Alignment and Benchmarking for Drone Vision Dataset

Mihai David
EPFL
mihai.david@epfl.ch

Baran Ozaydin
EPFL
baran.ozaydin@epfl.ch

Abstract

This study deals with semantic segmentation of high-resolution drone imagery and the performance of deep convolutional neural networks (CNNs) on this task, by using automatically generated drone labels. As human annotation of the training data is a very expensive process, especially in the case of semantic segmentation where each pixel in the image is assigned a specific class, this paper addresses the feasibility of using generated drone labels instead of human annotated drone labels in the context of supervised semantic segmentation. By analysing a varied suite of algorithms for obtaining drone labels from already existing satellite labels, such as descriptor-based image matching, dense pixel correspondence, feature matching and warping, this work shows a promising direction in overcoming the time-consuming process of manually annotating the unmanned aerial vehicle (UAV) imagery with the purpose of reducing the time and costs of creating datasets required by various supervised deep learning tasks.

1. Introduction

In the past decade, aerial imagery was mainly represented by satellite images that were made publicly available through satellite image providers, such as Google Earth, USGS Earth Explorer, Copernicus Open Access Hub, NOAA or Sentinel Open Access Hub. Even though these providers are usually exposing an API for querying captures of the Earth, obtaining clear, high-quality satellite images of a fixed size, from a specific GPS location is not an easy task. Some impediments to this task involve the malfunctioning of the satellites that these providers are using, the tiles misalignment, processing errors, local cloudiness, low spatial resolution or lightning conditions. However, there is a lot of ongoing research on satellite images datasets for deep learning tasks such as scene classification [33], semantic segmentation [23], object detection [35] or instance segmentation [16]. This is mainly due to the impressive performance of the convolutional neural networks and their robustness to noise in the training labels, making them the state-of-the-art approach in majority of tasks. Moreover, satellite labels can already be obtained from online maps in an automatic way, making the process of creating a supervised learning dataset easier, by avoiding human annotation.

In the recent years, there has been a shift from satellite imagery to drone imagery. The reason for this are the commercial drones available on the market, that are equipped with high-resolution cameras and are becoming more and more affordable. Therefore, the acquisition of aerial photographs is made simple and does not rely on third-parties anymore. However, images captured by drones also show high variations in brightness, illumination and weather conditions. Still, the biggest difference from the satellite images are the wide range of altitudes and the multiple viewing angles that the drone photos were captured at. Thus, compared to satellite images, harsh perspective effects appear on the corners of drone images, see Fig. 1. This is not happening with the satellite images as the acquisition altitude is much higher and the images are usually orthorectified, meaning that the distortions, which are induced by sensor orientation, topographical variation and the curvature of the earth are corrected in the final images. Additionally, compared to satellite images, there is no automatic pipeline to annotate the drone images, thus the need of manual annotation.

At first glance, one could use the satellite labels for the drone images. In Fig. 1, we show the corner displacements between the two types of images, by overlaying the satellite label on both drone and satellite images. Even though CNNs are robust to noise in the training labels, the semantic segmentation task requires qualitative enough labels for a good prediction. Therefore, simply using satellite labels for the drone images will prove unfeasible (Sec. 5.3) because these corner displacements highly affect the overall performance of the models. To this end, we focus on generating aligned image-label drone pairs and assess the performance of different neural networks using these samples in the context of supervised semantic segmentation. We further compare the quality of the predictions from networks using the generated labels or the human annotated labels in training. Next, the paper is divided into the following sections: re-

lated work, dataset, alignment methods, experiments and models performance, ablation studies, future work and conclusion.

1.1. Contributions

- We obtained satellite images and labels by using already available online maps, satellite images APIs and maps drawing software
- We show the need of accurate training labels when using drone images, in order to obtain a high performing model
- We assessed the performance of several image matching, dense pixel correspondence and warping methods for aligning the drone labels to the drone images
- We compared the performance of multiple neural networks for semantic segmentation using drone images and aligned (generated) drone labels with neural networks using drone images and human annotated drone labels.

2. Related work

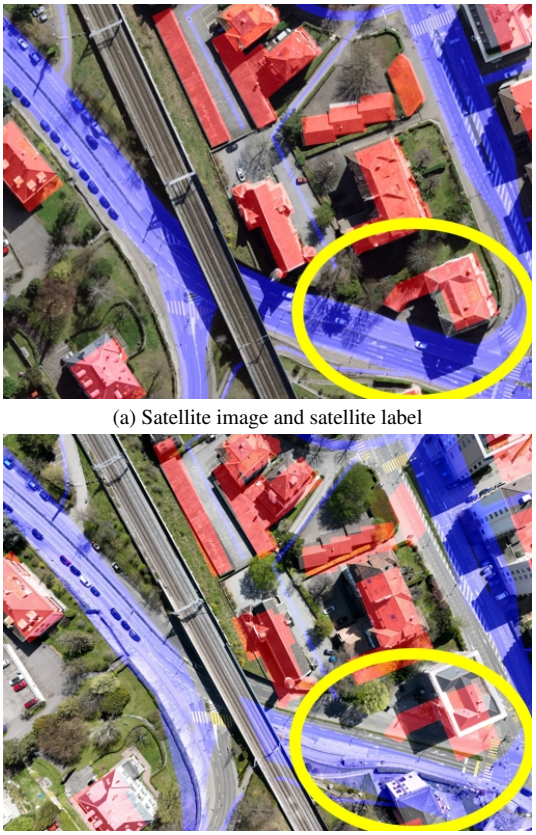
2.1. UAV datasets

As stated in Sec. 1, a shift from satellite imagery to drone imagery has happened only in the recent years. Due to this, there are relatively few drone datasets available for use in research, despite the fact that this is an extremely intriguing subject. Some of the available drone datasets are VisDrone [40] and UAVDT [12], being suitable for object detection and tracking tasks, and UAVid [26], UDD [9] and ICG Semantic Drone Dataset [14], serving for semantic segmentation task focusing on urban scenes.

2.2. Semantic Segmentation architectures

Semantic segmentation is a computer vision tasks that involves classifying each pixel in a given image within user-defined classes. The seminal study and also the first work that used CNNs for this task was Fully Convolutional Network (FCN) [25]. Following this work, many model structures followed an encoder-decoder based architecture. One of the most well-known work, especially in the biomedical sector, is UNet [30], a model that uses skip connections, fusing the information of the encoder, into the decoder, in the upsampling branch. DeepLab [7] and PSPNet [39] use Pyramid Pooling Modules with the former also adopting atrous convolutions to increase the receptive field. Most recent works leverage the power of Transformers [37, 38].

Some works demonstrate their effectiveness directly on drone datasets. For example, Kumar et al. [19] use a transformer-based encoder-decoder with a special module in the encoder branch that helps to capture low-level spatial context details about neighboring pixels in image tokens. They are using two drone datasets, mainly UAVid [26] and UDD [9]. Another work is UVid-Net [13] and it deals with UAV video semantic segmentation by using an enhanced encoder-decoder based CNN architecture.



(a) Satellite image and satellite label

(b) Drone image and satellite label

Figure 1. Corner displacement visualization between orthorectified satellite image (*a*) and drone image (*b*). Satellite label overlay on both drone and satellite images.

3. Dataset

The drone used for data acquisition is DJI Mavic 3 [1]. Multiple frames were extracted from videos taken in different Swiss cities like Morges, Neuchatel, Vevey, Montreaux and Cully. Both .JPG and .DNG files are included for each frame, as the latter includes metadata about GPS location, relative altitude and flight roll, pitch and yaw degrees. The images are further divided into 3 altitude categories: 80 meters, 120 meters and other altitude. We further define a subset of 100 images, all taken from 120 meters altitude, that will represent our main dataset used in experiments. We take into consideration 4 categories across images, mainly: background, building, road and water. The original size of the images and labels is 5280x3956 pixels.

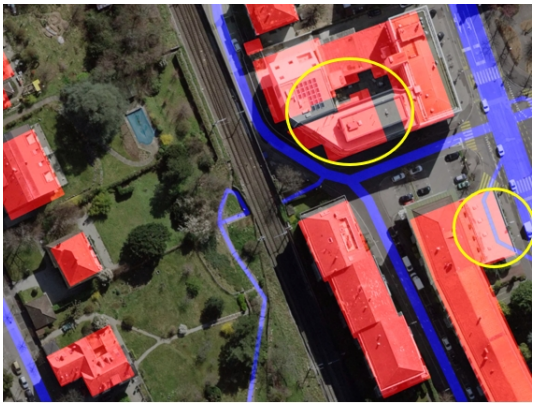


Figure 2. Satellite image with satellite label artefacts.

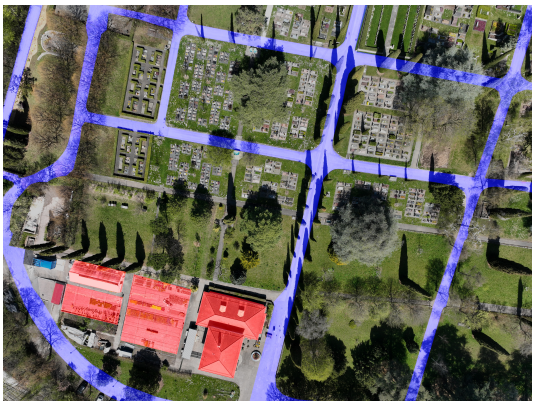
3.1. Satellite images and labels

In order to obtain the satellite images corresponding to drone images, we are using Mapbox [2] to query images based on drone GPS location and drone image size. For the satellite labels we find the corresponding raster image of the drone image by using OpenStreetMap (OSM) [4] - an open geographic database. Then, we use OverPass API [5] - an API for OSM data, and Maperitive [3] - a map drawing software that uses a custom set of drawing rules, for obtaining a categorical colormap. It is important to note that these automatically created satellite labels still present some inconsistencies with the satellite images, due to the specified drawing rules and satellite image offset and this can be seen in Fig. 2. Some challenging labeling situations are pedestrian tunnels, roads width and big areas of water. However, the obtained labels are robust enough for our further experiments.

3.2. Human annotated drone labels

Drone images were annotated by an external company. However, even by having human supervision, some labels present artefacts, especially for the road class, as a consequence of the company decision of not labeling all the types of roads found in the images, compared to the satellite labels where all types of roads are labeled. This difference can be seen in Fig. 3. We are obtaining the drone labels by aligning the automatically generated satellite labels. Therefore, our final drone labels will definitely not match the human annotated drone labels, impacting our metrics negatively, as we do not expect that the predictions obtained with models trained with annotated labels will exactly match the predictions made with models trained with generated drone labels.

(a) Satellite image and satellite label



(b) Drone image and human annotated drone label

Figure 3. Human annotated drone labels artefacts. We show the difference between automatically generated satellite labels(a) and human annotated drone labels(b) for the same area.

4. Alignment methods

In this section we present the main methods used to generate drone labels that are aligned with the drone images, starting from the already existing satellite labels.

4.1. SIFT Flow

Firstly, we focused on basic algorithms (non deep learning) for matching the satellite images with the drone images. One method, based on SIFT descriptors, is SIFT Flow [22]. This method was designed for solving dense correspondence across scenes, meaning aligning an image to its nearest neighbors in a large image corpus containing a variety of scenes. Unfortunately, it requires a very high computational power for our high-resolution images as gradients of each pixel in a patch need to be computed, and it cannot deal with high displacements in some drone-satellite image pairs, therefore we found this method unfeasible for our case.

4.2. Patch Match

Another basic matching algorithm is Patch Match [6]. It is a method for quickly finding approximate nearest-neighbor matches between image patches by a sequence of texture-based propagation and random sampling search steps. This method is more robust to high displacements between images and is much faster compared to SIFT Flow or other methods leveraging nearest-neighbor searches [18, 36]. Another advantage is that it computes a pixel-wise matching. The downside is that it is based on RGB features, being sensitive to shadows, lightning or color palette differences between drone and satellite images. The initialization of the algorithm can be adapted and the random search of patches can be constrained according to own needs. Our two initialization methods consisted of a random map and a map with prior information, more exactly the original satellite label map. This algorithm's performance is presented in Sec. 5.4.

4.3. SCOT

Semantic Correspondence as an Optimal Transport Problem (SCOT) [24] is a method for finding dense correspondence across semantically similar images. It aims at solving the many-to-one matching problem - many pixels in a source image are assigned to one target pixel, and the background matching problem - some object pixels are assigned to the background pixels. In our case, the former problem is of high interest because in aerial images we encounter many low-texture areas, therefore the high possibility of many-to-one correspondences. The background matching is not a very important issue, as the classes (road, building, water) that we are interested in the satellite and drone images are not clearly differentiated from the actual background. Moreover, the background matching solution involves class activation maps that are out of our reach, therefore we do not use this module in the implementation of the method.

The many-to-one problem is solved by global feature matching, which maximizes the total matching correlations between images. Each column in the matching matrix represents scores from all source pixels to target pixels, while each row represents matching scores from a source to all targets. Avoiding large values in a row or a column is the solution for the problem, therefore each row sum and column sum is a fixed value based on the prior distributions of pixels. Now, the one-to-one pixel correspondence relies only on the way of choosing a target pixel for each source pixel, based on the confidence matrix, for example taking the maximum value between all target pixels.

The advantage of the method is that it only uses a pretrained CNN backbone for extracting the features that the matching matrix is based on. There is no need for training and it does not require ground-truths. However, the method is computationally expensive, forcing us to work with heavy downsampled images. In Sec. 5.5 we present the performance of this method.

4.4. LoFTR and TPS

Detector-Free Local Feature Matching with Transformers (LOFTR) [34] is an image feature matching technique that uses cross attention layers in Transformer to obtain feature descriptors that are conditioned on both images. Due to several reasons like inadequate texture, repeating patterns, changing viewpoints, varying illumination, and motion blur, feature detectors may not be able to extract enough recurring interest points across images. A large receptive field in the feature extraction network is crucial but CNNs are not capable of covering such receptive fields. That is why by relying on the global receptive field and positional encoding provided by Transformer, LoFTR is able to produce good matches in low-texture areas too. Another key-point is that instead of performing feature detection, feature description and feature matching sequentially, it relies on 2 pixel-wise dense matching modules, one at coarse-level and another on fine-level that refines the good matches from the former module. LoFTR outperformed the existing state-of-the-art feature matching algorithms based on feature detectors such as SuperPoint [11] or SuperGlue [32], but also current best detector-free local feature matching algorithms such as NCN-Net [29] or DRC-Net [20].

The output of LoFTR consists of pairs of matched pixel coordinates. The drawback is that it does not perform a pixel-wise matching, therefore directly transforming the satellite labels based on satellite and drone image matching is not possible. However, having a dense set of correspondences, we follow [27] and compute Thin Plate Spline (TPS) [17] transformation parameters and apply the transformation on the satellite labels to align them with the drone images. Another drawback is that LoFTR is a supervised method that requires camera poses and depth maps for computing the ground-truth labels. As we are not able to compute ground-truths for our dataset, we cannot train our model from scratch and only use a pretrained version of LoFTR on outdoor scenes from MegaDepth dataset [21]. This method's performance is also presented in Sec. 5.5.

5. Experiments

According to our dataset definition in Sec. 3, we split the dataset into 75 images for training and 25 images for validation. An important note is that even if data was visually inspected, some small image regions in the validation set are overlapping with regions in the train set. Even though leakage of any validation data into the training data is a poor mistake, this will not impact our work as this overlap was kept constant during all our experiments.

5.1. Semantic segmentation networks

We are considering 2 semantic segmentation networks in our work, mainly DeepLabV3+ [8] and UNet [30]. They originally use pretrained backbones on ImageNet dataset [31]. DeepLabV3+ uses a ResNet50 backbone and UNet uses a FCN backbone. By making use of the provided models from Open-MMLab Segmentation computer vision system [28] we use versions of this two architectures that are further pretrained on CityScapes dataset [10], containing urban street scenes, as this is the most appropriate dataset to ours. DeepLabV3+ is pretrained on 80k iterations on image crops of size 1024x512 and UNet is pretrained for 160k iterations also on crops of 1024x512. Both networks will be trained on crops of 1025x512 pixels and will use multi-scale flip augmentations at test time.

5.2. Metrics and validation

Even though we train the aforementioned networks on 4 classes (background, building, road and water areas), because the water class is heavily under-represented in the training and validation data and the background class is not of high interest in the saliency maps, we only report the mIoU (mean Intersection Over Union) metric for road and building classes. The best model is saved according to the highest mIoU during the validation step.

5.3. The need of accurate drone labels

As suggested in the introduction (Sec. 1) we firstly investigate how does a network perform when using drone images and satellite labels compared to a network that uses satellite images and satellite labels. In Tab. 1 the model type convention corresponds to the following format: *train image - train label - test image - test label*, and we will stick with this notation in the rest of our work. At validation time, the images are re-scaled back to the original image size, therefore the IoU is computed on image sizes of 5280x3956. By analyzing the results we conclude that we definitely need more accurate labels for the drone images, as simply using satellite labels cannot overcome the displacements that appear in the drone images due to various altitudes and viewing angles, thus the model is not able to learn properly. In Fig. 4 we show how the two models perform on drone images. Even if the sat-sat model was trained with satellite images and satellite labels, it performs better on drone images and have a better generalisation capability than drone-sat model. This is thanks to the accurate labels used in training.

5.4. Patch Match performance

In Fig. 5 we present visual results of the Patch Match algorithm for the two initialization methods that we used. When initializing the algorithm with a map with prior information, more exactly the satellite label map, the results

(a) Drone-sat model (b) Sat-sat model

Figure 4. Predictions on drone images.

Model type	Class IoU (val)		mIoU (val)
	Building	Road	
DeepLabV3+ drone-sat-drone-sat	25.38	10.50	17.94
DeepLabV3+ sat-sat-sat-sat	**66.83**	**37.99**	**52.41**

Table 1. Comparison between model performances trained with drone images and satellite labels and model trained with satellite images and satellite labels.

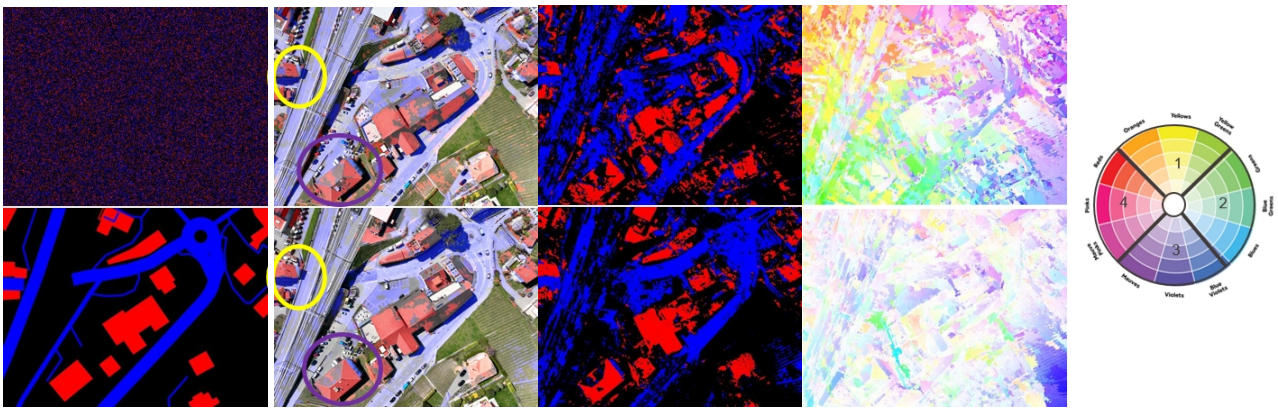
Image size	Method	Class IoU (val)		mIoU (val)
		Building	Road	
1056x792	Patch Match	42.47	17.92	30.19
1056x792	DeepLabV3+ sat-sat-drone-drone(h)	**57.38**	**29.48**	**43.43**

Table 2. Comparison of generated drone labels with human annotated drone labels. *sat-sat-drone-drone(h)* means that prediction was made on drone images using *sat-sat* model and the mIoU was computed against human drone labels.

are better. This can also be seen in the displacement maps in the last column that show in different colors from which positions were the pixels taken in the satellite label in order to generate the drone label. In Tab. 2 we use Patch Match algorithm with the satellite map initialization. On the first row we compute the mIoU between drone labels generated, for the validation set, by Patch Match and the human annotated drone labels. On the second row we use the sat-sat model to make prediction on drone images, and compare them with human annotated drone labels - *drone(h)*. This is also our baseline in term of generated predictions. Due to computational constraints we only used images of size 1056x792 (keeping the original aspect ratio) for Patch Match. For a fair comparison we also used the sat-sat model to predict on drone images of the same reduced size. As Patch Match performed worse that our baseline model, we discarded it as a possible method for generating drone labels.

5.5. Other alignment methods

In Tab. 3 we further compare the performance of SCOT and LoFTR+TPS. On the first row there is the mIoU between predictions on drone images using sat-sat model and

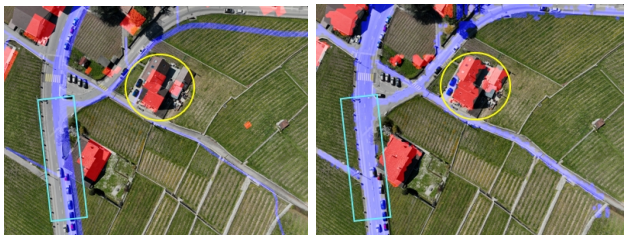


(a) Initialization method maps (b) Drone images and generated drone labels (c) Generated drone labels (d) Displacements visualization for generated drone labels

Figure 5. Patch Match visual results. Each row corresponds to a different initialization method: *random initialization* at the top and *satellite label initialization* at the bottom. From left to right, in the first column there are the 2 initialization maps. In the second column there are the generated drone labels for each initialization method together with the drone images. In the third column there are only the generated labels for a more clear view. In the last column there are the patch correspondence displacement visualization. The lighter the maps, the better.

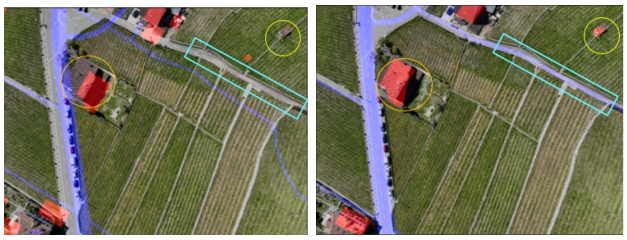
human annotated drone labels, as in Tab. 2, but the full image size of 5280x3956 is kept. The next two lines correspond to drone labels generated by SCOT and LoFTR+TPS, respectively. There is a clear improvement, as both methods outperformed the baseline for both classes with at least 5% IoU. For SCOT, we used an image size of 480x360 at inference time, upsampling it to the original image size in the end, and used a maximum-value protocol when computing the source-target correspondences from the confidence matrix. A visual example of this algorithm performance can be seen in Fig. 6. For LoFTR+TPS we used an image size of 640x480 at inference time (default size) and we set the matching threshold in the coarse-level module to 0.85. Also, we selected the 1000 most confident matches after the fine-level module, due to computational constraints. We then applied TPS on the resulted set of correspondences and upsampled the generated label to the original size of 5280x3956. Some visual results of this method can be seen in Fig. 7. On the last row of the table there is the target in terms of performance. *drone-drone(h)-drone-drone(h)* means that we used a model trained with drone images and drone labels and then compared the predictions on drone images, by this model, with human annotated drone labels. As stated in Sec. 3.2, we do not expect to reach the performance of this model in terms of mIoU as our generated drone labels come from satellite labels which initially differ from the human annotated drone labels. However, the mIoU is a good enough metric to compare different methods for generating the drone labels.

Next, we are interested to see if training semantic segmentation networks with the already generated drone labels could further improve the results. Thus, in Tab. 4 we present



(a) Satellite label overlayed on drone image (b) SCOT generated label overlayed on drone image

Figure 6. SCOT performance. Image *b* shows how SCOT generated labels are aligning very well with the drone image.



(a) Satellite label overlayed on drone image (b) LoFTR aligned label overlayed on drone image

Figure 7. LoFTR+TPS performance. Image *b* shows how LoFTR+TPS is capable of aligning the satellite label to the drone image.

our results. For the first three lines of the table, we trained 3 DeepLabV3+ semantic segmentation networks with 3 types of image-label pairs. Firstly, we used drone images and predictions from sat-sat model as labels. Secondly we used drone images and generated labels from LoFTR. Thirdly,

Image size	Method	Class IoU (val)		mIoU (val)
		Building	Road	
5280x3956	DeepLabV3+ sat-sat-drone-drone(h)	57.59	29.75	43.67
5280x3956	SCOT	64.05	36.53	50.29
5280x3956	LoFTR+TPS	63.15	38.92	51.03
5280x3956	DeepLabV3+ drone-drone(h)-drone-drone(h)	**78.62**	**60.11**	**69.39**

Table 3. Comparison of generated drone labels by different methods with human annotated drone labels. The last row corresponds to our target performance.

we used drone images and SCOT generated labels. We took these 3 trained models and computed the predictions on the drone images from the validation set in order to compare them with the human annotated drone labels. Predicting on drone images utilizing these trained models using generated drone labels was beneficial as all the classes' IoUs are higher in this table than in the previous one (Tab. 3). The biggest increase in mIoU is for SCOT, with over 5% mIoU. This is also our best method so far, reducing the gap to the target model (last row) to 13% mIoU.

6. Ablation studies

We further explore the impact of the type of semantic segmentation network used for making predictions. In this sense, we compare DeepLabV3+ network with a UNet network. Tab. 5 is similar to Tab. 4 and it additionally contains the results for the UNet network. UNet predictions are worse than DeepLabV3+ predictions for almost all the drone label generating scenarios. UNet only surpasses DeepLabV3+ for the road class, when using predictions generated from sat-sat model. Therefore, the type of the semantic segmentation network used definitely affects the overall performance of the drone label generation process.

Even though LoFTR+TPS performs well in aligning the satellite label to the drone image, we noticed that it is still have troubles in the corner of the drone images as these are the areas with the highest displacement compared to the satellite images. For example, some buildings that would appear in satellite images would be missing from the drone images as the the camera angle is far from nadir. Because of this, the satellite label will not include those buildings either. LoFTR will not be able to generate labels for unseen objects, as it only align the existing satellite labels with the drone image, based on drone image-satellite image correspondence. Therefore, when generating labels with LoFTR, we center-cropped the labels (*LoFTR_warped(c)*) to have more accurate ones, and used this labels for training and predicting with the DeepLabV3+ network. In Tab. 6 we compared these predictions with full-size human annotated labels and with cropped human annotated labels (*drone(c)(h)*). The predictions were almost the same when comparing the full size generated labels with the human annotated labels, but there was a small increase when we compared them with cropped annotated labels.

The dataset used so far consisted of 100 images taken from an altitude of 120 meters and it was only a subset of the available data. Moreover, as stated in the beginning of Sec. 5, there was a small overlap between validation and training data. For this, we designed a function that automatically checks the overlapping between training and validation data based on latitude and longitude coordinates. Thus, we create 2 new datasets that do not contain any overlapping regions or samples. One of them contains 132 images taken from 120 meters. For this dataset we consider 107 images for training and the same 25 images for validation as up until now. The second dataset contains 446 images taken from different altitudes. We use the same 25 images for validation and the rest of them for training. With this experiment we are interested to see if enhancing the dataset with multiple images from different altitudes improves the performance of the segmentation networks. In Tab. 7 we show the comparison of using the 2 aforementioned datasets. We train a sat-sat model on each of the dataset and compare the predictions of these models on the drone images, with the human annotated labels. Enhancing the dataset with more images, even if they are captured from different altitudes, proves to be beneficial.

7. Future work

Even though SCOT [24] involves a quite simple architecture, it showed promising results for aligning the drone labels. We plan to further investigate how different image sizes used at inference time affect the performance of the model and if there is a way to improve this method.

Another possible direction for generating drone labels is Few Shot Semantic Segmentation. This task involves segmenting unseen-class objects given only a handful of densely labeled samples. By using the satellite image as support image and the satellite label as support mask, we can input the drone image as query, in order to obtain a drone label. MSANet [15] is the state-of-the-art in this domain and will be included in our future research.

Moreover we plan to analyze how different datasets used

Image size	Method	Class IoU (val)		mIoU (val)
		Building	Road	
5280x3956	DeepLabV3+ drone - prediction_from_sat-sat - drone - drone(h)	63.02	32.69	47.85
5280x3956	DeepLabV3+ drone - LOFTR_warped - drone - drone(h)	66.01	40.43	53.22
5280x3956	DeepLabV3+ drone - SCOT_warped - drone - drone(h)	**66.86**	**45.89**	**56.37**
5280x3956	DeepLabV3+ drone-drone(h)-drone-drone(h)	**78.62**	**60.11**	**69.39**

Table 4. Comparison of generated drone labels, from trained networks using already generated drone labels, with human annotated labels.

Image size	Method	Class IoU (val)		mIoU (val)
		Building	Road	
5280x3956	DeepLabV3+ drone - prediction_from_sat-sat - drone - drone(h)	**63.02**	32.69	**47.85**
5280x3956	UNet drone - prediction_from_sat-sat - drone - drone(h)	56.01	**37.14**	46.57
5280x3956	DeepLabV3+ drone - LoFTR_warped - drone - drone(h)	**66.01**	**40.43**	**53.22**
5280x3956	UNet drone- LoFTR_warped - drone - drone(h)	56.84	39.25	48.04
5280x3956	DeepLabV3+ drone - drone(h) - drone - drone(h)	**78.62**	**60.11**	**69.39**
5280x3956	UNet drone - drone(h) - drone - drone(h)	78.56	54.96	66.76

Table 5. Comparison of generated drone labels, from trained UNet and DeepLabV3+ models using already generated drone labels, with human annotated labels.

Train_img_size Test_img_size	Method	Class IoU (val)		mIoU (val)
		Building	Road	
4280x3581 5280x3956	DeepLabV3+ drone(c) - LoFTR_warped(c) - drone - drone(h)	63.88	42.45	53.16
4280x3581 4280x3591	DeepLabV3+ drone(c) - LoFTR_warped(c) - drone(c) - drone(c)(h)	**66.17**	**43.35**	**54.76**

Table 6. The impact of using cropped LoFTR generated labels for training, and predicting new drone labels.

Dataset	Method	Class IoU (val)		mIoU (val)
		Building	Road	
132 images 120m	DeepLabV3+ sat-sat-drone-drone(h)	60.09	28.42	44.25
446 images multiple alt.	DeepLabV3+ sat-sat-drone-drone(h)	**65.92**	**33.63**	**49.77**

Table 7. Impact of using clean, larger datasets.

in the pretraining of semantic segmentation models are impacting the model performance when using drone image - generated label pairs in training. Additionally, we look forward to test the state-of-the-art semantic segmentation networks [37] for predicting the drone labels.

8. Conclusion

In this paper, we investigated if it is feasible to replace the human annotated labels with generated drone labels in the context of semantic segmentation on drone imagery, with the purpose of reducing the time required by careful annotation. We analyzed the performance of several methods for aligning the satellite labels created in an automatic way, to the drone images, for obtaining qualitative drone labels. We compared these generated labels with the human annotated labels and analyzed how different semantic segmentation networks performed in training and at test time when using either generated labels or manually annotated labels. As there was a significant gap of around 13% in terms of mIoU between predictions obtained with generated labels and human annotated labels, further research into drone labels generation or alignment is definitely needed.

References

[1] Dji mavic 3. https://www.dji.com/ch/mavic-3. 2

[2] Mapbox. https://www.mapbox.com/. 3

[3] Maperitive. http://maperitive.net/. 3

[4] Openstreetmap. https://www.openstreetmap.org/. 3

[5] Overpass api. https://wiki.openstreetmap.org/wiki/Overpass_API. 3

[6] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009. 4

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2016. 2

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. 5

[9] Yu Chen, Yao Wang, Peng Lu, Chen Yisong, and Guoping Wang. *Large-Scale Structure from Motion with Semantic Constraints of Aerial Images: First Chinese Conference, PRCV 2018, Guangzhou, China, November 23-26, 2018, Proceedings, Part I*, pages 347–359. 11 2018. 2

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016. 5

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2017. 4

[12] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking, 2018. 2

[13] S. Girisha, Ujjwal Verma, M. M. Manohara Pai, and Radhika M. Pai. Uvid-net: Enhanced semantic segmentation of uav aerial videos by embedding temporal information. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:4115–4127, 2021. 2

[14] TU Graz. Icg semantic drone dataset. http://dronedataset.icg.tugraz.at/. 2

[15] Ehtesham Iqbal, Sirojbek Safarov, and Seongdeok Bang. Msanet: Multi-similarity and attention guidance for boosting few-shot segmentation, 2022. 7

[16] İsmail Karakaya, Berkan Demirel, Orkun Öztörk, Murat Bal, and Emre Başeski. Hvlseg: An ensemble model for instance segmentation on satellite images. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2020. 1

[17] W. Keller and Andrzej Borkowski. Thin plate spline interpolation. *Journal of Geodesy*, 93, 02 2019. 4

[18] Neeraj Kumar, Li Zhang, and Shree Nayar. What is a good nearest neighbors algorithm for finding similar patches in images? In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, pages 364–378, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 4

[19] Satyawant Kumar, Abhishek Kumar, and Dong-Gyu Lee. Semantic segmentation of uav images based on transformer framework with context information. *Mathematics*, 10(24), 2022. 2

[20] Xinghui Li, Kai Han, Shuda Li, and Victor Adrian Prisacariu. Dual-resolution correspondence networks, 2020. 4

[21] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos, 2018. 4

[22] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman. Sift flow: Dense correspondence across different scenes. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, pages 28–42, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 3

[23] Qun Liu, Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna Nemani. DeepSat v2: feature augmented convolutional neural nets for satellite image classification. *Remote Sensing Letters*, 11(2):156–165, nov 2019. 1

[24] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4462–4471, 2020. 4, 7

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2014. 2

[26] Ye Lyu, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang. Uavid: A semantic segmentation dataset for uav imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 165:108 – 119, 2020. 2

[27] Juhong Min and Minsu Cho. Convolutional hough matching networks, 2021. 4

[28] Open-MMLab. Open-mmlab. https://github.com/open-mmlab/mmsegmentation. 5

[29] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions, 2020. 4

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2, 5

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2014. 5

[32] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks, 2019. 4

[33] Amsa Shabbir, Nouman Ali, Ahmed Jameel, Bushra Zafar, Aqsa Rasheed, Muhammad Sajid, Afzal Ahmed, and Saadat Dar. Satellite and scene image classification based on transfer learning and fine tuning of resnet50. *Mathematical Problems in Engineering*, 2021, 07 2021. 1

[34] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers, 2021. 4

[35] Di Wang, Qiming Zhang, Yufei Xu, Jing Zhang, Bo Du, Dacheng Tao, and Liangpei Zhang. Advancing plain vision transformer towards remote sensing foundation model, 2022. 1

[36] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007. 4

[37] Haotian Yan, Chuang Zhang, and Ming Wu. Lawin transformer: Improving semantic segmentation transformer with multi-scale representations via large window attention, 2022. 2, 8

[38] Yuhui Yuan, Xiaokang Chen, Xilin Chen, and Jingdong Wang. Segmentation transformer: Object-contextual representations for semantic segmentation. 2019. 2

[39] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2016. 2

[40] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision meets drones: A challenge, 2018. 2