
Nacala-Roof-Material: Drone Imagery for Roof Detection, Classification, and Segmentation to Support Mosquito-borne Disease Risk Assessment

Venkanna Babu Guthula
University of Copenhagen, DK

Stefan Oehmcke
University of Copenhagen, DK

Remigio Chilaule
Royal Danish Academy, DK
#MapeandoMeuBairro, MZ

Hui Zhang
University of Copenhagen, DK

Nico Lang
University of Copenhagen, DK

Ankit Kariryaa
University of Copenhagen, DK

Johan Mottelson
Royal Danish Academy, DK

Christian Igel
University of Copenhagen, DK

Abstract

Low-quality housing is associated with an increased risk of mosquito-borne diseases. As certain roof characteristics are linked to an elevated risk of malaria, the classification of roof types based on remote sensing imagery can support the assessment of malaria risk and help prevent the disease. To support research in this area, we release the Nacala-Roof-Material dataset, which contains high-resolution drone images from Mozambique with corresponding labels delineating houses and specifying their roof types. The dataset defines a multi-task computer vision problem, comprising object detection, classification, and segmentation. In addition, we benchmarked various state-of-the-art approaches on the dataset. Canonical U-Nets, YOLOv8, and a custom decoder on pretrained DINOv2 served as baselines. We show that each of the methods has its advantages but none is superior on all tasks, which highlights the potential of our dataset for future research in multi-task learning. While the tasks are closely related, accurate segmentation of objects does not necessarily imply accurate instance separation, and vice versa. We address this general issue by introducing a variant of the deep ordinal watershed (DOW) approach that additionally separates the interior of objects, allowing for improved object delineation and separation. We show that our DOW variant is a generic approach that improves the performance of both U-Net and DINOv2 backbones, leading to a better trade-off between semantic segmentation and instance segmentation.

1 Introduction

Mosquito-borne diseases refer to a group of infectious illnesses transmitted by the bite of mosquitoes. Malaria is a mosquito-borne disease caused by single-celled parasites of the Plasmodium group spread through bites of infected female Anopheles mosquitoes. It ranks among the world’s most severe public health problems and is a leading cause of mortality and disease in many developing

countries. It is therefore crucial to improve prevention, control, and surveillance measures of malaria, particularly in sub-Saharan Africa (Venkatesan, 2024; WHO, 2023). Living in low-quality housing built with natural materials such as grass or palm (thatch) as a roof and cane, grass, shrub, or mud as internal and external walls is associated with an increased risk of malaria infection (Dlamini et al., 2017).

Low-quality housing is an important risk factor for transmission of malaria, as sub-standard houses have more mosquito entry points and most malaria transmissions in sub-Saharan Africa occur inside dwellings while the inhabitants are asleep Tusting et al. (2020, 2017); Jatta et al. (2018); Tusting et al. (2019). Houses with metal roofs are hotter in the daytime than houses with thatched roofs. This may reduce mosquito survival and inhibit parasite development within the mosquito in metal roof houses. On this basis, the proliferation of modern construction materials in sub-Saharan Africa may have contributed decisively to the reduction of malaria cases Tusting et al. (2019). Classification of roof characteristics thus holds potential to support malaria surveillance and control programs. Roof characteristics, such as geometry, construction material, and condition can be monitored using remote sensing imagery to advance risk assessment of mosquito-borne diseases and guide mitigation strategies, especially when detailed health and socioeconomic data are scarce.

Here, we introduce the Nacala-Roof-Material drone-imagery dataset to support the development of machine learning algorithms for automated building *and* roof type mapping in low-income areas prone to malaria risk. Our dataset is based on high-resolution drone imagery of peri-urban and rural settlements in Nacala, Mozambique. The Mozambican NGO #MapeandoMeuBairro has delineated buildings and categorized them according to five roof types and the authors again carefully verified all annotations. We define three tasks on the Nacala-Roof-Material dataset, (i) semantic segmentation of building areas, (ii) instance segmentation, and (iii) multi-class roof type classification.

While these tasks may be related, closer inspection reveals a misalignment between their objectives. Accurate segmentation as measured by the intersection over union (IoU) does not necessarily imply accurate object separation, and vice versa. Furthermore, for accurate detection and classification, it would be sufficient to only detect the interior of an object as long as the segmented area allows to correctly classify the type. If the roofs of two buildings are (almost) touching, then segmentation may have a high IoU, but could make it difficult to separate buildings for counting. This is also a common issue in other applications, e.g., when studying cells in medical images (Ronneberger et al., 2015) or trees from satellite images (Brandt et al., 2020; Mugabowindekwe et al., 2022)).

We benchmark three existing state-of-the-art approaches on our multi-task dataset. First, we evaluate YOLOv8 (Jocher et al., 2023) developed for object detection, classification, and instance segmentation. Second, we build a segmentation model based on DINov2 (Oquab et al., 2024), a state-of-the-art pretrained vision transformer. Lastly, we evaluate U-Net (Ronneberger et al., 2015) a fully-convolutional encoder-decoder architecture, designed for semantic segmentation. To address the potential conflicts between pixel-level segmentation and correct object separation as outlined above, we propose a simple approach based on the recent work by Cheng et al. (2024), which we refer to as the Deep Ordinal Watershed (DOW) method. We extend both U-Net and DINov2 for semantic segmentation with DOW to produce an additional output map that predicts the interior of objects. While the original exterior segmentation map maximizes the IoU, we show that this interior map supports object separation.

The main contributions of our work are the following:

1. We provide the Nacala-Roof-Material dataset containing drone imagery at ≈ 4.4 cm resolution from peri-urban and rural areas in a sub-Saharan African region. These contain accurate segmentation labels of 17 954 buildings, categorized into five roof types.
2. We define a multi-task benchmark based on the data, which covers semantic segmentation, instance segmentation, and multi-class segmentation. We implemented and benchmarked different carefully adopted baseline methods, reflecting three different approaches to address the tasks.
3. We propose a general and simple approach to extend models for semantic segmentation to yield good segmentation *and* object separation results.

The data and code for reproducing the experiments are made freely available at <https://mosquito-risk.github.io/Nacala>.

The next section presents the Nacala-Roof-Material data, provides some background about roof types and risk of vector-borne diseases, and briefly discussed related datasets. Section 3 describes the deep

learning models we evaluated with an emphasis on the deep watershed methods. Experimental results are presented in Section 4 before we conclude.

2 Nacala-Roof-Material Data

Background: Housing conditions and risk of mosquito-borne diseases. In sub-Saharan Africa, housing conditions, health outcomes, and socioeconomic status of the residents are interrelated (Gram-Hansen et al., 2019; Degarege et al., 2019; Tusting et al., 2020). As poverty is widespread, diseases are more prevalent, and data are scarce in this region, automatic profiling of housing conditions based on analysis of satellite imagery holds the potential to estimate the socioeconomic status of the inhabitants and assess the risk of disease. This may in turn support targeted public health interventions.

Mosquitoes are vectors for diseases such as malaria, dengue, Zika, West Nile fever, Chikungunya, and Yellow fever. In 2022, more than 600 000 deaths occurred due to malaria globally and out of the approximately 249 million documented cases, around 233 million occurred within the WHO African Region, accounting for roughly around 94% of the total documented cases. The economic impact of malaria in Sub-Saharan Africa not only impedes progress towards achieving Sustainable Development Goal 3 (Good Health and Well-being) but also undermines efforts to attain SDG 1 (No Poverty) and SDG 8 (Decent Work and Economic Growth) by compromising economic productivity. Extreme weather conditions caused by climate change will likely exacerbate problems with mosquito-borne diseases in sub-Saharan Africa, as floods are expected to increase in frequency and have been linked to outbreaks of malaria in Africa (Githcko et al., 2000).

Low-quality housing increases the risk of transmission of diseases by mosquitoes, as sub-standard houses have more mosquito entry points and thereby increase human exposure to infection in the home (Tusting et al., 2015). Mosquito survival is lower in metal-roof houses compared to thatched-roof houses due to higher daytime temperatures (Tusting et al., 2015). Most malaria transmissions in sub-Saharan Africa occur indoors at night, and poor climatic performance of housing has been linked to increased malaria risk (Jatta et al., 2018). This is because elevated indoor temperatures can cause discomfort for inhabitants, which may result in decreased use of mosquito nets during the night. Roof materials, geometry, and conditions are critical for indoor climate, as roofs comprise the primary surface exposed to the sun. Automatic classification of roof characteristics thus holds potential for informing risk assessment of malaria and support targeted interventions.

The Nacala-Roof-Material dataset. We collected drone imagery of the Nacala region in Mozambique. The burden of malaria in Mozambique is approximately 10-fold the world average (number of documented cases compared to the total population, Venkatesan, 2024). The data covers three informal settlements of Nacala, a city of 350 000 inhabitants on the northern coast of Mozambique. Aerial imagery was collected using a DJI Phantom 4 Pro drone and processed using AgiSoft Metashape software. All data was recorded between October and December 2021, under a development project led by the Nacala Municipal Council. The image resolution is ≈ 4.4 cm, and we made all raw imagery available in OpenAerialMap (OpenAerialMap) with slightly decreased resolution. The total number of buildings in the study areas is 17 954. We distinguished five major types of roof materials in Nacala, namely metal sheet, thatch, asbestos, concrete, and no-roof, and their counts are 9776, 6428, 566, 174, and 1010, respectively. The region is mostly dominated by metal sheets and thatch roofs.

From the three informal settlements, see Figure 1, the first two areas were split into training $\mathcal{D}_{\text{train}}$, validation \mathcal{D}_{val} , and test $\mathcal{D}_{\text{test}}$ using stratified sampling. We created a square grid of 225 meters and counted the roof types in these cells. Then we partitioned the cells into three sets based on the class counts to achieve a similar class distribution in each set, where we prioritized the distribution of minority classes (i.e., concrete and asbestos). We defined that a building only belongs to a specific grid cell if its centroid falls into the cell. If a building area falls into two grid cells and those two cells belong to two different sets (e.g., training and test set), we choose to have data pixels in the set where the centroid of the building is placed. The remaining part of the building in the other set was masked to avoid data leaking between sets.

Although objects in training, validation, and test sets are from different cells, they stem from the same two areas. To evaluate the generalization to a new area without adjacent training data, we hold out the third settlement as a second test set referred to as \mathcal{D}_{ext} .

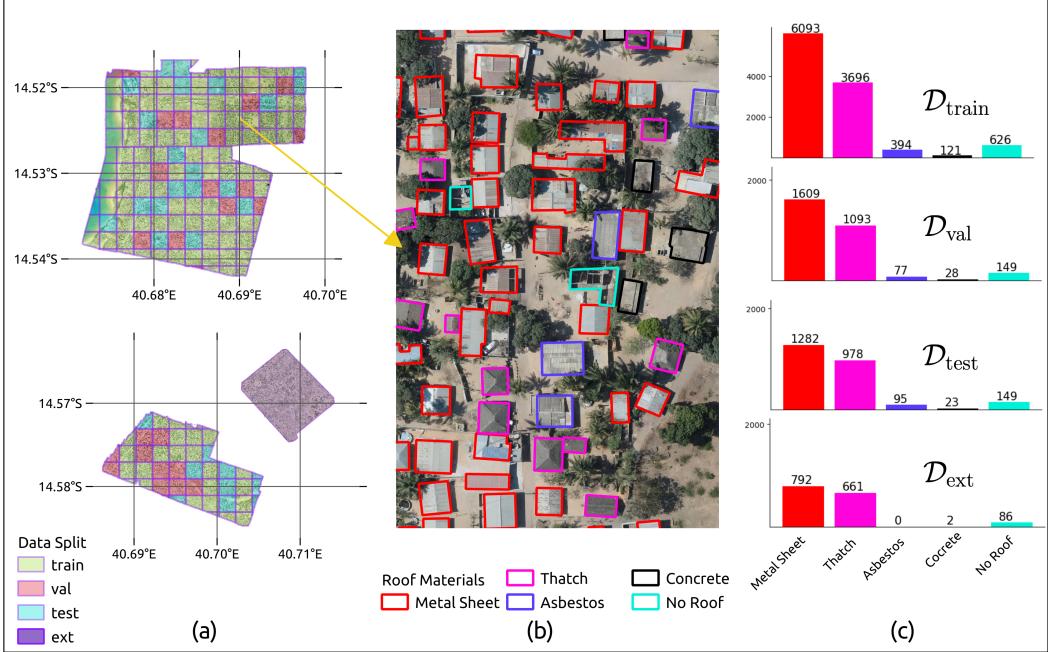


Figure 1: (a) Visualisation of the training, validation and test sets with reference to longitude and latitude; (b) Drone imagery with labels; (c) Instance counts for each class in all sets.

Related datasets. The project “Mapping Informal Settlements in Developing Countries using Machine Learning with Noisy Annotations and Multi-resolution Multi-spectral Data” is most closely related to our work Helber et al. (2018); Gram-Hansen et al. (2019). They used freely available 10m/pixel resolution imagery from the Sentinel-2 satellite and obtained labels for three roof types (metal, shingles, thatch) from geo-located survey data provided by Afrobarometer¹. These labels are very noisy in space and time. The labels are often not aligned with buildings because the geo-located coordinates were distorted for privacy reasons. Furthermore, the survey questions and satellite image observation may not be aligned in time. While the low spatial resolution of the Sentinel-2 imagery might allow to cover large geographic regions, it makes roof type classification challenging (Helber et al., 2018).

There are many datasets that contain remote sensing imagery with building labels, which, however, typically do not distinguish roof types. In particular, *Open Buildings* is a freely available continental-scale building dataset covering the whole of Africa (Sirk et al., 2021). In comparison, Nacala-Roof-Material is much more focused, providing significantly higher resolution images, more accurate delineations, and in particular provides roof type classifications.

Alidoost and Arefi (2018) distinguish between roof types in aerial images. However, they map a rather high-income town in Germany, where they distinguish between three roof shapes common in that region (flat, gable, and hip). Another dataset for classifying roof geometry is provided by Persello et al. (2023), who distinguish 12 fine-grained details of roof geometry.

3 Benchmarked Methods

This section presents the approaches we benchmarked on the Nacala-Roof-Material data set. The goal is to accurately segment the buildings (as assessed by metrics based on the IoU), separate individual buildings, and classify the roof materials. As baselines, we considered U-Net (Ronneberger et al., 2015), YOLOv8 (Jocher et al., 2023), and a model performing segmentation based on DINOv2 (Oquab et al., 2024). Furthermore, we extend the U-Net and the DINOv2 based systems with the deep ordinal watershed method recently proposed by Cheng et al. (2024). These approaches are compared in two settings. In the *two-stage* setting, we first solved the building segmentation and separation tasks

¹www.afrobarometer.org

and afterwards classified the roof material for each detected building. In the *end-to-end* setting, segmentation and classification were done in parallel.

3.1 Baseline Models

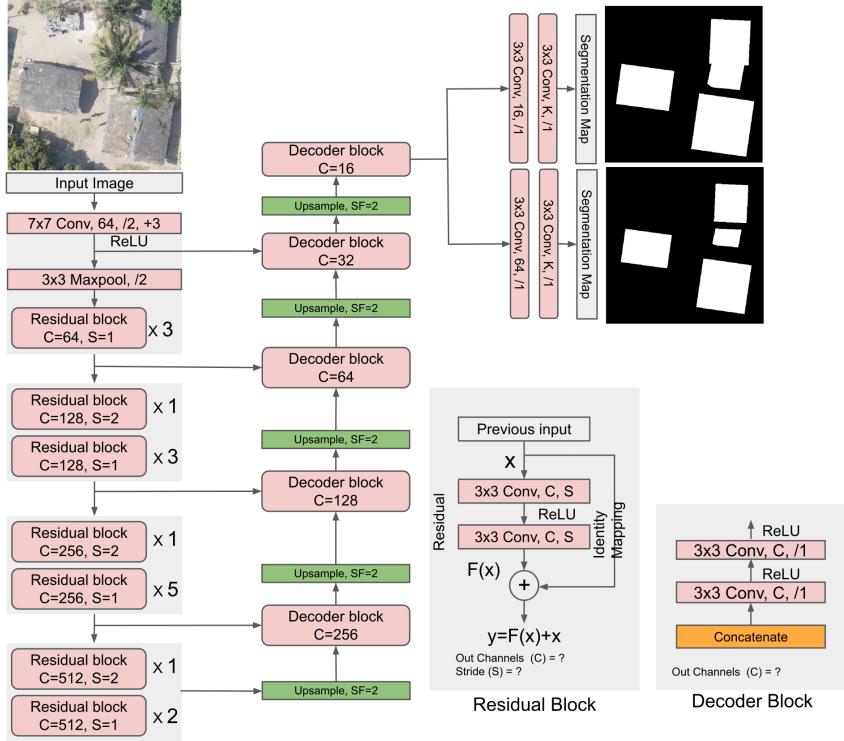


Figure 2: U-Net_{DOW} architecture producing two output maps, segmenting objects and their interiors, respectively. The architecture differs from the baseline U-Net only in the output heads.

U-Net. The U-Net is arguably the most common architecture for semantic segmentation (Ronneberger et al., 2015). We utilized a ResNet34 (He et al., 2016) encoder pretrained on ImageNet and a decoder similar to the original U-Net, except that we used nearest-neighbor upsampling instead of transposed convolutions (Odena et al., 2016), see Figure A.5 in the Appendix.

To identify individual instances in the semantic segmentation output map, the connected components in the map were determined (Brandt et al., 2020). To better separate individual buildings, we used a pixel-wise weight map during training that puts more emphasis on the space between buildings as already suggested by Ronneberger et al. (see Appendix A.1 for details) and commonly used in remote sensing (e.g. Brandt et al., 2020). However, this is not sufficient to separate buildings that are very close to each other or touch each other. Thus, we modified the target segmentation masks during training: Some border pixels were relabeled as background to ensure that there is a minimum gap of $n_{\text{gap}} = 7$ pixels between roofs. This modification of the target masks was only applied during training, before computing the weight map but not when calculating any performance metrics.

YOLOv8. We trained YOLOv8 (Jocher et al., 2023), which is among the state-of-the-art methods for instance segmentation. We used a model pretrained on the COCO dataset. While the original YOLO architecture was designed for object detection, YOLOv8 allows for instance segmentation by integrating concepts from YOLACT (Bolya et al., 2019).

DINOv2. We benchmarked an approach based on DINOv2 (Oquab et al., 2024) a state-of-the-art pretrained vision transformer. It uses the DINOv2 *Base* model as encoder, which is extended by a convolutional decoder. The DINOv2 output, a patch embedding with the shape of $\mathbb{R}^{1024 \times 768}$, is reshaped into feature maps of size $\mathbb{R}^{32 \times 32 \times 768}$. Then convolutional and linear upsampling layers

are used on top of these feature maps as a decoder (see Appendix A.3). We used the same loss function, weighting function, training label adjustment, and training strategy as for U-Net. We froze the encoder weights and only the convolutional decoder was trained.

3.2 Deep Ordinal Watershed

U-Nets and the DINOv2 based method described above try to classify each pixel as accurately as possible. However, for proper separation of objects it is sufficient – and typically preferable – if only the interior of an object is segmented. If the border of a building can be classified as background, even touching buildings can be separated. This reasoning leads to the deep ordinal watershed (DOW) model introduced by Cheng et al. (2024).

In the watershed approach, each pixel is assigned a height and the image is viewed as a topological map (Soille and Ansoult, 1990). A DOW architecture, see Figure 2, does not only predict a single segmentation mask but n_{lev} feature maps for $n_{\text{lev}} + 1$ discrete height levels, $\{0, 1, \dots, n_{\text{lev}}\}$, where 0 corresponds to the highest and n_{lev} to the lowest elevation. Background pixels are assumed to have level 0. The Euclidean distance transformation is computed for each object, and the distances are discretized into the remaining n_{lev} height levels. Target feature map $m \in \{1, \dots, n_{\text{lev}}\}$ marks all pixel with a distance level of m or higher. That is, the objects in the target feature maps get smaller with increasing m (if $n_{\text{lev}} = 1$ we recover the standard U-Net). Learning the discrete height levels of pixels this way solves an ordinal regression task (Frank and Hall, 2001; Cheng et al., 2008). Given the pixel heights, the watershed algorithm can be applied as a post-processing step for instance segmentation (Soille and Ansoult, 1990). Local minima in the elevation map define basins, each of which defines a distinct object. Adopting a flooding metaphor, the watershed algorithm now floods the basins until basins attributed to different starting points meet on watershed lines. Pixels attributed to the same basin belong to the same object.

Cheng et al. (2024) employ a DOW U-Net for individual tree segmentation, however, without a comparison with a standard U-Net or exploring different numbers of levels. For our task, we hypothesize that a minimal number of $n_{\text{lev}} = 2$ different non-background heights is sufficient. In this setting, the system outputs two masks representing the full object and its interior, respectively (for a comparison with a DOW U-Net with $n_{\text{lev}} = 6$ we refer to Appendix A.2 and Appendix B). We empirically evaluated DOW variants of both our U-Net architectures and DINOv2 based systems. We describe the U-Net extension in more detail in Appendix A.2, the DINOv2 based systems were modified analogously. Let n_{pix} denote the difference in distance between two levels. The smallest building in our data set has size 1.463 m^2 . Thus, for the given image resolution, the number of pixels per side is approximately $\sqrt{1.46}/0.044$. This suggests to discretize such that $n_{\text{pix}} < 13$, and we picked $n_{\text{pix}} = 10$.

Although the approaches are related, we would like to stress the DOW method is conceptually different from *deep level sets*, where deep neural networks learn a (continuous) level set function, the zero-set of which defines object boundaries (Hu et al., 2017; Hatamizadeh et al., 2020), as well as from predicting interior and border of an object as, for instance, done by Girard et al. (2021).

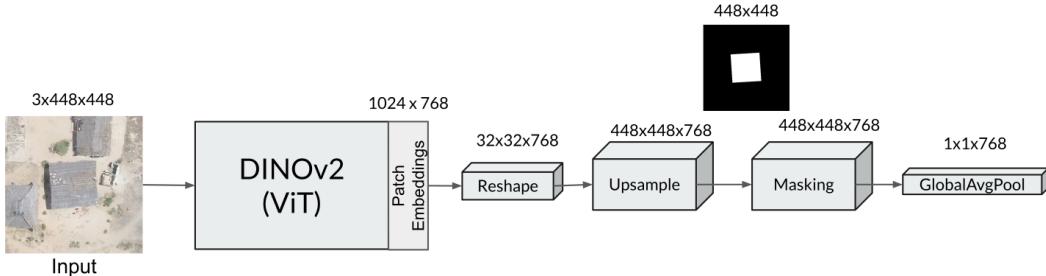


Figure 3: The architecture of the DINOv2 based roof material classifier. A classifier (e.g., logistic regression) is applied to the resulting feature vector.

3.3 Two-stage vs. End-to-end

All the neural network architectures described above can directly classify the roof types of detected buildings by predicting multi-class segmentation masks. However, encouraged by good classification results using DINOv2 features, we also studied an alternative two-stage approach: First we segmented and separated the buildings using the algorithms described above ignoring the roof material information. That is, we reduced the multi-class problem to a binary task. After that, we predicted the roof material of each detected building. We used DINOv2 to process a 448×448 patch centered around each building, see Figure 3. The output of DINOv2, a patch embedding with the shape of $\mathbb{R}^{1024 \times 768}$ was reshaped into feature maps of $\mathbb{R}^{32 \times 32 \times 768}$. These feature maps were then upsampled to the input patch size, masked with a target binary building mask, and average pooling was applied to obtain the final feature vector for the building. Standard machine learning classifiers were applied to this embedding to predict the roof material, where linear probing gave the best results (see Appendix B.2 for a comparison of different classifiers).

4 Experiments and Results

4.1 Experimental Setup and Evaluation Metrics

All models, except for YOLOv8 where we followed its original training protocol, were trained using cross-entropy loss with pixel-wise weighting. We employed the AdamW optimizer Loshchilov and Hutter (2019) with an initial learning rate of 0.0003. All models were trained for 300 epochs, utilizing a learning rate scheduler that decreased the learning rate by a factor of 10 every 50 epochs. The final weight configuration and hyperparameters for each model were selected based on the highest IoU score achieved on the validation dataset. The hyperparameters of the U-Net were chosen by observing results on the validation data set in an iterative process. The high training speed of YOLOv8 allowed for more systematic model selection: We applied the genetic algorithm that comes as part of the YOLOv8 framework for hyperparameter optimization (Jocher et al., 2023). The input patch sizes for the U-Net variants, YOLOv8, and DINOv2 models were 512, 640, and 448, respectively.

The semantic segmentation performance was evaluated by the IoU. We considered both the IoU of the binary building segmentation and the mean IoU for class-specific roof segmentation. The roof materials concrete and asbestos are very rare. While $\mathcal{D}_{\text{train}}$, \mathcal{D}_{val} , and $\mathcal{D}_{\text{test}}$ are stratified samples containing all classes, the spatially distinct data \mathcal{D}_{ext} does not contain any example of the two roof types, see Figure 1. To allow for a better comparison between the two test sets and to see the effect of the rare classes on the macro-averaged mean IoU, we provide the mean IoU of the three main classes (mIoU^3) alongside with the mean IoU of all five classes (mIoU^5).

Instance segmentation was assessed using the AP_{50} score, that is, the average precision evaluated at an IoU threshold of 0.5 (Everingham et al., 2010; Lin et al., 2014). We evaluated the AP for both the predictions of building instances and the predictions of multi-class roof type instances (i.e., in the latter case an object is only detected if the roof material is correctly identified). Similar to IoU, mAP_{50}^3 and mAP_{50}^5 denote the mean AP_{50} over three and five classes. To estimate the average precision, a confidence score is required for each building segment. The confidence score of binary and multi-class segmentation models was obtained by interpreting the neural networks’ outputs as probability distributions over classes and calculating the mean probability of belonging to the predicted class over all pixels within a predicted segment. The exception was YOLOv8, which provides its own confidence score. When a classifier using DINOv2 features was used on top of binary segmentation models, the confidence score was derived from the canonical probability score of the classifier. Additional metrics, AP_{50-95} and TP_s , are shown in Appendix B. Information on the computer resources is provided in Appendix A.4.

4.2 Results and Discussion

Our experimental results on $\mathcal{D}_{\text{test}}$ and \mathcal{D}_{ext} are presented in Table 1, additional details can be found in Appendix B. All metrics on the test sets were computed on raw images instead of patches to avoid artifacts when splitting images. We report averages over five trials on the corresponding standard deviations. The methods reached AP_{50} and IoU values on the spatially separated test set of up to 0.963 and 0.880, respectively. Thus the tasks can be solved with accuracies high enough for subsequent analysis while still leaving room for improvement. Detecting thatch roofs is particularly

Table 1: Benchmarking results on the Nacala-Roof-Material dataset. The table reports averages over five trials \pm standard deviations. The upper five models were trained in the two-stage setting. The lower half of the models was trained in the end-to-end setting, where multi-class classification is performed together with the segmentation as indicated by the subscript *Multi*. Models that used the DOW extension are indicated by the subscript *DOW*. IoU and AP₅₀ were computed on the binary output, where the predictions of multi-class models were binarized. mIoU and mAP₅₀ are macro averages, the superscripts indicate whether the averaging was done over all five classes or over the three frequent roof types. Results for individual roof types can be found in Appendix B.

Model Name	$\mathcal{D}_{\text{test}}$						\mathcal{D}_{ext}			
	pixel level			object level			pixel level		object level	
	IoU	mIoU ³	mIoU ⁵	AP ₅₀	mAP ₅₀ ³	mAP ₅₀ ⁵	IoU	mIoU ³	AP ₅₀	mAP ₅₀ ³
YOLOv8	0.866 ± 0.012	0.713 ± 0.019	0.568 ± 0.015	0.941 ± 0.003	0.815 ± 0.011	0.698 ± 0.018	0.896 ± 0.002	0.761 ± 0.006	0.963 ± 0.005	0.846 ± 0.008
DINOv2	0.833 ± 0.002	0.755 ± 0.004	0.562 ± 0.003	0.882 ± 0.004	0.789 ± 0.006	0.683 ± 0.008	0.905 ± 0.000	0.747 ± 0.011	0.919 ± 0.005	0.806 ± 0.008
DINOv2 _{DOW}	0.884 ± 0.001	0.763 ± 0.002	0.565 ± 0.004	0.930 ± 0.005	0.836 ± 0.002	0.725 ± 0.004	0.905 ± 0.001	0.852 ± 0.007	0.956 ± 0.001	0.852 ± 0.007
U-Net	0.895 ± 0.003	0.757 ± 0.024	0.570 ± 0.016	0.910 ± 0.005	0.810 ± 0.008	0.688 ± 0.014	0.909 ± 0.001	0.748 ± 0.007	0.929 ± 0.000	0.787 ± 0.011
U-Net _{DOW}	0.895 ± 0.002	0.775 ± 0.013	0.577 ± 0.009	0.935 ± 0.001	0.836 ± 0.005	0.730 ± 0.011	0.911 ± 0.002	0.764 ± 0.006	0.947 ± 0.004	0.812 ± 0.008
YOLOv8 _{Multi}	0.824 ± 0.023	0.708 ± 0.010	0.550 ± 0.017	0.910 ± 0.005	0.816 ± 0.009	0.597 ± 0.007	0.885 ± 0.002	0.785 ± 0.006	0.948 ± 0.003	0.849 ± 0.015
DINOv2 _{Multi}	0.880 ± 0.002	0.774 ± 0.004	0.699 ± 0.012	0.899 ± 0.003	0.820 ± 0.010	0.689 ± 0.025	0.899 ± 0.002	0.818 ± 0.005	0.946 ± 0.001	0.880 ± 0.011
DINOv2 _{DOW-Multi}	0.885 ± 0.001	0.786 ± 0.006	0.734 ± 0.006	0.918 ± 0.003	0.824 ± 0.011	0.702 ± 0.013	0.902 ± 0.001	0.819 ± 0.006	0.950 ± 0.005	0.875 ± 0.010
U-Net _{Multi}	0.879 ± 0.012	0.783 ± 0.010	0.634 ± 0.024	0.924 ± 0.004	0.850 ± 0.011	0.716 ± 0.018	0.903 ± 0.002	0.805 ± 0.020	0.943 ± 0.010	0.844 ± 0.039
U-Net _{DOW-Multi}	0.892 ± 0.001	0.777 ± 0.012	0.672 ± 0.042	0.928 ± 0.002	0.829 ± 0.011	0.671 ± 0.022	0.904 ± 0.002	0.794 ± 0.014	0.942 ± 0.005	0.812 ± 0.021

relevant, as they are associated with an increased malaria risk (Tusting et al., 2019), and these roofs can be identified particularly well, see Table B.3 in the Appendix.

When comparing the different approaches, we find that there is no method that was better than the others across all metrics. The U-Nets and YOLOv8 did well on their home grounds: YOLOv8 gave good object detection results (e.g., the best AP₅₀ scores), while the U-Nets performed well for semantic segmentation as measured by IoU. DINOv2 combined with a simple decoder was also competitive. Exemplary results are shown in Figure 4. As could be expected, classifying the minority roof types asbestos and especially concrete (which resembles concreted background areas) was most difficult, in particular for end-to-end YOLOv8, see Table B.3. YOLOv8 had the tendency to produce artefacts when applied to the larger images, especially when predictions of smaller overlapping patches were combined. This is one of the reasons for its lower IoU score.

In general, the DOW extension improved both U-Nets and DINOv2 based architectures. Comparing DINOv2 with DINOv2_{DOW} and U-Net with U-Net_{DOW}, the DOW variants were better in all ten performance indices (except for IoU on $\mathcal{D}_{\text{test}}$ where U-Net and U-Net_{DOW} gave the same result). Comparing DINOv2_{Multi} with DINOv2_{DOW-Multi}, the latter was better in all indicators except mAP₅₀³ on \mathcal{D}_{ext} . Only for U-Net_{DOW-Multi} the results were mixed, using DOW gave lower values for five indices and higher values for the other half. Overall, the DOW extension had a statistically significant positive effect on the object separation as intended. If we pool all 20 DOW trials and compare with the corresponding trials predicting a single mask, then the AP50 improved significantly (two-sided Wilcoxon rank sum test, $p < 0.001$) while the difference in IoU was not significant ($p > 0.05$).

Limitations. The Nacala-Roof-Material dataset is not a large-scale data set by current standards and it is restricted to a single region. However, considering the proliferation of low-cost drone technologies, high-resolution geospatial surveying is becoming increasingly affordable and common in sub-Saharan Africa. Accordingly, similar but unlabelled data will likely become available in the coming years at large scale, which makes it important to develop methods to make good use of these

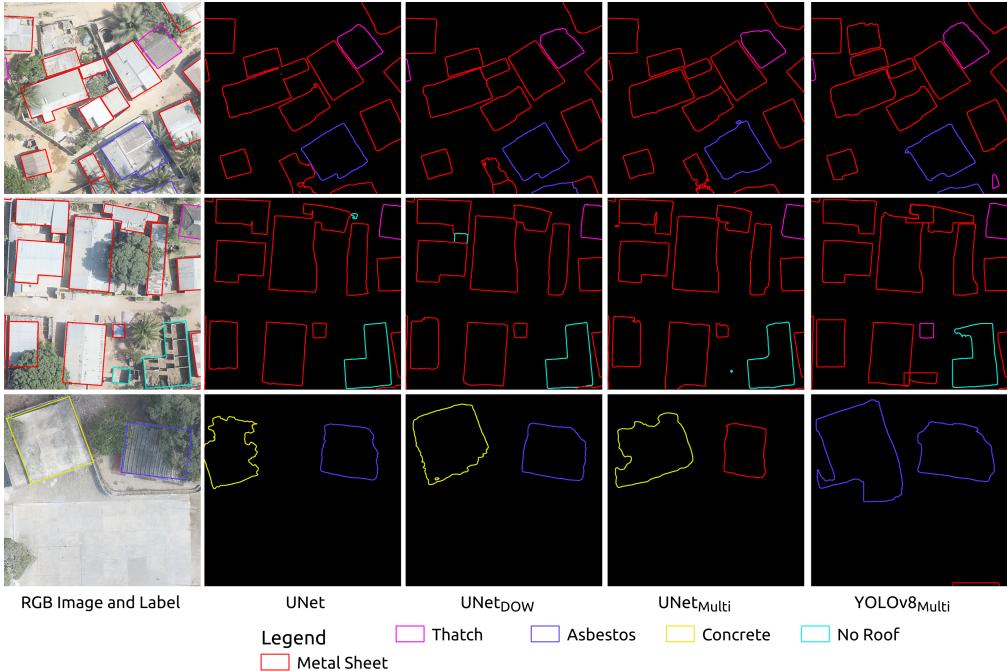


Figure 4: Exemplary predictions on $\mathcal{D}_{\text{test}}$ by different models. The predictions are polygonized and colored by class. The roof types with few training examples, asbestos and concrete, are particularly difficult, see bottom row.

data now. The Nacala-Roof-Material dataset covering informal settlements is a good example for the target areas of our risk disease monitoring and prevention research. In this context, Mozambique is particularly relevant because the country suffers from a high malaria incidence rate (Venkatesan, 2024). The second test set allows for testing generalization in an area geographically separated from the main training/test/validation data. In general, we would argue that there is a need for medium size benchmark data sets such as the Nacala-Roof-Material data to support equity in machine learning research, as we need benchmarks that can be utilized by researchers with limited compute resources.

5 Conclusions

The Nacala-Roof-Material dataset contains high-resolution drone imagery from informal settlements in Mozambique, where buildings and their roof material were carefully annotated. We curated the dataset as part of an intercontinental and interdisciplinary research project on risk assessment of mosquito-borne diseases, especially malaria, with the goal to predict risk maps and to develop and support measures for risk reduction. From a methodological perspective, the dataset defines a multi-task problem. We are interested in accurate semantic segmentation to determine the roof areas and also in identifying the individual buildings and classifying their roof types. Thus, the dataset adds to the landscape of computer vision benchmarks by providing a relevant resource for the development and evaluation of frameworks that strive at solving semantic segmentation as well as object detection and classification simultaneously with a high accuracy. For example, working on the Nacala-Roof-Material data has led us to the proposed deep ordinal watershed (DOW) approach, a reduced variant of the method described by Cheng et al. (2024). This variant method first segments objects along with their interiors into two elevation levels and then performs a watershed segmentation to separate objects. The DOW idea is applicable beyond the Nacala-Roof-Material data, on which it improved both the standard U-Net architectures as well as a system based on DINoV2 features for segmentation. Implementations of all algorithms are made publicly available together with the data (<https://mosquito-risk.github.io/Nacala>). With our proposed dataset, we invite the machine learning community to develop new approaches for interpreting high-resolution drone images that can ultimately support risk assessments of vector-borne diseases.

References

- F. Alidoost and H. A. Arefi. CNN-based approach for automatic building detection and recognition of roof types using a single aerial image. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 86:235–248, 2018.
- D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. YOLACT: Real-time instance segmentation. In *International Conference on Computer Vision (ICCV)*, pages 9157–9166, 2019.
- M. Brandt, C. J. Tucker, A. Kariryaa, K. Rasmussen, C. Abel, J. Small, J. Chave, L. V. Rasmussen, P. Hiernaux, A. A. Diouf, L. Kergoat, O. Mertz, C. Igel, F. Gieseke, J. Schöning, S. Li, K. Melocik, J. Meyer, S. Sinno, E. Romero, E. Glennie, A. Montagu, M. Dendoncker, and R. Fensholt. An unexpectedly large count of trees in the western Sahara and Sahel. *Nature*, 587:78–82, 2020.
- J. Cheng, Z. Wang, and G. Pollastri. A neural network approach to ordinal regression. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1279–1284. IEEE, 2008.
- Y. Cheng, S. Oehmcke, M. Brandt, L. Rosenthal, A. Das, A. Vrieling, S. Saatchi, F. Wagner, M. Mungabowindekwe, W. Verbruggen, C. Beier, and S. Horion. Scattered tree death contributes to substantial forest loss in California. *Nature Communications*, 15:641, 2024.
- A. Degarege, K. Fennie, D. Degarege, S. Chennupati, and P. Madhivanan. Improving socioeconomic status may reduce the burden of malaria in sub Saharan Africa: A systematic review and meta-analysis. *PLoS ONE*, 14(1), 2019.
- N. Dlamini, M. S. Hsiang, N. Ntshalintshali, D. Pindolia, R. Allen, N. Nhlabathi, J. Novotny, M.-S. Kang Dufour, A. Midekisa, R. Gosling, A. LeMenach, J. Cohen, G. Dorsey, B. Greenhouse, and S. Kunene. Low-Quality Housing Is Associated With Increased Risk of Malaria Infection: A National Population-Based Study From the Low Transmission Setting of Swaziland. *Open Forum Infectious Diseases*, 4(2):ofx071, 2017.
- M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88:303–338, 2010.
- E. Frank and M. Hall. A simple approach to ordinal classification. In *European Conference on Machine Learning (ECML)*, pages 145–156. Springer, 2001.
- N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka. Polygonal building extraction by frame field learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5891–5900, 2021.
- A. K. Githeko, S. W. Lindsay, U. E. Confalonieri, and J. A. Patz. Climate change and vector-borne diseases: a regional analysis. *Bulletin of the World Health Organization*, 78(9):1136 – 1147, 2000.
- B. Gram-Hansen, P. Helber, I. Varatharajan, F. Azam, A. Coca-Castro, V. Kopackova, and P. Bilinski. Mapping informal settlements in developing countries using machine learning and low resolution multi-spectral data. In *AAAI/ACM Conference on AI, Ethics, and Society*, 2019.
- A. Hatamizadeh, D. Sengupta, and D. Terzopoulos. End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery. In *European Conference on Computer Vision (ECCV)*, page 730–746. Springer, 2020. ISBN 978-3-030-58609-6.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- P. Helber, B. Gram-Hansen, I. Varatharajan, F. Azam, A. Coca-Castro, V. Kopackova, and P. Bilinski. Generating material maps to map informal settlements. In *NeurIPS 2018 Workshop on Machine Learning for the Developing World*, 2018.
- P. Hu, B. Shuai, J. Liu, and G. Wang. Deep level sets for salient object detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2300–2309, 2017.
- E. Jatta, M. Jawara, J. Bradley, D. Jeffries, B. Kandeh, J. B. Knudsen, A. L. Wilson, M. Pinder, U. D’Alessandro, and S. W. Lindsay. How house design affects malaria mosquito density, temperature, and relative humidity: an experimental study in rural Gambia. *The Lancet Planetary Health*, 2(11):e498–e508, 2018.
- G. Jocher, A. Chaurasia, and J. Qiu. Ultralytics YOLO. <https://github.com/ultralytics/ultralytics>, 2023. Accessed 20/03/2024.
- T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

- I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
- M. Mugabowindekwe, M. Brandt, J. Chave, F. Reiner, D. L. Skole, A. Kariyaa, C. Igel, P. Hiernaux, P. Ciais, O. Mertz, X. Tong, S. Li, G. Rwanyiziri, T. Dushimiyimana, A. Ndoli, V. Uwizeyimana, J.-P. Barnekow Lillesø, F. Gieseke, C. J. Tucker, S. Saatchi, and R. Fensholt. Nation-wide mapping of tree-level aboveground carbon stocks in Rwanda. *Nature Climate Change*, pages 91–97, 2022.
- A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.
- OpenAerialMap. OpenAerialMap. <https://openaerialmap.org/>, 2024. Accessed on 02/19/2024.
- M. Oquab, T. Darct, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINoV2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.
- C. Persello, R. Hänsch, G. Vivone, K. Chen, Z. Yan, D. Tang, H. Huang, M. Schmitt, and X. Sun. 2023 IEEE GRSS data fusion contest: Large-scale fine-grained building classification for semantic urban reconstruction [technical committees]. *IEEE Geoscience and Remote Sensing Magazine*, 11(1):94–97, 2023.
- O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.
- W. Sirko, S. Kashubin, M. Ritter, A. Annkah, Y. S. E. Bouchareb, Y. Dauphin, D. Keysers, M. Neumann, M. Cisse, and J. Quinn. Continental-scale building detection from high resolution satellite imagery. *arXiv preprint arXiv:2107.12283*, 2021.
- P. J. Soille and M. M. Ansoult. Automated basin delineation from digital elevation models using mathematical morphology. *Signal Processing*, 20(2):171–182, 1990.
- L. S. Tusting, M. Ippolito, B. Willey, I. Kleinschmidt, G. Dorsey, R. Gosling, and S. Lindsay. The evidence for improving housing to reduce malaria: A systematic review and meta-analysis. *Malaria Journal*, 14, 12 2015.
- L. S. Tusting, C. Bottomley, H. Gibson, I. Kleinschmidt, A. J. Tatem, S. W. Lindsay, and P. W. Gething. Housing improvements and malaria risk in sub-Saharan Africa: a multi-country analysis of survey data. *PLoS Medicine*, 14(2):e1002234, 2017.
- L. S. Tusting, D. Bisanzio, G. Alabaster, E. Cameron, R. E. Cibulskis, M. Davies, S. Flaxman, H. S. Gibson, J. B. T. Knudsen, C. M. Mbogo, F. O. Okumu, L. von Seidlein, D. J. Weiss, S. W. Lindsay, P. W. Gething, and S. Bhatt. Mapping changes in housing in sub-Saharan Africa from 2000 to 2015. *Nature*, 568(7752):391 – 394, 2019.
- L. S. Tusting, P. Gething, H. Gibson, B. Greenwood, J. Knudsen, S. Lindsay, and S. Bhatt. Housing and child health in sub-Saharan Africa: A cross-sectional analysis. *PLoS Medicine*, 17:e1003055, 03 2020.
- P. Venkatesan. The 2023 WHO world malaria report. *The Lancet Microbe*, 5(3):e214, 2024.
- WHO. World malaria report 2023. <https://www.who.int/publications/i/item/9789240086173>, 2023. (Accessed on 05/16/2024).

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The data and source code are provided with the submission, the results are summarized in the paper and supplementary material.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations of the data set are discussed in the paper, a distinguished limitations section has been added to the main paper; the goal of the empirical evaluation is precisely pointing out limitations and perspectives of different approaches.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: There are no new formal theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The data and source code are provided with the submission.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Data and code are made freely available, see <https://mosquito-risk.github.io/Nacala>

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Data and code are made available, the data and the splits are described in detail in the paper, most important hyperparametrs are also given in the paper/appendix

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report mean and standard deviation over five trials

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Compute resources are discussed in Section A.4 in the appendix. We report the total number of GPU hours used in the project, including experiments not discussed in the paper. Given the large amount of different experiments, we do not discuss each experimental setup individually.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We could not identify any ethical issues; we also refer to the attached data sheet

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes, we discuss the importance of roof type analysis for malaria prevention in Section 2.1.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We do not see a high risk of misuse; we also refer to the attached data sheet

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer:[Yes]

Justification: We are the owners of the data set

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The data set is very well described; we regard the data set as the main asset of the paper. The code is in a public GitHub repository; it is of OK quality but currently not yet documented in great detail.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: the paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: the paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Details on Models and Training Procedure

A.1 U-Net

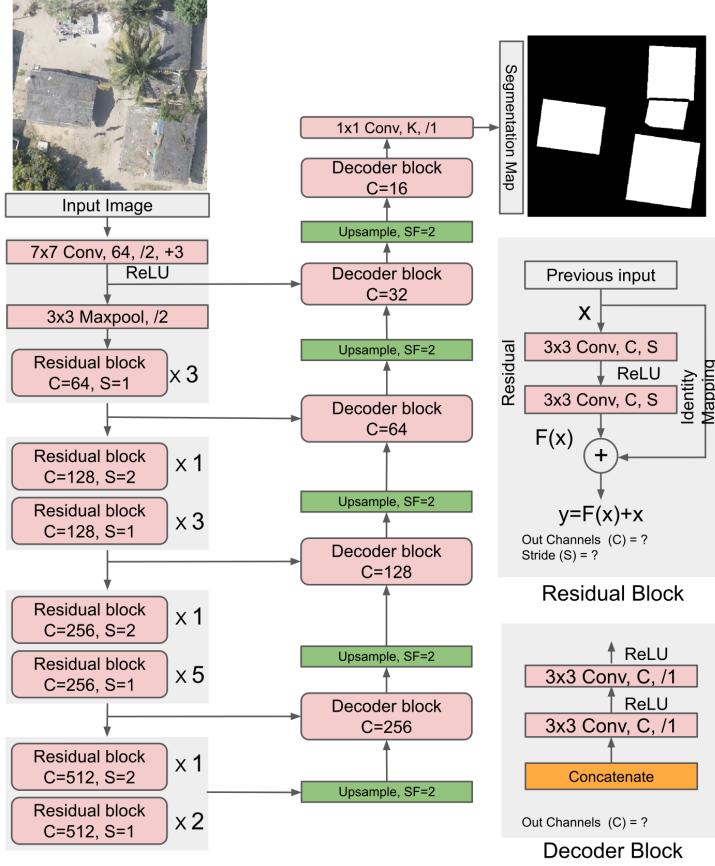


Figure A.5: Basic U-Net architecture

The basic U-Net architecture we used is shown in Figure A.5.

During training, the loss of each background pixel x is multiplicatively weighted by $w(x)$ defined as

$$w(x) = w_0 \cdot \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right) \quad (\text{A.1})$$

following Ronneberger et al. (2015). Here, $d_1(x)$ denotes the distance to the border of the nearest segment, and $d_2(x)$ is the distance to the border of the second nearest segment. We set $w_0 = 10$ and $\sigma = 5$ according to Ronneberger et al. (2015).

During training, we modified the target masks to ensure that $d_1(x) + d_2(x) \geq n_{\text{gap}} = 7$ for each background pixel x before we computed the weights $w(x)$.

A.2 Deep Ordinal Watershed U-Nets

We considered a stripped down version of the DOW U-Net proposed by Cheng et al. (2024) and set the number of elevation levels to $n_{\text{lev}} = 2$. The architecture of the resulting DOW network is depicted in Figure 2, which extends the basic U-Net architecture shown in Figure A.5. In contrast to the original U-Net, the DOW model has two heads. One is predicting the object's area, while the other predicts its interior. The interior is defined by removing pixels within a 10-pixel distance from the border of the building segment. Each head comprises a convolutional layer, batch normalization, ReLU activation, and finally a pointwise convolutional layer with outputs equal to the number of

classes. While the first head had filters of size 3×3 in its first convolutional layer, the second head for the interior used 64 filters. The class label of an object was derived from the second head and if the interior is not predicted in the case of small objects, the output from the first head is considered for the class.

We compared this DOW variant, referred to as U-Net_{DOW}, to the original DOW with several elevation levels, in which the levels are added to the standard U-Net architecture (Figure A.5) simply by increasing the number of output masks. We considered $n_{\text{lev}} = 6$ discrete height levels and accordingly refer to the model as U-Net_{DOW-6}. The pixel margin n_{pix} for each height level was determined experimentally by testing $n_{\text{pix}} \in \{1, 3, 5, 7, 9, 11, 13, 15\}$ on validation data, leading to $n_{\text{pix}} = 5$ for U-Net_{DOW-6}. An experimental comparison of U-Net_{DOW} and U-Net_{DOW-6} can be found in the extended results in Section B in the appendix.

A.3 Segmentation and Classification Using DINOV2

The segmentation architecture, which we refer to as DINOV2, is illustrated in Figure A.6. We refer to the DOW variant of DINOV2 as DINOV2_{DOW}. The same two-head implementation as in U-Net_{DOW}, is used for DINOV2_{DOW}.

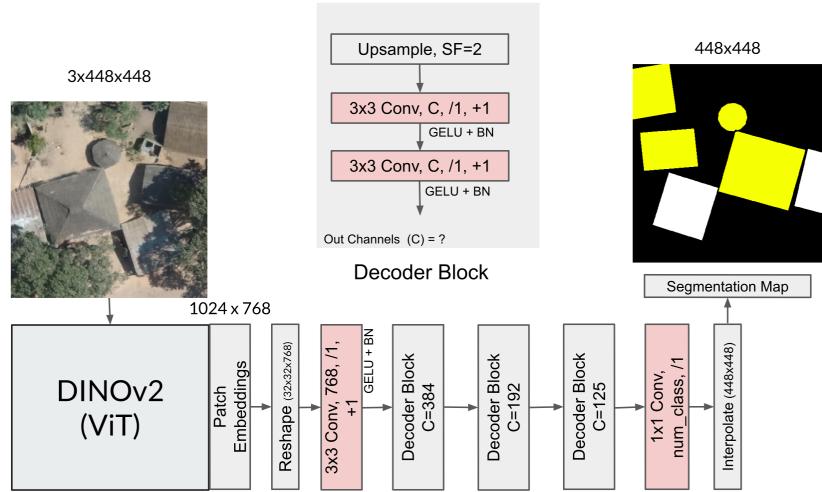


Figure A.6: DINOV2 Architecture

A.4 Compute resources

All experiments were conducted on AMD MI250X GPUs with 64 GB VRAM provided by LUMI². A total of 8550 GPU hours were used for the project, including preliminary experiments not included in the paper. The computation time for training semantic segmentation model was approximately 20 hours for 300 epochs when the entire data were loaded to GPU memory.

B Additional Results

B.1 Detailed Results for Different Roof Materials

Additional results on $\mathcal{D}_{\text{test}}$ are presented in Table B.2 and Table B.3. The tables report the IoU scores for the individual roof material classes. They also show the true positive rates TP_s , in addition to the AP_{50} the AP_{50-95} . The AP_{50-95} is defined as the mean AP over IoU thresholds from 50 % to 95 % with an interval of 5 %. The mean of AP_{50-95} over all classes is mAP_{50-95} . TP_s are the number of segments that overlap with ground truth segments with a minimum IoU of 0.5, we used this metric to assess the counting of buildings.

²<https://lumi-supercomputer.eu>

Beyond the performance metrics already discussed, we have included the results for U-Net_{DOW}-6 as described in Section A.2 in the appendix, showing that the two DOW architectures perform on par.

The corresponding results on \mathcal{D}_{ext} are given in Table B.4 and Table B.5. The mean IoU in Table B.4, and mAP₅₀ and mAP₅₀₋₉₅ in Table B.5 estimated on only four classes as there are no asbestos buildings in \mathcal{D}_{ext} . Also, there are only two buildings of concrete found in \mathcal{D}_{ext} and these two buildings were not identified from any of the experimental models, so results for the concrete class were not added to both tables.

Table B.2: Pixel-level accuracies on $\mathcal{D}_{\text{test}}$. IoU refers to the IoU computed on the binary outputs, where the predictions of multi-class models were binarized. mIoU⁵ refers to the macro average of the IoUs for the individual classes. The subscript *Multi* indicates the end-to-end setting.

Model Name	IoU-Score of each class						
	Metal Sheet	Thatch	Asbestos	Concrete	No Roof	mIoU ⁵	IoU
YOLOv8	0.807 ± 0.003	0.852 ± 0.038	0.450 ± 0.023	0.250 ± 0.027	0.480 ± 0.034	0.568 ± 0.015	0.866 ± 0.012
DINOv2	0.804 ± 0.003	0.854 ± 0.003	0.349 ± 0.010	0.196 ± 0.004	0.608 ± 0.013	0.562 ± 0.003	0.883 ± 0.002
DINOv2 _{DOW}	0.810 ± 0.003	0.867 ± 0.001	0.348 ± 0.009	0.188 ± 0.015	0.613 ± 0.005	0.565 ± 0.004	0.884 ± 0.001
U-Net	0.813 ± 0.009	0.881 ± 0.002	0.408 ± 0.012	0.171 ± 0.021	0.577 ± 0.073	0.570 ± 0.016	0.895 ± 0.003
U-Net _{DOW}	0.824 ± 0.005	0.879 ± 0.010	0.384 ± 0.042	0.174 ± 0.010	0.623 ± 0.028	0.577 ± 0.009	0.895 ± 0.002
U-Net _{DOW} -6	0.824 ± 0.006	0.887 ± 0.002	0.424 ± 0.055	0.160 ± 0.026	0.591 ± 0.057	0.577 ± 0.011	0.888 ± 0.009
YOLOv8 _{Multi}	0.750 ± 0.030	0.824 ± 0.004	0.405 ± 0.021	0.223 ± 0.059	0.549 ± 0.026	0.550 ± 0.017	0.824 ± 0.023
DINOv2 _{Multi}	0.821 ± 0.003	0.862 ± 0.002	0.490 ± 0.026	0.682 ± 0.031	0.640 ± 0.014	0.699 ± 0.012	0.880 ± 0.000
DINOv2 _{DOW-Multi}	0.839 ± 0.002	0.870 ± 0.002	0.542 ± 0.009	0.773 ± 0.009	0.649 ± 0.015	0.734 ± 0.006	0.885 ± 0.001
U-Net _{Multi}	0.819 ± 0.012	0.880 ± 0.004	0.514 ± 0.025	0.306 ± 0.091	0.650 ± 0.029	0.634 ± 0.024	0.879 ± 0.012
U-Net _{DOW-Multi}	0.827 ± 0.011	0.887 ± 0.002	0.511 ± 0.044	0.290 ± 0.105	0.636 ± 0.013	0.630 ± 0.026	0.889 ± 0.009

B.2 Performance of Different Classifiers

In the two-stage approach, we used a classifier based on DINOv2 features, as described in Section 3.3 and illustrated in Figure 3. The input representation was fixed and was processed by standard classification algorithms. We compared linear probing based on logistic regression with L_2 -regularization and k-nearest neighbours (kNN) classification trained on our data. For evaluating the classifiers and tuning their hyperparameters, we combined the training and validation data and performed 10-fold cross-validation (CV) with F1-score as performance metric. The best CV results gave logistic regression with L_2 -regularization, and this model was used for all subsequent two-stage experiments, see Table B.2.

We also performed an ablation study to show the importance of the masking and the upsampling in our architecture shown in Figure 3. The results are also depicted in Table B.2. When we omitted the masking and considered all features, the results got considerably worse. If we omitted the upsampling of the DINOv2 output and downsampled the masks instead, the performance also slightly dropped.

Table B.3: Object-level accuracy on $\mathcal{D}_{\text{test}}$. We report the AP for each roof type, and mAP_{50} and mAP_{50-95} are macro averages over the roof types. The rightmost three columns give the results when we discard the roof type information and just consider building detection. The TP_s columns count true positives, where TP_s are the number of objects that overlap with ground truth objects with a minimum IoU of 0.5. The total number of ground truth objects in the $\mathcal{D}_{\text{test}}$ is 2527.

Model Name	AP ₅₀ of each class					average over classes			ignoring roof type		
	Metal Sheet	Thatch	Asbestos	Concrete	No Roof	mAP ₅₀	mAP ₅₀₋₉₅	TP _s	AP ₅₀	AP ₅₀₋₉₅	TP _s
YOLOv8	0.841 ± 0.003	0.945 ± 0.008	0.505 ± 0.032	0.542 ± 0.055	0.661 ± 0.026	0.698 ± 0.018	0.548 ± 0.010	2262.2 ± 7.386	0.941 ± 0.003	0.798 ± 0.002	2405.0 ± 5.514
DINOv2	0.807 ± 0.005	0.885 ± 0.006	0.470 ± 0.011	0.579 ± 0.025	0.673 ± 0.015	0.683 ± 0.008	0.531 ± 0.005	2135.6 ± 7.761	0.882 ± 0.004	0.733 ± 0.003	2261.6 ± 9.351
DINOv2 _{DOW}	0.852 ± 0.002	0.940 ± 0.001	0.517 ± 0.016	0.600 ± 0.028	0.715 ± 0.006	0.725 ± 0.004	0.573 ± 0.008	2238.4 ± 5.238	0.930 ± 0.005	0.781 ± 0.005	2376.2 ± 7.194
U-Net	0.826 ± 0.005	0.924 ± 0.006	0.499 ± 0.016	0.511 ± 0.042	0.679 ± 0.015	0.688 ± 0.014	0.578 ± 0.014	2191.2 ± 11.25	0.910 ± 0.005	0.797 ± 0.003	2323.0 ± 6.033
U-Net _{DOW}	0.855 ± 0.005	0.946 ± 0.005	0.545 ± 0.019	0.596 ± 0.049	0.707 ± 0.012	0.730 ± 0.011	0.614 ± 0.007	2249.4 ± 4.128	0.935 ± 0.001	0.819 ± 0.003	2383.6 ± 5.314
U-Net _{DOW-6}	0.851 ± 0.006	0.943 ± 0.004	0.551 ± 0.011	0.587 ± 0.049	0.687 ± 0.022	0.724 ± 0.007	0.606 ± 0.005	2243.2 ± 3.487	0.929 ± 0.004	0.818 ± 0.002	2374.4 ± 5.783
YOLOv8 _{Multi}	0.849 ± 0.006	0.923 ± 0.007	0.467 ± 0.035	0.070 ± 0.027	0.676 ± 0.020	0.597 ± 0.007	0.481 ± 0.003	2195.6 ± 15.383	0.910 ± 0.005	0.751 ± 0.007	2328.2 ± 9.988
DINOv2 _{Multi}	0.869 ± 0.003	0.923 ± 0.005	0.474 ± 0.043	0.508 ± 0.100	0.669 ± 0.033	0.689 ± 0.025	0.512 ± 0.021	2231.6 ± 2.332	0.899 ± 0.003	0.733 ± 0.004	2311.4 ± 2.728
DINOv2 _{DOW-Multi}	0.888 ± 0.004	0.937 ± 0.004	0.536 ± 0.018	0.504 ± 0.056	0.647 ± 0.028	0.702 ± 0.013	0.558 ± 0.011	2270.4 ± 9.308	0.918 ± 0.003	0.766 ± 0.002	2366.0 ± 7.266
U-Net _{Multi}	0.883 ± 0.009	0.940 ± 0.010	0.531 ± 0.022	0.498 ± 0.051	0.728 ± 0.040	0.716 ± 0.018	0.603 ± 0.011	2262.4 ± 6.119	0.924 ± 0.004	0.797 ± 0.007	2358.4 ± 9.091
U-Net _{DOW-Multi}	0.864 ± 0.001	0.918 ± 0.006	0.533 ± 0.024	0.537 ± 0.048	0.702 ± 0.018	0.711 ± 0.010	0.609 ± 0.012	2216.2 ± 7.305	0.903 ± 0.004	0.786 ± 0.003	2308.0 ± 5.044

Table B.4: Pixel-level accuracies on \mathcal{D}_{ext} . IoU refers to the IoU computed on the binary outputs, where the predictions of multi-class models were binarized. mIoU⁵ refers to the macro average of the IoUs for the individual classes. The subscript *Multi* indicates the end-to-end setting.

Model Name	IoU-Score of each class				
	Metal Sheet	Thatch	No Roof	IoU (Mean)	IoU (Binary)
YOLOv8	0.888 ± 0.003	0.879 ± 0.003	0.516 ± 0.011	0.761 ± 0.006	0.896 ± 0.002
DINOv2	0.867 ± 0.014	0.853 ± 0.004	0.523 ± 0.022	0.747 ± 0.011	0.905 ± 0.000
DINOv2 _{DOW}	0.891 ± 0.003	0.880 ± 0.002	0.560 ± 0.018	0.777 ± 0.007	0.905 ± 0.001
U-Net	0.896 ± 0.005	0.883 ± 0.005	0.463 ± 0.017	0.748 ± 0.007	0.909 ± 0.001
U-Net _{DOW}	0.905 ± 0.002	0.895 ± 0.003	0.493 ± 0.018	0.764 ± 0.006	0.911 ± 0.002
U-Net _{DOW-6}	0.900 ± 0.008	0.889 ± 0.002	0.452 ± 0.031	0.747 ± 0.009	0.902 ± 0.003
YOLOv8 _{Multi}	0.890 ± 0.006	0.860 ± 0.006	0.606 ± 0.019	0.785 ± 0.006	0.885 ± 0.002
DINOv2 _{Multi}	0.905 ± 0.002	0.875 ± 0.004	0.674 ± 0.018	0.818 ± 0.005	0.899 ± 0.002
DINOv2 _{DOW-Multi}	0.912 ± 0.001	0.881 ± 0.002	0.663 ± 0.017	0.875 ± 0.010	0.902 ± 0.001
U-Net _{Multi}	0.913 ± 0.005	0.884 ± 0.003	0.617 ± 0.061	0.805 ± 0.020	0.903 ± 0.002
U-Net _{DOW-Multi}	0.921 ± 0.001	0.888 ± 0.002	0.613 ± 0.033	0.807 ± 0.011	0.909 ± 0.002

Table B.5: Object-level accuracies on \mathcal{D}_{ext} . We report the AP for each roof type, and mAP_{50} and mAP_{50-95} are macro averages over the classes. The rightmost three columns give the results when we discard the roof type information and just consider building detection. TP_s are the number of objects that overlap with ground truth objects with a minimum IoU of 0.5. The total number of ground truth objects in the \mathcal{D}_{ext} is 1541.

Model Name	AP ₅₀ of each class			Objects with Classes			Only Building Objects		
	Metal Sheet	Thatch	No Roof	mAP ₅₀	mAP ₅₀₋₉₅	TP _s	AP ₅₀	AP ₅₀₋₉₅	TP _s
YOLOv8	0.928 ± 0.001	0.947 ± 0.000	0.661 ± 0.023	0.846 ± 0.008	0.428 ± 0.002	1447.2 ± 4.534	0.963 ± 0.005	0.838 ± 0.002	1493.8 ± 3.826
DINOv2 _{Multi}	0.898 ± 0.004	0.885 ± 0.009	0.635 ± 0.024	0.484 ± 0.005	0.393 ± 0.005	1381.6 ± 7.172	0.919 ± 0.005	0.786 ± 0.006	1428.8 ± 7.305
DINOv2 _{DOW-Multi}	0.932 ± 0.002	0.942 ± 0.005	0.681 ± 0.020	0.852 ± 0.007	0.423 ± 0.003	1441.8 ± 4.400	0.956 ± 0.001	0.828 ± 0.003	1486.2 ± 1.939
U-Net	0.915 ± 0.006	0.921 ± 0.006	0.520 ± 0.027	0.590 ± 0.006	0.407 ± 0.004	1399.4 ± 4.758	0.929 ± 0.000	0.836 ± 0.002	1438.4 ± 4.499
U-Net _{DOW}	0.932 ± 0.003	0.946 ± 0.004	0.559 ± 0.027	0.812 ± 0.008	0.528 ± 0.003	1429.0 ± 6.229	0.947 ± 0.004	0.858 ± 0.004	1468.6 ± 6.499
U-Net _{DOW-6}	0.935 ± 0.001	0.940 ± 0.004	0.509 ± 0.022	0.795 ± 0.008	0.518 ± 0.005	1421.0 ± 3.688	0.939 ± 0.000	0.851 ± 0.004	1458.8 ± 4.118
YOLOv8 _{Multi}	0.949 ± 0.004	0.934 ± 0.007	0.664 ± 0.044	0.849 ± 0.015	0.423 ± 0.008	1446.0 ± 4.899	0.948 ± 0.003	0.808 ± 0.005	1477.2 ± 3.655
DINOv2 _{Multi}	0.955 ± 0.004	0.935 ± 0.007	0.749 ± 0.030	0.880 ± 0.011	0.539 ± 0.005	1454.4 ± 3.878	0.946 ± 0.001	0.801 ± 0.003	1468.8 ± 2.926
DINOv2 _{DOW-Multi}	0.956 ± 0.001	0.943 ± 0.005	0.727 ± 0.026	0.875 ± 0.010	0.521 ± 0.047	1460.8 ± 3.868	0.950 ± 0.005	0.820 ± 0.002	1478.8 ± 3.311
U-Net _{Multi}	0.956 ± 0.004	0.926 ± 0.008	0.651 ± 0.107	0.844 ± 0.039	0.548 ± 0.017	1439.0 ± 16.358	0.943 ± 0.010	0.838 ± 0.006	1463.6 ± 13.063
U-Net _{DOW-Multi}	0.951 ± 0.004	0.920 ± 0.006	0.632 ± 0.029	0.834 ± 0.010	0.458 ± 0.005	1426.8 ± 5.418	0.947 ± 0.001	0.854 ± 0.004	1472.6 ± 2.417

Table B.6: Cross-validation accuracies on combined training and validation data of k-nearest neighbour classification (kNN) and logistic regression applied to the DINOv2 features. The baseline is the architecture depicted in Figure 3, *w/o mask* refers to omitting the masking and averaging the DINOv2 features across the whole input patch, and *w/o upsampling* did not upsample the DINOv2 features but downsampled the building mask instead.

Classifier	F1-Score					
	baseline		w/o mask		w/o upsampling	
	Mean	Std	Mean	Std	Mean	Std
Logistic Regression	0.770	0.063	0.573	0.077	0.768	0.067
kNN	0.734	0.045	0.389	0.029	0.733	0.051