AGRICULTURAL EVOLUTION WITH DEEP LEARNING

Deep learning is pivotal in modern agriculture, especially for advanced agri-robots. Using vast data, these robots discern patterns, optimising tasks from herbicide application to disease detection. However, their consistent performance hinges on a deep-learning system adept at agricultural complexities. This article explores challenges in crafting such systems, touching on economic impacts and design trade-offs. The AutoDL Platform is introduced, which is a solution for merging data & model management, task automation, and application insights.

ALBERT VAN BREEMEN

Introduction

Recent advances in deep learning are transforming agriculture (Figure 1). By processing extensive data, agri-robots recognise patterns and make precise decisions. Image recognition lets them differentiate between crops and weeds, optimising herbicide use and aiding disease prevention. They autonomously grade produce by size, colour and defects, and employ predictive analytics for improved navigation.

Agri-robots offer substantial benefits in farming. They diminish reliance on manual labour, leading to cost savings and uninterrupted operations. Their accuracy optimises seed, fertiliser, and pesticide use, boosting yields while reducing waste. Continuous monitoring facilitates early problem detection, supporting prompt interventions and informed decisions, especially in irrigation.

Agriculture's complex visuals (Figure 2), from varied crops to disease patterns, challenge traditional computer vision, which often misreads these intricacies. In contrast, deep-learning-based vision algorithms autonomously capture these details with superior precision, underpinning the advanced features of modern agri-robots.

However, deep-learning models are perpetually evolving. As fresh data surfaces and scenarios shift, models require updates to stay relevant. The ever-changing tech landscape and varying application needs drive these adjustments. Thus, a deep-learning operations platform is crucial, managing a model's lifecycle from training to monitoring, ensuring sustained peak performance. However, achieving this peak performance is not straightforward; there are numerous challenges and considerations to account for, as will be explored below.

Economics of agri-robots

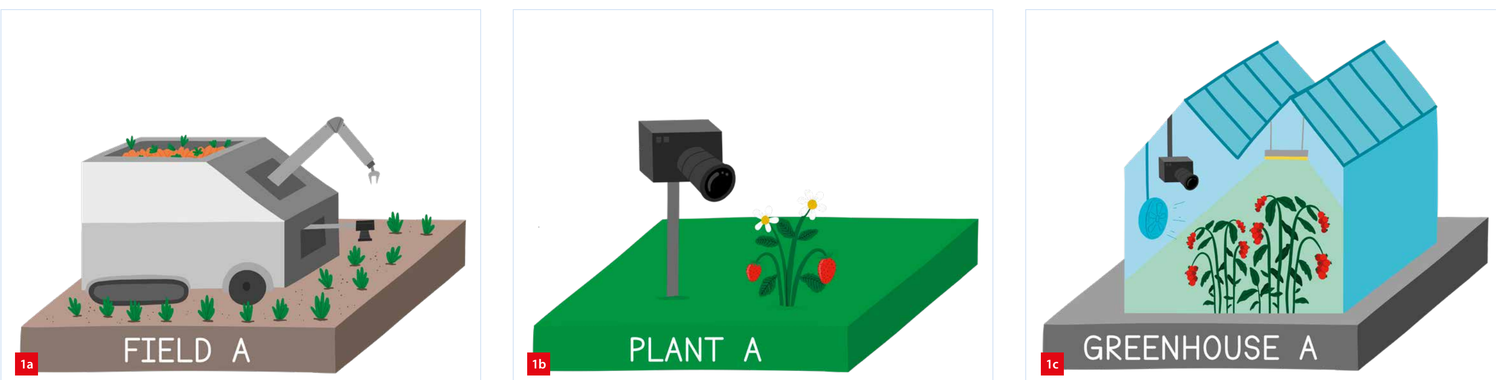
A basic economic model might help in understanding the intricacies of deep learning for agri-robots (Figure 3). To understand this model, envision a field with *M* objects ready for harvest. The agri-robot mainly consists of two parts: the visual detection system and the mechanical harvesting apparatus. In the model, if the visual system detects an object,

AUTHOR'S NOTE

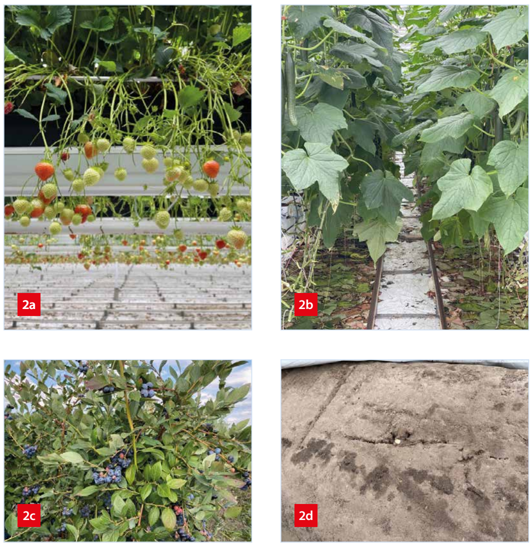
Albert van Breemen is the CEO/CTO of VBTI, an AI engineering company that develops deep-learning solutions for companies in areas such as agriculture and manufacturing.

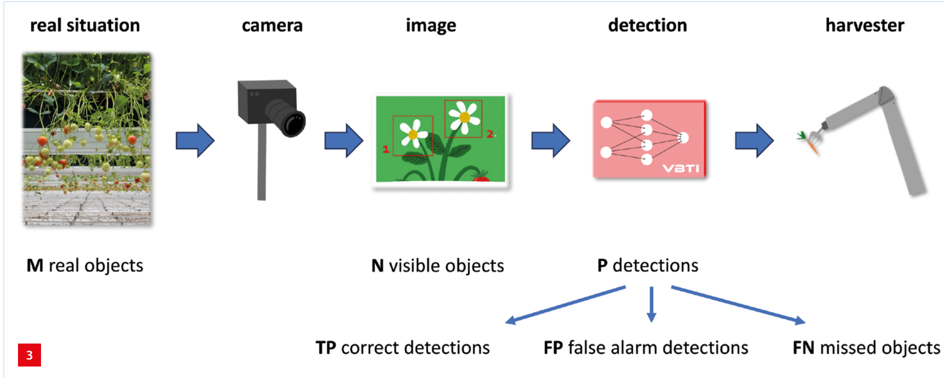
He has over 25 years of experience in artificial intelligence, high-tech technology and innovation.

albert.van.breemen@vbti.nl
www.vbti.nl



Applications of deep learning in agriculture.
(a) Agri-robots.
(b) Intelligent monitoring.
(c) Greenhouse automation.





Main factors impacting system performance:
1) Object visibility v and camera properties limit the actual number of objects visible on image.
2) Detection systems are never perfect.
3) Mechanical harvester actions are for h [%] successful.

Typical visual agriculture scenarios.
(a) Count the number of strawberries and estimate their ripeness.
(b) Differentiate between main stem and leaf stem of cucumber plant for a leaf-cutting robot.
(c) Count number of blue berries.
(d) Detect and locate asparagus just growing above ground.

the mechanical harvester acts with h [%] 'success rate'. There is also a 'visibility factor' v [%], the ratio between the total number of objects in the field M and those visible in the image N. This factor can be enhanced by tweaking camera angles or using mechanisms to provide an unobstructed view.

Detection systems are not infallible. Given N actual objects visible in the image and P predictions, the outcomes might be:
- true positive (TP): the system correctly detects an object;
- false negative (FN): the system misses an object;
- false positive (FP): the system inaccurately detects a non-existent object.

Here, $TP + FN = N = vM$ and $TP + FP = P$.

This rudimentary economic model for agri-robots centres on two aspects. Firstly, unharvested crops result from missed detections (e.g., false negatives). Secondly, needless harvester activations, stemming from false detections where no actual object exists (e.g., false positives). The maximum revenue and nominal cost are defined by:

$$Revenue_{max} = M \cdot p$$

$$Cost_{nominal} = (M / A) \cdot c$$

Here, A [actions/hour] is the harvester throughput, c [€/hour] is the cost per machine hour, and p [€/obj] is the price per harvested object.

Depending on object visibility, harvester success rate and detection failure, the actual revenue and cost will be:

$$\begin{aligned}
Revenue &= M \cdot v \cdot [TP / (FN + TP)] \cdot h \cdot p \\
&= v \cdot TPR \cdot h \cdot Revenue_{max} \\
Cost &= [(TP + FP) / A] \cdot c \\
&= [(TP + FP) / (M \cdot v)] \cdot v \cdot Cost_{nominal} \\
&= [(TP + FP) / (FN + TP)] \cdot v \cdot Cost_{nominal} \\
&= (TPR + FPR) \cdot v \cdot Cost_{nominal}
\end{aligned}$$

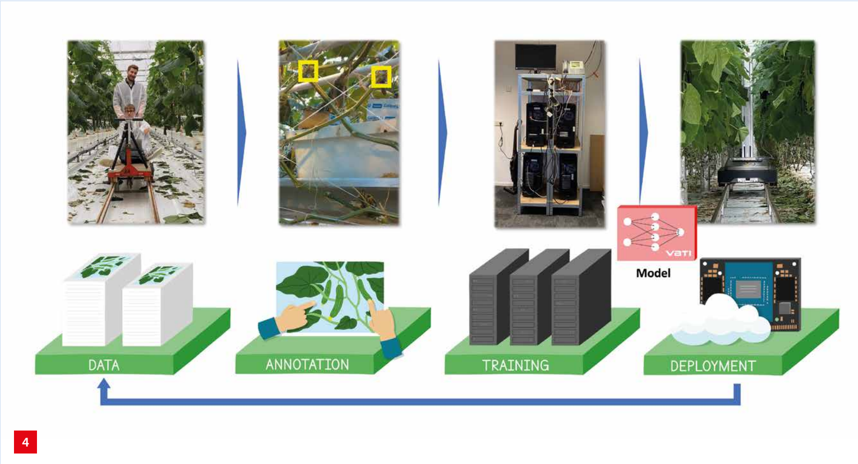
Here, TPR is the true positive rate (or recall), and FPR is the false positive rate (or false alarm rate).

From this model, it is evident that even a slight decrease in the TPR can dent farmer profits. For example, a TPR of 95% means 5% of potential harvest is overlooked. Moreover, every inaccurate detection diminishes efficiency. For example, if for every accurate detection there is an inaccurate one, the robot's harvest efficiency drops by half. In agriculture, a mere 5% dip in yield or profit is already impactful.

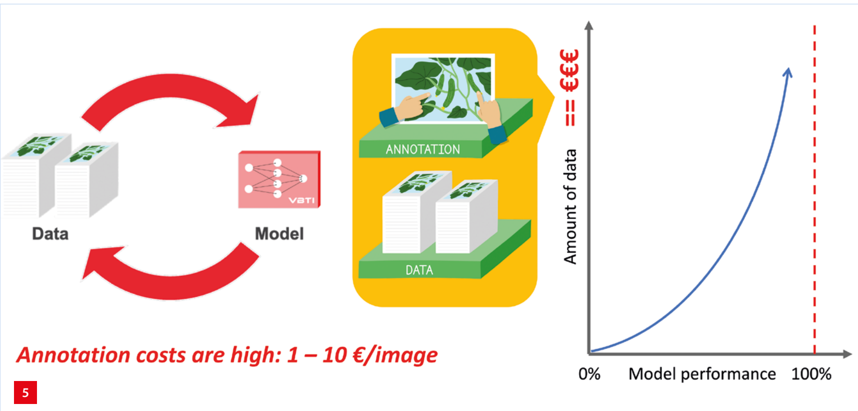
This underscores the high accuracy requirements of agri-robot visual systems. In specialised tasks such as leaf cutting, where incorrect detections risk damaging a plant's main stem, precision requirements can reach a remarkable 99.9997% (which equals cutting 5 main stems in a 6-months season @ 1000 leaves/hour, 8 hours/day), underscoring the demanding nature of developing robust deep-learning solutions.

Design trade-offs

Deep-learning model development involves a complex, iterative optimisation process (Figure 4), as the efficacy of a model is deeply influenced by its architecture and the dataset it is trained on.



The deep-learning 'big loop': collecting data, annotating data, training a model, deploying a model. The loop starts again when model performance is not sufficient anymore.



In general, more data increases the model performance, but also increases the cost to preprocess and annotate the data.

Two situations in different cucumber greenhouses. Lighting conditions, plant variety and surroundings are different, demanding for optimising the model per operational context.

Model architecture and size
Selecting the right architecture is pivotal for the model's performance. Distinct architectures cater for specific problem types. For image object detection, industry-recognised architectures include R-CNN, YOLO, and SSD, with each having multiple variations and distinct backbones such as VGG-16, ResNet, and MobileNet. Moreover, new architectures like DETR and ViT have emerged, leveraging vision transformers for object detection. Given the rapid advancements in this field, the most recent architecture might provide the solution you seek.

Model input size
Input dimensions, especially in images, are critical. Larger inputs capture intricate details but at the cost of computational power and memory. While they can increase the true positive rate by identifying subtle details, they may also increase the false alarm rate due to noise or non-relevant details.

Dataset size
A vast dataset presents diverse scenarios for the model, aiding its learning (Figure 5). But as the dataset size grows, ensuring its quality and relevance becomes crucial to maintain high accuracy. However, it's essential to acknowledge that at the onset of any deep-learning project, a 'complete' dataset is rarely available. As field conditions and scenarios evolve, so does the dataset, emphasising the need for periodic model training and refinements.

Operational context
Every specific robot has its own operational context (Figure 6) and because of the demanding performance of the detection systems requires its own optimised deep-learning model. These robots, working in varied environments, encounter unique data distributions based on their tasks and surroundings. This diversity in data distribution impacts the model's performance. As the operational context shifts, the data nature varies too, making it imperative for the model to remain adaptable and precise.

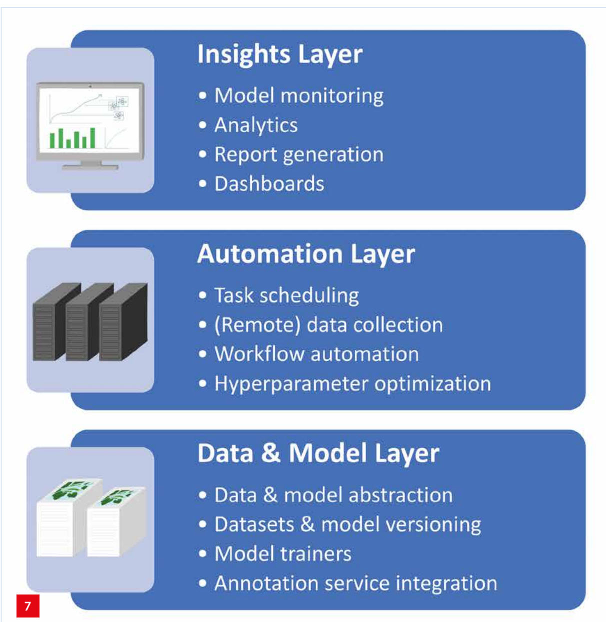
In summary, while the foundation of a deep-learning model lies in its architecture and the data it is trained on, it is essential to understand the interconnectedness of design decisions, such as input dimensions, the scope of the dataset, and the operational nuances of the robot's environment. The computational constraints of agri-robots directly influence both the model's architecture and input size, thereby limiting the ultimate potential of the vision system. Achieving the ideal balance necessitates an exhaustive exploration across possible combinations of model architecture, dataset and model training parameters.

Given the time-intensiveness of training and evaluation, automating this process is not just beneficial, but almost essential. Additionally, the sheer volume of data generated by agri-robots mandates a strategic approach to identify and integrate unique samples, further emphasising the role of automation.

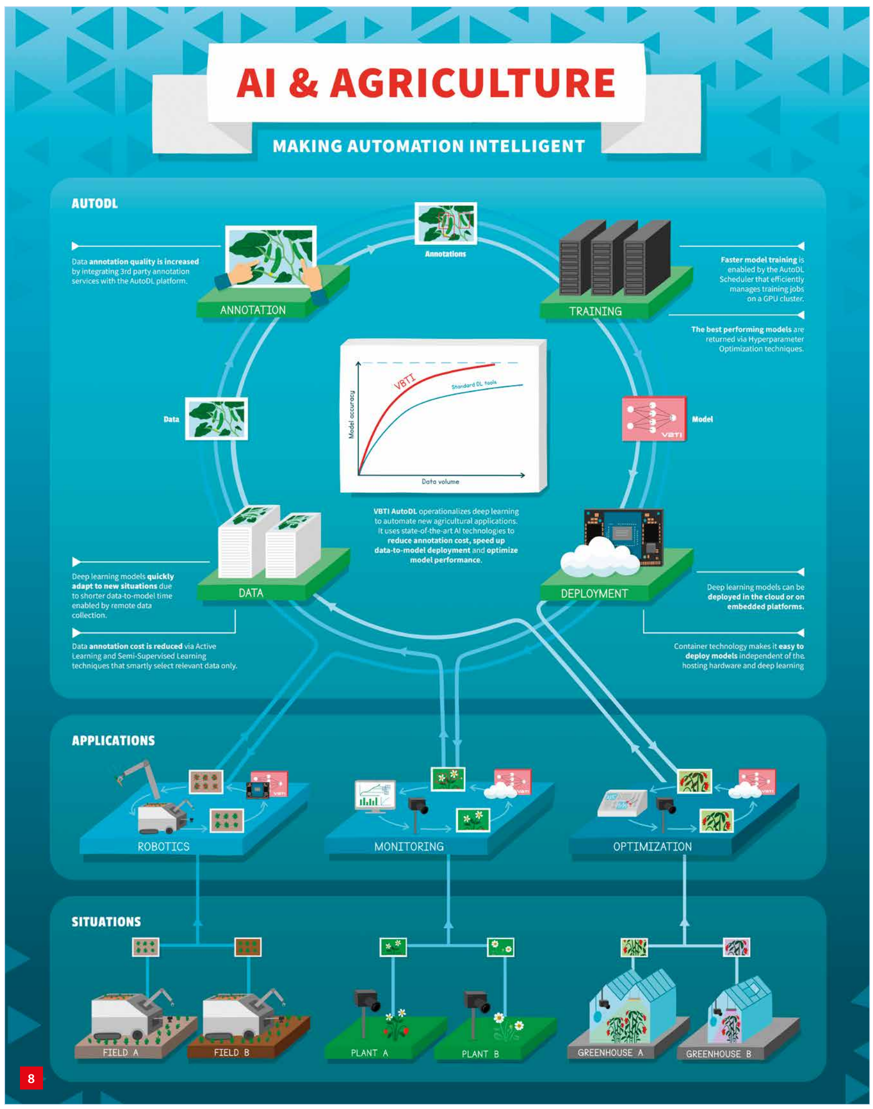
Deep-learning platform
Given the multifaceted challenges in crafting and refining deep-learning models for agri-robots, a specialised operational

platform is paramount. As agri-robots evolve and the intricacies of deep-learning models grow, a platform specifically tailored to the agri-robotics domain becomes indispensable. For this platform to truly address the unique needs of this sector, it should integrate the following key features:

- Data management capabilities
  Comprehensive functionalities for (remotely) collecting, storing, and preparing datasets are essential. Given agriculture's multifaceted nature, marked by diverse field conditions and a slew of variables, an agile and systematic data management framework is of utmost importance.
- Annotation support
  Annotation remains pivotal in priming datasets for deep-learning applications. To this end, the platform must either incorporate a state-of-the-art annotation tool or be compatible with external annotation services, ensuring fast and accurate data labelling.
- Framework neutrality
  To prevent inefficiencies and circumvent the hurdles of migrating between deep-learning frameworks (such as TensorFlow, PyTorch, ONNX, Ultralytics, SuperGradient, and more), the platform should remain deep-learning implementation framework neutral. Such a feature guarantees smooth transitions between different deep-learning frameworks without the burden of exorbitant switching costs.
- Training scaling
  At its essence, the platform should excel in model training. Equipped with the necessary computational and software tools, it should adeptly manage varied datasets, a spectrum of architectures, and the inherent challenges posed by agri-robotics.
- Automated model optimisation
  Considering the multitude of hyperparameters and configurations in deep learning, an automatic model-optimisation mechanism is indispensable. Leveraging cutting-edge algorithms, the platform should pinpoint the most suitable model configurations, minimising manual, and often tedious, experimentation.
- Performance monitoring
  It is vital to have a continuous (remote) model performance monitoring system in place. This ensures the early detection of any performance drift, facilitating swift rectifications.
- Business insights dashboard
  Beyond its technical capabilities, the platform should adeptly convert raw model metrics and operational statistics into tangible business insights. A user-friendly dashboard, providing a concise view of pivotal performance indicators, efficiency measures, and other pertinent data, aids stakeholders in drawing informed conclusions.



The deep-learning operations platform AutoDL consists of three layers: data & model layer, automation layer and insights layer.



The AutoDL platform is a deep-learning operations platform that manages data, models and insights in order to optimise model performance for every specific operational context.

In conclusion, the envisioned deep-learning platform for agri-robots must strike a balance – it should seamlessly intertwine the sophisticated technical procedures with tangible business insights while upholding its user-centric design, adaptability and scalability.

AutoDL platform

VBTI developed a deep-learning operations platform called AutoDL that implements the features listed previously. This platform unfolds across three distinct yet interconnected layers, each offering solutions tailored to agri-robots (Figure 7).

Data & model layer

This layer delves into data abstraction and framework integration. With the evolving landscape of robotics, data formats such as RGB, RGBD and pointclouds, and stereo images have become crucial. Recognising this diversity, the platform seamlessly integrates and manages these varying formats.

This layer further introduces an advanced datastore that not only stores data but also incorporates detailed metadata, embedding vectors and dataset versioning. The latter is crucial for reproducible model training results and testing different models against the same evaluation datasets. Such a set-up streamlines smart searches, particularly valuable when sifting through extensive datasets.

Additionally, to simplify the often overwhelming deep-learning landscape for users, 'model trainer' abstractions have been integrated. By abstracting the complexities of deep-learning frameworks and leveraging Docker software containerisation technology (Docker is a tool for packaging and running applications in a consistent and isolated way, making them easier to deploy and manage), users can focus on modelling without the distractions of underlying technical intricacies.

Automation layer

This layer implements automation, scalability and remote data collection. The deep-learning lifecycle, spanning from data collection to model deployment, is intricate. To navigate this complex loop, the platform introduces automation at every stage, ensuring efficient and optimised agri-robot performance. Given the compute-intensive nature of deep-learning tasks, especially training, the platform is fortified with GPU acceleration, offering the much-needed scalability.

Complementing this is a scheduler that manages various jobs, from training to evaluations, guaranteeing optimal resource usage. As real-time data collection becomes increasingly paramount for model monitoring, the platform not only supports remote data collection but also smartly selects the most valuable data, ensuring consistently high-quality insights.

Insights layer

This layer highlights continuous monitoring and actionable business insights. As agri-robots are set into action, the platform's strength in continuous data monitoring comes alive. By interpreting this data, it can be distilled into business dashboards, shedding light on the robot's efficacy, potential areas for enhancement, and more. The data's depth allows the platform to undertake predictive analytics, such as forecasting plant growth stages, thereby optimising robot tasks.

Furthermore, the platform's flexibility ensures its evolution in parallel with agricultural needs. For instance, it can be trained to recognise specific disease patterns or pests, ensuring the agri-robots remain state-of-the-art. Beyond robot performance, the platform expands its monitoring horizon to evaluate the productivity of the field and individual plants, granting users a holistic view of their operations.

Multi-layered

In summation, the multi-layered deep-learning operations platform (Figure 8) has been designed to meet the specific demands of agri-robots. It encompasses everything from data handling and model training to real-time insights, ensuring a holistic and streamlined approach to agricultural deep learning.

Conclusion

Integrating deep learning into agriculture, notably through agri-robots, marks a transformative turn in modern farming. Utilising vast data, these robots detect intricate patterns, enhancing operations like disease identification and precision irrigation. Yet, their commercial potential rests on the strength and adaptability of the embedded deep-learning systems. These models, central to agri-robotics' growth, bring challenges from data acquisition to real-time oversight.

The AutoDL Platform's tri-layered design streamlines this complex landscape, blending data abstraction, automation and insights. It forges a crucial link between deep learning and hands-on agricultural tasks. As agri-robotics progresses, platforms such as AutoDL become vital, shaping commercially viable robots and elevating traditional farming through technology.

11 *Cutter impression.*

Single camera and deep-learning model set-up gives insufficient performance.

Case: leaf-cutting robot

9

Prototype leaf-cutting robot by VDL CropTeq.

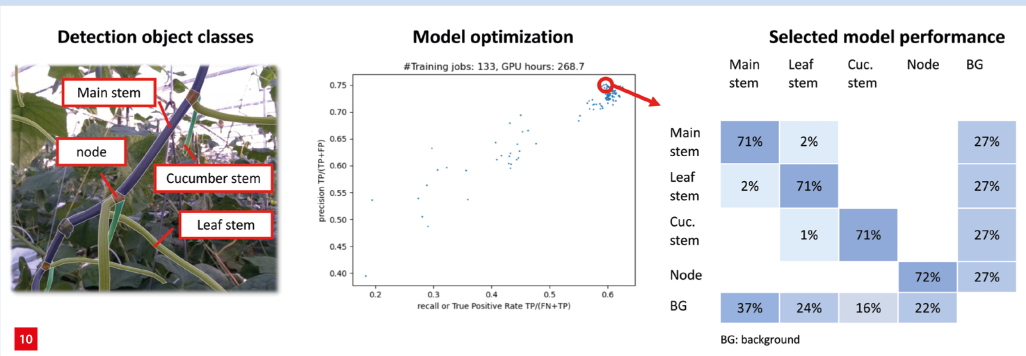
Cucumber plant maintenance in greenhouses involves leaf cutting, a crucial activity to maintain plant productivity. However, this task is labour-intensive. In collaboration with VDL Cropteq, VBTI aimed to innovate by introducing an autonomous robot to undertake this duty (Figure 9).

In the initial approach, a single-camera vision system was adopted for detection. The system's primary objective was to classify four essential plant components: main stem, leaf stem, cucumber stem, and nodes. To derive the best-performing model tailored to the dataset, a model optimisation search was undertaken (Figure 10). Typically, a single search encompassed several hundred training jobs, consuming hundreds of GPU-hours. This task is fully automated in the AutoDL platform.

Given the high costs associated with mistakenly cutting the main stem, the detection system had to exhibit unparallelled precision. A challenge emerged: distinguishing between the main stem and the leaf stem, as their visual differences are minute, posing a constraint on the system's performance.

Recognising the need for refinement, the subsequent design incorporated multiple overview cameras alongside a close-up camera. Unlike the previous model that relied on a single snapshot, the enhanced system (Figure 12) utilises the overview cameras to track objects over time. Now, the system's performance is enhanced by integrating redundancy in the vision system – both temporally (e.g., tracking) and spatially (using varied camera angles, including a detailed close-up for better pixel representation of plant details). With this set-up, there are now two models to train: one model for the overview cameras and one model for the close-up camera.

Once the robot becomes operational, the performance of the models is continuously monitored. Since every model is trained and evaluated on a specific dataset, should the operating conditions begin to deviate from the training dataset's statistics, there may be a need for new data collection and model retraining.

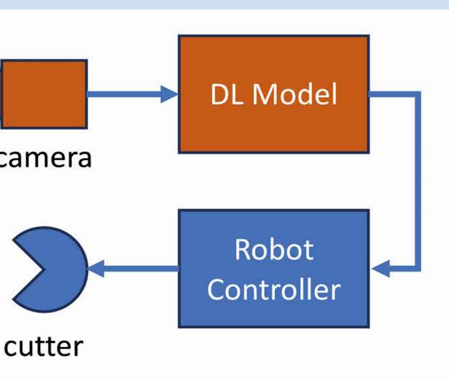


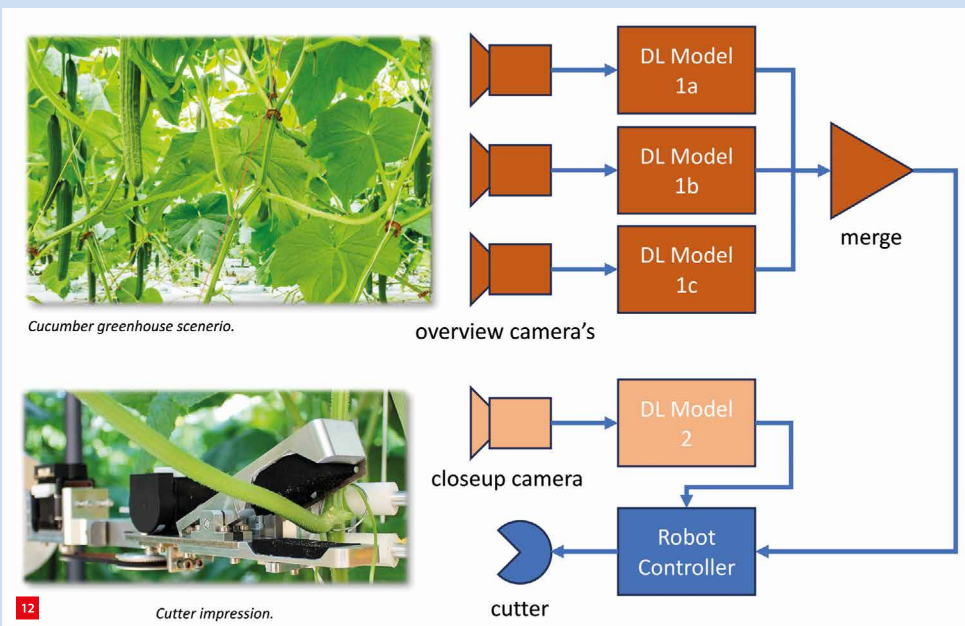
	Main stem	Leaf stem	Cuc. stem	Node	BG
Main stem	71%	2%			27%
Leaf stem	2%	71%			27%
Cuc. stem		1%	71%		27%
Node				72%	27%
BG	37%	24%	16%	22%	

BG: background

10

Detection model definition, model optimisation and model performance. Note that the model performance needs to be corrected for the fact that not all objects in the background have been annotated.





12

Utilising multiple cameras and deep-learning models enhances performance relative to a singular camera and model set-up.