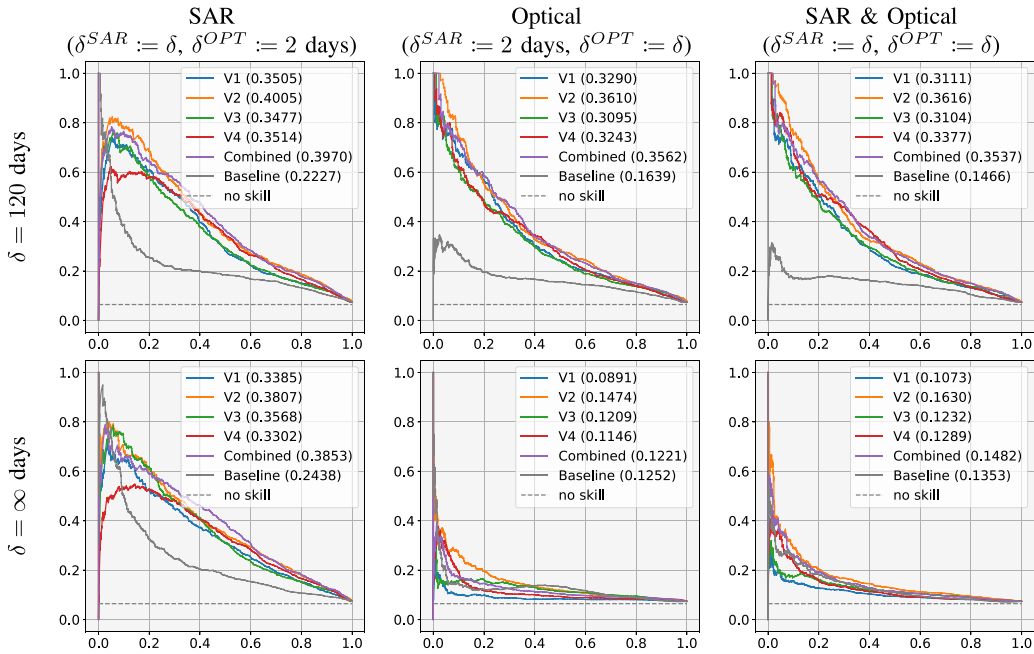


Fig. 16. PR curves of the different models applied on $testing^-$ with different sampling steps.

and long enough to detect man-made construction/destruction activities without a large influence of Level 1 data outliers. If very slow long term changes, such as decaying buildings over decades, shall be detectable, we suggest to consider a larger window duration, if observational data allows. This would, however, require a new pretraining of ERCNN-DRS to cater for a different window duration (Δ) and also a coarser sampling rate (δ) to reduce memory requirements.

In addition, the pretraining of ERCNN-DRS [12] relied on the enhanced normalized difference impervious surface index [31], [32] by aliasing impervious surfaces as urban. The transfer learning step, then, enables to specify more precise patterns that are intended as urban changes and are not (only) driven by impervious surface characteristics. Due to the limitations in time and space, from when and where samples are drawn, which is approximately 536 km² from Mariupol in the period

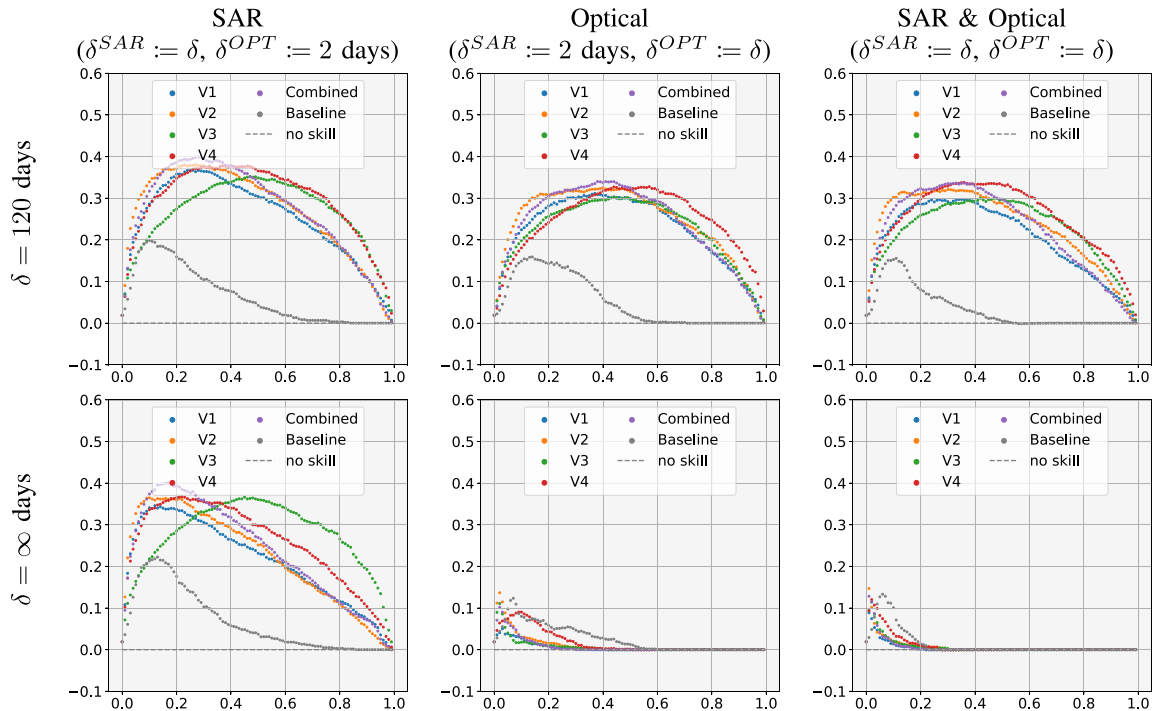


Fig. 17. Cohen's Kappa of the different models applied on $testing^-$ with different sampling steps and a changing threshold.

2017–2020 for the case at hand, the selected samples impact the overall performance. If little to no urban changes are present in that time and space, the transfer will be limited in tailoring the change patterns of interest. Similarly, the diversity of changes in the transfer period impacts the type of detectable change patterns as well. Hence, our method requires sufficient urban changes to be present before the time frame of interest for the same area⁸ for a useful transfer.

C. Changing the Sampling Rate

The original two day sampling rate served as a measure to reduce memory requirements. As overlapping swaths of nearby rows can, dependent on the selected region, result in more frequent updates than the expected repeat cycle, redundancies occur. In the scope of urban change monitoring, redundant and partially overlapping observations within a day are less likely to unveil urban changes. Hence, we decided to use a two day sampling step and merged overlapping observations within that step duration. This reduces significantly the memory footprint for each window (from over 150 observations down to less than 80; cf. Fig. 5). As demonstrated in the ablation studies, a sublinear decline in performance was observed as the sampling step has been increased. In turn, a coarser sampling can help to reduce the memory needs and still deliver acceptable performance. However, further analyses have to be done to understand if different change patterns are affected differently by a change in the sampling step.

D. Size of Transfer Set and Overfitting

The herein selected transfer data set size has been chosen to demonstrate the low labeling efforts while providing a converging transfer. We restricted the selected samples to contain a maximum of 15% of all pixels in a tile as a subject of change. This avoided an overrepresentation of similar change patterns that would bias the transfer. Our mitigation of bias was to select smaller and, thus, diverse change patterns instead. As discussed above, the selection of samples influences the transfer and so do their immanent change patterns. For example, a large area construction of similar buildings with the same materials would be overrepresented in the transfer samples and, thus, learned predominantly [33] with the tendency of memorization [34] that lowers the network's generalization.

E. Size of ERCNN-DRS and Large Dropout Rates

The ERCNN-DRS is an intentionally small network with less than 100 k parameters. It aims at identifying urban changes less by shapes but more by the spectral (optical) or polarized backscatter energy (SAR) responses with a limited receptive field. This was needed due to missing details in the medium resolution observation data and to cater for the limited memory available on GPUs. Since the ratio of the amount of training data to model size under a limited receptive field is large, overfitting happens very early, resulting in a large variance. A large dropout rate was used as a countermeasure. In earlier experiments, we considered more layers (i.e., a deeper architecture) and more convolutional filters. This, however, increased the memory needs so that the resulting network did not converge due to a too small batch size and increased gradient noise [35] or consumed too

⁸We did not yet explore the feasibility of using nearby regions as proxies.

much memory to be trainable on state-of-the-art GPUs (even if data parallelism was employed).

VIII. CONCLUSION

In this study, we demonstrated the applicability of detecting and monitoring urban changes for the AoI of Mariupol, Ukraine, by using transfer learning. It was shown that transferring for the years 2017–2020 with publicly available historic VHR data enabled monitoring during the times of war in 2022/23. During that time frame availability of VHR data was limited and a transfer for that time frame would only have been possible with significant costs. We applied four different transfer variants and their bagged ensemble to both the transfer and monitoring periods, for which the ensemble provided robust results.

We further analyzed the impact of the frequency of available observations in an ablation study. It showed that our method was resilient to even a large loss of observations. However, it also indicated that our method, despite the multimode input, is more dependent on optical observations than SAR observations. With this understanding, we can conclude that the loss of Sentinel 1B at the end of December 2021 did not significantly impact the monitoring capabilities of our method.

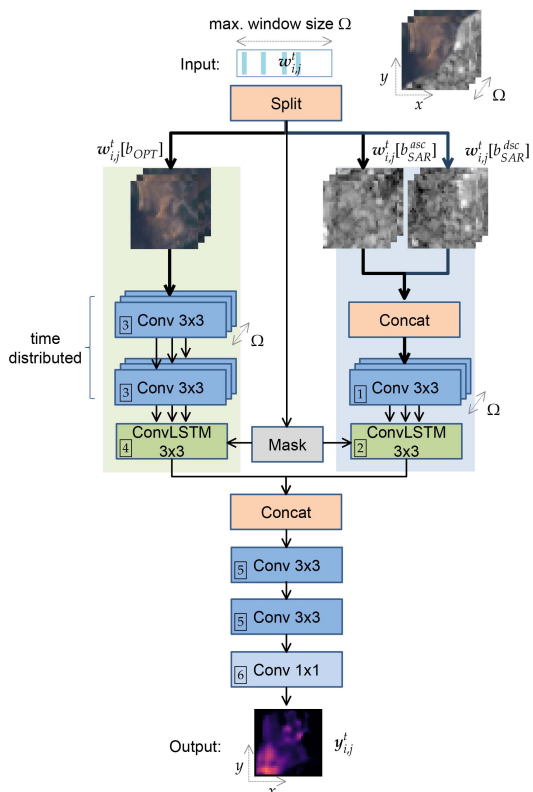
Disclosures: No potential conflict of interest was reported by the authors.

Ethical Statement: Due to the ongoing Russo-Ukrainian War, the selection of locations of visual samples was done with care to minimize risks of influence and harm. To the best of the authors' knowledge, we only selected locations and data that did not give direct insight to the ongoing war, but only documented the resulting (urban) changes. We also would like to underline that our monitoring methods used six-month windows and, hence, did not and shall not provide real-time information that could be used for military purposes. Our methods were optimized for inertial urban changes that manifest over longer periods.

Data Availability Statement: The labeled data used in this work and trained network models are available on Github https://github.com/It4innovations/urban_change_monitoring_mariupol_ua. The authors also provide collateral information such as GeoTIFF files of the prediction outputs.

Credit Authorship Contribution Statement: G. Zitzlsberger: Conceptualization, methodology, software, investigation, data curation, writing—original draft, writing—review and editing, visualization, funding acquisition, resources, supervision, project administration (90%). M. Podhoranyi: Validation, writing—review and editing (10%).

APPENDIX A ERCNN-DRS ARCHITECTURE



	Hyper-Parameters					
	Filters	Kernel	Stride	Activation(s)	Dropout	
Configurations	1	10	3×3	1×1	ReLU	
	2	10	3×3	1×1	tanh, hard sigmoid	0.4
	3	26	3×3	1×1	ReLU	
	4	26	3×3	1×1	tanh, hard sigmoid	0.4
	5	8	3×3	1×1	ReLU	
	6	1	1×1	1×1	sigmoid	

Fig. 18. Architecture as inherited from the pretraining stage. In the transfer learning stage, all layers were trained. A windowed multimodal input was expected (green background: multispectral optical; blue background: SAR in ascending and descending orbit directions). Hyperparameters of the respective layers are detailed in the table on the right.

APPENDIX B
METRICS FOR THE TESTING DATASET

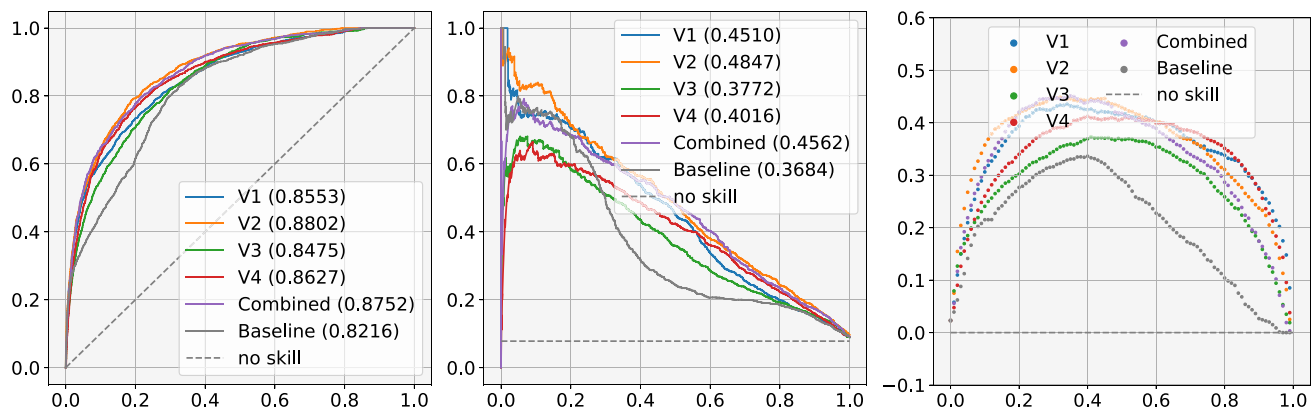


Fig. 19. For the testing dataset set: ROC (left) and PR (middle) curves; Cohen's Kappa is shown for different thresholds (right). Area under the ROC/PR curves are in parenthesis.

APPENDIX C
ABLATION METRICS FOR THE TESTING DATASET

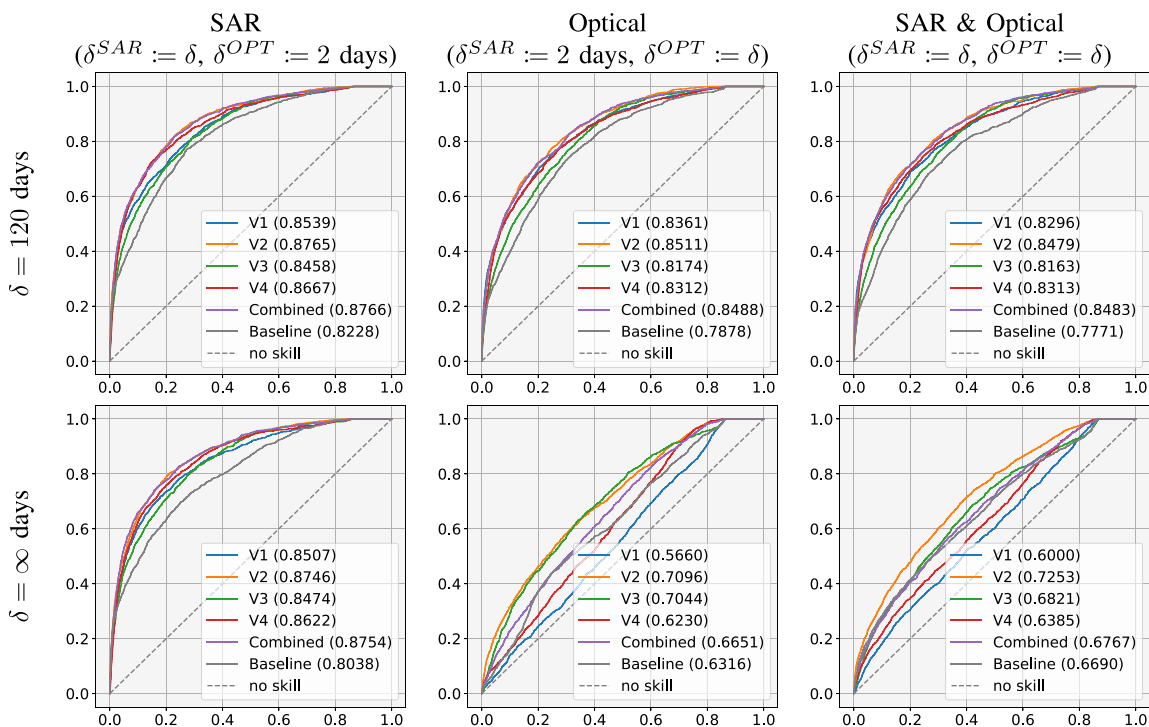


Fig. 20. ROC curves of the different models applied on testing with different sampling steps.

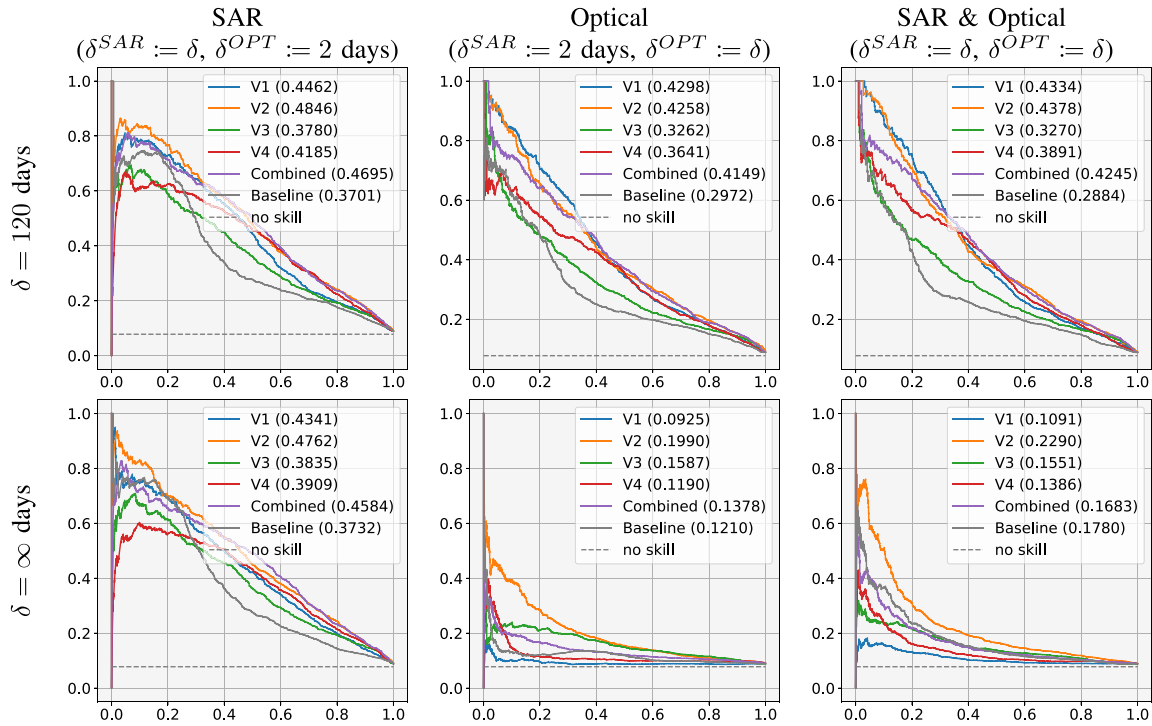


Fig. 21. PR curves of the different models applied on *testing* with different sampling steps.

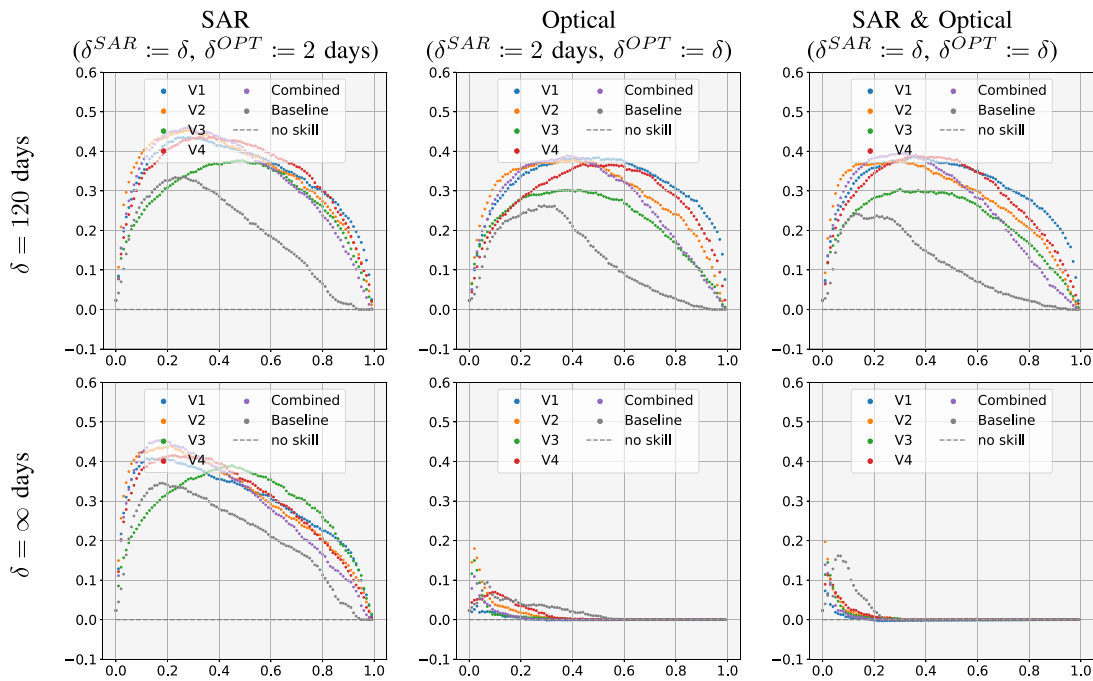


Fig. 22. Cohen's Kappa of the different models applied on *testing* with different sampling steps and a changing threshold.

APPENDIX D
URBAN CHANGES IN MARIUPOL 2022/23

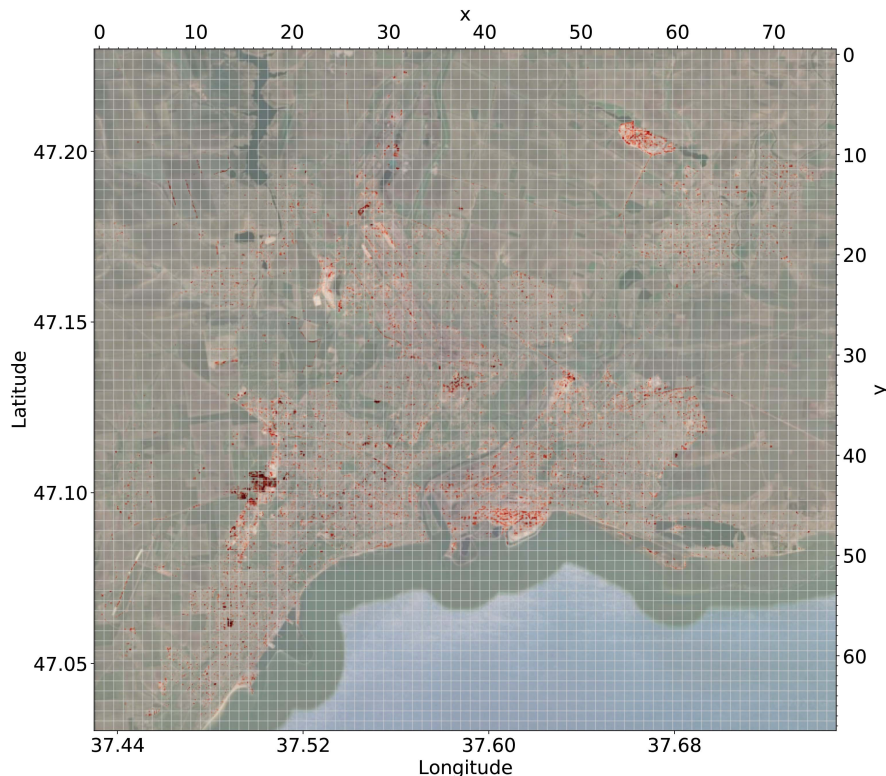


Fig. 23. Urban changes in Mariupol during the Russian invasion 2022/23 with combined models. Highlights in red show identified urban changes using $y_{i,j}^C$ for every tile. Background image ©2019/20 Google Earth, for reference only.

ACKNOWLEDGMENT

The authors would like to thank CESNET MetaCentrum for providing them access to a DGX H100 node.

REFERENCES

- [1] J. R. Shepard, "A concept of change detection," in *Proc. 30th Annu. Meeting Amer. Soc. Photogrammetry*, 1964, vol. 30, pp. 648–651.
- [2] A. Singh, "Review article digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, 1989, doi: [10.1080/01431168908903939](https://doi.org/10.1080/01431168908903939).
- [3] M. Hemati, M. Hasanlou, M. Mahdianpari, and F. Mohammadimanes, "A systematic review of landsat data for change detection applications: 50 years of monitoring the earth," *Remote Sens.*, vol. 13, no. 15, 2021, Art. no. 2869. [Online]. Available: <https://www.mdpi.com/2072-4292/13/15/2869>
- [4] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, "Change detection based on artificial intelligence: State-of-the-art and challenges," *Remote Sens.*, vol. 12, no. 10, 2020, Art. no. 1688. [Online]. Available: <https://www.mdpi.com/2072-4292/12/10/1688>
- [5] Y. You, J. Cao, and W. Zhou, "A survey of change detection methods based on remote sensing images for multi-source and multi-objective scenarios," *Remote Sens.*, vol. 12, no. 15, 2020, Art. no. 2460. [Online]. Available: <https://www.mdpi.com/2072-4292/12/15/2460>
- [6] L. Khelifi and M. Mignotte, "Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis," *IEEE Access*, vol. 8, pp. 126 385–126 400, 2020.
- [7] T. Bai et al., "Deep learning for change detection in remote sensing: A review," *Geo-Spatial Inf. Sci.*, vol. 26, no. 3, pp. 262–288, 2022, doi: [10.1080/10095020.2022.2085633](https://doi.org/10.1080/10095020.2022.2085633).
- [8] A. Shafique, G. Cao, Z. Khan, M. Asad, and M. Aslam, "Deep learning-based change detection in remote sensing images: A review," *Remote Sens.*, vol. 14, no. 4, 2022, Art. no. 871. [Online]. Available: <https://www.mdpi.com/2072-4292/14/4/871>
- [9] H. Jiang et al., "A survey on deep learning-based change detection from high-resolution remote sensing images," *Remote Sens.*, vol. 14, no. 7, 2022, Art. no. 1552. [Online]. Available: <https://www.mdpi.com/2072-4292/14/7/1552>
- [10] E. J. Parelius, "A review of deep-learning methods for change detection in multispectral remote sensing images," *Remote Sens.*, vol. 15, no. 8, 2023, Art. no. 2092. [Online]. Available: <https://www.mdpi.com/2072-4292/15/8/2092>
- [11] A. Lehner and T. Blaschke, "A generic classification scheme for urban structure types," *Remote Sens.*, vol. 11, no. 2, 2019, Art. no. 173. [Online]. Available: <https://www.mdpi.com/2072-4292/11/2/173>
- [12] G. Zitzlsberger, M. Podhorányi, V. Svatoň, M. Lazecý, and J. Martinovič, "Neural network-based urban change monitoring with deep-temporal multispectral and sar remote sensing data," *Remote Sens.*, vol. 13, no. 15, 2021, Art. no. 3000. [Online]. Available: <https://www.mdpi.com/2072-4292/13/15/3000>
- [13] G. Zitzlsberger, M. Podhoranyi, and J. Martinovic, "A practically feasible transfer learning method for deep-temporal urban change monitoring," *Int. J. Remote Sens.*, vol. 44, no. 17, pp. 5172–5206, 2023.
- [14] G. Zitzlsberger, M. Podhoranyi, and J. Martinovič, "rsdtlib: Remote sensing with deep-temporal data library," *SoftwareX*, vol. 22, 2023, Art. no. 101369. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352711023000651>
- [15] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Nov. 2011.
- [16] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

- [17] A. Sergeev and M. D. Balso, "Horovod: Fast and easy distributed deep learning in TensorFlow," 2018, *arXiv:1802.05799*.
- [18] J. Chen, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," in *Proc. Workshop Track Int. Conf. Learn. Representations*, 2016. [Online]. Available: <https://openreview.net/forum?id=D1VDZ5kMAu5JEJ1zfEWL>
- [19] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 2595–2603. [Online]. Available: <https://proceedings.neurips.cc/paper/2010/file/abea47ba24142ed16b7d8fbf2c740e0d-Paper.pdf>
- [20] F. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-A: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 162, pp. 94–114, Apr. 2020.
- [21] G. A. Reina, R. Panchumarthy, S. P. Thakur, A. Bastidas, and S. Bakas, "Systematic evaluation of image tiling adverse effects on deep learning semantic segmentation," *Front. Neurosci.*, vol. 14, 2020, Art. no. 65. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnins.2020.00065>
- [22] B. Huang, D. Reichman, L. M. Collins, K. Bradbury, and J. M. Malof, "Dense labeling of large remote sensing imagery with convolutional neural networks: A simple and faster alternative to stitching output label maps," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 6899–6902.
- [23] B. Huang, D. Reichman, L. M. Collins, K. Bradbury, and J. M. Malof, "Tiling and stitching segmentation output for remote sensing: Basic challenges and recommendations," 2019, *arXiv:1805.12219*.
- [24] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. M.-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021, doi: [10.1038/s41592-020-01008-z](https://doi.org/10.1038/s41592-020-01008-z).
- [25] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, vol. 2, pp. 3320–3328.
- [26] M. Majnik and Z. Bosnić, "RoC analysis of classifiers in machine learning: A survey," *Intell. Data Anal.*, vol. 17, no. 3, pp. 531–558, May 2013.
- [27] V. Raghavan, P. Bollmann, and G. S. Jung, "A critical investigation of recall and precision as measures of retrieval system performance," *ACM Trans. Inf. Syst.*, vol. 7, no. 3, pp. 205–229, Jul. 1989, doi: [10.1145/65943.65945](https://doi.org/10.1145/65943.65945).
- [28] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240, doi: [10.1145/1143844.1143874](https://doi.org/10.1145/1143844.1143874).
- [29] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.
- [30] R. Coluzzi, V. Imbrenda, M. Lanfredi, and T. Simoniello, "A first assessment of the sentinel-2 level 1-C cloud mask product to support informed surface analyses," *Remote Sens. Environ.*, vol. 217, pp. 426–443, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425718303742>
- [31] J. Chen, K. Yang, S. Chen, C. Yang, S. Zhang, and L. He, "Enhanced normalized difference index for impervious surface area estimation at the plateau basin scale," *J. Appl. Remote Sens.*, vol. 13, no. 1, pp. 1–19, 2019, doi: [10.1117/1.JRS.13.016502](https://doi.org/10.1117/1.JRS.13.016502).
- [32] J. Chen, S. Chen, C. Yang, L. He, M. Hou, and T. Shi, "A comparative study of impervious surface extraction using sentinel-2 imagery," *Eur. J. Remote Sens.*, vol. 53, no. 1, pp. 274–292, 2020, doi: [10.1080/22797254.2020.1820383](https://doi.org/10.1080/22797254.2020.1820383).
- [33] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 761–769.
- [34] D. Arpit et al., "A closer look at memorization in deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, D. Precup and Y. W. Teh, Eds., Aug. 6–11, 2017, vol. 70, pp. 233–242. [Online]. Available: <https://proceedings.mlr.press/v70/arpit17a.html>
- [35] S. L. Smith, P.-J. Kindermans, and Q. V. Le, "Don't decay the learning rate, increase the batch size," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=BIYy1BxCZ>



Georg Zitzlsberger received the Ph.D. degree in computational sciences from the VSB - Technical University of Ostrava, Ostrava, Czech Republic, within the university study program with IT4Innovations, the National Supercomputing Center, in 2023.

He is currently a Research Specialist with IT4Innovations and covers the interdisciplinary fields of machine/deep learning, remote sensing, and high-performance computing. His current research focuses on the challenge of detecting and monitoring urban

changes with remote sensing data.

Michal Podhoranyi received the Ph.D. degree in geoinformatics with a focus on hydraulic modeling from the VSB - Technical University of Ostrava, Ostrava, Czech Republic, in 2013.

He is a data analyst by background. He is currently a Researcher with IT4Innovations National Supercomputing Center, Ostrava. He has participated in several national and international projects related to data processing and artificial intelligence. He has authored or coauthored many papers in international peer-reviewed journals and conferences with a focus on data processing, artificial intelligence, and hydraulic modeling. His widespread research interests include artificial intelligence, databases, data processing, remote sensing, and cybersecurity.