Bounding Box Quality Assurance by Clustering LabelMe Annotation

Shuoran Zhang
San Jose State University

Abstract

Currently, computer vision is working to enable the numerous exciting applications ranging from safe autonomous driving, to accurate face recognition, to automatic reading of radiology images. Convolutional neural network (CNN) is an useful model to process the image data. As a result, the object bounding boxes are generated to represent the objects. However, the bounding boxes may not be very accurate, and the inaccurate data of bounding boxes may cause very serious consequences. I want to apply the quality assurance and machine learning methods to test the bounding boxes.

Approach

The goal of my personal project is to quality assurance the Object Bounding Box by applying the K mean clustering model. The quality assurance data are from an annotation tool called LabelMe. The QA testers will generate many annotations of objects, and I will use K-mean to clustering the annotation from QA, and generate the best QA object annotation, and compare with the existing bounding box. We will say the bounding box is ‘Pass’ if the coordinate is very similar to the clustered QA bounding box data, and the bounding box is ‘Failed’ if there are certain differences between the object bounding box and clustering QA bounding box.

Dataset

- QA annotation data from annotation tool, called LabelMe.
- Object Annotation data set is formatted in XML
- Data is made by the quality assurance testers.
- Data describe the coordinate and name of the objects.

Data Preprocessing

- Install LabelMe in my AWS ec2.
  http://ec2-54-218-178-103.us-west-2.compute.amazonaws.com/LabelMeAnnotationTool/tool.html?collection=LabelMe&mode=f&folder=example_folder&image=img2.jpg
- Annotate the objects in the image.
- Execute script to convert the LabelMe's XML to CSV.
- Convert the four points (left top, left bottom, right top, right bottom) into four features, which are center X, center Y, width, height of the box.
- Applying K-means clustering.
- Generate the XML file with the best QA annotation.

Solution Implementation

- Data processing, convert the XML coordinate to the feature arrays, which are the X, Y, of center point, and width, height of the bounding boxes.
- Initial K = 1; K++; K<=range, for each K, fit the four-feature data into the K-mean with minibatch. The range is equal to the # of object if # of object less than 10. Else then set the range to 10.
- Calculate the cost function for each K by Elbow analysis. Find the optimized K.
- Use the optimized K to calculate the new center point of each clusters. The optimized K represents the amount of objects in the image.
- Generate the new XML by the new center points, and then upload to LabelMe to view.
- Future implement: replace the human-force QA bounding boxes with the deep learning generated bounding box. This is the way of automate the QA processing.

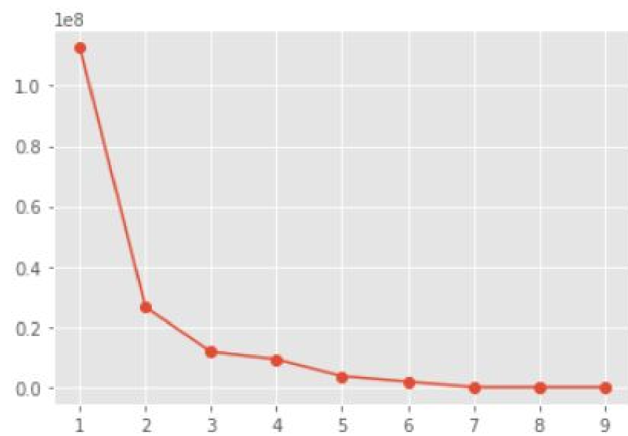
Evaluations

Elbow Analysis: The Elbow method is a method of interpretation and validation of consistency within cluster analysis designed to help finding the appropriate number of clusters in a dataset. This method looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. If one plots the percentage of variance explained by the clusters against the number of clusters the first clusters will add much information (explain a lot of variance), but at some point the marginal gain in explained variance will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion"

Given a set of observations (x1, x2, …, xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n) sets S = {S1, S2, …, Sk} so as to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the K center). Elbow Analysis objective is to find:

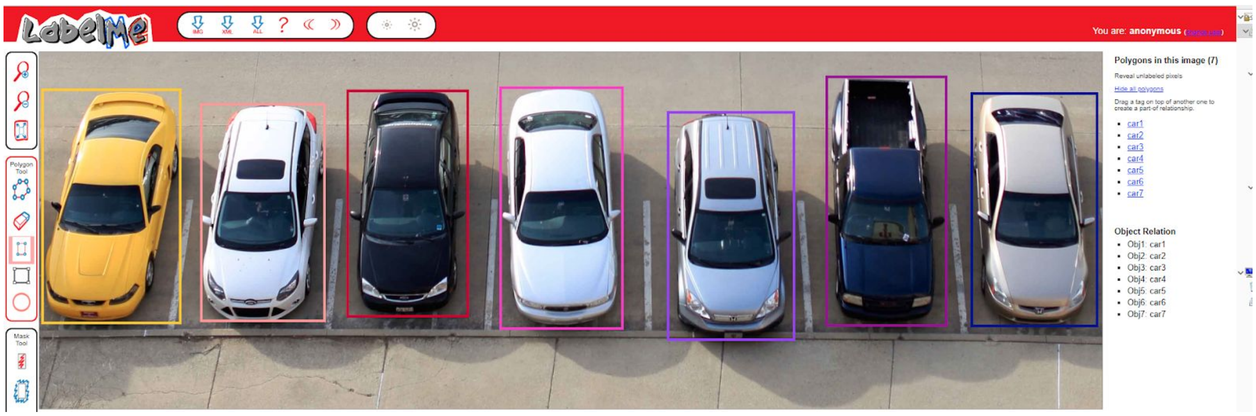
$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

```
1  11.2640174727
2  0.235812984896
3  0.440443673729
4  0.791078302668
5  0.391537918783
6  0.487289363317
7  1.01361327165e-08
8  0.937430767721
9  0.955150491794
Found 7 objects
```



The elbow diagram shows that the actual number of clusters chosen is 7.

Generate XML and view result in LabelMe



Impact

- Combine machine learning model and human-force to test the bounding box made by artificial intelligence (e.g. cnn)
- In the future, we can replace the human-force quality testing by other independent AI models, to achieve the automated QA testing.

References

[1] https://arxiv.org/pdf/1712.08470.pdf
[2] https://www.youtube.com/watch?v=GSwYGkTfOKk
[3] http://labelme.csail.mit.edu/Release3.0/
[4] http://www.awesomestats.in/python-cluster-validation/