



# Evaluating the effect of super-resolution for automatic plant disease detection: application to potato late blight detection

Mazari Sarah<sup>1</sup> · Meddoudi Abdlemadjid<sup>1</sup> · Benameur Sarah<sup>1</sup> · Hadjadj Yacine<sup>1</sup> · Chikr El Mezouar Miloud<sup>1</sup>

Received: 12 October 2023 / Revised: 23 January 2024 / Accepted: 30 January 2024 /

Published online: 26 February 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

## Abstract

Precise and automatic plant disease detection represent a fundamental research topic. Indeed, traditional or manual disease detection can be laborious, inaccurate and time-consuming. In this work, a particular attention is given to automatic potato late blight disease detection. In fact, this devastating pathogen leads every year to a significant reduction in potato yields. In addition, since potatoes are a major food source for many nations, decreased production generate a real food insecurity. Therefore, considering these challenges, using advanced techniques in computer vision and machine learning allowed farmers to swiftly and accurately identify disease. The main objective of this research work is to generate a super-resolved labeled dataset (SRD) and evaluate its impact on plant disease detection. Three states of the art object detection methods (Faster-RCNN, Detr and Yolo V8) have been used to conduct an exhaustive evaluation on the effect of using a super-resolved dataset to perform detection. The obtained results show that the detection of potato late blight disease is enhanced using SRD. Training Yolo V8 model on SRD outperform other trained models in detecting very small lesions. In fact, training Yolo V8 on SRD reached higher mAP, lower Loss values and a reasonable inference time, making it suitable for real time applications.

**Keywords** Deep learning · Object detection · Super-resolution · Late blight disease

---

✉ Mazari Sarah  
mazari\_sarah@yahoo.fr

Meddoudi Abdlemadjid  
abdlemadjidmeddoudi@gmail.com

Benameur Sarah  
sarah\_mail14@yahoo.fr

Hadjadj Yacine  
mail@ycinhdj.com

Chikr El Mezouar Miloud  
chikrelmezouar@gmail.com

<sup>1</sup> Djillali Liabes University of Sidi Bel Abbas, BP89, Cité Larbi BenMhidi, 22000 Sidi Bel Abbas, Algeria

## 1 Introduction

Currently, more than 60% of yield losses in agriculture are due to plant diseases. In order to identify plant diseases, traditional agricultural management practices use plant pathology technique, namely methods related to the chemical analysis of the infected area of the plant, and physical methods such as spectroscopy. These techniques obviously require qualified people intervention and can be very expensive. So, developing automated and low-cost plant disease detection methods becomes imperative. The majority of automated methods proposed in the literature are based on computer vision and artificial intelligence techniques. In fact, software-based products can perform vision-based tasks with greater accuracy than human experts. The main objective of automatically detecting plant disease is to rapidly and accurately diagnose plant disease. This process involves capturing images of plants (plants leaves), analyzing them and then providing a diagnosis. However, vision-based plant disease detection algorithms are faced with many challenges such as complexity of disease, lighting variations, low resolution of acquired image, lack of labeled data (for training accurate machine/deep learning model). This article proposes to evaluate effect of super-resolved plant disease images for early automatic and precise detection of potato blight disease (caused by *phytophthora infestans*). In fact, increasing resolution of images is an important processing step, allowed to obtain precise and reliable data, offering advantages in several areas such as vegetation/urban detection, precision agriculture, land use/land cover, change detection and in many other fields of computer vision applications. The main objective of super-resolution techniques is to generate a higher resolution image (HR) from lower resolution images (LR). In this work, a new method is developed, that allow to accurately detect disease on plant leaf. This method involves Super-resolved plant leaf images dataset by training EDSR model on the healthy/diseased plant leaves dataset. Then training three recent object detection models in order to detect late blight disease on plant leaf diseased images, using standard and super-resolved dataset.

## 2 Related works

Detecting diseases of plant leaf from images use traditional image processing technique and advanced one. Traditional classical image processing technique analyze visually observable patterns seen on the plant. Segmentation-based methods determine the difference in the color, texture and shape of the affected area, by dividing plant leaf image into different regions then analyzing each region for signs of disease. These methods can also be used to identify the type of disease present, as well as to monitor the progression of the disease over time. Dey et al. [1] describe the process of capturing images of the leaves, pre-processing the images to remove noise and enhance contrast then implement Otsu thresholding-based image processing for segmentation of leaf rot diseases in betel vine leaf. Features extraction methods involves extracting various features from plant leaves images. These features can then be used as input of classification algorithms to classify plant as healthy or diseased. The authors in [2] select the most relevant features using fuzzy feature selection technique (Fuzzy curve and Fuzzy surface) to improve classification accuracy.

Advanced techniques mainly use deep learning architecture. Neural networks models can be trained on a large dataset of plant leaves images to learn the features that differentiate healthy and diseased plants. The model can then be used to make prediction on new

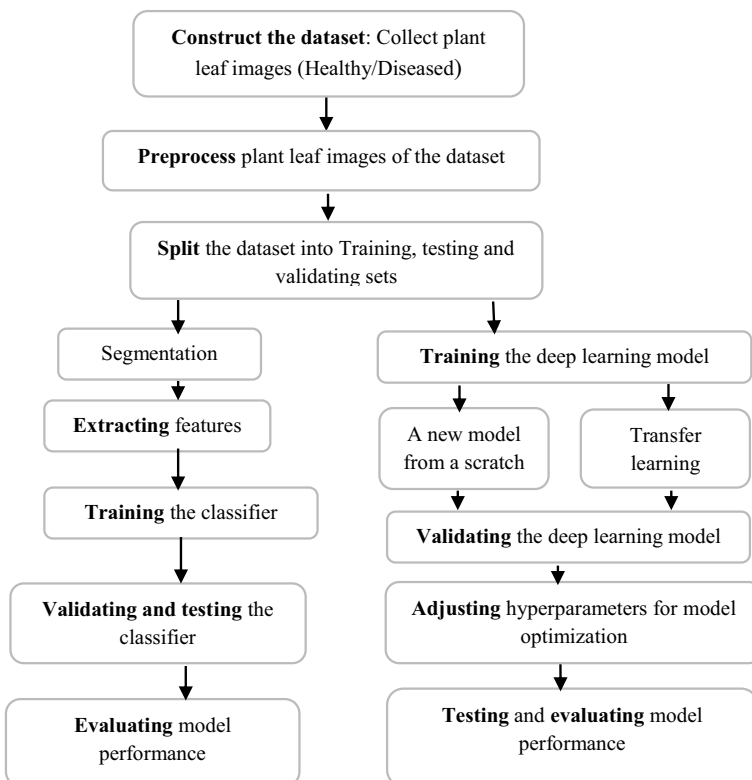
plant leaves images. Figure 1 shows the required steps for automatic identification of plant diseases using machine/deep learning approaches.

A summary of most recent proposed approaches for automatic plant disease classification and detection are shown in Table 1.

From Table 1, we can observe that deep learning-based technique model outperform segmentation, feature extraction and classification techniques. Segmentation based techniques present a smaller computational complexity but don't consider low contrast and defaults illumination of acquired image, so hardly achieve good plant disease detection results.

In [3, 5, 7], the authors used SVMs to classify leaf images as healthy or infected with various types of diseases, SVMs are generally used after different features extraction steps. However, SVMs require hand-crafted features, which can be time-consuming and limit the model's ability to learn complex features. SVMs are also combined with deep learning models to recognize four types of rice diseases [11], using the strengths of both approaches allow the authors to achieve high accuracy.

Deep learning and specifically Convolutional Neural Networks (CNNs) architecture are commonly used in plant disease classification task. Ferentinos et al. [8] fine tune five different CNNs architectures (AlexNet, GooLeNet, VGG-16, ResNet50, InceptionV3) on new plant disease dataset. The authors compared their performance in terms of accuracy,



**Fig. 1** Automatic identification of plant disease

**Table 1** Summary of most recent proposed approaches for automatic plant disease classification and detection

Authors -years	Description	Dataset	Accuracy/Precision
Mokhtar et al.- 2015 [3]	Each input image is segmented and descriptor created for each segment. Some geometric measurements are employed to identify an optimal feature subset. Support vector machine (SVM) algorithm with different kernel functions is used for classification	200 infected tomato leaf images	90%
Mohanty et al. – 2016 [4]	A deep convolutional neural network is trained to identify 14 crop species and 26 diseases	PlantVillage	99.35%
Prajapati et al. [5] (2017)	Features extraction is performed using centroide feeding based K-means clustering for segmentation. Three categories of features are extracted: color, shape and texture. SVM is then used for multi-class classification	120 captured images of rice leaf	80%
Fuentes et al. [6] (2017)	Using deep-learning-based approach combined with three detectors (Faster-RCNN- (R-FCN), and (SSD) to detect diseases and pests in tomato plants	5000 images from farm	83%
Kaur et al. – 2018 [7]	Using colour features, texture features, and their combinations to train three models based on SVM classifier	Plant village	90%
Ferentinos et al. [8] 2018	Comparing between five deep learning CNN models (Alexnet Alexnet OWTBn,GoogLeNet, Overfeat and VGG)	Open database of 87,848 image	99.53%
Singh et al. [9] 2019	A multilayer CNN is used to identify mango leaf diseases	1070 images of the Mango tree leaves	97.13%
Lin, Z. et al. [10] 2019	A unified convolutional neural network (CNN) denoting the matrix-based convolutional neural network (M-bCNN) was proposed	83,260 augmented images of winter wheat leaf diseases	90.1%
Sethy et al. [11]-2020	Four types of rice leaf diseases are identified using transfer learning and deep feature plus SVM	5932 on-field images of four types of rice disease	98.38%
Shrivastava et al. – 2021 [12]	14 different color spaces are used, 4 features are extracted from each color channel leading to 172 features. SVM method is used for classification	619 images collected from the real agriculture	94.65%
Atila et al. [13] -2021	B5 and B4 models of EfficientNet architecture achieved the highest values compared to transfer learning models in original and augmented datasets	PlantVillage	99.1%

**Table 1** (continued)

Authors -years	Description	Dataset	Accuracy/Precision
Zhang et al. [14]-2022	A Transvolution detection network is proposed, different GAN modules are incorporated with the CNN	PlantDoc,	51.7%
Ashwinkumar et al. [15]-2022	Optimal Mobile Network-based convolutional neural network (OMNCNN) is performed: the hyperparameters are optimized by the use of emperor penguin optimizer (EPO) algorithm and extreme learning machine (ELM) based classifier is utilized to allocate proper class labels to the applied plant leaf images	Not mentioned	98%
Borhani et al. [16]-2022	A combination of CNN and Vision Transformer have been implemented for the plant disease classification	-Wheat Rust Classification dataset - Rice Leaf Disease dataset - Plant Village	98% 99% 100%
Liu et al. [17]-2023	Proposition of a re-parameterization method to improve the boosting accuracy of convolutional neural networks and introduction of a dynamic pruning gate to dynamically control the network structure, enabling operation on hardware platforms with widely varying computational power;	collection of datasets for four crops with a total of 12 diseases over a three-year history	94%
Sarawast et al. [18]-2023	Using the artificial intelligence techniques like neural network model and dynamic SURF method in order to classify the plant diseases for fungus and bacteria	PlantDoc	99.5%

precision and recall and found that VGG-16 achieved the highest accuracy. When fine-tuning a pretrained model, the weights of the model are updated during training on a new dataset. By doing this, the new model can benefit from the feature extraction capabilities of the pretrained model. Fine tuning a pretrained model has been shown to improve accuracy and reduce the amount of time and resources required for training. Ashwinkumar et al. [15] propose a modified CNN-based model and introduce a new optimization technique called emperor penguin optimizer (EPO), to prevent overfitting and improve the generalization performance of the model. Their method outperforms other CNN-based model in terms of both accuracy and computational efficiency. Zhang et al. [14] improved CNN network by adding two generative networks models and use a modified Vision transformer network (ViT) to improve CNNs ability to capture global feature as a branch network. The model has a low inference speed but achieve high precision. Borhani et al. [16] compared between a simplified version of the ViT model, a CNN-based model and the hybrid model which included the combination of the CNN and ViT. Higher accuracy was achieved by ViT based model, however, an improvement in speed prediction was attempt by combining the attention blocks with convolutional blocks. Recently, a deep neural network based on an automatic pruning mechanism is developed to enable high-accuracy plant disease detection even under limited computational power was proposed by Liu et al. [17]. Sarawast et al. [18] proposed (Modified-MDNN) algorithm by using DSURF(Dynamic-SURF) features and also shows the comparative analysis between DNN and MDNN performance.

It is important to note that the performance of all machine learning proposed methods depends on the diversity of the dataset used for training and testing the model. In fact, the dataset should be diverse and representative of the type of desired disease recognition. Increasing diversity of training dataset is necessary especially when large datasets aren't available. This process is performed by data augmentation. Shorten et al. [19] make a survey on different data augmentation technique proposed in the literature. They present existing methods for Data Augmentation (color space augmentations, geometric transformations, kernel filters, mixing images, random erasing, feature space augmentation, Noise injection, Adversarial training, neural style transfer), promising developments, and meta-level decisions for implementing Data Augmentation.

### 3 Methodology

In this section, we describe the proposed methodology to perform plant disease detection. We firstly super resolve dataset using enhanced deep super-resolution network architecture (EDSR) and then perform detection and recognition steps using Faster R-CNN, YoloV8 and Detr models. A short description of super-resolution concept, EDSR architecture, Faster R-CNN, YoloV8 and Detr architectures is given below.

#### 3.1 Super-resolution

Super-resolution methods use information from several different images (multi-image super-resolution (MISR)) or only one image (single-image super resolution (SISR)) to create one upsized image. Three categories of SISR approaches can be distinguished: reconstruction-based approaches, use a priori information to recover the HR images; interpolation-based approach that reconstruct HR images using existing pixels to interpolate missing pixels, learning-based approaches use dictionary pairs of training and testing

images to predict HR images [20]. The recent developments in deep learning have furnish many tools to resolve SISR problem. Several deep learning architectures like CNNs [21], Residual CNNs [22] GANs [23] and DNNs [24] have been proposed to reconstruct high resolution image from low resolution image. Multi-image super-resolution approaches expect to reconstruct high-resolution details using multiple low-resolution images of the same scene (acquired from the same or different sensors). Deep learning based MISR methods achieved satisfactory results [25, 26] but depends on the fidelity of the training datasets. Optimization-based iterative MISR methods solve the problem in the spatial and frequency domains [27]. In frequency domain, multiple images are combining with sub-pixel displacements to enhance resolution. Due to difficulties to use a priori knowledge as regularization, many spatial domain MISR techniques were proposed. These include the maximum likelihood (ML), the maximum a posteriori (MAP), and the projection onto convex sets (POCS) [28, 29].

### 3.2 Edge enhanced deep super- resolution network

A particular attention is given to SISR learning methods. Recent comparative studies [30, 31] identified the Enhanced deep super-resolution network architecture (EDSR) proposed by Lim et al. [32] and the Enhanced super-resolution generative adversarial networks (ESRGAN) proposed by [33] as competitive approaches in the field of super-resolution task. While ESRGAN architecture is well- known for its capability to reconstruct fine details and textures and enhance perceptual quality, it may introduce artificial details which is undesirable in the domain where precision and accuracy are imperative. Furthermore, ESRGAN is a GAN based model that requires careful training process (leading to a significant challenge when computational resources are limited). In contrast, the training process in EDSR architecture is simple, it requires only a single loss function to optimize. EDSR architecture is relatively intuitive and represents an optimization of super resolution residual network architecture (SRResNet) [34]. Unnecessary modules are removed to simplify the network. In fact, it requires only a single loss function to optimize. The authors in [32] found that removing Batch normalization allow to save 40% of memory usage during training compared to SRResNet. Moreover, EDSR enhance image resolution while maintaining high fidelity to the original image (Better PSNR and SSIM values compared to ESRGAN super-resolution results). This is crucial in plant disease identification context where accurate representation of vein patterns, disease lesions and edges are essential.

### 3.3 Region-convolutional neural network (R-CNN)

Region with CNN features (R-CNN model) are a popular object detection and localization framework developed by Ren S et al. [35]. R-CNN architecture involves three modules. The first generates category-independent region proposals. The second extract features from each proposed region using pretrained CNN. The last module ensures object classification and bounding box regression. R-CNN suffers from several limitations such as slow inference speed, high memory consumption and multistage training. These limitations have motivated the development of optimized version of R-CNN such as Fast-R-CNN [36], Faster-R-CNN [37] and Mask-R-CNN [38]. Fast R-CNN uses a single CNN to extract features from the entire image and then generates region proposals from the feature map. Fast R-CNN model consists of a single-stage compared to three stages in R-CNN. Fast R-CNN samples multiple ROIs from the same image by using a new layer called ROI pooling that

extracts equal-length feature vectors from all proposals. This makes Fast- R-CNN detection model faster and more efficient. Faster R-CNN is an extension of fast R-CNN. The architecture of Faster R-CNN consists of 2 modules:

- Region Proposal Network (RPN) is built for generating region proposals.
- Fast R-CNN is used for detecting objects in the proposed region

### 3.4 Detection transformer model (Detr)

Detr perform prediction and generate the final set of detections by combining a common CNN with a transformer architecture. The authors in [39] propose to resolve object detection task as a set prediction problem and use transformers and bipartite matching loss. Detr object detection model uses self-attentional transformers to process input images, producing a fixed set of detection directly, without using proposal networks (RPNs) and non-maximum suppression. It is important to note that the transformer architecture used in Detr is computationally intensive and training large Detr models consume a large amount of memory.

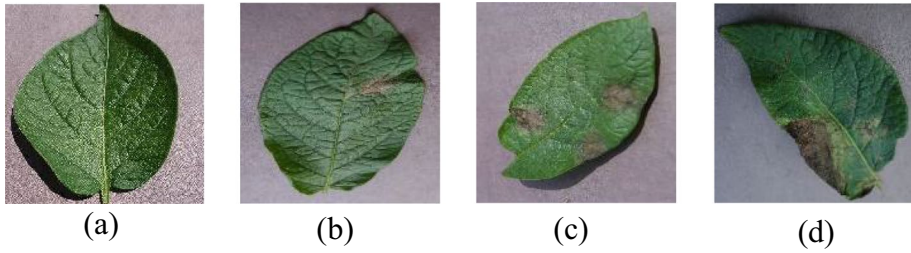
### 3.5 You only look one model (Yolo)

Yolo detection model [40] addresses object detection as a single-stage, end-to-end process. It uses a simple deep convolutional neural network to detect objects in the image. The input image is divided up into a grid of cells. The grid size is typically determined by the architecture of the YOLO model. After that, Yolo predicts bounding boxes that contain the objects found in each grid cell. The bounding box values ((x, y) coordinates of the box's center, width (w), and height (h)) are relative to the cell's size. A confidence score is then added to each bounding box to indicate the presence of an object. A class probability is also assigned using SoftMax activation. Only the most confident bounding box with high probability are selected. This model has 24 convolution layers, 4 max-pooling layers, and 2 fully connected layers. 1\*1 convolution followed by 3\*3 convolution are used to reduce number of channels. Yolo predicts multiple bounding boxes for each grid cell and filters them based on their confidence score and overlap with the anchor box. There are several versions of Yolo, the most recent one being YoloV8 [41]. YoloV8 architecture focuses on performance and speed achieving cutting-edge results on a number of benchmarks. YoloV8 predicts the center of an object directly instead of predicting the offset from predefined anchor boxes. This is called anchor free detection and allows reducing the number of box prediction. YoloV8 uses Mosaic augmentation (stitching four images together) and augments images during training online. At each epoch, the model sees a slightly different variation of the images it has been provided.

### 3.6 Proposed method

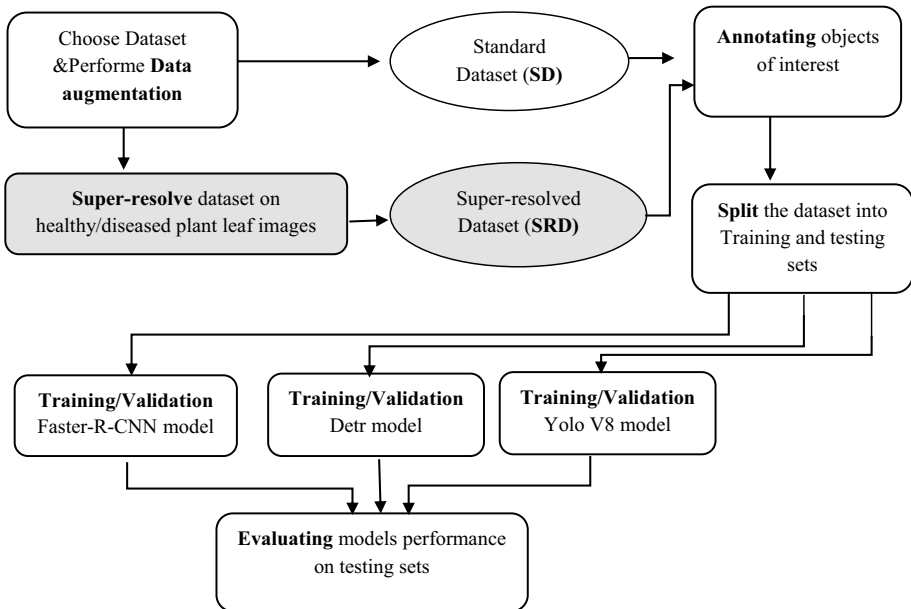
The proposed method was tested on potato plant leaf to detect Late Blight disease. Model detection training process was release using the plant leaf images taken from the popular PlantVillage dataset (<https://www.kaggle.com/datasets/emmarex/plantdisease>). Figure 2 displays examples of potato plant leaves images. Figure 2(a) shows a healthy potato plant leaf, while Fig. 2(b), (c) and (d) show various forms of late blight lesions on potato plant leaves.





**Fig. 2** Healthy and diseased potato plant leaves images. (a) Healthy potato plant leaf, (b), (c) and (d) infected potato plant leaves

In order to diversify and increase the training set, the potato plant leaf (Healthy and diseased) dataset was firstly augmented using geometric and color space transformations. A pretrained EDSR model for transfer learning was then used to upscale images by factor 4x. The evaluation of the effect of super-resolution on plant leaf disease detection is done by training three object detection models with transfer learning on the two datasets (Standard Dataset (SD) and super-resolved dataset SRD). Typically, training Faster-RCNN, Detr and Yolo V8 object detection models requires input images to be of a specific size for optimal processing. In this work, the Standard dataset contains potato plant leaves images (healthy and diseased) of size  $256 \times 256$ . The Super-resolved dataset contains potato plant leaves images (healthy and diseased) upscaled to  $1024 \times 1024$ . The Super-resolved dataset (SRD) and the standard dataset (SD) are then split into training set (70%), testing set (20%) and validation set (10%). The block diagram of the proposed method is depicted in Fig. 3.



**Fig. 3** Block diagram of the proposed method

## 4 Experiments and results

### 4.1 Hyperparameters and setting

All training experiments were run on a machine with 10 GB RTX 3080 GPU, 10th generation i7 CPU, and 32 GB of RAM. Faster R-CNN, Detr and YOLOV8 training process are performed on PyTorch framework. The learning rate is set to 0.0001 and the batch size to 8 with Adam optimizer. The smooth L1 is used for bounding box regression tasks. In this work, Faster R-CNN use ResNet50 as backbone network [39], region proposal consists of convolutional layers followed by fully connected layers and use the feature map coming from the backbone network as input. Detr use the ResNet-101 architecture as the backbone network [42] and incorporates deformable convolutions networks (DCN). Yolo V8 use CSPDarkNet as backbone network [43]. The size of SD and SRD datasets is 918 before data augmentation process and 2204 after data augmentation process.

### 4.2 Metrics evaluation

The quality assessment of super-resolution results is generally performed using PSNR and SSIM metrics. It describes the similarity between the predicted (upscaled) image and the ground truth image.

- Peak signal to noise ratio (PSNR)

The PSNR is calculated using the Mean-Square-Error (MSE) of the pixels and the maximum possible pixel value ( $MAX_I$ ) as follows [44]:

$$PSNR(x, y) = 10. \log \left( \frac{MAX(x, y)^2}{MSE(x, y)} \right)$$

- Structural Similarity Index Measure (SSIM)

The SSIM measure the similarity between two images. SSIM is designed to improve PSNR metric, which have proven to be inconsistent with human eye perception.

$$SSIM = \frac{(2 \cdot \bar{x} \cdot \bar{y} + C_1)(2 \cdot \sigma_{xy} + C_2)}{\left( (\bar{x})^2 + (\bar{y})^2 + C_1 \right) + (\sigma_x^2 + \sigma_y^2 + C_2)}$$

where  $\sigma_x$ ,  $\sigma_y$  are the variance of x and y respectively,  $\sigma_{xy}$  is the covariance of x and y.  $C_1 = (k_1 L)^2$  and  $C_2 = (k_2 L)^2$  are two variables used to stabilize the division with weak denominator; and L represents the dynamic range of the pixel values. The common values for  $k_1$  and  $k_2$  are 0.01 and 0.03, respectively [45]

The evaluation of object detection results, in our case Late Blight disease detection is doing by calculating some metrics [46]. The detection results are compared to the manually labeled data (ground truth annotations) to determine how accurately the detection method performed.

- Precision/Recall

Precision measures the proportions of correctly predicted bounding boxes among all predicted bounding boxes.

$$\textit{Precision} = \frac{TP}{TP + FP}$$

Recall measures the proportion of correctly predicted bounding boxes among all ground truth bounding boxes.

$$\textit{Rcall} = \frac{TP}{TP + FN}$$

- mean Average Precision (mAP50-95/mAP 50)

mAP 50–95 refers to the mAP computed over a range of intersection over Union (IoU) thresholds from 0.5 to 0.95. IoU measures the overlap between the predicted bounding boxes and the ground truth bounding boxes. To compute mAP 50–95, the precision and recall values are calculated at each IoU threshold within the range.

mAP50 refers to the mAP at IoU threshold of 0.5 mAp 50 is used as a standard benchmark to evaluate the performance of object detection models. It provides a measure of how well the model can detect objects with a reasonable level of overlap with the ground truth.

$$\textit{IoU} = \frac{\textit{Area of Overlap}}{\textit{Area of Union}}$$

AP is calculated across a set of IoU thresholds for each class  $k$  and then take the average of all AP values.

$$\textit{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} \textit{AP}_k$$

$\textit{AP}_k$  is the AP of class and  $n$  the number of classes.

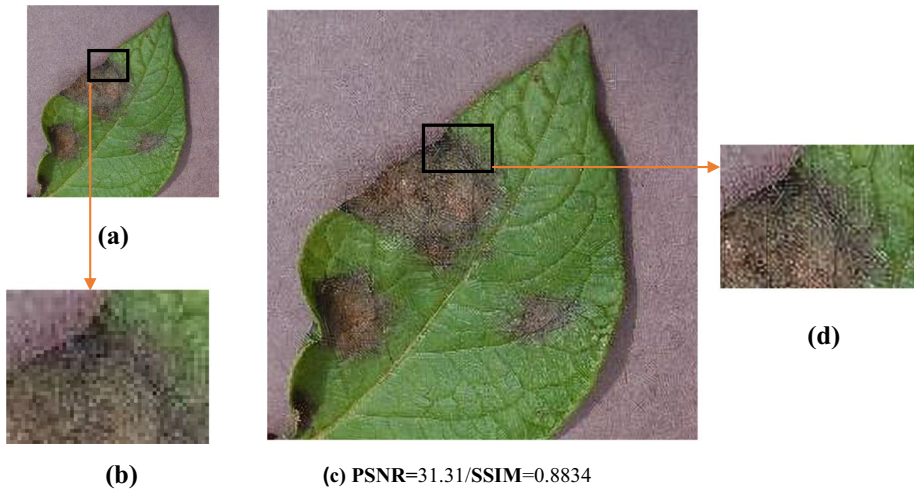
- Training/validation loss box

The training and validation loss is usually used to diagnose the model's performance and identify which aspects need tuning. Training and validation Loss Box measure the difference between the predicted bounding box coordinates and the ground truth bounding box coordinates. Common loss functions include mean squared error and smooth L1.

### 4.3 Results and performance analysis

Figure 4 shows an example of potato plant leaf infected by late blight disease. Figures (a) and (b) shows the original images. Figure (c) and (d) shows the corresponding super-resolved images (Upscaled image by factor  $\times 4$ ).

A zoomed-in patches of the original image and super-resolved one are displayed in Fig. 4. The PSNR and SSIM value are mentioned at the bottom of figure (c). From the obtained results, we can see that the EDSR architecture allow a good preservation of high frequency details and achieve a high SSIM and PSNR values.



**Fig. 4** Infected potato plant leaf: (a) Example from SD dataset, (c) Example from SRD dataset

Table 2 summarizes the obtained automatic detection results. Each model was run for 50 epochs. Performance assessment of the proposed method is performed by comparing mAP, Precision and Recall of the six trained models:

- Train and validate Faster-RCNN on SD and SRD
- Train and validate Detr on SD and SRD
- Train and validate Yolo V8 on SD and SRD

From Table 2, it can be observed that training detection models on super-resolved dataset (SRD) improve the detection results. In fact, Faster-RCNN SRD, Detr SRD and Yolo V8 SRD achieve high mAP at 0.5 IoU threshold compared to Faster-RCNN ND, Detr SD and Yolo V8 SD. Yolo V8 SRD model outperform all the other trained models in term of mAP, precision and Recall.

The performance of the proposed methods is also performed by analyzing the training and validation process for the six models, represented by mAP 50–95 validation curves (Fig. 5) and by Training/Validation Loss Box curves (Fig. 6).

**Table 2** Performance comparison of the six detection models

Model	mAP 50	Precision	Recall
Faster-RCNN SD	0.76	0.71	0.69
Faster-RCNN SRD(Ours)	<b>0.84</b>	<b>0.81</b>	<b>0.82</b>
Detr SD	0.75	0.69	0.67
Detr SRD (Ours)	<b>0.82</b>	<b>0.77</b>	<b>0.80</b>
Yolo V8 SD	0.88	0.84	0.86
Yolo V8 SRD (Ours)	<b>0.94</b>	<b>0.90</b>	<b>0.88</b>

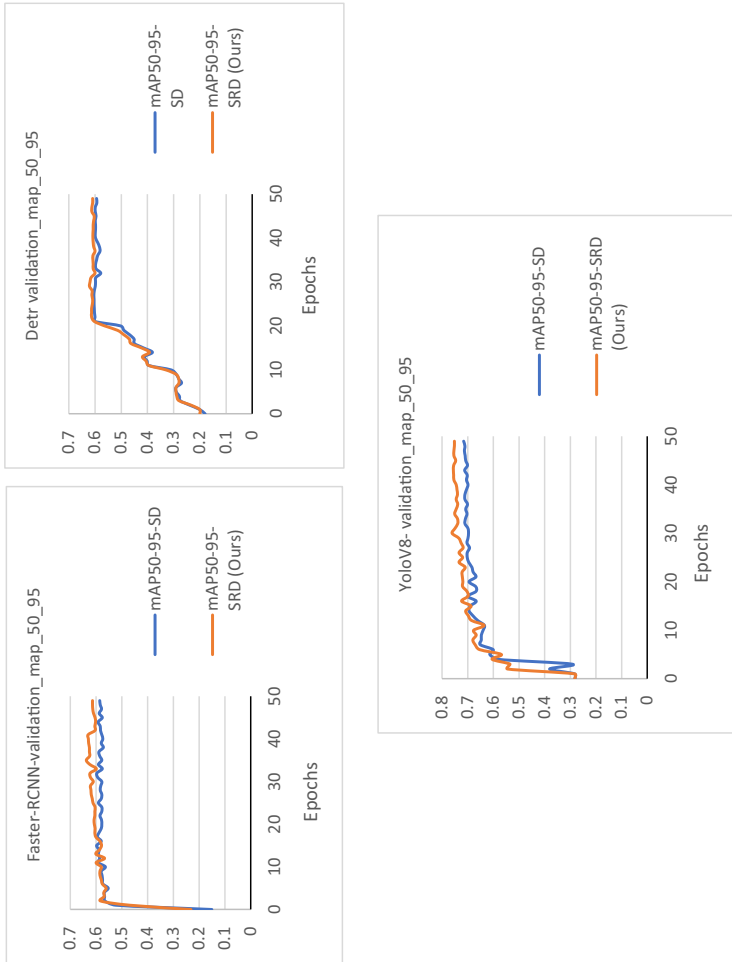
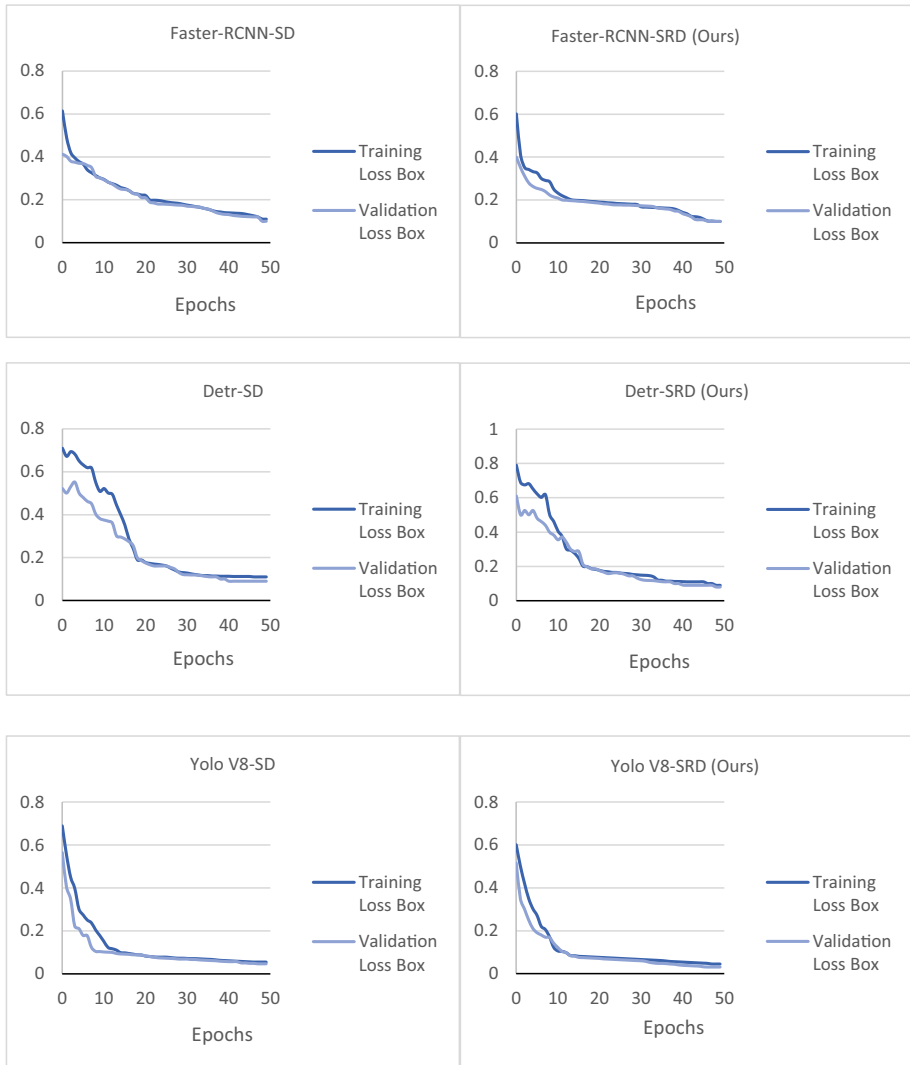


Fig. 5 Mean average precision curves



**Fig. 6** Loss curves for training and validation data

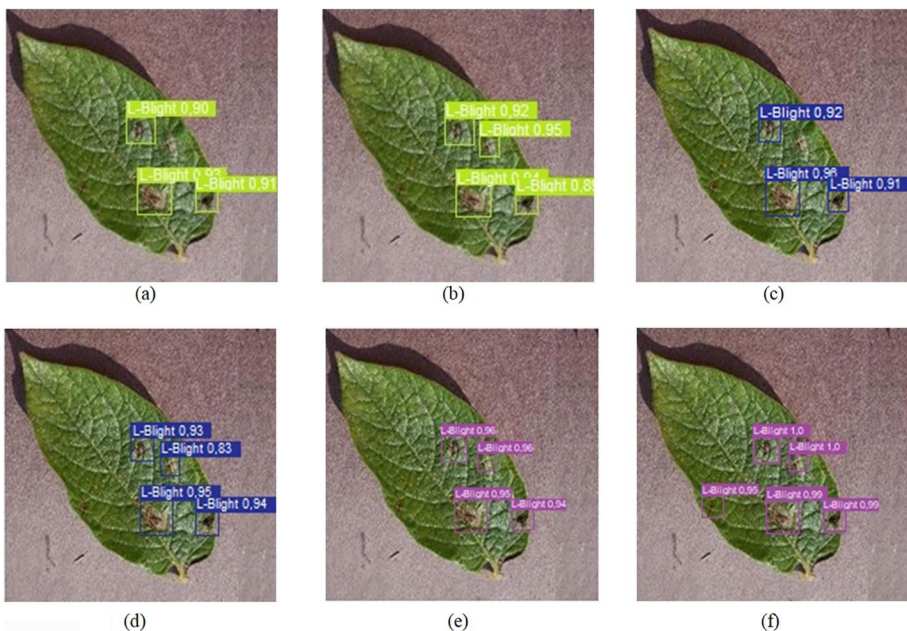
For Faster-RCNN SD, after the last epoch, the mAP (50–95) is 0.58 but this is not the mAP for the best epoch. The model achieves the best mAP after epoch 32 where the mAP is 0.6. We observe that best mAP(50–95) for the other models are not achieved at last epoch. Indeed, the mAP (50–95) for Faster RCNN SRD is 0.63 at epoch 35, 0.6 for Detr SD at epoch 27, 0.62 for Detr SRD at epoch 29, 0.71 for Yolo V8 at epoch 32 and finally 0.76 for Yolo V8 at epoch 30. The best mAP (50–95) is achieved by Yolo V8 SRD model. The models trained and validated on SRD achieve better mAP(50–95) overall.

Figure 6 shows training / validation Loss Box, they are visualized together on a graph for the six models.

All the models perform good localization of the object (in our case detecting late blight disease on potato plant leaf). In fact, the training loss box and validation loss box both decrease, converge to a low value and stabilize at a specific point. However, the lower value of Training/Validation Loss Box is attempted by Yolo V8 models: 0.055 for training on SD dataset, 0.047 for validation on SD dataset, 0.045 for training on SRD dataset and 0.031 for validation on SRD dataset. The training results can be further optimized using more powerful hardware, that allow using larger batch sizes and improve convergence (Fig. 6).

Figure 7(a)–(f) display detection results (localization of late blight lesions on potato leaf according with their corresponding confidence) using the six trained models on infected leaf image taken from test set. Figure 6(a)–(b) show detection results using Fast-R-CNN-SD and Fast-R-CNN-SRD. Figure 7(c)–(d) show detection results using Detr-SD and Detr-SRD. Figure 7(e)–(f) show detection results using Yolo V8-SD and Yolo V8-SRD. By comparing the obtained results, we can see that training models on super-resolve dataset enhance detection. In fact, five late blight lesions were detected using Yolo V8-SRD compared to Yolo V8-SD which detect four lesions. Yolo V8-SRD detect a very small lesion with high confidence (0.95). In addition, four late blight lesions were detected using Fast-R-CNN-SRD and Detr-SRD compared to Fast-R-CNN-SD and Detr-SD which detect only three lesions.

Finally, the training and inference time of the six trained models is calculated. This evaluation is interesting in scenario when real-time plant disease detection and monitoring is required.



**Fig. 7** Automatic potato late blight detection results on test image. (a) Faster-RCNN-SD detection, (b) Faster-RCNN-SRD detection, (c) Detr-SD detection, (d) Detr-SRD detection, (e) YoloV8-SD detection, (f) YoloV8-SRD detection

Table 3 shows the training time of the six trained models. According to adjusted training parameters (previously presented in Sect. 4.1), Yolo V8 SD/SRD and Faster-RCNN SD/SRD take the least training time. Faster-RCNN require more training time than Yolo V8 because, Faster-RCNN model employs selective search to propose regions and uses a CNN architecture to process these regions (which can be time consuming). Yolo V8 combines the tasks of object classification and localization into a single stage making the training process significantly faster. Detr achieve the longer training time due to the large number of parameters that need to be trained and the complexity of transformers.

Inference time refers to the amount of time it takes for a trained model to make predictions on unseen data. For more accurate assessment, the inference time is calculated for 20 different input images (20 Standard inputs with original resolution and 20 super-resolved input images). The obtained results, depicted in Table 4 show that the lowest inference time is achieved by Yolo V8 SRD, 12 ms, when the model process Standard input images and the highest inference time, 91 ms is achieved by Detr SD when the model process Super-resolved input images. As demonstrated in previous results, training Faster-RCNN, Detr and Yolo V8 models on super-resolved dataset improve the model's ability to precisely detect small lesions, However, processing high resolution images lead to increasing inference time of the models. In fact, inference time of the six trained models is higher when input images are super-resolved. From Tables 3 and 4, we can see that using higher resolution images increase both training and inference time. Based on all the obtained results, we can summarize that Yolo-V8 (SRD) maintain a good balance between detection precision and inference time.

**Table 3** Training time evaluation

Model	Train- ing time (Hours)
Faster-RCNN SD	14.78
Faster-RCNN SRD(Ours)	16.37
Detr SD	22.92
Detr SRD (Ours)	24.10
Yolo V8 SD	8.42
Yolo V8 SRD (Ours)	9.10

**Table 4** Average inference time evaluation

Model	Average inference time (Milliseconds)	
	Standard input images	Super- resolved input images
Faster-RCNN SD	52	65
Faster-RCNN SRD(Ours)	50	54
Detr SD	80	91
Detr SRD (Ours)	73	86
Yolo V8 SD	16	22
Yolo V8 SRD (Ours)	12	19



## 5 Conclusion

In this work, three object detection models (Faster-RCNN, Detr and Yolo V8) were trained on two types of datasets (Standard dataset (SD) and super-resolved dataset (SRD)). A comparison between the six trained model were conducted. The experimental results showed that achieving a precise recognition and localization of potato late blight disease are improved by training object detection models on super-resolved dataset. The obtained results showed promising results in achieving high mean Average Precision (mAP). Training Faster-RCNN, Detr and Yolo V8 on super-resolved dataset (SRD) outperformed other trained models. Using high resolution images dataset allow to detect very small lesions on the leaves at the cost of a small increase in training and inference time. The hardware resources and specifications must be taken into consideration, more powerful hardware can process complex architectures or higher-resolution input images faster. Note that, to deploy a plant disease detection model on smart device, a specialized inference engines like TFLite or OpenVINO must be used to optimize models for specific hardware. We can then conclude, that there is a fundamental compromise between precision and inference time, the increased resolution can capture more details, which is important for detecting small lesions on infected plant leaves, but it results in slower inference time and require more computational resources. In the future, performance of the proposed method could be evaluated and generalized to automatically detect and localize other types of plant diseases.

**Authors' contributions** All the authors contributed equally and significantly in writing this article and read and approved the final manuscript.

**Funding** No funding was received to assist with the preparation of this manuscript.

**Data availability** The data that support the findings of this study are openly available in: <https://www.kaggle.com/datasets/emmarex/plantdisease>

Additional data related to this work can be obtained on request from the corresponding author.

## Declarations

**Conflicts of interest** All authors declare that they have no conflicts of interest.

## References

1. Dey AK, Sharma M, Meshram M (2016) Image processing-based leaf rot disease, detection of betel vine (Piper betle L.). *Procedia Comput Sci* 85:748–754. <https://doi.org/10.1016/j.procs.2016.05.262>
2. Yan-Cheng Zhang, Han-Ping Mao, Bo Hu, Ming-Xi Li (2007) Features selection of cotton disease leaves image based on fuzzy feature selection techniques," 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing 124–129. <https://doi.org/10.1109/ICWAPR.2007.4420649>
3. Mokhtar U, Ali MA, Hassanien AE, Hefny H (2015) Identifying two of tomatoes leaf viruses using support vector machine. In *Information Systems Design and Intelligent Applications*; Springer: Berlin, Germany 771–782
4. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. *Front Plant Sci* 7:1419. <https://doi.org/10.3389/fpls.2016.01419>
5. Prajapati HB, Shah JP, Dabhi VK (2017) Detection and classification of rice plant diseases. *Intell Decis Technol* 11:357–373. <https://doi.org/10.3233/IDT-170301>
6. Fuentes A, Yoon S, Kim SC, Park DS (2017) A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* 17(9):2022. <https://doi.org/10.3390/s17092022>
7. Kaur S, Pandey S, Goel S (2018) Semi-automatic leaf disease detection and classification system for soybean culture. *IET Image Proc* 12:1038–1048. <https://doi.org/10.1049/iet-ipr.2017.0822>

8. Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric* 145:311–318. <https://doi.org/10.1016/j.compag.2018.01.009>
9. Singh UP, Chouhan SS, Jain S, Jain S (2019) Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. *IEEE Access* 7:43721–43729. <https://doi.org/10.1109/ACCESS.2019.2907383>
10. Lin Z et al (2019) A unified matrix-based convolutional neural network for fine-grained image classification of wheat leaf diseases. *IEEE Access* 7:11570–11590. <https://doi.org/10.1109/ACCESS.2019.2891739>
11. Sethy PK, Barpanda NK, Rath AK, Behera SK (2020) Deep feature-based rice leaf disease identification using support vector machine. *Comput Electron Agric* 175:105527. <https://doi.org/10.1016/j.compag.2020.105527>
12. Shrivastava VK, Pradhan MK (2021) Rice plant disease classification using color features: a machine learning paradigm. *J Plant Pathol* 103:17–26. <https://doi.org/10.1007/s42161-020-00683-3>
13. Atila Ü, Uçar M, Akyol K, Uçar E (2021) Plant leaf disease classification using Efficient Net deep learning model. *Ecol Informatics* 61:101182. <https://doi.org/10.1016/j.ecoinf.2020.101182>
14. Yan Zhang, Shiyun Wa, Longxiang Zhang, Chunli Lv (2022) Automatic Plant Disease Detection Based on Tranvolution Detection Network with GAN Modules Using Leaf Images. *Front Plant Sci Sec Technical Advances in Plant Science* 13. <https://doi.org/10.3389/fpls.2022.875693>
15. Ashwinkumar S, Rajagopal S, Manimaran V, Jegajothi B (2022) Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks, *Materials Today: Proceedings*, Volume 51, Part 1:480–487. <https://doi.org/10.1016/j.matpr.2021.05.584>
16. Borhani Y, Khoramdel J, Najafi E (2022) A deep learning-based approach for automated plant disease classification using vision transformer. *Sci Rep* 12:11554. <https://doi.org/10.1038/s41598-022-15163-0>
17. Liu Y, Liu J, Cheng W, Chen Z, Zhou J, Cheng H, Lv C (2023) A high-precision plant disease detection method based on a dynamic pruning gate friendly to low-computing platforms. *Plants* 12:2073. <https://doi.org/10.3390/plants12112073>
18. Saraswat S, Singh P, Kumar M et al (2023) Advanced detection of fungi-bacterial diseases in plants using modified deep neural network and DSURF. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-023-16281-1>
19. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. *J Big Data* 6:60. <https://doi.org/10.1186/s40537-019-0197-0>
20. Yao T, Luo Y, Chen Y, Yang D, Zhao L (2020) Single-Image Super-Resolution: A Survey. In: Liang, Q., Liu, X., Na, Z., Wang, W., Mu, J., Zhang, B. (eds) *Communications, Signal Processing, and Systems. CSPA 2018. Lecture Notes in Electrical Engineering* vol 516. Springer, Singapore. [https://doi.org/10.1007/978-981-13-6504-1\\_16](https://doi.org/10.1007/978-981-13-6504-1_16)
21. Arun PV, Buddhiraju KM, Porwal A, Chanussot J (2020) CNN-based super-resolution of hyperspectral images. *IEEE Trans Geosci Remote Sens* 58(9):6106–6121. <https://doi.org/10.1109/TGRS.2020.2973370>
22. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3147–3155. <https://doi.org/10.1109/CVPR.2017.298>
23. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. *Adv Neural Inform Process Syst* 2:2672–2680. <https://doi.org/10.5555/2969033.2969125>
24. Dong C, Loy CC, He K, Tang X (2016) Image super-resolution using deep convolutional networks. *IEEE Trans Patt Anal Machine Intellig* 38(2):295307. <https://doi.org/10.1109/TPAMI.205.2439281>
25. Haris M, Shakhnarovich G, Ukita N (2019) Recurrent back-projection network for video super-resolution. In: *Proc IEEE Conf Comput Vis Pattern Recognit* 3897–3906. <https://doi.org/10.1109/CVPR.2019.00402>
26. Kawulok M, Benecki P, Piechaczek S, Hrynczenko K, Kostrzewa D, Nalepa J (2020) Deep Learning for Multiple-Image Super-Resolution. *IEEE Geosci Remote Sens Lett* 17(6):1062–1066. <https://doi.org/10.1109/LGRS.2019.2940483>
27. Elad M, Hel-Or Y (2001) A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. *IEEE Trans Image Process* 10(8):1187–1193. <https://doi.org/10.1109/83.935034>
28. Stark H, Oskoui P (1989) High-resolution image recovery from image plane arrays, using convex projections. *JOSA A* 6(11):1715–1726. <https://doi.org/10.1364/JOSAA.6.001715>
29. Tipping ME, Bishop CM (2003) Bayesian image super-resolution. In *Adv Neural Inf Process Syst* 2033:1303–1310

30. Hamdi A, Chan YK, Koo VC (2021) A new image enhancement and super resolution technique for license plate recognition. *Heliyon* (11): e0834. 110.1016/j.heliyon.2021.e08341
31. A Lugmayr, M Danelljan, R Timofte (2019) "Unsupervised learning for real-world super-resolution, IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 3408–3416. <https://doi.org/10.1109/ICCVW.2019.00423>
32. B Lim, S Son, H Kim, S Nah, KM Lee (2017) Enhanced deep residual networks for single image super-resolution, 2017, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 1132–1140. <https://doi.org/10.1109/CVPRW.2017.151>
33. Wang X et al. (2019) ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: Leal-Taixé, L., Roth, S. (eds) Computer Vision – ECCV 2018 Workshops. ECCV 2018. Lecture Notes in Computer Science 11133. Springer, Cham. [https://doi.org/10.1007/978-3-030-11021-5\\_5](https://doi.org/10.1007/978-3-030-11021-5_5)
34. Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. IEEE Conf Comput Vision Pattern Recog (CVPR). <https://doi.org/10.1109/CVPR.2017.19>
35. Girshick R, Donahue J, Darrell T, Malik J (2016) Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans Pattern Anal Mach Intell* 38(1):142–158. <https://doi.org/10.1109/TPAMI.2015.2437384>
36. R. Girshick, Fast R-CNN (2015) IEEE International Conference on Computer Vision (ICCV), Santiago, Chile 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>
37. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans Pattern Anal Mach Intell* 39:1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
38. K He, G Gkioxari, P Dollár, R Girshick (2017) Mask R-CNN, IEEE International Conference on Computer Vision (ICCV), Venice, Italy 2980–2988. <https://doi.org/10.1109/ICCV.2017.322>
39. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, JM. (eds) Computer Vision – ECCV 2020. Lecture Notes in Computer Science() 12346. Springer, Cham. [https://doi.org/10.1007/978-3-030-58452-8\\_13](https://doi.org/10.1007/978-3-030-58452-8_13)
40. J Redmon, S Divvala, R Girshick, A Farhadi (2016) You Only Look Once: Unified, Real-Time Object Detection, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 779–788. <https://doi.org/10.1109/CVPR.2016.91>
41. Hussain M (2023) YOLO-v1 to YOLO-v8 and its complementary nature toward digital manufacturing and industrial defect detection. *Machines* 11:677. <https://doi.org/10.3390/machines11070677>
42. K He, X Zhang, S Ren, J Sun (2016) "Deep residual learning for image recognition," 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
43. Bochkovskiy A, Wang C, Liao HM (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv, abs/2004.10934*. <https://doi.org/10.48550/arXiv.2004.10934>
44. Horé, Alain, Ziou, Djemel (2010) Image quality metrics: PSNR vs. SSIM. 20th International Conference on Pattern Recognition, Istanbul, Turkey. <https://doi.org/10.1109/ICPR.2010.579>
45. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612. <https://doi.org/10.1109/TIP.2003.819861>
46. Padilla SL, Netto, EAB da Silva (2020) A survey on performance metrics for object-detection algorithms International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil 237–242. <https://doi.org/10.1109/IWSSIP48289.2020>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.