Fruit Detection

Le Minh Khoi, Huynh Bao Thanh

Vietnamese-German University

Machine Learning Final Project

E-mail: 17153@student.vgu.edu.vn

08 December, 2022

1 Introduction

1.1 Problem Statement

Nowadays, machine learning has taken an important position in academia and industry because of its noticeable applications. In depth, the usage of machine learning attaches to the usage of enormous amounts of data processing by some technical methods to take advantage of the information. Then we are capable of analyzing and extracting beneficial information in real life. The more using historical data, the more efficient information we gain. This technology goes along with AI creating ideas for researching and improvement. For example, smart building, smart houses, smart transport, etc were applied successfully in many countries especially the developed and technologically advanced countries.

The above discussions, there is one big question: how can a developing country with the agricultural strengths as Vietnam utilize this technology in producing, farming, storing and even exporting? With this motivation our group came up with an idea to bring computer vision closer to the farmer. Specifically in this project, we built up an experiment to detect three different fruits: apple, watermelon and cantaloupe using different YOLOv5 models and make comparison between them.

1.2 Application

Before operating the experiment, there are some fields that fruit detection can be applied:

In industry:

-Classify fruits

-Recognize rotten fruits for eliminating

In storagement:

-Manage the amount of fruit

-Measurement the fruits ripeness by information images

2 Methodology

In this section we will illustrate our approach in choosing YOLOv5 version , building dataset and operating pipeline

2.1 Analyze and Design the System

Since the time releases, there have been a lot of YOLOv5 models developed. However, we desire a general view about the application so we decided to train our dataset with different YOLOv5 models and make comparisons between them.

We focus more about validation and training speed, therefore YOLOv5n6, YOLOv5m6, YOLOv5x6 were chosen for our experiment.

Model	size (pixels)	mAP^{val} 50-95	mAP^{val} 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	**45**	**6.3**	**0.6**	**1.9**	**4.5**
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6 + TTA	1280 1536	55.0 **55.8**	72.7 **72.7**	3136 -	26.2 -	19.4 -	140.7 -	209.8 -

Figure 1: YOLOv5 versions

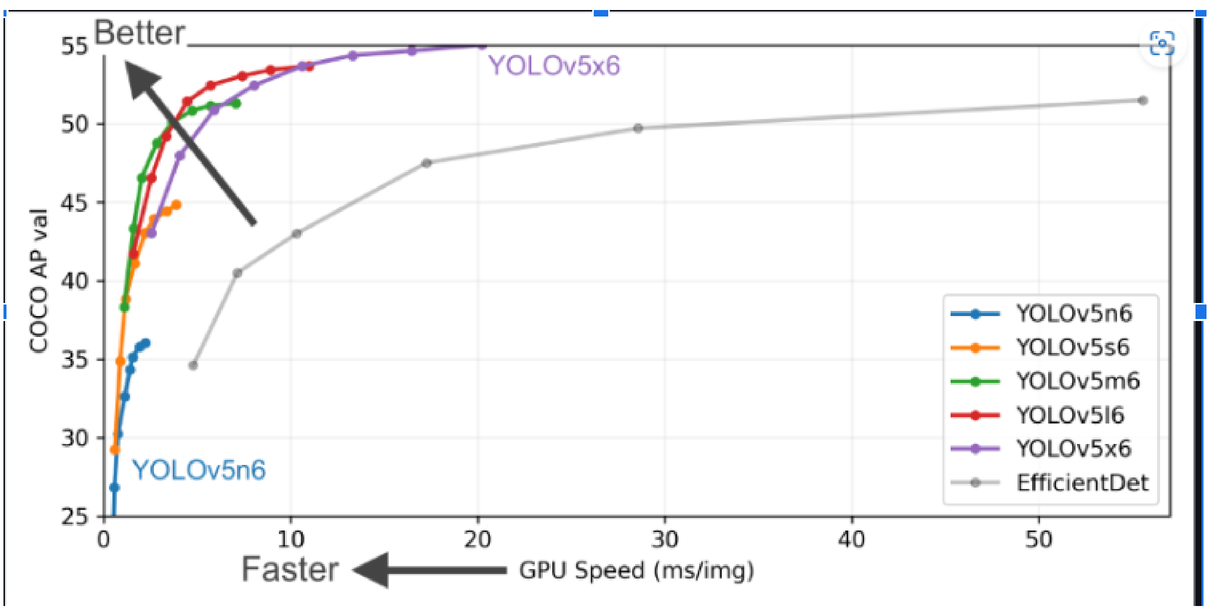

Figure 2: YOLOv5 versions

2.2 Data Collection and Processing

We divided our dataset into two parts, the first part is a dataset1 collected on the internet with 90 samples and the second part is a dataset2 collected manually by taking photos with 600 samples. The dataset1 and dataset2 in comparison: dataset1 is less samples than dataset2; every dataset2's samples are in uniform because they were captured by the camera frame; dataset2 has better image information.

2.2.1 Tools and Technology Used

For Labeling, we used LabelImg.py-a graphical image annotation tool run by anaconda. We found this is a useful tool to make a dataset in different types of models, the tool and interface are easy to approach and manipulate.

2.2.2 Image Augmentation

To gain the better training result, we should increase the number of data. The image augmentation method we choose were:

-Blur

-Rotating

-Flipping

-Adding noise

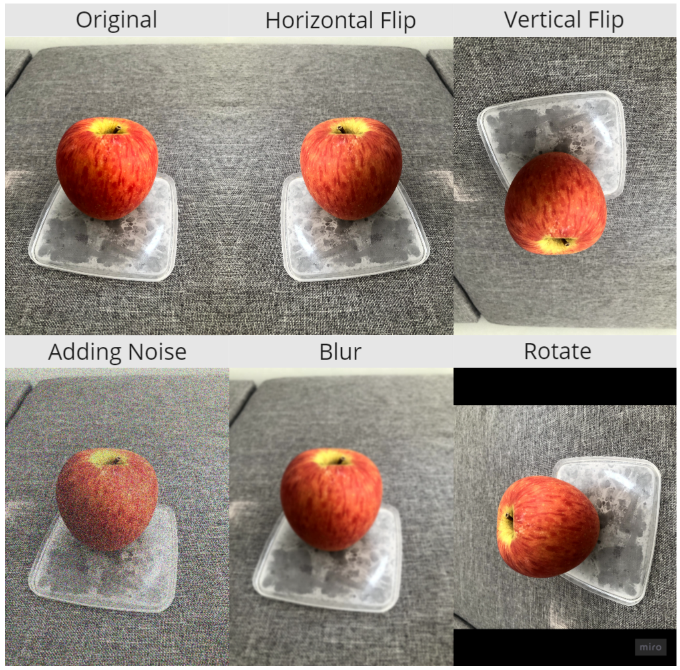

Figure 3: Image Augmentation Sample

2.3 System Construction

All of our process follow the pipeline below:

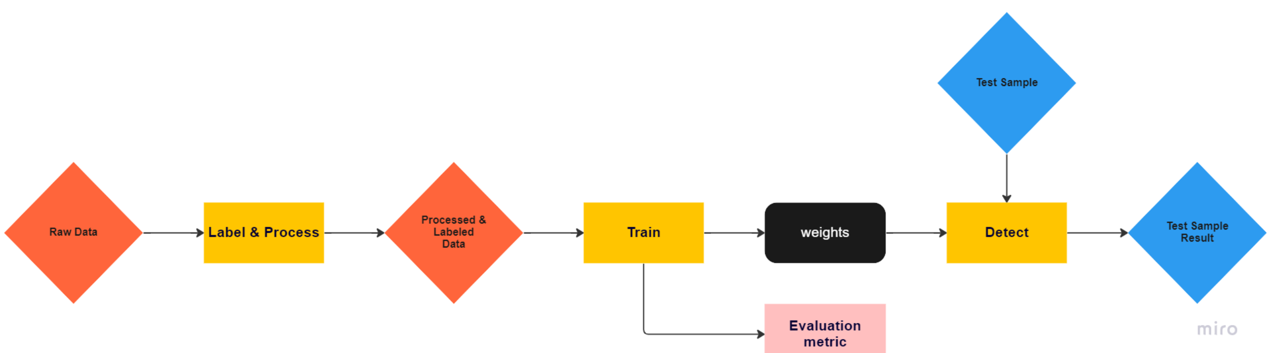

Figure 4: Pipeline

3 Experiments

3.1 Evaluation Metric

After training the dataset by each model, there are several tools called evaluation metrics to observe the performance of models and comparison between models. In this project, we used precision, recall, confusion matrix, F1_curve, P_curve, PR_curve, R_curve.

-**Recall metric**: It is how many actually positive predicted in the observations of positive.

-**Precision metric**: It is how many predicted in the observations of actually positive.

-**Confusion matrix**: The most intuitive and descriptive metrics used to find the accuracy and correctness. it is mainly used for the output that contains two or more types of classes.

-**F1_curve**: It determines the optimum confidence that balances the precision and recall, it spans on a domain from 0 to 1.

-**P_curve**: It determines the precision of each confidence.

-**PR_curve**: It summarizes the trade-off between the true positive rate and false positive rate for a predictive model.

-**R_curve**: It determines the precision of each recall.

After training dataset1 and dataset2 in 20 epochs with three YOLOv5 models. We collected the results below:

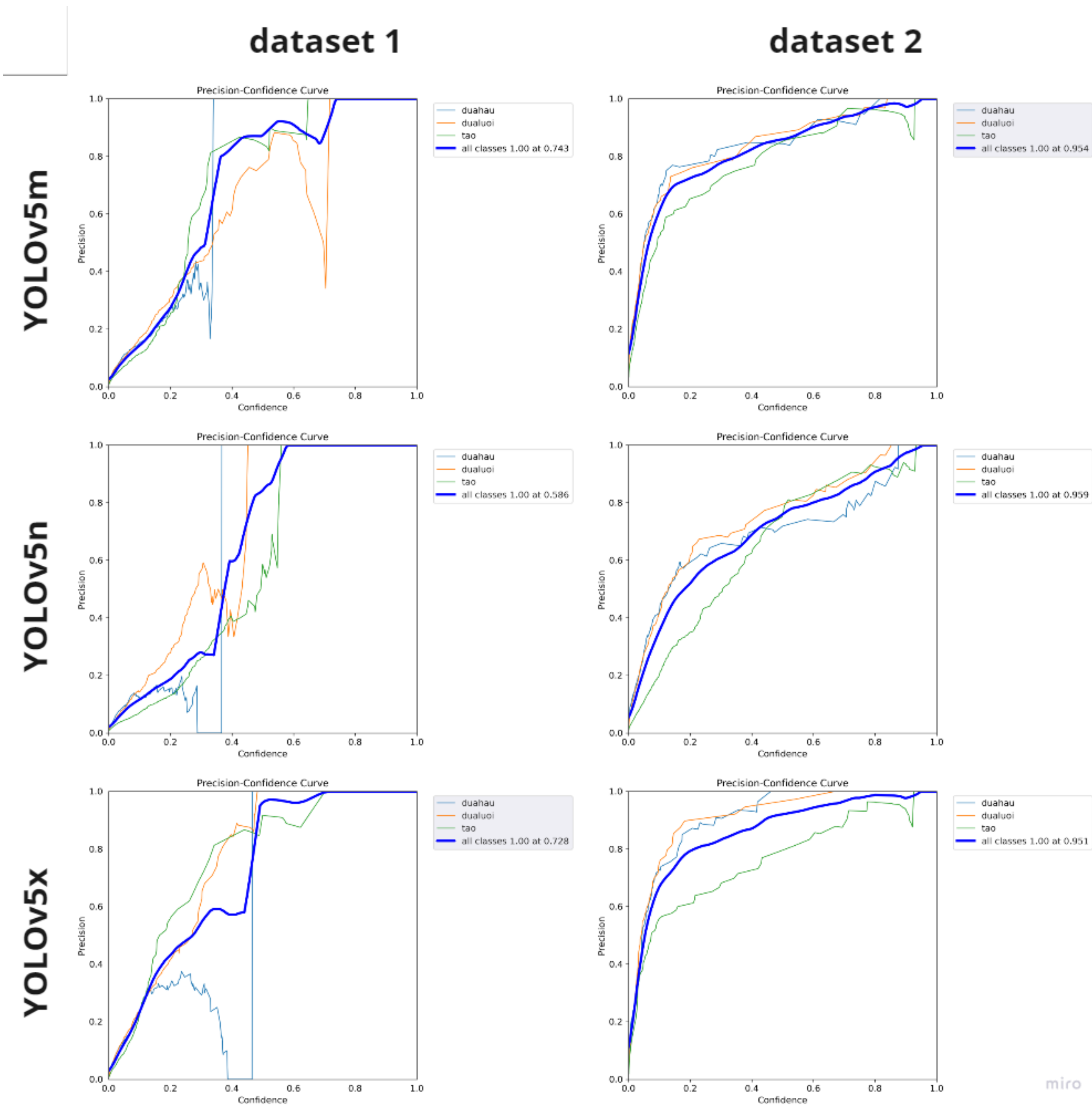

Figure 5: P_curve

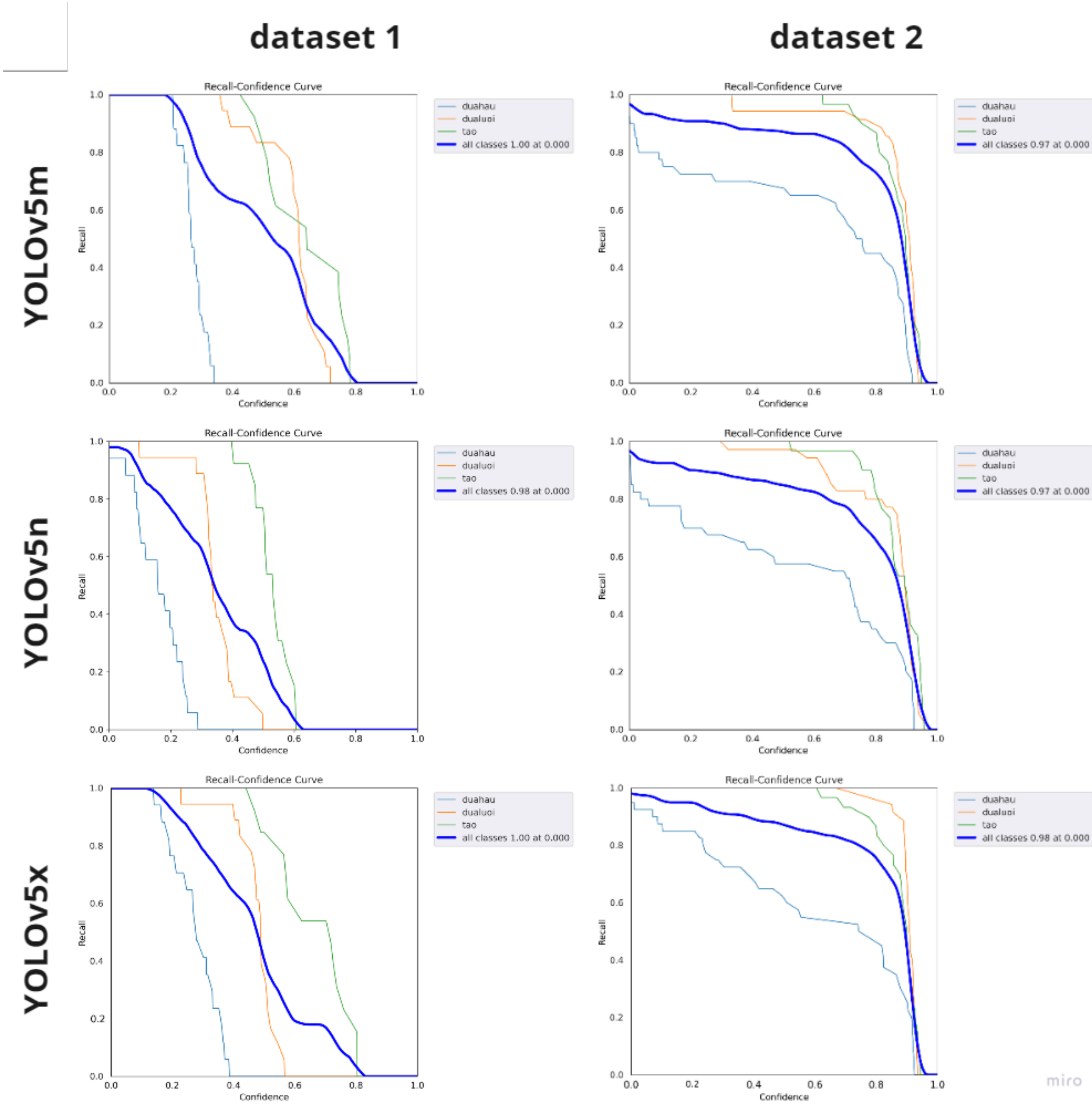

Figure 6: R_curve

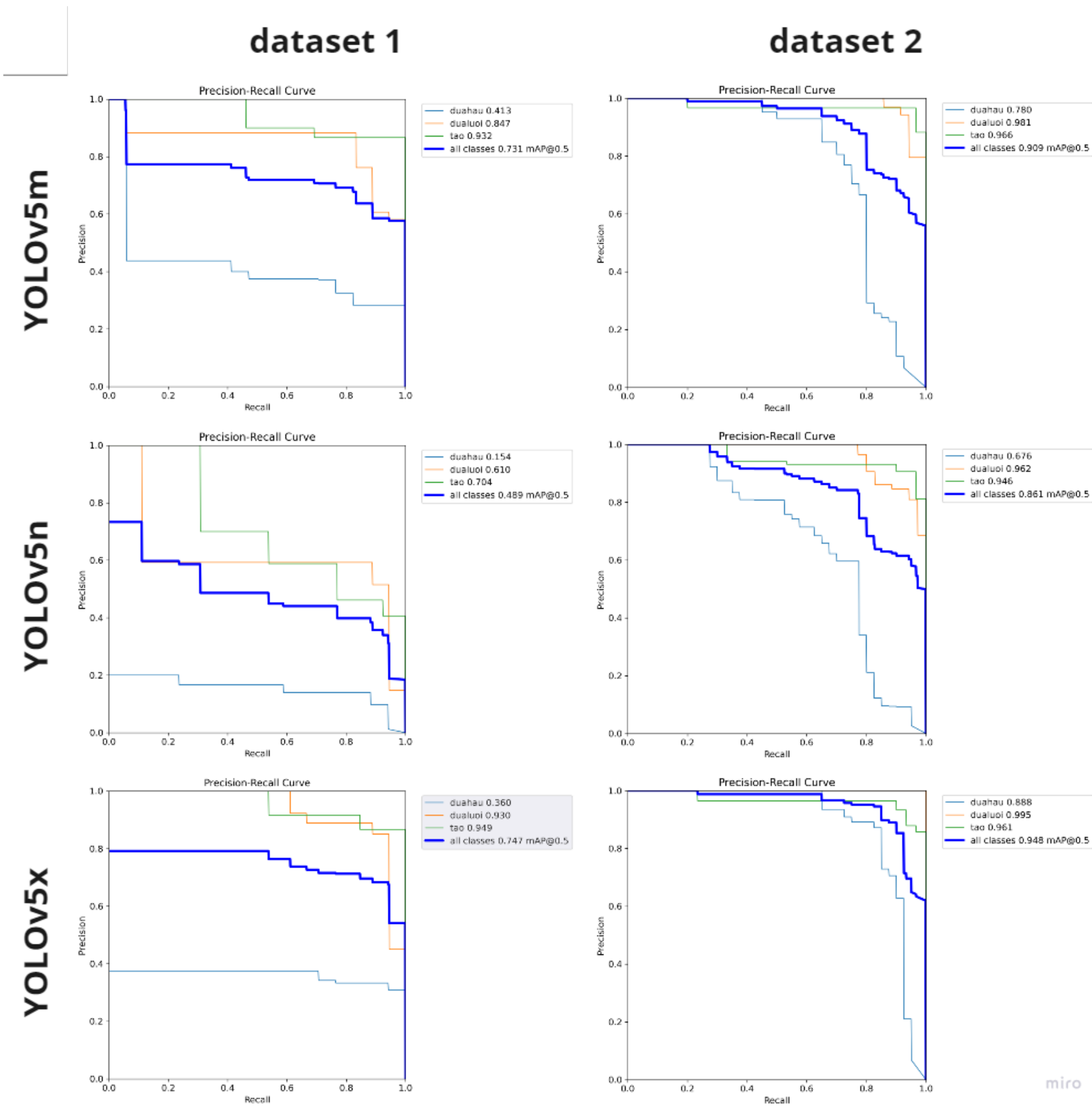

Figure 7: PR_curve

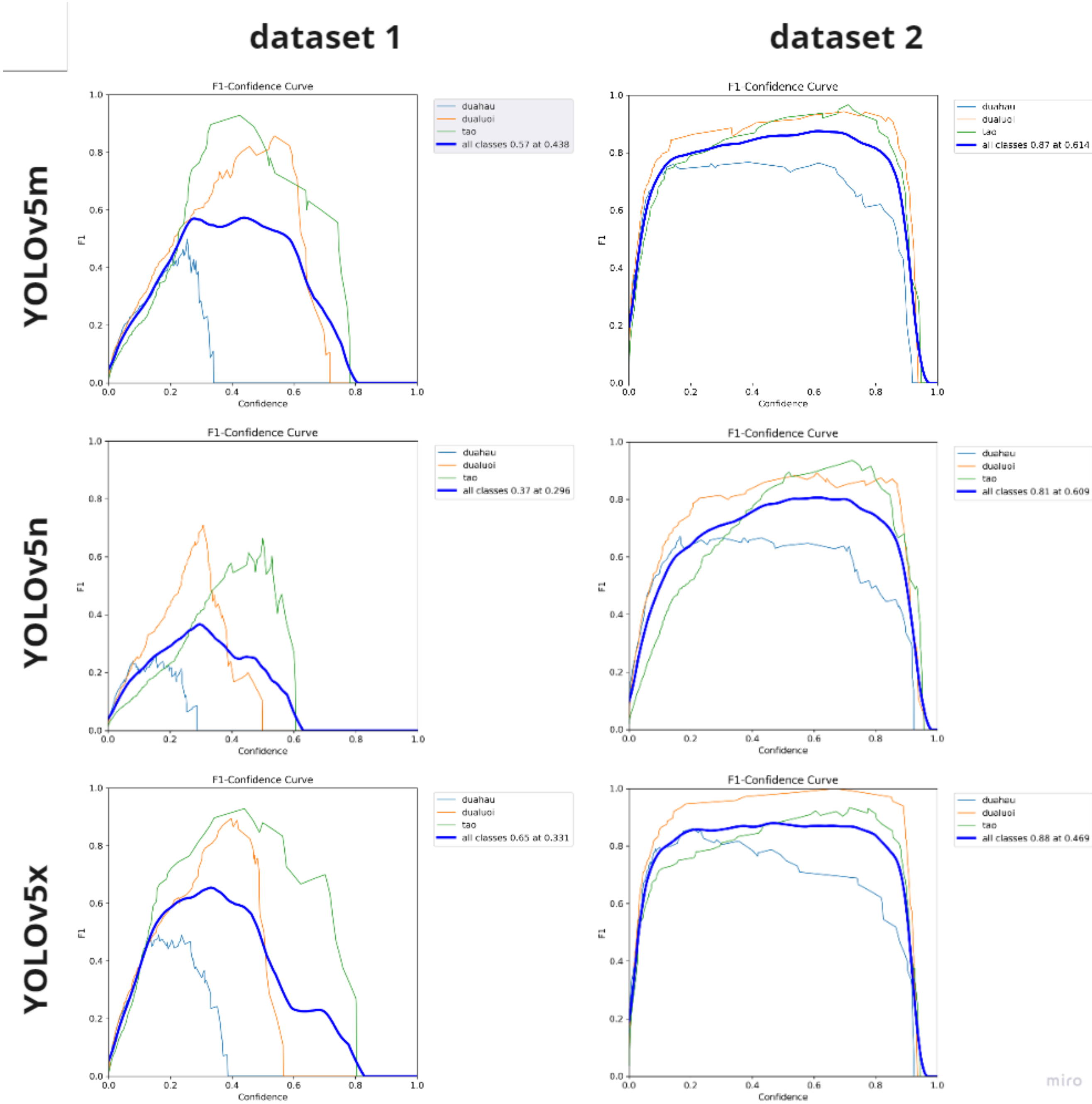

Figure 8: F1_curve

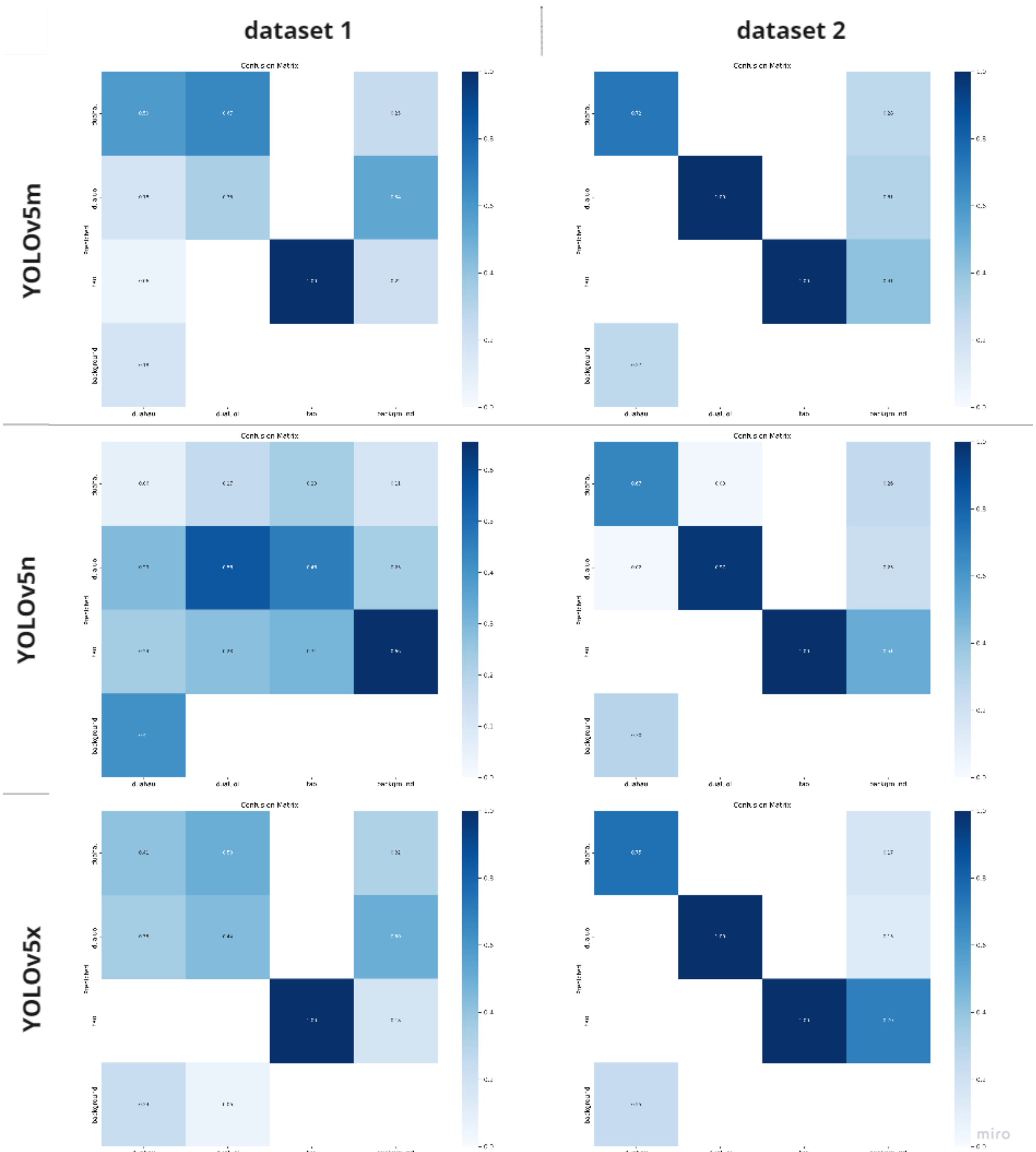

Figure 9: Confusion Matric

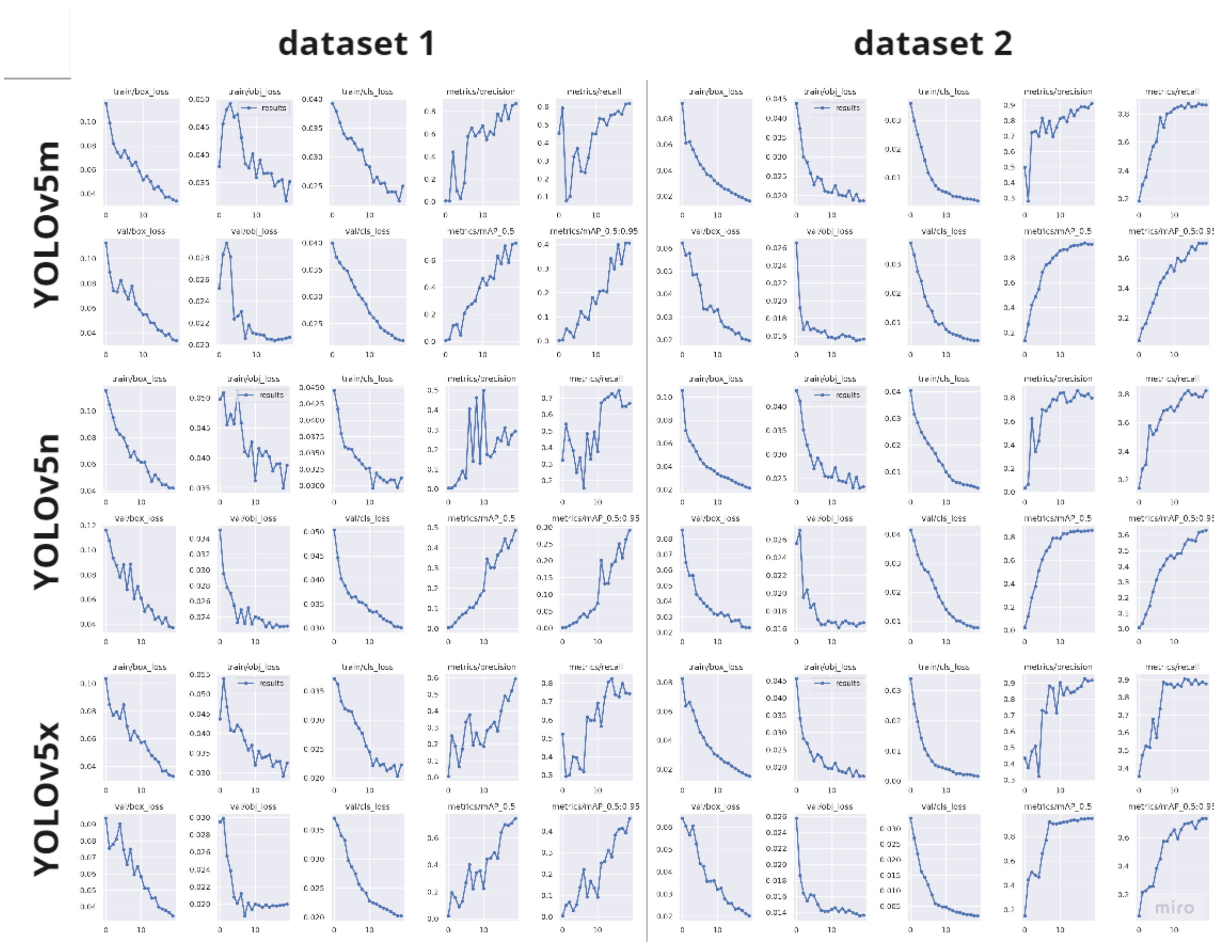

Figure 10: Result

Illustration

-Confusion Matrix: The YOLOv5n was still affected considerably by other objects and background, while two other models showed better distinguishing.

-PR_curve, P_curve, R_curve of YOLOv5n all showed worse than the others.

Conclusion: According to statistical data above, we are able to predict that YOLOv5m and YOLOv5x will produce good prediction than YOLOv5n

3.2 Benchmark data

We improved the performance of detection model by adjusting two factor:

-Data Capacity: At the begin, we tried to train our data on CPU, quickly, the CPU's memory was full immediately. Then we switched to virtual GPU on Google Colab, it provided better performance: it was redundant to take over our dataset, the only considerable thing was that it would turn off if we left the computer, so, we created a python programming for mouse clicking to avoid the disconnecting.

-Training Speed: We experienced data training on both CPU(Intel core i7 10th Gen) and virtual GPU on Google Colab that the virtual GPU was faster than CPU. On the other hand, we also tuned the batch: 4 batch with 20 epochs; 16 batch with 80 epochs. There bigger batch size, the faster training.

3.3 Results

According to our detection, there are some noticeable features:

Firstly, dataset1 and dataset2 were trained by YOLOv5m. By observation the detection result below, the training on dataset2 performed better.

Secondly, dataset2 were trained by YOLOv5m, YOLOv5n and YOLOv5x. While YOLOv5m and YOLOv5 provided good detection on different labels, they performed better than YOLOv5n.

The detecting results affirm that the previous prediction is correct.

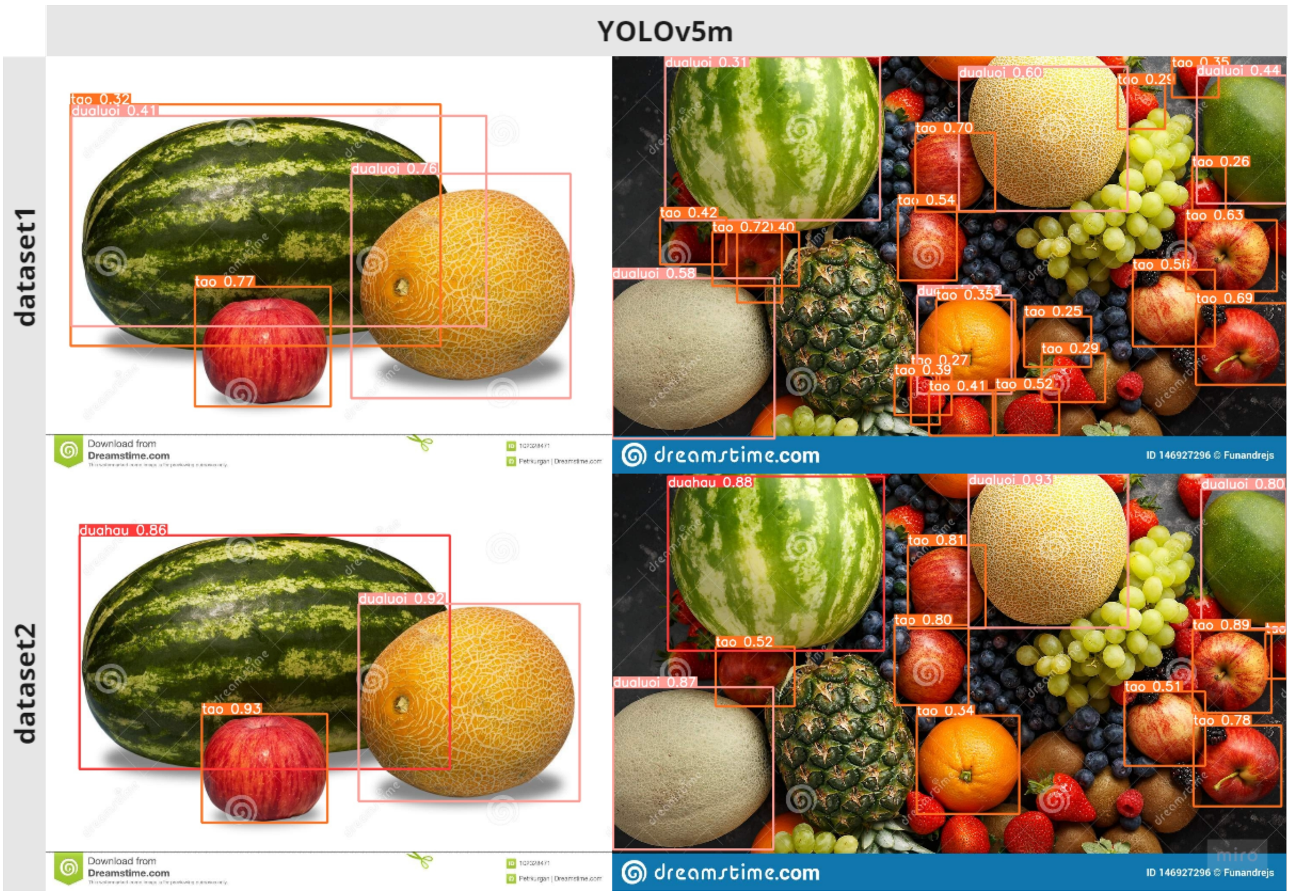

Figure 11: Compare Detection between dataset1 and dataset2

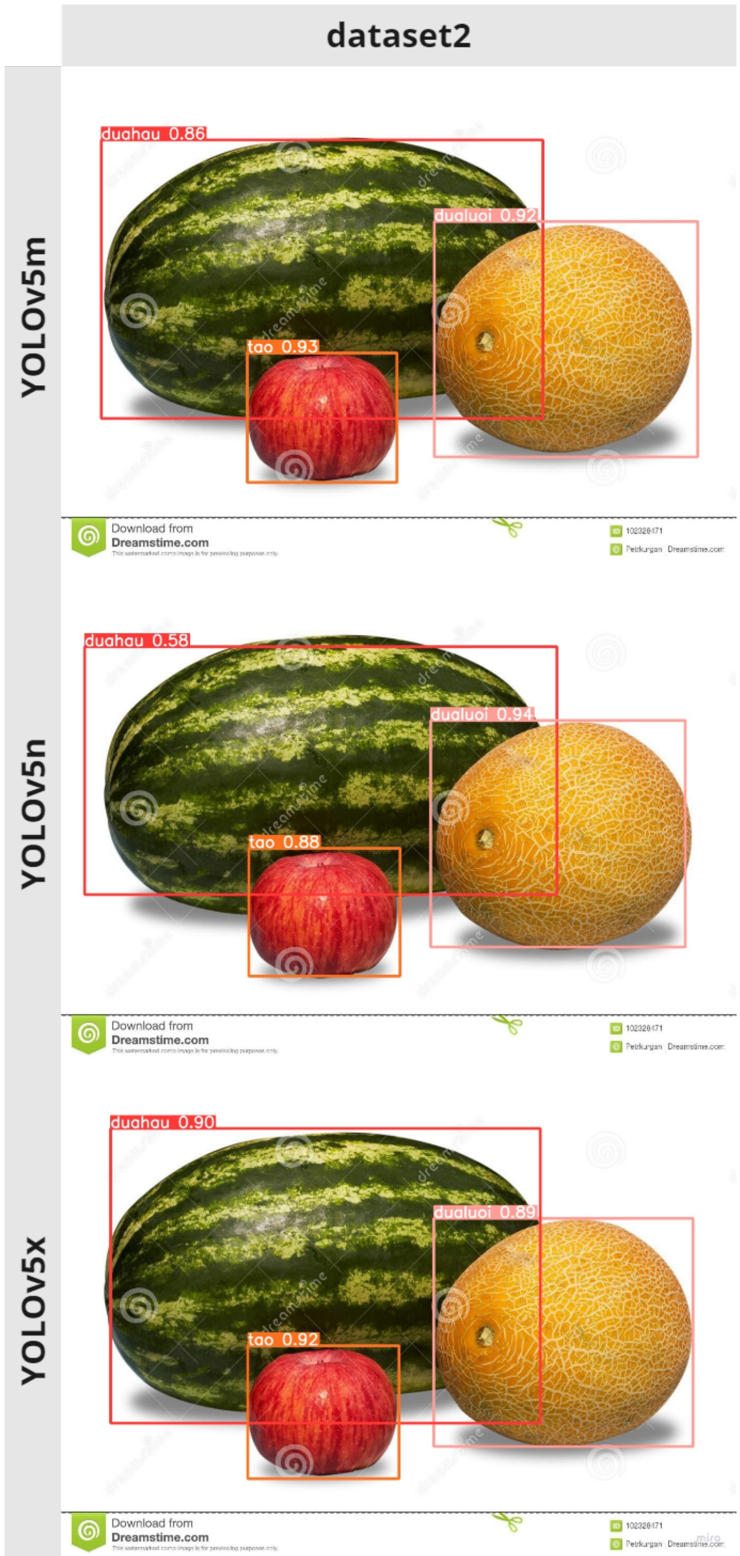

Figure 12: Compare Performances of Three Models

-Weakness and Strength: Several pros of our machine learning model are detecting with high precision, able to detect hidden objects, distinguishing similar fruits (for example apple and tomato) and less confusing the objects standing together. However, the cons are including small sample size, low epoch, it is hard to detect all of the objects if the frame has too much. There is also a trade off between detecting all of the objects that the model supposes it potential and detecting only the objects with high precision.

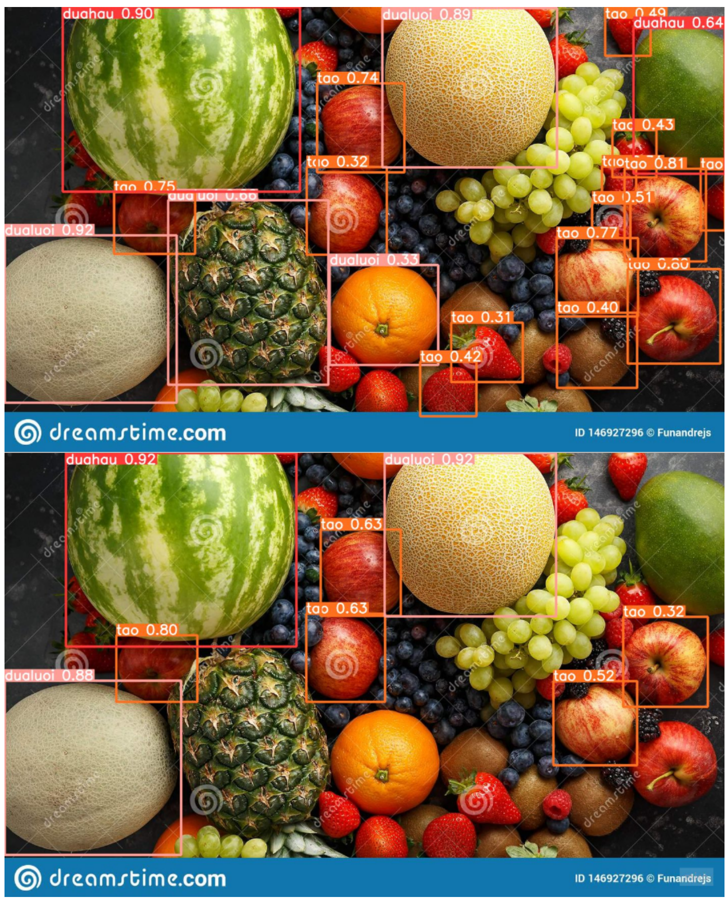

Figure 13: Trade off Features

4 Conclusion and Future Work

In this project, we walked through the experiments to observe the performances of different YOLOv5 model versions and detect successfully three different fruits: apple, watermelon and cantaloupe. We tried to take over a small application which has high potential in utilizing A.I to enhance the agriculture field. We carried one performance of models in terms of evaluation metrics and one for Benchmark data. For further future work, this small project can be expanded in some way for us to improve. For example: put fruit detection in robotics to classify fruit automatically in industry and farm; using fruit detection to determine the ripeness of fruit and predict the using date; using fruit detection to count and recognize the fruit for autonomous preservation planning.

References

https://github.com/ultralytics/yolov5