Ending Overfitting for UAV Applications - Self-Supervised Pretraining on Multispectral UAV Data

Jurrian Doornbos[1], Önder Babur[1,2]

[1] Information Technology Group, Wageningen University, Wageningen, the Netherlands – jurrian.doornbos@wur.nl, onder.babur@wur.nl
[2] Software Engineering and Technology, Eindhoven University of Technology, Eindhoven, the Netherlands

Keywords: foundation models, self-supervised learning, drones, UAV, multispectral

Abstract

While UAVs have revolutionized data collection for remote sensing, the practical application of Deep Learning remains severely limited by the scarcity of labelled training data, creating a stark contrast between laboratory successes and field performance. This research investigates whether transfer learning techniques can overcome this "small data problem" by enabling UAV-based deep learning models to generalize effectively across diverse environments without requiring prohibitive amounts of labelled examples. We present the use of an efficient self-supervised learning framework (FastSiam) tailored specifically for multispectral UAV imagery to overcome this generalization gap. Our approach enables effective feature learning without requiring extensive labelled data, bridging the gap between the potential of foundation models and the resource constraints of UAV remote sensing applications. We evaluate our method on a vineyard segmentation task across multiple geographic locations, demonstrating that models with FastSiam pretrained backbones significantly outperform their end-to-end trained counterparts, even with extremely limited labelled data. The most sophisticated architecture tested, Swin-T with a pretrained backbone, achieved an average F1 score of 0.80 across diverse test sites, showcasing robust generalization capabilities. Importantly, our results show that pretrained models benefit more from diversity in training samples than from sheer volume, suggesting new pathways for efficient model development in UAV applications. This work establishes that self-supervised pretraining serves as an effective regularizer for remote sensing tasks. Pretraining limits overfitting and improves generalization across varying environmental conditions, whilst requiring only modest computational resources, making advanced Deep Learning techniques more accessible for practical UAV applications.

1. Introduction

Uncrewed Aerial Vehicles (UAVs) have emerged as powerful tools for data collection across numerous applications within the remote sensing community (Toth & Jóźków, 2016). Their flexibility, relatively low operational costs, and ability to capture high-resolution, spectral imagery have made them invaluable for environmental monitoring, precision agriculture, infrastructure inspection, and disaster response (Doornbos et al., 2024).

Despite the ease of raw data that UAVs can collect, a significant challenge persists in the field: the "small data problem" (Safonova et al., 2023). While UAVs can generate substantial volumes of multispectral imagery, the labelled datasets required for supervised learning approaches are typically limited in size. This limitation stems from the resource-intensive nature of data annotation, which often requires domain expertise (Gao et al., 2022), substantial time investments and cost (Elezi et al., 2022).

The small data problem becomes particularly evident when implementing Deep Learning-based (DL) approaches for UAV-based remote sensing applications. DL approaches, with their significant number of parameters, are inherently data hungry (Simonyan & Zisserman, 2014). When trained on limited supervised datasets, these models tend to overfit to the training examples rather than learning generalizable patterns. This results in models that perform well on the specific conditions represented in the training data but fail to maintain performance when applied to new scenarios or locations (Goldblum et al., 2023). This generalization challenge substantially limits the practical utility of sophisticated UAV-based remote sensing applications. While academic publications may report impressive accuracy metrics, these results often do not translate to real-world implementations (Diez et al. 2021). Models trained in controlled research environments struggle to perform consistently across varying conditions, including changes in illumination, seasonality, geographic locations, or sensor characteristics (Doornbos et al., 2025).

The popularity of DL however is exactly due to its ability of generalization. For example, the Segment Anything Model (SAM) from Meta AI (Kirillov et al., 2023) is trained on over 1 billion masks across 11 million images, SAM demonstrates zero-shot transfer capabilities—it can segment objects in images without specific training on those particular objects or scenes. Similarly, text-to-image generative models like Stable Diffusion showcase DL's capability to internalize visual concepts and generate novel images across an astounding range of styles and content. Diffusion models learn latent representations of visual information that capture both the structural and semantic properties of images. In another Meta AI model, DINOv2 leverages a self-distillation approach with no labels to learn rich, transferable visual representations (Oquab et al., 2023). Trained on a diverse dataset of over 142 million images without any labels, DINOv2 demonstrates that DL models can develop sophisticated visual understanding through carefully designed self-supervision objectives. Whilst these examples from the broader computer vision domain can learn representations that generalize across domains, it also illustrates a need for sufficient data, architectural sophistication, and training methodologies.

The gap for DL in UAV remote sensing stems from two critical missing components: comprehensive multispectral UAV datasets and efficient self-supervised learning approaches tailored to UAV remote sensing imagery (Doornbos & Babur, 2025).

The contribution of this work is therefore as follows:

- Showcasing efficient self-supervised pretraining on a large-scale, diverse UAV multispectral dataset;
- Determining the effectiveness of the pretraining process on task-specific evaluation.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W2-2025-31-2025 | © Author(s) 2025. CC BY 4.0 License.

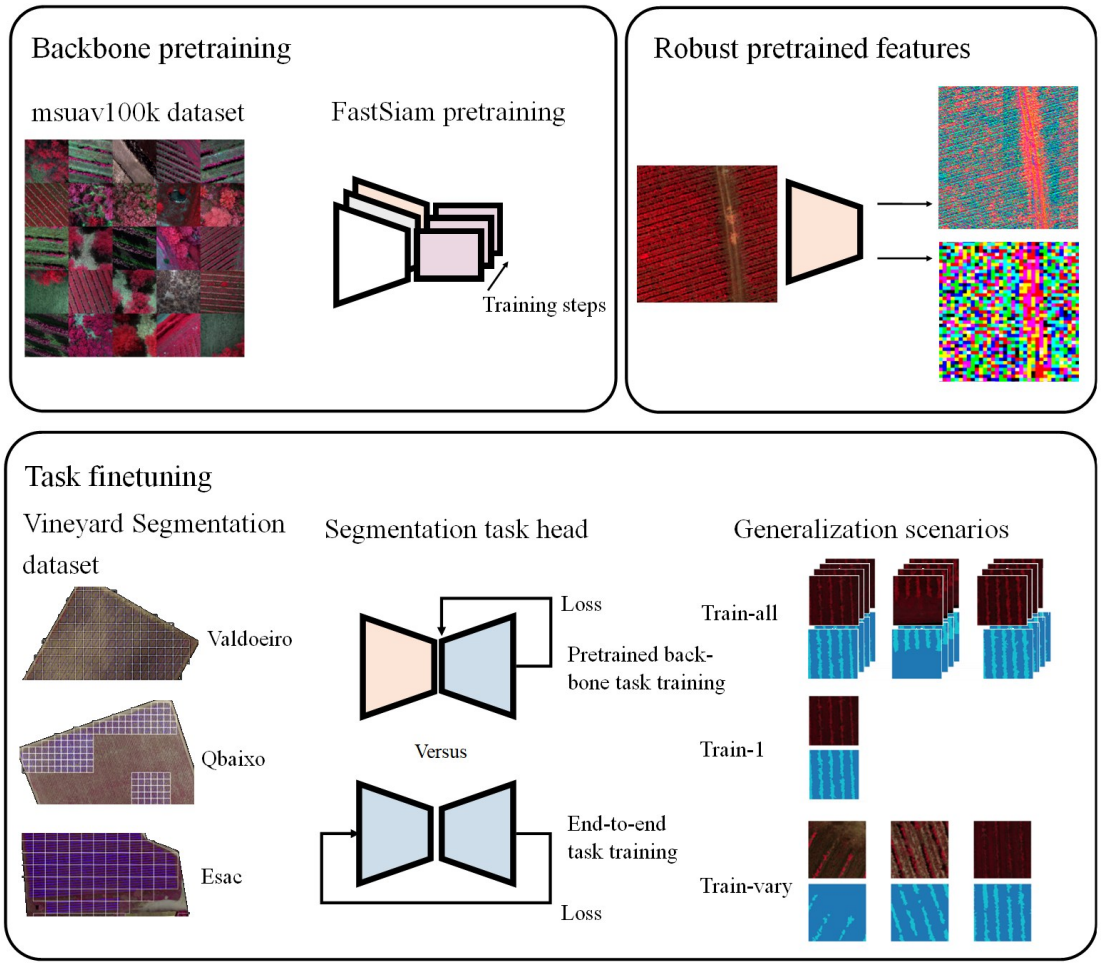

Figure 1. Methodology overview, with backbone pretraining using *msuav100k*, and the segmentation task finetuning.

The paper is structured as follows: Section 2 will present the pretraining dataset, the self-supervised learning approach and task-specific evaluation. Section 3 will demonstrate the effectiveness of pretraining on a typical segmentation task. Section 4 will discuss the implications and limitations of the findings. Section 5 concludes with a pathway forward.

2. Background

There are many different vision DL architectures to choose from, as well as different techniques to train them. This background section introduces the concepts of pretraining, backbones and self-supervised learning.

2.1 Pretraining backbones

The best performing image-based DL architectures are constructed of a backbone and a head, sometimes denoted as encoder and decoder. The backbone serves to extract all the important features from the image, and the head serves to execute a specific task. Goldblum et al., (2023) conducted over 1,500 training runs to evaluate a wide range of backbone architectures (including Convolutional Neural Networks, Vision Transformers, and hybrid models) with various pretraining methods (supervised learning, self-supervised learning, and vision-language training) across multiple tasks (classification, object detection, segmentation, retrieval, and out-of-distribution generalization). Their method involved standardized evaluation protocols with consistent hyperparameter optimization across all backbones to ensure fair comparisons. Key findings revealed that supervised ConvNeXt and SwinV2 architectures trained on large datasets performed the best overall, though self-supervised models showed competitive performance when compared on equal-sized pretraining datasets. The study also found that performance correlates strongly across different tasks, suggesting the emergence of universal vision backbones, and that transformers benefit more from increased scale than convolutional networks. Finally, when reducing to smaller models, more assumptions in the network architecture often increases accuracy, benefitting less from large amounts of data.

2.2 Self-supervised learning

The core idea behind self-supervised learning is to define a pretext task that the model can solve without external supervision (Chen et al., 2024). Common pretext tasks include predicting missing parts of an image, determining the relative position of image patches, or recognizing applied transformations like rotations, through a prediction and projection head. By solving these tasks, the model learns general features that can be transferred to downstream tasks like classification or segmentation.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W2-2025-31-2025 | © Author(s) 2025. CC BY 4.0 License.

2.2.1 Siamese Networks

Siamese networks are a specific architecture used in self-supervised learning that consists of two identical neural networks with shared weights. These twin networks process different views or transformations of the same input data and are trained to produce similar representations for related inputs and dissimilar representations for unrelated inputs.

The original SimSiam (Chen & He, 2020) approach uses simple Siamese architecture without negative pairs. It takes two augmented views of the same image, processes them through the twin networks and applies a stop-gradient operation over the projection network to prevent representation collapse. This approach demonstrates that contrastive learning doesn't necessarily require negative samples or momentum encoders to learn meaningful representations.

2.2.2 FastSiam:

FastSiam (Pototzky et al., 2022) builds upon the SimSiam framework but introduces several optimizations to make it more efficient and effective, optimizing for small batch sizes and limited computational resources. FastSiam employs multiple views of the same image. This multi-view approach effectively samples from a distribution with the same mean but reduced variance, helping to avoid outliers in the training process. This stabilization leads to faster convergence—FastSiam typically requires only 25 epochs compared to SimSiam's 100+ epochs to achieve comparable performance.

The fundamental insight of FastSiam is that averaging multiple samples reduces the standard error of the mean, creating more stable learning targets. This simple but effective modification allows for training with smaller datasets, fewer epochs, and limited computational resources while maintaining competitive performance compared to more resource-intensive approaches.

3. Methodology

This section outlines our approach to addressing the "small data problem" in UAV-based remote sensing through self-supervised learning. Figure 1 showcases the employed methodology, starting with the pretraining dataset *msuav100k*, which is solely used for pretraining a feature extractor using the FastSiam method. This results in robust pretrained features. Using vineyard segmentation as a task, we trained a vineyard segmentation model using the pretrained backbone (kept frozen during training) and a randomly initialized backbone (not frozen during training). The models were compared using different training data regimes to assess generalization. The methodology is structured as follows: Section 3.1 covers the pretraining dataset *msuav100k*, Section 3.2 presents our implementation of FastSiam, Section 3.3 shows the vineyard segmentation dataset and training splits, Section 3.4 explains the design choices for the segmentation head.

3.1 Pretraining dataset: *msuav100k*

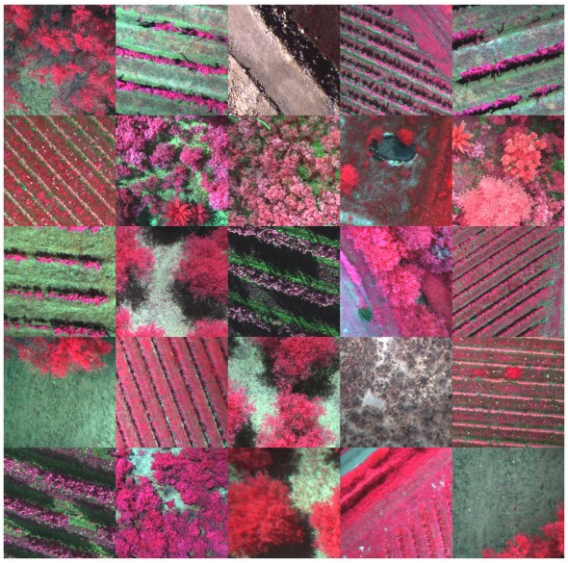

Figure 2. 25 sample false colour images from msuav100k.

The pretraining dataset employed was msuav100k. This dataset was compiled from a systematic online search of open-access data repositories, this collection encompasses 28 diverse datasets with imagery containing at least four spectral bands (Green, Red, RedEdge, and Near-Infrared). The dataset incorporates multispectral imagery captured by various sensor types including DJI Mavic 3M, DJI Phantom 4 Multispectral, Parrot Sequoia, MicaSense RedEdge, and MicaSense Altum/PT. Included datasets feature diverse applications and settings, such as vineyards and blueberry fields, mining waste, olive groves, rivers and more. A summary of these datasets is presented in Appendix I The total of 63,000 included images were radiometrically corrected and aligned per band if needed, afterwards they were cut into 512×512 patches, resulting in a total of 104,840 image chips.[1] Some examples of the chips are shown in false colour in Figure 2.

3.2 FastSiam pretraining

The intention of the FastSiam pretraining task is to train a robust feature extractor for downstream tasks. For this study, two different feature extractor architectures were selected: ResNet18 (He et al., 2015) and Swin Transformer (Liu et al., 2021). ResNet18 is a lightweight convolutional neural network with 11.2M parameters that employ residual connections to mitigate the vanishing gradient problem. It consists of 18 layers organized into 4 major blocks with progressively increasing channel dimensions (64, 128, 256, 512) and decreasing spatial resolution. Each residual block contains two 3×3 convolutional layers with batch normalization and ReLU activations. ResNet18 offers a good balance between computational efficiency and performance, making it suitable for resource-constrained environments while still providing strong feature extraction capabilities. Swin Transformer (Tiny) is a hierarchical vision transformer with 27.5M parameters that processes images using shifted windows of self-attention. It operates progressively by merging neighboring patches at each stage, creating a hierarchical representation with 4 stages of different feature resolutions. Each stage contains Swin

[1] This 100GB dataset is planned to be open source licensed, in the meantime, it is available upon request.

Transformer blocks with shifted window-based multi-head self-attention and MLP layers. The "tiny" variant uses a base dimension of 96 channels which expands through the network (96, 192, 384, 768). Swin-T's window-based attention mechanism reduces computational complexity while maintaining the transformer's ability to model long-range dependencies. In theory, Swin-T provides stronger feature extraction than ResNet18 as there are more parameters and less convolutional assumptions. Larger backbone networks were out of scope for the available resources.

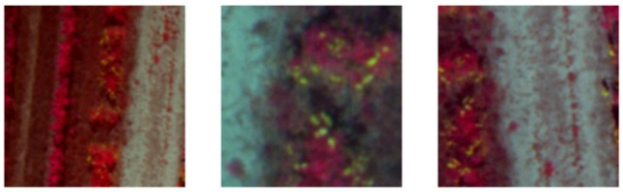

Figure 3. Three of the same input images with combinations of the augments: flip, spectral shift, noise, blur and zoom.

FastSiam uses augmentations for the projection and prediction heads. The selected augmentations for this study were randomly applied to the input images with a 50% chance. Augmentations included random resized crop, horizontal and vertical flip, gaussian blur, gaussian noise, brightness, and spectral increase or decrease, see Figure 3 for an example. Additional training parameters are shown in Table 1. Training was performed with an NVIDIA GTX1660Ti 6GB, for a duration of 5 hours for the ResNet18 model, and 52 hours for Swin-T-tiny[2].

Table 1. FastSiam training settings.

Parameter	Setting
Epochs	2
Learning rate	0.02
Optimizer	SGD with Cosine Annealing learning rate decay
Weight decay	0.0001
Batch size	32
Input channels	4
Projection head	2048 → 256
Prediction head	256 → 128 → 256
Backbone	ResNet18, Swin-T-tiny

3.3 Vineyard segmentation dataset

To evaluate the effectiveness of a pretrained backbone, the task of vineyard row segmentation was chosen. Vineyard row segmentation, while not the most challenging task in agricultural computer vision, serves as a good demonstrator for investigating vision DL model optimization and generalization characteristics for UAV applications. The dataset to support this task is from Barros et al (2022). This dataset is based on three distinct vineyards located in the central region of Portugal: Valdoeiro, Quinta de Baixo (further referred to as QBaixo), and Esac (which is further divided into two plots: Esac1 and Esac2). This dataset was specifically created for multispectral vineyard segmentation research. This dataset is not part of *msuav100k.*
The data was captured using a DJI drone equipped with a dual imaging sensor payload: a high-definition RGB camera (DJI Zenmuse X7 with 6016×4008 resolution) and a five-band multispectral and thermal camera (Micasense Altum, capturing Red, Green, Blue, Red-Edge, and Near-Infrared bands at 2064×1544 resolution, plus a thermal band at 57×44 resolution). Each vineyard was surveyed at different times and altitudes:

- *Esac1 & Esac2*: Surveyed October 2022 at 120m altitude (2.5cm GSD) post-harvest. 2.3ha vineyard planted in 1999 on sloping terrain (2°-5°) at 28m elevation. Cultivars: Alfrocheiro, Aragonez, Touriga Nacional, Marselan. Density: 2,800-3,400 vines/ha.
- *Valdoeiro*: Surveyed April 2022 at 60m altitude (1.25cm GSD) during early growth. 2.9ha vineyard planted in 2005 on flat terrain (<2°) at 99m elevation. Cultivar: Baga. Density: 3,200 vines/ha.
- *Quinta de Baixo:* Surveyed July 2022 at 70m altitude (1.45cm GSD) during advanced growth. 3.2ha vineyard planted in 2002 on sloping terrain (2°-5°) at 90m elevation. Cultivars: Syrah, Pinot, Baga. Density: 4,400 vines/ha.

Figure 4. False color sample imagery from Esac2, Valdoeiro and QBaixo respectively. The vineyards show different vegetation and soil structure and angle.

The collected raw images were processed to generate orthomosaics and digital surface models (DSMs) for each vineyard. The HD images were used to create both HD orthomosaics and DSMs, while the multispectral images underwent radiometric corrections including vignetting, dark pixel offset, and conversion to reflectance space using calibration panels.
The dataset includes annotated segmentation masks for binary classification, where pixels belonging to vine plants are labelled as positive class (1) and the rest as negative class (0). For the experiments, the orthomosaics were divided into 224×224 pixel sub-images (e.g. chips), see Figure 4 for some example chips.
Esac1 was used for training. Esac2 was used for selecting the best model from training. Whilst for testing, Valdoeiro and Qbaixo were used. One chip of each set was also kept back for testing one-shot training performance. For an overview of splits and chips, table 2 is provided.

- *Train-all*: The highest number of images, all from Esac1.
- *Train-1*: Uses only a single image from Esac1 to evaluate single-shot learning performance under extremely limited data conditions.
- *Train-vary*: Designed to investigate whether model performance benefits more from data quantity (*Train-all*) or diversity, comparing results when trained on varying vineyard examples.

Table 2. Dataset splits for vineyard segmentation task.

	Esac1	Esac2	Valdoeiro	QBaixo
Total chips	96	97	157	141
Train-1	1	0	0	0
Train-all	96	0	0	0
Train-vary	1	0	1	1
Validation	0	97	0	0
Test-V	0	0	156	0
Test-Q	0	0	0	140

[2] All code and weights are available on https://github.com/jurriandoornbos/multi_ssl

3.4 Segmentation task head design

The architecture employed for this segmentation task followed an encoder-decoder 'U-Net' structure with skip connections between corresponding encoder and decoder layers (purple and blue sections in Figure 5 respectively). These skip connections are vital as they allow the model to preserve fine spatial details that might otherwise be lost during downsampling.

Despite sharing the same segmentation head design, the implementation varies between the two backbone models. The ResNet18-based model features a segmentation head with 3.9 million parameters, while the Swin Transformer version has a larger head comprising 8.5 million parameters. This difference in parameter count reflects the size in output feature dimensions inside the respective backbone layers.

For different experiments on the effectiveness of pretraining, the weights from the pretrained backbone were used in the encoder and the whole backbone was kept frozen, or the backbone was randomly initialized with full end-to-end updates of the weights during training.

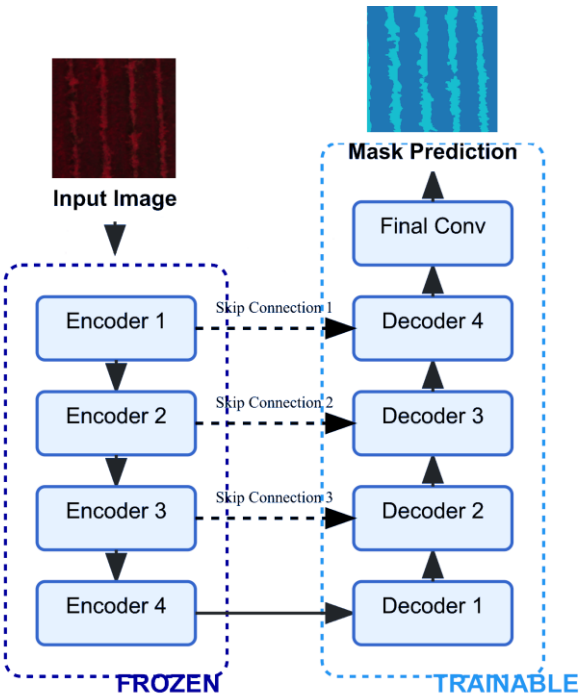

Figure 5. Segmentation head architecture diagram. A 4-channel multispectral image is passed into the encoder backbone, and through every feature dimension is decoded into a segmentation mask. Backbone encoder in purple dotted block, whereas the decoder head is in the blue dotted block.

Task training was performed on the same machine as FastSiam. Different experiments were performed: varying the training dataset between Train-1, Train-all and Train-vary, varying the backbone between ResNet18 and Swin-T-tiny, and varying between a frozen, pretrained FastSiam backbone, and a randomly initialized fully trainable backbone. Additional training settings are presented in Table 3. Every trained model was evaluated on the same test-sets: Test-V and Test-Q, with F1-scores being tracked for each image, calculating the mean and standard error over the test set. Using these metrics, a t-statistic and p-value was measured between the pretrained and end-to-end model, assuming a normal distribution. Finally, like Barros et al. (2022), a RandomForest was trained on the same dataset as a baseline comparison.

Table 3. Segmentation head training settings.

Parameter	Setting
Epochs	300
Learning rate	0.0003
Optimizer	SGD
Weight decay	0.0001
Batch size	8
Input channels	4
Backbone	ResNet18, Swin-T-tiny

4. Results

The results contain the accuracy scores for two different backbone architectures: ResNet18 and Swin-T-Tiny. These backbones were pretrained or randomly initialized. For self-supervised pretraining, FastSiam was used. In FastSiam, the aim of the model is to make augmented images close in feature space to their source image. As the name suggests, the *msuav100k* contains 100,000 images. These images were used to train the backbone. Two full epochs on the data resulted in around 6400 training steps. The negative cosine similarity loss is shown in Figure 6. After half the training time: 3200 steps (one epoch), both the backbones have already achieved their optimum. This indicates that even though it is the most comprehensive multispectral dataset, the models can capture the variance quickly, many of the images in *msuav100k* look very similar.

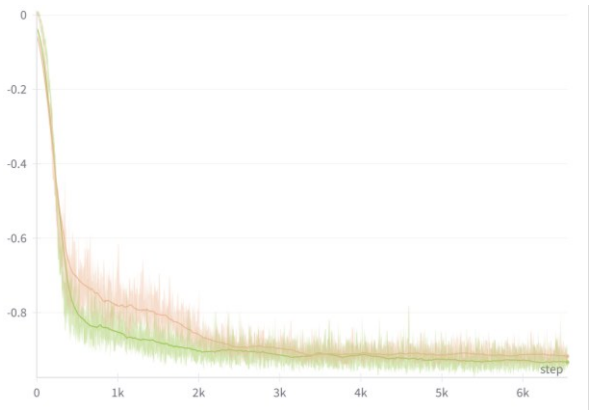

Figure 6. Train loss FastSiam over training steps (time), green line is ResNet18, orange line is Swin-T-tiny. ResNet18 drops significantly quicker to below -0.8 negative cosine similarity.

We evaluated these models across three distinct training regimes to understand performance patterns when applied to different vineyard test sites.

Our experimental design incorporated varying training datasets: *Train-all* (comprehensive dataset), *Train-1* (single image training), and *Train-vary* (diverse samples), enabling direct comparison between dataset size and variety effects. We examine end-to-end trained models against those using pretrained FastSiam backbones, while also including Random Forest as a traditional machine learning baseline.

The following results demonstrate how model performance varies across different vineyard conditions, highlighting the impact of pretraining as well as data characteristics on segmentation accuracy. Importantly, we find that variety of data

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W2-2025-31-2025 | © Author(s) 2025. CC BY 4.0 License.

is more important than dataset size, especially when the backbone is already pretrained.

4.1 Evaluation on *Train-all*

Table 4. Model F1 score results for *Train-all* subset. 'ee' denotes an end-to-end trained model, 'bb' denotes a pretrained FastSiam backbone.

	Esac2	Valdoeiro	QBaixo	Mean
ResNet18-ee	0.44	0.61	0.46	0.53
ResNet18-bb	0.75*	0.85*	0.59*	0.72*
Swin-T-ee	0.49	0.69	0.52	0.60
Swin-T-bb	0.56*	0.83*	0.42	0.63*
RandomForest	0.81	0.8	0.06	0.43

'*' Indicates a statistically significant (p-value < 0.01) improvement of using the pretrained backbone.

Table 4 presents the F1 scores for models trained on the *Train-all* subset and tested across three vineyard sites. The results demonstrate a clear performance advantage when using pretrained FastSiam backbones compared to end-to-end training approaches.

For ResNet18, the pretrained backbone ('bb') achieved significantly higher F1 scores across all testing sets, with an average improvement of 19 percentage points over its end-to-end ('ee') counterpart. Similarly, the Swin-T architecture showed statistically significant improvements with pretraining on two of the three test sites.

It's particularly noteworthy that while Esac2 comes from the same flight mission as the training data (Esac1), the pretrained backbones still delivered substantial performance gains even in this scenario where domain shift is minimal. This suggests that the self-supervised pretraining captures fundamental vineyard features that benefit segmentation regardless of data similarity to the training set.

The conventional RandomForest classifier performed competitively on the Esac2 and Valdoeiro sites but failed dramatically on Quinta de Baixo, highlighting its poor generalization to more distinct vineyard conditions.

4.2 Evaluation on *Train-1*

Table 5. Model F1 score results for *Train-1* subset. 'ee' denotes an end-to-end trained model, 'bb' denotes a pretrained FastSiam backbone.

	Esac2	Valdoeiro	QBaixo	Mean
ResNet18-ee	0.78	0.44	0.17	0.33
ResNet18-bb	0.74	0.49*	0.26*	0.38*
Swin-T-ee	0.63	0.77	0.36	0.57
Swin-T-bb	0.8*	0.70	0.39*	0.55
RandomForest	0.76	0.73	0.62	0.67

'*' Indicates a statistically significant improvement of using the pretrained backbone.

Table 5 shows the F1 scores for models trained using only a single image from Esac1. This extremely limited training scenario reveals that just one labelled image is only sufficient when the application target (Esac2) comes from a very similar domain as the training data. Both ResNet18 and Swin-T architectures achieve relatively high F1 scores (0.74-0.80) on Esac2, but their performance degrades substantially when tested on the more distinct Valdoeiro and Quinta de Baixo vineyards.

Interestingly, the pretrained FastSiam backbones ('bb') show statistically significant improvements over end-to-end training ('ee') on the more challenging test sites, particularly for ResNet18 on both Valdoeiro and Quinta de Baixo. This suggests that self-supervised pretraining provides valuable knowledge transfer that helps mitigate the severe data limitation. Even though pretrained is shown to be better, the model outputs are not useable at F1-scores well under 0.7.

4.3 Evaluation on *Train-vary*

Table 6. Model F1 score results for *Train-vary* subset. 'ee' denotes an end-to-end trained model, 'bb' denotes a pretrained FastSiam backbone.

	Esac2	Valdoeiro	QBaixo	Mean
ResNet18-ee	0.59	0.67	0.26	0.47
ResNet18-bb	0.57	0.76*	0.68*	0.72*
Swin-T-ee	0.80	0.75	0.62	0.69
Swin-T-bb	0.79	0.82*	0.78*	0.80*
RandomForest	0.42	0.86	0.72	0.79

'*' Indicates a statistically significant (p-value < 0.01) improvement of using the pretrained backbone.

Table 6 presents the F1 scores for models trained on the *Train-vary* subset, confirming the previous patterns observed across different training regimes. Pretrained FastSiam backbones ('bb') consistently outperform their end-to-end ('ee') counterparts, with statistically significant improvements on the more challenging Valdoeiro and Quinta de Baixo test sites. This pattern holds true regardless of the amount of training data, reinforcing that the diversity and similarity to application data are more important than sheer volume of training examples.

Most notably, the Swin-T architecture with pretrained backbone achieves the highest mean F1 score (0.80) across all test sites, outperforming all other models. The superior performance of the most complex pretrained backbone is consistent across all training regimes, highlighting that in low-data scenarios, transformer-based architectures with self-supervised pretraining can effectively leverage their representational power without overfitting.

The RandomForest classifier shows strong performance on Valdoeiro and Quinta de Baixo but struggles significantly on Esac2, indicating that the increase in diversity of the training dataset is too much for the model to capture.

5. Discussion

The results from *Train-all* and *Train-vary* demonstrate that models with a frozen backbone obtained through FastSiam pretraining exhibit superior generalization capabilities compared to those with random initialization. Additionally, the largest model has the best performing scores.

5.1 Pretraining as a regularizer

During the segmentation training task, only the model head receives gradient updates while the pretrained backbone remains fixed. The pretrained backbone has already developed robust feature representations from a much larger dataset, allowing it to identify key patterns. Consequently, similar features such as vineyard rows trigger consistent outputs from the segmentation head, suggesting that vineyard-row detection represents a learnable concept across different locations. Therefore, the pretrained backbone acts as a strong regularizer during training. This approach aligns with findings from Goldblum et al. (2023). In which they demonstrate and compare pretrained backbones and end-to-end trained backbones for generic computer vision tasks and benchmarks.

However, our findings reveal an important nuance: while abundant training data produces strong results on samples resembling the training distribution, a pretrained backbone significantly enhances generalization compared to end-to-end training approaches.

The most substantial improvement, however, lies in reducing the quantity of labels needed for effective model performance. In the *Train-vary* dataset, pretrained backbones achieved optimal scores, indicating that performance depends less on the volume of similar training samples and more on the diversity of samples used to optimize the downstream task on a pretrained backbone. This efficiency in label utilization parallels Goldblum et al. (2023) for self-supervised and vision-language pretrained models often excel with limited labelled data, suggesting that the representations learned during pretraining effectively compress the knowledge needed for downstream tasks.

Despite these promising outcomes, we have not yet reached the point where a single label suffices for learning tasks that generalize across diverse environments. The *Train-1* dataset failed to produce usable results, highlighting current limitations. This limitation reflects the boundary conditions identified in Goldblum et al. (2023), which shows that while pretraining significantly reduces data requirements, there remains a minimum threshold of task-specific data needed for effective transfer learning. Further research incorporating semi-supervised learning approaches will be necessary to develop DL models for UAV applications that generalize effectively across various conditions.

5.2 Model size and task complexity

The largest backbone in our study, Swin-T-Tiny with 27.5M parameters, demonstrated the strongest generalization capabilities across varied conditions (*Train-vary*). This raises the question whether scaling to even larger models would yield further improvements in performance. Goldblum et al. (2023) provide intriguing insights here, observing that "ViTs benefit more from scale than CNNs" with significantly higher correlation between parameter count and performance for transformer architectures. This suggests that scaling Swin Transformer architectures might indeed yield further improvements for complex, varied datasets.

However, the *Train-all* section also indicates that the smaller ResNet18 architecture can still outperform the larger models despite having fewer parameters, suggesting that the relationship between model size and performance is not strictly linear.

This apparent contradiction highlights the complex interplay between model architecture, parameter count, generalization ability and task complexity. The superior performance of ResNet18 in the data-rich *Train-all* scenario may indicate that certain architectural properties, such as residual connections and convolutional filters, provide particular advantages for the relatively simple vineyard segmentation tasks when sufficient training data is available. This is also suggested in Goldblum et al. (2023): "convolutional networks excel under linear probing," suggesting that CNN architectures may form more immediately useful representations for straightforward visual tasks without extensive fine-tuning.

Additionally, it remains uncertain whether significantly larger models, beyond the scale tested in our experiments, would continue to improve performance or simply introduce unnecessary computational overhead without proportional gains. Perhaps pretraining datasets with more variance could be introduced to increase the difficulty for DL models.

Furthermore, the benefits of larger models only show with more complex tasks, such as multi-task or more fine-detailed or nuanced objectives (such as disease prediction from leaves).

Future work could explore even smaller and larger backbone architectures to understand what type of task and dataset size would demand what type of model size for UAV-based applications. Although Goldblum et al. (2023) provides some guidance here, suggesting that modern hybrid architectures like Swin might offer the best compromise between the spatial inductive biases of CNNs and the scaling advantages of transformers for real-world deployment scenarios.

6. Conclusion and future work

The growing adoption of UAVs in remote sensing has created a paradoxical challenge: while data collection capabilities have expanded dramatically, our ability to extract meaningful insights remains constrained by the "small data problem" - the scarcity of labelled examples needed for supervised learning approaches. Current DL models for UAV applications often fail to generalize beyond their training environments, creating a significant gap between impressive laboratory results and disappointing field performance. This research investigated whether pretraining backbones, which have revolutionized general computer vision tasks, can bridge this critical gap for UAV-based environmental monitoring without requiring prohibitive amounts of labelled data. Our investigation demonstrates that pretrained backbones significantly enhance model performance in UAV-based segmentation tasks, primarily by constraining overfitting. The FastSiam pretraining approach functions as an excellent regularization mechanism, effectively limiting the backbone's tendency to overfit to training data while substantially improving generalization capabilities across diverse test scenarios. This finding aligns with the growing recognition that self-supervised pretraining offers considerable advantages for downstream tasks with limited labelled data.

The strong performance observed with pretrained backbones suggests that the challenge in UAV remote sensing applications may not necessarily lie in feature extraction capabilities, but rather in the availability and diversity of labelled data. The feature extractors developed through our pretraining approach appear robust enough to capture relevant patterns across varying conditions, indicating that the primary bottleneck has shifted toward label efficiency.

To further validate the effectiveness of our approach, future work should test these pretrained models on additional applications and datasets. The Multispectral UAV benchmark (UAVM) introduced by Li et al. (2024) represents a particularly promising avenue for evaluating the transferability of our findings across different remote sensing contexts and sensor modalities.

Building on these promising results, several research directions merit exploration. Semi-supervised learning approaches could leverage the strong representations from our pretrained models while requiring fewer labelled examples (Gao et al., 2022). Similarly, active learning strategies might help identify the most informative samples for labelling, further improving label efficiency (Elezi et al., 2022). These approaches could address the current limitations in label availability while capitalizing on the robust feature extraction capabilities we have developed. Finally, proper model selection remains an open issue and guidelines should be assembled, considering model task, model architecture and training dataset size.

Acknowledgements

This research was funded by the Horizon Europe program in the scope of the ICAERUS project (contract number 101060643).

References

Barros, T., Conde, P., Gonçalves, G., Premebida, C., Monteiro, M., Ferreira, C. S. S., & Nunes, U. J. (2022). Multispectral vineyard segmentation: A Deep Learning comparison study. Computers and Electronics in Agriculture, 195, 106782.

Chen, Y., Mancini, M., Zhu, X., & Akata, Z. (2024). Semi-Supervised and Unsupervised Deep Visual Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(3), 1327–1347.

Chen, X., & He, K. (2020). Exploring Simple Siamese Representation Learning. ArXiv preprint ArXiv:2011.10566

Doornbos, J., Bennin, K. E., Babur, Ö., & Valente, J. (2024). Drone Technologies: A Tertiary Systematic Literature Review on a Decade of Improvements. IEEE Access, 12, 23220–23240.

Doornbos, J., Babur, Ö., & Valente, J. (2025). Evaluating Generalization of Methods for Artificially Generating NDVI from UAV RGB Imagery in Vineyards. Remote Sensing 2025, Vol. 17, Page 512, 17(3), 512.

Doornbos, J. and Babur, Ö. (2025) Features from Multispectral Drone Data: Curating, training and distributing Transformers for all, EGU General Assembly 2025, Vienna, Austria, 27 Apr–2 May 2025, EGU25-1534, https://doi.org/10.5194/egusphere-egu25-1534, 2025.

Diez, Y., Kentsch, S., Fukuda, M., Caceres, M. L. L., Moritake, K., & Cabezas, M. (2021). Deep Learning in forestry using uav-acquired rgb data: A practical review. Remote Sensing, 13(14).

Elezi, I., Yu, Z., Anandkumar, A., Leal-Taixé, L., & Alvarez, J. M. (2022). Not All Labels Are Equal: Rationalizing the Labeling Costs for Training Object Detection. ArXiv preprint ArXiv:2106.11921

Gao, J., Burghardt, T., & Campbell, N. W. (2022). Label a Herd in Minutes: Individual Holstein-Friesian Cattle Identification. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 13374 LNCS, 384–396.

Goldblum, M., Souri, H., Ni, R., Shu, M., Prabhu, V., Somepalli, G., Chattopadhyay, P., Ibrahim, M., Bardes, A., Hoffman, J., Chellappa, R., Wilson, A. G., & Goldstein, T. (2023). Battle of the backbones: a large-scale comparison of pretrained models across computer vision tasks. Proceedings of the 37th International Conference on Neural Information Processing Systems.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. ArXiv preprint ArXiv:1512.03385

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo W.Y., et al. (2023). Segment anything. ArXiv preprint ArXiv:2304.02643

Li, Q., Yuan, H., Fu, T., Yu, Z., Zheng, B., & Chen, S. (2024). Multispectral Semantic Segmentation for UAVs: A Benchmark Dataset and Baseline. IEEE Transactions on Geoscience and Remote Sensing, 1–1.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. v, Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., … Bojanowski, P. (2023). DINOv2: Learning Robust Visual Features without Supervision. ArXiv preprint ArXiv:2304.07193

Pototzky, D., Sultan, A., Schmidt-Thieme, L. (2022). FastSiam: Resource-Efficient Self-supervised Learning on a Single GPU. In: Andres, B., Bernard, F., Cremers, D., Frintrop, S., Goldlücke, B., Ihrke, I. (eds) Pattern Recognition. DAGM GCPR 2022. Lecture Notes in Computer Science, vol 13485. Springer, Cham.

Safonova, A., Ghazaryan, G., Stiller, S., Main-Knorn, M., Nendel, C., & Ryo, M. (2023). Ten Deep Learning techniques to address small data problems with remote sensing. International Journal of Applied Earth Observation and Geoinformation, 125, 103569.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Toth, C., & Jóźków, G. (2016). Remote sensing platforms and sensors: A survey. ISPRS Journal of Photogrammetry and Remote Sensing, 115, 22–36.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. ArXiv preprint ArXiv:2103.14030

7. Appendix

msuav100k overview.

Sensor platform	Topic	GSD (cm)	Chips
DJI Mavic 3M	Mining-waste	2	6390
	Nitrogen	1	1232
	Olivegrove	1	15166
	Portugal Vineyard	1.5	9084
	UK Vineyard	2	45035
DJI Phantom 4M	Cacao	4.2	7642
	Tropical	4.2	1172
MicaSense Altum	Beechforest	3	324
	Macroalgae	1.5	600
MicaSense RedEdge	Blueberry	Variable	341
	Botrytis	1	6530
	Contamination	5	1855
	Forestfuel	Variable	4444
	Potato	Variable	368

	Rivers	25	37
Parrot Sequoia	Cherry	Variable	18
	Diurnal	7.5	62
	Localization	Variable	449
	Nature	Variable	2010
	Subtropical	Variable	1951