Machine Learning Models for Segment Object Detection: TensorFlow Detection Model Zoo

Zhang Ting
zhangting.199612@gmail.com
Boston, MA

Introduction

Nowadays machine learning is more and more useful in the field of object detection. However, knowing this truth, there are still many people don't know how to take advantage of the machine learning mathematical knowledge to train their model and predict on the new objects when given several datasets. So, as an opensource library, TensorFlow provides several machine learning models for users to use. These models are API indeed and could work out well as long as users install and utilize them correctly through the guide. They are user friendly and the only thing the users need to do is to make sure that the category they are interested in is already in the datasets provided by the model zoo.

Also, machine learning problems can be various segment. Object segmentation is one extension of the object detection in machine learning problem conditions. In some specific conditions, the simple bounding box is not enough to outline the profile of the object. As an extra mission, we need to segment the object with its specific featured profile once its rough region is detected. Some of the models in the TensorFlow Detection Model Zoo can implement the segmentation and output the profile masks.

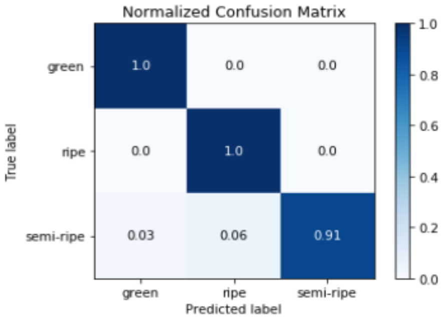
Analysis

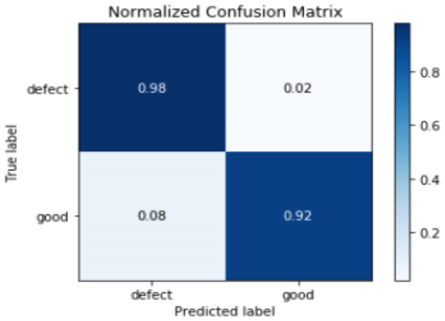
"We provide a collection of detection models pre-trained on the COCO dataset, the Kitti dataset, the Open Images dataset, the AVA v2.1 dataset and the iNaturalist Species Detection Dataset."[1] According to the description provided by Github TensorFlow developer, the main datasets provided by the model zoo cover the COCO dataset which is provided by Microsoft and covers 91 object categories in daily life, KITTI dataset which includes the driving condition object detection, OpenImage dataset which is provided by Google, AVA v2.1 dataset which contains multiple human action images, and iNaturalist Species dataset which includes almost all the species images in nature.

So, for the category we are interested in which is covered by the above dataset, we could simply download these pre-trained models from the TensorFlow model zoo github and run them locally or just run on the jupytor notebook. The API could give out a result which contains boxes or masks as well as the prediction scores.

Results:

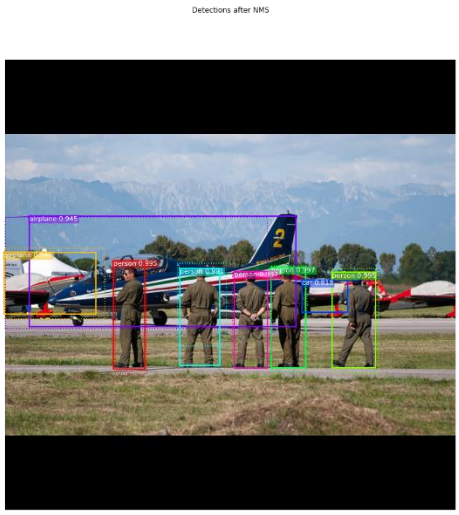
1. Example: ssd mobile-net v1
   "The model chosen is called the SSD mobile-net v1 COCO, this is essentially the mobilenet model pretrained on the COCO dataset (Lin et al, 2015). This model was chosen because despite having the lowest score among the COCO models, it has shown to have achieved the highest speed among them (TensorFlow Detection Model Zoo, 2017)." [2]

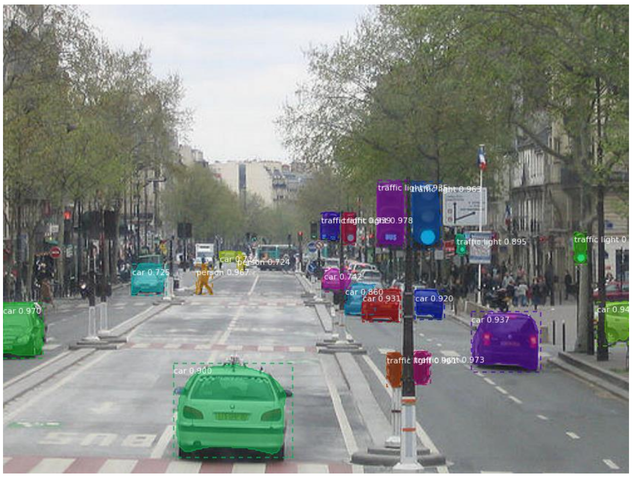


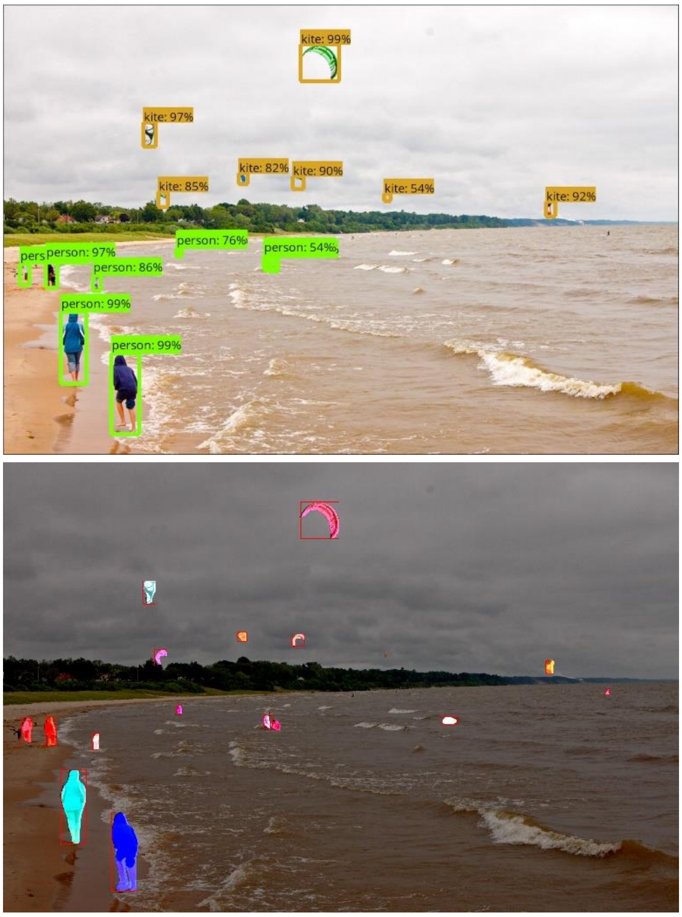


The result is quite good and we can see that this ssd model-net v1 COCO performs well in predicting different types of mango in their experiment.

2. Example: Mask R-CNN (Bounding box and masks)
   "This is an implementation of Mask R-CNN on Python 3, Keras and TensorFlow. The model generates bounding boxes and segmentation masks for each instance of an object in the image. It's based on Feature Pyramid Network (FPN) and a ResNet101 backbone" [3]

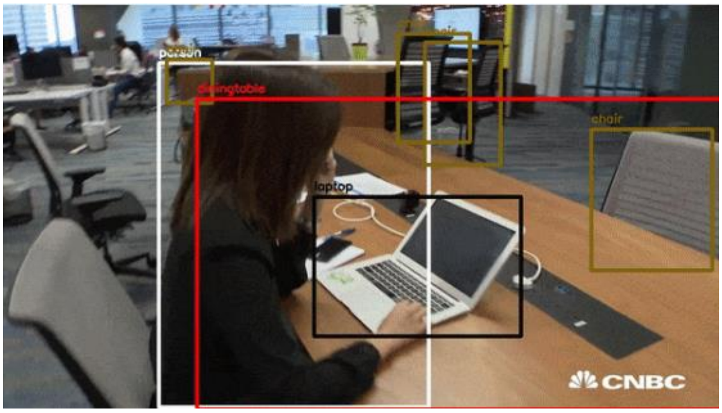






[4]

3. Example: YOLO TensorFlow ++
   “TensorFlow implementation of 'YOLO: Real-Time Object Detection', with training and an actual support for real-time running on mobile devices.” [5]



Pros & Cons

Pros:

1. As an opensource API model, the model provided by the TensorFlow Detection Model Zoo are easy to use and run, especially easy on jupytor notebook.
2. Since the datasets that the models pre-trained on are provided by companies like Microsoft and Google who are good at collect big data, thus the datasets can be really big enough for the training. The training and prediction accuracy can be reliable.
3. The datasets cover most of object fields in our daily life and it cannot be difficult to find your interest category in their API.
4. Even instance segmentation could be implemented using mask-rcnn in some specific conditions. Not only the rough bounding box but also the featured segmentation could be shown to the users.

Cons:

1. The hardware condition needs to be considered since the performance of the hardware could directly influence the speed of training and prediction of the model
2. To train an instance segmentation model, a groundtruth mask must be supplied for every groundtruth bounding box. Then this supervised machine learning labelling period can be complex for the users if they want to implement the instance segmentation.

Recommendation

For people who want to use the TensorFlow Object Detection Model Zoo, I have three suggestions:

1. First, make sure that the category or the object types you are interested in is covered in the TensorFlow Detection Model Zoo and then continue your training and prediction following the opensource guideline.
2. Second, be sure to have a python environment to run in and check whether your hardware condition is good enough to hold the training and detection problem of the usage of model. If your hardware condition is not met, you could apply for a cloud computing platform usage or just execute your API on some opensource platforms.

3. Last, according to the description of the TensorFlow Model Zoo, only the mask-rcnn models could output the masks instead of bounding box, which means that at present only mask-rcnn models could do the instance segmentation in this model zoo. So, make sure that you choose the correct model before you install and execute the model. The goal of whether doing the instance segmentation needs to be clarified first before using the system.
4. According to Pirkko Mustamo, there can be something you need to check with your model, "1) Pre-trained model will not work on the data without fine-tuning. 2) Fine-tuned model will work reasonably well on the given data…4) Using more variable training data will improve results on new images." [6] Thus, maybe sometimes adding some new training images can be necessary.

Conclusion

Generally speaking, TensorFlow Object Detection Model Zoo is quite a good model library for machine learning users. Users could take advantage of the API provided by the zoo and just easily install them following the guide. While users may need to add some new images to the training data to increase the accuracy, the general accuracy of the original pre-trained model is actually enough high. Also, users should make sure their hardware condition and detecting goal meet the features of the model. As a conclusion, using the model from TensorFlow Zoo can really be a nice experience.

Reference

[1]. Github: Machine Learning Object Detection Model Zoo
https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

[2]. Ryan Joshua H. Liwag, Kevin Jeff T. Cepria, Anfernee S. Rapio, Karlos Leo F. Castillo, Melvin K. Cabatuan, Edwin J. Calilung. "*Single Shot Multi-Box Detector with Multi Task Convolutional Network for Carabao Mango Detection and Classification using Tensorflow*". New York. De La Salle Universty. 2017. Printed.

[3]. TensorFlow Detection Model Zoo: Mask R-CNN
https://modelzoo.co/model/mask-r-cnn-keras

[4]. Github: TensorFlow Instance Segmentation
https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/instance_segmentation.md

[5]. TensorFlow Detection Model Zoo: YOLO TensorFlow++
https://modelzoo.co/model/yolo-tensorflow

[6]. Pirkko Mustamo. *"Object detection in sports: TensorFlow Object Detection API case study"*.University of oulu. 2018. Printed.

Extra Topic: Summary of Others

After discussing about our topic, I get a much clearer hierarchy level architecture about the machine learning models.

Firstly, according to Reardon Michaela:

- Machine learning is the art of using data to predict the outcome of future data. It is widespread in today's technology including predictive text, facial recognition, fraud alert and shopping recommendations.
- There are multiple types of machine learning tools such as Business Rule Management Systems (BRMS), Neural Networks, Deep Learning, Data Mining, Case-Based Reasoning (CBR), and Genetic Algorithms. These tools can be used in conjunction with each other to help hone in on features and model parameters.

It is a general and correct description of current art of machine learning.
Then Reardon Michaela discusses about three famous big models: Tensorflow, Pytorch and Keras.
TensorFlow

- Developed by Google
- Open Source
- Static Model Development
- Option to use TensorBoard, a graphical representation of TensorFlow
- Has multiple use methods (Python, JavaScript, Mobile, Production)
- Lots of add ons available
- Extended documentation, multiple tutorials

PyTorch

- Developed by Facebook
- Open Source
- Able to build Dynamic modeling
- Simplistic, very similar to Python
- Good support

Keras

- Lightweight
- Developed by someone at Google
- To be run on top of TensorFlow (or other)
- Intended for fast prototyping

Reardon Michaela recommends for the tensorflow for its parallelism and faster, more efficient training.

Then Zuo Hao, Lu Jie and I focus our work on the tensorflow.

About tensorflow zoo:

These models are API indeed and could work out well as long as users install and utilize them correctly through the guide. They are user friendly and the only thing the users need to do is to make sure that the category they are interested in is already in the datasets provided by the model zoo

Zuo Hao gives out a general model of the tensorflow package usage:

- Detection: Model interface for users to easily implement their own model
- Preprocess: shift the images before using as input in the detector
- Predict: Produce “raw” prediction tensors which can be passed to further functions
- Postprocess: Convert predicted output into final detections
- Loss: Based on the groundtruth to compute the loss
- Restore: Load a checkpoint into the TensorFlow graph

I think it is good and help the readers get clearer knowledge about the using of tensorflow model.

(1) CNN – RCNN – Fast RCNN – Faster RCNN

CNN: Firstly, the reason for choosing CNN is that the information from image can be large. So, it is important to reduce the calculation by reducing the connections between pieces of data. By using CNN, the number of dimensions can be reduced to a rational level

RCNN: Then, R-CNN is a kind of upgraded CNN, which can determine the location of the objects in an image. In R-CNN, the first step is to determine the possible regions, which may contain objects. Then, use CNN to determine the label.

Fast RCNN: In Fast R-CNN, different RoI share the characteristic values, so the input value to CNN will not be the regions but the whole image, which means it reduces the computation by only generating the feature value map once

Faster RCNN: Faster R-CNN is using anchor boxes, fixed size segmentations to label the location.

(2) The most frequent model we mentioned is Mask RCNN. If you want do the instance segmentation, then you need to use the Mask RCNN

For Mask-RCNN:

Object Detection

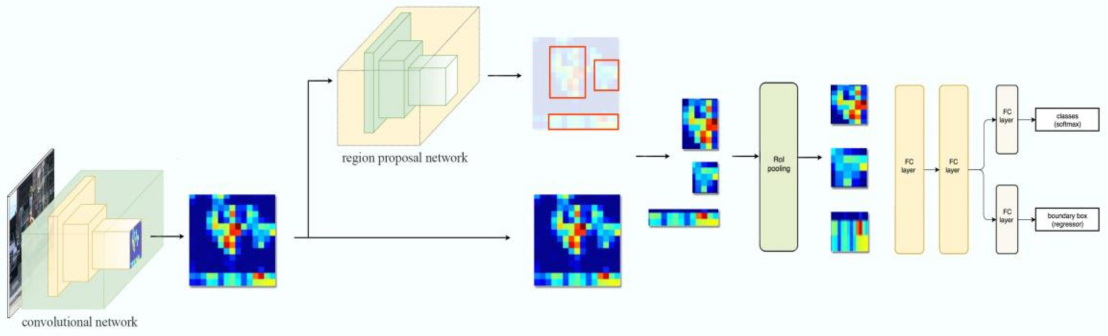
To solve the computer vision problems such as achieving object detection, a model Mask-RCNN can make a good performance on the inference of the object detection

Mask-RCNN

Mask-RCNN extends Faster-RCNN via adding a branch for inferring segmentation masks on each Region of Interest (ROI), in parallel with the existing branch for classification and bounding box regression.

Also, the mask RCNN could be used for instance segmentation which is stronger than Faster RCNN which only uses the bounding box.

Architecture:



Then in the part of Li Ganghao, he introduced the unsupervised learning.

(3) If you want to make the real-time detection for the video stream, YOLO is good because of its detection speed.

Introduction:

The various problems in pattern recognition are solved according to training samples whose categories are unknown (not marked)

One of the main methods: **Cluster analysis**. That is used in unsupervised learning to group, or segment, datasets with shared attributes in order to extrapolate algorithmic relationships.

Also, he lists some pros & cons:

Pros:

(1) No label & Clustering

(2) Reduce dimension

(3) Non-independent

(4) Interpretable

(5) Expandability

Cons:

(1) Low accuracy and validity

Unsupervised learning could be applied into several fields:

(1) Data mining

(2) Abnormal detection

(3) Detect a segment object
(4) Advertisement