montanha viva

Intelligent Predictive System for Decision Support in Sustainability

T4.2. Especificação dos algoritmos de IA, da plataforma de dados de entrada e indicadores de apoio à decisão

Specification of AI algorithms, input data platform, and decision support indicators

junho de 2023

Fundação para a Ciência e a Tecnologia

BPI | Fundação "la Caixa"

Contents

Contents 2

Executive Summary 2

1. Introduction 4

2. Objectives 4

3. Project Application 4

4. Project Description 4

5. Image Classification 5

5.1. Overview of image classification models 6

6. Image Segmentation 11

6.1. Overview of image segmentation methods 12

7. Overview of growth rate and plant vigor methods 16

7.1. Plant vigor, detection, and identification methods 16

8. Datasets 18

8.1. Plant identification 18

8.2. Not-exclusively plant datasets 22

9. Conclusions and final remarks 26

Bibliographic References 27

Executive Summary

The Montanha Viva Project aims to develop a decision support system, which is intelligent and with real-time operation in the economic exploitation of mountain plants, especially in remote locations (without internet connection), to stimulate the economical use of existing plants, increase production, reduce consumption of natural resources, contributing to the promotion of biodiversity and preservation of environmental sustainability, particularly of wild mountain plants. It will start from the identification and characterization of mountain plants with potentiating characteristics for natural mitigation of pests and diseases in agricultural crops and with application of this properties in health and well-being, to the creation of a local and remote sensing system for the analysis of plant vigor allied to artificial intelligence algorithms for decision support in carrying out cultural activities in existing plants or in new agroforestry exploitations.

Its objectives are:

- Collect basic information and produce data of the identification and characterization of mountain plants with application properties in health and welfare and with potentiating characteristics of natural mitigation of pests and diseases in agricultural crops in the mountain region of Serra da Gardunha, promoting the sustainability of existing agroforestry farms and the development of new products and new businesses from the use of wild flora.

- To evaluate and characterize the biological properties of selected species based on the collection of information from ethnobotanical surveys.

- Adapt existing technological solutions and/or develop specific solutions for local monitoring in remote (without access to electricity sources or communications) and inhospitable areas (with very high thermo-hygrometric gradients).

- Analyzing the potential of high-resolution remote sensing for near-real-time determination of plant vigor and growth rate.

- Develop an intelligent predictive system of mountain plant vigor, information and decision support in environmental sustainability in order to optimize the cultivation/exploitation of wild plants in the mountain region.

- Promote sustainable awareness, through the installation of interpretative tables and digital information with identification and dissemination of the environmental value, landscape and heritage of flora that aim to raise awareness and planning of visitation to mountain areas.

- Dynamize tourist trails for the promotion of mountain sustainability by raising awareness of local biodiversity.

- Communicate, disseminate, transfer data and technology and disseminate project results.

This document aims to evaluate state-of-art AI algorithms for analyzing plant vigor and growth rate.

Keywords: Convolutional Neural Networks, Segmentation, Plant Identification, Wild flowers.

1. Introduction

In recent years, the gap between agriculture and technology has narrowed with the development of Agriculture 4.0 that uses tools such as the Internet of Things, robotics and artificial Intelligence to optimize various agricultural processes. The scientific bibliography describes numerous solutions with the use of this resources that allow better decisions to be made regarding irrigation and weed management, and better and faster analysis of soil and plant health. This subsection examines the development of technologies that make it possible to monitor not only the phenology of plants but also their growth. It should be noted that these technologies will work on a microcontroller, which has limitations in terms of processing and performance. Therefore, we will also discuss some solutions for such cases.

2. Objectives

This work can be divided into two parts: Remote sensing to classify and monitor wild plants using sensors and cameras in the field. Each part plays an important role in the success of this project. On the one hand, we will use remote sensing to evaluate vast areas of land in the Serra da Gardunha to classify the native wild plants in this area. On the other hand, we will use cameras and sensors on the ground to monitor plant phenology and soil properties to obtain data on the health and growth of wild plants. This paper will focus on the latest.

3. Project Application

The use of classification and segmentation is intended to develop a system that can quickly detect and identify plants. This process is possible after the data has been collected using a data collection station. Once the system is in possession of the data, it will use segmentation pre-trained segmentation models, to separate the background of the presumed plant and obtain a segmented image. Once the plant is detected, a classification model will be used to determine the phenology of the plant, such as whether the plant has flowers or fruits, and which specie is it.

4. Project Description

Initially, the main goal is to gather information in order to have a robust algorithm capable to provide the information needed for the next phase of this project. This will be accomplished using a data collection station. This station will be equipped with the necessary hardware to collect the required data. This data will be used to create a dataset that will be useful for future studies of this nature and, more importantly, for the project. Then the data will go through several processes to extract useful information. These processes will be mainly related to image classification and segmentation. Using this technique, it is possible to extract the pixels that contain the plant in question and, remove unnecessary information such as the background. After separating the data between the plant and the background, we will be able to identify the phenology of the plant and its stages, such as whether it is flowering and/or bearing fruits. With this information, it is possible to present which regions

worthwhile to visit, and so develop a dynamic application that offers users with up-to-date information about the species in question.

5. Image Classification

The Classification method, a Supervised Learning approach, is used to categorize fresh observations with the help of training data. In classification, a computer learns from the dataset or observations provided before classifying fresh observations into various classes or groups.

Convolutional Neural Network (CNN) and related models are the most often used Deep Learning model in image classification. Adopting cutting-edge methods like attention mechanisms, new lightweight models, and single-stage detection models can improve the model's performance since even a little improvement in runtime and accuracy can produce observably improved results.

At the core of a CNN are convolutional layers, which apply mathematical operations known as convolutions to input data. Convolution involves sliding a small window, called a filter or kernel, over the input data and performing element-wise multiplications and sums. This process captures local patterns and detects features such as edges, corners, and textures.

By using convolutions, CNNs can effectively process inputs with many parameters while maintaining parameter sharing. This characteristic allows them to learn hierarchical representations of visual features, starting from simple features at lower layers to more complex and abstract features at higher layers.

Another important component of CNNs is pooling, which reduces the spatial dimensions of the data while preserving essential information. Pooling layers aggregate neighbouring features and down sample them, making the network more robust to variations in the input and reducing the computational requirements.

Typically, CNNs also include fully connected layers towards the end of the network, which perform high-level reasoning and decision-making based on the learned features. These layers connect all the neurons in the previous layer to the subsequent layer, enabling the network to learn complex relationships and make predictions.

Training a CNN involves optimizing its parameters using a large, labelled dataset through a process called backpropagation. This technique calculates the gradients of the network's parameters with respect to a loss function and updates them iteratively using optimization algorithms such as stochastic gradient descent.

With their ability to automatically learn relevant features and hierarchical representations, CNNs have achieved remarkable performance in tasks such as image classification, object detection, image segmentation, and more. They have become an indispensable tool for solving challenging problems in the realm of computer vision and continue to push the boundaries of what machines can achieve in understanding visual data.

When it comes to flora classification, Convolutional Neural Networks (CNNs) can play a significant role. By leveraging the power of CNNs, we can automatically learn and extract relevant features from plant images, allowing for accurate and efficient classification of different plant species.

To train a CNN for flora classification, a large dataset of labelled plant images is required. By utilizing this dataset and employing techniques like backpropagation and optimization algorithms, CNN can learn to accurately classify plant species based on the features it extracts from the images.

The application of CNNs in flora classification has the potential to revolutionize fields such as botany, agriculture, and conservation. It can speed up identifying and cataloging plant species, aid in biodiversity research, and contribute to the development of automated plant recognition systems. Ultimately, CNNs provide a powerful tool that enables us to leverage the capabilities of deep learning to enhance our understanding and conservation efforts of the plant kingdom.

5.1. Overview of image classification models

CNN models for image classification have made significant advancements in recent years. Here are some notable CNN architectures commonly used, and that will be studied to evaluate the best, for image classification tasks:

- **AlexNet**

Eight layers make up AlexNet, comprising three fully linked layers and five convolutional layers. Following the convolutional layers are the max-pooling layers, which reduce the spatial dimensionality by downsampling the feature maps. Rectified linear units (ReLU) served as the network's activation function. ReLU adds non-linearity and aids in solving the vanishing gradient issue, making it possible to train deeper networks effectively. Additionally, a normalization method known as Local Response Normalisation (LRN) was used following ReLU activation. By encouraging competition between nearby neurons, LRN improves the network's capacity to generalize and adapt to changes in input. AlexNet uses overlapping max pooling, which is different from conventional pooling methods. This method created overlapping areas by doing pooling operations with a stride that was less than the pool size, hence minimizing the loss of spatial information. Dropout regularization was used by AlexNet in order to avoid overfitting. During training, dropout randomly removes a portion of the neurons, driving the network to learn more durable and universal properties (Iandola et al., 2016).

- **ConvNeXt**

ConvNeXt uses group convolutions, a method that divides input channels into various groups and applies different filters to each group. As a result, the network can simultaneously collect several feature types, which encourages efficient feature learning. Path aggregation, which includes merging data from many convolutional pathways, is the main emphasis of the ConvNeXt architecture. By using a concatenation process, ConvNeXt combines features picked up by many groups, allowing the network to gather a variety of complementary data. It introduces the idea of cardinality, which describes how many groups there are in group convolutions. The network may regulate the complexity and capacity of the model using the cardinality parameter. It strikes a compromise between the overall network's computational efficiency and the trade-off between the representational strength of different groups. Similar to the well-known ResNet design, ConvNeXt has a bottleneck structure. A series of convolutions with dimensions of 1x1, 3x3, and 1x1 make up the

bottleneck structure. This architecture enables the network to successfully capture both low-level and high-level information while reducing the number of parameters and computational cost. It also offers many variations with various depths and capacities (Simonyan & Zisserman, 2015).

- **DenseNet**

By establishing feed-forward connections between every layer and every other layer, DenseNet creates dense connectivity. This indicates that all levels before it directly feeds data to the layer above it. This connection structure improves gradient flow and mitigates the vanishing gradient issue by facilitating feature reuse and allowing information to travel over shorter channels. It is made up of dense blocks, which represent a network's stack of layers. A dense block's layers are all linked to each other, enabling the concatenation of features. By mixing local and global information, dense blocks promote feature reuse and help the network learn more discriminative features. Between dense blocks, transition layers are added to limit the number of feature maps and spatial dimensions. 1x1 convolutional layers are frequently used as transition layers, which are then followed by average pooling. They decrease the number of feature maps, which helps to compress the data and improve computing performance. A growth rate hyperparameter introduced by DenseNet controls how many feature mappings are added to each layer inside a dense block. The growth rate controls the network's capacity and complexity, enabling a variable trade-off between model size and performance. Batch normalisation and rectified linear unit (ReLU) activation functions, which are incorporated into DenseNet, increase the network's stability and nonlinearity. Each layer's input is normalised by batch normalisation, which decreases the internal covariate shift and enhances training efficiency. In transition layers, it also introduces a compression factor hyperparameter. By lowering the amount of feature maps by a certain ratio, the compression factor lowers computing complexity while maintaining crucial data. It enables model size control while retaining computational efficiency (Ma et al., 2018).

- **EfficientNet**

The compound scaling mechanism used by EfficientNet scales the network's depth, breadth, and resolution consistently. The scaling of these dimensions is controlled by a compound coefficient. EfficientNet strikes a balance between model capacity and computing efficiency by scaling all facets of the network. The network's breadth and depth are measured in terms of the number of channels or filters present in each layer, respectively. By scaling these dimensions appropriately, EfficientNet permits more intricate representations while keeping computation costs under control. It adjusts the resolution of the input picture, allowing the network to gather more minute features. Improved accuracy may result from higher resolution input, but this comes at a higher processing cost. The compound scaling approach strikes the ideal balance between computing speed and resolution. Building blocks for MobileNetV2 that are depth-wise separable convolutions and inverted residual blocks are adjusted for use in EfficientNet. These building components effectively capture geographical and channel-wise information and are computationally efficient. The channel-wise feature responses in this model are adaptively recalibrated using SE blocks. SE blocks enhance valuable traits while suppressing unhelpful ones, increasing the strength of representation. To

determine the ideal scaling coefficients for depth, breadth, and resolution, it uses an AutoML-based methodology (Radosavovic et al., 2020).

- **GoogLeNet**

The GoogLeNet architecture introduced inception modules to capture multi-scale characteristics. These modules consist of parallel convolutional processes with filter sizes of 1x1, 3x3, and 5x5, along with a max-pooling layer. By using multiple filter sizes, the network can gather local and global information effectively. To reduce computational cost and parameters, 1x1 convolutional layers were employed for dimensionality reduction. These layers decreased the number of input channels before larger filter sizes, enabling efficient processing in subsequent layers. Auxiliary classifiers were added to address the vanishing gradient problem during training. These classifiers, consisting of fully connected and 1x1 convolutional layers, provided additional gradients and encouraged the learning of discriminative features. Instead of fully connected layers, GoogLeNet employed global average pooling at the network's end. This technique computed the average value of each feature map across spatial dimensions, resulting in a fixed-size feature vector for classification and reduced overfitting. GoogLeNet had 22 layers and used a higher number of filters per layer (Simonyan & Zisserman, 2015).

- **Inception V3**

Inception V3, inspired by GoogleNet, introduces the concept of inception modules featuring dimensionality reduction and parallel convolutional operations. These modules employ 1x1 convolutions and filters of various sizes (1x1, 3x3, and 5x5) to capture local and global information effectively. Factorization techniques replace the 5x5 convolution with two consecutive 3x3 convolutions, reducing parameters while preserving the receptive field. Batch normalization is incorporated to normalize inputs and improve convergence, gradient flow, and network stability. Reduction blocks are added between inception modules to downscale spatial dimensions using 1x1 convolutions and max pooling, increasing channel capacity. Auxiliary classifiers are used at intermediate levels during training to address the vanishing gradient problem, provide regularization, and enhance gradient flow. Inception V3 combines these features to create a powerful and efficient convolutional neural network architecture (Szegedy et al., 2015).

- **MNASNet**

MNASNet is a neural network architecture that utilizes neural architecture search (NAS) to discover optimal network configurations. It focuses on reducing model size and improving computational efficiency for deployment on resource-constrained devices. MNASNet employs depth-wise separable convolutions, which reduce complexity while maintaining spatial and channel relationships. The design incorporates mobile-specific constraints for optimized mobile deployment. The neural architecture search process explores a wide range of architectures to identify high-performing configurations. MNASNet aims to provide compact and efficient neural networks suitable for mobile devices (Tan et al., 2019).

- **MobileNet**

A neural network design called MobileNet makes use of depth-wise separable convolutions, which are made up of a depthwise convolution and a point-wise convolution. This factorization approach maintains spatial and channel interdependence while reducing computational cost and parameter count. It is appropriate for devices with limited resources since it seeks to reduce model parameters while keeping adequate accuracy. In order to manage the number of channels and layers, respectively, MobileNet offers a width multiplier and a depth multiplier, permitting trade-offs between model size and accuracy. Pre-trained models developed using transfer learning and fine-tuning on massive datasets like ImageNet are offered, allowing for quicker development and deployment (Ma et al., 2018).

- **RegNet**

RegNet is a neural network architecture that employs a systematic approach to network design. It explores a large design space of architectures while adhering to architectural design principles to guide the search for high-performing models. It incorporates compound scaling, which uniformly scales the depth, width, and resolution of the network, allowing for easy adjustment of model capacity and computational cost. RegNet introduces an adaptive scaling rule to find the optimal balance between model size and accuracy based on desired constraints. It emphasizes network width and depth, varying the number of channels and layers to explore different architectural configurations. To improve generalization and prevent overfitting, RegNet employs regularization techniques such as weight decay, dropout, and stochastic depth. It can be trained using standard deep learning optimization techniques like stochastic gradient descent (SGD) with momentum, benefiting from learning rate schedules, data augmentation, and batch normalization. Overall, RegNet provides a framework for designing high-performing models with efficient computational characteristics (Radosavovic et al., 2020).

- **ResNet**

ResNet is a deep neural network architecture that addresses the vanishing gradients problem in deep networks. It introduces skip connections or shortcuts that allow gradients to flow easily during backpropagation. The fundamental building block, called a residual block, consists of convolutional layers with a skip connection that adds the original input to the output. This enables the network to learn residual mappings and focus on the residual error. ResNet is renowned for its ability to build extremely deep networks by stacking residual blocks, allowing for hundreds or thousands of layers. Deeper networks can learn more complex features, leading to improved accuracy. ResNet also introduced pre-activation, where batch normalization and activation functions are applied before convolutional layers, aiding optimization. To reduce computational complexity, ResNet incorporates bottleneck structures in some residual blocks, using 1x1 and 3x3 convolutions to decrease and then increase the dimensionality. This approach reduces parameters and computations (Boesch, 2023).

- **ShuffleNet V2**

ShuffleNet V2 is a convolutional neural network architecture that incorporates a channel shuffle operation to enable information exchange between channels. This operation is performed within Shuffle units, which consist of pointwise and depthwise convolutions. The channel shuffle promotes information flow and enhances the network's representational power. To reduce computational cost, ShuffleNet V2 employs group convolutions in the depthwise convolution, reducing parameters and computation. It also employs a feature map alignment technique called channel split and concat to preserve information during downsampling. The architecture is designed hierarchically, capturing features at different abstraction levels, and uses different block configurations to balance accuracy and efficiency (Ma et al., 2018).

- **SqueezeNet**

SqueezeNet is a compact neural network architecture that achieves high performance with fewer parameters. It uses 1x1 convolutional filters to reduce the number of input channels, reducing computational complexity. The model incorporates skip connections to improve gradient flow and address the vanishing gradient problem. SqueezeNet also utilizes aggressive down-sampling and efficient fire modules to further reduce parameters. It strikes a balance between model size and accuracy, making it suitable for resource-constrained devices (Iandola et al., 2016).

- **SwinTransformer**

SwinTransformer is a cutting-edge computer vision model that combines the power of Transformers and convolutional neural networks (CNNs) to excel in various visual tasks. It introduces a hierarchical design with shifted windows, enabling efficient processing of image patches. By using shifted windows instead of traditional self-attention, SwinTransformer captures global and local information effectively. The model consists of multiple stages, each containing a hierarchical transformer encoder. This design allows the model to capture information at different scales and resolutions. SwinTransformer incorporates a shifted window self-attention mechanism within each stage, facilitating parallel processing and reducing computational complexity. To incorporate positional information, SwinTransformer utilizes window-based position embedding, encoding spatial relationships at the patch level. This ensures accurate modeling of object locations and improves overall performance. Furthermore, SwinTransformer employs a layer-wise feature map shifting strategy to enhance information propagation and gradient flow across different network layers. This facilitates efficient learning and optimization. SwinTransformer has demonstrated exceptional performance in tasks such as image classification and object detection. It strikes a balance between accuracy and computational efficiency, making it suitable for various computing systems and devices with different resource constraints (Liu et al., 2021).

- **VGG**

VGG (Visual Geometry Group) is a popular convolutional neural network (CNN) architecture known for its simplicity and effectiveness in image classification. VGG employs a deep architecture with multiple convolutional layers and max-pooling layers. The key idea is to stack small-sized filters

(typically 3x3) to increase the network's depth and capture complex features. VGG's architecture follows a consistent pattern, with 3x3 filters and 2x2 max-pooling layers. This regularity simplifies implementation and training. Although VGG is primarily known for image classification, its pre-trained models have also been used for other computer vision tasks, including object detection and segmentation (Simonyan & Zisserman, 2015).

- **VisionTransformer**

The Vision Transformer (ViT) is a deep learning model that applies the transformer architecture to computer vision tasks. It treats images as sequences of patches and uses a transformer encoder to capture global dependencies among these patches. ViT replaces traditional convolutional layers with self-attention layers and feed-forward networks to process the patch embeddings. It has achieved impressive results in image classification, object detection, and image segmentation tasks. However, to handle high-resolution images, a hybrid approach called Hybrid Vision Transformer (HViT) combines the Vision Transformer with a convolutional neural network (CNN) stem for local feature extraction (Dosovitskiy et al., 2021).

6. Image Segmentation

The main goal of computer vision is to identify objects on an image. There are several methods to accomplish this but the far most popular is segmentation or image segmentation. Segmentation is a method that aims to isolate or detect objects within a picture. A cluster of pixels belonging to the same class is called a segment. With a segmented image, pixels can be thought of as class labels rather than real pixel values (Barreto, 2022) . It's useful to divide objects in two categories: Stuff and Things. Things refers to objects that are properly bounded and can be counted, such as humans. Stuff is all of things that are not geometrically defined but are identified by the material, such as the sky (Barla, 2023). We can define three types of segmentation: Instance Segmentation, Semantic Segmentation and Panoptic Segmentation.

Instance Segmentation defines a unique label to each detected object within the picture, allowing objects to be counted and thus studying things. On the other hand, in Semantic Segmentation a label is shared if the objects belong to the same class, which enables to look at the stuff. Lastly, Panoptic Segmentation is a hybrid between instance segmentation and semantic segmentation. Therefore, this method not only recognizes the items according to their class labels but also all the instances present in the image. Which enables a study of both stuff and things. The distinction between these three segmentation methods can be seen in Figure 12.

Additionally, there are two classes of segmentation raising popularity, which each of the methods fall, interactive segmentation and automatic segmentation. In interactive segmentation, it is required a person to refine the resulting mask of any class of object by clicking in it, in order to guide the deep neural network, where the input is used as supervised information (X. Chen et al., 2022). As for automatic segmentation is a fully automated process that relies only in algorithms to do the

partition, thus categorizing specific objects ahead of time. Nevertheless, it still requires a substantial amount of fully annotated data to train the segmentation model.

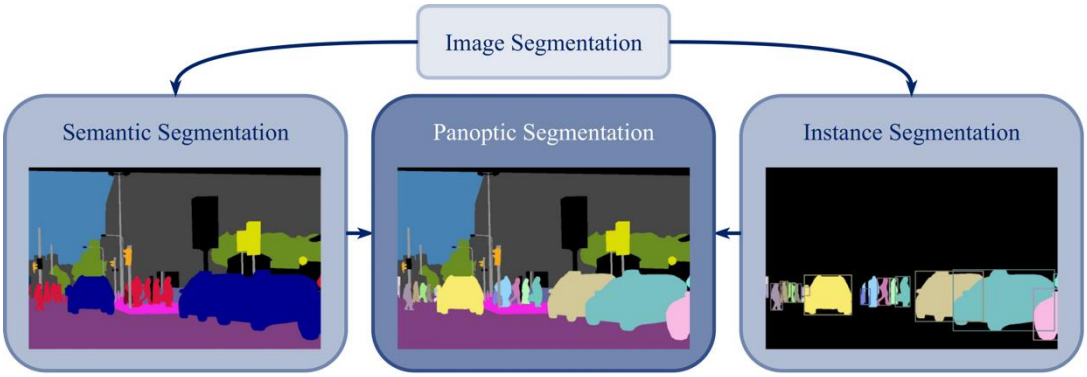

Figure 12: Representation of three different methods of segmentation. (*Left*) Semantic Segmentation: labels equally the same object class. (*Right*) Instance Segmentation: attributes unique labels to each object. (Center) Panoptic Segmentation: recognizes instances while at the same time recognizing the items according to their class.

6.1. Overview of image segmentation methods

When working with instance segmentation is common to use color information almost exclusively. (C.-Y. Wu et al., 2020) executes instance segmentation by fusing images and disparity information to regress object masks. This is achieved by using stereo cameras to provide geometric estimations and by introducing disparity information at ROI (Region of interest method) heads. GAIS-Net model was tested on a self-compile *HDQS* testing set and on *Cityscapes* dataset. On the *HDQS*, GAIS-Net outperformed other models on the bounding box evaluation (46.0% AP) and mask evaluation (40.7% AP), reaching a state-of-art performance. As for the *Cityscapes*, the comparison was only with Mask-RCNN where GAIS-net still had a better performance on mask evaluation, 32.5% AP with only the fine annotations and 37.1% AP with fine annotations and pre-training with the COCO dataset.

(Liang et al., 2021)suggest an approach that takes the segmentation network's masks and turns them into a collection of polygons that the deforming network can better fit on the object's boundaries. The solution was tested on the *Cityscapes* reaching 40.1% AP accuracy outperforming PANet (S. Liu et al., 2018) (36.4% AP). Additionally, this technique also demonstrates a 35% speed increase in annotating. It performs better than the boundary metric by 2.0% when compared to earlier work on annotation-in-the-loop.

Proposal-based segmentation use the detect and segment method, where objects are first detect using bounding box approach and then created a binary mask, which has still low resolution and is ineffective for real-time applications. Proposal-free instance segmentation has improved binary mask resolution and report faster than proposed-based ones, due the fact that are hinged on embedding loss functions, pixel affinity and dense prediction networks. Tackling the problem of maintaining a good segmentation accuracy in real-time applications, (Neven et al., 2019) proposes a new clustering loss function for proposal-free instance segmentation. By grouping together, the spatial embeddings of pixels from the same instance, the loss function maximizes the intersection-over-union of the resulting instance mask and concurrently learns an instance-specific clustering

bandwidth. The network can carry out instance segmentation in real-time while maintaining high accuracy when combined with a quick architecture. To evaluate the method, it was used the ERFNet (Romera et al., 2018) network architecture as base network and tested it on *Cityscapes* instance segmentation challenge. Comparing with fine only methods, it achieved an AP-score of 27.6 which is better than the PANet (S. Liu et al., 2018) method (36.4%). For the person (34.5 vs 30.5), rider (26.1 vs 23.7) and car class (52.4 vs 46.9) it performs better than Mask R-CNN (He et al., 2017) which holds the first place. Regarding execution speed, the ERFNet method, achieved the better trade-of of execution speed (11 fps) with accuracy (27.6 AP), where Mask R-CNN only achieved 26.2 AP with a execution time of 2.2 fps. As for PANet, it has a better accuracy (31.8 AP) but its execution time is one of the slowest with less than 1 fps.

(Yuan et al., 2021) tackle the context aggregation problem in semantic segmentation by developing a three-phase method to describe a pixel using the representation of the analogous object class. To learn object areas, supervised learning of the ground truth segmentation is used first. Next, after totaling the representations of the pixels located within the region, the value is computed. The relationship between each pixel and each object region is then calculated, and each pixel's representation is then enhanced with the object-contextual representation, which is a weighted aggregation of all the representations of the object regions. The Object-Contextual Representation (OCR) method was tested in several datasets such as *Cityscapes*, *ADE20K*, *Pascal-context* and *COCO-stuff*, and used the dilated ResNet-101 as backbone to conduce the experiment. In each test it outperforms the baseline models and on *Cityscapes* test reach a performance of 81.8% which is comparable with some methods bases on advanced baselines, such as DANet (Fu et al., 2019), ACFNet (Zhang et al., 2019).

Nowadays, in semantic segmentation is common to use a two-branch network architecture because of the improvement on its efficiency especially on real-time tasks. However, this type of architecture suffers from an event which the small-scale object is overwhelmed by its adjacent bigger objects. (Xu et al., 2022) names this event as overshoot and compares it to the overshoot on a PID controller. Regarding this comparison, it was proposed a three-branch network architecture, named Proportional-Integral-Derivative Network or PIDNet to mitigate this issue. In order to create an architecture that impersonate a PID controller in spatial domain they add a third branch called Auxiliary Derivative Branch (ADB). Therefore, the model works with a Proportional Branch which parses and saves the detailed information in its high-resolution feature maps, an Integral (I) branch parses long-range relationships by combining local and global context information and to forecast the border areas, the derivative (D) branch extracts the high-frequency information. An important note is that this architecture requires a precise annotation around boundary to perform at its best. The PIDNet family was tested on *Cityscapes*, *CamVid* and *COCO-stuff*. On the *CamVid* test set the PIDNet with and accuracy of 82.0% is only comparable to the DDRNet-23 (Hong et al., 2021) which is fastest but with an accuracy of 80.6%. As for the *Cityscapes* test set it was proved that PIDNet shows the best trade-off between inference speed and accuracy, achieving an accuracy of 80.4% and 80.6% in real-time. On the *COCO-stuff* the model reaches a good performance in comparison with other models of the same type, reaching an accuracy of 33.5%.

Regarding the efficiency in real-time applications, (Peng et al., 2022) suggests a lightweight model, PP-LiteSeg, which is constitute of three models: encoder, aggregation and decoder. The innovation is found on the decoder block where they integrate three new modules. The Flexible and Lightweight Decoder (FLD) is used to increase feature spatial size and reduce the channels in gradually manner, which can be adjusted depending on the decoder. For efficiently strengthening feature representation they propose a Unified Attention Fusion Module (UAFM) which makes use of channel and spatial attention. Lastly it is integrated a Simple Pyramid Pooling Module (SPPM) to aggregate global context and increase segmentation accuracy. PP-LiteSeg was tested on *Cityscapes* and *CamVid* test sets. PP-LiteSeg-B (which uses a STDC1 decoder) achieved an accuracy of 77.5% on the *Cityscapes* test set. As for the *CamVid* test set, PP-LiteSeg-B had 75.0% performance. It is important to notice that PP-LiteSeg-T which has the STDC2 encoder, while not presenting a state-of-art accuracy (thou still competitive), its inference speed on real-time tasks clearly outperforms other state-of-art models. Hence, this proposed model accomplished a state-of-art trade-off between inference speed and accuracy.

A redefiniton of atrous convolution, a commonly used tool on semantic segmentation was presented by (L.-C. Chen et al., 2017). Astrous convolution is a solution found for the problem of spatial resolution reduction of the feature maps that is also very computational efficient. (L.-C. Chen et al., 2017) introduce astrous convolution in modules (like the Atrous Spatial Pyramid Pooling module) with several atrous rates that are in cascade or in parallel to capture multi-scale context. The DeepLabv3 was tested in *PASCAL VOC 2012* set where it achieved an 85.7% accuracy. Additionally, the method was tested with the implementation on the ResNet-101 model where it accomplishes a performance of 86.9%. Furthermore, DeepLAb3 was also tested on the *Cityscapes* test set where the method achieved an 81.3% accuracy, which is comparable to other state of art methods such as PSPNet (Zhao et al., 2017).

Semantic Segmentation accuracy improves when are obtained high resolution feature maps with either astrous convolution and feature pyramid fusion. The methods used to this end are either ineffective or computationally intensive. Therefore (X. Li et al., 2021a) developed the Flow Alignment Module (FAM) that tackles the issue of misalignment of high-level feature maps fusion with low level feature maps. This problem requires explicit and dynamic position correspondence which the Flow Alignment Module provides. (X. Li et al., 2021b) defined a flow field, called Semantic Flow, based on the idea of the alignment of two adjacent video frames features in the video processing task. The Semantic Flow takes the distinct levels of the feature pyramid and creates a flow of adjacent level feature maps, allowing a more flexible fusion and refining the low-level features in the semantic representation. The FAM module was proved on the *Cityscapes* dataset, *CamVid*, *ADE20K* and *Pascal-context*. Using ResNet-18 (He et al., 2015) as backbone, the method achieved a 78.9% mIoU and 80.4% with 26 FPS using Mapillary Vistas (Neuhold et al., 2017) dataset for pretraining on the *Cityscapes* test set. Also in this test set, it was made an evaluation while performing multi-scale and horizontal flip inference which achieved 81.8% mIoU surpassing the DAnet (Fu et al., 2018) model by 0.3% requiring only 30% of computation. As for the *ADE20K* and *Pascal-context*, this model outperforms the state-of-art models while using less computation. Furthermore for *CamVid* using DF2 (X. Li et al., 2019) as backbone it improves the baseline by 3.2% mIoU, reaching 70.4% mIoU and using ResNet-18 achieves and accuracy of 73.8% mIoU.

Much of the suggested models for semantic segmentation use as backbone the ImageNet pre-trained, which doesn't have a wide field-of-view, so normally is introduced a special contextual module to mitigate this issue. (Gao, 2021), takes an alternative approach by designing a backbone specialize in semantic segmentation called RegSeg. By introducing a D block which is a dilated block structure and keeping the number of channels low, they can increase the field-of-view without losing local detail of the picture. The innovation is that D block uses group convolutions, and each group uses different dilation rates in order to extract multi-scale features. Testing this solution on the *CamVid* test set it achieved an accuracy of 80.9% with 70 FPS outperforming other models. For the *Cityscapes* test set it got a performance of 78.13% with 30 FPS being only excelled by DDRNet-23 (Hong et al., 2021) which uses as a backbone ImageNet pretraining that has more parameters.

A new bottom-up system for panoptic segmentation is proposed by (Cheng et al., 2020). The goal is to accomplish comparable results with top-down approaches while increasing the inference speed. The Panoptic-DeepLab has the particulatiry of having a dual-ASPP commonly used on semantic segmentation and a dual-decoder used on instance segmentation. While the first branch possesses a regular design, the instance branch holds a class-agnostic detector with an instance center regression. Something deserving of mentioning is the fact that this architecture during training only uses three loss functions and adds marginal parameters to semantic segmentation. The model was tested on *Cityscapes*, *Mapillary Vistas* and *COCO* test sets. On the *Cityscapes* test set, Panoptic-Deeplab achieved a performance of 84.2%. As for the *Mapillary Vistas* test set, it accomplished the best of panoptic quality (PQ) with 42.7% compared to the winners of 2018. Finally for the *COCO* test set, it performed comparability well with other top-down methods which use heavier backbones or deformable convolution.

There have been recent breakthroughs in the field of computer vision that came dismantle the CNN paradigm. (Z. Chen et al., 2023) developed an adaption of a vision-specific transformer (ViT) to try to solve some of the performance gap that this architecture still holds when applied to computer vision. The innovation was to introduce a vision-specific inductive biases, thus avoiding modifying the original architecture. These biases are achieved by three modules: a spatial prior module for capturing the local semantics from input images, a spatial feature injector for incorporating spatial prior into the ViT and a multi-scale feature extractor to reconstruct the multi-scale features required by dense predictions tasks. This implementation was tested for the ViT-T/S/B models with the use of ImageNet-1K pre-training and with ImageNet-22K weights for the ViT-L model. By evaluating this implementation on the *COCO* dataset with the use of MMDetection on object detection and instance segmentation, it achieved better performance than the other two similar approaches ViT (Y. Li et al., 2021) and ViTDet (Y. Li et al., 2022) and has comparable results to recent vision-specific models. The Vit-L model also outperforms the other two approaches with the same initialization. Another conclusion taken was that ViT adapter gains precision using multi-model pre-training instead of using different pre-trained weights. As for semantic segmentation, following the same logic as before, ViT-adapter outperforms the ViT (Y. Li et al., 2021) and other vision-specific transformers with the use of ImageNet-1k pre-training. And for ImageNet-22k pre-trained weights, ViT-Adapter proved to be more fitting for semantic segmentation than the Swin Transformer (Z. Liu et al., 2021), due the improvements over different model sizes. Again Vit-Adapter benefits from multi-model pre-training.

Another adaption for ViT specifically on the semantic segmentation scope is the Lawin Transformer

(Yan et al., 2022). Which implements a multi-scale representation through a window attention mechanism to increase the efficiency of the ViT architecture and at the same time decrease the computation cost that this strategy which holds. Hence, they developed a decoder that applies a large window attention, which enables the local window to query a wider range of the context window with a negligible processing overhead. Thus, enabling capture of contextual information at multiple scales. Furthermore, it is added an efficient hierarchical vision transformer (HVT) below the Lawin Transformer, which enables the multi-scale representations into the semantic segmentation ViT. This strategy was tested on *Cityscapes*, *ADE20K* and *COCO-stuff* datasets. Compared to SegFormer (Xie et al., 2021) which is also built with a attention window and an MiT (HVT method) encoder, the Lawin Transformer surpass it in all the tests. When switching the MiT for a Swin-Transformer on the Lawin and comparing with the Swin-UperNet (Z. Liu et al., 2021) and MaskFormer (Cheng et al., 2021) on the *ADE20K* dataset, it is concluded that by increasing the capacity of the encoder the Swin-Lawin outperforms the MaskFormer and UperNet. However, if the capacity is small the Swin-Lawin performs worse than MaskFormer.

Meta AI researchers (Kirillov et al., 2023) just published a project aimed to be the first foundation model for image segmentation. This project includes a new task, named promptable segmentation, inspired by natural language learning (NLP) foundation models, which is able return a valid segmentation mask when fed with any segmentation prompt, such as red flower. Additionally includes a model, Segment Anything Model (SAM), composed of three parts: an image encoder, specifically the MAE pre-trained Vision Transformer, adapted to process high resolution inputs, a flexible prompt encoder which considers two types of prompts: sparse (points, boxes, text) and dense(mask) and a fast mask decoder which is a modify version of a Transformer decoder block. Moreover, the Segment anything project includes a segmentation Anything Data Engine which is used to improve SAM's notion of objects by applying a generalization of two segmentation approaches, interactive segmentation, and automatic segmentation. This on-going part of the project enabled the creation of a 11 million images with 1.1 billion reliable segmentation masks all gathered on the SA-1B dataset.

7. Overview of growth rate and plant vigor methods

7.1. Plant vigor, detection, and identification methods

Plant detection and identification can be achieved by passing an image through a CNN algorithm, which then identifies the plant species. However, to achieve these results and use them for real life applications it is important to be aware of concepts such as plant phenology and the role that it plays.

Plant phenology is the timing of events in the life-cycle of plants, such as leaf bud break, flowering and fruiting. Phenology affects not only the fitness of individual plants, but also the fitness of organisms that depend on them, which in terrestrial ecosystems include virtually all animals (Stucky et al., 2018). Therefore, it is important to monitor changes in plants to ensure that they survive and do not have a negative impact on the environment. Therefore, the term "plant phenotyping" was introduced, which uses different methods to find and qualify plant characteristics (such as the leaf, the flower or the whole plant) by analyzing periodically taken images with precision (Das Choudhury

et al., 2019).There are several approaches ranging from detection to classification to segmentation to extract more complex parts of the plant such as the flower, the leaf or even the roots. Nowadays, this technique is used in various applications (apps) available to the public. In this section we will present some of these solutions as well as other studies that address this issue.

In the work of (Wang et al., 2021), a leaf recognition-based solution for plant identification is proposed. A multi-scale CNN with attention (AMSCNN) was used, which consists of several different scaled feature learning modules with an attention mechanism that assigns higher weights to important features to improve feature extraction. It achieved an accuracy of 95.28%.

(Barré et al., 2017) developed a deep neural network called LeafSnap to identify plants via its leaves. The CNN architecture applies the concept of dimension reduction modules, where each module consists in 2 convolutional layers followed by a MAX-pooling layer in order to reduce the number of connections between the last convolution layer and the first Dense layer. It was tested in three different datasets: Flavia (with 32 species), Foliage (with 60 species) and LeafSnap (with 184 species), having an accuracy of 97.9%, 95.8% and 86.3% respectively.

Other plant identification model using leaves is described in the following work by (Ganguly et al., 2022), which developed a classification model named BLeafNet combining deep neural networks with Bonferroni fusion learning. BLeafNet uses 5 deep learning models with the backbone algorithm ResNet-50, for different features. The model was tested in Malayakew, Leafsnap and Flavia datasets. This model had an accuracy of 92.22% for the Leafsnap dataset, 98.54% on the Malayakew dataset and 98.70% on the Flavia dataset. It's pointed out that before the fusion stage BLeafNet had an accuracy of 95.32% on the Malayakew dataset, showing an improvement of accuracy after fusion.

For the world of medicinal plant classification, this study from (Patil & Sasikala, 2023) shows a variation of the K-Nearest Neighbours (KNN) approach, called Weighted KNN, which changes the procedures to assign weights to the k points. After being tested, this algorithm had an accuracy of 98.62%.

This study from (Gogul & Kumar, 2017) aimed to identify medicinal plants in environments such as forests or mountains. Using images from a mobile phone camera, CNN was used to classify the species. This approach provided a solution to the problem of using hardware with lower capabilities by avoiding the use of CNN for feature extraction. Three architectures were used for feature extraction: OverFeat, Inception-v3 and Xception, with additional machine learning models such as Logistic Regression and Random Forest (RF). These results were fed into the CNN by resorting to Transfer Learning. CNN combined with OverFeat, Xception and Inception-v3 had an accuracy of 73.05%, 90.60% and 93.41% respectively.

The work of (Indra et al., 2021) also dealt with medicinal plants, more specifically, Indonesian medicinal plants. This work combined two algorithms, Principal Component Analysis (PAC) for feature extraction and the CNN (KNN) as a classifier. This study achieved an accuracy of 88.87%.

The work (Ibrahim Nehad M. Abdulrahman and Gabr, 2022) offers a solution for identifying wild plants by leaves, fruits or even both. This is the innovation compared to other studies. A wild plant dataset was collected from a natural habitat in Egypt and three techniques were tested: AlexNet, Random Forest (RF) and Support Vector Machine (SVM). It concluded that the convolutional neural

network (AlexNet) achieved the best results with an accuracy of 98.2%, compared to 96.7% and 96% for SVM and RF respectively.

8. Datasets

8.1. Plant identification

The *Flowers-299* (Cretu, 2020) dataset contains 299 flower species with a total amount of 115944 images. This dataset was developed by a bachelor's degree student for a final project of flowers identification with help of machine learning algorithms and was gathered with help of google image search. The images were resized with an average size of 271x242 pixels and filtered. Each class has between 222 and 483 images. It is worth to notice that this dataset holds three species that are of interested in this project: Armeria Maritima, Echium and Erica. Figure 13 represent two samples of the Flowers-299 dataset.

Figure 13: Representation of two images that the Flowers-299 holds. *(Left)* A representation of an Erica. *(Right)* Representation of an Echium(Cretu, 2020).

The *102 Category flower dataset*(Nilsback & Zisserman, 2008), developed by Maria-Elena Nilsback and Andrew Zisserman, is another dataset for flower identification. However, this dataset was a tool to study the impact of multiple combinations of features on the performance of a model. With this end, they used a support vector machine classifier and computed four different features for the flowers. This dataset has 102 categories of flowers. Each class holds between 40 to 258 images, which have large scale, pose and light variations. To visualize the classes is used isomaping with shape and color features. The dataset includes 5 files: image dataset, image segmentation, image labels, data splits and Chi^2 (Chi-Square) distances. Figure 14 shows to images of this dataset.

Figure 14: Representation of two images that the 102 Category flower dataset (Nilsback & Zisserman, 2008).

For weed identification, (Olsen et al., 2019)developed a dataset, *DeepWeeds*, that includes eight different weed species native of Australia on a total of 17509 images. Due the fact that robotic weed control faces a general obstacle of classifying weed on its natural environment, the goal was to use this tool to help develop systems for weed managing. Additionally, the images have unique filenames with the ID number of the instrument that take the image and the date and time when that happen. There is a representation of 2 images on Figure 15.

Figure 15: Representation of two images of weeds that are in DeepWeeds(Olsen et al., 2019).

Another dataset developed for the problem of weed identification was *4Weed* (Aggarwal et al., 2022). It contains 150 Giant Ragweed images, 170 Redroot Pigweed images, 139 Foxtail images, and 159 Cocklebur images, which correspond to the four most prevalent weed species reported in corn and soybean production systems. In order to prepare the dataset for training both image classification and object detection deep learning networks capable of precisely finding and recognizing weeds inside maize and soybean fields, bounding box annotations were made for each image. Figure 16 shows to images of two species that appear in this dataset.

Figure 16: Representation of two images of weeds that are in 4Weed dataset. *(Left)* Sample of the weed species Cocklebur *(Right)* Sample of the weed species Pigweed (Aggarwal et al., 2022).

Additionally, (Koklu et al., 2022) developed a dataset *Grapevine Leaves Image* to classify grapevine leaves. This dataset, prepared by collaboration between Selcuk University, Necmettin Erbakan University and Koramanoglu Mehmetbey University, based in Turkey, contains 5 species of vine leaves with a total of 500 images captured in a controlled light environment. The species are specific to the Central Anatolia region of Turkey and are as follows: Ak, Ala Idris, Buzgulu, Dimnit and Nazli. Each species is associated with 100 images.

Figure 17: Representation of two images of leaves that are in Grapevine Leaves Image dataset. *(Left)* Sample of the leaf species Ak. *(Right)* Sample of the leaf species Dimnit (Koklu et al., 2022).

Another leaf dataset was developed by (S. G. Wu et al., 2007). *Flavia* dataset was feed to a probabilistic neural network (PPN) with the purpose of classify plants by its leaves. The classification model was created on Matlab and the input vector feed to the PPN was created by extracting 12 leaf features and orthoganized into 5 principal variables. Afterwards the model outputted the Latin name

of the plant. It has total of 32 classes (1800 pictures) of leaves and each image is a single leaf with a white background. Figure 18 is an example of leaf image in this dataset.

Figure 18: Representation of one leaf image on the dataset(S. G. Wu et al., 2007).

Open Plant Phenotyping Database (OPPD) (Madsen et al., 2020) is a database of plant seedlings. This dataset is a tool for problems of visual recognition. It contains 7590 images of 47 different species which were cultivated using three growth conditions (ideal, drought and natural) and tracked while growing to have multiple images of several stages of the same plant which includes sown pictures. It has bounding box annotations and holds to different tasks instance detection and plant species classification. Some samples are presented in Figure 19.

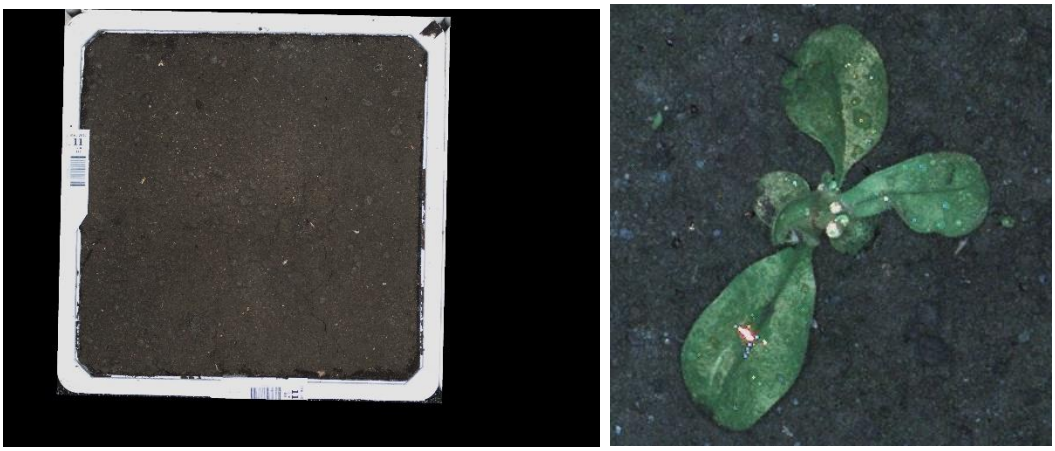

Figure 19: Representation of images on the OPPD dataset. *(Left)* Full Object Image - Representation of individual polystyrene boxes with the plants. *(Right)*Individual Plant Cut-Outs - Representation of all the bounding boxes annotated on the Full Object Image image (Madsen et al., 2020).

Zenkl et al. (2022) built a dataset for a wheat semantic segmentation task. The *Eschikon wheat segmentation (EWS)* dataset has 190 images collected over four years (2017, 2018, 2019, 2020) which holds all the stages of the wheat during all the seasons. This dataset was manually chosen from 100000 images gathered and annotated pixel-wise. It is divided according to illumination direct and diffuse light. Some examples are illustrated in Figure 20.

Figure 20: Representation of several images on the EWS dataset (Zenkl et al., 2022).

8.2. Not-exclusively plant datasets

The *ADE20K* is a dataset specialized on object detection and image segmentation, developed by (Zhou et al., 2017, 2019). It has a total of 27000 images of everyday scenes which were taken from Sun and Places dataset. Almost 3K object categories are thoroughly annotated in the images. Several photos also include object components and parts of other parts. Additionally, the original images have annotated polygons and object instances. Additionally, pictures are deidentified, masking faces and license plates. Figure 21 shows a sample of what the dataset offers.

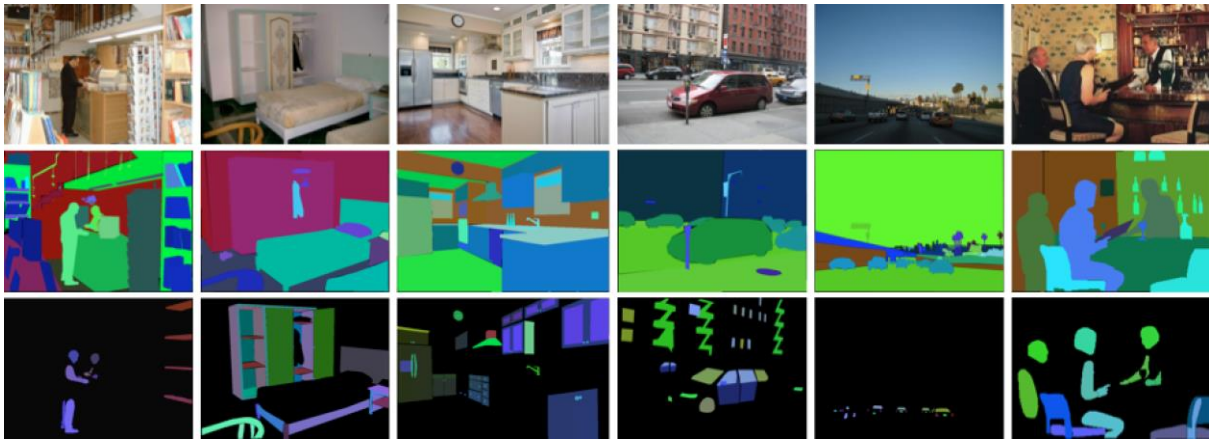

Figure 21: Representation of the images on the dataset. *(Top)* Images of the scenes. *(Middle)* Object Segmentation. *(Down)* Parts segmentation (Zhou et al., 2017, 2019).

The *Cityscapes* dataset was created to fill the gap of understanding urban street scenes, essential for autonomous driving. (Cordts et al., 2016) acquired data for several months in multiple cities during daytime, ultimately gathering over 25 000 images ready to be used in segmentation training. The dataset contains 30 classes created by polygonal annotations which 5 000 images are fine annotated and 20 000 images are coarse annotated, represented in Figure 22. These classes go from people to road to sky to vehicle. Furthermore, other researchers have added some extensions to the dataset such as bounding box annotations and augmented data by adding fog and rain.

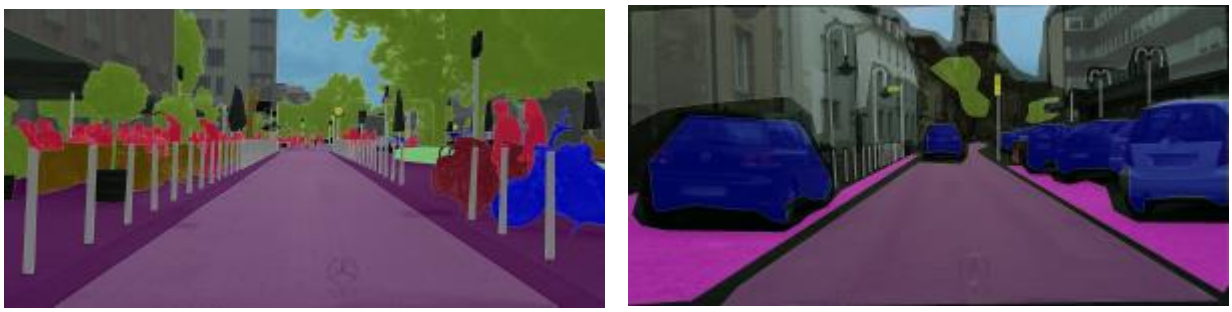

Figure 22: Representation of the images on the dataset. *(Left)* Images with fine annotation. *(Right)* Images with coarse annotation. (Cordts et al., 2016).

Still in the urban scene recognition, (Lin et al., 2015) developed a dataset containing objects from everyday life in its natural status. The *Common Objects in Context*, or *COCO*, contains 235000 images with over 91 object classes and with per-instance-segmentation that provides exact localization and labeling of objects. This dataset is one of the most used datasets for object detection and segmentation. Additionally, was created a ramification of this dataset named *Common Objects in Context-stuff*, or *COCO-stuff*,(Caesar et al., 2018) which focus on stuff classes. Including to the original dataset semantic segmentation annotations for scene understanding tasks, augmenting to 180 thing and stuff classes. This upgrade enables researchers to do scene parsing. The Figure 23 presents image examples of both datasets.

Figure 23: Representation of the images of the two datasets. *(Left)* Example of COCO dataset (Lin et al., 2015). *(Right)* Example of COCO-stuff dataset (Caesar et al., 2018).

The *SA-1B* dataset was developed by (Kirillov et al., 2023) when creating the project Segment Anything that aims to create a foundation model for image segmentation. The SA-1B dataset has the purpose of being a tool to researchers while training and evaluating models. Therefore, it was designed with 11 million wide world high resolution images taken directly from photographers in order to enable general purpose object segmentation. It also includes 1.1 billion mask segmentation collected by the segment anything engine, represented in Figure 24.

Figure 24: Mask examples of the SA-1B dataset (Kirillov et al., 2023).

Additionally, a dataset is currently being created. This dataset will be unique to this study and contain pictures of the species under investigation. The dataset will be made up of pictures shot over the course of several months from various perspectives. This is necessary in order to study the phenotype of this plants. This is necessary to analyze the phenotype of these plants.

A summary of the above-described datasets and related works is reported in Table 1 and Table 2, highlighting their applications.

Table 1: Datasets for plant identification.

Datasets	Types of plants classified	Task	Annotations	Type of annotations	Total number of images
Flowers-299(Cretu, 2020)	Flower	Classification	No	_____	115944
102 Category Flower(Nilsback & Zisserman, 2008)	Flower	Classification	Yes	Unknown	8189
DeepWeeds(Olsen et al., 2019)	Weeds	Classification	Yes	Bounding box Instance segmentation mask	17509
4Weed(Aggarwal et al., 2022)	Weeds	Classification Object Detection	Yes	Unknown	618
Grapevine Leaves Image(Koklu et al., 2022)	Leaves	Classification	No	_____	500
Flavia(S. G. Wu et al., 2007)	Leaves	Classification	No	_____	1800
Open Plant Phenotyping Database (OPPD) (Madsen et al., 2020)	Plant	Instance Segmentation Classification	Yes	Bounding box	7590
Eschikon wheat segmentation (EWS) (Zenkl et al., 2022)	Wheat	Semantic Segmentation	Yes	Pixel-wise Segmentation	190

Table 2: Non-exclusively plant datasets.

Datasets	Type	Task	Annotations	Type of annotations	Total number of images
ADEKA20(Zhou et al., 2017, 2019)	Objects	Object detection Segmentation Classification	Yes	Pixel-wise annotations Polygon annotations	27000
Cityscapes (Cordts et al., 2016)	Urban street scenes	Segmentation	Yes	Polygon annotations Bounding Box	25000
COCO (Lin et al., 2015)	Objects	Object detection Instance Segmentation	Yes	Per-instance segmentation	235000
COCO-stuff (Caesar et al., 2018)		Semantic Segmentation Scene Parsing	Yes	Bounding boxes Pixel-level annotations	164000
SA-1B (Kirillov et al., 2023)	Objects	Object detection Instance segmentation Instance Segmentation Scene Parsing	Yes	Mask annotations	1100000

9. Conclusions and final remarks

In order to provide an overview of the current state of the art in assessing the vigor and growth rate of wild plants, this report has assembled some important data on computer vision strategies that will be useful in solving future problems. Considering that the goal is to perform on-site monitoring using small processors and cameras, it is important to implement strategies that can work under these conditions. The algorithms to be implemented must take this into account. It is also extremely important to have a solid dataset of the wild plants selected for monitoring. If this is not possible, the datasets collected may be helpful in overcoming this limitation.
In summary, the information in this report will be a great tool to overcome the obstacles of this project and make it more economically feasible.

Bibliographic References

Aggarwal, V., Ahmad, A., & Etienne, A. (2022). *4Weed Dataset*. OSF. osf.io/w9v3j

Barla, N. (2023). *The Complete Guide to Panoptic Segmentation [+V7 Tutorial]*. https://www.v7labs.com/blog/panoptic-segmentation-guide#h4

Barré, P., Stöver, B. C., Müller, K. F., & Steinhage, V. (2017). LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics*, *40*, 50–56. https://doi.org/https://doi.org/10.1016/j.ecoinf.2017.05.005

Barreto, S. (2022). *Instance Segmentation Vs. Semantic Segmentation*. https://www.baeldung.com/cs/instance-semantic-segmentation-cnn

Caesar, H., Uijlings, J., & Ferrari, V. (2018). *COCO-Stuff: Thing and Stuff Classes in Context*.

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). *Rethinking Atrous Convolution for Semantic Image Segmentation*.

Chen, X., Zhao, Z., Zhang, Y., Duan, M., Qi, D., & Zhao, H. (2022). *FocalClick: Towards Practical Interactive Image Segmentation*.

Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., & Qiao, Y. (2023). *Vision Transformer Adapter for Dense Predictions*.

Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., & Chen, L.-C. (2020). *Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation*.

Cheng, B., Schwing, A. G., & Kirillov, A. (2021). *Per-Pixel Classification is Not All You Need for Semantic Segmentation*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). *The Cityscapes Dataset for Semantic Urban Scene Understanding*.

Cretu, B. (2020). *Flowers-299*. https://www.kaggle.com/datasets/bogdancretu/flower299

Das Choudhury, S., Samal, A., & Awada, T. (2019). Leveraging image analysis for high-throughput plant phenotyping. In *Frontiers in Plant Science* (Vol. 10). Frontiers Media S.A. https://doi.org/10.3389/fpls.2019.00508

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2018). *Dual Attention Network for Scene Segmentation*. arXiv. https://doi.org/10.48550/ARXIV.1809.02983

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). *Dual Attention Network for Scene Segmentation*.

Ganguly, S., Bhowal, P., Oliva, D., & Sarkar, R. (2022). BLeafNet: A Bonferroni mean operator based fusion of CNN models for plant identification using leaf image classification. *Ecological Informatics*, *69*, 101585. https://doi.org/https://doi.org/10.1016/j.ecoinf.2022.101585

Gao, R. (2021). *Rethink Dilated Convolution for Real-time Semantic Segmentation*.

Gogul, I., & Kumar, V. S. (2017). Flower species recognition system using convolution neural networks and transfer learning. *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, 1–6. https://doi.org/10.1109/ICSCN.2017.8085675

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask R-CNN*. arXiv. https://doi.org/10.48550/ARXIV.1703.06870

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*.

Hong, Y., Pan, H., Sun, W., & Jia, Y. (2021). *Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes*.

Ibrahim Nehad M. Abdulrahman and Gabr, D. G. and E. A.-H. M. (2022). A New Deep Learning System for Wild Plants Classification and Species Identification: Using Leaves and Fruits. In F. and G. F. Saeed Faisal and Mohammed (Ed.), *Advances on Intelligent Informatics and Computing* (pp. 26–37). Springer International Publishing.

Indra, R., Napianto, R., Nugroho, N., Pasha, D., Rahmanto, Y., & Yudoutomo, Y. (2021). *Implementation of PCA and KNN Algorithms in the Classification of Indonesian Medicinal Plants*. 46–50. https://doi.org/10.1109/ICOMITEE53461.2021.9650176

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). *Segment Anything*.

Koklu, M., Unlersen, M. F., Ozkan, I. A., Aslan, M. F., & Sabanci, K. (2022). A CNN-SVM study based on selected deep features for grapevine leaves classification. *Measurement*, *188*, 110425. https://doi.org/https://doi.org/10.1016/j.measurement.2021.110425

Li, X., You, A., Zhu, Z., Zhao, H., Yang, M., Yang, K., & Tong, Y. (2021a). *Semantic Flow for Fast and Accurate Scene Parsing*.

Li, X., You, A., Zhu, Z., Zhao, H., Yang, M., Yang, K., & Tong, Y. (2021b). *Semantic Flow for Fast and Accurate Scene Parsing*.

Li, X., Zhou, Y., Pan, Z., & Feng, J. (2019). *Partial Order Pruning: for Best Speed/Accuracy Trade-off in Neural Architecture Search*.

Li, Y., Mao, H., Girshick, R., & He, K. (2022). *Exploring Plain Vision Transformer Backbones for Object Detection*.

Li, Y., Xie, S., Chen, X., Dollar, P., He, K., & Girshick, R. (2021). *Benchmarking Detection Transfer Learning with Vision Transformers*.

Liang, J., Homayounfar, N., Ma, W.-C., Xiong, Y., Hu, R., & Urtasun, R. (2021). *PolyTransform: Deep Polygon Transformer for Instance Segmentation*.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). *Microsoft COCO: Common Objects in Context*.

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). *Path Aggregation Network for Instance Segmentation*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*.

Madsen, S. L., Mathiassen, S. K., Dyrmann, M., Laursen, M. S., Paz, L. C., & Jørgensen, R. N. (2020). Open plant phenotype database of common weeds in Denmark. *Remote Sensing*, *12*(8). https://doi.org/10.3390/RS12081246

Neuhold, G., Ollmann, T., Bulò, S. R., & Kontschieder, P. (2017). The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 5000–5009). https://doi.org/10.1109/ICCV.2017.534

Neven, D., Brabandere, B. De, Proesmans, M., & Gool, L. Van. (2019). *Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth*.

Nilsback, M.-E., & Zisserman, A. (2008, December). Automated Flower Classification over a Large Number of Classes. *Indian Conference on Computer Vision, Graphics and Image Processing*.

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., Banks, W., Girgenti, B., Kenny, O., Whinney, J., Calvert, B., Rahimi Azghadi, M., & White, R. D. (2019). DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning. *Scientific Reports*, *9*(1). https://doi.org/10.1038/s41598-018-38343-3

Patil, S., & Sasikala, M. (2023). Segmentation and identification of medicinal plant through weighted KNN. *Multimedia Tools and Applications*, *82*(2), 2805–2819. https://doi.org/10.1007/s11042-022-13201-7

Peng, J., Liu, Y., Tang, S., Hao, Y., Chu, L., Chen, G., Wu, Z., Chen, Z., Yu, Z., Du, Y., Dang, Q., Lai, B., Liu, Q., Hu, X., Yu, D., & Ma, Y. (2022). *PP-LiteSeg: A Superior Real-Time Semantic Segmentation Model*.

Romera, E., Álvarez, J. M., Bergasa, L. M., & Arroyo, R. (2018). ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Transactions on Intelligent Transportation Systems*, *19*(1), 263–272. https://doi.org/10.1109/TITS.2017.2750080

Stucky, B. J., Guralnick, R., Deck, J., Denny, E. G., Bolmgren, K., & Walls, R. (2018). The plant phenology ontology: A new informatics resource for large-scale integration of plant phenology data. *Frontiers in Plant Science*, *9*. https://doi.org/10.3389/fpls.2018.00517

Wang, X., Zhang, C., & Zhang, S. (2021). Multiscale Convolutional Neural Networks with Attention for Plant Species Recognition. *Computational Intelligence and Neuroscience*, *2021*, 5529905. https://doi.org/10.1155/2021/5529905

Wu, C.-Y., Hu, X., Happold, M., Xu, Q., & Neumann, U. (2020). *Geometry-Aware Instance Segmentation with Disparity Maps*.

Wu, S. G., Bao, F. S., Xu, E. Y., Wang, Y.-X., Chang, Y.-F., & Xiang, Q.-L. (2007). *A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network*.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers*.

Xu, J., Xiong, Z., & Bhattacharyya, S. P. (2022). *PIDNet: A Real-time Semantic Segmentation Network Inspired from PID Controller*.

Yan, H., Zhang, C., & Wu, M. (2022). *Lawin Transformer: Improving Semantic Segmentation Transformer with Multi-Scale Representations via Large Window Attention*.

Yuan, Y., Chen, X., Chen, X., & Wang, J. (2021). *Segmentation Transformer: Object-Contextual Representations for Semantic Segmentation*.

Zenkl, R., Timofte, R., Kirchgessner, N., Roth, L., Hund, A., Van Gool, L., Walter, A., & Aasen, H. (2022). Outdoor Plant Segmentation With Deep Learning for High-Throughput Field Phenotyping on a Diverse Wheat Dataset. *Frontiers in Plant Science*, *12*. https://doi.org/10.3389/fpls.2021.774068

Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., & Ding, E. (2019). *ACFNet: Attentional Class Feature Network for Semantic Segmentation*.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). *Pyramid Scene Parsing Network*.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., & Torralba, A. (2017). Scene Parsing through ADE20K Dataset. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., & Torralba, A. (2019). Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, *127*(3), 302–321.