I2C-UHU-PERSEUS at *PlantClef 2025*: Multi-Label Identification and Classification of Plant Species in Images Using Object Detection Techniques

Notebook for the PLantCLEF Lab at CLEF 2025

Jesús Tejón Carrillo[1], David Prieto Araujo[1], Victoria Pachón Álvarez[1] and
Jacinto Mata Vázquez[1]

[1] *I2C Research Group, University of Huelva, Spain*

Abstract

This work presents a hybrid pipeline for the automatic identification and classification of plant species using object detection and deep learning techniques. Specifically, the approach combines YOLOv11 for the detection of relevant regions in images and InceptionV3 for multi-label classification of the detected species. The methodology was evaluated within the context of the PlantCLEF 2025 challenge, which involves multi-species classification from natural vegetation quadrat images. To address the high class imbalance, random undersampling and data augmentation techniques were applied. Despite computational constraints that required dataset reduction, the proposed pipeline achieved an F1-score of 0.02203, ranking 28th out of 38 participating teams. In comparison, the top-performing team achieved a score of 0.38132. These results, although modest, highlight the potential of integrating object detection as a pre-classification step in complex natural scenarios.

Keywords

Deep Learning, Python, Convolutional Neural Networks, Machine Learning, Image Detection.

1. Introduction

Nowadays, precise plant species classification tasks are carried out in the fields of botany, environmental conservation, and biodiversity monitoring. Traditionally, this work was performed by specialists in the area, which implies high time and resource costs, in addition to the possibility of human errors. It is for this reason that, given these limitations, the use of techniques that combine computer vision with deep learning becomes essential to automate and scale the recognition of different plant species.

Recent advances in the field of Convolutional Neural Networks (CNNs) have demonstrated that this type of network has a remarkable ability to extract and learn features from plant images, including variable backgrounds, scales, and positions [1]. Examples of this type of network include ResNet[2], DenseNet[3], and more recent ones such as Vision Transformers[4], which, together with a rich and diversified dataset in term of classes, number of images datasets such as ImageNet, can significantly improve performance in multiclass classification tasks in the field of computer vision, including automatic classification of plant species, flowers, or fruits. Although architectures like ResNet or Vision Transformers have shown good results, InceptionV3 was chosen due to its balance between performance and efficiency, particularly useful in training environments with limited resources.

However, multiclass classification of plant species presents significant challenges, such as visual similarity between species, proximity between species that can lead to confusion, class imbalance in the data, and differences between training and test images.[5]

In this work, a solution was proposed a solution based on a hybrid detection and classification pipeline, combining the YOLOv11 model for localization of relevant regions in images with an InceptionV3 network for multi-label classification of detected species. This approach allows us to simultaneously

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

†These authors contributed equally.

jesus.tejon@alu.uhu.es (J. T. Carrillo); david.prieto707@alu.uhu.es (D. P. Araujo); vpachon@uhu.es (V. P. Álvarez); mata@uhu.es (J. M. Vázquez)

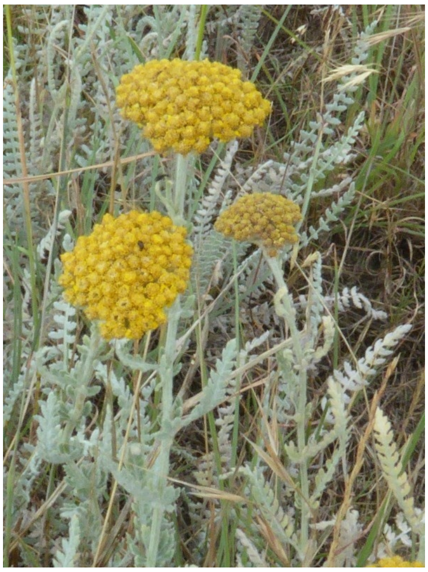

Figure 1: Typical image from the training set.

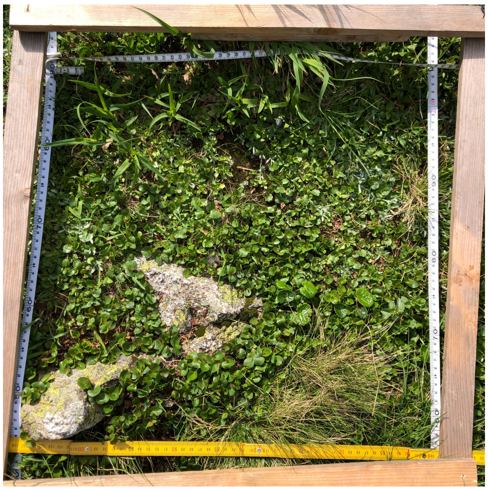

Figure 2: Typical image from the test set.

address the detection of multiple species within the same image, as well as mitigate problems arising from class imbalance and variability between training and test domains. Through this approach, the objective was to improve accuracy and robustness in plant species classification scenarios in real and uncontrolled images.

2. Context

This work is focused on providing a solution to the problem posed in the *PlantCLEF 2025* [6] competition, which provides a dataset composed of a total of 7806 distinct plant classes that must be classified. The dataset presents a marked class imbalance, with some species represented by only a single image, while others have up to 823 samples.

Another important challenge lies in the difference between training and test images. Due to computational resource constraints, the original training dataset had to be reduced in size. This selection was carried out with the goal of preserving class diversity while ensuring feasible training times and memory usage. Although this reduction may limit overall model generalization, it was necessary to adapt the approach to the available hardware. Training images typically show a single plant, generally taken from a vertical or lateral perspective, with the object centered. In contrast, test images are captured from a zenithal (top-down) perspective and usually include multiple plants in the same frame, which introduces greater complexity to the inference process.

This type of CLEF competitions, specifically those of LifeCLEF [7], help to perform different comparative evaluations in the field of computer vision applied to biodiversity.

3. Related Work

Previous studies have investigated plant species detection through the use of computer vision techniques. Among these, we can see studies related to fruit detection [8], disease detection in leaves [9] as well as in complete plants [10], and detection of medicinal plants [11]. Furthermore, these technologies have demonstrated great importance in the field of agriculture, where they contribute to optimizing crop monitoring, pest control, and management of different resources [12].

However, most of these works focus on controlled scenarios or direct classification tasks, with good quality images and homogeneous conditions. Few address the problem from a prior detection approach or multi-label classification in unstructured natural environments, which limits their applicability in real situations.

In the context of competitions like PlantCLEF, which poses realistic challenges from images collected by citizens (with noise, variable lighting, partially visible species, etc.), some recent works have explored more robust solutions. For example, in PlantCLEF 2023, the NEUON team employed architectures such as Inception-v4 and Inception-ResNet-v2, along with data augmentation strategies and organ-specific training (leaves, flowers, fruits, etc.), achieving outstanding performance [13]. Additionally, in PlantCLEF 2024, approaches based on Vision Transformers for multi-label classification in vegetation plot images were explored, addressing the challenges of variability in capture conditions and the presence of multiple species per image [14].

Despite these advances, many previous approaches remain limited in terms of their generalization, especially when facing poorly represented species or when a single image contains multiple relevant labels. In this work, we propose a hybrid pipeline based on prior detection with YOLOv11 and multi-label classification with InceptionV3, specifically designed to address these challenges. Unlike traditional direct classification approaches, the proposed methodology attempts to detected regions of interest before classification, which may help improve accuracy in complex scenarios such as those posed by PlantCLEF 2025.

4. Methodology

4.1. Models Used

- **InceptionV3**
  In this project, this model was used for the multiclass classification part of plant species in different images. InceptionV3 is a convolutional network architecture, developed by Google, that stands out for its efficiency and high performance in tasks related to computer vision [15]. This model has been previously employed in plant species classification tasks, such as flower classification, obtaining very good results [16]. More recent architectures, such as EfficientNet, were not chosen due to hardware limitations and training time constraints. Additionally, although domain-specific models like BioCLIP have shown promise in plant identification tasks, their relative novelty, lack of extensive documentation, and limited availability of open-source implementations and pretrained weights made them less practical for this study. In contrast, InceptionV3 provided a balanced trade-off between performance, computational efficiency, and support within the deep learning community, making it a more feasible and robust option under the constraints of this project.
  To leverage the pre-trained weights of the model with the ImageNet dataset [17], the last layer of the model was replaced with a global average pooling layer, followed by a dense layer with ReLU activation, and finally a softmax output layer was added to adapt the model to multiclass classification. The final architecture was trained using the Adam optimizer and freezing some upper layers, which allowed the model to learn complex and determining relationships, enabling it to show good capacity for extracting visual features in the botanical domain. Figure 3 illustrates the complete InceptionV3 architecture used, highlighting the modular structure of the network—including the Inception modules A, B, and C—and the progressive reduction of spatial dimensions through grid reduction layers. The diagram also emphasizes the parallel convolutional paths within each Inception module, which contribute to the model's efficiency and ability to capture multi-scale features.

- **YOLOv11**

  YOLOv11 is a model that belongs to the YOLO family, which has a series of models specialized in object detection with high speed and good accuracy even in real-time systems [18]. YOLOv11 is nothing more than an evolution within this family that incorporates improvements to the

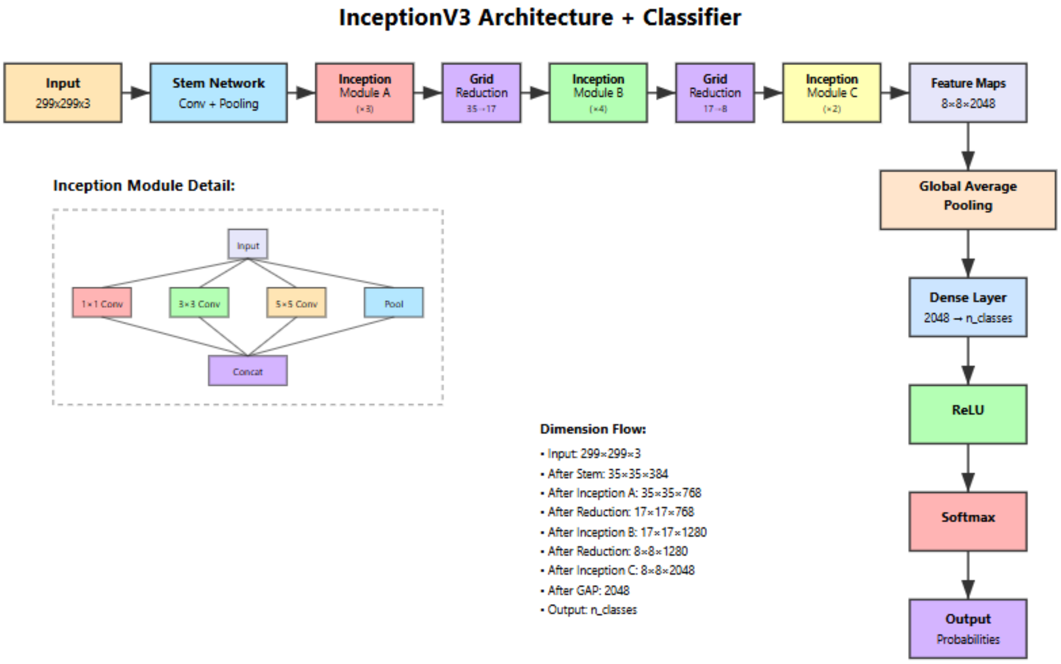

Figure 3: InceptionV3 neural network architecture with custom classifier head. The diagram illustrates the complete flow from image input (299×299×3) through characteristic Inception modules (A, B, C) with intermediate grid reductions, followed by Global Average Pooling (GAP) and a dense layer with ReLU and Softmax activations for classification. Bottom-left detail shows the parallel structure of a typical Inception module with its multiple convolutional branches.

architecture, such as new prediction modules or more efficient training strategies, which enable it to obtain better results [19].
The motivation for using YOLOv11 in this context was twofold. On one hand, it allows automatic detection of regions of interest within images, such as plants, leaves, or flowers. This way non-relevant areas of the image were filtered out from what does not interest us in the image to favor the performance of the classifier model (InceptionV3). On the other hand, this addressed the problem of multiple plants in one image, since we only pass individual images to the classifier model, which facilitates its work by not having to classify more than one image per plant, which could generate confusion and errors. Figure 4 shows an example of what an image would look like after detecting plant species using the YOLO model, which returns the same image with bounding boxes and their corresponding confidence scores added.
In case the YOLO model fails, or does not find any plant, the complete image is passed to the classifier model for classification.

With this combination between detection and classification, a more robust pipeline could be built, where predictions are not based on the complete image, but on key fragments that are identified by the detector. This proved useful when classifying images taken in different natural environments, where the position and framing of photos vary significantly, such as those offered in the PlantCLEF challenge.

4.2. Balancing Techniques

Given such an imbalanced dataset, the following techniques were chosen:

- Random Under-Sampling

  To reduce class imbalance and thus improve training performance, a random under-sampling process was chosen to eliminate those classes that contained an insignificant amount of examples.

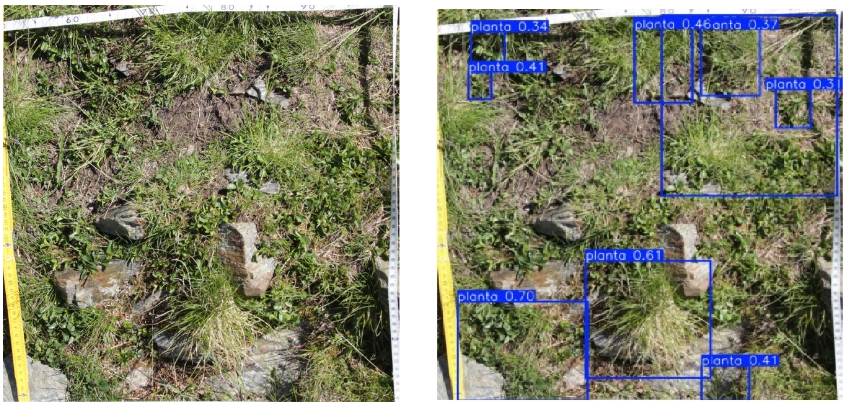

Figure 4: Example of an image before and after being processed by the YOLO model, with bounding boxes added.

Specifically, classes that had only a single data point were eliminated, as they did not provide sufficient information for the model to predict correctly [20], and thus solve the space problem on the equipment that performed the training.
This technique allowed reducing the total number of classes, which in turn improved the overall training perspective, allowing the model to focus on classes with more information. Furthermore, this technique helps prevent data overfitting, while improving training stability [21].

- Data Augmentation

  As mentioned several times before, the dataset presented a clearly imbalanced distribution. To mitigate this problem, data augmentation techniques were applied, which have proven to be really effective in image classification tasks, by generating new ones from transformations of the originals [22]. These transformations included rotations, scaling, random zoom, and horizontal flipping, among others. This way, the training set could be artificially expanded without the need to collect new data, since in this context it was impossible.
  For the practical implementation of this technique, the ImageDataGenerator method offered by the Keras library[1] was used, which allows applying these transformations at runtime during training. This strategy avoids having to store augmented images on the hard drive, reduces memory usage, and allows greater data variability in each epoch.

5. Experimental Setup

For the experimental setup, several Python libraries for machine learning were used. Some of those used were "Keras", "TensorFlow" [23], "Pandas" [24], and "YOLO" [25], among others.

Regarding the data, first the random under-sampling previously described was performed, reducing the classes from 7806 to 4950. After this, a stratified split of the original dataset was applied—allocating 70% of the samples for training, 20% for validation, and 10% for testing—while preserving the original class imbalance. Finally, data augmentation was applied at runtime during training.

On the other hand, to train the YOLO model, a sample of 30 images from the test dataset provided by PlantCLEF had to be taken, which was manually labeled using the "Roboflow"[2] tool. After this, data augmentation was performed, going from 30 to 61 images divided into 52 for training, 6 for validation, and 3 for testing, and the YOLO model was retrained for plant detection in images.

The complete training and implementation notebooks, along with code and documentation, are publicly available at https://github.com/JesusTejon/PlantCLEF-2025-TFG

[1] https://keras.io
[2] https://roboflow.com/

Table 1
Results obtained from the experiments

Models	F1-score	Technique
InceptionV3	0.00088	None
InceptionV3	0.00596	Augmentation
InceptionV3 + YOLOv11	0.02203	Augmentation

6. Results

Regarding performance evaluation, the F1-score has been taken as the main metric, as it offers a balance between precision and recall, which is especially relevant in contexts with imbalanced classes or when it is equally costly to commit false positives and false negatives. Unlike metrics such as precision or recall separately, which focus solely on one of these aspects, the F1-score provides a more complete view of the model's behavior. On the other hand, the use of the confusion matrix has been discarded due to its high dimensionality in this multiclass problem, which would hinder its visual interpretation and detailed analysis.

After experimenting with different configurations, the results shown in Table 1 were obtained.

The results clearly show the progressive improvement in performance as additional techniques are applied to the base model. The use of data augmentation produces a significant improvement compared to direct training with the original dataset, which suggests that the introduced variability contributes positively to the model's generalization capacity.

Likewise, the integration of a prior detection stage with YOLOv11 followed by classification with InceptionV3 notably improves the F1-score, tripling the value obtained with the exclusive use of data augmentation. This demonstrates that first localizing the relevant regions of the image before classifying them helps the model focus on areas with significant information, thus reducing the impact of background noise and improving labeling precision.

Despite the absolute F1-score values still being low, these initial experiments show a clear trend of improvement that validates the usefulness of the proposed hybrid approach. These initial tests laid a promising foundation that these initial tests lay a promising foundation for future iterations of the system, in which both the architecture and the quality of the input data could be further optimized to achieve more competitive performance in scenarios such as those posed by PlantCLEF.

After a manual review, it was observed that the main failure in the system occurs when the YOLO model is unable to detect any plant species or detects an incorrect one, which subsequently causes confusion for the classification model. An example of this can be seen in Figure 5, where the model fails to detect any plant species, likely because it is somewhat camouflaged with the environment. On the other hand, Figure 6 shows a case where the same plant is detected twice, with one of the detections including a rock. This may lead the classification model to classify two different species where there is only one.

As shown in the images, both types of YOLO model errors can also lead to incorrect classification by the InceptionV3 model on the extracted regions.

Additionally, another expected error occurs with the classes that were removed during Random Undersampling, which for obvious reasons cannot be classified by the model.

As shown in Table 2, the top-performing teams achieved F1-scores close to 0.38, while the overall average performance remained low (media = 0.2008), reflecting the difficulty of the task. The I2C-UHU-PERSEUS team obtained an F1-score of 0.02203, ranking 28th out of 38, which, although modest in absolute terms, still positioned the team above 26% of the participants. These results underscore both the technical complexity of the challenge and the value of the proposed approach as a baseline for future improvements.

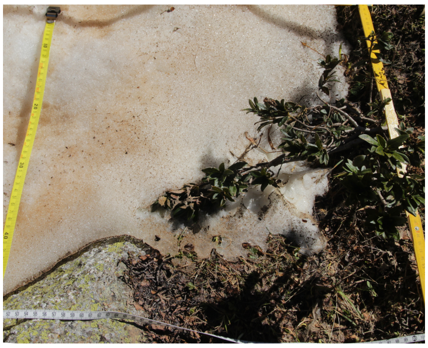

Figure 5: Example of no detection by the YOLO model.

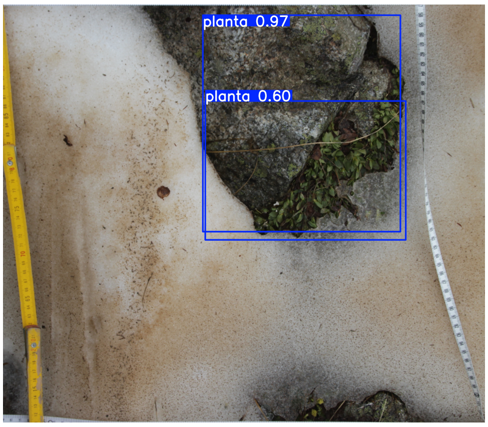

Figure 6: Example of errors in the YOLO model.

Participante	F1-score	Ranking
webmaking	0.38132	1
Chlorophyll Crew	0.37555	2
...	...	...
I2C-UHU-PERSEUS	**0.02203**	**28**
...	...	...
kaggleuser	0.00013	38
media	0.2008	–

Table 2
Official results published by the task organizers.

7. Conclusions

First, the critical importance of having a high-quality dataset that is representative of the real testing environment is evident. The substantial differences between the training and test sets likely negatively affected the model's performance, compromising its generalization capacity. Additionally, the marked imbalance between classes—including a considerable number of categories with only one available sample—represented a significant limitation during the training process.

Regarding the results, while they did not achieve the expected performance, they clearly identify several improvement factors. One of the most relevant lines to explore is the class reduction strategy: eliminating those with few samples may harm the system's ability to identify less frequent, but equally important species. Alternatively, the use of few-shot learning techniques or synthetic sample generation could be considered. Likewise, the adoption of classification models more specialized in the botanical domain, as well as the use of more recent architectures such as Vision Transformers, could provide significant improvements.

In terms of detection, the use of YOLO as a prior detector presents advantages, but also introduces errors that are transferred to the classification system. It would be pertinent to evaluate other finer detections techniques or attention-based ones, with the objective of improving the quality of the extracted regions of interest.

Overall, this work highlights both the challenges and potential of automated plant species recognition in complex environments. Despite the limitations encountered, the proposed pipeline establishes a solid foundation for future research, and offers a clear direction toward more robust, adaptive, and scalable solutions in real contexts such as those posed by PlantCLEF.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 and Claude Sonnet 4 in order to: Grammar and spelling check. Further, the authors used GPT-4 for figures 3 in order to: Generate images. After using these services, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] S. H. Lee, C. S. Chan, S. J. Mayo, P. Remagnino, How deep learning extracts and learns leaf features for plant classification, Pattern recognition 71 (2017) 1–13.

[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[3] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

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

[6] G. Martellucci, H. Goëau, P. Bonnet, F. Vinatier, A. Joly, Overview of PlantCLEF 2025: Multi-species plant identification in vegetation quadrat images, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, 2025.

[7] L. Picek, S. Kahl, H. Goëau, L. Adam, T. Larcher, C. Leblanc, M. Servajean, K. Janoušková, J. Matas, V. Čermák, K. Papafitsoros, R. Planqué, W.-P. Vellinga, H. Klinck, T. Denton, J. S. Cañas, G. Martellucci, F. Vinatier, P. Bonnet, A. Joly, Overview of lifeclef 2025: Challenges on species presence prediction and identification, and individual animal identification, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2025.

[8] A. Koirala, K. B. Walsh, Z. Wang, C. McCarthy, Deep learning–method overview and review of use for fruit detection and yield estimation, Computers and electronics in agriculture 162 (2019) 219–234.

[9] C. Sarkar, D. Gupta, U. Gupta, B. B. Hazarika, Leaf disease detection using machine learning and deep learning: Review and challenges, Applied Soft Computing 145 (2023) 110534.

[10] M. H. Saleem, J. Potgieter, K. M. Arif, Plant disease detection and classification by deep learning, Plants 8 (2019) 468.

[11] S. Ravikumar, I. Eugene Berna, R. Babu, Y. Arockia Raj, K. Vijay, Detection of medicinal plants using machine learning, in: International Conference on Recent Trends in Computing, Springer, 2024, pp. 199–208.

[12] U. Barman, P. Sarma, M. Rahman, V. Deka, S. Lahkar, V. Sharma, M. J. Saikia, Vit-smartagri: vision transformer and smartphone-based plant disease detection for smart agriculture, Agronomy 14 (2024) 327.

[13] S. Chulif, Y. L. Chang, S. H. Lee, Deep learning for large-scale plant classification: Neuon submission to plantclef 2023., in: CLEF (Working Notes), 2023, pp. 2035–2042.

[14] H. Goëau, V. Espitalier, P. Bonnet, A. Joly, Overview of plantclef 2024: multi-species plant identification in vegetation plot images, CEUR-WS, 2024.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[16] X. Xia, C. Xu, B. Nan, Inception-v3 for flower classification, in: 2017 2nd international conference on image, vision and computing (ICIVC), IEEE, 2017, pp. 783–787.

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

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

[19] G. Jocher, J. Qiu, Ultralytics yolo11, GitHub: https://github.com/ultralytics/ultralytics (2024). URL: https://github.com/ultralytics/ultralytics.

[20] M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural networks 106 (2018) 249–259.

[21] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on knowledge and data engineering 21 (2009) 1263–1284.

[22] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of big data 6 (2019) 1–48.

[23] F. J. J. Joseph, S. Nonsiri, A. Monsakul, Keras and tensorflow: A hands-on experience, Advanced deep learning for engineers and scientists: A practical approach (2021) 85–111.

[24] W. McKinney, et al., pandas: a foundational python library for data analysis and statistics, Python for high performance and scientific computing 14 (2011) 1–9.

[25] P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma, A review of yolo algorithm developments, Procedia computer science 199 (2022) 1066–1073.