Progress Report – Pixel Dawgs: Image Classifier

11/3/2015

Abstract

The main idea behind this study is to create an image classification algorithm that will read the image and produce several tags. These tags will be generated using image-processing tools like SLIC (Simple Linear Iterative Clustering), SOBEL filtering, and DBSCAN (Density Based Spatial Clustering of Applications with Noise). Initially ~4000 images will be used to train the code using machine learning algorithm. Later the classifier will tag the images by itself. Each tag will then represent a certain layer of the image. Out of all the tags created top 5 tags will be used to define the image with % accuracy.

Manish Borse, Nick Rosetti, Satyanarayana Reddy, Shreya Gupta

1. Introduction:

This study performed to design an image classifier using image processing tools, big data and data science tools. The main idea was to design a tool that will read any image and generate tags defining the objects present in the image. To do so we started with the open source data provided by Yahoo Flickr as the training set.

2. Data collection and analysis:

The training set data provided by Flickr's server was required to be processed and reformed to use. The training set contains a text file, which contains automatically generated tags along with an associated image id. There is another set of text files, which contain an image id and an associated URL. Each team member ran a script to parse the text containing image URLs and downloaded a total of ~300,000 images. A machine-learning algorithm automatically tags each image.

Later for post processing each image was saved with its associated ID. Further the text file containing the tags is then split into three separate 5GB files so that a dictionary can be used to find tags for a particular image. These tags are saved into a separate text file that contains image IDs which are used for processing later. Using tags as an index reformats the text file, which contains image IDs and tags. This means that each tag has an associated list of images. The top tags are calculated by rank and a selection of the top 0.5% is used to seed general tags. Each RGB image is converted to Hue Saturation Value space. A histogram for each image within a tag is calculated and a composite histogram is computed. This operation is completed for each tag. The composite histogram is written to a file by tag. This histogram is loaded back into memory to use to tag images.

In order to tag images, a list of valid images is loaded into memory. Histogram back projection is used to "back project" an image onto a histogram. This produces a resulting image that contains a per pixel representation of the probability that each pixel belongs to the tagged composite histogram. The current output of this algorithm is a 0 to 1 probability that the image is tagged correctly. A probability threshold will be used to select "valid" tags.

3. Current status and progress since last presentation:

We have successfully applied SLIC (simple linear iterative clustering) algorithm to break the image in to super pixels. Fig 1(a) shows the segmentation of super pixels. Here we are also using Gaussian filtering to reduce the noise before the super pixels are created. This gives a far better distribution of the super pixel over the image. Further SOBEL filtering operator is used for edge response. The SOBEL filtering Operator is a discrete differentiation operator. It computes an approximation of the gradient of an image intensity function [1]. Furthermore a four dimensional vector is created which represents a super pixel. The four dimensional vector represents the median color of a super pixel combined with the average edge response. Each super pixel in the image is represented by a vector and all the vectors a combined together to form one super vector representing the whole image.

Later DBSCAN solver is used to separate out the vectors representing same color, and those are collected in one cluster as shown in fig 1(b). But if the same color vectors have significantly different edge response then they are not a part of same cluster. These clusters are now used to

represent each layer of the image as shown in fig 1(c). Each layer will represent a tag. For simplification purposes while training the code the color scheme of the layers is changed. Now each layer is represented as the average color of the corresponding cluster. This will help the programmer to train the code more efficiently. Fig 1(d) shows simplified color scheme.

Future work includes

- Finalizing the segmentations accuracy

- Manually tagging 4000 images

- Create a feature vector for each tagged layer in every image

- Create list for feature vectors for every tag

- Use a supervised learning method such as SVM (Support Vector Machine)
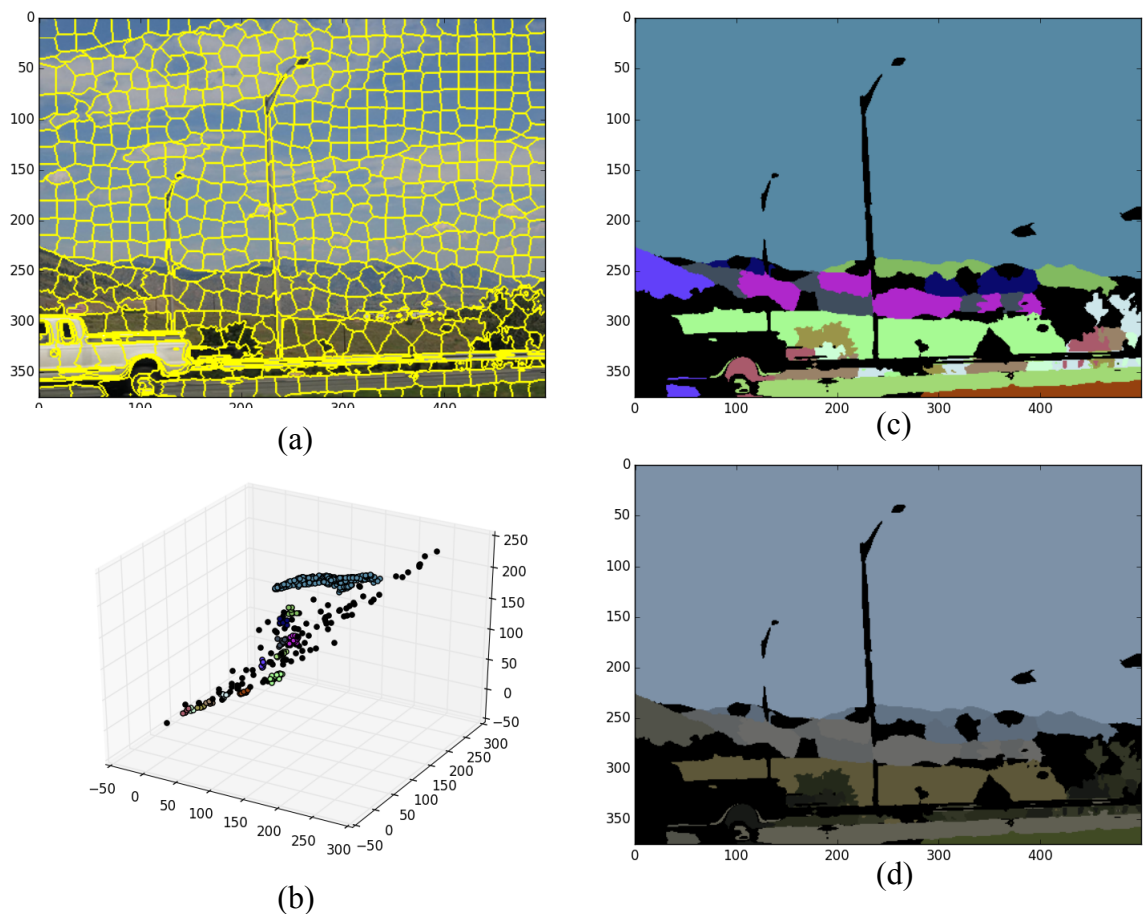

(a)

(b)

(c)

(d)

Figure 1. Stages of image processing (a) segmentation and creating super pixels, (b) cluster of vectors representing each layer, (c) image separated in different layers, (d) layers represented as the average color of the layer.

References

[1] Sobel Derivatives — OpenCV 2.4.12.0 Documentation. N.p., n.d. Web. 04 Nov. 2015.