Deep Learning and Continual Learning Techniques for Plant Image Analysis

Submitted by

Muhammad Sohaib Younis, M.Sc.

This dissertation is submitted for the degree of

Doctor of Natural Sciences

Department of Mathematics and Computer Science

University of Marburg

First reviewer: Prof. Dr. Bernhard Seeger

Second reviewer: Prof. Dr. Dominik Heider

Date of Submission: 20. February 2024
Date of Disputation: 19. April 2024
Place of publication: Marburg
Year of publication: 2024
University code: 1180

Declaration

I hereby declare that this thesis is my independent work without any external assistance. All the sources or aiding material utilized for writing the thesis have been appropriately acknowledged through proper citations. The research mentioned in this thesis was conducted exclusively for my PhD at the University of Marburg. Furthermore, I confirm that this is my first attempt at a doctoral degree and this thesis has not been previously submitted, either in its present or similar form, to any other academic institution to obtain a degree or qualification.

Muhammad Sohaib Younis
Munich 24. April 2024

Acknowledgements

First and foremost, I express gratitude to Allah the Almighty, the Most Gracious, and the Most Merciful for His assistance and for granting me the opportunity and determination to bring this work to fruition.

I would like to express my deepest appreciation to my supervisor, Prof. Dr. Bernhard Seeger for his steadfast support, patience, and guidance, throughout my doctoral journey, particularly over the past three years. Additionally, I am profoundly grateful to Prof. Dr. Thomas Hickler for facilitating my research initiation and doctoral pursuit, as well as to Dr. Claus Weiland and Dr. Marco Schmidt for their invaluable mentorship and collaboration. Recognition is also owed to the German Research Foundation (DFG) and Philipps University of Marburg for funding my research.

I extend my gratitude to all my educators, from my early years in kindergarten to my Ph.D. studies; their teachings have been instrumental throughout my academic development. My utmost appreciation is reserved for my family, especially my parents, for their unwavering support, love, and prayers, without whom I would not have been able to reach this milestone.

Abstract

The research presented in this thesis addresses the application of deep learning on digital images, particularly plant images. The exponential growth of publicly available image datasets, mainly due to the wide accessibility of smartphones and digital cameras, has sparked a surge in deep learning research across various domains. Online platforms like iNaturalist, GBIF, and Zooniverse offer hundreds of millions of images, including digitized herbarium scans from museums and collections worldwide. These serve as invaluable resources for ecological and biodiversity research. While plant images from natural environments can provide an excellent resource for studying species distributions and ecological traits, herbarium scans offer additional advantages, such as analysis of visual and structural plant features in a standardized format relevant for analyzing phenological traits of species spanning hundreds of years. This thesis presents innovative methods for species recognition, trait extraction, and plant organ detection by leveraging novel deep learning techniques for image recognition and object detection. While recognizing the successful implementation of these approaches, the thesis also highlights crucial challenges such as data imbalance and limited availability of labeled datasets. The thesis addresses these challenges and proposes an innovative, data-free continual learning approach for training a model on continuously arriving data while also mitigating data imbalance. This approach enables the integration of new data of unknown distribution into existing models while preserving the previously learned knowledge without access to the prior data. Through a combination of practical deep learning applications and theoretical insights, the research presented in this thesis contributes significantly to advancements in ecological research and continual learning.

Zusammenfassung

Die in dieser Arbeit vorgestellte Forschung befasst sich mit der Anwendung von Deep Learning auf digitale Bilder, insbesondere Pflanzenbilder. Das exponentielle Wachstum öffentlich verfügbarer Bilddatensätze, das vor allem auf die breite Verfügbarkeit von Smartphones und Digitalkameras zurückzuführen ist, hat zu einem starken Anstieg der Deep-Learning-Forschung in verschiedenen Bereichen geführt. Online-Plattformen wie iNaturalist, GBIF und Zooniverse bieten Hunderte von Millionen von Bildern, darunter digitalisierte Herbarbelege aus Museen und Sammlungen weltweit. Diese dienen als unschätzbare Ressourcen für die ökologische und Biodiversitätsforschung. Während Pflanzenbilder aus natürlichen Umgebungen eine hervorragende Ressource für die Untersuchung von Artenverteilungen und ökologischen Merkmalen darstellen, bieten eingescannte Herbarbelege zusätzliche Vorteile, wie die Analyse visueller und struktureller Pflanzenmerkmale in einem standardisierten Format, die für die Analyse phänologischer Merkmale von Arten über Hunderte von Jahren hinweg verwendbar ist. In dieser Arbeit werden innovative Methoden zur Erkennung von Arten, zur Extraktion von Merkmalen und zur Erkennung von Pflanzenorganen vorgestellt, indem neuartige Deep-Learning-Techniken zur Bilderkennung und Objekterkennung eingesetzt werden. Neben der erfolgreichen Umsetzung dieser Ansätze werden in dieser Arbeit auch entscheidende Herausforderungen wie die Unausgewogenheit der Daten und die begrenzte Verfügbarkeit von markierten Datensätzen aufgezeigt. Die Arbeit befasst sich mit diesen Herausforderungen und schlägt einen innovativen, datenfreien Ansatz für kontinuierliches Lernen vor, um ein Modell auf kontinuierlich anfallenden Daten zu trainieren und gleichzeitig das Datenungleichgewicht zu verringern. Dieser Ansatz ermöglicht es, neue Daten mit unbekannter Verteilung in bestehende Modelle zu integrieren und gleichzeitig das zuvor gelernte Wissen zu bewahren, ohne auf die vorherigen Daten zugreifen zu müssen. Durch eine Kombination aus praktischen Deep-Learning-Anwendungen und theoretischen Erkenntnissen trägt die in dieser Arbeit vorgestellten Ergebnise wesentlich zu Fortschritten in der ökologischen Forschung und im kontinuierlichen Lernen bei.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AP Average Precision

CLS Complementary Learning System

CNN Convolutional Neural Network

DFCIL Data-free class incremental learning

DFGR Data-Free Generative Replay

ELU Exponential Linear Unit

EWC Elastic Weight Consolidation

Faster R-CNN Faster region-based CNN

FLOPO Flora Phenotype Ontology

FPN Feature Pyramid Network

GAN Generative Adversarial Network

GBIF Global Biodiversity Information Facility

GPU Graphics Processing Unit

ILSVRC ImageNet Large Scale Visual Recognition Challenge

LwF Learning without Forgetting

LwM Learning without Memorizing

MAS Memory Aware Synapses

MFGR Memory-free generative replay

MLP Multilayer Perceptron

MNHN Muséum national d'Histoire naturelle

NMS Non-Maximum Suppression

OBO Open Biomedical Ontologies

ReLU Rectified Linear Unit

ResNet Residual Network

RGB Red Green Blue (colours)

ROI Region of Interest

RPN Region proposal network

SI Synaptic Intelligence

SSD Single Shot MultiBox Detector

VAE Variational Autoencoder

VGG Visual Geometry Group

YOLO You Only Look Once

List of publications

Main publications

- **Younis, S.**, Weiland, C., Hoehndorf, R., Dressler, S., Hickler, T., Seeger, B. & Schmidt, M. Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks. *Botany Letters*. **165**, 377-383 (2018) doi:10.1080/23818107.2018.1446357

- **Younis, S.**, Schmidt, M., Weiland, C., Dressler, S., Seeger, B. & Hickler, T. Detection and annotation of plant organs from digitised herbarium scans using deep learning. *Biodiversity Data Journal*. **8** (2020) doi:10.3897/BDJ.8.e57090

- **Younis, S.** & Seeger, B. Data-Free Generative Replay for Class-Incremental Learning on Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*. (Submitted)

Supporting publications

- **Younis, S.**, Schmidt, M., Weiland, C., Dressler, S., Tautenhahn, S., Kattge, J., Seeger, B., Hickler, T. & Hoehndorf, R. Extracting Trait Data from Digitized Herbarium Specimens Using Deep Convolutional Networks. (Friedrich-Schiller-Universität Jena, 2018) doi:10.22032/dbt.37836

- **Younis, S.**, Schmidt, M., Seeger, B., Hickler, T. & Weiland, C. A workflow for data extraction from digitized herbarium specimens. *Biodiversity Information Science And Standards*. **3** pp. e35190 (2019) doi:10.3897/biss.3.35190

Other contributions not included

- Grieb, J., Weiland, C., Hardisty, A., Addink, W., Islam, S., **Younis, S.** & Schmidt, M. Machine learning as a service for DiSSCo's digital specimen architecture. (Pensoft, 2021) doi: 10.3897/biss.5.75634

- Möglich, J., Lampe, P., Fickus, M., **Younis, S.**, Gottwald, J., Nauss, T., Brandl, R., Brändle, M., Friess, N., Freisleben, B. & Others Towards reliable estimates of abundance trends using automated non-lethal moth traps. *Insect Conservation And Diversity*. (2023) doi: 10.1111/icad.12662

- Zeuss, D., Bald, L., Gottwald, J., Becker, M., Bellafkir, H., Bendix, J., Bengel, P., Beumer, L., Brandl, R., Brändle, M., Dahlke, S., Farwig, N., Freisleben, B., Friess, N., Heidrich, L., Heuer, S., Höchst, J., Holzmann, H., Lampe, P., Leberecht, M., Lindner, K., Masello, J., Möglich, J., Mühling, M., Müller, T., Noskov, A., Opgenoorth, L., Peter, C., Quillfeldt, P., Rösner, S., Royauté, R., Runge, C., Schabo, D., Schneider, D., Seeger, B., Shayle, E., Steinmetz, R., Tafo, P., Vogelbacher, M., Wöllauer, S., **Younis, S.**, Zobel, J. & Nauss, T. Nature 4.0: A networked sensor system for integrated biodiversity monitoring. *Global Change Biology*. (2024) doi: 10.1111/gcb.17056

1. Introduction

1.1 Background and motivation

There has been a significant rise in the number of in digital images in recent years mostly due to the widespread accessibility of smartphones and digital cameras. In particular, an increasing number of amateur photographers has led to huge image collections related to nature, especially plants, animals, and insects. The wealth of these images captured by researchers and citizen scientists have been collected in various repositories of ecological data and made available online, such as iNaturalist (675,000 images of 5,000 species [107]) and the Zooniverse (1.2 million images of 40 species [71]). Furthermore, the digitization efforts by many museums and collectors worldwide have also resulted in large digital datasets of numerous specimens, especially plants. For example iDigBio portal hosted over 1.8 million georeferenced images of vascular plant specimens as of 2017 [112] and the GBIF platform has more than 27 million plant specimen records with images, the vast majority of these images being herbarium scans [23].

These massive repositories for digital images of plants and other organisms, including herbarium scans, serve as invaluable resources for ecological research. Images captured in the natural environment are invaluable for studying species distribution, phonology and ecological traits. However, digitized herbarium scans present distinctive advantages for analyzing the structure and visual traits. For centuries, herbarium collections have been instrumental in botanical research and academia, documenting the species and other visual traits of the specimen. According to some estimates, around 3000 herbaria have accumulated around 400 million specimens [101]. This is mainly because herbarium specimens typically follow a standard format collection by fixing a dried and preserved specimen on a white A3 size sheet. These herbarium specimens are also accompanied by labels containing their scientific names, traits, locations, and collector notes. Another advantage digitized herbarium scans offer is their large image size, due to being scanned by high-resolution cameras, making it easier for computers to detect species and trait information.

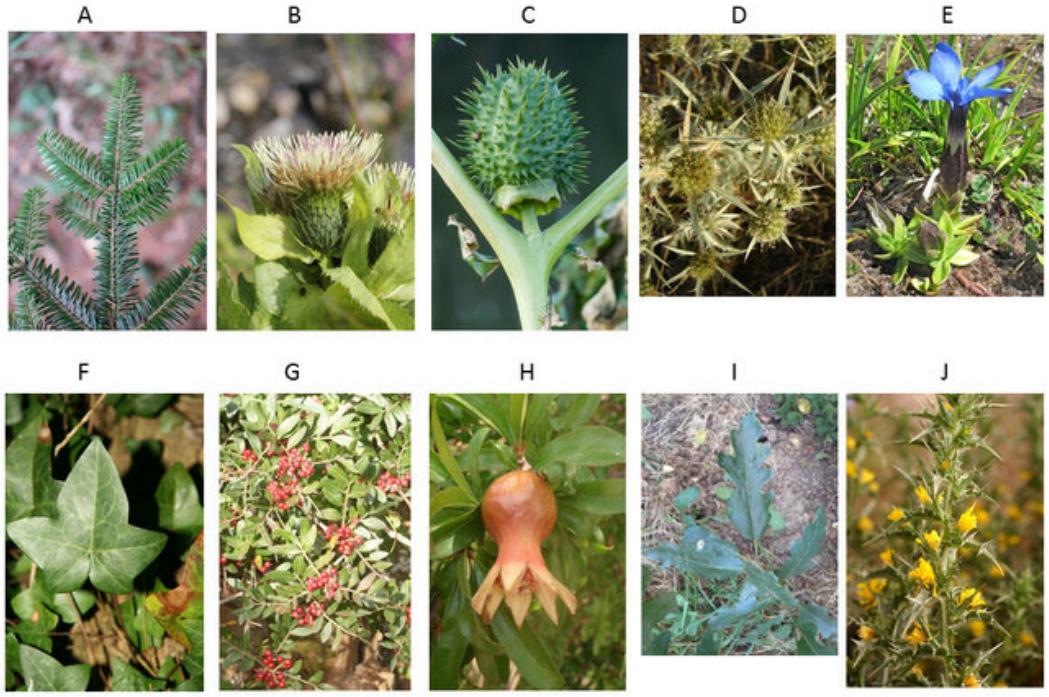


Fig. 1.1 Images of different species used in the PlantCLEF dataset [45].

The proliferation of digital images has given rise to a lot of computer vision and traditional machine learning techniques for feature detection and trait extraction [123]. With the increasing availability of huge image datasets, the evolution of deep learning has heralded a new era in the scientific community for automating species identification and trait recognition [91, 7]. The data-centric approach of deep learning, as opposed to traditional data modeling and feature engineering techniques, has enabled the development of models capable of making highly accurate predictions on natural images, camera trap images, and digitized herbarium scans.

The incorporation of deep learning methods in recognizing plant species from images and herbarium scans has demonstrated remarkable success [27, 109]. There have also been some deep learning techniques proposed for identifying plant phenotypes and traits. The accessibility of millions of digitized herbarium specimens online has facilitated the application of deep learning algorithms, hence accelerating species detection, facilitating trait recognition, and enhancing our understanding of biodiversity. However, before the first publication of the thesis [118], the utilization of deep learning on herbarium specimens was limited to only a few contributions in this domain [94, 9]. Our research proposes a few methods of recognizing plant species, identifying their traits and also detecting multiple plant organs from herbarium scans using deep learning.

Real-world datasets of natural organisms, including those specially curated for competitions like PlantCLEF [27], often exhibit inherent imbalances due to the naturally occurring uneven distribution of species or lack of sufficient samples. This imbalance significantly affects the performance of deep learning, particularly for trait recognition and classification tasks where the inter-class differences are visually minor. The application of data augmentation and transfer learning can help overcome the limitations imposed by imbalanced datasets. However, traditional data augmentation techniques are ineffective for addressing high-class imbalance on large datasets [42], such as plant images encompassing numerous species. Our novel approach addresses these limitations of data imbalance by using a combination of data augmentation, by a generative model, with an innovative data rebalancing method.

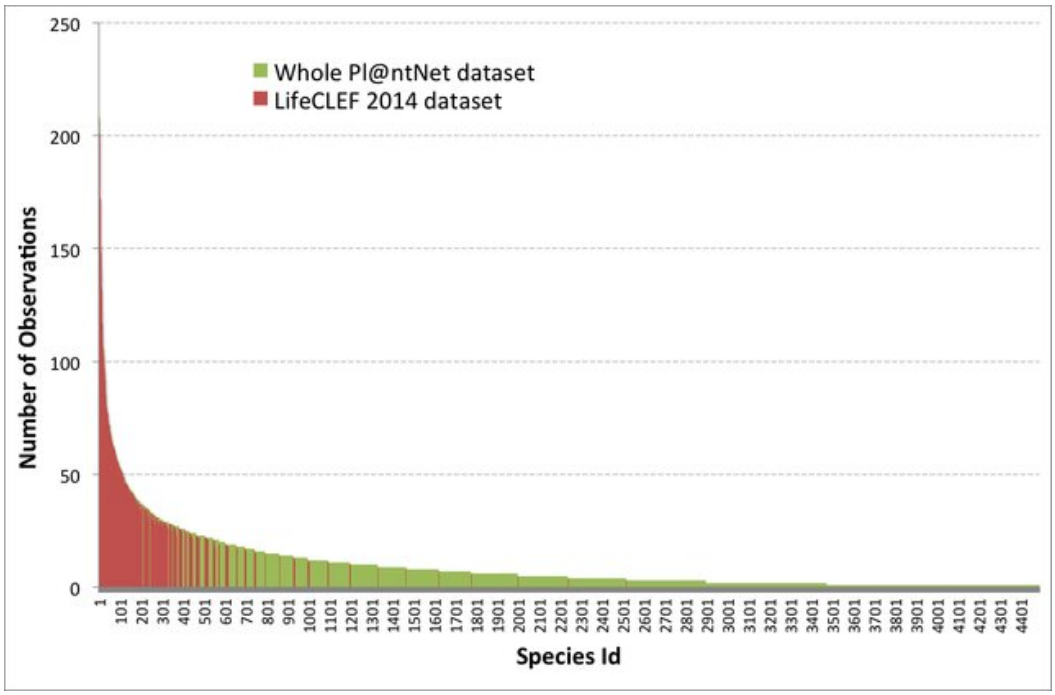


Fig. 1.2 Long tail distribution of the whole Pl@ntNet dataset (with PlantCLEF 2014 subset in red) [44].

As more and more data from new collections or rare species becomes available online, it is highly desirable to incorporate this data in an already trained model to increase its knowledge base or to keep it up to date. Retraining the model on new data combined with the original data is desirable but not always possible, due to availability or memory constraints. However, learning only on new data can drastically affect the model's performance. The challenge is to learn new data while sufficiently memorizing the knowledge learned from the original data. Our research proposes a method to repeatedly learn new data while mitigating any negative effects on performance.

1.2 Thesis overview

This thesis consolidates our research on the specific applications of deep learning in the domain of biodiversity and ecology, especially on plant images. It explores image recognition techniques for identifying plant species and extracting their traits from herbarium specimens. It then presents an approach for detecting various plant organs within herbarium scans. Throughout these two approaches, the challenges and limitations posed by the natural data imbalances and manual labeling processes are discussed. To address these issues caused by the dataset imbalance and infrequent availability of data, a novel data-free continual learning approach is presented.

This thesis is organized as follows. In the first section of this chapter, the motivation behind using deep learning on herbarium scans is indicated and the need to learn on new data is mentioned. Chapter 2 gives an overview of basic concepts of deep learning and continual learning relevant to this thesis. In Chapter 3, the approach for each project and their relevant publications are discussed. Chapter 4 lists all three main publications with their summary and contributions. Finally, a conclusion of our work is provided in Chapter 5, along with proposed future research directions. The supporting publications relevant to this thesis are attached in the Appendix.

2. Literature Review

Artificial Intelligence (AI) has seen a transformative evolution from its early days of rule-based approaches to the gradual automation facilitated by machine learning and, subsequently, deep learning. As AI initially relied heavily on rigid rule-based systems that were defined by domain experts, leading to many constraints in scalability and applicability across various tasks and domains. The emergence of machine learning introduced a paradigm shift, enabling the learning algorithms to discover patterns within data and use these patterns to make predictions. Deep learning, a subset of machine learning, experienced a surge in the research community due to advancements in computational capacities. This progress in memory and processing power, especially due to Graphics Processing Units (GPUs) [76], empowered the deep learning methods to train large models that could assimilate the knowledge from substantial volumes of data efficiently. Deep learning has been applied in many applications like computer vision, natural language processing, and robotics. In ecology and botany, deep learning finds applications in diverse areas, such as identifying species via audio and images, monitoring animal behavior, and aiding in biodiversity research.

However, the exponential surge in new and diverse data poses a challenge for traditional machine learning methods, including deep learning algorithms. These algorithms can struggle to efficiently incorporate new incoming information without compromising the predictive performance of previously learned knowledge. Continual learning has emerged as a solution to this problem, which facilitates the model's ability to assimilate and adapt to new data while retaining the previously learned knowledge.

This chapter delves into the advancements made in deep learning in Section 2.1 and provides some fundamentals about image classification, object detection, and image generation. Section 2.2 provides an overview of continual learning, its biological inspiration, and some techniques for learning new data without any access to the previous data.

2.1 Deep learning

Deep learning is a sub-field of machine learning for learning data representations. It processes the data in multiple layers by employing hierarchical architectures that facilitate the creation of intricate levels of data abstractions or features for learning the data representation, particularly in image processing [54]. Drawing inspiration from biological systems, a typical deep learning network consists of multiple layers of artificial neurons [72]. The main objective of deep learning is to learn patterns or discover complex structures in large datasets by leveraging the depth and interconnections of neural networks. The layers within the networks are interconnected, each governed by numeric weights that regulate signal transmission between neurons. These weights can be adjusted by using a backpropagation algorithm, based on the training data, to make the network capable of learning [88]. Backpropagation is a fundamental technique for learning in neural networks. It operates iteratively by fine-tuning the weights of the connections between neurons to minimize the error or loss function, which is a measure of deviation from the model's desired performance. Following each forward pass, the backpropagation algorithm calculates the gradients at each layer with respect to the loss function. These gradients guide the update of network parameters via an optimization function like gradient descent. This optimization function determines the direction for weight adjustments in order to minimize the overall error. In essence, backpropagation incrementally refines model parameters, enabling it to learn the desired patterns in the data and enhance prediction accuracy.

Over the past decade, deep learning has experienced a remarkable surge in popularity, due to its wide-ranging applications across various domains, such as computer vision, natural language processing, and automation. This rise can be attributed to the exponential growth of available data and advancements in computational hardware. Deep learning's ability to automatically extract high-level features from raw data has transformed the landscape of machine learning, eliminating the need for labor-intensive tasks of feature engineering and architecture design. As evidenced by recent surveys documenting its history and applications, the emergence of deep learning as a dominant research area has had a profound impact on the artificial intelligence landscape [20, 81].

2.1.1 Convolutional neural networks

Among many types of deep neural networks, convolutional neural networks (CNNs) have been most extensively studied, particularly for computer vision. They draw inspiration from the innate mechanism in the animal visual cortex for detecting light in receptive

fields, discovered in 1959 by Hubel & Wiesel [40]. Inspired by this revelation, Kunihiko Fukushima in 1980 proposed a neural network based model for pattern recognition [22]. The pivotal moment in deep learning happened in 1990 when LeCun et al. [55] published a groundbreaking paper that established the modern framework for CNN, which was further improved in 1998 [56]. Their innovation was the development of a multi-layer artificial neural network called LeNet-5, shown in Figure 2.1, designed to classify handwritten digits from original images without any preprocessing or feature engineering. It consists of five layers that are trained with the backpropagation algorithm [35].

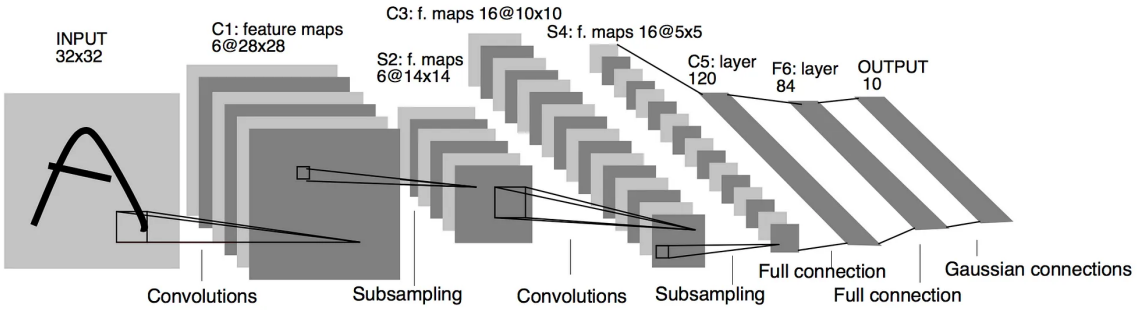


Fig. 2.1 LeNet-5 architecture as originally published in [56].

LeNet-5 is widely considered a precursor to the modern CNN architectures. However, due to the scarcity of memory and computational resources during its era, notably the absence of powerful GPUs (graphics processing units), little progress on CNN-related research happened until about 2010. In 2012 Krizhevsky et al. developed AlexNet [50], a CNN model bigger and deeper than LeNet-5. It won the most difficult object recognition in ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012) [89], with an error rate of 15.3%. AlexNet achieved the best classification against all the traditional machine learning and computer vision approaches. It was a significant breakthrough for deep learning, which also renewed the interest of researchers in modern CNN architectures.

In 2014, the Visual Geometry Group (VGG) at the University of Oxford proposed a new model called VGGNet with 13 convolutional layers with small (3x3) convolutional filters [98], as shown in Figure 2.2, whereas AlexNet only had 2 convolutional layers. VGGNet performed very well in ILSVRC 2014 and was the runner up with an error rate of 7.3% [89]. Even though VGG and AlexNet were not very deep, they were sometimes prone to overfitting. In 2015, Sergey Ioffe and Christian Szegedy proposed batch normalization, a method to make the training of deep learning models faster and stable by normalizing the inputs of each layer [41]. This helped to mitigate the problem of internal covariate shift and provided a regularization effect while also preventing extreme gradients during backpropagation.

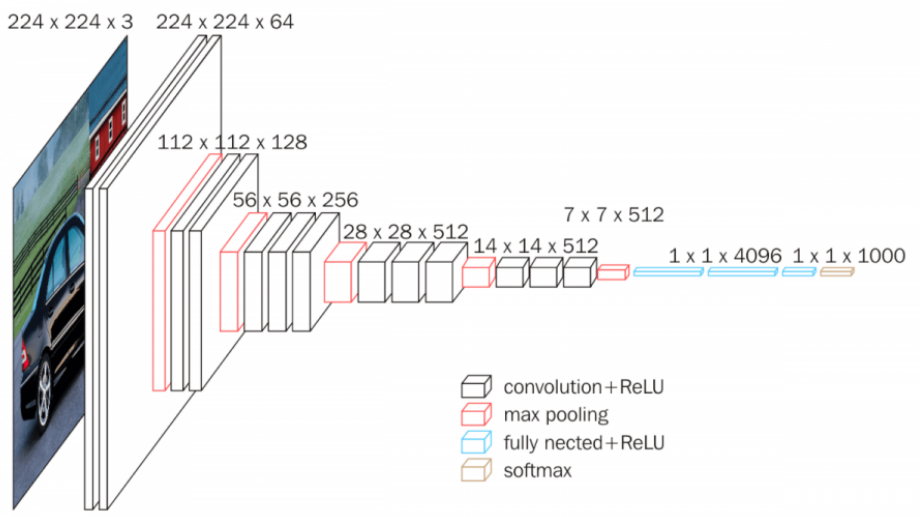


Fig. 2.2 VGG16 architecture containing 13 convolutional and 3 fully connected layers [98].

Later in 2015, He et al. discovered that a subsequent increase in the number of layers of a model did not provide any significant performance improvement but rather a gradual decrease, even after using batch normalization. Hence they suggested degradation was most likely due to notorious vanishing/exploding gradients [32]. To overcome this problem they came up with ResNet, which won the ILSVRC 2015 competition having an error rate of only 3.57% [89]. ResNet introduced a novel concept in CNN architecture called residual blocks. ResNet is composed of many residual blocks, each consisting of two to three layers stacked together with a skip connection, as shown in Figure 2.3. The skip connections allow the gradient to flow unimpeded to the initial layers during backpropagation, enabling networks to have more depth while maintaining accuracy without degradation.

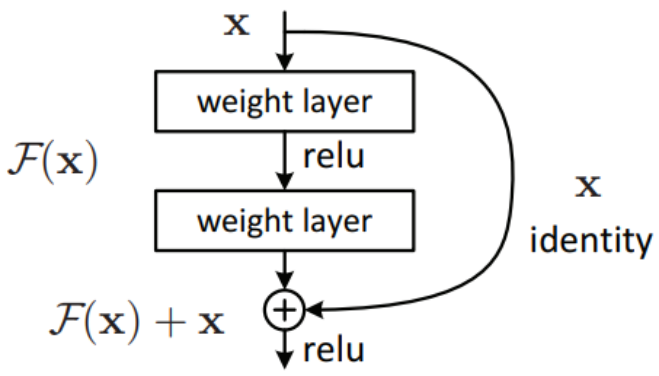


Fig. 2.3 A residual block with skip connection [32].

2.1.2 CNN architecture

A Convolutional Neural Network (CNN) is a feedforward neural network that automatically extracts features directly from data given in the form of multiple arrays like images, sequences, or video data. Unlike conventional fully connected neural networks or multilayer perceptron (MLP), where each neuron is linked to all the neurons in the previous layer as shown in Figure 2.4, CNN employs a different approach that requires less learnable parameters, thus improving training speed.

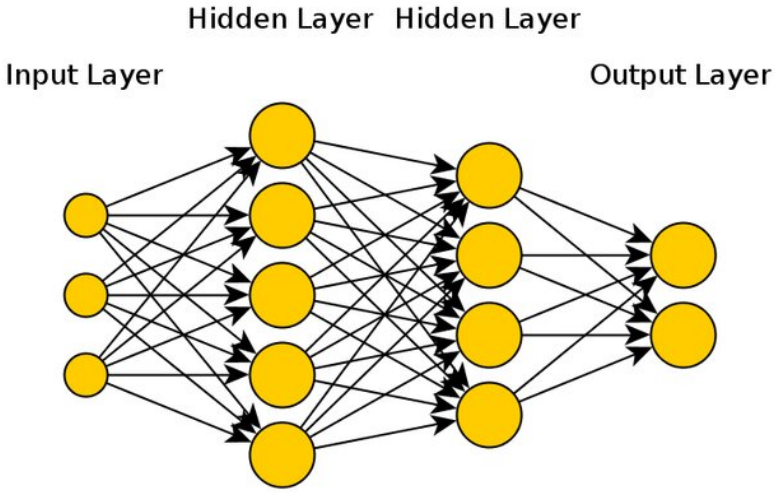


Fig. 2.4 An example of multilayer perceptron architecture [79].

In convolutional layers, each neuron can only receive input within a receptive field, which encompasses a small group of neurons of the preceding layer in local patches. A convolutional neural network consists of multiple stacked convolutional layers and optionally a fully connected layer at the end depending on the application. The convolutional layer uses convolution kernels to generate feature maps from input data of the previous layer. The feature maps for each layer are obtained by sliding the convolution kernels across the receptive fields to calculate localized dot products between the input and kernel matrices. This process is known as a convolution and it represents the input data and intermediate feature maps that highlight specific features within the input at each spatial position. An element-wise nonlinear activation function, most commonly ReLU [26], is then applied to the convolved results. Activation functions enable the model to learn complex non-linear features in the data.

In convolutional neural networks, channels represent distinct dimensions or features within various types of input data. For example, an input image will generally have only

one channel if it is grayscale and three channels if it is colored (RGB image). Each channel represents a specific pattern or feature of the input. In CNNs, as the data progresses through the layers, it is typical for the number of channels to gradually increase for the feature maps, allowing the network to learn more intricate and diverse patterns such as shapes, edges, or color gradients. Channels of the feature maps need to increase as they enable the network to learn hierarchical representations and complex structures in the input data. As the number of channels increases, it is typical for the spatial dimensions of the feature maps to decrease the overall dimensions of the feature map, thus reducing the number of learnable parameters in the convolutional layers while keeping the necessary information. This reduction in spatial dimensions or subsampling has the added benefits of reducing redundancy in feature maps and curbing the risk of overfitting. The spatial dimension of the maps can be decreased by employing a pooling layer (e.g. max pooling, average pooling) after each layer or block of layers, or with strided convolution [49], which involves skipping certain sliding positions of the kernel. The two main components of a CNN, the extraction of feature maps after convolution and the subsequent subsampling of those feature maps are shown in Figure 2.1.

As the network learns, the kernels in each layer are updated to improve the feature maps representative of the input data. The initial layers in the network learn primitive features such as corners and edges while the deeper layers use these basic features to learn more complex features like curves and basic shapes. This process of hierarchical feature extraction and progressive refinement from primitive elements to intricate and sophisticated shapes achieved through the convolutional neural network's layered architecture can be visualized [119], and even reused for similar applications [116].

2.1.3 Image classification

Image classification is a computer vision task that assigns labels to input images by categorizing them into predefined classes. Convolutional neural networks have revolutionized image classification due to their ability to automatically learn spatial hierarchies of features, such as edges, textures, and shapes directly from the input image, which is important for recognizing objects in images, without the need for traditional manual feature engineering. A typical network for image classification consists of the following layers:

- **Input layer**: This is the first layer in the network and is responsible for receiving the raw image as input. The input layer does not perform any computation but only reads the pixel values of the image as a tensor.
- **Feature extraction layers**: These layers are formed by combining convolutional layers and activation functions. Pooling and batch normalization can also be added in these

layers depending on the architectures, as shown in Figure 2.2. These layers perform the bulk of the computation in the network and are responsible for extracting the features from the image.

- **Fully connected layers**: An image classification can consist of one or more fully connected layers at the end. These layers receive the processed features from the previous layers, flattened into a single vector, and learn complex relationships between those features, as displayed in Figure 2.2.

- **Output layer**: The last layer of the model is called the output layer. It is a fully connected layer with several neurons equal to the number of predefined classes. This layer performs the image classification by assigning the probability of the image belonging to each class, using softmax or sigmoid activation functions.

2.1.4 Object detection

Object detection is a computer vision task of locating and identifying multiple objects of predefined classes in an image. It is more challenging than image classification, which predicts the class of only one object in an image. Object detection on the other hand is a combination of two tasks, object localization and object classification. Object localization refers to identifying the location and size of one or more objects in an image by drawing a bounding box around them. Object classification assigns each of these objects to a class, similar to image classification.

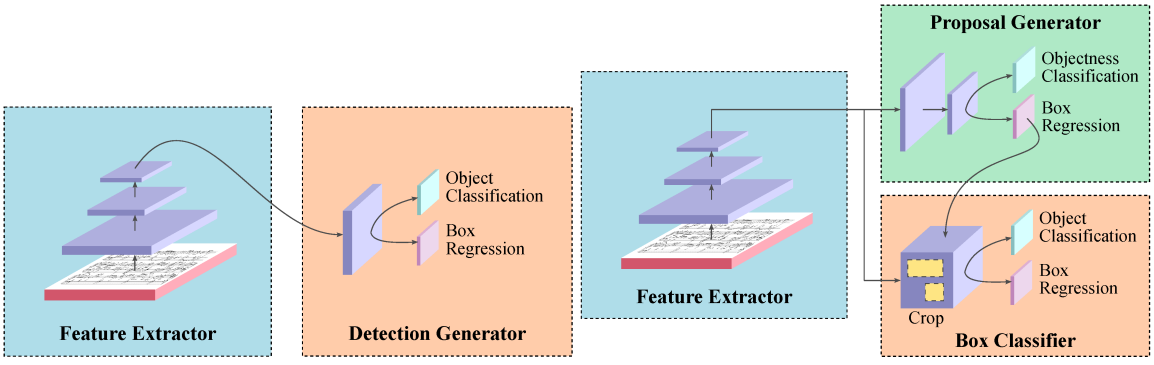


Fig. 2.5 a) Basic architecture of one-stage detector. b) Basic architecture of two-stage detector [75].

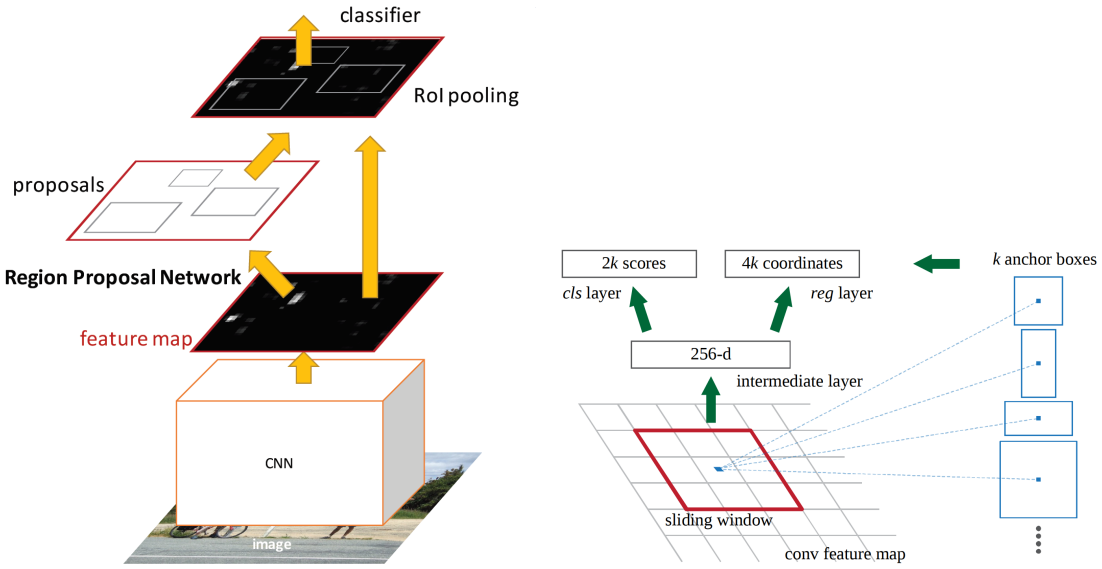
The object detection methods can be categorized into two main types: one-stage detectors and two-stage detectors. One stage detectors like SSD [61] and YOLO [84] use a single

network to directly predict the bounding boxes and class probabilities for all objects detected in an image, as shown in Figure 2.5a. Whereas two stage detectors like Faster R-CNN [86] first generate region proposals (candidate object regions) then perform classification of each proposed region and refine the size and location of the bounding boxes, as shown in Figure 2.5b. One-stage detectors are much faster and memory efficient but two-stage detectors have higher object recognition and localization accuracy [38]. Since in our research accuracy is more important than speed, especially due to similar looking objects, we opted for a two-stage object detection method.

The first successful framework for a two-stage method also called the region based method, was presented in 2014 by Girshick et al [25]. It performed object detection by generating region proposals using selective search and then classifying these regions with a CNN. It was an intuitive idea but very slow because CNN-based feature extraction was required for each candidate region, thus needing a lot of memory and computational resources. In 2015, Girshick proposed Fast R-CNN [24], which improves on R-CNN with two major contributions: 1) Region of Interest (ROI) pooling, which allows features from the entire image to be extracted just once instead of feature extraction from CNN for each proposed region and 2) a single network instead of three independent models for localization and classification of objects, thus making the process faster and more efficient.

While Fast R-CNN made strides in speed, it still relies on region proposals generated externally by selective search, which is a computational bottleneck. To mitigate this problem, Ren et al. proposed Faster R-CNN [86], just three months after Fast R-CNN, which introduced the Region Proposal Network (RPN). This innovation directly generated region proposals from the convolutional feature maps, eliminating the need for external proposal methods and significantly enhancing both speed and accuracy compared to its predecessor. Despite advancements in speed in subsequent models, only a few object detection methods have surpassed the performance of Faster R-CNN, solidifying its position as one of the leading object detection techniques.

Faster R-CNN consists of two modules, the region proposal network (RPN) for proposing regions and Fast R-CNN for predicting bounding box and class labels of objects, both sharing convolutional feature layers from a single CNN, as shown in Figure 2.6a. RPN generates thousands of anchor boxes from the feature map, which are predefined bounding boxes of varying scales and aspect ratios, as shown in Figure 2.6b. These anchor boxes serve as templates that are placed over the image to localize potential objects of different sizes and shapes in the image. RPN then adjusts the dimensions of these anchor boxes to align them with potential objects while simultaneously predicting the probability, also known as objectness score, of each anchor box representing a foreground object or background.



(a) (b)

Fig. 2.6 a) An illustration of Faster R-CNN model. b) The region proposal network (RPN) in Faster R-CNN [86].

Following this approach, the anchor boxes are sorted according to their objectness scores and filtered with Non-Maximum Suppression (NMS), which removes redundant or overlapping proposals and passes them to the Fast R-CNN model. Thus, RPN acts as an attention mechanism for the Fast R-CNN network by suggesting regions more likely to contain an object.

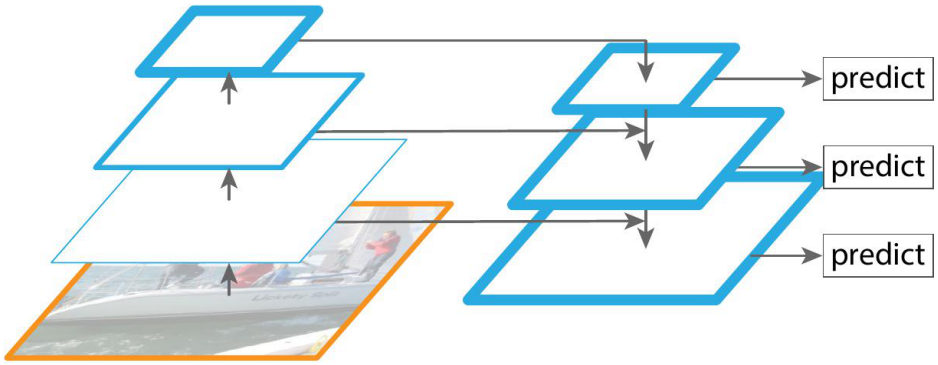


Fig. 2.7 Illustration of the feature pyramid network (FPN) [58].

Object detection can be challenging when there are multiple objects in an image with varying scales, particularly for small objects. To overcome this problem, there is an architecture called Feature Pyramid Network (FPN), which generates feature maps at multiple scales [58]. FPN is composed of a bottom-up pathway and a top-down pathway, as shown

in Figure 2.7. The bottom-up pathway is a traditional CNN, such as ResNet, for feature extraction. After each layer of CNN, the spatial resolution of the feature map decreases while the semantic value of the features increases, thus creating a hierarchy or pyramid of layers. From the network's feature map of the last layer having semantic value, FPN creates a top-down path by progressively upsampling the feature map. These are then merged, using lateral connections, with the bottom-up feature maps of the same spatial size. FPN provides object detectors like Faster R-CNN with multi-scale feature maps, enhancing their ability to detect objects of different shapes and sizes within an image.

2.1.5 Image generation

There are several machine learning architectures based on CNNs that can generate realistic-looking synthetic images. The two most common architectures are Variational Autoencoder (VAE) and Generative Adversarial Network (GAN). Variational autoencoder was the first kind of deep learning generative model, introduced by Kingma and Welling in 2013 [47], that tried to reconstruct or generate images as close as possible to the training images. VAE consists of an encoder that maps the input images to a latent space and a decoder that generates images, using their latent space representation, that resembles the input images, as shown in Figure 2.8. The figure shows an example reconstruction of a small grey-scale image by the decoder, based on the latent representation of the original image created by the encoder. Although VAE can provide better control over image generation, it produces images with less quality and realism than GANs [67].

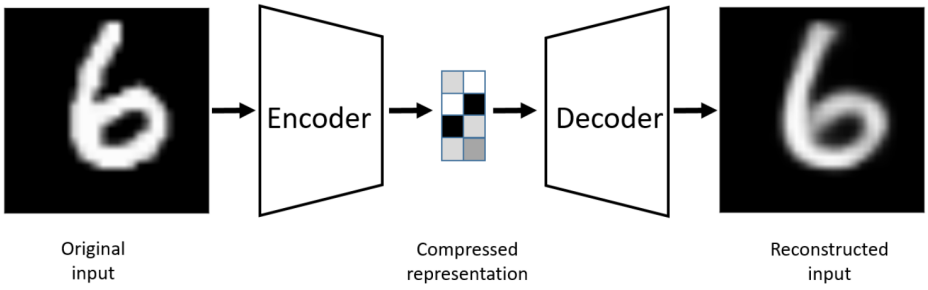


Fig. 2.8 A Variational Autoencoder model. (Diagram taken from [6])

Generative Adversarial Network (GAN), introduced by Goodfellow et al. in 2014 [29], is a pioneering framework in machine learning specially designed for data generation. The core architecture consists of two neural networks, called generator and discriminator. These networks have a competitive relationship with each other, which lets them generate authentic and realistic data. The generator network produces synthetic data by mapping random noise or latent space vectors to a representation that closely resembles the distribution of real

data. Simultaneously, the discriminator network tries to distinguish between the real and artificial data created by the generator. This adversarial process forces the generator and the discriminator into a min-max optimization scenario, where the generator tries to produce data that is realistic enough to deceive the discriminator while the discriminator continuously learns to accurately differentiate between the real and generated data [29].

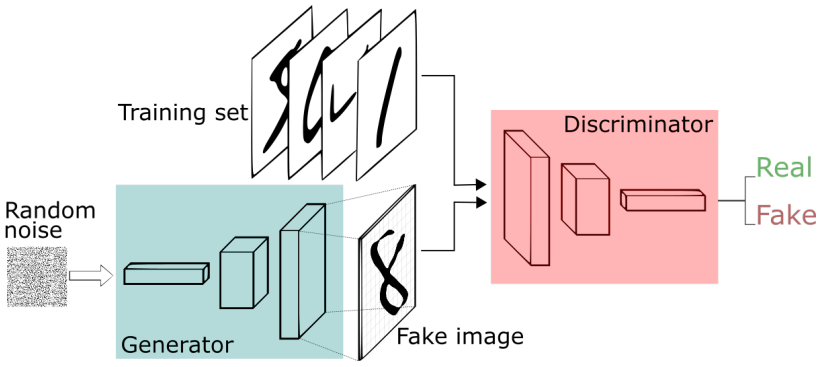


Fig. 2.9 An illustration of Generative Adversarial Network. (Adapted from [97])

Over time, as both networks undergo iterative training, the generator becomes increasingly adept at crafting highly realistic artificial data samples, thus making it more challenging for the discriminator to differentiate between real and synthetic data. This adversarial learning process is illustrated in Figure 2.9. The figure shows a typical training setup of the generator and the discriminator, where the generator tries to create images of digits that resemble the original images and the discriminator learns to identify whether the image is generated or real.

The success of GANs spans across various domains, notably in computer vision where they have shown excellent results in generating high-quality realistic images [8]. The applications of GAN extend beyond just image generation to natural language processing, audio synthesis, and style transfer [46].

As the generator network is responsible for image synthesis, it does not necessarily need a discriminator for training. It has been shown that the generator can be trained to produce images by utilizing the internal representations of the pre-trained classifier [31, 87]. Through this approach of knowledge transfer, the generator can gain insights from the trained model to synthesize images with similar features and distributions [39]. This mode of training proves invaluable in scenarios where GAN has no access to the authentic original images necessary for training the discriminator, thus relying solely on the classifier or a comparable model trained on the original dataset. This method of generator training can be essential for continual learning of a model without needing access to real images, a concept which is elaborated in the following section.

2.1.6 Applications in ecology

Deep learning has been successfully applied in ecology and botany to perform classification and identification of taxa and traits of animals, insects, and plants from natural or lab images. With the exceptional performance of CNN based models in competitions such as ImageNet [89], they have also demonstrated impressive results in recognition of plants in LifeCLEF challenges since 2015 [45, 27].

With the increase in digitization efforts and sudden rise of citizen science portals, there is a huge amount of data available for the identification of plants [7, 91] and animals [108, 71, 22]. Many online platforms and apps such as Pl@ntnet [43], iNaturalist [106], LeafSnap [51] and Flora Incognita [63] simplify the identification of plant species from images. Researchers have also used CNN to detect various plant organs, such as flowers, fruits and seedlings [93, 100, 111].

2.2 Continual learning

The recent deep learning models have been able to surpass human level performance in image recognition and object detection. As more and more data is becoming available in online repositories, it is desirable to assimilate it into existing models in order to improve their accuracy or incorporate new knowledge. Although deep learning models are impressive when trained on a large static dataset, their performance or knowledge can only improve from new data over time by continuously repeating the entire training process. The reason is their inherent static design.

This is where continual learning becomes a pivotal role. Continual learning is also sometimes called lifelong learning or incremental learning [102]. Continual learning is based on the neurocognitive process of humans for retraining knowledge and incrementally learning from new experiences and observations. There can be several reasons for continual learning:

- Retraining the model from scratch as new data accumulates over time can be computationally and memory expensive.
- Frequent retraining on the entire dataset is required if the incoming data has new patterns or new unseen classes.
- Learning on new data, especially with multiple categories, can lead to catastrophic forgetting of previously learned knowledge or tasks.

2.2.1 Stability–plasticity dilemma

Catastrophic forgetting refers to the decrease in performance of the model on previous data while adapting the model to new information because it leads to overwriting existing knowledge [65]. It is the main issue in continual learning, and it arises due to the inherent nature of models derived from neural networks. As the model learns on new data, it adjusts its parameters, thus overwriting their learned values, which leads to interference with previously learned knowledge. To overcome catastrophic forgetting, continual learning tries to create a balance between the preservation of old knowledge and the gradual absorption of new knowledge into the model. This is known as the stability–plasticity dilemma and has been widely studied in both biological systems and computational models [66, 19]. Stability refers to the capacity to retain current knowledge without being easily disrupted. Plasticity refers to the network's ability to adapt to new data or patterns. The dilemma is that there is a trade-off between stability and plasticity. Increasing the stability of a model makes it inflexible to adapt to new information, whereas reinforcing plasticity leads to the loss of previous knowledge.

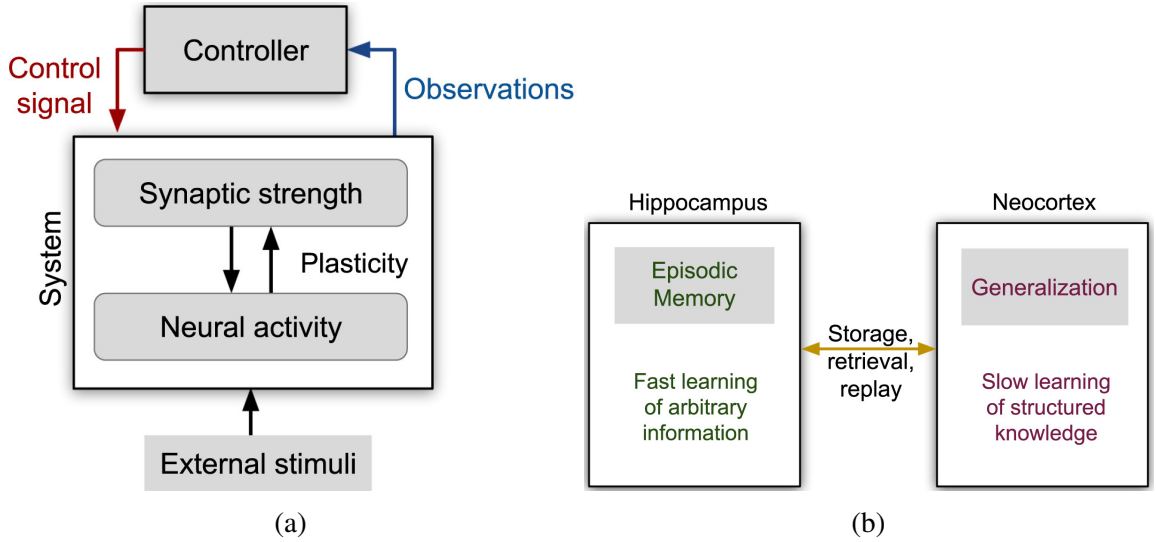
2.2.2 Biological inspiration

As humans and animals are able to incrementally acquire new knowledge throughout their lives, many high level approaches for continual learning take inspiration from nature and biology. The brain's ability to accumulate novel information continuously while retaining relevant knowledge and memories due to its synaptic plasticity is the main reason that inspires numerous continual learning methods in artificial intelligence. There are two main neurological mechanisms that are adapted for continual learning discussed in the following.

Hebbian and homeostatic plasticity

Hebbian plasticity is based on the mechanism proposed by Donald Hebb in 1949 for the process of learning or adaptation of neurons in the brain to external stimuli by synaptic plasticity [33]. Also known as Hebbian learning, it theorizes that the more two neurons are active together or simultaneously the more their synaptic connection strengthens. This idea also resonated with Hebb's famous phrase "Cells that fire together, wire together". This mechanism of reinforced neural connections is believed to be the elementary process of learning associations in the brain. Hebbian learning principles have been applied to neural networks to learn patterns and associations. Thus, Hebbian plasticity can be used to explain the flexibility or adaptability of neural networks, as the weights of the neuron connections change based on the input patterns.

However, Hebbian plasticity alone is inherently unstable [2], leading to runaway neural activity, which can potentially cause instability in neural networks. To rectify this problem, stability is achieved with homeostatic plasticity [16]. It regulates excessive strengthening or weakening of synaptic connections by imposing constraints on the strength of neural activity [68]. Homeostatic plasticity can be considered as a feedback control mechanism on the instability caused by Hebbian plasticity. Figure 2.10a displays the schema of a learning system based on Hebbian plasticity with homeostatic plasticity.



(a) (b)

Fig. 2.10 a) Hebbian learning with homeostatic plasticity [120]. b) Complementary learning systems (CLS) theory [64]. Schematics adapted from [78].

Complementary learning system

Another biologically-inspired approach for continual learning is derived from the complementary learning system (CLS) theory [64]. CLS tries to replicate the fast and slow complementary functions of the hippocampus and neocortex in the human brain. The hippocampus part of the brain is responsible for the rapid learning of novel information and short-term storage of new memories and experiences. In contrast, the neocortex gradually learns generalized and long-term memories. As the hippocampus stores the memories temporarily, they are fragile and susceptible to interference. In contrast, the neocortex can store vast amounts of information for an extended time. The quickly learned new information in the hippocampus is slowly replayed over time to the neocortex for long-term storage during sleep and rest periods. This process strengthens and stabilizes memories in the neocortex, reducing their reliance on the hippocampus for retrieval and thus allowing it to store new memories quickly. Therefore, CLS theory suggests two complementary systems: 1) the

hippocampus that rapidly encodes and retrieves new experiences and 2) the neocortex that gradually consolidates these experiences into long-term memory. Figure 2.10b illustrates the roles of the hippocampus and neocortex in complementary learning systems. Many recent continual learning frameworks for deep learning take inspiration from the complementary learning system due to its ability to rapid learning and gradual consolidation of experiences over a long period [82].

2.2.3 Continual learning approaches

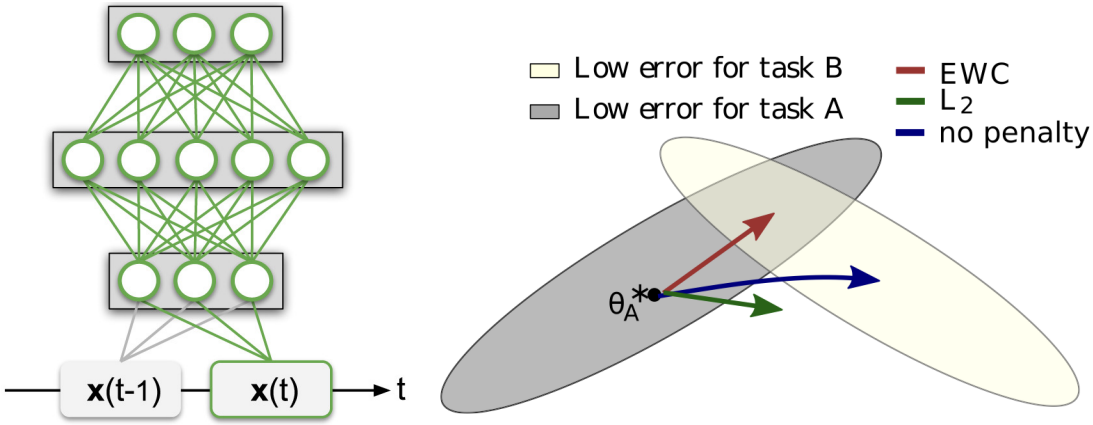
Continual learning for machine learning models can be defined as learning on a stream of data that may belong to related or new tasks or classes. Data streams have specific properties, see [83] for details, that differentiate them from static data sets. For example, the size of a data stream is potentially infinite, thus it is generally not feasible to store the entire data stream. Thus, a continual system is required to adapt the model to new data and tasks without storing any data or revisiting only a small portion of the previous data. There are many types of approaches to continual learning [82], which include regularization methods, memory and replay methods, generative replay methods, architecture based methods, and knowledge distillation based methods. This thesis will focus on data-free continual learning method. These methods do not store any previous data and as soon as the model is trained on a task or batch of data the underlying data is not accessible anymore.

Inspired by the foundational concepts of data-free lifelong learning in biological systems, only a few high-level strategies align with this principle [114]. Two approaches stand out: regularization-based methods and knowledge distillation methods utilizing generative replay. Based on Hebbian and homeostatic plasticity theories, regularization methods aim to enhance the generalization capabilities of the model and prevent overfitting by imposing some constraints during the learning phase. Conversely, knowledge distillation methods employing generative replay draw inspiration from the concept of complementary learning systems observed in biological cognition. This approach emulates how biological systems store and utilize past experiences to inform and facilitate new learning processes by retaining the previously learned knowledge in a generator and then replaying the old simulated data while learning new tasks.

Regularization based methods

One of the fundamental strategies for catastrophic forgetting in continual learning is based on regularization techniques, which aim to strike a delicate equilibrium between plasticity and stability. Regularization-based methods impose constraints on the parameters and hyper-

parameters of the model during training, to retain the previously learned knowledge while learning new tasks, as depicted in Figure 2.11a. The figure shows a neural network model getting input in two-time steps/tasks $x(t-1)$ and $x(t)$. The model uses regularization to retain knowledge from both tasks.



(a) (b)

Fig. 2.11 a) Training model on new task with regularization (Schematic adapted from [78]). b) Elastic Weight Consolidation (EWC) [48].

Regularization-based methods are often memory-free, meaning they do not retain prior data. Among these methods, Elastic Weight Consolidation (EWC) [48] stands out as a prevalent approach. EWC operates by restraining the parameter updates during training by penalizing significant alterations that may disrupt previously learned tasks. By regularizing the parameters, EWC helps to maintain a balance between old and new tasks, thus reducing the interference effect between tasks, as depicted in Figure 2.11b. The figure shows two regions of parameter space that optimize the model for task A and task B, where the model has already been trained on task A. The arrows in the figure point to the trajectories the parameters take when the model learns task B using three different methods. If the model learns task B without any regularization (blue arrow), it will optimize its parameters for task B only and forget about task A. The green arrow directs the parameters outside the optimum region of task A and task B when constraints are applied to all weights equally during the update. EWC (red arrow) finds a solution for the model to learn task B without drastically interfering with task A. This usually leads to model parameters where the optimum region for task A and task B overlap.

Beyond EWC, several other regularization-based techniques exist to overcome catastrophic forgetting. For instance, Synaptic Intelligence (SI) [121] and Memory Aware Synapses (MAS) [4] function similarly to EWC in preventing the detrimental effects of catas-

trophic forgetting. These methods reinforce the retention of previously learned knowledge while adapting to new information, contributing to stable and continual learning paradigms.

Knowledge distillation and generative replay

Another approach for data-free continual learning is more complex than the regularization technique, as it combines knowledge distillation and generative replay. Generative replay is a data augmentation technique inspired by a complementary learning system. Generative replay is based on generative adversarial networks (GANs) or variational autoencoders (VAEs) to generate synthetic data resembling previously encountered tasks, as depicted in Figure 2.12. Knowledge distillation, on the other hand, is the process of transferring knowledge from a well-trained, often complex, model to a newer one. This distillation process aids in consolidating previous knowledge while training on new tasks. One of the earliest approaches leveraging knowledge distillation for continual learning is Learning without Forgetting (LwF) [57].

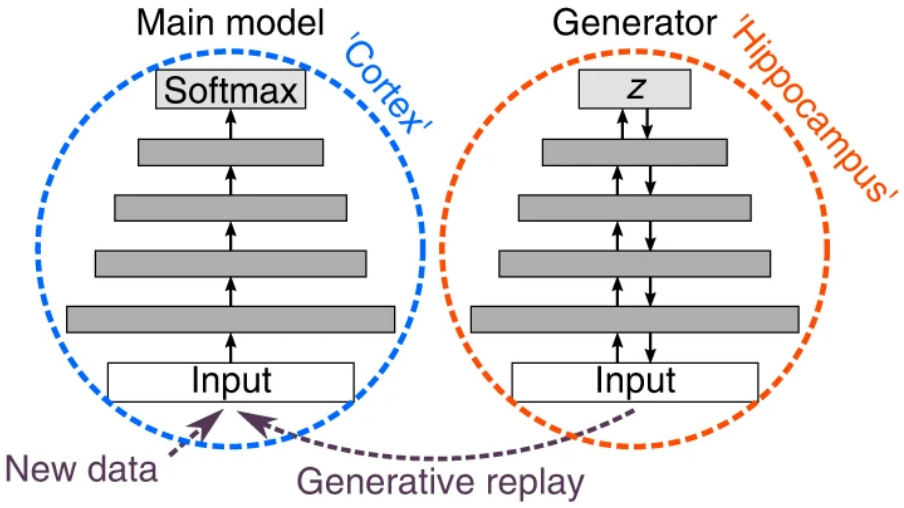


Fig. 2.12 An illustration of generative replay based on biological approach [105].

The fusion of knowledge distillation and generative replay presents several data-free integration methodologies. Firstly, the distilled knowledge and the generated samples can be combined to fortify and reinforce prior knowledge while learning new tasks. Secondly, they can function sequentially, wherein the generator is trained without any training data but by using the knowledge distillation technique to extract the data distribution of the previous tasks from the trained model. A visual representation of this synergy between knowledge distillation and generative replay is shown in Figure 2.13. The strategic union of knowledge distillation and generative replay is a robust technique for data-free continual learning. By

leveraging the strengths of knowledge transfer and synthetic data generation, this hybrid approach empowers models to assimilate new tasks while retaining knowledge from previous experiences.

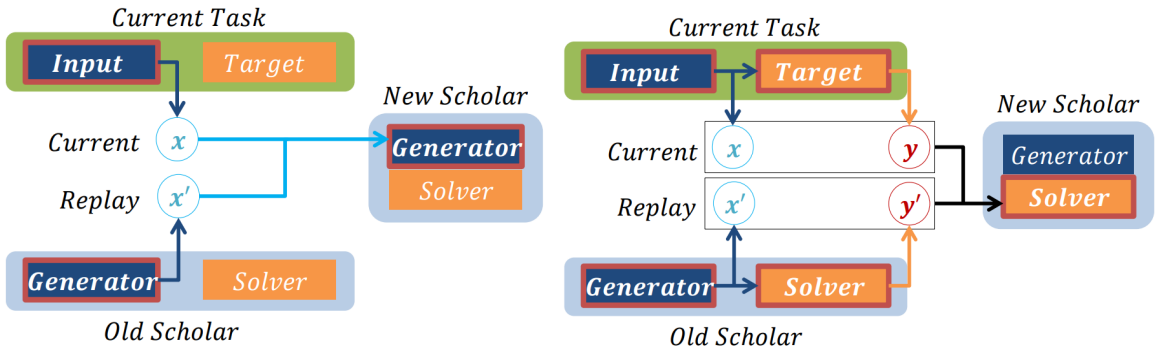


Fig. 2.13 Training sequence of a) generator and b) solver/classifier on combination of real and generated data. (Diagram adapted from [96])

3. Methods

The increasing digitization of herbarium collections and their public accessibility has propelled biodiversity research into uncharted territories. Especially the advent of deep learning has revolutionized image analysis and has offered many new avenues in botany and ecology. This chapter consists of three sections, and in the first section we present a deep learning approach based on convolutional neural networks for taxon identification of plants and trait extraction from digitized herbarium scans. This approach was successfully able to identify the species of specimen from a large dataset of herbarium images and extract the leaf features of the plants, from a smaller dataset. In addition to image classification, deep learning based object detection techniques offer promising applications in herbarium scans. One of these methods, presented in the second section, was used to detect and locate plant organs on herbarium scans. It was able to detect five types of plant organs, although with varying degrees of success due to their skewed distribution in the images. As datasets continue to expand exponentially, there is a growing need for machine learning models to be adaptable, despite many computational constraints and data availability challenges. Our approach discussed in the third section is designed to enable a model to incrementally learn from new data, that may be imbalanced while retaining the previously learned knowledge. The sections below provide a detailed explanation of these approaches.

3.1 Species classification and trait extraction

Herbarium collections are increasingly becoming available online to the scientific community due to the ongoing digitization of herbarium specimens worldwide. Concurrently, rapid advancements in deep learning algorithms are revolutionizing pattern recognition on images, which are increasingly being applied in ecology. In this section, we will discuss our first publication in which deep convolutional neural networks were used for taxon identification from digitized herbarium scans, consisting of a diverse collection of 1000 most frequently documented species in GBIF [23]. We will also discuss extraction of morphological traits from herbarium scans on a smaller collection, by using their identified taxonomy.

A herbarium is a collection of preserved plant specimens, typically dried, pressed, and annotated, and stored in archival storage like a library for scientific study. These specimens can be whole plants or plant parts from different geographical locations and habitats. The specimens in a herbarium are attached on a large piece of paper, accompanied by detailed information about their taxonomic classification, place origin, and other relevant data. A digitized herbarium specimen is a high-resolution scan of the physical herbarium specimen. Figure 3.1 shows an example of such a specimen.

Despite recent efforts to digitize herbarium collections and make them accessible online, thousands of herbarium specimens remain unidentified. Additionally, a significant number of herbarium annotations need to be updated following more recent taxonomic knowledge and nomenclature, a task that is incredibly labor-intensive for botanists to accomplish on huge collections. As a result of digitization, herbarium specimens can be analyzed with computer vision and machine learning approaches, thus enabling automated species identification and image recognition, particularly with the emergence of deep learning methods. Deep learning methods, such as convolutional neural networks have gained a lot of attention for their performance in various image recognition tasks like ImageNet [89], and notably for identifying plant species from images in PlantCLEF [45].

Deep learning has revolutionized biological and ecological research by aiding taxonomists and botanists in identifying new species and facilitating biodiversity studies. While the use of deep learning in botany has traditionally focused on living plants in natural environments through citizen science applications and specific apps like LeafSnap [51], Pl@ntnet [43], and iNaturalist [106], the application of deep learning on herbarium specimen images is relatively new with only a few applications [9, 94]. These advances have enabled taxon recognition and species identification from herbarium images, paving the way for innovative applications in the botanical sciences.

3.1.1 Taxon and trait recognition from herbarium scans

In our research, we not only focus on taxon recognition but also the identification of morphological traits from herbarium specimens. So far, there has been limited exploration of trait recognition from plant images. However, there are however some approaches for extracting leaf traits from plant images and herbarium scans with specialized software and semi-automated workflows for feature selection, but they can be quite laborious [14]. We first created a model for taxon recognition on millions of herbarium scans of the most abundant species from MNHN (Muséum national d'Histoire naturelle) vascular plant herbarium collection data-set in Paris [53]. We then performed trait recognition on a subset of these images which belonged to the African taxa. The trait recognition on African taxa was performed

Fig. 3.1 An example of a digitized herbarium specimen [23].

due to the wealth of knowledge base of morphological trait data accessible through the Flora Phenotype Ontology (FLOPO) [37] and African Plants - a photo guide [21]. Another reason for the emphasis on African taxa was due to the excellent performance of the model for the North American and European taxa, the geographical region where taxonomic expertise and resources for identification are still less available and much needed.

Taxon and trait data

For our approach to identify taxa of herbarium scans, the taxon names were extracted from the GBIF metadata entries for each specimen. To resolve the synonymies in nomenclature arising from discrepancies in the FLOPO database and herbarium dataset, we utilized the GBIF taxon backbone [95]. For the trait recognition task, the trait data for each specimen was connected via the taxon name. The name of the plant trait from FLOPO matched the GBIF backbone using the Global Names Resolver (http://resolver.globalnames.org/), with only the taxon names that matched above the 0.9 score were considered for annotating the images. As the traits were not directly extracted from the herbarium specimen but were linked via the taxon names, all the traits were assigned to the herbarium scans based on their taxon, regardless of whether they were visible on the image or not. The assignment of all traits of taxa to herbarium scans could have resulted in many scans being labeled with traits not recognizable from the given plant material (e.g., flower symmetry in a specimen without flowers). To address this issue, we concentrated on a refined subset of leaf traits deemed visually recognizable from herbarium scans. These leaf traits encompass leaf arrangement, leaf structure, leaf form, leaf margin, and leaf venation. Table 3.1 reports the list of selected leaf traits.

The Image dataset and its preprocessing

The herbarium images used in this study were sourced from the open-access datasets available on the GBIF portal [23], primarily from the MNHN vascular plant herbarium collection located in Paris. Our dataset consisted of 830,408 full-scale images belonging to the 1000 species with most herbarium scans available in the MNHN collection. The non-uniform distribution of species in nature resulted in imbalanced data ranging from 5494 to 532 images per species. For the trait recognition part of the study, a subset of 170 species consisting of 152,223 images was selected, supported by trait data from the previously mentioned sources.

A herbarium typically has labels for its taxon and other annotations about the specimen at the bottom of the sheet. It also contains barcodes on the top and bottom of the sheet and sometimes reference color bar on the sides. Since these labels can act as background noise or

Fig. 3.2 Image preprocessing steps: The herbarium scans are cropped and reduced to a standard size before given as input to the convolutional neural network [118].

create potential biases for the deep learning model, all images after they were downloaded from GBIF went through a preprocessing step where they were uniformly cropped and resized in a portrait format, as suggested by Carranza-Rojas et al. [9]. Figure 3.2 shows an example of this preprocessing step.

Implementation

For recognizing the herbarium species from the images, we implemented a deep convolution neural network model, a detailed explanation of deep learning and convolutional neural networks is given in Section 2.1. We utilized a modified ResNet bottleneck network with Exponential Linear Unit (ELU) activation function between the convolutional layers [12], instead of the traditional Rectified Linear Unit (ReLU) function, as shown in Figure 3.3. We also modified the average pooling layer to accommodate the custom rectangular image dimensions. For recognizing the leaf traits, a smaller ResNet model without bottlenecks was utilized due to the limited number of leaf traits in the dataset. As multiple traits could be present in a scan, a sigmoid activation function was used in the last layer instead of softmax. We trained the model on the herbarium dataset from GBIF without transfer learning from other sources [116], to avoid potential training biases. These models we trained and implemented on a TITAN Xp GPU using the TensorFlow framework [1].

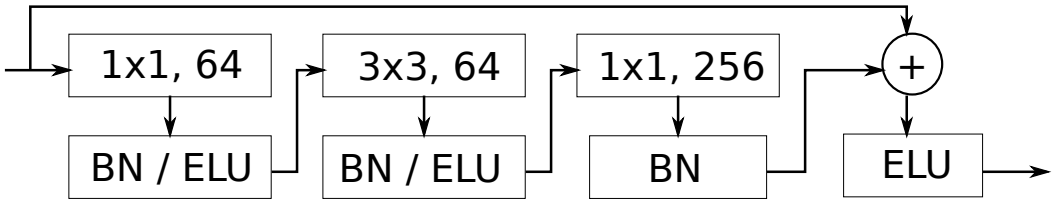


Fig. 3.3 The modified bottleneck ResNet block with ELU.

Results and Discussion

Our model trained for species recognition exhibited very good performance with an average accuracy of 82.4% of the test dataset. Notably, in 96.3% of the cases, the correct species appeared within the top five probable predictions, shown in Figure 3.4. Due to the imbalanced nature of the dataset, species with fewer images tended to yield lower prediction accuracy. For the trait recognition task, the trained model achieved an 89.6% accuracy rate in predicting the correct trait, including correctly identifying all leaf traits in 30.9% of the herbarium images. Table 3.1 reports the accuracy for all the traits.

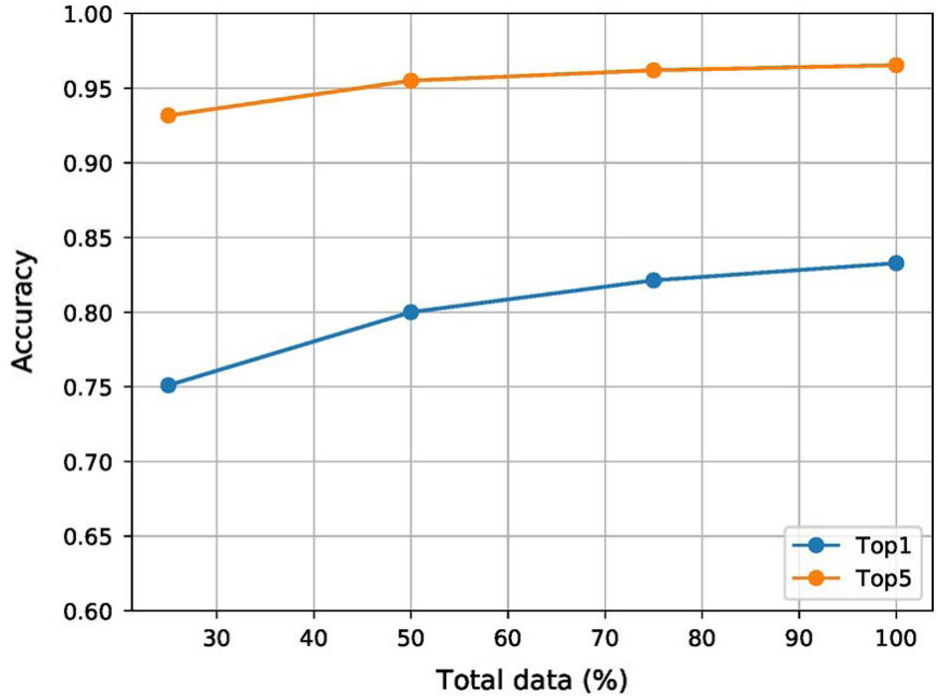


Fig. 3.4 Accuracy of the model for species recognition, depending on the proportion of dataset used for training [118].

In this research, we demonstrated a successful application of deep learning on herbarium scans for taxon identification and trait recognition, with an excellent top-five accuracy of 96.3% for species prediction due to a very large dataset of herbarium images compared to

other similar studies. The results showed that it was easier for the model to predict most of the species of the herbarium specimen than the traits, mostly due to the smaller training dataset and complexities of some traits. These findings suggest promising implications for digitization efforts by herbarium collectors globally, further advancing the research in automated species recognition and highlighting the potential of deep-learning algorithms. This research was published in 2018 [118], a copy of which is attached in Section 4.1.

Trait	State	OBO ID	No. of Scans	Accuracy
Leaf - Arrangement	alternate	FLOPO:0001032	120514	92.98
Leaf - Arrangement	opposite	FLOPO:0000420	34262	36.80
Leaf - Arrangement	rosulate	FLOPO:0900066	37459	63.00
Leaf - Arrangement	whorled	FLOPO:0002264	7550	44.61
Leaf - Form	cordate	FLOPO:0900069	10378	29.81
Leaf - Form	deeply lobed	FLOPO:0006834	28900	59.79
Leaf - Form	oblong to linear	FLOPO:0000103	86644	81.00
Leaf - Form	orbicular	FLOPO:0017811	8032	23.78
Leaf - Form	ovate or elliptic etc.	FLOPO:0000286	91954	89.83
Leaf - Margin	entire	FLOPO:0900073	118297	87.30
Leaf - Margin	not entire	FLOPO:0900074	59148	72.50
Leaf - Structure	palmately compound	FLOPO:0018499	2268	46.42
Leaf - Structure	pinnately compound	FLOPO:0907004	46827	68.62
Leaf - Structure	simple	FLOPO:0000693	128391	97.00
Leaf - Structure	trifoliolate	FLOPO:0900067	8711	9.10
Leaf Venation	palmate	FLOPO:0900070	17275	48.11
Leaf Venation	parallel	FLOPO:0900072	40710	89.57
Leaf Venation	pinnate	FLOPO:0000561	102663	90.35
Leaf Venation	triplinerve	FLOPO:0900071	7372	21.00

Table 3.1 Leaf traits selected for detection, with Open Biomedical Ontologies (OBO) ID and number of herbarium scans for each trait. The last column shows the accuracy of predicted scans for each trait.

3.1.2 Trait extraction from herbarium and collector notes

Our research efforts in taxon and trait recognition from herbarium scans have yielded promising results. In seeking further enhancement for trait recognition, we further improved the process by directly extracting trait data from herbarium specimens by examination of their annotations and collector notes, instead of associating the traits of the specimen through their respective species. The collector notes for this task were compiled from four distinct herbarium collections, obtained from the GBIF platform [23]. These selected herbarium collections were:

1. Royal Botanic Gardens, Kew [30]
2. Royal Botanic Garden Edinburgh Herbarium [15]
3. Herbarium of Universite de Montpellier, Institut de Botanique [92]
4. Herbarium of the Museum National d'Histoire Naturelle [53]

Figure 3.5 shows two of the collector notes extracted from two herbarium scans. These collector notes are also available via GBIF as metadata for each herbarium scan. We focused on the traits belonging to three primary plant organs: leaves, flowers, and fruits.

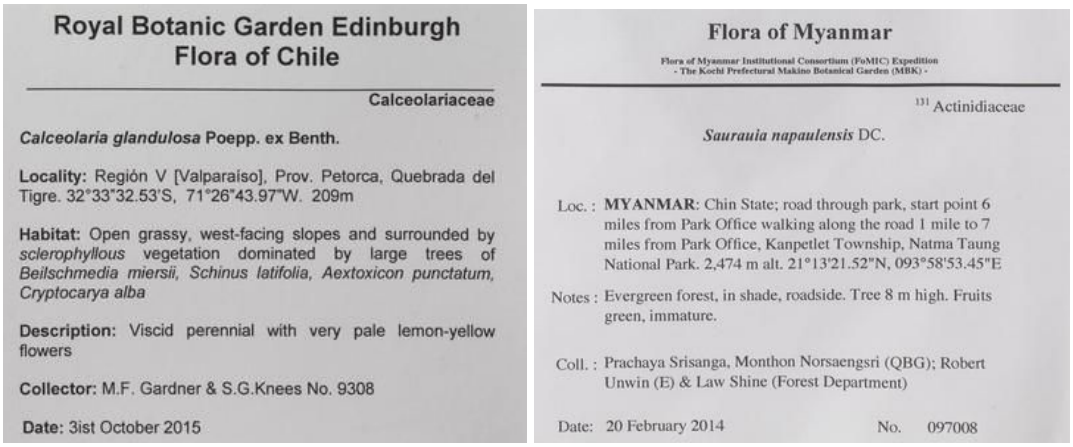


Fig. 3.5 Examples of annotations on the herbarium sheets containing the collector notes [23].

As a herbarium specimen may not possess all these organs in every instance, we further improved the process of trait extraction by only selecting the traits of the flowers and fruits explicitly mentioned in the accompanying collector notes for each herbarium specimen. All the traits for leaves were always selected as it was assumed that all selected herbarium specimens contained leaves. Leveraging the trait knowledge base from FLOPO and merging

the herbarium scans with their corresponding collector notes, we generate comprehensive trait data that directly corresponds with the plant organs on each specimen. This process of combining the herbarium scans, their corresponding collector notes, and the FLOPO knowledge is shown in Figure 3.6.

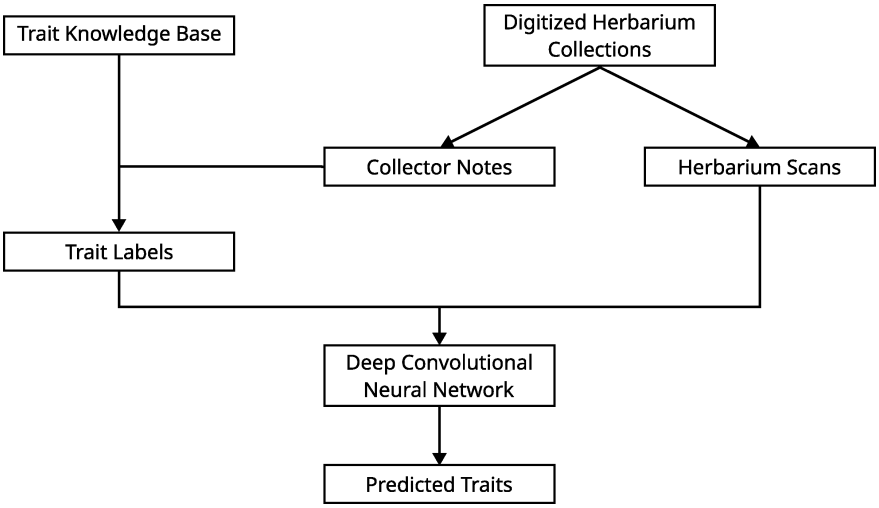


Fig. 3.6 The workflow for extracting the trait data from the herbarium scan and collector notes, merging it with FLOPO knowledge base and feeding it to a neural network as input for training.

A total of 27 traits were selected for the training of the model, consisting of 14 leaf traits, 9 flower traits, and 4 fruit traits. A visual depiction of some of the leaf and flower traits is illustrated in the Figure 3.7.

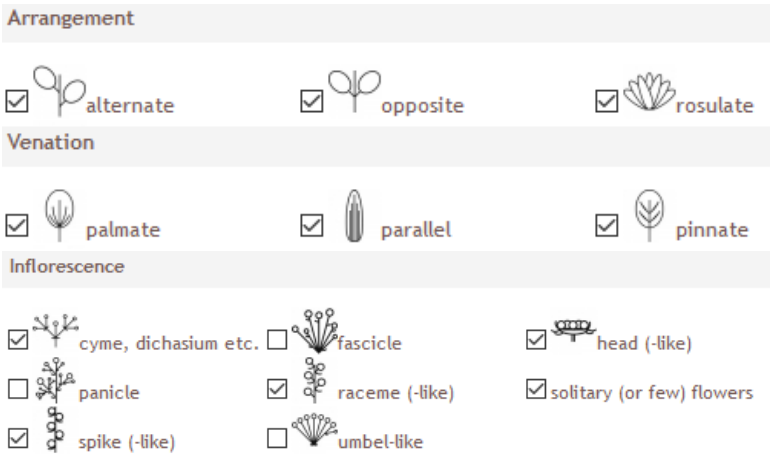
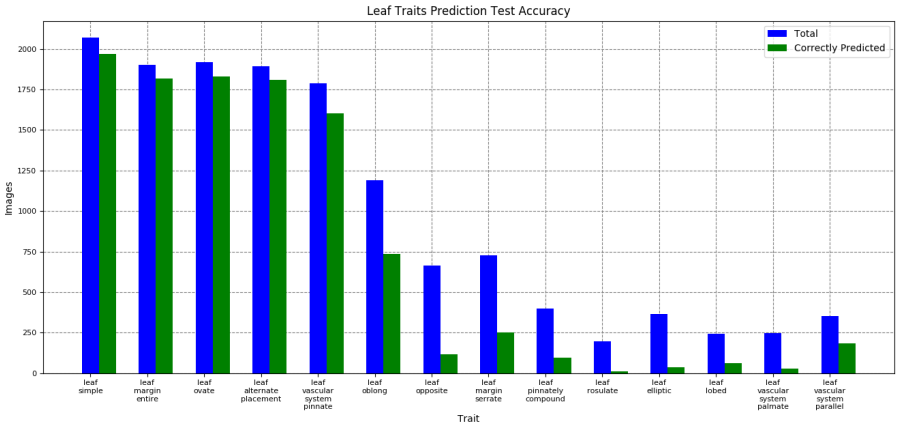
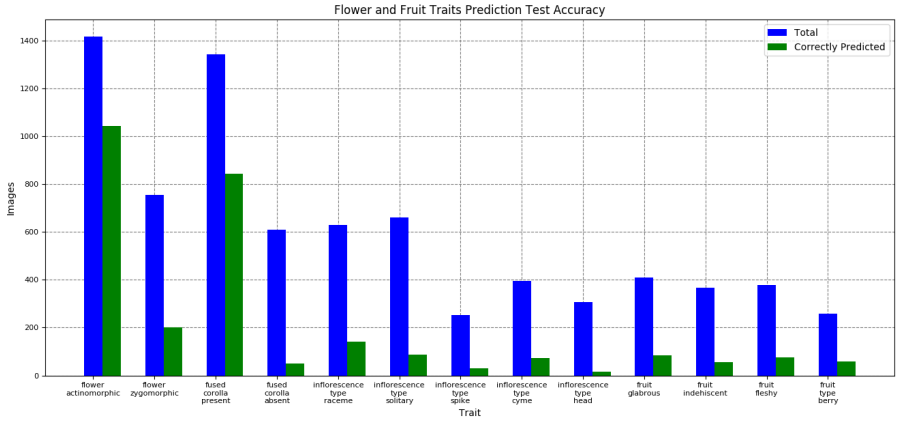


Fig. 3.7 The figure illustrates a visual representation of several selected leaf and flower traits [21]



(a)



(b)

Fig. 3.8 Total and correctly predicted images for a) leaf traits and b) flower and fruit traits.

Despite consolidating herbarium specimens from four distinct collections for the trait dataset, the resulting compilation was of a relatively modest size of 13,157 images, spanning 2,339 unique species. Many factors contributed to this constrained size of the dataset, primarily the scarcity of herbarium specimen with digitized annotation in their metadata. While the act of scanning herbarium specimens for creating high-resolution images can be automated with relative ease, the process of digitizing and incorporating associated annotations remains labor-intensive. Consequently, only a fraction of these herbarium specimens available online have their corresponding annotations and collector notes available, thus significantly limiting the available dataset. Another limiting factor is the infrequent occurrence of keywords linked to flower and fruit organs within the digitized collector notes. Since the presence of these plant organs is crucial for selecting the relevant traits, their sporadic mention in the collector notes further reduced the number of eligible specimens. Lastly, the primary factor influencing the selection of only 27 traits for this study is the low representation of numerous species within the FLOPO knowledge base. This inadequacy in species representation potentially stems from the focus of the herbarium collections and the knowledge base to certain geographical regions, thereby limiting the number of species for analysis. These multifaceted limitations underscore the challenges faced in curating a comprehensive and diverse herbarium image dataset for trait analysis.

3.2 Object detection in herbarium and camera traps

The abundance of plant images, encompassing both natural and herbarium specimens, has presented both a multifaceted challenge and an unprecedented opportunity. The emergence of machine learning as a powerful tool for extracting complex patterns from digital images is becoming invaluable in contemporary scientific pursuits [89]. Many datasets that were previously once considered impenetrable due to their sheer size and diversity, now hold the promise of yielding invaluable insights. The progress in this domain has resulted in a multitude of machine learning tools designed not only for species identification from plant images [51, 63], but have drastically transformed the significance of herbarium scans [9, 118]. The advancements in machine learning, particularly in deep learning, have made it possible to handle the chronic backlog of unprocessed and misidentified herbarium specimens by automating the process of identifying and cataloging them [94]. These techniques, specifically deep learning, extend beyond mere tools for species identification, demonstrating remarkable potential in extracting features from images. As plant images hold a wealth of visual information, they can be used to extract phenotypes and traits of the plant [80, 104]. The deep learning methods for object detection can be applied to detect and recognize plant organs from images, which can offer much insight into the ecology and the impact of climate on the species or individual plants [93, 100, 111].

However, the task of training a model for organ detection is much more challenging than species identification. In species identification, the requisite taxonomic information for each image is either readily available in the metadata or can be quickly identified by an expert. In contrast, organ detection is significantly labor intensive due to the absence of annotations specifying the position and size of plant organs within each image. This requires a time-intensive process of manual labeling of images [77], often restricting the scale of datasets. Despite these challenges, several tools have recently emerged that can partially automate the annotation process [90]. The collaborative efforts of computer science and botanical experts have facilitated the extraction of invaluable information from previously inaccessible or overlooked data. Through the synergy between these distinct fields, we are witnessing a technological leap in our understanding of ecology, evolution, and the impacts of climate change.

3.2.1 Detection and annotation of plant organs

In our research, we employ deep learning to detect and locate plant organs within herbarium scans. The object detection network employed in our approach can identify individual plant organs and pinpoint their location on the image with bounding boxes. There are several types of neural network architectures, based on convolutional neural networks, for location and detecting objects in images. We chose Faster R-CNN for this task, based on its remarkable track record and widespread application in previous studies for detecting plant organs in natural images. Faster region-based CNN (Faster RCNN) [86], a member of the R-CNN family, is specifically designed for object detection tasks. Faster R-CNN operates by identifying objects and their respective locations in stages, thus giving it the ability to detect objects of varying shapes and sizes. An overview of Faster RCNN can be found in Section 2.1.4. This has shown state-of-the-art performances in various object detection applications and competitions [122].

The adoption of CNNs, particularly Faster R-CNN, has been embraced by numerous researchers exploring diverse plant organs such as flowers, fruits, and seedlings in natural images [93, 100], and herbarium scans [73, 111]. While previous studies often focused on leaves or fruits in natural settings, our work is the first endeavor to detect both vegetative and reproductive plant organs from herbarium scans. Our novel approach holds for diverse applications. Identification and precise localization of plant organs on herbarium specimens, such as leaves, flowers, and fruits, can facilitate phenological studies spanning extensive periods. Furthermore, it can also help give us insight into the effects of climate change dating back to the Industrial Revolution [112, 52]. Our pioneering approach opens doors to understanding the evolution of plant traits across centuries and the intricate effects of evolving environmental conditions.

Herbarium annotation dataset

To train the Faster RCNN object detection network for plant organs, the herbarium scans were sourced from the Muséum National d'Histoire Naturelle (MNHN) vascular plant herbarium collection [53], from the online GBIF portal [10]. We meticulously selected a diverse selection of 653 herbarium images spanning 351 distinct species. This manual curation of scans required meticulous selection criteria to minimize visual overlap between plant organs while covering a broad range of taxa and morphology. The images were downloaded and rescalced from the original average dimensions of approximately 5100 by 3500 pixels to 1200 by 800 pixels, to reduce the training time for the model while preserving their aspect ratio.

The subsequent annotation process of these images required manually creating bounding boxes for each plant organ, with their corresponding label. For this task, we employed LabelImg [103], a Python graphical toolkit designed for annotating herbarium images. Due to the complexity and the large number of plant organs, particularly leaves, within the images, the manual annotation process was slow with an average rate of about 8 to 15 herbarium sheets labeled per hour. There were also many other challenges during this labeling process, such as difficulty in identifying the current stage in the life cycle of the reproductive organs, whether they were in the form of fruit, flower, or bud. Additionally, in certain instances, the proximity and dense coverage of small plant organs by leaves made identification challenging. The culmination of the labeling process for the 653 herbarium images resulted in a total of 19,654 annotated bounding boxes. The distribution of annotation boxes for each organ, for a selection of 15 plant families is shown in Figure 3.9. Notably, 155 of these were either annotated or verified by an expert. This subset of verified images served as the test set for validating the model. The detailed list of annotations for each plant organ is shown in Table 3.2.

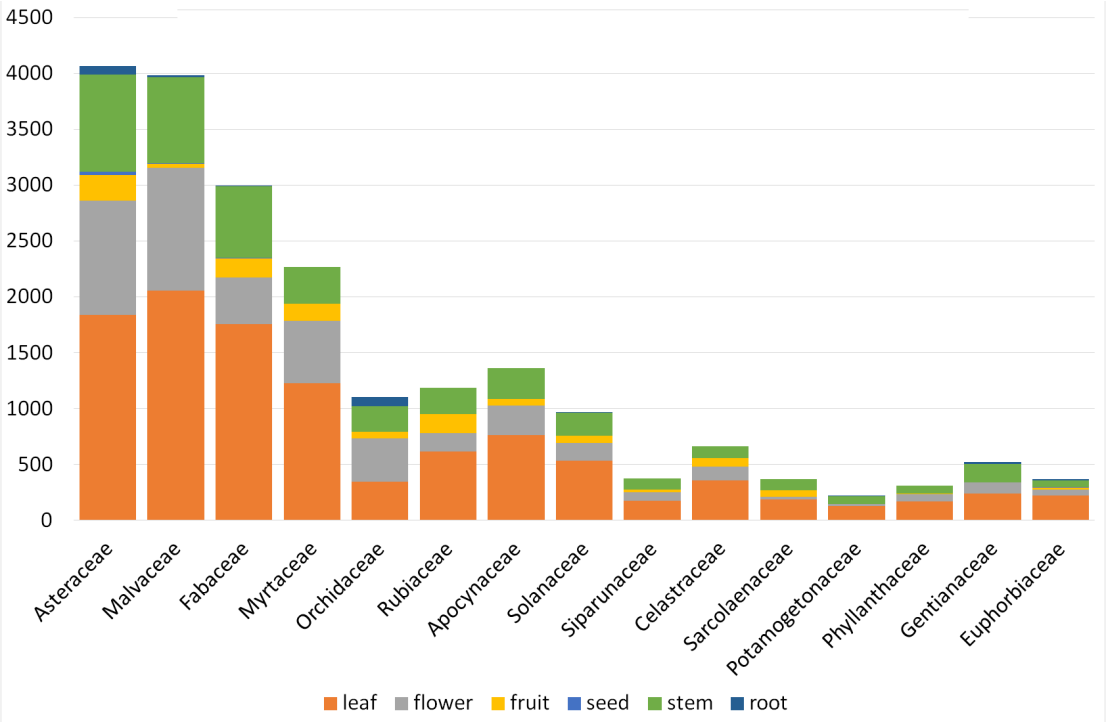


Fig. 3.9 Number of annotated organs for plant families. The variance in annotation per family can be due to different factors, such as phenology, season, and number of herbarium specimens.

Category	Training subset (498 images)	Test subset (155 images)	Complete dataset (653 images)
Leaf	7886	2051	9937
Flower	3179	763	3942
Fruit	1047	296	1343
Seed	4	6	10
Stem	3323	961	4284
Root	78	60	138
Total	15517	4137	19654

Table 3.2 The number of annotated bounding boxes for each plant organ in training and test set of MNHN dataset.

Implementation and Results

The plant organ detection task was performed with Faster R-CNN, with the Feature Pyramid Network backbone [58], as described in the Section 2.1.4. Given the constraints of a small training dataset, training the model from scratch was deemed impractical. As the initial layers of the convolutional neural network capture generic features about the basic shapes in the image, any large and diverse dataset can be used to train these layers. Therefore, in our approach we used a ResNet model pre-trained on the ImageNet dataset [17] and applied transfer learning [116], to fine-tune the parameters of the herbarium dataset, enhancing the efficiency of the training process. The object detection model was implemented using the Detectron2 library within the PyTorch framework [113], and trained using the Stochastic Gradient Descent optimizer on three TITAN Xp GPUs. The model trained on the MNHN herbarium scans was evaluated on a dataset of 708 full-scale herbarium scans from Herbarium Senckenbergianum (FR) [74], with a different set of 136 species and geographical origins, thus providing a robust evaluation scenario beyond the training data. The model performed very well and was able to successfully detect almost all plant organs in the Herbarium Senckenbergianum dataset. An example of plant organs detected on a herbarium scan, with their corresponding bounding boxes and confidence probability is shown in Figure 3.10.

The trained organ detection model was employed to generate a list of bounding boxes for each plant organ within herbarium scans, along with their respective names or class labels. Each prediction was also accompanied by the confidence level of the model. To assess the model's performance in organ detection, a widely recognized COCO evaluation method was employed [59]. This method determines the accuracy of each detected object, considering both the size and location of its bounding box. The COCO evaluation method calculates

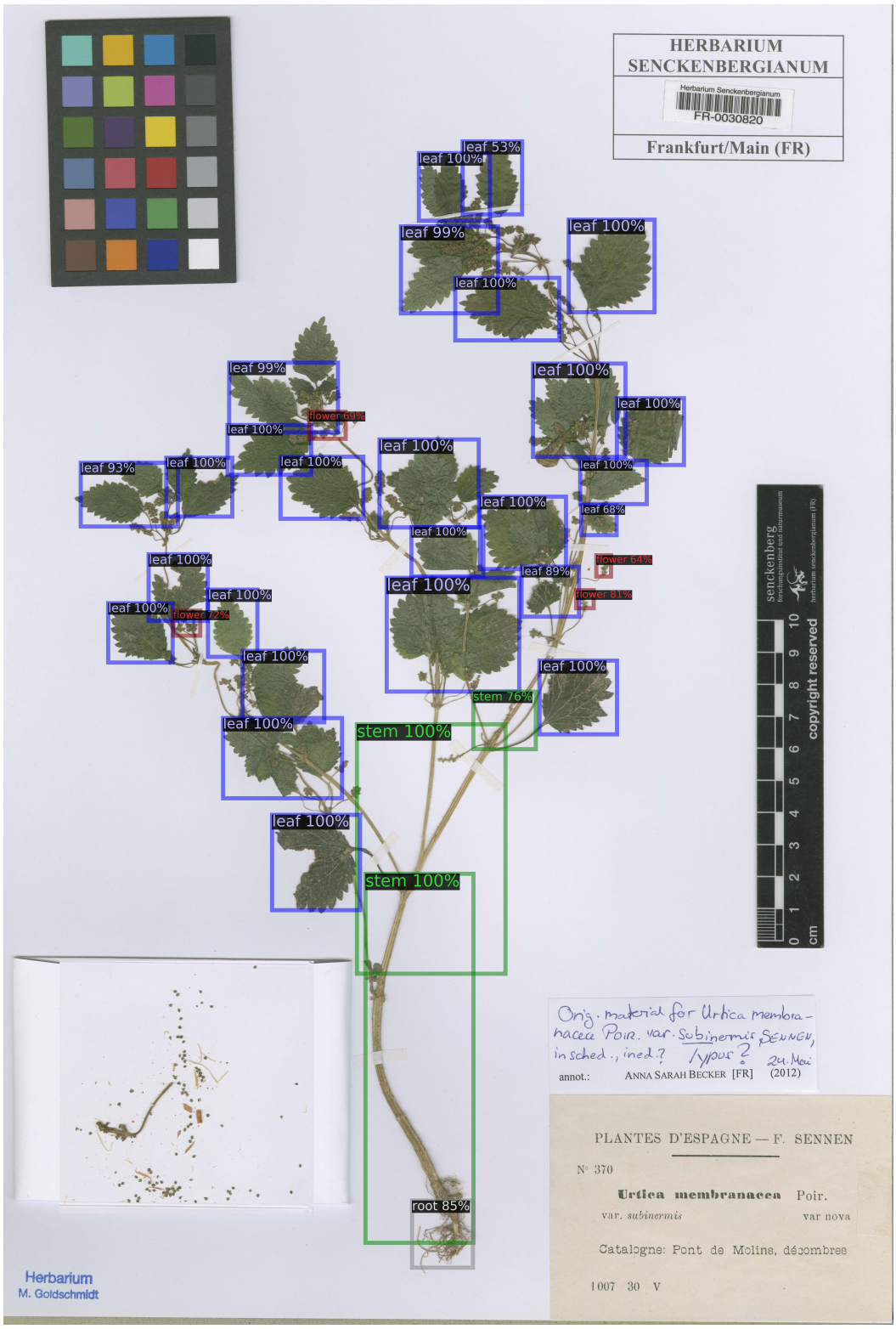


Fig. 3.10 A sample result of organ detection performed on a full scale Herbarium Senckenbergianum scan.

average precision (ranging from 0 to 100), which is a metric that combines both precision and recall of all the detections. The results of COCO evaluation for organ detection on the MNHN test dataset are presented in Table 3.3. Additionally, Table 3.4 showcases the model's performance on the Herbarium Senckenbergianum dataset.

Category	Bounding Boxes	AP
Leaf	2051	26.5
Flower	763	4.7
Fruit	296	7.8
Seed	6	0.0
Stem	961	9.9
Root	60	9.4

Table 3.3 Average Precision (AP) of each plant organ, with the total number bounding boxes in the MNHN test set.

Category	Bounding Boxes	AP
Leaf	3362	37.9
Flower	1921	18.3
Fruit	183	7.9
Seed	47	0.0
Stem	1063	25.1
Root	117	11.8

Table 3.4 Average Precision (AP) of each type of organ, with the number of bounding boxes in the Herbarium Senckenbergianum dataset.

As evident from these tables, the average precision (AP) values for leaves are the highest. This can be attributed to leaves having the highest number of annotations, as noted in Table 2. Conversely, the good precision values for stems and roots can be attributed to relative uniformity across the plant kingdom. In contrast, the morphological diversity of flowers and fruits posed a challenge to the model, while seeds presented difficulties due to their infrequent occurrence in herbarium specimens.

Our study stands as a pioneering effort in the detection of multiple plant organ types from herbarium scans, acknowledging the inherent biases in the annotated dataset arising from the natural distribution of different organs on a plant. The MNHN Paris Herbarium and Herbarium Senckenbergianum datasets used in this study had different geographical and

taxonomic focus. While these two datasets had some overlap at the family level, and partially at the genus level, the species-level overlap was minimal. This eliminates the possibility of species-specific features influencing organ detection in the Herbarium Senckenbergianum dataset. The observed imbalance of different organs in our dataset reflected the natural distribution of organs in the wild and the selection biases inherent in herbarium collections.

The detection of plant organs on herbarium specimens has applications across fields of research like ecology, botany, and agriculture. Beyond their taxonomic value, flowers and fruits in specimens can be a source of valuable data for phenological studies, particularly in the context of climate change [112]. Similarly, analysis of the roots may be used for identifying specimens that contain root symbionts, such as mycorrhizal fungi or nitrogen-fixing bacteria, thus offering an opportunity for exploration through microbiological or genetic methods [34]. As most computer vision approaches focus on live plants, especially in the context of agriculture, their focus is often limited to a specific set of taxa. In contrast, our approach stands out for its inclusivity, encompassing a broader and more diverse range of species, similar to many applications in citizen science on natural images [110]. Furthermore, our method extends its applicability across an extensive time scale by utilizing the vast collections of herbaria. This distinguishes our approach from recent similar methods, such as GinJinn [73] and LeafMachine [111]. GinJinn employs an object-detection pipeline for automating the detection of features such as from herbarium scans, by recognizing leaves. LeafMachine is another approach that detects the size, number, and type of leaves from digitized specimens using machine learning.

3.2.2 Detection of insects and moths in camera trap images

The object detection method used in the previous section was also applied to identify and monitor moths in camera trap images. The primary objective of this research was the evaluation of an automated moth trap (AMT) by comparing it to a conventional which is a lethal trap for all the captured insects. The designed automated moth trap was intended to be low cost, with robust design and ease to install overnight at the site.

It captures images of insects attracted by ultraviolet light against a white screen. An image of AMT deployed in the field is shown in Figure 3.11. This automated monitoring approach captures images of insects at high resolution throughout various phenological changes, making it possible to track shifts in ecological patterns and insect populations [13]. To detect the moths and insects, the object detection model, based on Faster R-CNN, was trained in two stages. In the first stage, the model was trained on 203 camera trap images recorded at different locations, which were annotated with bounding boxes and moth/insect labels with Labelbox. The predictions from this model served as unverified labels for the

second round. These predicted labels and bounding boxes underwent review and correction in the online image annotation tool PhotoDB, for the second round of training. As it is less labor-intensive to review and adjust annotated images than to manually label them, 1827 images were annotated in the second round. The images were combined with the images from the first round to train the model, which increased the size of the dataset. This resulted in the model having improved performance, especially reducing the false detection of large insects, such as moths, as multiple detections of insects in close proximity to each other. Figure 3.12 shows an example of moth and insect detection by the model on a camera trap image.

Fig. 3.11 Photo of automated moth trap setup in the field [69].

Fig. 3.12 Camera trap picture taken by AMT, with predicted bounding boxes around moths and insects.

3.3 Beyond stationary models: Incremental learning

In the rapidly evolving landscape of ecology and botany, the availability of newer datasets is continuously expanding. This growth is mainly attributed to the digitization efforts by museums and the invaluable contributions made by citizen scientists. With these datasets increasingly becoming available online, it is essential to keep the machine learning models up to date by integrating the rich and diverse wealth of new knowledge. This is particularly relevant in our case for plant organ detection. The labor-intensive task of annotating herbarium scans with bounding boxes limits the availability of labeled data. Consequently, the machine learning models need to be updated frequently to incorporate the latest training data and any modifications of existing data. This iterative learning approach is essential for machine learning models to maintain relevance and accuracy on trending datasets.

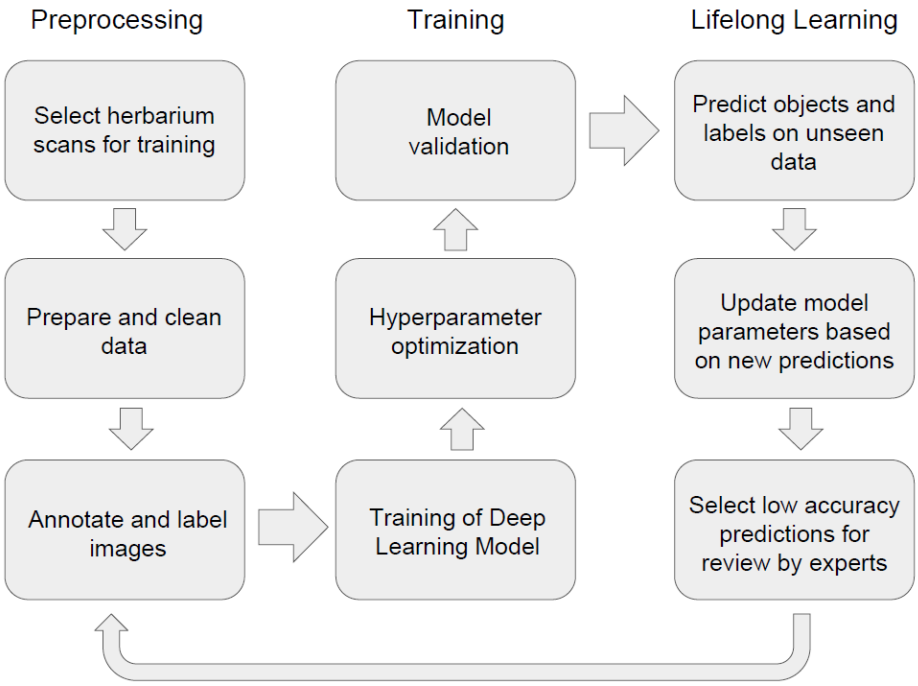


Fig. 3.13 A workflow for plant organ detection from herbarium scans with lifelong learning.

This necessity led us to devise a workflow for training based on lifelong learning for plant organ detection, as depicted in Figure 3.13. In the first stage, the herbarium scans are preprocessed and annotated. In the second stage, a model is trained from the annotated data, and finally, in the third stage, the model is deployed to predict labels on new data. Notably, a subset of the data on which the model had low confidence is selected for annotation. This newly annotated data is then used to retrain the model. This iterative learning process is called continual learning or lifelong learning. An overview of continual learning can be found

in Section 2.2. Through continual learning, the model can refine itself and stay attuned to the evolving datasets. The poster presented in the conference [117], illustrating the workflow of plant organ detection from herbarium scans using lifelong learning is included in the Appendix A.2.

3.3.1 Data-Free Continual Learning on Imbalanced Data

In many real-world scenarios, like the one mentioned above, data is often available incrementally instead of all at once. The new data could be from a previously unknown source or due to a dynamically changing environment. This requires the model to continuously learn from the evolving dataset. One conventional approach is to train a new model from scratch on the new data and existing data merged. While this method can produce a high-performance model due to the abundance of data, is impractical for many applications. For instance, there could be a scenario where the model needs prolonged training time due to a large dataset, and the new data becomes available before the training is complete, rendering the model outdated even before it is fully trained. Another scenario could be that the data cannot be saved permanently or locally due to privacy concerns. Additionally, the growing dataset may require an increasing amount of computing and memory resources, making it computationally or financially unfeasible.

An alternative approach to training on merged data is to exclusively train the model on the new data, making it more efficient. Although this approach offers some advantages such as reduced memory requirement and relatively constant training time, it leads the model to forget about the previously trained data and optimize only for the new data. This phenomenon, known as catastrophic forgetting, results from the interference of training on new data with patterns of previous data, resulting in a significant decline in the model's performance on old data [65].

To address this challenge of catastrophic forgetting, a balance is required between acquiring new knowledge and preserving old knowledge. This delicate equilibrium is termed a stability-plasticity dilemma [66]. It denotes that the stability of the model on the previous data with its plasticity must be balanced to make the model adaptable to new data. Achieving this balance is essential for machine learning to continuously adapt to new data without compromising the knowledge already learned.

Apart from the challenge of catastrophic forgetting, another significant obstacle for training deep learning models on images of nature is that real-world datasets, like Pl@ntNet [44], are inherently imbalanced and often exhibit long-tail distribution. This nonuniform distribution of images results in a scenario where a majority of images belong to a small number of classes. This kind of class imbalance presents considerable difficulties to the

deep learning process. To address this problem of imbalance, several strategies can be implemented. Some common approaches involve undersampling the images from majority classes, employing oversampling or applying data augmentation for the minority classes, and introducing synthetic data to rebalance the datasets. Within continual learning, rehearsal or replay-based methods are frequently employed [11], which share many similarities with the oversampling strategy.

In this thesis, we present a pioneering approach for data-free continual learning, named Data-Free Generative Replay (DFGR), which is specifically designed to cater to data-free class-incremental learning and imbalanced datasets. Class incremental learning refers to the scenario where the incoming data does not just consist of new instances of the same classes as previous data but introduces entirely new unseen classes. In this context of plant species recognition, where the species can be considered as classes, the new data contains images of plant species not present in the previous dataset.

In the DFGR approach, a generator is employed to synthesize the previous image dataset. Notably, this generator is not trained on the previous data but relies on knowledge transfer from the existing trained model. To effectively address the imbalanced data problem, DFGR incorporates focal loss during the model training and dynamically adjusts the generated images to balance the dataset. This dual-pronged strategy of addressing data-free learning on novel classes and imbalances within incoming data, results in making DFGR a robust approach to many real-world continual learning scenarios.

Related data-free continual learning approaches

Unfortunately, many continual learning methods require storing a subset of previous data either to merge it with the new data, or for training a generator that will reconstruct it during training. This can be impractical for many applications where the previous data is challenging to store or is no longer accessible, e.g., for privacy reasons [5]. In such cases, it is essential to have a continual learning approach that is not dependent on previous data. Two main approaches for continual learning without needing previous data are regularization-based methods and knowledge distillation.

Regularization-based methods overcome the need to store previous data by imposing restrictions on model parameters while learning new data, thus mitigating catastrophic forgetting. Regularization methods vary, with some regularizing the model weights, like Elastic Weight Consolidation (EWC) [48] and Synaptic Intelligence (SI) [121], by penalizing the changes in parameters considered important to previous data. Some other approaches, such as Learning without Forgetting (LwF) [57] and Learning without Memorizing (LwM)

[18], aim to prevent activation drift, which is the change in activations of the old network while learning new tasks by employing knowledge distillation.

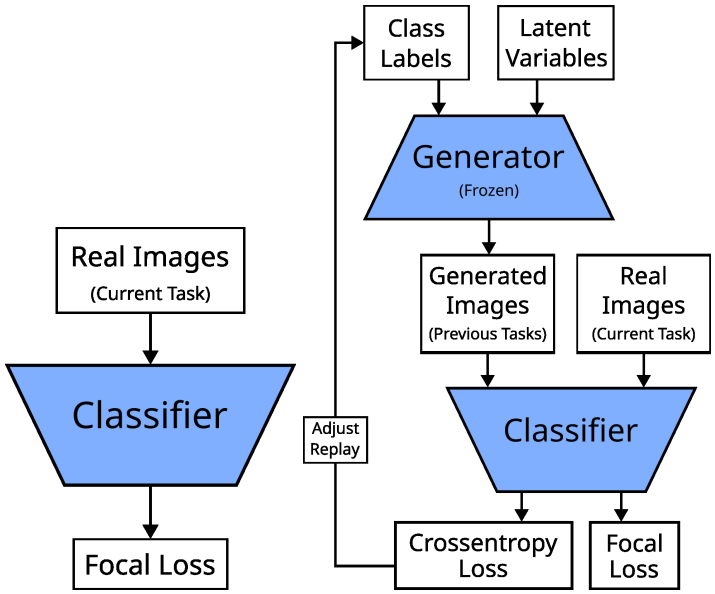
Knowledge distillation on the other hand transfers knowledge from the model trained on the previous data to the model trained on new data [36]. This is often achieved by retaining the previous knowledge in a generator and replaying it alongside the new data. Early attempts at synthesizing images without needing the previous training data include DeepDream [70], and DeepInversion [115]. DeepDream tries to generate realistic-looking images by minimizing the loss on the trained classification model. DeepInversion extends this by introducing a regularization term to improve image quality. Contemporary methods utilize multiple losses, including cross-entropy, batch normalization alignment, image smoothness, and information entropy, to enhance diversity in generated images [99, 114]. A detailed overview of various regularization and knowledge distillation methods is presented in Section 2.2.3.

Proposed approach

Our approach, named Data-Free Generative Replay (DFGR) addresses the challenge of imbalanced datasets with data-free continual learning by combining regularization methods and knowledge distillation. As regularization relies on a single model for learning new data and retaining old data, it falls short in terms of information retention compared to the knowledge distillation approach. Conversely, knowledge distillation requires a model trained on prior data to act as a teacher, which can have substantial memory and computational requirements. In our approach, to circumvent these challenges we directly transfer knowledge from the previous model to a generator, which is significantly smaller than the teacher. We achieve this by combining data reconstruction and regularization techniques.

The proposed DFGR method implements the learning process through two sequential stages. The initial stage is training or retraining the classifier model, and the second stage is training the generator. For training the classifier, we use focal loss to cater to the data imbalance [85], instead of the conventional cross-entropy loss [28]. DFGR first trains a classifier on all the available data illustrated in Figure 3.14a. Every time new data arrives belonging to novel classes, the classifier gets merged with the generated data for classifier training. The generated data is a reconstruction of all the previously encountered classes. The training process, explained in Figure 3.14b, illustrates the classifier's simultaneous training on current and generated data. To address the imbalance in the previous data, the generator automatically balances the replay data by adjusting the ratio of the classes of generated images.

Figure 3.15 shows the training process of the generator. The generator is based on a class-conditional BigGAN architecture [8]. It is trained to reconstruct images belonging



(a) (b)

Fig. 3.14 a) Workflow of the classifier training with real images, b) and retraining the classifier with real and generated images.

to previously seen data by leveraging the insights of the trained classifier and transferring knowledge from it. This is achieved by using various loss functions, to achieve the following objectives for the generated images.

Cross-entropy Loss: The images belong to the classes the model was trained on.

Batchnorm Loss: The generated images have similar data distribution to the real images.

Features Loss: The images exhibit similar high-level features as observed in real images.

Divergence Loss: There is diversity among the images, both inter-class and intra-class.

Smoothing Loss: The images look realistic with minimum artifacts.

Classifier training/retraining phase adopts the focal loss, initially designed for detecting objects of varying sizes, to overcome the challenge of data imbalance with unknown class distribution. Focal loss is a modified version of cross-entropy loss with a reweighing of losses for different classes. Besides focal loss, another technique employed to cater to class imbalance was generator replay adjustment. The replay adjustment dynamically changes the frequency or probabilities of classes within a batch of generated images, based on the average loss per class. During training, the total loss for the classifier is the combination of the focal loss of the current data and the cross-entropy loss of the generated data.

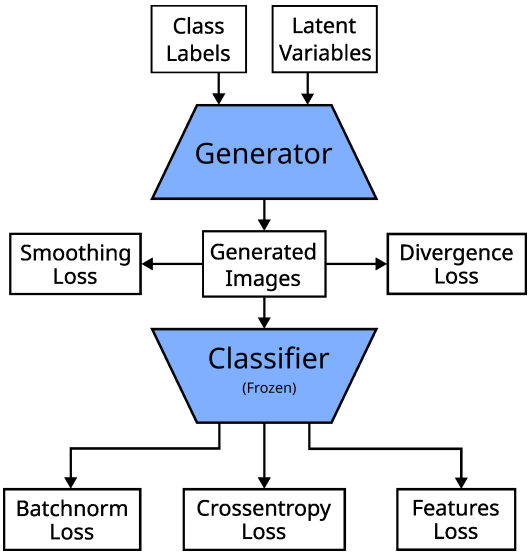


Fig. 3.15 Workflow for training the generator.

Generator training phase is designed to attain the five main objectives for the generated images outlined above. These objectives were achieved through a combination of various loss functions. The first loss function employed was traditional cross-entropy loss for the classification task. The second loss function (batch normalization loss) was introduced to minimize the disparity in batch normalization statistics between real and generated image samples. By aligning itself to the batch normalization means and variances of the trained model, the generator was trained to emulate the feature maps and distribution of the real data. Similar to the batch normalization loss, the feature map loss aimed to minimize the distance between features, representing the high-level features of the images, for both real and generated images. To enhance the image diversity, a sample diversification loss was applied to prevent potential overfitting, by minimizing Jensen-Shannon divergence between two random subsets of the generated images. Finally, to conform to a realistic look, image smoothing was applied to the generated images, through the use of a Gaussian kernel. All these five loss functions were then combined for training the generator. Furthermore, extensive tests of different combinations of these loss functions evaluated their impact on the DFGR's learning performance, both in balanced and unbalanced scenarios.

Implementation and Experimental Setup

DFGR combines ResNet and BigGAN architectures for its classifier and generator models. ResNet, renowned for extracting image features, is used to classify images. Conversely, BigGAN is an advanced generator model for creating realistic images. DFGR was imple-

mented in the PyTorch framework on a single RTX 3060 GPU. In addition, it was subjected to extensive tests with the benchmark MNIST Digits and FashionMNIST datasets. While the original datasets are inherently balanced, the tests also introduce artificial imbalances to evaluate the effectiveness of our approach. For incremental learning, the datasets were divided into three sub-datasets or tasks, each exclusively containing certain classes of images, as denoted in Eq 3.1.

$$BalancedData: \begin{cases} T_1: & \{3,4,9\} \\ T_2: & \{5,6,0\} \\ T_3: & \{1,2,8,7\} \end{cases} \tag{3.1}$$

Additionally, a predefined portion of images were selected from each class to simulate imbalanced data, ranging from 10% to 100%, as depicted in Eq 3.2.

$$ImbalancedData: \begin{cases} T_1: & \{3:1.0,4:0.6,9:0.3\} \\ T_2: & \{5:0.9,6:0.4,0:0.2\} \\ T_3: & \{1:0.5,2:0.7,8:0.1,7:0.8\} \end{cases} \tag{3.2}$$

Results

Our approach, DFGR, was extensively evaluated on MNIST and FashionMNIST datasets, both with balanced and imbalanced dataset scenarios. This involved exploring four combinations of three loss functions for class incremental learning: standard loss, cross-entropy loss, and feature map loss. The standard loss is a combination of three separate loss functions, responsible for image smoothing, sample diversification, and batch normalization losses.

Additionally, the impact of replay adjustment, for re-balancing generated data, was also tested. The results, presented in Table 3.5, indicate that the combination of replay adjustment with loss functions mentioned above yielded the highest accuracy for our approach, both on balanced and imbalanced datasets. The table shows that cross-entropy loss played a significant role in enhancing the accuracy of the model, particularly when combined with replay adjustment. Furthermore, the feature map loss only marginally improved the accuracy compared to the standard loss.

We also compared our method with two data-free baseline methods [48, 57] and two similar advanced methods, named MFGR [114] and Always Be Dreaming (DFCIL) [99]. As listed in Table 3.6, the results demonstrate that DFGR outperformed other methods across many metrics, namely accuracy, training time, and memory efficiency. DFGR's ability to

Methods	MNIST Bal.		MNIST Imbal.		FMNIST Bal.		FMNIST Imbal.	
	Acc.	Avg. Time	Acc.	Avg. Time	Acc.	Avg. Time	Acc.	Avg. Time
l_s	45.8	10:38	52.7	8:31	40.0	13:20	39.8	12:23
$l_s + l_{feat}$	71.7	10:53	58.7	8:49	40.6	13:11	39.7	10:10
$l_s + l_{ce}$	79.1	11:59	80.5	11:13	43.1	11:54	40.2	11:07
$l_s + l_{ce} + l_{feat}$	81.7	9:25	81.5	8:09	45.1	11:33	40.3	9:32
$l_s + ra$	58.0	11:43	53.9	9:43	41.4	10:47	40.0	9:54
$l_s + l_{feat} + ra$	78.9	9:30	59.7	8:38	43.3	9:51	40.4	7:56
$l_s + l_{ce} + ra$	87.5	8:45	87.4	8:24	43.6	8:29	42.2	8:31
$l_s + l_{ce} + l_{feat} + ra$	88.4	8:35	88.5	7:10	46.6	8:40	43.6	8:20

Table 3.5 Accuracy (in %) and average run times (hh:mm) for every dataset, with different combinations of loss functions and replay adjust (*ra*).

address overall class imbalance during testing was unmatched by other methods, enhancing its applicability and effectiveness in real-world scenarios.

In conclusion, DFGR stands out as a novel approach for incremental learning on imbalanced datasets, offering high accuracy without the need to store previous data. The main feature distinguishing it from other similar ones like MFGR and DFCIL is employing a small generator trained using the classifier for replaying previous data instead of using a student-teacher approach. Incorporating various loss functions, such as feature map loss, focal loss, and generator replay adjustment, contributes to the effective training of the generator, handling imbalanced data, and augmenting past data during replay, respectively. DFGR represents a promising step toward data-free continual learning with limited resources and imbalanced datasets. The research article for this approach is attached below in Section 4.3.

Methods	Models Saved	Model Size (Millions)	MNIST Bal.		MNIST Imbal.		FMNIST Bal.		FMNIST Imbal.	
			Acc.	Avg. Time	Acc.	Avg. Time	Acc.	Avg. Time	Acc.	Avg. Time
Naive (Lower Limit)	-	-	41.5	0:34	41.1	0:26	39.9	0:44	39.2	0:22
EWC [48]	-	-	47.5	1:23	46.9	0:52	39.9	1:28	39.5	0:58
LWF [57]	Classifier	19.6 M	58.2	1:10	55.5	0:38	41.4	1:08	40.3	0:44
MFGR [114]	Classifier + Generator	19.6 M + 3.2 M	66.2	15:32	65.8	16:35	42.3	16:05	41.2	16:32
DFCIL [99]	Classifier + Generator	19.6 M + 3.2 M	83.2	13:17	81.1	14:43	48.3	13:27	32.9	14:25
DFGR (Ours)	Generator + Features	3.2 M + 41 K	88.4	8:35	88.5	7:10	46.6	8:40	43.6	8:20

Table 3.6 Accuracy (in %) and average run times (hh:mm) for baseline and competitive methods.

4. Publications

4.1 Publication 1:

Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks

This publication presents a novel method for species and trait identification of vascular plants from digitized herbarium scans using deep learning. This is the first study to be best our knowledge that addresses several traits across a large number of taxa. Our proposed method is a Convolutional Neural Network (CNN) image recognition model for identifying the 1000 most frequently documented species of herbarium scans in the Muséum national d'Histoire Naturelle (MNHN) collection, accessible on the GBIF portal. This method also integrates the identification of morphological traits of the herbarium specimen, via their species. The approach demonstrates excellent performance in accurately recognizing taxa from herbarium specimens, with well-rounded recognition of traits.

Contribution Role: Lead Author

Published in:

Botany Letters, 2018

BOTANY LETTERS, 2018
https://doi.org/10.1080/23818107.2018.1446357

Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks

Sohaib Younis[a,b], Claus Weiland[a], Robert Hoehndorf[f], Stefan Dressler[e], Thomas Hickler[a,d], Bernhard Seeger[b] and Marco Schmidt[a,c,e]

[a]Data and Modelling Centre, Senckenberg Biodiversity and Climate Research Centre (SBiK-F), Frankfurt am Main, Germany; [b]Department of Mathematics and Computer Science, University of Marburg, Marburg, Germany; [c]Palmengarten der Stadt Frankfurt am Main, Frankfurt am Main, Germany; [d]Department of Physical Geography, Goethe University, Frankfurt am Main, Germany; [e]Department of Botany and Molecular Evolution, Senckenberg Research Institute and Natural History Museum Frankfurt, Frankfurt am Main, Germany; [f]Computer, Electrical and Mathematical Sciences and Engineering Division, Computational Bioscience Research Center, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

ABSTRACT

Herbaria worldwide are housing a treasure of hundreds of millions of herbarium specimens, which are increasingly being digitized and thereby more accessible to the scientific community. At the same time, deep-learning algorithms are rapidly improving pattern recognition from images and these techniques are more and more being applied to biological objects. In this study, we are using digital images of herbarium specimens in order to identify taxa and traits of these collection objects by applying convolutional neural networks (CNN). Images of the 1000 species most frequently documented by herbarium specimens on GBIF have been downloaded and combined with morphological trait data, preprocessed and divided into training and test datasets for species and trait recognition. Good performance in both domains suggests substantial potential of this approach for supporting taxonomy and natural history collection management. Trait recognition is also promising for applications in functional ecology.

ARTICLE HISTORY
Received 8 December 2017
Accepted 20 February 2018

KEYWORDS
Herbarium specimens; species recognition; convolutional neural networks; morphological traits; trait recognition; digitization

Introduction

Herbaria have been the foundation for systematic botanical research for centuries, harboring the type specimens defining plant taxa and documenting variability within them. Up to now, about 3000 herbaria have accumulated nearly 400 million specimens (Thiers 2017). These immense collections are rapidly becoming more accessible: the Natural History community presently experiences a major increase of digitization activities, with botany in Africa profiting from the large-scale digitization of type specimens in the context of the African Plants Initiative (Smith et al. 2011).

While the availability of digitized plant specimens is constantly improving due to automated digitization streets and major collections are achieving almost complete coverage of their collections (Le Bras et al. 2017), simultaneously the opportunities for computer-based image recognition are also rapidly improving, especially with the rise of deep-learning methods.

Deep learning is a part of machine learning methods for learning data representation. It processes the data in multiple layers, which leads to multiple levels of data abstractions, also known as features in image processing, for learning the representation (LeCun, Bengio, and Hinton 2015). A typical deep-learning network consists of multiple layers of artificial neural networks, inspired by biological neural networks (Olshausen and Field 1996). The main objective of a neural network is to learn a pattern or an approximation function based on its input. The connections have numeric weights that control the signals between neurons, which can be tuned based on experience, making neural net works adaptive to input and capable of learning (Schmidhuber 2015).

There are many types of deep-learning networks but in our experiment we use convolutional neural networks (CNN) (LeCun and Bengio 1995), as their main application is in image classification. The organization and connectivity of neurons in convolutional network is biologically inspired by animals' visual cortex (Matsugu et al. 2003; Hubel and Wiesel 1968). Their success in the field of computer vision and image classification can be attributed to their need of relatively little or no preprocessing of images compared to other machine-learning algorithms, which require the calculation of certain statistical properties of the image before learning or in some cases hand-engineered feature design. This is achieved by stacking multiple convolution layers, which

CONTACT Sohaib Younis Muhammad-Sohaib.Younis@senckenberg.de

Published online 13 Mar 2018

apply a convolution operation to the input with a kernel, producing a feature map and passing it as input to the next layer. As the network learns, the kernels in each layer are updated to improve the feature maps for the classification task. The initial layers in the network compute primitive features on the image such as corners and edges. The deeper layers use these features to compute more complex features consisting of curves and basic shapes and the deepest layers combine these shapes and curves to create recognizable shapes of objects in the image (Yosinski et al. 2014; Zeiler and Fergus 2014).

Convolutional neural networks have received attention recently due to their exceptional performance in ImageNet competitions. They have also demonstrated impressive results in recognition of plants in PlantCLEF challenges since 2015 (Goëau, Bonnet, and Joly 2017).

Botanical applications of image recognition include taxon recognition from photos in the context of citizen science, eg by Pl@ntnet (Joly et al. 2016) or iNaturalist (Van Horn et al. 2017) or specially designed apps such as LeafSnap (Kumar et al. 2012). Up to now, there have been only few and very recent applications on images of herbarium specimens (Carranza-Rojas et al. 2017; Schuettpelz et al. 2017; Unger, Merhof, and Renner 2016). Our present approach includes taxon recognition as well as the recognition of morphological traits from herbarium specimens. Traits have so far only rarely been a subject in image recognition from plant images. There are however some approaches to extract leaf traits from plant images including specialized software and semiautomated workflows that are comparatively work-intensive (eg Corney et al. 2012). We are focussing on taxa with a high number of images; in the context of trait recognition we focus further on African taxa, because of good availability of morphological trait data via a knowledge base generated within the framework of the Flora Phenotype Ontology (FLOPO; Hoehndorf et al. 2016) and African Plants - a photo guide (Dressler, Schmidt, and Zizka 2014), but also because the existing taxon recognition presently works best for North American and European taxa and we are aiming to improve taxon recognition for a region, where taxonomic expertise and resources for identification are still less available and much needed.

Some deep-learning approaches for recognising traits already exist that focus mainly on features like leaf count (Ubbens and Stavness 2017) and leaf tip/base (Pound et al. 2017) on selected model organisms to monitor performance of different cultivars or under different growth conditions. To our knowledge, this is the first study to deal with several traits in a large number of taxa, implying more abstraction in the concept of a trait and variability within a trait to be recognized.

Materials and Methods

Our general approach in the recognition of taxa and traits from specimen images is to label the images with both of these informations, the taxon name being already included in the image data from the original data provider and the trait data being connected via the taxon name.

Taxon data

In order to resolve synonymies resulting from different concepts in our trait databases and the data accompanying the herbarium scans, we used the GBIF taxon backbone (GBIF 2017). Since scans have been found and downloaded via GBIF, names and taxon IDs from GBIF have already been attached to the images. Names from our trait data (FLOPO knowledge base/African Plants – a photo guide) have been matched to the GBIF backbone using the Global Names Resolver (http://resolver.globalnames.org/). Only taxon name matches with a score > 0.9 have been used to label images with trait data.

Trait data

While herbarium scans in GBIF are labelled with a taxon name, traits are not directly connected with the scan and need to be linked via the taxon name, the approach is described above. We assigned these traits to all herbarium scans of a given taxon. This implies, that individual herbarium scans may or may not show all traits connected with the taxon. The trait data-set used in this study is from the multi-entry identification key of 'African Plants - a photo guide' (Brunken et al. 2008; Dressler, Schmidt, and Zizka 2014) enhanced by trait data extracted from Floras using text-mining, annotated by a domain ontology (FLOPO) and combined in a knowledge base of plant traits which we consulted via a SPARQL endpoint (http://semantics.senckenberg.de/sparql).

The assignment of traits to all herbarium scans of a taxon will result in individual scans being labelled with traits not recognizable from the given plant material (eg flower symmetry in a specimen without flowers). In order to use traits, which are really shown on a herbarium scan, we focused on a reduced set of leaf traits, considered to be recognizable in the majority of herbarium scans. These leaf traits include leaf arrangement, leaf structure, leaf form, leaf margin and leaf venation.

Images

Our specimen imagery is from open access images contributed to the GBIF portal, a large part of these consisting of the MNHN (Muséum national d'Histoire naturelle) vascular plant herbarium collection data-set in Paris (Le Bras et al. 2017). From several million digitized specimens on GBIF, we downloaded scans of the 1000 species with most herbarium scans available via GBIF, consisting of a total of 830,408 images. The distribution of scan images per species is heterogeneous, *Thymus pulegioides* L. with the most images (5494) and *Orchis anthropophora* (L.) All. with the least images (532).

For the trait recognition, we extracted a subset of these (170 species / 152,223 scans) with trait data available via the sources mentioned above. The data-set has been divided into 70% for training, 10% for validation and 20% for testing. The division is done uniformly for all species, regardless of the number of images belonging to it.

In order to investigate the role of the number of images on the accuracy of species recognition, we further extracted reduced datasets with 75, 50 and 25% of the total number of images, evenly reduced for each species.

Image preprocessing

All images in the data-set have been uniformly cropped and resized in portrait format after downloading them from GBIF. As shown in Figure 1, a typical herbarium sheet in this data-set contains a label with information on the collection event and the taxon as well as collector and number, plus, in many cases, further annotations by other scientists working with the collection object. It also contains barcodes on the top and bottom of the sheet and sometimes reference color bar on the sides. In order to reduce the background noise in the picture from the barcodes and color bars, they were uniformly cropped from the pictures, as suggested by Carranza-Rojas et al. (2017). One more reason to crop the pictures' labels and notes is not to let the deep-learning network learn any author or collector bias from the labels and tags on the herbarium sheets – even if individual characters may not be readable at the final resolution, shapes of words, specimen labels, annotation labels used by taxon specialists, etc. may roughly correspond with taxa. All images were cropped 7.5% from left and right in order to remove the reference color bars, 5% from top and 20% from bottom, to remove the bar codes and notes on the specimen. The original images had an average size of c. 5100 by 3500 pixels, therefore the images were then resized to 292 by 196 pixels, in order to preserve the aspect ratio of the sheet (Figure 1) and reduce the number of pixels for the network to learn in order to speed up the learning. The network was slightly modified to process this custom dimension of the images as shown in Table 1.

Network architecture

The image recognition task was done by using a slightly modified ResNet model (He et al. 2016) implemented using the Tensorflow framework (Abadi et al. 2016). A typical residual convolution network uses a raw RGB image with dimensions of 225 by 225 as input with 3 three color channels (red, green and blue) and consists of blocks of convolution layers, a few pooling layers and a fully connected layer at the end, as shown in Table 1. It also contains skip connections to cater for vanishing gradients and degradation of information due to the depth of the network from large number of layers. These connections bypass the convolution layers in each block by carrying the output of the previous block and combining it with the output of the current block without any processing.

Figure 2 shows a block in ResNet, consisting of three convolutional layers. The first layer of 1×1 convolution reduces the number of filters for the next 3×3 layer, thus creating a bottleneck for the second layer. The third layer of 1×1 increases the number of filters again for the next block. The bottleneck in the block leads to a higher number of filters without increasing the model complexity. Each layer is followed by batch normalization (Ioffe and Szegedy 2015) and an activation function, except the last layer where only batch normalization is used. The output of this batch normalization and previous block is added and then passed through an Exponential Linear Units

Figure 1. Image processing: herbarium scans as downloaded have been cropped and reduced to standard size in order to prepare them for treatment in deep learning algorithms.

Table 1. Sequence of layers in the ResNet and dimensions of features for each layer. Since the images for herbarium scans are resized to dimensions of 196 by 292, the last average pooling layer had to be modified to cater for the change in image size.

Layer Type	Filter Size / Stride	Output Size
Convolution	7 × 7 / 2	146 × 98 × 64
Max Pool	3 × 3 / 2	72 × 48 × 64
Convolution	[1 × 1, 3 × 3, 1 × 1] × 3	72 × 48 × 256
Average Pool	2 × 2 / 2	36 × 24 × 256
Convolution	[1 × 1, 3 × 3, 1 × 1] × 4	36 × 24 × 512
Average Pool	2 × 2 / 2	18 × 12 × 512
Convolution	[1 × 1, 3 × 3, 1 × 1] × 6	18 × 12 × 1024
Average Pool	2 × 2 / 2	9 × 6 × 1024
Convolution	[1 × 1, 3 × 3, 1 × 1] × 3	9 × 6 × 2048
Average Pool	9 × 6 / 9 × 6	1 × 1 × 2048
Fully Connected (Softmax)	1000 dense	1000

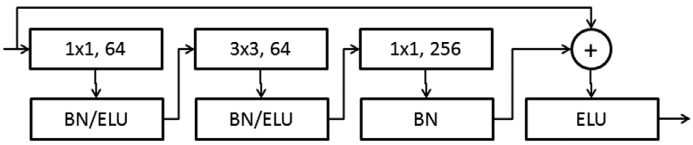


Figure 2. A bottleneck ResNet block.

(ELU) activation function to speed up learning (Clevert, Unterthiner, and Hochreiter 2015), which is then fed as input to the next block. As shown in Table 1, the network is exactly the same as a conventional ResNet, except the average pooling layer where the filter size and strides of the layer are changed to cater for the custom dimension of the image.

The network was only trained on the herbarium data-set as downloaded via the GBIF network and transfer learning from any other data-set (Yosinski et al. 2014) was not used, as the total number of images was considered sufficient for training, and we preferred to avoid introducing a training bias by transfer learning from another similar data-set.

The model was implemented using the Tensorflow framework on a TITAN Xp GPU, we chose an approach using an Adam optimizer with Nestrov Momentum (Dozat 2016) in training. We started training the network with a batch size of 60 and learning rate of 1e-4 for 20 k steps. It was then trained further with a batch size of 120 with learning rate of 5e-5 for 20 k and then rate of 1e-5 for another 30 k steps. The model processed the validation data-set every 500 steps during the last two stages of training to calculate the validation error. The training was early stopped if the validation error did not decrease for the last 2500 steps.

For recognizing the traits, a smaller plain ResNet without bottleneck was used, since the total number of leaf traits in the data-set is limited to 19. As the scans have more than one unique trait, a sigmoid activation function was used in the last layer instead of softmax.

Results

The network successfully predicted the correct species for the vast majority of images in the test data-set. Out of 165,689 test images, the network was able to correctly predict the species of 82.4% of the images correctly, in 96.3% of the images, the correct species was among the five most probable predictions. The most accurately predicted species was *Phlegmariurus phlegmaria* (L.) Holub, with 137 out of 138 images correctly predicted. The least accurately predicted species was *Rosa corymbifera* Borkh, with only 11 out of 157 images correctly predicted.

Generally, probabilities for the correct species and accuracy values were high (> 0.8) throughout the data-set; species with fewer images, however, more often had lower values (Figure 3).

The network modified for trait recognition was able to predict, out of 30,374 test images, 89.6% of the traits successfully, including 30.9% of images with all the traits correctly. However, these numbers vary considerably according to the individual traits. The most accurately predicted trait was Leaf - Structure (simple), with an accuracy of 97% and 21,928 out of 22,595 test images correctly predicted. The least accurately predicted trait was Leaf - Structure (trifoliolate), with an accuracy of only 9.1% and 115 out of 1266 test images correctly predicted (see Table 2 for all traits).

Running the network with a reduced number of images in order to investigate the role of the number of images on accuracy (Figure 4) showed accuracy rising with number of images most strongly for the Top1 predictions between 25% and 50% of the images in our data-set and slightly levelling with a higher number of images.

Discussion

The paper presents an application of deep learning on herbarium scans in order to identify the taxon represented by the herbarium specimen and to recognize its traits. Our approach proved to be very efficient in recognizing taxa from herbarium specimens and on average also performed well for traits.

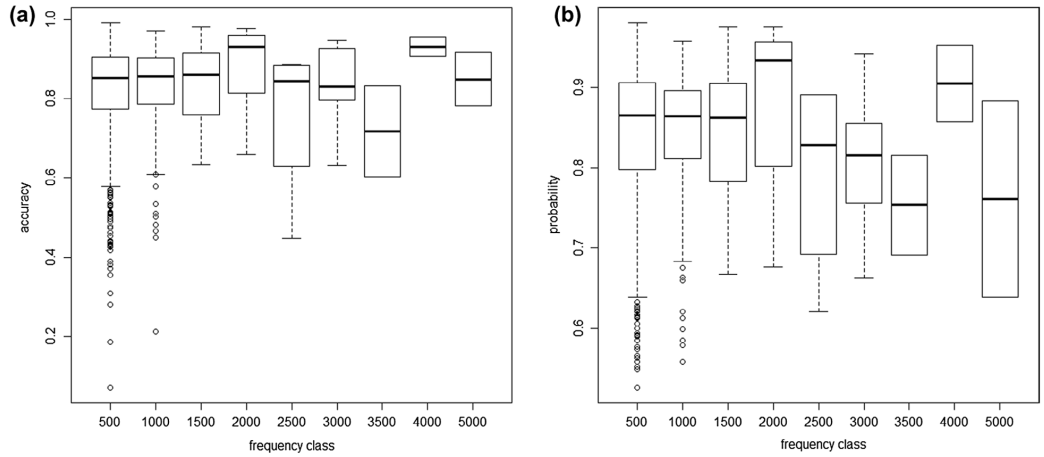


Figure 3. Species recognition from herbarium scans: performance measures depending on number of herbarium images (the frequency class includes all species with a number of images between × and ×+500): (a) boxplots of accuracy depending on number of test images per species . While the best-recognized species in each frequency class all have values > 0.9, median values increase slightly, but especially minimum values increase strongly with number of images. (b) boxplots of probability values for scans to be assigned to the correct taxon (mean probability of all test images of a taxon).

Table 2. Leaf traits used in this study with OBO ID and number of herbarium scans labelled to have this trait. The last column shows the accuracy of predicted scans for each trait.

trait	state	OBO ID	# scans	accuracy(%)
Leaf - Arrangement	alternate	FLOPO:0001032	120,514	92.98
Leaf - Arrangement	opposite	FLOPO:0000420	34,262	36.8
Leaf - Arrangement	rosulate	FLOPO:0900066	37,459	63
Leaf - Arrangement	whorled	FLOPO:0002264	7550	44.61
Leaf - Form	cordate	FLOPO:0900069	10,378	29.81
Leaf - Form	deeply lobed	FLOPO:0006834	28,900	59.79
Leaf - Form	oblong to linear	FLOPO:0000103	86,644	81
Leaf - Form	orbicular	FLOPO:0017811	8032	23.78
Leaf - Form	ovate or elliptic etc.	FLOPO:0000286	91,954	89.83
Leaf - Margin	entire	FLOPO:0900073	118,297	87.3
Leaf - Margin	not entire (serrate or crenate etc.)	FLOPO:0900074	59,148	72.5
Leaf - Structure	palmately compound	FLOPO:0018499	2268	46.42
Leaf - Structure	pinnately compound	FLOPO:0907004	46,827	68.62
Leaf - Structure	simple	FLOPO:0000693	128,391	97
Leaf - Structure	trifoliolate	FLOPO:0900067	8711	9.1
Leaf Venation	palmate	FLOPO:0900070	17,275	48.11
Leaf Venation	parallel	FLOPO:0900072	40,710	89.57
Leaf Venation	pinnate	FLOPO:0000561	102,663	90.35
Leaf Venation	triplinerve	FLOPO:0900071	7372	21

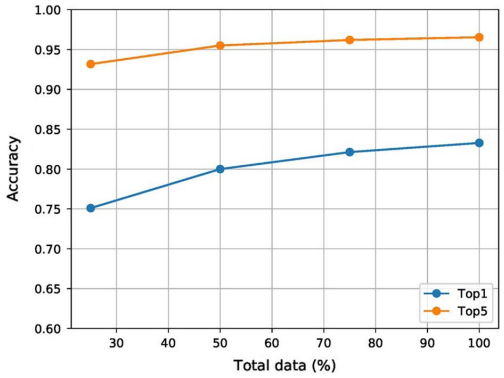


Figure 4. Accuracy of species recognition depending on the number of images available for training (25%, 50%, 75%, 100%).

The only other similar approach for deep learning-based species recognition of herbarium specimens with a large number of species is by Carranza-Rojas et al. (2017) with comparable results of 90.3% top five accuracy on 1204 species on a total of 253,733 images. To compare these results, it needs to be considered that in our study we had c. 16% less species and four times more images.

The performance of species recognition from herbarium images in our study was very high, to 96.3% when the top five predictions were considered. However, this is largely due to high numbers of training data, as demonstrated in Figure 4. Especially, the number of badly recognized species was decreasing with an increase of digitized images available (Figure 3).

Although the number of images for most traits was much higher than the number of images for individual species, trait recognition was not performing equally well. While for humans it is much easier to recognize a simple trait as used in our study than to correctly identify a species, from the networks' perspective, taxon-specific patterns seem to be easier to grasp than the more generalized and variable concept of traits.

Trait recognition differed much between traits and seemed to depend to a large part on the number of training samples: The best recognized traits with accuracies > 80% usually were labelled to at least 80,000 scans; traits with less than 10,000 scans never reached accuracies > 50%. However, especially in the traits with fewer available scans, accuracy varies considerably and cannot be explained only by the number of scans. As traits are often characteristic for taxa from genus to family level, correlations between the documented traits (and many more undocumented ones) may play a role here and taking phylogenetic relations and trait correlations into account would be interesting in further studies on trait recognition.

Conclusion

The good performance of species and trait recognition from herbarium scans even for a data-set consisting of a large number of species is promising in the context of digitization activities worldwide. Automated species recognition may become a valuable tool for the taxonomist and technical staff facilitating identification and bringing possible misidentifications of herbarium specimens to the attention of the responsible curator. In the context of our study, identification performance is high enough to give valuable suggestions in the identification process and pre-identifications that may be used before assigning collection material to a taxon expert. We could well imagine the implication of deep-learning algorithms in herbarium workflows, especially if material is digitized already at an early stage. However, we used species with a large number of herbarium images but for the majority of the c. 350,000 known species of plants, there are presently only very few or no image available. Further study is needed to include these. Trait recognition is interesting also from an ecological perspective, considering the adaptations of leaf form (Givnish 1987) that could be extracted from herbarium material and live plant photos.

The application of deep learning on natural history collections is a very recent development. Further improvements in the near future may be expected especially from improved algorithms, higher availability of herbarium scans by ongoing digitization activities and faster computers allowing also larger image sizes to be processed and thereby to include more details.

Acknowledgements

We acknowledge the efforts of herbaria worldwide to digitize their collections and make scan images available via GBIF. We further want to thank all contributors to our trait data in the context of the FLOPO knowledge base and African Plants - a photo guide.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

SY, MS and SD received funding from the DFG Project *Mobilization of trait data from digital image files by deep learning approaches* (grant 316452578). Parts of RH's & CW's work were funded by the National Bioscience Database Center (NBDC) and the Database Center for Life Science (DBCLS) Biohackathon 2017 grants. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the TITAN Xp GPU to CW used for this research.

Notes on contributors

Sohaib Younis is a computer scientist at Senckenberg Biodiversity and Climate Research Center with focus on deep learning and image processing. *Contributions:* convolutional network modeling, image preprocessing, description of results and preparation of manuscript.

Claus Weiland is scientific programmer at SBIK-F's Data & Modelling Centre with main interests in large-scale machine learning, trait semantics and scientific data management. *Contributions:* Flora Phenotype Ontology and knowledge base, design of the GPU platform, data analysis and preparation of the manuscript.

Robert Hoehndorf is assistant professor for computer science at the King Abdullah University of Science and Technology. His research interests are artificial intelligence, knowledge representation, biomedical informatics, ontology. *Contributions:* stimulation and concept of study, design and implementation of workflow for the Flora Phenotype Ontology and knowledge base.

Stefan Dressler is curator of the phanerogam collection of the Herbarium Senckenbergianum Frankfurt/M., which includes its digitization and curation of associated databases. Taxonomically he is working on Marcgraviaceae, Theaceae, Pentaphylacaceae and several Phyllanthaceous genera. *Contribution:* Trait data.

Thomas Hickler is head of SBIK-F's Data & Modelling Centre and Professor for Biogeography at the Goethe University Frankfurt. He is particularly interested in interactions between climate and the terrestrial biosphere, including potential impacts of climate change on species, ecosystems and associated ecosystem services. *Contribution:* Preparation of manuscript, comprehensive concept of study within biodiversity sciences.

Bernhard Seeger is professor of computer science systems at the Philipps University of Marburg. His research fields include high-performance database systems, parallel computation and real-time processing of high-throughput data with a focus on spatial biodiversity data. *Contribution:* Provision of support in machine learning and data processing.

Marco Schmidt is a botanist at Senckenberg Biodiversity and Climate Research Center (SBIK-F) with a focus on African

savannas and biodiversity informatics (eg online databases like *African Plants - a photo guide* and *West African vegetation*) and is working at Palmengarten's scientific service, curating living collections and collection databases. *Contributions:* concept of study, workflow, taxon and trait data, preparation of manuscript.

ORCID

Sohaib Younis http://orcid.org/0000-0001-9171-783X
Claus Weiland http://orcid.org/0000-0003-0351-6523
Robert Hoehndorf http://orcid.org/0000-0001-8149-5890
Thomas Hickler http://orcid.org/0000-0002-4668-7552
Marco Schmidt http://orcid.org/0000-0001-6087-6117

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, et al. 2016. "Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." *ArXiv Preprint ArXiv*:1603.04467.

Brunken, U., M. Schmidt, S. Dressler, T. Janßen, A. Thiombiano, and G. Zizka. 2008. "www.westafricanplants.senckenberg.de - an Image-Based Identification Tool for West African Plants." *Taxon* 57 (3): 1027–1028.

Carranza-Rojas, J., H. Goeau, P. Bonnet, E. Mata-Montero, and A. Joly. 2017. "Going Deeper in the Automated Identification of Herbarium Specimens." *BMC Evolutionary Biology* 17 (1): 181.

Clevert, D.-A., T. Unterthiner, and S. Hochreiter. 2015. "Fast and Accurate Deep Network Learning by Exponential Linear Units (Elus)." ICLR 2016 conference paper. *ArXiv Preprint ArXiv*:1511.07289.

Corney, D. P. A., J. Y. Clark, H. L. Tang, and P. Wilkin. 2012. "Automatic Extraction of Leaf Characters from Herbarium Specimens." *Taxon* 61 (1): 231–244.

Dozat, T. 2016. Incorporating Nesterov Momentum into Adam. ICLR workshop paper, 2016. https://openreview.net/pdf?id=OM0jvwB8jIp57ZJjtNEZ

Dressler, S., M. Schmidt, and G. Zizka. 2014. "Introducing African Plants - a Photo Guide - an Interactive Identification Tool for Continental Africa." *Taxon* 63 (5): 1159–1161.

GBIF Secretariat. 2017. "GBIF Backbone Taxonomy." Accessed via https://www.gbif.org/species/6 in 2017

Givnish, T. J. 1987. "Comparative Studies of Leaf Form: Assessing the Relative Roles of Selective Pressures and Phylogenetic Constraints." *New Phytologist* 106 (Suppl.): 131–160.

Goëau, H., P. Bonnet, and A. Joly. 2017. "Plant Identification Based on Noisy Web Data: The Amazing Performance of Deep Learning (LifeCLEF 2017)." CEUR Workshop Proceedings.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep Residual Learning for Image Recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.

Hoehndorf, R., M. Alshahrani, G. V. Gkoutos, G. Gosline, Q. Groom, T. Hamann, J. Kattge, et al. 2016. "The Flora Phenotype Ontology (FLOPO): Tool for Integrating Morphological Traits and Phenotypes of Vascular Plants." *Journal of Biomedical Semantics* 7 (1): 309.

Hubel, D. H., and T. N. Wiesel. 1968. "Receptive Fields and Functional Architecture of Monkey Striate Cortex." *The Journal of Physiology* 195 (1): 215–243.

Ioffe, S., and C. Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." *International Conference on Machine Learning*, 448–456.

Joly, A., P. Bonnet, H. Goëau, J. Barbe, S. Selmi, J. Champ, S. Dufour-Kowalski, et al. 2016. "A Look inside the Pl@ntNet Experience." *Multimedia Systems* 22 (6): 751–766.

Kumar, N., P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. B. Soares. 2012. "Leafsnap: A Computer Vision System for Automatic Plant Species Identification." *Computer Vision–ECCV 2012*, 502–516. Berlin, Heidelberg: Springer.

Le Bras, G., M. Pignal, M. L. Jeanson, S. Muller, C. Aupic, B. Carré, G. Flament, et al. 2017. "The French Muséum National d'Histoire Naturelle Vascular Plant Herbarium Collection Dataset." *Scientific Data* 4: 170016.

LeCun, Y., and Y. Bengio. 1995. "Convolutional Networks for Images, Speech, and Time Series." *The Handbook of Brain Theory and Neural Networks* 3361 (10): 276–279.

LeCun, Y., Y. Bengio, and G. Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–444.

Matsugu, M., K. Mori, Y. Mitari, and Y. Kaneda. 2003. "Subject Independent Facial Expression Recognition with Robust Face Detection Using a Convolutional Neural Network." *Neural Networks* 16 (5–6): 555–559.

Olshausen, B. A., and D. J. Field. 1996. "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images." *Nature* 381: 607.

Pound, M. P., A. J. Burgess, M. H. Wilson, J. A. Atkinson, M. Griffiths, A. S. Jackson, A. Bulat, et al. 2017. "Deep Machine Learning Provides State-of-the-Art Performance in Image-Based Plant Phenotyping." *GigaScience* 6 (10): 1–10.

Schmidhuber, J. 2015. "Deep Learning in Neural Networks: An Overview." *Neural Networks* 61: 85–117.

Schuettpelz, E., P. B. Frandsen, R. B. Dikow, A. Brown, S. Orli, M. Peters, A. Metallo, V. A. Funk, and L. J. Dorr. 2017. "Applications of Deep Convolutional Neural Networks to Digitized Natural History Collections." *Biodiversity Data Journal* 5: e21139.

Smith, G. F., J. P. Roux, P. Raven, and E. Figueiredo. 2011. "African Herbaria Support Transformation on the Continent 1." *Annals of the Missouri Botanical Garden* 98 (2): 272–276.

Thiers, B. 2017. "The World's Herbaria 2016: A Summary Report Based on Data from Index Herbariorum." http://sweetgum.nybg.org/science/docs/The World's Herbaria 2016 18 Jan 2017.pdf

Ubbens, J. R., and I. Stavness. 2017. "Deep Plant Phenomics: A Deep Learning Platform for Complex Plant Phenotyping Tasks." *Frontiers in Plant Science* 8: 1476.

Unger, J., D. Merhof, and S. Renner. 2016. "Computer Vision Applied to Herbarium Specimens of German Trees: Testing the Future Utility of the Millions of Herbarium Specimen Images for Automated Identification." *BMC Evolutionary Biology* 16 (1): 25.

Van Horn, G., O. Mac Aodha, Y. Song, A. Shepard, H. Adam, P. Perona, and S. Belongie. 2017. "The iNaturalist Challenge 2017 Dataset." Retrieved from http://arxiv.org/abs/1707.06642

Yosinski, J., J. Clune, Y. Bengio, and H. Lipson. 2014. "How Transferable Are Features in Deep Neural Networks?" *Advances in Neural Information Processing Systems* 3320–3328.

Zeiler, M. D., and R. Fergus. 2014. "Visualizing and Understanding Convolutional Networks." In *European Conference on Computer Vision*, edited by D. Fleet, T. Pajtla, B. Schiele and T. Tuytelaars, 818–833. Cham: Springer.

4.2 Publication :

Detection and annotation of plant organs from digitised herbarium scans using deep learning

In this publication, a Faster R-CNN based approach for detecting plant organs, on herbarium scans is presented. These plant organs are leaves, flowers, stems, seeds, and roots. While prior research focused on detecting leaves and fruits on plants, especially in their natural environment, our approach stands out as the first known research effort to detect a large number of plant organs, representing both vegetative and reproductive plant organs on herbarium scans. The dataset for this study comprised of herbarium scans sourced from Muséum national d'Histoire naturelle (MNHN) and Herbarium Senckenbergianum collections. Our approach demonstrated the highest detection accuracy on leaves, with good precision for roots and stems. Detecting fruits and flowers was challenging due to their infrequent occurrence and the observed imbalance of reproductive organs, across various herbarium specimens and species.

Contribution Role: Lead Author

Published in:

Biodiversity Data Journal, 2020

Biodiversity Data Journal 8: e57090
doi: 10.3897/BDJ.8.e57090

open access

Research Article

Detection and annotation of plant organs from digitised herbarium scans using deep learning

Sohaib Younis[‡,§], Marco Schmidt[‡,|], Claus Weiland[‡], Stefan Dressler[¶], Bernhard Seeger[§], Thomas Hickler[‡]

‡ Senckenberg Biodiversity and Climate Research Centre (SBiK-F), Frankfurt am Main, Germany
§ Department of Mathematics and Computer Science, Philipps-University Marburg, Marburg, Germany
| Palmengarten der Stadt Frankfurt, Frankfurt am Main, Germany
¶ Senckenberg Research Institute and Natural History Museum, Frankfurt am Main, Germany

Corresponding author: Sohaib Younis (sohaibyounis89@gmail.com)

Academic editor: Ross Mounce

Received: 30 Jul 2020 | Accepted: 16 Nov 2020 | Published: 10 Dec 2020

Citation: Younis S, Schmidt M, Weiland C, Dressler S, Seeger B, Hickler T (2020) Detection and annotation of plant organs from digitised herbarium scans using deep learning. Biodiversity Data Journal 8: e57090. https://doi.org/10.3897/BDJ.8.e57090

Abstract

As herbarium specimens are increasingly becoming digitised and accessible in online repositories, advanced computer vision techniques are being used to extract information from them. The presence of certain plant organs on herbarium sheets is useful information in various scientific contexts and automatic recognition of these organs will help mobilise such information. In our study, we use deep learning to detect plant organs on digitised herbarium specimens with Faster R-CNN. For our experiment, we manually annotated hundreds of herbarium scans with thousands of bounding boxes for six types of plant organs and used them for training and evaluating the plant organ detection model. The model worked particularly well on leaves and stems, while flowers were also present in large numbers in the sheets, but were not equally well recognised.

Keywords

herbarium specimens, plant organ detection, deep learning, convolutional neural networks, object detection and localisation, image annotation, digitisation

Introduction

Herbarium collections have been the basis of systematic botany for centuries. More than 3000 herbaria are active on a global level, comprising ca. 400 million specimens, a number that has doubled since the early 1970s and is growing steadily (Thiers 2020). Accessibility of these collections has been improved by international science infrastructure aggregating specimen data and increasingly also digital images of the specimens. Plant specimens, being usually flat and of a standard format approximating A3 size, are easier to digitise than most other biological collection objects. The Global Plants Initiative (Smith and Figueiredo 2014) has been very successful in digitising type specimens around the world. Single collections, such as the National Museum of Natural History in Paris, have digitised their collections completely (Le Bras et al. 2017) and large scale national or regional digitisation initiatives are already taking place or are planned for the near future (Borsch et al. 2020). Presently, there are more than 27 million plant specimen records with images available via the GBIF platform (www.gbif.org), the vast majority of these images being herbarium scans.

This rising number of digitised herbarium sheets provides an opportunity to employ computer-based image processing techniques, such as deep learning, to automatically identify species and higher taxa (Carranza-Rojas et al. 2017, Younis et al. 2018, Carranza-Rojas et al. 2018) or to extract other useful information from the images, such as the presence of pathogens (as done for live plant photos by Mohanty et al. 2016). Deep learning is a subset of machine learning methods for learning data representation. Deep learning techniques require huge amounts of training data to learn the features and representation of those data for the specified task by fine tuning parameters of hundreds or thousands of neural networks, arranged in multiple layers. Learning the value of these parameters can take vast computer and time resources, especially on huge datasets.

The most common type of deep learning network architecture being used for extracting image features is the Convolutional Neural Network (CNN) (LeCun and Bengio 1995). A convolutional neural network extracts the features of an image by passing through a series of convolutional, non-linear, pooling (image downsampling) layers and passes them to a fully connected layer to obtain the desired output. Each convolutional layer extracts the visual features of the image by applying convolution operations to the image with kernels, using a local receptive field, to produce feature maps and passing it as input to the next layer. The initial layers in the network compute primitive features on the image, such as corners and edges, the deeper layers use these features to compute more complex features consisting of curves and basic shapes and the deepest layers combine these shapes and curves to create recognisable shapes of the concepts in the image (Yosinski et al. 2014, Zeiler and Fergus 2014).

In this paper, we use deep learning for detecting plant organs on herbarium scans. The plant organs are detected using an object detection network, which works by localising each object with a bounding box on the image and classifying it. There are many types of networks, based on CNN, used for this application. In this study, a network called Faster

R-CNN (Ren et al. 2015) was used, which is part of the R-CNN family for object detection. Region-based Convolutional Networks (R-CNN) identify objects and their locations in an image. Faster R-CNN networks have shown state-of-the-art performances in various object detection applications and competitions (Zhao et al. 2019). Therefore, many researchers have explored the use of CNN and particularly Faster R-CNN for detecting various plant organs, such as flowers, fruits and seedlings (Sa et al. 2016, Stein et al. 2016, Häni et al. 2020, Mai et al. 2018, Sun et al. 2018, Bargoti and Underwood 2017, Jiang et al. 2019, Ott et al. 2020, Weaver et al. 2020). To our knowledge, this is the first time object detection has been used to detect both vegetative and reproductive plant organs on herbarium scans. Identifying and localising plant organs on herbarium sheets is a first necessary step for some interesting applications. The presence and state of organs, such as leaves, flowers and fruits, can be used in phenological studies over long time periods and may give us more insight into climate change effects since the time of the Industrial Revolution (Willis et al. 2017, Lang et al. 2019).

Methods

Network architecture

A typical object detection network consists of object localisation and classification integrated into one convolutional network. There are two main types of meta-architectures available for this application: single stage detectors like Single Shot Multibox Detectors (SSD) (Liu et al. 2016) and 'You only look once' (YOLO) (Redmon et al. 2016) and two-stage, region-based CNN detectors, such as Faster R-CNN. Single stage detectors use a single feed-forward network to predict object class probabilities along with bounding box coordinates on the image. Faster R-CNN is composed of three modules: 1) a deep CNN image feature extraction network, 2) a Region Proposal Network (RPN), used for detection of a predefined number of Regions of Interests (RoIs) where the object(s) of interest could reside within the image, followed by 3) Fast R-CNN (Girshick 2015), computes a classification score along with class-specific bounding box regression for each of these regions. The main reason for choosing Faster R-CNN for organ detection is because it is generally more accurate, particularly for large and small objects, than single stage detectors like SSD when speed and memory consumption are not as important as overall accuracy (Huang et al. 2017).

The CNN feature extraction network used in this paper is based on the ResNet-50 architecture (He et al. 2016), without the final fully-connected layer. The Region Proposal Network (RPN) creates thousands of prior or anchor boxes to estimate the location of objects in the image. The anchor boxes are predefined bounding boxes of certain height and width tiled across the image, determined by their scale and aspect ratios, in order to capture different sizes of objects of specific classes. The RPN generates these proposals by adjusting these anchors with coordinate offsets of the object bounding boxes and predicts the possibility of each anchor being a foreground object or a background. These proposals are sorted according to their score and top N proposals are selected by Non-Maximum Suppression (NMS), which are then passed to Fast R-CNN stage. NMS reduces

the high number of proposals for the next stage by short-listing the proposals with the highest score having minimum overlap with each other by removing the proposals with overlap above a predefined threshold for each category. In the next stage, the proposals with feature maps of different shapes are pooled with a ROI pooling layer, which performs max-pooling on the inputs of non-uniform sizes to obtain a fixed number of uniform size feature maps. These feature maps are propagated through fully-connected layers, which end in two siblings fully-connected layers for object classification and bounding box regression, respectively. An illustration of Faster R-CNN is shown in Fig. 1.

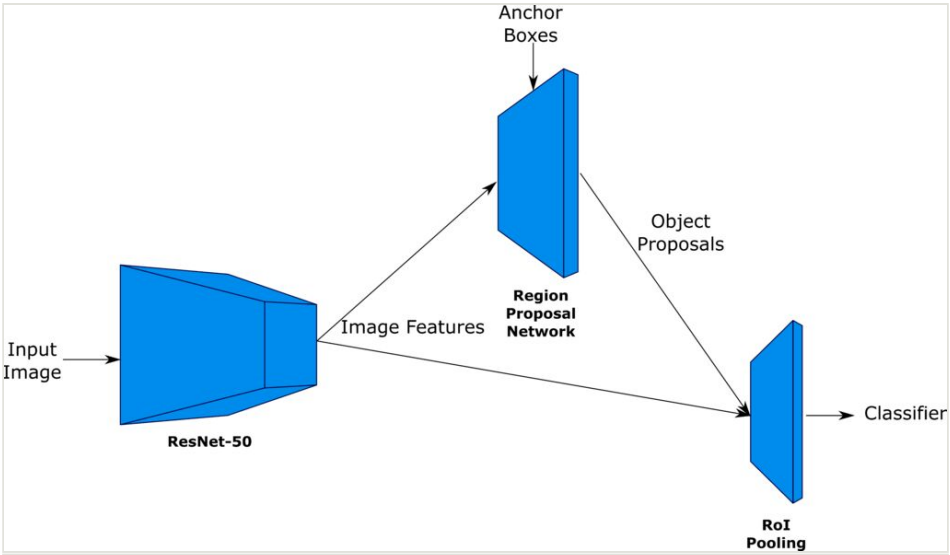


Figure 1. doi

An illustration of the Faster R-CNN architecture, with ResNet for image feature extraction, RPN for generating object proposals and RoI Pooling for creating fixed-size feature maps for each proposal.

Image Annotation

The herbarium scans annotated for training the object detection network were selected from the MNHN (Muséum national d'Histoire naturelle) vascular plant herbarium collection dataset in Paris (Le Bras et al. 2017), from open access images contributed to the GBIF portal (MNHN and Chagnoux 2020). A total of 653 images were downloaded and rescaled from their original average size of ca. 5100 by 3500 pixels to 1200 by 800 pixels, in order to preserve the aspect ratio of the scans and to speed up the learning by reducing the number of pixels. The images were selected manually from a large collection of scans, having minimum visual overlap between organs, while covering a broad range of taxa and morphology (Fig. 2, Suppl. material 2). All these images were annotated for six different types of organs (Suppl. material 1) using LabelImg (Tzutalin 2015), a Python graphical toolkit for image annotation using bounding boxes. The average rate for manual image annotation was 8 to 15 herbarium sheets per hour, depending on the difficulty and number

of bounding boxes to be annotated. The total number of annotated bounding boxes for all 653 images was 19654, with an average of 30.1 bounding boxes per image. From these 653 annotated images, 155 of them were either annotated or verified by an expert, making a validated subset hence used for testing and the 498 were used for training, as shown in Fig. 3 and Fig. 4 and in more detail in Table 1.

Table 1.

The number of annotated bounding boxes for each plant organ in training and test subset.

Category	Training subset (498 images)	Test subset (155 images)	Complete dataset (653 images)
Leaf	7886	2051	9937
Flower	3179	763	3942
Fruit	1047	296	1343
Seed	4	6	10
Stem	3323	961	4284
Root	78	60	138
Total	15517	4137	19654

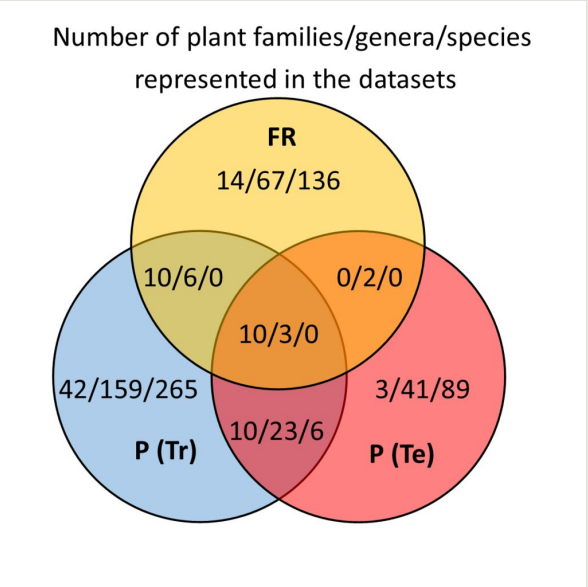


Figure 2. doi

Number of taxa of different rank for the three datasets with overlaps at family, genus and species level. P(Tr), P(Te): MNHN Paris Herbarium training and test datasets, FR: Herbarium Senckenbergianum dataset.

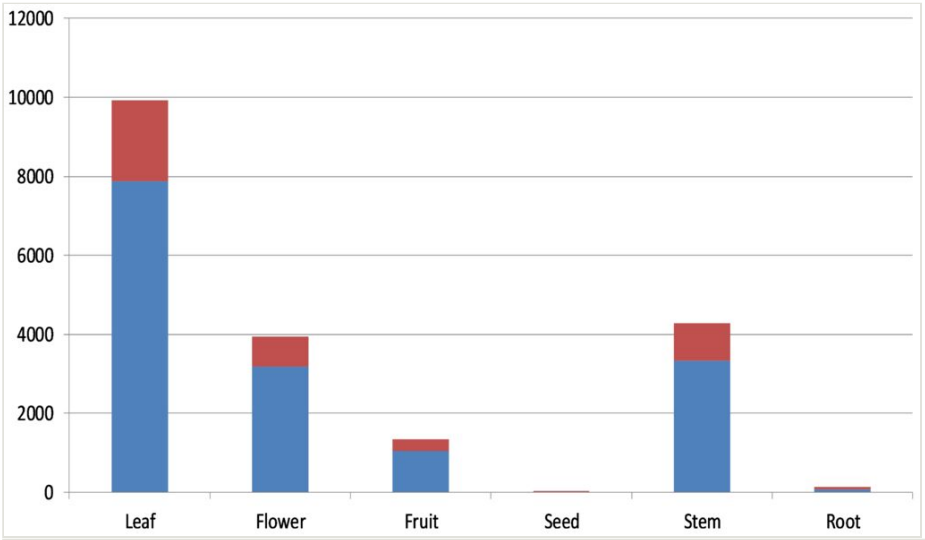


Figure 3. doi

A column chart showing the number of annotated bounding boxes for each organ. Red: Test subset, Blue: Training subset.

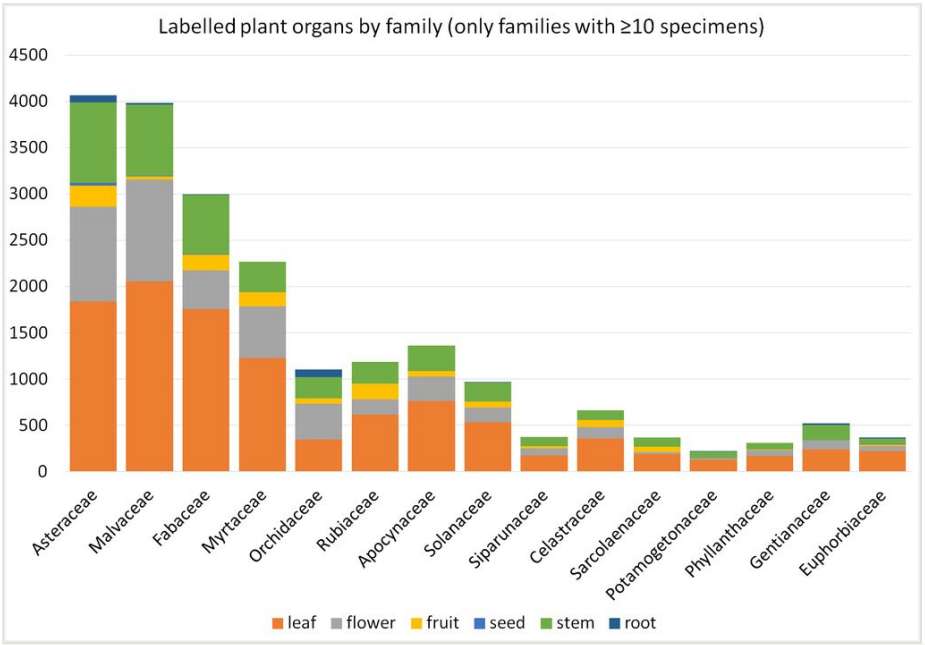


Figure 4. doi

Families of labelled specimens (ordered by number of specimens) with number of labelled plant organs. The share of the plant organs differs between families, which may be due to factors depending on the plant itself and collecting habits (season, selection of identifiable specimens).

Preparing our data was not always straight-forward. The manual localisation and labelling of plant organs from specimens encountered the following difficulties: buds, flowers and fruits are different stages emerging in the life cycle of plant reproductive organs and, in some cases, it was therefore difficult to find a clear distinction between these structures. In some taxa, different plant organs were impossible to separate as these were small and crowded, for example, in dense inflorescences with bracts and flowers or stems densely covered by leaves. In a few cases, it was also hard to differentiate from the digital image between roots and stolons or other stem structures. In all of these cases, we placed our labelled boxes in a way to best characterise the respective plant organ. Sometimes, this involved including parts of other organs and, at other times, if sufficient clearly assignable material were available, difficult parts were left out.

Implementation

The object recognition task was performed using Faster R-CNN, as described in the network architecture, with the Feature Pyramid Network (Lin et al. 2017) backbone. The Feature Pyramid Network increases the accuracy of the object detection task by generating multi-scale feature maps from a single scale feature map of ResNet output, by making top-down pathways in addition to the usual bottom-up pathways used by a regular convolutional network for feature extraction, where each layer of the network represents one pyramid level. The bottom–up pathway increases the semantic value of the image features, from corners and edges in the initial layers to detecting high level structures and shapes of objects in the image in the final layers, while reducing its resolution at each layer. The top-down pathway then reconstructs higher resolution layers from the most semantically rich layer, with predictions made independently at all levels as shown in Fig. 5. This approach provides Faster R-CNN with feature maps at different resolutions for detecting objects of multiple scales.

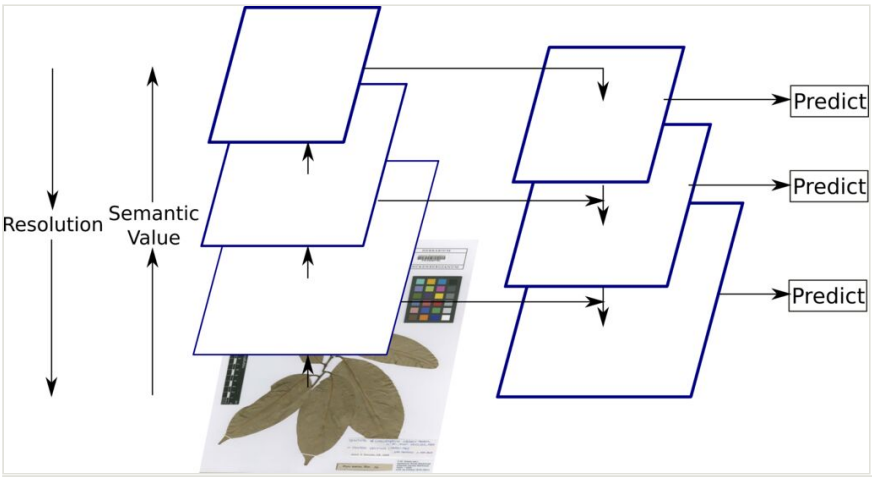


Figure 5. doi

An illustration of Feature Pyramid Network, where feature maps are indicated by blue outlines and thicker outlines denote semantically stronger features (Lin et al. 2017).

In order to reduce the training time and, more importantly, because of the small size of the training dataset, transfer learning (Yosinski et al. 2014) was implemented to initialise the model weights pre-trained on the ImageNet dataset (Deng et al. 2009). Since the initial layers of a CNN usually learn very generic features that can also be used in new contexts, pre-trained weights can initialise the weights for these layers. For the deeper layers, transfer learning is used to initialise the parameter weights pre-trained on the ImageNet dataset and then fine-tuned during training, using the annotated herbarium scan dataset until convergence.

The model was implemented with the Detectron2 (Wu et al. 2019) library in PyTorch framework and trained using Stochastic Gradient Descent optimiser with a learning rate of 0.0025 and momentum of 0.9. The anchor generator in the Region Proposal Network (see section above on network architecture) had six anchor scales [32, 64, 128, 256, 512, 1024] (square root of area in absolute pixels) each with three aspect ratios of [1:2, 1:1, 2:1]. The thresholds for non-maximum suppression (NMS) were 0.6 for training and 0.25 for testing, respectively.

Due to the large image size and additional parameters of Faster R-CNN, a minibatch size of four images per GPU (TITAN Xp) was selected for training the model. The model was trained twice, once with a training subset of 498 images on a single GPU for 9000 iterations and performance evaluated on the test subset of 155 images, also on a single GPU and then trained again on all 653 annotated images on three GPUs for 18000 iterations for predicting plant organs on another un-annotated independent dataset to evaluate our method. This dataset consists of 708 full scale herbarium scans, with an average size of ca. 9600 by 6500 pixels, from the Herbarium Senckenbergianum (FR) (Otte et al. 2011) with a different set of species (Fig. 2) and geographical origins, which is also available at GBIF (Senckenberg 2020). The Python code and the trained model have been made available at GitHub (Younis 2020).

Results

The predictions of the organ detection model provides a list of bounding boxes for each organ, along with the confidence levels and their class labels. The performance of the model was evaluated using the COCO evaluation metric (Lin et al. 2014), which determines whether the predicted organs and their locations are correct. The minimum threshold chosen for any prediction to be acceptable is having a confidence score (probability) of 0.5. The COCO method calculates average precision (with values from 0 to 100), which is a metric that encapsulates both precision and recall of the detection, for the entire predictions and each class of organs at different levels of Intersection over Union (IoU). IoU is an evaluation metric that quantifies the overlap of the predicted bounding boxes with the ground-truth bounding boxes. The IoU score ranges from 0 to 1, the higher the overlap, the higher the IoU score. The evaluation method considers all predictions as positive that have IoU of at least 0.5 and the average precision at this level of IoU is called AP50. Similarly, the average precision with a minimum IoU of 0.75 is called AP75, whereas AP is the average over 10 IoU levels from 0.5 to 0.95 with a step size of 0.05. The

precision metrics evaluated on the predicted organs on the test subset are shown in Table 2. The COCO method also calculates the AP for each category, as shown in Table 3, along with the total bounding boxes for each category in the test subset.

Table 2.

The precision of the predictions on the MNHN Paris Herbarium test subset with COCO evaluation method.

AP50	AP75	AP
22.8	6.8	9.7

Table 3.

Average Precision of each type of organ along with the total bounding boxes for each category in the test subset.

Category	Bounding Boxes	AP
Leaf	2051	26.5
Flower	763	4.7
Fruit	296	7.8
Seed	6	0.0
Stem	961	9.9
Root	60	9.4

From the predicted annotations of the model for plant organs on 708 full scale herbarium scans from the Herbarium Senckenbergianum dataset, trained on the 653 annotated MNHN Paris Herbarium dataset, 203 were manually verified and corrected to evaluate the predictions. The organ detection model was successfully able to detect almost all plant organs in the majority of scans, as shown by the images in Fig. 6. The dataset of these 203 herbarium scans, along with the result of detections and the annotations, is available at PANGAEA Younis et al. 2020.

The performance of the model on the verified annotated Herbarium Senckenbergianum dataset is shown in Table 4 and Table 5. The average precision on these 203 scans is generally higher than the MNHN Paris Herbarium test subset, there being two main reason for this: 1) The organ detection model for full scale detection was trained on all 653 images of the MNHN Paris Herbarium annotated dataset before detection on the Herbarium Senckenbergianum dataset, 2) The annotation of these 203 images from the Herbarium Senckenbergianum dataset were done, based on the predictions of organs on scans as shown in Fig. 6.

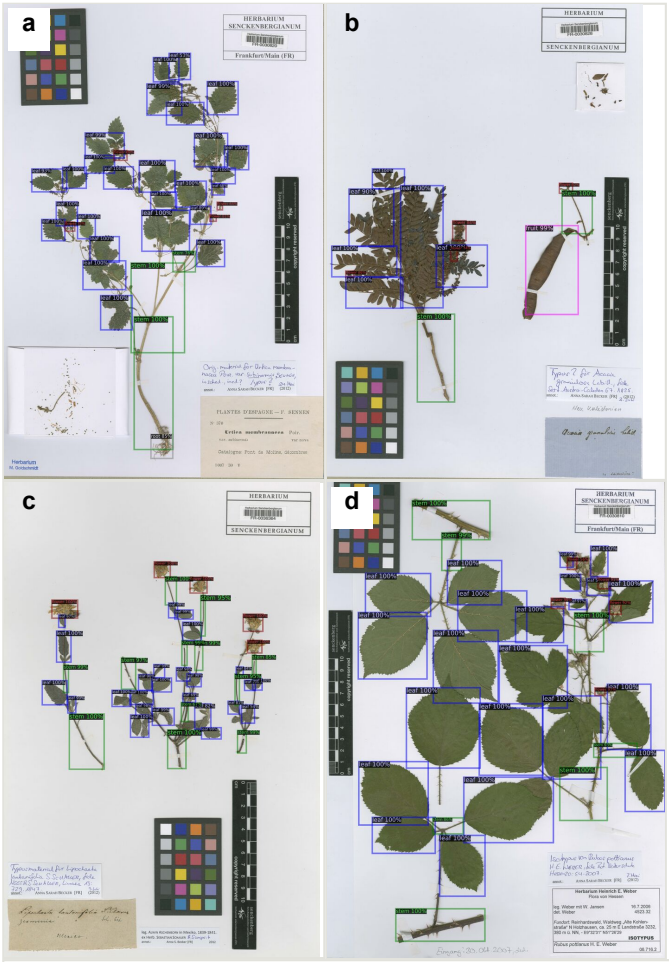


Figure 6.

Sample results of organ detection performed on unseen full scale Herbarium Senckenbergianum scans. Colour scheme for bounding boxes is; Leaf:Blue, Flower:Maroon, Fruit:Magenta, Seed:Yellow, Stem:Green, Root:Grey.

Discussion

This paper presents a method to detect multiple types of plant organs on herbarium scans. For this research, we annotated hundreds of images with thousands of bounding boxes by hand for each possible plant organ. A subset of these annotated scans was then used for training of deep learning for organ detection. After training, the model was used to predict the type and location of plant organs on the test subset. The automated detection of plant organs in our study was most successful for leaves and stems (Table 3 and Table 5). Best AP values for leaves are likely due to the largest set of annotated bounding boxes. Good values for stems and roots may be explained by the relative uniformity of these organs throughout the plant kingdom, as compared to the morphologically more diverse flowers

and fruits in between these. Seeds are rarely visible on herbarium sheets and require more training material.

Table 4.

Result of model evaluation on the Herbarium Senckenbergianum annotated dataset.

AP50	AP75	AP
32.1	16.1	16.8

Table 5.

Average Precision of each type of organ along with the total bounding boxes for each category in the Herbarium Senckenbergianum annotated dataset.

Category	Bounding Boxes	AP
Leaf	3362	37.9
Flower	1921	18.3
Fruit	183	7.9
Seed	47	0.0
Stem	1063	25.1
Root	117	11.8

The model was trained again on all the annotated scans earlier and tested on a different un-annotated dataset. The model performed well, based on visual inspection. In order to evaluate the performance of the model with an average precision metric, around 200 of these scans were annotated by hand, based on the predicted bounding boxes. The predicted bounding boxes dramatically reduced the time to annotate these scans, since the predictions for leaves and stems were fairly accurate. After being annotated, these scans were compared with the predictions to evaluate the precision of the organ detection model on this dataset.

We consider our study as a 'real-life' pioneer study with inherent biases. The training and test datasets from MNHN Paris Herbarium are from the same collection, while the Herbarium Senckenbergianum specimens are from an independent collection with different geographical and taxonomic focus, but still with a number of higher taxa in common with MNHN Paris Herbarium. The different datasets overlap mainly on the family level, partly on genus level and only slightly between the MNHN Paris Herbarium training and test datasets at species level (Fig. 2, Suppl. material 2). Therefore, we can exclude organ recognition being based upon species-specific features. As in nature itself and the collections represented here, families are not represented equally. Likewise, the number of labelled organs, represented in our dataset, is far from balanced and biased both by the natural distribution of these organs in the sampled taxa and by the selection of material by the collectors. Roots, for example, are mainly represented in Asteraceae and Orchidaceae,

families with many small and herbaceous species (Fig. 4, Suppl. material 3). In order to better understand the difference in average precision of organ detection across different taxa, further studies are necessary. A promising strategy would be to employ data augmentation to create artificially-balanced distributions of organs and taxa (Shorten and Khoshgoftaar 2019). The current study focuses on the analysis and the provision of annotated datasets of actual herbarium specimens, involving the aforementioned constraints rooted in the morphology of the specimens concerned and not simulated data. It would also be interesting to compare a general organ recognition with taxon-specific approaches. Especially for fruits and flowers, we have very different shapes between taxa and also the possible distinction between different developmental stages depends a lot on the taxon.

Most computer vision approaches on plants focus on live plants, often in the context of agriculture or plant breeding and, therefore, include only a limited set of taxa. The present approach not only targets a much larger group of organisms and morphological diversity, comparable to applications in citizen science (Wäldchen and Mäder 2019), but can also be applied on a wider time-scale by including collection objects from hundreds of years of botanical research. Some significant recent similar approaches to detect plant organs on herbarium scans are GinJinn (Ott et al. 2020) and LeafMachine (Weaver et al. 2020). GinJinn uses an object-detection pipeline for automated feature extraction from herbarium specimens. This pipeline can be used to detect any type of plant organ, which the authors of this research demonstrated by detecting leaves on a sample dataset. LeafMachine is another approach which tries to automate extraction of leaf traits, such as class, size and number, from digitised herbarium specimens with machine learning.

Conclusions

Our present work focuses on the detection of plant organs from specimen images. The presence of flowers and fruits on specimens is a new source of data for phenological studies (Willis et al. 2017), interesting in the context of climate change. Presence of roots would identify plant specimens potentially containing root symbionts, such as mycorrhizal fungi or N-fixing bacteria, for further study by microbiological or genetic methods (Heberling and Burke 2019). Up to now, this requires visual examination of the specimens by humans; however, an automated approach using computer vision would considerably reduce the effort. Furthermore, the detection and localisation of specific plant organs on a herbarium sheet would also enable or improve further computer-vision applications, including quantitative approaches, based on counting these organs, improved recognition of qualitative organ-specific traits, such as leaf shape, as well as quantitative measures, such as leaf area or fruit size.

Localisation of plant organs will improve automated recognition and measurements of organ-specific traits, by preselecting appropriate training material for these approaches. The general approach of measuring traits from images instead of the specimen itself has been shown to be precise, except for very small objects (Borges et al. 2020). Of course,

measurements that involve further processing of plant parts, as often done in traditional morphological studies on herbarium specimens, are not possible from images.

Automated pathogen detection on collection material will also profit from the segmentation of plant organs from Herbarium sheet images, as many pathogens or symptoms of a plant disease only occur on specific organs. Studies on gall midges (Veenstra 2012) have found herbarium specimens to be interesting study objects and would potentially profit from computer vision.

Manual annotation of herbarium specimens with bounding boxes, as done for the training and test datasets in this study, is a rather time-consuming process. Verification and correction of automatically-annotated specimens is considerably faster, especially if the error rate is low. By iteratively incorporating expert-verified computer-generated data into new training datasets, the results can be further improved with reasonable efforts using Continual Learning (Parisi et al. 2019).

Acknowledgements

TH, SY, MS and SD received funding from the DFG Project "Mobilization of trait data from digital image files by deep learning approaches" (grant 316452578). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the TITAN Xp GPU to CW used for this research. Digitisation of the Senckenberg specimens used in this study has taken place in the frame of the Global Plants Initiative.

Author contributions

Sohaib Younis is computer scientist at Senckenberg Biodiversity and Climate Research Center with focus on deep learning and image processing. Contributions: convolutional network modelling, image preprocessing, annotation of herbarium scans, organ detection, description of results and preparation of the manuscript.

Marco Schmidt is botanist at Senckenberg Biodiversity and Climate Research Center (SBIK-F) with a focus on African savannahs and biodiversity informatics (e.g. online databases like African Plants - a photo guide and West African vegetation) and is working at Palmengarten's scientific service, curating living collections and collection databases. Contributions: concept of study, annotation and verification of herbarium scans, preparation of the manuscript.

Claus Weiland is theoretical biologist at SBIK-F's Data & Modelling Centre with main interests in graph neural networks and bio-ontologies. Contributions: Design of the GPU platform, data analysis and preparation of the manuscript.

Stefan Dressler is curator of the phanerogam collection of the Herbarium Senckenbergianum Frankfurt/M., which includes its digitisation and curation of associated databases. Taxonomically, he is working on Marcgraviaceae, Theaceae, Pentaphylacaceae

and several Phyllanthaceous genera. Contribution: Herbarium Senckenbergianum collection, preparation of the manuscript.

Bernhard Seeger is professor of computer science systems at the Philipps University of Marburg. His research fields include high-performance database systems, parallel computation and real-time processing of high-throughput data with a focus on spatial biodiversity data. Contribution: Provision of support in machine learning and data processing.

Thomas Hickler is head of SBIK-F's Data & Modelling Centre and Professor for Biogeography at the Goethe University Frankfurt. He is particularly interested in interactions between climate and the terrestrial biosphere, including potential impacts of climate change on species, ecosystems and associated ecosystem services. Contribution: Preparation of the manuscript, comprehensive concept of study within biodiversity sciences.

Conflicts of interest

No potential conflict of interest was reported by the authors.

References

- Bargoti S, Underwood J (2017) Deep fruit detection in orchards. 2017 IEEE International Conference on Robotics and Automation (ICRA) 3626-3633. https://doi.org/10.1109/icra.2017.7989417
- Borges L, Reis VC, Izbicki R (2020) Schrödinger's phenotypes: images of herbarium specimens are both good and (not so) bad sources of morphological data. BioRxiv https://doi.org/10.1101/2020.03.31.018812
- Borsch T, Stevens A, Häffner E, Güntsch A, Berendsohn W, Appelhans M, Barilaro C, Beszteri B, Blattner F, Bossdorf O, Dalitz H, Dressler S, Duque-Thüs R, Esser H, Franzke A, Goetze D, Grein M, Grünert U, Hellwig F, Hentschel J, Hörandl E, Janßen T, Jürgens N, Kadereit G, Karisch T, Koch M, Müller F, Müller J, Ober D, Porembski S, Poschlod P, Printzen C, Röser M, Sack P, Schlüter P, Schmidt M, Schnittler M, Scholler M, Schultz M, Seeber E, Simmel J, Stiller M, Thiv M, Thüs H, Tkach N, Triebel D, Warnke U, Weibulat T, Wesche K, Yurkov A, Zizka G (2020) A complete digitization of German herbaria is possible, sensible and should be started now. Research Ideas and Outcomes 6 https://doi.org/10.3897/rio.6.e50675
- Carranza-Rojas J, Goeau H, Bonnet P, Mata-Montero E, Joly A (2017) Going deeper in the automated identification of Herbarium specimens. BMC Evolutionary Biology 17 (1): 1-14. https://doi.org/10.1186/s12862-017-1014-z
- Carranza-Rojas J, Joly A, Goëau H, Mata-Montero E, Bonnet P (2018) Automated identification of herbarium specimens at different taxonomic levels. Multimedia Tools and Applications for Environmental & Biodiversity Informatics 151-167. https://doi.org/10.1007/978-3-319-76445-0_9

- Deng J, Dong W, Socher R, Li L, Kai Li, Li Fei-Fei (2009) ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition 248-255. https://doi.org/10.1109/cvpr.2009.5206848
- Girshick R (2015) Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV) 1440-1448. https://doi.org/10.1109/iccv.2015.169
- Häni N, Roy P, Isler V (2020) A comparative study of fruit detection and counting methods for yield mapping in apple orchards. Journal of Field Robotics 37 (2): 263-282. https://doi.org/10.1002/rob.21902
- Heberling JM, Burke D (2019) Utilizing herbarium specimens to quantify historical mycorrhizal communities. Applications in Plant Sciences 7 (4). https://doi.org/10.1002/aps3.1223
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 770-778. https://doi.org/10.1109/cvpr.2016.90
- Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K (2017) Speed/Accuracy trade-offs for modern convolutional object detectors. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) https://doi.org/10.1109/cvpr.2017.351
- Jiang Y, Li C, Paterson A, Robertson J (2019) DeepSeedling: deep convolutional network and Kalman filter for plant seedling detection and counting in the field. Plant Methods 15 (1): 141. https://doi.org/10.1186/s13007-019-0528-3
- Lang PM, Willems F, Scheepens JF, Burbano H, Bossdorf O (2019) Using herbaria to study global environmental change. New Phytologist 221 (1): 110-122. https://doi.org/10.1111/nph.15401
- Le Bras G, Pignal M, Jeanson M, Muller S, Aupic C, Carré B, Flament G, Gaudeul M, Gonçalves C, Invernón V, Jabbour F, Lerat E, Lowry P, Offroy B, Pimparé EP, Poncy O, Rouhan G, Haevermans T (2017) The French Muséum national d'Histoire naturelle vascular plant herbarium collection dataset. Scientific Data 4 (1). https://doi.org/10.1038/sdata.2017.16
- LeCun Y, Bengio Y (1995) Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks 3361 (10): 276-279.
- Lin T, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: Common Objects in Context. Computer Vision – ECCV 2014 740-755. https://doi.org/10.1007/978-3-319-10602-1_48
- Lin T, Dollar P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2117-2125. https://doi.org/10.1109/cvpr.2017.106
- Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg A (2016) SSD: Single Shot MultiBox Detector. Computer Vision – ECCV 2016 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
- Mai X, Zhang H, Meng M-H (2018) Faster R-CNN with classifier fusion for small fruit detection. 2018 IEEE International Conference on Robotics and Automation (ICRA) 7166-7172. https://doi.org/10.1109/icra.2018.8461130
- MNHN, Chagnoux S (2020) The vascular plants collection (P) at the Herbarium of the Muséum national d'Histoire naturelle (MNHN - Paris). 69.186. GBIF. URL: https://doi.org/10.15468/nc6rxy

- Mohanty S, Hughes D, Salathé M (2016) Using deep learning for image-based plant disease detection. Frontiers in Plant Science 7: 1419. https://doi.org/10.3389/fpls.2016.01419
- Otte V, Wesche K, Dressler S, Zizka G, Hoppenrath M, Kienast F (2011) The New Herbarium Senckenbergianum: Old institutions under a new common roof. Taxon 60 (2): 617-618.
- Ott T, Palm C, Vogt R, Oberprieler C (2020) GinJinn: An object-detection pipeline for automated feature extraction from herbarium specimens. Applications in Plant Sciences 8 (6). https://doi.org/10.1002/aps3.11351
- Parisi G, Kemker R, Part J, Kanan C, Wermter S (2019) Continual lifelong learning with neural networks: A review. Neural Networks 113: 54-71. https://doi.org/10.1016/j.neunet.2019.01.012
- Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 779-788. https://doi.org/10.1109/cvpr.2016.91
- Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems 91-99.
- Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C (2016) DeepFruits: A fruit detection system using deep neural networks. Sensors 16 (8). https://doi.org/10.3390/s16081222
- Senckenberg (2020) Herbarium Senckenbergianum (FR). GBIF. URL: https://doi.org/10.15468/ucmdjy
- Shorten C, Khoshgoftaar TM (2019) A survey on Image Data Augmentation for Deep Learning. J Big Data 6 https://doi.org/10.1186/s40537-019-0197-0
- Smith G, Figueiredo E (2014) The Global Plants Initiative: Where it all started. Taxon 63 (3): 707-709. https://doi.org/10.12705/633.33
- Stein M, Bargoti S, Underwood J (2016) Image based mango fruit detection, localisation and yield estimation using multiple view geometry. Sensors 16 (11). https://doi.org/10.3390/s16111915
- Sun J, He X, Ge X, Wu X, Shen J, Song Y (2018) Detection of key organs in tomato based on deep migration learning in a complex background. Agriculture 8 (12): 196. https://doi.org/10.3390/agriculture8120196
- Thiers B (2020) The World's Herbaria 2019: A summary report based on data from Index Herbariorum. New York Botanical Garden (3). URL: http://sweetgum.nybg.org/science/docs/The_Worlds_Herbaria_2019.pdf
- Tzutalin (2015) LabelImg. Github. URL: https://github.com/tzutalin/labelImg
- Veenstra AA (2012) Herbarium collections—an invaluable resource for gall midge taxonomists. Muelleria 30 (1): 59-64.
- Wäldchen J, Mäder P (2019) Flora Incognita – wie künstliche Intelligenz die Pflanzenbestimmung revolutioniert. Biologie in unserer Zeit 49 (2): 99-101. https://doi.org/10.1002/biuz.201970211
- Weaver W, Ng J, Laport R (2020) LeafMachine: Using machine learning to automate leaf trait extraction from digitized herbarium specimens. Applications in Plant Sciences 8 (6). https://doi.org/10.1002/aps3.11367
- Willis C, Ellwood E, Primack R, Davis C, Pearson K, Gallinat A, Yost J, Nelson G, Mazer S, Rossington N, Sparks T, Soltis P (2017) Old plants, new tricks: Phenological

research using herbarium specimens. Trends in Ecology & Evolution 32 (7): 531-546. https://doi.org/10.1016/j.tree.2017.03.015

- Wu Y, Kirillov A, Massa F, Lo W, Girshick R (2019) Detectron2. Github. URL: https://github.com/facebookresearch/detectron2
- Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Lawrence ND (Eds) Advances in neural information processing systems. Neural Information Processing Systems Conference. 3320– 3328 pp. URL: https://papers.nips.cc/book/advances-in-neural-information-processing-systems-27-2014
- Younis S, Weiland C, Hoehndorf R, Dressler S, Hickler T, Seeger B, Schmidt M (2018) Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks. Botany Letters 165: 377-383. https://doi.org/10.1080/23818107.2018.1446357
- Younis S (2020) Plant Organ Detection. GitHub. URL: https://github.com/2younis/plant-organ-detection
- Younis S, Schmidt M, Dressler S (2020) Plant organ detections and annotations on digitized herbarium scans. PANGAEA. URL: https://doi.pangaea.de/10.1594/PANGAEA.920895
- Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Fleet D, Pajtla T, Schiele B, Tuytelaars T (Eds) European conference on computer vision. Springer, 818-833 pp. https://doi.org/10.1007/978-3-319-10590-1_53
- Zhao Z, Zheng P, Xu S, Wu X (2019) Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems 30 (11): 3212-3232. https://doi.org/10.1109/tnnls.2018.2876865

Supplementary materials

Suppl. material 1: Plant Organ Annotations doi

Authors: Sohaib Younis, Marco Schmidt, Claus Weiland
Data type: XML Files
Brief description: The zip archive provides annotations for both Herbarium Senckenbergianum and MNHN Paris Herbarium datasets.
Download file (778.04 kb)

Suppl. material 2: Specimen List doi

Authors: Sohaib Younis, Marco Schmidt
Data type: CSV File
Brief description: The file provides a list for all the specimens, showing their taxonomy, organ count and URLs.
Download file (119.27 kb)

Suppl. material 3: Family organ count doi

Authors: Sohaib Younis, Marco Schmidt
Data type: CSV File
Brief description: The file provides a list of the total annotated organs for each family.
Download file (2.53 kb)

4.3 Publication 3:

Data-Free Generative Replay for Class-Incremental Learning on Imbalanced Data

The following publication presents Data-Free Generative Replay (DFGR), a pioneering approach to continual learning, specifically designed to address imbalanced dataset scenarios for data-free class-incremental learning. DFGR comprises of a ResNet as an image classifier and BigGAN functioning as a generator, to synthesize the previous dataset. Noteworthy, our approach introduces a novel technique, generator replay adjustment, tackling data imbalance, besides applying focal loss to the classifier. This contributed to achieving high accuracy without the necessity of storing previous data.

Contribution Role: Lead Author

Submitted in:

IEEE Transactions on Knowledge and Data Engineering

Data-Free Generative Replay for Class-Incremental Learning on Imbalanced Data

Sohaib Younis*, Bernhard Seeger†
Department of Mathematics and Computer Science, University of Marburg
Marburg, Germany
Email: *sohaibyounis89@gmail.com, †seeger@mathematik.uni-marburg.de

***Abstract*—Continual learning is a challenging problem in machine learning, especially for image classification tasks with imbalanced datasets. It becomes even more challenging when it involves learning new classes incrementally. One method for incremental class learning, which helps addressing dataset imbalance, is rehearsal using previously stored data. In rehearsal-based methods, access to previous data is required for either training the classifier or the generator, but it may not be feasible due to storage, legal, or data access constraints. Although there are many rehearsal-free alternatives for class incremental learning, such as parameter or loss regularization, knowledge distillation, and dynamic architectures, they do not consistently achieve good results, especially on imbalanced data. This paper proposes a new approach called Data-Free Generative Replay (DFGR) for class incremental learning, where the generator is trained without access to real data. In addition, DFGR also addresses dataset imbalance in continual learning of an image classifier. Instead of using training data, DFGR trains a generator using mean and variance statistics of batch-norm and feature maps derived from a pretrained classification model. The results of our experiments demonstrate that DFGR performs significantly better than other data-free methods and reveal the performance impact of specific parameter settings. DFGR achieves up to 88.5% and 46.6% accuracy on MNIST and FashionMNIST datasets, respectively. Our code is available at https://github.com/2younis/DFGR**

I. Introduction

Recent advances in neural networks and deep learning have been reported to surpass human capabilities in a wide range of individual tasks or similar multiple tasks [37]. However, these architectures often remain static after training and cannot adapt their behavior over time or learn from new data. They also require a lot of labeled data to be read as input to the network multiple times, thus requiring significant training time. However, data arrives continuously as a stream of data items or batches in many real-world scenarios. Therefore, machine learning models should be able to learn from a data stream and adapt to a changing environment. This ability to acquire new knowledge is known as continual learning or lifelong learning [30].

Neural networks learn from new data by retraining the network on the entire old and new dataset or by transfer learning only on the new dataset. Transfer learning is a method in which a model trained on a similar domain is partially retrained on new data. One of the main problems is that when the network tries to learn from new data or tasks, it interferes with the previously learned knowledge, leading to catastrophic forgetting [26], which causes a significant decrease in performance or a complete loss of old knowledge.

To avoid catastrophic forgetting, the learning architecture must acquire new knowledge while preventing it from interfering with old knowledge. This phenomenon is called the stability-plasticity dilemma [27], where stability and plasticity refer to how strongly the systems retain learned knowledge and how much the systems can adapt to learn new knowledge, respectively. Too much stability will impede efficient learning from new data, whereas too much plasticity will result in forgotten knowledge previously learned.

There have been many different techniques for overcoming catastrophic forgetting, where the most common strategies belong to one of three main categories, namely architectural, regularization, and rehearsal [30] [16] [7]. In an architectural approach, the network tries to accommodate new knowledge by dynamically changing the number of layers or neurons in the network while also keeping some parts of the network static by fixing the weights of some neurons [38] [48] [9]. In the regularization approach, the plasticity of the network is controlled by imposing some restrictions on the update of the neurons' weights. This approach generally adds an extra adjustable regularization term to the loss function [21] [43]. Finally, in the rehearsal approach, old data is replayed to the model mixed with the current data for joint training. In this approach the network retains old knowledge while also learning new knowledge [39] [24] [15]. There are two ways of replaying old data: rehearsal and pseudo-rehearsal [36]. In simple rehearsal, the old data replayed to the model during training is stored in memory. This method becomes inefficient in many real-world settings because of its high storage space consumption [2], especially in the case of edge devices [33]. On the contrary, pseudo-rehearsal avoids storing old data by generating data on demand either randomly or from a generative model.

In addition to catastrophic forgetting, another challenge of deep learning in a real-world setting is that many image datasets are rarely balanced and follow a long-tail distribution, as shown in Fig. 1. The plot depicts the frequency distribution of species from the Pl@ntNet data set sorted by the number of occurrences. Obviously, the distribution of images in a dataset is not uniform but skewed towards some classes, resulting in a small number of classes containing the majority of images or samples and many classes containing only a

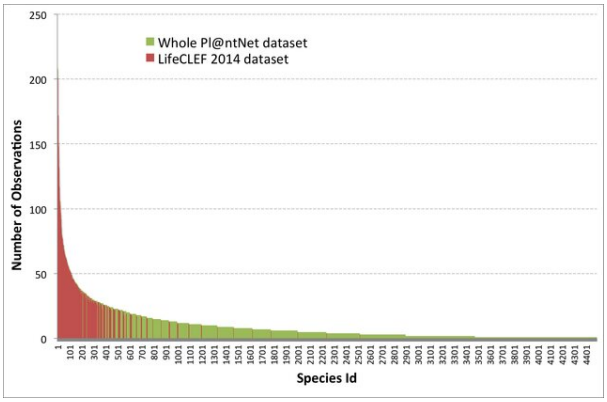


Fig. 1: Long tail distribution of the whole Pl@ntNet dataset (with PlantCLEF 2014 subset in red) [14].

small number of images. These class imbalances cause severe learning problems and are an active area of research in machine learning [42] [19]. The most common approaches to address class imbalance are undersampling of majority classes, oversampling of minority classes, data augmentation, or using synthetic data to balance the classes. In continual learning, an approach similar to oversampling is used mainly, especially in rehearsal or replay-based methods [6].

In this paper, we propose a new approach to image classification called Data-Free Generative Replay (DFGR) that addresses the problems of continual class-incremental learning and imbalanced datasets using pseudo-rehearsal from a generator. The generator in DFGR is not trained on real images, as it may not always be available due to privacy concerns or storage limitations, but uses the means and variances of batch normalization layers [13] of the trained classifier and the final layer feature maps of each trained class. Since DFGR does not store the data or part of it for replay but solely relies on the pseudo-rehearsal from the generator, it uses focal loss on the classifier to cope with imbalanced data and dynamically adjust the replay probabilities of classes for the generated images. In particular, we present two main contributions in this paper for DFGR: First, a feature map loss for estimating high level features of the images during the training of the generator and second, a generator replay adjustment for data augmentation of the generated images.

Rest of this paper is organized as follows. Section II investigates related works on data-free class incremental learning. Section III gives an overview to DFGR. Section IV and Section V explain the methodology for DFGR and its implementation, respectively. Finally the results of an experimental comparison for our approach are shown in Section VI.

II. Background and Related Work

This paper focuses on class incremental learning [34], one of the three continual learning scenarios presented in [41]. In this scenario, the training model arrives incrementally as a stream of labeled data belonging to different classes. The model accesses the data in stages or episodes, where each stage is called a task. Thus, the model trains on the data in a sequence of tasks, where each task consists of a non-overlapping subset of classes. Each task can consist of one or more classes, but each class can only appear in a single task. The class-incremental learning scenario can be challenging because the model has to learn to discriminate between all classes seen so far, either from current or previous tasks, mainly when they belong to different tasks. This becomes even more challenging in a data-free learning setup where storing any training data from the previous tasks is not allowed or possible, and the only available information is the model trained on previous tasks (with some meta-data) and the training data for the current task.

In the following, we focus on the most recent and effective techniques for class-incremental learning [3] [25]. We only cover the methods that support data-free continual learning, whereas replay-based methods are not in the scope of the paper and hence are omitted. We first provide an overview of the methods and then sketch the differences to our approach.

A. Regularization-based Methods

These types of methods avoid storing data previously seen, but rely on imposing additional constraints on the update process of various model parameters and hyper-parameters during training in order to mitigate catastrophic forgetting. Thus, these methods are inherently data-free. There are various options for regularization during training. Some regularize the model weights, [18] [47] [1], while others focus on remembering important feature representations of previous data [21] [8].

In image classification, one renowned method called Elastic Weight Consolidation (EWC) [18] adds a quadratic penalty to the loss function, which restricts the update of model parameters considered important to the previously learned classes. The importance of the parameters is approximated by the diagonal of the Fisher information matrix [31]. Another similar method is Synaptic Intelligence (SI), which calculates the importance of the learned parameters with the help of synapses [47]. Memory Aware Synapses (MAS) calculates the importance of weights in an unsupervised manner with Hebbian learning by observing the sensitivity of trained model's output function [1].

Other approaches to regularization aim to prevent activation drift [21] [8], which is the change in activations of the old network while learning new tasks. This approach is based on knowledge distillation from a model trained on the previous classes to the model being trained on the new data. A commonly known method with this approach is Learning without Forgetting (LwF) [21]. It takes the output of the trained model on the new data as the soft labels for previously seen classes and uses them as targets for the new classifier. A recent improvement on this is Learning without Memorizing (LwM), which introduces an attention distillation loss to preserve the attention maps of the classifier on previous classes while training on new data [8].

B. Knowledge distillation

Knowledge distillation is widely used to approach transfer knowledge from a pretrained model to another model [12]. Conventionally most methods using knowledge distillation require access to the previous dataset, however there are some recent approaches that follow the data-free constraints. Lopes et al. [23] attempt to reconstruct the original data from the meta-data (e.g. means and standard deviation of activations from each layer) to reconstruct the original data. Chen et al. [5] use a pretrained classifier as a fixed discriminator to train a new generator which could generate images with maximum discriminator responses.

One of the earliest attempts to synthesize images from a trained model without any access to the training data is DeepDream [29], which generates images from random noise by minimizing classification loss and an image regularization term to steer the generated image away from being unrealistic. DeepInversion [46] extends DeepDream with a new feature distribution regularization term that minimizes the distance between the feature map statistics and respective batch normalization layers' (BN) running means and variances. It shows that using the BN based regularization significantly improves the quality of the images. Recent methods employ a combination of losses like cross-entropy, BN alignment, image smoothness, and information entropy to increase diversity in generated images [40] [45].

Based on these previous methods, we provide a new approach to the scenario of imbalanced datasets. Since regularization methods only use a single model for training and retraining, they are inferior to the methods that use a generator to reconstruct previous data, hence helping the classifier retain prior knowledge. On the other hand, knowledge distillation methods mostly require a pretrained teacher model, which is not feasible for our approach because we are retraining the classifier for incremental learning with only a generator model. Therefore, we cannot use these methods [40] [45]. Instead, we use the same data reconstruction and regularization techniques of these methods without any pretrained teacher model. In the next section, we show how our approach entitled Data-Free Generative Replay (DFGR) combines all these incremental learning methods and the techniques for mitigating the ramifications of imbalanced training data.

III. Overview of DFGR

This section gives an overview of the three essential workflows of DFGR and describes its different stages of learning. For each task of the data, the model learns in two sequential stages: 1) Classifier training or re-training, 2) Generator training. The loss functions used for the classifier and the generator in the workflows shown in Fig. 2 and Fig. 3 contain references (equation numbers) to their definitions detailed in Section IV.

In the first stage, the classifier is trained on all the available classes, using focal loss [35] instead of cross-entropy loss [10]. For training the classifier, the first task uses real data only, while all subsequent tasks use a mix of real and generated data. Fig. 2a shows the corresponding workflow for the first task.

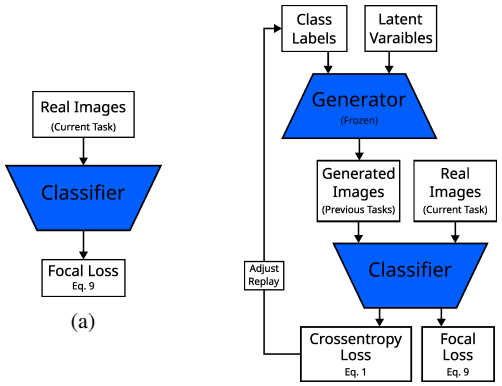


Fig. 2: a) Workflow of the classifier training with real images (for the first task). b) Workflow for retraining the classifier with real and generated images.

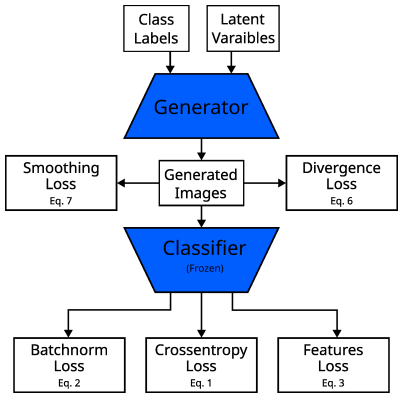


Fig. 3: Workflow for training the generator.

After the first task of data, the generated images of the previous task mixed with real images of the current task are used to retrain the old classifier. This is called classifier re-training stage, and Fig. 2b shows the associated workflow. In this stage, the real images are trained with focal loss as before due to the imbalanced dataset, but the generated images use cross-entropy loss since they are balanced by default. However, the sample ratio for the classes can be adjusted on the fly in case more generated samples are required for a sparsely populated class. This feature of adjusting replay for the generator is one of our main novelties not known from previous work.

Fig. 3 depicts the workflow of the second stage (generator training) where a class-conditional generator [28] based on BigGAN architecture [4] is trained to create images similar to the ones of the previous tasks using the trained classifier. The generator can use many different loss functions to achieve the following goals for the generated images:

1) An image corresponds to same classes the classifier has been trained on, Eq. (1)
2) An image follows a similar data-distribution as real ones

(i.e. small domain gap), Eq. (2)
3) An image has similar high-level features as real ones, Eq. (3)
4) There is a sufficient inter-class and intra-class divergence between generated images, Eq. (6)
5) An image looks realistic with minimum noise, Eq. (7)

IV. Methodology

A. Generator Training

In the generator training stage, the generator is trained with loss functions, as shown in Fig. 3, in order to achieve the five goals for the generated images mentioned in the previous section. We introduce these loss functions in the following subsections.

1) Cross-entropy loss: Cross-entropy loss is the most common loss function for training a classifier for multi-class classification. We use the Pytorch function for this loss, which implements the negative log likelihood of the softmax of the logits, given in equation (1).

$$p_y = \frac{exp(z_y)}{\sum_j exp(z_j)}$$
$$l_{ce} = -log(p_y) \tag{1}$$

As in [10], z_y indicates logits for class y and p_y indicates the softmax probability for class y. Thus, l_{ce} is the softmax loss for class y. The total softmax loss is the mean of all softmax losses in a batch.

2) Batch-normalization statistics loss: In order to reduce the data distribution gap between the generated and previous real samples, we introduce a loss that aligns batch normalization (BN) statistics, as used in DeepInversion [46]. A batch normalization layer in the model keeps the running means and variances of the feature maps of each layer, which are learned during training. These means and variances are used to normalize the input for the next layer, and thus, they reduce internal co-variate shifts of the data [13]. In particular, they can be utilized to approximate the feature map statistics of real dataset. If we assume the dataset follows Gaussian distribution, then the generated samples should essentially have similar means and variances across the feature maps as the original dataset, whose running means are stored in the pretrained network. To enforce the similarity of features in all layers, the distance between feature map statistics of real data and generated data should be minimized as expressed in equation (2).

$$l_{bn} = \sum_l \|\mu_l(\tilde{x}) - \mu_{l_{bn}}\|_2 + \sum_l \left\|\sigma_l^2(\tilde{x}) - \sigma_{l_{bn}}^2\right\|_2 \tag{2}$$

As in [45], $\tilde{x}$ indicates the feature map vector of the generated data. For layer l, $\mu_l(\tilde{x})$ and $\sigma_l^2(\tilde{x})$ denote the estimated batch-wise means and variances of the feature map, respectively, and $\mu_{l_{bn}}$ and $\sigma_{l_{bn}}^2$ are the means and variances feature map vectors of the real data stored in the batch normalization layers of the pretrained network, respectively.

3) Feature map loss: Similar to the batch normalization loss, we can also strive for minimizing the distance between features extracted from the last convolutional layer of the model, just before the fully connected layer. Since there is no BN layer between the convolutional network and the fully connected layer, the means and variances of the last layer's feature map are saved for all the classes after the classifier is trained on all the tasks. During the generator training, equation (3) is used for minimizing the distance between the last feature map's statistics of generated data and real data.

$$l_{feat} = \|\mu(\hat{x}) - \mu_m\|_2 + \left\|\sigma^2(\hat{x}) - \sigma_m^2\right\|_2 \tag{3}$$

In equation (3), $\hat{x}$ denotes the convolutional network's last layer feature maps obtained from the generated data. Moreover, $\mu(\hat{x})$ and $\sigma^2(\hat{x})$ are the estimated batch-wise means and variances of the feature maps. Here, μ_m and σ_m^2 are the overall means and variances statistics obtained from the saved feature maps of the last layer, depending on the classes in the current batch. μ_m and σ_m^2 are calculated for each batch by merging the Gaussian distributions of the classes weighted by the number of samples of each class, hence represented by subscript m. The minimization of the feature map loss ensures that the generated images have similar high level features as the real images.

In the equations (4) and (5), μ_c and σ_c^2 are the mean and variance of the last layer's feature maps for class c, respectively. Moreover, n_c denotes the number of the images in the current batch belonging to class c.

$$\mu_m = \frac{\sum_c n_c \mu_c}{\sum_c n_c} \tag{4}$$

$$\sigma_m^2 = \frac{\sum_c n_c(\sigma_c^2 + \mu_c^2)}{\sum_c n_c} - \mu_m^2 \tag{5}$$

The use of feature maps for generating images is the second main contribution of this paper. Recall that replay adjustment controls the ratio of generated images per class.

4) Sample diversification loss: Suppose the generator is forced to adjust the BN and feature map statistics of the generated samples to the statistics of the real data. In that case, this can lead to overfitting and reduced image diversity. To increase image diversity within each batch of generated images, we add another loss term as introduced by Xin et. al. [45].

$$l_{div} = D_{JS} = -\frac{1}{2}\Big(D_{KL}(s_1||s_2) + D_{KL}(s_2||s_1)\Big) \tag{6}$$

As specified in equation (6), the generated images also called fakes consist of two samples s_1 and s_2. They are obtained by first dividing the batch of generated images in two halves and then randomly selecting $2/3$ of the images from each half, respectively. The Jensen Shannon divergence (D_{JS}) between these two samples is then calculated and used as a loss for maximizing the diversity among generated images.

5) Image smoothing loss: Because generated images can have low level noise whereas natural images are generally locally smooth in pixel space, image smoothing loss is introduced as inspired by Smith et al. [40]. The minimization of image smoothing provides more natural looking images or in our case generated images that look similar to the real images previously seen. Hence, it also lowers other classifier based losses like BN loss and feature map loss. This loss corresponds to the mean square error between the generated images (gen) and a blurred version of the same images by using a Gaussian kernel ($gen_{blurred}$), as shown in equation (7).

$$l_{sm} = \|gen - gen_{blurred}\|_2^2 \tag{7}$$

6) Total generator loss: Finally, the total loss for the generator consists of a linear combination of the individual losses. The hyper-parameters can also be adjusted to only activate some of the individual losses, which we used for assessing their impact. The losses that are always activated are image smoothing loss, sample diversification loss and batch-norm loss. These losses are independent of the class labels of the generated images. The other two losses, cross-entropy loss and feature map loss depend on the classes of the generated image in each batch. Since they were considered to have high impact in mitigating the effects of imbalanced datasets, we conducted extensive tests with different combinations of these losses to determine their impacts in our approach for balanced and imbalanced data.

$$l_{total} = \delta l_{ce} + \alpha l_{feat} + \beta l_{bn} + \gamma l_{div} + \epsilon l_{sm} \tag{8}$$

In equation (8), δ, α, β, γ and ϵ denotes the hyper-parameters of the loss functions. In our experiments the values of β, γ and ϵ are always set to 1, and the associated partial sum returns the standard loss as defined in equation (11). This standard loss is the part of the total loss that is independent of either the number of images in a class of each batch or the class imbalance of the batch. To the contrary, the cross-entropy loss and features loss largely depend on the classes of images in each batch. Thus, the values of δ and α are set to 0 or 1 to create different total loss functions and control their contribution to learning imbalanced data.

B. Classifier Training/Retraining

As the classifier must train on imbalanced data on each task with unknown distribution, a method is required to make the model learn despite the imbalance in training data. The standard method for image classification is to use cross-entropy loss. However, it can only overcome minor random imbalance per batch but not major imbalance in the whole dataset. To address the imbalance in each class, the training loss for different classes must be reweighed by multiplying them with some weights. We examined Focal loss [22] and Balanced Softmax loss [35]. Other losses were also considered but were deemed similar in performance. Unfortunately, Balanced Softmax loss could not perform as desired in class-incremental learning setup but Focal loss on the other hand was able to retain some knowledge of previously seen classes.

1) Focal loss: Focal loss was initially introduced by He et. al [22] for dense object detection. Most object detection algorithms detect objects of varying size and location by searching millions of bounding boxes per image, which causes class imbalance. Focal loss is a modified version of Cross-Entropy loss that tries to handle the class imbalance problem by down-weighting easy or abundant classes and putting focus on training hard classes. For that, focal loss inversely reweights the classes using their prediction probabilities, as given in equation (9).

$$l_{fl} = -(1 - p_y)^{\gamma} log(p_y) \tag{9}$$

The value of γ in the above equation is set to 2.

2) Generator Replay Adjustment: While training the image classifier on generated images, a generated sample from a particular class can exhibit higher loss compared to other classes. We have tried to fix this difference by implementing a technique most commonly for data augmentation, by repeating or increasing the number of samples for the class with higher loss. As the generated images act as replay for previously seen data, we monitor average loss per class and adjust the frequency or probabilities of classes for the next batch of replay images. This is done by increasing the probabilities of classes in each batch with lower average losses, allowing them to have a higher chance of being replayed in the next batch.

3) Total classifier loss: The total loss for the classifier is a combination of Focal loss for the real data and cross-entropy loss for the generated data, as shown in equation (10).

$$l_{total} = (1 - m) l_{fl}^{real} + m l_{ce}^{replay} \tag{10}$$

Parameter m is the ratio of the number of previously seen classes and all classes seen so far including in the current task. This parameter prevents the classifier from forgetting the previous classes by increasing their importance depending on their number.

V. Implementation

Our implementation of DFGR is based on a combination of well-established classifier and generator architectures, as shown in Fig. 2 and Fig. 3. The classifier is derived from the ResNet model [11], a renowned deep learning architecture known for its effectiveness in image classification tasks. The generator component uses BigGAN model [4], an advanced conditional generative model capable of generating high-quality images for different classes. These models were implemented in PyTorch framework [32], on a single RTX 3060 GPU.

In order to test various scenarios and loss functions, we opted for using two well-known benchmark datasets, namely MNIST digits [20] and FashionMNIST [44]. These two datasets are chosen due to their relatively small size, making them efficient for experimentation while still being complex enough to thoroughly evaluate our hypotheses. Both MNIST and FashionMNIST contain 10 distinct classes and images are rescaled to 32×32 pixel to match the resolution requirements

of our models. Since these datasets are mostly balanced, we created new imbalanced datasets from them with predefined ratio of images per class, ranging from 100% to 10% images for each class. The original datasets served as our baseline, while the newly created imbalanced datasets were employed to rigorously test our proposed methodology. Both datasets were assigned to three tasks T_1, T_2, T_3 such that every tasks exclusively contains the images from the following list of classes:

$$BalancedData : \begin{cases} T_1 : & \{3, 4, 9\} \\ T_2 : & \{5, 6, 0\} \\ T_3 : & \{1, 2, 8, 7\} \end{cases}$$

For the specification of imbalanced datasets, we added the fraction of total images for each class after the class label (separated by a colon). Again, we examined three tasks T_1, T_2, T_3 with the following specifications:

$$ImbalancedData : \begin{cases} T_1 : & \{3 : 1.0, 4 : 0.6, 9 : 0.3\} \\ T_2 : & \{5 : 0.9, 6 : 0.4, 0 : 0.2\} \\ T_3 : & \{1 : 0.5, 2 : 0.7, 8 : 0.1, 7 : 0.8\} \end{cases}$$

For example, task T_1 receives 100%, 60%, and 30% of the images from class 3, 4, and 9, respectively. Thus, we introduce a fixed amount of class imbalance to ensure repeatability of the experiments.

The hyper-parameters for training our model were selected with careful consideration after checking many different combinations. For ResNet and BigGAN, we chose a batch size of 128 (for training the classifier) and a batch size of 32 (for training the generator), respectively. Both models were trained for 1000 epochs with early stopping, ensuring that training terminated when performance improvements stagnated, with a maximum patience of 50 epochs and 75 epochs for the classifier and generator respectively. We utilized the Adam optimizer [17] with a learning rate of $1e-4$ and a β_1 value of 0.5 to facilitate model convergence and optimization during training. Because of long training time required for training the generator and then learning each task sequentially by the classifier, it took around 8 to 12 hours on average for each experiment to run all tasks. Each experiment was repeated at least 3 times, while most of them around 5 times.

VI. Results

We conducted a series of experiments on MNIST [20] and FashionMNIST [44] datasets, considering both balanced and imbalanced dataset configurations for each. We examined four different combinations of three loss functions (standard loss l_s, cross-entropy loss l_{ce}, and feature map loss l_{feat}) for class incremental learning, with and without the incorporation of a generator replay adjustment. The standard loss function l_s combines image smoothing loss, sample diversification loss, and batch normalization loss, as shown in equation (11).

$$l_s = l_{sm} + l_{div} + l_{bn} \tag{11}$$

Method	Task I	Task II	Task II
Class 3	99.9	91.1	87.2
Class 4	99.7	91.7	89.9
Class 9	99.4	84.8	61.0
Class 5	-	99.6	60.0
Class 6	-	99.7	91.3
Class 0	-	99.9	91.5
Class 1	-	-	99.9
Class 2	-	-	99.6
Class 8	-	-	99.4
Class 7	-	-	99.5
Average	99.7	94.3	88.4

TABLE I: Per class and average accuracy (in %) of DFGR on MNIST Balanced after training of each task.

The four combinations of losses are standard loss with cross-entropy loss, standard loss with feature map loss, standard loss with both cross-entropy loss and feature map loss, and finally just the standard loss. The experiments were conducted using these losses, with and without replay adjustment (ra) in order to test its effectiveness with each loss function. Table II shows the results of these experiments for the four datasets. For each dataset, we report the final accuracy of the classifier on the test set and the average runtime after training on all tasks. For the MNIST Balanced dataset, Table I shows in more detail the development of the average accuracy for each class after the tasks completed their training. Our method DFGR provides a high class accuracy for an increasing number of classes.

As shown in Table II, the combination of all loss functions and replay adjustment gives the highest overall accuracy in our class incremental learning setup, both on balanced and imbalanced datasets. The results show that cross-entropy loss plays most significant role in improving the accuracy, especially when combined with the replay adjustment. Feature map loss also improves accuracy compared to the standard loss, but its impact is more subtle. These findings emphasized the significance of incorporating cross-entropy loss and the advantages of replay adjustment in our method DFGR to enhance the overall performance of class incremental learning models.

To provide a more comprehensive comparison of DFGR, we conducted experiments comparing it against data-free baseline and two similar methods (MFGR and DFCIL) recently proposed. We tested each method on balanced and imbalanced versions of MNIST and FashionMNIST. Table III shows the final test accuracy and average runtime after training on all tasks. In addition, the table reports the previously stored models required for each method and the model size (number of parameters).

As expected the naive method showed the lowest accuracy, underlining the necessity for sophisticated incremental learning techniques. Elastic Weight Consolidation (EWC) [18], which is a regularization based approach, performed slightly better than the naive approach. Learning without Forgetting (LwF) [21], another regularization based approach, exhibited

Methods	MNIST Balanced		MNIST Imbalanced		FashionMNIST Balanced		FashionMNIST Imbalanced	
	Acc.	Avg. Time	Acc.	Avg. Time	Acc.	Avg. Time	Acc.	Avg. Time
l_s	45.8	10:38	52.7	8:31	40.0	13:20	39.8	12:23
$l_s + l_{feat}$	71.7	10:53	58.7	8:49	40.6	13:11	39.7	10:10
$l_s + l_{ce}$	79.1	11:59	80.5	11:13	43.1	11:54	40.2	11:07
$l_s + l_{ce} + l_{feat}$	81.7	9:25	81.5	8:09	45.1	11:33	40.3	9:32
$l_s + ra$	58.0	11:43	53.9	9:43	41.4	10:47	40.0	9:54
$l_s + l_{feat} + ra$	78.9	9:30	59.7	8:38	43.3	9:51	40.4	7:56
$l_s + l_{ce} + ra$	87.5	8:45	87.4	8:24	43.6	8:29	42.2	8:31
$l_s + l_{ce} + l_{feat} + ra$	88.4	8:35	88.5	7:10	46.6	8:40	43.6	8:20

TABLE II: Accuracy (in %) and average run times (hh:mm) for every dataset and every combination of loss functions and replay adjust (ra).

Methods	Models Saved	Model Size (Millions)	MNIST Balanced		MNIST Imbalanced		FashionMNIST Balanced		FashionMNIST Imbalanced	
			Acc.	Avg. Time	Acc.	Avg. Time	Acc.	Avg. Time	Acc.	Avg. Time
Naive (Lower Limit)	-	-	41.5	0:34	41.1	0:26	39.9	0:44	39.2	0:22
EWC [18]	-	-	47.5	1:23	46.9	0:52	39.9	1:28	39.5	0:58
LWF [21]	Classifier	19.6 M	58.2	1:10	55.5	0:38	41.4	1:08	40.3	0:44
MFGR [45]	Classifier + Generator	19.6 M + 3.2 M	66.2	15:32	65.8	16:35	42.3	16:05	41.2	16:32
DFCIL [40]	Classifier + Generator	19.6 M + 3.2 M	83.2	13:17	81.1	14:43	48.3	13:27	32.9	14:25
DFGR (Ours)	Generator + Features	3.2 M + 41 K	88.4	8:35	88.5	7:10	46.6	8:40	43.6	8:20

TABLE III: Accuracy (in %) and average run times (hh:mm) for baseline and competitive methods.

improved accuracy compared to EWC, primarily due to its utilization of a classifier trained on previous data to preserve knowledge. In our experimental comparisons, we examined two methods that are designed for data-free learning like our method: Always be dreaming (DFCIL) [40] and MFGR [45]. Both of these methods employ a classifier trained on previous tasks as a teacher for training the generator and the new classifier on the current task. As a consequence, they have to store both of these models, significantly increasing the memory requirements for incremental learning.

Our method DFGR emerged as a clear winner in our experiments as it demonstrates higher accuracy than all other methods, as shown in Table III. Among those, only MFGR and DFCIL partly obtain similar results. For one dataset (FashionMNIST Balanced), the accuracy of DFCIL is even slightly better. For FashionMNIST Imbalanced, however, the accuracy of DFGR is more than 10% higher than DFCIL. In addition, DFGR only requires 15% the storage (for the generator parameters and feature map statistics) compared to DFCIL and MFGR. Thus, we conclude that it is not necessary to maintain large classifier models as teachers even if the datasets are balanced. Moreover, these methods require also more time on average for training compared to DFGR. In summary, DFGR offers a balance between memory efficiency, runtime, and accuracy, making it a promising solution for class incremental learning in resource-constrained data-free environments. It also offers the unique feature to address class imbalance during training.

VII. Conclusion

This paper proposes DFGR, a novel approach for incremental learning on imbalanced datasets with data-free generative replay. DFGR offers high accuracy without storing any replay data or data from previous tasks, but training only a generator using the classifier trained on previous data, in order to preserve knowledge and to reconstruct data for incremental learning. Unlike other similar approaches, which require a trained classifier as teacher and generator to reconstruct data, DFGR only needs a generator. We examined various loss functions like feature map loss for training the generator and focal loss for imbalanced data, and proposed generator replay adjustment for data augmentation of sparse classes. Based on a preliminary set of experiments, we obtained parameter settings and showed the superiority of DFGR compared to other data-free learning approaches.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

[2] Yogesh Balaji, Mehrdad Farajtabar, Dong Yin, Alex Mott, and Ang Li. The effectiveness of memory replay in large scale continual learning. *arXiv preprint arXiv:2010.02418*, 2020.

[3] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3514–3522, 2019.

[6] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pages 1952–1961. PMLR, 2020.

[7] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[8] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019.

[9] Timothy J Draelos, Nadine E Miner, Christopher C Lamb, Jonathan A Cox, Craig M Vineyard, Kristofor D Carlson, William M Severa, Conrad D James, and James B Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *2017 international joint conference on neural networks (IJCNN)*, pages 526–533. IEEE, 2017.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[14] Alexis Joly, Hervé Goëau, Pierre Bonnet, Concetto Spampinato, Hervé Glotin, Andreas Rauber, Willem-Pier Vellinga, Robert Fisher, and Henning Müller. Are species identification tools biodiversity-friendly? In *Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data*, pages 31–36, 2014.

[15] Nitin Kamra, Umang Gupta, and Yan Liu. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368*, 2017.

[16] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[19] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[20] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[21] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[23] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

[24] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[25] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

[26] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[27] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504, 2013.

[28] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[29] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog*, 2015, 2015.

[30] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

[31] Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[33] Lorenzo Pellegrini, Vincenzo Lomonaco, Gabriele Graffieti, and Davide Maltoni. Continual learning at the edge: Real-time training on smartphone devices. *arXiv preprint arXiv:2105.13127*, 2021.

[34] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[35] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

[36] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[38] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[39] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[40] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9374–9384, 2021.

[41] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[42] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. *Advances in neural information processing systems*, 30, 2017.

[43] Shixian Wen and Laurent Itti. Overcoming catastrophic forgetting problem by weight consolidation and long-term memory. *arXiv preprint arXiv:1805.07441*, 2018.

[44] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[45] Xiaomeng Xin, Yiran Zhong, Yunzhong Hou, Jinjun Wang, and Liang Zheng. Memory-free generative replay for class-incremental learning. *arXiv preprint arXiv:2109.00328*, 2021.

[46] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[47] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[48] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online incremental feature learning with denoising autoencoders. In *Artificial intelligence and statistics*, pages 1453–1461. PMLR, 2012.

5. Conclusion

The novel research contributions presented in this thesis consist of a comprehensive multifaceted exploration of deep learning applications within the domain of biodiversity and ecology, with a primary emphasis on the image analysis of digitized herbarium scans. The thesis comprises three main studies, supported by accompanying publications and conference papers. The first study discusses implementing a deep learning framework for automated species and trait recognition of herbarium species, encompassing 1000 species. This pioneering effort and other automated species recognition methods have marked a significant leap forward in biodiversity research. With the increasing availability of deep learning algorithms, domain scientists can now efficiently analyze species distribution patterns and ecological traits of herbarium specimens while mitigating potential misidentifications in the archived herbarium collections and accelerate conservation of threatened species [3].

Based on the fundamental results of the first study, the second study delves into the problem of detecting various plant organs within herbarium scans. Detecting the presence of specific plant organs, such as flowers, fruits, leaves, and stems, not only offers invaluable insights for phenological studies but also holds practical applications across various fields. For instance, identifying leaf structure and fruit characteristics is essential in agriculture, aiding in crop yield estimation, pest management, and nutrient level assessment. In essence, these approaches presented in the thesis showcase the potential of deep learning in automating previously labor-intensive tasks of species identification and plant organ detection on herbarium specimens.

Despite these valuable achievements, several bottlenecks persist in the deep learning process, such as inherently imbalanced data in nature, a scarcity of labeled data for specialized applications, and a constant influx of new data, which can render trained models quickly outdated. To address these problems, a novel continual learning approach tailored for imbalanced data is proposed in the third study. This approach leverages data-free generative replay to preserve the learned knowledge and reconstruct previous data. Based on the experimental testing on two benchmark datasets, DFGR exhibited exceptional performance compared to similar data-free learning methods. In addition, DFGR makes efficient use of

computational sources, such as memory, and significantly reduces the training time while exhibiting high accuracy. These attributes make DFGR a promising solution for learning in resource-constrained data-free environments, such as edge devices.

The future of deep learning for herbarium scans appears promising, as demonstrated by recent advancements. Notably, transfer learning shows potential by leveraging a models pre-trained on similar large datasets to compensate for limited herbarium resources in certain regions. The performance on herbarium scans can be further enhanced by designing a tailor-made training pipelines, incorporating preprocessing steps to remove background tags and collector notes. Moreover, plant organ detection can be adapted to real-world conditions for various applications, including leaf and fruit counting, disease and pest detection, and automated harvesting platforms.

Although DFGR is a proof of concept, its testing was confined to a small benchmark dataset due to time and computational constraints. Subsequent steps should include testing it on larger generic datasets like CIFAR10/100 and ImageNet, to evaluate its scalability on a large number of classes. Additionally, future work should explore specialized continual learning datasets, such as CORe50 and CLEAR [62, 60], to access DFGR's ability to learn incremental tasks. To train on these large datasets, the model size needs to be increased to facilitate learning, potentially impacting its performance while providing an opportunity to further study the impact of the individual loss functions. Currently, limited attention is given to optimizing loss function weights, which needs to be explored further. Given the evolving nature of continual learning, novel learning techniques can be incorporated to generate data reflective of real distributions and mimic previous data as closely as possible, to increase DFGR's performance and adaptability.

A. Appendix

A.1 Supporting Publication A

This publication extends taxon and trait recognition from herbarium scans by refining the process of directly extracting plant organ traits, instead of inferring them from their species. The herbarium scans containing collector notes were obtained from four distinct collections, which were used to detect the presence of flowers, leaves, and fruits. The trait information of these organs, leveraged from the FLOPO knowledge base, was merged with the herbarium scans to provide a comprehensive database consisting of 13,157 images representing 2,339 unique species. Despite the model's overall good performance for detecting traits, particularly for leaves, limitations that affected the performance such as a limited dataset and some underrepresented organs were acknowledged.

Contribution Role: Lead Author/Presenter

Presented in:

10^{th} International Conference on Ecological Informatics, 2018

Proceedings of the 10th International Conference on Ecological Informatics

Extracting Trait Data from Digitized Herbarium Specimens Using Deep Convolutional Networks

Sohaib Younis[1,2], Marco Schmidt[1,3], Claus Weiland[1], Stefan Dressler[4], Susanne Tautenhahn[5], Jens Kattge[5], Bernhard Seeger[2], Thomas Hickler[1,6], Robert Hoehndorf[7]

[1]SBiK-F, Germany; *[2]University of Marburg, Germany*; *[3]Palmengarten, Germany*; *[4]SRI-NHM, Germany;* *[5]MPI-BGC, Germany*; *[6]Goethe University, Germany*; *[7]KAUST, Saudi Arabia*

Corresponding author e-mail: Muhammad-Sohaib.Younis@senckenberg.de

ABSTRACT:

Herbarium collections have been the foundation of taxonomical research for centuries and become increasingly important for related fields such as plant ecology or biogeography. Herbaria worldwide are estimated to include c. 400 million specimens, by inclusion of type specimens cover with few exceptions all known plant taxa (c. 350 000 species) and have a temporal dimension that is reached by only few other botanical data sources.

Presently, c. 13.5 million digitized herbarium specimens are available online via institutional websites or aggregating websites like GBIF. We used these specimen images in combination with morphological trait data obtained from TRY and the FLOPO knowledge base in order to train deep convolutional networks to recognize these traits as well as phenological states from specimen images. To improve trait recognition, we expanded our approach to include high resolution scans to enable fine grain feature extraction. Furthermore we analyze differences in recognizability of traits depending on trait group (e.g. leaf traits) or higher taxa. Newly mobilized trait data will be used to improve our trait databases. Our approach is described in detail and performance in the recognition of different traits is analyzed and discussed.

KEYWORDS: Trait Recognition, Deep Convolutional Neural Network, Plant Phenotyping, Digitized Natural History Collections, Image Processing

A.2 Supporting Publication B

This poster illustrates the workflow of plant organ detection using lifelong learning. It outlines the various steps involved in the process, from preprocessing and annotation of herbarium scans to training of the deep learning model, followed by continuous improvement of the model performance with iterative refinement of the annotated data.

Contribution Role: Lead Author/Presenter

Presented in:

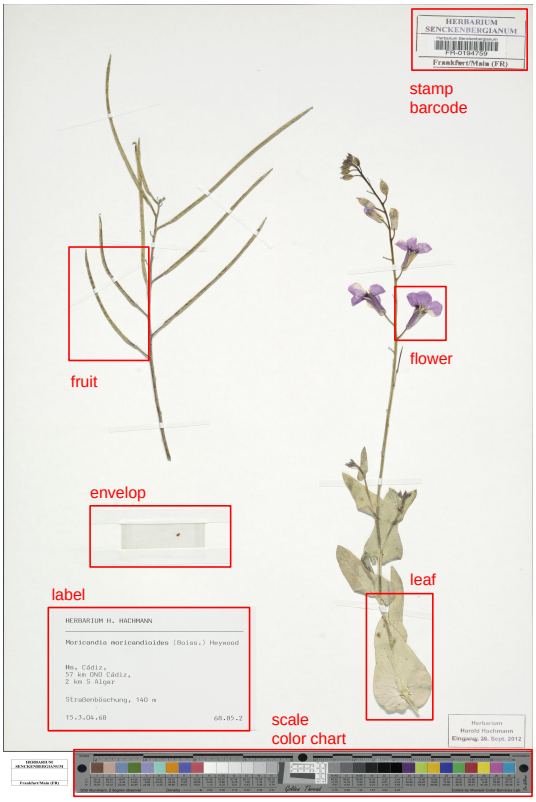
Biodiversity Information Science And Standards, 2019

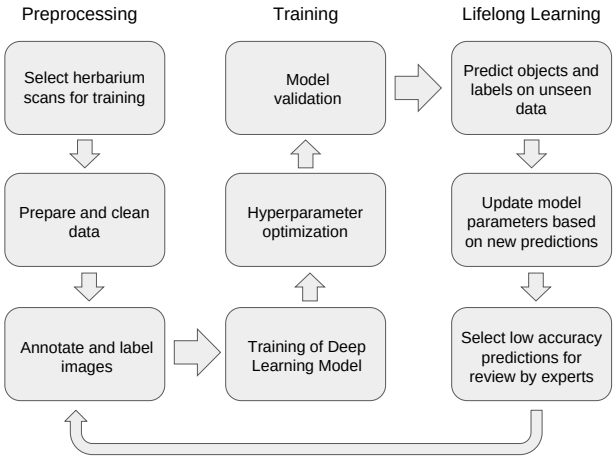
A workflow for data extraction from digitized herbarium specimens

Sohaib Younis [1,2], Marco Schmidt [1,3,4], Bernhard Seeger [2], Thomas Hickler [1,5], Claus Weiland [1]

[1] Senckenberg Biodiversity and Climate Research Centre (SBiK-F), Frankfurt am Main, Germany, [2] Department of Mathematics and Computer Science, University of Marburg, Marburg, Germany, [3] Palmengarten der Stadt Frankfurt am Main, Frankfurt am Main, Germany, [4] Department of Botany and Molecular Evolution, Senckenberg Research Institute and Natural History Museum Frankfurt, Frankfurt am Main, Germany, [5] Department of Physical Geography, Goethe University, Frankfurt am Main, Germany, e-mail: Muhammad-Sohaib.Younis@senckenberg.de

Herbarium specimens are collection objects with a high density of information. Different working groups have recently made progress in extracting such information. We propose here a workflow including all these approaches for comprehensive data extraction.





1 - Object detection and segmentation

In order to specifically analyse different parts of the herbarium sheet, it is necessary to identify these:
1. preserved plant material as well as additional objects,
2. the label containing information on the collection event and identification,
3. annotations such as revision labels, or notes on material extraction,
4. identifiers such as barcodes or numbers,
5. envelopes for loose plant material and
6. often scale bars and color charts used in the digitization process.

This step could be taken follow the methods of Triki et al. (2018).

2 - Plant organ measurements and trait extraction

Leafs, flowers and fruits may be measured for quantitative traits and morphological traits specific for these organs may be identified. Such approaches have been provided by Gaikwad et al. (2018) and Younis et al. (2018).

3 - Extraction of text from labels and annotations

Different text elements on the herbarium sheet often including cryptic abbreviations (Schröder 2019) contain crucial information documenting the plant material and collection event. Kirchhoff et al. (2018) developed an OCR-based approach that could be applied here. Similarities on labels may be helpful even where readability is restricted.

4 - Taxon recognition

Deep-learning-based taxon recognition (as in Younis et al. 2018) may be helpful not only as a step in identifying unidentified material (e.g. vegetative 'ecologist specimens'), but also in finding misidentifications or mislabellings.

Gaikwad J, Triki A, Bouaziz B, Hamed H, Hentschel J (2018) TraitEx: tool for measuring morphological functional traits from digitized herbarium specimens. Proceedings of the 10th International Conference on Ecological Informatics. Jena
Kirchhoff A, Bügel U, Santamaria E, Reimeier F, Röpert D, Tebbje A, Güntsch A, Chaves F, Steinke K, Berendsohn W (2018) Toward a service-based workflow for automated information extraction from herbarium specimens. Database 2018 https://doi.org/10.1093/database/bay103
Schröder CN (2019) Katalog der auf Herbarbelegen gebräuchlichen Abkürzungen. Kochia 12: 37-67.
Triki A, Bouaziz B, Gaikwad J (2018) Refined methodology for accurately detecting objects from digitized herbarium specimens. Proceedings of the 10th International Conference on Ecological Informatics. Jena
Younis S, Weiland C, Hoehndorf R, Dressler S, Hickler T, Seeger B, Schmidt M (2018) Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks. Botany Letters 165: 377-383. https://doi.org/10.1080/23818107.2018.1446357

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

[2] Abbott, L. F. and Nelson, S. B. (2000). Synaptic plasticity: taming the beast. *Nature neuroscience*, 3(11):1178–1183.

[3] Albani Rocchetti, G., Armstrong, C. G., Abeli, T., Orsenigo, S., Jasper, C., Joly, S., Bruneau, A., Zytaruk, M., and Vamosi, J. C. (2021). Reversing extinction trends: new uses of (old) herbarium specimens to accelerate conservation action on threatened species. *New Phytologist*, 230(2):433–450.

[4] Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. (2018). Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154.

[5] Balaji, Y., Farajtabar, M., Yin, D., Mott, A., and Li, A. (2020). The effectiveness of memory replay in large scale continual learning. *arXiv preprint arXiv:2010.02418*.

[6] Bank, D., Koenigstein, N., and Giryes, R. (2023). Autoencoders. *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*, pages 353–374.

[7] Barré, P., Stöver, B. C., Müller, K. F., and Steinhage, V. (2017). Leafnet: A computer vision system for automatic plant species identification. *Ecological Informatics*, 40:50–56.

[8] Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

[9] Carranza-Rojas, J., Goeau, H., Bonnet, P., Mata-Montero, E., and Joly, A. (2017). Going deeper in the automated identification of herbarium specimens. *BMC evolutionary biology*, 17(1):1–14.

[10] Chagnoux, S. et al. (2020). The vascular plants collection (p) at the herbarium of the muséum national d'histoire naturelle (mnhn-paris). *MNHN-Museum national d'Histoire naturelle*.

[11] Chrysakis, A. and Moens, M.-F. (2020). Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pages 1952–1961. PMLR.

[12] Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

[13] Collett, R. A. and Fisher, D. O. (2017). Time-lapse camera trapping as an alternative to pitfall trapping for estimating activity of leaf litter arthropods. *Ecology and Evolution*, 7(18):7527–7533.

[14] Corney, D. P., Clark, J. Y., Tang, H. L., and Wilkin, P. (2012). Automatic extraction of leaf characters from herbarium specimens. *Taxon*, 61(1):231–244.

[15] Cubey, R. (2018). Royal botanic garden edinburgh living plant collections (e).

[16] Davis, G. W. (2006). Homeostatic control of neural activity: from phenomenology to molecular design. *Annu. Rev. Neurosci.*, 29:307–323.

[17] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[18] Dhar, P., Singh, R. V., Peng, K.-C., Wu, Z., and Chellappa, R. (2019). Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146.

[19] Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25.

[20] Dong, S., Wang, P., and Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, 40:100379.

[21] Dressler, S., Schmidt, M., and Zizka, G. (2014). Introducing african plants—a photo guide—an interactive photo database and rapid identification tool for continental africa. *Taxon*, 63(5):1159–1164.

[22] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

[23] GBIF, G. (2020). The global biodiversity information facility (year) what is gbif. *Available from [13 January 2020]*.

[24] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

[25] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

[26] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

[27] Goeau, H., Bonnet, P., and Joly, A. (2017). Plant identification based on noisy web data: the amazing performance of deep learning (lifeclef 2017). CEUR Workshop Proceedings.

[28] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[29] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

[30] Grant, S. and von Konrat, M. (2016). Royal botanic gardens, kew-herbarium specimens. *Royal Botanic Gardens, Kew. DOI: https://doi. org/10.15468/ly60bx*.

[31] Gupta, S., Hoffman, J., and Malik, J. (2016). Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2827–2836.

[32] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[33] Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology press.

[34] Heberling, J. M. and Burke, D. J. (2019). Utilizing herbarium specimens to quantify historical mycorrhizal communities. *Applications in plant sciences*, 7(4):e01223.

[35] Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier.

[36] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[37] Hoehndorf, R., Alshahrani, M., Gkoutos, G. V., Gosline, G., Groom, Q., Hamann, T., Kattge, J., De Oliveira, S. M., Schmidt, M., Sierra, S., et al. (2016). The flora phenotype ontology (flopo): tool for integrating morphological traits and phenotypes of vascular plants. *Journal of biomedical semantics*, 7(1):1–11.

[38] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2017a). Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311.

[39] Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., and Belongie, S. (2017b). Stacked generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5077–5086.

[40] Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243.

[41] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

[42] Johndrow, J. E., Smith, A., Pillai, N., and Dunson, D. B. (2016). Inefficiency of data augmentation for large sample imbalanced data. *arXiv preprint arXiv:1605.05798*.

[43] Joly, A., Bonnet, P., Goëau, H., Barbe, J., Selmi, S., Champ, J., Dufour-Kowalski, S., Affouard, A., Carré, J., Molino, J.-F., et al. (2016). A look inside the pl@ ntnet experience: The good, the bias and the hope. *Multimedia Systems*, 22:751–766.

[44] Joly, A., Goëau, H., Bonnet, P., Spampinato, C., Glotin, H., Rauber, A., Vellinga, W.-P., Fisher, R., and Müller, H. (2014). Are species identification tools biodiversity-friendly? In *Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data*, pages 31–36.

[45] Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.-P., Planqué, R., Rauber, A., Palazzo, S., Fisher, B., et al. (2015). Lifeclef 2015: multimedia life species identification challenges. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 6th International Conference of the CLEF Association, CLEF'15, Toulouse, France, September 8-11, 2015, Proceedings 6*, pages 462–483. Springer.

[46] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.

[47] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[48] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

[49] Kong, C. and Lucey, S. (2017). Take it in your stride: Do we need striding in cnns? *arXiv preprint arXiv:1712.02502*.

[50] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[51] Kumar, N., Belhumeur, P. N., Biswas, A., Jacobs, D. W., Kress, W. J., Lopez, I. C., and Soares, J. V. (2012). Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12*, pages 502–516. Springer.

[52] Lang, P. L., Willems, F. M., Scheepens, J., Burbano, H. A., and Bossdorf, O. (2019). Using herbaria to study global environmental change. *New phytologist*, 221(1):110–122.

[53] Le Bras, G., Pignal, M., Jeanson, M. L., Muller, S., Aupic, C., Carré, B., Flament, G., Gaudeul, M., Gonçalves, C., Invernón, V. R., et al. (2017). The french muséum national d'histoire naturelle vascular plant herbarium collection dataset. *Scientific Data*, 4(1):1–16.

[54] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[55] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2.

[56] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[57] Li, Z. and Hoiem, D. (2017). Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

[58] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

[59] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

[60] Lin, Z., Shi, J., Pathak, D., and Ramanan, D. (2021). The clear benchmark: Continual learning on real-world imagery. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*.

[61] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer.

[62] Lomonaco, V. and Maltoni, D. (2017). Core50: a new dataset and benchmark for continuous object recognition. In *Conference on robot learning*, pages 17–26. PMLR.

[63] Mäder, P., Boho, D., Rzanny, M., Seeland, M., Wittich, H. C., Deggelmann, A., and Wäldchen, J. (2021). The flora incognita app–interactive plant species identification. *Methods in Ecology and Evolution*, 12(7):1335–1342.

[64] McClelland, J. L., McNaughton, B. L., and O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.

[65] McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

[66] Mermillod, M., Bugaiska, A., and Bonin, P. (2013). The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects.

[67] Mi, L., Shen, M., and Zhang, J. (2018). A probe towards understanding gan and vae models. *arXiv preprint arXiv:1812.05676*.

[68] Miller, K. D. and MacKay, D. J. (1994). The role of constraints in hebbian learning. *Neural computation*, 6(1):100–126.

[69] Möglich, J. M., Lampe, P., Fickus, M., Younis, S., Gottwald, J., Nauss, T., Brandl, R., Brändle, M., Friess, N., Freisleben, B., et al. (2023). Towards reliable estimates of abundance trends using automated non-lethal moth traps. *Insect Conservation and Diversity*.

[70] Mordvintsev, A., Olah, C., and Tyka, M. (2015). Inceptionism: Going deeper into neural networks.

[71] Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., and Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725.

[72] Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

[73] Ott, T., Palm, C., Vogt, R., and Oberprieler, C. (2020). Ginjinn: An object-detection pipeline for automated feature extraction from herbarium specimens. *Applications in Plant Sciences*, 8(6):e11351.

[74] Otte, V., Wesche, K., Dressler, S., Zizka, G., Hoppenrath, M., and Kienast, F. (2011). The new herbarium senckenbergianum: Old institutions under a new common roof. *Taxon*, 60(2):617–618.

[75] Pacha, A., Hajič Jr, J., and Calvo-Zaragoza, J. (2018). A baseline for general music object detection with deep learning. *Applied Sciences*, 8(9):1488.

[76] Pandey, M., Fernandez, M., Gentile, F., Isayev, O., Tropsha, A., Stern, A. C., and Cherkasov, A. (2022). The transformational role of gpu computing and deep learning in drug discovery. *Nature Machine Intelligence*, 4(3):211–221.

[77] Papadopoulos, D. P., Uijlings, J. R., Keller, F., and Ferrari, V. (2017). Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision*, pages 4930–4939.

[78] Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71.

[79] Pinto, R. C. (2011). Online incremental one-shot learning of temporal sequences.

[80] Pound, M. P., Atkinson, J. A., Townsend, A. J., Wilson, M. H., Griffiths, M., Jackson, A. S., Bulat, A., Tzimiropoulos, G., Wells, D. M., Murchie, E. H., et al. (2017). Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *Gigascience*, 6(10):gix083.

[81] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., and Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36.

[82] Qu, H., Rahmani, H., Xu, L., Williams, B., and Liu, J. (2021). Recent advances of continual learning in computer vision: An overview. *arXiv preprint arXiv:2109.11369*.

[83] Read, J. and Žliobaitė, I. (2022). Learning from data streams: An overview and update. *arXiv preprint arXiv:2212.14720*.

[84] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

[85] Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al. (2020). Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186.

[86] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

[87] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. (2014). Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

[88] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

[89] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252.

[90] Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77:157–173.

[91] Rzanny, M., Seeland, M., Wäldchen, J., and Mäder, P. (2017). Acquiring and preprocessing leaf images for automated plant identification: understanding the tradeoff between effort and information gain. *Plant methods*, 13(1):1–11.

[92] S, C. (2018). Herbarium specimens of université de montpellier 2, institut de botanique (mpu).

[93] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *sensors*, 16(8):1222.

[94] Schuettpelz, E., Frandsen, P. B., Dikow, R. B., Brown, A., Orli, S., Peters, M., Metallo, A., Funk, V. A., and Dorr, L. J. (2017). Applications of deep convolutional neural networks to digitized natural history collections. *Biodiversity data journal*, (5).

[95] Secretariat, G. (2017). Gbif backbone taxonomy. *Accessed via https://www.gbif.org/species/6 in 2017*.

[96] Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

[97] Silva, T. S. (2017). A short introduction to generative adversarial networks. *https://sthalles.github.io*.

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

[99] Smith, J., Hsu, Y.-C., Balloch, J., Shen, Y., Jin, H., and Kira, Z. (2021). Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9374–9384.

[100] Sun, J., He, X., Ge, X., Wu, X., Shen, J., and Song, Y. (2018). Detection of key organs in tomato based on deep migration learning in a complex background. *Agriculture*, 8(12):196.

[101] Thiers, B. M. (2018). *The world's herbaria 2017: A summary report based on data from Index Herbariorum*. William and Lynda Steere Herbarium, The New York Botanical Garden.

[102] Thrun, S. and Mitchell, T. M. (1995). Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46.

[103] Tzutalin, D. (2015). tzutalin/labelimg. *Github*.

[104] Ubbens, J. R. and Stavness, I. (2017). Deep plant phenomics: a deep learning platform for complex plant phenotyping tasks. *Frontiers in plant science*, 8:1190.

[105] Van de Ven, G. M., Siegelmann, H. T., and Tolias, A. S. (2020). Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):4069.

[106] Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. (2018). The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778.

[107] Van Horn, G., Mac Aodha, O., Song, Y., Shepard, A., Adam, H., Perona, P., and Belongie, S. (2017). The inaturalist challenge 2017 dataset. *arXiv preprint arXiv:1707.06642*, 1(2):4.

[108] Villa, A. G., Salazar, A., and Vargas, F. (2017). Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. *Ecological informatics*, 41:24–32.

[109] Wäldchen, J. and Mäder, P. (2018). Machine learning for image based species identification. *Methods in Ecology and Evolution*, 9(11):2216–2225.

[110] Wäldchen, J. and Mäder, P. (2019). Flora incognita–wie künstliche intelligenz die pflanzenbestimmung revolutioniert: Botanik. *Biologie in unserer Zeit*, 49(2):99–101.

[111] Weaver, W. N., Ng, J., and Laport, R. G. (2020). Leafmachine: Using machine learning to automate leaf trait extraction from digitized herbarium specimens. *Applications in Plant Sciences*, 8(6):e11367.

[112] Willis, C. G., Ellwood, E. R., Primack, R. B., Davis, C. C., Pearson, K. D., Gallinat, A. S., Yost, J. M., Nelson, G., Mazer, S. J., Rossington, N. L., et al. (2017). Old plants, new tricks: Phenological research using herbarium specimens. *Trends in ecology & evolution*, 32(7):531–546.

[113] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2. https://github.com/facebookresearch/detectron2.

[114] Xin, X., Zhong, Y., Hou, Y., Wang, J., and Zheng, L. (2021). Memory-free generative replay for class-incremental learning. *arXiv preprint arXiv:2109.00328*.

[115] Yin, H., Molchanov, P., Alvarez, J. M., Li, Z., Mallya, A., Hoiem, D., Jha, N. K., and Kautz, J. (2020). Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724.

[116] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27.

[117] Younis, S., Schmidt, M., Seeger, B., Hickler, T., and Weiland, C. (2019). A workflow for data extraction from digitized herbarium specimens. *Biodiversity Information Science and Standards*, 3:e35190.

[118] Younis, S., Weiland, C., Hoehndorf, R., Dressler, S., Hickler, T., Seeger, B., and Schmidt, M. (2018). Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks. *Botany Letters*, 165(3-4):377–383.

[119] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer.

[120] Zenke, F., Gerstner, W., and Ganguli, S. (2017a). The temporal paradox of hebbian learning and homeostatic plasticity. *Current opinion in neurobiology*, 43:166–176.

[121] Zenke, F., Poole, B., and Ganguli, S. (2017b). Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR.

[122] Zhao, Z.-Q., Zheng, P., Xu, S.-t., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232.

[123] Zheng, L., Yang, Y., and Tian, Q. (2017). Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244.