Sugar-beet Stress Detection using Satellite Image Time Series

Author information scrubbed for double-blind reviewing

Keywords: Satellite Image Time Series (SITS), Autoencoder, 3D Convolution, Representation Learning

Abstract: Satellite Image Time Series (SITS) data has proven effective for agricultural tasks due to its rich spectral and temporal nature. In this study, we tackle the task of stress detection in sugar-beet fields using a fully unsupervised approach. We propose a 3D convolutional autoencoder model to extract meaningful features from Sentinel-2 image sequences, combined with acquisition-date-specific temporal encodings to better capture the growth dynamics of sugar-beets. The learned representations are used in a downstream clustering task to separate stressed from healthy fields. The resulting stress detection system can be directly applied to data from different years, offering a practical and accessible tool for stress detection in sugar-beets.

1 INTRODUCTION

Sugar beet (Beta vulgaris) is a major industrial crop in Europe, particularly in countries like Germany, France, and Poland. The European Union accounts for nearly 50% of the world's sugar-beet production, with Germany playing a significant role in both cultivation and processing (European Commission, nd). Within Germany, the region of Bavaria is especially important due to its favourable soil, climate, and established farming practices. Sugar beet is not only crucial for granulated sugar production but also supports livestock feed, fermentation industries, and biofuel applications, making it economically and environmentally valuable.

Sugar-beet cultivation is subject to a wide range of stress factors including biotic stress such as diseases and pests, and abiotic stress like drought, heat, and nutrient deficiencies. These conditions negatively affect both yield and sugar content, and reduce the overall quality and productivity of the crop. Therefore, identifying stress within sugar beet fields is essential for gaining insights into crop health and guiding better management practices.

Sugar-beet fields are large and widely distributed, making manual inspection for stress time-consuming and labor-intensive. High-resolution imagery and labeled datasets are often costly and impractical for small-scale farmers. This study proposes a method for sugar-beet stress detection using publicly available Sentinel-2 satellite data. The main challenge in this study is the limited availability of labeled data, with only 5% of sugar-beet fields being labeled. Therefore, we use labeled data exclusively for evaluation, and develop a fully unsupervised framework leveraging both spectral and temporal characteristics of Sentinel-2 data. The code used in this study is available at: `https://anonymous.4open.science/r/Stress-Detection-7343/`.

2 RELATED WORK

Traditionally, classical computer vision and machine learning methods have been used for plant disease detection from satellite images, relying on hand-crafted features such as statistical descriptors and vegetation indices like Normalised Difference Vegetation Index (NDVI)(Shanmugam et al., 2017; Raza et al., 2020; Rumpf et al., 2010). Inspired by these approaches, we include histogram-based features as a baseline for comparison with the proposed model.

Recent studies(Victor et al., 2024; Yu et al., 2021; Ji et al., 2018) have explored deep learning for agricultural tasks like crop classification and disease detection. For instance, TempCNNs (Ji et al., 2018) use 3D convolutions to model temporal satellite data. These studies treat disease detection as a classification task requiring labeled datasets. While effective, such supervised methods can be impractical for small-scale industries that lack the resources to label massive datasets. To address this, recent work(Zhao et al., 2021; He et al., 2022; Cong et al., 2022) has applied autoencoders for unsupervised feature extraction from satellite data. Additionally, Vision Transformer (ViT) based models(Tarasiou et al., 2023; He et al., 2022; Cong et al., 2022) demonstrate the value of temporal encodings in learning data representations. Inspired by these studies, we propose an autoencoder model with 3D convolution and temporal

encodings derived from acquisition dates for feature learning, followed by clustering for stress detection– thereby eliminating the need for labeled data.

2.1 Auto-encoders

Autoencoders are neural networks used for unsupervised representation learning, enabling tasks such as dimensionality reduction and feature extraction. They encode the high-dimensional input $\mathbf{x}$ into a lower-dimensional latent representation $\mathbf{z}$ through an encoder, capturing essential features while discarding irrelevant details. A decoder then reconstructs the original input from $\mathbf{z}$, producing $\hat{\mathbf{x}}$. This process is summarised as:

$$\hat{\mathbf{x}} = \text{Decoder}(\text{Encoder}(\mathbf{x}))$$

Loss Function: The model is trained to minimise reconstruction error between $\mathbf{x}$ and $\hat{\mathbf{x}}$, most commonly using Mean Squared Error (MSE):

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \quad , \tag{1}$$

where $\mathbf{x}$ is the input data, $\hat{\mathbf{x}}$ is the reconstructed data, and n is the number of examples. This optimisation enables the model to learn compact, informative representations of the data.

3 DATA

This study uses Sentinel-2 Level 2A images (Sentinel Copernicus, nd) of agricultural areas in Germany, each containing multiple sugar-beet fields. Images have spatial dimensions of $(1000, 1000)$, with float32 reflectance values typically between 0 and 1, occasionally exceeding 1 in highly reflective areas. The data covers June to early September 2019, covering the sugar-beet growth cycle from early development to harvest. We use 10 spectral bands (excluding Sentinel-2 bands 1, 9, and 10 due to redundancy and cloud mask availability), all resampled to 10 m resolution. Each image includes multiple temporal instances, from which we select seven cloud-free temporal instances spread accross the growth season. An image is represented as $I \in \mathbb{R}^{H \times W \times C \times T}$, where $H = 1000$, $W = 1000$, $C = 10$, and $T = 7$ denote the height, width, number of channels, and timepoints, respectively (Fig. 1). Three auxiliary masks are used in pre-processing: a cloud mask derived from the Sentinel-2 Scene Classification Map (Sentinel Hub, nd) to remove clouds, a sugar-beet field ID mask using the field numbers assigned by us for tracking and identification, and an acquisition date mask for temporal encoding.

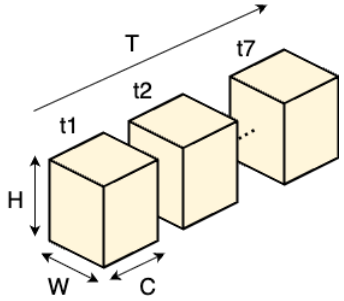

Figure 1: Image data represented as a sequence of temporal instances from $t1$ to $t7$. Each cube corresponds to a single instance with dimensions (H, W, C).

The evaluation set for the 2019 season currently includes 35 stressed and 26 healthy sugar-beet fields. However, class imbalance remains a potential concern—both in the unlabeled 2019 training set and in future datasets such as the upcoming 2024 season. This imbalance may arise from two primary factors. First, environmental conditions can lead to widespread stress across fields, making healthy cases relatively rare. Second, ground-truth labels are primarily collected by farmers, who tend to report stressed fields more frequently due to their agronomic relevance. As a result, the datasets may potentially over-represent stressed cases.

3.1 Data Preprocessing

The preprocessing pipeline converts raw Sentinel-2 images into model-ready tensors (Fig. 2). We first binarise the sugar-beet field ID mask and apply it pixel-wise across all channels and timepoints to retain only the sugar-beet pixels. We then extract each sugar-beet field as a temporal patch and pad it symmetrically with zeros to a uniform spatial size of $(64, 64)$. Using the cloud mask, we remove cloud-covered temporal instances, as accurate reflectance values are critical for stress detection. We select seven cloud-free instances– two per month from June to August and one from September– to ensure seasonal coverage. To avoid noise from adjacent vegetation, we exclude border pixels of the sugar-beet fields.

Table 1: Summary of tensor variants used. Each tensor represents a pre-processed sugar-beet sub-patch over seven temporal instances.

Tensor	Channel Composition	#Channels
B10	All Sentinel bands (excluding bands 1, 9, 10)	10
MVI	NDVI, EVI, and MSI vegetation indices	3
B4	Sentinel bands 2, 4, 8, 11	4

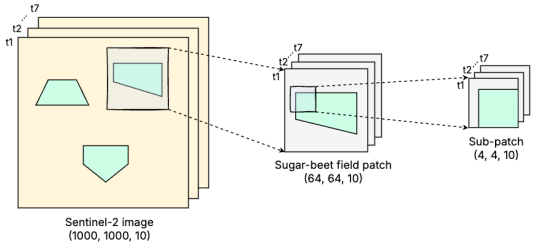

Figure 2: Data preprocessing overview. Each temporal instance is represented as (H,W,C), and the sequence of seven instances is denoted by $t_1,t_2,\ldots,t_7$.

Next, we split each patch into non-overlapping $(4,4)$ sub-patches. We discard the empty sub-patches that contain no sugar-beet pixels, and fill the partially empty sub-patches with the channel-wise average of the sugar-beet field pixels within that sub-patch. We then create three tensor variants using different channel combinations for experimentation, as detailed in Table 1. The vegetation indices Normalised Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), and Moisture Stress Index (MSI) are used as channels in the MVI tensor. The B10 variant uses all Sentinel-2 bands (except bands 1, 9 and 10), and the B4 variant uses Sentinel-2 bands 2, 4, 8, and 11– used to calculate vegetation indices in MVI as channels. Each sub-patch is represented as a 4D tensor $\mathbf{x} \in \mathbb{R}^{4\times4\times C\times7}$, where C depends on the tensor variant.

Figure 3 illustrates the RGB composites of sample images from the dataset.

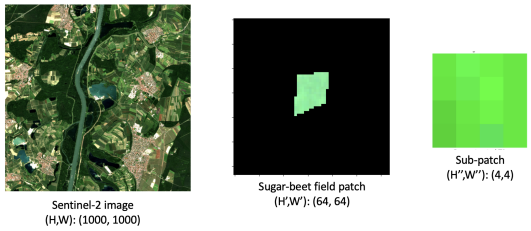

Figure 3: Sample Sentinel-2 data according to the terminologies, shown for a single temporal instance using the RGB composite.

4 MODEL

In this section, we first discuss temporal encodings used in this study. The following three subsections outline the modeling process, which comprises of feature extraction, downstream clustering, and converting sub-patch predictions to field-level labels. Finally, we discuss the generation of localized stress maps as deliverables.

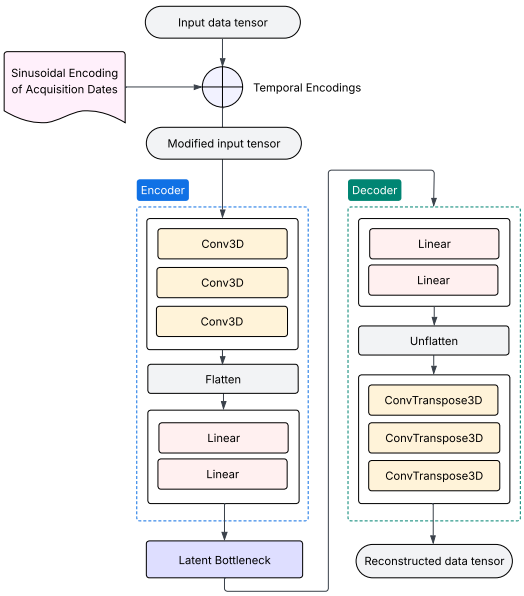

Figure 4: 3D_AE model with temporal encodings.

4.1 Temporal Encodings

Temporal information is essential for understanding the sugar-beet growth and evolution of stress patterns. Although Sentinel-2 provides frequent images, cloud cover often limits its usability. We select seven non-clouded images per field spread across the growth season. However, the acquisition dates vary across sugar-beet fields and the temporal sequences are not uniformly spaced. We address this by adding acquisition-date-specific temporal encodings to the input tensor. We use sinusoidal encodings (Vaswani et al., 2017) to represent acquisition dates, mapping each day $d \in [1,365]$ to a continuous annual cycle. The encodings are computed as follows:

$$\mathrm{e}_s = \sin\left(\frac{2\pi d}{365}\right), \quad \mathrm{e}_c = \cos\left(\frac{2\pi d}{365}\right)$$

These are added element-wise to all pixels of the input data tensors during the forward pass.

4.2 Feature Extraction

To obtain compact and meaningful representations from the preprocessed sub-patch tensors, we use the autoencoder architecture, with 3D convolutions. Temporal encodings are first added to the input tensor, which is then passed through 3D convolutional layers that preserve the spatiotemporal dimensions while

progressively increasing the number of feature maps, to capture both temporal and spectral characteristics. The resulting tensor is flattened and passed through fully connected layers to produce a latent representation $\mathbf{z} \in \mathbb{R}$. The decoder reconstructs the input tensor from the latent representation using transposed 3D convolutions, thus restoring the original dimensions. We train the model using the MSE loss (equation (1)), after which, we extract the latent representations and use them as feature vectors for downstream clustering. We denote the model as 3D_AE, with the architecture as shown in Fig. 4.

4.3 Downstream Clustering

Due to limited labeled data, we categorise the sugar-beet sub-patches as stressed or healthy by applying unsupervised clustering on the learned features $\mathbf{z}$ instead of supervised classification. We use k-means to group the features into two clusters, which we interpret as the target classes.

4.4 Obtaining patch-level labels

Since ground truth labels are only available for the entire sugar-beet fields, we aggregate sub-patch predictions using a threshold $\alpha \in [0,1]$. If more than $\alpha\%$ of the sub-patches are predicted as stressed, the entire sugar-beet field is labeled stressed. Using a configurable threshold provides flexibility in defining the severity of stress required, to categorize an entire field as stressed. This flexibility is useful in different scenarios—for example, a lower threshold can be set to detect fields with minimal stress for closer monitoring, while a higher threshold helps avoid false alarms by only marking fields with significant stress. We use $\alpha = 0.5$ as the default value and further analyse the effect of varying this threshold.

4.5 Generation of localized stress maps

Apart from the sugar-beet field predictions, we also generate visual deliverables in the form of field-level stress maps. These stress maps are RGB images with localized stress regions highlighted using bounding boxes drawn over predicted stressed sub-patches. Since the default sub-patch size is 4, each bounding box corresponds to a $(4,4)$ pixel region in the original image space. These visualizations are generated for both unlabeled fields and the labeled evaluation set, making them a valuable output for analysis. As more labeled data becomes available, the stress detection system can be further refined using a downstream classification task, making the visual outputs even more accurate and directly actionable for farmers. Figure 5 presents representative stress maps.

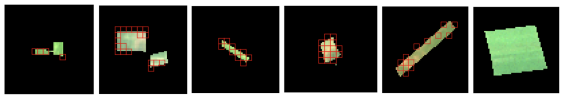

Figure 5: Localized stress maps for sugar-beet fields as deliverables, marked on RGB composite of the last temporal instance.

Figure 6 illustrates the end-to-end workflow of the stress detection system.

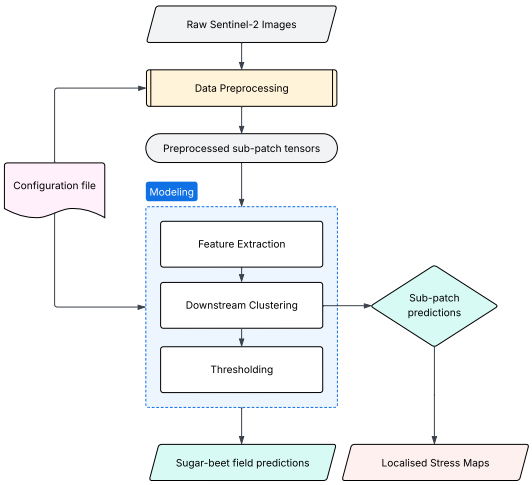

Figure 6: Sugar-beet stress detection system workflow.

5 RESULTS AND DISCUSSION

A total of 1857 unlabeled cloud-free sugar-beet fields corresponding to 1228 patches are partitioned into 33128 sub-patches. Of these, 80% are used for training and 20% for testing the 3D_AE model. The latent features extracted from all 1857 unlabeled fields are then used for k-means clustering. The evaluation set comprises of 61 cloud-free sugar-beet fields corresponding to 48 patches, further divided into 1197 sub-patches. We use the B10 sub-patch tensor as default, and sub-patch-to-patch threshold (α) as 0.5.

Model performance is calculated over sugar-beet fields, specifically across the 61 fields in the evaluation set, using accuracy, precision, recall and F1-score (in percentages). While accuracy reflects the overall correctness, the F1 score provides a deeper insight, since the primary focus is detecting stressed fields while minimizing false positives. Due to potential class imbalance in the dataset, as highlighted in the data section, accuracy by itself can be mislead-

Table 2: K-means clustering results on sugar-beet fields using various feature extraction methods.

Method	Temporal Encodings	Data Tensor	Accuracy	Precision	Recall	F1-score
Raw Data	✗	B10	63.93	62.75	**91.43**	74.42
Histogram features	✗	B10	52.46	55.77	82.86	66.67
PCA	✗	B10	57.38	59.18	82.86	69.05
2D_AE	✗	B10	60.11	63.96	73.34	67.78
3D_AE	✗	B10	59.56	62.80	72.38	67.25
2D_AE	✓	B10	59.01	60.07	84.76	70.26
3D_AE	✓	B10	**69.40**	**70.36**	80.95	**75.21**
3D_AE	✓	MVI	52.46	56.27	75.24	64.32
3D_AE	✓	B4	60.65	62.81	76.19	68.84

ing. The results are averaged over three independent executions to account for randomness in initialisation and training.

5.1 Model Comparison

The proposed 3D_AE model is evaluated against four baselines. The first applies k-means directly on the sub-patch tensors, without any feature abstraction. The second uses k-means on histogram features (Wikipedia, nda) computed across all temporal instances. The third, applies Principal Component Analysis (PCA) (Wikipedia, ndb) on the channel dimension, in order to retain the top three principal components. The fourth, 2D_AE, is a 2D convolutional variant of the 3D_AE model. It processes input with temporal instances stacked along the channel dimension and omits temporal encodings, resulting in shape $\mathbb{R}^{H\times W\times C\cdot T}$. We also evaluate the 2D_AE and the 3D_AE model without temporal encodings to assess their impact on performance.

As shown in Table 2, the 3D_AE model with temporal encodings achieves the best overall results, with a high F1-score of 75.21%, demonstrating a strong balance between precision (70.36%) and recall (80.95%). This balance indicates that the 3D model can effectively identify stressed fields while maintaining a relatively low false positive rate. In contrast, the 2D_AE model with temporal encodings exhibits a slightly higher recall, but noticeably lower precision, suggesting a tendency to over-predict stress. This pattern highlights that while temporal encodings enhance both architectures, the ability of 3D convolutions to jointly capture spatial and temporal dependencies provides a more reliable representation of stress progression. When temporal encodings are removed, performance drops for both autoencoders. The F1-score of the 3D variant decreases from 75.21% to 67.25%, while the 2D model falls from 70.26% to 67.78%. The consistent reduction across F1-score, accuracy, precision, and recall highlights the importance of temporal information in distinguishing subtle temporal variations associated with stress development.

The raw data method achieves a high recall, indicating that the original input already contains discriminative temporal and spectral cues useful for identifying stress. However, its lower precision reveals a tendency to over-predict stress, which may result in false positives. A similar trend is observed for the histogram and PCA-based baselines, which maintain relatively high recall but lower precision, leading to moderate F1-scores. These results suggest that while simple feature representations can capture stressed samples, they often have the tendency to over-predict stress.

While B10 is the primary input for model comparison, we additionally evaluate the 3D_AE architecture using two alternative tensors: MVI with multiple vegetation indices, and B4 with four Sentinel-2 bands used to compute MVI (Table 1). The result of this experiment suggests that the choice of input tensor impacts the model performance significantly. As shown in Table 2, the B10 tensor achieves the highest F1-score, highlighting the value of its richer spectral content for feature learning. In comparison, the MVI tensor performs worst, showing that vegetation indices alone may lack the spectral detail needed for effective unsupervised learning. Interestingly, the B4 tensor outperforms MVI, indicating that even limited raw band data can retain more useful information than pre-computed vegetation indices.

5.2 Effect of varying the sub-patch-to-patch threshold

We further examine the impact of varying the sub-patch-to-patch threshold $\alpha \in [0,1]$. A sugar-beet field is labeled as stressed if the proportion of its sub-patches predicted as stressed exceeds the threshold α. Fig. 7 presents the precision-recall and F1-score curves for the 3D_AE model and the best-performing

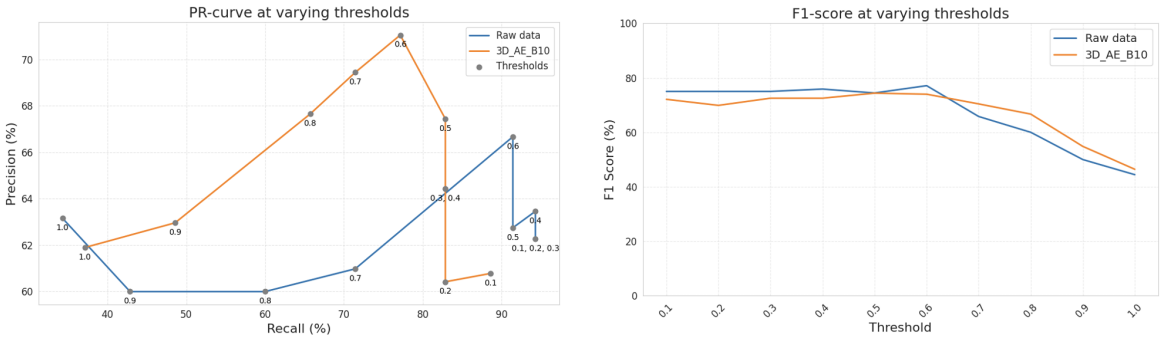

Figure 7: Precision-Recall curve for varying the sub-patch-to-patch threshold (α).

baseline (raw data), with α varying from 0.1 to 1.0.

At lower thresholds, the baseline achieves higher recall than 3D_AE and nearly equivalent F1-scores, highlighting its strength in detecting stressed fields. As the threshold increases, a larger proportion of sub-patches must be predicted as stressed for the entire field to be labeled as stressed. Under these stricter conditions, 3D_AE begins to outperform the baseline, particularly in precision—and consequently in F1-score. This shift reflects the tendency of the baseline model to incorrectly label healthy fields as stressed when the threshold is high. While both models perform comparably around the default threshold $\alpha = 0.5$, 3D_AE shows greater robustness at higher thresholds, maintaining better precision and a slightly higher F1-score. This suggests that the spatiotemporal representations learned by 3D_AE lead to more reliable stress detection under higher thresholds.

5.3 Applying the stress detection system on 2024 data

The developed stress detection system is implemented as a single, modular script that handles data loading, preprocessing, training of the 3D_AE model with temporal encodings, and downstream clustering, producing field-level predictions and localized stress maps. We further apply the this system to the 2024 season, where high rainfall limited the availability of cloud-free observations. Four temporal instances (one each from June to September) were used instead of the default seven. The dataset comprises of 2026 unlabeled fields (50529 sub-patches), with 80% for training and 20% for testing the autoencoder model. Evaluation set includes 40 labeled fields (1213 sub-patches), with an imbalanced distribution (35 stressed, 5 healthy).

As shown in Table 3, the stress detection system achieves a high F1-score with a modest balance of precision and recall, indicating strong performance in detecting stressed instances while minimizing false alarms. The 2024 evaluation set was heavily skewed toward stressed fields. Despite this imbalance, the system successfully identified the stressed instances from healthy. This can be attributed to the severity and consistency of stress signals during the 2024 season, which likely provided clearer spectral distinctions for the model to learn from.

Metric	Score (%)
Accuracy	77.50
Precision	86.09
Recall	88.57
F1-score	87.28

Table 3: Clustering results on 2024 data using 3D_AE with temporal encodings.

Despite being trained with fewer temporal instances, the model captures discriminative spectral–temporal patterns, highlighting the effectiveness of the temporal encodings and its adaptability to real-world conditions, where temporal sequences are often incomplete due to atmospheric constraints. Although the evaluation set was relatively small, limiting the statistical strength of the results, they provide encouraging evidence that the proposed system generalizes well across growing seasons and varying field conditions.

6 CONCLUSION

In this study, we developed a fully unsupervised system for stress detection in sugar-beet fields using publicly available satellite images. The approach requires no labeled training data and specifically highlights the importance of incorporating temporal acquisition dates as part of the model input. Given the limited availability of labeled data, the current results are exploratory in nature. Nevertheless, they demonstrate the potential of unsupervised methods to gen-

erate actionable insights in scenarios with limited labeled data. Moreover, the developed stress detection system is designed to generalise across years, making it easily deployable on future datasets.

7 FUTURE WORK

Once labeled data becomes available, we plan to transition from downstream clustering to supervised classification, enabling more reliable model performance and clearer benchmarking. We also aim to validate our system on the 2025 sugar-beet season once the corresponding ground truth labels are available. This study forms part of a broader effort focused on early stress detection in sugar-beet fields, with the goal of enabling timely interventions and reducing crop losses. As part of this effort, we will explore using temporal data from June to July instead of the full sugar-beet growth cycle from June to September, for facilitating earlier detection. Additionally, while the current pipeline operates at the sub-patch level, future work will explore pixel-level stress detection for more fine-grained analysis.

By providing a reusable and generalisable framework, this study lays the groundwork for a more robust stress detection system, and for advancing unsupervised representation learning in agriculture.

REFERENCES

Cong, Y., Khanna, S., Meng, C., Liu, P., Rozi, E., He, Y., Burke, M., Lobell, D., and Ermon, S. (2022). Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *Advances in Neural Information Processing Systems*, 35:197–211.

European Commission (n.d.). Sugar production. Accessed September 3, 2025.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

Ji, S., Zhang, C., Xu, A., Shi, Y., and Duan, Y. (2018). 3d convolutional neural networks for crop classification with multi-temporal remote sensing images. *Remote Sensing*, 10(1):75.

Raza, M. M., Harding, C., Liebman, M., and Leandro, L. F. (2020). Exploring the potential of high-resolution satellite imagery for the detection of soybean sudden death syndrome. *Remote Sensing*, 12(7):1213.

Rumpf, T., Mahlein, A.-K., Steiner, U., Oerke, E.-C., Dehne, H.-W., and Plümer, L. (2010). Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Computers and electronics in agriculture*, 74(1):91–99.

Sentinel Copernicus (n.d.). Copernicus sentinel-2 collection 1 msi level-2a (l2a). Accessed September 3, 2025.

Sentinel Hub (n.d.). Sentinel-2 collection scene classification map. Accessed September 3, 2025.

Shanmugam, L., Adline, A. A., Aishwarya, N., and Krithika, G. (2017). Disease detection in crops using remote sensing images. In *2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, pages 112–115. IEEE.

Tarasiou, M., Chavez, E., and Zafeiriou, S. (2023). Vits for sits: Vision transformers for satellite image time series. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Victor, B., Nibali, A., and He, Z. (2024). A systematic review of the use of deep learning in satellite imagery for agriculture. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Wikipedia (n.d.a). Image histogram. Accessed September 3, 2025.

Wikipedia (n.d.b). Principal component analysis. Accessed September 3, 2025.

Yu, R., Luo, Y., Zhou, Q., Zhang, X., Wu, D., and Ren, L. (2021). Early detection of pine wilt disease using deep learning algorithms and uav-based multispectral imagery. *Forest Ecology and Management*, 497:119493.

Zhao, C., Cheng, H., and Feng, S. (2021). A spectral–spatial change detection method based on simplified 3-d convolutional autoencoder for multitemporal hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.