Politecnico di Torino

Mechatronic Engineering

Title: Machine Learning Algorithms for Service Robotics Applications in Precision Agriculture
Supervisor: prof. Marcello Chiaberge
Candidate: Angelo Tartaglia [S242338]
Date: 29 October 2018

1. Abstract

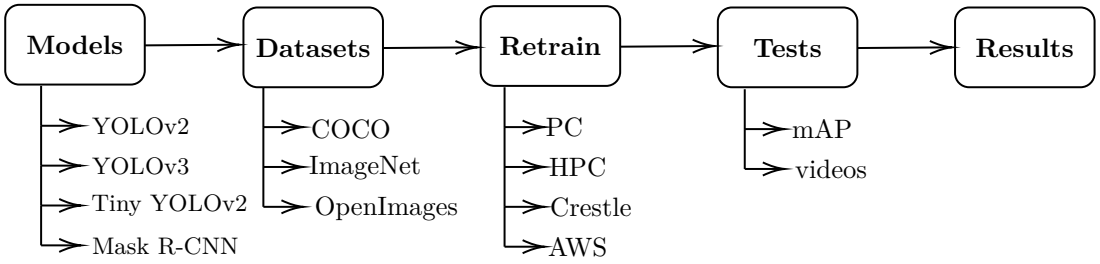
World population is growing faster than expected. Every year Earth resources are consumed faster, as proven by the early fall of the *Earth Overshoot Day* that claims the end of year resources. One of the simplest solutions to this problem would be investing in technologies and innovations towards a smart extraction of what population needs. In the last few years a lot of companies introduced automation and robots to improve production in terms of time and obviously cost. Agricultural world is not distant from this evolution; in fact many of the works attempted are now helped by a set of machines that simplifies human work. Hitherto the application in precision agriculture has been always under the human control; there are a lot of applications that see the participation of a worker and in parallel a machine used for a lot of tasks. In the future may an entire field will be managed by a group of automated robots that work together. Those machines will require a lot of specific features likes autonomous navigation, mapping, visual object recognition and many others. This thesis is part of that project and it regards the detection and classification of fruits for future application of auto-harvesting and health control. In a more specific way apples will be considered in this thesis work for fruit application and methods and algorithms as Y.O.L.O. and Mask R-CNN to do the processing of images will be described.These techniques permits the detection of apples in post processing and in real-time with accuracies that range over 32% to 78%. The final result can be used in future applications for spatial localization of fruits and for the detection of possible diseases. It should be emphasised that, even if the thesis shows the results of the object class apple, the algorithms can be applied in a wide range of objects with the only requirement of a different training images dataset.

2. Objectives

All the solutions found until this thesis work are referred to a classification of fruits into images. The aim of my project is instead the *detection* of fruits that implies not only the comprehension of the presence of an apple in pictures but also the identification of its localization. This is a big deal considering the power needed by the hardware and the huge dataset needed for the training of a network. The final platform can be applied in a lot of manners, from auto-harvesting to the monitoring of health conditions of the plant and fruits. The purpose of this thesis is to set the base for a general methodology for future applications and to make clear how a neural network works.

3. Developed Works

In order to correctly explain the work done, the result section will be divided in the following way:



3.1 Models

For the purpose of this thesis the utilisation of models for object detection was necessary. All the models have the purpose of recognising selected object with their localisation in the picture, assigning to each one the *accuracy* (a factor that indicates the confidence of the system, from 0 to 1) of the prediction. As the available models are numerous and various, a selection of the better ones was carried out at the beginning of this work. The chosen models were Y.O.L.O. and Mask R-CNN. Y.O.L.O. (*You Only Look Once*) is one of the last neural networks for the creation of bounding boxes that enclose detected objects. The operation principle of this model is the following: i) dividing the input image into a grid; ii) for

each grid cell the model has to predict with the usage of priors (boxes of different sizes) the presence of an object; iii) assigning the detected object to a class and calculate the accuracy; iv) purifying the prediction removing the bounding boxes with low accuracy (usually 0.3) to give the final result. Hence the output image is the original one with the addition of boxes over the detected objects with the reference classes and accuracies. Y.O.L.O. has various versions and three of them have been tested: YOLOv2, TinyYOLOv2 and YOLOv3. The major differences among these versions can be summarised as following: YOLOv2 is a big evolution from the original one and includes a lot of features for speed up the processing of images and improve the results (YOLOv2 was one of the most used networks); TinyYOLOv2 is a lightweight version of the original with a reduced dimension of the network that however can be used for embedded application as on *RaspberryPi* with *MovidiusNCS*; YOLOv3 is the latest version of this network with a huge modification on how the classes are predicted. Mask R-CNN is instead the last evolution of the R-CNN (*Regional Convolutional Neural Networks*) and gives a slightly different result. After the recognition of objects with the classification and accuracy results, the network creates a *mask* of all pixels attributable of that specific object. This new type of model belongs to *instance segmentation* category (a step further object detection), giving as result the original image with all the detected objects enclosed in their recognised shapes.

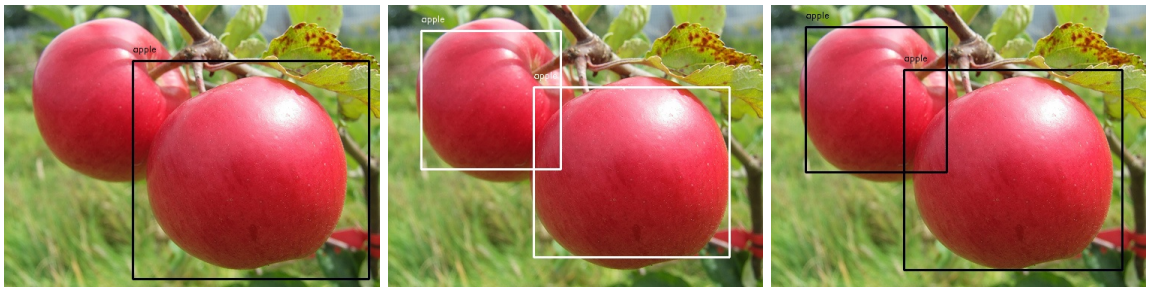
3.2 Datasets The models briefly explained require huge datasets for the original training and the specific re-train done during the thesis period. In fact all deep learning networks base their functionality on the features extracted from the training images. The used ones were: *COCO*, *ImageNet* and *Open Images*. COCO (*Common Object in COntext*) is a dataset containing over 200k divided into 80 classes with bounding boxes and masks. The neural network used, YOLO and Mask R-CNN, have been trained on COCO as base dataset. ImageNet contains over 14mln images divided into 21841 classes used for classification purposes. It has been used for the creation of a new dataset for mask detection using mask R-CNN; over 300 images have been hand-annotated with the shape of apples and hence they have been given to the network during the re-training phase. Open Images is one of the last and huge datasets for object classification and detection. The server that hosts this dataset does not permit the download of single classes so a new *Python* script has been produced during the thesis (*OIDv4 Toolkit*) in order to solve this problem; this work has been published also in the main page of the Open Images website for its benefits. After the download of apple images, the re-trainings of YOLO networks have been done.

3.3 Re-Train The models used have been trained for the detection of 80 different classes. This can be useful for several applications but, with the perspective of an utilisation of this model into a field of apple, the re-training can enhance the performances of the networks. In fact with the *tranfer learning* technique it is possible to inherit the good performance of the original networks and re-train them on only one class, apple in this case. The networks have been subjected to modifications of their structures: the number of possible classes and other specific hyper-parameters as batch dimension, learning rate and input image size. Hence re-trainings have been done using different hardware platforms: a personal notebook has processed the re-training of TinyYOLOv2 in approximately 13 hours, *Crestle* (a web platform) has been used for the retraining of mask R-CNN and finally *AWS* (Amazon Web Services) hosted the re-training of YOLOv2 and YOLOv3.

3.4 Test Phase The networks obtained and the original ones have been tested using principally two methods: *mAP* evaluation and video processing. mAP (*mean Average Precision*) is one of the most used metrics for network evaluation. Briefly, it compares the predicted bounding boxes of the test images with the ground-truth (the boxes defined by the user) and gives as output a result between 0 and 1, where 0 means no pixel of correlation and 1 means a perfect overlapping. This operation is repeated for all the classes predicted by the network and then a mean is produced. This number can be used to compare easily all algorithms for object detection like the ones used during this thesis work. The other parameter is the simple evaluation of the processed videos from the different networks in post-processing (for mask R-CNN, YOLOv2 and YOLOv3) or in real-time (for TinyYOLOv2).

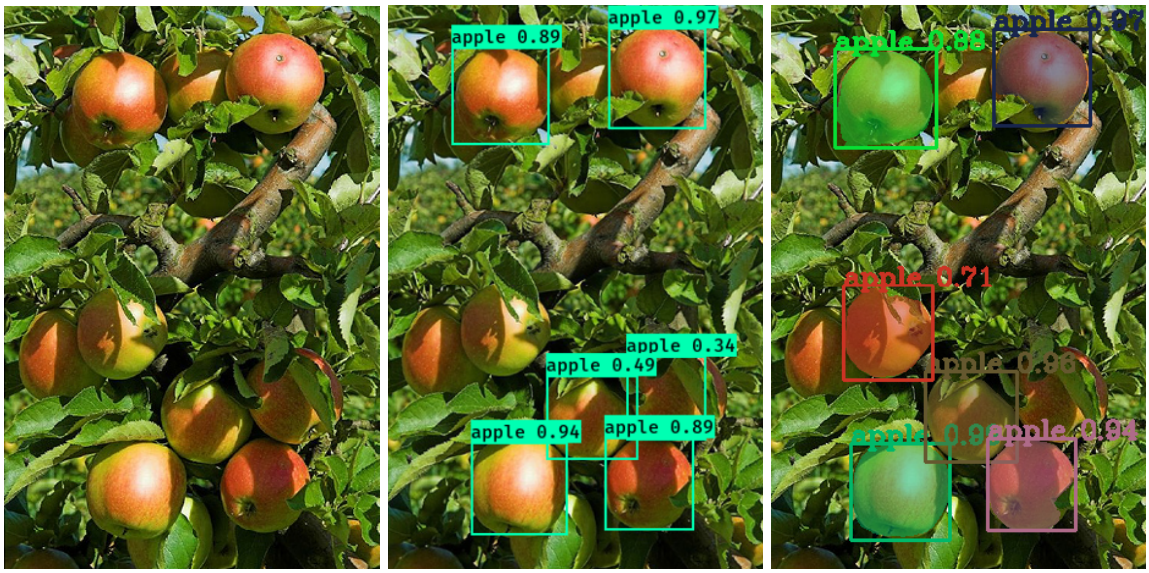
3.5 Results The mAP results are showed in the table below. It can be easily seen that for every network tested there is an enhancing of the values that determines a general better performance of the different models.

Architecture	Dataset		
	COCO	Apple	%
Tiny Y.O.L.O.v2	31.47%	49.96%	+58.75%
Y.O.L.O.v2	60%	76.88%	+28.13%
Y.O.L.O.v3	63.2%	70.76%	+11.96%
Mask R-CNN	68.8%	77.65%	+12.86%



(A) Tiny YOLOv2 COCO (B) Tiny YOLOv2 APPLE (C) YOLOv2 COCO

FIGURE 1: YOLO outputs.



(A) Original image (B) YOLOv3 APPLE output (C) Mask R-CNN APPLE output

FIGURE 2: Same images processed over two different networks.