Early Detection of Sugarcane Smut Disease in Hyperspectral Images

Dong Bao*, Jun Zhou*, Shamsul Arafin Bhuiyan[†], Ali Zia*, Rebecca Ford[‡], Yongsheng Gao*

*Institute for Integrated and Intelligent Systems, Griffith University, Queensland, Australia

[†]Sugar Research Australia, Woodford, Queensland, Australia

[‡]Centre for Planetary Health and Food Security, Griffith University, Queensland, Australia

Abstract—Sugarcane smut, caused by the fungus *Sporisorium scitamineum*, is a serious sugarcane disease in Queensland, which can cause 30-100% production loss. Early detection of smut disease is a key step towards disease management. However, early-stage smut symptoms are not visible to the human eye. To address this challenge, we leverage the capability of hyperspectral imaging in data acquisition beyond the human visual spectrum and propose a deep Convolutional Neural Network (CNN) to classify sugarcane images as infected with *S. scitamineum* or healthy. A key component of the CNN is the Dual Self-Attention Block (DSAB) module that is proposed to identify important image features both spectrally and spatially. Experiments on a collected hyperspectral image dataset show the effectiveness of our proposed method in detecting smut disease before visible symptoms appear.

Keywords—Sugarcane, Smut Disease Detection, Hyperspectral Imaging, Deep Learning, Self-attention

I. INTRODUCTION

Sugarcane is Australia's primary source of raw sugar. The sugarcane industry in Queensland accounts for approximately AUD 4 billion in value, which supports more than 22,000 jobs and 10,000 businesses¹. One of the challenges sugarcane farmers face is sugarcane disease prevention and management. Smut caused by a fungus *Sporisorium scitamineum* is one of the most important sugarcane diseases, causing significant production losses under favourable conditions if intermediate susceptible and susceptible varieties are planted, which accounts for $\geq 30\%$ of the current breeding lines [1]. Therefore, early detection of sugarcane smut in screening trials will allow breeders to discard susceptible lines early in the selection program.

Computer vision and machine learning have been used for the automatic detection of plant diseases. Early research captured grayscale or colour images and used traditional machine learning methods such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) [2]–[4] as the detectors. With the success of deep learning, Convolutional Neural Networks (CNNs) have recently been adopted for plant disease detection [5], [6]. They can be integrated with traditional classifiers [7] to achieve improved results.

Since visual smut symptoms may take months to develop, by which time the disease epidemic may be well established, disease detection at the early stages of breeding programs offers a faster decision to discard susceptible lines. However,

Fig. 1: An example of a hyperspectral sugarcane leaf image. On the right is the spectral curve of a sugarcane pixel.

this is a challenging task because the only visible symptom is the appearance of a long black sorus. This motivates us to explore the information beyond the visible spectrum to gain a stronger capability of early disease detection. For this, we leverage hyperspectral imaging technology to capture data at a fine spectral resolution in both visible and near-infrared spectra. Unlike RGB images which contain three colour channels, hyperspectral images typically contain tens or hundreds of bands (channels), as shown in Fig. 1, providing rich spectral information for early detection of smut.

With the development of imaging technologies, hyperspectral images have been increasingly used in plant disease detection [4], [8], [9]. Moghadam *et al.* [4] extracted features from vegetation indices, full-spectrum, and probabilistic topic models on hyperspectral images to detect tomato spotted wilt virus in capsicum plants. Xie *et al.* [3] detected the early-stage symptoms of grey mould disease on tomato leaves one day after the inoculation. Rumpf *et al.* [10] built an automatic approach for the early detection of three sugar beet diseases based on the classification of spectral reflectance data using support vector machines. However, to our knowledge, there has not been reported work on the early detection of sugarcane smut using hyperspectral imaging.

In this paper, we assume that *S. scitamineum* infection can cause the change of spectral response on leaves of sugarcane and this type of change is not detectable by the human eye. Therefore, the change can be considered as invisible symptoms. We also assume that invisible symptoms can be detected beyond the visible spectrum in the early stage before the visible smut symptoms appear. The detection of invisible symptoms is formulated as a binary classification problem, for which CNN models are trained on a collected hyperspectral sugarcane dataset. We propose a Dual Self-Attention Block (DSAB) module which can be inserted into an adopted ResNet

¹https://www.bdbcanegrowers.com.au/wp-content/uploads/2019/11/310175_economic-contribution-of-the-sugarcane-industry-to-queensland.pdf

Fig. 2: A DSAB-integrated residual block. DSAB contains two sub-modules: the channel self-attention module and the spatial self-attention module.

backbone network architecture [11]. This module studies the importance of features both spectrally and spatially. The structure of DSAB is shown in Fig. 2. In this research, we also intend to answer the question that if invisible symptoms exist, whether they are distributed evenly on the leaf surface or locally in certain regions. To achieve this goal, we apply Class Activation Mapping (CAM) [12] to visualize Class Activation Maps (CAMs).

Our main contributions can be summarized as follows:

- This is the first work on early detection of sugarcane smut disease using hyperspectral imaging and deep neural networks.
- We propose a DSAB module and embed it in the deep neural networks to improve the detection of invisible smut symptoms.
- We build a novel hyperspectral sugarcane dataset for smut disease detection and conduct experiments to validate the assumption and effectiveness of the proposed method.

II. HYPERSPECTRAL SUGARCANE SMUT DATASET

The dataset was built from 50 inoculated sugarcane plants and 50 control (healthy) sugarcane plants. To prepare the infected plants, one-budded setts of each plant were inoculated by dipping in 1×10^6 teliospores/L water suspension and planted individually in a 1L pot with sterile soil. The control one-budded setts were dipped into sterile deionized water. Plants were maintained on a bench outside. Inoculated and uninoculated plants were maintained on separate benches and were arranged using a randomized complete block design.

Visible smut symptoms were confirmed on several infected plants at 15 weeks after inoculation and regularly visually inspected for disease symptomology. The image data capture occurred at 12 and 13 weeks after inoculation. We assumed the invisible symptoms already appear on most inoculated sugarcane plants at this stage although they are not detectable by the human eye. We do not know whether the symptoms are evenly presented throughout the whole sugarcane plant or not.

A. Image Capture

We captured 200 hyperspectral images (100 images from week 12, and 100 images from week 13) using an imec Snapscan VNIR hyperspectral camera, i.e. each image was

from a single plant. This camera covers the spectral range from 470 nm to 900 nm with 161 bands. The size of captured images is $1088 \times 2048 \times 161$, with the first two dimensions indexing the spatial locations of pixels, and the third dimension being the spectral bands.

Before image capture, a dark reference image was captured by covering the lens of the camera with a cap. The dark reference image provides information on system noises. The plant image acquisition was done directly under the sunlight with a white spectral calibration board, also called spectralon, placed next to the plant. The white calibration board reflects $\geq 99\%$ of the light from 400 nm to 1500 nm and is used to ground-truth the lighting conditions.

B. Hyperspectral Image Calibration

Hyperspectral image calibration aims to remove the noise and reduce the influences caused by different lighting conditions. It includes dark calibration and white calibration. The dark calibration step subtracts the dark reference image from the original image so that the system noise of the camera can be removed.

The white calibration requires the extraction of the white calibration board from the image. This was done by a k-means clustering method to retrieve the brightest cluster in the image. After that, the top 1% of the brightest pixels (L_2 -norm was used to rank the pixels) were removed to avoid saturation. The calibration process then calculated the mean spectral curve of the remaining top 1000 brightest pixels, which were subsequently used as the reference curve of the lighting condition for the captured sugarcane image. All pixels in the image were then divided by the mean spectral curve band-by-band.

C. Semantic Segmentation of Sugarcane

This step aims to reduce the influence of background so model learning and prediction are only performed on sugarcane plants. To reach this goal, the sugarcane plant was segmented from its background in the hyperspectral image. Here we treated the task as a semantic segmentation process and adopted a supervised learning approach. We first manually annotated some images so that the sugarcane plant was marked as the foreground and the rest of the image was treated as the background. The labelled images were then used to train an SVM classifier [13]. The training was done at the pixel level with the spectral curve and foreground-background labels as

the input. Afterwards, the trained SVM was used to generate the semantic labels of the rest of the images to automatically segment sugarcane from the background. Small holes on the sugarcane were filled using morphological operations. Fig. 3 shows an example input image, the generated semantic labels, and the segmented sugarcane image, indicating that the sugarcane segmentation is complete and highly accurate.

Fig. 3: Separating sugarcane from the background.

D. Data Generation

In total 200 images were captured, containing 100 infected plants and 100 healthy plants. However, this is insufficient to train a good deep learning model considering the high number of bands in each image. Furthermore, the spatial size of the original image is too big to be used for training. To address this problem, we used a sliding window to travel across each image and extracted $32 \times 32 \times 161$ image cubes without overlapping. The image cubes were saved if the sugarcane foreground pixels occupied $\geq 45\%$ of all the pixels. Each image cube inherited the label from the parent image, i.e. infected or healthy. We ended up with 30,261 image cubes. These data were then split into training, validation, and testing sets with a ratio of 6:2:2. Details of the sugarcane dataset are summarized in TABLE I.

TABLE I: The sugarcane smut dataset.

	Training Set	Validation Set	Testing Set
Total	17,664	6,144	6,453
Healthy	9,225	3,596	3,856
Infected	8,559	2,591	2,597

III. DUAL SELF-ATTENTION NETWORK

With the sugarcane hyperspectral image dataset collected, we treat the early detection of smut disease as a binary classification problem. To this end, we leverage the recent progress on deep learning and propose a dual self-attention network for this classification task. This network adopts a ResNet [11] as the backbone. The key component of the network is the DSAB module placed at the end of the residual block right before the addition of the shortcut and refined features. This module can help ResNet to better identify informative regions from the image. The proposed DSAB contains two self-attention modules: a channel self-attention module and a spatial self-attention module, so it is capable of capturing the joint spectral-spatial importance within the images. Fig. 2 shows how the DSAB module is integrated into the residual block.

Self-attention achieved success in natural language processing [14]. Due to its capability to extract global information and long-range dependencies of the features, it was soon introduced to computer vision for scene segmentation and image generation etc. [15], [16]. The advantage of self-attention is that it allows different spatial positions or channels of a feature map to interact with each other, so the feature discriminability can be augmented. This is achieved by learning the correlations and dependencies between features extracted at any two positions or channels. This mechanism addresses the drawbacks of traditional CNN models which focus on building image representation from local features in each layer. When applied to smut disease detection, the invisible disease symptoms located at distant positions can be linked and detected effectively.

A. General Architecture

The ResNet architecture consists of multiple CNN blocks with the residual mechanism, i.e. residual blocks. The residual block used in our deep neural network is shown in Fig. 2, in which we insert the proposed DSAB module to increase the discriminative capability of the block. A feature map $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$ from the last convolutional layer of the original residual block is fed into the DSAB module. The channel self-attention module infers the channel self-attention map first which is then fed into the spatial self-attention module to generate a spatial self-attention map. After that, the intermediate feature map $\mathbf{F}$ and the spatial self-attention map are summed together to obtain the final refined feature map $\mathbf{F}'$. Eventually, the output of the residual block is produced by an addition operation that merges the final refined feature map $\mathbf{F}'$ and the shortcut. The overall process of DSAB can be summarized as:

$$\mathbf{F}' = \mathbf{M}_s(\mathbf{M}_c(\mathbf{F})) + \mathbf{F}, \quad (1)$$

where $\mathbf{M}_c$ denotes the channel self-attention module and $\mathbf{M}_s$ denotes the spatial self-attention module.

B. Channel Self-attention Module

The channel self-attention module computes the correlations between channels. The relationships within each possible channel pair are learned and augmented. Our channel attention module learns the global information and long-range dependencies without losing the channel information.

This step is motivated by the Convolutional Block Attention Module (CBAM) [17] which was proposed to boost the performance of feed-forward convolutional neural networks. The channel attention module in CBAM performs the global average pooling and the global max pooling over the feature map to obtain two 1D representations of the channels before they are fed into a shared dense layer that learns the channel correlations. The application of global pooling in CBAM may cause information loss. To address this issue, in our method, the self-attention mechanism replaces the global pooling operations, making full use of the information in the feature map which ensures that even minor correlations and features can be extracted.

As shown in the left panel of Fig. 4, the module receives an intermediate feature map $\mathbf{F}$ as the input. The feature map

Fig. 4: Diagram of two sub-modules of DSAB. Their architectures are similar, the difference lies in the operations on queries, keys, and values. The channel self-attention module focuses on extracting channel-wise correlations and dependencies, while the spatial self-attention module focuses on extracting pixel-wise correlations and dependencies.

goes through Batch Normalization (BN) and ReLu activation layers before being fed into the module. Afterwards, three 1×1 convolutions are placed in parallel to serve as three transformation matrices: $\mathbf{W}_Q$, $\mathbf{W}_k$, and $\mathbf{W}_v$. Query $\mathbf{Q} \in \mathbb{R}^{H \times W \times C}$, key $\mathbf{K} \in \mathbb{R}^{H \times W \times C}$, and value $\mathbf{V} \in \mathbb{R}^{H \times W \times C}$ are then generated by these three transformation matrices respectively.

The channel self-attention module aims to extract the correlations between channels maps. To do so, we reshape and transpose $\mathbf{Q}$ to $C \times N$, where $N = H \times W$, and reshape $\mathbf{K}$ and $\mathbf{V}$ to $N \times C$. A row vector $\mathbf{q}_i \in \mathbb{R}^{1 \times N}$ in $\mathbf{Q}$ is the corresponding query for i^{th} channel map. Similarly, column vectors $\mathbf{k}_j, \mathbf{v}_j \in \mathbb{R}^{N \times 1}$ are the corresponding key and value for j^{th} channel map. $\mathbf{Q}$ is then multiplied by $\mathbf{K}$ to obtain $\mathbf{I}$:

$$\mathbf{I} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1C} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{C1} & \alpha_{C2} & \cdots & \alpha_{CC} \end{bmatrix} \quad (2)$$

where

$$\alpha_{ij} = \frac{q_i \cdot k_j}{\sqrt{N}} \quad (3)$$

is the scaled dot product. Using a scaled dot product can prevent Softmax gradient from being too small. We then apply Softmax on $\mathbf{I}$ to obtain the activation map $\mathbf{A}_c$, where the Softmax formula is defined by

$$\alpha'_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{j=1}^C \exp(\alpha_{ij})} \quad (4)$$

and the derived activation map is

$$\mathbf{A}_c = \begin{bmatrix} \alpha'_{11} & \alpha'_{12} & \cdots & \alpha'_{1C} \\ \alpha'_{21} & \alpha'_{22} & \cdots & \alpha'_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha'_{C1} & \alpha'_{C2} & \cdots & \alpha'_{CC} \end{bmatrix}. \quad (5)$$

Eventually, $\mathbf{V}$ is multiplied by the transpose of $\mathbf{A}_c$ to obtain

the channel self-attention map $\mathbf{F}_c$, where

$$\mathbf{F}_c = \begin{bmatrix} \sum_{j=1}^C \alpha'_{1j} v_{1j} & \sum_{j=1}^C \alpha'_{2j} v_{1j} & \cdots & \sum_{j=1}^C \alpha'_{Cj} v_{1j} \\ \sum_{j=1}^C \alpha'_{1j} v_{2j} & \sum_{j=1}^C \alpha'_{2j} v_{2j} & \cdots & \sum_{j=1}^C \alpha'_{Cj} v_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^C \alpha'_{1j} v_{Nj} & \sum_{j=1}^C \alpha'_{2j} v_{Nj} & \cdots & \sum_{j=1}^C \alpha'_{Cj} v_{Nj} \end{bmatrix}. \quad (6)$$

C. Spatial Self-attention Module

This module focuses on modelling the global information and long-range dependencies between spatial locations. Our module employs all the channel information to extract the global and local features.

As illustrated in the right panel of Fig. 5, the module receives the channel self-attention map $\mathbf{F}_c \in \mathbb{R}^{H \times W \times C}$ as the input. The steps of generating Query $\mathbf{Q} \in \mathbb{R}^{H \times W \times C}$, key $\mathbf{K} \in \mathbb{R}^{H \times W \times C}$, and value $\mathbf{V} \in \mathbb{R}^{H \times W \times C}$ are the same as those used in the channel self-attention module. Note that in this section we use the same notations $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ as in the channel self-attention module, but they refer to different matrices. We first reshape $\mathbf{Q}$ to $N \times C$, and then reshape and transpose $\mathbf{K}$ and $\mathbf{V}$ to $C \times N$. A row vector $\mathbf{q}_i \in \mathbb{R}^{1 \times C}$ in $\mathbf{Q}$ is the corresponding query for i^{th} pixel. Similarly, the column vectors $\mathbf{k}_j, \mathbf{v}_j \in \mathbb{R}^{C \times 1}$ are the corresponding key and value for j^{th} pixel. We perform a matrix multiplication between $\mathbf{Q}$ and $\mathbf{K}$ to obtain $\mathbf{J}$ and apply Softmax to get the activation map $\mathbf{A}_s$. Finally, $\mathbf{V}$ is multiplied by the transpose of $\mathbf{A}_s$ to obtain the spatial self-attention map $\mathbf{F}_s$. Detailed equations for these steps are similar to those of channel self-attention, so they are not included here.

IV. EXPERIMENTS

We compared the proposed method with the baseline ResNet models [11] and the CBAM method [17] on the collected sugarcane smut dataset. Both CBAM and the proposed DSAB are attention modules that can be inserted into ResNet. As shown in TABLE II, six models were trained based on two ResNet architectures with a different number of layers.

Accuracy, F1-score, specificity, and sensitivity were used to measure the performance of different models. Accuracy is

Fig. 5: Comparison of Receiver Operating Characteristic (ROC) curves of six models.

calculated as the correctly classified samples among all testing samples. F1-score is the harmonic mean of the precision and recall, where precision is the number of true positive samples divided by the number of all positive samples, and recall is the number of true positive samples divided by the number of all samples that should have been identified as positive. Specificity is used to measure the ability to correctly classify healthy samples, while sensitivity is used to measure the ability to detect the infected samples.

A. Implementation Details

All the reported methods were implemented using TensorFlow 2². ResNet18 and ResNet34 were adopted with pre-activation residual blocks [18] because they make it easier for signals to propagate in the networks in both forward and backward directions and hence improve the model generalization. We removed the max pooling layer from the ResNet and only halved the feature map size at the first 7×7 convolutional layer and stage three. This is because the spatial size of the input image is small. ResNet usually halves the input five times, which makes most intermediate feature maps to be very small and generates trivial features. Our modification of the network structure maintains the large size of feature maps so more spatial details can be maintained. Besides, large feature maps also help to generate better CAMs [12]

TABLE II: Classification results on the sugarcane dataset.

Model Name	Accuracy	F1-Score	Specificity	Sensitivity
ResNet18	89.42%	0.8648	0.9353	0.8340
ResNet18+CBAM	88.59%	0.8530	0.9340	0.8156
ResNet18+DSAB	90.56%	0.8818	0.9319	0.8672
ResNet34	90.81%	0.8850	0.9335	0.8710
ResNet34+CBAM	88.03%	0.8526	0.8988	0.8533
ResNet34+DSAB	90.86%	0.8851	0.9364	0.8679

B. Training

The models were trained on two GPUs (Nvidia Titan V and Titan Xp). The mirrored strategy was used as the distributed training strategy. We trained the models for 200 epochs with the batch size set as 256. An exponential decay on learning rate was adopted with the decay rate set as 0.7 and the initial learning rate set as 0.0003. The learning rate decayed every 20 epochs. Adam and binary cross-entropy were used as the optimizer and loss function respectively. We also used the ModelCheckpoint function to monitor the validation accuracy during training and saved the model with the highest validation accuracy which was used to obtain the testing results.

Fig. 6: CAM visualisation results on infected and healthy sugarcane image patches.

²<https://www.tensorflow.org/>

C. Results

TABLE II summarizes the experimental results. The networks with DSAB outperform other networks in terms of accuracy. ResNet34 with DSAB achieves 90.86% accuracy which is the highest among all the models. The accuracy of ResNet18 with DSAB is 1.14% higher than that of ResNet18. The F1-scores of ResNet34 and ResNet34 with DSAB are very close. However, the F1-score of ResNet18 with DSAB is 0.017 higher than that of ResNet18. The sensitivity of ResNet18 with DSAB is 0.0332 higher than that of ResNet18, which is a remarkable lead. ResNet34 with DSAB achieves the highest specificity and ResNet34 achieves the highest sensitivity. In general, the ResNet34 based networks perform better than ResNet18 based networks due to ResNet34's capability in coping with more complex data. Also, the introduction of DSAB can improve detection performance in most cases. This shows the benefit of self-attention modules to better model the relationships between features in channels and locations.

ROC curves of six trained models are presented in Fig. 5. These curves are very close to each other. In terms of Area Under the Curve (AUC), ResNet34 with DSAB reaches 0.96692 which is the highest among six trained models. All the six trained models have their AUCs reach over 0.95, which suggests that ResNet-based networks can effectively identify the invisible symptoms on the hyperspectral images of infected sugarcane.

We also show the CAMs [12] generated from different methods for some infected and healthy sugarcane in Fig. 6. The bright areas in the maps are the important regions on sugarcanes that make major contributions to the final classification outcomes. The figure shows some interesting observations. Firstly, the bright areas, in general, occupy higher percentages on leaves of healthy plants than on leaves of infected plants. This implies that invisible smut symptoms are likely to be more localized rather than evenly distributed on the plant surface. Secondly, the DSAB-integrated models have similar but broader important areas than the baseline ResNet models. This is because a self-attention module computes the correlation and dependencies between any two spatial locations and channels. This enables the inclusion of pixels with larger distances to jointly contribute to disease detection. Thirdly, the CBAM method discretizes the contribution of local features. This may be caused by its global pooling operations that reduce the capability of capturing finely detailed information from images. This is consistent with the results presented in TABLE II.

V. CONCLUSION

In this paper, we have introduced a method to automatically detect invisible early-stage symptoms of sugarcane smut disease. This method is based on hyperspectral imaging and deep convolutional neural networks. We propose the DSAB module to extract joint spectral-spatial attention information for improving disease detection performance. The experimental results have proven the effectiveness of our method in smut detection before symptoms become visible to human vision. Analysis of visualization results also demonstrates that the distribution of invisible smut symptoms is likely to be local. In the future, we will work with biologists to further verify the

existence of invisible early-stage sugarcane smut symptoms. We will also extend our work to other sugarcane diseases, such as the sugarcane mosaic virus.

REFERENCES

- [1] S. A. Bhuiyan, R. Magarey, M. McNeil, and K. Aitken, "Sugarcane smut, caused by *sporisorium scitamineum*, a major disease of sugarcane—a contemporary review," *Phytopathology*, no. ja, 2021.
- [2] K. R. Gavhale and U. Gawande, "An overview of the research on plant leaves disease detection using image processing techniques," *IOSR Journal of Computer Engineering*, vol. 16, no. 1, pp. 10–16, 2014.
- [3] C. Xie, C. Yang, and Y. He, "Hyperspectral imaging for classification of healthy and gray mold diseased tomato leaves with different infection severities," *Computers and Electronics in Agriculture*, vol. 135, pp. 154–162, 2017.
- [4] P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka, and E. Hernandez, "Plant disease detection using hyperspectral imaging," in *2017 International Conference on Digital Image Computing: Techniques and Applications*, 2017, pp. 1–8.
- [5] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- [6] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
- [7] M. Türkoğlu and D. Hanbay, "Plant disease and pest detection using deep learning-based features," *Turkish Journal of Electrical Engineering & Computer Science*, vol. 27, pp. 1636–1651, 05 2019.
- [8] G. Polder, P. M. Blok, H. A. C. de Villiers, J. M. van der Wolf, and J. Kamp, "Potato virus Y detection in seed potatoes using deep learning on hyperspectral images," *Frontiers in Plant Science*, vol. 10, p. 209, 2019.
- [9] K. Nagasubramanian, S. Jones, A. K. Singh, A. Singh, B. Ganapathysubramanian, and S. Sarkar, "Explaining hyperspectral imaging based plant disease identification: 3d CNN and saliency maps," *CoRR*, vol. abs/1804.08831, 2018.
- [10] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance," *Computers and Electronics in Agriculture*, vol. 74, no. 1, pp. 91–99, 2010.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.
- [12] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.
- [13] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing systems*, 2017, pp. 5998–6008.
- [15] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019.
- [16] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7354–7363.
- [17] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision*, September 2018.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proceedings of the European Conference on Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, pp. 630–645.