Right for the Wrong Scientific Reasons: Revising Deep Networks by Interacting with their Explanations

Patrick Schramowski[1], Wolfgang Stammer[1], Stefano Teso[2], Anna Brugger[3], Hans-Georg Luigs[4], Anne-Katrin Mahlein[5] & Kristian Kersting[1,6]

[1]TU Darmstadt, Computer Science Department, Darmstadt, Germany
[2]KULeuven, Computer Science Department, Leuven, Belgium
[3]University of Bonn, Institute for Crop Science and Resource Conservation (INRES) – Plant Diseases and Plant Protection, Bonn, Germany
[4]LemnaTec GmbH, Aachen, Germany
[5]Institute for Sugar Beet Research, Goettingen, Germany
[6]TU Darmstadt, Centre for Cognitive Science, Darmstadt, Germany

Abstract

Deep neural networks have shown excellent performances in many real-world applications such as plant phenotyping. Unfortunately, they may show "Clever Hans"-like behaviour— making use of confounding factors within datasets—to achieve high prediction rates. Rather than discarding the trained models or the dataset, we show that interactions between the learning system and the human user can correct the model. Specifically, we revise the models decision process by adding annotated masks during the learning loop and penalize decisions made for wrong reasons. In this way the decision strategies of the machine can be improved, focusing on relevant features, without considerably dropping predictive performance.

Introduction

Imagine a plant phenotyping team attempting to characterize crop resistance to plant pathogen. The plant physiologist records a larger amount of hyperspectral imaging data. Impressed by the results of deep learning in other scientific areas, she wants to establish similar results for phenotpying. Consequently, she asks a machine learning expert to apply deep learning to analyse the data. Luckily, the resulting predictive accuracy is very high. The plant physiologist, however, remains sceptical. The results are "too good, to be true". Checking the decision process of the deep model using explainable artificial intelligence (AI), the machine learning expert is flabbergasted to find that the learned deep model actually uses clues within the data that do not actually relate to the biological problem at hand, so called confounding factors. The physiologist loses trust in AI and turns away from it, proclaiming it to be useless.

This example encapsulates an important issue of current explainable AI. Indeed, the seminal paper of Lapuschkin *et al.* [1] helps in "unmasking Clever Hans predictors and assessing what machines really learn". However, rather than proclaiming, as the plant physiologist might, that the machines have learned the right predictions for wrong reasons and can therefore not be trusted, we here showcase that interactions between the learning system and the human user can correct the model towards making the right predictions for the right reasons. This may also increase the trust into machine learning models.

Indeed, trust lies at the foundation of major theories of interpersonal relationships in psychology, as argued by Simpson [2]. In particular, Hoffman *et al.* [3] argue that interpersonal trust depends on the perceived competence, benevolence (or malevolence), understandability, and directability—the degree to which the trustor can rapidly assert control or influence when something goes wrong. They and others such as Waytz *et al.* [4] and Wang *et al.* [5] also show that trust into machines follows similar patterns, with some notable differences: it is often inappropriate to attribute benevolence/malevolence to machines, and trust into machines suffers from different biases than trust into individuals. The differences, however, do not affect the argument that interaction and understandability are central to trust in learning machines, too. The competence of a classifier can be assessed by monitoring its behavior and beliefs over time, directability can be achieved by allowing the user to actively teach the model how to act and what to believe, while understandability can be approached by explaining the models decisions.

Surprisingly, the link between interacting, explaining and building trust has been largely ignored by the machine learning literature. On one hand, existing approaches focus on passive learning only, and do not consider interaction

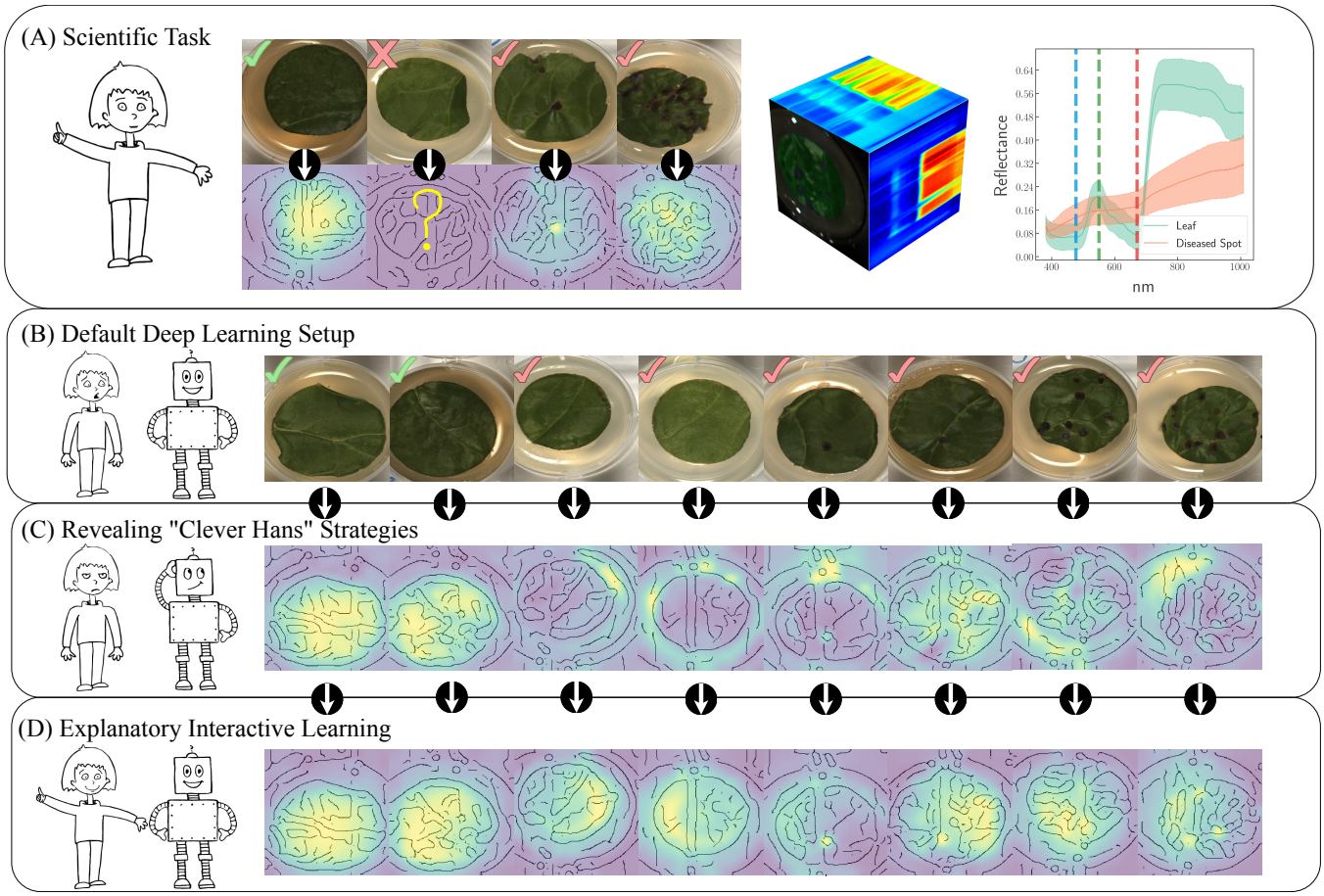

Figure 1: **Explanatory Interactive Learning (XIL)**—Human users revise learning machines results towards trustworthy decision strategies. (A-Left) Data samples and respective explanations that an expert expects of a ML model. Note even an expert can be uncertain about a valid explanation. (A-Middle) Visualization of the hyperspectral data, corresponding to the spatial dimensions and the spectral dimension. The planes on the top and left sides of the cube correspond to slices taken from the center of the cube, but placed on the edges for visualization. (A-Right) Visualization of the characteristic reflectance of healthy tissue vs. disease spots. The vertical red, green and blue lines depict the three wavelengths of the RGB dataset. (B+C) Classifications of a deep neural network and its explanations. The learned model clearly uses confounding factors to explain it's decision. The human user provides feedback on the reasons. In turn, the machine gets new information and can continue learning. (D) The human-revised deep network yields to improved classifications, in a large part matching the expected strategies. (We note here that for visualization the RGB images shown in all figures correspond to real RGB images, whereas the edge overlays result from the pseudo-RGB images of the original data set (*cf.* Methods RGB/HS classification).)

between the user and the learner [6, 7, 8]. On the other hand, interactive learning frameworks such as active [9] and coactive learning [10] do not consider the issue of trust. In active learning, for instance, the model presents unlabelled instances to a user, and in exchange obtains their label. This is completely opaque—the user is oblivious to the models beliefs and reasons for predictions and to how they change in time, and cannot see the consequences of her own instructions. In coactive learning, the user sees and corrects the systems prediction, if necessary, but the predictions are not explained to her. So, why should users trust models learned interactively?

Furthermore, although an increasing amount of research investigates methods for explaining machine learning models, even here the notion of interaction has been largely ignored. Reconsider the study by Lapuschkin *et al.* [1]. They showed that one can in fact find "Clever Hans"-like behaviour in popular computer vision models which based their decisions on confounding factors. These factors may act as good indicators within the particular dataset, but would prove to be useless in real world settings. Based on these findings, Lapuschkin *et al.* recommended a word of caution towards the interest in such models, but they did not offer a solution for correcting their behaviour. Particularly in real world applications, where monitoring for every possible confounding factor or acquiring a new dataset due to existing confounders is time and resource consuming, it is inevitable to move beyond revealing the (wrong) reasons step towards correcting the reasons underlying a models decisions.

This is exactly the main technical contribution of the present study. We introduce the novel learning setting of "explanatory interactive learning" (XIL) and illustrate its benefits in an important scientific endeavour, namely,

plant phenotpying. Starting from a learning system that does not deliver biologically plausible explanations for a relevant, real-world task in plant phenotyping, we add the scientist into the training loop, who interactively revises the original model via explanations so that it produces trustworthy decisions without major drop in performance. Specifically, the interaction takes the form illustrated in Fig. 1. In each step, the learner explains its interactive query to the domain expert, and she responds by correcting the explanations, if necessary, to provide feedback. This allows the user not only to check whether the model is right or wrong on the chosen instance, but also if the answer is right (or wrong) for the wrong reasons, e.g., when there are ambiguities in the data such as confounders [11]. By witnessing the evolution of the explanations, similar to a teacher supervising the progress of a student, the human user can see whether the model eventually "gets it". Actually, the user can even correct the explanation presented to guide the learner. This correction step is crucial for more directly affecting the learners beliefs and is integral to modulating trust [3, 12].

To demonstrate the significance of XIL, we demonstrate XIL for deep plant phenotpying, a growing and relevant field of research [13, 14, 15, 16, 17]. To this end, we recorded a scientific, real-world dataset—a plant phenotyping dataset consisting of RGB and hyperspectral images (HS) of healthy and diseased sugar beet leaves. Then, we applied convolutional neural networks to classify the plants' leaves into the categories *control* (healthy) and *inoculated* (diseased) and investigated the underlying reasons for the network's predictions. As a model disease, Cercospora Leaf Spot (CLS) was used. This is caused by Cercospora beticola, and is the most destructive leaf disease of sugar beet with worldwide economic importance.

We expected the networks to use obvious symptoms of the plant disease such as the characteristic CLS typically occurring at later stages of the disease, but also non-visible symptoms for early stages of disease progression that are hidden in the spectral domain. Examples from varying stages of the disease progression are shown in Fig. 1 (A) together with feature importance maps (saliency maps) in the bottom row that human experts consider as right reasons. Specifically, the left most example shows a healthy tissue sample. The second to the left example shows an inoculated sample at an early stage of disease progression, where no obvious disease characteristics are visible and even the human expert is unsure of the right reason for classifying such a sample. The two right most samples show inoculated samples displaying the characteristic disease spots with varying extent. Training a deep neural network to classify the two categories resulted in surprisingly high performance (B). Unmasking the reasons for the classifications using an approach similar to that of Lapuschkin *et al.* [1], different reasons of the model can be identified, showing that the model is right for the wrong reasons, focusing incorrectly on areas outside of the tissue (C). Using XIL, however, a human user can correct this behaviour (D). The corrected model produced accurate predictions for the right reasons.

We proceed as follows. We start off by formally introducing Explanatory Interactive Machine Learning (XIL) and instantiate it in the CAIPI method [18] as well as the RRR method [11]. After introducing XIL, we discuss qualitatively and quantitatively several empirical results. They demonstrate the importance of explaining decisions for building trustful machines by including the human user into the training process. Finally, we provide the details on how domain experts can revise learning machines and in turn enable the machines to correct their abilities to solve the scientific real-world task of plant disease prediction. Our contributions thus addresses a main part of building trustworthy AI methods by providing an end-to-end, interactive method to evaluate and revise black-box models. This provides an important alternative to Rudin's [19] message: "*Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead*", namely:

> *Continue to explain black box models since they can alleviate "Clever Hans"-like problems when used to revise the model interactively.*

A preliminary version of this manuscript appeared as conference paper [18]. It is extended by XIL towards latent layers of deep neural networks and presents the first application of XIL to an important scientific task, namely, plant phenotyping.

Explanatory Interactive Machine Learning (XIL)

In XIL, a learner is able to interactively query the user (or some other information source) to obtain the desired outputs of the data points. The interaction takes the following form. At each step, the learner considers a data point (labeled or unlabeled), predicts a label, and provides explanations of its prediction. The user responds by correcting the learner if necessary, providing a slightly improved – but not necessarily optimal – feedback to the learner.
Let us now instantiate this schema to *explanatory active learning*—combining active learning with local explainers (*cf.* Methods). Indeed, other interactive learning can be made explanatory too, including coactive learning [10], active imitation learning [20], and mixed-initiative interactive learning [21], but this is beyond the scope of this paper.

Explanatory Active Learning. In Explanatory Active Learning, we require black-box access to an active learner and an explainer. We assume that the active learner provides a procedure SELECTQUERY($f, \mathcal{U}$) for selecting an informative instance $x \in \mathcal{U}$ based on the current model f, and a procedure FIT($\mathcal{L}$) for fitting a new model (or update

Algorithm 1 CAIPI takes as input a set of labelled examples $\mathcal{L}$, a set of unlabelled instances $\mathcal{U}$, and iteration budget T.

$f \leftarrow \textsc{Fit}(\mathcal{L})$
repeat
  $x \leftarrow \textsc{SelectQuery}(f, \mathcal{U})$
  $\hat{y} \leftarrow f(x)$
  $\hat{z} \leftarrow \textsc{Explain}(f, x, \hat{y})$
  Present x, $\hat{y}$, and $\hat{z}$ to the user
  Obtain $\bar{y}$ and explanation correction $\mathcal{C}$
  if CAIPI: $\{(\bar{x}_i, \bar{y})\}_{i=1}^{c} \leftarrow \textsc{ToCounterExamples}(\mathcal{C})$
  else if RRR: $\{(x, \bar{y}, A)\} \leftarrow \textsc{ToBinaryCorrectionMask}(\mathcal{C})$
  if CAIPI: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x, \bar{y})\} \cup \{(\bar{x}_i, \bar{y})\}_{i=1}^{c}$
  else if RRR: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x, \bar{y}, A)\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{x\}$
  $f \leftarrow \textsc{Fit}(\mathcal{L})$
until budget T is exhausted or f is good enough
return f

the current model) on the examples in $\mathcal{L}$. The explainer is assumed to provide a procedure $\textsc{Explain}(f, x, \hat{y})$ for explaining a particular prediction $\hat{y} = f(x)$. The framework is intended to work for any reasonable learner and explainer.

When using LIME for computing an interpretable model locally around the queries in order to visualize explanations for current predictions, this results in CAIPI as summarized in Alg. 1. At each iteration $t = 1, \ldots, T$ an instance $x \in \mathcal{U}$ is chosen using the query selection strategy implemented by the SELECTQUERY procedure. Then its label $\hat{y}$ is predicted using the current model f, and EXPLAIN is used to produce an explanation $\hat{z}$ of the prediction. The triple $(x, \hat{y}, \hat{z})$ is presented to the user as a (visual) artifact. The user checks the prediction and the explanation for correctness, and provides the required feedback. Upon receiving the feedback, the system updates $\mathcal{U}$ and $\mathcal{L}$ accordingly and re-fits the model. The loop terminates when the iteration budget T is reached or the model is good enough.

During interactions between the system and the user, three cases can occur: **(1) Right for the right reasons:** The prediction and the explanation are both correct. No feedback is requested. **(2) Wrong for the wrong reasons:** The prediction is wrong. As in active learning, we ask the user to provide the correct label. The explanation is also necessarily wrong, but we currently do not require the user to act on it. **(3) Right for the wrong reasons:** The prediction is correct but the explanation is wrong. We ask the user to provide an *explanation correction* $\mathcal{C}$.

Explanatory Interactive Learning with counterexamples. The "right for the wrong reasons" case is novel in active learning, and we propose *explanation corrections* to deal with it. They can assume different meanings depending on whether the focus is on component relevance, polarity, or relative importance (ranking), among others. In our experiments we ask the annotator to indicate the components that have been wrongly identified by the explanation as relevant, that is,

$$\mathcal{C} = \{j : |w_j| > 0 \wedge \text{the user believes the } j\text{th component to be irrelevant}\} .$$

In document classification, $\mathcal{C}$ would be the set of words that are irrelevant according to the user but relevant for the model.

Given the correction $\mathcal{C}$, we are faced with the problem of explaining it back to the learner. We propose a simple strategy to achieve this. This strategy is embodied by TOCOUNTEREXAMPLES. It converts $\mathcal{C}$ to a set of *counterexamples* that teach the learner not to depend on the irrelevant components. In particular, for every $j \in \mathcal{C}$ we generate c examples $(\bar{x}_1, \bar{y}_1), \ldots, (\bar{x}_c, \bar{y}_c)$, where c is an application-specific constant. Here, the labels $\bar{y}_i$ are identical to the prediction $\hat{y}$. The instances $\bar{x}_i$, $i = 1, \ldots, c$ are also identical to the query x, except that the jth component (i.e. $\psi_j(x)$) has been either randomized, changed to an alternative value, or substituted with the value of the jth component appearing in other training examples of the same class. This process produces $c \cdot |\mathcal{C}|$ counterexamples, which are added to $\mathcal{L}$.

Why is this data augmentation a sensible idea? To see this, consider the case of linear max-margin classifiers. Let $f(x) = \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle + b$ be a linear classifier over two features, ϕ_1 and ϕ_2, of which only the first is relevant. Fig. 2 shows that $f(x)$ (red line) uses ϕ_2 to correctly classify a negative example x_i. In order to obtain a better model (e.g. the green line), the simplest solution would be to enforce an orthogonality constraint $\langle \boldsymbol{w}, (0, 1)^\top \rangle = 0$ during learning. Counterexamples follow the same principle. In the separable case, the counterexamples $\{\bar{x}_{i\ell}\}_{\ell=1}^{c}$ amount to additional max-margin constraints [22] of the form $y_i \langle \boldsymbol{w}, \boldsymbol{\phi}(\bar{x}_{i\ell}) \rangle \geq 1$. The only ones that influence the model

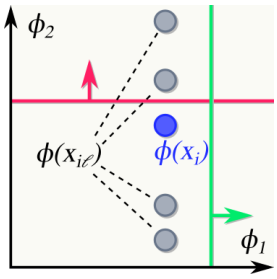

Figure 2: Mathematical intuition for the counterexample strategy. (Best viewed in color).

are those on the margin, for which strict equality holds. For all pairs of such counterexamples ℓ, ℓ' it holds that $\langle \boldsymbol{w}, \boldsymbol{\phi}(\bar{x}_{i\ell}) \rangle = \langle \boldsymbol{w}, \boldsymbol{\phi}(\bar{x}_{i\ell'}) \rangle$, or equivalently $\langle \boldsymbol{w}, \boldsymbol{\delta}_{i\ell} - \boldsymbol{\delta}_{i\ell'} \rangle = 0$, where $\boldsymbol{\delta}_{i\ell} = \boldsymbol{\phi}(\bar{x}_{i\ell}) - \boldsymbol{\phi}(x_i)$. In other words, the counterexamples encourage orthogonality between $\boldsymbol{w}$ and the correction vectors $\boldsymbol{\delta}_{i\ell} - \boldsymbol{\delta}_{i\ell'}$, thus approximating the orthogonality constraint above.

Most importantly, this data augmentation procedure is model-agnostic, although alternatives indeed exist: Contrastive examples [23], feature ranking [24] for SVMs and constraints on the input gradients for differentiable models [11].

Explanatory Interactive Learning with right for the right reason loss. Another method to regularize the learner to be right for the right reasons is the RRR introduced by Ross *et al.* [11]. This additional regularization term function adds a penalty to gradients that lie outside of a binary mask that indicates which features of the input are relevant. We modified the original loss function to:

$$L(\theta,\ X,\ y,\ A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -c_k y_{nk} \log(\hat{y}_{nk})}_{\text{Right answers}} + \underbrace{\lambda_1 \sum_{n=1}^{N}\sum_{d=1}^{D}\left(A_{nd}\frac{\delta}{\delta h_{nd}}\sum_{k=1}^{K} c_k \log(\hat{y}_{nk})\right)^2}_{\text{Right reasons}} + \underbrace{\lambda_2 \sum_i \theta_i^2}_{\text{Weight regularization}}, \quad (1)$$

where θ describes the parameters of the network, X the input, y the ground truth and A the binary mask used in the regularization term that discourages the input gradient from being large in regions marked by A. Instead of regularizing the gradients with respect to X, as originally described in [11], we regularize the gradients of the final convolutional layer h, corresponding to the GRAD-CAMS. Further c is a rescaling weight given to each class of the unbalanced dataset and $\hat{y}$ corresponds to the network prediction. The objective function is split into three terms. The first and the last are the familiar cross entropy and weight (θ) regularization terms. The second term is the new regularization term. The λ values are used to weight the different regularizations. Ross *et al.* [11] state that the regularization parameter λ_1 should be set such that the "right answers" and "right reasons" terms have similar orders of magnitude.
RRR can be adjusted to the XIL framework as in Alg. 1.

Let us now present several showcases that demonstrate the effectiveness of explanatory machine learning methods for understanding, validating, and correcting the behavior of a learned model. We finally investigate how providing explanations to users changes their trust in the model's choices.

Evaluation on a toy dataset. We begin by considering simulated users—as it is common for active learning—to evaluate the contribution of explanation feedback.Indeed, counterexample strategies (e.g. CAIPI) can trivially accommodate more advanced models than the one employed here. As is common in active learning, we simulate a human annotator that provides correct labels. Explanation corrections are also assumed to be correct and complete (i.e. they identify all false positive components), for simplicity[1].

Specifically, we applied our data augmentation strategy to a decoy variant of fashion MNIST, a fashion product recognition dataset[2]. The dataset includes 70,000 images over 10 classes. All images were corrupted by introducing confounders, that is, 4×4 patches of pixels in randomly chosen corners whose shade is a function of the label in the training set and random in the test set (see [11] for details). The average test set accuracy of a multilayer perceptron (with the same hyperparameters as in [11]) is reported in Tab. 1 (right) for three correction strategies: no corrections, our counterexample strategy (CE), and the input-gradient constraints proposed by [11] (RRRLoss). The model's explanations for CE are computed with LIME[3], additionally, for every training image we added $c = 1, 3, 5$ counterex-

[1] In practice corrections may be incomplete or noisy, especially when dealing with non-experts. This can be handled by, e.g., down-weighting the counterexamples.

[2] `https://github.com/zalandoresearch/fashion-mnist`

[3] Due to sampling, LIME may output different explanations for the same prediction. To reduce variance, we ran it 10 times and kept the k components identified most often.

User Study

	Q1	Q2	Q3
S1	65%	35%	82%
S2	29%	12%	41%
S3	77%	65%	71%

Fashion-MNIST (Toy) Dataset

	no corr.	Counterexamples $c=1$	Counterexamples $c=3$	Counterexamples $c=5$	RRR IG
Train	**97**%	93%	92%	92%	89%
Test	48%	82%	**85**%	**85**%	**85**%

Scientific Dataset

	no. corr.	RRR GRAD-CAM
RGB	88%	88%*
HS	99%	95%

Table 1: Explanatory feedback can boost trust and performance. (Left) User study: percentage of "yes" answers. (Middle) Accuracy on the fashion MNIST dataset of an MLP without corrections (no corr.), with our counterexample corrections using varying c (middle), and RRR with input gradient (IG) constraints [11]. (Right) The mean model balanced accuracy of applying the right for the right reason constraints with GRAD-CAM over 5 cross validation runs. With "*" we denote situations where decisions made based on the background could not be fully removed.

amples where the decoy pixels are randomized. When no corrections are given, the accuracy on the test set is 48%: the confounders completely fool the network, *cf.* Tab. 1(Middle). Providing even a single counterexample increases the accuracy to 82%, i.e., the effect of confounders drops drastically. With more counterexamples the accuracy of CE is similar to that of RRR, however, both methods pose valid improvements, thus showing that explanatory interactive learning is an effective measure for improving the model in terms of both predictive performance and beliefs.

Plant Phenotpying: Classification with Deep Learning results in high performance. Next we showcase the extent, importance, and usability of XIL. To this end, we performed classification and revised corrections of the learned models on a real-world, scientific dataset. This dataset corresponds to RGB and HS (ref. Methods) images of leaf tissue from inoculated (Cercospora beticola) and healthy sugar beet plants. Notably, there is a strong variability in the extent of disease severity over all samples, with some samples clearly showing characteristic CLS (two right most samples in Fig. 1) while others do not (second to the left sample in Fig. 1) and for the human eye appear indistinguishable —at least in RGB— from healthy leaves (top sample in Fig. 1). Roughly 50% of inoculated tissue samples showed visible CLS.

We performed classification using convolutional neural networks (CNNs) using 2D convolutions for the RGB dataset and 3D convolutions for the HS dataset (*cf.* Methods). The task was to classify the leaf samples into the classes healthy and diseased. The corresponding mean balanced accuracies determined over 5 cross-validation runs are shown in the left column (Default) of Tab. 1(Right), showing high accuracies on the RGB dataset and nearly perfect accuracies on the HS dataset. Indeed it seems the HS data to contain more relevant information for such a classification task.

Be careful! The explanations indicate that the classification is right for the wrong reasons. Given the difficult classification task, we want to know what the explanations of the networks are that lead them to such accurate predictions. We visualized the explanations of the networks using Gradient weighted Class Activation Mapping (GRAD-CAM) [25]. The resulting feature activation maps indicate which region of an input image is important for predicting a specific class and is based on the gradients of a given target class flowing into the last convolutional layer.

Following Lapuschkin et al. [1], we applied spectral clustering and t-SNE [26] analysis on the resulting explanations to better visualize and evaluate the decision strategies. Fig. 3 shows the different strategies of one example cross-validation fold for the corresponding validation set. The top figure shows the strategies of the CNN trained on the RGB data and the bottom figure shows the strategies of the CNN trained on the HS data (*cf.* Methods for details on the visualization). From Fig. 3 and Fig. 8 (upper left) one can identify different decision strategies, based on the data type (RGB or HS) and class prediction. The RGB-CNN has one strategy for control tissue, namely to focus on large regions of the healthy tissue. Interestingly, for samples wrongly classified as control the RGB-CNN shows a similar strategy as for truly control samples. When CLS were clearly visible the RGB-CNN correctly identifies these as relevant features for classifying the samples as inoculated. However, for many inoculated samples, for which no spots are visible the CNN surprisingly focuses on regions in the background, specifically often on the nutrition solution (agar), which the tissue was embedded in.

This background strategy is even more radically developed by the HS-CNN. In the bottom graphic of Fig. 3 one can identify that the HS-CNN has altogether two prediction strategies, one for each predicted class label (*cf.* Fig. 8 (bottom left)). In the case of control samples, the HS-CNN focuses on large areas of the tissue, however for inoculated samples, even if CLS are visible, the network focuses on the nutritional solution (agar) in order to classify these as inoculated. Moreover, when analyzing the reflectance of the agar across different stages of disease development, we could indeed identify differences between control and inoculated nutrition solution. This can be seen in the left panel of Fig. 7. Given the much smaller data dimensionality of the RGB images compared to the HS data, it seems likely

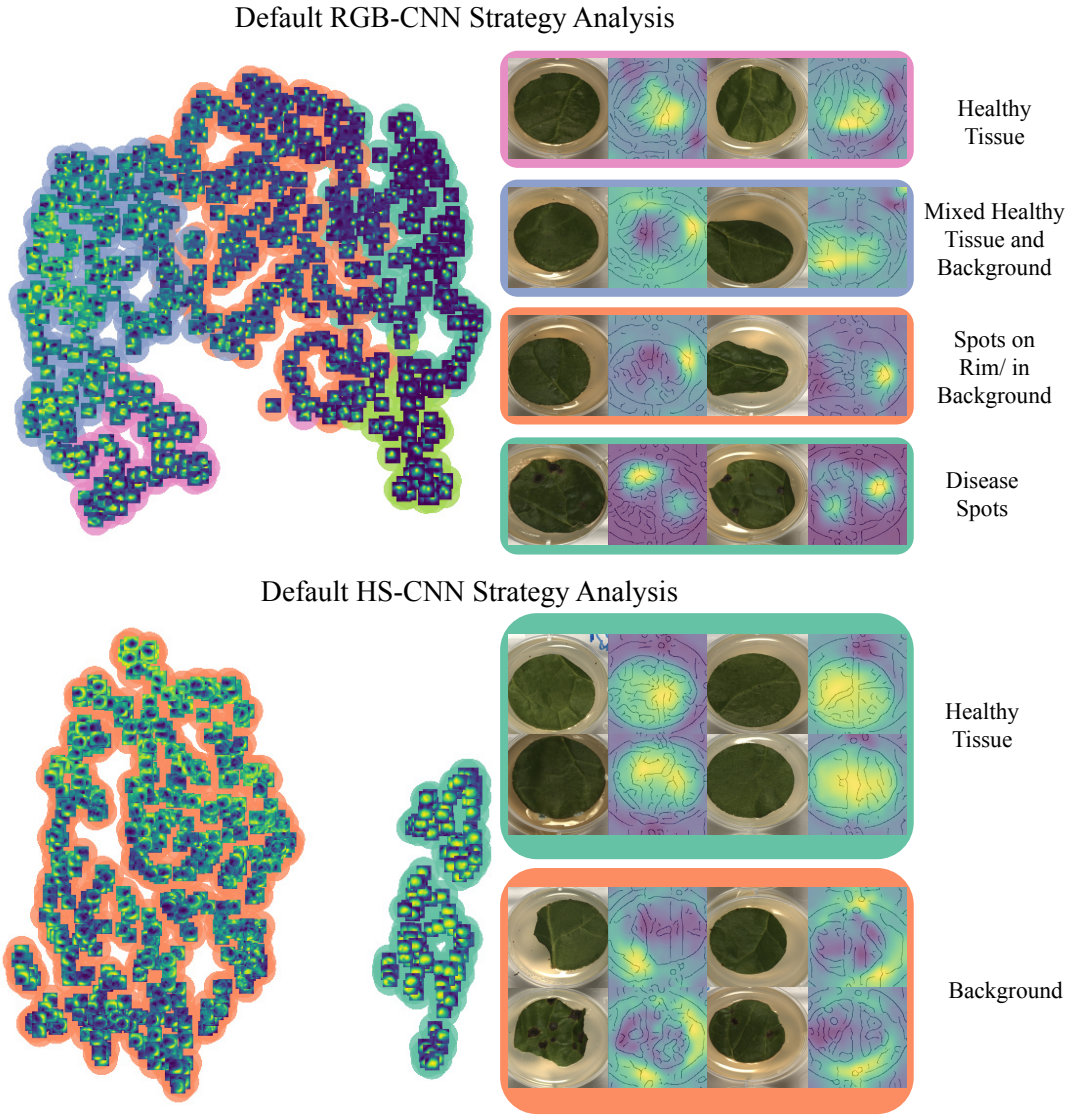

Figure 3: Cluster analysis of the different decision strategies after training CNNs with the cross entropy loss (Default). The top row specifies the strategies of a CNN trained with RGB images. The bottom row specifies the strategies of a CNN trained with hyperspectral images. The images are visualized in a two-dimensional t-SNE embedding and colored by the spectral clustering assignments.

that the RGB-CNN would have more difficulties focusing only on the agar as a classification feature, thus explaining the different classification strategies between HS and RGB-CNNs as well as the reduced classification performance of the RGB-CNN, compared to the HS-CNN.

To summarize, the CNNs showed high to very high performances by largely using confounding factors within the dataset. Indeed the trained neural networks used strategies which a human would consider as cheating rather than valid problem-solving behavior. In this state the accuracies may not correspond to the true performance when measured in an environment outside of the lab setting. Possibly even leading to dangerous consequences if left unchecked.

Correcting the model to classify right for the right reasons. Now, it is actually too simple to say that we can not trust these models and even question if machines are truly "intelligent". In this work we show, that with the human in the loop revising the machine, as in a XIL setting, the models are actually able to recover from so called "Clever Hans" strategies to trustful behavior by constraining it's explanations if they are not reconcilable.

For this, we let an expert revise the learning of the machine by constraining the machine's explanations to match domain knowledge. For simplicity, we focused our results on using the RRRLoss [11], rather than using a CE strategy, though both would be valid here within the XIL framework. Specifically, we added a second regularization term to our loss function such that we penalize the model during training when it uses an area which might exploit a dataset artifact. We simulated an expert user by predefining these areas before training and, as an initial step, these areas

corresponded to binary masks of the whole tissue (*cf.* Methods). We modified the original RRRLoss by adding class balancing weights and using Gradient weighted Class Activation Mapping [27] for creating the model explanations (*cf.* Methods).

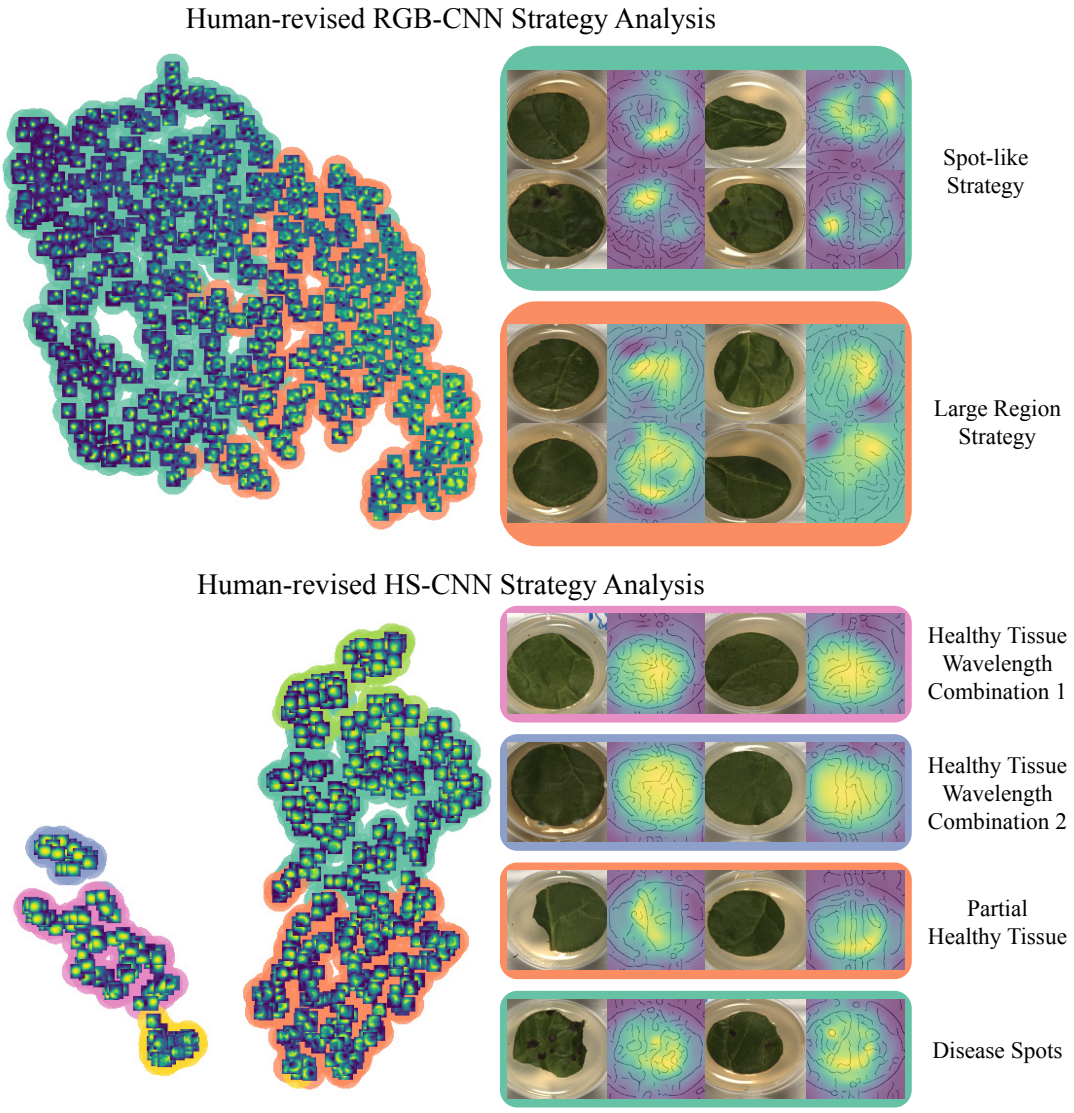

Figure 4: Cluster analysis of the different decision strategies after training CNNs with a cross entropy loss and a right for the right reason loss. The top row specifies the strategies of a CNN trained with RGB images. The bottom row specifies the strategies of a CNN trained with hyperspectral images.

Similar to the default training mode (i.e. cross entropy loss and L2 weight regularization) we analyzed the decision strategies of the RRRLoss training mode using t-SNE and spectral clustering. The results can be seen in Fig. 4, with the explanations of the RGB-CNN in the top panel, whereas those of the HS-CNN located in the bottom panel. As one can see, the number of decision strategies has largely decreased after training the RGB-CNN with the RRRLoss. Essentially two strategies could be recognized, corresponding to a small scale, spot-like strategy and large-region strategy. In general the model could be improved to focus more on regions of the tissue, however, still there are cases in which the model focuses on features outside of the tissue region. We further observed single examples in which the explanations in the default loss were located on the tissue, however after training with RRRLoss these were located on the background. This behaviour could not be removed with varying hyper-parameter settings.

In comparison, the number of decision strategies of the RRRLoss trained HS-CNN increased. Upon closer investigation, however, this was largely due not to different spatial strategies, but rather different combinations of wavelengths —the final convolutional layer of the HS-CNN contains four channels corresponding to four different spectral regions (*cf.* Fig. 6). One can see that in comparison to the RGB-CNN, after training with the RRRLoss the model focuses on image regions lying only on the tissue. The strategies of control samples correspond to nearly full activation of the whole tissue, whereas for inoculated samples the identified relevant image regions are often not as large-scale. Particularly the model now focuses on the CLS, which it had previously largely ignored. Fig. 1 (Bottom panel) shows in more detail several examples of the observed strategies used by the corrected HS-CNN in comparison

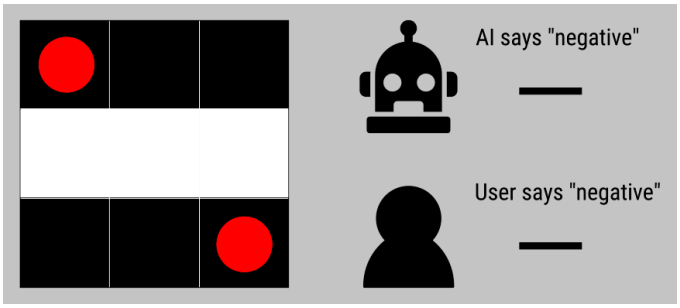

Figure 5: Example training round as presented in the questionnaire. The classification is correct but the explanation shows that the two most relevant pixels do not match the true classification rule (as in S3). (Best viewed in color).

to the observed "Clever Hans" strategies of the default, not revised machine.

Although the model's performance slightly decreased, *cf.* Tab. 1(Right), it is still able to classify samples without visible symptoms. It is important to note here that although applying several different hyper-parameters for the RRRLoss it was not possible to entirely prevent the RGB-CNN from focusing on the background. The unchanged accuracy for the RGB-CNN with RRRLoss of Tab. 1 must therefore be taken with a grain of salt.

With reference to the left panel of Fig. 7 one can indeed see that the HS-CNN has much more information at hand to focus on the confounding factors in the first place. However, also after revision with the RRRLoss it is easier for the HS-CNN to make accurate predictions based on the reflectance of the tissue in comparison to the RGB-CNN (middle panel of Fig. 7). Particularly, the HS-CNN mainly uses a spectral area —the third left heatmap of Fig. 6— for prediction, which is beyond the RGB area. Thus explaining the difficulty of correcting the RGB-CNN.

Does using explanations gain trust of users? We designed a questionnaire about a machine that learns a simple concept by querying labels (but *not* explanation corrections) to an annotator. The questionnaire, available in the Supplementary Material, was administered to 17 randomly selected undergraduate students from an introductory course on deep learning. Specifically, we designed a toy binary classification problem (inspired by [11]) about classifying small (3×3) black-and-white images. The subjects were told that an image is positive if the two top corners are white and negative otherwise. Then they were shown three learning sessions consisting of five query/feedback rounds each.

In session 1 (**S1**) every round included the images chosen by the model, the corresponding prediction, and the label provided by a knowledgeable annotator. No explanations were shown. The predictions are wrong for the first three rounds and correct in the last two. Sessions 2 and 3 (**S2**, **S3**) were identical to S1, meaning that at every round *the same example, prediction and feedback label* were shown, but now explanations were also provided. The explanations highlighted the two most relevant pixels, as in Fig. 5. In S2 the explanations did not converge to the correct rule—they highlight the two top corners—from the fourth round onwards, while in S3 they did. Removing the explanations reduces both S2 and S3 to S1.

After each session, the subjects were asked three questions: (**Q1**) "Do you believe that the AI system eventually learned to classify images correctly?", (**Q2**) "Do you believe that the system eventually learned the correct classification rule?", and (**Q3**) "Would you like to further assess the system by checking whether it classifies 10 random images correctly?". The first two questions test the subject's uncertainty in the predictive ability and beliefs of the classifier, respectively, while the last one tests the relationship between predictive accuracy (but *not* explanation correctness) and expected uncertainty reduction. The percentage of "yes" answers is down in Tab. 1(Left).

As expected, the uncertainty in the model's correctness depends heavily on what information channels are enabled. When no explanations are shown (S1), only 35% of the subjects assert to believe that the model learned the correct rule (Q2). This percentage almost doubles (65%) when explanations are shown and converge to the correct rule (S2). The need to see more examples also lowers from 82% to 71%, but does not drop to zero. This reflects the fact that five rounds are not enough to reduce the subject's uncertainty to low enough levels. The percentage of subjects asserting that the classifier produces correct predictions (regardless of the learned rule, Q1) also increases from 65% to 77% when correct explanations are shown (S2). When the explanations do not converge (S3), the trend is reversed: Q1 drops to 29% and Q2 to 12%, i.e., most subjects do not believe that the model's behavior and beliefs are in any way correct. This is the only setting where Q3 drops below 50% (41%): witnessing that the model's beliefs do not match the target rule induces distrust (with high certainty). This confirms the previous finding that trust into machines drops when wrong behavior is witnessed [3]. Thus, augmenting interaction with explanations does appropriately drive trust into the model.

Discussion

In recent years, AI methods, especially machine learning with various directions and algorithms [28, 29], have become more and more successful in a wide range of areas like computer vision, natural language processing, and robotics, among others. Consider e.g. AlphaZero surpassing human level performance in playing chess and Go. During its self-play training process, AlphaZero discovered a remarkable level of Go knowledge. This included not only fundamental elements of human Go knowledge, but also non-standard strategies beyond the scope of traditional human Go knowledge [30]. Thus exemplifying the potential of these methods to discover strategies previously unknown even to experts of the domain. However, studies from various applications such as [31], [32], [33] and [1], have revealed that learning machines can also result in human-undesired strategies, e.g., the machine exploiting dataset artifacts.

We introduced the novel learning setting of "explanatory interactive learning" (XIL) and illustrated its benefits on an important research task, namely, plant phenotpying. XIL adds the scientist into the training loop. She interactively revises the original model via providing feedback on its explanations. Our experimental results demonstrate that XIL can help avoiding Clever Hans moments in machine learning and encourages (or discourages, if appropriate) trust into the underlying model. This shows that the vision of Donald E. Knuth—"*Let us change our traditional attitude to the construction of programs. Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.*"—is not insurmountable for machine learning.

There are several possible extensions for future work. Other interactive learning approaches such as coactive [10], active imitation [20], mixed-initiative interactive [21] and guided probabilistic learning [34] should be made explanatory. While it is not fully established yet what the important features are that make up an optimal human interpretable explanation [35], in order to allow better interaction and communication between the user and model, it is necessary to extend the explanations to multiple modalities, e.g. expand visual with textual and counterexemplar information [36]. Ras *et al.* [37] already postulated that multi-modal explanation methods are required to answer specific questions in a simple human interpretable language and mentioned the necessity to adapt the explanation complexity to the requirement of the relevant setting and end-user. Building inherently interpretable machine learning models [38] is an exciting approach to understanding a model's decision. We argue that even in this setting it is necessary to learn and explain interactively, for the user to understand and appropriately build trust in the model's decisions.

Methods

Active learning. The active learning paradigm targets scenarios where obtaining supervision has a non-negligible cost. Here we cover the basics of pool-based active learning, and refer the reader to two excellent surveys [39, 40] for more details. Let $\mathcal{X}$ be the space of instances and $\mathcal{Y}$ be the set of labels (e.g. $\mathcal{Y} = \{\pm 1\}$). Initially, the learner has access to a small set of labelled examples $\mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$ and a large pool of unlabelled instances $\mathcal{U} \subseteq \mathcal{X}$. The learner is allowed to query the label of unlabelled instances (by paying a certain cost) to a user functioning as annotator, often a human expert. Once acquired, the labelled examples are added to $\mathcal{L}$ and used to update the model. The overall goal is to maximize the model quality while keeping the number of queries or the total cost at a minimum. To this end, the query instances are chosen to be as informative as possible, typically by maximizing some informativeness criterion, such as the expected model improvement [41] or practical approximations thereof. By carefully selecting the instances to be labelled, active learning can enjoy much better sample complexity than passive learning [42, 43]. Prototypical active learners include max-margin [44] and Bayesian approaches [45]; recently, deep variants have been proposed [46].

However, active (showing query data points) and even coactive learning (showing additionally the prediction of the query data point) do not establish trust: informative selection strategies just pick instances where the model is uncertain and likely wrong. Thus, there is a trade-off between query informativeness and user "satisfaction", as noticed and explored in [47]. In order to properly modulate trust into the model, we argue it is essential to present explanations.

Local explainers. There are two main strategies for interpreting machine learning models. Global approaches aim to explain the model by converting it *as a whole* to a more interpretable format [6],[48]. Local explainers instead focus on the arguably more approachable task of explaining *individual predictions* [8]. While explainable interactive learning can accommodate any local explainer, in our implementations we use either LIME [7] or GRAD-CAM [25], both described next.

The idea of LIME (Local Interpretable Model-agnostic Explanations) is simple: even though a classifier may rely on many uninterpretable features, its decision surface around any given instance can be locally approximated by a simple, interpretable *local model*. In LIME, the local model is defined in terms of simple features encoding the presence

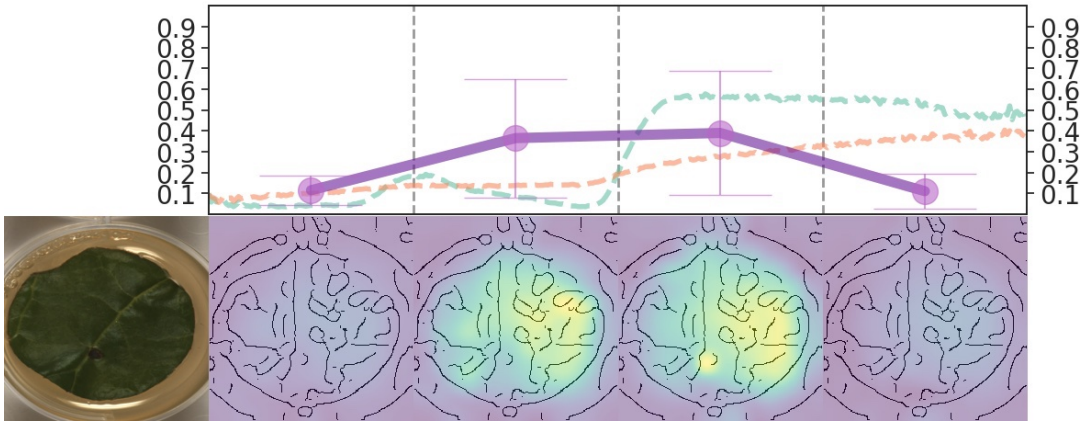

Figure 6: GRAD-CAMS with spatial and spectral explanations of a corrected network. Bottom-left shows the sample followed by the corresponding spatial activations maps mapped to four different hyperspectral areas. Top shows the activation (mean and standard deviation) in the hyperspectral domain (purple) with pixel-wise hyperspectral samples for visualization.

or absence of *basic components*, such as words in a document or objects in a picture[4]. An explanation can be readily extracted from such a model by reading off the contributions of the various components to the target prediction and translating them to an interpretable visual artifact. For instance, in document classification one may highlight the words that support (or contradict) the predicted class.

GRAD-CAMS are a generalization of Class Activation Maps, introduced by [27] and take advantage of the facts that, firstly, deeper layers of a CNN capture higher-level visual constructs and, secondly, that convolutional features retain spatial information. As such, the last convolutional layer represents a trade-off between high visual representation and spatial information. Specifically, a GRAD-CAM is computed by forward passing an image through the network, applying a backpropagation of a one-hot encoding vector which specifies the class label of interest up to the last convolutional layer. The resulting gradients of each channel are global average pooled, multiplied with the corresponding feature maps, summed and finally passed through a RELU activation function. In this way the final feature maps of the convolutional feature extractor are weighted by the importance of these features. The resulting two-dimensional heatmap can finally be interpolated to the original input size for visualization. In case a 3D convolutional network is used to classify hyperspectral data the resulting heatmap is three dimensional also showing activations along the spectral dimension of the data, *cf.* Fig. 6.

Sample collection. The dataset used in this study corresponds to HS and RGB images of leaf discs of sugar beet cv. Isabella (KWS, Einbeck, Germany) inoculated with *Cercospora beticola*. For this sugar beet seeds were pre-grown in small pots and piqued after the primary leaves were fully developed. The seedlings were then transferred into plastic pots (diameter of 17 cm) on commercial substrate (Topfsubstrat 1.5, Balster Erdenwerk, GmbH, Sinntal-Altengronau, Germany) under greenhouse conditions and watered as necessary. After reaching growth stage 16 according to BBCH scale [49] the plants were inoculated with *C. beticola* conidia which were collected from infested sugar beet leaves after incubation in a moist chamber for 48 hours. A spore suspension of 5×10^5 was sprayed onto leaves before the plants were transferred into plastic bags to achieve 100% RH for 48 hours. For image acquisition leaf discs were stamped out with a cork borer of 2 cm diameter and placed on 10g/l pyhtoagar (Duchefa Biochemie B.V, Haarlem, Netherlands), containing 0.34 mM benzimidazole, 10 g sucrose and 3 mg kinetin. To observe different symptom classes sugar beet leaves of 9, 14 and 19 days after inoculation (dai) were used since first symptoms appeared 9 dai. As a control group 18 leaf discs of untreated sugar beet plants were measured as well and five technical replications with 6 discs each were used for each symptom group.

Each sample was measured over five consecutive days such that a sample from 9 dai reappears four further times in the dataset as 10-13 dai. Untreated leaf discs were also recorded over five consecutive days. The percentage of healthy leaves to unhealthy leaves was approximately 17% to 83%, respectively. For image acquisition leaf discs on agar were placed on a linear stage at a distance of 53 cm to a Hyperspec VNIR E-series imaging sensor (Headwall Photonics, Bolton, MA, USA) in the range of 380 nm to 1000 nm and a Hyperspec SWIR imaging sensor (Headwall Photonics, Bolton, MA, USA) in the range of 900 to 2500 nm. The VNIR sensor has a spectral resolution of 2-3 nm and a pixel pitch of 6.5 μm. The sensor was surrounded by eight lamps (Ushio Halogen Lamp J12V-150WA/80 (Marunouchi, Chiyoda-ku, Tokyo, Japan)) and the distance between lamps and leaves was 60 cm with a vertical

[4]While not all problems admit explanations in terms of elementary components, many of them do [7]; in this case, LIME assumes these to be provided in advance.

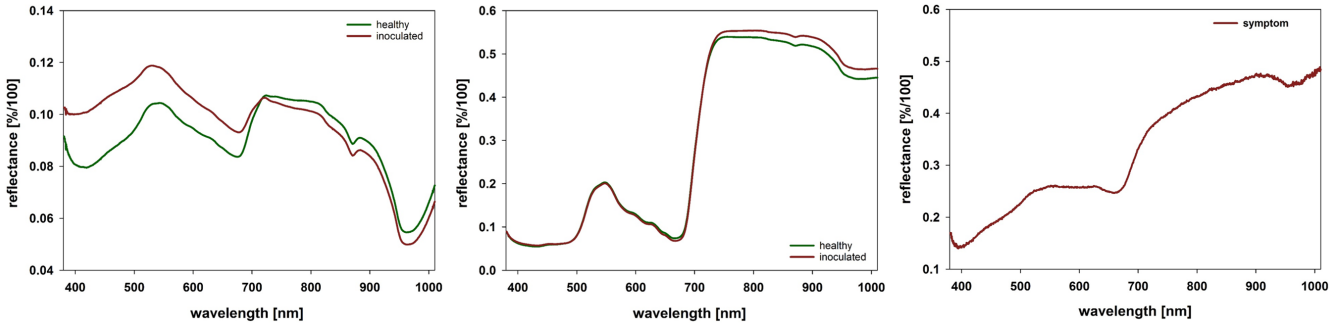

Figure 7: Spectral signatures of measured agar plates with sugar beet leaf discs. Signatures were extracted of agar on which healthy and inoculated sugar beet leaf discs were placed (left), of healthy and inoculated sugar beet leaf discs (middle) and of *C. beticola* symptoms on sugar beet leaves (right). Signatures were extracted from 100 pixel for each group and the mean value is presented.

orientation of 45. Exposure times of 44 ms were used for the VNIR sensor.Hyperspectral images were taken daily for five consecutive days.

Data preparation. As mentioned above, each sample was imaged over five consecutive days such that each sample, though slightly differing from day to day, is represented up to 5 times within the full dataset. In this way a sample from 9 dai would occur for 4 further days (10-13 dai). In order to prevent the models from memorizing the structure of the individual leaf samples and correlating this to the corresponding labels, a precaution was taken to exclusively contain all days of one sample either in the training or validation dataset.

RGB/HS classification. The RGB images used for training the classifiers were generated from the hyperspectral data, by slicing the data at the corresponding RGB channels that were provided by the camera system (*cf.* Fig. 1 (A-Right)). Prior to training the RGB classifiers, the data was standard scaled following $z = (x - u)/s$, where u is the mean and s the standard deviation of the training samples.

To train a classifier on the RGB images of sugar beet leaves we used a VGG16 [50] network pretrained on ImageNet [51] to finetune the network parameters using the RGB plant images. Additional training information and hyperparameters that were used were a batch size of 32, learning rate of 1e-4 and a step learning rate scheduler set to reduce the learning rate at epochs 5 and 15 by a factor of 0.1. Furthermore the ADAM optimizer was used with L2 regularization 1e-5. Five separate cross validation folds were trained until convergence, using a data split of 0.75 for training and 0.25 for testing. Convergence was reached after 30 epochs.

To classify the HS data we trained a convolutional neural network (CNN) architecture with batch normalization using 3D convolution filters, rather than standard 2D filters, learning features not only along the image dimensions, but also over the spectral dimensions. The used network is build up with four residual blocks, each containing one to three convolutional layers. The last two layers are fully connected layers with a final softmax activation function. The other layers use ReLU activations. During training the networks we use dropout to prevent overfitting. The network's parameters are trained with an stochastic gradient descent optimizer with momentum using a batch size of 10 HS images, a learning rate of 1e-4 and a L2 regularization of 1e-5.

Five separate cross validation folds were trained until convergence, using a data split of 0.75 for training and 0.25 for testing. Convergence was reached after 100 epochs.

Analyzing classification strategies of the model. Based on the results of [52], in which the authors performed sanity checks over a variety of saliency methods, we chose to investigate our model's explanations using Gradient-weighted Class Activation Mapping (GRAD-CAM)[25].

In order to analyze the resulting strategies produced by the layer-wise relevance propagation method (LRP), the authors of [1] revert to using spectral clustering on the resulting heatmaps in a pipeline they termed 'SpRAy'. We apply SpRAy in a similar way, however, rather than using the raw GRAD-CAM heatmaps, we perform a discrete Fourier transformation on these beforehand in order to better differentiate the different strategies. In detail the pipeline is as follows

- Perform a discrete Fourier transform on downsized GRAD-CAM heatmaps, as in [1].
- Using the Cityblock metric compute a k-nearest neighbor graph of the Fourier transformed heatmaps, represented as an adjacency matrix, C.
- Compute the affinity matrix as suggested in [53] as $A = max(C, C^T)$.

- Perform an eigengap analysis [53] to estimate the number of clusters, k, within the dataset.
- Perform spectral clustering on the affinity matrix, given k from the previous step
- Perform a t-SNE analysis [26] on the similarity matrix, estimated from the affinity matrix as in [1] as $S = \frac{1}{A+\epsilon}$, whereby $\epsilon \in [0, 1]$, here we used $\epsilon = 0.05$.

Applying XIL to CNNs for scientific dataset. We produced the matrix A (Eq. 1) corresponding to full tissue masks for each sample. Specifically, for each sample we created a binary mask having values of zero within the tissue and values of one everywhere else, i.e. the background. In this way during training the gradients everywhere but on the tissue are to be minimized.

The network models were retrained from the same initial values as in the default training mode (using only the cross-entropy loss), however now using RRR. To choose the optimal λ_1 value, the resulting explanations were visually assessed. The five cross validation folds of HS-CNN were thus trained until convergence between 200 and 280 epochs using a $\lambda_1 = 20$ value, with all other hyperparameters as in the default training mode. For training the RGB-CNN with RRR the learning rate was reduced to a constant learning rate of 5e-05. Although applying a range of λ_1 values from 0.1 to 1000, using the RGB-CNN, no satisfactory convergence state could be reached in which the regularized model showed acceptable explanations. The accuracy in Tab. 1 and the strategies presented in the Fig. 4 and 8 correspond to the GRAD-CAMS of training the five cross-validation folds for 60 epochs with $\lambda_1 = 1$.

Acknowledgments ST an KK thank Antonio Vergari, Andrea Passerini, Samuel Kolb, Jessa Bekker, Xiaoting Shao, and Paolo Morettin for very useful feedback on the conference version of this article. Furthermore, the authors are thankful to Cigdem Turan for providing the figure sketches, and to Ulrike Steiner and Stefan Paulus for very useful feedback. PS, AKM, AB and KK acknowledge the support by BMEL funds of the German Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the innovation support program, project "DePhenS" (FKZ 2818204715). WS an KK were also supported by BMEL/BLE funds under the innovation support program, project "AuDiSens" (FKZ 28151NA187). ST acknowledges the supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme, grant agreement No. [694980] "SYNTH: Synthesising Inductive Data Models". KK also acknowledges the support by the German Science Foundation project "CAML" (KE1686/3-1) as part of the SPP 1999 (RATIO).

References

[1] Lapuschkin, S. *et al.* Unmasking clever hans predictors and assessing what machines really learn. *Nature communications* **10**, 1096 (2019).

[2] Simpson, J. A. Psychological foundations of trust. *Current directions in psychological science* **16**, 264–268 (2007).

[3] Hoffman, R. R., Johnson, M., Bradshaw, J. M. & Underbrink, A. Trust in automation. *IEEE Intelligent Systems* **28**, 84–88 (2013).

[4] Waytz, A., Heafner, J. & Epley, N. The mind in the machine: Anthropomorphism increases trust in an autonomous vehicle. *Journal of Experimental Social Psychology* **52**, 113–117 (2014).

[5] Wang, N., Pynadath, D. V. & Hill, S. G. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, 109–116 (IEEE Press, 2016).

[6] Buciluǎ, C., Caruana, R. & Niculescu-Mizil, A. Model Compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, 535–541 (ACM, 2006).

[7] Ribeiro, M. T., Singh, S. & Guestrin, C. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144 (ACM, 2016).

[8] Lundberg, S. & Lee, S.-I. An unexpected unity among methods for interpreting model predictions. *arXiv preprint arXiv:1611.07478* (2016).

[9] Settles, B. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the conference on empirical methods in natural language processing*, 1467–1478 (Association for Computational Linguistics, 2011).

[10] Shivaswamy, P. & Joachims, T. Coactive learning. *Journal of Artificial Intelligence Research* **53**, 1–40 (2015).

[11] Ross, A. S., Hughes, M. C. & Doshi-Velez, F. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2662–2670 (2017).

[12] Kulesza, T. *et al.* Principles of explanatory debugging to personalize interactive machine learning. In *Proc. of IUI*, 126–137 (2015).

[13] Lau, E. High-throughput phenotyping of rice growth traits. *Nature Reviews Genetics* **15**, 778–778 (2014).

[14] de Souza, N. High-throughput phenotyping. *Nature Methods* 36–36 (2009).

[15] Tardieu, F., Cabrera-Bosquet, L., Pridmore, T. & Bennett, M. Plant Phenomics, From Sensors to Knowledge. *Current Biology* **27**, R770–R783 (2017).

[16] Pound, M. P. *et al.* Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *Gigascience* **6**, gix083 (2017).

[17] Mochida, K. *et al.* Computer vision-based phenotyping for improvement of plant productivity: a machine learning perspective. *GigaScience* **8**, giy153 (2018).

[18] Teso, S. & Kersting, K. Explanatory interactive machine learning. In *Proceedings of AIES19* (AAAI, 2019).

[19] Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* **1**, 206–215 (2019). URL `https://doi.org/10.1038/s42256-019-0048-x`.

[20] Judah, K. *et al.* Active imitation learning via reduction to iid active learning. In *UAI*, 428–437 (2012).

[21] Cakmak, M. *et al.* Mixed-initiative active learning. *ICML 2011 Workshop on Combining Learning Strategies to Reduce Label Cost* (2011).

[22] Cortes, C. *et al.* Support-vector networks. *Machine learning* **20**, 273–297 (1995).

[23] Zaidan, O. *et al.* Using "annotator rationales" to improve machine learning for text categorization. In *NAACL HLT*, 260–267 (2007).

[24] Small, K. *et al.* The constrained weight space svm: learning with ranked features. In *ICML*, 865–872 (Omnipress, 2011).

[25] Selvaraju, R. R. *et al.* Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 618–626 (2017).

[26] Maaten, L. v. d. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9**, 2579–2605 (2008).

[27] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929 (2016).

[28] Jordan, M. I. & Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science* **349**, 255–260 (2015).

[29] Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature* **521**, 452–459 (2015). URL `https://doi.org/10.1038/nature14541`.

[30] Silver, D. *et al.* Mastering the game of go without human knowledge. *Nature* **550**, 354–359 (2017).

[31] Zech, J. R. *et al.* Confounding variables can degrade generalization performance of radiological deep learning models. *arXiv preprint arXiv:1807.00431* (2018).

[32] Badgeley, M. A. *et al.* Deep learning predicts hip fracture using confounding patient and healthcare variables. *npj Digital Medicine* **2**, 31 (2019).

[33] Chaibub Neto, E. *et al.* A permutation approach to assess confounding in machine learning applications for digital health. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 54–64 (ACM, 2019).

[34] Odom, P. & Natarajan, S. Human-guided learning for probabilistic logic models. *Frontiers in Robotics and AI* **5**, 56 (2018).

[35] Narayanan, M. *et al.* How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682* (2018).

[36] Kanehira, A. & Harada, T. Learning to explain with complemental examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8603–8611 (2019).

[37] Ras, G., van Gerven, M. & Haselager, P. Explanation methods in deep learning: Users, values, concerns and challenges. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, 19–36 (Springer, 2018).

[38] Chen, C. *et al.* This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 8928–8939 (2019). URL `http://papers.nips.cc/paper/9095-this-looks-like-that-deep-learning-for-interpretable-image-recognition`.

[39] Settles, B. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* **6**, 1–114 (2012).

[40] Hanneke, S. *et al.* Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning* **7**, 131–309 (2014).

[41] Roy, N. *et al.* Toward optimal active learning through monte carlo estimation of error reduction. *ICML* 441–448 (2001).

[42] Castro, R. M. *et al.* Upper and lower error bounds for active learning. In *The 44th Annual Allerton Conference on Communication, Control and Computing*, 2.1, 1 (2006).

[43] Balcan, M.-F. *et al.* The true sample complexity of active learning. *Machine learning* **80**, 111–139 (2010).

[44] Tong, S. & Koller, D. Support vector machine active learning with applications to text classification. *JMLR* **2**, 45–66 (2001).

[45] Krause, A. *et al.* Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *ICML*, 449–456 (ACM, 2007).

[46] Gal, Y. *et al.* Deep bayesian active learning with image data. In *Proc. of ICML*, 1183–1192 (2017).

[47] Schnabel, T. *et al.* Short-term satisfaction and long-term coverage: Understanding how users tolerate algorithmic exploration. In *Proc. of WSDM*, 513–521 (ACM, 2018).

[48] Bastani, O. *et al.* Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504* (2017).

[49] Meier, U. *et al.* Phenological growth stages of sugar beet (*Beta vulgaris* l. ssp.) codification and description according to the general bbch scale (with figures). *Nachrichtenblatt des Deutschen Pflanzenschutzdienstes* **45**, 37–41 (1993).

[50] Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[51] Deng, J. *et al.* ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of CVPR09* (2009).

[52] Adebayo, J. *et al.* Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 9505–9515 (2018).

[53] Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing* **17**, 395–416 (2007).

Supplementary materials

Questionnaire used in User Study

The Questionnaire document used can be found at `https://github.com/stefanoteso/caipi/blob/master/form/questionnaire.pdf`

Strategy analysis classification errors

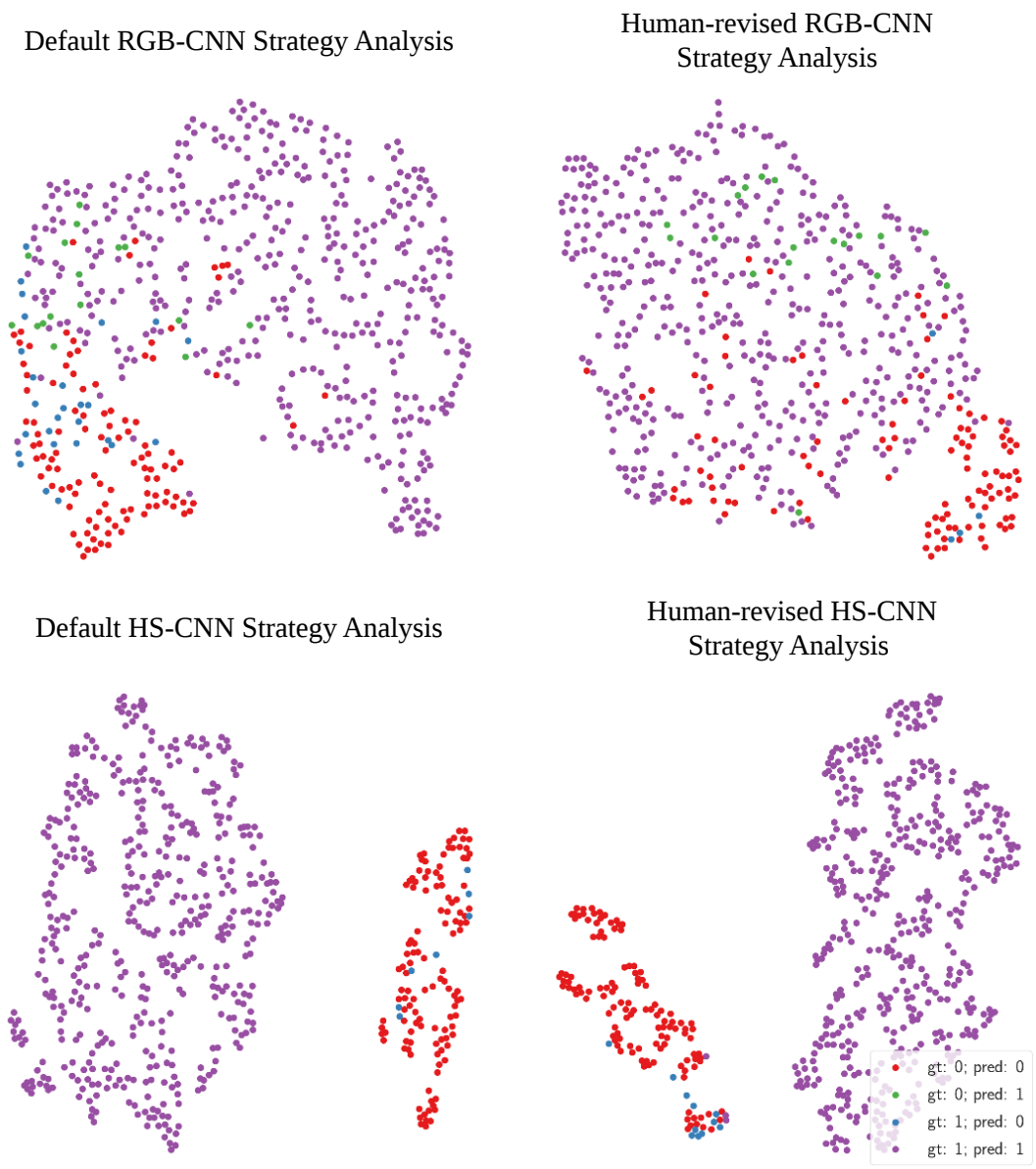

Figure 8: Appendix. T-SNE embeddings of the different image type and training loss configurations colored by the ground truth (gt) and prediction (pred) labels of each sample. The top row depicts the results of training with RGB images, the bottom row with hyperspectral images. The left column shows the results of training only with the cross-entropy loss, whereas the right column shows the results of training with the right for the RRRLoss. When comparing to Figures 3 and 4 note the strong differentiation of decision strategies of the HS-CNN between healthy and diseased samples.

Explanations along hyperspectral data dimension

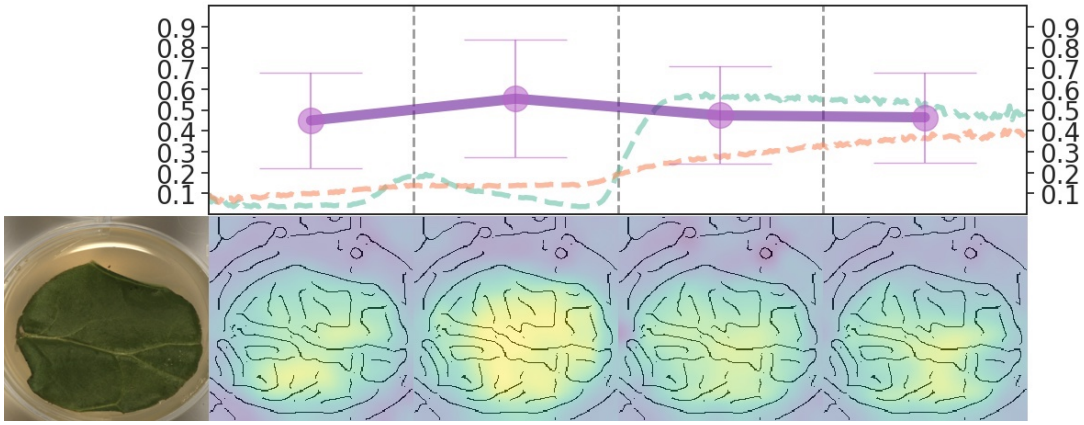

Figure 9: Appendix. Spatial and spectral activations from healthy sample of not regularized network.

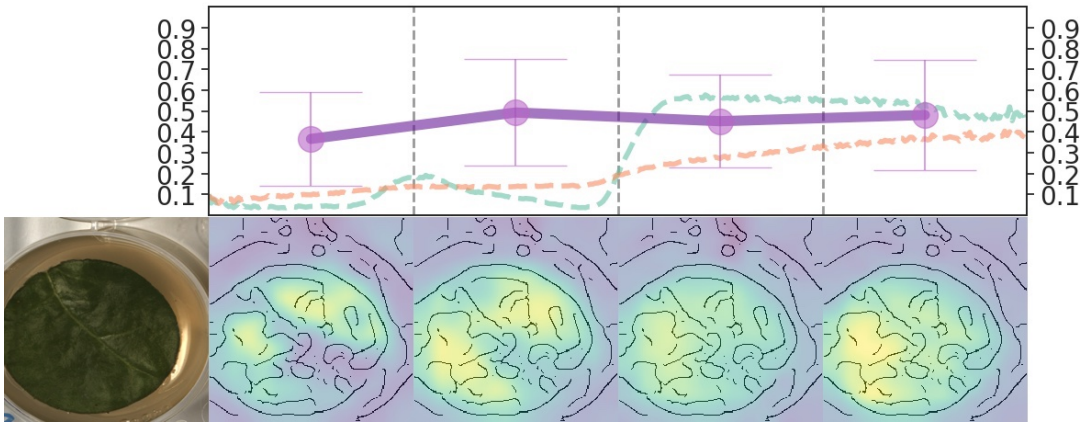

Figure 10: Appendix. Spatial and spectral activations from healthy sample of not regularized network.

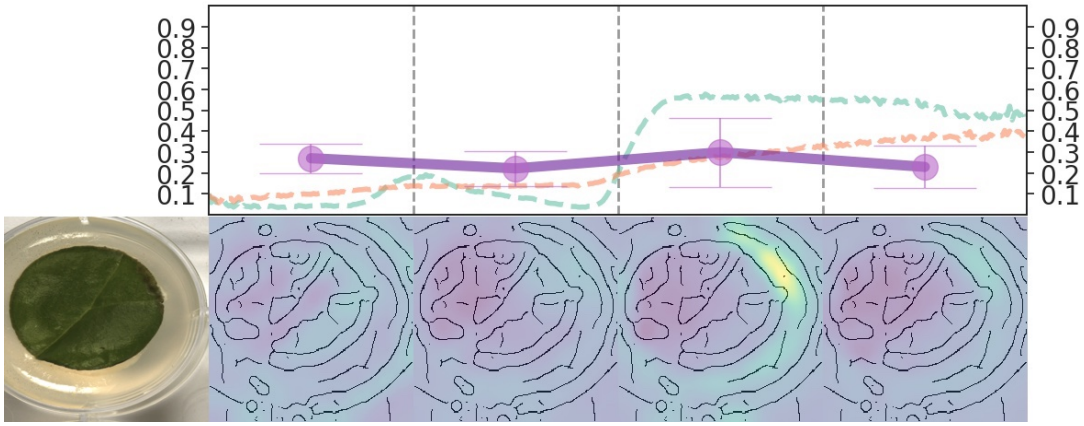

Figure 11: Appendix. Spatial and spectral activations from inoculated sample without visible symptoms of not regularized network.

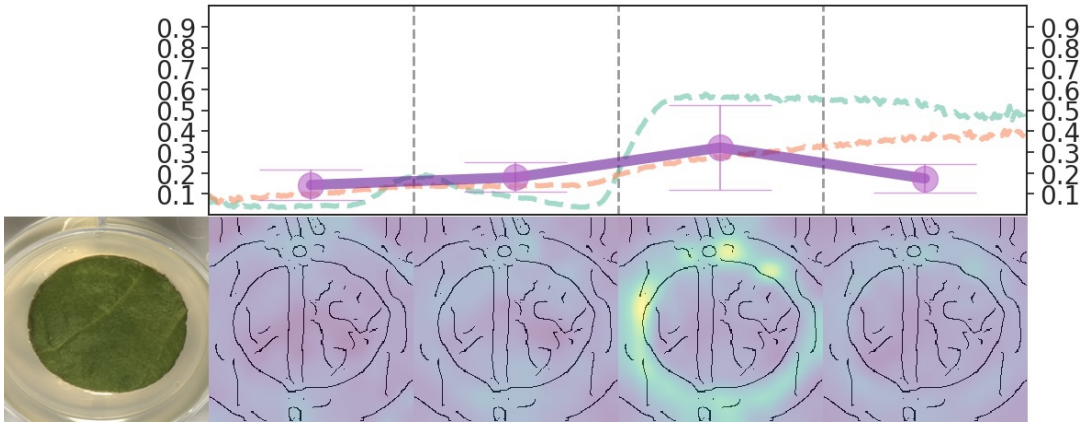

Figure 12: Appendix. Spatial and spectral activations from inoculated sample without visible symptoms of not regularized network.

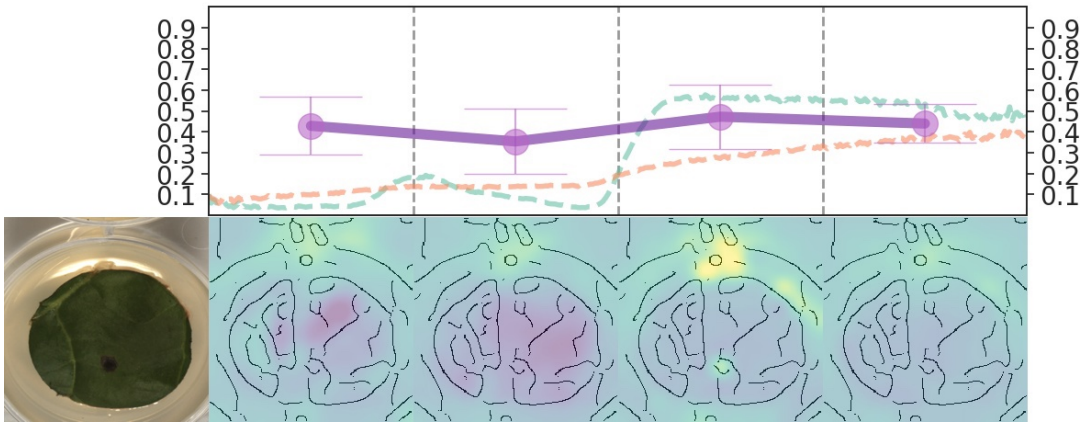

Figure 13: Appendix. Spatial and spectral activations from inoculated sample with single visible symptoms of not regularized network.

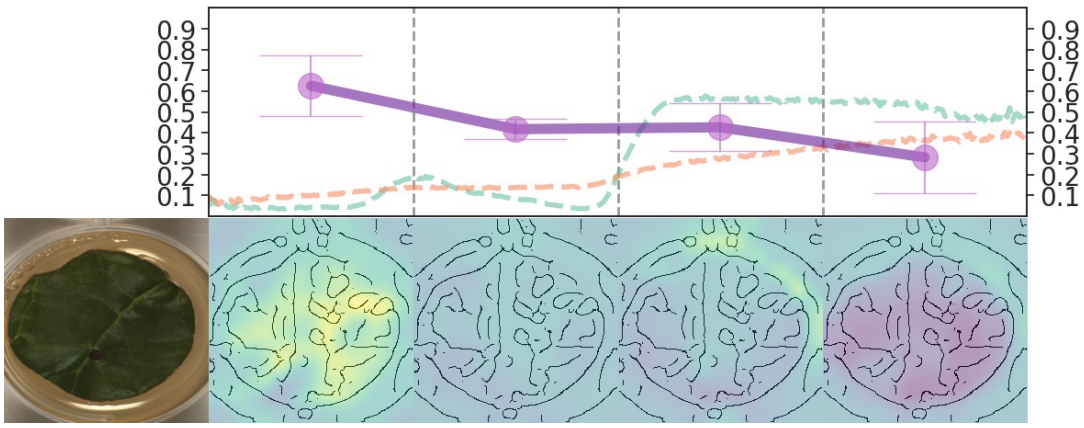

Figure 14: Appendix. Spatial and spectral activations from inoculated sample with single visible symptoms of not regularized network.

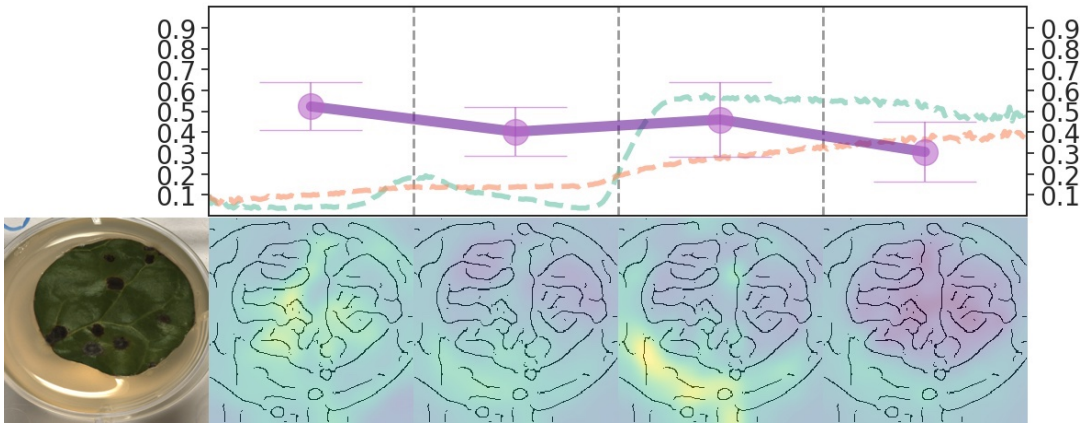

Figure 15: Appendix. Spatial and spectral activations from inoculated sample with multiple visible symptoms of not regularized network.

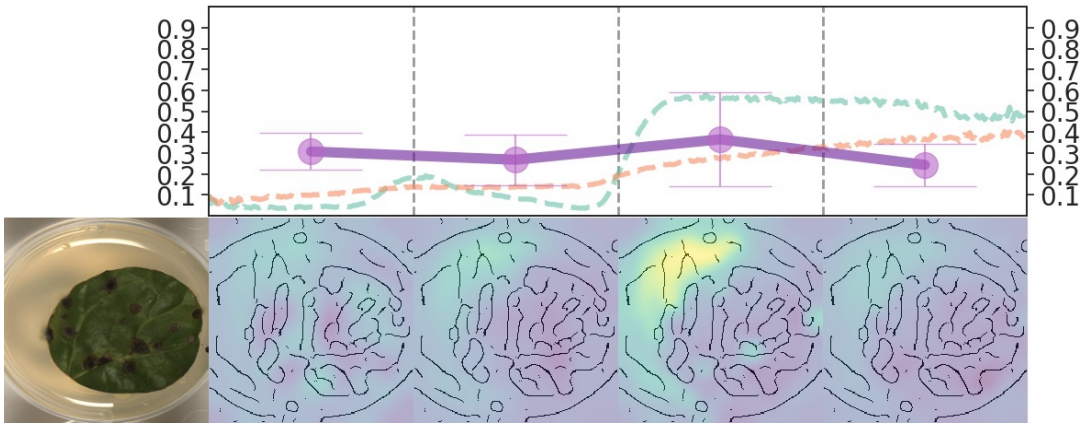

Figure 16: Appendix. Spatial and spectral activations from inoculated sample with multiple visible symptoms of not regularized network.

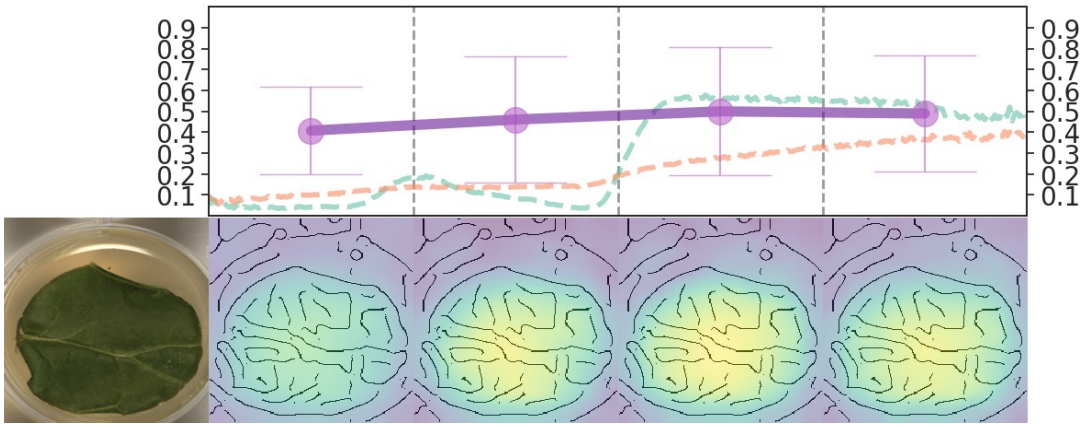

Figure 17: Appendix. Spatial and spectral activations from healthy sample of regularized network.

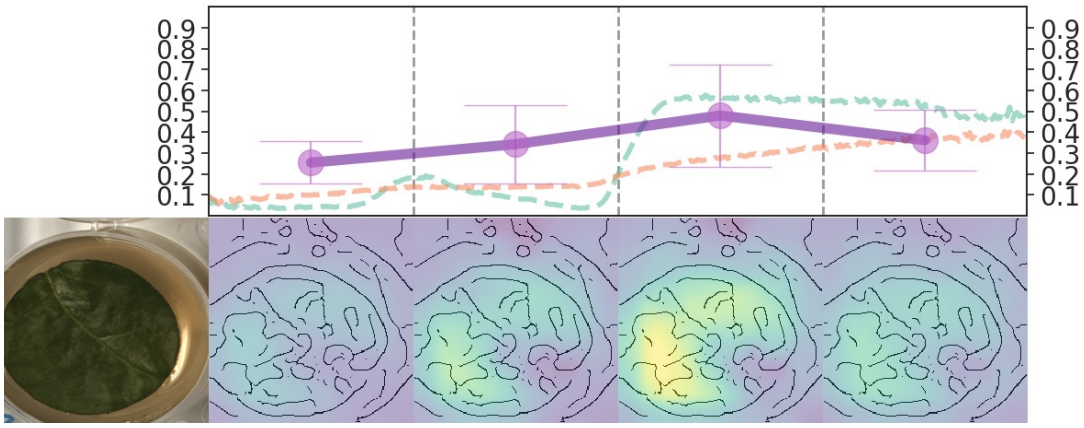

Figure 18: Appendix. Spatial and spectral activations from healthy sample of regularized network.

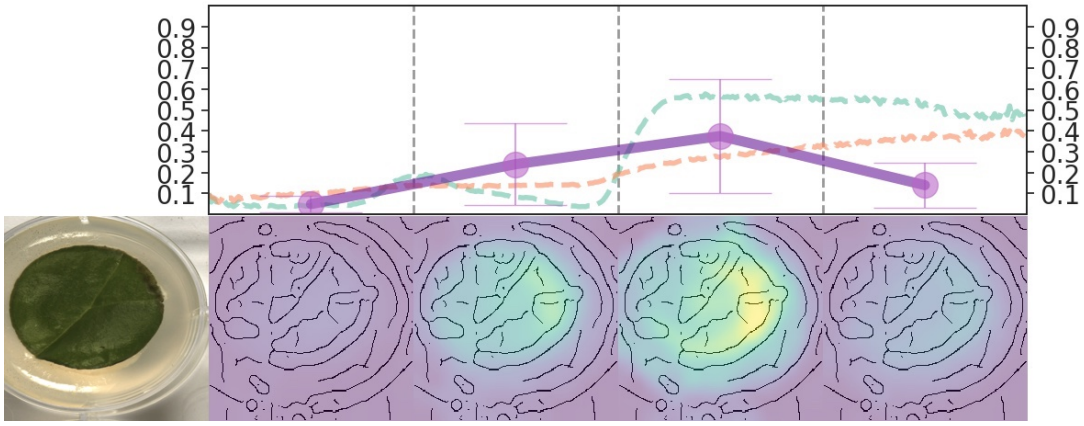

Figure 19: Appendix. Spatial and spectral activations from inoculated sample without visible symptoms of regularized network.

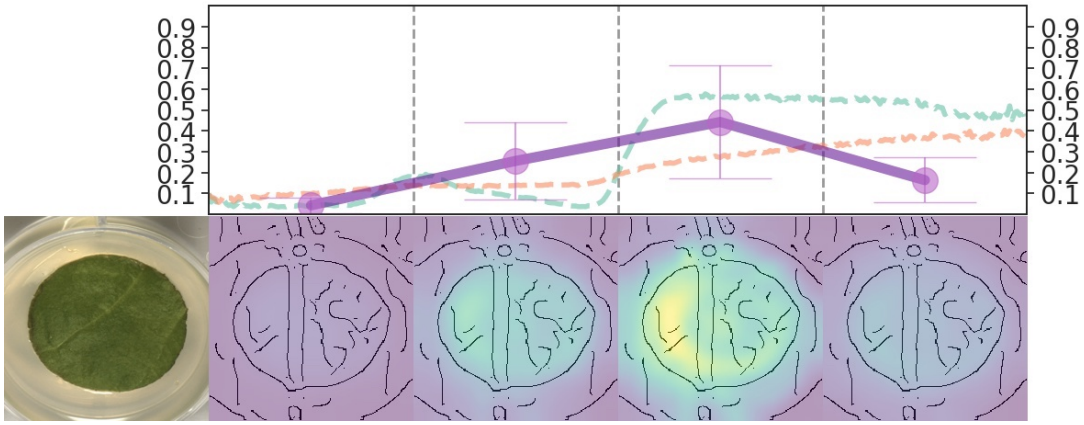

Figure 20: Appendix. Spatial and spectral activations from inoculated sample without visible symptoms of regularized network.

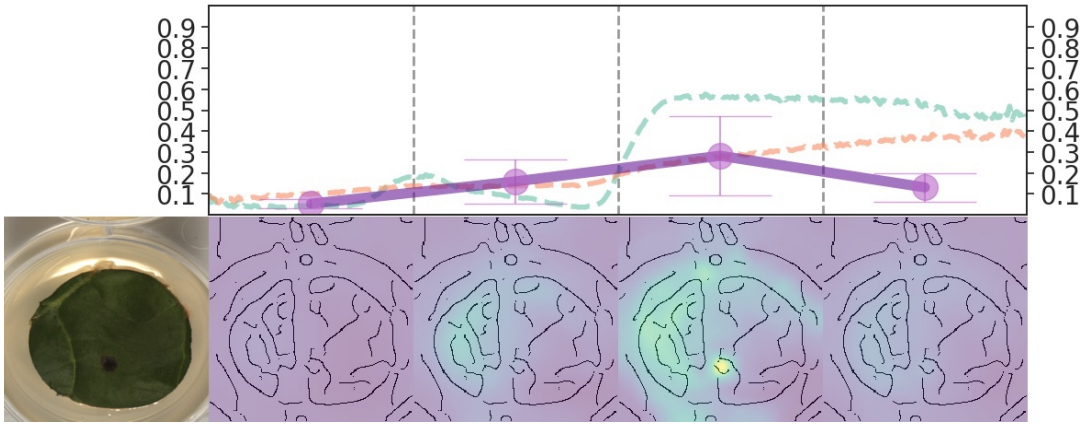

Figure 21: Appendix. Spatial and spectral activations from inoculated sample with single visible symptoms of regularized network.

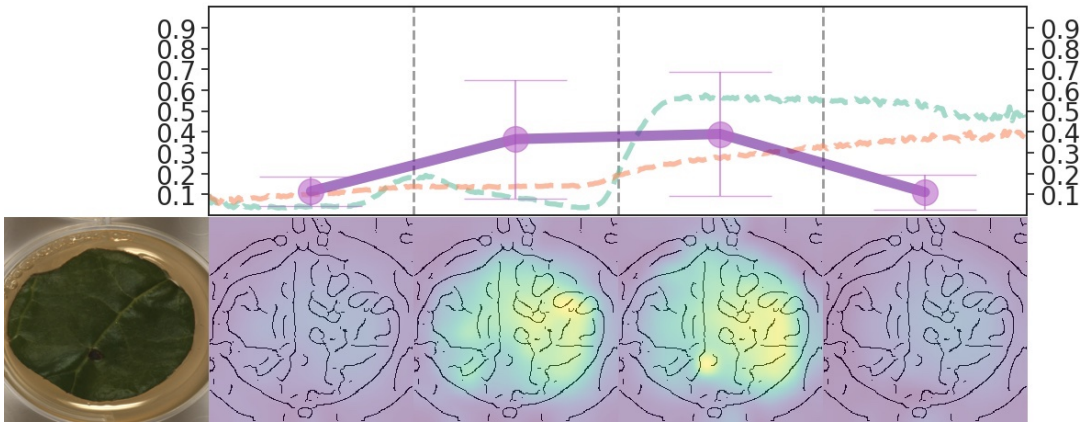

Figure 22: Appendix. Spatial and spectral activations from inoculated sample with single visible symptoms of regularized network.

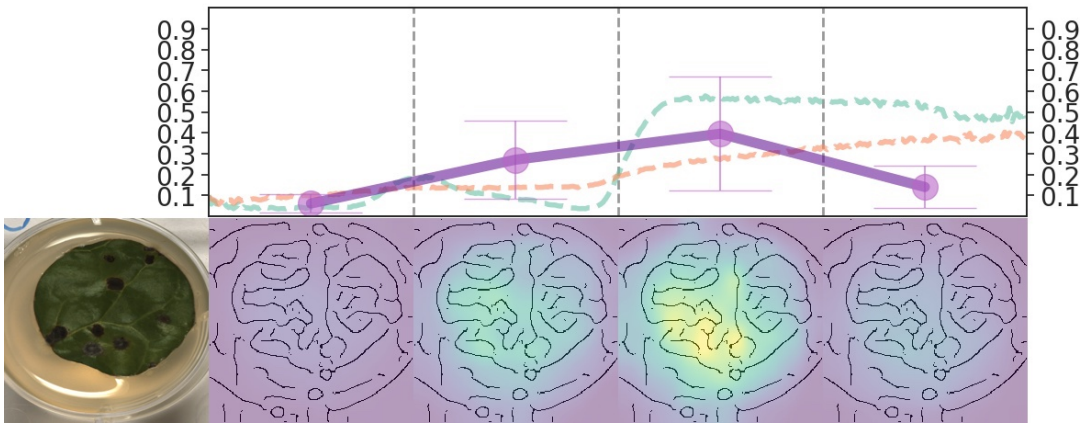

Figure 23: Appendix. Spatial and spectral activations from inoculated sample with multiple visible symptoms of regularized network.

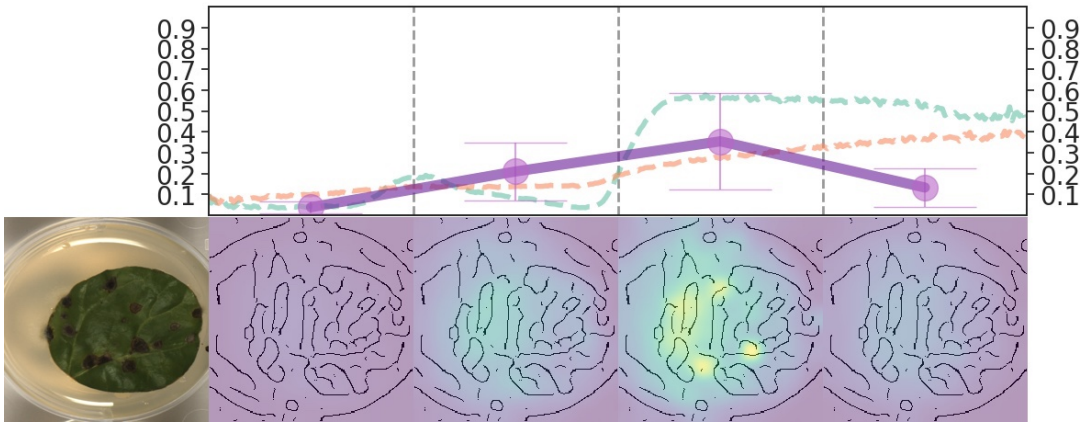

Figure 24: Appendix. Spatial and spectral activations from inoculated sample with multiple visible symptoms of regularized network.