Towards Better Pixabay Tags

Willie Maddox

November 1, 2017

This document serves as the proposal for the final Capstone project for the Machine Learning Engineer Nanodegree offered through Udacity.

1 Domain Background

Pixabay is a website where photographers can publish and share copyright free images and videos. Since all the contents are released under the CC0 license, they are safe to use without having to ask permission or give credit to the original artist. When a user submits a new image it must first be reviewed by the Pixabay admins. They look at:

- Copyright and Duplicates
- Noise and JPEG Compression Artifacts
- Focus and Blurring
- Lighting and Colors
- Tilted and Crooked Images
- Image Dimensions
- Image Manipulations
- Image Hygiene and Composition

If the image satisfies the above categories then, most likely, it will be approved [1]. This is probably the primary reason Pixabay is so popular among photographers and artists alike; the overall quality of images in the database is professional grade.

2 Problem Statement

Along with the image upload, the user must also provide at least 3 tags describing the content of the image. The average number of tags per image is around 10. Tags make the image easily searchable by other users. Pixabay provides a tagging tutorial on their website but in general the tags are not required to meet the same level of quality standards that are placed on a newly uploaded image [2]. Nor can they really be enforced. The metric for measuring the quality of an image is

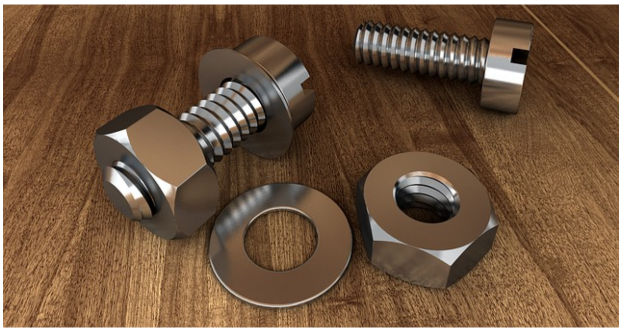

Figure 1: *Nuts n Bolts n Boots n Pants*

well defined. All images must have at least 1920 along the long dimension. The image should be sharp and in focus. Avoid embedded timestamps. These are all acceptable forms of objective measurement. Tags, on the other hand, represent a persons description or interpretation of what is contained in an image and as such they (the tags) are difficult to use as a source of measurement. For example, Fig.1 looks like *nuts and bolts* to me, but to someone else it might be *hardware* or maybe even *wood*. Which tags are *correct*? Well all of them. The final decision on which labels to use for tagging an image will be up to the owner of the image. Hence, it is probably not a good idea to use tags as a measure of whether or not an image should or should not be approved.

Assuming that the user is the original author (or photographer) of the image, then coming up with 3 relevant tags should be trivial. Authors have first hand knowledge of their own images and can generally be trusted to tag the image correctly. After all, an image with mislabeled tags is an image that no one will ever find. And since so much effort is required on the part of the author to get an image approved in the first place, it seems highly unlikely that anyone would ever mislabel one of their own images on purpose. And yet many of the images on pixabay have tags that are incorrect or at least appear to be.

major tag problem areas:

1. Misspelled tags. siberian husky not sibirian husky.
2. Irrelevant tags.
3. Different translations.
4. Incorrect tags. papillon returns lots of butterflies (no papillon tag either)

Show examples of 2-3 images that have bogus tags.

3 Datasets and Inputs

For this study we will use a custom Pixabay dataset as our primary dataset. By custom we mean that we will only choose images with tags related to the 1000 classes in Imagenet. We will also supplement our primary dataset with a secondary dataset consisting of both single-label and multi-label image datasets. For single-label we will use Imagenet CLS-LOC[3] dataset and for multi-label we will use the Imagenet DET[3], MSCOCO[4] and NUS-WIDE[5] datasets. These secondary datasets are very organized; they come with verified ground truth and can be easily downloaded and extracted for immediate use. Data from Pixabay does not come prepackaged. You must submit multiple search queries to build up your own database. Fortunately they provide an API for registered users[6]. The Pixabay API is well documented and it's usage is relatively straight forward. At the minimum you need to pass it an API key for authentication and a query string of labels to search. For example, to retrieve web format photos about "yellow flowers", the query string q needs to be URL encoded[1]. `https://pixabay.com/api/?key=1234567-a1b2c3d4e5f6g7h8i9j0k1l2m&q=yellow+flowers&image_type=photo`. The response for this request is a JSON encoded data structure containing metadata for a list of images.

Snippet 1: *Pixabay API JSON response*


```
{
"total": 19177,
"totalHits": 500,
"hits": [
  {
    "id":2895728,
    "pageURL": ...,
    "type":"photo",
    "tags":"flower, pink, yellow",
    "previewURL": ...,
    "previewWidth":150,
    "previewHeight":112,
    "webformatURL": ...,
    "webformatWidth":640,
    "webformatHeight":480,
    "imageWidth":4608,
    "imageHeight":3456,
    "views":56,
    "downloads":25,
    "favorites":1,
    "likes":4,
    "comments":5,
    "user_id":5394567,
    "user":"GeorgeB2",
    "userImageURL": ...,
  },
  {
    "id": 195893,
    "tags": "blossom, bloom, flower",
    "webformatURL": ...,
    "..."
  },
  "..."
]
}
```


A sample of the API response is shown in Snippet 1. There are three top level parameters: The `"total"` number of images in the Pixabay database with tags matching the query, the maximum `"totalHits"` that can be retrieved with the present query, and the actual `"hits"` which are a list of python dictionaries, each containing metadata about a specific image in the database. For our purposes, we only need a subset of this metadata: A url to fetch the image, the set of labels that describe the image, and a mapping to help us keep track of which set of labels goes with each image. Each "hit" contains a url for a low, medium, and high resolution version of an image. We choose the medium sized image `"webformatURL"`. Since most pretrained models use images with dimensions between 200 and 300 pixels as input, this is an appropriate choice. As we can see in the code block, the `"tags"` represent the labels for the image. We will store the `"tags"` in a dictionary using the `"id"` as the key since they are unique across images. Each downloaded image will be saved using `"id"` as the base of the file name. So

[1]The key used in the url is invalid so don't expect it to work. The url is meant to illustrate the basic structure of a request.

for the example above we will have 2895728.jpg and 195893.jpg. This will make it easy to determine which image goes with which tags and vice versa.

The details (usage, directory layout, data formats, etc.) of the other datasets are available online and will not be discussed here. However, it is worth mentioning how each of the datasets will be merged together. The syntax rules for labels depend on the dataset and are generally incompatable across datasets. Some datasets capatilize proper nouns, some do not (Chihuahua vs chihuahua). Some use spaces to separate multi-word labels, others use underscores (golden retriever vs golden_retriever). Some use alternate spellings (airplane vs aeroplane). Before we can merge the above databases together into a complete database we will first need to decide on our own set of rules for labels.

4 Solution Statement

The solution is to train a model that is capable of generating (or recommending) a finite set of tags based soley on an input image. We will measure the performance of our model by comparing the top k predicted labels from our model against the known ground truth labels for each of our images. The exact value of k and the specific metric used are discussed in the following sections. We set aside a subset of our training data which will be used exclusively for testing. For each image in this testing set, the goal is to reproduce similar results for each one.

5 Benchmark Model

To benchmark the solution above, we will compare the testing set against three separate models. The first one will simply be a pretrained Imagenet model[2] (Just the base Imagenet model with 1000 classes. No fine tuning.) Since we will be using this same base model for transfer learning, we should expect similar performance classifing single-label images from the base Imagenet classes.

For the second model, we will use Clarifai's image recognition API[3]. Clarifai's image recognition systems recognize various categories, objects, and tags in images, as well as find similar images. The company's image recognition systems allow its users to find similar images in large uncategorized repositories using a combination of semantic and visual similarities. We will use the evaluation metrics below to quantify how well the Clarifai model does on our training set.

The third benchmark model, Akiwi, is a semi-automatic image tagging system able to suggest keywords for uploaded images with minimal user input[4]. Akiwi does not offer a public API, instead you must drag and drop images in one at a time. We will most likely not run the entire testing set through this benchmark, but rather use it to study edge cases and outliers. It will be interesting to see how well our model compares to these state-of-the-art systems.

[2] https://keras.io/applications/

[3] https://www.clarifai.com/

[4] http://www.akiwi.eu/

6 Evaluation Metrics

There are two groups of evaluation metrics that apply to multi-label classification: *example-based* metrics and *label-based* metrics. In regard to classification problems, there are a good many evaluation metrics to choose from. However, there are two in particular that are commonly used in multi-label classification problems. The first is

The second is the Ja

$$\mathrm{IOU} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y^i \wedge \hat{y}|}{|y^i \vee \hat{y}|}, \tag{1}$$

7 Project Design

Can we improve the classification by adding more types of dogs, cats, etc.

Search for images with Imagenet Labels

There are restrictions that make getting the exact data you want a bit tricky. For one, when you submit a query you get back a

We will use WordNet to create consistent labels across the datasets.

Transfer learning on imagenet. Add k classes where k is the number of classes in pixabay images that are not classified by Imagenet.

The training, validation, and testing datasets will be drawn from the complete dataset.

For this project, I will use a pretrained model of Imagenet

The plan is to use the Imagenet model as a fixed feature extractor

1. How many images per category are there in Imagenet. (between 732 and 1300 per synset)
2. How many nouns (or physical entities) are there in WordNet.
3. How many hypernyms classes are there in Imagenet.
4. How many hyponyms per hypernym are there in Imagenet.
5. What about holonyms and meronyms. Can they be of any use with this problem?

Because the images are of such high quality on Pixabay they make great specimens for training on CNN's.

- Tokenization
- Tagging - Nouns only
- Stemming
- Lemmatization

- Lexical semantics: synonym, antonym, hypernym, hyponym, meronym, holonym
- StopWord removal

$k - fold$ cross validation

too compensate for class imbalance, We will use stratification to construct the $k - fold$ cross validation subsets[7;8].

References

[1] Simon Steinberger. Tagging Tutorial for Pixabay Images, 2014.

[2] Simon Steinberger. Photography Training and Image Quality Standards, 2012.

[3] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[4] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312, 2014.

[5] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao Zheng. NUS-WIDE: A Real-World Web Image Database from National University of Singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., July 8-10, 2009.

[6] Simon Steinberger. Pixabay API, 2014.

[7] R. Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, volume 14, page 1137–1143, 1995.

[8] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. *On the Stratification of Multi-label Data*, pages 145–158. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-23808-6. doi: 10.1007/978-3-642-23808-6_10. URL `https://doi.org/10.1007/978-3-642-23808-6_10`.