Tree Leaves Detection Based on Deep Learning

Lei Wang and Wei Qi Yan

Auckland University of Technology, Auckland 1010, New Zealand

Abstract. In this paper, digital images related to five kinds of leaves which are available at New Zealand are collected as our dataset, two deep learning models, namely, Faster R-CNN and YOLOv5, representing two-stage and one-stage algorithms, are employed to conduct tree leaves detection. Our results show that YOLOv5 model obviously outperforms to the Faster R-CNN in the speed of both model training and object detection. The difference between these two methods is not significant in the comparison of mAPs. We conclude that YOLOv5, as the representative of one-stage algorithm, is obviously better than Faster R-CNN, as the representative of two-stage algorithms, especially, its advantage in speed makes sure it a bright prospect in deep learning applications.

Keywords: YOLOv5 · Faster R-CNN · Leaf detection · Deep learning

1 Introduction

1.1 Background and Motivation

New Zealand is one of the well-known scenic countries in the world. The forest resources are quite abundant. There are a variety of tree species growing everywhere, many of them are unique worldwide [1]. The research work for leaf classification has been conducted for decades, plant recognition via digital images has become an interdisciplinary subject in the field of plant taxonomy and computer vision [2]. The research work of plant automatic taxonomy has made great progress, but with the continuous expansion of its applications, it is still hard to fully meet the needs of reality [3].

Object detection is an important branch of image processing and computer vision [20, 21]. In deep learning, object detection algorithms have been explored rapidly. However, visual objects often have distinctive poses and are often blocked. Taken into consideration of the complexity and diversity of scenes, this is still a challenging topic, there are rooms to be improved.

Visual object detection is roughly grouped into two categories. The first one is R-CNN family algorithms based on regional proposals, whose representative network is R-CNN, Fast R-CNN, SPPNet, Faster R-CNN, FPN, Mask R-CNN, etc. They are characterized by using two-stage methods, which require the algorithms to generate the candidate boxes of an object, namely, position of the object, and conduct the classification and regression of the candidate boxes.

One-stage algorithms, which do not need the region proposals, directly generate the class probability and position coordinates of the object. Its representative network is...
YOLO series, like YOLO, YOLOv2, YOLOv3, YOLOv4, YOLOv5, SSD, RetinaNet, etc.

There have been a lot of research work for tree leaves classification. Compared with traditional image classification, visual object detection is obviously in line with practical needs. In reality, it is often impossible to have only one visual object in a scene. The requirements for visual object detection are much complicated, required algorithms not only verify what an object is, but also determine where the object is in the image. There is relatively little research work on leaf detection, but more focus on automatic driving, video monitoring, mechanical processing, intelligent robot, and other fields. Therefore, in this paper, we endeavor to explore the performance of deep learning algorithms for tree leaves detection.

Pertaining to object detection algorithms over the past decades [23–30, 31], this project adopts two representative algorithms based on deep learning for visual object detection, namely, YOLOv5 based on one-stage algorithms and Faster R-CNN based on two-stage algorithms. Generally, the detectors based on two-stage algorithms have high positioning and recognition accuracy, while one-stage detectors have the advantage of speed. The difference between them is that the two-stage detectors have a step of generating region proposals, while the one-stage detectors directly regress and classify the prediction boxes. In this paper, our two algorithms are applied to the leaf dataset collected locally in New Zealand.

1.2 Contribution

There has been a great deal of literature on leaf classification for decades. However, with the emergence of various new models in deep learning, computer vision has gradually shifted from image classification to object detection, object tracking, semantic segmentation, and instance segmentation. In view of 3D shape of leaves in object detection, leaf images observed from different angles of view may be completely different, object detection is more difficult than image classification. In addition, different from traditional leaf classification, this project is not only to identify and classify single leaf on digital images, but also to realize the recognition of multiple types of leaves in videos.

Pertaining to object detection, there are three issues that make object detection more difficult: (1) Multiple objects with different classes and the number of objects, (2) multiple scales, for example, visual objects with different sizes, (3) external environment interference, such as the changes of illumination, the existence of occlusion, and the quality of images.

In this project, we trained two different models representing one-stage and two-stage algorithms respectively. At the same time, local leaves of New Zealand were collected as datasets. Experimental results of two models were compared and analyzed through the detection of actual leaves in our videos.

2 Literature Review

There is literature regarding leaf classification, the research work on object detection is also very ample and full in recent years. However, the research work on leaf detection is relatively limited.
In 2012, deep convolutional neural network showed excellent performance in ILSVRC-2012 large-scale image classification task. The model received training of more than one million images, the error rates of the top 5 in the 1000 classes are reduced to 15.3% which almost halves the error rate of the best competing methods. This success led to a revolution in computer vision [4].

In 2017, three deep learning networks GoogLeNet, AlexNet, and VGGNet were used to identify plant species on LifeCLEF 2015 dataset [5], with an overall accuracy 80% for the best model [6].

In essence, object detection is also an image classification technique. In addition to object classification, the algorithms identify the locations of object instances from numerous predefined classes of natural images. Object detection is one of the most basic and challenging tasks in computer vision [7].

After CNN achieved great success in the 2012 ImageNet classification task [8], Girshick firstly proposed the region-based convolutional neural networks (R-CNN) in 2014, its algorithmic structure became the classical one of two-stage algorithms. Since then, object detection algorithms have been developed at an unprecedented speed [9].

At present, deep learning methods for object detection are mainly grouped into two directions: Two-stage algorithms and one-stage algorithms. The former refers to the algorithm generating a series of candidate frames as samples and classifying the samples through CNN; the latter does not need to generate candidate frames, but directly transforms the problem of object border locating into a regression problem. The performance of the two methods is also different. Generally, the former is superior in the accuracy of object detection and positioning, while the latter outperforms in algorithmic speed.

Fast R-CNN algorithm was proposed in 2015 [10], which designs a pooling layer structure for ROI and effectively solves the problem that R-CNN algorithm must crop and scale the image to the same size. The idea of multi-task loss function is also proposed. Gradient is transmitted directly through ROI Pooling layer. But it still does not get rid of the limitations of generating positive and negative candidate boxes in the selective search algorithm.

In order to solve the defects of Fast R-CNN algorithm, Faster R-CNN algorithm was proposed in 2015 [11]. Region Proposal Network (RPN) was designed, its advantage is that the whole network process can share the feature information extracted by CNN, which saves the computational costs, solves the problem that Fast R-CNN algorithm is slow in generating positive and negative candidate boxes, and avoids the decrease of algorithmic accuracy due to too much extraction of candidate boxes. RPN network can generate multi-size candidate boxes in the convolution feature map of fixed size, resulting in the inconsistency between the size of the variable object and the fixed receptive field. This is a shortcoming of the Faster R-CNN model.

In view of the existence of RPN structure, the two-stage methods are represented by R-CNN algorithms, though the detection accuracy is getting higher and higher, its speed of detection reaches a plateau, which makes it difficult to meet the real-time requirement. Therefore, one-stage algorithm is represented by YOLO based on regression method. The one-stage algorithm can classify the class and location information directly through the trunk network without using RPN network. It’s faster, but initially less accurate.
In 2015, YOLO algorithm inherits the idea of overfeating, its speed of detection reaches 45 frames per second. P-Relu activation function is adopted during model training. However, it has problems such as the accuracy of positioning and recall rate are dissatisfied. Besides, the detection for very small object is ineffective. At last, generalization ability is relatively weak [12].

YOLOv2 [13] and YOLOv3 [14] algorithms were proposed on CVPR 2017, focusing on solving the poor recall rate and positioning accuracy. It uses DarkNet-19 as the feature extraction network and adds batch normalization as the pretreatment. The original YOLO uses the full connection layer to directly predict the coordinates of the bounding box, while YOLOv2 refers to the idea of Faster R-CNN, introduces anchor mechanism, and calculates a better anchor template in the training set.

The operation of anchor boxes is applied to the convolutional layer so as to improve the prediction of the bounding boxes. Meanwhile, the positioning method with strong constraints is adopted to greatly improve the recall rate of the algorithm. Combined with the fine-grained features, the shallow feature and the deep feature are connected, which is helpful to the detection of small-size objects.

In 2020, YOLOv4 was released [15], which is a significant update to the YOLO family, with an increase in AP and FPS of 10% and 12% based on COCO datasets, respectively. On June 25, 2020, Ultralytics released YOLOv5 on Github, which performs excellently, especially the frame rate 140FPS of YOLO v5s model is amazing.

3 Methodology

YOLO and Faster R-CNN, namely, two-stage algorithms, were taken into account in this project to detect local leaves, compare and analyze the differences between them. The reason why these two algorithms are chosen is that they are the most representative ones of single-stage and two-stage algorithms, respectively.

3.1 Working Principle and Structure Analysis of YOLO

The bounding boxes and class labels of Pascal-VOC and Kaggle datasets are saved in .xml file. However, this format of annotation cannot be imported directly into MATLAB. We wrote a program for converting XML format files to MATLAB files.

In YOLOv5, four versions of object detection network are given, namely, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. YOLOv5s model was used in this project. YOLOv5s is the network with the minimum depth and the minimum width in YOLOv5 series. The YOLOv5s is the fastest one, but the AP is relatively less accurate. However, this model is also a good choice if the focus of object detection is on larger targets and less complex scenarios, which are speed oriented. The other three networks of YOLOv5 series are based on YOLOv5s to continuously deepen and widen the network, the AP is also continuously improved, but the speed will become slower.

The structure of YOLOv5 is very similar to that of YOLOv4, both use CSPDarkNet53 (i.e., Cross Stage Partial Network) as the Backbone net. PANET (i.e., Path Aggregation Network) and SPP (i.e., Space Pyramid Pooling) were employed as the Neck net.
The input of YOLOv5 adopts the same way of mosaic data enhancement as YOLOv4. Mosaic refers to the method of CutMix data enhancement [16]. However, CutMix only takes use of two images for splicing, while mosaic data enhancement utilizes four images, which are randomly scaled, cropped, and resized. In addition, YOLOv5 is also optimized in terms of adaptive image scaling.

The function of backbone is to aggregate and combine different fine-grained images to form a convolutional neural network. The backbone of YOLOv5s adopts CSPNet, which takes use of gradient information in other CNN frameworks.

Neck is a series of network layers that mix and combine image features and transfer them to the prediction layer, also known as Head. The Neck part of YOLOv5s adopts PANET [17]. Generating feature pyramids is the main function of the Neck. The feature pyramid enhances the detection of visual objects with multiple scales, thus it recognizes the same object with various sizes and scales.

The Head of the model is mainly employed for final judgment. In YOLOv5 model, the model Head is as same as the previous versions of YOLOv3 and YOLOv4. Each Head has \((80 \text{ classes} + 1 \text{ probability} + 4 \text{ coordinates}) \times 3 \text{ anchor frames}, 255 \text{ channels in total.}\)

The loss function for object detection is generally composed of classification loss and recurrence loss of bounding boxes. GIOU is used as the loss function of bounding boxes in YOLOv5. In the postprocessing of object detection, NMS operation is usually needed for filtering visual objects. In YOLOv5, weighted NMS is accommodated.

The selection of activation function is very important for deep neural networks. In YOLOv5, Leaky ReLU is used in the middle/hidden layer, sigmoid function is applied to the final detection layer.

YOLOv5 provided us with two optimization functions Adam and SGD. The default one is SGD. If we train a smaller dataset, Adam is a better choice. Adam is offered in this project.

### 3.2 Analysis of the Working Principle of Faster R-CNN

After the accumulation of R-CNN and Fast R-CNN, Faster R-CNN algorithm was proposed in 2016. Faster R-CNN integrates feature extraction, proposal extraction, bounding box regression, and classification into one network, which greatly uplifts the overall performance, especially the detection speed. The main implementation steps of Faster R-CNN are as follows.

The first step is to extract features. Faster R-CNN firstly extracts feature map of the candidate image. The feature map is shared for subsequent RPN (Region Proposal Network) layer and fully connected layer.

The second step is to enter RPN (i.e., Region Proposal Network), which is used to generate regional image blocks. This layer determines through “softmax” that anchors belong to the foreground or background and take use of bounding box regression to get precise proposals.

The third step is ROI pooling. This layer collects the input feature map and candidate object regions. The feature map of the object region is extracted and sent to the following fully connected layer to determine the object class.
The fourth step is classification. The feature map of object region is used to classify the class of the visual object, the regression of boundary boxes is used to obtain the final precise location of the box. Therefore, we see that the highlight of Faster R-CNN is that it proposes an effective method to locate the object area, which greatly reduces the time consumption of convolution operation, the speed has been greatly accelerate.

3.3 Environmental Deployment

Since model training and validation need a lot of computing power, we employ Google Colab platform for the computations. Google Colab is a free cloud-based deep learning platform based on Jupiter notebook, which provides a free Tesla P100 GPU for deep learning users. In this project, the requirement of computational power needed to make and annotate datasets is relatively weak, we only utilize local computing power to complete it. The dataset is imported into the Colab platform for model training and validation. In the local environment, we use LabelMe software to label the collected images.

3.4 Data Set Preparation

The basic dataset encapsulates five types of tree leaves which were collected from a park at Auckland, New Zealand: (1) Magnolia grandiflora, (2) Boehmeria nivea, (3) Clausena lansium, (4) Euphoria longan, (5) Hibiscus. Firstly, we collected these five kinds of leaves from the park, we took pictures as the collected visual data, including various distances, view angles, and so on. The shooting background includes wood floor and white paper. The shooting includes single leaves and the combination of multiple leaves. A total of eight groups of video footages were captured as the samples, the video durations are various from 30 s to 80 s, the frame rate was 60 FPS. An example of a screenshot taken with a white paper background and a wood floor background is shown in Fig. 1.

![Fig. 1. Leaf images collected](image-url)

We extracted frames from the video to obtain 419 images. We used software LabelMe, a data annotation software, to label the obtained images. In the process of labeling, leaf abbreviations are combined with numbers and initials of leaf names. 1 –“m.g.” stands for Magnolia Grandiflora, 2 –“b. n.” refers to Boehmeria nivea, 3 –“c. l.” means Clausena Lansium, 4 –“e.l.” shows Euphoria longan, and 5 –‘h.’ represents for Hibiscus. Given the
relatively small number of overall datasets, data augmentation is necessary. The specific operations include flipping, zooming in, zooming out, clipping, and combining. The 419 images were expanded four times to 1676 images. The data is split into training set and validation set according to the ratio of 8:2. The final training set has 1,340 images and 336 images in the validation set. The final test was conducted by using a 37 s video shot having a mixture of leaves against a wood floor background.

3.5 Evaluation Methods

AP means average precision, which is the area under the precision-recall curve. mAP is the mean average precision, which is the average AP value of multiple categories of AP. The size of mAP must be in the interval [0, 1.0], the larger the better. This index is a key index to measure the detection accuracy in object detection. The loss function of object detection consists of two parts: Classification loss and regeneration loss of bounding boxes. The loss of classification is mainly based on cross-entropy loss function, which has not been changed much in recent years. IOU and GIOU are representative methods in the development of regression of bounding boxes. YOLO v5 adopts the more advanced GIOU.

RPN network is introduced into Faster R-CNN network. At the same time, the loss function of RPN network is also used as the loss function of Faster R-CNN algorithm whilst training RPN network. In addition to mAP, another important performance of object detection algorithm is speed. A measure of speed is FPS (frame per second), the number of images that can be processed per second.

4 Analysis and Discussions

4.1 Comparison of Object Detection Results

In order to compare the differences between the two models accurately, it is necessary to be tested in multiple scenarios, including model testing in single leaf scene, in a small number of mixed leaves, and in complex scenarios with multiple leaves.

In the single-leaf scenario, we see that YOLOv5 accurately identified visual objects, the selection of bounding boxes is much appropriate. Meanwhile, Faster R-CNN identifies the majority of objects accurately. Based on the “E.L.” leaf, two results were
exported. In addition, when the model Faster R-CNN identifies the “B.N.” leaf, it only recognizes a part of the leaf, the bounding box is too large when it identifies the ‘H’ leaf (Fig. 2).

In the scenario of model testing with a small number of mixed leaves, we see that Faster R-CNN model accurately identifies most of visual objects, but it also missed a few of individual leaves. At the same time, there are few cases where the box is too large or too small (Fig. 3).

For YOLOv5, even if a partial leaf is able to be recognized, the box size is appropriate. But one of them was wrongly identified, the ‘H’ leaf was wrongly identified as the “BN leaf”.

Fig. 3. The results of Faster R-CNN and YOLOv5 in a small number of mixed leaves

In the complex scene where all the five kinds of leaves are contained, YOLOv5 model identified almost all the leaves accurately. However, the toy wheel in the lower left corner of the first set of pictures is mistakenly identified as a leaf. The Faster R-CNN model can only identify part of the leaves in this scene, and the box selection is not very accurate. But it also doesn’t recognize the toy wheel in the lower left corner of the first set of images as a leaf (Fig. 4).

Fig. 4. A small number of mixed leaves in complex scenarios with multiple leaves
4.2 Comparative Analysis of the Two Proposed Models

In Fig. 5, in terms of classification task, both algorithms decline steadily, but the training loss of YOLOv5 falls faster.

![Classification](image)

**Fig. 5.** The training loss curve of YOLOv5 and Faster R-CNN model for leaf classification

In Fig. 6, the two algorithms maintain the trend decline for regression loss. YOLOv5 started with a brief rise, but tends to be stable in the later stage, the whole process presents a bit amplitude. Faster R-CNN drops much steadily and has a small amplitude of shock.

![Objectness](image)

**Fig. 6.** The training loss curve of YOLOv5 and Faster R-CNN model for regression

For the total loss curve, YOLOv5 decreases faster in the early stage and then tends to slow down, while Faster R-CNN also decreases slowly and steadily in the whole process as shown in Fig. 7.

![GIOU](image)

**Fig. 7.** The training loss curve of YOLOv5 and Faster R-CNN model
The network structure of the model determines the training speed and execution speed of the model as well as the memory usage. The training speed, execution speed and memory usage of the two models are shown in Table 1. YOLOv5 has obvious advantages in speed and memory consumption. Compared with Faster R-CNN, YOLOv5 is nearly 32 times faster in training speed, nearly 39 times faster in execution speed, and nearly 8 times smaller in memory occupancy.

| Types          | YOLOv5       | Faster R-CNN |
|----------------|--------------|--------------|
| Training speed | 26 ms/step   | 814ms/step   |
| Execution speed| 0.011        | 0.432s       |
| Memory usage   | 14MB         | 109MB        |

Based on the IoU threshold value of 0.5, the mAP results obtained by using the two models are shown in Fig. 8. Both methods kept the accuracy increasing gradually, but YOLOv5 increased from 0, while Faster R-CNN grew from 0.80, which indicates that it increases much slowly. The primary reason why the initial accuracy of Faster R-CNN is so high is that the model is a two-stage algorithm. In the training for RPN candidate boxes in the first stage, there are many candidate boxes in each column, the boxes that have not objects are classified as negative classes, resulting in a high accuracy even if all the results are negative.

After the model training, we selected the one with the lowest loss as the optimal model and carried out validation. Finally, we calculated that the mAP of YOLOv5 was 0.932 and the mAP of Faster R-CNN was 0.918.

**Fig. 8.** The training curve of YOLOv5 model and Faster R-CNN model

### 4.3 Discussions

From the experimental results of this project, the YOLOv5 model representing the one-stage algorithms is superior to the Faster R-CNN model on behalf of the two-stage
algorithms in most indicators. The differences between them in training speed, execution speed, model size, accuracy, and others are shown in Table 2. YOLOv5 model also performs better in rectangular box regression. In addition, YOLOv5 takes up less memory and executes faster, so it is able to be adapted to more devices and scenarios.

| Type               | YOLOv5       | Faster R-CNN  |
|--------------------|--------------|---------------|
| Training speed     | 26 ms/step   | 814 ms/step   |
| Execution speed    | 0.011 s      | 0.432 s       |
| Memory usage       | 14 MB        | 109 MB        |
| mAP                | 0.932        | 0.918         |
| Total loss         | 0.032        | 1.028         |

To sum up, YOLOv5 has made various optimizations in data augmentation, feature extraction, loss function, and other aspects, which shows its excellence in both accuracy and speed and is suitable for more complex and diverse applications.

5 Conclusion and Future Work

In this paper, YOLOv5 model and Faster R-CNN model were employed to implement object detection of leaves collected locally in New Zealand. Experimental results show that YOLOv5 algorithm is superior in almost all indicators. Especially, YOLOv5 algorithm is superior to Faster R-CNN algorithm in terms of speed, memory occupancy, and accuracy of object position prediction. From the viewpoint of loss functions, YOLOv5 model is better.

In this project, the number and species of leaves as datasets are relatively small. In order to ensure the outperformance experimentally, the shapes of leaves are significantly different. However, in practical applications, the kinds of leaves with very similar shapes may appear, the difficulty of leaf identification will be greatly increased. Therefore, it is necessary to train the model with more numbers and types of tree leaves in future.

The background of this experiment is relatively simple, but in practical application, the background of leaves is often very complex. How to implement object detection with high recognition rate under complex background is our future work [18, 19].

References

1. McGlone, M., Buitenwerf, R., Richardson, S.: The formation of the oceanic temperate forests of New Zealand. NZ J. Bot. 54(2), 128–155 (2016)
2. Sun, Y., Liu, Y., Wang, G., Zhang, H.: Deep learning for plant identification in natural environment. Comput. Intell. Neurosci. (2017)
3. Goëau, H., Bonnet, P., Joly, A.: Plant identification in an open-world. Lifeclef (2016)
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
5. Joly, A., et al.: LifeCLEF 2015: multimedia life species identification challenges. In: Mothe, J., et al. (eds.) CLEF 2015. LNCS, vol. 9283, pp. 462–483. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24027-5_46
6. Ghazi, M.M., Yanikoglu, B., Aptoula, E.: Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235, 228–235 (2017)
7. Zou, Z., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: a survey. arXiv preprint arXiv:1905.05055 (2019)
8. Hinton, G.E., Krizhevsky, A., Sutskever, I.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1106–1114 (2012)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
10. Girshick, R.: Fast R-CNN. In: IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
11. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
12. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
13. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
14. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
15. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
16. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: regularization strategy to train strong classifiers with localizable features. In: IEEE International Conference on Computer Vision, pp. 6023–6032 (2019)
17. Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: PANet: few-shot image semantic segmentation with prototype alignment. In: IEEE International Conference on Computer Vision, pp. 9197–9206 (2019)
18. Yan, W.Q.: Computational Methods for Deep Learning - Theoretic. Practice and Applications. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-030-61081-4
19. Yan, W.Q.: Introduction to Intelligent Surveillance - Surveillance Data Capture, Transmission, and Analytics, 3rd edn. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-319-60228-8
20. Pan, C., Yan, W.Q.: Object detection based on saturation of visual perception. Multimed. Tools Appl. 79(27–28), 19925–19944 (2020). https://doi.org/10.1007/s11042-020-08866-x
21. Pan, C., Li, X., Yan, W.: A learning-based positive feedback approach in salient object detection. IEEE IVCNZ (2018)
22. Liu, Z., Yan, W., Yang, B.: Image denoising based on a CNN model. In: ICCAR, pp. 389–393 (2018)
23. Zheng, K., Yan, W., Nand, P.: Video dynamics detection using deep neural networks. IEEE Trans. Emerg. Top. Comput. Intell. 2(3), 224–234 (2018)
24. Xiao, B., Nguyen, M., Yan, W.: Apple ripeness identification using deep learning models. ISGV (2021)
25. Zhao, K., Yan, W.: Fruit detection from digital images using CenterNet. ISGV (2021)
26. Al-Sarayreh, M., Reis, M., Yan, W., Klette, R.: Potential of deep learning and snapshot hyperspectral imaging for classification of species in meat. Food Control (2020)
27. Al-Sarayreh, M., Reis, M.M., Yan, W.Q., Klette, R.: A sequential CNN approach for foreign object detection in hyperspectral images. In: Vento, M., Percannella, G. (eds.) CAIP 2019. LNCS, vol. 11678, pp. 271–283. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29888-3_22
28. Al-Sarayreh, M., Reis, M., Yan, W., Klette, R.: Detection of red-meat adulteration by deep spectral–spatial features in hyperspectral images. J. Imaging 4(5), 63 (2018)
29. Al-Sarayreh, M., Reis, M.M., Yan, W.Q., Klette, R.: Detection of adulteration in red meat species using hyperspectral imaging. In: Paul, M., Hitoshi, C., Huang, Q. (eds.) PSIVT 2017. LNCS, vol. 10749, pp. 182–196. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75786-5_16
30. An, N., Yan, W.: Multitarget tracking using Siamese neural networks. ACM TOMM (2021)