Deep Learning - Final Project: Xmas Tree Pet detection

Ting An, Chien
Yuan Ze University
Taiwan

Abstract

Nowadays, people keep pets as children. However, sometimes pets can be mischievous, vandalizing people's properties. In this project, I fine-tuned a transfer learning model based on YOLOv7 to do object detection, distinguishing the pets and the properties, Xmas tree in this case, and notify owners once their pets stay around the Xmas tree for too long.

1. Introduction

Videos about pets destroying Xmas trees are trending when it comes to Christmas every year. Although the viewers might find it cute and funny, as a pet owner, I have to say it is really tiring. The reason why I chose this topic for my final project is to keep my pets away from the Xmas tree. With the help of object detection, whenever our fur babies approach or even attack the tree, we will receive a notification and take action before anything terrible happens. After all, saving our Xmas tree and decorations is saving us money.

2. Data

The dataset is a series of labeled image sequences converted from videos captured by the security camera in my living room. I labeled 496 images with LabelImg as training data and the YOLOv7 tool does augmentation automatically, as shown in Figure 2.

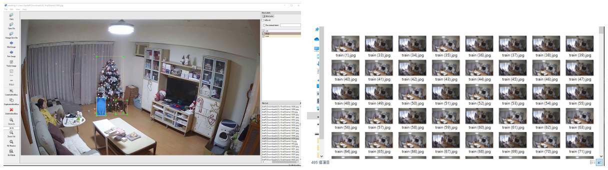

Figure 1. Labeling with LabelImg and labeled dataset

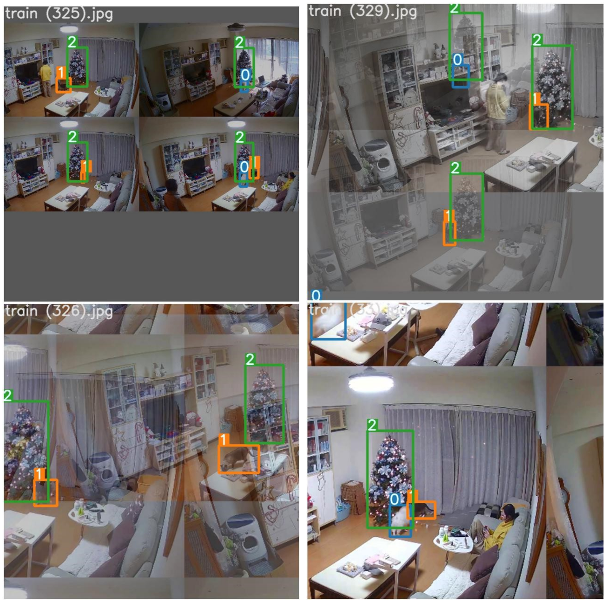

Figure 2. YOLOv7 Augmentation

3. Methods

The method applied in this project is transfer learning based on YOLOv7, with fine-tuning using a customized dataset. I've considered using the original pre-trained model YOLOv7 along with open-source Coco dataset images including cat and dog labels, but the result shows low accuracy and confidence, as shown in Figure 3.

4. Experiments

In this project, I use an android emulator to stream the security camera filming the Xmas tree. Next, capture every single frame and determine whether any cat or dog has come across the scene and overlaps with the Xmas tree. If a dog, a cat, or both a dog and a cat approach the tree for more than 40 frames within 50 frames. It will trigger the Line notify bot to send a message along with a labeled image to inform owners, "Your dog/cat is being naughty!". The time interval between two Line notifications will be no less than

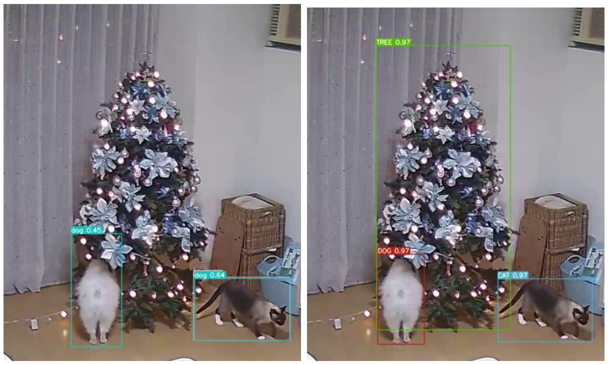

Figure 3. Comparison of pre-trained YOLOv7 with Coco dataset weight and after fine-tuning with both conf-threshold and iou-threshold is 0.3

1 minute.

The outcomes with different confidence thresholds of my model are shown in Figure 4. Figure 5 includes a comparison of day and night conditions, with different natural lights and indoor lights, which shows the model works efficiently in different environments. The F1 score is shown in Figure 6. When the confidence falls between 0.9-0.95, we get the F1 score of 1. Therefore, the confidence threshold is set to 0.9 in this case and the result came out fairly accurate from practical observation without misjudgment. Figure 7, respectively shows the loss of bounding boxes, objects, and classes.

The common failure modes of my model would be when pets go behind and are blocked by the Xmas tree. In this case, pets are not detected, and thus failed to inform owners.

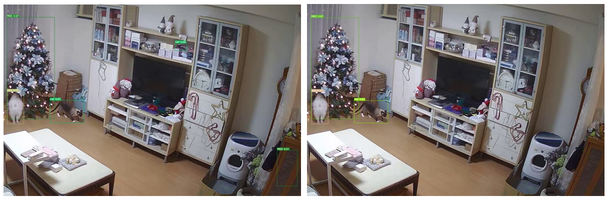

Figure 4. Comparison of 0.003 confidence threshold and 0.7 confidence threshold of my model

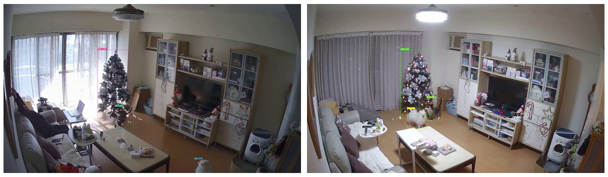

Figure 5. Comparison of day and night conditions with different natural lights and indoor lights

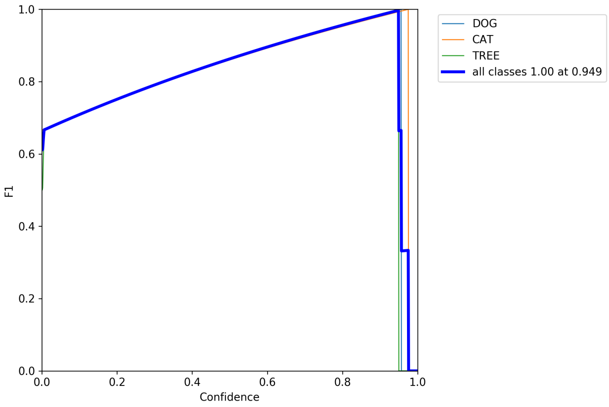

Figure 6. F1 curve

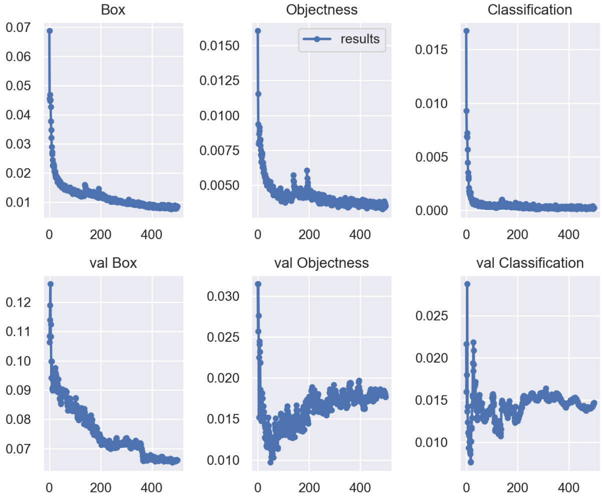

Figure 7. Loss

5. Conclusion

In conclusion, the Xmas tree is rescued. To make it more functional, my ideas for future extensions are:

1) Buy a better security camera, and connect it to edge devices coordinate with sirens. So, whenever pets go near Xmas tree, the alarm will go off and keep them away immediately.

2) Probably combine the model with RNN to eliminate the pet-behind-tree hazard.