Detecting Looted Archaeological Sites from Satellite Image Time Series

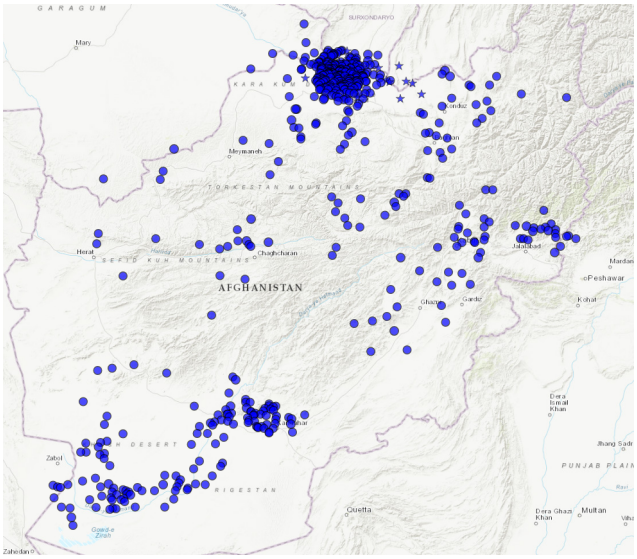
Supplementary Material

We first provide additional visualizations (Section A) and an analysis of learned temporal attention weights (Section B), and then complete the benchmark with implementations details (Section C) and further clarifications (Section D). The code and dataset can be found at https://github.com/ElliotVincent/DAFA-LS.

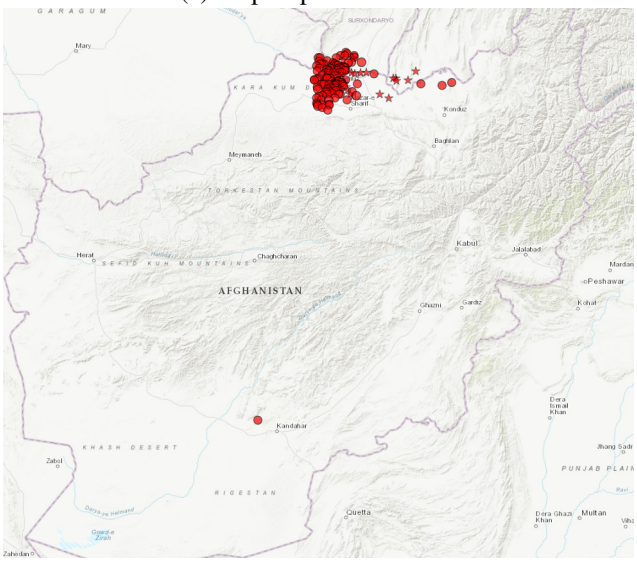
A. Additional visualizations

A.1. Location of DAFA-LS sites

We report in Figure A1 the location of DAFA-LS sites. We note a high concentration of archaeological sites in the northern region of Afghanistan in general, which is amplified for looted sites in particular. Note that, in order to prevent misuse of our data by malevolent individuals or organizations, we do add random noise to the point coordinates before plotting these maps. Hence, some sites might look like they are located outside Afghanistan, which is not actually the case.



(a) Map of preserved sites



(b) Map of looted sites

Figure A1. **Sites location.** We show the location of preserved (a) and looted (b) sites of DAFA-LS, adding strong random noise to their coordinates to prevent misuse of the data. Test sites are marked with a star (⋆).

A.2. Examples of looting marks

We show in Figure A2 three examples of visible looted marks. For the selected sites, we show images before and after the looting, with a zoom on the damaged area. The scars are typical of mechanical looting performed with bulldozers for example.

A.3. Failure cases

We report in Figure A3 all the time series for which our best baseline (DOFA+LTAE) predicts the wrong label with a confidence higher than 95%. We note that, for these sites, it is very difficult for a non-expert human eye to identify looting marks if any, or even to clearly see the structure of an archaeological site.

B. Temporal attention weights

We visualize in Figure B1 the temporal attention weights, gathered by year and averaged over all looted test sites, for two (out of eight) attention heads of DOFA+LTAE. We can see that the years 2020 and 2021 draw more attention from the model.

C. Implementation details

C.1. Single-frame methods

We use the PyTorch implementation of ResNet20 for CIFAR-10 by Yerlan Idelbayev[2], the torchvision implementation of ResNet18 and ResNet34, and the official PyTorch implementation of SatMAE[3], Scale-MAE[4] and DOFA[5]. We use a base version of foundations models when available (DOFA) and a

[2] https://github.com/akamaster/pytorch_resnet_cifar10

[3] https://github.com/sustainlab-group/SatMAE

[4] https://github.com/bair-climate-initiative/scale-mae

[5] https://github.com/zhu-xlab/DOFA

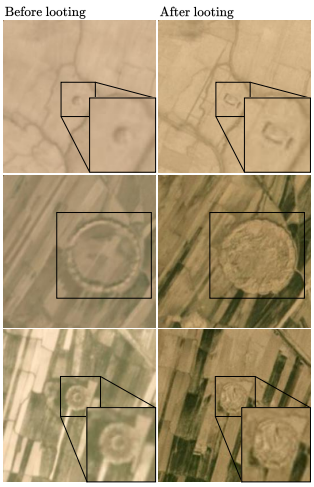


Figure A2. **Example of visible looting marks.**

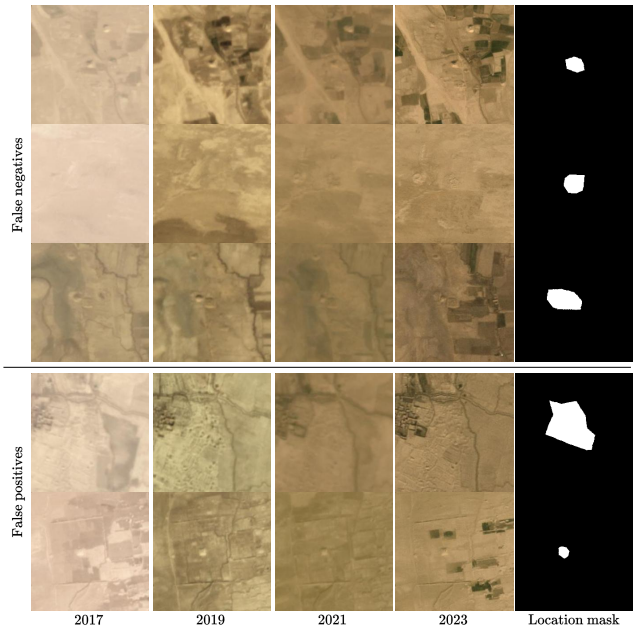
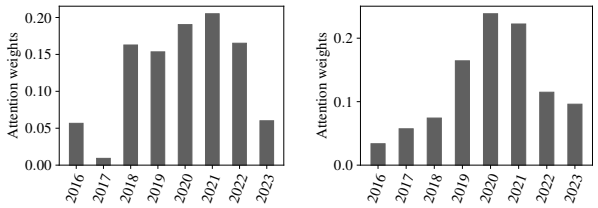


Figure A3. **Example of failure cases.** We show all the time series for which our best baseline (DOFA+LTAE) predicts the wrong label with a confidence higher than 95%.

large version otherwise (SatMAE, Scale-MAE). The models are trained with a learning rate of 10^{-3} for 60 epochs and a batch size of 32 (13,380 iterations). ResNet20 is trained from scratch. We train a single linear layer on top of a pretrained frozen DOFA.

C.2. Pixel-wise multi-frame methods

We use the PyTorch implementations of DuPLo and TempCNN available in Transformer's official public repository[6] and the PyTorch implementation of LTAE available in UTAE's official public repository[7]. These models are trained with a learning rate of 10^{-4} for 2 epochs and a batch size of 128 (25,000 iterations). Each batch contains 128 pixel-wise time series, sampled from possibly different SITS. For all these methods, we use smaller versions of the architecture compared to their default setting, taking into account the relative small size of DAFA-LS and to limit over-fitting. The used configurations are available in our official repository[8].



(a) Head n°2/8 (b) Head n°3/8

Figure B1. **Temporal attention.** We report the attention weights for two of the eight heads of DOFA+LTAE. We report the mean of the weights for all looted test sites. We sum all the monthly weights for each year.

C.3. Whole-image multi-frame methods

We use the official PyTorch implementations of UTAE[9] and TSViT[10]. These models are trained with a learning rate of 10^{-4} for 100 epochs and a batch size of 4 (14,900 iterations). For PSE+LTAE, SatMAE+LTAE, Scale-MAE+LTAE and DOFA+LTAE, we use SatMAE, Scale-MAE, DOFA and LTAE implementations mentioned above and the official implementation of PSE[11]. These models are trained with a learning rate of 10^{-4} for 200 epochs and a batch size of 8 (14,900 iterations). For PSE+LTAE, 1024 randomly sampled in-mask pixels are used during training and all in-mask pixels are used at inference. A majority voting rule is applied at inference to determine the final prediction for a given SITS. For SatMAE+LTAE, Scale-MAE+LTAE and DOFA+LTAE, the backbone is loaded with pretrained weights and frozen during training. We use smaller versions UTAE and TSViT compared to their default setting, taking into account the relative small size of DAFA-LS and to limit over-fitting. The used configurations can be found in our official repository[12].

[6] https://github.com/MarcCoru/crop-type-mappin
[7] https://github.com/VSainteuf/utae-paps
[8] https://github.com/ElliotVincent/DAFA-LS
[9] See footnote 7.
[10] https://github.com/michaeltrs/DeepSatModels
[11] https://github.com/VSainteuf/pytorch-psetae
[12] See footnote 8

Id	Category	Image level	Dates used	Input type	Task	Inference strategy
(i)	Single-frame	Whole-image	2023	Image	Classification	12-month image voting
(ii)	Multi-frame	Pixel-wise	2016-2023	SITS	Classification	In-mask pixel voting
(iii)	Multi-frame	Whole-image	2016-2023	SITS	Segmentation	In-mask pixel voting
(iv)	Multi-frame	Whole-image	2016-2023	SITS	Classification	Direct prediction

Id	Model name
(i)	ResNet20/18/34 [38], SatMAE [19], Scale-MAE [68], DOFA [89]
(ii)	DuPLo [42], Transformer [71], LTAE [32], TempCNN [63]
(iii)	UTAE [33], TSViT (seg. head) [79]
(iv)	{PSE [34], SatMAE [19], Scale-MAE [68], DOFA [89]}+LTAE [32], TSViT (cls. head) [79]

Table D1. **Categorization of baseline methods.** We explicit the main characteristics of the different evaluated baseline methods.

D. Benchmark: additional details

We detail in Table D1 the main characteristics of the evaluated baselines. In particular, we report the dates available at training time (*Dates used*), the learning task (*Task*), and the inference procedure (*Inference strategy*).

Suggested use cases of our benchmark. We have benchmarked methods that distinguish between looted and preserved sited given a SITS and a coarse location mask. An example use case of such approaches would be the monitoring of known endangered sites. SITS could be accumulated regularly over time for such sites, and alerts raised if the SITS is flagged as looted by the model. We encourage human verifications with higher resolution imagery when possible, before visiting a flagged site on the ground.