scientific reports

Check for updates

OPEN

Ensembles of data-efficient vision transformers as a new paradigm for automated classification in ecology

S. P. Kyathanahally[1]✉, T. Hardeman[1], M. Reyes[1], E. Merz[1], T. Bulas[1], P. Brun[2], F. Pomati[1] & M. Baity-Jesi[1]✉

Monitoring biodiversity is paramount to manage and protect natural resources. Collecting images of organisms over large temporal or spatial scales is a promising practice to monitor the biodiversity of natural ecosystems, providing large amounts of data with minimal interference with the environment. Deep learning models are currently used to automate classification of organisms into taxonomic units. However, imprecision in these classifiers introduces a measurement noise that is difficult to control and can significantly hinder the analysis and interpretation of data. We overcome this limitation through ensembles of Data-efficient image Transformers (DeiTs), which we show can reach state-of-the-art (SOTA) performances without hyperparameter tuning, if one follows a simple fixed training schedule. We validate our results on ten ecological imaging datasets of diverse origin, ranging from plankton to birds. The performances of our EDeiTs are always comparable with the previous SOTA, even beating it in four out of ten cases. We argue that these ensemble of DeiTs perform better not because of superior single-model performances but rather due to smaller overlaps in the predictions by independent models and lower top-1 probabilities, which increases the benefit of ensembling.

Biodiversity monitoring is critical because it serves as a foundation for assessing ecosystem integrity, disturbance responses, and the effectiveness of conservation and recovery efforts[1–3]. Traditionally, biodiversity monitoring relied on empirical data collected manually[4]. This is time-consuming, labor-intensive, and costly. Moreover, such data can contain sampling biases as a result of difficulties controlling for observer subjectivity and animals' responses to observer presence[5]. These constraints severely limit our ability to estimate the abundance of natural populations and community diversity, reducing our ability to interpret their dynamics and interactions. Counting wildlife by humans has a tendency to greatly underestimate the number of individuals present[6,7]. Furthermore, population estimates based on extrapolation from a small number of point counts are subject to substantial uncertainties and may fail to represent the spatio-temporal variation in ecological interactions (e.g. predator-prey), leading to incorrect predictions or extrapolations[7,8]. While human-based data collection has a long history in providing the foundation for much of our knowledge of where and why animals dwell and how they interact, present difficulties in wildlife ecology and conservation are revealing the limitations of traditional monitoring methods[7].

Recent improvements in imaging technology have dramatically increased the data-gathering capacity by lowering costs and widening the scope and coverage compared to traditional approaches, opening up new paths for large-scale ecological studies[7]. Many formerly inaccessible places of conservation interest may now be examined by using high-resolution remote sensing[9], and digital technologies such as camera traps[10–12] are collecting vast volumes of data non-invasively. Camera traps are low-cost, simple to set up, and provide high-resolution image sequences of the species that set them off, allowing researchers to identify the animal species, their behavior, and interactions including predator-prey, competition and facilitation. Several cameras have already been used to monitor biodiversity around the world, including underwater systems[13,14], making camera traps one of the most widely-used sensors[12]. In biodiversity conservation initiatives, camera trap imaging is quickly becoming the gold standard[10,11], as it enables for unparalleled precision monitoring across enormous expanses of land.

However, people find it challenging to analyze the massive amounts of data provided by these devices. The enormous volume of image data generated by modern gathering technologies for ecological studies is too large to

[1]Eawag, Überlandstrasse 133, 8600 Dübendorf, Switzerland. [2]WSL, Zürcherstrasse 111, 8903 Birmensdorf, Switzerland. ✉email: sreenath.kyathanahally@eawag.ch; marco.baityjesi@eawag.ch

be processed and analyzed at scale to derive compelling ecological conclusions[15]. Although online crowd-sourcing platforms could be used to annotate images[16], such systems are unsustainable due to the exponential expansion in data acquisition and to the insufficient expert knowledge that is most often required for the annotation. In other words, we need tools that can automatically extract relevant information from the data and help to reliably understand how ecological processes act across space and time.

Machine learning has proven to be a suitable methodology to unravel the ecological insights from massive amounts of data[17]. Detection and counting pipelines have evolved from imprecise extrapolations from manual counts to machine learning-based systems with high detection rates[18–20]. Using deep learning (DL) to detect and classify species for the purpose of population estimation is becoming increasingly common[18–27]. DL models, most often with convolutional neural network (CNN) like architectures[18–20,22,24,26], have been the standard thus far in biodiversity monitoring. Although these models have an acceptable performance, they often unreliably detect minority classes[22], require a very well-tailored model selection and training, large amounts of data[20], and have a non-negligible error rate that negatively influences the modeling and interpretation of the outcoming data. Thereupon, it is argued that many DL-based monitoring systems cannot be deployed in a fully autonomous way if one wants to ensure a reliable-enough classification[28,29].

Recently, following their success in natural language processing applications[30], transformer architectures were adapted to computer vision applications. The resulting structures, known as vision transformers (ViTs)[31], differ from CNN-based models, that use image pixels as units of information, in using image patches, and employing an attention mechanism to weigh the importance of each part of the input data differently. Vision transformers have demonstrated encouraging results in several computer vision tasks, outperforming the state of the art (SOTA) in several paradigmatic datasets, and paving the way for new research areas within the branch of deep learning.

In this article, we use a specific kind of ViTs, Data efficient image Transformers (DeiTs)[32], for the classification of biodiversity images such as plankton, coral reefs, insects, birds and large animals (though our approach can also be applied in different domains). We show that while the single-model performance of DeiTs matches that of alternative approaches, ensembles of DeiTs (EDeiTs) achieve very good performances without requiring any hyperparameter tuning. We see that this mainly happens because of a higher disagreement in the predictions, with respect to other model classes, between independent DeiT models.

Results

A new state of the art. We trained EDeiTs on several ecological datasets, spanning from microorganisms to large animals, including images in color as well as in black-and-white, with and without background; and including datasets of diverse sizes and with varying numbers of classes, both balanced and unbalanced. Details on the datasets are provided in section "Data". As shown in Fig. 1, the error rates of EDeiTs are sometimes close to or even smaller than those of previous SOTA. In the SI, we provide a detailed comparison between our models' accuracy and F1-score and that of the previous SOTA. Details on models and training are provided in sections "Models", "Implementation" and "Ensemble learning".

Individual models comparison. We now show that the better performance of EDeiTs is not a property of the single models, but that it rather stems from the ensembling step. To do this, we focus on the ZooLake dataset where the previous state of the art is an ensemble of CNN models [22] that consisted of EfficientNet, MobileNet and DenseNet architectures. In Table 1, we show the single-model performances of these architectures, and those of

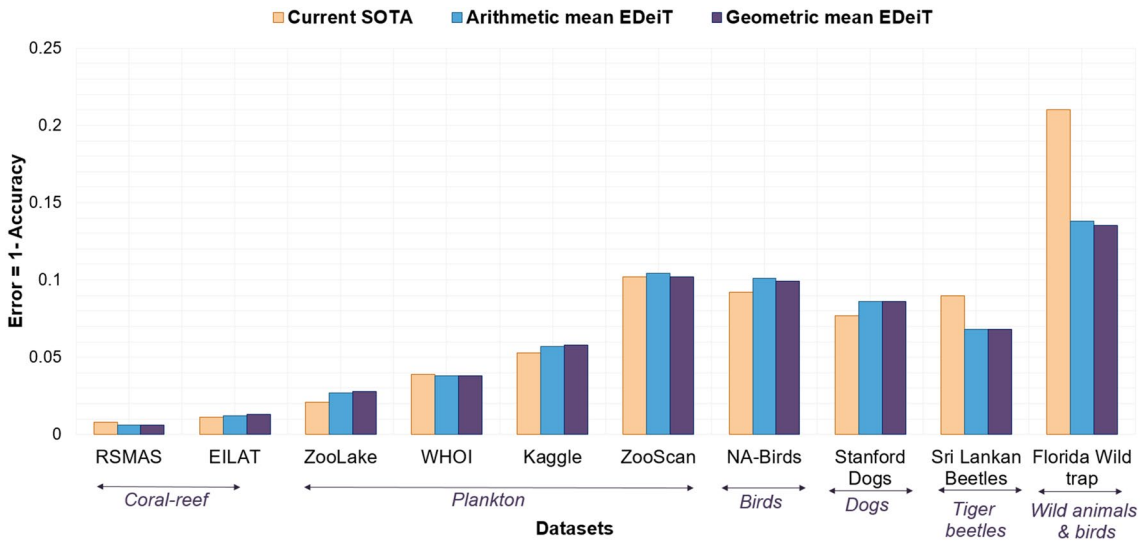

Figure 1. Comparing EDeiTs to the previous SOTA. For each dataset, we show the error, which is the fraction of misclassified test images (1 − accuracy). The error of the existing SOTA model is shown in orange. For the ensembles of DeiTs, we show two ways of combining the individual learnings: through arithmetic (blue) and geometric (purple) averaging.

Model	No. of params for each model	Accuracy mean	F1-score mean	Arithmetic ensemble (accuracy/F1-score)	Geometric ensemble (accuracy/F1-score)
Dense121	8.1M	0.965 (3)	0.86 (1)	0.976/0.916	0.977/0.917
Efficient-B2	9.2M	0.9670 (4)	0.894 (2)	0.975/0.915	0.975/0.914
Efficient-B5	30.6M	0.964 (2)	0.87 (1)	0.971/0.891	0.971/0.898
Efficient-B6	43.3M	0.965 (1)	0.880 (7)	0.971/0.904	0.972/0.906
Efficient-B7	66.0M	0.968 (1)	0.893 (4)	0.974/0.913	0.974/0.920
Mobile-V2	3.5M	0.961 (2)	0.881 (5)	0.971/0.907	0.973/0.909
Best_6_avg	–	–	–	0.978/0.924	0.977/0.923
DeiT-Base	85.8M	0.962 (3)	0.899 (2)	0.973/0.924	0.972/0.922

Table 1. Summary of the performance of the individual models on the ZooLake dataset. The ensemble score on the rightmost column is obtained by averaging across either 3 or 4 different initial conditions. The Best_6_avg model is an ensemble of DenseNet121, EfficientNet-B2, EfficientNet-B5, EfficientNet-B6, EfficientNet-B7 and MobileNet (combining learners through an arithmetic mean) models[22]. The numbers in parentheses are the standard errors, referred to the last significant digit.

the DeiT-Base model ("Implementation" section), which is the one we used for the results in Fig. 1. The accuracies and (macro-averaged) F1-scores of the two families of models (CNN and DeiT) when compared individually are in a similar range: the accuracies are between 0.96 and 0.97, and the F1-scores between 0.86 and 0.90.

Ensemble comparison. We train each of the CNNs in Table 1 four times (as described in Ref.[22]), with different realisations of the initial conditions, and show their arithmetic average ensemble and geometric average ensemble ("Ensemble learning" section) in the last two columns. We also show the performance of the ensemble model developed in Ref.[22], which ensembles over the six shown CNN architectures. We compare those with the ensembled DeiT-Base model, obtained through arithmetic average ensemble and geometric average ensemble over three different initial conditions of the model weights.

As can be expected, upon ensembling the individual model performance improves sensibly. However, the improvement is not the same across all models. The CNN family reaches a maximum F1-score ≤ 0.920 for ensemble of Efficient-B7 network across initial conditions. When the best CNNs are picked and ensembled the ensemble performance (Best_6_avg) reaches F1-score ≤ 0.924 . In the case of DeiT models, the ensemble was carried out without picking the best model across different DeiTs but still reaches similar classification accuracy (with the F1-score reaching 0.924) with no hyperparameter tuning.

Why DeiT models ensemble better. To understand the better performance of DeiTs upon ensembling, we compare CNNs with DeiTs when ensembling over three models. For CNNs, we take the best EfficientNet-B7, MobileNet and Dense121 models from Ref.[22] (each had the best validation performance from 4 independent runs). For DeiTs, we train a DeiT-Base model three times (with different initial weight configurations) and ensemble over those three.

Since the only thing that average ensembling takes into account is the confidence vectors of the models, we identify two possible reasons why EDeiTs perform better, despite the single-model performance being equal to CNNs:

(a) Different CNN models tend to agree on the same wrong answer more often than DeiTs.
(b) The confidence profile of the DeiT predictions is better suited for average ensembling than the other models.

We will see that both (a) and (b) are true, though the dominant contribution comes from (a). In Fig. 2a we show a histogram of how many models gave a right (R) or wrong (W) prediction (*e.g.* RRR denotes three correct predictions within the individual models, RRW denotes one mistake, and so on).

On Fig. 2b and c, we show the same quantity, but restricted to the examples that were correctly classified by the arithmetic and geometric averaged ensemble models. The CNN ensemble has more RRR cases (2523) than the EDeiT (2515), but when the three models have some disagreement, the EDeiTs catch up with the CNN ensembles

In particular:

The correct RWW cases are 2.0x more common in the geometric average and arithmetic average EDeiT (Geometric CNN: 8, Geometric EDeiT: 15; Arithmetic CNN: 8, Arithmetic EDeiT: 16). In the SI (See Footnote 1) we show that the probability that a RWW ensembling results in a correct prediction depends on the ratio between the second and third component of the ensembled confidence vector, and that the better performance of DeiT ensembles in this situation is justified by the shape of the confidence vector.We thus measure the mutual agreement between different models. To do so, we take the confidence vectors, $\vec{c}_0$, $\vec{c}_1$ and $\vec{c}_2$ of the three models, and calculate the similarity

$$S = \frac{1}{3}(\vec{c}_0 \cdot \vec{c}_1 + \vec{c}_0 \cdot \vec{c}_2 + \vec{c}_1 \cdot \vec{c}_2)\,, \tag{1}$$

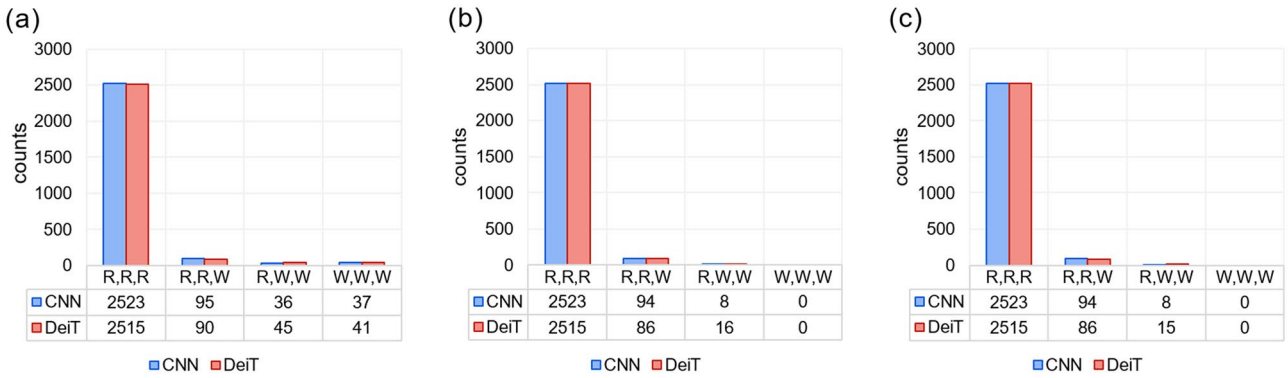

Figure 2. Comparison between three-model ensemble models based on CNNs and on DeiTs on the ZooLake test set. The bar heights indicate how often each combination (RRR, RRW, RWW, WWW) appeared. RRR indicates that all the models gave the right answer, RRW means that one model gave a wrong answer, and so on. The numbers below each bar indicate explicitly the height of the bar. On panel (**a**) we consider the whole test set, on panel (**b**) we only consider the examples which were correctly classified by the *arithmetic* ensemble average, and on panel (**c**) those correctly classified through *geometric* ensemble average.

averaged over the full test set. For DeiTs, we have $S = 0.799 \pm 0.004$, while for CNNs the similarity is much higher, $S = 0.945 \pm 0.003$. This is independent of which CNN models we use. If we ensemble Eff2, Eff5 and Eff6, we obtain $S = 0.948 \pm 0.003$. Note that the lower correlation between predictions from different DeiT learners is even more striking given that we are comparing the *same* DeiT model trained three times, with *different* CNN architectures. This suggests that the CNN predictions focus on similar sets of characteristics of the image, so when they fail, all models fail similarly. On the contrary, the predictions of separate DeiTs are more independent. Given a fixed budget of single-model correct answers, RWW combinations result more likely in a correct answer when the two wrong answers are different (see SI (See Footnote 1)). The situation is analogous for geometric averaging (Fig. 2c).

Comparison to vanilla ViTs : For completeness, in the SI (See Footnote 1) we also provide a comparison between DeiTs[32] and vanilla ViTs[31]. Also here, we find analogous results: despite the single-model performance being similar, DeiTs ensemble better, and this can be again attributed to the lower similarity between predictions coming from independent models. This suggests that the better performance of DeiT ensembles is not related to the attention mechanism of ViTs, but rather of the distillation process which is characteristic of DeiTs ("Models" section).

Discussion

We presented Ensembles of Data Efficient Image Transformers (EDeiTs) as a standard go-to method for image classification. Though the method we presented is valid for any kind of images, we provided a proof of concept of its validity with biodiversity images. Besides being of simple training and deployment (we performed no specific tuning for any of the datasets), EDeiTs achieve results comparable to those of earlier carefully tuned state-of the-art methods, and even outperform them in classifying biodiversity images in four of the ten datasets.

Focusing on a single dataset, we compared DeiT with CNN models (analogous results stem from a comparison with vanilla ViTs). Despite the similar performance of individual CNN and DeiT models, ensembling benefits DeiTs to a larger extent. We attributed this to two mechanisms. To a minor extent, the confidence vectors of DeiTs are less peaked on the highest value, which has a slight benefit on ensembling. To a major extent, independently of the architecture, the predictions of CNN models are very similar to each other (independently of whether the prediction is wrong or right), whereas different DeiTs have a lower degree of mutual agreement, which turns out beneficial towards ensembling. This greater independence between DeiT learners also suggests that the loss landscape of DeiTs is qualitatively different from that of CNNs, and that DeiTs might be particularly suitable for algorithms that average the model weights throughout learning, such as stochastic weighted averaging[33], since different weight configurations seem to interpret the image in a different way.

Unlike many kinds of ViTs, the DeiT models we used have a similar number of parameters compared to CNNs, and the computational power required to train them is similar. In addition to their deployment requiring similar efforts, with higher performances, DeiTs have the additional advantage of being more straightforwardly interpretable than CNNs by ecologists, because of the attention map that characterizes transformers. The attention mechanism allows to effortlessly identify where in the image the model focused its attention (Fig. 3), rendering DeiTs more transparent and controllable by end users.

All these observations pose EDeiTs as a solid go-to method for the classification of ecology monitoring images. Though EDeiTs are likely to be an equally solid method also in different domains, we do not expect EDeiTs to beat the state of the art in mainstream datasets such as CIFAR[34] or ImageNet[35]. In fact, for such datasets, immense efforts were made to achieve the state of the art, the top architectures are heavily tailored to these datasets[36], and their training required huge numerical efforts. Even reusing those same top architectures, it is hard to achieve high single-model performances with simple training protocols and moderate computational resources. In addition, while ensembling provides benefits[37], well-tailored architectural choices can provide the

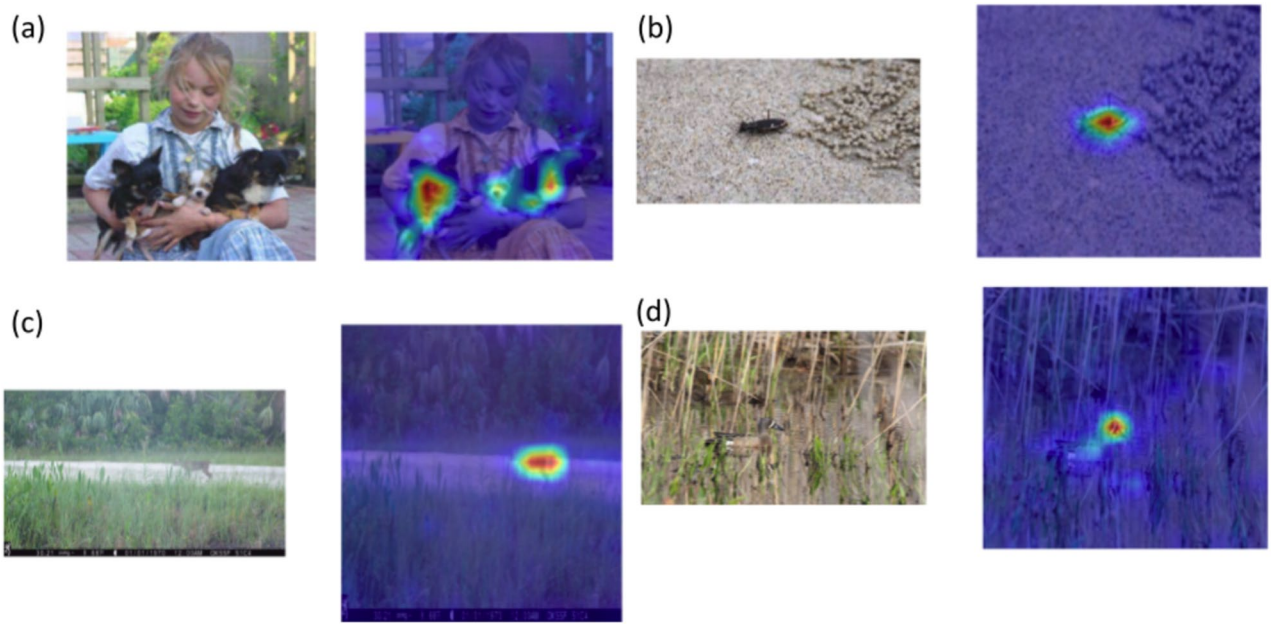

Figure 3. Examples of DeiTs identifying images from different datasets: (**a**) Stanford Dogs, (**b**) SriLankan tiger beetles, (**c**) Florida wild-trap, and (**d**) NA-Birds datasets are visualized. The original image is shown on the left in each panel, while the right reveals where our model is paying attention while classifying the species in the image.

same benefits[38]. Therefore, it is expected that the SOTA models trained on these datasets will benefit less from ensembling.

Finally, we note that the nominal test performance of machine learning models is often subject to a decrease when the models are deployed on real world data. This phenomenon, called *data shift*, can be essentially attributed to the fact that the data sets often do not adequately represent the distribution of images that is sampled at the moment of deployment[39]. This can be due to various reasons (sampling method, instrument degradation, seasonal effects, an so on) and is hard to harness. However, it was recently shown that Vision Transformer models (here, ViT and DeiT) are more robust to data shift[40–42] and to other kinds of perturbations such as occlusions[41], which is a further reason for the deployment of EDeiTs in ecological monitoring.

Methods

Data. We tested our models on ten publicly available datasets. In Fig. 4 we show examples of images from each of the datasets. When applicable, the training and test splits were kept the same as in the original dataset. For example, the ZooScan, Kaggle, EILAT, and RSMAS datasets lack a specific training and test set; in these cases, benchmarks come from k-fold cross-validation[43,44], and we followed the exact same procedures in order to allow for a fair comparison.

RSMAS This is a small coral dataset of 766 RGB image patches with a size of 256 × 256 pixels each[45]. The patches were cropped out of bigger images obtained by the University of Miami's Rosenstiel School of Marine

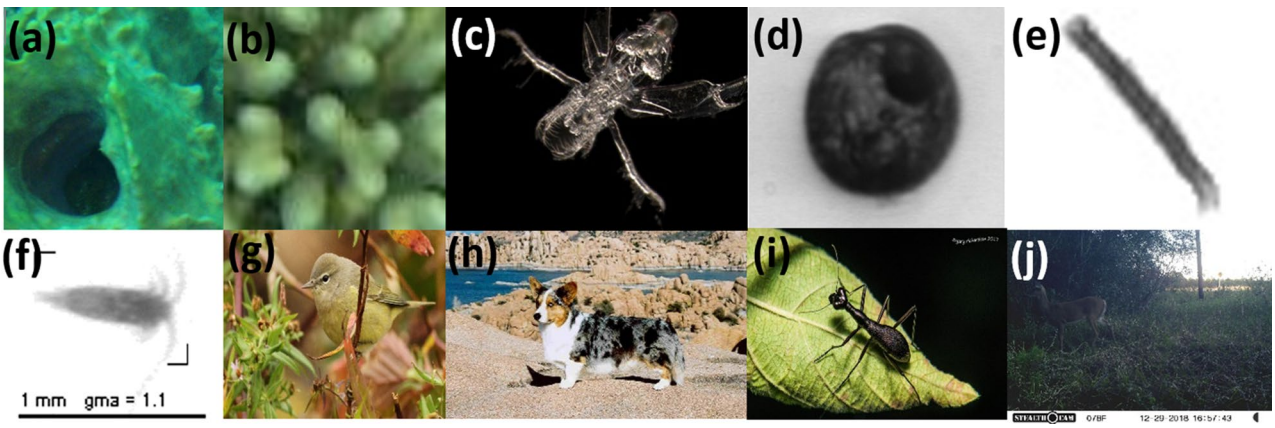

Figure 4. Examples of images from each of the datasets.(**a**) RSMAS (**b**) EILAT (**c**) ZooLake (**d**) WHOI (**e**) Kaggle (**f**) ZooScan (**g**) NA-Birds (**h**) Stanford dogs (**i**) SriLankan Beetles (**j**) Florida Wildtrap.

and Atmospheric Sciences. These images were captured using various cameras in various locations. The data is separated into 14 unbalanced groups and whose labels correspond to the names of the coral species in Latin. The current SOTA for the classification of this dataset is by[44]. They use the ensemble of best performing 11 CNN models. The best models were chosen based on sequential forward feature selection (SFFS) approach. Since an independent test is not available, they make use of 5-fold cross-validation for benchmarking the performances.

EILAT This is a coral dataset of 1123 64-pixel RGB image patches[45] that were created from larger images that were taken from coral reefs near Eilat in the Red sea. The image dataset is partitioned into eight classes, with an unequal distribution of data. The names of the classes correspond to the shorter version of the scientific names of the coral species. The current SOTA[44] for the classification of this dataset uses the ensemble of best performing 11 CNN models similar to RSMAS dataset and 5-fold cross-validation for benchmarking the performances.

ZooLake This dataset consists of 17943 images of lake plankton from 35 classes, acquired using a Dual-magnification Scripps Plankton Camera (DSPC) in Lake Greifensee (Switzerland) between 2018 and 2020 [14,46]. The images are colored, with a black background and an uneven class distribution. The current SOTA[22] on this dataset is based on a stacking ensemble of 6 CNN models on an independent test set.

WHOI This dataset [47] contains images of marine plankton acquired by Image FlowCytobot[48], from Woods Hole Harbor water. The sampling was done between late fall and early spring in 2004 and 2005. It contains 6600 greyscale images of different sizes, from 22 manually categorized plankton classes with an equal number of samples for each class. The majority of the classes belonging to phytoplankton at genus level. This dataset was later extended to include 3.4M images and 103 classes. The WHOI subset that we use was previously used for benchmarking plankton classification models[43,44]. The current SOTA[22] on this dataset is based on average ensemble of 6 CNN models on an independent test set.

Kaggle-plankton The original Kaggle-plankton dataset consists of plankton images that were acquired by In-situ Ichthyoplankton Imaging System (ISIIS) technology from May to June 2014 in the Straits of Florida. The dataset was published on Kaggle (https://www.kaggle.com/c/datasciencebowl) with images originating from the Hatfield Marine Science Center at Oregon State University. A subset of the original Kaggle-plankton dataset was published by[43] to benchmark the plankton classification tasks. This subset comprises of 14,374 greyscale images from 38 classes, and the distribution among classes is not uniform, but each class has at least 100 samples. The current SOTA[22] uses average ensemble of 6 CNN models and benchmarks the performance using 5-fold cross-validation.

ZooScan The ZooScan dataset consists of 3771 greyscale plankton images acquired using the Zooscan technology from the Bay of Villefranche-sur-mer[49]. This dataset was used for benchmarking the classification models in previous plankton recognition papers[43,44]. The dataset consists of 20 classes with a variable number of samples for each class ranging from 28 to 427. The current SOTA[22] uses average ensemble of 6 CNN models and benchmarks the performance using 2-fold cross-validation.

NA-Birds NA-Birds[50] is a collection of 48,000 captioned pictures of North America's 400 most often seen bird species. For each species, there are over 100 images accessible, with distinct annotations for males, females, and juveniles, totaling 555 visual categories. The current SOTA[51] called TransFG modifies the pure ViT model by adding contrastive feature learning and part selection module that replaces the original input sequence to the transformer layer with tokens corresponding to informative regions such that the distance of representations between confusing subcategories can be enlarged. They make use of an independent test set for benchmarking the model performances.

Stanford Dogs The Stanford Dogs dataset comprises 20,580 color images of 120 different dog breeds from all around the globe, separated into 12,000 training images and 8,580 testing images[52]. The current SOTA[51] makes use of modified ViT model called TransFG as explained above in NA-Birds dataset. They make use of an independent test set for benchmarking the model performances.

Sri Lankan Beetles The arboreal tiger beetle data[53] consists of 380 images that were taken between August 2017 and September 2020 from 22 places in Sri Lanka, including all climatic zones and provinces, as well as 14 districts. *Tricondyla* (3 species), *Derocrania* (5 species), and *Neocollyris* (1 species) were among the nine species discovered, with six of them being endemic . The current SOTA[53] makes use of CNN-based SqueezeNet architecture and was trained using pre-trained weights of ImageNet. The benchmarking of the model performances was done on an independent test set.

Florida Wild Traps The wildlife camera trap[54] classification dataset comprises 104,495 images with visually similar species, varied lighting conditions, skewed class distribution, and samples of endangered species, such as Florida panthers. These were collected from two locations in Southwestern Florida. These images are categorized in to 22 classes. The current SOTA[54] makes use of CNN-based ResNet-50 architecture and the performance of the model was benchmarked on an independent test set.

Models. Vision transformers (ViTs)[31] are an adaptation to computer vision of the Transformers, which were originally developed for natural language processing[30]. Their distinguishing feature is that, instead of exploiting translational symmetry, as CNNs do, they have an *attention mechanism* which identifies the most relevant part of an image. ViTs have recently outperformed CNNs in image classification tasks where vast amounts of training data and processing resources are available[30,55]. However, for the vast majority of use cases and consumers, where data and/or computational resources are limiting, ViTs are essentially untrainable, even when the network architecture is defined and no architectural optimization is required. To settle this issue, Data-efficient Image Transformers (DeiTs) were proposed[32]. These are transformer models that are designed to be trained with much less data and with far less computing resources[32]. In DeiTs, the transformer architecture has been modified to allow native distillation[56], in which a student neural network learns from the results of a teacher model. Here, a

CNN is used as the teacher model, and the pure vision transformer is used as the student network. All the DeiT models we report on here are DeiT-Base models[32]. The ViTs are ViT-B16, ViT-B32, and ViT-L32 models[31].

Implementation. To train our models, we used transfer learning[57]: we took a model that was already pre-trained on the ImageNet[35] dataset, changed the last layers depending on the number of classes, and then fine-tuned the whole network with a very low learning rate. All the models were trained with two Nvidia GTX 2080Ti GPUs.

DeiTs We used DeiT-Base[32] architecture, using the Python package TIMM[58], which includes many of the well-known deep learning architectures, along with their pre-trained weights computed from the ImageNet dataset[35]. We resized the input images to 224 x 224 pixels and then, to prevent the model from overfitting at the pixel level and help it generalize better, we employed typical image augmentations during training such as horizontal and vertical flips, rotations up to 180 degrees, small zoom up's to 20%, a small Gaussian blur, and shearing up to 10%. To handle class imbalance, we used class reweighting, which reweights errors on each example by how present that class is in the dataset[59]. We used sklearn utilities[60] to calculate the class weights which we employed during the training phase.

The training phase started with a default pytorch[61] initial conditions (Kaiming uniform initializer), an AdamW optimizer with cosine annealing[62], with a base learning rate of 10^{-4}, and a weight decay value of 0.03, batch size of 32 and was supervised using cross-entropy loss. We trained with early stopping, interrupting training if the validation F1-score did not improve for 5 epochs. The learning rate was then dropped by a factor of 10. We iterated until the learning rate reached its final value of 10^{-6}. This procedure amounted to around 100 epochs in total, independent of the dataset. The training time varied depending on the size of the datasets. It ranged between 20min (SriLankan Beetles) to 9h (Florida Wildtrap). We used the same procedure for all the datasets: no extra time was needed for hyperparameter tuning.

ViTs We implemented the ViT-B16, ViT-B32 and ViT-L32 models using the Python package vit-keras (https://github.com/faustomorales/vit-keras), which includes pre-trained weights computed from the ImageNet[35] dataset and the Tensorflow library[63].

First, we resized input images to 128×128 and employed typical image augmentations during training such as horizontal and vertical flips, rotations up to 180 degrees, small zooms up to 20%, small Gaussian blur, and shearing up to 10%. To handle class imbalance, we calculated the class weights and use them during the training phase.

Using transfer learning, we imported the pre-trained model and froze all of the layers to train the model. We removed the last layer, and in its place we added a dense layer with n_c outputs (being n_c the number of classes), was preceded and followed by a dropout layer. We used the Keras-tuner[64] with Bayesian optimization search[65] to determine the best set of hyperparameters, which included the dropout rate, learning-rate, and dense layer parameters (10 trials and 100 epochs). After that, the model with the best hyperparameters was trained with a default tensorflow[63] initial condition (Glorot uniform initializer) for 150 epochs using early stopping, which involved halting the training if the validation loss did not decrease after 50 epochs and retaining the model parameters that had the lowest validation loss.

CNNs CNNs included DenseNet[66], MobileNet[67], EfficientNet-B2[68], EfficientNet-B5[68], EfficientNet-B6[68], and EfficientNet-B7[68] architectures. We followed the training procedure described in Ref.[22], and carried out the training in tensorflow.

Ensemble learning. We adopted average ensembling, which takes the confidence vectors of different learners, and produces a prediction based on the average among the confidence vectors. With this procedure, all the individual models contribute equally to the final prediction, irrespective of their validation performance. Ensembling usually results in superior overall classification metrics and model robustness[69,70].

Given a set of n models, with prediction vectors $\vec{c}_i$ $(i = 1, \ldots, n)$, these are typically aggregated through an arithmetic average. The components of the ensembled confidence vector $\vec{c}_{AA}$, related to each class α are then

$$c_{AA,\alpha} = \frac{1}{n} \sum_{i=1}^{n} c_{i,\alpha} \,. \tag{2}$$

Another option is to use a geometric average,

$$c_{GA,\alpha} = \sqrt[n]{\prod_{i=1}^{n} c_{i,\alpha}} \,. \tag{3}$$

We can normalize the vector $\vec{c}_g$, but this is not relevant, since we are interested in its largest component, $\max_{\alpha}(c_{GA,\alpha})$, and normalization affects all the components in the same way. As a matter of fact, also the nth root does not change the relative magnitude of the components, so instead of $\vec{c}_{GA}$ we can use a product rule: $\max_{\alpha}(c_{GA,\alpha}) = \max_{\alpha}(c_{PROD,\alpha})$, with $c_{PROD,\alpha} = \prod_{i=1}^{n} c_{i,\alpha}$.

While these two kinds of averaging are equivalent in the case of two models and two classes, they are generally different in any other case[71]. For example, it can easily be seen that the geometric average penalizes more strongly the classes for which at least one learner has a very low confidence value, a property that was termed *veto* mechanism[72] (note that, while in Ref.[72] the term *veto* is used when the confidence value is exactly zero, here we use this term in a slightly looser way).

Data availability

All the data we used is open access. The datasets analysed during the current study are available in the repositories, that we indicate in "Data" section.

Code availability

The code for the reproduction of our results is available at https://github.com/kspruthviraj/Plankiformer.

Received: 22 April 2022; Accepted: 5 October 2022
Published online: 03 November 2022

References

1. Kremen, C., Merenlender, A. M. & Murphy, D. D. Ecological monitoring: A vital need for integrated conservation and development programs in the tropics. *Conserv. Biol.* **8**, 388–397 (1994).
2. Jetz, W. *et al.* Essential biodiversity variables for mapping and monitoring species populations. *Nat. Ecol. Evol.* **3**, 539–551 (2019).
3. Kühl, H. S. *et al.* Effective biodiversity monitoring needs a culture of integration. *One Earth* **3**, 462–474. https://doi.org/10.1016/j.oneear.2020.09.010 (2020).
4. Witmer, G. Wildlife population monitoring: Some practical considerations. *Wildl. Res.*https://doi.org/10.1071/WR04003 *(2005).*
5. McEvoy, J. F., Hall, G. P. & McDonald, P. G. Evaluation of unmanned aerial vehicle shape, flight path and camera type for waterfowl surveys: Disturbance effects and species recognition. *Peer J.* **4**, e1831–e1831. https://doi.org/10.7717/peerj.1831 (2016).
6. Hodgson, J. C. *et al.* Drones count wildlife more accurately and precisely than humans. *Methods Ecol. Evol.* **9**, 1160–1167. https://doi.org/10.1111/2041-210X.12974 (2018).
7. Tuia, D. *et al.* Perspectives in machine learning for wildlife conservation. *Nat. Commun.* **13**, 792. https://doi.org/10.1038/s41467-022-27980-y (2022).
8. Soranno, P. A. *et al.* Cross-scale interactions: Quantifying multi-scaled cause-effect relationships in macrosystems. *Front. Ecol. Environ.* **12**, 65–73. https://doi.org/10.1890/120366 (2014).
9. Luque, S., Pettorelli, N., Vihervaara, P. & Wegmann, M. Improving biodiversity monitoring using satellite remote sensing to provide solutions towards the 2020 conservation targets. *Methods Ecol. Evol.* **9**, 1784–1786. https://doi.org/10.1111/2041-210X.13057 (2018).
10. Burton, A. C. *et al.* Review: Wildlife camera trapping: A review and recommendations for linking surveys to ecological processes. *J. Appl. Ecol.* **52**, 675–685. https://doi.org/10.1111/1365-2664.12432 (2015).
11. Rowcliffe, J. M. & Carbone, C. Surveys using camera traps: Are we looking to a brighter future? *Anim. Conserv.* **11**, 185–186. https://doi.org/10.1111/j.1469-1795.2008.00180.x (2008).
12. Steenweg, R. *et al.* Scaling-up camera traps: Monitoring the planet's biodiversity with networks of remote sensors. *Front. Ecol. Environ.* **15**, 26–34. https://doi.org/10.1002/fee.1448 (2017).
13. Orenstein, E. C. *et al.* The scripps plankton camera system: A framework and platform for in situ microscopy. *Limnol. Oceanogr. Methods* **18**, 681–695. https://doi.org/10.1002/lom3.10394 (2020).
14. Merz, E. *et al.* Underwater dual-magnification imaging for automated lake plankton monitoring. *Water Res.* **203**, 117524. https://doi.org/10.1101/2021.04.14.439767 (2021).
15. Farley, S. S., Dawson, A., Goring, S. J. & Williams, J. W. Situating ecology as a big-data science: Current advances, challenges, and solutions. *Bioscience* **68**, 563–576. https://doi.org/10.1093/biosci/biy068 (2018).
16. Jamison, E. & Gurevych, I. Noise or additional information? Leveraging crowdsource annotation item agreement for natural language tasks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* pp 291–297 (2015).
17. Kwok, R. Ai empowers conservation biology. *Nature* **567**, 133–134. https://doi.org/10.1038/d41586-019-00746-1 (2019).
18. Norouzzadeh, M. S. *et al.* Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proc. Natl. Acad. Sci.* **115**, E5716–E5725. https://doi.org/10.1073/pnas.1719367115 (2018).
19. Willi, M. *et al.* Identifying animal species in camera trap images using deep learning and citizen science. *Methods Ecol. Evol.* **10**, 80–91. https://doi.org/10.1111/2041-210X.13099 (2019).
20. Tabak, M. A. *et al.* Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods Ecol. Evol.* **10**, 585–590. https://doi.org/10.1111/2041-210X.13120 (2019).
21. Henrichs, D. W., Anglès, S., Gaonkar, C. C. & Campbell, L. Application of a convolutional neural network to improve automated early warning of harmful algal blooms. *Environ. Sci. Pollut. Res.* pp 1–12 (2021).
22. Kyathanahally, S. P. *et al.* Deep learning classification of lake zooplankton. *Front. Microbiol.*https://doi.org/10.3389/fmicb.2021.746297 *(2021).*
23. Py, O., Hong, H., & Zhongzhi, S. Plankton classification with deep convolutional neural networks. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference* pp 132–136. https://doi.org/10.1109/ITNEC.2016.7560334 (2016).
24. Dai, J., Yu, Z., Zheng, H., Zheng, B. & Wang, N. A hybrid convolutional neural network for plankton classification. In Chen, C.-S., Lu, J. & Ma, K.-K. (eds.) Computer Vision – ACCV 2016 Workshops, 102–114 (Springer International Publishing, Cham, 2017).
25. Lee, H., Park, M. & Kim, J. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp 3713–3717. https://doi.org/10.1109/ICIP.2016.7533053 (2016).
26. Luo, J. Y. *et al.* Automated plankton image analysis using convolutional neural networks. *Limnol. Oceanogr. Methods* **16**, 814–827. https://doi.org/10.1002/lom3.10285 (2018).
27. Islam, S. B. & Valles, D. Identification of wild species in texas from camera-trap images using deep neural network for conservation monitoring. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)* pp 0537–0542, https://doi.org/10.1109/CCWC47524.2020.9031190 (2020).
28. Green, S. E., Rees, J. P., Stephens, P. A., Hill, R. A. & Giordano, A. J. Innovations in camera trapping technology and approaches: The integration of citizen science and artificial intelligence. *Animals*https://doi.org/10.3390/ani10010132 *(2020).*
29. Schneider, S., Greenberg, S., Taylor, G. W. & Kremer, S. C. Three critical factors affecting automated image species recognition performance for camera traps. *Ecol. Evol.* **10**, 3503–3517. https://doi.org/10.1002/ece3.6147 (2020).
30. Vaswani, A. et al. Attention is all you need. CoRR arXiv:1706.03762 (2017).
31. Dosovitskiy, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. CoRR arXiv:2010.11929 (2020).
32. Touvron, H. et al. Training data-efficient image transformers & distillation through attention. CoRR arXiv:2012.12877 (2020).
33. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. & Wilson, A. G. Averaging weights leads to wider optima and better generalization (2018).
34. Krizhevsky, A. Learning multiple layers of features from tiny images. (2009). https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

35. Deng, J. et al. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* pp 248–255 (Ieee, 2009).
36. Recht, B., Roelofs, R., Schmidt, L. & Shankar, V. Do imagenet classifiers generalize to imagenet?. In *International Conference on Machine Learning* pp 5389–5400 (PMLR, 2019).
37. d'Ascoli, S., Refinetti, M., Biroli, G. & Krzakala, F. Double trouble in double descent: Bias and variance (s) in the lazy regime. In *International Conference on Machine Learning* pp 2280–2290 (PMLR, 2020).
38. Nakkiran, P., Venkat, P., Kakade, S. & Ma, T. Optimal regularization can mitigate double descent. arXiv preprint arXiv:2003.01897 (2020).
39. Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V. & Herrera, F. A unifying view on dataset shift in classification. *Pattern Recogn.* **45**, 521–530. https://doi.org/10.1016/j.patcog.2011.06.019 (2012).
40. Minderer, M. *et al.* Revisiting the calibration of modern neural networks. *Adv. Neural. Inf. Process. Syst.* **34**, 15682–15694 (2021).
41. Naseer, M. M. *et al.* Intriguing properties of vision transformers. *Adv. Neural. Inf. Process. Syst.* **34**, 23296–23308 (2021).
42. Paul, S. & Chen, P.-Y. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence* **36**, pp 2071–2081 (2022).
43. Zheng, H. *et al.* Automatic plankton image classification combining multiple view features via multiple kernel learning. *BMC Bioinf.* **18**, 570. https://doi.org/10.1186/s12859-017-1954-8 (2017).
44. Lumini, A., Nanni, L. & Maguolo, G. Deep learning for plankton and coral classification. *Appl. Comput. Inform.* https://doi.org/10.1016/j.aci.2019.11.004 (2020).
45. Gómez-Ríos, A. *et al.* Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation. *Expert Syst. Appl.* **118**, 315–328. https://doi.org/10.1016/j.eswa.2018.10.010 (2019).
46. Kyathanahally, S. *et al.* Data for: Deep learning classification of lake zooplankton. *Front. Microbiol.*https://doi.org/10.25678/0004DY *(2021).*
47. Sosik, H. & Olson, R. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. *Limnol. Oceanogr. Methods* **5**, 204–216 (2007).
48. Olson, R. J. & Sosik, H. M. A submersible imaging-in-flow instrument to analyze nano-and microplankton: Imaging flowcytobot. *Limnol. Oceanogr. Methods* **5**, 195–203. https://doi.org/10.4319/lom.2007.5.195 (2007).
49. Gorsky, G. *et al.* Digital zooplankton image analysis using the ZooScan integrated system. *J. Plankton Res.* **32**, 285–303. https://doi.org/10.1093/plankt/fbp124 (2010).
50. Van Horn, G. et al. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp 595–604. https://doi.org/10.1109/CVPR.2015.7298658 (2015).
51. He, J., *et al.* Transfg: A transformer architecture for fine-grained recognition. CoRR arXiv:2103.07976 (2021).
52. Khosla, A., Jayadevaprakash, N., Yao, B. & Fei-Fei, L. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition* (Colorado Springs, CO, 2011).
53. Abeywardhana, D., Dangalle, C., Nugaliyadde, A. & Mallawarachchi, Y. Deep learning approach to classify tiger beetles of Sri Lanka. *Eco. Inform.* **62**, 101286. https://doi.org/10.1016/j.ecoinf.2021.101286 (2021).
54. Gagne, C., Kini, J., Smith, D. & Shah, M. Florida wildlife camera trap dataset. CoRR arXiv:2106.12628 (2021).
55. Xu, Y., Zhang, Q., Zhang, J. & Tao, D. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. CoRR arXiv:2106.03348 (2021).
56. Allen-Zhu, Z. & Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. CoRR arXiv:2012.09816 (2020).
57. Tan, C. et al. A survey on deep transfer learning. In *International conference on artificial neural networks* pp 270–279 (Springer, 2018).
58. Torch image models (2022). Available at https://fastai.github.io/timmdocs/.
59. Johnson, J. M. & Khoshgoftaar, T. M. Survey on deep learning with class imbalance. *J. Big Data* **6**, 1–54 (2019).
60. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
61. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32**, 8024–8035 (2019).
62. Loshchilov, I. & Hutter, F. Fixing weight decay regularization in adam. CoRR **abs/1711.05101** (2017).
63. Abadi, M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. CoRR **abs/1603.04467** (2016).
64. O'Malley, T. et al. Keras Tuner. https://github.com/keras-team/keras-tuner (2019).
65. Mockus, J. *Bayesian Approach to Global Optimization: Theory and Applications* Vol. 37 (Springer Science & Business Media, 2012).
66. Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks (2018).
67. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp 4510–4520. https://doi.org/10.1109/CVPR.2018.00474 (2018).
68. Tan, M. & Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* pp 6105–6114 (PMLR, 2019).
69. Seni, G. & Elder, J. *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions* Vol. 2 (Morgan & Claypool Publishers, 2010).
70. Zhang, C. & Ma, Y. *Ensemble Machine Learning: Methods and Applications* (Springer, 2012).
71. Alexandre, L. A., Campilho, A. C. & Kamel, M. On combining classifiers using sum and product rules. *Pattern Recogn. Lett.* **22**, 1283–1289 (2001).
72. Tax, D. M., Duin, R. P. & Breukelen, M. V. Comparison between product and mean classifier combination rules. In In Proc. Workshop on Statistical Pattern Recognition, 165–170 (1997).

Acknowledgements

This project was funded by the Eawag DF project Big-Data Workflow (#5221.00492.999.01), the Swiss Federal Office for the Environment (contract Nr Q392-1149) and the Swiss National Science Foundation (project 182124).

Author contributions

M.B.J. designed the study, S.K. mined the data, S.K. built the models, S.K., T.H., E.M., T.B., M.R., P.B., F.P. and M.B.J. were actively involved in the discussion while building and improving the models and data, S.K. and M.B.J. wrote the paper. All the authors contributed to the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-21910-0.

Correspondence and requests for materials should be addressed to S.P.K. or M.B.-J.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.