Pixel Dawgs: Image Classifier

Manish R. Borse, Nick Rosetti, Veera Karri, and Shreya Gupta
Date: 12/9/2015

***Abstract*— The main objective behind this project is to make a tool that will classify the images based on the visual content. This study was performed using Python, Java Script and PHP. Image classification will be preformed based on image segmentation in various different layers. Each layer holds a word value as its tag. Image processing tools like SLIC (simple linear iterative clustering), DBSCAN (density based spatial clustering of applications with noise), and Sobel filtering are used for segmenting the image in different layers. Layers that are in the same cluster are associated with same tag. Few thousand images are used as the training set. SVM (Support Vector Machine) is the most common machine learning algorithm used for classification and hence it is used to predict the tags for the test data set. Finally the predicted result will show five tags based on the visual content of the image. These five tags are short-listed based on the layer the covers the maximum are of the image. This tools hold a decent accuracy.**

***Index Terms* — Image Classification, Image processing, Machine Learning, Big Data, Data Science, Python, RDBMS.**

I. INTRODUCTION

THIS study is performed as a class project for the Big data and data science course. The goal of this project is use tools of big data and data science to design an image classifier. Classification of the images is done based on the visual content of the particular image. Data was collected form open source data that was provide by Yahoo – Flickr. Further the available data was process in a way that can be used. The open source data contains over one million images, but due to lack of computational power around 300,000 images were used in this study.

Image processing was performed in the order of, first Sobel-filtering algorithm was used for edge response and edge detection in the image later SLIC algorithm was used for fragmenting the image in super pixels and then finally DBSCAN for clustering together all the super pixels with same color.

Around two thousand images are used as training set for the machine-learning segment. The images in the training set were tagged manually since the initial tags were not associated with the segmented layers. To do this a Graphical User Interface was designed. Test data made up of 200 – 400 images is used to predict the results. Results show a decent accuracy.

This study is a class project in Big Data and Data Science (CSE 4990/6990) course. Date: December 9, 2015. Team 5 – Pixel Dawgs, performed this study.

Manish Borse with the Department of Mechanical Engineering, Mississippi State University, Mississippi State, MS, 39762 USA. (e-mail: mrb393@msstate.edu).

Nick Rosetti with the Department of Computer Science, Mississippi State University, Mississippi State, MS, 39762 USA. (e-mail: ncr55@msstate.edu).

Veera Karri with the Department of Computer Science, Mississippi State University, Mississippi State, MS, 39762 USA. (e-mail: svk38@msstate.edu).

Shreya Gupta with the Department of Computer Science, Mississippi State University, Mississippi State, MS, 39762 USA. (e-mail: sg1312@msstate.edu).

II. DATA ACQUISITION AND DATA ENGINEERING

Open source raw data is provided by Yahoo-Flickr. Data set includes two types of text files.

1) *Image URL (10 files, 5GB each)* – These set of files contains all the image ID's and the associated URL to download the image, among other attributes.

2) *Auto-tags file (14GB – after extraction)* – This file contains all the image IDs and associated (auto-generated) tags, which were created by Flickr's image processing tool.

Using the *Image URL file* ~300,000 images were downloaded. These images were later used for image processing. Due to computational limitation, subset of these images was used for segmentation. Further the tags for these images need to be associated with the image ID. This involved searching for the image ID through the *Auto-tags file*. Since the tags file was large (14GB) for traditional processing, it was divided into two different files. Hereby each image ID was associated with several different tags. For post processing and further using data in the correct format, index column was changed from image ID to tags. Now each tag is associated with different image IDs.

III. IMAGE PROCESSING TOOLS

Three different image-processing algorithms were used for this project. This includes, in the order of application, Sobel Filtering, SLIC and

A. *Sobel-Filtering*

Irwin Sobel and Gray Fledman from Stanford Artificial intelligence Laboratory came up with "Isotropic 3×3 image Gradient Operator". It is often referred as Sobel-Fledman operator [1]. This algorithm basically creates an image focusing on the edges.

Sobel operator performs a 2D spatial gradient measurement on an image this results in highlighting the regions of high spatial frequency that correspond to edges. It is used to find the approximate absolute gradient magnitude at each point the image. The formulation of this operator is given as [2],

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A$$

Here A is the source image; G_x and G_y are two images, which at point contain the horizontal and vertical derivative approximation respectively. The * operator denotes 2–Dimensional signal processing convolution operation. Sobel kernels can be decomposed as the products of an averaging and a differentiation kernel and G_x can be written as,

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

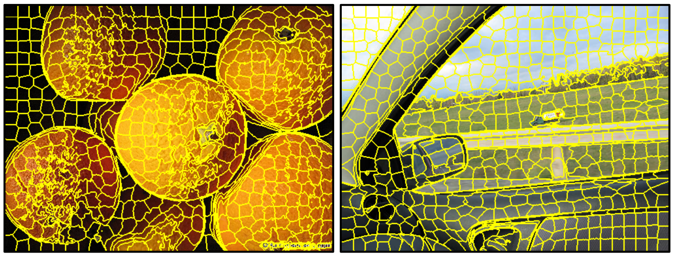
The gradient magnitude and direction is calculated using equation (1) and (2),

Fig 1. Left hand side image is the raw image and the image on the right hand side is the gray scale image showing the edge response [2].

B. *Simple Linear Iterative Clustering (SLIC)*

Super-pixels are playing a important role in the current state-of-art in computer vision applications. There are several different algorithms that generate compact super-pixels with low computational overhead. Image and visual representation lab (IVRL), came up with an algorithm SLIC that clusters pixels in the combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform super-pixels. IVRL claims that their experiments show that this approach produces super-pixels at a lower computational cost while achieving a segmentation quality equal to or greater than four state-of-the-art methods, as measured by boundary recall and under-segmentation error. Considering this, SLIC super-pixel is used in this project. Fig 1 (a) shows the raw image that was processed using the Sobel filtering for edge response and then SLIC was implemented for super-pixel generation. Fig 1 (b) shows the super pixels that are generated on the image based on the five-dimensional color and image plane space.

(a) Raw image



(b) Super-pixels generated on the picture.

Fig 2. This figure shows the super-pixels created on the image. Each box with yellow borders in (b) is considered as a super-pixel.

C. Density Based Spatial Clustering of Application Noise

Clustering algorithms play a vital role for the task of class identification in spatial database. However when applying to a large spatial database there are some requirements for the clustering algorithm; minimum input parameters, discovery of clusters for arbitrary shapes and good efficiency on large database. DBSCAN provides good efficiency on large data set and given an edge response it can create a cluster for an arbitrary shape. DBSCAN requires only one input parameter and it supports the user in determining an appropriate value for it. Literature review shows that DBSCAN is significantly more effective in discovering clus- ters of arbitrary shape than the well-known algorithm CLAR- ANS, and that (2) DBSCAN outperforms CLARANS by a factor of more than 100 in terms of efficiency.

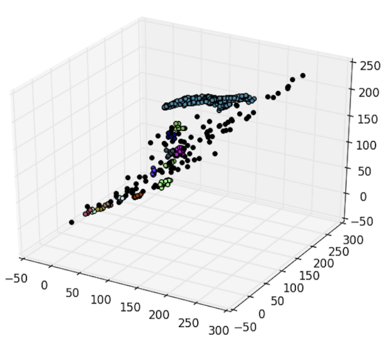


Fig 3. Graph of DBSCAN clustering using Lab color channel

For this project we are providing Lab color space and edge responce from the Sobel filtering algorithm as the input parameter for the DBSCAN. Lab color space is based on one channel for Luminance (L) and two color channels (a and b). The two color channels a and b can represent any two colors depending upon the shade of the color and L defines the lightness of the color. L = 0 is the darkest shade of the color generated by and b. L = 100 is the lightest shade of the same.

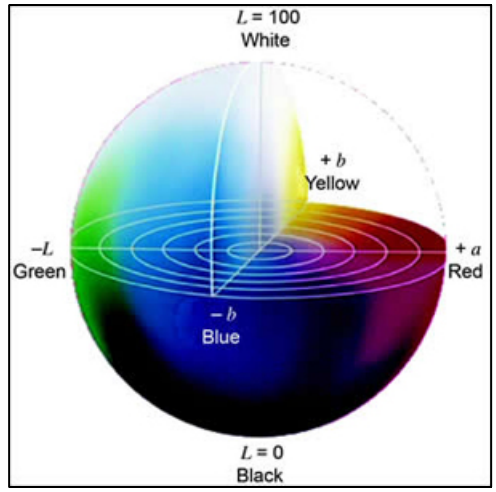


Fig 4. LAB color space, featuring L – Luminus (lighteness) and, a and b represent any two colors.

Later after the using DBSCAN clustering algorithm the super pixels with same color value were clustered to form one layer with median color value. Fig 5 shows the layered images.

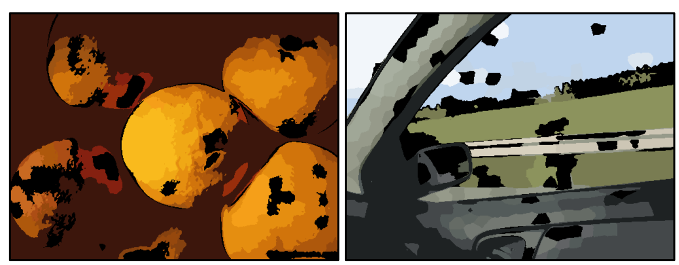


Fig 5. Segemented images with median color value showing diffrent layers of the visual content.

IV. Machine Learning

A. Graphical User Interface

Since now all images a processed and are segmented into separate layers now these layers needed to be tagged. The main difficulty was the auto-tags that are generated by Flickr's image classifier are associated to the entire image. So to overcome this problem a GUI was created using PHP and JavaScript.

GUI was used to manually tag ~3000 images that was used as the training set for the machine learning algorithm. Also from the auto-tags file that was mentioned in Section II was used to find top 10 tags that are commonly used in the dataset. Another 10 tags were selected based on commonly seen features in an image such as sky, trees, clouds, wall, people, etc.

Fig 6 shows the GUI. Each layer in the image holds a co-ordinate value. Therefore co-ordinates on each layer are selected and assigned to the considerable tags.

Fig 6. Graphical user interface, used for manual tagging of images

Further this data, inclusive of the spatial co-ordinates and associated tags, are stored in a excel file. This data is further used to generate feature vector. The feature vector will be provided as the input for the Support Vector Management (SVM), the machine-learning algorithm.

B. Support Vector Management

Machine learning is a sub set of Artificial Intelligence. It focuses on computer programs that can teach themselves; basically computer code is trained on a dataset and then it can predict results when exposed to a new data value. For this project a supervised machine learning technique is used. SVM is a supervised machine-learning algorithm, which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs [5]. Few advantages of SVM are it is highly effective on high dimension spaces, it uses a subset of input to create decision vectors and therefore it is very memory efficient.

The code is initially trained based on the feature vectors, with six dimensions, that were created using the image processing tools and manually tagging the images. Further completely new sets of images were used as test data. Here the code reads the new image and based on the visual content it predicts the possible tags. Results are discussed in the result section.

V. RESULTS

Results show that the image classifier predicts the tags based on the visual content Fig 7 shows the image and the top five predicted tags. As shown in the figure it predicts probability of people occurring in the image is 0.60. Classifier also predicts P = 0.40 for trees and P = 0.27 for tress and grass respectively, that is because of the green color present in the image. Wall is P = 0.32, considering this is a indoor image its is a good agreement. Similarly Fig 8 shows only three tags because of the threshold value given to the probability variable. The classifier only shows the tags that have probability of higher than 0.2. Fig 8 shows accurate probability for P (sky) = 0.71, P (trees) = 0.42 and, P (water) = 0.23.

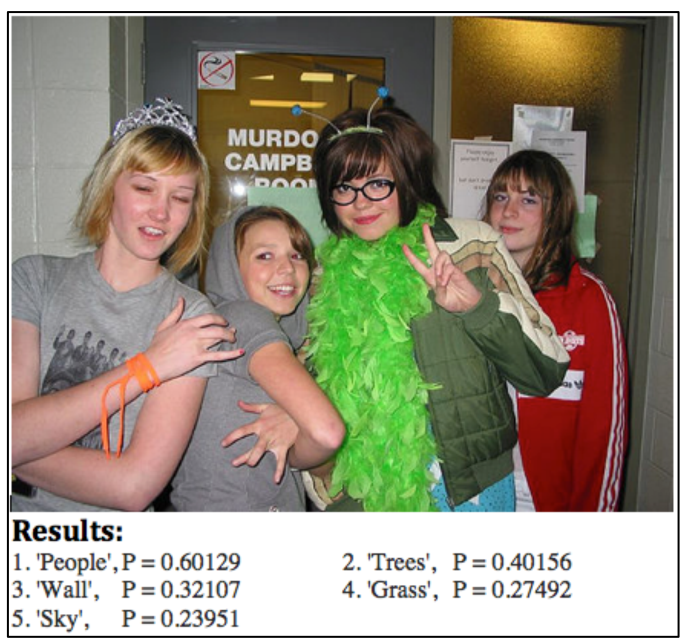


Fig 7. Example image 1 showing the probaby of tags occuring in the image.

Fig 8. Example image 2 showing the probaty of tags occuring in the image.

The code was tested on the test data, which is different, then the training set. Fig 9 shows the bar graph of four probability buckets each bucket is of value P = 0.25 and four tags, sky, tree, water and, people were chose for the study. Data was tested for the occurrence for the tags in an image.

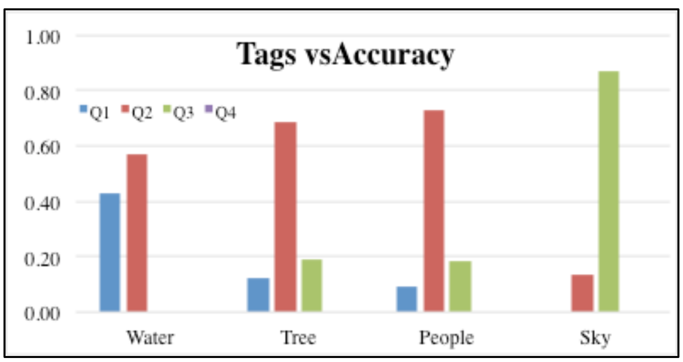


Fig 9. Graph showing Tags vs Accuracy

VI. Conclusion and Future Work

In conclusion it can be said that the goal of the project was successfully achieved. The image classifier works with acceptable amount of probability of predicting a particular tag. Future work includes investigations of different factors to form the basis for a feature vector that classifies the images with higher accuracy and draw conclusions about relations between various entities within the images.

References

[1] R. Fisher, S. Perkins, and A. Walker, “Sobel Edge Detector,” *Feature Detectors* -, 2003. [Online]. Available at: http://homepages.inf.ed.ac.uk/rbf/hipr2/sobel.htm. [Accessed: 2015].

[2] “Sobel Operator,” *Wikipedia*. [Online]. Available at: https://en.wikipedia.org/wiki/sobel_operator#cite_note-2. [Accessed: Sep-2015].

[3] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, p. 2274 - 2282, May 2012.

[4] “Color Space Color Model,” - *Color Hex Color Codes*, Jan-2012. [Online]. Available at: https://www.colorcodehex.com/color-model.html. [Accessed: 2015].

[5] “1.4. Support Vector Machines,” *1.4. Support Vector Machines — scikit-learn 0.17 documentation*. [Online]. Available at: http://scikit-learn.org/stable/modules/svm.html. [Accessed: May-2015].

[6]