ENV-540
Large Rocks Detection Dataset
Group 4

Image Processing for Earth Observation
Autumn 2024

Prof. DEVIS Tuia
Supervised by : Thien-Anh Nguyen, Manon Béchaz, Giacomo May, Emanuele Dalsasso

DARMON Samuel
CLEVORN Jan
HOMINAL Sven

Table of contents

1 GitHub and Code of the project

The code for this project is available on the following public GitHub repository : IPEO_Project_Group_4.

2 Topic and challenges

Nowadays, technology is being used to make certain time-consuming tasks in our daily lives easier and more automated. This has its application in machine learning, and in particular applied to the domain of image processing for earth observation and object detection.

The subject of this project is large rocks detection in Valais, Ticino and Graubunden. The Federal Office for Topography swisstopo still proceeds to manual annotations of all large rocks in Switzerland to produce topographic maps. The aim here is to observe what could be done with recent automatic methods, this would save a great deal of time. A dataset is provided and contains annotations, the aim will be to detect large rocks (5 over 5 meters) in Switzerland based on high-resolution RGB images [1] and DSM [2]. Different sources of data, machine learning approaches and more recent object detection models are used.

Automatization of large rock detection in satellite imagery through machine learning methods faces several significant challenges, mainly due to the complex nature of remote sensing data and the variability of objects in the environment. One of the main difficulties is dealing with objects of varying sizes and shapes. Rocks can appear in many different sizes and configurations, which makes them harder to detect across a wide landscape [3]. According to Li et al. (2023), accurate object detection, such as rocks, requires obtaining both precise bounding frame around the object and its precise features [4].

Another challenge is the high inter-class similarity and cluttered backgrounds present in remote sensing. The rocks might blend with other features like trees, soil, or debris, which all can be of important size, and this confounds detection models. It is subject also to light variations, where shadows or other effects may obscure the appearance of rocks [5]. According to Zhang et al. (2023), the YOLO series algorithms have advanced remote sensing image object detection but face challenges with detecting small, distant, or low-contrast objects and adapting to variations in data across regions, seasons, and weather conditions [6]. Furthermore, the scale of objects in satellite imagery varies a lot, depending on the resolution and the sensor's altitude. Models may struggle with detecting both small and large rocks in the same image, especially when they occur across a big area. Automated interpretation of remote sensing data is challenging due to its extensive spatial coverage and complex image backgrounds, according to Liu et al. (2023) [7]. To address these challenges, it requires advanced solutions like multi-scale detection methods, data augmentation techniques and combination of RGB images with digital surface models (DSMs). By integrating these complementary methods, i.e. RGB imagery, DSMs, hillshade rasters, and actual deep learning models like YOLOv8, this project achieves a framework for detecting and analyzing large rocks in Valais.

3 Method

3.1 Dataset

One dataset with 3 modalities is given in the scope of project. This "Large Rocks Dataset" is based on 63 hand-annotated tiles by the Federal Office for Topography (swisstopo) distributed across the Swiss Alps. The annotations comprehensively cover all rocks measuring at least 5x5 meters, with a geographic split to minimize biases related to specific areas or spatial characteristics. While the original data had smaller spatial resolutions, all tiles have a standardised resolution of 50 cm, and images are segmented into tiles of 640x640 pixels with a 25% overlap. The dataset contains 2'625 annotated large rocks.

- swissIMAGE_50cm_patches : RGB Images with 50cm resolution (originally 10 cm).
- SwissSURFACE3D_patches : DSM based on LiDAR data. Contains actual elevation values in meters.
- swissSURFACE3D_hillshade_patches : terrain and landscape views using brightness values derived from the DSM data, generated with QGIS with the hillshade function (Azimut 0, Vertical angle 0)

Digital Surface Models (DSMs) capture detailed terrain elevation data. Hillshade rasters, derived from DSMs, simulate light and shadow on the terrain, enhancing surface visibility. Hillshade rasters emphasize topographic anomalies, such as the edges of rocks or depressions, that might otherwise go unnoticed in raw imagery. This method is widely used for improving interpretability in elevation-based analysis.

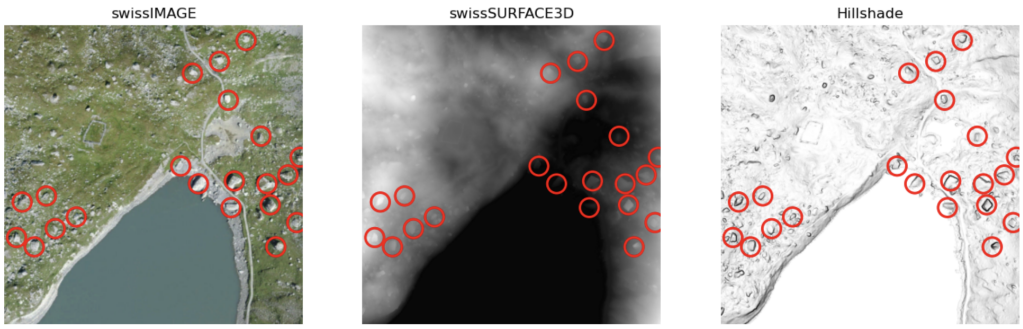


Figure 1: Aerial image from swissIMAGE, digital surface model from swissSURFACE3D and the corresponding hillshade raster with large rocks annotations from swisstopo annotators.

3.2 Model

In this project, YOLOv8 is used, a state-of-the-art object detection model known for its real-time detection capabilities and high accuracy. YOLOv8 introduces several architectural enhancements over its predecessors, including an advanced backbone and neck architecture for improved feature extraction, and an anchor-free detection head that simplifies the model while enhancing accuracy [8].

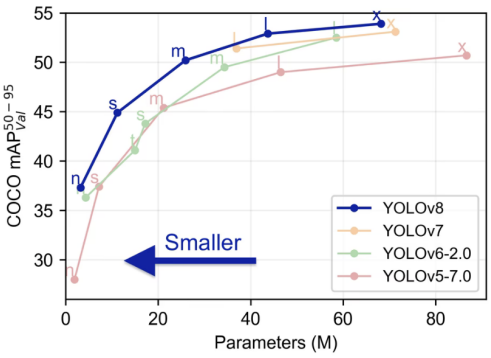


Figure 2: Ultralytics YOLOv8 Model comparison to predecessors. It offers better performance for a lower number of parameters.

While YOLOv8 is among the latest iterations in the YOLO series, newer versions like YOLOv9 and YOLOv10 have been introduced, each bringing further advancements in speed and accuracy. In our experiments however, various YOLO models have been evaluated and it has been found that YOLOv8x provided the optimal balance between performance and computational efficiency for our specific application. There were no noticeable performance improvements using YOLOv10x. Figure 2 shows the different YOLOv8 models. The `x` model, while having many more parameters, is far more accurate than YOLOv8n. This was also seen in practice, section 4 will showcase this.

3.3 Workflow

3.3.1 Train/ Test/ Split

After visualizing the json dataset, a number of training (65%) and test (35%) images were initially noticeable. This is suboptimal, especially noting that the training set currently only comprises 640 images. This was modified to an appropriate split of about 80-10-10 (train-test-val) by splitting the original test set into test and validation respectively.

3.3.2 Bounding box conversion

Then, round bounding circles have been converted to bounding boxes taking the relative location of the object, bounding box size, and image size as inputs. YOLO requires each bounding box to be represented by one line, formatted as follows: `class center_x center_y width height`. Some experimentation with the bounding box width and height showed that 30 was the ideal width for these boxes. Figure 4 shows an example of the red bounding circles and blue bounding boxes, clearly showing the conversion worked.

3.3.3 Removing Duplicates in Labels and Checking Image Size

Duplicates are removed in each label file, that ensures unique entries. The number of images is checked to see if it matches the number of annotation entries in the JSON file.

3.3.4 Hillshade-RGB Combination

Experiments showed that combining RGB and Hillshade data by replacing the second RGB-band by the Hillshade raster, which is normalised between [0,255] to have similar values as the RGB values.. This is described

and visualised in section 4.2. Once this operation is done, we save the YOLO dataset, organized into `images/` and `labels/` directories, each containing `train/, val/`, and `test/` subfolders.

3.3.5 Training

A YAML configuration file to train the YOLO model has been created, specifying dataset paths, class names and augmentation parameters. This is also where the YOLO parameters have been chosen, including epochs, batch size, image size, optimizer, and augmentation settings are set. Notably the epochs, batch size, image size, optimizer and augmentation settings.

3.4 Feature Extraction

Before running any CNN based models such as YOLO, manual feature extraction was tested. Manual feature extraction would have allowed to focus more on specific, domain relevant features, that might not be captured by YOLO directly. Moreover, manual feature extraction would require significant less computational resources for training and inference. Initially, 9 different channels were plotted in order to investigate if any easy segmentation could be done with soly the RGB images. The motivation was that gray rocks could be potentially segmented from green forests using different thresholds. Seen from the Figure 3, no segmentation was able to be done.

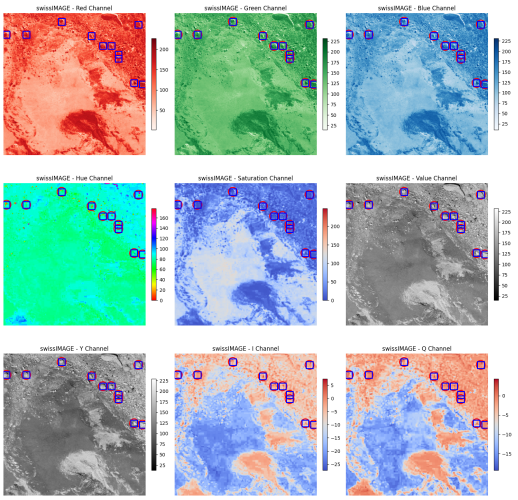


Figure 3: Feature Extraction Test using Color Space

Some further feature extraction was tested. Rocks usually have more irregeular, non-continious shapes, compared to trees or other vegetation, which was the motivation to use edge detection methods to find boundaries of rocks, after the boundaries would have been found, a threshold to the surroudned area could have been applied to distinguish small and large rocks. Unfortunately, none of the algorithms worked, potentially because the quality of images wasn't good enough. Lastly, some testing was done with texture descriptions, also without success. Using spectral information, such as vegetation indexes like NDVI could have helped to exclude vegetation and focus on non-vegetated areas, which are likely to include rocks, though this data was not provided.

3.5 Data Augmentation

Data augmentation is a technique used to artificially increase the size of a dataset by applying various transformations to the original data, enhancing model generalization and robustness. Although image augmentation is directly performed by YOLO, it was of essence to ensure the augmentations make sense, getting an overview of the different types of augmentations performed. Not all augmentations automatically performed by YOLO make sense. Transformations such as flipping, cropping and rotation seem to be applicable for all type of images. However, changes to illumination, brightness and color jitter may only be suitable for RGB images. The next subsections briefly explain the augmentations used in the best performing model.

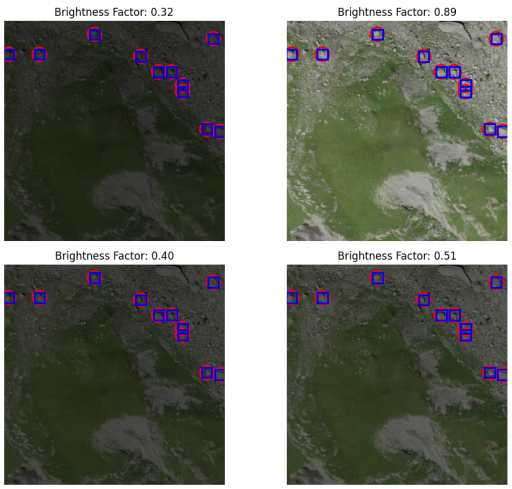


Figure 4: Brightness Augmentation for swissIMAGE_50cm_patches

In our training, several augmentation parameters beyond their default settings to enhance model performance are customized. Notably, `hsv_s` (Hue-Saturation-Value Saturation) is set to 0, deviating from the default value of 0.7, in order to control color saturation variations. We also disabled certain augmentations by setting `degrees`, `scale`, `shear`, and `perspective` to 0, thereby reducing geometric transformations during

training. Additionally, we applied the `randaugment` policy for `auto_augment` and set `erasing` to 0.4, introducing random erasing with a 40% probability to improve model robustness. These deliberate adjustments were made to tailor the augmentation process to the specific characteristics of our dataset, potentially enhancing the model's generalization capabilities.

3.5.1 Brightness Augmentation

One type of augmentation that was tested was brightness augmentation, which deemed to be be particularly useful for rock detection in images as it helps simulate varying lighting conditions that occur naturally in outdoor environments. Rocks can be exposed to direct sunlight, shadows, or diffuse light, leading to significant differences in their appearance in terms of brightness and contrast. Brightness augmentation was only applied to the swissIMAGE_50cm_patches images, and not to the DSM images (swissSURFACE3D_hillshade_patches and SwissSURFACE3D_patches). Figure 4 test different brightness factors.

3.5.2 Translation

Translation, or shifting the image along the x and y axes, can be valuable in training YOLOv8 for large rock detection from aerial imagery. It helps the model generalize better by seeing objects in different positions within the image. This is especially useful for the given case, where objects may not always be centered or in predictable locations, allowing YOLOv8 to detect rocks at various positions across the frame.

3.5.3 Flipping

Flipping, or mirroring the image helped since rocks in aerial imagery can appear in various orientations, thus helps the model generalize by presenting flipped versions of the same image. This simulates real-world scenarios where rocks may be captured from different angles or perspectives. By training the model with both upright and flipped images, it becamse more robust in detecting rocks regardless of their orientation

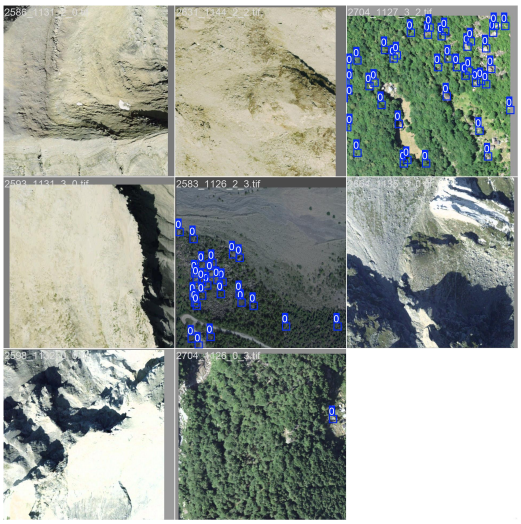


Figure 5: Random batch with all used Augmentations (Batch Size = 8)

3.5.4 Erasing

Erasing is a data augmentation technique that randomly removes parts of the image by setting pixel values to a neutral color, this can be seen in the random training batch in figure 5. This technique deemed to be successfily as it simulates occlusion or missing data. Rocks can be partially covered by vegetation, shadows, or other objects, leading to incomplete visibility.

3.6 Modalities

In order to incorporate more modalities than just RGB Images, we decided to incorporate hillshade values, as these are clearly more discriminative for detecting large rocks, as can be seen in figure 1. Since YOLOv8 only accepts three bands, we decided to replace one of the RGB bands by the normalized hillshade values. As seen in figure 9, replacing the second band by hillshade is what keeps the image visually discriminative, while adding more information.

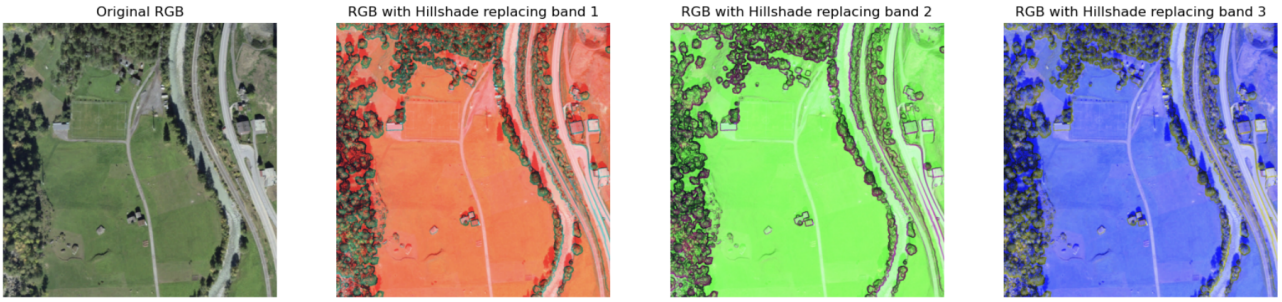


Figure 6: RGB with Hillshade replacing color bands

3.7 Metrics

YOLO provides several performance metrics that evaluate both the accuracy and efficiency of the model. These metrics are essential for understanding the model's performance in detecting and localizing large rocks, notably:

- IoU (Intersection over Union): It quantifies the overlap between predicted bounding boxes and ground truth boxes. A higher IoU indicates better localization accuracy.
- mAP (mean Average Precision): This metric combines classification and localization performance. Typically reported as mAP50, mean Average Precision at an IoU threshold of 0.50, assessing detection quality with moderate overlap requirements and mAP50-95 as the mean Average Precision averaged over IoU thresholds from 0.50 to 0.95 in increments of 0.05, providing a more comprehensive evaluation.
- Precision: It represents the proportion of correctly identified objects among all detections, indicating the model's ability to minimize false positives.
- Recall: It reflects the proportion of ground truth objects that were successfully detected, measuring the model's ability to avoid false negatives.

3.8 SCITAS

For this project and for the training run, we utilized the Scientific IT and Application Support (SCITAS) facilities at EPFL, which provide advanced computational resources and High-Performance Computing (HPC), leveraged for our Large Rocks Detection Project. Utilizing SCITAS allowed for efficient processing of large parameter numbers and complex computations involved in training the YOLOv8x model and its 68,229,648 parameters. Using SCITAS' Tesla V100-PCIE-32GB, 32501MiB GPU allowed a training run of 150 epochs to be run in just under one hour.

3.9 Flowchart of the methods

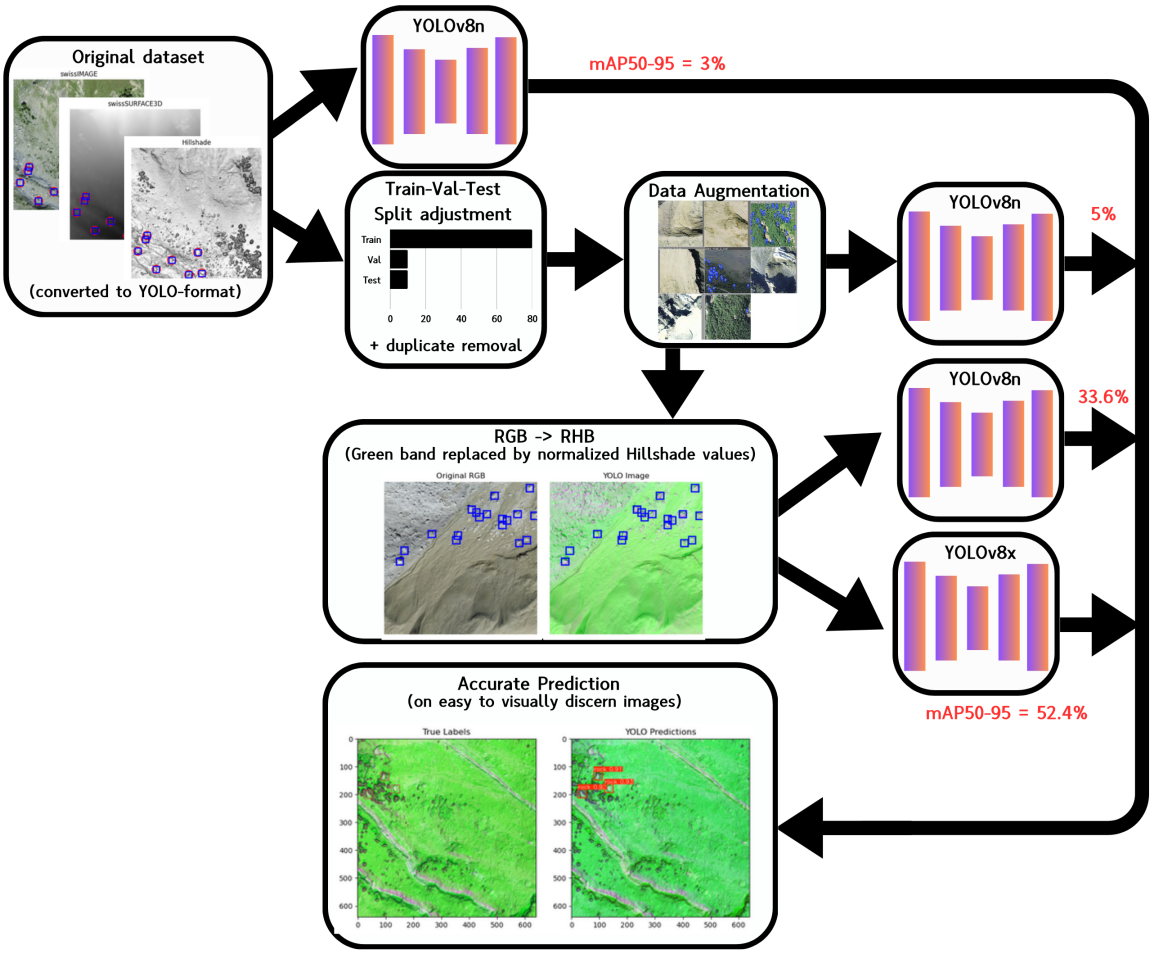


Figure 7: Flowchart of the methods

4 Results

4.1 YOLOv8n - RGB

To establish a baseline, models were trained using only RGB images. Initially, some testing was also done with using preset YOLOv8 network configurations and customized one. The rationale for this was based on the fact that the dataset on which the YOLOv8n model was trained (the COCO dataset) does not include a "Rock" class [8], suggesting a potential mismatch. This did not show any major differences.

Table 1 below presents the validation results of the baseline models. The training process involved either enabling or disabling all YOLO-inherent augmentations. Table 1 shows that the use of YOLO-inherent augmentations alone did not result in substantial improvements compared to models without augmentations. One possible reason for this could be that some of the augmentations designed for general use may not be suitable for a large rock dataset, as explained in Section 3.5. Hence, a custom set of augmentations was created to better address the specific characteristics of the rock dataset, which may explain the lack of significant improvement with the default YOLO augmentations.

YOLO model	Channels	Augmentation	mAP50	mAP50-95
v8n	RGB	No	0.15	0.03
v8n	RGB	Yes	0.2	0.05

Table 1: Validation set metrics during training for different model configurations.

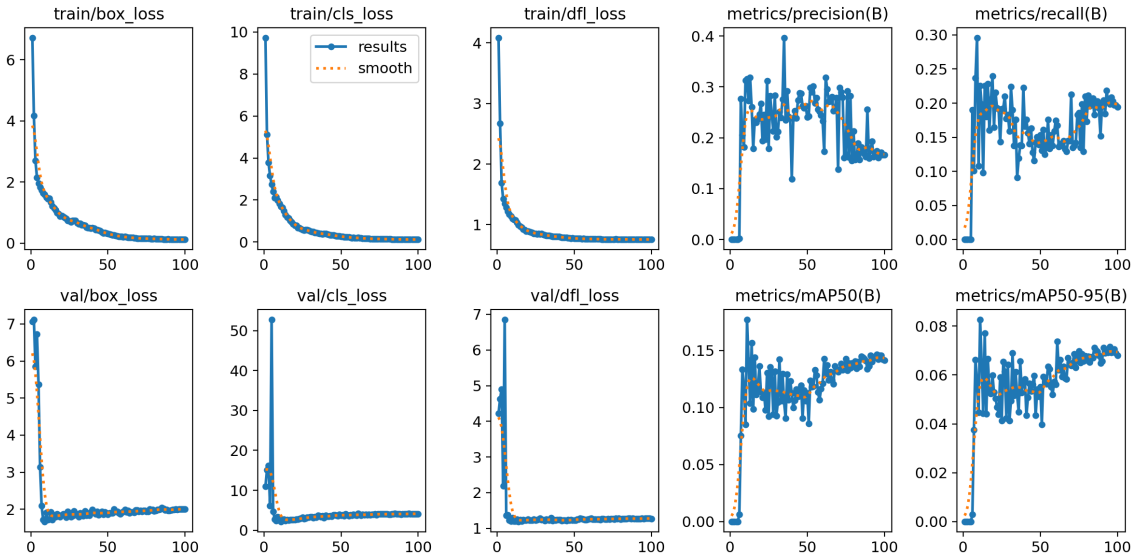


Figure 8: Training metrics for 100 epochs of training YOLOv8n on non-augmented RGB images

As seen from the above Table 1 and Figure 8, the obtained reults are not satisfactory and even show some signs of overfitting.

4.2 YOLOv8n - RHB - custom augmentation

After having gotten unsatisfactory results with only RGB images, even with default YOLO augmentation, we incorporated Hillshade data. Since YOLO is not made to accept more than 3 bands, we replaced one of the original bands with normalized hillshade data, as described in Section 3.6. Band 2 replaced by hillshade is the best choice and gives the highest "net" detection of the object. Figure 9 illustrates a comparison between the original RGB images, the RGB image with band 2 replaced, and the `.tif` file displaying the RGB with the hillshade as an overlay. A training batch with augmentation values as defined in section 3.5 is illustrated in figure 10.

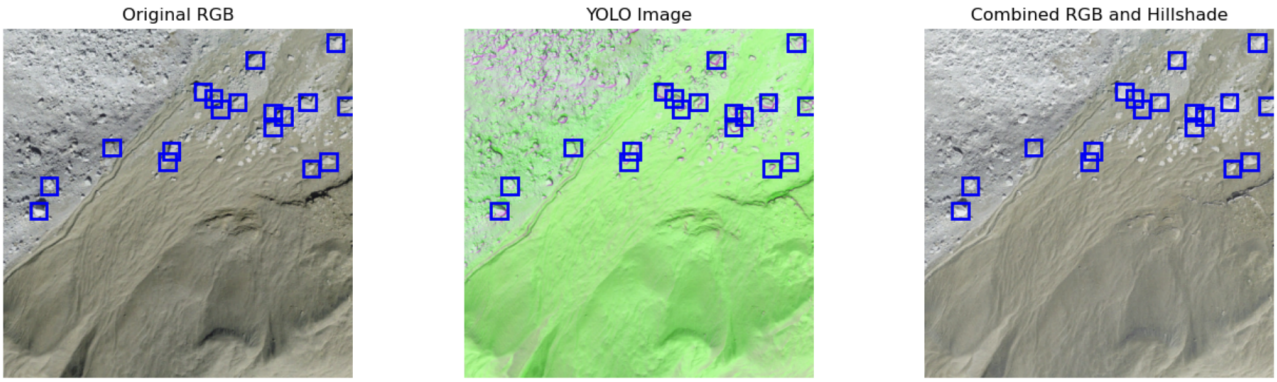


Figure 9: RGB with Hillshade replacing color bands

Initial tests with YOLOv8n gave much more satisfactory results, with mAP50 passing 55%.

4.3 YOLOv8x - RHB - custom augmentation

Knowing this was the right track, we upgraded the model to YOLOv8x, the largest YOLOv8 model, passing from 3.5M to 68.7M parameters. This is how we obtained our best model. It's noticeable here that precision is high, reaching about 80% over a hundred epochs, the recall reaches 55 % and the mean average precision at an IoU threshold of 0.50 reaches 65% on validation data. Overall the results are good when using RGB images combined here with hillshade layers. This can be seen in figure 11

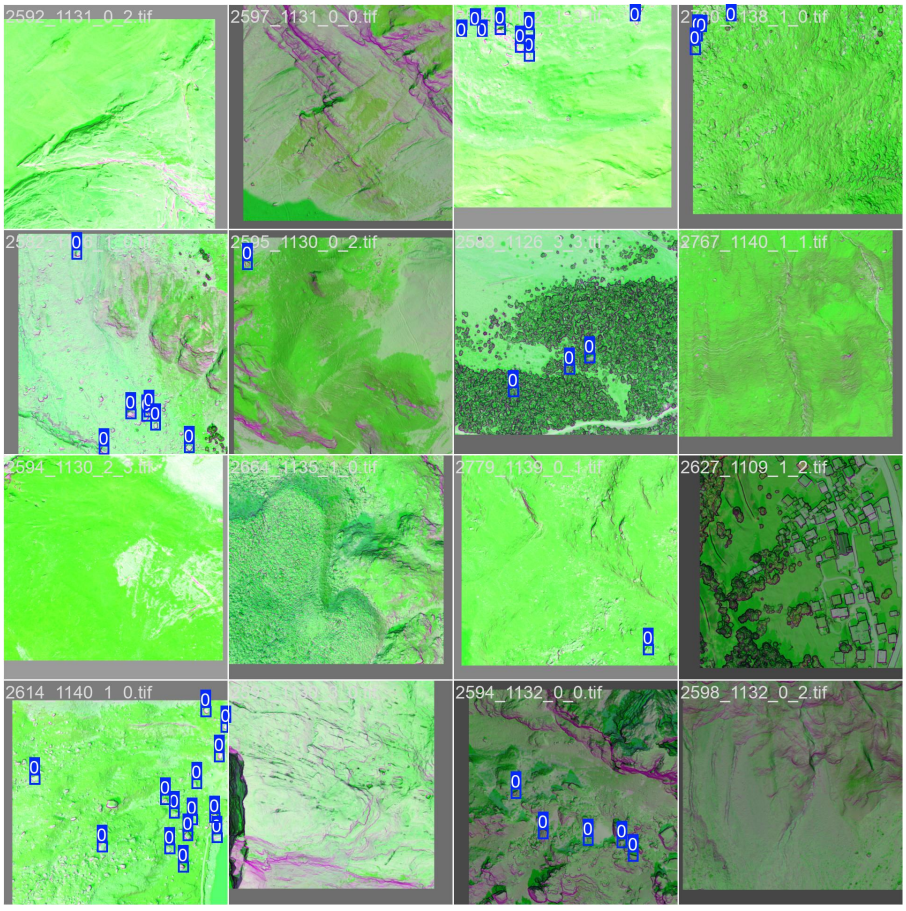


Figure 10: Training batch post-augmentation, with band 2 replaced by normalized hillshade values.

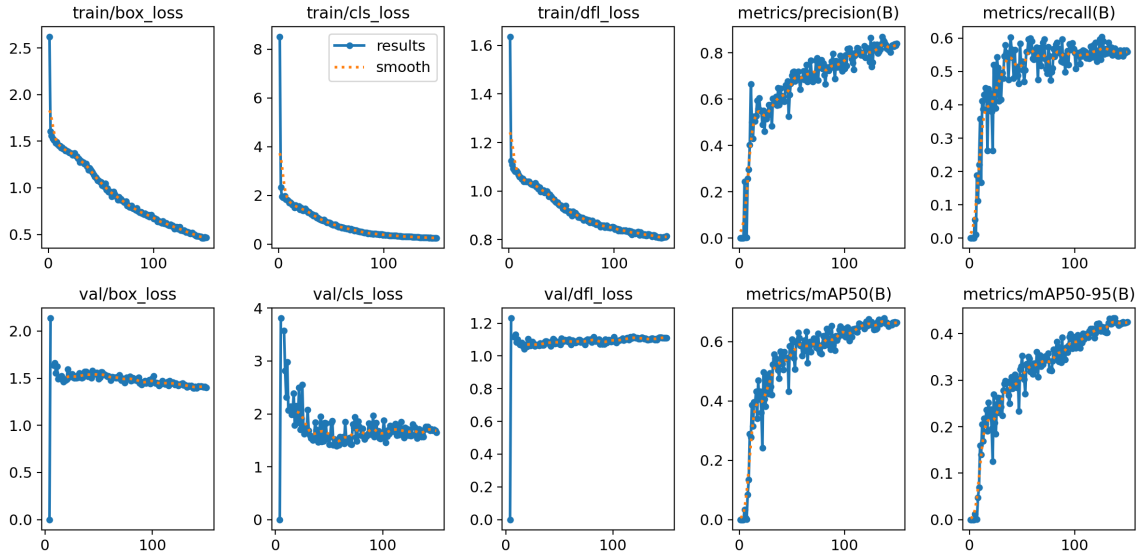
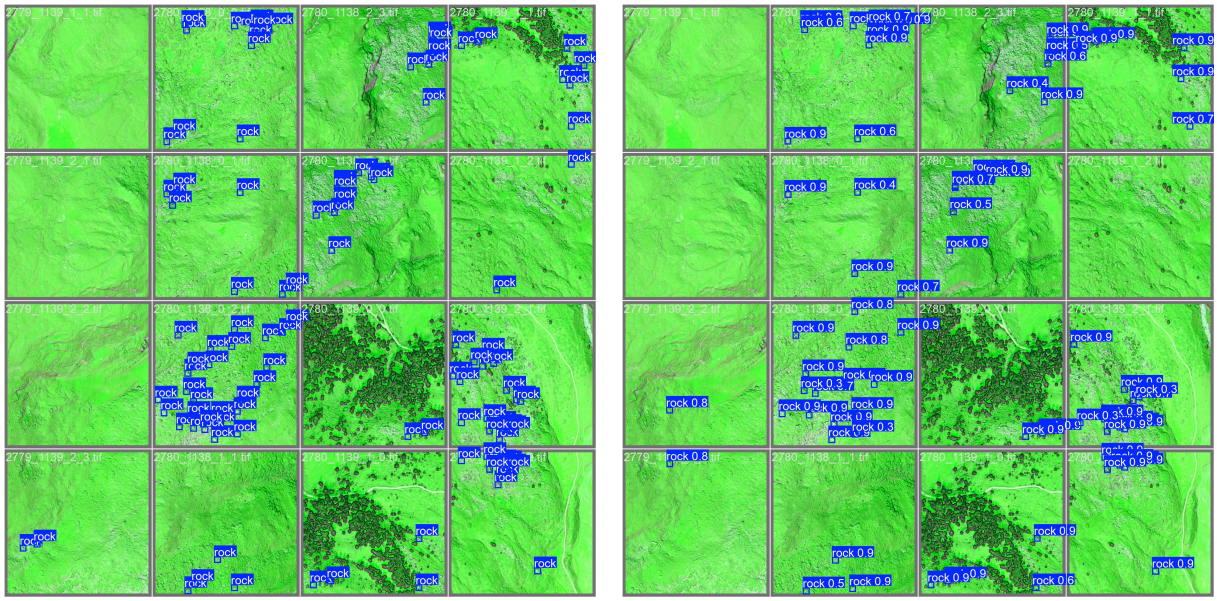


Figure 11: Training metrics: RHB (Red Hillshade Blue) - YOLOv8x, with custom augmentations

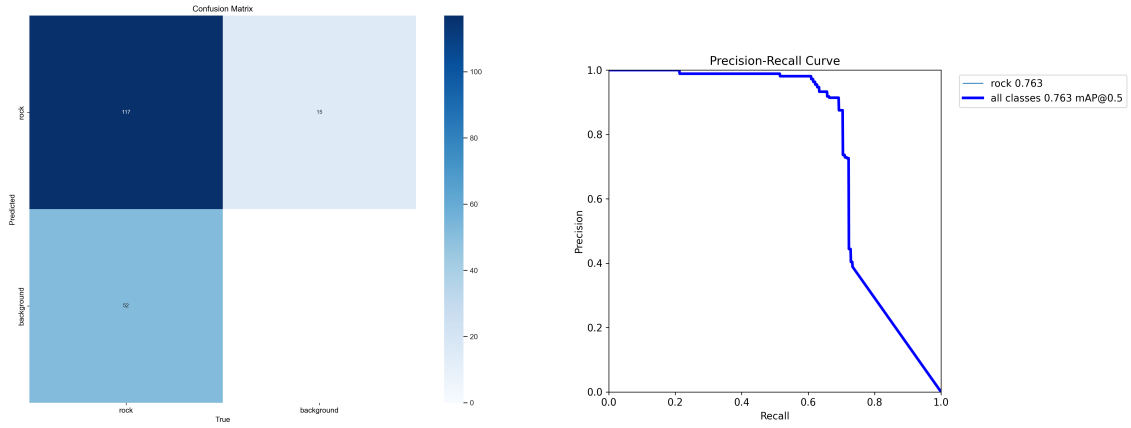


(a) Labels

(b) Predictions

Figure 12: Label and predictions for an example batch

As an example of model predictions as seen on Figure 12, the predictions for rocks detection are accurate. With high precision but lower recall, the model returns here less of the large rocks, but most of its predicted labels are correct when compared to the actual labels.



(a) Confusion Matrix

(b) Precision-Recall curve

Figure 13: Confusion Metrics and P-R curve on the test set

The curve shown in figure 13b computed on the test set (111 never seen before images) shows a trade-off between precision (how many of the detected large rocks are correct) and recall (how many of the actual large rocks are detected). Here, a steep drop in precision as recall increases suggests that the model starts detecting more false positives as it tries to capture all true positives. Inference is conducted using the test set to evaluate the model's performance under real-world conditions. Specific samples from the test set are used to analyze detection in scenarios such as cluttered backgrounds, varied lighting, and terrain variability.

4.4 Overall Results

The overall metrics over our training and test runs are summarized below, in table 2. It is notable to see at which point increasing the size of the training set and custom augmentations for this task have an impact on precision.

YOLO model	Channels	Augmentation	split	mAP50	mAP50-95
v8n	RHB	default	65-35-0	0.311	0.135
v8n	RHB	default	75-15-10	0.449	0.203
v8n	RHB	custom	75-15-10	0.587	0.336
v8x	RHB	default	75-15-10	0.502	0.325
v8x	RHB	custom	80-10-10	0.692	0.524

Table 2: Validation set metrics during training for different model configurations.

For the best model overall, a validation run on the test set gives a mAP50-95 of 0.574 and a mAP50 of 0.762, which are even better values than on the validation set, which assures us that our model is not overfitting.

5 Discussion & Challenges

The application of YOLOv8 models (YOLOv8n and YOLOv8X) for detecting large rocks in high-resolution satellite imagery demonstrates the potential of deep learning in automating labor-intensive tasks like manual annotation for topographic mapping. Integrating different data sources, such as combining RGB images with hillshade overlays, significantly improved object detection performance, particularly in complex terrains. The hillshade layer enhanced the model's ability to distinguish large rocks despite cluttered backgrounds.

In terms of model performance, the YOLOv8X model, trained with combined datasets (RGB images and hillshade) and advanced augmentation techniques and pretrained parameters of the model, achieved a precision of 80%, a recall of 55% and an mAP50 of 60%. These metrics reflect the model's capacity for accurate detection and localization of large rocks. High precision signifies that the model is correct for its predited labels as compared to the true ones, even though not all of them are detected.

Despite these advancements, the project for large rocks detection faced significant challenges related to both data and model limitations. One major challenge was the variability in rock features, such as size, shape, and texture, which made it difficult for the model to generalize across diverse instances. Rocks often appeared in

cluttered backgrounds, including vegetation, shadows, and other natural elements, which could obscure their features or mimic their appearance, leading to false positives or missed detections. Additionally, the dataset size, although standardized to a 50 cm resolution, was relatively small, with only 2,625 annotated rocks. This limited dataset size increased the risk of overfitting and reduced the model's robustness.

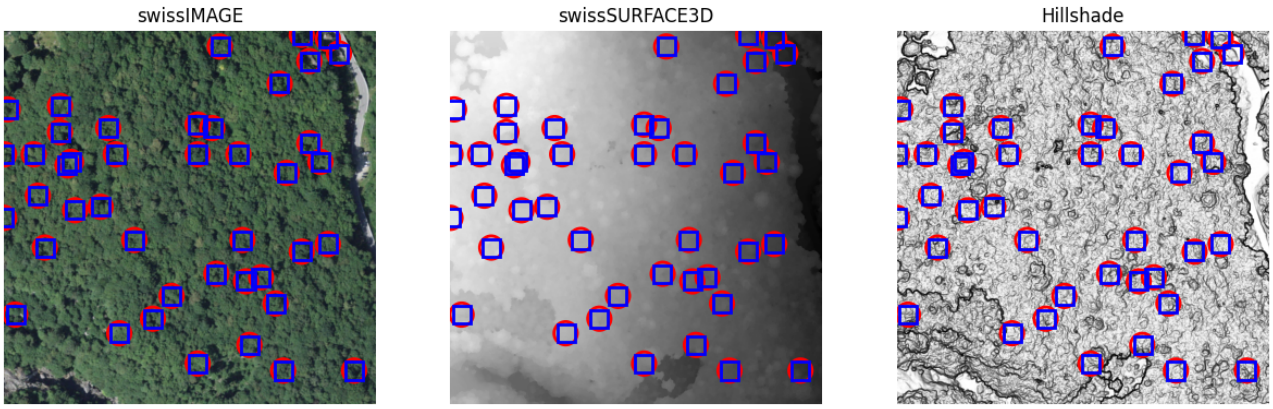


Figure 14: Example Annotation - Even manual extraction through visual inspection alone would be challenging

Another key challenge was related to annotation quality. Manual annotations by swissTOPO annotators introduced the potential for human errors, including mislabeling and inconsistencies in object boundaries. As seen on Figure 14, a manual feature extraction on such images would be very challenging for a human. These issues were compounded by the ambiguity in defining precise object boundaries, particularly for rocks blending with their surroundings. The conversion of annotations to compatible YOLO formats also posed risks of inaccuracies if not carefully managed. Moreover, geographic bias in the dataset, despite efforts to minimize it through spatial splitting, remained a concern. This bias could limit the model's ability to perform well in regions outside the training areas (here in Valais).

Also, some factors added further complexity. The model's performance was challenged by terrain variability, lighting changes, and seasonal effects, such as snow-covered rocks or dense vegetation, which introduced inconsistencies in the representation of large rocks. For example, rocks obscured by snow or hidden within dense foliage during certain seasons were less likely to be detected accurately. These environmental variations highlight the need for models that are more adaptive to changing conditions.

To address these limitations, incorporation of more diverse transformations during training, such as simulating seasonal variations, could improve generalization and prediction accuracy. Enhancing the dataset with additional annotated samples, especially from varied geographic regions, would reduce bias and improve robustness. A cross-regional validation approach, where the model is tested on datasets from different areas, could further assess its adaptability and reliability. Additionally, refining preprocessing steps, improving annotation standardization, and leveraging advanced augmentation techniques remain essential to overcome the challenges and enhance the model's overall performance.

6 Conclusion

This project has highlighted the effectiveness of deep learning methods, specifically the YOLOv8 model, in automating the detection of large rocks in Switzerland using high-resolution satellite imagery. While significant progress was made, challenges such as the variability in rock characteristics, complex backgrounds, and limited dataset size underscore the difficulty of remote sensing tasks.

Also, integrating RGB imagery with hillshade overlays and applying advanced data augmentation techniques greatly enhanced model preicision and effectiveness to detect large rocks. However, overcoming challenges like human-based annotations, potential geographic bias and adaptability to environmental changes over the seasons remains essential and difficult to overcome. These insights emphasize the importance of developing robust approaches to improve the scalability and reliability of automated detection systems, especially in the context of large rocks detection.

References

[1] Admin.ch. (2024) Swissimage 10 cm. [Online]. Available: https://www.swisstopo.admin.ch/en/orthoimage-swissimage-10

[2] admin.ch. (2024) swisssurface3d. [Online]. Available: https://www.swisstopo.admin.ch/de/hoehenmodell-swisssurface3d

[3] A. T. et al. (Feb. 2022) Automatic target detection from satellite imagery using machine learning, sensors, vol. 22, no. 3, p. 1147,. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1147

[4] Y. F. L. Wen, Y. Cheng and X. Li. (2023) A comprehensive survey of oriented object detection in remote sensing images, expert systems with applications, vol. 224, p. 119960. [Online]. Available: https://doi.org/10.1016/j.eswa.2023.119960

[5] S. V. Adekanmi Adeyinka Adegun, Jean Vincent Fonou-Dombeu and J. Odindi. (Jun. 2023) State-of-the-art deep learning methods for objects detection in remote sensing satellite images, sensors, vol. 23, no. 13, pp. 5849–5849. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1147

[6] X. B. C. Bai and K. Wu. (2023) A review: Remote sensing image object detection algorithm based on deep learning," electronics, vol. 12, no. 24, pp. 4902–4902. [Online]. Available: https://doi.org/10.3390/electronics12244902

[7] Y. Y. et al. (2023) Automated object recognition in high-resolution optical remote sensing imagery," national science review, vol. 10, no. 6. [Online]. Available: https://doi.org/10.1093/nsr/nwad122

[8] Ultralytics, "YOLOv8." [Online]. Available: https://docs.ultralytics.com/models/yolov8