YOLO Models for Fresh Fruit Classification from Digital Videos

Yinzhe Xue, Wei Qi Yan

Auckland University of Technology, Auckland 1010 New Zealand

ABSTRACT

Identifying food freshness is a very important and a long history action for our humans, because fruit freshness can tell us the information about the quality of foods with the advancement of machine learning and computer science, which will be broadly employed in factories and markets, instead of manual classification. Recognizing the freshness of food is rapidly being replaced by computers or robots. In this book chapter, we conduct the research work on fruit freshness detection, we make use of YOLOv6, YOLOv7, and YOLOv8 in this project to implement fruit classifications based on a variety of digital images, this can incredibly improve the efficiency and accuracy of the classification, after the classification, the output will showcase the result of fruit fresheness classification, namely, fresh, or rotten, etc. We also compare the results of different deep learning models to discover which architecture is the best one in terms of speed and accuracy. At the end of this book chapter, we made use of the majority vote method to combine the results of different models to get better accuracy and recall scores. To generate the final result, we trained the three models individually, and we also propose a majority vote to get a better performance for fresh fruit detection. Compared with the previous work, our method has higher accuracy and much faster speed than the previous methods, because we use the clustering method to generate the final result, it will be easy for us to change the backbone and get a better result in the future.

Keywords: Fruit classification, freshness detection, ensemble method, YOLOv8, orange, apple, banana

INTRODUCTION

Fruit freshness detection is a very interesting topic in machine vision that is also a very important task for human ordinary lives, because every day we need to know which food is safe to be eaten, which will cause illness or diseases, rotten foods may lead to poisoning, hence, we develop a number of ways to classify fruits and detect as well as predict the freshness.

In this book chapter, we use deep neural networks for freshness and rotten fruit classification, YOLO is a very famous architecture that can be employed for almost all types of fruit classification and freshness detection, meanwhile, we also introduce the potential methods such as Transformers in this project. This project will mainly make use of deep learning methods (like YOLO) to classify the digital images for fresh or rotten fruits, we take advantage of three YOLO models and compare the results.

The focus of this book chapter is to detect fruit freshness or classify fresh and rotten fruits from

the input digital images. According to the most advanced YOLO architecture, it will be easy for us to get a high precision and recall for fruit freshness detection compared with the human labor method. Our contributions to this book chapter include: (1) Collecting a large dataset for three classes of fruits (i.e., apple, banana, and orange) (2) Classifying each image with YOLOv6, YOLOv7 or YOLOv8 models (3) Detecting the freshness of the given fruits (4) Proving machine learning methods to detect the freshness of the fruit (5) Seeking an ensemble method to combine the detection result from different architecture, and finding the best clustering weights for different architecture.

The structure of this book chapter is that we show our literature review and discuss the relevant studies of visual object detection and classification in Section 2. Meanwhile, we also introduce the details related to Transformer in deep learning. In Section 3, we introduce our research methods and dataset. In Section 4, we implement the proposed algorithms, collect experimental data and demonstrate our outcomes. Additionally, the limitations of these proposed methods will be detailed. In Section 5, we summarize and analyze the experimental results. We draw the conclusion and state our future work in Section 6.

We collected our dataset through the Kaggle website. In this dataset, we totally have six different types of fruits, there are images for fresh and rotten apples, bananas and oranges, and the number for each group is different. To give more information about the dataset, we will show the distribution of the dataset for this project.

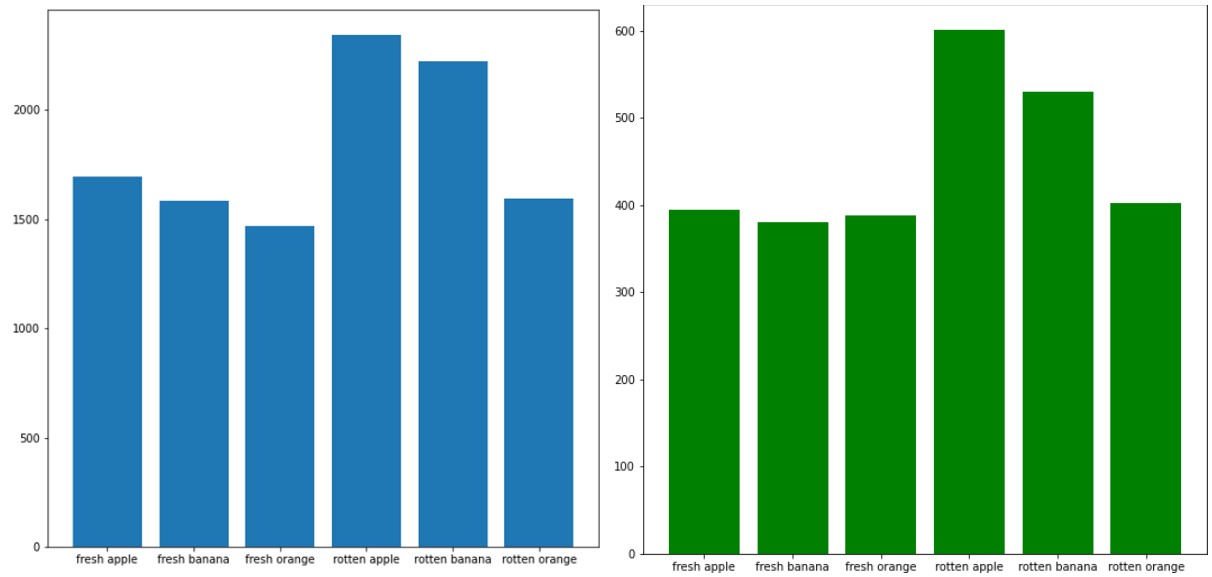
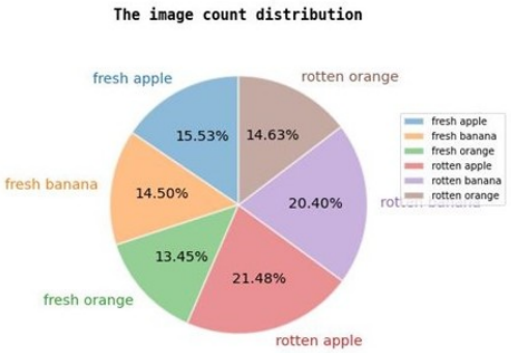


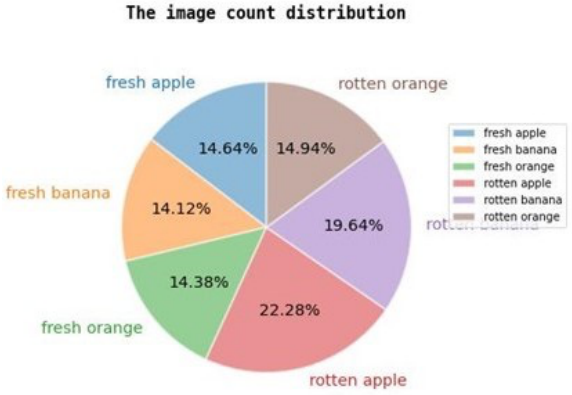
Fig.1. Our dataset (a) Training dataset (b)Test dataset

We take advantage of around 2% samples as the test set and the rest of samples will be used as our training set, Fig. 1. Shows the distribution of the samples in our training set and test set. We see that the training and test dataset almost have the same distributions of the samples. We also show the pie chart for the training and testing dataset to offer a more intuitive reflection of the dataset for this project.



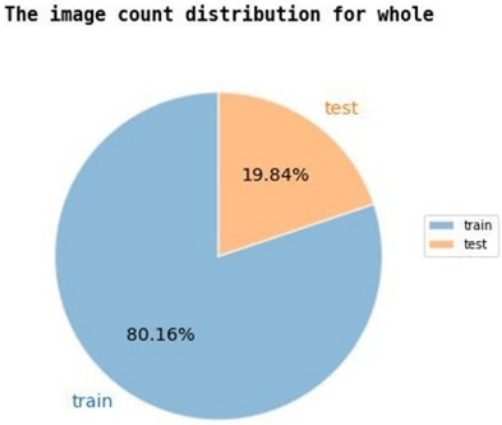
(a)

Pie chart for training dataset



(b)

Pie chart for test dataset



The distribution for training and test datasets (c)

Fig.2. Pie charts for training and test dataset

RELATED WORK

YOLO (Fang, 2019; Fang, 2021; Redmon, 2021; Parico, 2021) is the abbreviation of "You Only Look Once", which is a very famous and widely employed in computer vision. There are a lot of advantages of YOLO models, firstly, YOLO is a "lightweight" architecture, which means it will be trained in a very fast way. The trained weights will not consume a large space. Secondly, YOLO can provide visual object detection in real time. The real-time is the network which can make detections on the input image much faster than human reaction, so it feels like the network "promptly" generates the result without delay. Compared to Faster R-CNN or Transformer model, the accuracy of YOLO is not high, the differences of the performance between YOLO and other models are pretty minor.

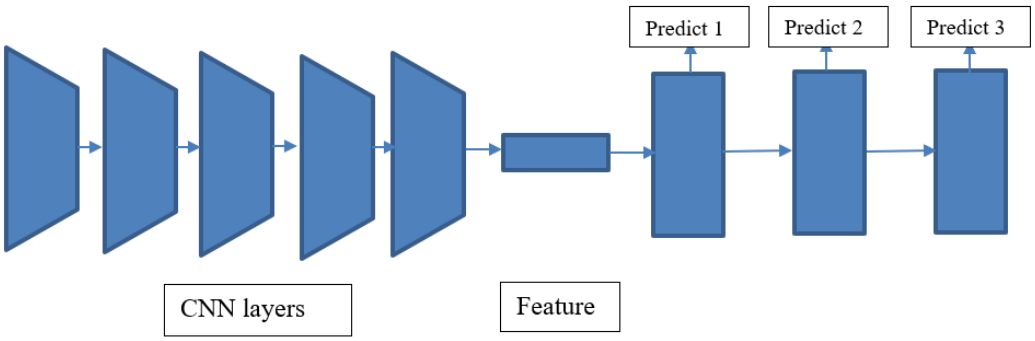


Fig. 3. YOLO architecture

In Fig. 3, the middle box is the residual layers, the right three blocks are the upsampling layer, which is accommodated to increase the size of the feature, the white box marked with "Predict" is the concatenate layer, this layer is offered to concatenate the output from the previous layers. There are four very important components in YOLO model: (1) Residual blocks (2) Bounding box regression (3) Intersection over Union (IOU) (4) Non-maximum suppression (NMS).

In the residual block, the main idea is to find the correct location of the visual object. To do this, it firstly splits the given image into regions, each with a dimension of $N \times N$. the network will perform the detection based on each grid. If an object appears in a grid, then the network will mark that grid as a candidate for final detection.

YOLO is the network with a single-box regression to predict the four pieces of information for a bounding box. Each grid cell is responsible for predicting the bounding boxes and their confidence scores. The IOU is equal to 1.0 if the predicted bounding box is as same as the real box. If we assume the blue square presents the prediction, and the red one is the ground truth. This mechanism eliminates bounding boxes that are not equal to the real box.

The red bounding box is the ground truth, we need to conduct object detection, and the blue square is the prediction that the architecture works on the image. On the right side, the green box indicates that our prediction is overlapped with the ground truth (GT).

Compared with the previous architectures, YOLOv6 is use of the same dataset to pre-train, but the backbone has been changed to EfficientRep, and the neck has been changed to Rep-PAN.

YOLOv7 has a higher accuracy for multiple tasks shown in the book chapter. The main difference of YOLOv7 is that YOLOv7 modifies the ELAN architecture (efficient layer aggregation network), Themodified ELAN network is called E-Ellan, which is just a combination of multiple convolution layers and two more concatenation layers.

YOLOv8 was published in 2022, which is the latest version of YOLO models. YOLOv8 is an open-source architecture. Leveraging previous YOLO models, YOLOv8 models are faster and more accurate, while providing a unified framework for training models to perform object detection, instance segmentation, and image classification.YOLOv8 is similar to YOPLOv7, but YOLOv8 replaces C3 architecture to C2f architecture (Lin, 2017), which has two backbones for YOLOv8. Pertaining to YOLOv8, we will take use of different backbones for the different tasks (like detection or classification). YOLOv8 looks like a combination of the previous YOLO architecture, but YOLOv8 is much faster than the previous versions of YOLO models.

Compared to the architecture of C3 and C2f, there are a few changes: (1) The kernel size of the first convolution layer is changed to 6×6 instead of 3×3; (2) Two convolution layers in neck are deleted; (3) The number of blocks in backbone has been changed to 3-6-6-3; (4) A split part is added beforebottleneck; (5) The residual connections are added between the input and output; (6) The kernel parameters are modified; (7) Instead of using the parallel bottlenecklayers, the serial connected layers are employed; (8) The head of YOLOv8 changes to the anchor-free architecture, but the previous YOLO is anchor-based architecture.

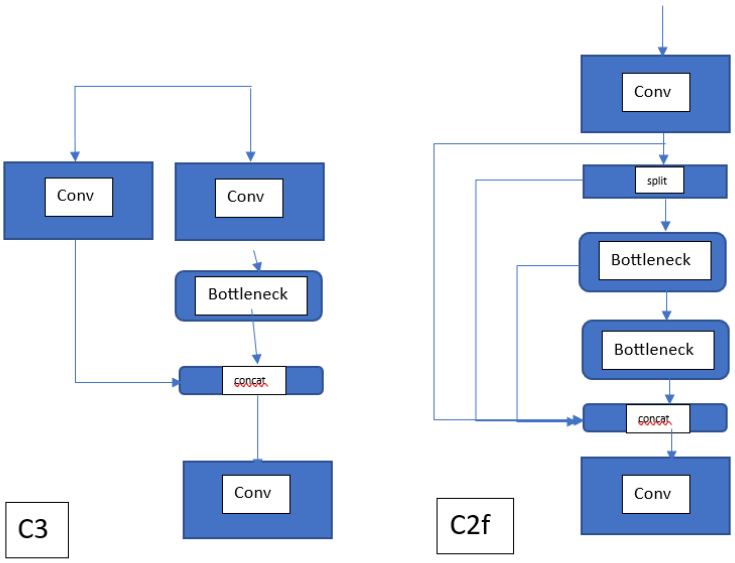


Fig.4.Architecture of C3 and C2f

OUR WORK

Data argumentation is a broadly used method to improve the network performance, it adds more types or more samples in the training set based on the existing training dataset, more types of training data will make the proposed model more robust.

Regarding image classification using data arguments, the normal method is always to rotate or flip the original images, because from the previous experiments, these two methods are very simple to implement and can obviously increase the performance of the proposed network. In this project, we also add a blurring method in our data augmentation. Pertaining to this blurring method, a Gaussian kernel is adopted to convolve the original image, then we get a blurred image with the same fruit location and class of the original input. Because we use kernels to process the image, it is also easy for us to undertake different degrees to the image and find out the best one to increase the network performance as much as possible as shown in Fig.5.

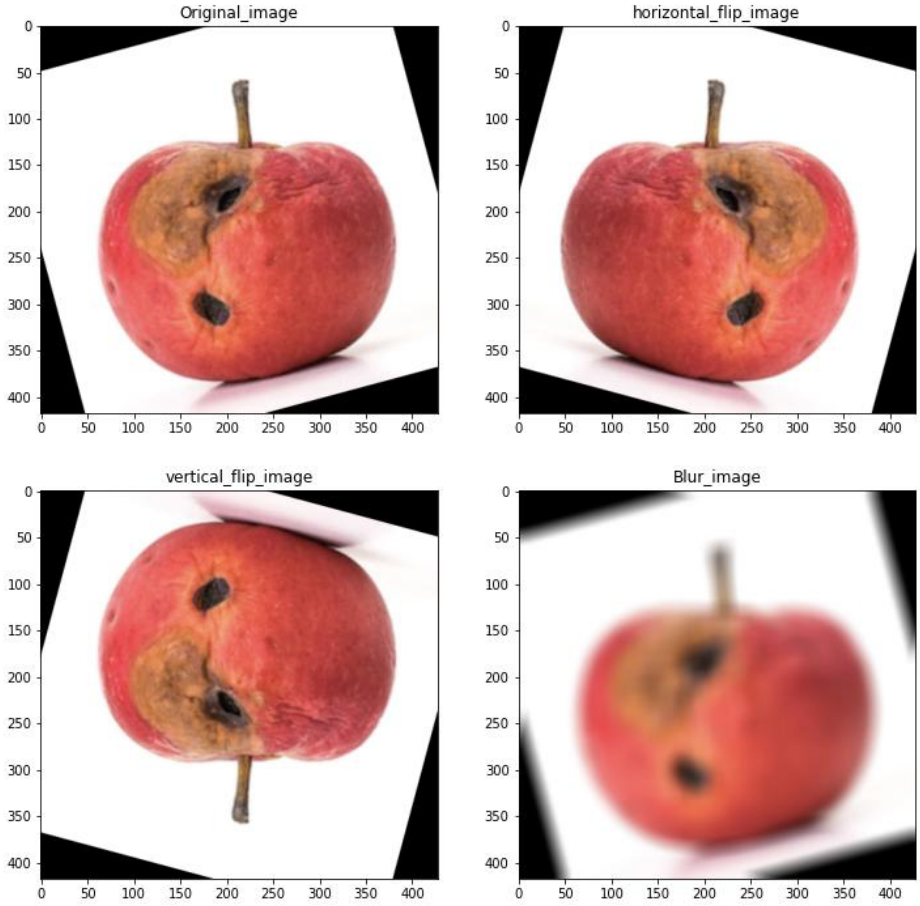
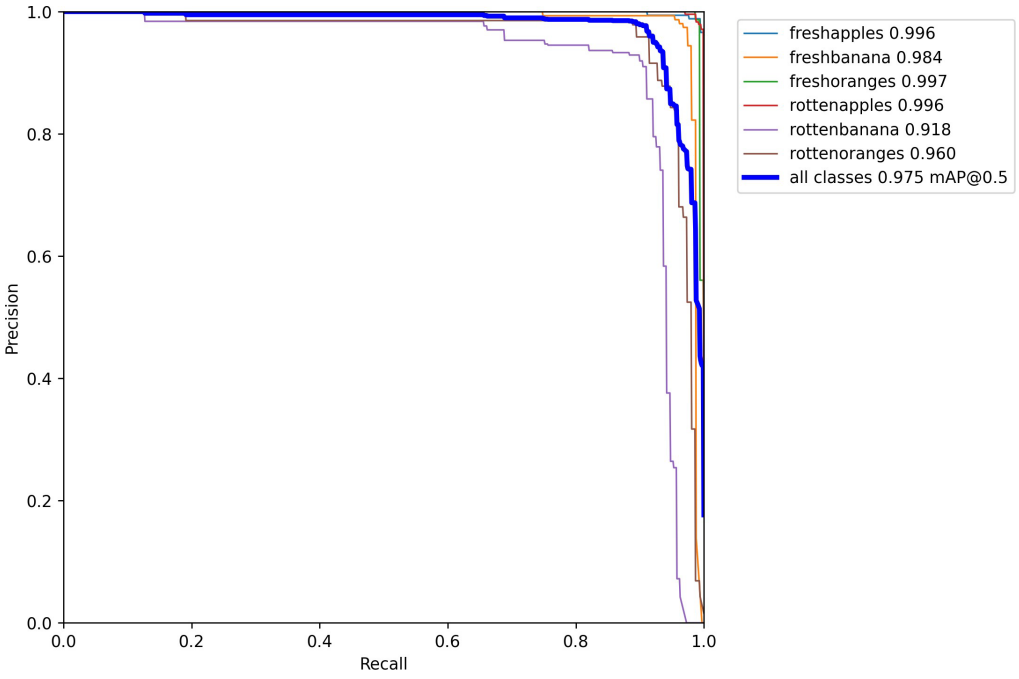


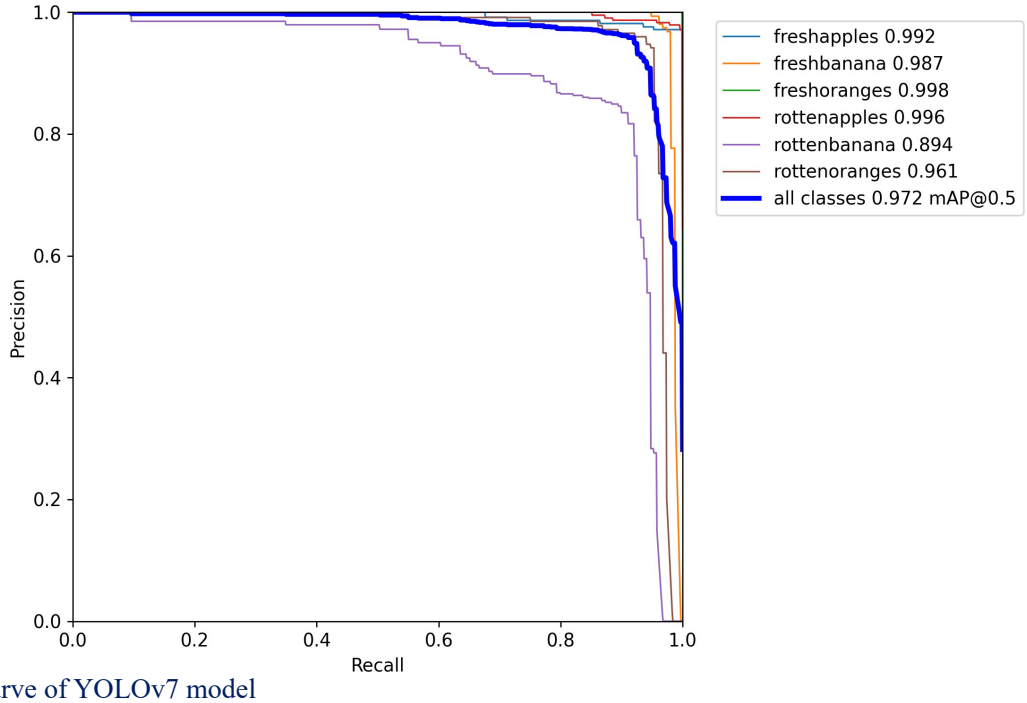
Fig.5. An example of image augmentation

In Fig. 5, the top left is the original input image, we add three different augmentations in our augmented dataset, namely, horizontal flipping, vertical flipping and image blurring, the two kinds of flips will add more visual appearances to the network, these visual features will show in the cases when we take photos for the object from different angles, we rotate the images (no only 180 degrees or 90 degrees), the step will add more “ambiguous” features to the architecture, then the network model will have the ability to classify visual objects in the images with a variety of shapes and colors, instead only detect a class of the fruits. From our experiments, the blurring augmentation will always aid the proposed model to increase the performance.

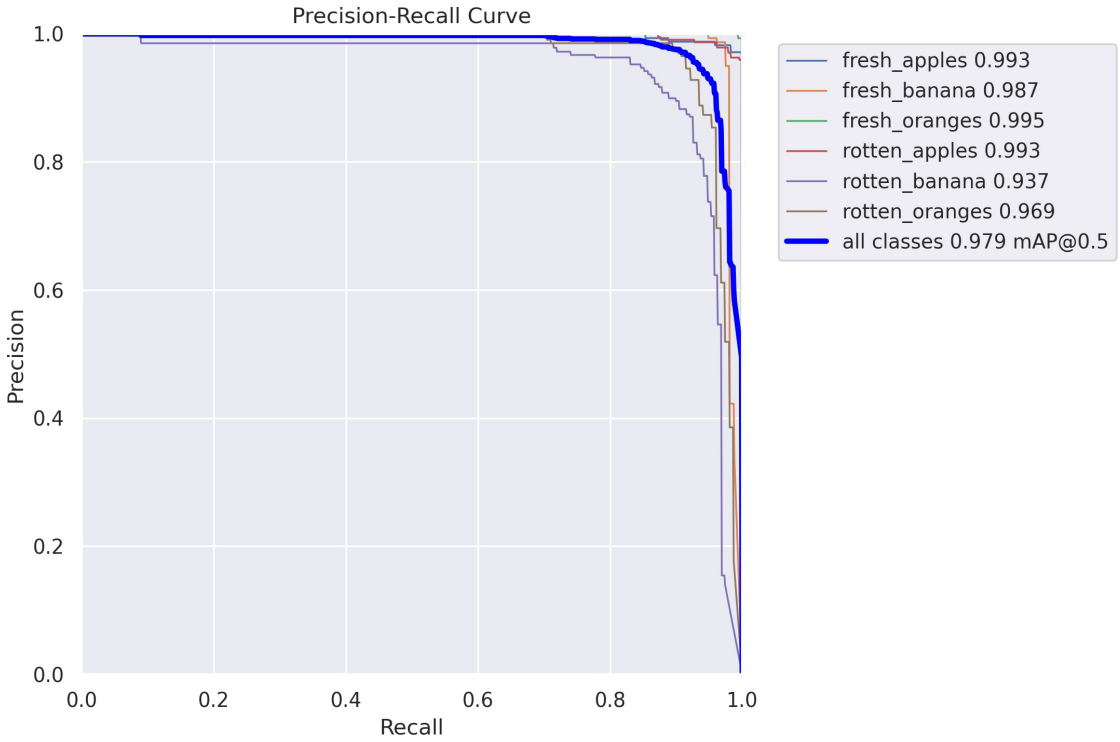
RESULT ANALYSIS



(a) PR curve of YOLOv6 model

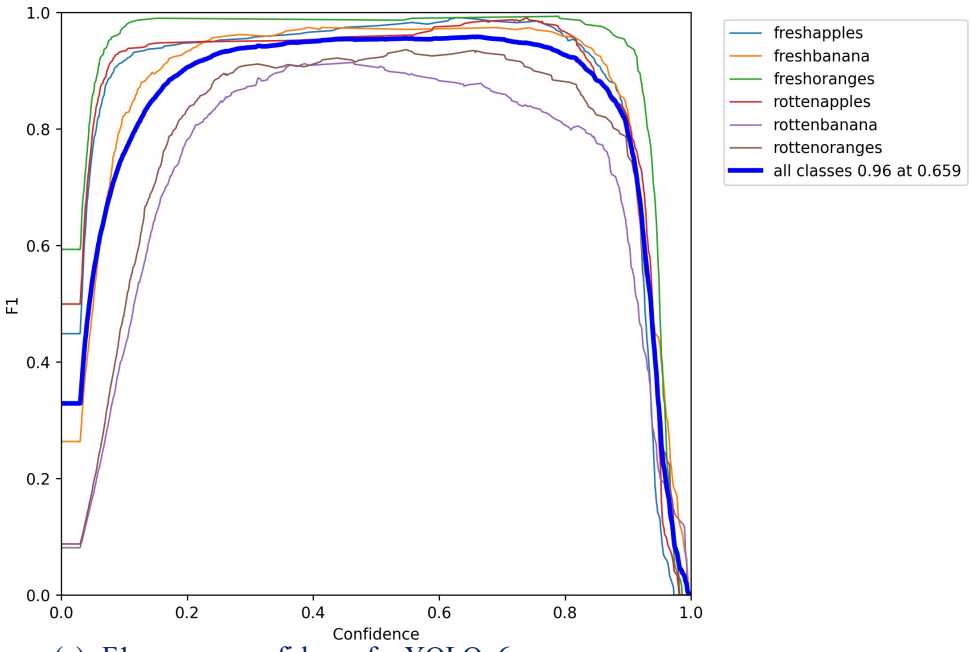


(b) PR curve of YOLOv7 model

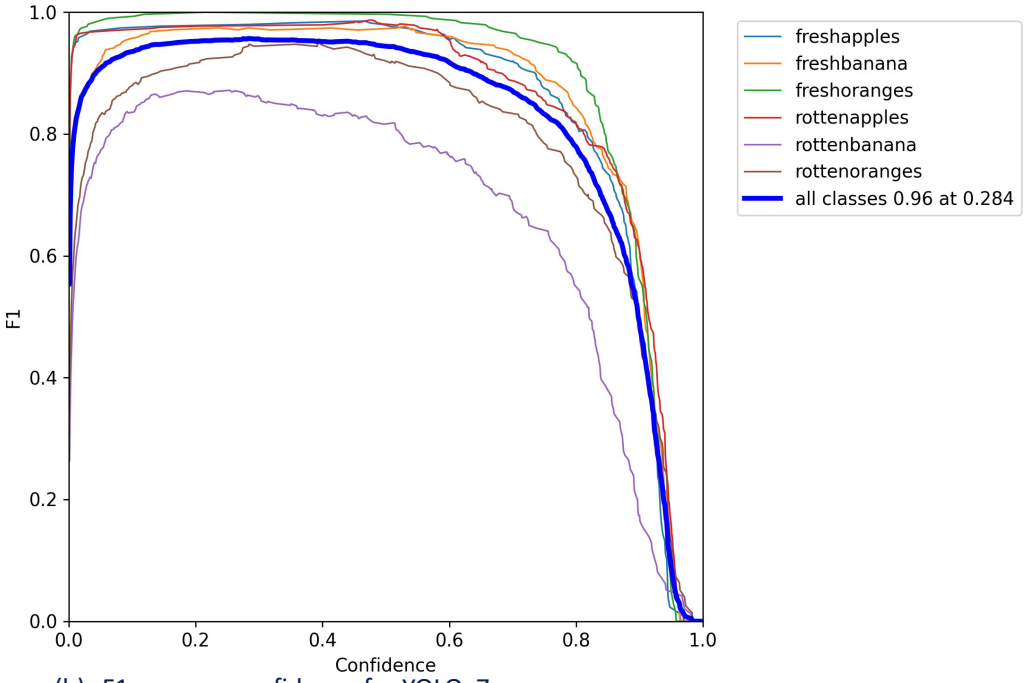


(c) PR curve of YOLOv8 model

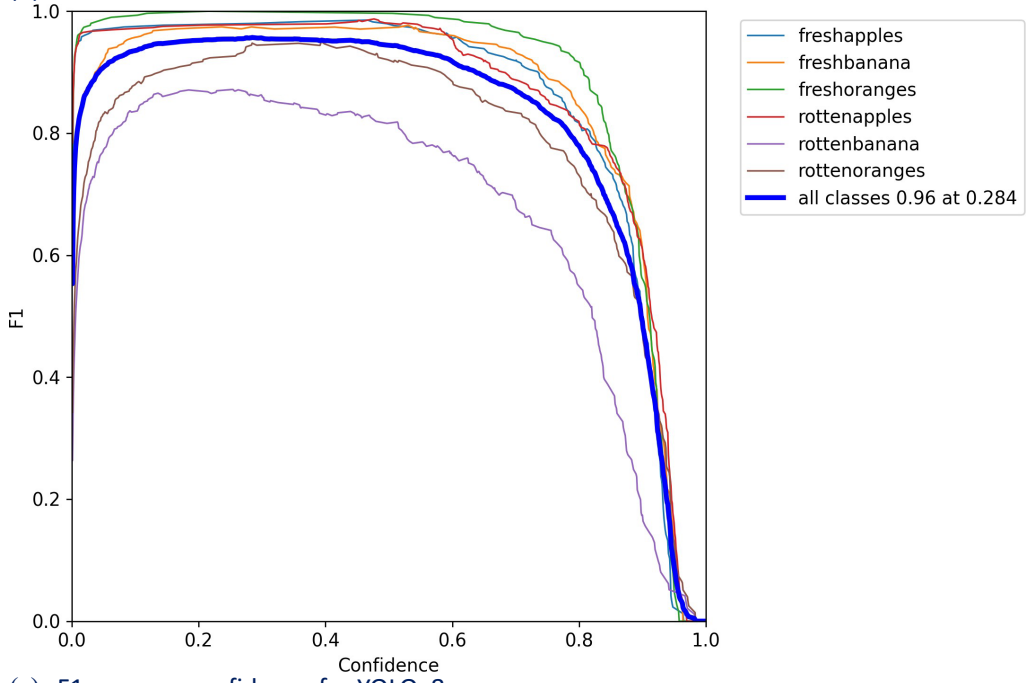
Fig. 6. PR curves of YOLO models



(a) F1 score vs confidence for YOLOv6



(b) F1 score vs confidence for YOLOv7



(c) F1 score vs confidence for YOLOv8

Fig.7. F1 score vs confidence of YOLO models

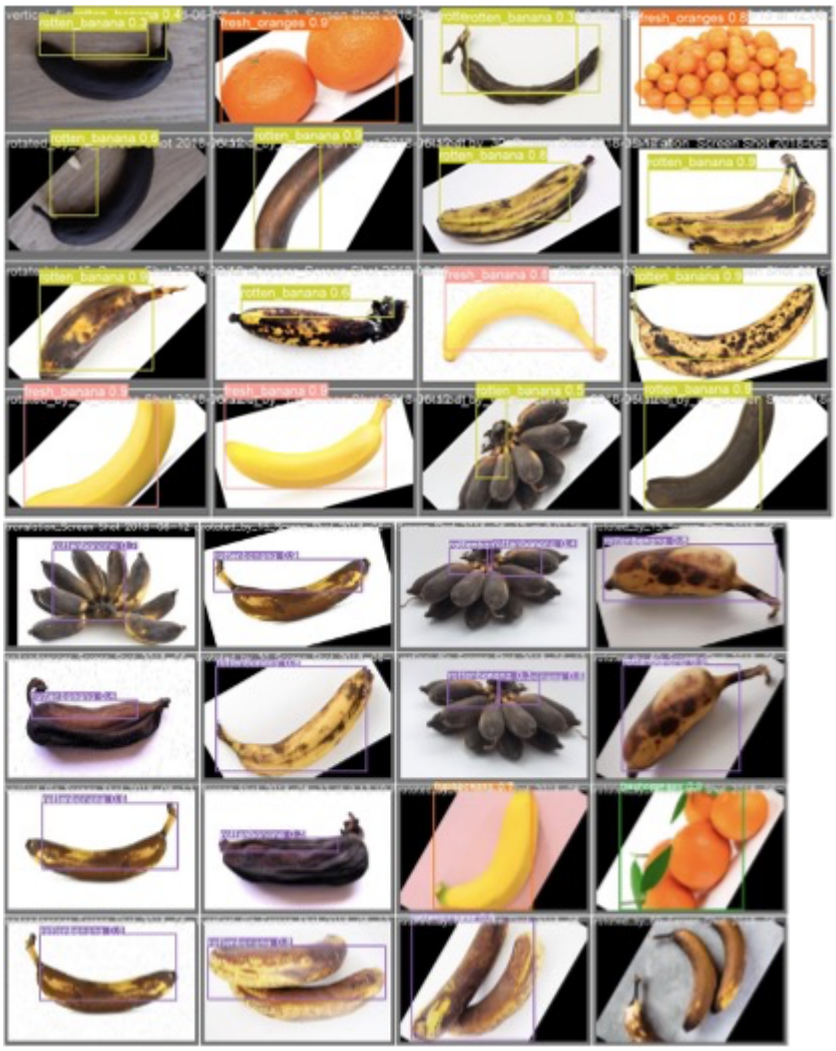


Fig. 7 Our classification results using YOLOv8 model

In Fig. 7, we show our prediction results generated by YOLOv8 model. By following the given images, the purple bounding box means rotten banana, green bounding boxes refer to fresh oranges, and orange bounding boxes indicate fresh banana. As we see from the test image, the test images also have a rotation around the center, this kind of argumentation methods can give us a better estimation of the classification in the real scene, because in our real world, it is impossible for us or for a machine to take a photo for a fruit from a constant angle, for example, we always take photos for apple from a side perspective, this will increase the workload of human.

To better compare the results, we will show the final prediction results from three different models. Because this is a classification task, we finally make use of precision and recall to estimate the average performance, as we discussed in the previous section, precision means how many objects the network can find with the correct label from the given images, and recall is the score to show the ratio of our correct detection. A good classifier should have a balance between precision and recall; but for this fresh fruit classification task, the results from different architectures are shown in Table 1.

In Table 1, YOLOv6 model has a better result compared with other different architectures, but actually, from the confusion matrix, YOLOv8 model has a better classification accuracy, a large error only occurs for rotten orange. So in the real scene, if we can only make use of a single network to detect fruits from the image dataset, we highly recommend consideringthe YOLOv8 for the first try.

MAJORITY VOTE

Majority vote is a traditional method used to improve the detection performance, because the YOLOv8 architecture is very easy to implement, we are planning to propose it as a base method to get a better result. There are two main reasons why we are use of majority vote as a method in this book chapter. The first reason is that the majority vote is a method that can be employed for "outside" of the model. The second reason is with a majority vote, we can easily determine how many results and which one we will use it to generate the final output, and the majority vote method also has the best adaptability, replacing the backbone (like YOLOv8) that will not influence the architecture, we still can take use of the same code to change the input that we get from the new backbone, this will also help us easily change our architecture in the future. Our results are shown in Table 3.

Table 1 The results of training the deep nets.

Models	Avg precision	Avg recall
YOLOv6	0.846	0.884
YOLOv7	0.852	0.879
YOLOv8	0.861	0.892

Table 2 The results for testing the deep nets.

Models	All classes IOU	All classes F1	The best threshold	accumulate error rate
YOLOv6	0.975	0.96	0.659	12%
YOLOv7	0.972	0.96	0.284	9%
YOLOv8	0.979	0.96	0.569	14%

Table 3. Comparisons of various models

	Precision	Recall
YOLOv6	0.956	0.943
YOLOv7	0.933	0.92
YOLOv8	0.957	0.941
Majority vote	0.961	0.957

ALATION STUDY

While we train the network models, we make use of the original fruit images, because the network with an augmentation dataset will increase the performance of our proposed model, there are three types of datasets: The first is the original dataset, which includes the images without any augmentation. The second dataset is that we add horizontal flip and vertical flip to the training dataset. The third dataset is that we add flips and blur to the training dataset, the result are shown in Table 4.

Table 4. Ablation experiments.

	Train on original dataset		Train on dataset with flip augmentation		Train on dataset with flip and blur	
Architecture	Precision	Recall	Precision	Recall	Precision	Recall
YOLOv6	0.956	0.943	0.955	0.94	0.967	0.96
YOLOv7	0.933	0.92	0.937	0.921	0.952	0.931
YOLOv8	0.957	0.941	0.96	0.934	0.97	0.964

From the results in Table 4, it is obvious that when we add flipping operations to the original dataset, the performance increases but only a little bit. Compared with only the flip augmentation dataset, we add the blurry images to the dataset which enhances the performance a lot, both precision and recall from different architectures ramp up at least one percent, this means compared with the flipped image, blurring images will provide more new information to the model training.

Ablation study

It is obvious that with the majority vote method, our network can perform better than any individual network. In general, the majority vote method is,

$$voting(w_1 * net_1, w_2 * net_2 \dots \dots w_n * net_n) \quad (1)$$

where w_1 to w_n is the weights for different architecture, the net_1 to net_n is the network detection result from different network architecture, the weights will always in range [0,1]. All the other numbers refer to the confidence of the network prediction we use, if a network has a high weight (high confidence), the result is more accurate than the other results. With various weights, we can make the majority vote result become more accurate and robust, we show different weight results if we combine YOLOv6, YOLOv7 and YOLOv8 together.

From Table 5, we easily see that the weights may not be the best strategy to generate the result, given the best performance of our proposed model, relatively large weights will have a better performance, but increasing the weights for a certain network too much will make the majority vote result get close to the single network result.

Table 5. Ablation results with various weights

Series	Weights for YOLOv6 model	Weights for YOLOv7 model	Weights for YOLOv8 model	Majority voting results	
				precision	recall
1	0.33	0.33	0.33	0.961	0.957
2	0.6	0.2	0.2	0.96	0.942
3	0.2	0.6	0.2	0.94	0.92
4	0.2	0.2	0.6	0.97	0.963
5	0.8	0.1	0.1	0.954	0.94
6	0.1	0.1	0.8	0.955	0.942

CONCOLUSION

In this book chapter, we summary our work and show our future research directions. For the dataset, we are use of the fruit freshness dataset from Kaggle which includes 6 different types of image samples. In this dataset, fruits are fresh apple, fresh banana, fresh orange, rotten apple, rotten banana, and rotten orange, After getting the dataset, we split them into training and test dataset. Following the normal preprocess steps, we split the dataset with a ratio 8:2, which means we will randomly choose 80 % data to train and the rest for the test. Pertaining to the methods for fruit classification, we took use of YOLOv6, YOLOv7, and YOLOv8 to generate the classification result.

Compared to the result from other architectures, during the training and testing, YOLOv8 will have the highest accuracy and the fastest training speed. YOLOv8 is always very easy after we got the result from all YOLO architectures, we make use of the majority voting method to ensemble their result and finally, we got a higher accuracy than the previous work for fresh fruit detection with the ensemble method.

In the future, we are planning to use YOLOv8 as a baseline of the architecture and combine with

other models, we believe this will be a great choice. With the development of the detection method created in recent years, a lot of new models, like diffusion model and attention models will always have a very good performance of object detection classification. In our next step, we can also add those architectures in our ensemble group and generate result, we believe such a method can help us get a higher accuracy even we detect with more types of fruits.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., & Ghemawat, S. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467.*

Abdulrahman, A., & Iqbal, K. (2014) Capturing human body dynamics using RNN based on persistent excitation data generator. *International Symposium on Computer-Based Medical Systems (CBMS),* (pp. 221-226).

Anderson, C., Burt, P., & Van Der Wal, G. (1985). Change detection and tracking using pyramid transform techniques. *Cambridge Symposium* (pp. 72-78) International Society for Optics and Photonics.

Baum, L., E., & Sell, G. (1968). Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, *27*(2), 211-227.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157-166.

Barnea, E., Mairon, R., & Ben-Shahar, O. (2016). Colour-agnostic shape-based 3D fruit detection for crop harvesting robots. *Biosystems Engineering*, 146, 57-70.

Chatzis, S. P., & Kosmopoulos, D. I. (2011). A variational Bayesian methodology for hidden Markov models utilizing Student's-*t* mixtures. *Pattern Recognition*, *44*(2), 295-306.

Chen, Y. N., Han, C. C., Wang, C. T., Jeng, B. S., & Fan, K. C. (2006) The application of a convolution neural network on face and license plate detection. *International Conference on Pattern Recognition,* (Vol. 3, pp. 552-555).

Collins, R. T., Lipton, A. J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., & Wixson, L. (2000) A system for video surveillance and monitoring. *Research Report. Carnegie Mellon University, USA*.

Eickeler, S., & Muller, S. (1999). Content-based video indexing of TV broadcast news using hidden Markov models. *IEEE International Conference on Acoustics, Speech, and Signal Processing.*(Vol. 6, pp. 2997-3000). IEEE

Fu, R., Zhang, Z. and Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. *Youth Academic Annual Conference of Chinese Association of Automation (YAC).*

Fu, Y., Nguyen, M., Yan, W. (2022) Grading methods for fruit freshness based on deep learning. Springer Nature Computer Science.

Fu, Y. (2020) Fruit Freshness Grading Using Deep Learning. Master's Thesis. Auckland University of Technology, New Zealand.

Fang, W., Lin, W., Ren, P. (2019) Tinier-YOLO: A real-time object detection method for constrained environments. *IEEE Access*, 8: 1935 - 1944.

Fang, Y., et al. (2021) You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34: 26183 – 26197.

Galvez, R., Bandala, A., Dadios, E., et al. (2018) Object detection using convolutional neural networks. *IEEE TENCON*, 2023-2027.

Gers, F. A., & Schmidhuber, J. (2000). Recurrent nets that time and count. Neural Networks, *IEEE – INNS* (Vol. 3, pp. 189-194).

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation, 12*(10), 2451 – 2471.

Haritaoglu, I., Harwood, D., & Davis, L. S. (2000). W/sup 4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22* (8), 809 – 830.

Han, X., Gao, Y., Lu, Z., Zhang, Z., & Niu, D. (2015). Research on moving object detection algorithm based on improved three frame difference method and optical flow. *International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)* (pp. 580-584).

Heikkila, M., & Pietikainen, M. (2006). A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(4), 657-662.

Joseph, R., Divvala, S., Girshick, R., Farhadi, A. (2021) You only look once: Unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779 – 788.

Khan, R., Debnath, R. (2015) Multi class fruit classification using efficient object detection and recognition techniques. *International Journal of Image, Graphics and Signal Processing.*

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale

video classification with convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1725 - 1732).

Katagiri, S., & Lee, C. H. (1993). A new hybrid algorithm for speech recognition based on HMM segmentation and learning vector quantization. *IEEE Transactions on Speech and Audio Processing, 1* (4), 421 – 430.

LeCun, Y. (1989). Generalization and network design strategies. *Connectionism in Perspective,* 143 – 155.

LeCun, Y., & Ranzato, M. (2013). Deep learning tutorial. *Tutorials in International Conference on Machine Learning (ICML'13).*

Liu, Y., Sun, P., Wergeles, N., et al. (2021) A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 172: 114602.

Liu, Z., Yan, W., Yang, B. (2018) Image denoising based on a CNN model. International Conference on Control, Automation and Robotics.

Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks, 16* (5), 555 - 559.

Maryam, K., & Reza, K. M. (2012). An analytical framework for event mining in video data. *Artificial Intelligence Review, 41* (3), pp. 401 – 413.

Mikolov, T., Kombrink, S., Burget, L., Černocký, J., & Khudanpur, S. (2011). Extensions of recurrent neural network language model. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* (pp. 5528 - 5531).

Narendra, K. S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks, 1* (1), 4 – 27.

Noorit, N., & Suvonvorn, N. (2014). Human activity recognition from basic actions using finite state machine. *International Conference on Advanced Data and Information Engineering (DaEng - 2013)* (pp. 379 - 386). Springer.

Pan, C., Yan, W. (2018) A learning-based positive feedback in salient object detection. International Conference on Image and Vision Computing New Zealand.

Pan, C., Yan, W. (2020) Object detection based on saturation of visual perception. Multimedia Tools and Applications, 79 (27-28), 19925-19944.

Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient object detection based on visual perceptual saturation and two-stream hybrid networks. IEEE Transactions on Image Processing.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *ICML (3), 28,* 1310 – 1318.

Petrushin, V. A. (2005). Mining rare and frequent events in multi-camera surveillance video using self-organizing maps. *ACM SIGKDD international Conference on Knowledge discovery in Data Mining,* pp. 794 – 800.

Popoola, O. P., & Wang, K. (2012). Video-based abnormal human behavior recognition—A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42* (6), 865 – 878.

Parico, A., Ahamed, T. (2021) Real time pear fruit detection and counting using YOLOv4 models and deep SORT. *Sensors*, 21(14): 4803.

Xia, Y., Nguyen, M., Yan, W. (2022) A real-time Kiwifruit detection based on improved YOLOv7. International Conference on Image and Vision Computing New Zealand (IVCNZ)

Xia, Y., Nguyen, M., Yan, W. (2023) Kiwifruit counting using KiwiDetector and KiwiTracker. IntelliSys conference.

Xiao, B., Nguyen, M., Yan, W. (2023) Apple ripeness identification from digital images using transformers. Multimedia Tools and Applications, Springer Science and Business Media LLC.

Xiao, B., Nguyen, M., Yan, W. (2023) Fruit ripeness identification using transformers. Applied Intelligence, Springer Science and Business Media LLC.

Xiao, B., Nguyen, M., Yan, W. (2021) Apple ripeness identification using deep learning. International Symposium on Geometry and Vision.

Yan, W. (2019) Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer London.

Yan, W. (2021) Computational Methods for Deep Learning: Theoretic, Practice and Applications. Springer London.

Yu, Y., Zhang, K., Yang, L., & Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on Mask R-CNN. *Computers and Electronics in Agriculture*, 163, 104846.

Zhang, Y., et al. (2022) Complete and accurate holly fruits counting using YOLOX object detection. *Computers and Electronics in Agriculture*, 198: 107062.

Zhao, K. (2021) Fruit Detection Using CenterNet. Master's Thesis, Auckland University of Technology, New Zealand.

Zhao, K., Yan, W. (2021) Fruit detection from digital images using CenterNet. International Symposium on Geometry and Vision.

Zou, Z., et al. (2023) Object detection in 20 years: A survey. *Proceedings of the IEEE.*