Enhancing Early Disease Diagnosis: Analysis of a Cassava Plant Dataset

YUCETURK SALIH EREN, University of Twente, The Netherlands

Smart agriculture applications have become a common way of letting farmers conduct basic, but timely intervention on their crops. This can save entire regions from crop failure. Areas dependent on subsistence farming, and high in population density are under pronounced risk of unexpected food shortages. This application is still new and more data is required to expand our early reactive capabilities against crop disease. Certain diseases can become fatal for a plant even before symptoms are visible, and early diagnosis can help mitigate crop loss. In this paper, we aim to analyse a dataset of cassava plants. This study contributes to the field of early plant disease diagnosis by improving our understanding of the utility of spectral data in classifying crop diseases. We propose two goals - to investigate and analyze the dataset, and to build, and evaluate supervised models to classify the crop diseases. This study will consider research questions related to disease growth, the usefulness of spectral readings for early diagnosis, and the impact of different classifiers on disease classification.

Additional Key Words and Phrases: Cassava, Manihot esculenta, Early disease detection, Crop diagnosis, Smart farming, CBSD, CMD

1 INTRODUCTION

The cassava plant is an important crop for large parts of Sub-Saharan Africa, and other tropical regions of the world. The crop may be eaten directly or be processed into other kinds of food, like flour, or be used as animal feed. It is chosen for its resilience, and high return potential. Large populations of people depend on the proper functioning of the chain of production of the cassava plant. "More than 500 million people in tropical and sub-tropical Africa, Asia and Latin America" [2] This chain is sensitive to disruptions, as only a few percentage of change may mean millions will not get the food they need, for a price they can afford.

Plant diseases spread quickly around the fields of these crops. A related study [6] done around the region of Lake Victoria, and the Tanzanian coast, has demonstrated the rapid and expansive spread of the cassava brown streak virus. Tanzania alone is larger than twice the size of Germany, and the disease can result in "estimated yield loss of up to 70%" [6].

The two diseases that concern us are the Cassava Brown Streak (CBSD), and the Cassava Mosaic (CMD) Diseases. These diseases initially do not show symptoms [3] on the part of the plant above soil. Therefore by the time symptoms are visible to the farmer, it is too late for the plant. Early intervention can save entire regions from crop failure.

Smartphones are central [4] to the implementation of current smart farming solutions. Recommender systems [1, 10] are a good application of smartphones. The advantage of such an approach is the relatively low cost of implementation, and that it provides basic help immediately to a large area at once. "Conventional smartphones are equipped with several sensors that could be useful to support near real-time usual and advanced farming activities", [7] lowering the risk of late crop disease diagnoses.

The quality of the help a smartphone is able to provide depends on the quality of dataset, data analysis done, and the algorithms used. Early disease recognition applications [5, 8, 13] using smart phone cameras, and machine learning models trained on relevant data are a common use case of this scheme. The camera is not always enough for diagnosing all cases of plant health. The diseases that effect the cassava plant typically don't become visible until it is too late, as mentioned before. Plant health can be better understood through the measure of other kinds of data, particularly spectral data.

In this paper we aim to analyse and interpret a dataset [3] created by Owomugisha et al. This dataset contains spectral data, images, expert rating of plant health, and biochemical data of many cassava plants in different stages of growth.

Based on these, our goals are defined as:

- **Goal 1:** To investigate and analyse the dataset.
- **Goal 2:** To build supervised models to classify the crop diseases.

The following research questions will be kept in mind throughout the research.

- **RQ1:** How is disease growth effected by the screenhouse, versus open field cases?
- **RQ2:** What is the extent of the benefit provided by the spectral readings, for the early diagnosis of diseases?
- **RQ3:** What effect do the different type of classifiers have on disease classification?

By the end of this research we aim to contribute to the field of early disease diagnosis by exploring different modalities for the task of crop disease recognition. This will be done in two ways. The first is by analysing the dataset. Particularly, by investigating the degree of usefulness of the spectral data, and the differences the various descriptors have on the plant health. The second is by using various models to classify the crop diseases. This has the added possibility of being used for scanning cassava plants where it is feasible to take spectral readings of the required kind.

2 RELATED WORK

Labelled Cassava Dataset [12]

A detailed dataset of cassava plants in various stages of growth, with and without diseases. The diseases whose effects are studied are the Cassava Brown Streak Disease, and Cassava Mosaic Diseases. This collection consists of raw spectral data readings, images (taken by smartphone), biochemical (lab) data, and expert scoring of the disease stage. The biochemical data is taken as ground truth in determining the disease onset. The disease scores are from 1 to 3 for field, and 1 to 4 for screenhouse. 1 meaning healthy, and 3/4 meaning diseased. Data has been collected from both a controlled environment (screenhouse), and an open field where the plants are

TScIT 39, July 7, 2023, Enschede, The Netherlands

unsheltered from the natural elements. The data from the screenhouse has been collected for a time period of 19 weeks, and a time period of 15 weeks for the open field.

Recommender System For Disease Detection [10]

An investigation of open field crop disease monitoring with real time information feedback to three farmers. Such systems can provide basic help to many farmers over a large area. This implementation also works offline as farmers are not always connected to the internet. It uses natural language processing, and question and answer pairs to give the useful answer to the farmer. Multiple models are built and tested against each other. It is pointed out in the paper that the a major limiting factor for the quality of answers is the volume of available data.

Detection Of Plant Diseases Using Spectral Data [11]

The disease may not be visible on the cassava plant leaf by visible light, but signs may be picked up by invisible spectral data. This is a paper aimed at "developing methods for diagnosing cassava diseases before they are visibly symptomatic on the plant" [11]. This method uses spectral data and presupposes that using it can result in a significant time advantage over visible light. It builds upon past research [9] on this method, and presents the methodology of applying this technique. The plants are grown in a closed, controlled environment. The data is preprocessed, and dimensional reduction using the PCA model is applied. Multiple training models are used with both the original, and PCA data, and their performances are compared. This study demonstrates the viability of detecting disease using spectral readings. The steps taken for analysing spectral data in our paper, follows those presented in this study.

3 METHODOLOGY

The dataset includes spectral, and image data, alongside other feature descriptors. The complete analysis is divided into four parts; exploratory analysis, spectral classification, image classification, and combined analysis. Each one of these parts have been coded in a separate Jupyter notebook. The first section, *Exploratory Analysis* serves to introduce the dataset. The data is plotted in various ways to answer a number of questions. *Spectral Classification* and *Image Classification* contain code that prepares relevant data, presents some example cases, and feeds them to supervised classification algorithms. Finally, the *Combined Analysis* section combines the model training of both the image and spectral data into one notebook and compares their performances directly. The description of the process for each of these parts will be presented in the following sections.

3.1 Preprocessing

3.1.1 Exploratory Analysis.

The dataset consists of multiple .csv files, and hundreds of .jpg files. The spectral readings for each experiment group (field/screenhouse) are stored in two .csv files; labelled "b", and "g". The reason for having two spectral readings instead of one is because the sensor device reads only a single point on the leaf. Therefore, having two readings on different positions of each leaf will increase the information recorded.

The data-frames are imported and preprocessed. The "b", and "g" readings are joined into one. Inconsistencies in the column names of the dataset are removed, and rows of data with missing values for the disease_class and image_name columns are removed. At the end of this process, we end up with two data-frames. One containing field readings, and the other containing the screenhouse readings. The columns of these data-frames are listed below:

Spectral Data (many columns, both "b" and "g"), **disease_class**, **week**, **cassava_variety**, **plant_number**, **leaf_number**, **image_name**, **chemical_test**, **expert_score**, **image_label**.

Spectral, and image classification have been done on these two data-frames. Each row in these data-frames represents a reading from a single leaf.

After the preprocessing, the data is visualised according to these queries:

- What is the number of leaves per week, for each disease_class?
- What is the number of leaves per week, for each expert_score?
- How do the chemical readings for all leaves change by week?
- How do the chemical readings for leaves grouped by disease_class change by week?
- How do the mean, median and standard deviation of chemical readings, grouped by disease_class change by week?
- Plot the spectral data onto 2-d space using PCA.
- Plot the spectral data onto 2-d space using t-SNE.
- Plot the spectral data onto 3-d space using PCA.
- Plot the spectral data with 4 PCs.
- Plot the spectral data with 10 PCs.

The results of the exploratory analysis will be discussed in detail under the *The Dataset* section.

3.1.2 Spectral Classification.

This section describes the process of analysing and classifying the spectral data. The aforementioned data-frames contain the spectral readings, but they will need to be processed before feeding them to classification algorithms.

The first 4 weeks of disease data are removed. Healthy data for all weeks are kept. This is because the first 4 weeks of diseased data do not provide much information and can be removed without significant loss. The noise in data is then reduced using a simple moving average algorithm. All rows of spectral data are rewritten using this moving average algorithm with a window length of 10 columns.

The next step is for the data to be regularised. The spectral readings are scaled and this makes each (spectral) feature have a mean of 0 and a standard deviation of 1. This ensures that all features are on a similar scale and have a comparable impact on the classifiers.

Two versions of the same data will be compared. One with the complete spectral readings, and another version with reduced number of spectral features using a dimensionality reduction algorithm. Plotting the cumulative variance allows us to choose the appropriate number of principal components. *Fig. 16* shows the explained cumulative variance by the number of components for both sets of data. Alternate versions of both sets of data are created using the PCA algorithm with 10 as the number of components. The final data-frames and their columns are given in the table below:

Table 1. Spectral data-frames

Data-frame	Columns
field_image_spectral_averaged_df_scaled	disease_class, image_name, Spectral Features
screenhouse_image_spectral_averaged_df_scaled	disease_class, image_name, Spectral Features
field_image_pca_df	disease_class, image_name, PC Features
screenhouse_image_pca_df	disease_class, image_name, PC Features

3.1.3 Image Classification.

This section describes the similar process done on the image data. The number of images present are fewer than the total number of rows for both field and screenhouse data. Therefore the rows for which an image isn't present are filtered from the data-frames.

Out of the 3638 rows of field data, 1074 of them have an image. 808 healthy, 23 CBSD, and 243 CMB cases are present. For screenhouse data, out of the total of 1361 rows, 351 images are present. 249 healthy, and 102 cases of CBSD are present. There are no images for CMD. *Fig. 17* shows examples of some images.

Feature extraction is applied to these images. We use a pretrained ResNet-18 model for this purpose. This model has already been trained on a large dataset and has learned weights and parameters. We focus on the "avgpool" layer of the ResNet-18 model. This layer performs average pooling, reducing the spatial dimensions of the feature maps to 512 while retaining important information.

Before feeding the images into the model, we apply preprocessing steps. Resizing the images to a standard size, normalizing the pixel values to a common range, and converting them into a suitable format for the model. Each image is:

- resized to 224 by 224 pixels
- converted to a PyTorch tensor
- normalized the pixel values of the image
- put into a a zero vector of size 512

The features are concatenated to their data-frame. The final data-frames and their columns are given in the table below:

Table 2. Image data-frames

Data-frame	Columns
filtered_field_feature_vectors_disease_class_df	disease_class, image_name, Image Features
filtered_screenhouse_feature_vectors_disease_class_df	disease_class, image_name, Image Features

3.1.4 Combined Analysis.

The purpose of this section is to directly compare the usefulness of the spectral data versus the image data in detecting and classifying plant diseases. It uses the data-frames prepared under the *Spectral Classification*, and *Image Classification* sections.

Since not all rows have an image present, the corresponding rows of data for the spectral readings also need to be removed to compare them meaningfully. We end up with 4 data-frames. They are described in the table below:

Table 3. Spectral and image data-frames

Data-frame	Columns
filtered_field_image_spectral_averaged_df_scaled	disease_class, image_name, Spectral Features
filtered_screenhouse_image_spectral_averaged_df_scaled	disease_class, image_name, Spectral Features
filtered_field_feature_vectors_disease_class_df	disease_class, image_name, Image Features
filtered_screenhouse_feature_vectors_disease_class_df	disease_class, image_name, Image Features

3.2 Classification

The spectral classification, image classification, and combined analysis all use the same classification models. The following models are used:

- XGBoost: xgb.XGBClassifier(objective='multi:softmax')
- Sklearn: KNeighborsClassifier(n_neighbors=3)

To monitor the model's performance during training, we define an evaluation set consisting of the training and validation data. We use early stopping based on validation loss to prevent overfitting. The early stopping rounds parameter is set to 5. The evaluation metrics used are multi-class error rate (merror) and logarithmic loss (mlogloss).

Other metrics; accuracy, precision, recall, and a confusion matrix are recorded. The accuracy score measures the overall correctness of the predictions. Precision shows the proportion of correctly predicted positive cases out of all cases predicted as positive. Recall measures the proportion of correctly predicted positive cases out of all actual positive cases. The confusion matrix visually represents the predicted versus actual class labels. We then calculate True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) for each class.

To make it easier to interpret the the confusion matrix, we plot two versions: one with actual values and another with percentage values, giving 2 ways of seeing the the distribution of predictions.

We then provide examples of correctly and incorrectly classified images for each class. These qualitative examples allow us to comment more on its performance.

Finally, we visualize the training and validation mlogloss and merror over epochs, giving information about the model's learning progress and potential overfitting.

4 THE DATASET

4.1 Basics

The tables below show detailed information about our data-frames.

Table 4. Information on the field data-frame

Query	Value
Number of rows	3638
Number of columns	7301
Number of unique values per column:	
disease_class	3
week	14
cassava_variety	3
plant_number	10
leaf_number	3
image_label	3638
image_name	3638
expert_score	3
chemical_test	969
Value sets of some columns:	
disease_class	1 (%46), 2 (%26), 3 (%29)
expert_score	1 (%54), 2 (%35), 3 (%12)

Table 5. Information on the screenhouse data-frame

Query	Value
Number of rows	1361
Number of columns	7299
Number of unique values per column:	
disease_class	3
week	17
cassava_variety	3
plant_number	10
leaf_number	1
image_label	1361
image_name	1351
expert_score	4
chemical_test	1160
Value sets of some columns:	
disease_class	1 (%48), 2 (%25), 3 (%28)
expert_score	1 (%87), 2 (%11), 3 (%4), 4 (%1)

The field data-frame has 3638 rows, while the screenhouse data-frame has 1361 rows. The field data-frame contains a larger amount of data compared to the screenhouse data-frame. However the screenhouse data-frame has a longer observation period, a different distribution of expert scores, and a wider range of unique values for chemical test results compared to the field data-frame. For field, the "leaf_number" column has three different leaf numbers. However, in the screenhouse data-frame, the "leaf_number" column has only 1 unique value, suggesting that only one leaf per plant. In the field data-frame, both "b_expert_score" and "g_expert_score" columns have 3 unique values each, whereas in the screenhouse data-frame, these columns have 4 unique values each.

4.2 Principal Component Visualization

Plots of the spectral data in 2-d and 10-d space and shown below. 3-d and 4-d plots can be found under the *Appendix*.

Fig. 1. Visualisation of field spectral data using PCA

Fig. 2. Visualisation of screenhouse spectral data using PCA

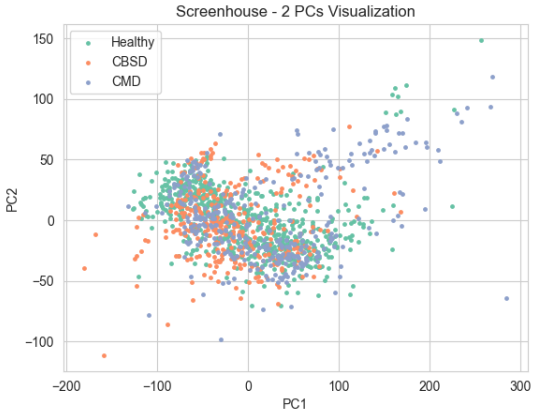


Fig. 3. Visualisation of field spectral data using t-SNE

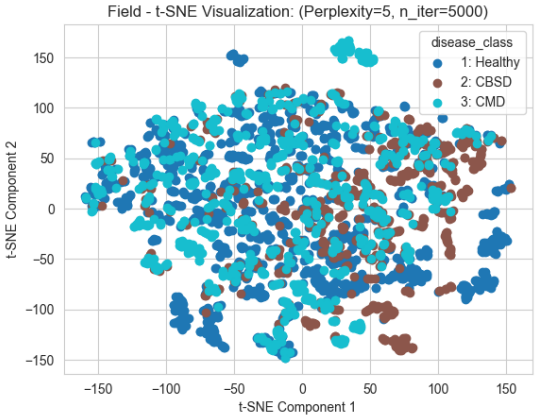
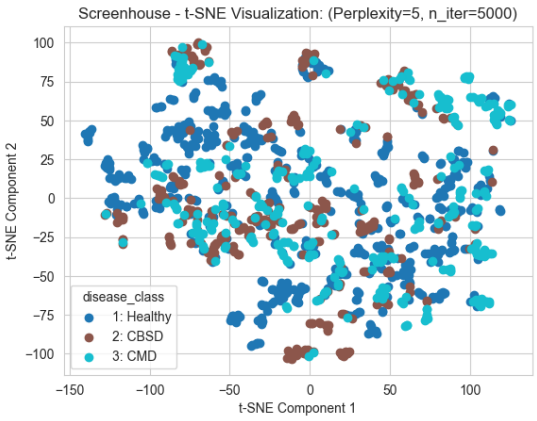


Fig. 4. Visualisation of screenhouse spectral data using t-SNE



The two plots above fail to show distinct cluster patterns. This could be due to several factors. Firstly, the data points are distributed in a continuous manner and this might make it challenging to identify distinct groups. Additionally, PCA and t-SNE are dimensionality reduction techniques, and if the patterns in the data are spread across many dimensions, these methods may struggle to represent the data in a lower dimensional space. Moreover, datasets with high variability or heterogeneity make it difficult for clustering algorithms to identify clear clusters. Lastly, some datasets may lack clear clusters due to their inherent complexity.

Fig. 5. Visualisation of field spectral data using PCA

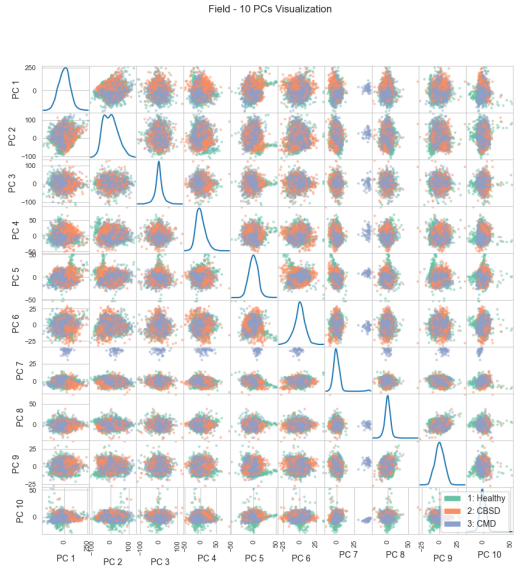
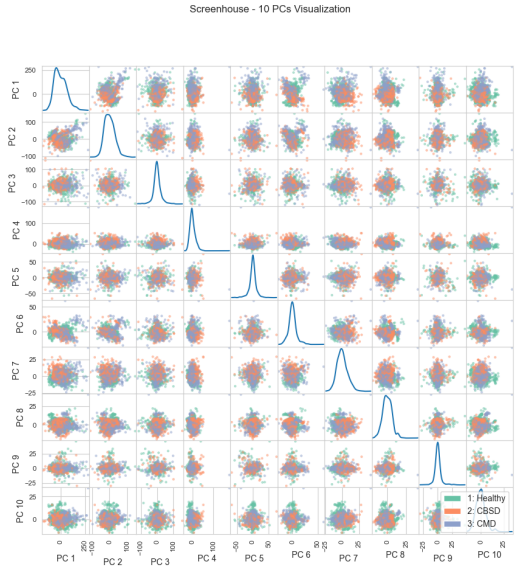


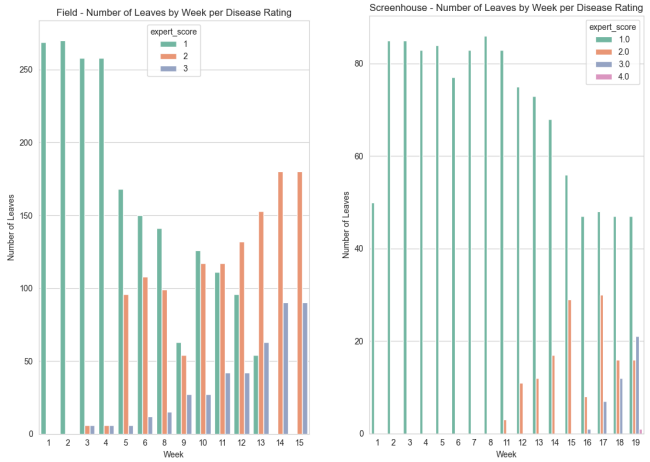
Fig. 6. Visualisation of screenhouse spectral data using PCA



Visualising the data in 10 dimensions starts to show some clustering for a subset of values. This suggests that the spectral data clusters are too complex to be distinctly visualised in lower dimensions. This is not unexpected since the original data readings have a high number of dimensions.

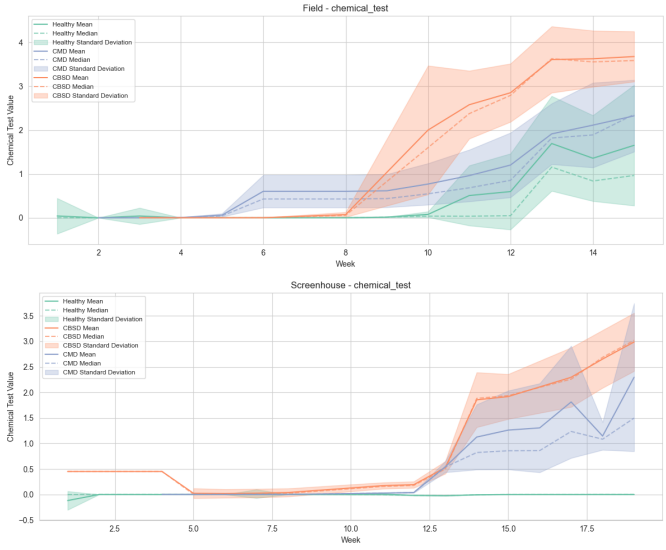
4.3 Disease Spread

Fig. 7. Number of leaves per expert score by week for field and screenhouse data



For both datasets, the number of healthy leaves steadily decrease with each week. The number of healthy plants for screenhouse are always a majority, even at the end of the experiment. For field data, the healthy plants become a minority by the middle of the experiment, and at the end, completely healthy plants are nonexistent.

Fig. 8. Chemical readings per disease class by week, for field and screenhouse data



The above plots of chemical readings show a clear difference between the healthy, CBSD, and CMD classes. The chemical readings are taken as ground truth.

5 RESULTS

The table below lists the performances metrics for all training data. The "filtered" label indicates that the rows in that data-frame for which an image isn't present have been removed. This insures parity between field and screenhouse data when comparing them directly. The expanded table with FP, TP, FN, TN can be found under the *appendix*. Examples of disease class prediction have also been provided.

Table 6. Results table of classification models

Training Data	Model	Performance Metrics			
		Accuracy	Precision	Recall	F-score
Field (Non-PCA) - Spectral	XGBoost	0.856	0.854	0.856	0.851
	KNN	0.724	0.729	0.724	0.723
	Average	0.790	0.791	0.790	0.787
Field (PCA) - Spectral	XGBoost	0.689	0.674	0.689	0.675
	KNN	0.606	0.596	0.606	0.596
	Average	0.648	0.635	0.648	0.636
Screenhouse (Non-PCA) - Spectral	XGBoost	0.816	0.814	0.816	0.812
	KNN	0.704	0.711	0.704	0.705
	Average	0.760	0.763	0.760	0.759
Screenhouse (PCA) - Spectral	XGBoost	0.730	0.727	0.730	0.717
	KNN	0.644	0.632	0.644	0.625
	Average	0.687	0.680	0.687	0.671
Field - Image	XGBoost	0.762	0.698	0.762	0.698
	KNN	0.762	0.736	0.762	0.744
	Average	0.762	0.717	0.762	0.721
Screenhouse - Image	XGBoost	0.647	0.639	0.647	0.643
	KNN	0.718	0.705	0.718	0.711
	Average	0.683	0.672	0.683	0.677
Field (Filtered) - Spectral	XGBoost	0.936	0.935	0.936	0.930
	KNN	0.882	0.873	0.882	0.877
	Average	0.909	0.904	0.909	0.904
Field (Filtered) - Image	XGBoost	0.861	0.817	0.861	0.808
	KNN	0.851	0.810	0.851	0.824
	Average	0.856	0.814	0.856	0.816
Screenhouse (Filtered) - Spectral	XGBoost	0.942	0.946	0.942	0.940
	KNN	0.914	0.914	0.914	0.914
	Average	0.928	0.930	0.928	0.927
Screenhouse (Filtered) - Image	XGBoost	0.714	0.659	0.714	0.671
	KNN	0.771	0.757	0.771	0.761
	Average	0.743	0.708	0.743	0.716

6 DISCUSSION

In this section the results are interpreted, compared, and commented on.

6.1 Spectral

6.1.1 *PCA vs. Non-PCA.*

A performance drop when using PCA for the "Field - Spectral" and "Screenhouse - Spectral" data is apparent. The accuracy, precision, recall, and F-score values are consistently lower for the PCA data compared to the non-PCA. The F-scores for the non-PCA data range between 0.851 and 0.705. Those with PCA have F-scores between 0.596 and 0.717, indicating a decrease in the overall balance between precision and recall.

The difference in performance between the non-PCA and PCA versions can be attributed to the nature of Principal Component Analysis. PCA reduces the dimensionality of the data by projecting it onto a lower-dimensional space while maximizing the variance. This reduction may cause a loss of information, including relevant features that contribute to accurate classification. Consequently, the reduced feature set used in the PCA version may lead to decreased discrimination power and result in lower accuracy, precision, recall, and F-scores compared to the non-PCA version.

6.1.2 *Screenhouse vs. Field.*

Comparing the performance values of the field and screenhouse data, we can observe that field generally outperformed screenhouse in terms of accuracy, precision, recall, and F-score. Particularly for class 0 (appendix).

One potential reason for the difference in performance may be due to the field environment having a more diverse range of conditions, leading to a larger variation in spectral patterns and potentially making it easier for the models to differentiate between classes. In contrast, the screenhouse environment is more controlled, resulting in less variation and potentially making classification more challenging. Another reason could be that the spectral characteristics captured by the sensors in the field and screenhouse environments could be different, causing variations in the performance.

6.2 Image

We can see that the "Field - Image" model generally outperforms the "Screenhouse - Image" model in terms of accuracy, precision, recall, and F-score. Field data achieves an accuracy of 0.762 whereas screenhouse has an accuracy of 0.647 and 0.718. It is worth noting that KNN performs slightly better than the XGBoost model.

The number of images present are 1074 for field, and 351 for screenhouse. Having more images in the field dataset increases the chances of capturing a wider range of variations in plant growth, lighting conditions, and other factors that affect image features. On the other hand, the screenhouse dataset, with a smaller number of images, might suffer from limited representation and potential bias.

6.3 Combined

Looking at field and screenhouse data, field spectral data performs worse than screenhouse spectral data for both models. However, for images the field data performs better than screenhouse. Overall, there doesn't seem to be a major performance difference between field and screenhouse data. Image data from screenhouse performs the worst out of the four data sets of this section. This is consistent with the results from the *Image* section.

Comparing image and spectral data, we can observe that both models achieve high accuracy scores for spectral, with values between 0.882 and 0.942. Accuracy values of models for image data are between 0.714 and 0.861. Image data also has worse F-scores. Between 0.877 and 0.940 for spectral and between 0.671 and 0.824 for image. This indicates that for image data, the classification model's performances in terms of balancing precision and recall is worse.

The spectral data provides richer information for classification tasks in this context. When compared directly, and only for those leaves that a image exists, the spectral data performs better.

Spectral data captures the reflectance properties of different wavelengths, providing more detailed information about the leaf than the

image taken from a phone camera. The spectral data, being more specific to the properties of the samples, appears to be better suited for distinguishing between different disease classes in this particular comparison.

6.4 XGBoost vs. KNN

For the image data, the XGBoost model tends to have higher accuracy, precision, recall, and F-score compared to the KNN model on average. Similarly, for spectral data, XGBoost generally performs better than KNN. However, it's worth noting that the performance difference between the two models is relatively small. XGBoost consistently shows slightly higher performance.

XGBoost is based on ensemble learning, and uses gradient boosting algorithms. Ensemble learning is a learning technique that combines multiple individual models to create a stronger predictive model. Gradient boosting is a method where base models are trained sequentially, and each subsequent model is trained to correct the mistakes of the previous models. The models are combined by giving more weight to the models that perform well. XGBoost generally performs well with high-dimensional data.

On the other hand, KNN is a lazy learning algorithm that classifies data points based on the majority vote of their nearest neighbors. It is comparatively a simple algorithm that performs well when the data has clear boundaries or local structures. It tends to work better with low-dimensional data where local proximity plays the primary role in determining the class labels.

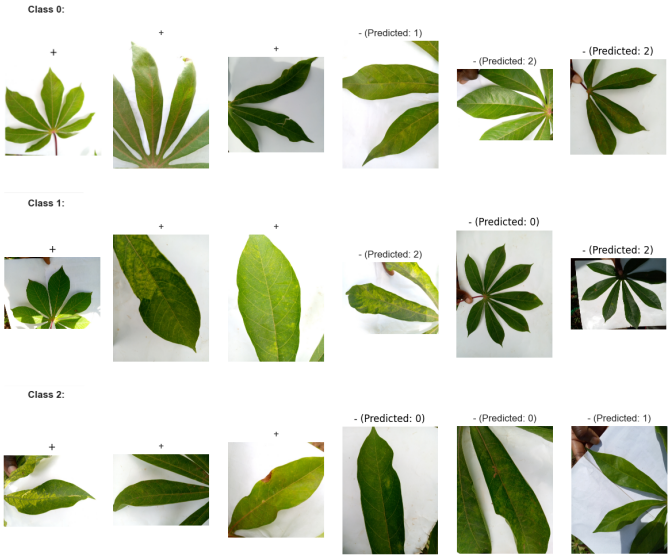
Therefore, the reason why KNN performs worse may be due to it struggling to capture the complex patterns present in high-dimensional data, where picking up on global relationships are necessary for accurate classification.

6.5 Example Prediction Cases

A collection of example cases of disease class predictions are given below. It is worth noting that the correct predictions all have uniform lighting, background and the photo contains only the leaf itself in clear focus. Similarly, the incorrect predictions appear to have been taken under different lighting conditions. The experimenter's hand is visible in some, and the background of the leaf is not always entirely covered by the white sheep of paper.

The preprocessing done on both the images and spectral data were minimal. To improve disease class predictions, expanding the preprocessing of the images by standardizing lighting, removing the background, cropping the leaf, aligning the image, and normalizing colors should be considered. These steps will enhance image consistency, reduce variations, and aid the model in accurately predicting disease classes.

Fig. 9. Examples of disease class prediction



7 CONCLUSION

This study aimed to investigate the utility of spectral in comparison to image data for classifying crop diseases, and explore the impact of different classifiers on disease classification.

The results revealed that spectral data outperformed image data in predicting diseases. Image data showed slightly lower performance, indicating that the classification models struggled to balance precision and recall when using image-based features. Spectral data provided more detailed information about the leaves and proved to be better suited for distinguishing between different disease classes.

Among the classifiers evaluated in this study, the XGBoost model consistently performed better than the KNN model. This performance advantage may be attributed to XGBoost's ability to capture complex patterns and relationships in high-dimensional data, which is crucial for accurate disease classification.

This study contributes to the field by highlighting the advantage of utilizing spectral data for early disease diagnosis in crop plants. By demonstrating the superior performance of spectral data over image data, we emphasize the potential of spectral readings for timely disease detection.

Furthermore, the comparison of different classifiers provided insights into the potential edge that ensemble learning algorithms like XGBoost in handling high-dimensional data for disease classification may have over "lazy learning".

Future research in this field can explore several avenues for improvement. Firstly, addressing the limitations of this study, such as the relatively small size of the screenhouse image dataset, or the limited preprocessing of both the image and spectral data can enhance the generalizability and robustness of the findings.

Secondly, the study can be expanded by including other classification models of different types, which can provide a more comprehensive evaluation of the performance of various algorithms. By including these additional models with different algorithmic

types, we can perform a more comprehensive evaluation of various classification approaches. This expanded analysis will provide a better understanding of which algorithms are most effective for crop disease classification, and potentially reveal alternative models that outperform ensemble algorithms.

REFERENCES

[1] Soham Chakraborty and Sushruta Mishra. [n. d.]. A Smart Farming-Based Recommendation System Using Collaborative Machine Learning and Image Processing. In *Cognitive Informatics and Soft Computing* (Singapore, 2022) *(Lecture Notes in Networks and Systems)*, Pradeep Kumar Mallick, Akash Kumar Bhoi, Paolo Barsocchi, and Victor Hugo C. de Albuquerque (Eds.). Springer Nature, 703–716. https://doi.org/10.1007/978-981-16-8763-1_58

[2] Mabrouk A. El-Sharkawy. [n. d.]. Cassava biology and physiology. 56, 4 ([n. d.]), 481–501. https://doi.org/10.1007/s11103-005-2270-7

[3] Godliver Owomugisha, Joyce Nakatumba-Nabende, Joshua Jeremy Dhikusooka, Estefania Taravera, Ephraim Nuwamanya, and Ernest Mwebaze. [n. d.]. A Labeled Spectral Dataset with Cassava Disease Occurrences using Virus Titre Determination Protocol. ([n. d.]), 7.

[4] Godwin Idoje, Tasos Dagiuklas, and Muddesar Iqbal. [n. d.]. *Survey for smart farming technologies: Challenges and issues | Elsevier Enhanced Reader.* https://doi.org/10.1016/j.compeleceng.2021.107104

[5] Yan Guo, Jin Zhang, Chengxin Yin, Xiaonan Hu, Yu Zou, Zhipeng Xue, and Wei Wang. [n. d.]. Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming. 2020 ([n. d.]), e2479172. https://doi.org/10.1155/2020/2479172 Publisher: Hindawi.

[6] D. R. Mbanzibwa, Y. P. Tian, A. K. Tugume, S. B. Mukasa, F. Tairo, S. Kyamanywa, A. Kullaya, and J. P. T. Valkonen. [n. d.]. Simultaneous virus-specific detection of the two cassava brown streak-associated viruses by RT-PCR reveals wide distribution in East Africa, mixed infections, and infections in Manihot glaziovii. 171, 2 ([n. d.]), 394–400. https://doi.org/10.1016/j.jviromet.2010.09.024

[7] Jorge Mendes, Tatiana M. Pinho, Filipe Neves dos Santos, Joaquim J. Sousa, Emanuel Peres, José Boaventura-Cunha, Mário Cunha, and Raul Morais. [n. d.]. Smartphone Applications Targeting Precision Agriculture Practices—A Systematic Review. 10, 6 ([n. d.]), 855. https://doi.org/10.3390/agronomy10060855 Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[8] Rodrigo Moreira, Larissa Ferreira Rodrigues Moreira, Pablo Luiz Araújo Munhoz, Everaldo Antônio Lopes, and Renato Adriane Alves Ruas. [n. d.]. AgroLens: A low-cost and green-friendly Smart Farm Architecture to support real-time leaf disease diagnostics. 19 ([n. d.]), 100570. https://doi.org/10.1016/j.iot.2022.100570

[9] Ephraim Nuwamanya, R. Patrick, Settumba Mukasa, Samuel Kyamanywa, Joseph Hawumba, and Baguma Yona. [n. d.]. Influence of spectral properties on cassava leaf development and metabolism. 13 ([n. d.]), 834–843. https://doi.org/10.5897/AJB2013.12795

[10] Jonathan Omara, Estefania Talavera, Daniel Otim, Dan Turcza, Emmanuel Ofumbi, and Godliver Owomugisha. [n. d.]. A field-based recommender system for crop disease detection using machine learning. 6 ([n. d.]), 1010804. https://doi.org/10.3389/frai.2023.1010804

[11] Godliver Owomugisha, Ephraim Nuwamanya, John A. Quinn, Michael Biehl, and Ernest Mwebaze. [n. d.]. Early detection of plant diseases using spectral data. In *Proceedings of the 3rd International Conference on Applications of Intelligent Systems* (New York, NY, USA, 2020-02-17) *(APPIS 2020)*. Association for Computing Machinery, 1–6. https://doi.org/10.1145/3378184.3378222

[12] Owomugisha Godliver, Biehl Michael, Joyce, Nakatumba-Nabende, Ephraim Nuwamanya, Dalton Kanyesigye, Nicholas Muhumuza, Maurice Katusiime, Joshua Jeremy Dhikusooka, Tobius Saolo, Bamundaga Aloyzius, Joan Nabadda, Nakalyango Molly, and Nahima Musa. [n. d.]. Cassava Spectral and Image Dataset. https://doi.org/10.7910/DVN/R0KL7R

[13] Diego Inácio Patrício and Rafael Rieder. [n. d.]. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. 153 ([n. d.]), 69–81. https://doi.org/10.1016/j.compag.2018.08.001

8 APPENDIX

A EXPLORATORY ANALYSIS

Fig. 10. Leaves by week per disease class for field and screenhouse data

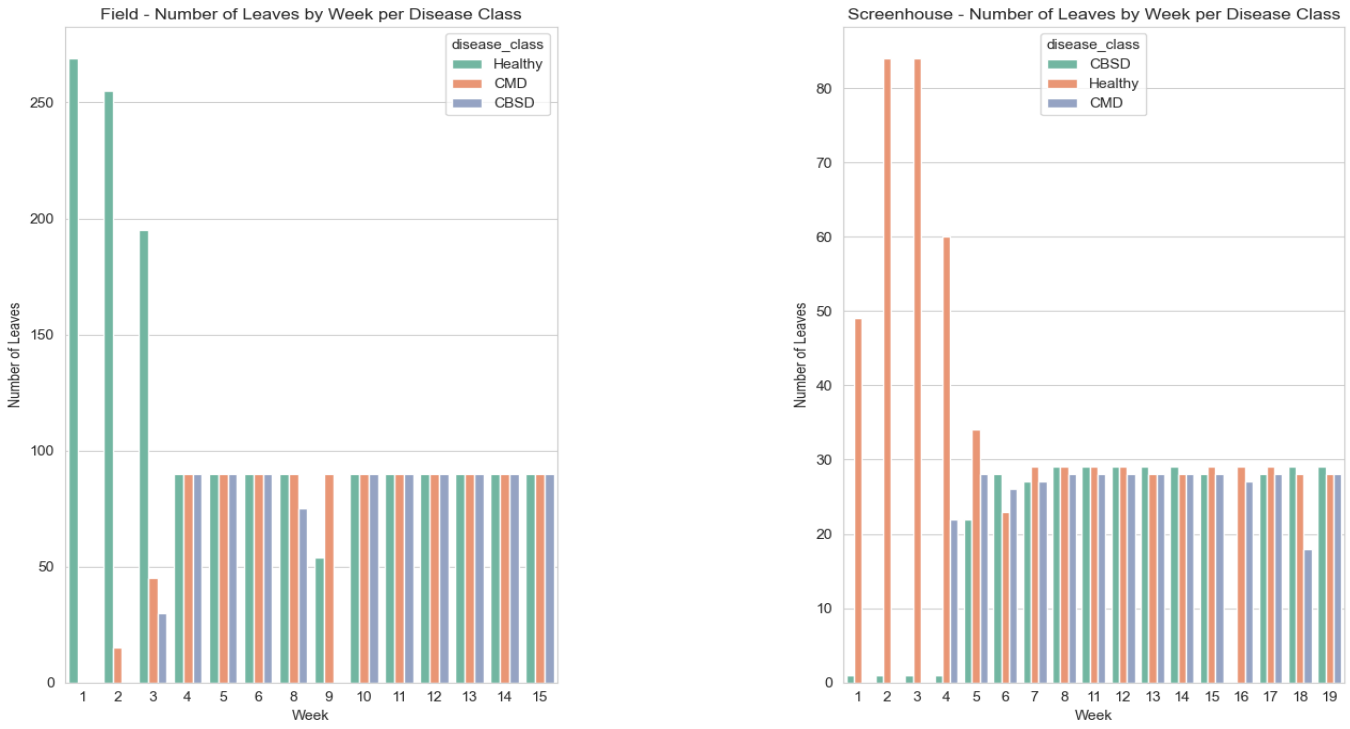


Fig. 11. Chemical readings by week for field and screenhouse data

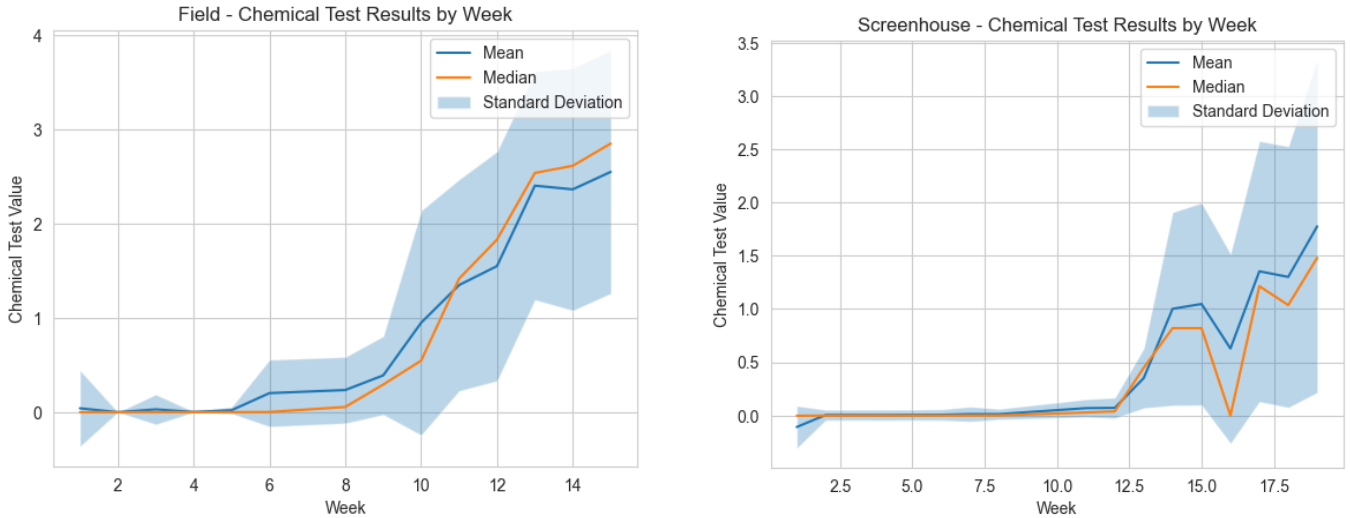


Fig. 12. Chemical readings of leaves per disease class by week for field and screenhouse data

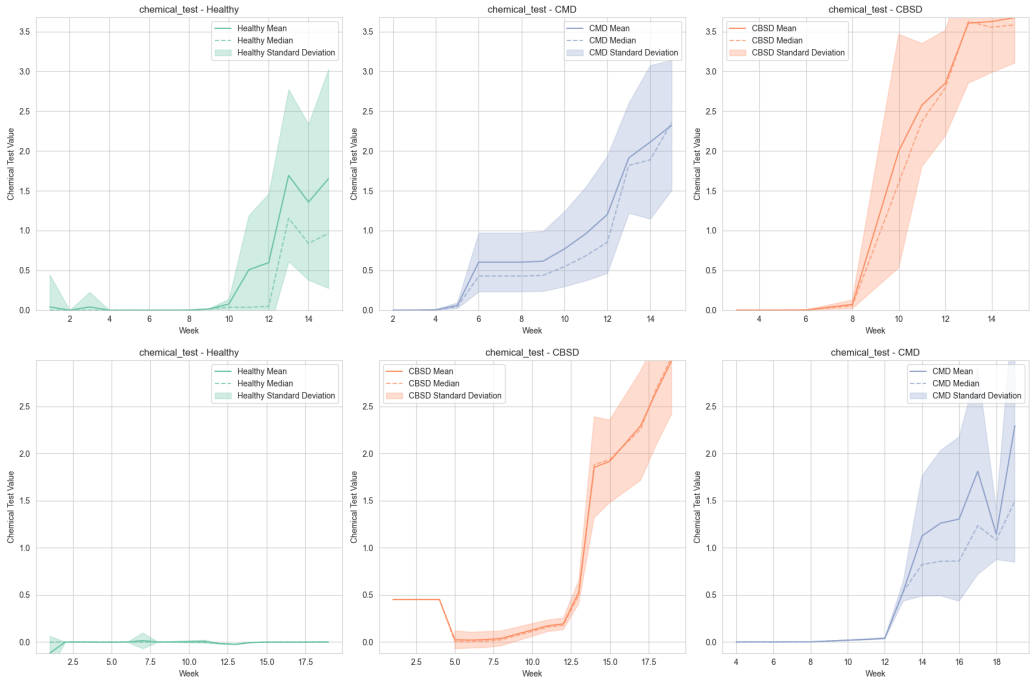


Fig. 13. Mean, median and standard deviation of chemical readings by week per disease class for field data

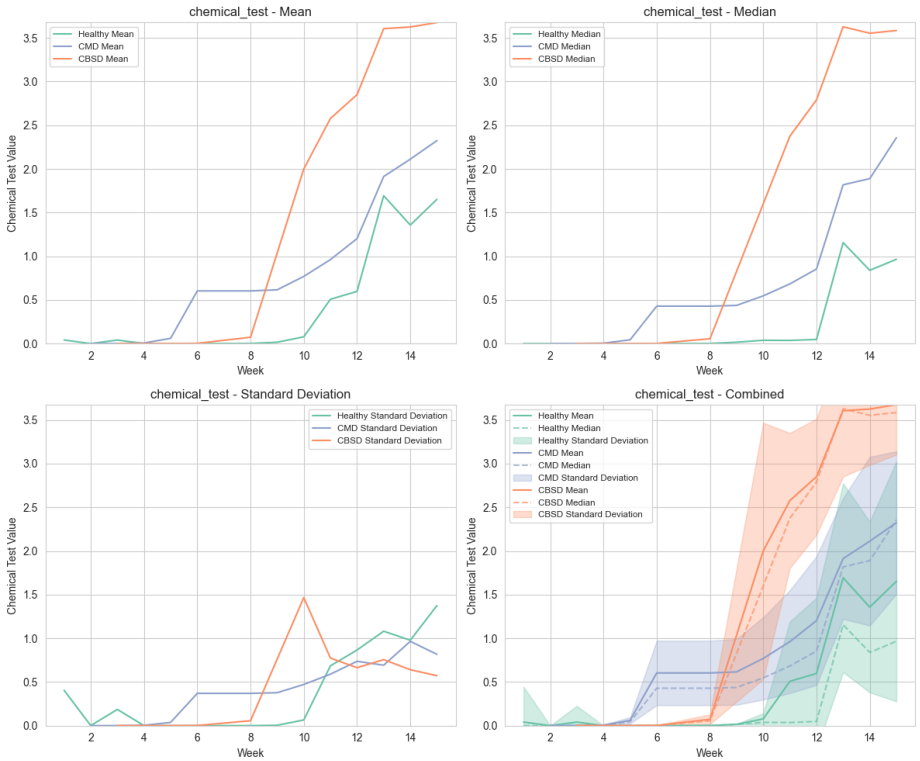


Fig. 14. Mean, median and standard deviation of chemical readings by week per disease class for screenhouse data

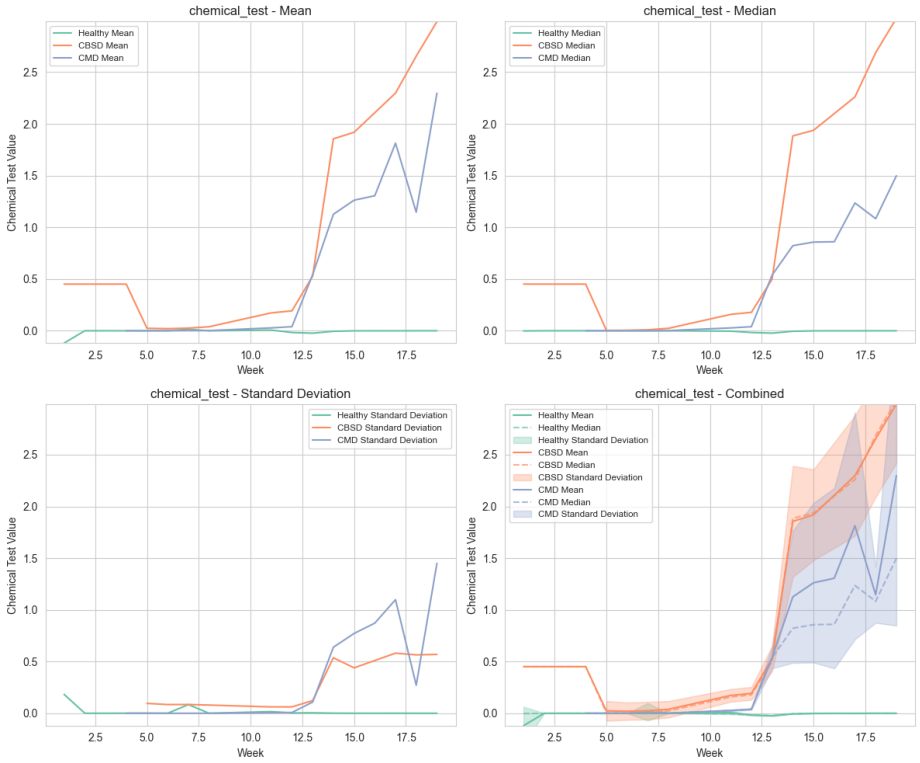


Fig. 15. Visualisation of spectral data using PCA in 3 PCs for field and screenhouse data

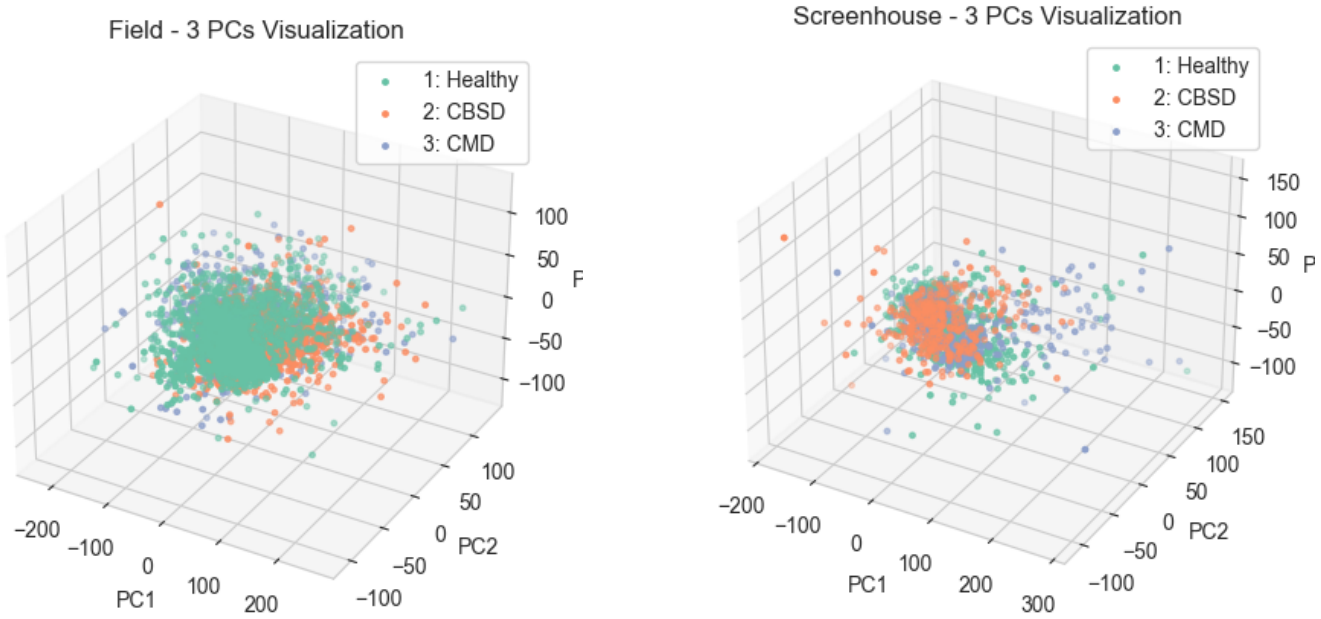
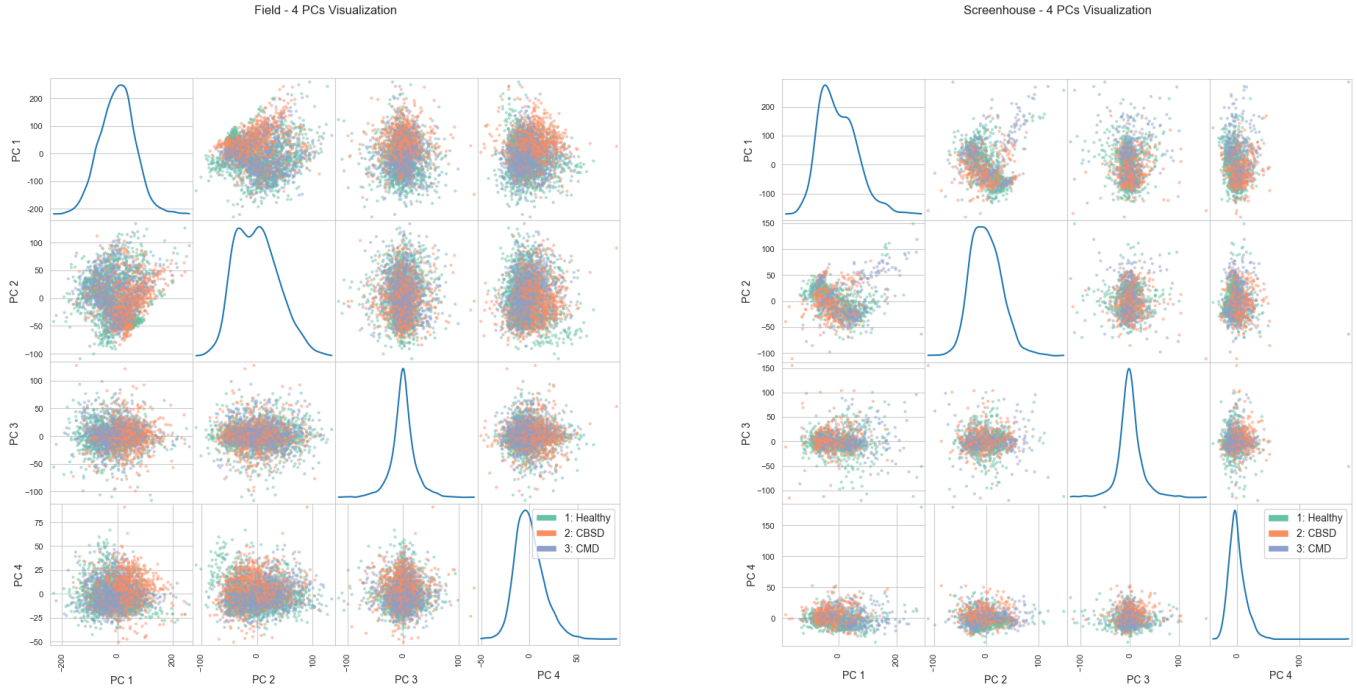


Fig. 16. Visualisation of spectral data using PCA in 4 PCs for field and screenhouse data



B SPECTRAL CLASSIFICATION

Fig. 17. Cumulative variance explained by the number of principal components for field and screenhouse data

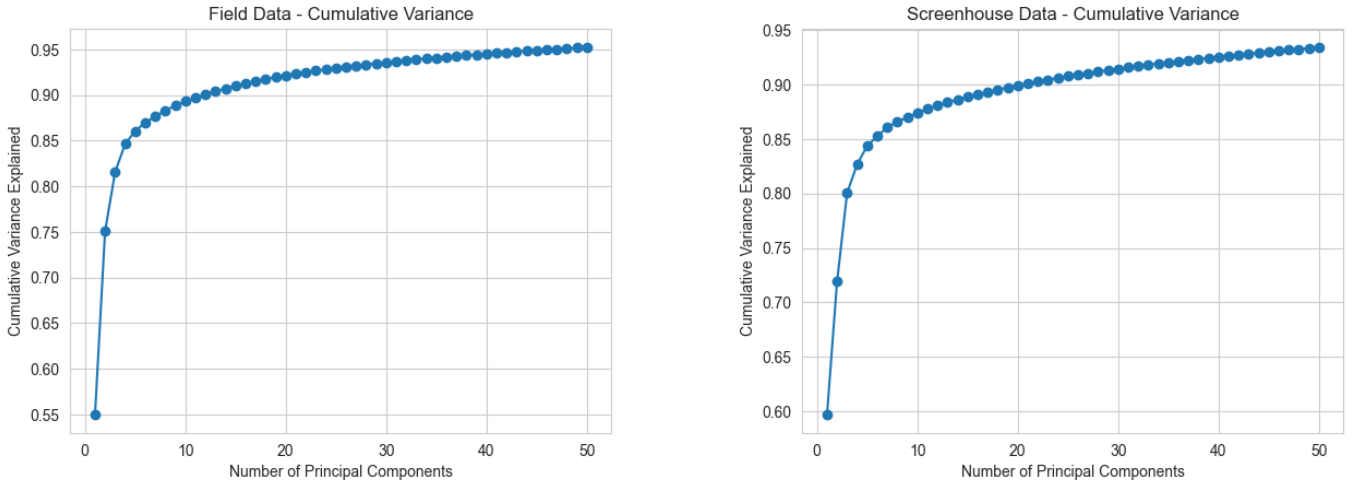


Fig. 18. Examples of spectral data for field and screenhouse before reducing noise

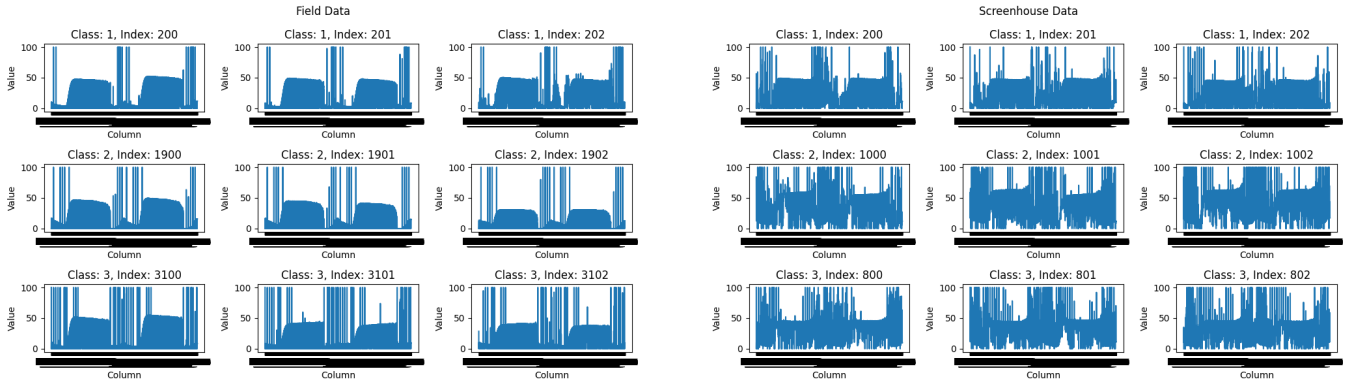
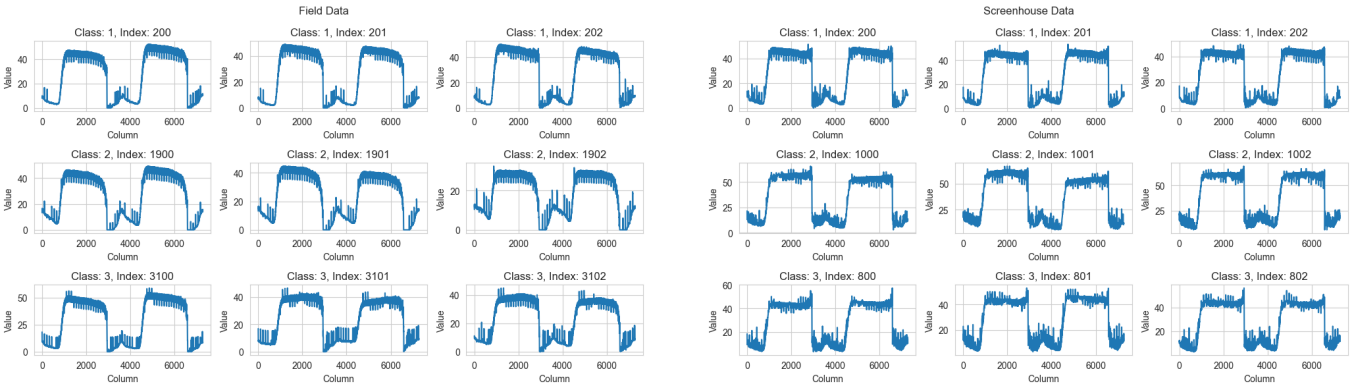
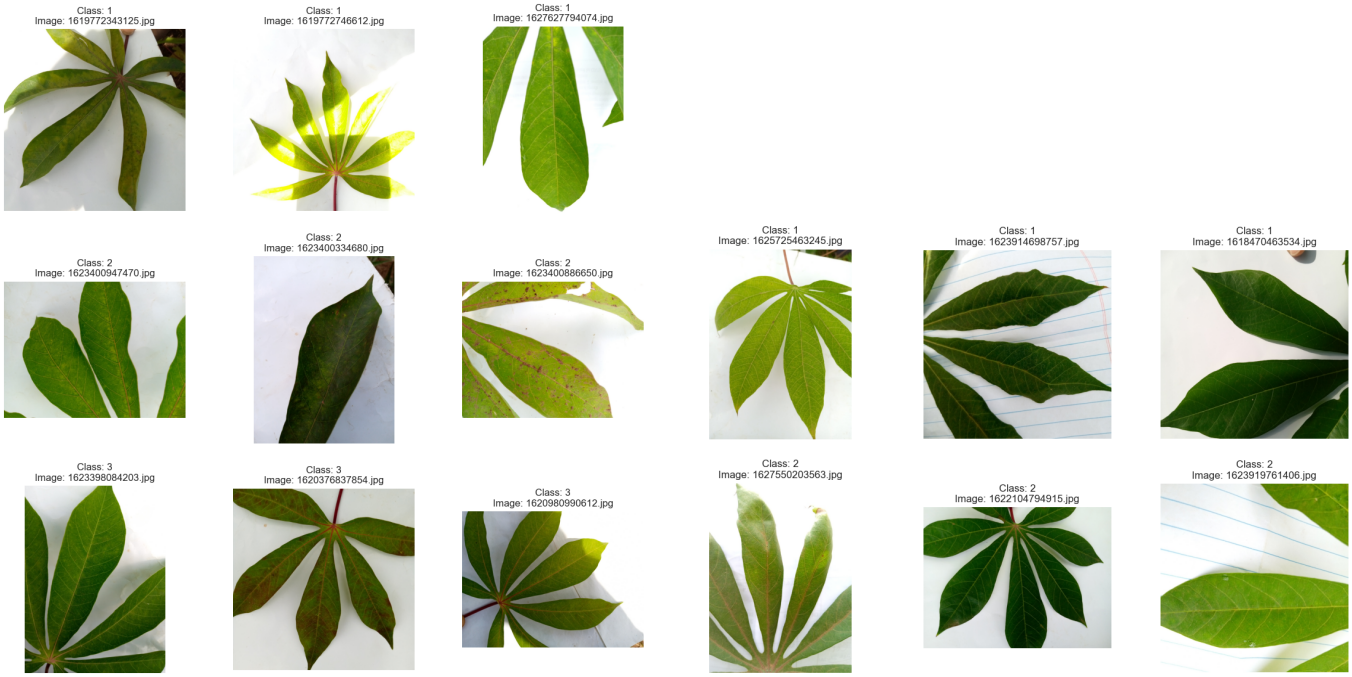


Fig. 19. Examples of spectral data for field and screenhouse after reducing noise



C IMAGE CLASSIFICATION

Fig. 20. Example images from field and screenhouse data



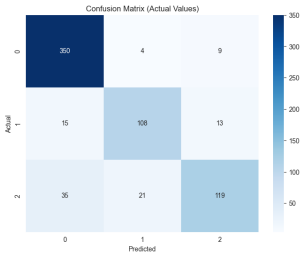
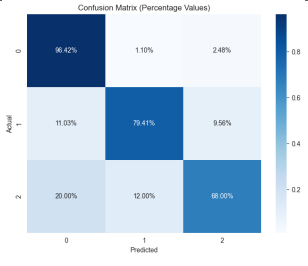
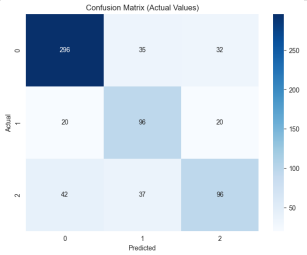
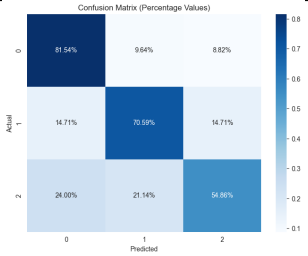
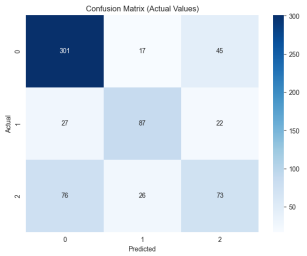
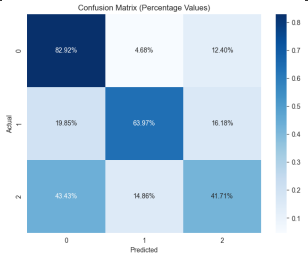
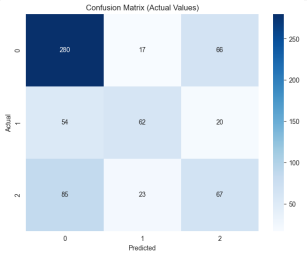
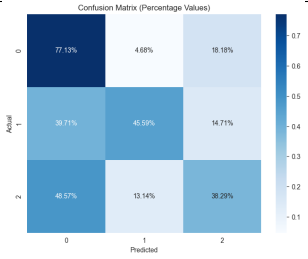
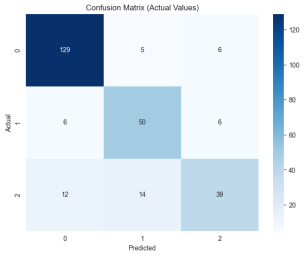
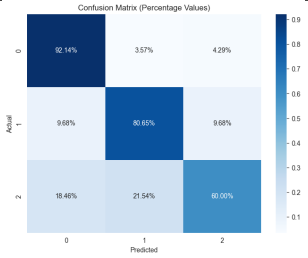
D COMPLETE TRAINING RESULTS

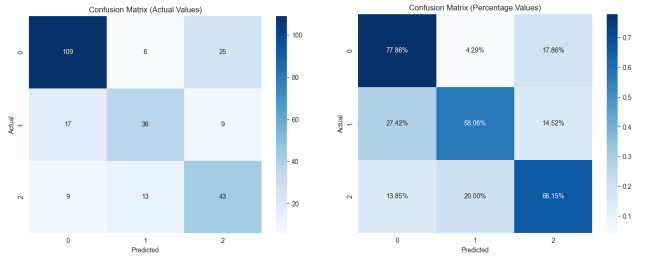
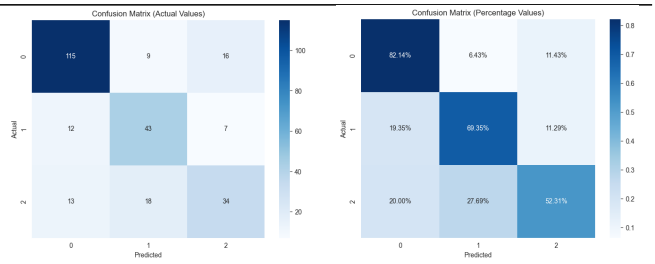
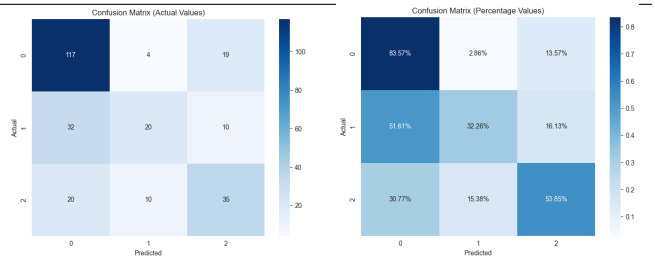
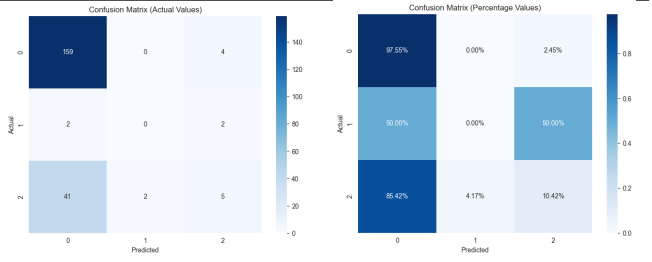
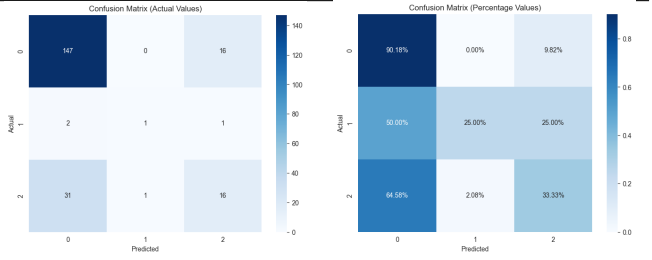
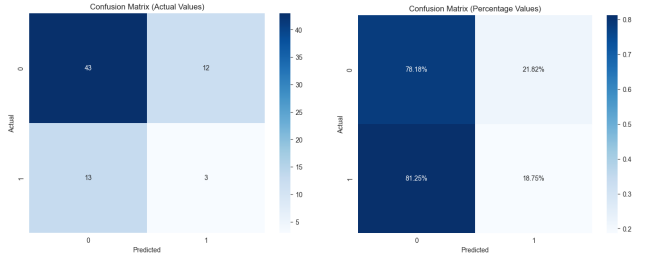
Table 7. Extended results table of classification models

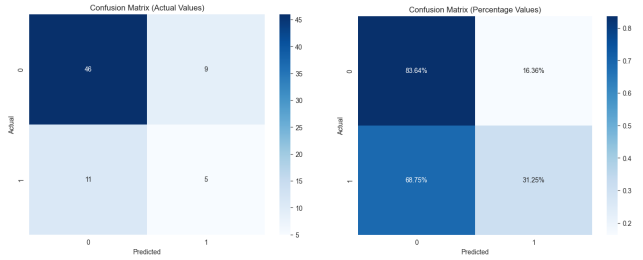
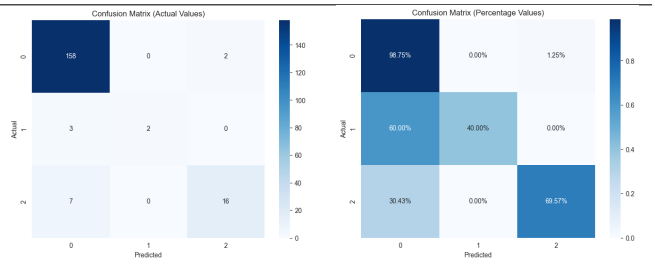
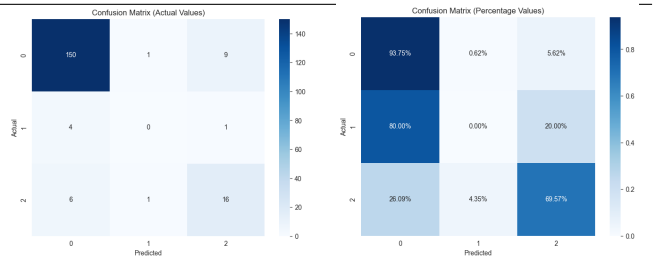
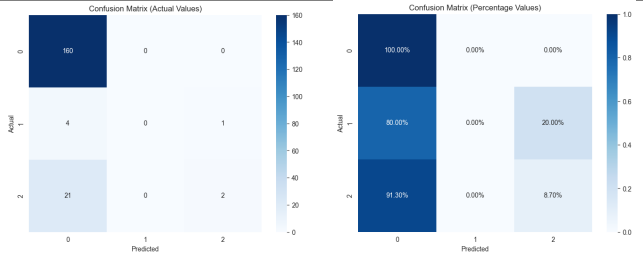
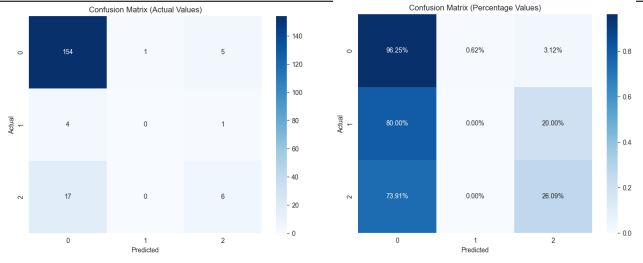
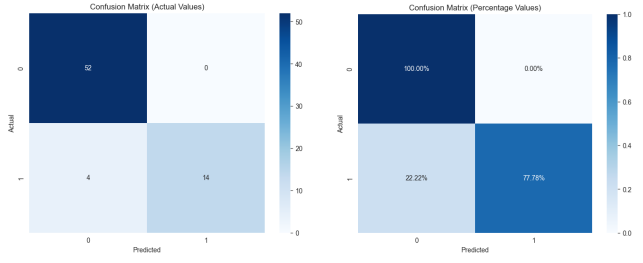
Training Data	Model	Class	Performance Metrics							
			TP	TN	FP	FN	Accuracy	Precision	Recall	F-score
Field (Non-PCA) - Spectral	XGBoost	0	350	261	50	13	0.856	0.854	0.856	0.851
		1	108	513	25	28				
		2	119	477	22	56				
	KNN	0	296	249	62	67	0.724	0.729	0.724	0.723
		1	96	466	72	40				
		2	96	447	52	79				
	Average	-	-	-	-	-	0.790	0.791	0.790	0.787
Field (PCA) - Spectral	XGBoost	0	307	206	105	56	0.689	0.674	0.689	0.675
		1	91	493	45	45				
		2	67	440	59	108				
	KNN	0	280	172	139	83	0.606	0.596	0.606	0.596
		1	62	498	40	74				
		2	67	413	86	108				
	Average	-	-	-	-	-	0.648	0.635	0.648	0.636
Screenhouse (Non-PCA) - Spectral	XGBoost	0	129	109	18	11	0.816	0.814	0.816	0.812
		1	50	186	19	12				
		2	39	190	12	26				
	KNN	0	109	101	26	31	0.704	0.711	0.704	0.705
		1	36	186	19	26				
		2	43	168	34	22				
	Average	-	-	-	-	-	0.760	0.763	0.760	0.759

Screenhouse (PCA) - Spectral	XGBoost	0	116	100	27	24	0.730	0.727	0.730	0.717
		1	45	181	24	17				
		2	34	181	21	31				
	KNN	0	117	75	52	23	0.644	0.632	0.644	0.625
		1	20	191	14	42				
		2	35	173	29	30				
	Average	-	-	-	-	-	0.687	0.680	0.687	0.671
Field - Image	XGBoost	0	159	9	43	4	0.762	0.698	0.762	0.698
		1	0	209	2	4				
		2	5	161	6	43				
	KNN	0	147	19	33	16	0.762	0.736	0.762	0.744
		1	1	210	1	3				
		2	16	150	17	32				
	Average	-	-	-	-	-	0.762	0.717	0.762	0.721
Screenhouse - Image	XGBoost	0	43	3	13	12	0.647	0.639	0.647	0.643
		1	3	43	12	13				
	KNN	0	46	5	11	9	0.718	0.705	0.718	0.711
		1	5	46	9	11				
	Average	-	-	-	-	-	0.683	0.672	0.683	0.677
Field (Filtered) - Spectral	XGBoost	0	158	18	10	2	0.936	0.935	0.936	0.930
		1	2	183	0	3				
		1	16	163	2	7				
	KNN	0	150	18	10	10	0.882	0.873	0.882	0.877
		1	0	181	2	5				
		1	16	155	10	7				
	Average	-	-	-	-	-	0.909	0.904	0.909	0.904
Field (Filtered) - Image	XGBoost	0	160	3	25	0	0.861	0.817	0.861	0.808
		1	0	183	0	5				
		1	2	164	1	21				
	KNN	0	154	7	21	6	0.851	0.810	0.851	0.824
		1	0	182	1	5				
		1	6	159	6	17				
	Average	-	-	-	-	-	0.856	0.814	0.856	0.816
Screenhouse (Filtered) - Spectral	XGBoost	0	52	14	4	0	0.942	0.946	0.942	0.940
		1	14	52	0	4				
	KNN	0	49	15	3	3	0.914	0.914	0.914	0.914
		1	15	49	3	3				
	Average	-	-	-	-	-	0.928	0.930	0.928	0.927
Screenhouse (Filtered) - Image	XGBoost	0	47	3	15	5	0.714	0.659	0.714	0.671
		1	3	47	5	15				
	KNN	0	46	8	10	6	0.771	0.757	0.771	0.761
		1	8	46	6	10				
	Average	-	-	-	-	-	0.743	0.708	0.743	0.716

Table 8. Confusion matrices

Training Data	Model	Confusion Matrix	
		Actual	Percentage
Field (Non-PCA) - Spectral	XGBoost		
	KNN		
Field (PCA) - Spectral	XGBoost		
	KNN		
Screenhouse (Non-PCA) - Spectral	XGBoost		

	KNN	
Screenhouse (PCA) - Spectral	XGBoost	
	KNN	
Field - Image	XGBoost	
	KNN	
Screenhouse - Image	XGBoost	

	KNN	
Field (Filtered) - Spectral	XGBoost	
	KNN	
Field (Filtered) - Image	XGBoost	
	KNN	
Screenhouse (Filtered) - Spectral	XGBoost	

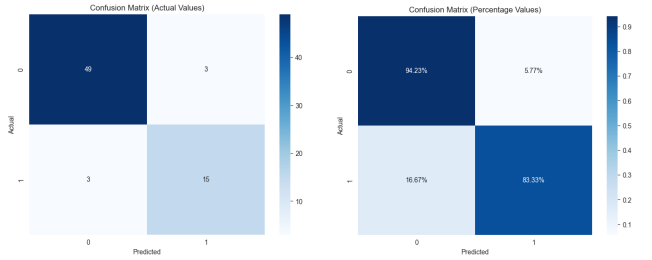
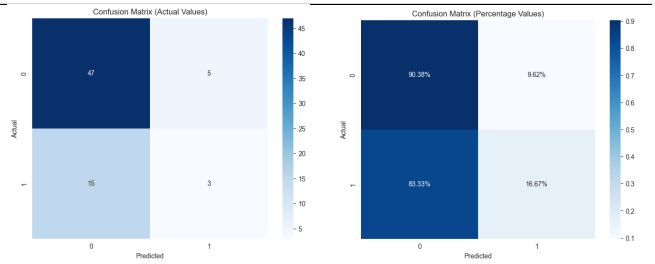
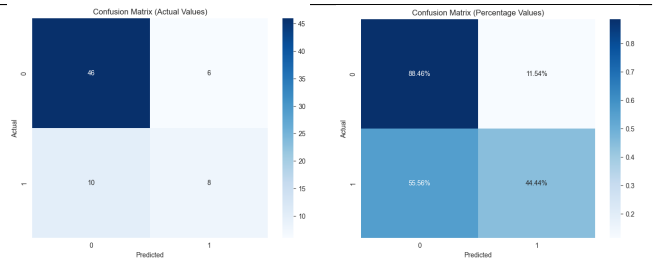
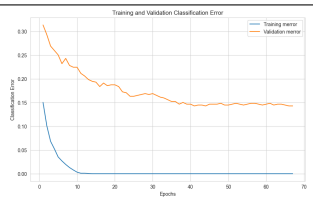
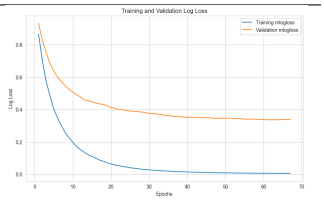
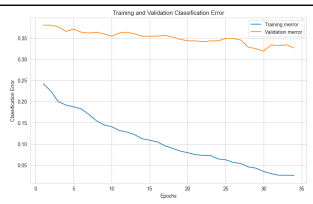
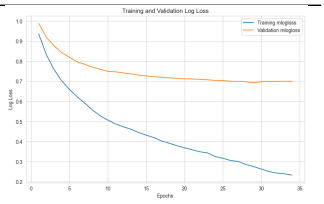
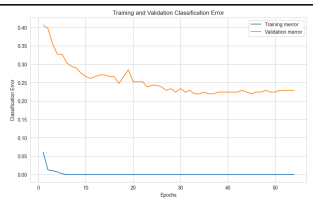
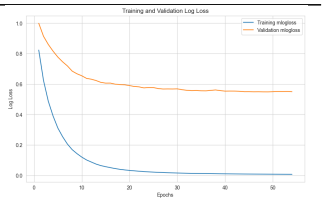
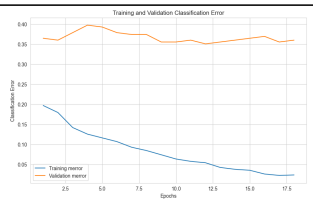
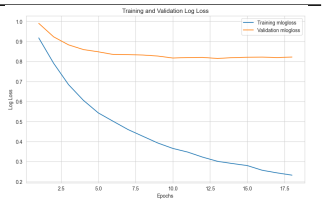
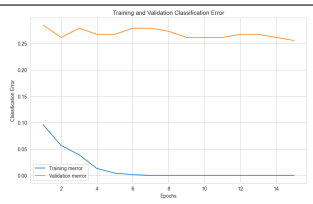
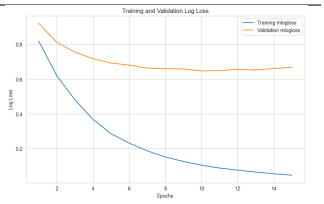
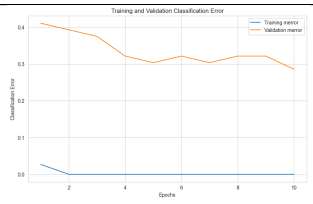
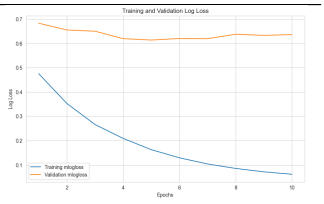
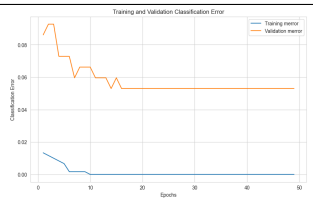
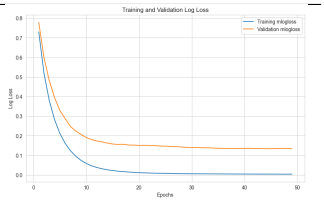
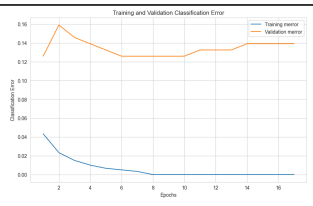
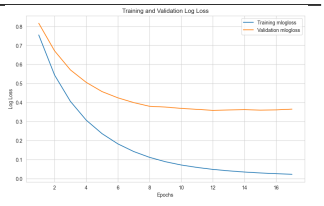
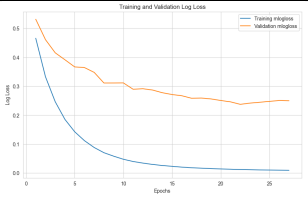
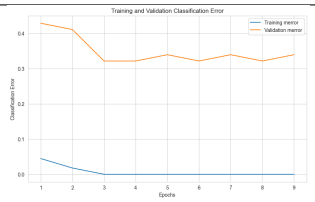
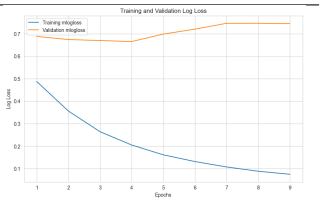
	KNN	
Screenhouse (Filtered) - Image	XGBoost	
	KNN	

Table 9. Log Loss, and Classification Error

Training Data	Model	merror	mlogloss
Field (Non-PCA) - Spectral	XGBoost		
	KNN	-	-
Field (PCA) - Spectral	XGBoost		
	KNN	-	-
Screenhouse (Non-PCA) - Spectral	XGBoost		

	KNN	-	-
Screenhouse (PCA) - Spectral	XGBoost		
	KNN	-	-
Field - Image	XGBoost		
	KNN	-	-
Screenhouse - Image	XGBoost		
	KNN	-	-
Field (Filtered) - Spectral	XGBoost		
	KNN	-	-
Field (Filtered) - Image	XGBoost		
	KNN	-	-
Screenhouse (Filtered) - Spectral	XGBoost		
	KNN	-	-

Screenhouse (Filtered) - Image	XGBoost		
	KNN	-	-