Machine Learning pipeline to estimate Soybean Rust severity using UAV-Derived multispectral indices

Salathiel Antunes Teixeira[1*]; Rodrigo Valdivino[2]; Rodrigo Y. Tsukahara[3]; Maurício da Rosa Ribeiro[4]

[1]Researcher/Data Analyst, [2]Research Assistant, [3]Coordinator/Senior Researcher, [4]Researcher.
Biosystems Engineering Departament - Fundação ABC, Castro, Paraná, Brazil.
*salathiel.teixeira@fundacaoabc.org

A paper from the Proceedings of the

17th International Conference on Precision Agriculture and 11th Brazilian Congress on Precision and Digital Agriculture

13-15 July 2026

Porto Alegre, Brazil

Abstract.

Asian Soybean Rust is one of the most destructive diseases affecting soybean crops worldwide and can result in yield losses of up to 90% when control measures are not implemented in a timely manner. Conventional disease monitoring based on field scouting is time-consuming, labor-intensive, and inherently subjective, often failing to adequately represent the spatial variability of disease across production fields. These limitations highlight the need for automated, objective, and high throughput approaches capable of supporting management decisions. The objective of this study was to develop and evaluate an end-to-end machine learning pipeline for real-time monitoring of Asian Soybean Rust severity using multispectral imagery acquired by Unmanned Aerial Vehicles (UAVs). Multispectral data were collected on multiple assessment dates, and corresponding visual disease severity ratings from experimental plots were combined into a single, comprehensive dataset. Individual plots were spatially delineated using circular buffers with a radius of 1 m, within which spectral band values were extracted. Disease severity was defined as the response variable and modeled as a continuous outcome predicted by 18 vegetation indices derived from the multispectral imagery. Two supervised machine learning algorithms were evaluated for this task: Random Forest and Support Vector Machine. Model performance was quantified using independent training and validation datasets, with evaluation based on the coefficient of determination and root mean square error. Both algorithms demonstrated high predictive accuracy during the training phase. However, the Random Forest model showed a pronounced reduction in predictive performance when applied to the validation dataset, suggesting susceptibility to overfitting under the evaluated conditions. In contrast, the Support Vector Machine model maintained stable performance across both training and validation datasets, indicating superior generalization capability. Overall, the results suggest that Support Vector Machine represents a consistent approach for predicting continuous values of Asian Soybean Rust severity from UAV-based multispectral data. The proposed framework demonstrates potential for operational disease monitoring and decision support applications in precision agriculture.

Keywords.

Soybean Rust, machine learning, multispectral imagery, Randon Forest, Support Vector Machine.

Introduction

Soybean (*Glycine max* L. Merrill) cultivation is a cornerstone of global food and economic security; however, its productivity is constantly threatened by fungal pathogens. Among these, Asian Soybean Rust (ASR), caused by the fungus *Phakopsora pachyrhizi*, stands out as the most devastating disease, with the potential to cause yield losses ranging from 10% to 90% if management is not performed in a timely manner (Godoy et al. 2024; Hossain et al. 2024). The rapid dissemination of the pathogen and its ability to cause early defoliation demand rigorous chemical control strategies based on the efficiency of fungicides (Godoy et al. 2024) as well a correct identification and appropriate quantification in plants.

Traditionally, the identification of disease severity relies on visual field scouting, a process that is inherently subjective, labor-intensive, and prone to failures in capturing spatial variability, especially across large areas (Bajwa et al. 2017; Hossain et al. 2024). Furthermore, the timing of fungicide application is critical; delays in detecting the first symptoms can compromise control efficacy and crop profitability (Negrisoli et al. 2022). Although canopy-level leaf spectral monitoring has shown potential for differentiating stress levels, fungicide application alone may not significantly alter the spectral response of plants in certain bands, suggesting that changes detected by remote sensing are predominantly due to disease progress or leaf (Santos et al. 2022).

The evolution of remote sensing, particularly the use of Unmanned Aerial Vehicles (UAVs) equipped with multispectral and hyperspectral sensors, has opened new frontiers for real-time disease monitoring (Feng et al. 2024). Recent studies indicate that integrating hyperspectral data with Machine Learning algorithms allows not only for the classification of disease severity - such as target spot and bacterial wildfire - but also for the early prediction of pathogens even before human visualization (Lay et al. 2023; Otone et al. 2024). Through the analysis of vegetation indices and spectral signatures in the visible and near-infrared regions, deep learning models, such as convolutional neural networks, have achieved high precision in detecting ASR under field conditions (Feng et al. 2024; Ferraz et al. 2024).

Given this scenario, the development of automated, high-throughput tools becomes essential. This study proposes and evaluates an end-to-end machine learning pipeline for real-time monitoring of ASR severity, integrating UAV-derived multispectral indices, aiming to provide a scalable solution for precision management in modern agriculture.

Methodology

Characterization of experiments and location

To evaluate the control of ASR using different management strategies and products, several trials were conducted during the 2023/2024 growing season at the Fundação ABC Research Station, Castro, Paraná, Brazil (Field 3D: -24.79316, -49.90081 and Field 11C: -24.795159, -49.90099). Cultivars sowed were recommended genotypes for location. Fertilization management, weed control, insecticide applications were the same across all trials. Details of the study location and the zones delimited by circles (1 m radius) for spectral band value extraction are shown in Figure 1.

Experimental plots, as well as the different assessment dates and their respective disease severity ratings, were grouped into a single dataset. In total, 1,428 disease severity scores were assigned to the experimental plots by trained evaluators at different reproductive growth stages of the crop.

Data acquisition

Multispectral imagery

Multispectral images were acquired using a UAV-mounted camera MicaSense Altum™

equipped with a multispectral sensor (“MicaSense Altum™ Integration Guide” 2024). A summary of the sensor specifications is provided in

Table 1. Image acquisitions were carried out at different crop development stages, in accordance with the disease severity assessment schedule (Table 2). Flight parameters were set to ensure a ground sample distance (GSD) of 5 cm per pixel.

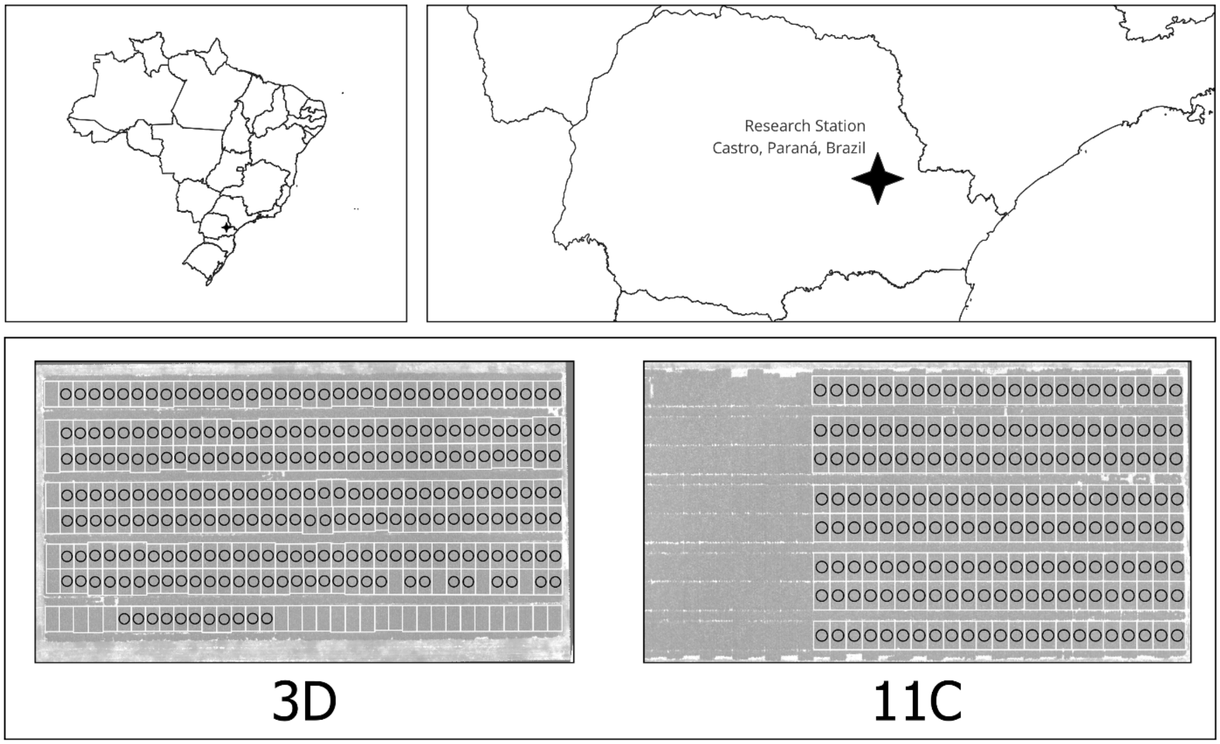

Figure 1. Details of the location of the fields and experiments conducted during the 2023/2024 growing season. Top left – location in Brazil. Top right – highlighted location within the state of Paraná. Bottom – detailed view of the area delimited by 1 m radius circles used for spectral band value extraction (raster images on 06/02/2024).

After the flights, the images were processed and assembled into mosaics that combined the different scenes captured into a single raster using the PIX4Dmapper™ (version 4.8.0) software for each flight date. The raster files from each flight were georeferenced using known coordinates in QGIS (version 3.40.5 – codename “Bratislava”) software. Within the same GIS environment, the experimental plots were properly delineated, and a 1 m radius circular zone at the center of each plot (Figure 1) was used as the sampling area for extracting the values recorded for each spectral band captured by the sensor.

The different experiments conducted had distinct dates of assessments. These differences were experimental in nature and aimed at adapting the trials to different windows of disease occurrence or manegement of treatments applied, as well as to operational constraints. Table 2 presents the dates on which disease severity assessments were performed for each experiment, along with the corresponding flight days after or before the dates of assessments.

Computations and indices determinations

For each flight date and raster image of the trials, several vegetation indices were calculated. The calculations followed the correspondence scheme between spectral regions and sensor bands as follows: Blue = Band 1, Green = Band 2, Red = Band 3, NIR (Near-infrared) = Band 4, and Red-edge = Band 5 (“MicaSense Altum™ Integration Guide” 2024). A summary of the vegetation indices is provided in Table 3. The calculations across different dates and experiments were performed using expressions that are available in the code excerpt provided in the Appendix.

Table 1. Summary of specification and parameters of multispectral sensor MicaSense Altum^{TM}.

Band number	Band name	Spectral region	Center wavelength (nm)
1	Blue	Visible (Blue)	475
2	Green	Visible (Green)	560
3	Red	Visible (Red)	668
4	Near Infrared (NIR)	Near Infrared	840
5	Red Edge	Red Edge	717

Table 2. Summary of growth stages, assessment dates, and days relative to flight across trials.

Trial name	Growth stage (date) and days relative to flight		
	1	2	3
FP23PRSO15	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	R6 (11/03/2024) Flight 3 before
FP23PRSO40	R2 (09/02/2024) Flight 3 before	R5.1 (26/02/2024) Flight 3 after	R5.3 (11/03/2024) Flight 3 before
FP23PRSO41	R2 (09/02/2024) Flight 3 before	R5.1 (26/02/2024) Flight 3 after	R5.3 (11/03/2024) Flight 3 before
FP23PRSO42	R2 (09/02/2024) Flight 3 before	R5.1 (26/02/2024) Flight 3 after	R5.3 (11/03/2024) Flight 3 before
FP23PRSO57	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	R6 (11/03/2024) Flight 3 before
FP23PRSO58	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	R6 (11/03/2024) Flight 3 before
FP23PRSO59	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	R6 (11/03/2024) Flight 3 before
FP23PRSO61	R2 (08/02/2024) Flight 2 before	R5.1 (23/02/2024) Flight 0 day	R5.5 (14/03/2024) Flight 0 day
FP23PRSO62	R2 (08/02/2024) Flight 2 before	R5.1 (23/02/2024) Flight 0 day	R5.5 (14/03/2024) Flight 0 day
FP23PRSO68	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	R6 (11/03/2024) Flight 3 before
FP23PRSO69	R5.1 (23/02/2024) Flight 0 day		R5.5 (14/03/2024) Flight 0 day
FP23PRSO70	R5.1 (23/02/2024) Flight 0 day		R5.5 (14/03/2024) Flight 0 day
FP23PRSO71	R5.1 (23/02/2024) Flight 0 day		R5.5 (14/03/2024) Flight 0 day
FP23PRSO72	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	R6 (11/03/2024) Flight 3 before
FP23PRSO73	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	
FP23PRSO77	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	R6 (11/03/2024) Flight 3 before
FP23PRSO78	R4 (08/02/2024) Flight 2 before	R5.3 (20/02/2024) Flight 3 before	R6 (11/03/2024) Flight 3 before

Table 3. List of vegetation indices and corresponding descriptions.

Acronym	Name	Brief description
Cigreen	Chlorophyll Index Green	Estimates chlorophyll content using the green band; sensitive to nutritional variations.
CVI	Chlorophyll Vegetation Index	Highlights vegetative vigor with emphasis on chlorophyll content.
EVI	Enhanced Vegetation Index	Improves NDVI performance by reducing saturation in dense vegetation areas.
ExG	Excess Green Index	Measures green color dominance; widely used with RGB drone imagery.
GLI	Green Leaf Index	Estimates green leaf proportion in RGB images.
GNDVI	Green Normalized Difference Vegetation Index	NDVI variant uses the green band instead of red.
Intensity	Reflectance or Luminance Intensity	Overall brightness or reflected light intensity; not a spectral index itself.
MTCI	MERIS Terrestrial Chlorophyll Index	Related to chlorophyll content; useful for nitrogen monitoring.
NDRE	Normalized Difference Red Edge	Use the red-edge region; sensitive to plant nutritional status.
NDVI	Normalized Difference Vegetation Index	Classic index for estimating biomass and vegetation vigor.
NDVIre	NDVI Red Edge	NDVI variation using the red-edge band; more sensitive than the classic NDVI.
OSAVI	Optimized Soil-Adjusted Vegetation Index	Reduces soil background effects under low vegetation cover.
RECI	Red Edge Chlorophyll Index	Related to chlorophyll absorption in the red-edge region.
SAVI	Soil-Adjusted Vegetation Index	NDVI variant correcting for exposed soil influence.
SIPI	Structure Insensitive Pigment Index	Detects chlorophyll-to-carotenoid ratio, less affected by canopy structure.
TGI	Triangular Greenness Index	Estimates chlorophyll content using visible reflectance geometry.
TVI	Transformed Vegetation Index	Modified NDVI version to enhance contrast between areas.
VARI	Visible Atmospherically Resistant Index	Estimates vegetation vigor using visible bands; resistant to haze and dust.

Computational approach

Computational Setup and Software Environment

All data processing, vegetation index calculation, and predictive modeling were conducted using the R programming language running in RStudio IDE codename “Kousa Dogwood” (Posit Software - PBC 2025; R Core Team 2024). The core data manipulation and functional programming workflow were built upon the tidyverse (Wickham 2016) ecosystem. For spatial data handling, the terra (Hijmans 2020) and sf (Pebesma 2016) packages were employed to manage raster and vector files, respectively.

Pre-processing and Data Partitioning

Multispectral rasters were processed to extract average reflectance values per experimental plot. Zonal statistics were computed using functions from the exactextractr (Baston 2019), which ensured precision by considering the fraction of pixels covered by each plot polygon (Figure 1). Following extraction, 18 vegetation indices (Table 3) were calculated. The final dataset was consolidated by joining spectral metrics with ground-truth severity notes synchronized by plot ID and flight date.

The dependent variable, Soybean Rust Severity (SRS) range between 0-100%, was treated as a continuous numerical variable. To ensure model robustness and unbiased evaluation, the dataset was partitioned into training (70%) and validation (30%) sets. This split was performed using function from the caret (Kuhn 2007), which preserves the overall distribution of the response variable across both subsets.

Machine Learning Algorithms and Hyperparameter Tuning

In this study, SRS was defined as the target (dependent) variable, while 18 different vegetation indices served as the predictors (independent variables). Two supervised learning algorithms were implemented:

- Random Forest (RF): managed via the ranger (Wright 2015) engine within the tidymodels (Kuhn and Wickham 2018) framework. The model was tuned to find the optimal number of trees increasing step sizes: 25-tree increments from 25 to 100, 50-tree increments from 100 to 1,000, 100-tree increments from 1,000 to 3,000, and 250-tree increments from 3,000 to 5,000, evaluated iteratively. The stabilization point was defined using a convergence threshold where the relative improvement in Root Mean Square Error (RMSE) was less than 0.1%, resulting in the selection of 3,500 trees. The number of variables randomly sampled at each split was set to p - 1 (where p is the number of predictors), and the minimum node size was set to 1 to allow for maximum tree depth.
- Support Vector Machine (SVM): a radial basis function kernel was employed using the kernlab engine. Hyperparameter optimization was conducted via a comprehensive grid search with 5-fold cross-validation, implemented through a custom function utilizing the tidymodels (Kuhn and Wickham 2018). The search space for the cost of constraints (C) and the sigma (ς) parameter was defined using validation values in a logarithmic scale ($2x10^{-3}$ to $2x10^{5}$ for C and $2x10^{-5}$ to $2x10^{3}$ for ς). The "optimal" model was automatically identified using the tune package (part of tidymodels), selecting the parameter combination that minimized the RMSE across the validation folds.

Model’s performances were quantified by comparing observed vs. predicted values using the metrics from the yardstick (Kuhn et al. 2017). The evaluation criteria were the Coefficient of Determination (R^2) and the RMSE (measure unit of target variable). Comparative visualizations, including SRS "Observed vs. Predicted" scatter plots, were generated using ggplot2 (Wickham et al. 2007) and arranged via the patchwork (Pedersen 2019) to provide a comprehensive summary of efficiency in both training and validation phases.

Results and Discussion

The comparison between the two models reveals that while both perform well, however RF model exhibits signs of overfitting (Figure 2). Although RF achieves a near-perfect fit during training (R^2 = 0.991, RMSE = 2.3), its performance drops significantly during validation (R^2 = 0.917, RMSE = 7). The sharp increase in error and the wider dispersion of points around the identity line in the validation plot (Figure 2, B) indicate that the RF model has likely "memorized" the training noise rather than capturing the underlying trend as effectively.

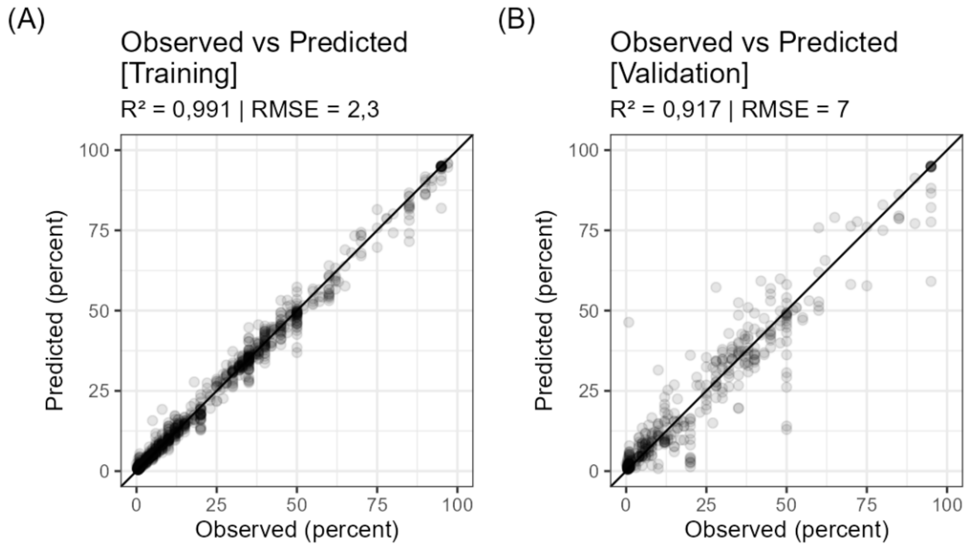

Figure 2. Scatter plots of real versus predicted percentages for the Random Forest (RF) model. The solid line represents the 1:1 identity line, while the model shows a near-perfect fit during (A) training (R^2 = 0.991), the increased dispersion in the (B) validation set with RMSE above training indicates a tendency toward overfitting.

In contrast, the SVM demonstrates superior generalization and stability (Figure 3). The SVM maintains a high consistency between its training (R^2 = 0.95) and validation (R^2 = 0.927) phases (Figure 3, A and B respectivaly), with only a slight increase in the RMSE from 5.4 to 6.6. This suggests that the SVM is robust and less prone to overfitting, making it a reliable choice for predicting these percentage-based outcomes on unseen data. Consequently, the SVM was the more balanced and preferable model for this specific application.

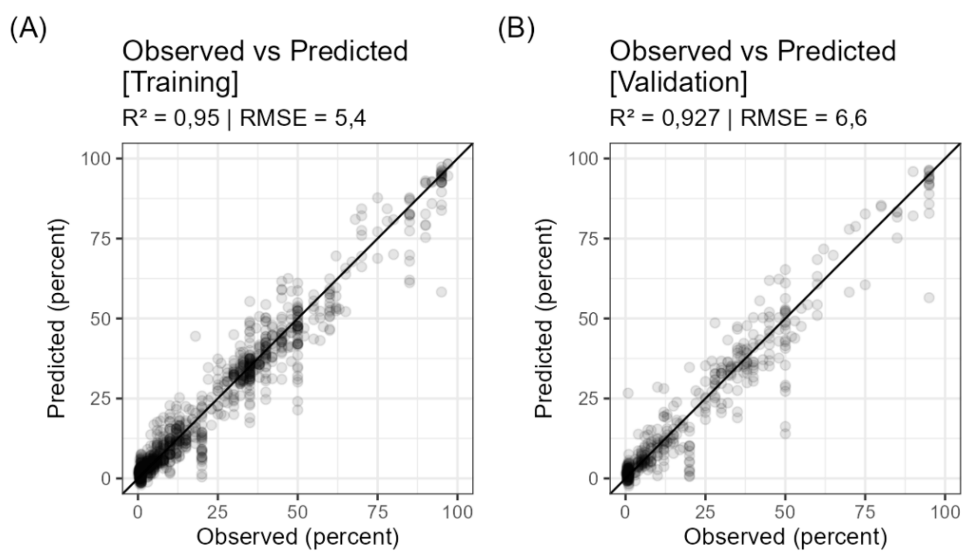

Figure 3. Scatter plots of real versus predicted percentages for the Support Vector Machine (SVM) model. The solid line represents the 1:1 identity line, demonstrating model stability between (A) training (R^2 = 0.95) and (B) validation (R^2 = 0.927) phases.

Despite the strong performance of the SVM, it should be noted that model behavior may vary under different feature sets or sample sizes, and further validation using independent datasets would strengthen these conclusions. This behavior is consistent with reported superior performance of SVM compared to tree-based methods when dealing with high-dimensional spectral inputs, highlighting its robustness and lower sensitivity to overfitting findings (Otone et al. 2024). Yet, although spectral responses may remain stable under different management conditions (Santos et al. 2022), model generalization still depends on algorithmic robustness, reinforcing the superior stability observed for the SVM in this study.

Although RF models have demonstrated high predictive accuracy in soybean disease classification tasks (Ferraz et al. 2024), results in this work indicate that, for continuous percentage prediction, RF may exhibit sensitivity to training data, resulting in reduced generalization performance. The use of a high number of variables randomly sampled at each split likely increased the similarity among individual trees, reducing the diversity introduced by random feature selection. This configuration may have amplified variance, leading to excellent training performance but reduced generalization.

Conclusion

Both machine learning approaches demonstrated strong predictive capability. Random Forest model exhibited signs of overfitting, possible driven by high model complexity and limited feature-level regularization, resulting in reduced performance on unseen data. In contrast, the Support Vector Machine showed stability and consistency between training and validation phases, indicating robustness for continuous percentage prediction under the evaluated conditions. Future studies incorporating independent datasets and alternative feature-selection or hyperparameter strategies are recommended to further validate and extend these conclusions.

References

Bajwa, S., Rupe, J., & Mason, J. (2017). Soybean Disease Monitoring with Leaf Reflectance. *Remote Sensing*, *9*(2), 127. https://doi.org/10.3390/rs9020127

Baston, D. (2019, August 1). exactextractr: Fast Extraction from Raster Datasets using Polygons. https://doi.org/10.32614/CRAN.package.exactextractr

Feng, J., Zhang, S., Zhai, Z., Yu, H., & Xu, H. (2024). DC2Net: An Asian Soybean Rust Detection Model Based on Hyperspectral Imaging and Deep Learning. *Plant Phenomics*, *6*, 0163. https://doi.org/10.34133/plantphenomics.0163

Ferraz, M. A. J., Santiago, A. G. D. S. G., Bruzi, A. T., Vilela, N. J. D., & Ferraz, G. A. E. S. (2024). Defoliation Categorization in Soybean with Machine Learning Algorithms and UAV Multispectral Data. *Agriculture*, *14*(11), 2088. https://doi.org/10.3390/agriculture14112088

Godoy, C. V., Utiamada, C. M., Meyer, M. C., Campos, H. D., del Ponte, E. M., & others. (2024). *Eficiência de fungicidas para o controle da ferrugem-asiática da soja, Phakopsora pachyrhizi, na safra 2023/2024: resultados sumarizados dos ensaios cooperativos* (Circular Técnica No. 206). Londrina, PR: Embrapa Soja. https://www.infoteca.cnptia.embrapa.br/handle/doc/1165843. Accessed 5 January 2026

Hijmans, R. J. (2020, March 20). terra: Spatial Data Analysis. https://doi.org/10.32614/CRAN.package.terra

Hossain, Md. M., Sultana, F., Yesmin, L., Rubayet, Md. T., Abdullah, H. M., Siddique, S. S., et al.

(2024). Understanding Phakopsora pachyrhizi in soybean: comprehensive insights, threats, and interventions from the Asian perspective. *Frontiers in Microbiology*, *14*, 1304205. https://doi.org/10.3389/fmicb.2023.1304205

Kuhn, M. (2007, October 5). caret: Classification and Regression Training. https://doi.org/10.32614/CRAN.package.caret

Kuhn, M., Vaughan, D., & Hvitfeldt, E. (2017, November 12). yardstick: Tidy Characterizations of Model Performance. https://doi.org/10.32614/CRAN.package.yardstick

Kuhn, M., & Wickham, H. (2018, July 27). tidymodels: Easily Install and Load the "Tidymodels" Packages. https://doi.org/10.32614/CRAN.package.tidymodels

Lay, L., Lee, H. S., Tayade, R., Ghimire, A., Chung, Y. S., Yoon, Y., & Kim, Y. (2023). Evaluation of Soybean Wildfire Prediction via Hyperspectral Imaging. *Plants*, *12*(4), 901. https://doi.org/10.3390/plants12040901

MicaSense Altum™ Integration Guide. (2024, August 7). *MicaSense Knowledge Base*. https://support.micasense.com/hc/en-us/articles/360010025413-Altum-Integration-Guide. Accessed 8 January 2026

Negrisoli, M. M., Silva, F. N. D., Negrisoli, R. M., Lopes, L. D. S., Souza Júnior, F. D. S., Freitas, B. R. D., et al. (2022). Impact of Fungicide Application Timing Based on Soybean Rust Prediction Model on Application Technology and Disease Control. *Agronomy*, *12*(9), 2119. https://doi.org/10.3390/agronomy12092119

Otone, J. D. Q., Theodoro, G. F., Santana, D. C., Teodoro, L. P. R., Oliveira, J. T., Oliveira, I. C., et al. (2024). Hyperspectral Response of the Soybean Crop as a Function of Target Spot (Corynespora cassiicola) Using Machine Learning to Classify Severity Levels. *AgriEngineering*, *6*(1), 330–343. https://doi.org/10.3390/agriengineering6010020

Pebesma, E. (2016, October 26). sf: Simple Features for R. https://doi.org/10.32614/CRAN.package.sf

Pedersen, T. L. (2019, December 1). patchwork: The Composer of Plots. https://doi.org/10.32614/CRAN.package.patchwork

Posit Software - PBC. (2025). RStudio: Integrated Development Environment for R. Boston, MA, USA. https://dailies.rstudio.com/rstudio/

R Core Team. (2024). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/

Santos, J. M. D., Batista, P. F., Sperandio, E. M., Nogueira, L. C. A., Gonçalo, T. P., Castoldi, G., et al. (2022). Spectral responses at visible and near-infrared wavelengths of soybean plants to fungicides. *Ciência Rural*, *52*(10), e20210380. https://doi.org/10.1590/0103-8478cr20210380

Wickham, H. (2016, September 9). tidyverse: Easily Install and Load the "Tidyverse." https://doi.org/10.32614/CRAN.package.tidyverse

Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., et al. (2007, June 1). ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. https://doi.org/10.32614/CRAN.package.ggplot2

Wright, M. N. (2015, July 28). ranger: A Fast Implementation of Random Forests. https://doi.org/10.32614/CRAN.package.ranger

Appendix

The snippet following is an excerpt of code to calculate values of different indices used as predictors of SRS in this work. These values were derived from plots located in Field 3D at Fundação ABC Research Station (Castro, Paraná, Brazil). A similar structure, with necessary adaptations, was applied to calculate the indices for the Field 11C plots.

```
# ------------------------------------------
# Packages
# ------------------------------------------
library(terra)
library(sf)
library(exactextractr)
library(dplyr)
library(tools)

# ------------------------------------------
# Statistical Zone Calculation Function
# ------------------------------------------
processa_raster <- function(raster_path, buffer) {
  # read raste as SpatRaster
  r <- rast(raster_path)
  # mean calculation by polygon
  res <- exact_extract(r, buffer, fun = "mean", progress = FALSE)
  df <- as.data.frame(res)
  # detecting ID in shapefile
  id_col <- setdiff(names(buffer), attr(buffer, "sf_column"))
  # cola o campo de ID
  df[[id_col[1]]] <- buffer[[id_col[1]]][seq_len(nrow(df))]
  # raster name (whitout file extention)
  df$raster <- file_path_sans_ext(basename(raster_path))
  # new names of bands
  names(df)[1:nlyr(r)] <- paste0("Banda", 1:nlyr(r))
  return(df)
}


# ------------------------------------------
# Data from Field 3D
# ------------------------------------------
# --- load data ---
dir <- "J:/PROJETOS_SAZONAIS/AGR25 - CARACTERIZACAO RADIOMETRICA DA FERRUGEM SOJA/3D/Dados_Processados_Analise"
buffer <- st_read(file.path(dir, "avaliacao_phakopsora3d_Id_Buffer.shp"))
# --- Rasters ---
raster1 <- "20240206_Micasense_CDECastro3D_GSD5.tif"
raster2 <- "20240223_Micasense_CDECastro3D_GSD5.tif"
raster3 <- "20240314_Micasense_CDECastro3D_GSD5.tif"
rasters <- c(file.path(dir, raster1),
             file.path(dir, raster2),
```

```
        file.path(dir, raster3))

# --- Apply function each raster ---
lista_resultados <- lapply(rasters, processa_raster, buffer = buffer)
# --- Unite all in just one table ---
tabela_final <- bind_rows(lista_resultados) %>%
  mutate(
    data_voo = as.Date(substr(raster, 1, 8), format = "%Y%m%d")
  ) %>%
  rename(
    Blue    = Banda1,
    Green   = Banda2,
    Red     = Banda3,
    Nir     = Banda4,
    RedEdge = Banda5,

  )


tabela_indices_3D <- tabela_final %>%
  mutate(
    # ------- classical indices -------
    NDVI   = (Nir - Red) / (Nir + Red),
    GNDVI  = (Nir - Green) / (Nir + Green),
    NDRE   = (Nir - RedEdge) / (Nir + RedEdge),
    NDVIre = (RedEdge - Red) / (RedEdge + Red),
    #  0.5 and 1.5 used as constant values
    SAVI   = (1.5 * (Nir - Red)) / (Nir + Red + 0.5),
    # 1.16 and 0.16 used as constant values
    OSAVI  = (1.16 * (Nir - Red)) / (Nir + Red + 0.16),
    # ------- Clorophyll -------
    Cigreen = (Nir / Green) - 1,
    RECI    = (Nir / RedEdge) - 1,
    # +eps p/ avoid  div/0
    CVI     = (Nir * Red) / (Green^2 + 1e-6),
    MTCI    = (Nir - RedEdge) / (RedEdge - Red),
    # ------- Others indices -------
    EVI   = 2.5 * ( (Nir - Red) / (Nir + 6*Red - 7.5*Blue + 1) ),
    SIPI  = (Nir - Blue) / (Nir - Red),
    VARI  = (Green - Red) / (Green + Red - Blue),
    ExG   = 2*Green - Red - Blue,
    GLI   = (2*Green - Red - Blue) / (2*Green + Red + Blue),
    # constant values from diff between length wave
    TGI   = -0.5 * ((190*(Red - Green)) - (120*(Red - Blue))),
    TVI   = sqrt(abs(NDVI + 0.5)),
    # intensity / brigth
    Intensity = (Red + Green + Blue) / 3
  ) %>%
  dplyr::select(id_Parcela, data_voo, everything(),
                -c(Blue, Red, Green, RedEdge, Nir, raster))
```