Deep Learning Algorithms in Agriculture

Subjects: Agricultural Engineering | Computer Science, Artificial Intelligence | Agronomy

Contributor: Pedro Gaspar

The field of agriculture is one of the most important fields in which the application of deep learning still needs to be explored, as it has a direct impact on human well-being. In particular, there is a need to explore how deep learning models can be used as a tool for optimal planting, land use, yield improvement, production/disease/pest control, and other activities. The vast amount of data received from sensors in smart farms makes it possible to use deep learning as a model for decision-making in this field. In agriculture, no two environments are exactly alike, which makes testing, validating, and successfully implementing such technologies much more complex than in most other industries.

Keywords: Agriculture ; Deep learning ; Disease Detection ; Fruit ; Weed detection ; Automation ; Artificial intelligence ; Computational vision ; Remote sensing

1. Disease Detection

Plant diseases cause losses in agricultural production and endanger food security. The most common practice in pest and disease control is to spray the crop area evenly with pesticides. This method requires significant amounts of pesticides, which results in high financial and human resources and significantly changes the environment. The use of a deep learning algorithm that detects the exact time, location, and affected crops and sprays the pesticides only on the affected plants can reduce resource consumption and environmental changes.

Kerkech et al. [1] used the SegNet model to segment and detect grapevine diseases using images with RGB and infrared ranges. The dataset was acquired using a UAV device with two MAPIR Survey2 camera sensors, including a visible light (RGB) sensor set for automatic illumination and an infrared sensor. The dataset was labeled using a semi-automatic method, i.e., a sliding window to identify potentially diseased areas. Then, each block was classified by a LeNet5 network for pre-labeling. In the end, the labeled images were manually corrected. The SegNet recognizes four classes including the shaded areas, soil, healthy and symptomatic vines. Two models were trained, one for the RGB images and the other for the infrared images. The segmentation results of the two models were also fused in two ways. The first case was called “Fusion AND”, which means that the symptom is considered detected if it is present in both the RGB and infrared images. The second case is called “fusion by the union” and has the symbol ”fusion OR”, which means that the symptom is considered detected if it is present in either the RGB or the infrared image. The model trained with RGB images (Acc = 85.13) outperformed the model trained with infrared images (78.72). The fusion AND had the best performance, and the fusion OR had the worst accuracy. The runtime of SegNet on a UAV image was reported to be 140 s for visible and infrared images. The fusion between the two segmented images takes less than 2 s.

Kerkech et al. [2] used a CNN model to detect Esca disease in grapevine using UAV RGB images. A CNN model was trained with different combinations of patch sizes 16 x 16, 32 x 32 and 64 x 64 with different color spaces including RGB, HSV, LAB, and YUV, and vegetation indices such as ExG, ExR, ExGR, GRVI, NDI, and RGI. All the different color spaces and vegetation indices used can be calculated from the RGB images. The results show that the CNN model trained with the RGB and YUV color spaces has a better performance compared to the models trained with HSV and LAB. It was pointed out that the lower accuracy of the models trained with HSV could be due to the Hue (H) channel in HSV, which combines all color information into a single channel and is less relevant for the network to learn the best color features. The LAB color space has one luminance channel (L) and two chrominance channels, which do not reproduce the colors of the diseased vineyard well.

In the next experiment, vegetation indices were added to the RGB and YUV data. Combining vegetation indices with RGB and YUV improved classification results in most cases. The final investigation concerned the models trained by combining the vegetation indices alone. The combination of ExR, ExG, and ExGR vegetation indices with a size of 16 x 16 gave the best performance among the other inputs (including color space) and sizes with an accuracy of 95.80%. Furthermore, the combination of YUV and ExGR vegetation indices with sizes of 32 x 32 and 64 x 64 achieved similar performance but the run-time was longer.

From the results, the color of the images and leaves is very important in the detection of grapevine diseases. Furthermore, the number of channels in the input does not affect the run-time of the model, but the size of the input and the model structure do.

Barbedo [3] investigated how dataset size and diversity affect the performance of DL techniques applied to plant diseases. An image dataset of leaves with a small number of samples for the CNN model was used to investigate the behavior of GoogLeNet under different conditions. Data augmentation and transfer learning methods were used to train the CNN. The results showed that even with transfer learning and augmentation techniques, CNN requires a large number of images to extract useful features from the data. Even though a large number of images can be easily acquired with the new technology, labeling the dataset is time-consuming. One option proposed was to share the dataset, but as mentioned earlier, the two environments are not the same in agriculture. The effect of removing the background of the images on the accuracy of the model was also investigated. The model has trained again with the images without background. The model has trained again with the images without background. The results show different behaviors with respect to the accuracy of the model, including no significant effect, a significant improvement in accuracy in some cases, a significant decrease in accuracy, and mixed results (improved accuracy in some diseases while the error rate increased in others). From the significant decrease in accuracy for some plants, it was inferred that the CNN model sometimes uses the background of the model to classify the images. An attempt that can be made here is to train the model with both datasets (with and without background) and investigate the accuracy of the model in this case.

Ferentinos (2018) [4] used AlexNet, AlexNetOWTBn [5], GoogLeNet, Overfeat [6], and VGG models to identify 25 plant disease in 58 different classes of (plant, disease) combinations, including some healthy plants. The dataset used was images of healthy and infected leaves of the plants from the Plantvillag database (https://github.com/spMohanty/PlantVillage-Dataset, accessed on 17 November 2021). More than 37.3% of the images in the dataset were taken under real cultivation conditions in the field, and the other images were taken under laboratory conditions. In the first experiment, the number of images acquired under laboratory conditions and real conditions was kept similar in the training and test set, and the models were trained using this dataset. The VGG model performed best on the test set with an accuracy of 99.53%.

They also investigated the significance of the presence of field-captured images in the training set. From the 58 available classes of the form (plant, disease), the 12 that contained images of both types were selected. Two experiments were conducted with these 12 classes, and two CNN models were developed: one was trained with images under laboratory conditions and tested with images under field conditions, and another was trained with images under field conditions and tested with images under laboratory conditions. Although the number of images acquired under field conditions was less than the number of images acquired under laboratory conditions, the CNN model trained only with images under field conditions performed better with an accuracy of 68% than the CNN model trained only with images under laboratory conditions with an accuracy of 33%. This result shows the importance of the presence of the images acquired under field conditions. From the misclassification image, they point out some problematic situations, including images with extensive partial shading on the leaves, images with multiple objects in addition to the image, images where the leaf occupies a very small and non-centric part of the frame, and images without leaves.

Jiang et al. [7] implemented the CNN models to detect five common apple leaf diseases using images from the field. By applying data augmentation, such as rotational transformations, horizontal and vertical flips, intensity disturbances, and Gaussian noise, 26,377 disease images were generated. The problem with SDD is that it cannot detect a small object, and also, an object can be detected multiple times. To overcome these drawbacks, they developed an SSD model with a VGG-INCEP as the backbone, where two GoogLeNet inception layers replace two convolutional layers of the VGG model. Moreover, the structure of feature extraction and fusion is designed by applying the Rainbow concatenation method instead of the pyramid (which is used in the SSD model) to improve the performance of feature fusion. The results showed that the data augmentation improved the accuracy of the model by 6.91% compared to the original dataset. Moreover, the proposed model achieved the best performance compared to faster R -CNN and SSD with VGG and Rainbow SSD (RSSD) model.

To investigate the effect of using a deep model as a backbone, ResNet-101 was used as a feature extractor for SSD. The results show that ResNet-101 does not lead to any improvement. In terms of speed, Faster RNN was the slower model with more accuracy. The proposed model with 78.80% mAP and 23.13 Frames Per Second (FPS) was more accurate than SDD and RSSD, but SSD outperformed the other models in terms of time inference. By examining the misclassified images, it was indicated that similarity between diseases, misclassified background, and light condition were the challenges in classification.

Karthik et al. [8] proposed a Residual Attention Network for disease detection in tomato leaves. The main idea of attention is to focus on the relevant parts of the input to produce outputs. The dataset was collected from the open-source website Plantvillage and contained one healthy class and three diseases. They implemented three models. One was a traditional CNN model, the second used the three residual layers introduced into the CNN model as part of the ResNet architecture, and the last used an attention mechanism on top of the residual CNN for effective feature learning. The traditional CNN-based methods focus on ordered feature learning, starting from basic image-level features such as edges, color, etc., to complex texture-based differences [8]. In the deeper layer, some important features extracted in the first layer may be lost. The residual layers are designed to avoid this problem [8]. They concatenate the extracted features from the earlier layers with the deeper layers. In addition, the attention mechanism is used to extract the relevant parts of the feature maps. The Residual Attention Network CNN performed better with an overall accuracy of 98% than the Vanilla Residual Network with an accuracy of 95% and the traditional CNN model with an accuracy of 84%.

Liu et al. [9] implemented the model Cascade Inception to detect four common apple leaf diseases in images captured in the field. The Cascade Inception was a modified AlexNet model with inception layers from GoogleNet. Various data enhancement methods such as image rotation, mirror symmetry, brightness adjustment, and PCA jittering were applied to the training images. Moreover, the fully connected layers were replaced by convolutional layers, which results in fewer parameters and avoids overfitting. The proposed model was trained using the optimization method Nesterov Accelerated Gradient (NAG) and achieved an accuracy of 97.62%. The performance of the proposed model was compared with SVM and BP neural networks, standard AlexNet, GoogLeNet, ResNet-20, and VGGNet-16. Transfer learning method was used to train VGGNet-16 and achieved 96% accuracy. Standard AlexNet, GoogLeNet, and ResNet-20 were trained from scratch using SGD-optimization and achieved a maximum accuracy of 95.69%. The SVM and BP, which achieved an accuracy of less than 60%, show that the traditional approaches rely heavily on the expert developed classification features to improve the recognition accuracy. They also investigated the effect of data augmentation methods and optimization algorithms on accuracy. The model with the SGD optimizer achieved 93.32% accuracy, while the model with NAG achieved 97.62% accuracy. The data augmentation methods improved the performance of the model by 10.83%. The advantage of the model is that it outperformed other CNN models in terms of training time and memory required. Furthermore, the number of parameters of the model was less than AlexNet and GoogleNet. One point that emerges from the paper is that GoogleNet and AlexNet were trained using the SGD optimizer, and the proposed model was trained using the NAG method, but as mentioned in the paper, SGD has the “local optimum” problem. In addition, the models were not compared based on the inference time.

Ramcharan et al. [10] trained the SSD object detection model to identify three diseases, two types of pest damage, and nutrient deficiency in cassava at the mild and pronounced stages. A dataset was collected from the field, which was divided into 80–20 as training and testing sets. The model achieved 94 ± 5.7% mAP on the test dataset. The trained model was used on a mobile phone to investigate the performance of the model in the real world. One hundred twenty images of leaves were captured using a mobile device of the experiment, and the model inference was run on a desktop and mobile phone to calculate the performance metrics of the trained model. The results show that the average precision of the model on the real dataset decreases by almost 5%, but the average recall decreases by almost half, and the F1-score decreases by 32%. Furthermore, the results show that the model performs better on the leaves with pronounced symptoms than on the leaves where the symptoms are only mildly pronounced.

Picon et al. [11] used DL model to detect seventeen diseases of five crops (wheat, barley, corn, rice, and rapeseed) in images captured in the field. The dataset contained several challenges, such as multiple diseases on the same plant, similar visual symptoms among diseases, images of early and early-stage diseases, and diseases of leaves, and stems. To improve the accuracy of the model, the crop ID was used in the network. The crop ID was defined as a categorical vector with K components, where K is the number of crops in the model. Several models were trained. The first approach was to train an independent model for each crop and the second approach was to train a single model for the entire dataset. The results show that the multi-crop model had a similar performance to splitting the training dataset into the different crops (1% increased accuracy). However, the class of diseases with fewer images in the training set may benefit from the multi-crop model. The second approach was to add the crop ID to the multi-crop model. The results show that by adding crop id, the model can still benefit from more images for training, while crop ID information helps the network to discriminate between similar diseases.

Chen et al. [12] used pre-trained MobileNet V2 networks to identify twelve rice plant diseases. The dataset was collected from online sources and real agricultural fields. An attention mechanism was added to the model, and transfer learning was performed twice during model training to achieve better performance. The MobileNet-V2 achieved 94.12% accuracy on the PlantVillage dataset, while the MobileNet-V2 with attention and transfer learning achieved 98.26%. The five CNNs such as MobileNetv1, MobileNet-V2, NASNetMobile, EfficientNet-B0, and DenseNet121 were selected for comparison with the proposed models. In addition, two other networks named MobileNetv2-2stage and MobileAtten-alpha were trained. In the MobileNetv2-2stage model, transfer learning was performed twice to identify the images of plant diseases.

Similarly, in MobileAtten-alpha, the attention method was used instead of transfer learning twice. The proposed model (using attention and transfer learning twice) achieved the the second best accuracy of 98.84%. The DenseNet121 with an accuracy of 98.93% outperformed other models. MobileNetv2-2stage and MobileAtten-alpha achieved an accuracy of 98.68% and 96.80%, respectively. The proposed model was trained with images from the field and achieved an accuracy of 89.78%.

2. Fruit Detection and Yield Forecast

Yield forecasting is one of the most important and popular topics in precision agriculture because it is the basis for yield mapping and estimation, supply and demand matching, and crop and harvest management. Modern approaches go far beyond simple forecasting based on historical data but incorporate methods from DL to provide a comprehensive multidimensional analysis of crop, climatic and economic conditions to maximize yields.

Silver and Monga [13] trained five CNN models to estimate grape yield from RGB images taken in a vineyard on harvest day using the camera of a Samsung Galaxy S3. The images of 40 grapevines were split into two parts, one for the left cordon and one for the right cordon, resulting in a total of 80 cordons. A simple CNN was trained by inputting the RGB images of the left and right cordons and estimating the grape yield. The second model was the same as the first model, but the input to the model was the right cordon images and the inversion of the left cordon images to look similar to right cordon images. The third model, an autoencoder network, is trained to a high level of accuracy and the CNN encoder weights are transferred as starting weights into the CNN model for the yield estimation model. In the fourth model, an autoencoder model was trained to output the density map of the grapes in the image. Then, the weights of the encoder are transferred as initial parameters into the CNN model for yield estimation. The last model was trained with the output of the autoencoder for the density map as input to the CNN model for yield estimation instead of the RGB images. The CNN models with flipped images outperformed the simple CNN model with an MAE % of 15.43. The models of transfer learning from Density Map Network Representation with MAE % of 11.79 achieved the best performance among the other models. The last model with MAE % of 15.99 did not perform as well as the fourth model because the accuracy of the density map estimates was quite low. The results show that transfer learning, when used properly, can improve recognition accuracy.

Aguiar et al. [14] trained SSD MobilenetV1 and the Inception model to detect Grape Bunch in Mid Stage and early stages and then transferred the trained model to the TPU-Edge device to investigate the temporal accuracy of the model. The same strategy in [15] was used to collect the dataset and it is publicly available (https://doi.org/10.5281/zenodo.5114142, accessed on 3 November 2021) with 1929 vineyard images and their annotation. Overall, SSD MobileNet-V1 performed better than the Inception model, as it had a higher F1 score, AP, and mAP. The early stage was more difficult to detect than the middle growth stage. The first class represented smaller clusters that were more similar in color and texture to the surrounding foliage, making them more difficult to detect. SSD MobileNet-V1 showed an AP of 40.38% in detecting clusters at an early growth stage and 49.48% at a medium growth stage. In terms of time, SSD MobileNet-V1 was more than four times faster than the Inception model on TPU-Edge Device.

Ghiani et al. [16] used Mask R-CNN with ResNet101 as a backbone which was pre-trained with the dataset COCO (https://cocodataset.org/#home, accessed on 18 November 2021) for detecting grape branches on the tree. An open-source dataset GrapeCS-ML [17] containing more than 2000 images without annotation of fifteen grape varieties at different stages of development in three Australian vineyards was used to train the model. In addition to the GrapeCS-ML dataset, 400 images were collected from the island of Sardinia (Italy). The result obtained by applying the detector to the test samples was an mAP value of 92.78%. To investigate the generalizability of the proposed model, the model trained on the GrapeCS-ML dataset was tested on its internal dataset. The dataset contained different grape varieties, vegetation, and colors than the GrapeCS-ML dataset. An mAP of 89.90% was obtained with the internal dataset, indicating that the model can be used in other fields. To investigate the significance of the size of the dataset used for training and the importance of the augmentation techniques, the size of the original dataset was reduced to 10% of the training images in one case with augmentation and the other case without augmentation of the dataset. The mAP was reduced by up to 5%, especially in the experiments performed without augmentation. It was observed that the recognition accuracy decreased for images with overlap between clusters.

In Milella et al. [18], a system for the automatic estimation of harvest volume and for detecting grapes in vineyards using an RGB-D sensor on board an agricultural vehicle has been proposed. An RGB-D sensor is a special type of depth detection device that works in conjunction with an RGB camera and is able to add depth information to the conventional image pixel by pixel. The approach to determine the crop volume involved three steps: the 3D reconstruction of the vine rows, the segmentation of the vines using a deep learning method, and the estimation of the volume of each plant. In the first step, the depth output from the camera was not used because the parameters of the algorithm are fixed and cannot be configured. Instead, the semi Global Matching (SGM) algorithm was used, which is a computer vision algorithm for estimating a dense disparity map from a rectified stereo image pair. A box-grid filter is then used to merge the point clouds.

In segmentation, a segmentation algorithm was first used to separate the canopy from the background using the green–red vegetation index (GRVI), and then k-means were used to identify each plant. Based on the results of the clustering algorithm, different plants were calculated and an estimate of the volume per plant was performed. Finally, the pre-trained AlexNet, VGG16 and VGG19, and GoogLeNet were trained to perform grape cluster detection in 5 different classes (grapes, vineyard stakes, vine stems, cordons, canes, leaves, and background). The VGG model performed the best with an accuracy of 91.52.

Santos et al. [19] used deep learning models and computer vision models to estimate grape wine yield from RGB images. Images were captured with a Canon EOS REBEL T3i DSLR camera and a Motorola Z2 Play smartphone from five different grape varieties. The dataset named Embrapa Wine Grape Instance Segmentation Dataset (WGISD) with 300 RGB images is publicly available (https://doi.org/10.5281/zenodo3361736, accessed on 18 November 2021). Models from DL such as Mask R-CNN, YOLOv2, and YOLOv3 were trained to detect and segment grapes in the images. Then, an image processing algorithm called Structure-from-Motion (SfM) was used to perform spatial registration, integrating the data generated by the CNN-based step. In the final step, the results of the CV model were used to remove the clusters detected in different images to avoid counting the same clusters in several images. The Mask R-CNN with ResNet101 as the backbone outperformed YOLOv2 and YOLOv3 in terms of object detection, but the YOLO model outperformed the Mask R-CNN in terms of detection time. The worst performance was obtained with YOLOv3. To verify the performance of Mask R-CNN+SfM, 500 key-frames of a video were used, and the result is shown in a video at https://youtu.be/1Hji3GS4mm4, accessed on 17 November 2021.

Palacios et al. [20] applied the method of deep learning to detect the flowering of the vine and used it for the estimation of early yield estimation. Images of six grapevine varieties were acquired using a mobile platform developed at the University of La Rioja. The RGB camera was a Canon EOS 5D Mark IV RGB with a full-frame CMOS sensor. The ground truth was acquired using the algorithm in [21]. This algorithm was developed to process only images with a single inflorescence and a dark background. In the first step, SegNet was used with VGG (VGG16 and VGG19) as a backbone to segment and extract the inflorescences contained in the images. Then, these regions were used to count the flowers in each inflorescence using three algorithms, including SegNet, Watershed Flower Segmentation, and a linear model. The SegNet with VGG19 as backbone outperformed the model with VGG16 in terms of IOU and F1-score. For flower recognition, the SegNet model with VGG19 was trained to classify a group of flowers per image into three classes including contour, center, and background. After segmentation, false-positive filtering of flower segmentation was performed. Here, the flowers whose center and contour was segmented, and whose contour surrounded the center above a certain threshold were considered as true positives.

Kang and Chen [22] implemented DaSNet-v2, which is an Encoder–Decoder with atrous convolution developed in Deeplab-v3+ to detect and segment the apple in an orchard for harvesting by a robot. Atrous convolution is a type of convolutional layer that allows control of the resolution of the features computed by the CNN. The dataset was collected from an apple orchard as RGB-D and RGB. The RGB-D was used to visualize the environment. A lightweight model, Resnet 18, was used as the backbone of the DaNet-v2 to ensure its deployment on the Jetson-TX2 with limited computing capacity. In addition, the model was trained with the Resnet50 and Resnet101 backbones. The performance of the model with the Resnet101 backbone was compared with DaSNet-v1, YOLO-v3, faster-RCNN, and the Mask-RCNN. DaSNet-v2 and Mask-RCNN with F1-score of 0.873 and 0.868, respectively, outperformed the other models. However, DaSNet-v2 outperformed mask-RCNN with a computational efficiency between 306 and 436 ms with a time of 1.3 s. The results also show that single-stage detection models such as Yolo have better computational efficiency compared to two-stage detectors. The model was implemented on Jetson-TX2, a lightweight backbone of Resnet-18, and the experimental results show that DaSNet-v2 with Resnet-18 can achieve similar performance in recall and precision of detection compared to YOLO-v3. Environmental factors such as strong sunlight reflection, shadows, and appearance variations of fruits in color, shape, occlusion, or viewing angle could lead to false-negative detection results. RGB-D images were processed using DaSNet-v2 and the PPTK toolbox, a Python package for image visualization, to deploy the robot in the orchard.

Koirala et al. [23] developed a DL model, called Mango-YOLO, based on YOLO-v3 and YOLO-v2 (tiny) for counting mangoes on trees. YOLOv2 (tiny) has a small architecture with only nine convolutional layers, six pooling layers, and one detection layer, sacrificing accuracy for speed. YOLOv3 [24] is based on Darknet-53 and improves upon YOLOv2 [25]. The Mango YOLO had 33 layers compared to 106 layers in YOLO-v3 and 16 layers in YOLO-v2 (tiny). The reduction in the number of layers is expected to reduce computation time and detect mangoes more accurately. The model was trained on the dataset collected from five orchards. The Mango-YOLO achieved better performance with an accuracy of 0.967% compared to YOLO-v2 (tiny) (0.9%) and YOLO-v3 (0.951%). In terms of time inference, the Mango YOLO with 15 ms was faster than YOLO-v3 (25 ms) and slower than YOLO-v2 (tiny) (10 ms). Moreover, Mango-YOLO was trained once from scratch on the augmented dataset, and the second time transfer learning was used using the COCO dataset. The models had the same performance on the test set, and the reason was explained by the fact that the COCO dataset does not contain Mango images. The false detection over images taken with Canon camera shows resizing of the images and results in image distortion with leaves taking a curved shape resembling the fruit and overexposed areas on branches, trunks, and leaves.

Tian et al. [26] developed YOLO-V3 with DenseNet as the backbone to detect apples on trees. They used two datasets for training. The first one contained images of apples at one growth stage, and the second one contained images taken at different growth stages. The model showed better performance for mature apples than for young and growing apples because the color features were more prominent, the individual volume was larger, and there was less overlap. The results also showed that the F1-score of the model trained with the first dataset was higher than that of the model trained with the second dataset. The performance of the trained model decreased for images with partial occlusion of apples with branches and leaves but is still an acceptable result (IOU = 0.889 for mature apples).

The model achieved the best performance compared to Faster R-CNN, YOLOV2, and the original YOLOv3. In terms of time inference, the model was faster than Faster R-CNN and similar to YOLOv3. The F1-score and IOU of the model without data augmentation methods decreased by 0.033 and 0.058, respectively.

Zhou et al. [27] implemented an SSD model with two lightweight backbones MobileNetV2 and InceptionV3, to develop an Android APP called KiwiDetector that detects kiwis in the field. Quantization is a technique for performing computations and storing tensors with bit widths smaller than floating-point numbers. When training a neural network, 32-bit floating-point weights and activation values are typically used. A quantized model performs some or all operations with tensors using integers instead of floating-point values. This allows the computational complexity to be reduced and the trained model can be used on devices with lower resource requirements. The quantization method was used to compress the model size and improve the detection speed by quantizing the weight tensor and activation function data of convolutional neural networks from 32 to bit floating-point numbers to an 8-bit integer. The results showed that MobileNetV2, quantized MobileNetV2, InceptionV3, and quantized InceptionV3 achieve a true detection rate of 90.8%, 89.7%, 87.6%, and 72.8%, respectively. The quantized MobileNetV2 on the Huawei P20 smartphone outperformed the other models in terms of detection speed (103 ms/frame) and size. Although the SSD with MobileNetV2 was more accurate than the SSD with quantized MobileNetV2, the SSD with quantized MobileNetV2 was 37% faster.

3. Weed Detection

Besides disease, weeds are considered to be a prevalent threat to agricultural production. These are plants considered undesirable in a particular situation, as they may compete with crops for sunlight and water, resulting in crop and economic loss. One significant problem in weed control is that they are difficult to detect and distinguish from crops. DL algorithms can improve weed detection and discrimination at a lower cost with reduced environmental problems and side effects. These technologies could power robots that detect and remove weeds.

Bah et al. [28] proposed a CNN model with unsupervised training dataset annotation collection for weed detection in UAV images of bean and spinach fields. They assumed that crops are grown in regular rows and that plants growing between the rows are considered weeds. The Hough transform was applied to the skeleton to detect the rate of plant rows, and then a simple linear iterative clustering (SLIC) algorithm was applied to create a marker and delineate the plant rows. This algorithm generated superpixels based on k-mean clustering. After row detection, a blob coloring algorithm was used to identify the weeds. The unsupervised training dataset was used to train ResNet18 for weed detection in the images. The supervised learning method performed 6% better than the unsupervised learning method in the bean field and about 1.5% better in the spinach field. The low number of weeds between rows may explain the difference in performance in the bean field. The performance of the model was compared with SVM and RF. In general, ResNet18 shows better performance in supervised and unsupervised learning methods than SVM and RF.

Ferreira et al. [29] implemented the unsupervised learning models JULE and DeepCluster to detect weeds in the field. The JULE consists of stacked multiple combinations of convolutional layer, batch normalization layer, ReLU layer, and pooling layer. AlexNet and VGG16 were implemented as the basis of the DeepCluster model to extract features, and then K-means is used as the clustering algorithm. Two datasets, Grass-Broadleaf [30] and DeepWeeds [31] were used. The DeepCluster model performed better than the JULE model on both datasets with a large number of clusters. The DeepCluster with a base of Alexnet and VGG achieved similar performance, with Alexnet performing better on DeepWeed and VGG on the other dataset. To investigate the effect of transfer learning on unsupervised learning, the pre-trained VGG and AlexNet were used on ImageNet. The pre-trained model did not improve the accuracy of Grass-Broadleaf, but it did improve the accuracy of DeepWeed. The data augmentation also improved the accuracy of the unsupervised learning methods. They also used semi-supervised data labeling. Semi-supervised learning is a machine learning approach that deals with the use of labeled and unlabeled data. In the semi-supervised method, labeled images from the DeepCluster model were used to train Inception-V3, VGG, and ResNet. Inception-V3 and VGG outperformed ResNet on the Grass-Broadleaf and DeepWeeds dataset, respectively.

Milioto et al. [32] modified Encoder–Decoder CNN architecture in [33] to distinguish weeds from crops and soil. The number of convolutional layers was decreased to reduce time inference, and additional vegetation indices (14 channels) were included in the input for more accurate detection. The dataset of three fields was used. Three networks were trained with different inputs: one with RGB images, another with RGB and near-infrared (NIR) images, and the last one was trained with 14 channels such as RGB, Excess Green (ExG), Excess Red (ExR), color index of Vegetation Extraction (CIVE), and Normalized Difference Index (NDI). To investigate the generalization ability of the proposed model, the model was trained on the images of one field and tested on images of all fields. The results indicate that feeding these 14-channels into the input can speed up the training and improve the performance of the model on the unseen dataset compared to the model trained on RGB images and RGB+NRI but still, the recall and precision of the model on the unseen field drop sharply (11–50%).

Another experience was conducted to investigate the generalization capacity of the proposed model. The trained model was retrained on the unseen dataset with 10, 20, 50, or 100 images in the training set. The performance of the RGB network when retrained with 100 images is almost the same as the performance of the proposed model trained with ten images. The inference time of the proposed model on PC and Jetson TX2 platform was also found to be 44 ms and 210 ms, respectively, which is slower than the model trained with RGB images with 31 ms and 190 ms, respectively.

Lottes et al. [34] developed an encoder–decoder Fully Convolutional Network (FCN) with the sequential model for weed detection in sugar beet fields. The encoder–decoder FCN was used to extract features from the input images, and the sequential model processed the five images in a sequence using 3D convolution and output a sequence code that was used to learn sequential information about the weeds in five images in a sequence. The results showed that the encoder-decoder with a sequential model improved the F1-score of the module by almost 11–14% compared to the encoder-decoder FCN. The results indicated that the changes in the visual appearance of the images in the training and test dataset could lead to a decrease in model performance, and adding additional information, such as vegetation indices, leads to better generalization for other fields.
Wang et al. [35] used the encoder–decoder with Atrous Convolution for pixel-wise semantic segmentation of crops and weeds. The two datasets for sugar beet and oilseed included in the paper were taken under completely different lighting conditions. To mitigate the effects of the different lighting conditions, three image enhancement methods were evaluated, including Histogram Equalization (HE), Auto Contrast, and Deep Photo Enhancer. The models were also trained with various inputs, including YCrCb and YCgCb color spaces and vegetation indices such as Normalized Difference Index (NDI), Normalized Difference Vegetation Index (NDVI), Excess Green (ExG), Excess Red (ExR), Excess Green minus Excess Red (ExGR), Color Index of Vegetation (CIVE), Vegetative Index (VEG), and Modified Excess Green Index (MExG), Combined Indices (COM1), and COM2. For the sugar beet dataset, the model trained with NIR images enhanced by Auto Contrast outperformed the other models with a mean IOU of 87.13%. For the Oilseeds dataset, the models were trained with RGB images only, and the model trained with images enhanced by Deep Photo Enhancer outperformed the other models (mIOU = 88.91%).

4. Species Recognition

The classification of species (e.g., insects, birds, and plants) is another critical aspect of agricultural management. The traditional human approach to species classification requires specialists in the field and is time-consuming. Deep learning can provide more accurate and faster results by analyzing real-world data.

Rußwurm and Körner [36] trained LSTM and GRU models for multitemporal classification, which achieved high accuracy in crop classification tasks for many crops. Images were collected from SENTINEL 2A images between 31 December 2015 and 30 September 2016. For consistency with the LANDSAT series, the blue, green, red, near-infrared, and shortwave infrared 1 and 2 bands were selected for this assessment. The performance of the LSTM model was compared to the RNNs, the CNN model, and a baseline SVM. The LSTM model outperformed the other models in terms of accuracy in land cover classification.

Lee et al. [37] used CNN to classify 44 plants based on leaf images. Two datasets were prepared. One contained the entire image of a leaf, and for the other, each leaf image was manually cropped. The accuracy of the CNN model trained with the second dataset was higher than that obtained with the first dataset. The results showed that CNN can extract high-level features such as structural subdivisions, leaf tip, leaf base, leaf margin, etc., and is not limited to shape, color, and texture.

Ayhan et al. [38] implement DeeplabeV3+, a CNN model developed from scratch and a machine learning method that uses NDVI (NDVI+ML) to segment vegetation and non-vegetation in the images. The dataset of [39] belongs to two studied sites, Vasiliko in Cyprus and Kimisala in Rhodes Island, were used. The images were acquired using a UAV and a modified, uncalibrated near-infrared camera. The image resolution of the Vasiliko image is 20 cm per pixel and these images were used to train the models. On the other hand, the Kimisala dataset was used as a test set. The images in this dataset contain two different resolutions of 10 cm per pixel and 20 cm per pixel.

Two DeeplabeV3+ models were trained with RGB images as input and G, B bands plus NDVI as the third channel. Although the loss of the DeeplabeV3+ model trained from scratch with NDVI+GB was decreased, the test result was very poor. However, the same model trained with transfer learning and NDVI+GB channels improved the accuracy of the model compared to the model trained with RGB images (almost one percent). The CNN developed from scratch was also trained with two different channels. The first was trained with RGB images and the second with four channels of RGB and NIR. The model trained with RGB and NIR outperformed the model trained with RGB images with an accuracy of 80.9% and an accuracy of 76%.

In the last experience, the NDVI and the Gaussian Mixture Model (GMM) of the machine learning model were used to classify the images. This method includes several thresholds that need to be adjusted by the user. The NDVI+ML method outperformed the models from DL with an accuracy of 87%. Note that the deep learning method was trained with a dataset from one location and tested at another location with a different land cover. This could be a reason why NDVI+ML outperformed the DL models.

Bhusal et al. [40] used the pre-trained MobileNet on the ImageNet to classify bird pests in the image. Video data from a commercial vineyard captured with a GoPro Hero 4 outdoor camera with 1080P resolution was used as the dataset. In their work, a motion detection algorithm was used that is capable of detecting moving objects. For each of these moving objects, the abounded rectangle was extracted. These moving objects were cropped from the original RGB image and reduced to

. Each of these cropped images was referred to as motion instance images (MIIs). More than 5000 MIIs were collected from different videos and classified as bird or non-bird. In the next step, a CNN model developed in [41] was used to improve image resolution. Five MobilNet models were trained with the MIIs, 2e-MIIs, 3e-MIIs, 4e-MIIs, and the entire Dataset. The worth results in terms of accuracy were obtained with a model trained with MIIs, and the best result was obtained with the model trained with all the datasets. Mac Aodha et al. [42] used the CNN models from scratch to detect bats from audio files. Two CNN models called CNNFULL with three convolutional layers and 32 filters and CNNFAST consisting of two convolutional layers and 16 filters were trained. The audio files were converted to a spectrogram and used as input to the CNN model. CNNFULL and CNNFAST took 53 and 9.5 s, respectively, to run the entire detection pipeline on the 3.2-min full-spectrum test dataset. CNNFAST showed a trade-off between speed and accuracy with slightly lower performance compared to CNNFULL.

The performance of the models was compared with three existing commercial closed-source detection systems, including SonoBat (version 3.1.7p) (https://sonobat.com/, accessed on 26 November 2021); SCAN'R (version 1.7.7), and Kaleidoscope (version 4.2.0 alpha4) (https://www.wildlifeacoustics.com/products/kaleidoscope-pro, accessed on 26 November 2021), as well as a machine learning method RF. The CNN model significantly outperformed the other algorithms in terms of mAP.

Ramalingam et al. [43] used Internet of Things (IoT) based architecture for insect detection. The Internet is a global system of interconnected computer networks that use Transmission Control Protocol/Internet Protocol (TCP/IP) to connect billions of computers worldwide [44]. The IoT, on the other hand, is a global network of physical objects equipped with sensors and actuators that connect to the Internet in real-time to be identified, sensed, and controlled remotely [44]. The IoT architecture used in [43] consists of four layers: perception layer, transport layer, processing layer, and application layer. In the perception layer, a smart wireless camera captures an image of a sticky insect trap. The transport layer uses WiFi communication and TCP/IP to send images to the processing layer and transmit processed data to the application layer. In the processing layer, an Fast RCNN with ResNet50 is used to detect the insect in the images. In the final stage, smartphones and web interfaces are used to perform the application layer tasks. The experimental results show that the trained model achieves 96% accuracy in insect detection and outperforms YOLO and SSD in terms of accuracy.

5. Soil Management

For experts in agriculture, soil is a heterogeneous natural resource with complex processes and unclear mechanisms. Its temperature alone can provide insight into the impact of climate change on regional yields. Deep learning algorithms study the processes of evaporation, moisture, and soil temperature to understand the dynamics of the ecosystem and the implications for agriculture.

Li et al. [45] used a bidirectional LSTM model to estimate soil temperature at 30 sites under five different climate types. Soil temperature (ST) measurements were obtained from the U.S. Department of Agriculture’s National Water and Climate Center, which has established more than 200 sites across the country to collect data on meteorology, soil, and solar radiation. Two models were trained with different inputs and outputs. The first model received the meteorological weather conditions including daily hours, minimum and maximum air temperature, minimum and maximum relative humidity, vapor pressure, average solar radiation, and average wind speed, and outputted the soil temperature amplitude obtained by subtracting the daily average soil temperature from the hourly soil temperature. The second model obtained the meteorological weather conditions including month, day of the month, observed air temperature, dew point temperature, minimum and maximum air temperature, minimum and maximum relative humidity, vapor pressure, average solar radiation, average wind speed, and outputted the daily average ST. To calculate the hourly ST, the output of the first model was added with the output of the second model and called the integrated BiLSTM model. The result of the model was compared with the BiLSTM model, which directly estimates the hourly ST, the LSTM model, the deep neural network, random forest, SVR, and Linear Regression. The integrated BiLSTM model outperformed the other models in terms of MAE, RMSE, and R2. The LSTM model achieved the second-best performance.

It was also found that the performance of each model is not as good in snowy areas as in warm or dry areas and that, the accuracy of the other models increases with soil depth (except for RF). This behavior could be due to a change in the standard deviation of the soil temperature at different depths and climate types, but this is not investigated.

Yu et al. [46] implement CNN (Conv2D, Conv3D), ConvLSTM to estimate the soil temperature. The difference between Conv2D and Conv3D is the size of the input channels. In the ConvLSTM model [47], there are convolution structures in both the input-to-state and state-to-state transitions. The model obtains the last ten days of historical data from spatiotemporal ST and predicts the ST one, three, and five days in advance. Each model was trained with two different input channels. The first time the raw data from ST was fed into the model, the second time, the input was processed using the Empirical Mode Decomposition (EMD) method, which is a proposed method for processing signals. In the EMD method, the number of channels was increased from one to ten. To complement the model DL, persistent prediction (PF) is used, a simple prediction method that treats the temperature of the first day as a prediction for the next day. When forecasting ST with one-day historical data, PF outperformed the models of DL with raw data input. On the other hand, when the models of DL used the EEMD- processed data as input, the prediction performance was significantly improved. Among all the models, EEMD-Conv3D performed the best in predicting the spatiotemporal ST. It could be noted that ST depends not only on the historical data of ST but also on the meteorological weather conditions, which can be used as input to the model to improve the accuracy.

6. Water Management

Water management in agriculture has implications for hydrological, climatological, and agronomic balance. Two research focus on the topic, where machine learning models are employed to estimate evapotranspiration, allowing for more effective use of irrigation systems, and water table detection, which helps determine crop water needs.

Saggi and Jain [48] used multilayer perceptrons (MLP) to evaluate daily evapotranspiration for irrigation scheduling. They developed their model from scratch. The developed DL model performed well in estimating evapotranspiration and outperformed ML models such as RF, Generalized Linear Model (GLM), and Gradient Boosting Models (GBM).

Zhang et al. [49] presented an LSTM-based model for water management in agricultural areas. The LSTM model with a value of 0.789–0.952 outperformed the fully connected network. As mentioned in the paper, the prediction of water table depth can help engineers and decision-makers to develop an optimal irrigation scheduling system while controlling the effects of salinity on intensive irrigation.

Loggenberg et al. [50] applied hyperspectral sensing and machine learning to model water stress in vineyards. Stem water potential (SWP) was measured in the field by using a pressure chamber to determine vine water stress status. Vines with SWP values between −1.0 MPa and −1.8 MPa were classified as water-stressed, while vines with SWP values −0.7 MPa were classified as not stressed. Images were acquired using the SIMERA HX MkII hyperspectral sensor, which detects 340 spectral wavebands in the VIS and NIR. A spectral subset consisting of 176 wavebands with a spectral range of 473–708 nm was used as input to the models. Two ML models Random Forest (RF) and Extreme Gradient Boosting (XGBoost) were used for classification. In the first experience, RF and XGBoost models used all wavebands (176 bands) as input. RF with an accuracy of 83.3% outperformed XGBoost with an accuracy of 78%. When using a subset of important wavebands (18 bands), the accuracy of RF and XGBoost was improved (RF = 93.3% and XGBoost = 90%). The results show that the choice of input to the model in ML is crucial and should be carefully selected. The effect of smoothing the spectral data with the Savitzky–Golay filter in the data preprocessing step was also investigated in the paper. The Savitzky–Golay filter reduces the model accuracy ranging between 0.7% and 3.3%.

Chen et al. [51] used a deep Q-learning (DQN) model for an irrigation decision strategy based on short-term weather forecasts for rice. Daily observed meteorological data of the rice-growing period at three stations, including daily minimum and maximum temperature, average temperature, average wind speed, sunshine duration, average relative humidity, and precipitation were collected.

Alibabaei et al. [52] uses the DQN model to schedule irrigation of a tomato field in Portugal using climate Big Data. Historical data are collected from various sources and processed for use as input. Two LSTM models are trained on the obtained historical data to predict the total soil water in the soil profile for the next day and the tomato yield at the end of a season, respectively.

The trained LSTM models were used in the DRL training environment, which takes the current state (historical climate data) and action (amount of irrigation) and then returns the next state and reward. The reward was calculated as the net return and the Q-value was estimated using an LSTM model. The results show that the agent learns to avoid water waste at the beginning of the season and water stress at the end of the season. Compared to the threshold and fixed irrigation method, the DQN agent increases productivity by 11% and avoids water waste by 20–30%.

7. Automation in Agriculture

Deep learning is also being used to control sensors and robots that enable automation and optimization of agricultural processes. These robots can be used for a variety of purposes, including automated seeding, pesticide, and crop nutrient application, damage repair, irrigation, weed detection and removal, harvesting, etc.

Li et al. [53] used a deep-learning algorithm to quickly and accurately detect and locate longan fruit in imagery and pick the fruit with a UAV device. An RGB-D camera on the UAV was used to collect images of the fruits. As mentioned in the paper, one of the disadvantages of UAVs is that they are easily affected by local circulation and airflow when capturing images, resulting in blurred images of the fruits. In the preprocessing of the data, a Fuzzy image processing algorithm was used to remove the blurred images from the data set.

FPN, YOLOv3, YOLOv4, MobileNet-SSD, YOLOv4-tiny, and MobileNet-YOLOv4 models were trained on the images to detect and locate string fruits, simple fruits, and fruit branches. In the final phase, the detection results are mapped onto the optimized depth image to extract the contours and spatial information of the three targets. Based on this information, the drone can detect and locate the fruits and use them in harvesting. In general, the YOLO models had faster detection speed and achieved better accuracy than the FPN and SSD models. MobileNet-YOLOv4 achieved the best performance in terms of accuracy (mAP = 89.73) and inference time (68 s), while FPN achieved the worst performance.

To test the model in orchards, a picking platform was developed with a UAV, a Jetson TX2, an RGB-D camera, a set of scissors with clamps, and a support frame. The accuracy rate of successful harvesting in four cases was reported as 75, 75, 69.23, and 68.42.

Aghi et al. [54] presents a low-cost, energy-efficient local motion planner for autonomous navigation of robots in vineyards based on RGB-D images, low-range hardware (a low-cost device with low power and limited computational capabilities), and two control algorithms. The first algorithm uses the disparity map and its depth representation to generate proportional control for the robot platform. The second backup algorithm, based on a deep learning algorithm, takes control of the machine when the first block briefly fails and generates high-level motion primitives.

An Intel RealSense Depth Camera D435i RGB-D camera was used to capture images and compute the depth map on the platform. The video was captured in different terrain, quality, and time of day. Then, a light depth map-based algorithm processes the depth maps to detect the end of the vineyard row and then calculates control values for linear and angular velocities using a proportional controller. Since, as in many outdoor applications, sunlight negatively affects the quality of the results and interferes with the control provided by the local navigation algorithm, MobileNet was trained to classify whether the camera was pointed at the center of the end of the vineyard row or one of its sides, and it was used when the first algorithm failed due to outdoor conditions. The CNN model was trained with transfer learning and classified the images into three classes: Middle, Left, and Right. For the middle class, the video was taken in rows with the camera pointed at the center, and for the other two classes, the videos were rotated left and right with the camera at a 45-degree angle to the long axis of the row. The accuracy of the model on the test set was one. The model was trained on a small portion of the data set to investigate the significance of the size of the data set. The accuracy of the model decreased by 6% with the small data set.

Finally, the CNN model was optimized by discarding all redundant operations and reducing the floating-point accuracy from 32 to 16 bits. The accuracy of the optimized model was the same as the original and the time inference was improved. The proposed model was implemented on a robot and the tests were performed in a new vineyard scenario. The first algorithm can detect the end of the vineyard regardless of the direction of the long axis of the robot. When the first algorithm fails, the CNN model jumps in and detects the end of the vineyard.

Badeka et al. [55] trained YOLOv3 to detect grape crates in the field for use on robots harvesting grapes. The images of the crate were taken under natural field conditions. Three data enhancement techniques were used, including rotation, noise, processing, and blur processing. The model achieved an accuracy of 99.74% (mAP%) with an inference time of 0.26 s. To use the trained model on robots, it must be deployed on edge devices and report the inference time and accuracy of the model on these devices. Another interesting problem is that the robot can detect whether the box is full or not.

Majeed et al. [56] combined the DL model with mathematical models to detect cordons in grape canopies and determine their trajectories. Images were taken so that a tree trunk was approximately in the center of the field of view of a high-resolution camera (Sony Cyber-shot RX100 IV). A total of 191 images of random RGB images were captured from two different vineyard blocks with different cultivars. Each image in the dataset was classified into three classes: Background, Trunk, and Cordon (http://hdl.handle.net/2376/16939, accessed on 1 September 2021). Since the background pixels in the labeled images covered most of the images, a weight was assigned to each class in the preprocessing of the data, which was calculated using a median frequency class balancing method to avoid bias during training. Data augmentation and transfer learning techniques were used.

References

1. Kerkech, M.; Hafiane, A.; Canals, R. Vine disease detection in UAV multispectral images using optimized image registration and deep learning segmentation approach. Comput. Electron. Agric. 2020, 174, 105446.
2. Kerkech, M.; Hafiane, A.; Canals, R. Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. Comput. Electron. Agric. 2018, 155, 237–243.
3. Barbedo, J.G.A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Comput. Electron. Agric. 2018, 153, 46–53.
4. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. 2018, 145, 311–318.
5. Krizhevsky, A. One Weird Trick for Parallelizing Convolutional Neural Networks. 2014. Available online: http://xxx.lanl.gov/abs/1404.5997 (accessed on 1 November 2021).
6. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. arXiv 2013, arXiv:1312.6229.
7. Jiang, P.; Chen, Y.; Liu, B.; He, D.; Liang, C. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. IEEE Access 2019, 7, 59069–59080.

8. Karthik, R.; Hariharan, M.; Anand, S.; Mathikshara, P.; Johnson, A.; Menaka, R. Attention embedded residual CNN for disease detection in tomato leaves. Appl. Soft Comput. 2020, 86, 105933.

9. Liu, B.; Zhang, Y.; He, D.; Li, Y. Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks. Symmetry 2018, 10, 11.

10. Ramcharan, A.; McCloskey, P.; Baranowski, K.; Mbilinyi, N.; Mrisho, L.; Ndalahwa, M.; Legg, J.; Hughes, D.P. A Mobile-Based Deep Learning Model for Cassava Disease Diagnosis. Front. Plant Sci. 2019, 10, 272.

11. Picon, A.; Seitz, M.; Alvarez-Gila, A.; Mohnke, P.; Ortiz-Barredo, A.; Echazarra, J. Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. Comput. Electron. Agric. 2019, 167, 105093.

12. Chen, J.; Zhang, D.; Zeb, A.; Nanehkaran, Y.A. Identification of rice plant diseases using lightweight attention networks. Expert Syst. Appl. 2021, 169, 114514.

13. Silver, D.L.; Monga, T. In Vino Veritas: Estimating Vineyard Grape Yield from Images Using Deep Learning. In Advances in Artificial Intelligence; Meurs, M.J., Rudzicz, F., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 212–224.

14. Aguiar, A.S.; Magalhães, S.A.; dos Santos, F.N.; Castro, L.; Pinho, T.; Valente, J.; Martins, R.; Boaventura-Cunha, J. Grape Bunch Detection at Different Growth Stages Using Deep Learning Quantized Models. Agronomy 2021, 11, 1890.

15. Aguiar, A.S.; Monteiro, N.N.; Santos, F.N.d.; Solteiro Pires, E.J.; Silva, D.; Sousa, A.J.; Boaventura-Cunha, J. Bringing Semantics to the Vineyard: An Approach on Deep Learning-Based Vine Trunk Detection. Agriculture 2021, 11, 131.

16. Ghiani, L.; Sassu, A.; Palumbo, F.; Mercenaro, L.; Gambella, F. In-Field Automatic Detection of Grape Bunches under a Totally Uncontrolled Environment. Sensors 2021, 21, 3908.

17. Seng, K.P.; Ang, L.M.; Schmidtke, L.M.; Rogiers, S.Y. Computer Vision and Machine Learning for Viticulture Technology. IEEE Access 2018, 6, 67494–67510.

18. Milella, A.; Marani, R.; Petitti, A.; Reina, G. In-field high throughput grapevine phenotyping with a consumer-grade depth camera. Comput. Electron. Agric. 2019, 156, 293–306.

19. Santos, T.T.; de Souza, L.L.; dos Santos, A.A.; Avila, S. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. Comput. Electron. Agric. 2020, 170, 105247.

20. Palacios, F.; Bueno, G.; Salido, J.; Diago, M.P.; Hernández, I.; Tardaguila, J. Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions. Comput. Electron. Agric. 2020, 178, 105796.

21. Millán, B.; Aquino, A.; Diago, M.P.; Tardáguila, J. Image analysis-based modelling for flower number estimation in grapevine. J. Sci. Food Agric. 2017, 97 3, 784–792.

22. Kang, H.; Chen, C. Fruit detection, segmentation and 3D visualisation of environments in apple orchards. Comput. Electron. Agric. 2020, 171, 105302.

23. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. Precis. Agric. 2019, 20, 1107–1135.

24. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. 2018. Available online: http://xxx.lanl.gov/abs/1804.02767 (accessed on 1 September 2021).

25. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.

26. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agric. 2019, 157, 417–426.

27. Zhou, Z.; Song, Z.; Fu, L.; Gao, F.; Li, R.; Cui, Y. Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation. Comput. Electron. Agric. 2020, 179, 105856.

28. Bah, M.D.; Hafiane, A.; Canals, R. Deep Learning with Unsupervised Data Labeling for Weed Detection in Line Crops in UAV Images. Remote Sens. 2018, 10, 1690.

29. Ferreira, A.D.S.; Freitas, D.M.; da Silva, G.G.; Pistori, H.; Folhes, M.T. Unsupervised deep learning and semi-automatic data labeling in weed discrimination. Comput. Electron. Agric. 2019, 165, 104963.

30. Ferreira, A.D.S.; Freitas, D.M.; da Silva, G.G.; Pistori, H.; Folhes, M.T. Weed detection in soybean crops using ConvNets. Comput. Electron. Agric. 2017, 143, 314–324.

31. Olsen, A.; Konovalov, D.A.; Philippa, B.; Ridd, P.; Wood, J.C.; Banks, W.; Girgenti, B.; Kenny, O.; Whinney, J.; Calvert, B.; et al. DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning. Sci. Rep. 2019, 9, 2058.

32. Milioto, A.; Lottes, P.; Stachniss, C. Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2229–2235.

33. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 2481–2495.

34. Lottes, P.; Behley, J.; Milioto, A.; Stachniss, C. Fully Convolutional Networks with Sequential Information for Robust Crop and Weed Detection in Precision Farming. IEEE Robot. Autom. Lett. 2018, 3, 2870–2877.

35. Wang, A.; Xu, Y.; Wei, X.; Cui, B. Semantic Segmentation of Crop and Weed using an Encoder-Decoder Network and Image Enhancement Method under Uncontrolled Outdoor Illumination. IEEE Access 2020, 8, 81724–81734.

36. Rußwurm, M.; Körner, M. Multi-Temporal Land Cover Classification with Long Short-Term Memory Neural Networks. ISPRS 2017, 42W1, 551–558.

37. Lee, S.H.; Chan, C.S.; Mayo, S.J.; Remagnino, P. How deep learning extracts and learns leaf features for plant classification. Pattern Recognit. 2017, 71, 1–13.

38. Ayhan, B.; Kwan, C.; Budavari, B.; Kwan, L.; Lu, Y.; Perez, D.; Li, J.; Skarlatos, D.; Vlachos, M. Vegetation Detection Using Deep Learning and Conventional Methods. Remote Sens. 2020, 12, 2502.

39. Skarlatos, D.; Vlachos, M. Vegetation Removal From Uav Derived Dsms, Using Combination of Rgb and Nir Imagery. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 2018, IV-2, 255–262.

40. Bhusal, S.; Bhattarai, U.; Karkee, M. Improving Pest Bird Detection in a Vineyard Environment using Super-Resolution and Deep Learning. IFAC-PapersOnLine 2019, 52, 18–23.

41. Yamanaka, J.; Kuwashima, S.; Kurita, T. Fast and Accurate Image Super Resolution by Deep CNN with Skip Connection and Network. In Neural Information Processing; Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S.M., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 217–225.

42. Mac Aodha, O.; Gibb, R.; Barlow, K.E.; Browning, E.; Firman, M.; Freeman, R.; Harder, B.; Kinsey, L.; Mead, G.R.; Newson, S.E.; et al. Bat detective—Deep learning tools for bat acoustic signal detection. PLoS Comput. Biol. 2018, 14, e1005995.

43. Ramalingam, B.; Mohan, R.E.; Pookkuttath, S.; Gómez, B.F.; Sairam Borusu, C.S.C.; Wee Teng, T.; Tamilselvam, Y.K. Remote Insects Trap Monitoring System Using Deep Learning Framework and IoT. Sensors 2020, 20, 5280.

44. Verdouw, C.; Wolfert, S.; Tekinerdogan, B. Internet of Things in agriculture. CAB Rev. 2016, 11, 1–12.

45. Li, C.; Zhang, Y.; Ren, X. Modeling Hourly Soil Temperature Using Deep BiLSTM Neural Network. Algorithms 2020, 13, 173.

46. Yu, F.; Hao, H.; Li, Q. An Ensemble 3D Convolutional Neural Network for Spatiotemporal Soil Temperature Forecasting. Sustainability 2021, 13, 9174.

47. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.c. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Advances in Neural Information Processing Systems; Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2015; Volume 28.

48. Saggi, M.K.; Jain, S. Reference evapotranspiration estimation and modeling of the Punjab Northern India using deep learning. Comput. Electron. Agric. 2019, 156, 387–398.

49. Zhang, J.; Zhu, Y.; Zhang, X.; Ye, M.; Yang, J. Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. J. Hydrol. 2018, 561, 918–929.

50. Loggenberg, K.; Strever, A.; Greyling, B.; Poona, N. Modelling Water Stress in a Shiraz Vineyard Using Hyperspectral Imaging and Machine Learning. Remote Sens. 2018, 10, 202.

51. Chen, M.; Cui, Y.; Wang, X.; Xie, H.; Liu, F.; Luo, T.; Zheng, S.; Luo, Y. A reinforcement learning approach to irrigation decision-making for rice using weather forecasts. Agric. Water Manag. 2021, 250, 106838.

52. Alibabaei, K.; Gaspar, P.D.; Assunção, E.; Alirezazadeh, S.; Lima, T.M. Irrigation optimization with a deep reinforcement learning model: Case study on a site in Portugal. Agric. Water Manag. 2022, 263, 107480.

53. Li, D.; Sun, X.; Elkhouchlaa, H.; Jia, Y.; Yao, Z.; Lin, P.; Li, J.; Lu, H. Fast detection and location of longan fruits using UAV images. Comput. Electron. Agric. 2021, 190, 106465.

54. Aghi, D.; Mazzia, V.; Chiaberge, M. Local Motion Planner for Autonomous Navigation in Vineyards with a RGB-D Camera-Based Algorithm and Deep Learning Synergy. Machines 2020, 8, 27.

55. Badeka, E.; Vrochidou, E.; Papakostas, G.A.; Pachidis, T.; Kaburlasos, V.G. Harvest Crate Detection for Grapes Harvesting Robot Based on YOLOv3 Model. In Proceedings of the 2020 Fourth International Conference On Intelligent

Computing in Data Sciences (ICDS), Fez, Morocco, 21–23 October 2020; pp. 1–5.

56. Majeed, Y.; Karkee, M.; Zhang, Q.; Fu, L.; Whiting, M.D. Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks. Comput. Electron. Agric. 2020, 171, 105308.

Retrieved from https://encyclopedia.pub/entry/history/show/49664