Inductive Conformal Prediction for Harvest-Readiness Classification of Cauliflower Plants: A Comparative Study of Uncertainty Quantification Methods

Mohamed Farag[1], Jana Kierdorf[1], Ribana Roscher[2,1]

[1]*Remote Sensing Group, Institute of Geodesy and Geoinformation, University of Bonn*
[2]*Data Science for Crop Systems, Institute of Bio- and Geosciences, IBG-2: Plant Sciences, Forschungszentrum Jülich GmbH*

Abstract

Quantifying the uncertainty of machine learning models is a promising way to make better-informed decisions in digital agriculture. Efforts have been made to address this, ranging from understanding and segregating sources of uncertainty to utilizing diverse approaches for quantifying the cumulative amount. However, in order to fully realize the potential of uncertainty quantification in digital agriculture, more research is needed to compare and contrast different methods and determine which are most effective in different contexts. In this paper, we investigate inductive conformal prediction as another family of machine learning methods besides the commonly used softmax outputs and Monte Carlo dropout. Inductive conformal prediction constructs valid prediction sets by selecting a pre-defined level of predictive confidence in the system. In our experiments, we analyze this method for an image-based harvest-readiness classification task of cauliflower plants, and compare the results to softmax outputs and uncertainties derived from Monte Carlo dropout. Inductive conformal prediction turns out as a complementary tool offering distinct advantages and providing another level of information for decision support.

1. Introduction

Digital agriculture has gained popularity by utilizing data-driven models that rely on sensor observations and machine learning (ML)-based monitoring strategies. These models can improve current agricultural management or provide new insights into promising future management strategies. Deep learning (DL), in particular, has achieved significant success in analyzing vast amounts of complex data from various sensors, extracting meaningful patterns and representations [17]. This has enabled a wide range of applications, such as multi-model crop classification [19], generation and detection of plant growth stages [2, 18], and plant disease detection [3] [15].

The increased use of machine learning models in digital agriculture often comes with a need to assess the confidence in their predictions. For this, uncertainty quantification plays a crucial role in augmenting models by supplementary information about the outputs to increase confidence. Machine learning with supervision, in which labeled data is used to train a model for generating outputs on unseen data, is considered learning by induction. As such, models approximate the real world, creating a source of uncertainty, denoted as *epistemic* (model uncertainty). Additionally, noisy and imprecise data serve as another source of uncertainty, denoted as *aleatoric* (data uncertainty). In a nutshell, data uncertainty relates to the inherent randomness in the data-generating process, while model uncertainty represents the lack of knowledge about the best model. [10] offers a detailed explanation of uncertainty sources in supervised learning settings and provides a classification of various ML-based methods for dealing with uncertainty. However, it is essential to note that the definition of uncertainty, its causes, and the traditional separation of sources into the aforementioned types in ML remain somewhat ambiguous, as pointed out by [6]. They demonstrate the presence of additional sources of uncertainty, such as missing data and the deployment of ML-based approaches in a changing environment. [5] presents a comprehensive overview of uncertainty in DL, further pointing out that epistemic uncertainty is reducible, while aleatoric is generally irreducible. They demonstrate metrics to measure uncertainty and provide various approaches to address uncertainty in the context of real-world applications.

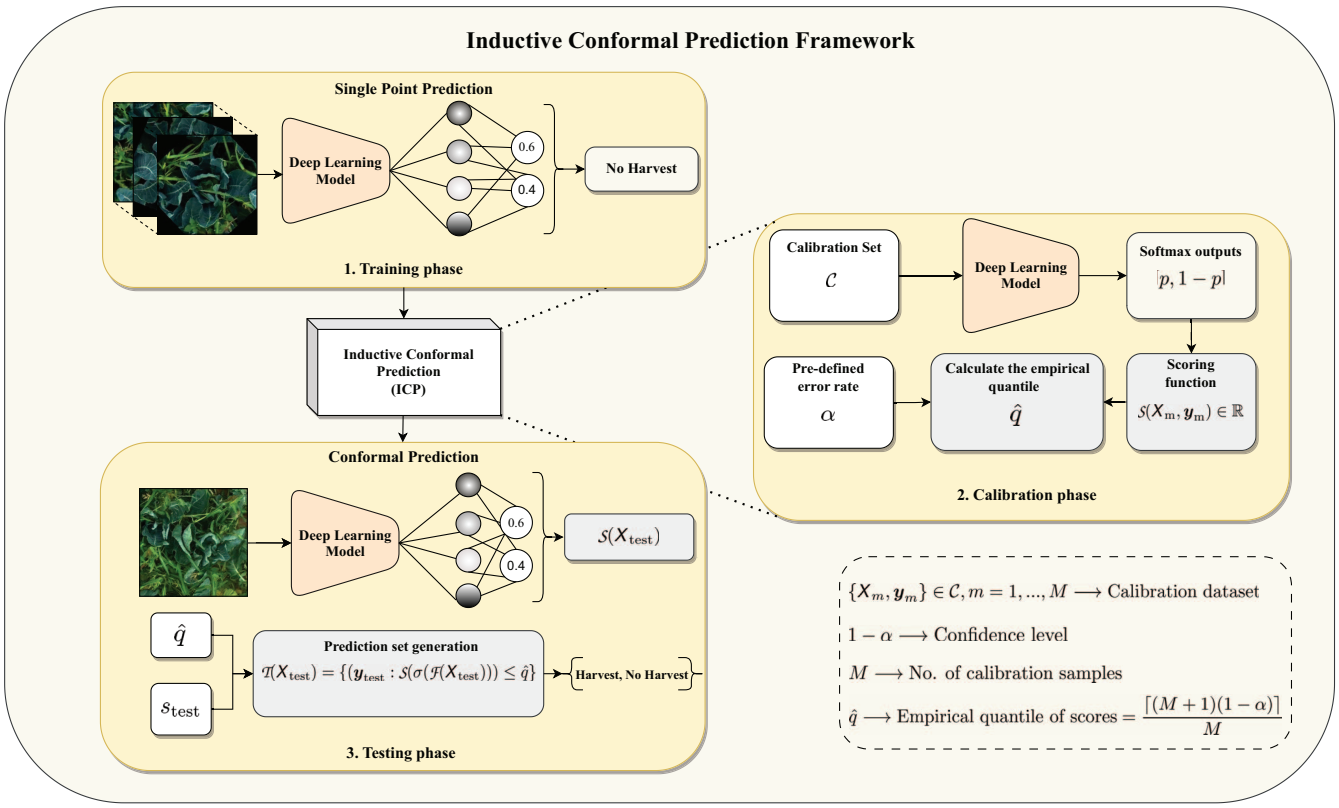

Figure 1: **The general framework of Inductive Conformal Prediction (ICP)**. After training the model, we initiate the calibration phase with new *i.i.d* samples. We choose a predefined confidence level of $1 - \alpha$ and a scoring function $\mathcal{S}(X)$ to measure conformity of future samples. Then, we calculate the empirical quantile $\hat{q}$ and compare the conformity score of the testing sample to the quantile during testing. This process generates a prediction set $\mathcal{T}_{\text{test}}(.)$.

Several techniques have been developed for estimating uncertainty and its types. Bayesian Neural Networks (BNN) are the most common method for estimating epistemic uncertainty [23]. The goal is to obtain the distribution of posterior weights and sample from it multiple times to obtain a probabilistic output. However, this method comes with high computational complexity. Monte Carlo Dropout (MC-Dropout) [4] is another widely adopted alternative for approximating BNNs. By applying random dropout masks during several forward runs, we can find the predictive posterior distribution. However, this method is relatively slow during inference. Ensemble learning [13] is another way to approximate BNNs [9]. It involves training models with different hyperparameters and datasets. However, the computational complexity of this method is generally high. Softmax outputs can be used to assess uncertainty quickly, but as stated in Section 2.3, it is prone to miscalibration, overconfidence, and the fact that it estimates only the aleatoric part of the uncertainty.

[10] explores set-valued prediction approaches, a further class of methods to estimate the uncertainty. One of the most prominent algorithms for this class is Conformal Prediction (CP) [20, 16, 14]. While it has had theoretical foundations since 2005, CP has recently gained attention in deep learning [1]. The fundamental concept of inductive conformal prediction (ICP) is to transform a single point prediction model into a set predictive algorithm that generates prediction sets for unseen samples. A prediction set is defined as a set of labels that guarantees to include the correct label with a pre-defined confidence level in a classification problem. In other words, the prediction set ensures coverage of the true reference value with an upper bound for errors will be created by the framework.

In this paper, we compare ICP, softmax outputs, and MC-Dropout, and discuss these approaches regarding their ability and efficacy to provide information about the prediction's uncertainty. Three experiments are conducted to highlight the characteristics of each method while putting a specific focus on ICP, a so far only rarely used method in digital agriculture. For our experiments, we use the GrowliFlower dataset [12], an image-based dataset for cauliflower harvest-readiness classification. Our main contributions are as follows:

1. We highlight and discuss the differences between ICP, softmax outputs, and MC-Dropout by evaluating the three methods on unseen data.

2. The interaction between the different components is highlighted showing the independence between ICP and softmax, in addition to analyzing ICP's sensitivity to the pre-defined confidence level.

3. We analyze the ability of each method to handle Out-of-Distribution (OOD) data in order to illustrate their level of informativeness.

4. We conduct a quantitative and qualitative comparison to analyze the usability of each method, and discuss their drawbacks, as well as the potential of hybrid tools.

2. Machine Learning Uncertainty Quantification Tools

In this section, we present the details of our employed methods. A dataset is defined as $\{\boldsymbol{X}_n, \boldsymbol{y}_n\} \in \mathcal{D}$, $n = 1, ..., N$, with $\boldsymbol{X}_n \in \mathbb{R}^{H \times W \times B}$ representing a B-dimensional image and $\boldsymbol{y}_n \in \mathbb{R} = [1, ..., C]$ denoting the class labels with C being the total number of classes.

2.1. Inductive Conformal Prediction (ICP)

One of the most commonly used types of conformal prediction is ICP, which is well-known for its computational efficiency. ICP transforms a conventional predictive algorithm into a so-called conformalized model by generating a prediction set that is guaranteed to include the correct label with a pre-defined confidence level for a new test sample $\boldsymbol{X}_{\text{test}}$ (see also Section 3.4). The dataset $\mathcal{D}$ is divided into four subsets: the training set $\mathcal{I}$, the validation set $\mathcal{G}$, the model calibration set $\mathcal{C}$, and the testing set $\mathcal{E}$. Although some works use identical sets $\mathcal{G}$ and $\mathcal{C}$, we use separate sets to avoid model exposure to data seen previously during validation in case of hyper-parameter tuning.

The ICP transformation framework goes as follows:

1. A proper notion of uncertainty such as softmax outputs in classification is selected; in the binary case, we have two classes represented by probability p for the positive class, and $1 - p$ for the negative class:

$$\sigma(\mathcal{F}(\boldsymbol{X}_n)) = [p_n, 1 - p_n], \tag{1}$$

   where $\sigma(\mathcal{F}(\boldsymbol{X}_n))$ is the probabilistic output for image $\boldsymbol{X}_n$ after feature extraction by the DL model $\mathcal{F}$.

2. We calibrate the model outputs using new unseen $i.i.d$ samples $\{\boldsymbol{X}_m, \boldsymbol{y}_m\} \in \mathcal{C}, m = 1, ..., M$ by getting the softmax outputs for each sample and applying the scoring function $\mathcal{S}(\boldsymbol{X}_m, \boldsymbol{y}_m)$ that needs to be selected as hyperparameter. The scoring function is defined in the range $[0, 1]$ as:

$$\mathcal{S}(\boldsymbol{X}_m, \boldsymbol{y}_m) = 1 - \sigma(\mathcal{F}(\boldsymbol{X}_m))_y \in \mathbb{R}\,. \tag{2}$$

   Where $\sigma(\mathcal{F}(\boldsymbol{X}_m))_y$ is the probability component corresponding to the true label. A suitable choice is important and careful design characteristics such as ranking the prediction errors when the model is applied to inputs should be considered [1]. The so-called conformal scores represent the predictive uncertainty, where a large score means less confidence because the original softmax value is low and vice versa.

3. The empirical quantile of the conformal scores, denoted as $\hat{q}$, is calculated based on a pre-defined confidence level $1 - \alpha$ and the number of calibration samples M:

$$\hat{q} = \frac{\lceil (M+1)(1-\alpha) \rceil}{M} \tag{3}$$

4. Finally, for a new test sample $\boldsymbol{X}_{\text{test}}$, we apply the scoring function as a measure of the similarity, typicalness, or conformality to the calibration subset. We include the labels $\boldsymbol{y}_{\text{test}}$ that fall below $\hat{q}$ to generate the prediction sets $\mathcal{T}$:

$$\mathcal{T}(\boldsymbol{X}_{\text{test}}) = \{(\boldsymbol{y}_{\text{test}} : \mathcal{S}(\sigma(\mathcal{F}(\boldsymbol{X}_{\text{test}}))) \leq \hat{q}\}. \tag{4}$$

5. The prediction set $\mathcal{T}$ is guaranteed to encompass the true label y by following the probability constraints below:

$$1 - \alpha \leq \mathrm{P}(\boldsymbol{y}_{\text{test}} \in \mathcal{T}(\boldsymbol{X}_{\text{test}})) \leq 1 - \alpha + \frac{1}{M+1}\,. \tag{5}$$

   The probability highlighted in Eq 5 indicates that there is a chance of obtaining a set that includes the correct label, with a probability of $1 - \alpha$. However, it is not possible to ensure adaptivity for every sample. known as conditional coverage, which is a stronger property compared to the marginal coverage in Eq 5, defined as:

$$\mathrm{P}(y_{\text{test}} \in \mathcal{T}(\boldsymbol{X}_{\text{test}}) | \boldsymbol{X}_{\text{test}}) \geq 1 - \alpha \tag{6}$$

   In most cases, marginal coverage in Eq 5 is achieved [1], while Eq 6 can only be approximated [21].

2.2. Monte Carlo Dropout (MC-Dropout)

Dropout is a regularization approach utilized in DL to reduce over-fitting [8]. During training at each forward pass j, dropout randomly deactivates some of the neurons controlled by a hyper-parameter w_{drop} which represents the

fraction of neurons to drop. During inference, the dropout layer is switched off to avoid getting stochastic outputs.

MC-Dropout exploits the previous idea to estimate the epistemic uncertainty by allowing the model to get a stochastic output during multiple forward runs $j = 1, ..., J$ at the inference phase. The mean prediction $\mathcal{U}(\boldsymbol{X}_{\text{test}})$ for a single test sample $\boldsymbol{X}_{\text{test}}$ is obtained as

$$\mathcal{U}(\boldsymbol{X}_{\text{test}}) = \frac{1}{J} \sum_{j=1}^{J} \mathcal{F}(\boldsymbol{X}_{\text{test}})_j. \quad (7)$$

Furthermore, the standard deviation $\mathcal{S}_{\text{DO}}(\boldsymbol{X}_{\text{test}})$ can be analyzed to check the distribution of the predictions

$$\mathcal{S}_{\text{DO}}(\boldsymbol{X}_{\text{test}}) = \sqrt{\frac{1}{J} \sum_{j=1}^{J} (\mathcal{F}(\boldsymbol{X}_{\text{test}})_j - \mathcal{U}(\boldsymbol{X}_{\text{test}}))^2}, \quad (8)$$

while a wide distribution reveals uncertainty and more precise distributions induce model confidence.

2.3. Softmax Outputs

The most commonly used method for assessing uncertainty is through the probabilistic output of the softmax function. However, as pointed out by [7], this measure can be misleading due to miscalibration. It occurs when the predicted probabilities do not align with the true likelihoods. In addition, softmax outputs often exhibit overconfidence [13], assigning high probabilities to incoming samples even if they are misclassified. This makes it challenging to detect and accurately quantify uncertainty. Furthermore, raw softmax outputs are unable to capture epistemic uncertainty [4], which relates to the model's lack of knowledge or limited data.

3. Experiments and Results

3.1. Dataset

GrowliFlower is an open-source dataset acquired in 2020 and 2021 [12]. It contains 14K samples of cauliflower plants with multiple modalities available, including RGB and multi-spectral images. The dataset also includes a comprehensive range of annotations, making it useful for computer vision tasks such as classification and segmentation. This paper considers harvest readiness as a binary classification problem, where class 0 indicates non-readiness for harvesting. Four time steps are obtained to form the training, validation, and testing subsets. The model is trained on 6224 samples, using 196 samples as a validation set $\mathcal{G}$ and 194 samples for the testing set $\mathcal{E}$, with an image size of $(256 \times 256 \times 3)$. Random rotation is applied as an augmentation technique to enrich the variability of the data and to avoid over-fitting.

3.2. Settings

3.2.1 Model and Training Settings

The model consists of three blocks, each with output channels set to 64, 128, and 256, respectively. Each block contains three convolutional layers with a kernel size of $(3, 3)$, and padding is used to maintain the same dimensions. We apply the ReLU function as a non-linear transformation after each convolutional layer, followed by a max pooling layer. Additionally, we add Batch Normalization (BN) [11] after the 64 and 128 convolutional blocks. Dropout layers, set to a probability of $w_{\text{drop}} = 0.1$, are added before each convolutional layer to prepare the model for MC-Dropout runs. A classification head is used, which includes a final dropout layer with a probability $w_{\text{drop}} = 0.25$ of randomly deactivating a neuron. Finally, a softmax layer is applied.

For training, we use 500 epochs and early stopping. Adam optimizer is used with a learning rate of 0.001. To address class imbalance, the Weighted Cross Entropy Loss (WCLE) to penalize the model more when predicting the minor class and vice versa is utilized, and NVIDIA RTX A4500 GPU is our hardware accelerator.

3.2.2 Uncertainty Estimation Tools Settings

In our model, we use ICP as a post-hoc technique. The size of the calibration set is an important factor that affects performance. While [1] suggests using 1000 samples, [22] recommends using $(10/\alpha)$ samples to ensure good performance, where α represents the desired error percentage in the system.

To evaluate the performance of ICP, we first split the original validation set $\mathcal{G}$ into two subsets $\mathcal{A}$ and $\mathcal{B}$ where $\mathcal{G} = \mathcal{A} \cup \mathcal{B}$. $\mathcal{A}$ allocates 100 samples for calibration and 96 samples to evaluate ICP's performance are included in $\mathcal{B}$. Because of the lack of data, the entire validation set $\mathcal{G}$ that encompasses 196 samples is utilized as a final calibration set $\mathcal{C}$ and we evaluate the approach on the entire testing set $\mathcal{E}$ which has 194 samples. The confidence level $1 - \alpha$ is selected to be 80%, and the scoring function is the same as mentioned in Section 2.1.

We conduct 1000 forward runs for MC-Dropout, keeping all architecture layers frozen except for the dropout layers. Finally, we set the threshold γ at which a sample is classified as class 1 to be 0.5 if the probability p is greater than it.

3.3. Experiment 1: Quantitative Evaluation

3.3.1 Results

Our aim is to evaluate the informativeness of our approaches based on the generated predictions. The model achieves a testing accuracy of 66.5%.

-	Singleton sets $\{0\}, \{1\}$	Double sets $\{0, 1\}$
Total	160	34
Correct	112	17
Incorrect	48 ($y \neq \hat{y}$) and ($y \notin \mathcal{T}$)	17 ($y \neq \hat{y}$)

Table 1: **ICP outcomes evaluated on the test set $\mathcal{E}$.**

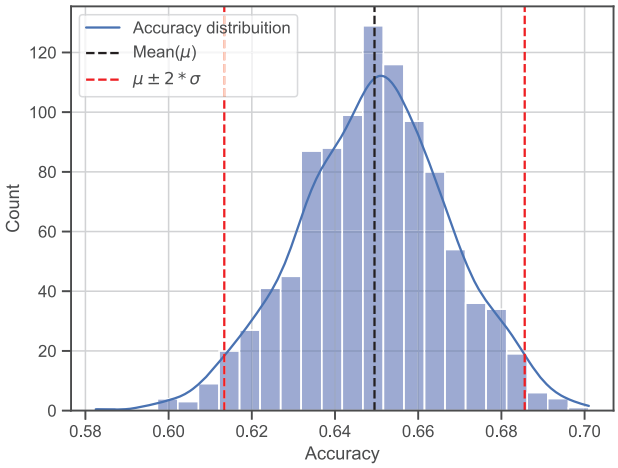

Figure 2: **Accuracy distribution after 1000 MC-Dropout runs applied to the testing set $\mathcal{E}$.** The distribution demonstrates uncertainty present in the model as it is wide, which could be interpreted as the model's lack of knowledge.

Prediction sets are generated based on a predefined confidence level of $1 - \alpha = 80\%$. Four different outputs are expected: $\{\phi\}, \{0\}, \{1\}$, and $\{0, 1\}$. Singleton sets indicate confidence, while a double-element set implies uncertainty and difficulty of the sample. The empty sets will be explained later in Section 3.4.

As illustrated in Table 1, there are 160 singleton sets and 34 double-element sets obtained. Consequently, the DL system exhibits caution with respect to 34 samples, requiring further assistance to determine the correct label. Among the singleton sets, 112 includes the correct label, while 48 reveals misclassifications and ICP fails to generate a prediction set that includes the correct label.

Furthermore, out of the 34 double-element sets, 17 are correctly classified by comparing the original output label before applying ICP to the reference value. However, the ICP framework remains uncertain, as indicated by the inclusion of both labels in the prediction set. This supplementary information emphasizes on the need for external supervision for the difficult samples where ICP hesitates.

For MC-Dropout, 1000 stochastic samples are acquired. The resulting average testing accuracy μ is 65%, and the standard deviation σ is 0.0180, as shown in Figure 2.

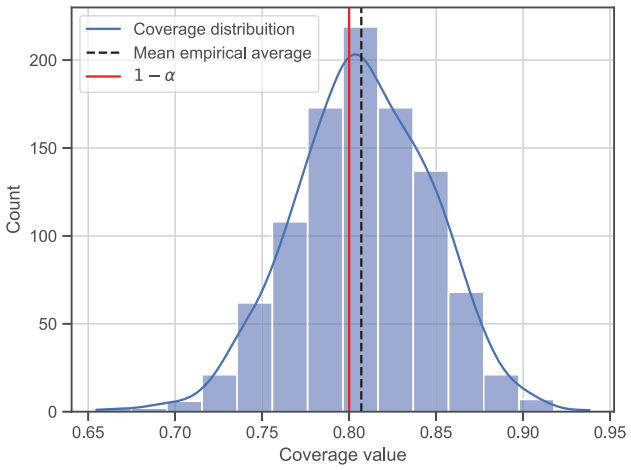

Figure 3: **ICP's empirical coverage distribution**. ICP operates properly according to [1] as for 1000 random splits between calibration and testing set, the mean empirical coverage is 0.806, and the theoretical limit $1 - \alpha$ is 0.8. Moreover, the distribution is almost centered around $1 - \alpha$.

A flatter distribution indicates greater uncertainty about the model's overall performance on the testing dataset. MC-Dropout augments predictions with an estimate of the model's knowledge, which can be used to make better decisions about the model's capacity for the given task.

3.3.2 Discussion

Softmax probabilities are a fast way to be used as an estimate for uncertainty, but have known issues as mentioned in Section 2.3. ICP provides more information about the same outputs in addition to the ability to assess the total predictive uncertainty. The prediction set itself and its size are measures for the uncertainty, while the type of the set indicates if the system is confident or not. The length $|\mathcal{T}(.)|$ is equivalent to adaptivity which is governed by Eq 6, where an increased length signifies sample difficulty revealing uncertainty situation.

In addition, as stated in [22], ICP is considered "always *valid*" as the frequency of errors occurs at a rate no higher than α at a confidence level of $1 - \alpha$ on the long run. When a ML model is well-calibrated, the probabilities it generates align with the actual probabilities or likelihood of the predicted event. Conservative validity is one of two validity types that explains a part of the calibration term in CP. This concept is similar to calibration because the measured probability of errors (i.e., miscoverage of the true class) for successive generations of prediction sets is roughly equal to the theoretical error α (i.e., true likelihood).

The 48 singleton sets that do not include the correct label

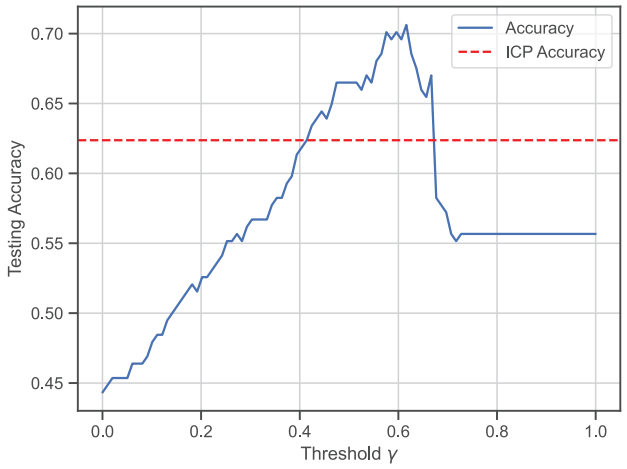

Figure 4: **Sensitivity analysis of ICP to softmax threshold** γ. The sensitivity of softmax output is observed for the testing accuracy when changing the threshold, ICP in contrast offers a consistent performance.

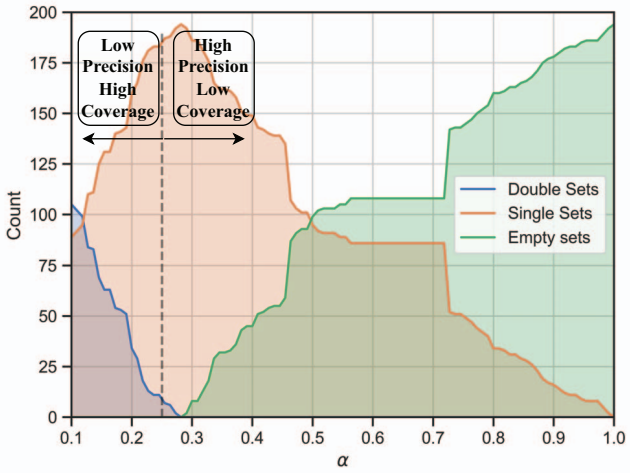

Figure 5: **Sensitivity analysis of ICP with respect to the error rate** α. A lower error rate leads to an increase in double sets as ICP becomes more cautious to the left of the vertical line. In the region to the right, single sets increase, however, the model also has a higher rate of not including the true class.

elaborate and emphasize the previous idea, as for one run only the error is $\frac{48}{194} = 24.7\%$ and the theoretical limit α is 20%. By applying the law of large numbers, 1000 runs of splitting the calibration and testing sets randomly generate an empirical mean coverage of 80.6% as demonstrated in Figure 3, which shows that the long frequency of true labels miscoverage is around 20%.

3.4. Experiment 2: The Effect of Hyperparameters and Mutual Influence

We begin by observing the effect of changing the threshold γ of the softmax as a hyperparameter and record ICP's sensitivity, as demonstrated in Figure 4. ICP offers an overall accuracy of 66.5%, which is independent of the threshold and signifies its consistent performance. ICP uses the scoring function and calibration dataset to mitigate the aforementioned deficiencies of softmax [1].

The reasons for generating empty sets are investigated by varying α and observing the changes in the total number of each set type as shown in Figure 5. To begin, we first explain the concept behind ICP. Being based on typicalness or similarity, ICP assigns the possible labels that would make a new unseen sample X_{test} conforms to a bag of samples or calibration dataset $\mathcal{C}$. At the start, the calculated conformal scores represent uncertainty, where higher scores show less confidence and vice versa.

For a given number of samples M, we calculate the empirical quantile $\hat{q}$ based on the pre-defined α. The quantile splits the range of the scoring function into two parts: the conformal region, where the scores are low and the model is certain, and the non-conformal region, where the scores are high and the model is uncertain. When $\alpha = 0.2$, this means that at least 80% of softmax outputs for the reference values (the true labels that should be predicted) fall below $\hat{q}$. In contrast, for the remaining 20%, ICP is unsure about them because the scores are too high, implying high uncertainty, as the model knows that there are 20% of the scores even if they are for the reference values that should be predicted, yet their values are too high and above the quantile.

For a new test sample, the framework assigns any label that makes the new sample conforms to the calibration samples after checking the scores produced. If a score is higher than the quantile, ICP's process this, first as the score is high, showing uncertainty, and second, being higher than the quantile, forcing ICP to reject assigning this label to the sample. The non-conformality is high and it would check the second score if it allows for assigning a label that would make the sample conforms to the calibration set.

Increasing α reduces the quantile value, meaning that more calibration samples are added to the non-conformal region (the percentage of the softmax values related to the true class is reduced below the quantile) where the framework is uncertain, which allows the model to deal with a new sample that has both scores higher than the quantile like an anomaly and produces an empty set $\{\phi\}$. The higher the value of α, the more similar the new test sample needs to be compared to the calibration set in order to have a non-empty prediction set (to be conformal), which explains the

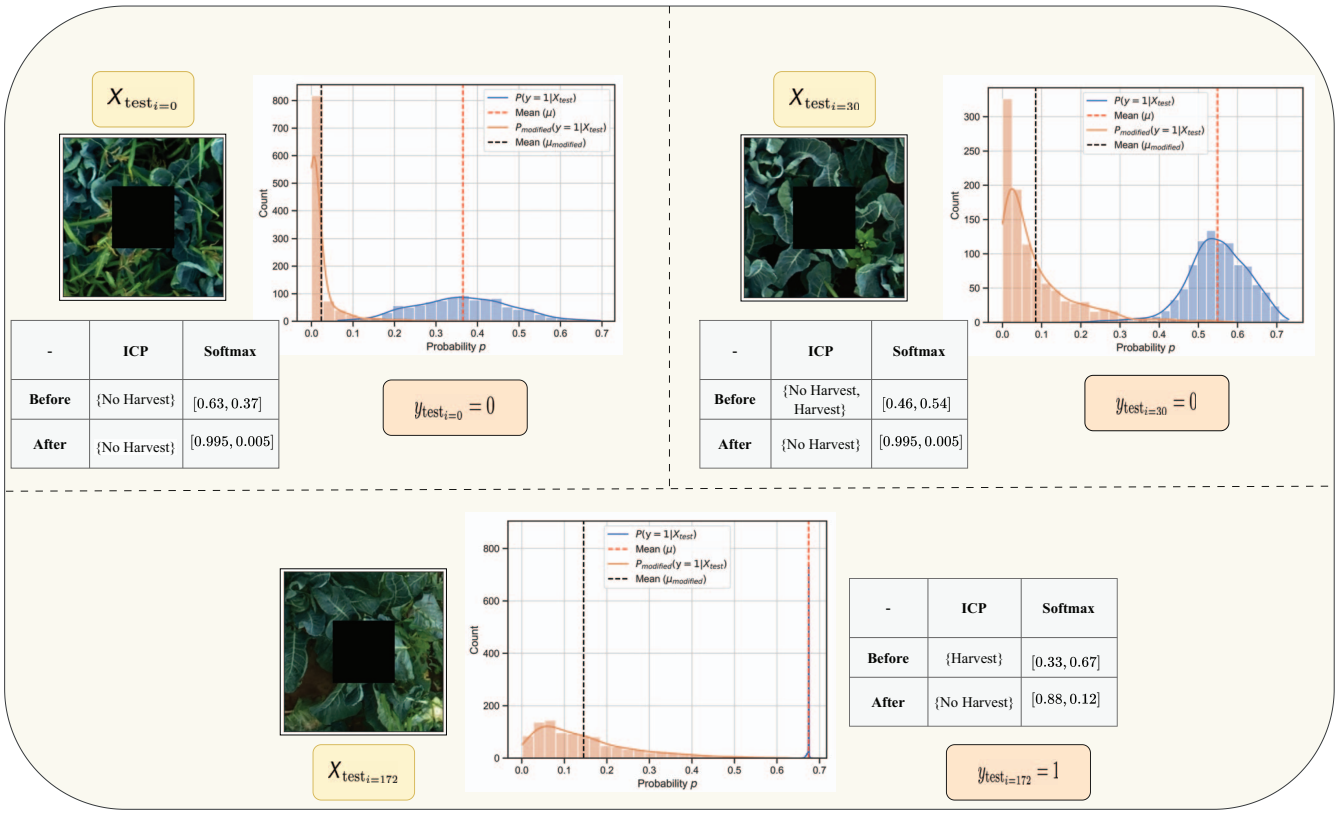

Figure 6: **A comparison of the outputs of each framework when applied to OOD**. At each partition, we have a sample modified by adding a black box at the center of the image. The blue distribution represents the conditional distribution $P(y = 1|X_{\text{test}})$ which shows the probability of the sample to be predicted as class 1, and the orange distribution $P_{\text{modified}}(y = 1|X_{\text{test}})$ illustrates the same distribution after applying the modification to the image. While y_{test_i} is the reference value for sample i.

significant increase in the number of empty sets observed when increasing α.

3.5. Experiment 3: Out of Distribution data (OOD)

We can evaluate the effectiveness and limitations of our methods by testing them utilizing OOD samples. To analyze the role of the cauliflower's head in the final decision, we overlay a black box at the center of the images. Three images are chosen from the testing set (indices $i = 0$, 30, and 172) to showcase the different information that can be provided by each framework for various scenarios.

As shown in Figure 6, the first sample has a reference value of 0. Before image modification, the softmax output generates $[0.63, 0.37]$, while the model produces $[0.995, 0.005]$ after modification. This severe change in softmax probabilities emphasizes the importance of the plant's head in guiding the decision, yet we are not heavily dependent on these values.

By comparing the conditional distribution $P(y = 1|X_{\text{test}})$ to the distribution $P_{\text{modified}}(y = 1|X_{\text{test}})$, MC-Dropout provides another piece of information through sampling 1000 times. Due to the elimination of the features of the plant's head, the model produces a more confident prediction, as shown by the orange histogram, in contrast to the original one where the model is less confident. Finally, ICP produces a correct prediction sets at both cases.

At index $i = 30$, the reference value is 0, but the model initially predicted 1 based on the softmax outputs. Although obtaining a confident prediction seems challenging, the model's misclassification of the sample is unavoidable. MC-Dropout improves the information by introducing flat distributions that display high output variability as a sign for a problem. As a tool for total predictive uncertainty, ICP is cautious and includes both labels requesting additional assistance. Blocking the center of the image shows enhanced model confidence from softmax perspective, indicating that the image belongs to class 0. However, the distribution provided by MC-Dropout is skewed and not centered around the mean, which again highlights an issue related to this case. Finally, the prediction set generated by ICP includes the true label, which shows limited information provided compared to the MC-Dropout approach.

For the last image, the model's prediction aligns with the reference value. Softmax produces $[0.33, 0.67]$, while MC-

Dropout reinforces the decision by presenting a very sharp distribution centered around the mean. This highlights the model's confidence in the prediction, and ICP includes the correct class in the prediction set. The result from MC-Dropout may provide insights into the model's knowledge regarding this specific sample, as it indicates that further investigation is needed since the model is too confident compared to other cases. Image modification altered the softmax values and ICP set in favor of class 0. However, the outcome from MC-Dropout demonstrates a distribution that is not centered around the mean, being flat and skewed.

3.5.1 Discussion

In this experiment, MC-Dropout and ICP provide different types of information. ICP provides prediction sets that guarantee coverage of the true label, serving as a means to assess total predictive uncertainty. On the other hand, MC-Dropout generates distributions that convey information about the model's knowledge. When comparing usability, ICP is better suited as a front-end tool for ensuring that the model makes predictions, with less emphasis on understanding the sources and measuring the contribution of each to the predictive uncertainty. In contrast, MC-Dropout can be seen as a back-end method for checking the model's knowledge about different concepts and comparing models based on their confidence levels.

ICP is a tool that deals with uncertainty in a distinct way from MC-Dropout [10]. It sets up predetermined confidence levels that we require at the system and creates prediction sets based on these levels. ICP is agnostic to the model and the data, and it is valid when dealing with limited samples [1]. Furthermore, it has a low computational overhead. In contrast, MC-Dropout can be viewed as an ensemble learning or a BNN approximation that estimates epistemic uncertainty using the generated distribution of outputs.

The ICP method has a limitation in that it deals only with total predictive uncertainty, which may not effectively reveal the sources present in the system. Its performance is influenced by both the calibration dataset and the scoring function, making the appropriateness of the scoring function questionable. Moreover, the quality and size of the calibration dataset could negatively impact its performance.

On the other hand, the MC-Dropout method concentrates on epistemic uncertainty only, and has three factors that can affect its performance: the dropout probability w_{drop}, the layer position in the architecture, and the number of samples taken. For highly parametric networks, the number of samples can influence the estimate and slow down inference. Hence, our objective is to merge different approaches and create hybrid solutions that can overcome the limitations inherited by each method.

4. Conclusion

This paper presents a comparison and discussion of three approaches to uncertainty estimation for an image-based prediction task of cauliflower harvest-readiness, both qualitatively and quantitatively. The three approaches are inductive conformal prediction, Monte Carlo dropout, and softmax raw outputs, which are evaluated based on their effectiveness in estimating predictive uncertainty.

ICP is a promising technique in the cluster of set prediction approaches. It provides valuable insights into the confidence of predictions by constructing prediction sets with a predefined confidence level. ICP is a robust choice in complex agricultural scenarios where the goal is to force the model to generate a prediction, while the separation and estimation of uncertainty sources are less important. Its ability to offer validity, adaptivity, and capture total predictive uncertainty is noteworthy.

On the other hand, MC-Dropout has demonstrated its complementary nature by offering a computationally efficient approach to estimating epistemic uncertainty or the model's lack of knowledge, as an approximation for BNNs. MC Dropout can be used in contexts where the main goal is to reduce uncertainty present in the system, such as in Active Learning (AL) or investigating the concepts learned by the model. In contrast, the softmax raw outputs exhibit limitations in uncertainty estimation. Although they provide a fast estimate of uncertainty, the method captures only aleatoric uncertainty, potentially leading to misinterpretations of model confidence.

Overall, the comparative analysis sheds light on the strengths and weaknesses of each method. It highlights the significance of accurate uncertainty estimation in digital agriculture, where critical decisions rely on confident predictions. The findings emphasize the value of using both ICP and MC-Dropout as they offer complementary insights into uncertainty and model performance.

We recommend exploring hybrid approaches that combine the strengths of different methods to further improve decision support systems for digital agriculture. It's also crucial to fine-tune these methods for specific agricultural tasks and datasets to ensure optimal performance and practical usability in the real world.

5. Acknowledgment

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2070 – 390732324 and DFG project AID4Crops (RO 4839/6-1).

References

[1] Anastasios Nikolas Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free

uncertainty quantification. *ArXiv*, abs/2107.07511, 2021.

[2] Lukas Drees, Immanuel Weber, Marc Rußwurm, and Ribana Roscher. Time dependent image generation of plants from incomplete sequences with cnn-transformer. In *DAGM German Conference on Pattern Recognition*, pages 495–510. Springer, 2022.

[3] Konstantinos P. Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018.

[4] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *ArXiv*, abs/1506.02142, 2015.

[5] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseo Lee, Matthias Humt, Jianxiang Feng, Anna M. Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, M. Shahzad, Wen Yang, Richard Bamler, and Xiaoxiang Zhu. A survey of uncertainty in deep neural networks. *ArXiv*, abs/2107.03342, 2021.

[6] Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. Sources of uncertainty in machine learning - a statisticians' view. *ArXiv*, abs/2305.16703, 2023.

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017.

[8] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[9] Lara Hoffmann and Clemens Elster. Deep ensembles from a bayesian perspective. *CoRR*, abs/2105.13283, 2021.

[10] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110:457 – 506, 2019.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456, Lille, France, 2015. JMLR.org.

[12] Jana Kierdorf, Laura Verena Junker-Frohn, Mike Delaney, Mariele Donoso Olave, Andreas Burkart, Hannah Jaenicke, Onno Muller, Uwe Rascher, and Ribana Roscher. Growliflower: An image time-series dataset for growth analysis of cauliflower. *Journal of Field Robotics*, 40(2):173–192, 2023.

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc.

[14] Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 76(1):71–96, 2014.

[15] Bipul Neupane, Teerayut Horanont, and Nguyen Duy Hung. Deep learning based banana plant detection and counting using high-resolution red-green-blue (rgb) images collected from unmanned aerial vehicle (uav). *PLoS ONE*, 14, 2019.

[16] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Machine Learning: ECML 2002*, pages 345–356, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[17] Michael P Pound, Jonathan A Atkinson, Alexandra J Townsend, Michael H Wilson, Marcus Griffiths, Aaron S Jackson, Adrian Bulat, Georgios Tzimiropoulos, Darren M Wells, Erik H Murchie, et al. Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *Gigascience*, 6(10):gix083, 2017.

[18] Longzhe Quan, Huaiqu Feng, Yingjie Lv, Qi Wang, Chuanbin Zhang, Jingguo Liu, and Zongyang Yuan. Maize seedling detection under different growth stages and complex field environments based on an improved faster r–cnn. *Biosystems Engineering*, 184:1–23, 2019.

[19] Igor Teixeira, Raul Morais, Joaquim J. Sousa, and António Cunha. Deep learning models for the classification of crops in aerial imagery: A review. *Agriculture*, 13(5), 2023.

[20] Glenn Shafer Vladimir Vovk, Alexander Gammerman. *Algorithmic Learning in a Random World*. Springer New York, NY, 1 edition, Dec. 2005.

[21] Vladimir Vovk. Conditional validity of inductive conformal predictors. *CoRR*, abs/1209.2673, 2012.

[22] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Mathematics and Statistics. Springer, Switzerland, 2nd edition, 2022. Hardcover ISBN: 978-3-031-06648-1. Published: 14 December 2022. Softcover ISBN: 978-3-031-06651-1. Due: 28 December 2023. eBook ISBN: 978-3-031-06649-8. Published: 13 December 2022.

[23] Hao Wang and Dit-Yan Yeung. A survey on bayesian deep learning. *ACM Comput. Surv.*, 53(5), sep 2020.