EC601 Mini Project 2 Report

Chenxi Li

State of the Art of Machine Learning Models to Detect an Segment Objects

Introduction

This reports summarizes my research into Image Segmentation using Mask R-CNN, after reading multiple blog posts. Due to overlaps in materials, I focused on two blogs, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms" by Ajay Aresanipalai and "State of the art deep learning: an introduction to Mask R-CNN." by Rohith Gandhi

Mask R-CNN is a very popular network to separate an object in the image form its background. Many deep learning networks can easily draw a bounding box around an object, while segmenting an object requires further processing of the detected object.

Summary of references

There are steps to segmenting an object. First, we just need to find a bounding box for each detected object. Models such as Faster R-CNN can do already do this. Besides, algorithms such as YOLO can even easily do this in real time. The second step, however, is tricky; for each object surrounded by a bounding box, we need to generate a segmentation mask based on the pixels. For this part, we need a separate semantic segmentation model, according to Ajay's blog, "State of the art deep learning: an introduction to Mask R-CNN." Despite the need of a separate model, the segmentation part is easy, as Ajay points out, because the model only has to predict if each pixel is or is not part of the object.

So let's focus on the deletion algorithm for now. The blog by Rohith explains a few different methods that achieves object detection and bounding box prediction. The fundamental difference between R-CNN and YOLO lies on their ways to split an image into regions. R-CNN selects 2000 regions using selective search algorithm. Fast R-CNN puts the image through a feature map before identifying region proposals. Faster R-CNN uses a separate "region proposal network" to identicy region proposals using anchor boxes. This method is faster than R-CNN and Fast R-CNN, which are based on selective-search, but it's more expensive to develop as you need to train a separate network.

Analysis of results

In terms of performance of object detection and bounding box predixtion, YOLO algorithm is unparalleled among all methods discussed before. As paper "YOLOv3: An Incremental Improvement" by Joseph Redmon and Ali Farhadi explains, the model is trained to do classification and bounding box at the same time. In terms of precision of prediction, YOLOv3 by far has the highest COCO AP (33mAP), meaning YOLOv3 can draw the best bounding box by average. In our image segmentation problem, the burden of computation does not lie on segmentation; in Ajay's blog, it states that pixel level instance segmentation is quite achievable, for the fact that this can be run on device as cheap as Raspberry Pi.

Recommendations

The choice of model for image segmentation largely depends on available dataset for training; YOLOv3 is fast, but training its model requires sophisticated labelling and a huge training set, which can be very expensive. If you need to build a model from scratch, R-CNN is the easiest way to get started with.

Conclusions

While there are many different approaches to detecting and segment objects from images, developers has to take real life factor such as budget for hardware, available time for training into factor in order to decide which method is the best.

State of the Art of Un-supervised Learning and Where It Is currently Being Used

Introduction

Un-supervised Learning means building a machine learning model without manually labelling training data. While complex ML problems such as image classification always requires labelling, un-supervised learning gets commonly used on less complex data such as 2D coordinates.

K-means clustering is a very popular method for un-supervised learning. It is very suitable for clustering analysis. It used Euclidean distances to separate data into groups.

Summary of references

“Understanding K-means Clustering in Machine Learning” by Dr. Michael J. Garbade gives us a glimpse on how it works. It tells us that K-means clustering starts with finding centroids of the cluster, and then runs iterative optimization to find perfect positions of centroids.

Analysis of results

For a fixed k(number of clusters) and d(number of dimensions), the complexity for K-means clustering is $O(n^{dk+1})$, where n is the number of data points. Using brute force is inevitable when finding the minimum inertia $\sum_{i=0}^{n} min_{\mu_j \in C}(||x_i - \mu_j||^2)$. However, a variant of K-means, called mini batch K-means, offers more scalability . In this approach, a number of sample data are picked, then k centroids are calculated for those sample data points. Then another batch of sample will be picked again and the centroids will be updated. According to *scikit-learn* website, this approach is only slightly worse than standard K-means, but it saves huge amount of computation time.

Recommendations

Un-supervised learning is useful when we need to find pattern inside randomly distributed data, for example, when we need to separate a country’s population into different income groups. Also, without human labeling, the data must be reliable when developer decides to use un-supervised learning, and the larger the data set, the more creditable the result. Simple binary classification can be solved with linear regression. For complex problems, such as classifying multi-demential data into multiple categories, K-means clustering can come in quite handy. Mini batch K-means should be considered when data set is too large for hardware to compute.

Conclusions

Sometimes certain Machine Learning problems can only be solved with un-supervised learning, such as analyzing arbitrary data. K-means clustering can be used when the size of test se is not overwhelmingly large. Hardware constraints should also be kept in mind in order to decide wether to go with the mini batch approach.