Multi-Task Cascaded Convolutional Networks based Intelligent Fruit Detection for Designing Automated Robot

Li Zhang[1, 2], Guan Gui[3], Senior Member, IEEE, Abdul Mateen Khattak[1,4], Minjuan Wang[1, 2], Wanlin Gao[1, 2*] and Jingdun Jia[1*]

[1]Key Laboratory of Agricultural Informatization Standardization, Ministry of Agriculture and Rural Affairs, Beijing, 100083, China
[2]College of Information and Electrical Engineering, China Agricultural University, Beijing, 100083, China
[3]Nanjing University of Posts and Telecommunications, Nanjing, 210003 China
[4]Department of Horticulture, The University of Agriculture Peshawar, 25120, Pakistan

*Corresponding author: Wanlin Gao (e-mail: wanlin_cau@163.com), Jingdun Jia (e-mail: jiajdun@most.cn)

This work was supported by Project of Scientific Operating Expenses, Ministry of Education of China (2017PT19) and China Postdoctoral Science Foundation (2018M630222). All the mentioned support is gratefully acknowledged.

ABSTRACT Effective and efficient fruit detection is considered crucial for designing automated robot (AuRo) for yield estimation, disease control, harvesting, sorting and grading. Several fruit detection schemes for designing AuRo have been developed during the last decades. However, conventional fruit detection methods are deficient in real-time response, accuracy and extensibility. This paper proposes an improved multi-task cascaded convolutional network (MTCNN) based intelligent fruit detection (InFD) method. This method has the capability to make the AuRo work in real-time and with high accuracy. Moreover, based on the relationship between the diversity samples of dataset and the parameters of neural networks evolution, this work presents an improved augmented method. A procedure that is based on image fusion to improve the detector performance. The experiment results demonstrated that the proposed detector performed immaculately, both in terms of accuracy and time-cost. Furthermore, the extensive experiment also demonstrated that the proposed technique has the capacity and a good portability to work with other akin objects conveniently.

INDEX TERMS Fruit detection, real-time, cascaded convolutional networks, automated robot

I. INTRODUCTION

Fruit detection for yield estimation, grade sorting, disease control and other applications in agricultural field have achieve intensive popularity over the past few decades [1-5]. Several systems have been deployed for automated harvesting robots, which have led to considerable improvement in the industry [6],[7]. Particularly, recognizing and classifying fruits according to their quality has been one of the most popular research fields attracting most of the farm enterprises. Fruit detection is undoubtedly the first and foremost parameter to be considered in order carry out more in-depth studies on the subject. Therefore, many researchers have made efforts for years to develop robust algorithms for fruit detection [8-10]. Although, the performance of fruit detection systems has been improved remarkably, they are still far from practical application. The basic difficulties in developing such fruit detection system are the uncertain and unrestrained environments of orchards. These include numerous challenging tasks, such as insufficient or over illumination, indistinguishable backgrounds, heavy occlusion by neighborhood fruits or foliage, low-resolutions, variation of pose and so on.

Fruit detection can be considered a special type of object detection that has many similarities with face detection task [11-13]. Due to the advantage of high precision, cascaded convolutional networks (CCN) based face detection has acquired a remarkable breakthrough [14], [15]. Among these state-of-the-art methods, multi-task cascaded convolutional network (MTCNN) [16] is the most popular one due to its outstanding performance in accuracy and time-consumption. Although MTCNN has achieved great progress in face detection task, deploying this method directly for fruit detection task is not suitable. It is due to the design of MTCNN, that its architecture includes many specificity functions for face detection, which are not suitable for the task of fruit detection. Thus, there is a need to improve this MTCNN framework by removing customized functionality.

The absence of a unified benchmark is another great challenge for fruit detection. A sufficient amount of sample images plays an important role in deep learning based model training. In this research, we collected images from apple orchard by digital camera. Then we selected the suitable ones and labelled them to create a dataset. Creating a dataset manually is a tedious and time-consuming task. So we devised a new augmented method based on fusion algorithm. The motivation for this fusion method came from the principle that the generated new samples should be close to authentic images. Supplementary samples were created for diversity by adding fusion images that would help improve the final result of this detector. In order to evaluate the structure whether it could be applied to other kinds of objects conveniently, we trained the detector on two other fruits species (strawberry and orange) as well.

To summarize, our contributions are as follows:

1. We proposed a new architecture for fruit detection called Fruit-MTCNN (F-MTCNN) by improving the baseline model of MTCNN. And this detector has the attributes of high accuracy and less time-consumption.
2. We proposed a novel augmented method called fusion augmentation (FA). We generate artificial images samples by adding negative patches from samples of dataset by random cropping that supplement the samples diversity.
3. The proposed approach can be deployed to other kinds of objects conveniently with a small amount of training samples.

The organization of the rest paper is arranged as follows. In Section II, we review prior related work in fruit detection. Section III, IV we introduce method used in this study. Our experiments in this research are shows in Section V. In Section VI, we analyze and discuss our results and present the conclusion of this work.

II. RELATED WORK

Automated harvesting robot is a potential solution for many challenges in agriculture such as the explosively increasing global old-age population, labor cost increase, increasing demand for of produce and so on. Identify and obtaining precise positions of fruits are the most important parts of the visual system for a harvesting robot. Due to this reason, fruits identification and detection has been extensively studied for years. Generally, these methods can be divided into three types by the technologies they employ.

A. IMAGE PROCESSING

Several image processing techologies are in use for fruit detection task [17-20]. For example, *Aggelopoulou et al,* [21] proposed an algorthim based on binary image technology for flower images of apple tree, and analyzed the correlation between yield and flower density. To segment branches from images, *Ji et al.* [22] converted RGB color space to $I_1I_2I_3$ and XYZ space by a transformation formula. Several classification techniques such as decision trees, K-nearest neighbor, and discriminant analysis image processing algorithms are used to choose appropriate wavelengths to classify images of codling moth infestation in apples [23]. To improve fruit detection, *Bulanon et al.* [24] proposed an image fusion method by obtaining thermal and visible images simultaneously. Moreover, the experiments on an orange canopy scene of orchard showed that this approach improved fruit detection compared to the one that only used thermal images. In general, these methods need to design a special algorithm for a specific task, and they are highly dependent upon the characteristics of the subject, which needs to be redesigned if there is a slight change in its condition. Therefore, the weaknesses in these methods hardly satisfy requirements of a farm manager.

B. MACHINE LEARNING

There are some machine learning based techologies for detection tasks, such as those reported by [25-30]. To detect and count immature citrus fruits, *Lu et al.* [9] extracted features of local binary pattern (LBP) and detected local intensity maxima around the immature fruits. *Benalia et al.* [31] developed a system to improve the quality control and sorting of dried fruits of fig (*Ficus carica*). These approaches employ computer vision techniques such as PLS-DA and PCA to analyze images and get better result ultimately. *Borges et al.* [32] also presented a classification system based on clustering. This technique was applied to classify the severity of bacterial spot in tomato filed. All these machine based learning methods greatly improved the detection performance. However, the shortcomings were that the features they used were extracted through experienced worker. In addition, the high performance achieves by these machine learning based methods was at the cost of high computational complexity. Therefore, there was a need to search for and find out some new procedures that would extract features automatically.

C. DEEP LEARNING

Over the past few years, deep neural networks procedures have made a considerable progress in many fields [33]. Wireless communication [34] [35], signal processing [36-38], image classification [39], saliency detection [40-44]. Many approaches have been developed in the field of agriculture as well [45-[48]. *Bargoti and Underwood,* [49] presented an approach for fruit detection and counting using images taken in orchard. They used two feature learning algorithms i.e. multi-scale Multi-Layered Perceptrons and Convolutional Neural Networks (CNN), to segment the fruit from its background. Their final results showed the performance closer to the state-of-the-art perfection. Faster-RCNN is one of the most advanced object detection methods, has provided good results in many detection tasks [50]. Recently, a Faster-RCNN framework approach was adopted for fruit detection for mango, almond and apple in orchards

[51]. This method also showed that data augmentation can signify performance and reduce training images by more than two-folds. The final result presented that this approach accomplished a remarkable detection performance for apples and mangoes. Similarly, *Sa et al.* [52] also used Faster-RCNN as a baseline fruit detector. The difference is they used imagery obtained from these two modalities i.e. Color and Near-Infrared. Thus they proposed a new approach by combining these two kinds of information earlier or later. This proposed multi-modal approach provides better performance compare to prior work. However, using Faster-RCNN architecture for fruit detection directly is inadequate. This is because the Faster-RCNN designed detection task for many categories of objects with large scale change. Whereas, the visual system in agriculture needs to detect one or only a few kinds of fruit in general, and usually the fruit size does not change significantly. Thus, the application of Faster-RCNN model for fruit detection task is complicated and time-consuming. Furthermore, providing a large amount of data is necessary to prevent over-fitting problems, because the structure of Faster-RCNN is of a deeper architecture that contains thirteen convolution layers. During the recent years, due to the rapid development of security, intelligent equipment and other applications, the detection accuracy has been highly improved.

III. F-MTCNN DETECTOR

A. MOTIVATION

There are many similarities between face detection and fruit detection, such as various poses, illuminations and occlusions. Nevertheless, there are some differences as well between both of them. Firstly, compared with facial features (eyes, nose, mouth), the information contained in fruit is usually relatively simple. In general, the fruit feature only includes the overall information (shape, color). Secondly, it is more likely to be confronted with heavy occlusion in the tasks of fruit detection. Thirdly, there is no uniform benchmark for fruit detection, and sufficient images acquisition and annotation are time-consuming tasks. Finally, real-time is one of the most important indices for fruit detection. This is because fruit detection model is generally applied to automatic equipment, such as picking robot, sorting robot, yield estimation robot and so on. So, for the design of fruit detection model, the above mentioned motives should be taken into consideration. Based on all that, we designed a fruit detector that can detect fruits with different pose, low resolution and occlusion. It also has the capability to count the number of fruits. **Fig.** 1 shows a typical example of apple fruits appearing at different poses, sizes, distances, resolutions and occlusions.

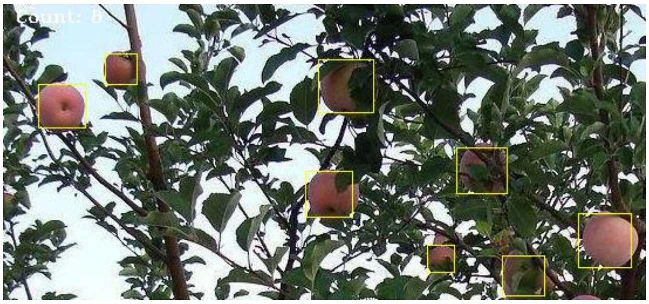

FIGURE 1. This is an example image for fruit detection, which contains pose, illumination, occlusion and scale variability. The rectangles with yellow line are the results using our methods.

B. THE OVERALL ARCHITECTURE

We improved the architecture of MTCNN by omitting landmark loss of three cascaded networks for fruit detection task. The overall architecture of this model includes three stages as shown in **Fig.** 2.

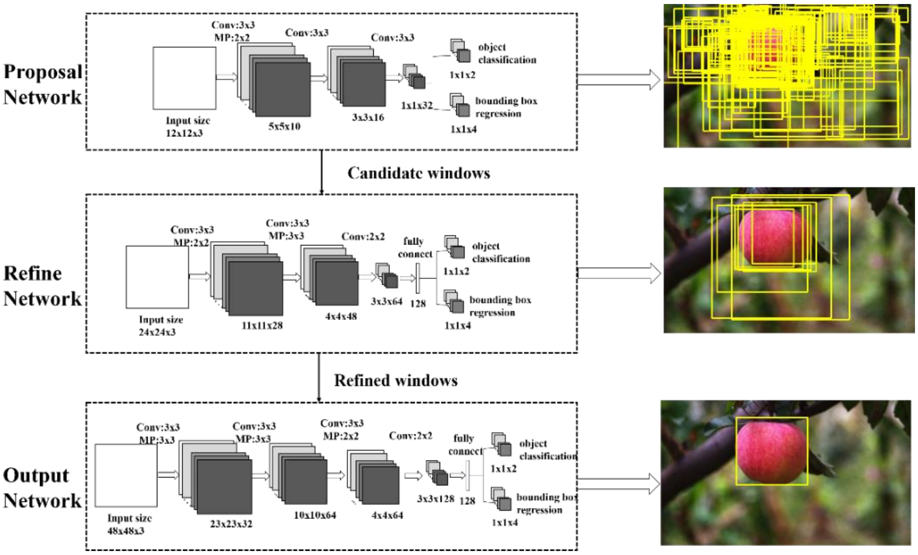

FIGURE 2. The overall architecture of the detection network. (top) the proposal network, (middle) the refine network, (bottom) the output network. "Conv" represents convolution and "MP" denotes max pooling.

The model first stage is the proposal network (PNet). This stage includes only three CNN layers to extract features and plays a key role for the proceeding two networks. After object classification and bounding box regression, this stage can obtain a high amount of candidate windows. To reduce the highly overlapped candidates, this stage exploits the non-maximum suppression method before furnishing the final output. Then these candidate windows from the first stage become inputs for the second stage, called refine network (RNet). At this stage, a lot of false candidate windows from stage one are rejected, and the candidate bounding boxes are calibrated. The third one is the output network (ONet) stage. The inputs to this stage come from the output of RNet. The significance of this stage is that it further rejects false candidate windows and obtains precise regression bounding boxes. These three networks gradually identify fruits against their background and obtain precise bounding boxes of these objects. The change in size of this cascaded convolutional networks work can be considered as a pyramid architecture.

C. LOSS FUNCTION

The loss function for fruit detection consists of two parts viz. classification and regression. We train F-MTCNN to acquire the classification of fruit or non-fruit objects, and then use regression bounding box for the fruit location detection.

1) FRUIT CLASSIFICATION

The classification task is to distinguish fruits from the background, so it can be regarded as a two-class classification problem. Thus, we exploit cross-entropy loss for each sample $\boldsymbol{x_i}$.

$$\boldsymbol{L_i^{cls} = -\left(y_i^{cls}\, log(p_i) + (1 - y_i^{cls})(1 - log(p_i))\right)} \qquad (1)$$

where $y_i^{cls} \in \{0, 1\}$ present ground-truth value, p_i is the probability of the input sample x_i , being a fruit.

2) BOUNDING BOX REGRESSION

The bounding box regression is to reduce the location information of each candidate window that is predicted by the detector to the nearest ground-truth. Here, each bounding box includes four coordinates i.e. left, top, height and width. So

$$L_i^{reg} = \left\| \hat{y}_i^{reg} - y_i^{reg} \right\|_2^2 \qquad (2)$$

Where $\hat{y}_i^{reg}$ represents the bounding box predicted by the detector, and y_i^{reg} is the ground truth object location.

D. TRAINING

For each unit network, there are fruit, partially fruit and non-fruit images for training. The overall loss function is as follows in Eq (3);

$$\mathrm{C} = \Sigma_{i=1}^{N}(\lambda_1 \alpha_i^{cls} L_i^{cls} + \lambda_2 \alpha_i^{reg} L_i^{reg}) \qquad (3)$$

Where $\boldsymbol{\alpha_i^{cls}} \in \{\boldsymbol{0}, \boldsymbol{1}\}$ denotes the input sample type of $\boldsymbol{x_i}$. We used $\boldsymbol{\lambda_1 = 1}$, $\boldsymbol{\lambda_2 = 0.5}$ for PNet, $\boldsymbol{\lambda_1 = 1}$, $\boldsymbol{\lambda_2 = 0.6}$ and $\boldsymbol{\lambda_1 = 1}$, $\boldsymbol{\lambda_2 = 0.7}$ for RNet and ONet respectively, to obtain more accurate bounding box.

IV. DATASET DESCRIPTION

A. IMAGE ACQUISITION

In this work, nearly 1800 images were collected from Beijing, China, using Canon EOS 100D camera. Besides, some 316 supplemental images from Internet and 511 from ImageNet (an open source database) [53] were acquired for diversity of samples. Specifically, each of these images contained at least one object of fruit, and the maximum count of fruits per image was 28. All the objects were labelled manually as individual image datasets.

B. TRAINING DATA

After acquisition, we divided these image into three different types, i.e. negative, positive or partial fruit samples. This division was made on the basis of the Intersection-over-Union (IoU) value with the ground truth, as shown in TABLE I.

TABLE I
IOU VALUE FOR THREE TYPE OF SAMPLES DIVISION

Sample	IoU
Positive	≥ 0.7
Partial	≥ 0.45 ∩ < 0.7
Negative	< 0.2

C. FUSION AUGMENTATION

Accumulating diverse samples for the model training can improve the performance of the detector. However, collecting a sufficient amount of samples is a tedious and time-consuming task, and it is not convenient to train the detector with a new category. Therefore, many studies have been conducted on augmentation methods such as rotation, translation, scaling, adding Noise and so on, and these methods have improved the performance to some extent as well. Through these methods, we found that the augmented samples were very close to the real environment and they improved the final performance as well. Motivated by this, we randomly extracted several sizes of negative patches from the original images. After that, we augmented our dataset by the fusion of positive or partial objects with one or several of small size negative patches. To the best of our knowledge, this was the first time proposed augmentation by fusion (FA). The flowchart can be seen in **Fig.** 3.

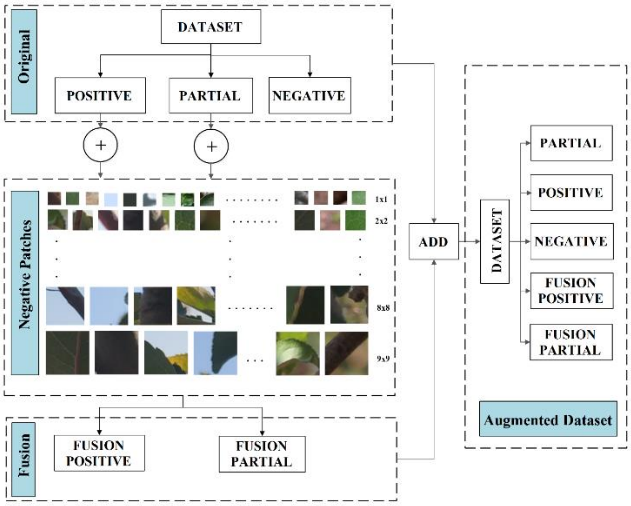

FIGURE 3. The illustration of our proposed fusion augmentation method.

In this article, we generated seven groups of Negative patches (NP) by randomly tailoring a fixed rectangle size from negative images of a dataset. The sizes of these seven NP groups are n × n pixels $n \in [1,9]$ $and\ n \in Z$. And we augmented our dataset by the following Algorithm:

Algorithm for FA:

```
Input: $I_j, j \in$ Positive or Part Positive image samples
  If: the size of sample is $12 \times 12$
    Count = random choose from 1 to 3
    If Count ≡ 1;        n = random choose size from 4 to 9
    If Count ≡ 2 or 3;   n = random choose size from 1 to 3
  Else if: the size of sample is $24 \times 24$
    Count = random choose from 1 to 6
    If Count ≡ 5 or 6;   n = random choose size from 1 to 3
    If Count ≡ 2 or 3;   n = random choose size from 4 to 6
    If Count ≡ 1;   n = random choose size from 7 to 9
  Else:
    Count = random choose from 3 to 8
    If Count ≡ 6 or 7 or 8;   n = random choose size from 1 to 3
    If Count ≡ 4 or 5;   n = random choose size from 4 to 5
    If Count ≡ 3 ;   n = random choose size from 6 to 9
Output: Fusion image $I_{jf} = \sum_{k=1}^{count} I_j \times p_n$ $p_n \in$ random choose one patch from the group of n
```

V. EXPERIMENTS

A. EVALUATION METHOD

In this work, we utilized true positive rate (TPR) and false positive rate (FPR) as evaluation method for fruit detection. And the TPR, FPR can be computed as:

$$TPR = \frac{TP}{TP+FN} \qquad \mathrm{FPR} = \frac{FP}{FP+FN} \tag{4}$$

where TP represents the number of correct detection results, FP the number false detections, and FN the number of missing objects.

B. ENVIRONMENTS AND TRAINING

We conducted our experiments through an Ubuntu 16.04 64-bits PC, equipped with an Intel(R) Core (TM) i5-7500 CPU @ 3.20GHz processor having 8 GB-RAM. We used NVIDIA (R) GeForce GTX 1060 graphics card having 3GB of memory to reduce our training time. The implementation of this fruit detection architecture used TensorFlow, which is an open-sourced deep learning framework developed by Google Brain Team. Besides this, we utilized Python as the programming language to adapt to the structure of TensorFlow. The time cost for each network is shown in **Table** II. As obvious from the table, the time cost of training for this whole detection network is about 2.5 hours. This low time consuming detection network is conducive to be applied in other fields as well.

TABLE II
TIME COST FOR EACH UNIT NETWORK

Model	Epochs	Time cost (minutes)
PNet	16571	98
RNet	6048	45
ONet	3726	25
Total	26345	168

C. COMPARISON WITH AUGMENTED DATASET

In order to know the function of FA in detail, we studied a fusion of the datasets of PNet, RNet and ONet in combination. The number of each unit patch of original dataset and that augmented by FA is presented in **Table** III.

TABLE III
A COMPOSITION OF THE ORIGINAL AND FUSION DATASET

Name	size	Original	Fusion
Negative	12x12	96337	--
	24x24	96206	--
	48x48	95071	--
Partial	12x12	64682	90554
	24x24	16198	80990
	48x48	16365	65460
Positive	12x12	23003	92012
	24x24	5712	28560
	48x48	5532	33193

To verify FA, we took our experimental datasets of all the three network units (PNet, RNet and ONet) and combined them with two other datasets, i.e. one was a fusion of these three networks (Fusion ALL), and the other was our original dataset. We trained the detector on these different datasets, and the True Positive Rate (TPR) result is shown in **Fig.** 4.

From the result we can observe that the model trained on all augmented networks by FA can improve true positive rate of 0.05 compared with the model trained on original dataset. While, only fusion of these networks got lower True Positive Rate when False positive samples were less than 100. This demonstrated that, if the detector is trained on one of these network units (PNet, RNet, ONet) individually, it will only affect part of result when used on a small number of false positive samples. However, with the increase in the number of false samples, these differences gradually decrease.

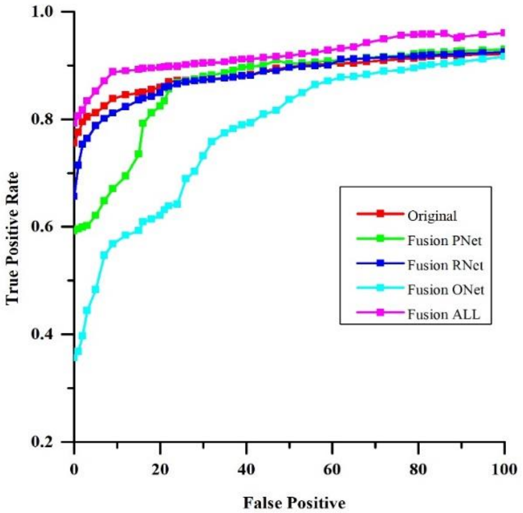

FIGURE 4. The true positive rate for the detection model trained on different datasets.

D. COMPARISON WITH DIFFERENT THRESHOLDS

Threshold value plays a key role in getting the final results and how to choose an appropriate threshold is very crucial. So, we set the threshold equal to 0.5, 0.6, 0.7, 0.8 and 0.9, and verified the detection model on the same test dataset with the five groups having different thresholds. The number of true positive samples and false positive samples are shown in **Fig.** 5.

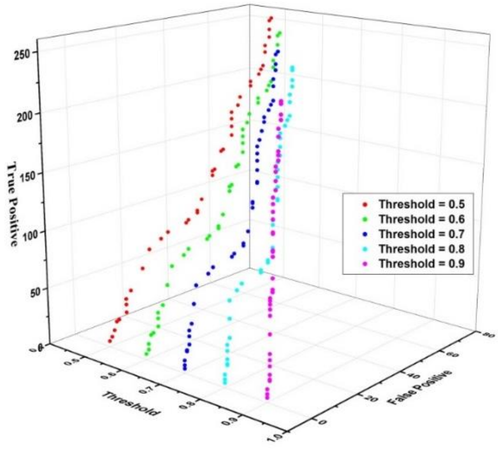

FIGURE 5. The number of the true positive samples and false positive samples for the detection model with different threshold value

From the figure, we can see that when the threshold value decreases from 0.9 to 0.5, the number of true positive samples increases, however, the number of false positive samples also increases. This result means that if the network threshold is weak, then there is a high probability of showing extra wrong objects and with a high threshold there is a chance of missing true objects. Therefore, we continued further experimentation on the TPR and FPR with the same conditions, as shown in **Fig.** 6 and **Fig.** 7. **Fig.** 6, reveals that the false positive rate was less than 0.2 when the threshold was 0.9. When the threshold decreased to 0.8, the false positive rate went above 0.5 sharply. Moreover, when the threshold value was 0.5, 0.6 or 0.7, the FPR raised to almost 0.7. Further, **Fig.** 7 shows that the TPR was as high as 0.98 when the threshold was 0.9. The other four groups threshold resulted in values below 0.9. After all the considerations, we adopted 0.9 as threshold value for our model.

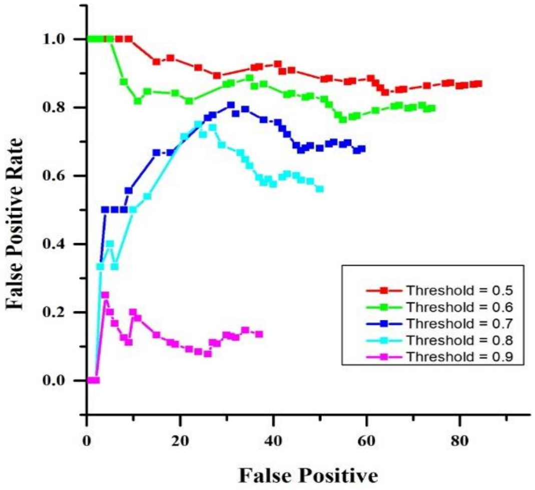

FIGURE 6. The false positive rate for the detection model with different threshold value

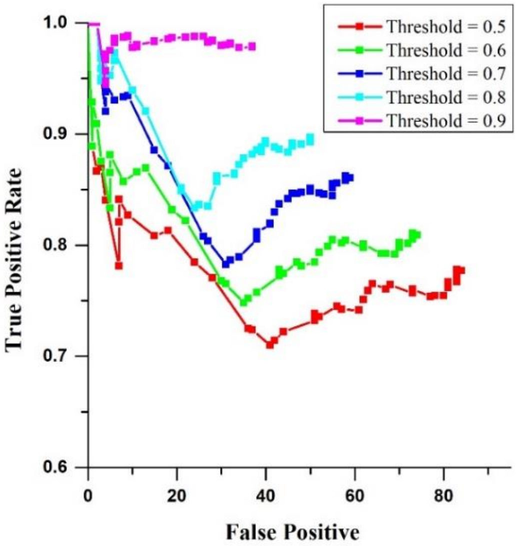

FIGURE 7. The true positive rate for the detection model with different threshold value

E. COMPARISON AT DIFFERENT STATUSES

It is known that when the objects are with minor disturbance or occlusion, the results detected by a model are improved and close to the real situation. However, the detection results will be abruptly affected when the object environment is complex or with heavily occlusion. To verify and analyze the performance of the model further, we divided the test images into three levels according to the complexity of the environment and/or the severity of occlusion. These three levels were easy, medium and hard, and we conducted further experiments on these three levels. **Fig.** 8, **Fig.** 9 and **Fig.** 10 show the results of the experiments at easy, medium and hard levels respectively.

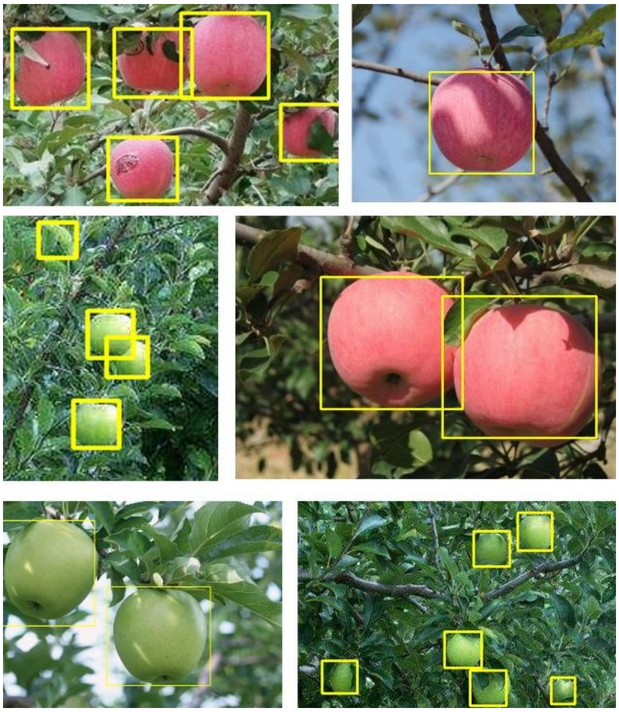

FIGURE 8. **Some examples of apple detection results at the easy level. The yellow boxes are the results of our detector.**

Fig. 8, indicates that for the fruits with no or less occlusion, the detector provided high accuracy, whether that was for counting the total number or the precision in position of each object. **Fig.** 9 demonstrates that these samples of foliage or tree branches were taken at a medium occlusion. Here some of the detected objects images were taken with varying light intensities and some with changes in their scale and sizes. In this case, the detector also achieved high accuracy, both on account of the total number and precision in the position of each object. Similarly, we took images with heavy occlusion and/or strong variations in light intensity as the hard level images. We can observe from **Fig.** 10 that in such conditions, although the detector missed some objects and got some error too, it detected most of the objects correctly, overall.

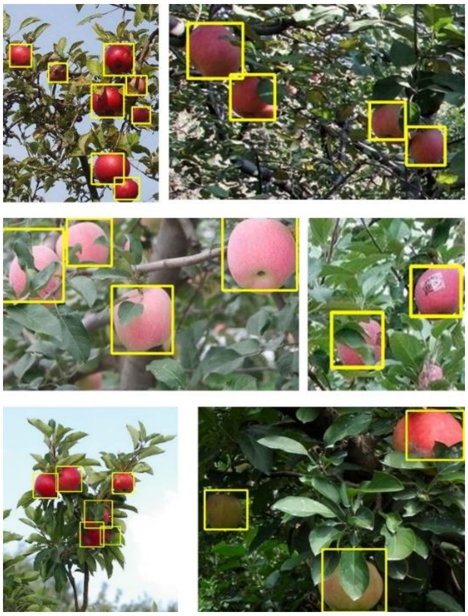

FIGURE 9. **Some examples of apple detection results at the medium level. The yellow boxes are the results of our detector.**

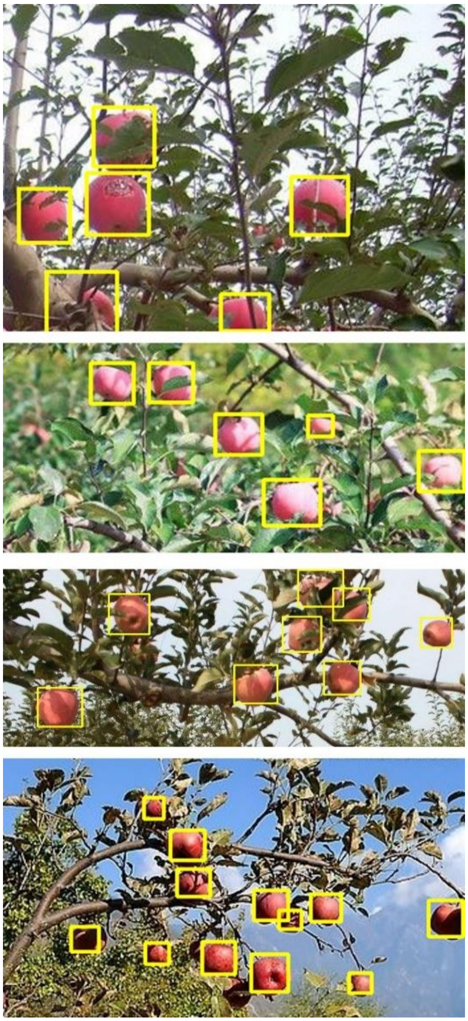

FIGURE 10. **Some examples of apple detection results at the hard level. The yellow boxes are the results of our detector.**

F. COMPARISON WITH OTHER KINDS OF FRUITS

In order to verify whether the detector can adapt to other fruits conveniently, we carried out experiments on strawberry and orange. We obtained the clear and usable images of the two fruit species from ImageNet dataset [53]. The number of images for both the fruit species can be found in each respective batch of **Table** IV**.** We trained our model on both the fruits images datasets.

TABLE IV
THE COMPOSITION OF STRAWBERRY AND ORANGE DATASET

Name	size	Strawberry	Orange
Negative	12x12	54418	107967
	24x24	54333	107846
	48x48	53873	107330
Partial	12x12	40345	84388
	24x24	10100	21033
	48x48	10144	21081
Positive	12x12	10791	26663
	24x24	2661	6715
	48x48	2711	6773

We took some untrained images from ImageNet dataset [53] and also collected some from Images of Baidu as the test dataset. We trained our model on these two datasets. After training and validation some samples of strawberry and oranges are presented in **Fig.** 11 and **Fig.** 12.

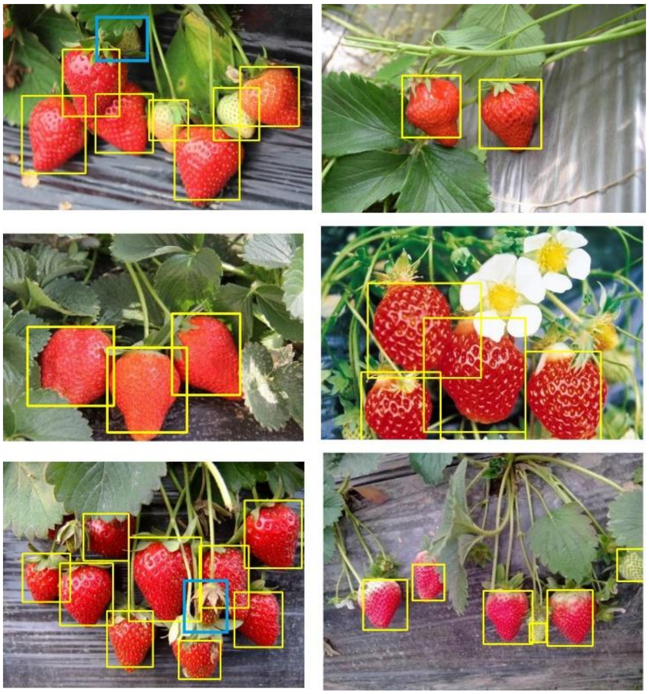

FIGURE 11. Some examples of strawberry detection results. The yellow boxes are the results of our detector and the blue boxes are the missing ones.

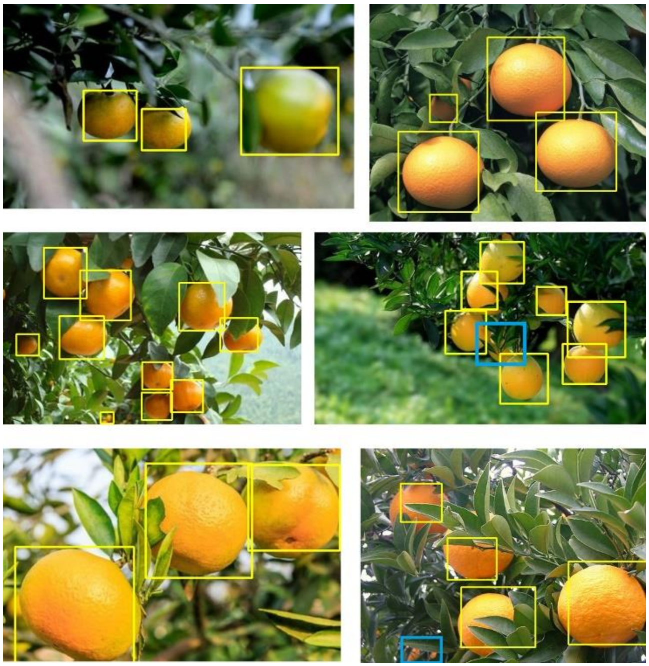

FIGURE 12. Some examples of orange detection results. The yellow boxes are the results of our detector and the blue boxes are the missing ones.

From above two figures, it is clear that most of the fruits can be correctly detected. This demonstrated that detector can be feasibly adapted for other kinds of fruits, though there is still room for perfection.

G. TIME COST

Time-cost is one of the very important indexes for a detector. This is because the automatic agricultural equipment needs to collect and analyze the image and make decision in real-time. We conducted our experiment on twelve different groups of images. Each group included one hundred images. Then we tested our model on these twelve groups separately. The results of our experiment are presented in **Fig.** 13.

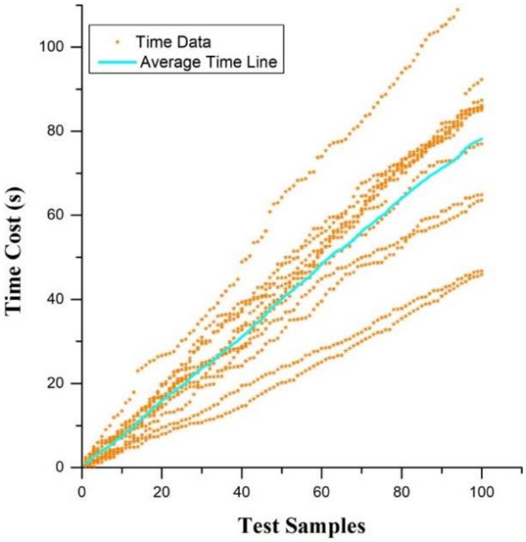

FIGURE 13. Time cost for the detector tested on twelve different groups of images.

Here, we analyze and discuss the detection result presented in the previous section. As shown in **Fig.** 11 and **Fig.** 12, there are some false positive and false negative samples. The small amount of dataset is probably one reason for that. Although we proposed a novel augmented method to improve the performance, the small number of training data limits the final result. Our detection model is based on deep learning method, which need sufficient samples to adjust parameters of a network. The diversity of size for detection network is another reason. If there is a great gap between test set and training set, the detector may miss some objects.

VI. CONCLUSION

In this study, we exploited a multi-task cascaded convolutional networks based detector for fruit detection. We chose apple for our study and collected more than one thousands of images from apple orchards and labeled them. Alongside this, we also added an appropriate amount of supplementary images from internet and ImageNet dataset to create a dataset. Furthermore, we proposed a novel augmented method called fusion augmentation. The comparative experiment results demonstrated that this augmented method can improve the final result. To verify whether the detector could be applied to other kinds of fruits as well, we selected strawberry and orange as two other test fruits. The dataset for training was obtained from ImageNet dataset, which contains hundreds of images. Our results showed that the detector can conveniently adapt to other kinds of fruit as well. Finally, we tested the detector on twelve groups of images with different resolutions. Each group had one hundred images. The average time cost of the detector was less than 80 seconds per one hundred images, which is very close to real-time response.

VII. FUTURE WORK

We find proposed multi-task cascaded convolutional networks based fruit detector have good performance of timeliness and accuracy to meet the requirements for the visual system of harvesting robot from the experimental results. However, there is still a long distance for practical application and promotion of the harvesting robot. One of the most important task is to determine the order for all detected fruits. In other words, is to decide which object should be first considered for picking. Compared with picking manually, by human visual attention can solve this kind of problem effectively. On the basis of this study, we will focus on the study and mimic the human visual attention when viewing the scene by relevant studies such as visual saliency detection and semantic segmentation.

In future, we will also study the characteristics of fruit deeply and design a more reasonable and effective network model for fruit recognition tasks. Besides this, improving and optimizing the accuracy of the detector is also an important task for the future.

REFERENCES

[1] L. Ma, "Deep Learning Implementation using Convolutional Neural Network in Mangosteen Surface Defect Detection," *7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE).*, no. November, pp. 24–26, 2017.

[2] A. Mohapatra, S. Shanmugasundaram, and R. Malmathanraj, "Grading of ripening stages of red banana using dielectric properties changes and image processing approach," *Comput. Electron. Agric.*, vol. 143, no. 382, pp. 100–110, 2017.

[3] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, "An in-field automatic wheat disease diagnosis system," *Comput. Electron. Agric.*, vol. 142, pp. 369–379, 2017.

[4] J. Ma, K. Du, L. Zhang, F. Zheng, J. Chu, and Z. Sun, "A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing," *Comput. Electron. Agric.*, vol. 142, pp. 110–117, 2017.

[5] N. Behroozi-Khazaei and M. R. Maleki, "A robust algorithm based on color features for grape cluster segmentation," *Comput. Electron. Agric.*, vol. 142, pp. 41–49, 2017.

[6] W. Mao, B. Ji, and J. Zhan, "Apple Location Method For the Apple Harvesting Robot," *2nd International Congress on Image and Signal Processing.*, pp. 0–4, 2009.

[7] S. Vougioukas and D. C. Slaughter, "Real-time segmentation of strawberry flesh and calyx from images of singulated strawberries during postharvest processing," *Comput. Electron. Agric.*, vol. 142, pp. 298–313, 2017.

[8] Y. Shi, W. Huang, J. Luo, L. Huang, and X. Zhou, "Detection and discrimination of pests and diseases in winter wheat based on spectral indices and kernel discriminant analysis," *Comput. Electron. Agric.*, vol. 141, pp. 171–180, 2017.

[9] J. Lu, W. Suk, H. Gan, and X. Hu, "ScienceDirect Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis," *Biosyst. Eng.*, vol. 171, pp. 78–90, 2018.

[10] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization : A review," *Comput. Electron. Agric.*, vol. 116, pp. 8–19, 2015.

[11] Tian Zhou, Sujuan Yang, Lei Wang, Jiming Yao, Guan Gui, "Improved Cross-Label Suppression Dictionary Learning for Face Recognition," *IEEE Access*, vol. 6, no. 1. pp. 48716 – 48725, 2018.

[12] S. Liao, A. K. Jain, and S. Z. Li, "A Fast and Accurate Unconstrained Face Detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 211–223, 2016.

[13] C. Zhu, "Towards a Deep Learning Framework for Unconstrained Face Detection." *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS).*, 6-9 Sept. 2016.

[14] Zhang, Kaipeng, et al. "Detecting faces using inside cascaded contextual cnn." *Proceedings of the IEEE International Conference on Computer Vision.*, pp. 3171-3179, 2017.

[15] Z. Yang and R. Nevatia, "A Multi-Scale Cascade Fully Convolutional Network Face Detector." *23rd International Conference on Pattern Recognition (ICPR).*,4-8 Dec. 2016.

[16] K. Zhang, Z. Zhang, Z. Li, S. Member, Y. Qiao, and S. Member, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, no. 1, vol. 23, pp. 1499 – 1503, 2016.

[17] Ji W, Meng X, Tao Y, et al. "Fast segmentation of colour apple image under all-weather natural conditions for vision recognition of picking robots," *International Journal of Advanced Robotic Systems.*, no. 1, vol. 13, pp. 1 – 24, 2016.

[18] H. Dang, J. Song, and Q. Guo, "A Fruit Size Detecting and Grading System Based on Image Processing," *2010 Second Int. Conf. Intell. Human-Machine Syst. Cybern.*, vol. 2, pp. 83–86, 2010.

[19] I. B. Mustaffa, S. Fikri, and B. Mohd, "Identification of Fruit Size and Maturity Through Fruit Images Using OpenCV-Python and Rasberry Pi." *International Conference on Robotics, Automation and Sciences (ICORAS).*, 27-29 Nov. 2017

[20] G. Moradi, "Fruit defect detection from color images using ACM and MFCM algorithms," *2011 Int. Conf. Electron. Devices, Syst. Appl.*, pp. 182–186, 2011.

[21] A. D. Aggelopoulou, D. Bochtis, S. Fountas, K. C. Swain, T. A. Gemtos, and G. D. Nanos, "Yield prediction in apple orchards based on image processing," *Precision Agriculture*, vol. 12, pp. 448–456,

2011.

[22] W. Ji, Z. Qian, B. Xu, Y. Tao, D. Zhao, and S. Ding, “Apple tree branch segmentation from images with small gray-level difference for agricultural harvesting robot,” *Optik - International Journal for Light and Electron Optics.,* vol. 127, pp. 11173–11182, 2016.

[23] A. Rady, N. Ekramirad, A. A. Adedeji, M. Li, and R. Alimardani, “Postharvest Biology and Technology Hyperspectral imaging for detection of codling moth infestation in GoldRush apples,” *Postharvest Biology and Technology.,*vol. 129, pp. 37–44, 2017.

[24] D. M. Bulanon, T. F. Burks, and V. Alchanatis, “Image fusion of visible and thermal images for fruit detection,” *Biosyst. Eng.*, vol. 103, no. 1, pp. 12–22, 2009.

[25] C. S. Nandi, B. Tudu, C. Koley, A. Seasonal, and M. Indica, “A Machine Vision-Based Maturity Prediction System for Sorting of Harvested Mangoes,” *IEEE Trans. Instrum. Meas.*, vol. 63, no. 7, pp. 1722–1730, 2014.

[26] X. Xu, D. Niu, Q. Wang, P. Wang, and D. D. Wu, “Intelligent Forecasting Model for Regional Power Grid With Distributed Generation,” *IEEE Systems Journal.,* vol. 11, no. 3, pp. 1836–1845, 2017.

[27] A. Rojas-dom ńguez, L. C. Padierna, J. Mart ń, C. Valadez, H. J. Puga-soberanes, and H. J. Fraire, “Optimal hyper-parameter tuning of SVM classifiers with application to medical diagnosis,” *IEEE Access,* vol. 6, pp. 7164 – 7176, 2017.

[28] Z. Ma, Y. Lai, W. B. Kleijn, Y. Z. Song, L. Wang, and J. Guo, “Variational Bayesian Learning for Dirichlet Process Mixture of Inverted Dirichlet Distributions in Non-Gaussian Image Feature Modeling,” *IEEE Trans. Neural Networks Learn. Syst.,* vol. PP, pp. 1–15, 2018.

[29] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, “Variational bayesian matrix factorization for bounded support data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 876–889, 2015.

[30] J. Han, K. N. Ngan, M. Li, and H. Zhang, “Unsupervised Extraction of Visual Attention Objects in Color Images,” *IEEE Transactions on Circuits and Systems for Video Technology.*, vol. 16, no. 1, pp. 141–145, 2006.

[31] S. Benalia, S. Cubero, J. M. Prats-montalb án, B. Bernardi, G. Zimbalatti, and J. Blasco, “Computer vision for automatic quality inspection of dried figs (Ficus carica L .) in real-time,” *Comput. Electron. Agric.,* vol. 120, pp. 17–25, 2016.

[32] D. L. Borges, S. T. C. D. M. Guedes, A. R. Nascimento, and P. Melo-pinto, “Detecting and grading severity of bacterial spot caused by Xanthomonas spp . in tomato (Solanum lycopersicon) fields using visible spectrum images,” *Comput. Electron. Agric.,* vol. 125, pp. 149–159, 2016.

[33] Z. Ma, J. H. Xue, A. Leijon, Z. H. Tan, Z. Yang, and J. Guo, “Decorrelation of Neutral Vector Variables: Theory and Applications,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 1, pp. 129–143, 2018.

[34] H. Huang, J. Yang, H. Huang, Y. Song and G. Gui, “Deep Learning for Super-Resolution Channel Estimation and DOA Estimation based Massive MIMO System,” *IEEE Transactions on Vehicular Technology.*, vol. 67, no. 9, pp. 8549-8560, Sept. 2018.

[35] M. Liu, J. Yang, T. Song, J. Hu, and G. Gui, “Deep Learning-Inspired Message Passing Algorithm for Efficient Resource Allocation in Cognitive Radio Networks,” *IEEE Transactions on Vehicular Technology,* vol. 68, no. 1, pp. 641-653, Jan.2019.

[36] G. Gui, H. Huang, Y. Song,and H. Sari, “Deep Learning for an Effective Nonorthogonal Multiple Access Scheme,” *IEEE Transactions on Vehicular Technology.*, vol. 67, no. 9, pp. 8440-8450, Sept. 2018.

[37] M. Liu, T. Song, G. Gui, J. Hu, and H. Sari, “Deep Cognitive Perspective: Resource Allocation for NOMA based Heterogeneous IoT with Imperfect SIC,” *IEEE Internet of Things Journal*, 2018. doi: 10.1109/JIOT.2018.2876152

[38] Y. Li, X. Cheng, G. Gui, “Co-Robust-ADMM-Net: Joint ADMM Framework and DNN for Robust Sparse Composite Regularization,” *IEEE Access.*, vol. 6, pp. 47943-47952, 2018.

[39] F. Zhu, Z. Ma, X. Li, G. Chen, J. Chien, J. Xue, and J. Guo, “Image-text Dual Model with Decision Strategy for Small-sample Image Classification,” *Neurocomputing*, accepted, 2018.

[40] G. Cheng, P. Zhou, and J. Han, “Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, 2016.

[41] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, “Object Detection in Optical Remote Sensing Images Based on Weakly Supervised Learning and High-Level Feature Learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, 2015.

[42] D. Zhang, D. Meng, and J. Han, “Co-Saliency Detection via a Self-Paced Multiple-Instance Learning Framework,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, 2017.

[43] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, “Detection of Co-salient Objects by Looking Deep and Wide,” *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 215–232, 2016.

[44] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, “Background Prior-Based Salient Object Detection via Deep Reconstruction Residual,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1309–1321, 2015.

[45] Z. M. Khaing, Y. Naung, and P. H. Htut, “Development of Control System for Fruit Classification Based on Convolutional Neural Network,” *2018 IEEE Conf. Russ. Young Res. Electr. Electron. Eng.*, pp. 1805–1807, 2018.

[46] G. Zeng, “Fruit and Vegetables Classification System Using Image Saliency and Convolutional Neural Network,” *IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC).*, pp. 613–617, 2017.

[47] L. Hou and Q. Wu, “Fruit Recognition Based On Convolution Neural Network,” *2016 12th Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Discov.*, pp. 18–22, 2016.

[48] T. Nishi, “Grading Fruits and Vegetables Using RGB-D Images and Convolutional Neural Network,” *2017 IEEE Symposium Series on Computational Intelligence.*, no. 4, pp. 1–6, 2017.

[49] S. Bargoti and J. P. Underwood, “Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards,” *J. F. Robot.*, vol. 34, no. 6, pp. 1039–1060, 2017.

[50] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[51] S. Bargoti and J. Underwood, “Deep fruit detection in orchards,” *IEEE Int. Conf. Robot. Autom.*, pp. 3626–3633, 2017.

[52] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. Mccool, “DeepFruits : A Fruit Detection System Using Deep Neural Networks,” *Sensors,* vol. 16, no. 8, 2016.

[53] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, 2009, pp. 248-255.