# Investigation of You Only Look Once Networks for Vision-based Small Object Detection

Li YANG[1]
Department of Electronic Information Engineering, Leshan Vocational and Technical College
Leshan 614099, Sichuan, China

***Abstract*—Small object detection is a challenging issue in computer vision-based algorithms. Although various methods have been investigated for common objects including person, car and others, small object are not addressed in this issue. Therefore, it is necessary to conduct more researches on them. This paper is focused on small object detection especially jewellery as current object detection methods suffer from low accuracy in this domain. This paper introduces a new dataset whose images were taken by a web camera from a jewellery store and data augmentation procedure. It comprises three classes, namely, ring, earrings, and pendant. In view of the small target of jewellery and the real-time detection, this study adopted the You Only Look Once (Yolo) algorithms. Different Yolo based model including eight versions are implemented and train them using our dataset to address most effective one. Evaluation criteria, including accuracy, F1 score, recall, and mAP, are used to evaluate the performance of the various YOLOv5, YOLOv6, and YOLOv7 versions. According to the experimental findings, utilizing YOLOv6 is significantly superior to YOLOv7 and marginally superior to YOLOv5.**

***Keywords—YOLOv7; YOLOv6; YOLOv5; computer vision; jewellery detection; small object detection; real-time detection***

## I. Introduction

Target detection techniques based on deep learning have recently received much attention because of their strong generalizability, which has coincided with the growth of deep-learning theory and improved computer performance [1]. Convolutional neural networks dominate deep learning (CNN). Using the input picture as training data, the convolution network may successfully learn the key characteristics of the recognised item. Repeated training steadily boosts the training model's performance to provide accurate target detection outcomes [2].

Whether or not candidate areas are formed, the current popular object detection methods may be classified into two-stage algorithms and one-stage detection techniques. The RCNN [3], SPP (Space Pyramid Pooling)-Net [4], Fast-RCNN [5], Faster-RCNN [6], Mask-RCNN [7], and other algorithms are examples of the former. The latter primarily consists of the YOLO algorithm (YOLOv1 [8], YOLOv2 [9], YOLOv3 [10], YOLOv4 [11], YOLOv5 [12], etc.) and the SSD [13] algorithm. While two-stage algorithms may achieve high accuracy, the lengthy detection time makes it challenging to meet the real-time requirement in common object identification applications. The one-step detection method has taken centre stage in the study of object detection due to its benefits of high precision and quick speed. Early object identification systems, such as YOLOv1, YOLOv2, etc., often had a network topology that consisted of a fully-connected layer on top of many convolution layers. The model's capacity to recognize several scales was significantly constrained by only calculating the feature map of a set size. In this case, the algorithm's detection accuracy for tiny objects could be more optimal [14].

Tiny/small object recognition is a difficult topic in computer vision, and several solutions have been put out to deal with it. Tiny/small object identification techniques have a long history that dates back to the early 2000s, a time when classic feature-based techniques were widely used. Later, with the development of deep learning, researchers started looking at object detection techniques based on deep neural networks. Faster R-CNN, YOLO, SSD, and RetinaNet, among other well-known object detection frameworks, were first developed for identifying large objects, but they have now been upgraded to handle small/tiny objects. With techniques like Yolo, EfficientDet, and Sparse R-CNN reaching cutting-edge performance on tiny/small object identification benchmarks, there has been tremendous research advancement in this field recently. By altering network topologies and incorporating and enhancing additional datasets, researchers attempt to enhance the outcomes for tiny object identification. Another clear answer to this problem is to improve the input image resolution, although it increases processing time [15].

In this paper, we introduce a new dataset whose images were taken by a web camera. The main objective of this study is to address an effective method to deal with small object detection challenge with generated dataset. In order to generate the dataset, data augmentation and YOLO are used to detect jewellery object as one of popular small objects. The reason to use Yolo based object detection in this study, because of their high efficiency in terms of accuracy rate and low computation complexity in object detection. In view of the small target of jewellery and the real-time detection, we chose YOLOv7 [16], YOLOv6 [17] and YOLOv5 models and compare their results.

In summary, this paper has the following contributions:

*1)* Generating a new dataset for tiny objects: We gathered about 2,500 photos from a webcam which includes three categories of jewellery (earrings, ring and pendant) and then augmented them to about 6,500 images.

*2)* Generating new Yolo-based model on different versions of YOLOv7, YOLOv6 and YOLOv5 algorithms.

*3)* Conducting extensive evaluations and analysis for the generated models and presented comprehensive performance comparison on the methods.

The rest of this paper consists as follows; Section II presents the background of the study. Section III discusses the research methodology. The results and analysis are presented in Section IV. Finally, his paper concludes in Section V.

## II. Background

In 2015, researcher Joseph Redmon and colleagues introduced YOLO algorithm. The YOLO series outperforms preceding versions in terms of speed and the ability to find tiny things [18]. It just takes one iteration of the algorithm to spread over a picture for the YOLO to detect things in real time [19]. In this study, various YOLO model iterations were used for both training and testing.

### *A. YOLOv5*

A number of object recognition architectures that have already been trained using the MS COCO dataset are available in YOLOv5, which was released in 2020. Because of its quick speed and great accuracy, it is one of the most well-known detection algorithms. The photos are divided into a grid system by YOLOv5, and each grid cell is in charge of identifying items inside its own area. When several objects are present, this method offers a particular benefit.

The YOLOv5 is a deep learning-based platform available in five distinct iterations, ranging in size from the tiny YOLOv5 nano version, designed for mobile and embedded devices, to the gigantic YOLOv5x large version [18, 20, 22].

The author did not publish a detailed paper, but only launched a repository on Github and updated improvements there. Fig. 1 illustrates the backbone, neck, and head of the YOLOv5 architecture, which may be understood by analyzing its structural code [18, 21, 23]: Cross Stage Partial Networks (CSP) and the focal structure make up the backbone. The focus structure down samples the input data dimension while the original data is kept. The model's capacity for learning is enhanced, and its memory use is decreased, thanks to the CSP Network's ability to extract important information.

The neck part combines the acquired characteristics and transmits them to the prediction layer using Feature Pyramid Networks (FPN) and the Path Aggregation Network (PAN). The FPN up samples the high-level feature data via top-to-bottom communication and prediction fusion. The underlying pyramid, PAN, communicates important positioning properties top-to-bottom, aiding in the distinguishing of similar objects of various scales and sizes.

The final output vectors, consisting of bounding boxes, class probabilities, and object scores, are provided by the output layer's head by applying anchor boxes to the features. Including the focus and CSP layers is the primary change in YOLOv5. The focusing layer decreases layers, parameters, FLOPS, and CUDA memory to increase forward and reverse speeds. The backbone layer's CSP layer tries to extract specific data and carry out more extensive activities. In YOLOv5, the meshing ideas from the original YOLO algorithm have been retained [18].

### *B. YOLOv6*

The single-stage object detection framework for industrial applications, MT-YOLOv6 (created by the Meituan firm, hence the prefix "MT"), has a hardware-friendly, effective design and excellent performance [24].

Numerous upgrades to the Backbone, Neck, and Head blocks as well as training methods are included in YOLOv6. For instance, utilizing Rep-PAN and EfficientRep structures, respectively, the Neck and Backbone in YOLOv6 have been redesigned in accordance with the notion of a hardware-aware neural network. Strong representational capabilities are combined with hardware computational capability, such as GPU, in the EfficientRep Backbone. Rep-Pan Neck outperforms PANet and SPP in terms of accuracy and speed.

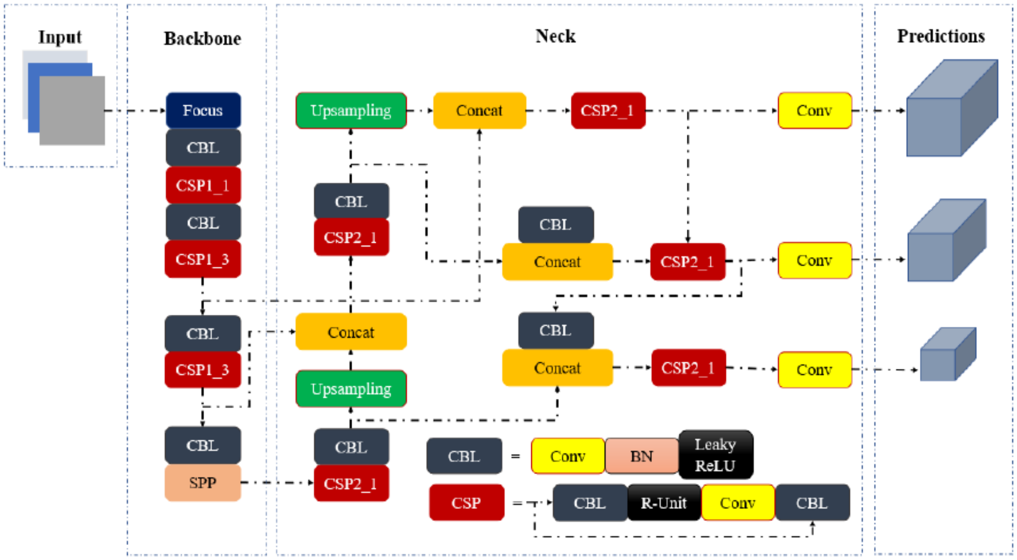


Fig. 1. Network structure of YOLOv5 [18].

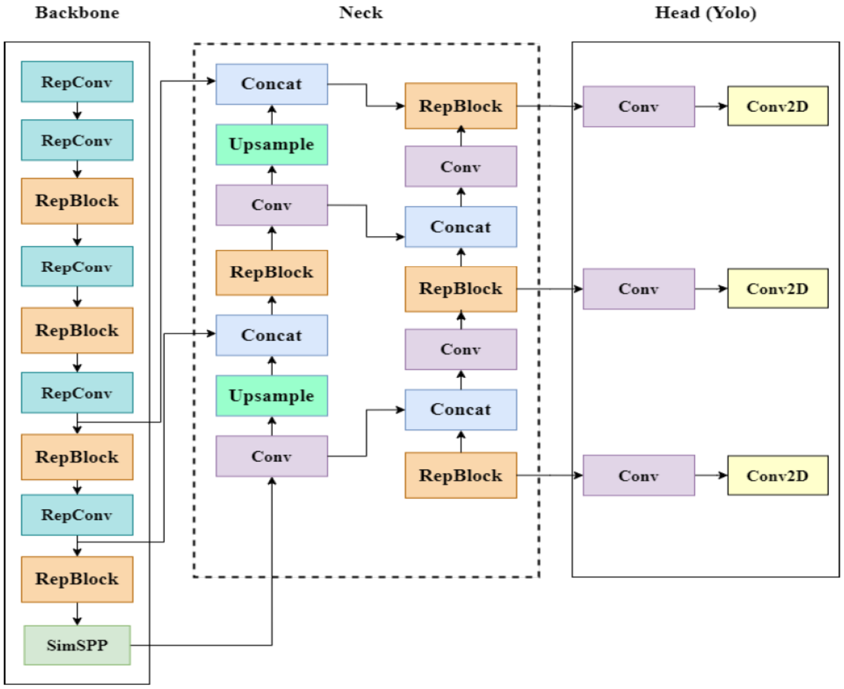


Fig. 2. Network structure of YOLOv6 [25].

By placing a layer between the network and the final Head, YOLOv6 Head is decoupled, which boosts speed. RepVGG style, a re-parameterizable structure that adopts a multi-branch topology and may be equivalently fused into a single 3*3 convolution, is introduced throughout the training phase. This fusion makes use of the memory and computing power. The training technique also incorporates an anchor-free paradigm, SimOTA label assignment policy, and Scale-Sensitive IOU (SIOU) Bounding Box regression loss for effective inferencing. [25]. The overall architecture of YOLOv6 is shown in Fig. 2.

### C. YOLOv7

The most recent entry in the YOLO series is YOLOv7 which developed by Alexey Bochkovskiy. The architecture is faster and more accurate at detecting threats than any of the earlier iterations. The backbone, head, and neck of YOLOv7 are the same as those of its earlier iterations. The main improvements made by the authors to the YOLOv7 model that enabled it to achieve this peak were:

*1)* In particular, the ELAN employs expand, shuffle, and merge cardinality to continually boost the network learning ability without breaking the original gradient route by considering the following design approach. This layer of aggregation is known as E-ELAN and is an improved version of the efficient layer aggregation (ELAN) computational block. E-ELAN also can direct different groups of computational blocks as they discover specific characteristics.

*2)* An original method of model scaling involves concatenating layers to scale the model's depth and width simultaneously; and

*3)* The addition of an additional head network to improve training and the use of a method called model re-parameterization to strengthen the model's robustness and improve its ability to generalize to new data [18, 26].

To provide additional gradient variation for diverse characteristic graphs, YOLOv7 was specifically suggested with direct access to the cascade of ResNet [27] or DenseNet [28]. Since RepConv [29] has an identity link, these structures dismantle the network structure. Because of this, the YOLOv7 was developed by deleting the identity connection in RepConv and developing the intended reparametrized convolution, accomplishing the effective combination of the reparametrized convolution and other networks.

YOLOv7 also takes advantage of the concept of deep supervision. It adds a further auxiliary head structure in the middle network layer as an auxiliary loss to direct the weight of the external network. The real-time inference is offered by these mono-modality object identification techniques, which also attain performance. These object detection models, however, only employ one stream. As a result, these models cannot use each stream's benefits, such as the accurate edges and appropriate lighting in IR photos and the object's colour and detail information in RGB images. Appropriate feature exploitation throughout each stream is necessary to improve object detection performance [26]. The structure of YOLOv7 is shown in Fig. 3.

### D. Related Work

*1) Small object detection:* Object detection has made tremendous strides in recent years. Despite these advancements, the performance between the detection of small and large items still exhibits a sizable performance disparity [31]. To improve the speed and accuracy of small object detection models, many studies have been performed, out of which we mention some of them.

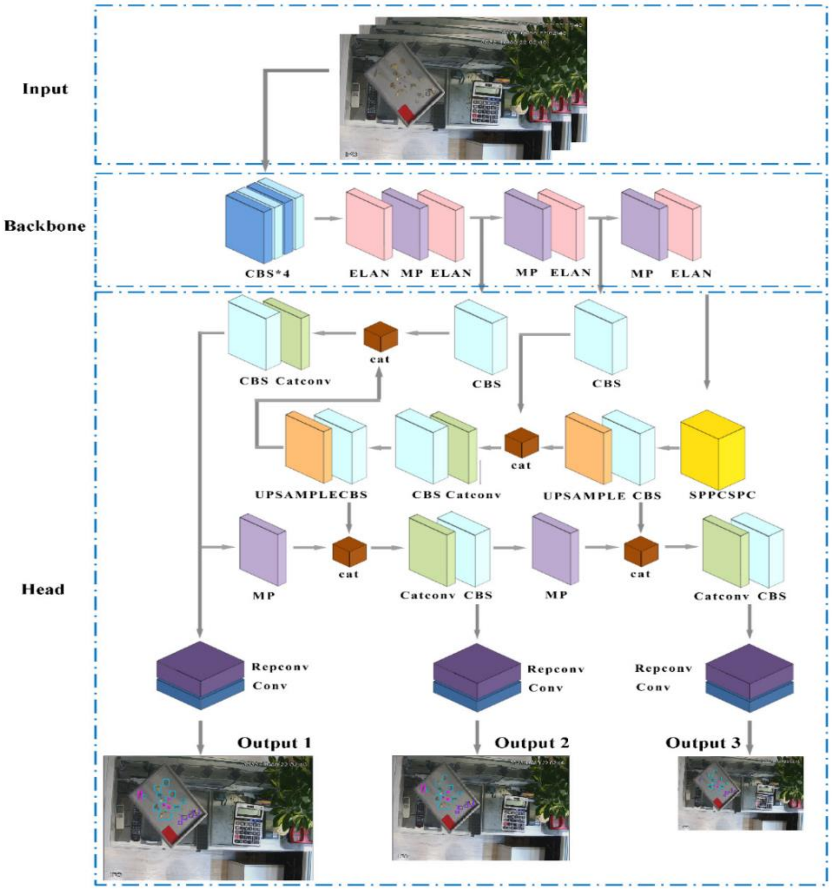


Fig. 3. Network structure of YOLOv7 [30].

In [32] YOLOv5 network modifications were made for aerial small target detection. By utilizing the first effective channel attention module, they altered the backbone, and the channel attention pyramid approach was suggested. Consequently, the module for identifying large items was removed in order to improve the identification of tiny things, and a detect layer was introduced to look for smaller objects. Finally, transposed convolution was used to produce upsampling rather than the already used closest neighbor interpolation. With the suggested technique, the mAP for the VEDAI dataset was 6.9%, for the xView dataset it was 6.7%, for the DOTA dataset it was 2.7%, and for the Arirang dataset it was around 2.4% for the small car class.

To address the problem of detecting small fish [33] presented a YOLOv5-based model. It tries to address the issues of inaccurate location and insufficient information for detecting underwater targets. First, they proposed combining the attention mechanisms CA and C3 structure to increase the network's ability to retrieve crucial information. Next, they suggested expanding the YOLOv5's three detection layers to four in order to address the issue of numerous, intensive detection tasks. Finally, in order to improve convergence time and lessen erroneous regression findings, GIOU loss was used in place of EIOU loss. The experimental findings demonstrated that the revised algorithm affected various indicators differently; mAP@0.50 achieved 94.9%, which was more accurate. The detection effect was 24.6 percent greater, and there were 248 picture detections overall—49 more than with YOLOv5. Performance for target detection was enhanced. Poor underwater fish swarm detection, tiny target location, few pixels, and low accuracy issues are all resolved.

In [31] some generic work for detecting tiny objects was done. They showed that one of the reasons for the poor average accuracy for small objects is the need for more representation of small things in training data. This is especially true for today's most advanced object detectors, which require a large enough training set of objects to guarantee that the predicted anchors match. They suggested two ways to improve the initial MS COCO database in order to resolve the problem. They first demonstrated how oversampling photos with small items during training may easily enhance performance on small objects. Second, they recommended an improved method based on pasting microscopic items. Compared to the state-of-the-art achieved by Mask R-CNN on MS COCO, their trials showed a 9.7% relative improvement, for instance, segmentation and a 7.1% relative improvement for object detection for tiny objects. The collection of augmentation techniques that have been suggested provides a trade-off between the accuracy of predictions for small and big objects.

In [34], their technique produced counting dense flocks of hemp ducks using positive detection findings. To improve the network structure of the YOLOv7 algorithm, three CBAM modules were added to the backbone network. SE-YOLOv7 and ECA-YOLOv7 were introduced for comparison studies along with an updated YOLOv7 algorithm that includes an attention mechanism. As a consequence of the experimental findings, CBAM-YOLOv7 was shown to have greater accuracy, as well as somewhat enhanced recall, mAP@0.5, and mAP@0.5:0.95 values. There was no change in the computational demand, and the FLOPS only slightly increased by 0.02 G. They also provided two labelling methods: whole-body labelling and head-only tagging, taking into consideration the overlap problem with hemp duck labelling frames. The full-body frame labelling approach showed a greater detection effect, whereas the head-only labelling method resulted in the loss of a significant amount of feature information.

*2) Jewellery detection:* Although there are not many articles available for jewellery recognition work, especially jewellery in store, some similar works can be mentioned.

Images of jewellery have been categorized using a machine-learning method [35]. They employed various methods. The first method takes advantage of the characteristics of AlexNet's support vector machine and support vector machine extracted from the input pictures. The Inception-v3 model is used in the second technique to carry out the same task. The results of the experiments showed that both techniques worked well, although Inception-v3 had a 0.9% higher success rate. In order to improve consistency, the Inception-v3 was then used to train the dataset from scratch. SVM has a 98.30% accuracy rating compared to 99.19% for Inception-v3.

In [36], the author focuses on picture recognition methods for automatically classifying stone-grinding flaws. After that, the stone quality is classified using the computed pertinent image attributes. A binary decision tree-based technique that uses decision thresholds modified from a training dataset does classification. In the end, the accuracy and time complexity of the proposed method are compared with more than twenty cutting-edge machine learning algorithms, and the results are competitive: on the D1 dataset, the best accuracy was reached among all tested algorithms; on the D2 dataset, 7% poorer prediction than the best algorithm was achieved. In addition, the algorithm consistently outperformed all others in all tests.

A novel technique for counting pearls was presented in [37]. The model is composed of an approach to counting and object detection. After a thorough investigation, they examine the key performance metrics of nine object detection algorithms. The results demonstrate that pearl identification can be accomplished using Faster R-CNN with ResNet152, which was pretrained on the pearl dataset, and only needs 15.8 ms of inference time with a counter following the initial loading of the model. Additionally, performance for precise counting and peal recognition of natural or artificial light is promising. Additionally, the network obtained 100% accuracy in counting pearls and a recall/AR@100 (medium) of 95% for pearl identification.

In [38], they generated a model for automated coin identification and recognition. In contrast to image processing techniques, which focus on the extraction of color, shape, and edge information, the majority of coin identification systems now in use are based on the physical characteristics of the coins. They have suggested a deep learning strategy for the recognition and detection of Indian coinage. AlexNet, a convolutional neural network that has already been trained, is trained using features including textures, colors, and forms. More than 1600 photos were used to train the model. They used a pre-assembled collection of photos to train AlexNet in their trials, which proved that there was sufficient training data. Results obtained demonstrated that the suggested technique outperformed more established methods.

## III. Methodology

YOLO was developed as a pre-trained object detector that can identify common items, including tables, chairs, automobiles, phones, and more [39]. We suggest a detection approach based on YOLO techniques to create a model that could recognize jewellery. Also useful in real-time applications are our models.

### A. Dataset

Our dataset consists of three various categories of jewellery (earrings, ring and pendant). The dataset includes images taken from our webcam. We collected our photos from a jewelry store webcam that was fixed in place. Fig. 4 shows some examples of our dataset. The dataset contains different image sizes including small target objects, which is more challenging to detect.

We selected photos of different types of shapes, sizes, resolutions, angles and different numbers of samples in each image. Using Roboflow, the 2456 photos from the dataset were increased to roughly 6500 images in order to improve our model. The following random processing steps were applied to each image: horizontal flip, rotation (between -15° and +15°), saturation (between -20% and +20%), exposure (between -10% and +10%), and brightness (between -20% and +20%). A maximum of three enhanced versions were produced for each image. In Fig. 5, examples of enhanced pictures are displayed. A training set (93%), a validation set (6%), and a test set (1%), each comprised of the labelled images, are then created.

### B. Google Colab

Utilizing Google Colab, which offers free usage of powerful GPUs, was helpful. A 12GB NVIDIA Tesla T4 GPU, shown in greater detail in Fig. 6, is used for all training and testing workloads. Our whole model was trained with an image size of 640 pixels and 50 training iterations. Other hyperparameters were adjusted using the YOLO default settings.

### C. Transfer Learning

It is always a good idea to begin the training process for an object detector with a model that has already been built using weights from extremely sizable datasets. Even if the training weights do not have the test items, this is OK. Transfer learning refers to this procedure. In order to help the network learn more quickly, a pre-trained model that uses weights from the COCO dataset as its initial weights is employed. Additionally advantageous is the fact that fewer data will be needed [39].

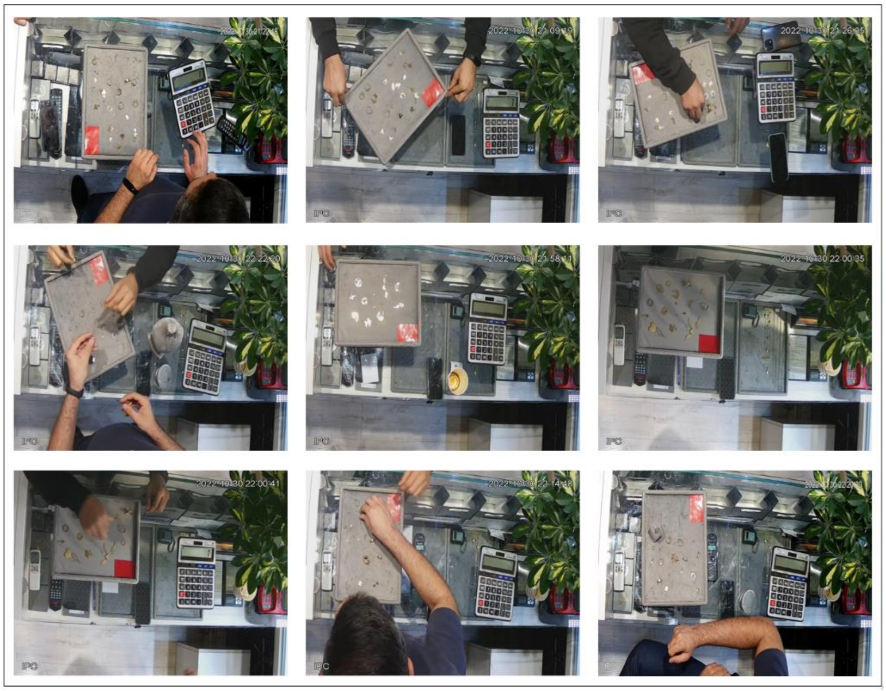


Fig. 4. Sample images from dataset.

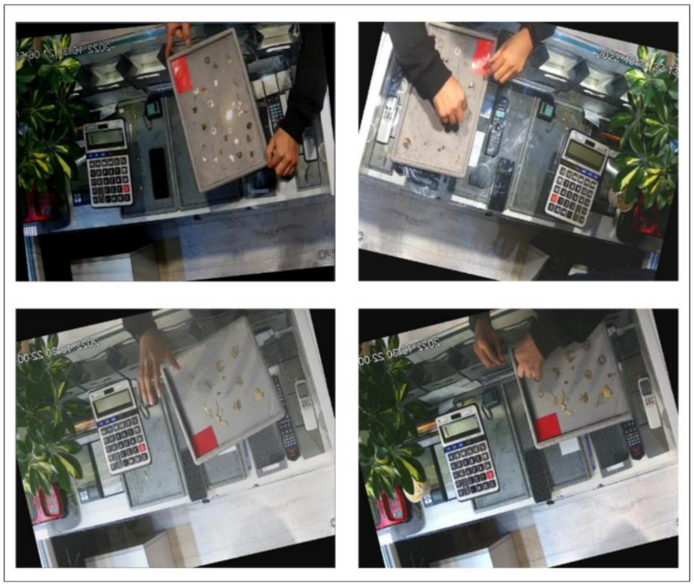
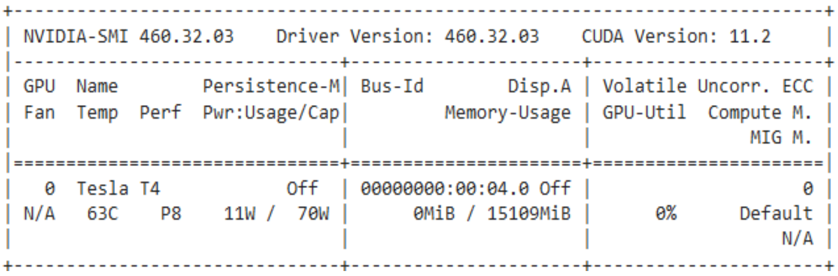


Fig. 5. Sample images of augmented images.



```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   63C    P8    11W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

Fig. 6. Details of Google Colab GPU.

## IV. Results and Analysis

In this section, we introduce the experiment's details, and then we show the training results using pretraining weights.

### A. Model Evaluation

In the proposed study, the gathered experimental findings were compared using a variety of assessment metrics, including accuracy, recall, F1-score, confusion matrix, and mean average precision (mAP). The true positive rate, or TPR, indicates how probable it is that things from the real world will be correctly identified. When a model produces no false negatives, which indicates that there are no bounding boxes that are not recognized but ought to be detected, it has a high recall. Eq. (1) provides the mathematical model for the recall [18]:

$$R = \frac{TP}{TP+FN} = \frac{TP}{\text{Total Ground Truths}} \tag{1}$$

The real positive and false negative are denoted in the equation above by the letters TP and TN, respectively. Eq. (2) defines precision as the percentage of correctly anticipated positives, often known as the positive predictive value. The exact model creates no false positives and only detects important things (FP) [18].

$$P = \frac{TP}{TP+FP} = \frac{TP}{\text{Total Predictions}} \tag{2}$$

The F-1 score, as stated in Eq., is the harmonic mean of the accuracy and recall scores (3).

$$F - 1 = 2 * \frac{P*R}{P+R} \tag{3}$$

Eq. (4) defines AP as the area under the PR curve and defines mAP as the average of all AP values across all classes/categories.

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{4}$$

where n is the number of classes [18].

We trained YOLOv5n, YOLOv5s, YOLOv5m, YOLOv6t, YOLOv6n, YOLOv6s, YOLOv7tiny and YOLOv7. In Fig. 7 and 8, we have shown F1-confidence and precision-recall curves respectfully. Correspondingly, the results of validation’s mAP, recall and precision are shown in Table 1. In addition, we put the training time and parameter values of each model for better evaluation.

### B. Model Losses

The results of loss functions of models are shown in Fig. 9 to 11:

### C. Analysis

We trained our model with YOLOv5n, YOLOv5s, YOLOv5m, YOLOv6n, YOLOv6t, YOLOv6s, YOLOv7tiny and YOLOv7. Fig. 12 to 14 provide examples of model predictions for the fresh and undiscovered images. Although YOLOv7 is the newest version and was launched in July 2022, it is allegedly better at object recognition than YOLOv5 and YOLOv6. All models that were trained on our jewellery dataset showed promising performance. However, we explored first YOLOv6 and next YOLOv5 are performing better than YOLOv7, and both YOLOv7tiny and YOLOv7 training accuracies were far worse on our dataset.

Due to their recent publication, the poor accuracies of the current YOLO versions may have resulted from insufficient experimentation, tweaking, and correction [40].

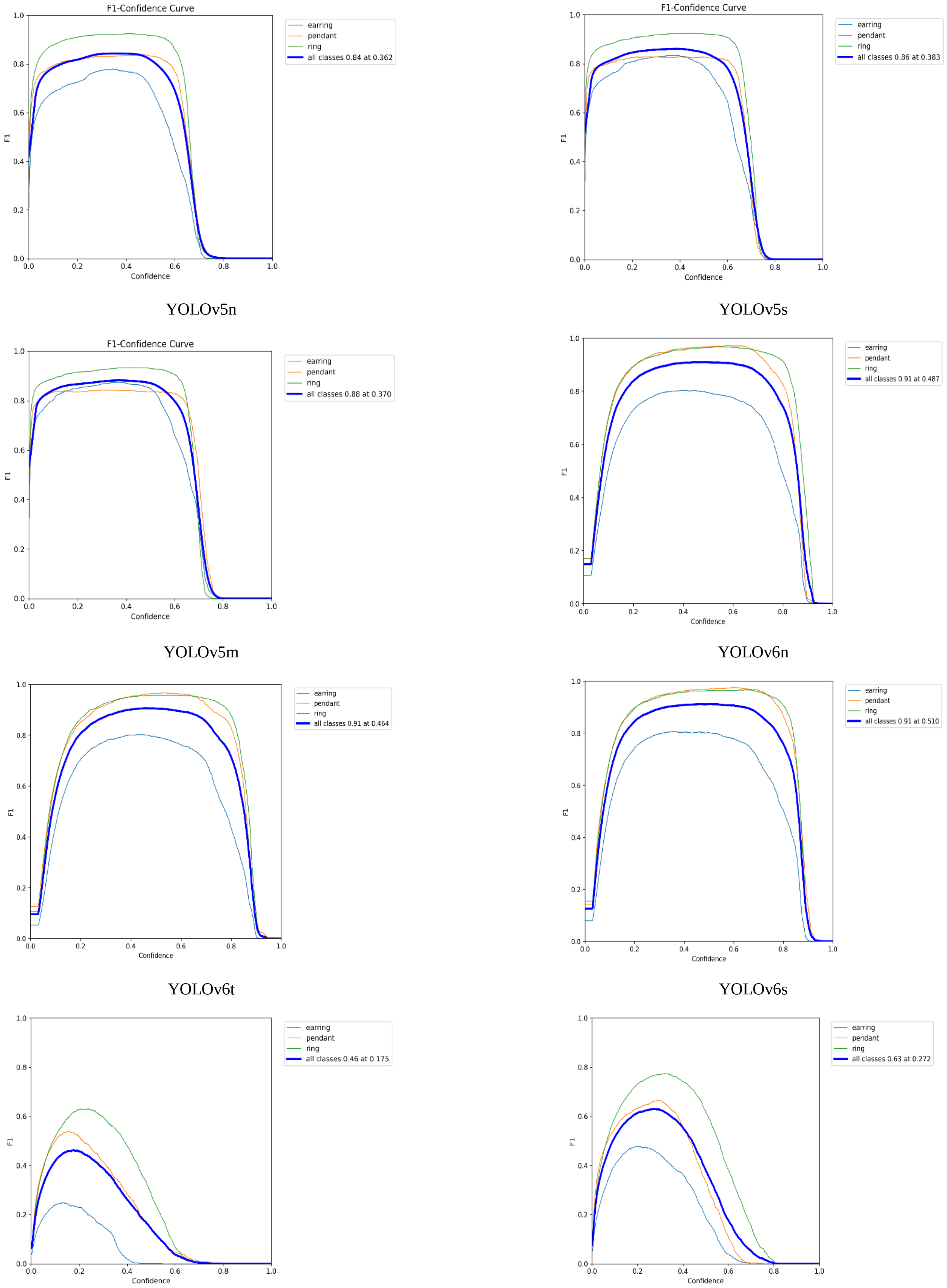
As shown by the results, the smallest model YOLOv5n achieved mAP@0.5 of 0.865 and the largest model, YOLOv7 achieved mAP@0.5 of 0.60. YOLOv6s model almost outperformed all other models and YOLOv6n is in second place. The worst result is for YOLOv7tiny except for training time which is the lowest with 2.393 hours. In the next place, the YOLOv7 has the worst performance. Due to the number of their parameters, YOLOv7 took the longest, clocking in at 4.518 hours. The detection speed of the model decreases as the number of parameters increases.

Another problem that should be investigated is that in all YOLOv5 models, the number of undetected objects is higher. It is feasible that expanding the dataset will boost the model's precision.

### D. Additional Analysis

We analyzed two other cases. One is to change the Augmentation strategy and the second is to train with more epochs. We augmented our images as described in Section III A with RoboFlow. We also know that there are techniques to augment data in the structure of YOLO algorithms. For further investigation, we once removed all data augmentation techniques in the YOLOv5s and YOLOv6s algorithms. And next time we only removed the mosaic and mixup techniques. The difference in accuracy and speed between the original models with the modified models is shown in Table II. Interestingly, we noticed that when we remove the three data augmentation, results in YOLOv5s increases slightly and in YOLOv6s is almost equal. But in both models the training time decreases by a large margin.

Finally, to evaluate training more epochs, we trained the YOLOv6s algorithm with 100 epochs, the results compared to 50 epochs are in shown Table III. It shows that training with more epochs is slightly better. Fig. 12to 14 show the experimental results for Yolov5, 6 and 7.



YOLOv5n

YOLOv5s

YOLOv5m

YOLOv6n

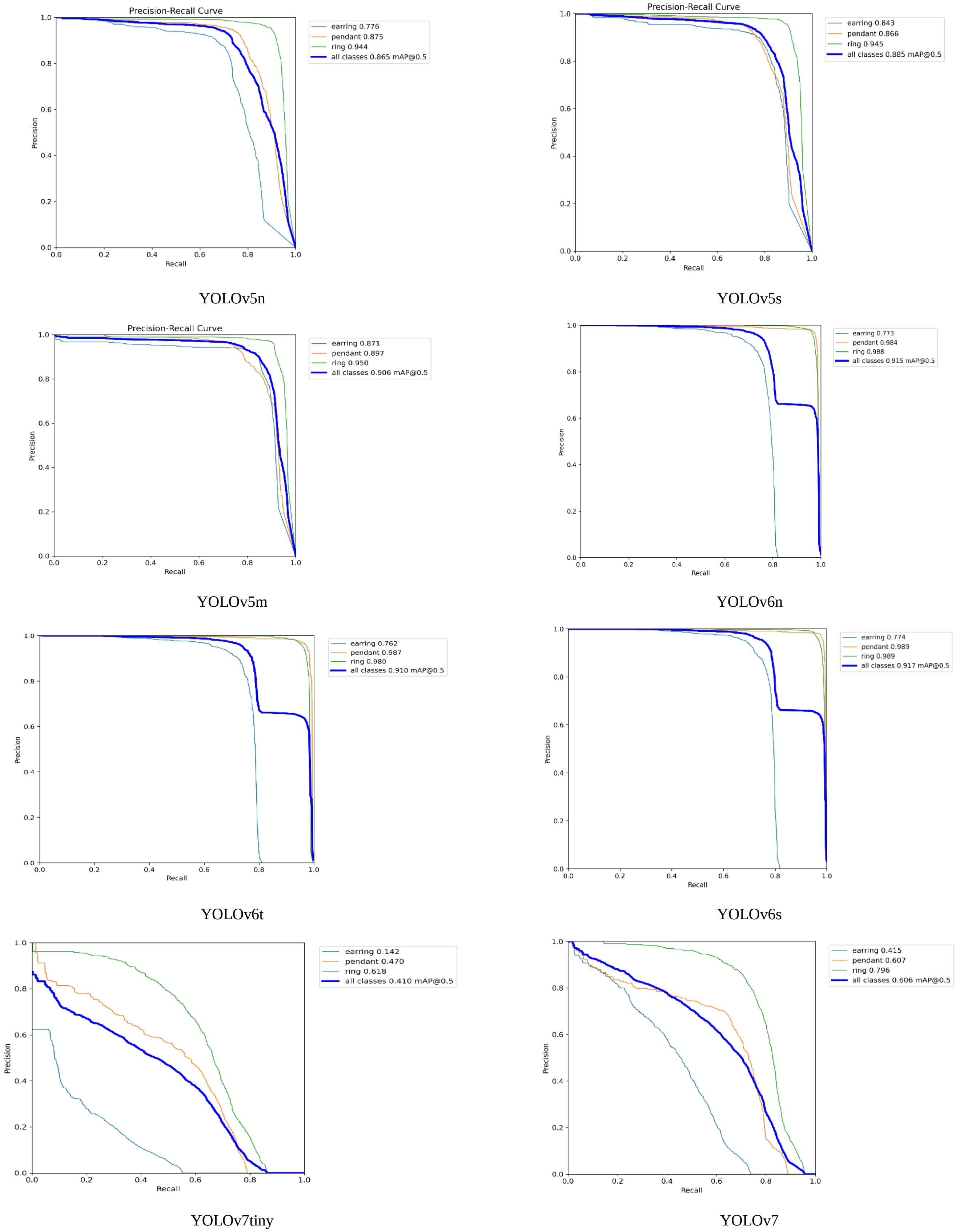
YOLOv6t

YOLOv6s

YOLOv7tiny

YOLOv7

Fig. 7. F1-Confidence curves.



YOLOv5n

YOLOv5s

YOLOv5m

YOLOv6n

YOLOv6t

YOLOv6s

YOLOv7tiny

YOLOv7

Fig. 8. Precision-Recall curves.

TABLE I. RESULT OF MAP, RECALL, PRECISION AND TRAINING TIME

| | *MAP@0.5* | *MAP@0.5:0.95* | *Precision* | *Recall* | *Params(M)* | *Trainig Time(hour)* |
|---|---|---|---|---|---|---|
| **YOLOv5n** | 0.865 | 0.547 | 0.902 | 0.794 | 1.76 | 4.206 |
| **YOLOv5s** | 0.885 | 0.585 | 0.913 | 0.817 | 7.01 | 4.431 |
| **YOLOv5m** | 0.906 | 0.615 | 0.932 | 0.84 | 20.86 | 4.404 |
| **YOLOv6n** | 0.915 | 0.598 | 0.946 | 0.883 | 4.30 | 3.670 |
| **YOLOv6t** | 0.91 | 0.595 | 0.936 | 0.884 | 9.67 | 3.823 |
| **YOLOv6s** | 0.917 | 0.61 | 0.955 | 0.881 | 17.19 | 3.726 |
| **YOLOv7tiny** | 0.41 | 0.14 | 0.48 | 0.45 | 6.01 | 2.393 |
| **YOLOv7** | 0.60 | 0.24 | 0.694 | 0.59 | 37.2 | 4.518 |

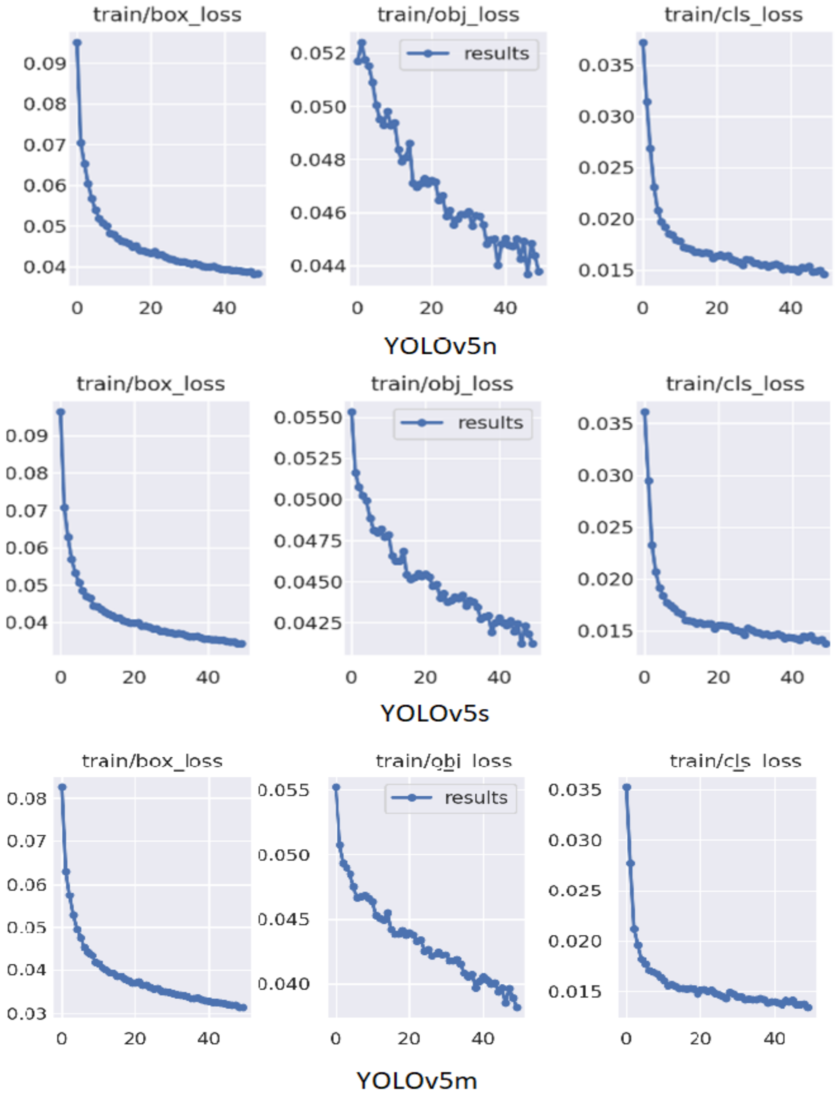


Fig. 9. Model Losses of YOLOv5.

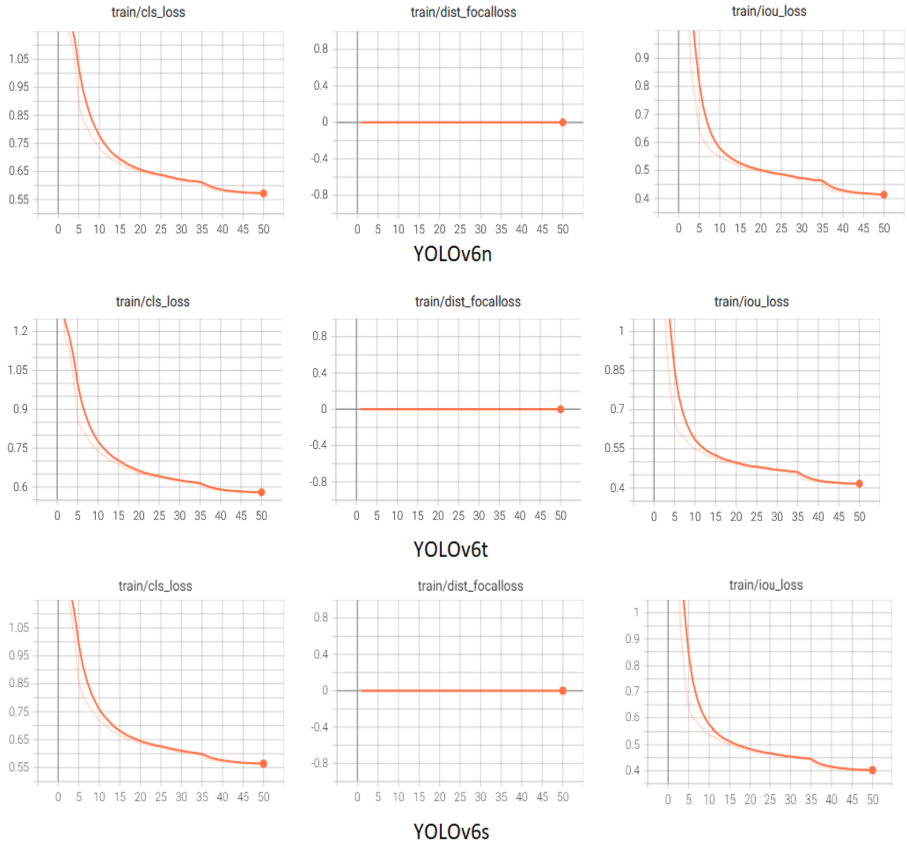


Fig. 10. Model Losses of YOLOv6.

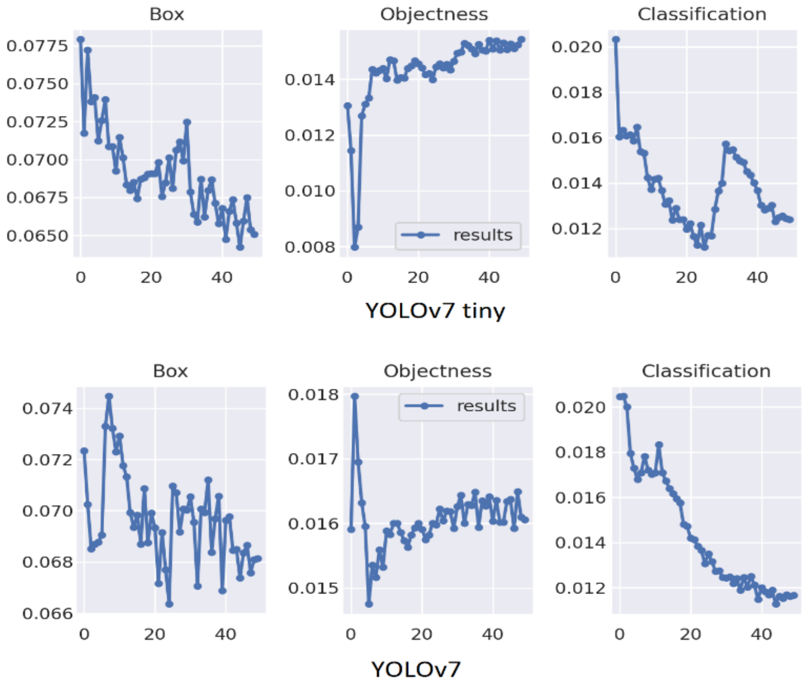


Fig. 11. Model Losses of YOLOv7.

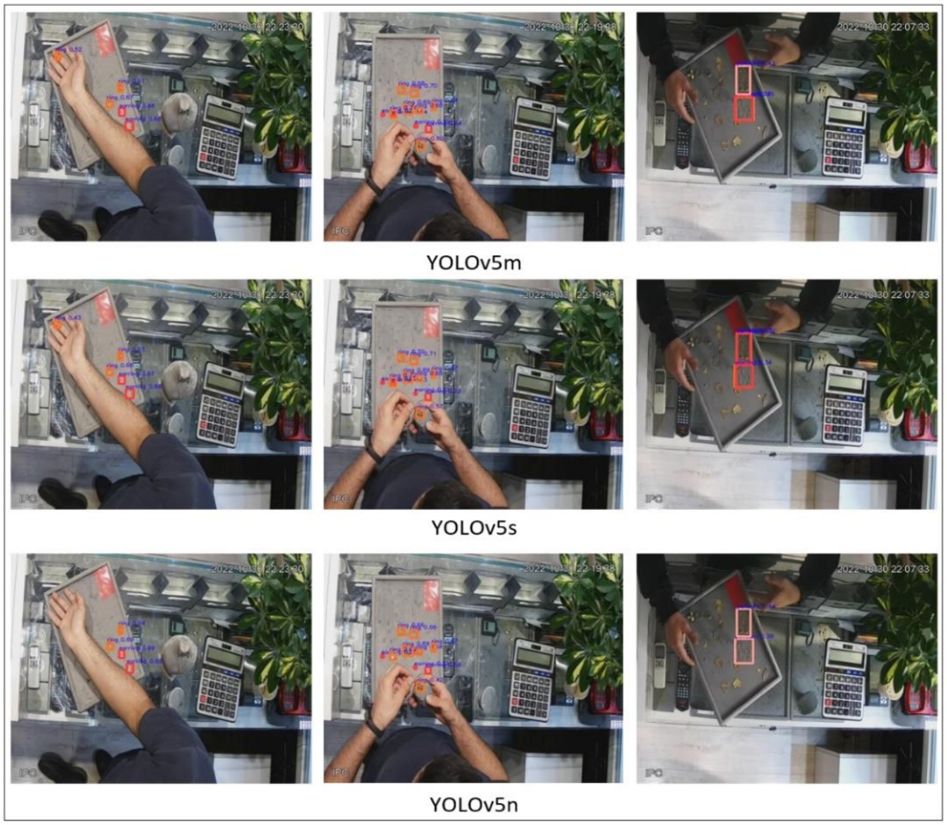


Fig. 12. Samples for detection with YOLOv5.

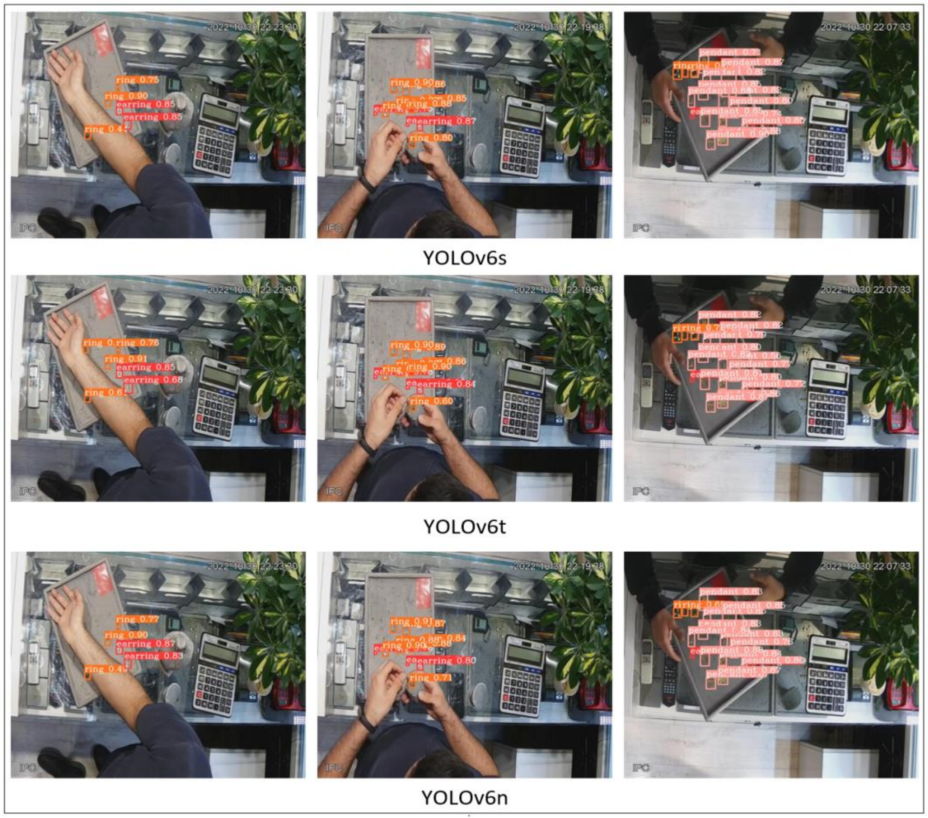


Fig. 13. Samples for detection with YOLOv6.

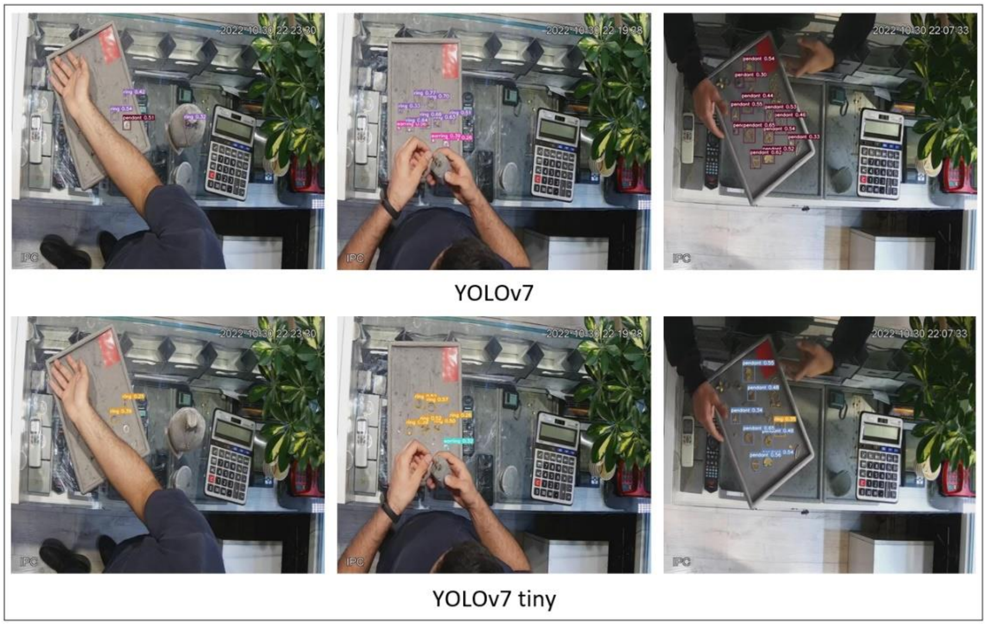


Fig. 14. Samples for detection with YOLOv7.

TABLE II. RESULTS OF AUGMENTATION ABLATION. THE “ORIGINAL” MODEL IS THE SAME AS THE PREVIOUS SECTIONS WITHOUT CHANGES AND WITH ALL THE AUGMENTATIONS. “W/O ALL” DENOTES WITHOUT ALL YOLO AUGMENTATION TECHNIQUES. AND “W/O 3” DENOTES WITHOUT JUST MOSAIC, MIXUP AND COPY_PASTE YOLO AUGMENTATION TECHNIQUES

| | $MAP@0.5$ | $MAP@0.5:0.95$ | $Precision$ | $Recall$ | $Trainig\ Time(hour)$ |
|---|---|---|---|---|---|
| **YOLOv5s (Original)** | 0.885 | 0.585 | 0.913 | 0.817 | 4.431 |
| **YOLOv5s (W/O All)** | 0.885 | 0.535 | 0.901 | 0.838 | 1.734 |
| **YOLOv5s (W/O 3)** | 0.905 | 0.62 | 0.915 | 0.839 | 2.117 |
| **YOLOv6s (Original)** | 0.917 | 0.61 | 0.955 | 0.881 | 3.726 |
| **YOLOv6s (W/O All)** | 0.893 | 0.505 | 0.935 | 0.869 | 2.129 |
| **YOLOv6s (W/O 3)** | 0.916 | 0.612 | 0.944 | 0.893 | 2.407 |

TABLE III. RESULTS OF INCREASING EPOCHS

| | $MAP@0.5$ | $MAP@0.5:0.95$ | $Precision$ | $Recall$ | $Trainig\ Time(hour)$ |
|---|---|---|---|---|---|
| **YOLOv6s (50 epochs)** | 0.917 | 0.61 | 0.955 | 0.881 | 3.726 |
| **YOLOv6s (100 epochs)** | 0.934 | 0.645 | 0.5 | 0.906 | 7.893 |

## V. CONCLUSION

In this work firstly a dataset consisting of 6k images of three classes, i.e., earrings, ring and pendant was created. Our photos were taken from a webcam fixed in a jewellery store. It has been tried that the photos be in different lighting conditions and angles as well as with different qualities. We have also used the benefits of data augmentation for our dataset. The performance of several YOLOv5, YOLOv6, and YOLOv7 variations was then compared based on the algorithm's accuracy, recall, mAP, and training time. YOLOv6s outperformed other algorithms for identifying jewellery, achieving the greatest mAP@0.5, of 0.917, and mAP@0.5:0.95, of 0.6, according to the results. The fastest algorithm was YOLOv7tiny with 2.393 hours of training time for 50 epochs. Also, we analyzed the model’s augmentation techniques and training with more epochs. The proposed method in this study has advantages compared to other existing methods because of high accuracy rate and low computation complexity. For future study, the results of the work show that we have reached good accuracy, but there is more work to be done, especially for the YOLOv7 algorithms, which can be compensated for by further optimization and investigation and the execution of more epochs. Another direction for future study on tiny/small object detection is to explore the use of attention mechanisms in deep neural networks. The use of attention mechanisms and generative models could help improve the performance of tiny/small object detection models and lead to better results on real-world applications.

## REFERENCES

[1] Zheng, J., et al., Insulator-Defect Detection Algorithm Based on Improved YOLOv7. Sensors, 2022. 22(22), pp. 8801.

[2] Xianbao, C., et al., An improved small object detection method based on Yolo V3. Pattern Analysis and Applications, 2021. 24(3), pp. 1347-1355.

[3] Girshick, R., et al. Rich feature hierarchies for accurate object detection and semantic segmentation. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[4] He, K., et al., Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 2015. 37(9), pp. 1904-1916.

[5] Girshick, R. Fast r-cnn. in Proceedings of the IEEE international conference on computer vision. 2015.

[6] Ren, S., et al., Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 2015. 28.

[7] He, K., et al. Mask r-cnn. in Proceedings of the IEEE international conference on computer vision. 2017.

[8] Redmon, J., et al. You only look once: Unified, real-time object detection. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[9] Redmon, J. and A. Farhadi. YOLO9000: better, faster, stronger. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[10] Redmon, J. and A. Farhadi, Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[11] Bochkovskiy, A., C.-Y. Wang, and H.-Y.M. Liao, Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.

[12] J, G. Ultralytics/yolov5 Available online: https://github.com/ultralytics/yolov5/releases/tag/v6.1. 2022.

[13] Liu, W., et al. Ssd: Single shot multibox detector. in European conference on computer vision. 2016. Springer.

[14] Liu, H., et al., SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. Sensors, 2022. 22(15), pp. 5817.

[15] Benjumea, A., et al., YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles. arXiv preprint arXiv:2112.11798, 2021.

[16] Wang, C.-Y., A. Bochkovskiy, and H.-Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696, 2022.

[17] Li, C., et al., YOLOv6: a single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976, 2022.

[18] Ali, L., et al., Development of YOLOv5-Based Real-Time Smart Monitoring System for Increasing Lab Safety Awareness in Educational Institutions. Sensors, 2022. 22(22), pp. 8820.

[19] Conley, G., et al., Using a deep learning model to quantify trash accumulation for cleaner urban stormwater. Computers, Environment and Urban Systems, 2022. 93, pp. 101752.

[20] Ahmad, T., et al., Detecting Human Actions in Drone Images Using YoloV5 and Stochastic Gradient Boosting. Sensors, 2022. 22(18), pp. 7020.

[21] Thuan, D., Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm. 2021.

[22] Yang, S.J., et al., Assessing microscope image focus quality with deep learning. BMC bioinformatics, 2018. 19(1), pp. 1-9.

[23] Guo, Y., et al., Improved YOLOV4-CSP Algorithm for Detection of Bamboo Surface Sliver Defects With Extreme Aspect Ratio. IEEE Access, 2022. 10, pp. 29810-29820.

[24] Horvat, M. and G. Gledec, A comparative study of YOLOv5 models performance for image localization and classification.

[25] Aburaed, N., et al. A Study on the Autonomous Detection of Impact Craters. in IAPR Workshop on Artificial Neural Networks in Pattern Recognition. 2023. Springer.

[26] Yun, J.-S., S.-H. Park, and S.B. Yoo, Infusion-Net: Inter-and Intra-Weighted Cross-Fusion Network for Multispectral Object Detection. Mathematics, 2022. 10(21), pp. 3966.

[27] He, K., et al. Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[28] Huang, G., et al. Densely connected convolutional networks. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[29] Ding, X., et al. Repvgg: Making vgg-style convnets great again. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[30] Wu, D., et al., Detection of Camellia oleifera Fruit in Complex Scenes by Using YOLOv7 and Data Augmentation. Applied Sciences, 2022. 12(22), pp. 11318.

[31] Kisantal, M., et al., Augmentation for small object detection. arXiv preprint arXiv:1902.07296, 2019.

[32] Kim, M., J. Jeong, and S. Kim, ECAP-YOLO: Efficient Channel Attention Pyramid YOLO for Small Object Detection in Aerial Image. Remote Sensing, 2021. 13(23), pp. 4851.

[33] Li, J., et al., CME-YOLOv5: An Efficient Object Detection Network for Densely Spaced Fish and Small Targets. Water, 2022. 14(15), pp. 2412.

[34] Jiang, K., et al., An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation. Agriculture, 2022. 12(10), pp. 1659.

[35] Singh, V. and P. Kaewprapha, A comparative experiment in classifying jewelry images using convolutional neural networks. Science & Technology Asia, 2018, pp. 7-17.

[36] Hurtik, P., M. Burda, and I. Perfilieva. An image recognition approach to classification of jewelry stone defects. in 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS). 2013. IEEE.

[37] Hou, M., et al., PDC: Pearl Detection with a Counter Based on Deep Learning. Sensors, 2022. 22(18), pp. 7026.

[38] Tajane, A., et al. Deep learning based indian currency coin recognition. in 2018 International Conference On Advances in Communication and Computing Technology (ICACCT). 2018. IEEE.

[39] Hatab, M., H. Malekmohamadi, and A. Amira. Surface defect detection using YOLO network. in Proceedings of SAI Intelligent Systems Conference. 2020. Springer.

[40] Chen, E., et al., Real-time detection of acute lymphoblastic leukemia cells using deep learning. bioRxiv, 2022.