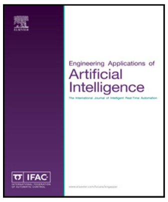
Engineering Applications of Artificial Intelligence 161 (2025) 111961

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

# Explainable automated wild-orchid identification combining deep neural networks and Bayesian networks

Diah Harnoni Apriyanti [a,b,*], Luuk J. Spreeuwers [a], Peter J.F. Lucas [a]

[a] Faculty of EEMCS, University of Twente, Drienerlolaan 5, Enschede, 7522NB, Overijssel, The Netherlands
[b] National Research and Innovation Agency (BRIN), Jl. M.H. Thamrin, Jakarta, 10340, DKI Jakarta, Indonesia

ARTICLE INFO

*Keywords:*
Explainable artificial intelligence
Bayesian networks
Deep neural networks
Machine learning
Orchid flower identification

ABSTRACT

Deep learning has been shown repeatedly to be a successful method of obtaining accurate classifiers. This also applies to orchid identification from digital photographs. However, deep neural networks possess the major weakness of lack of explainability, missing the ability to explain the reasons behind a decision. Nevertheless, most current research regarding automated orchid identification applies this blackbox approach. By contrast, in this paper we propose a new method for trustworthy automated orchid identification combining two complementary methods: deep neural networks and feature-based Bayesian networks, where the Bayesian network is also utilized for providing an explanation of the generated solutions. We use other deep neural networks to extract flower characteristics, the features, from the images which are subsequently fed into the Bayesian network as uncertain evidence. When combining the deep neural network and the Bayesian network as an ensemble classifier, both reaching the same conclusion, an accuracy of 89.4% is achieved, the most trustworthy outcome. With a human-in-the-loop ensemble classifier, validation results are even better, yielding an accuracy of 98.1%. Our approach also exploits the taxonomic knowledge represented in the Bayesian network to provide an explanation of the solutions for every case, reinforcing further trust in the method. The result is an explainable user-in-the-loop ensemble classifier. Providing explainability can help build user trust in a system and may play a major role when it is used as a learning aid for new orchid enthusiasts. Finally, the proposed method may be also of value in many fields other than plant determination.

## 1. Introduction

Plant determination has a very long history, with a major role in its development by the famous physician Carl Linnaeus, who firmly established the systematic method of plant categorization based on observed characteristic features (Linnaeus, 1735). Manual plant identification involves examining physical plant characteristics, such as the number of leaves, their shape, texture, and color, whether or not it bears flowers, or fruit, and if it does, how the flowers or fruits look, whether it produces seeds, the number and shape of seeds, etc., resulting in a plant's name. Any plant has a **binomial (two-part) name**, after Linnaeus, consisting of a **genus** name, describing a group of similar plants, and an epithet used to characterize a specific plant, usually called a **species** name. For example, the orchid named *Cypripedium parviflorum* has genus *Cypripedium* and its species is *Cypripedium parviflorum*; a species name is meaningless without its associated genus name.

The process of identifying a plant species can be quite challenging, even for the experts. In manual plant identification, the experts have to observe the characteristics of the plant accurately and compare it with the literature using what is called an identification key. An **identification key** is a list of plant characteristics that can lead the experts to a species name, the identity of the plant. This method has been used by botanists, also amateur botanists, for many centuries. Originally, and still today, there are botanic books on plant taxonomy (e.g. Casey (2020), O'Byrne (2008)) that can be consulted. In addition, within the laboratory setting, modern molecular, in particular genome analytic methods can be used (Behura et al., 2024). However, these are of little use in the field and are thus outside of the scope of this paper.

In the present paper, we take digital photographic images of orchids, as made by a smartphone's camera or a special-purpose digital camera, as the starting point for identifying orchids. Manual **orchid** identification involves examining physical flower characteristics such as the number of flowers, inflorescence, the shape of a flower, the texture of a flower, and so on, resulting in an orchid's name (de Vogel et al., 2025). Specialized botanic books on orchids, such as O'Byrne (2008),

[*] Corresponding author at: National Research and Innovation Agency (BRIN), Jl. M.H. Thamrin, Jakarta, 10340, DKI Jakarta, Indonesia.
*E-mail addresses:* diah006@brin.go.id (D.H. Apriyanti), l.j.spreeuwers@utwente.nl (L.J. Spreeuwers), p.j.f.lucas@utwente.nl (P.J.F. Lucas).

[1] https://gobotany.nativeplanttrust.org/

[2] https://www.cpbr.gov.au/cpbr/cd-keys/RFKOrchids/key/rfkorchids/Media/Html/identify.htm

https://doi.org/10.1016/j.engappai.2025.111961
Received 25 October 2024; Received in revised form 29 June 2025; Accepted 2 August 2025

that may be helpful in determining an orchid species, generally require considerable knowledge about orchids for their effective use. However, increasingly taxonomic plant information is on the World Wide Web, often specialized for particular genera of plants. There also exist several web-based applications that cover orchids, such as GoBotany[1] and the website of the Centre for Australian National Biodiversity Research (CAPBR[2]) and these are quite valuable for orchid enthusiasts, but they also presume that the user has considerable botanic knowledge. Alternatively, consulting a plant taxonomist for any plant that one wishes to identify is also not practically feasible. With the current international challenge of conserving our natural environment against further deterioration due to industrialization and climate change, there is an urgent need to archive information about plants, such as orchids, in their natural habitat before it is too late (Schiff, 2018). Thus, orchid identification is not only of interest to the orchid enthusiast.

Not surprisingly, in recent years computer-vision methods are increasingly applied to investigate automated orchid identification, with an emphasis on deep learning for identifying orchid species from digital photographic images (Liu et al., 2019; Seeland et al., 2017; Hiary et al., 2018). After training from image datasets, deep learning methods are often able to recognize features that may not be noticeable by humans, which can increase the identification accuracy. However, the advantage of high performance comes at a cost: lack of **explainability**. Deep neural networks (DNNs) act as a blackbox, which many users find unacceptable, in particular when they not only wish to obtain the right answer, but also wish to understand the reasons why this answer was provided. Lack of explainability undermines the trustworthiness of a system.

According to Petkovic (Petkovic, 2023), a **trustworthy system** requires both high accuracy and explainability. Providing users with understandable information on how machine learning models reach their decisions will increase user trust, and orchid identification is no exception to that rule. For several centuries, plant determination was done using characteristic features as described in botanic taxonomic books and manuals, a method at which taxonomists excel. It is this method that we chose as our starting point for our research with Bayesian networks (BNs) as the chosen machine learning method to represent and interpret orchid features and their uncertainty (Koller and Friedman, 2009a).

BNs are known as knowledge representation and machine learning methods that have the capability of explaining decisions based on their network structure plus their underlying joint probability distribution. An attractive feature is that expert knowledge can be incorporated to guide the learning process, which makes this method stand apart from other machine learning methods. The excellence of BNs for interpretable machine learning was already mentioned by Mihaljević et al. (2021), and Nicora et al. (Nicora et al., 2024).

In our research, orchid-flower features were extracted by end-to-end image-based deep learning; the extracted features were subsequently fused with a BN to express feature uncertainty. The resulting BN reflects traditional taxonomic determination based on flower features, taking into account the uncertainty of the DNN's feature extraction process. It is not only able to identify orchids from photographic images; in addition, it provides an **explanation** of why or why not a particular name of an orchid species has been associated with an image. However, although the above-mentioned limitations of DNNs are very relevant, their high performance remains an attractive reason for their use. Hence, we decided to combine both methods, **whole-image deep neural-network classification** with **feature-based Bayesian network classification and interpretation**, resulting in a new method that one could call an **explainable ensemble classifier** (Hastie et al., 2009). This way we achieve the best of both worlds: accuracy and interpretability.

To summarize, the scientific contributions of this paper are as follows:

- By combining two, very different, types of classifiers, a flower-feature-based Bayesian network and an end-to-end whole-image deep neural network, as a kind of ensemble classifier, solutions can be obtained that are augmented by an indication of their **trustworthiness**.
- The Bayesian network can be used to generate an explanation of provided solutions, taking into account ground flower features and uncertain, by deep neural networks extracted image features.
- The algorithm (Fig. 7) offers state-of-the-art performance for identification of orchids from photographic images.

Although we have not yet applied the algorithm to a field other than orchid identification, its principles are domain-agnostic, and, thus, there is no clear obstacle for applying it to other fields.

Details will be provided in the remainder of the paper. As far as we know, this is the first time that such a combined taxonomic-feature-based probabilistic and deep-neural-network-based method appears in literature.

The remaining part of the paper is organized as follows. Section 2 reviews related work in the different areas covered by our paper. In Section 3 the methods used and developed for the research are summarized, sometimes in detail when needed. Experiments and associated results are reported in Section 4. Finally, in Section 5 we discuss what has been achieved by our research. Its implications for the area of image-based plant, in particular orchid, identification are elaborated in Section 6.

## 2. Related work

### 2.1. Orchid flower recognition and deep learning

From a general point of view, Hindarto and Amalia, describe the kind of obstacles one may come across if one wishes to develop a flower recognition system using modern neural-network technology; they also argue why developing such systems is increasingly of importance given current ecological threats (Hindarto and Amalia, 2023). Nevertheless, there have been notable advancements in automated orchid identification in recent years, driven largely by the power of deep learning. Several studies have demonstrated the effectiveness of deep learning architectures in recognizing orchid species from images, which is why we restrict ourselves here to the description of these relatively new deep learning approaches.

Arwatchananukul et al. built a system that utilizes computer vision to identify *Paphiopedilum* orchids, also known as Venus slippers (Arwatchananukul et al., 2020). The system relies on a dataset of 1,500 images, with 100 samples for each of 15 different orchid species covered by the data. All images were captured at Paphiopedilum orchid gardens and meticulously classified by experts. The core of the system is a deep learning approach that combines a convolutional neural network with the Inception-v3 feature extractor from TensorFlow (Abadi et al., 2015). This combination achieved remarkable recognition rates, reaching up to 98.6%. Part of this success may be explained by the high photographic quality of the pictures: all orchids are similar in size and were positioned in the center of the picture. The good design of the employed deep neural network will also have played a role here. Finally, the researchers also developed a practical application —- a mobile app for Android devices. A limitation of the work is that only part of one orchid genus, viz. *Paphiopedilum*, is covered: 15 of about 100 different species.

Research conducted by Sarachai et al. (2022) focused on the development of another architecture specifically designed to classify orchid species from images, this time not restricted to one genus. The architecture tackles this challenge by incorporating a three-pronged approach:

(1) A Global Prediction Network (GPN). This network acts as a broad observer, analyzing the overall characteristics of the orchid flower in an image. By examining these global features, the GPN makes an initial prediction about the name of the orchid species.
(2) A Local Prediction Network (LPN) focuses on the finer details. It utilizes a spatial transformer network to zoom in on specific areas of the flower. This allows the network to analyze local features, such as the intricate details of individual flower organs. Based on these local analyses, the LPN generates its own prediction for the orchid species.
(3) An Ensemble Neural Network (ENN) acts as a harmonizer, combining the predictions from both the GPN (global features) and the LPN (local features) to a final classification.

Evaluation was done on three datasets: the well-known Oxford 17 and 102 datasets (Nilsback and Zisserman, 2008), containing 17 and 102 different species, respectively, and their own dataset containing 52 orchid species. The performance varied from 94.18% to 98.39%, where the lowest performance was achieved for their own, harder orchid dataset due to higher variation in picture quality, compared to the Oxford datasets.

An ensemble voting-scheme method, related to the previously described method, was used to predict orchid species from three pre-trained deep learning models by Ou, et al. Ou et al. (2023). The three pre-trained models are ResNet50, EfficientNet, and Big Transfer (BiT). The method could improve the accuracy of the best single pre-trained model by 2.8% to 3.1% when validated by different datasets.

The effectiveness of deep learning architectures was also observed by Wang and Wang (Wang and Wang, 2024). They applied transfer learning technology to recognize 12 different types of orchid from 12,227 images. The method was able to achieve an accuracy of 96.16%. In another research, Wang et al. (2024) proposed to merge features extracted from multiple layers and different stages, and subsequently trains the classifier on this integrated representation. The model obtained a 92.89% classification accuracy rate, which was higher than when only using Resnet34.

To conclude deep neural networks performance for orchid identification varies, dependent on size of the dataset, quality of the photographic images, number of species distinguished, and DNN architecture employed, between 93% and 98.5%.

### 2.2. XAI and deep neural networks

Due to the impact of deep learning and deep neural networks, many different methods have been proposed to improve the explainability of neural networks, and these methods have become known as **eXplainable AI**, **XAI** for short (Longo et al., 2024). The term XAI was introduced by DARPA and has since become widely recognized and used (Holzinger et al., 2022). One should realize that providing an explanation of solutions generated by an intelligent system has been explored already since the 1970s (Lucas and van der Gaag, 1991). As an example, the capability of providing an explanation of a solution (why a question was asked and how a solution was achieved) was already part of the 1970s rule-based MYCIN system; it was considered an essential ingredient for a system that was intended for use by clinicians (Shortliffe, 1976; Lucas and van der Gaag, 1991). Hence, the topic of explanation is by no means new, but has become an important issue of debate regarding the acceptance of blackbox neural networks.

A diagrammatic overview of the concept of XAI as proposed by DARPA is depicted in Fig. 1 (Gunning and Aha, 2019). An XAI system should be able to provide explanations that help users know the system's strengths, weaknesses, and future performance, while supporting making corrections. Humans have both explicit and implicit knowledge, which they use in tandem. Understanding and explaining things requires **explicit knowledge**, while a DNN that learns from data to create a probabilistic model essentially acquires **implicit knowledge**. Knowledge-representation systems, using e.g., logic, rules, or knowledge graphs, are based on explicit, symbolic knowledge. Whereas currently these two approaches to AI, neural and symbolic, are seen as each other's opposite, researchers are working to bridge the gap between the two (Xu et al., 2019).

According to Kenny et al. (2021), two main types of XAI system can be distinguished. Firstly, **transparency by design**, i.e., designing a system in a transparent fashion supporting the understanding of how a system or model is working. The second type is **post-hoc explanation**, which is an explanation that is based on evidence that supports the conclusion, e.g., by showing pixel importance. There exist several XAI methods based on the first approach. One of these, called **Prototypical Part Network** (**ProtoPNet**), is able to identify parts of the test image having similar appearance to the learned prototypes (Chen et al., 2019). Another method, called **Neural Prototype Tree** (**ProtoTree**), developed by Nauta et al. (2021), was built based on ProtoPNet but generates fewer prototypes. It consists of training a convolutional neural network followed by a binary tree. This approach simplifies model understanding and error detection by segmenting the reasoning into smaller steps.

One particularly influential approach that can be categorized as the second one is called **LIME** (**Local Interpretable Model-agnostic Explanations**). It explains a prediction made by a model by fitting a local surrogate model, e.g., a simple linear function, whose predictions are easier to explain than the original model (Ribeiro et al., 2016). Another XAI method that applies this second approach is rule-based and called **anchors** (Ribeiro et al., 2018). An anchor is a rule that 'anchors' – hence the name of the method – a prediction locally, such that changes of other features of a data instance does not change the anchor. This method is capable of interpreting the behavior of various models and can be applied to different domains such as tabular data, images and text. Finally, **SHapley Additive explanation** (**SHAP**) is also a type of XAI that uses post-hoc explanations. SHAP employs a method based on cooperative game theory that results in what are called 'Shapley values'. They are used to determine the contribution and importance of different feature combinations on a model's output (Lundberg and Lee, 2017).

Unfortunately, none of the XAI methods mentioned above are in our opinion particularly suitable for representing and explaining uncertain taxonomic knowledge.

### 2.3. Bayesian networks as means for XAI

**Bayesian classifiers** are Bayesian networks with a restrictive, relatively simple tree topology and they have been popular for a very long time, at least since the end of the 1960s (e.g. de Dombal et al. (1972), Chow and Liu (1968)). Nonrestrictive Bayesian networks are a more recent invention from the end of the 1980s (Pearl, 1988; Cowell et al., 1999; Koller and Friedman, 2009b). There are several papers that have shown that despite its restrictive nature, the simple, often called 'naive', Bayesian classifier performs remarkably well (Domingos and Pazzani, 1997; Friedman et al., 1997). Nevertheless, often, as in our research, it pays off when adding some extra complexity to a Bayesian network as it adds to their explainability and sometimes also performance.

Bayesian networks are examples of white-box representation methods, as they allow explaining their conclusions based on both their network structure and associated probability distribution. There are several papers that offer evidence that Bayesian networks are a promising XAI method, e.g. Butz et al. (2022), Lacave and Diez (2002), van Leeuwen et al. (2024). This evidence explains why we decided to use Bayesian networks as a backbone of our research.

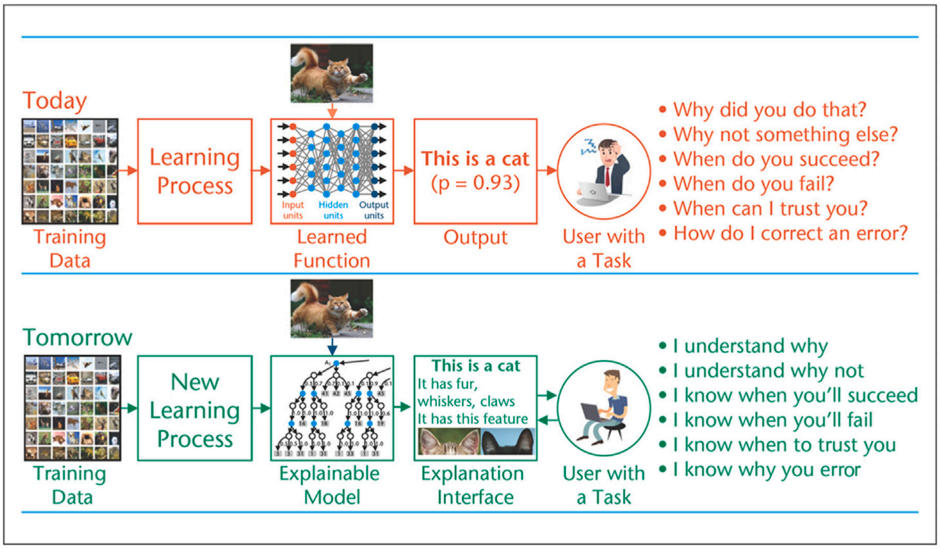


**Fig. 1.** The XAI concept as proposed by DARPA (diagram courtesy of DARPA (Gunning and Aha, 2019)).

## 3. Methods

Modern computer vision methods are good at providing end-to-end solutions, in particular, as those offered by deep neural networks (DNNs). The input is just an image that includes a particular object, and without explicitly taking into account the plant's visual characteristics, or features, that taxonomists would use to distinguish them from each other. That is the reason why we decided to explore a second method: plant feature extraction, based on exactly the same images used for training the end-to-end DNNs, where again DNNs, although different ones – specialized feature extractors – were employed. As feature extraction would yield uncertainty, meaning that some features obtained by the automated extraction process were wrong in comparison to the ground truth, all the features had to be combined to predict the plant's species. A modern way to do this is by building a Bayesian network (BN) (see below for details). By combining the two methods, we basically combine an implicit method, to some extent akin to the *thinking fast method* of Daniel Kahneman (Kahneman, 2011), and the explicit rational, taxonomic methods, related to the *thinking slow method*. The explainable ensemble method that combines a BN and DNN was born.

An overview of how the methods explored in this paper are intended to work together for practical use, is depicted in Fig. 2, whereas the actual algorithm is presented in Fig. 7, and will be discussed later. Explicit flower characteristics such as texture, number of flowers, and so on (see Section 3.1) are extracted by deep neural network (DNN) classifiers. The extracted features are then fed into a Bayesian network (BN) to predict the orchid species. The BN also supports the generation of an explanation. To increase the accuracy of the entire system, we also employ another DNN. Although similar in architecture, the main difference with the DNN feature classifiers is that it classifies an image, purely based on image data rather than flower features. Finally, the DNN classification is combined with the BN predictions, where the latter is also used to explain the solution. The result is an explainable, high accuracy automated orchid identification system, which may be seen as an **ensemble classifier**, although an uncommon one as will become clear below.

Before we can actually use the DNN and BN in the way just described, we first need to train both using data. Most of the remainder of this section is about the methods of machine learning we developed for that purpose. In addition, the features used to characterize orchid flowers are introduced. Also described is how image features can be extracted using DNNs and much is devoted to explaining how we can build a BN for flower identification using a mixture of descriptive flower features from textual data and uncertain image features obtained by DNNs.

### 3.1. Descriptive features of orchids

A flower of an orchid consists of several parts as depicted in Fig. 3. These parts are described by means of flower features, typically also used by taxonomists; they are essential for differentiating between various types of plants or flowers. In current research on automated plant recognition systems, researchers use **implicit** features, i.e., image features that say nothing specific about the flower color, form, texture, etc. Instead, in our research we also use **explicit**, descriptive features inspired by how taxonomists characterize plants. **Descriptive features** are based on the description of the characteristics of a flower. For example, the *Cypripedium parviflorum* has a flower with color 'green' and 'red', has a 'pouch shape' of the labellum; orchid *Cypripedium reginae* has a flower which is 'red', and has a 'simple shape' of the labellum, etc. The **identity** of a plant, i.e., its **species**, is represented by the variable 'CLASS' and refers to the name of an orchid species. In our research the focus is on the identification of species (which thus by default also gives the genus).

The following features $F$ and their associated domain, indicated by $D(F)$, were selected to describe the images in the dataset on the basis of being easily identifiable by both humans and computer vision systems and by their occurrence in the descriptions associated with the images in our dataset:

(1) **T**: Texture of the labellum with domain $\{\text{spots}, \text{nospots}\}$.
(2) **NF**: Number of Flowers, a count of the number of distinguishable flowers in the picture, with a qualitative domain of three values: 'single or pair', 'a few', and 'many'.
(3) **IN**: INflorescence with domain 'single or pair', 'panicle', 'raceme', and 'spike'.
(4) **CF**: Color of Flower describing the color of sepals and petals, in terms of 8 color pairs: 'GreenGreen' (all flowers are entirely green), 'GreenRed', 'GreenYellow', 'PurplePurple', 'PurpleYellow', 'RedRed', 'RedYellow', and 'YellowYellow'.
(5) **CL**: Color of Labellum described with the same 8 color pairs as for CF.
(6) **LC**: shape of Labellum with a domain consisting of 'fringed', 'simple', 'lobed', and 'pouched'.

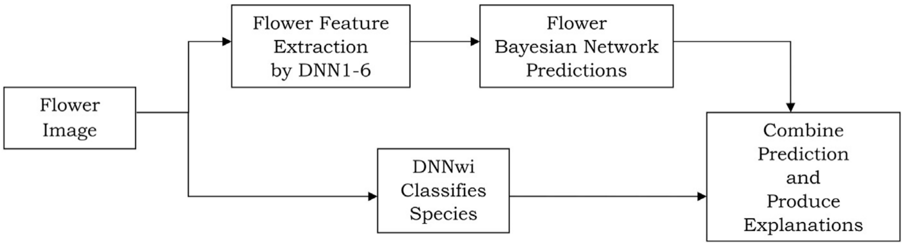

Flower Image
Flower Feature Extraction by DNN1-6
Flower Bayesian Network Predictions
DNNwi Classifies Species
Combine Prediction and Produce Explanations

**Fig. 2.** An overview of the various methods, and their relationship, for automatically identifying an orchid from a photographic digital image, augmented by an explanation provided by the orchid-flower Bayesian network; DNN: Deep Neural Network. There are two types of DNN mentioned in this flow diagram: feature classifiers, DNN1-6, (one DNN for each feature), and a whole-image classifier, DNNwi. The algorithm that corresponds to this flow chart is presented in Fig. 7.

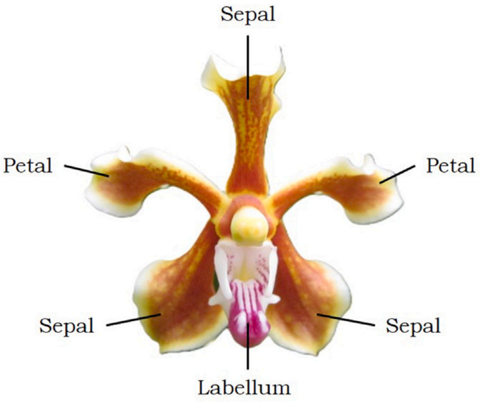


**Fig. 3.** Parts of an orchid flower. The flower consists of sepals and petals; there is only one labellum which may have its own separate color, different from the sepals and petals.

Fig. 4 illustrates some of the characteristic features of orchids. An effective method to describe color features, here CF and CL, using two colors in combination was developed in previous research (Apriyanti et al., 2021). Other features, e.g., geographical location and season of flowering, were not used by us as the orchids we have in our dataset come from one geographical area.

Descriptive features of orchid species provide the gold standard for describing orchids, and although there is some variation among species, e.g., a particular orchid species may vary in color, in general this variation is very limited. As the goal of the research is to determine an orchid species based on an analysis of digital photographs, the flower features mentioned above also exist in a digital image version, obtained by feature extraction using DNN. The descriptive features and the orchid species fill part of the orchid dataset – a small sample of it is shown in Table A.3 – we put together. How the descriptive features are complemented by image flower features is described in the next subsection.

### 3.2. Deep neural networks for feature extraction and flower–image interpretation

To extract the orchid image features from digital photographic image data, each descriptive feature was taken as the ground truth label for the corresponding image feature. Six different DNN classifiers were trained, using supervised learning, one for each individual feature mentioned above. After extensive experimentation with many different DNN architectures, described in detail in a previously published paper (Apriyanti et al., 2023), **Xception** (Chollet, 2017) was chosen as the backbone of the classifiers.

The training inputs for the DNNs were images with a dimension of 224 × 224 × 3. For this research, the pre-trained architecture of Xception was used by freezing the first layer and unfreezing the rest. We added a flatten layer and one dense layer with 256 neurons using a ReLU activation function (Agarap, 2018). We also added a dropout layer of which the value was 0.5. As final layer of the DNN we used the softmax function. The features derived using this process are uncertain because uncontrolled conditions of the images, such as lighting, pose variation, and scaling, may have a big influence on the actual feature value extracted. The Xception architecture and the hyperparameters used for the whole-image interpretation were the same as for feature classifiers, except for the output target (orchid species rather than an orchid feature).

As we will describe below, the probabilistic information needed for including the image features into a Bayesian network consists of the true positive rates and true negative rates for each of the features, which were already reported in a previous paper (Apriyanti et al., 2023) and are repeated for completeness in Table A.5. The sample dataset in Table A.3 contains the image features for the orchids; note that these are not always the same as the ground truth features. For example, for obtaining the color of flower, the human interpreter defines that the color is white, however, if the illumination of the image is low then the computer defines it as gray. This uncertainty, however, can be represented adequately in a Bayesian network.

### 3.3. Design of the Bayesian network

#### 3.3.1. Basic notions

A **Bayesian Network** $\mathcal{B} = (G, P)$ represents a joint probability distribution over a set of **random variables** $X = \{X_1, \dots, X_n\}$ in the shape of a graphical model $G = (V, A)$ that expresses conditional independence assumptions, and therefore also conditional dependence assumptions, the complement. It consists of a set of **nodes** $V = \{1, \dots, n\}$ and a set of **directed edges** $A \subseteq V \times V$, with an edge $(i, j) \in A$, also denoted as $i \to j$, that together form a directed acyclic graph $G$. Each node $i \in V$ corresponds one-to-one to a random variables $X_i$ and vice versa, while directed edges $i \to j$ represent the relationship between random variables $(X_i, X_j)$.

A Bayesian network specifies all the conditional probabilities that are needed to compute the joint probability distribution of the variables it contains using the following equation:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_{\mathrm{Pa}(i)}) \tag{1}$$

where $\mathrm{Pa}(i) = \{k \mid (k, i) \in A\}$ denotes the set of **parents** of node $i$, where $X_{\mathrm{Pa}(i)}$ is the associated set of random variables (Lucas et al., 2004). This expression implies that all ancestor variables $X_j$ (variables with a path to $X_i$) of $X_i$ are conditionally independent of variable $X_i$ given the parent variables $X_{\mathrm{Pa}(i)}$. This usually results in a compact specification of the joint probability distribution. One should also realize that a joint probability distribution allows one to compute any (conditional) probability of any subset of variables given any other subset of variables just by using the basic rules of probability theory.

To build a Bayesian network, we can use one of the following approaches: (i) exploiting (in particular) causal knowledge of experts

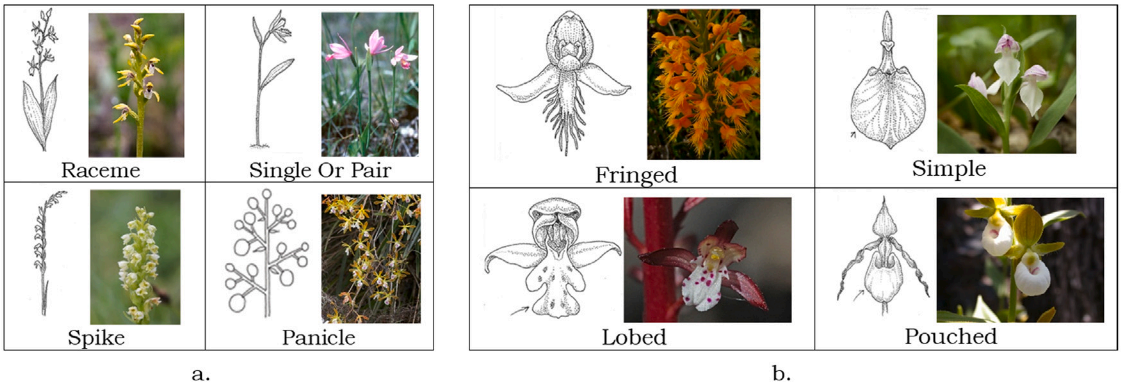


**Fig. 4.** Morphological features: (a) Inflorescence, (b) Labellum characteristic.

in a given domain; (ii) learning from data; (iii) combining the two approaches. In our research, the third approach is employed, i.e., learning from data combined with expert background knowledge.

### *3.3.2. Learning Bayesian networks from data*

There are two steps in learning a Bayesian network from data: firstly, its graphical structure has to be learned from the data, and secondly, the probabilistic parameters $P(X_i \mid X_{\mathrm{Pa}(i)})$ need to be estimated before we can do inference.

**Structure learning** is the process of finding the structure that fits the data best. Finding the best structure is a challenging task since the number of model structures is large (super-exponential), even if the network contains few nodes (Robinson, 1976). As a consequence, structure learning is not done by exhaustively exploring the entire space of directed acyclic graphs, as this would be prohibitive for graphs with more than 6 or 7 nodes. Rather, structure learning is accomplished by carrying out conditional independence tests in a local fashion, where only subsets of variables are considered. This type of algorithm is called **constraint-based**. One may also learn the graph structure by traversing the space of directed acyclic graphs in a heuristic manner, using various score metrics to guide the search; these algorithms are called **score-based**.

Examples of constraint-based structure learning algorithms are the stable PC algorithm (Colombo and Maathuis, 2014) and Grow-Shrink (GS) (Margaritis, 2003). All of these algorithms use conditional independence tests to find a structure that mirrors the independences reflected in the data.

The score-based approaches employ a **score function** to assess how well the structure explains the data, possibly taking into account prior knowledge about probabilistic parameters and network structure. Most score functions used are based on the idea that the **likelihood** of the data given the BN, $P(\mathrm{Data} \mid \mathrm{BN})$, is a suitable measure of goodness-of-fit. However, usually score measures such as Bayesian Dirichlet Equivalent (BDe) score (Heckerman et al., 1995; Chickering, 1995) and Bayesian Information Criterion (BIC) score (Chickering, 1995) are used as they penalize complexity of a BN, making complex networks less prone to overfitting than simpler networks (Needham et al., 2007). BIC is defined as follows:

$$\mathrm{BIC}(\mathcal{B} \mid \mathrm{Data}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \log P(X_j^i \mid X_{\mathrm{Pa}(j)}^i) - d\,\frac{\log m}{2} \tag{2}$$

with $m = |\mathrm{Data}|$ and $d$ a measure of the complexity (dimensions) of the probability tables of the BN. The first term in Eq. (2) is actually the log-likelihood of the data given the BN model, i.e., $\log P(\mathrm{Data} \mid \mathrm{BN})$, whereas the second term acts as a penalty term, which inhibits learning complex Bayesian networks. In this paper we will rather use the BDe score, which also incorporates the log-likelihood of the data, but in addition allows using prior knowledge (in our case we will use a uniform prior on the probabilistic parameters and start search with a prior network structure). Search algorithms using this approach are the hill-climbing algorithm (a greedy search) and tabu search, also greedy search but with a mechanism included to prevent revisiting previously visited nodes. When using structure learning it is also possible to restrict a score-based algorithm by the specification of a **white-list** (meaning that all arcs included should be present in the result) and **black-list** (mentioned arcs should not be included in the result) (Scutari, 2025; Scutari and Denis, 2021).

The algorithms mentioned above are purely machine-learning based. Ways to restrict the search process is offered by **restrictive** algorithms that are especially used for classification purposes: the naive Bayesian classifier (Domingos and Pazzani, 1997) and the tree-augmented network (Chow and Liu, 1968). The **naive Bayesian classifier** is a Bayesian network with a fixed structure (a single class variable and feature variables that are assumed to be conditionally independent of the class variable) (Friedman et al., 1997), whereas the tree-augmented network adds a tree structure to the naive network structure. Later in Section 3.6, it will become clear that we developed a type of Bayesian network learning that starts with a naive backbone to be followed by structure learning, which we call **semi-supervised structure learning**.

Following structure learning, the next step is **parameter learning**, i.e., given the Bayesian-network graph with its encoded conditional independence assumptions, the probabilistic parameters have to be estimated from the data. The simplest approach to learn the parameters is by maximizing the likelihood of the data, called **maximum likelihood**, resulting in the maximum likelihood of the data given the BN. For discrete random variables, maximum likelihood corresponds to counting the frequencies of the occurrences of values of variables in the data. This approach is based on the data only, and often called the **frequentist approach** (Hastie et al., 2009). Another approach is to learn the parameter by starting with a prior distribution and using a **Bayesian approach** by updating the probability distribution by the new data (Heckerman et al., 1995; Hastie et al., 2009).

### *3.4. Deterministic random variables and conditional independence*

By definition, all the conditional independences in the graph $G = (V, A)$, denoted by the triples $U \perp\!\!\!\perp_G W \mid Z$, with $U, W, Z$ disjoint subsets of nodes from $V$, are reflected in the probability distribution $P$, denoted by $X_U \perp\!\!\!\perp_P X_W \mid X_Z$, i.e.

$$U \perp\!\!\!\perp_G W \mid Z \Longrightarrow X_U \perp\!\!\!\perp_P X_W \mid X_Z$$

It is said that the graph $G$ is an **independence map** (I-map) of the probability distribution $P$. In words: “if paths between nodes are disconnected or paths are blocked, then there will be corresponding conditional independences in the probability distribution $P$”.

Independence statements $X_U \perp\!\!\!\perp_P X_W \mid X_Z$ are determined by carrying out independence tests that establish that $P(X_U \mid X_W, X_Z) =$

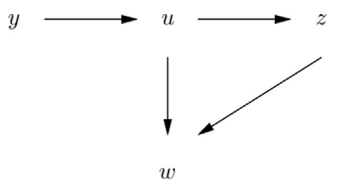


**Fig. 5.** The graph $G$ of a Bayesian network with functional dependence $X_u = f(X_y, X_z)$.

$P(X_U \mid X_Z)$ for any value of the set of variables $X_U, X_W$, and $X_Z$. If the independence test fails, dependence holds: $X_U \not\perp\!\!\!\perp_P X_W \mid X_Z$.

Statements of the form $U \perp\!\!\!\perp_G W \mid Z$ are read-off from the graph $G$ by applying the conditions of **d-separation**, telling that if all (undirected) trails from any vertex in $U$ to any vertex in $W$ is blocked by any vertex in $Z$ the statement holds. A trail is **blocked** if none of the vertices in $Z$ are **colliders**, i.e. of the form $\cdot \rightarrow w \leftarrow \cdot$ on the considered trail, or a descendant of $w$. If the d-separation test fails, the sets of vertices are d-connected, denoted as $U \not\perp\!\!\!\perp_G W \mid Z$. Often a directed edge of a directed acyclic graph can be reversed (using Bayes' rule to compute the new probabilities), yielding a network with exactly the same d-separation properties. Such networks are called **Markov equivalent**. Arcs that participate in a collider cannot be reversed without adding some extra dependences as edges.

However, the independence relationships do not hold when the probability distribution $P$ also contains **deterministic** probabilities $P(x_v \mid x_{\text{Pa}(v)})$, i.e., with $P(x_v \mid x_{\text{Pa}(v)}) \in \{0, 1\}$. Deterministic probabilities are expected to occur frequently in our research, as part of the data that will be used consists of descriptions of orchids, and these are always completely certain and precise (although sometimes ambiguous).

As an example, consider the Bayesian network shown in Fig. 5. Here we have that $\{u\} \not\perp\!\!\!\perp_G \{w\} \mid \{y, z\}$ (nodes $u$ and $w$ are directly connected), but because of the functional dependence $X_u = f(X_y, X_z)$ it holds that $X_u \perp\!\!\!\perp_P X_w \mid \{X_y, X_z\}$ as $X_u$ is fully determined by $X_y$ and $X_z$. From data one can only infer probabilistic independence statements $X_u \perp\!\!\!\perp_P X_w \mid X_z$, which means that deterministic variables might induce extra independences in the learned graphical structure.

In particular the constraint-based structure learning algorithms, such as the (stable) PC algorithm, have difficulty in dealing with deterministic data. This is due to the use of a test in the algorithm that examines the conditional independence of two variables given a subset of other variables. A typical example is the use of **mutual information** $I(X, Y \mid \mathbf{Z})$ defined as follows:

$$\begin{aligned} I(X, Y \mid \mathbf{Z}) &= H(X \mid \mathbf{Z}) - H(X \mid Y, \mathbf{Z}) \\ &= \sum_{\mathbf{Z}} \sum_{X} \sum_{Y} P(X, Y, \mathbf{Z}) \log \frac{P(X, Y \mid \mathbf{Z})}{P(X \mid \mathbf{Z}) P(Y \mid \mathbf{Z})} \end{aligned} \quad (3)$$

where $H(X \mid \mathbf{Z}) = \sum_X P(X \mid \mathbf{Z}) \log P(X \mid \mathbf{Z})$ is the **conditional entropy**. In case that variables $X$ and $Y$ are conditionally independent given the set of variables $\mathbf{Z}$ we have that $P(X, Y \mid \mathbf{Z}) = P(X \mid \mathbf{Z}) P(Y \mid \mathbf{Z})$, i.e., $I(X, Y \mid \mathbf{Z}) = 0$. However, if the conditional probability distributions $P(X_i \mid x_{\text{Pa}(i)})$ are deterministic, when computing the entropy $\lim_{p \downarrow 0} p \log p = 0$ and for the case that $p = 1$, we have that $\log p = 0$, hence, the entropy becomes 0. As a consequence, the arcs between the variables $X$ and $Y$ are omitted, despite the fact that there exists a deterministic relationship $Y = f(X)$ between $X$ and $Y$. Although a deterministic relationship between variables $X$ and $Y$ means that $X$ and $Y$ are **informational equivalent** from a probabilistic point of view, i.e., for example $Y$ could be omitted, this does not imply that deterministic relationships are useless. On the contrary, they can be useful for prediction (Lemeire et al., 2008), as in the present paper.

### 3.5. Virtual evidence

The method of including **virtual evidence**, sometimes also called **likelihood evidence**, is meant to express uncertainty about the observation of a particular value of a feature, in our case for example the uncertainty of observing a particular color of an orchid flower in a photograph, according to a deep neural network algorithm (Pearl, 1988). The idea is that a variable $F$ (feature) is linked to a virtual node, denoted $F_v$, in the following graphical form:

$F \rightarrow F_v$

With this subgraph of a bigger Bayesian network, a set of conditional probability distributions has to be associated: $P(F_v = yes \mid F = yes) = x$ and $P(F_v = no \mid F = yes) = 1 - x$, and $P(F_v = yes \mid F = no) = y$ and $P(F_v = no \mid F = no) = 1 - y$. Note that these probabilities correspond to the true positive rate (TPR), false negative rate (FNR), false positive rate (FPR), and the true negative rate (TNR) for each of the features. For the orchid flower Bayesian network, these numbers are provided in Table A.5.

In order to classify an orchid from a photograph, one first needs to extract the features using the feature neural networks obtained by deep learning. This yields a unique value for each of the features, which is used as **hard evidence** into the Bayesian network. Given a value for each of the features $F_v$, e.g. $F_v = yes$, computed is the probability distribution $P(F \mid F_v = yes)$ by means of probabilistic inference. This actually corresponds to using Bayes' theorem:

$$P(F \mid F_v = yes) = \frac{P(F_v = yes \mid F = yes) P(F = yes)}{P(F_v = yes)}$$

### 3.6. Semi-supervised structure learning

The ground-truth features that are used to describe the features verbally are for each orchid very similar and often identical. This results in much determinism in the probabilistic relationships between the ground-truth feature variables with the consequences summarized above in Section 3.4. Despite the fact that there is sometimes still some nondeterminism left, it is not possible to determine scoring functions (because of zero probabilities) or dependences among variables. One could of course use Laplace smoothing when computing conditional probabilities from the data; however, that would not do justice to the nature of the data in this specific case. A natural way to look at this part of the Bayesian network is to see the class variable as an (almost) deterministic function of the ground-truth feature variables. To represent this as a graph can be done by a naive Bayesian network structure with the class variable as root and the ground-truth feature variables $F_i$ as the leaves connected to the class variables, as follows: CLASS → $F_i$, for $i = 1, \ldots, 6$.

This determinism, however, does not concern the **image** feature variables, because in the extraction process from the images by deep learning there is always some, and sometimes significant, uncertainty involved. The uncertainty is exemplified in the results for the individual image features summarized in Table A.5. The general idea of the semi-supervised structure learning method we developed for this specific problem with partly deterministic variables and partly uncertain versions of the same variables, is illustrated in Fig. 6. Thus, each feature $F_i$ and image feature $\text{IF}_i$ is assumed to be independent (d-separated) and the role of structure learning is to find dependences among the features and image features variables $\text{IF}_i$ from the data that improve the score of the resulting Bayesian network.

However, instead of applying structure learning one could also link features and corresponding image features together manually as follows:

$$\text{CLASS} \rightarrow F_i \rightarrow \text{IF}_i, \quad \text{for } i = 1, \ldots, 6$$

resulting in what we will call a "double-naive" Bayesian network. Alternatively, one could manually add a few arcs based on background knowledge, resulting in what we will call a "semi-naive" Bayesian network below.

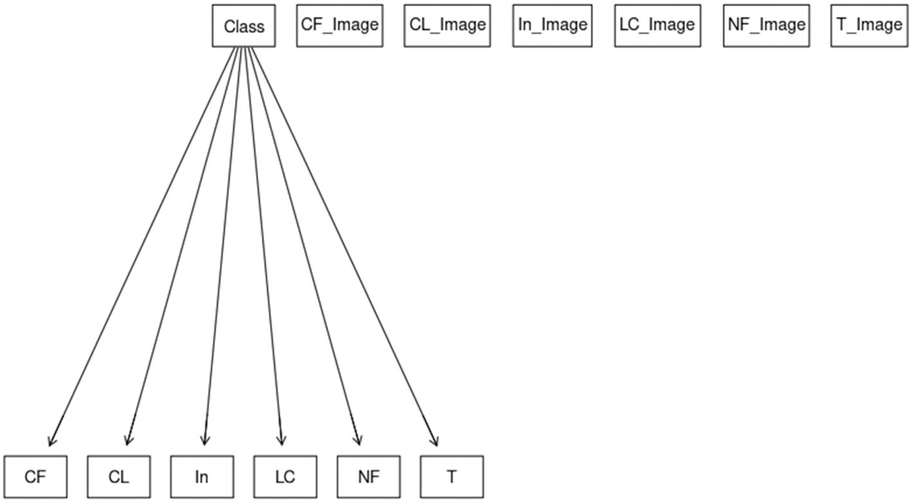


**Fig. 6.** General backbone of Bayesian network feature structure learning. Note that each image feature is independent of all other variables; the idea is that dependences between the features and image features are learned by structure learning restricted to features and image features. Thus, this Bayesian network is used as a start, or backbone, of the search process.

### *3.6.1. Complexity of the network*

When we observed the probability distribution of each feature in each class, it became clear that for any of the values of the CLASS variable there are deterministic probability distributions among the feature variables. This makes it less likely that dependences between the feature variables are discovered.

However, if one would allow the CLASS variable to encode every individual species of the plant considered in this paper, i.e., the orchids, double-naive or semi-naive Bayesian networks, as introduced in the previous section, would suffice to describe the individual orchids as long as the features offer sufficient discriminative power. As there are about 25,000–30,000 different orchid species, the dimension of the Cartesian product of the feature variables should be at least 25,000 in size. The advantage of using double-naive or semi-naive Bayesian networks as a representation is that probabilistic inference has linear time and space complexity, whereas Bayesian networks in general are NP-hard (Cooper, 1990). Thus, the chosen representation is very efficient.

The features should be powerful enough to differentiate between different types of flower. However, it is not always possible to differentiate between different orchid types based on a description of their flowers as sometimes the descriptions are quite similar. As a consequence, the only thing we can do is to use a sufficient number of easily identifiable features and sometimes accept that some orchids cannot be distinguished based on their features.

Using the definition of Bayesian networks it is in principle possible to describe flowers (of plants) in terms of their features. The joint probability distribution in this case would read as:

$$P(\mathrm{CLASS}, \mathrm{T}, \mathrm{NF}, \mathrm{IN}, \mathrm{CF}, \mathrm{LC}, \mathrm{CL})$$

and can be computed using Formula (1) given a particular graph structure associated with the Bayesian network. Note that given the various domains of the variables, in this case using multinomial data, the conditioning set of the set of conditional probability distributions

$$P(\mathrm{CLASS} \mid \mathrm{T}, \mathrm{NF}, \mathrm{IN}, \mathrm{CF}, \mathrm{LC}, \mathrm{CL})$$

would be able to index

$$|D(\mathrm{T})| \cdot |D(\mathrm{NF})| \cdot |D(\mathrm{IN})| \cdot |D(\mathrm{CF})| \cdot |D(\mathrm{LC})| \cdot |D(\mathrm{CL})|$$
$$= 2 \cdot 3 \cdot 3 \cdot 8 \cdot 3 \cdot 8 = 3{,}456$$

number of orchids. Given the estimated number of orchids, the result obtained above is approximately one-tenth of this number. By adding more features, for example, flowering date (FD) assuming that its value is expressed in terms of a specific month, we would in principle be able to index each of the 25,000 individual orchids. However, there may be overlap of some orchid features and, in addition, it is known that there exists some biological variation within one species, e.g., a species may have some variation in color. For our orchid database with 63 species, the features included in our models offer enough expressive power, although this does not imply that the image features are always able to discriminate between the species because of their uncertainty.

### *3.7. An explainable ensemble classifier using a DNN and a BN*

As Eq. (1) shows, a Bayesian network represents a factorized probability distribution and as such allows, in principle, computing any probability distribution of any subset of variables conditional on any evidence. Whereas at first sight this appears to offer only limited explanatory power, one should also realize that the graph structure of a Bayesian network allows one to follow the reasoning flow through the network, sometimes blocked or unblocked by the d-separation due to part of the evidence. In addition, a Bayesian network supports **counterfactual reasoning** where the effects of what-if assumptions can be explored just by looking at changes due to the assumptions (Pearl, 2009).

When restricted to orchid identification based on $m > 0$ explicit (descriptive) features, the following probabilistic queries would yield explanatory insight:

- The distribution of $P(\mathrm{CLASS} \mid \mathrm{Evidence})$;
- Given solution CLASS $= c$ (for example obtained by the deep neural network based on image data), show which image features $\mathrm{IF}_i$ are most likely:

  $$\arg\max_f P(\mathrm{IF}_i = f \mid \mathrm{CLASS} = c),$$

  for $i = 1, \ldots, m$;
- Counterfactual reasoning with some assumptions about the CLASS, features, and image features entered as evidence into the network. For example: Let us assume that CLASS $= c$ is offered as solution by one classifier, whereas another classifier gives CLASS $= d$ as result. Comparing the two predictions by computing the difference, alternatively one could compute the ratio, yields:

  $$P(\mathrm{IF}_i = f \mid \mathrm{CLASS} = c) - P(\mathrm{IF}_i = f \mid \mathrm{CLASS} = d)$$

  for $i = 1, \ldots, m$. The difference (or ratio) will give insight into the differences between the image characteristics between these two different suggested solutions.

```
algorithm ExplainableEnsemblePrediction

input BN, DNN, evidence, class, image, numfeatures, maxrank
classpredictions = ProbabilisticInference(BN, class, evidence)
dnn-classvalue = Classify(DNN, image)
if dnn-classvalue $\in$ classpredictions[1 : maxrank] then #(Case 1)
    Display "Trusted agreement DNN and BN solutions up to max rank ",
            maxrank, " solution: "
    Explain(BN, evidence, class, dnn-classvalue, classpredictions, image, numfeaturess,
            maxrank)
else #(Case 2)
    Display "DNN solution may be trusted, but should be checked, "
    Explain(BN, evidence, class, dnn-classvalue, classpredictions, image,
            numfeaturess, maxrank)
    if $\neg$UserSatisfied() then #(Case 3)
        Display "DNN not trusted; trusted BN solution "
        Explain(BN, evidence, class, classpredictions[1], classpredictions, image,
                numfeaturess, maxrank)

function ProbabilisticInference(BN, class, evidence)
    solutions = sort-high-low($P$(class | evidence; BN))
    return(solutions)

function Classify(DNN, image)
    input-layer = Encode(image)
    return(arg max Process(DNN, input-layer))

function Explain(BN, evidence, class, classvalue, solutions, image, $m$, $n$)
    Display classvalue, "; alternative BN solutions: ", solutions[1 : $n$]
    Display "Predicted Image Features: ", for $i = 1, \ldots, m$ :
            $\arg\max_f P(\mathrm{IF}_i = f \mid \text{class} = \text{classvalue}; \mathrm{BN})$
    Display "Compare to ", evidence, image

function UserSatisfied()
    Display "Satisfied? "; return(yes $\equiv$ ReadInput())
```

**Fig. 7.** DNN (deep neural network) and BN (Bayesian network) as an ensemble classifier for explainable predictions with user-in-the-loop. Process: a neural network computational engine. Class: target class variable of the classification; Numfeatures: number of features in the BN; maxrank: maximum number of BN predictions taken into account.

A possible algorithm for combining the two representations, DNN and BN, as an ensemble classifier is shown in Fig. 7. Selecting a particular rank, 'maxrank', the 'maxrank' first most likely orchid species are collected and it is checked whether the orchid identified by the DNN is among them. If this is the case, the explanation from the BN is presented to the user. However, even if the output of the DNN differs from the solutions of the BN, we still can provide an explanation by highlighting reasons in terms of the features represented in the BN. A correspondence between an identification of an orchid in terms of extracted features and a whole-image neural-network classification can be interpreted as that there is high confidence that the right orchid has been found, i.e., the solution can be **trusted**. Finally, if the DNN solution cannot be trusted, we only have the BN to resort to and we select the highest ranked solution, which is also explained in terms of the BN. There is a clear role for the explanation facility and the user, who in the end decides whether it is better to go for the BN solution. Thus, on the one hand DNN and BN are used together to increase the number of trusted solutions, compared to simply using them on their own, and on the other hand the BN is used to explain solutions in terms of the structure and probability distribution of the BN.

The parameter 'maxrank' is not given a fixed value; its value should be determined by experimentation with a target dataset. In our experimental setting we finally choose the value 'maxrank = 5 or 4', which will be discussed below in Experimental Results (Section 4). Note that by setting 'maxrank' to the value of the number of different class values ($|D(\mathrm{CLASS})|$ flower species in our case), the BN is completely ignored for the purpose of classifying an image, although it is still used for generating an explanation. By giving 'maxrank' a small value, say 1 or 2, the number of trusted solutions (agreed by DNN and BN) decreases, but the level of trust increases as the two classifiers in the ensemble use different methods for the same purpose.

To illustrate that the cases distinguished in the algorithm shown in Fig. 7 actually exist, and that the generated explanation may be valuable, we discuss three different examples from the dataset used for testing:

**(Case 1)** **(dnn-classvalue in classpredictions [1 to maxrank] ).** An orchid image with ground-truth label equal to *Amerorchis rotundifolia* is correctly predicted by both DNN and BN (top 1). This shared solution added by the explanation (evidence and the expected image features are equivalent) shown in Fig. 8 support trust in the solution.

**(Case 2)** **(dnn-classvalue outside classpredictions [1 to maxrank] ).** The ground truth for this instance is *Cypripedium californicum*. The DNN predicts *Cypripedium californicum*, whereas *Cypripedium californicum* is outside the first 5 ranked BN predictions. Hence, this solution may be doubted and by considering the explanation shown in Fig. 9, the user is given information

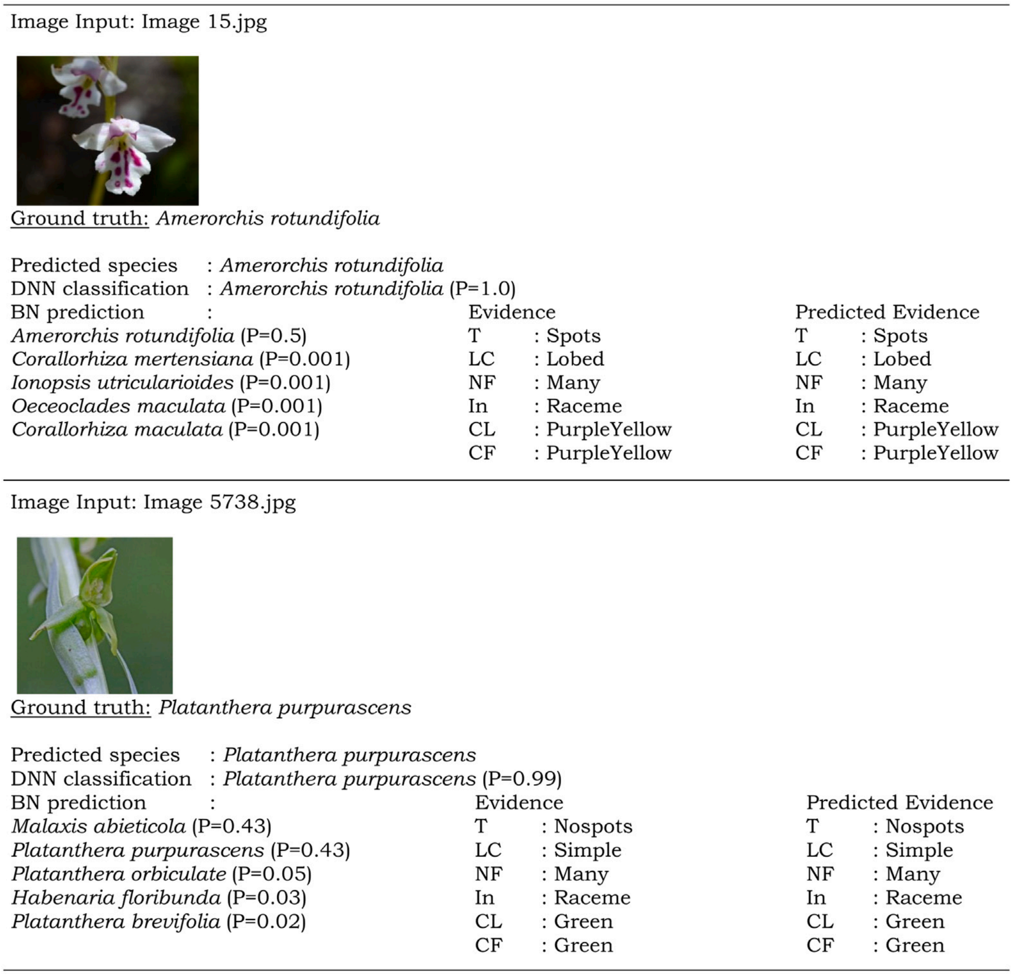


Image Input: Image 15.jpg

Ground truth: *Amerorchis rotundifolia*

Predicted species : *Amerorchis rotundifolia*
DNN classification : *Amerorchis rotundifolia* (P=1.0)

| BN prediction : | Evidence | | Predicted Evidence | |
|---|---|---|---|---|
| *Amerorchis rotundifolia* (P=0.5) | T | : Spots | T | : Spots |
| *Corallorhiza mertensiana* (P=0.001) | LC | : Lobed | LC | : Lobed |
| *Ionopsis utricularioides* (P=0.001) | NF | : Many | NF | : Many |
| *Oeceoclades maculata* (P=0.001) | In | : Raceme | In | : Raceme |
| *Corallorhiza maculata* (P=0.001) | CL | : PurpleYellow | CL | : PurpleYellow |
| | CF | : PurpleYellow | CF | : PurpleYellow |

Image Input: Image 5738.jpg

Ground truth: *Platanthera purpurascens*

Predicted species : *Platanthera purpurascens*
DNN classification : *Platanthera purpurascens* (P=0.99)

| BN prediction : | Evidence | | Predicted Evidence | |
|---|---|---|---|---|
| *Malaxis abieticola* (P=0.43) | T | : Nospots | T | : Nospots |
| *Platanthera purpurascens* (P=0.43) | LC | : Simple | LC | : Simple |
| *Platanthera orbiculate* (P=0.05) | NF | : Many | NF | : Many |
| *Habenaria floribunda* (P=0.03) | In | : Raceme | In | : Raceme |
| *Platanthera brevifolia* (P=0.02) | CL | : Green | CL | : Green |
| | CF | : Green | CF | : Green |

**Fig. 8.** Two explanations generated for case 1 instances (trustworthy cases) by the algorithm in Fig. 7. An explanation consists of a picture of the orchid in question, the predicted orchid species returned by the algorithm, the species predicted by the DNN (with probability), the first 5 ranked species predicted by the BN (again with probability attached), the feature evidence extracted by the feature extractors and used as input to the BN, and the predicted BN evidence that is compatible with the predicted species according to the DNN, being equal to the first BN solution.

to look into the matter. All the image evidence obtained by the feature classifiers corresponds to the BN's predicted values with the exception of T(exture), where the ground truth is 'Nospots', whereas the extracted feature tells that there are 'Spots'. This corroborates lack of trust in the BN predictions.

**(Case 3) (The user is not satisfied)**. If the user is not satisfied with the prediction of the DNN, the system will display the BN top 1 predictions. Fig. 10 shows an example of this case. The orchid image shown is predicted to be *Corallorhiza trifida*, with the associated extracted image features presented alongside, alongside to the expected image features. The system also displays the ground truth image from the species predicted. If the user is not satisfied, likely because there is a discrepancy between flower features shown in the picture of the flower, and the predicted feature evidence, as presented in an explanation, the system will display the solution from the BN rank 1 and also give an explanation based on the species predicted by the BN.

The time complexity of the algorithm depicted in Fig. 7 is determined by the subprocedures "ProbabilisticInference" and "Classify". As described in Section 3.6.1, time complexity of the BNs is determined by a linear-time probabilistic inference algorithm (due to the restricted structure of the BNs), followed by the application of a good sorting algorithm (e.g. heap sort) with an $O(n \log n)$ time complexity. A neural-network forward pass as implemented in the "Classify" procedure, can be considered as consisting of a sequence of vector multiplications and additions, and thus is polynomial time in terms of the dimensions of the layers and the depth of the network.

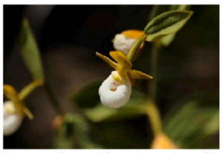
More in detail, Xception, the DNN architecture used by us, has a depth-separable convolutional architecture (depth-wise and point-wise). Let $k$ be the kernel size; $H$ the height and $W$ the width of the input feature maps; $C$ be the number of input channels; $F$ be the number of filters. Then, the time complexity for depth-wise computation is: $O(k^2 \cdot H \cdot W \cdot C \cdot F)$. The time complexity for point-wise computation is: $O(H \cdot W \cdot C \cdot F)$. The complexity for the dense layers is: $O(n \cdot m)$, and, finally, the complexity for flatten layers is equal to $O(n)$. As Xception consists of several depth-wise and point-wise layers, we have to combine several of the computations, but the result is still polynomial time. However, from a practical point of view, because of the size of the DNN, one would need a separate GPU unit for efficient computation, which at the time of writing are available even for standard laptop computers.

## 4. Experimental results

### 4.1. Dataset

We used data of 6300 images from 63 species of orchids, where each species was described by ground-truth and image-based features as described in Section 3.1; a small sample of data is shown in Table

Image Input: Image 1848.jpg



Ground truth: *Cypripedium californicum*

Predicted species : *Cypripedium californicum*
DNN classification : *Cypripedium californicum* (P=0.99)

| BN prediction : | Evidence | | Predicted Evidence | |
|---|---|---|---|---|
| *Cypripedium passerinum* (P=0.77) | T | : Spots | T | : Nospots |
| *Goodyera oblongifolia* (P=0.22) | LC | : Pouched | LC | : Pouched |
| *Goodyera pubescens* (P=0.005) | NF | : Many | NF | : Many |
| *Cypripedium candidum* (P=0.004) | In | : Raceme | In | : Raceme |
| *Epipactis palustris* (P=0.002) | CL | : Yellow | CL | : Yellow |
| | CF | : Green | CF | : Green |

Image Input: Image 6149.jpg

Ground truth: *Prosthechea cochleata*

Predicted species : *Prosthechea cochleata*
DNN classification : *Prosthechea cochleate* (P=0.99)

| BN prediction : | Evidence | | Predicted Evidence | |
|---|---|---|---|---|
| *Platanthera lacera* (P=0.56) | T | : Nospots | T | : Nospots |
| *Platanthera grandiflora* (P=0.27) | LC | : Fringed | LC | : Simple |
| *Cypripedium californicum* (P=0.08) | NF | : Many | NF | : Many |
| *Oeceoclades maculata* (P=0.04) | In | : Raceme | In | : Raceme |
| *Platanthera leucophaea* (P=0.03) | CL | : PurpleYellow | CL | : PurpleYellow |
| | CF | : Green | CF | : Green |

**Fig. 9.** Two explanations generated for case 2 instances by the algorithm in Fig. 7. See the caption of Fig. 8 for more detail. In contrast to Fig. 8, the chosen predicted species is the one corresponding to the DNN prediction.

A.3. To extract the image features that were added to the dataset that originally only contained the ground-truth features, the class variable in this experiment was balanced: each species was represented by 100 images. The dataset was split up into a training set with 5040 instances (80% of the dataset), a validation set with 630 instances (10% of the dataset) and a separate test set with 630 instances (10% of the dataset). The test set included all 63 orchids with an equivalent distribution of instances per species. The performance measure we are using below in evaluating the two types of classifier (DNN and BN) is **accuracy** with respect to the data, is defined as:

$$\text{accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \tag{4}$$

with $N_{\text{correct}}$ the number of correctly classified cases of the dataset.

### 4.2. Building orchid-identifying Bayesian networks

In our previous publication (Apriyanti et al., 2023), we carried out an ablation study with naive BNs, where groups of image features were omitted by going through parts of the power set of all 6 image features. This experiment demonstrated that including more features yielded better performance, and that all image features are needed for maximum classifier performance.

In Section 3.6 we have provided motivation for the use of a restricted form of structure learning of Bayesian networks, called semi-supervised structure learning, where the naive network part was kept fixed (by employing white-lists and black-lists, cf. Section 3.3.2). The arcs between the image feature variables were determined in four different ways (using manual design or learning). We refer to the resulting four different Bayesian network graphs as G1 to G4:

(G1) by interpreting an image feature purely as virtual evidence of the corresponding ground truth feature, which results in a double-naive network structure, presented in Fig. 11(a).

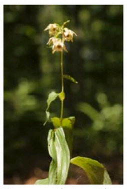
(G2) this "double-naive" Bayesian network was slightly modified by manually adding an arc between the color of flowers and the image color of the labellum, based on the idea that deep learning will not always be able to distinguish between the flower and labellum (as segmentation is not employed). The resulting semi-naive BN is depicted in Fig. 11(b).

(G3) score-based structure learning was used only with respect to the image features employing tabu search with the backbone network from Fig. 6 as input, See Fig. 11(c) for the result.

(G4) score-based structure learning was used only with respect to the image features using hill climbing and again with the backbone network from Fig. 6 as input. See Fig. 11(d) for the output.

The four Bayesian networks described above were subsequently evaluated using the data of 6300 cases described above in Section 4.1. In order to make sure that the performance results were not biased because of the selection of the test set (10% of the dataset), we employed *stratified random sampling* in combination with two different validation methods: (1) 5-fold cross-validation and (2) bootstrapping, respectively. In addition, each 5-fold cross-validation and bootstrapping run was repeated 200 times (with different stratified samples every time) to obtain information about the variability of the result. We used classification error as a measure to compare the four different Bayesian network structures, as can be seen in Fig. 12. Note that during each of the 200 runs of the cross-validation and bootstrapping algorithms the network structure remains the same, whereas the probabilistic

Image input: Image 3065.jpg



Ground truth: *Epipactis helleborine*

Predicted species : *Hexalectris spicata*
DNN classification : *Hexalectris spicata* (P=0.57)

| BN prediction : | Evidence | | Predicted Evidence | |
|---|---|---|---|---|
| *Epipactis helleborine* (P=0.97) | T | : Nospots | T | : Nospots |
| *Cypripedium fasciculatum* (P=0.02) | LC | : Pouched | LC | : Lobed |
| *Eulophia alta* (P=0.004) | NF | : Many | NF | : Many |
| *Goodyera oblongifolia* (P=0.001) | In | : Raceme | In | : Raceme |
| | CF | : Purple | CL | : PurpleYellow |
| | CF | : GreenRed | CF | : RedYellow |

Satisfied? Yes/No
If No, then Predicted species: *Epipactis helleborine*

| Evidence: | | Predicted Evidence | | |
|---|---|---|---|---|
| T | : Nospots | T | : | Nospots |
| LC | : Pouched | LC | : | Pouched |
| NF | : Many | NF | : | Many |
| In | : Raceme | In | : | Raceme |
| CL | : Purple | CL | : | Purple |
| CF | : GreenRed | CF | : | GreenRed |

Image input: Image 1217.jpg

Ground truth: *Corallorhiza trifida*

Predicted species : *Eulophia alta*
DNN classification : *Eulophia alta* (P=0.26)

| BN prediction : | Evidence | | Predicted Evidence | |
|---|---|---|---|---|
| *Corallorhiza trifida* (P=0.94) | T | : Spots | T | : Nospots |
| *Oeceoclades maculata* (P=0.03) | LC | : Lobed | LC | : Lobed |
| *Epipactis palustris* (P=0.01) | NF | : Many | NF | : Many |
| *Corallorhiza maculata* (P=0.004) | In | : Raceme | In | : Raceme |
| | CF | : Yellow | CL | : Purple |
| | CF | : GreenRed | CF | : GreenRed |

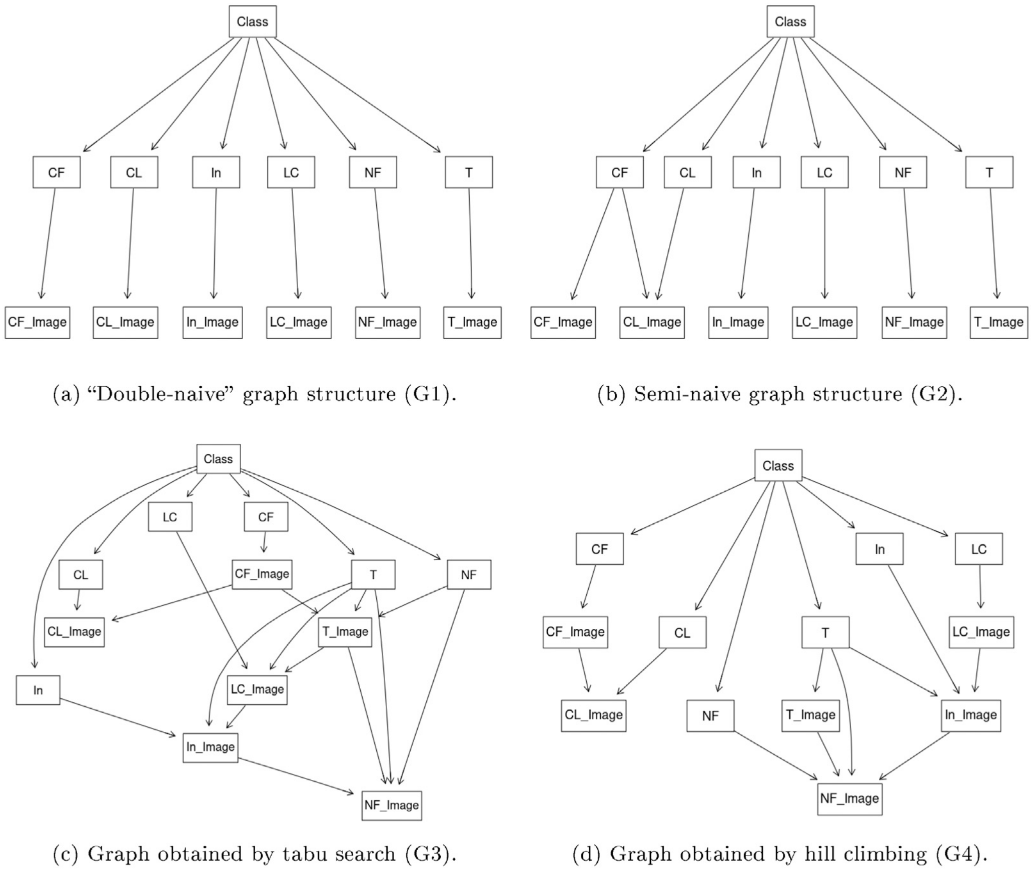
Satisfied? Yes/No
If No, then Predicted species: *Corallorhiza trifida*

| Evidence: | | Predicted Evidence | | |
|---|---|---|---|---|
| T | : Spots | T | : | Spots |
| LC | : Lobed | LC | : | Lobed |
| NF | : Many | NF | : | Many |
| In | : Raceme | In | : | Raceme |
| CL | : Yellow | CL | : | Yellow |
| CF | : Green | CF | : | Green |

**Fig. 10.** Two explanations generated for case 3 instances by the algorithm in Fig. 7. See the caption of Fig. 8 for more detail. Note that the user did not agree with the DNN solution, based on comparing the orchid picture with the predicted evidence; the BN feature evidence prediction shown next, matches the picture much better.

parameters are different, being based on a slightly different dataset, and the same applies to the test set, which also differs from run to run. As the box plots in Fig. 12 show, there is only very little variation in the classification performance for both cross-validation and bootstrapping. Nevertheless, it appears that the stratified bootstrapping produces slightly better results. Similar conclusions can be drawn about the four different Bayesian-network structures evaluated, although the best results were obtained by the networks obtained from tabu search. This network (G3) will be used in the following as our best Bayesian network.

This BN was tested on an independent test set of 630 cases (10% of the original orchid dataset). As a Bayesian network will produce a ranking of solutions, from high to low probability with sometimes equal rank because of equal posterior probability, we need to take equivalence into account in the ranking rather than just assume that probabilities are unique. It basically means that the posterior distribution of the CLASS needs to be partitioned into equivalence classes. However, as this is mostly only relevant for the solutions with maximum probability, we first determined the classification performance of the BN by simply taking the first 3, 5, and 10 ranked solutions, as shown in Table 1. The exception is the top 1 performance, where we took into account the cases of equal maximum probability. The distribution of cases with equal maximum probability was $\mu = 1.45$; s.d. $= 0.72$, thus, usually there was just a single solution or two solutions with equal maximum probability.



(a) "Double-naive" graph structure (G1).

(b) Semi-naive graph structure (G2).

(c) Graph obtained by tabu search (G3).

(d) Graph obtained by hill climbing (G4).

**Fig. 11.** Bayesian networks obtained by manual design ((a) and (b)), or by restrictive, semi-supervised structure learning ((c) and (d)).

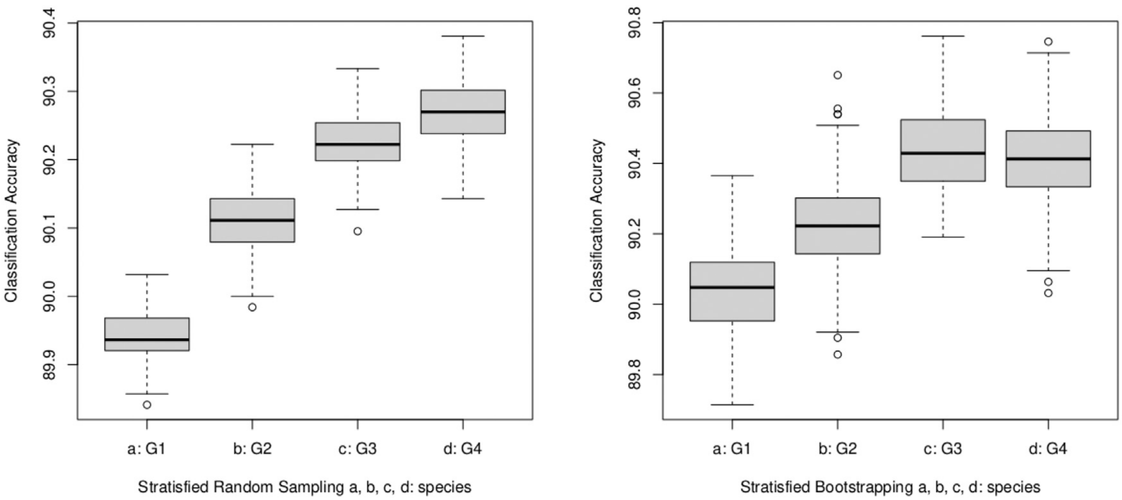


**Fig. 12.** Box plots of cross-validation and bootstrapping results obtained by 200 runs for each validation using stratified random sampling according to the class variable's distribution; a: naive BN (G1); b: semi-naive BN (G2); c: tabu-search BN (G3); d: hill-climbing BN (G4). Note that the y-axes of the two plots are not aligned.

### *4.3. Combining deep neural networks and Bayesian networks*

Using for fairness the same test set as for the Bayesian network, that was also not used as validation set for training, the whole-image deep neural network was evaluated, resulting in a classification performance of **95.1%**, which is higher than the BN up to the top 5 ranking, although there the performance becomes close. The performance of the chosen BN is less than that of the DNN due to the uncertain nature of feature extraction. Had we been able to improve the performance of the DNN feature extractors (the experimental results are reported and analyzed in our paper (Apriyanti et al., 2023)), the BN would also have yielded higher performance.

Deep learning excels at predicting the top result (rank 1). In this case, the BN also performed well with 552 cases being correct; the overlap between the two classifiers yields 530 correctly classified cases. An analysis of the results, where we combined the DNN and BN is shown in Table 2. For likely results (rank 2 or 3), there are 3.5% cases and for rank 4–5, there are 1.8% cases where results coincide. If we

**Table 1**
Classification performance of the tabu-search BN (G3 in Fig. 12) ranked solutions according to probability, using a 10% independent test set (630 cases) from the orchid dataset of 6300 cases.

| Ranked<br>Solution set | Accuracy<br>(Inclusion in set) |
|---|---|
| Top 1 | 87.6% |
| Top 3 | 91.4% |
| Top 5 | 93.5% |
| Top 10 | 96.0% |

**Table 2**
A combined analysis of the results of the DNN and BN classifiers. Note that these are not entirely the same as the results for the ensemble classifier of Fig. 7, because when combining predictions of the DNN and BN it is unknown whether the classifications are correct; this is only known after the algorithm has run. Details are provided in the main text.

| Case | Combined<br>ranked classifications | #Correct<br>classifications | Pct<br>(%) |
|---|---|---|---|
| 1 | DNN top 1, BN top 1 | 530 | 84.1 |
| 2 | DNN top 1, BN rank 2–3 | 22 | 3.5 |
| 3 | DNN top 1, BN rank 4–5 | 11 | 1.8 |
| **Subtotal 1** | Trusted agreement DNN & BN | 563 | 89.4 |
| 4 | DNN top 1, BN rank > 5 | 36 | 5.7 |
| **Subtotal 2** | DNN right | 599 | 95.1 |
| 5 | DNN wrong, BN top 1 | 22 | 3.5 |
| **Subtotal 3** | DNN or BN right | 621 | 98.6 |
| 6 | DNN wrong, BN rank 2–5 | 4 | 0.6 |
| 7 | DNN wrong, BN rank > 5 | 5 | 0.8 |
| **Total** | Explained solutions | 630 | 100 |

go beyond rank 5 for the BN predictions, the correspondence between DNN and BN is 5.7%, although clearly the BN results cannot be trusted. As may be expected, there are cases where the DNN comes up with the wrong classification, but where the BN is right; this happens in about 3.5% of cases for the top 1 results, increases to 4.1% for the top 5 ranked predictions (cases 5 and 6 in Table 2 added together). There are a few instances (0.8%) where neither model predicts the right outcome. The fact that there are cases where the DNN fails and the BN succeeds means that a combined approach offers means to leverage performance, which will be addressed next.

When using the two classifiers as an ensemble, i.e., the algorithm shown in Fig. 7, we have to experimentally determine the value for the parameter ‘maxrank’. The results obtained by the algorithm do not entire correspond to Table 2, because the ground truth is not utilized in the algorithm to select the right DNN or BN solution, simply because it should be assumed unknown and only afterwards, when the two classifier results have already been combined into one answer, it can be used for evaluation. It is for example clear that the number of cases 1 to 3 included in Table 2 will be larger than 563, because these categories will also include instances where the DNN predicts a value within the BN ranks 2–5, i.e., not top 1, and also part of case 5 from Table 2 (DNN wrong, BN top 1) will be handled as case 1 in the algorithm (with higher numbers of cases 2–5 as a result). However with the proviso that the user is able to handle the case-2 branch of the algorithm by making the right decision based on the provided explanation, and with a maxrank value equal to 5, an accuracy of **98.1%** was achieved. For this value of maxrank, a user is expected to be able to decide for each of the 59 of the 630 remaining cases whether the BN should be trusted more than the DNN, based on the explanation provided. This was the maximum performance of the algorithm, also achieved for ‘maxrank’ value equal to 4. This role of the user is clearly a limitation of the explainable ensemble classifier, although in daily practice it means that there are less than 1 in 10 cases ($\frac{59}{690} \cdot 100\%$) a user is asked to say whether or not the DNN solution can be trusted. However, a lazy solution is to simply always choose the DNN solution (guaranteed by setting ‘maxrank’ to 63), yielding an accuracy of 95.1%, the DNN’s accuracy. As we envision a clear role for explanation and trustworthiness in the decision process, there are good reasons to choose for the less lazy approach. The overall gain in performance is limited, but that cannot be said about the trust in the solutions provided by the algorithm.

### *4.4. Qualitative comparison with other XAI methods*

As mentioned above, Bayesian networks, being a representation of a joint (multivariate) probability distribution, are supplied with inference algorithms for computing any conditional probability distribution of any subset of variables. The other XAI methods described in Section 2.2 do not have this capability, and this was one reason to use BNs in our research.

However, methods such as ProtoPNet, ProtoTree, LIME, and SHAP have their merits as well, in particular when it comes to explaining the results obtained from DNNs. Next, we present a brief comparison between our proposed method with other XAI methods.

Supposed we need to classify an image of an orchid, as presented in Fig. 8. The BN gives us an explanation related to the botanic or taxonomic characteristics or features that appear in the picture, such as a spotted texture (T), a labellum (LC) that is lobed, many flowers (NF), a raceme inflorescence (In), and the colors of flower (CF) and labellum (CL) are a combination of purple and yellow. Thus, our BN-based XAI method can be considered as an image captioning method because it provides a caption or explanation of the content of an image, in this case, of an orchid flower.

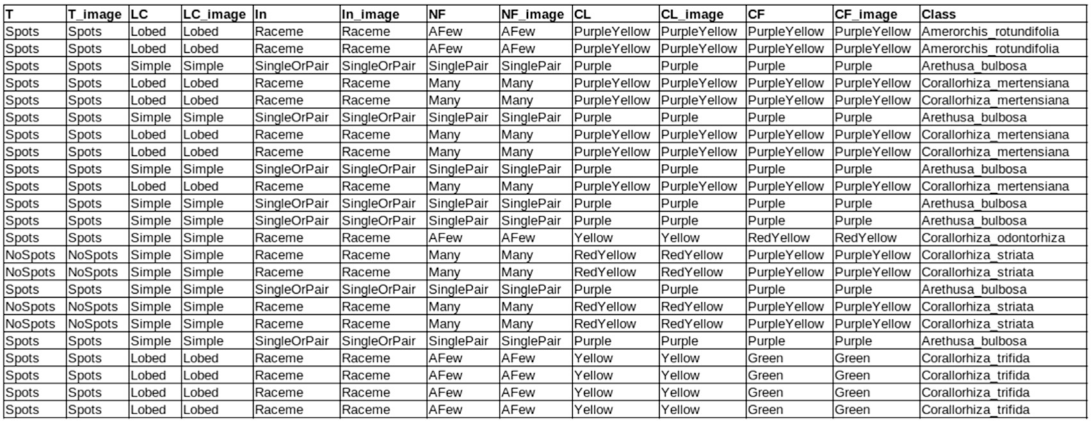
In contrast, LIME (Ribeiro et al., 2016) offers a local explanations by presenting saliency-like maps to show the reasons behind the prediction. It is easy to implement and the computational costs are low. However, based on some experimentation, LIME had difficulty in extracting relevant orchid parts and was far removed from generating an explanation in terms of orchid features as used by taxonomists.

SHAP (Lundberg and Lee, 2017) also provides an explanation by showing the flower features that are important for explaining an object in an image, which works reasonable well in this case. However, these features have again no direct relationship to the kind of taxonomic features which we wish to use in an explanation: only the important pixels are presented. Besides that, the method is computationally expensive and for large images, the kernel SHAP is slow.

ProtoPNet and ProtoTree (Chen et al., 2019; Nauta et al., 2021) offer very similar approaches, as they both learn from prototypes. ProtoPNet computes the similarity scores between the trained prototypes and the test image’s latent patches. These similarity scores are weighted and summed together to give a final score for the orchid belonging to a class. Instead of using all prototypes, ProtoTree build a decision tree to reduce the number of prototypes used in the ProtoPNet. Both methods show the patches of the images that draw attention and compare them to the learned prototypes. Both methods are more intuitive in comparison to LIME and SHAP for visualizing model decisions and are closer to human reasoning. However, the explanation consists only of patches of the image, without labels that can be interpreted as taxonomic features. Thus considerable user interpretation is still required.

To summarize, BNs have clearly advantages as an XAI method in comparison to other XAI method for the kind of applications we have considered in this paper.

**Table A.3**
Some examples of data instances of orchids of different type (indicated by the Class variable); T: texture; LC: shape of labellum; In: inflorescence; NF: number of flowers; CF: color of flower; CL: color of labellum. With '_image' is indicated the corresponding image feature extracted by a deep neural network.



| T | T_image | LC | LC_image | In | In_image | NF | NF_image | CL | CL_image | CF | CF_image | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | AFew | AFew | PurpleYellow | PurpleYellow | PurpleYellow | PurpleYellow | Amerorchis_rotundifolia |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | AFew | AFew | PurpleYellow | PurpleYellow | PurpleYellow | PurpleYellow | Amerorchis_rotundifolia |
| Spots | Spots | Simple | Simple | SingleOrPair | SingleOrPair | SinglePair | SinglePair | Purple | Purple | Purple | Purple | Arethusa_bulbosa |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | Many | Many | PurpleYellow | PurpleYellow | PurpleYellow | PurpleYellow | Corallorhiza_mertensiana |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | Many | Many | PurpleYellow | PurpleYellow | PurpleYellow | PurpleYellow | Corallorhiza_mertensiana |
| Spots | Spots | Simple | Simple | SingleOrPair | SingleOrPair | SinglePair | SinglePair | Purple | Purple | Purple | Purple | Arethusa_bulbosa |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | Many | Many | PurpleYellow | PurpleYellow | PurpleYellow | PurpleYellow | Corallorhiza_mertensiana |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | Many | Many | PurpleYellow | PurpleYellow | PurpleYellow | PurpleYellow | Corallorhiza_mertensiana |
| Spots | Spots | Simple | Simple | SingleOrPair | SingleOrPair | SinglePair | SinglePair | Purple | Purple | Purple | Purple | Arethusa_bulbosa |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | Many | Many | PurpleYellow | PurpleYellow | PurpleYellow | PurpleYellow | Corallorhiza_mertensiana |
| Spots | Spots | Simple | Simple | SingleOrPair | SingleOrPair | SinglePair | SinglePair | Purple | Purple | Purple | Purple | Arethusa_bulbosa |
| Spots | Spots | Simple | Simple | SingleOrPair | SingleOrPair | SinglePair | SinglePair | Purple | Purple | Purple | Purple | Arethusa_bulbosa |
| Spots | Spots | Simple | Simple | Raceme | Raceme | AFew | AFew | Yellow | Yellow | RedYellow | RedYellow | Corallorhiza_odontorhiza |
| NoSpots | NoSpots | Simple | Simple | Raceme | Raceme | Many | Many | RedYellow | RedYellow | PurpleYellow | PurpleYellow | Corallorhiza_striata |
| NoSpots | NoSpots | Simple | Simple | Raceme | Raceme | Many | Many | RedYellow | RedYellow | PurpleYellow | PurpleYellow | Corallorhiza_striata |
| Spots | Spots | Simple | Simple | SingleOrPair | SingleOrPair | SinglePair | SinglePair | Purple | Purple | Purple | Purple | Arethusa_bulbosa |
| NoSpots | NoSpots | Simple | Simple | Raceme | Raceme | Many | Many | RedYellow | RedYellow | PurpleYellow | PurpleYellow | Corallorhiza_striata |
| NoSpots | NoSpots | Simple | Simple | Raceme | Raceme | Many | Many | RedYellow | RedYellow | PurpleYellow | PurpleYellow | Corallorhiza_striata |
| Spots | Spots | Simple | Simple | SingleOrPair | SingleOrPair | SinglePair | SinglePair | Purple | Purple | Purple | Purple | Arethusa_bulbosa |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | AFew | AFew | Yellow | Yellow | Green | Green | Corallorhiza_trifida |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | AFew | AFew | Yellow | Yellow | Green | Green | Corallorhiza_trifida |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | AFew | AFew | Yellow | Yellow | Green | Green | Corallorhiza_trifida |
| Spots | Spots | Lobed | Lobed | Raceme | Raceme | AFew | AFew | Yellow | Yellow | Green | Green | Corallorhiza_trifida |

**Table A.4**
Orchid species covered by the developed Bayesian networks and deep neural networks.

| Species name | Species name | Species name |
|---|---|---|
| 1 Amerorchis rotundifolia | 2 Arethusa bulbosa | 3 Bletia purpurea |
| 4 Brassia caudata | 5 Calopogon barbatus | 6 Calypso bulbosa |
| 7 Cephalanthera austiniae | 8 Cleistesiopsis divaricata | 9 Coeloglossum viride |
| 10 Corallorhiza maculata | 11 Corallorhiza mertensiana | 12 Corallorhiza odontorhiza |
| 13 Corallorhiza striata | 14 Corallorhiza trifida | 15 Corallorhiza wisteriana |
| 16 Cyclopogon elatus | 17 Cypripedium acaule | 18 Cypripedium arietinum |
| 19 Cypripedium californicum | 20 Cypripedium candidum | 21 Cypripedium fasciculatum |
| 22 Cypripedium montanum | 23 Cypripedium passerinum | 24 Cypripedium reginae |
| 25 Cyrtopodium punctatum | 26 Dactylorhiza viridis | 27 Encyclia tampensis |
| 28 Epidendrum nocturnum | 29 Epipactis atrorubens | 30 Epipactis gigantea |
| 31 Epipactis helleborine | 32 Epipactis palustris | 33 Eulophia alta |
| 34 Eulophia graminea | 35 Galearis spectabilis | 36 Goodyera oblongifolia |
| 37 Goodyera pubescens | 38 Gymnadenia conopsea | 39 Habenaria floribunda |
| 40 Habenaria quinqueseta | 41 Hexalectris spicata | 42 Ionopsis utricularioides |
| 43 Malaxis abieticola | 44 Oeceoclades maculata | 45 Phaius tankervilleae |
| 46 Platanthera blephariglottis | 47 Platanthera brevifolia | 48 Platanthera chapmanii |
| 49 Platanthera dilatata | 50 Platanthera grandiflora | 51 Platanthera lacera |
| 52 Platanthera leucophaea | 53 Platanthera praeclara | 54 Platanthera psycodes |
| 55 Platanthera purpurascens | 56 Pogonia ophioglossoides | 57 Ponthieva racemosa |
| 58 Prosthechea cochleata | 59 Pseudorchis albida | 60 Sacoila lanceolata |
| 61 Spiranthes cernua | 62 Spiranthes lacera | 63 Tipularia discolor |

## 5. Discussion

The aim of our research was to develop a white-box method that is able to determine the species of an orchid from a digital photograph; achieving high accuracy and good explainability were our main goals. Although research on deep neural networks has shown repeatedly to yield remarkably good performing classifiers (Samek et al., 2021), their black-box nature has motivated researchers to focus on ways to explain neural-network output. The urgency of the matter is reflected by the creation of the new research field of "Explainable AI" (XAI). Initially it was our goal to develop a Bayesian network that would be able to do the job of orchid recognition, partly based on employing DNN feature classifiers. The reason is that a Bayesian network is able to handle the features or characteristics that human taxonomists would use in determining a plant, which offers an excellent opportunity to explain a classification to a user of the system. However, as our results show, BNs may be good classifiers, they are not optimal classifiers and have difficulty in interpreting the subtle details from images that even humans may not be able to see. Nevertheless, when properly built, a BN contains a lot of information that when combined with a DNN offers the kind of insight into a classification that one expects from an explanation. Thus, the idea of combining DNN and BNs was born.

When combining both methods – DNN and BN – the aim is to obtain a system that increases **trustworthiness** by combining two completely different methods: a whole-image classification based on computer-vision methods delivered by the DNN and a classification based on taxonomic image features extracted by feature classifying DNNs and interpreted by the BN. When both methods reach the same conclusion, it is clear that there is every reason to trust that conclusion. In the algorithm depicted in Fig. 7 that situation corresponds to when 'dnn-classvalue ∈ classpredictions[1 : maxrank]' is true. Not only does the system produce a trustworthy solution, it also explains it in a proper way using taxonomic knowledge represented in the Bayesian network.

When the solutions do not coincide, there is still the possibility (somewhere between 3.5–5.7% of cases) that only one method (BN or DNN) is right and the other (DNN or BN) is wrong, but the trustworthiness of the solution is lower. However, also in this case an explanation is provided by the Bayesian network, consisting of showing the image features corresponding to the DNN solution, the option to compare this to the extracted image features (the evidence) and ways to explore the taxonomic knowledge of the Bayesian network using alternative species (counterfactual reasoning (Pearl, 2009)). Hence, the algorithm offers a kind of decision-making with the human-in-the-loop, supported by the explanations provided by the BN. This means that although our method does not offer 100% accuracy (its maximum accuracy is **98.1%**), an explanation by the BN is always given to the user, who, supported by that, is given the opportunity to check whether disagreement between

**Table A.5**
The results of the Xception feature classifiers using multi-class and multi-label classification.

| | Features | Multi-class | | Multi-label | |
|---|---|---|---|---|---|
| | | TPR (%) | TNR (%) | TPR (%) | TNR (%) |
| T | NoSpots | 97.0 | 86.9 | 89.2 | 85.4 |
| | Spots | 86.9 | 97.0 | 79.9 | 93.6 |
| In | Panicle | 57.9 | 99.5 | 68.4 | 99.1 |
| | Raceme | 91.3 | 91.0 | 90.2 | 87.6 |
| | SingleOrPair | 94.8 | 97.9 | 95.9 | 97.9 |
| | Spike | 85.9 | 95.3 | 82.8 | 94.1 |
| NF | AFew | 75.9 | 88.1 | 71.2 | 92.8 |
| | Many | 84.2 | 86.7 | 84.2 | 87.1 |
| | SinglePair | 93.6 | 98.4 | 95.3 | 98.3 |
| LC | Fringed | 66.2 | 98.6 | 73.2 | 98.6 |
| | Lobed | 85.9 | 89.3 | 80.4 | 90.8 |
| | Pouched | 89.3 | 97.8 | 90.4 | 96.8 |
| | Simple | 87.7 | 92.7 | 83.8 | 90.9 |
| CF | Green | 87.0 | 94.0 | 86.8 | 93.9 |
| | GreenRed | 91.0 | 94.9 | 87.5 | 95.7 |
| | GreenYellow | 50.0 | 99.7 | 66.7 | 97.5 |
| | Purple | 86.0 | 98.0 | 88.7 | 97.6 |
| | PurpleYellow | 72.0 | 99.4 | 62.1 | 98.4 |
| | Red | 80.0 | 99.0 | 80.0 | 99.5 |
| | RedYellow | 90.0 | 95.4 | 95.8 | 96.8 |
| | Yellow | 84.0 | 96.6 | 82.5 | 95.8 |
| CL | Green | 82.0 | 92.8 | 90.6 | 94.3 |
| | GreenRed | 57.0 | 99.7 | 57.1 | 98.6 |
| | GreenYellow | 57.0 | 99.7 | 71.4 | 99.6 |
| | Purple | 82.0 | 97.1 | 83.2 | 96.4 |
| | PurpleYellow | 89.0 | 96.4 | 81.2 | 96.3 |
| | Red | 90.0 | 99.9 | 87.1 | 100 |
| | RedYellow | 65.0 | 99.9 | 58.8 | 99.2 |
| | Yellow | 87.0 | 87.5 | 86.2 | 91.3 |

DNN and BN may be due to the inaccuracy of one or both of these representations.

## 6. Conclusions

In this study, we develop an automated orchid species identification method that incorporates explainability by design through integrating low-level end-to-end whole-image interpretation by a deep neural network and taxonomic knowledge provided by a Bayesian network. Combining the two approaches when identifying an orchid increases the trustworthiness of the solutions, where in addition an explanation is offered for every case by the Bayesian network in understandable biological terms used in daily practice by taxonomists in determining plants. Our evaluation results, supported by stratified random sampling with cross-validation and bootstrapping, showed very little variation in the performance of the BN, indicating that our results are robust. As far as we know, the described method is new and demonstrates excellent classification performance close to 100% combined with explanations for every case in terms of taxonomic knowledge. As there is a role for the user in the classification algorithm, as described above, in future research we wish to explore whether potential users, including plant taxonomists, find the method useful and whether or not it should be further refined. Whereas we have shown in this paper that the method works in the domain of plant identification, it can be quite easily applied to other fields where deep learning is a popular method as well.

## CRediT authorship contribution statement

**Diah Harnoni Apriyanti:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Funding acquisition, Data curation, Conceptualization. **Luuk J. Spreeuwers:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Formal analysis, Conceptualization. **Peter J.F. Lucas:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The research described in this paper was funded by the Research and Innovation in Science and Technology Project (RISET-Pro) of the Ministry of Research, Technology, and Higher Education of Republic Indonesia (World Bank Loan No. 8245-ID) and supported by the Research and Innovation Agency (BRIN), Republic of Indonesia. The funder had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## Appendix

See Tables A.3–A.5.

## Data availability

Data will be made available on request.

## References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. URL https://www.tensorflow.org/.

Agarap, A.F., 2018. Deep learning using Rectified Linear Units (ReLU). http://dx.doi.org/10.48550/arXiv.1803.08375, CoRR abs/1803.08375.

Apriyanti, D.H., Spreeuwers, L.J., Lucas, P.J., 2023. Deep neural networks for explainable feature extraction in orchid identification. Appl. Intell. 53 (21), 26270–26285. http://dx.doi.org/10.1007/s10489-023-04880-2.

Apriyanti, D.H., Spreeuwers, L.J., Lucas, P.J.F., Veldhuis, R.N.J., 2021. Automated color detection in orchids using color labels and deep learning. PLoS One 16 (10), e0259036. http://dx.doi.org/10.1371/JOURNAL.PONE.0259036.

Arwatchananukul, S., Kirimasthong, K., Aunsri, N., 2020. A new paphiopedilum orchid database and its recognition using convolutional neural networks. Wirel. Pers. Commun. 115, 3275–3289. http://dx.doi.org/10.1007/s11277-020-07463-3.

Behura, N.A., Kothakota, N.J., Behera, B., Teja, S.A., Routray, S.K., Babu, R., 2024. Advancements in orchidaceae species identification: A comprehensive review of traditional and molecular methods. Int. J. Exp. Res. Rev. (IJERR) 43, 229–252. http://dx.doi.org/10.52756/ijerr.2024.v43spl.017.

Butz, R., Schulz, R., Hommersom, A., van Eekelen, M., 2022. Investigating the understandability of XAI methods for enhanced user experience: When Bayesian network users became detectives. Artif. Intell. Med. 134, 102438. http://dx.doi.org/10.1016/j.artmed.2022.102438.

Casey, F. (Ed.), 2020. Plant Taxonomy: Classical and Modern Methods. Syrawood Publishing House.

Chen, C., Li, O., Tao, C., Barnett, A.J., Su, J., Rudin, C., 2019. This looks like that: deep learning for interpretable image recognition. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. NIPS, Curran Associates Inc., Red Hook, NY, USA, pp. 8930–8941.

Chickering, D.M., 1995. A transformational characterization of equivalent Bayesian network structures. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. UAI '95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 87–98.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1800–1807. http://dx.doi.org/10.1109/CVPR.2017.195.

Chow, C., Liu, C., 1968. Approximating discrete probability distributions with dependence trees. IEEE Trans. Inform. Theory 14 (3), 462–467. http://dx.doi.org/10.1109/TIT.1968.1054142.

Colombo, D., Maathuis, M.H., 2014. Order-independent constraint-based causal structure learning. J. Mach. Learn. Res. 15, 3921–3962.

Cooper, G.F., 1990. The computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence 42 (2), 393–405. http://dx.doi.org/10.1016/0004-3702(90)90060-D.

Cowell, R., Dawid, A., Lauritzen, S., Spiegelhalter, D., 1999. Probabilistic Networks and Expert Systems. Springer, New York.

de Dombal, F., Leaper, D., Staniland, J., McAnn, A., Horrocks, J., 1972. Computer-aided diagnosis of acute abdominal pain. Br. Med. J. ii, 9–13.

de Vogel, E., Vermeulen, J., Schuiteman, A., 2025. Orchids of New Guinea. URL https://www.orchidsnewguinea.com.

Domingos, P., Pazzani, M., 1997. On the optimality of the simple Bayesian classifier under zero-one lo ss. Mach. Learn. 29, 103–130.

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. Mach. Learn. 29 (2), 131–163. http://dx.doi.org/10.1023/A:1007465528199.

Gunning, D., Aha, D.W., 2019. DARPA's explainable artificial intelligence program. AI Mag. 40 (2), 44–58. http://dx.doi.org/10.1609/aimag.v40i2.2850.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning, 2nd ed.. In: Springer Series in Statistics, Springer New York Inc..

Heckerman, D., Geiger, D., Chickering, D.M., 1995. Learning Bayesian networks: The combination of knowledge and statistical data. Mach. Learn. 20 (3), 197–243. http://dx.doi.org/10.1023/A:1022623210503.

Hiary, H., Saadeh, H., Saadeh, M., Yaqub, M., 2018. Flower classification using deep convolutional neural networks. IET Comput. Vis. 12 (6), 855–862.

Hindarto, D., Amalia, N., 2023. Implementation of flower recognition using convolutional neural networks. Int. J. Softw. Eng. Comput. Science (IJSECS) 3 (3), 341–351. http://dx.doi.org/10.35870/ijsecs.v3i3.1808.

Holzinger, A., Saranti, A., Molnar, C., Biecek, P., Samek, W., 2022. Explainable AI methods - A brief overview. In: XxAI - beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers. In: LNCS13200, Springer International Publishing, Cham, pp. 13–38. http://dx.doi.org/10.1007/978-3-031-04083-2_2.

Kahneman, D., 2011. Thinking Fast and Slow. Farrar, Straus and Giroux.

Kenny, E.M., Ford, C., Quinn, M., Keane, M.T., 2021. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies. Artificial Intelligence 294, 103459. http://dx.doi.org/10.1016/j.artint.2021.103459.

Koller, D., Friedman, N., 2009a. Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. The MIT Press.

Koller, D., Friedman, N., 2009b. Probabilistic Graphical Models: Principles and Techniques. MIT Press, California.

Lacave, C., Diez, F.J., 2002. A review of explanation methods for Bayesian networks. Knowl. Eng. Rev. 17 (2), 107–127. http://dx.doi.org/10.1017/S026988890200019X.

Lemeire, J., Steenhaut, K., Maes, S., 2008. Causal inference on data containing deterministic relations. URL https://api.semanticscholar.org/CorpusID:14787526.

Linnaeus, C., 1735. Systema Naturae, Sive Regna Tria Naturae Systematice Proposita Per Classes, Ordines, Genera, & Species. Haak, Leiden.

Liu, Z., Wang, J., Tian, Y., Dai, S., 2019. Deep learning for image-based large-flowered chrysanthemum cultivar recognition. Plant Methods 15 (1), http://dx.doi.org/10.1186/s13007-019-0532-7.

Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Del Ser, J., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., Jiang, R., Khosravi, H., Lecue, F., Malgieri, G., Páez, A., Samek, W., Schneider, J., Speith, T., Stumpf, S., 2024. Explainable artificial intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. Inf. Fusion 106, 102301. http://dx.doi.org/10.1016/j.inffus.2024.102301.

Lucas, P.J., van der Gaag, L.C., Abu-Hanna, A., 2004. Bayesian networks in biomedicine and health-care. Artif. Intell. Med. 30 (3), 201–214. http://dx.doi.org/10.1016/j.artmed.2003.11.001.

Lucas, P.J.F., van der Gaag, L.C., 1991. Principle of Expert Systems. Addison Wesley, Wokingham, England.

Lundberg, S.M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS '17, Curran Associates Inc., Red Hook, NY, USA, pp. 4768–4777.

Margaritis, D., 2003. Learning Bayesian Network Model Structure from Data (Ph.D. thesis). School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Mihaljević, B., Bielza, C., Larrañaga, P., 2021. Bayesian networks for interpretable machine learning and optimization. Neurocomputing 456, 648–665. http://dx.doi.org/10.1016/j.neucom.2021.01.138.

Nauta, M., van Bree, R., Seifert, C., 2021. Neural prototype trees for interpretable fine-grained image recognition. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 14928–14938. http://dx.doi.org/10.1109/CVPR46437.2021.01469.

Needham, C.J., Bradford, J.R., Bulpitt, A.J., Westhead, D.R., 2007. A primer on learning in Bayesian networks for computational biology. PLoS Comput. Biol. 3 (8), 1–8. http://dx.doi.org/10.1371/journal.pcbi.0030129.

Nicora, G., Catalano, M., Bortolotto at al., C., Preda, L., 2024. Bayesian networks in the management of hospital admissions: A comparison between explainable AI and black box AI during the pandemic. J. Imaging 10 (5), http://dx.doi.org/10.3390/jimaging10050117.

Nilsback, M.-E., Zisserman, A., 2008. Automated flower classification over a large number of classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing. pp. 722–729. http://dx.doi.org/10.1109/ICVGIP.2008.47.

O'Byrne, P., 2008. A to Z of South East Asian Orchid Species, first ed. Orchid Society of South East Asia.

Ou, C.-H., Hu, Y.-N., Jiang, D.-J., Liao, P.-Y., 2023. An ensemble voting method of pre-trained deep learning models for orchid recognition. In: 2023 IEEE International Systems Conference (SysCon). pp. 1–5. http://dx.doi.org/10.1109/SysCon53073.2023.10131263.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman, San Mateo, California.

Pearl, J., 2009. Causality: Models, Reasoning, and Inference, second ed. Cambride University Press.

Petkovic, D., 2023. It is not "accuracy vs. explainability"—We need both for trustworthy AI systems. IEEE Trans. Technol. Soc. 4 (1), 46–53. http://dx.doi.org/10.1109/TTS.2023.3239921.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "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. KDD '16, Association for Computing Machinery, New York, NY, USA, pp. 1135–1144. http://dx.doi.org/10.1145/2939672.2939778.

Ribeiro, M.T., Singh, S., Guestrin, C., 2018. Anchors: High-precision model-agnostic explanations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, pp. 1527–1535. http://dx.doi.org/10.1609/aaai.v32i1.11491.

Robinson, R.W., 1976. Counting unlabeled acyclic digraphs. In: Proceedings of the Fifth Australian Conference on Combinatorial Mathematics. In: LNM, Springer, vol. 622, pp. 28–43.

Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.-R., 2021. Explaining deep neural networks and beyond: A review of methods and applications. Proc. IEEE 109 (3), 247–278. http://dx.doi.org/10.1109/JPROC.2021.3060483.

Sarachai, W., Bootkrajang, J., Chaijaruwanich, J., Somhom, S., 2022. Orchid classification using homogeneous ensemble of small deep convolutional neural network. Mach. Vis. Appl. 33 (17), http://dx.doi.org/10.1007/s00138-021-01267-6.

Schiff, J.L., 2018. Rare and Exotic Orchids: Their Nature and Cultural Significance (corrected version). Springer International Publishing, Cham, p. XII, 186. http://dx.doi.org/10.1007/978-3-319-70034-2_1.

Scutari, M., 2025. Package 'bnlearn'. R Foundation for Statistical Computing (CRAN), Vienna, Austria, URL http://cran.r-project.org/web/packages/bnlearn/bnlearn.pdf.

Scutari, M., Denis, J.-B., 2021. Bayesian Networks with Examples in R, second ed. Chapman and Hall, Boca Raton, ISBN 978-0367366513.

Seeland, M., Rzanny, M., Alaqraa, N., Wäldchen, J., Mäder, P., 2017. Plant species classification using flower images: a comparative study of local feature representations. PLoS One 12 (2), 1–29. http://dx.doi.org/10.1371/journal.pone.0170629.

Shortliffe, E.H., 1976. Computer-based Medical Consultations: MYCIN. Elsevier, New York.

van Leeuwen, L., Verbrugge, R., Verheij, B., Renooij, S., 2024. Building a stronger case: Combining evidence and law in scenario-based Bayesian networks. In: Lorig, F., Tucker, J., Lindstrom, A.D., Dignum, F., Murukannaiah, P., Theodorou, A., Yolum, P. (Eds.), In: Hybrid Human AI Systems for the Social Good - Proceedings of the 3rd International Conference on Hybrid Human-Artificial Intelligence, vol. 386, IOS Press, pp. 291–299. http://dx.doi.org/10.3233/FAIA240202.

Wang, J., Wang, H., 2024. Efficiency in orchid species classification: A transfer learning-based approach. Int. J. Comput. Intell. Appl. 23 (01), 2350031. http://dx.doi.org/10.1142/S1469026823500311.

Wang, J., Wang, H., Long, Y., Lan, Y., 2024. An improved classification model based on feature fusion for orchid species. J. Electr. Eng. Technol. 19 (3), 1955–1964. http://dx.doi.org/10.1007/s42835-023-01705-7.

Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., Zhu, J., 2019. Explainable AI: A brief survey on history, research areas, approaches and challenges. In: Tang, J., Kan, M.-Y., Zhao, D., Li, S., Zan, H. (Eds.), Natural Language Processing and Chinese Computing. Springer, Cham, pp. 563–574.