PinterNet: A Thematic Label Curation Tool for Large Image Datasets

Ruoqian Liu[1], Diana Palsetia[2], Arindam Paul[1], Reda Al-Bahrani[1], Dipendra Jha[1],
Wei-keng Liao[1], Ankit Agrawal[1] and Alok Choudhary[1]

[1]Electrical Engineering and Computer Science, Northwestern University
{rll943, arindam.paul, rav650, dipendra.jha, wkliao, ankitag, choudhar}@eecs.northwestern.edu
[2]4C Insights, INC.
diana.palsetia@4cinsights.com

***Abstract*—Recent progress in big data and computer vision with deep learning models has gained a lot of attention. Deep learning has been performed on tasks such as image classification, object detection, image segmentation, image captioning, visual question and answering, using large collections of annotated images. This calls for more curated large image datasets with clearer descriptions, cleaner contents, and diversified usability. However, the curation and labeling of such datasets can be labor-intensive. In this paper, we present PinterNet, an algorithm for automatic curation and label generation from noisy textual descriptions, and also publish a big image dataset containing over 110K images automatically labeled with their themes. Our dataset is hierarchical in nature, it has high level category information which we refer as *verticals* with fine-grained thematic labels at lower level. This advocates a new type of hierarchical theme classification problem closer to human cognition and of business value. We provide benchmark performances using deep learning models based on AlexNet architecture with different pre-training schemes for this novel task and new data.**

***Keywords*-Computer vision; Dataset; Image classification; Theme classification; Label curation**

I. Introduction

Recent years witnessed breakthroughs in computer vision with the combined advancement of big data and deep learning. Deep Convolutional Neural Networks (CNNs) are used to understand image scenes; deep Recurrent Neural Networks (RNNs) are used to model speeches and languages unfolded in time. Popular CNN architectures such as AlexNet [1], VGG [2], GoogLeNet [3] and ResNet [4], have demonstrated significant performance on large scale visual recognition tasks such as image classification [5]. Streamlining image understanding with language modeling to achieve higher cognitive intelligence has been pursued in image captioning [6], sentence-based image retrieval [7], and question answering [8].

The success of deep learning models is inseparable with the advancement of big data. A chief contributing factor behind all the developments in deep learning is the availability of large datasets that are clean, diversified and clearly labeled. While there has been increasing efforts in the society to collect, annotate, and publicize datasets to serve as training and benchmarking for various tasks, a large extent of the work is carried out manually, through crowd workers, either locally recruited or through online services like Amazon Mechanical Turk [1].

Although crowd labeling has become the standard approach, its limitations cannot be overlooked. First, the reliability of labels is largely affected by each individual's own experience, preference, capability and even fidelity. Some labeling systems try to reduce the variability by passing the same sample to many workers and "merge" different results. However, there is little universally acknowledged principle as of how to synthesize crowd-sourced results. Secondly, crowd-sourcing is expensive, requiring necessary expenses at the pipeline design, strategy deployment, software development and payment to workers.

The third drawback of crowd source labeling, and the most important one, is the fact that each label is generated with no regards to the holistic view of the entire dataset, given each labeling worker is only exposed to a small portion of the dataset. We argue that the comprehension of data in its entirety is important in producing reasonable labels. The focus of this paper is to develop an algorithm that relies on word affinity and frequency to generate image labels from noisy, easy-to-get annotations or search terms. Our algorithm, while entirely automatic, can be easily inserted into a crowd-sourcing pipeline, either before the human labeling to produce a set of reasonable candidates to reduce individual variance, or after the human labeling to merge and summarize labels.

The developed algorithm automatically collects, cleans, and eventually produces labels from *verified* tags of images from Pinterest [2]. Labels created by this process turn out to be strongly related to a set of *themes*. Thematic labels are for example, "4th of July", "father's day gift", and "summer outfit". They tend to include a conceptually coherent set of objects that could span a wide spectrum of looks and types. Such thematic labels are different from commonly seen

[1]https://www.mturk.com
[2]https://www.pinterest.com

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

object labels (e.g. different breeds of dogs in ImageNet), for that it describes a higher level of abstraction of what human perceives in images.

We present PinterNet, the label curation tool along with the image dataset, currently containing over 110K images. Images are first categorized into verticals, and then into themes. Examples of verticals include "food", "fashion", and "home decor", and themes are "4th of July", and "christmas gift ideas". The same theme may appear across multiple verticals. Detailed information about the dataset can be found on the dedicated website[3]. Such a hierarchical label structure inspires us to build a hierarchical classification system, released as the first benchmark for theme classification.

The rest of the paper proceeds as follows. In Section II, we present the algorithm PinterNet, for data curation and labeling based on frequent itemset mining. In Section III, we describe the dataset, containing over 110K images crawled from Pinterest, organized by hierarchical verticals and thematic labels. Section IV presents the hierarchical classification system based on pre-training and fine-tuning CNNs with various architectures. The infrastructure of image collection is described in Section V. A literature review of related public datasets, automatic data labeling tools, as well as related recognition tasks is given in Section VI, and Section VII discusses future works and concludes the paper.

II. Automatic Label Curation

Nowadays, the continuous volume of images being uploaded on social sites, simply outpaces the rate of annotation that can be conducted using crowd-sourced workers. Image labeling simply cannot entirely rely on manual labeling anymore. Therefore the automation of label curation becomes essential. When an image is collected from a public, mostly social website, a series of information trails with it in the form of unstructured texts, e.g. annotations, tags, and comments from different participants. For each image, there could be many information pieces describing the same content from possibly different perspectives and in noisy ways. For a set of images, all those pieces collectively form a holistic understanding of the dataset as a whole. Labels of individual images should not only depend on their own tags, but it should also take into account such a holistic understanding of the whole data.

When images are passed to generate labels one at a time, as most manual labeling systems do, the labeling person may use random phrases of his/her choice that might be filtered out eventually due to the scarcity of such phrase, hence resulting in a waste of resource. In contrast, we propose an algorithm that incorporate collective information from all images in a set, automatically filter out the infrequent tag and produce a refined set of descriptions. Such descriptions can be directly used as labels, or passed on to labeling persons for further curation.

A. Overview

The common approach for labeling is conducted bottom-up, where a single image is shown to a labeling worker at a time, and a label is generated with no regards to its proximity to other generated labels. In contrast, we follow the top-down approach, using the textual information already existing in images during collection, either the search terms used in crawling, or the comments, tags, and annotations associated with them during propagation on social sites. Such information is abundant, easy-to-get, but can be too noisy to use directly as labels. On social sites an image can be described by many users from many perspectives with different keywords; even the same concept can be written out in redundant ways. For instance, an image[4] from Pinterest, shown in Figure 1, is annotated with a series of word phrases by human users, along with the occurrence of each phrase. All of them are trying to convey a similar idea, but as a result of human variation, are largely overlapping and repetitive. It is therefore necessary to rely on data-driven methods to refine its labels, by not only looking at the frequency of words appeared in this image, but also that appeared in the whole dataset. By focusing on this image only, it is easy to generalize labels like "mother's day" and "father's day" but leave out "diy" because it doesn't appear frequent enough. However in view of its appearance in the whole dataset it would be saved.

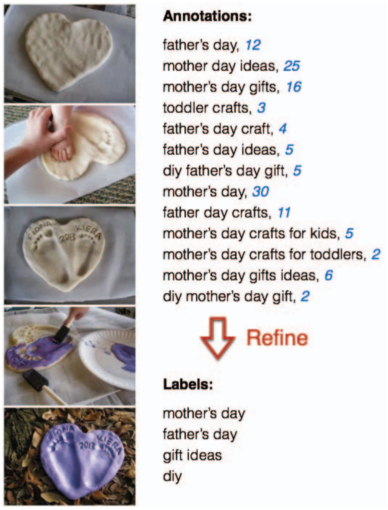

Figure 1. An example image from Pinterest. The annotations are those made by users when they "pin" an image they like. The four labels are generated automatically by frequent itemset mining algorithms relying on statistics of annotation on *all* images.

The algorithm we propose has the following benefits:

- Automated. It takes in a large set of existing noisy textual annotation segments and generates a refined set

[3] http://www.pinternet.org

[4] What's shown in Figure 1 is not five separated images but one. It is common on Pinterest to find long, collaged images to show working steps.

of keyword labels. No human supervision is required in this labeling process.

- Data-driven. The label generation is done mindfully, taking into account the word frequency, affinity, and correlations within the entire data. It produces not only single-word labels but also multi-word segments, without the assistance of hand-crafted rules in word concatenation.
- Modular. The developed tool can be used as a module, pipelined with other steps in the data acquisition and processing stage to ensure the quality of labels. In a system of manual label generation, this module can be used either in the beginning to generate candidate labels for crowd workers, and/or afterward to clean and refine human generated labels.

B. Problem definition

Suppose we have crawled a collection $\mathcal{C}$ of images using a number of search terms (details of the crawling infrastructure described in Section V). Each image $I \in \mathcal{C}$ is associated with a series of search terms used to query it, $I : \{s_1, s_2, \ldots\}$. Each search term further comprises a sequence of keywords, $s_i = \{w_1, w_2, \ldots\}$, and can have a high overlap of keyword usage with other search terms. We claim two search terms to be the same if they have the same set of keywords regardless of the word order. For example "gifts father's day" and "father's day gifts" are the same search term (this has been validated by the Pinterest search engine – querying with such two phrases returns the same set of images). In this section we develop an automatic label curation strategy that answers the following questions:

1) Given a collection of images $\mathcal{C}$ and the set of search terms $\mathcal{S}$ used to query $\mathcal{C}$, along with statistics such as the number of images acquired from each search term $s_i \in \mathcal{S}$, how can we determine a set of image labels that are meaningful, concise, and representative?
2) Given an unlabeled image $I \in \mathcal{C}$, and the associated search terms $\{s_1, s_2, \ldots\}$ used to acquire it, how can we determine what label, or set of labels, to assign to the image?

C. Association rule mining

The idea is to adapt concepts from association rule mining to the generation of labels. Frequent itemset mining is the fundamental strategy towards the discovery of association rules, in the form of $A \Rightarrow B$ which means given itemset A appears, B is likely to appear. The adapted procedure for label creation is composed of the following three steps.

Building word transaction. We start with building a many-many relationship graph between two kinds of elements, words that have ever appeared in a search term (analogous to "items" in a market-basket model), and a search term used to obtain an image (analogous to "baskets"). Each time a search term (a basket of words) is used to acquire an image, we record it as a transaction. A set of image data acquired through this searching procedure would create a transaction file. The number of transactions in the file is the same as number of images acquired. As the definition of items is now words, we change the term itemset to wordset from here on.

Word preprocessing. To create a transaction of wordsets, a series of processing steps is required to account for upper/lower cases, stop words, abbreviations, inflectional and derivational forms of words, etc. We use standard natural language processing (NLP) procedures, with four steps illustrated in Figure 2. The implementation is based on the Python Natural Language Toolkit [9].

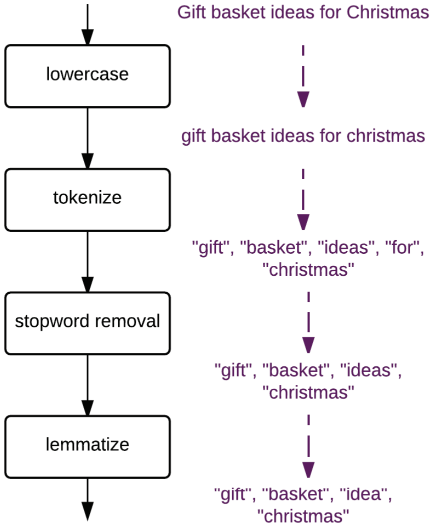
Gift basket ideas for Christmas
lowercase
gift basket ideas for christmas
tokenize
"gift", "basket", "ideas", "for", "christmas"
stopword removal
"gift", "basket", "ideas", "christmas"
lemmatize
"gift", "basket", "idea", "christmas"

Figure 2. Four steps of NLP procedure to clean a sequence of words and strip into word items. An example input and outputs at each step are shown.

Frequent itemset mining. After a clean transaction file is obtained, an algorithm finds frequent sets of items (itemsets) from examining the transactions. Let $W = \{w_1, w_2, \ldots\}$ be the collection of word items. We use the Apriori [10] algorithm, which works by assuming that a multi-item set is frequent only if all its subsets are frequent. Two threshold parameters are required, the *support* threshold τ_{supp}, and the *confidence* threshold τ_{conf}. The result of Apriori is: (1) a list of wordsets (consisting of both single words and frequently co-occurring multi-words) whose occurrence ratio is larger than τ_{supp}, and (2) a set of association rules, in the form of $P \Rightarrow Q (P \subset W^*, Q \subset W^*, P \cap Q = \emptyset)$ whose confidence measure is over τ_{conf}. W^* is the set of all unique words appeared in wordsets generated by (1). The confidence measure is defined as the ratio of the number of transactions containing both P and Q to the number of transactions containing P. When $P \Rightarrow Q$ is found in the rule set, most likely $Q \Rightarrow P$ can be found too, with a different confidence value.

D. Label curation algorithm

Given the wordset and association rule results generated by Apriori, labels are created with two branches, namely, Single-Label Curation (SLC) and Multi-Label Curation (MLC). They work in an interlaced fashion, generating respectively a single-word label set and multi-word label set. The two resulting sets of labels are merged to produce the final set of labels.

Let's denote the wordset result (the ranked list of sets of words that co-occur frequently) of Apriori as $S_0 \cup M_0$, where S_0 is the set of single words that by themselves occur in the transaction with a probability over τ_{supp}, and M_0 is the set of multi-words that satisfy the same criterion.

As of the association rule result of Apriori, we process it in the following fashion. For each rule generated in the form of $P \Rightarrow Q$, where P and Q are non-overlapping wordsets, we process each rule into $R = P \cup Q$ and keep only the unique Rs. The set of unique Rs forms M_1, the final multi-word label set. The final single-word label set, S_1, is generated by subtracting the support value of each word in M_1 from its value in S_0, and having the resulting set go through the support threshold filter again.

The entire process is denoted by an example in Figure 3. Given a transaction file (each line of transaction records a set of words being used to query one image), the Apriori algorithm finds the most frequent single wordset S_0 and multi-word set M_0 with the set parameter τ_{supp}. Another set parameter τ_{conf} filters association rules based on M_0. We curate multi-word labels M_1 by merging words appeared in rules. Then M_1 is used to update confidence values in S_0 and obtain curated single-word labels S_1.

III. PinterNet Dataset Statistics

Using the PinterNet automatic label curation tool, we collected a large set of images from Pinterest, organized them into categorized verticals, and generated theme-oriented labels. This hierarchically labeled dataset is publicized [5]. The data are extracted for a period of one year between January 2015 to January 2016, using a list of search terms to query from Pinterest API (see Section V for details of search term construction and image crawling). The dataset contains 110,828 images, with each image placed under one of 33 categories, which we call verticals, and assigned a number of themed labels. Verticals are defined by Pinterest category information [6]. The distribution of verticals is illustrated in Figure 4. Table I summarizes the label information of top 10 verticals (amounts to 72% of entire data) in this dataset. The number of classes in each vertical is determined by adjusting τ_{supp} and τ_{conf} so that it is close to about 1% of the number of images.

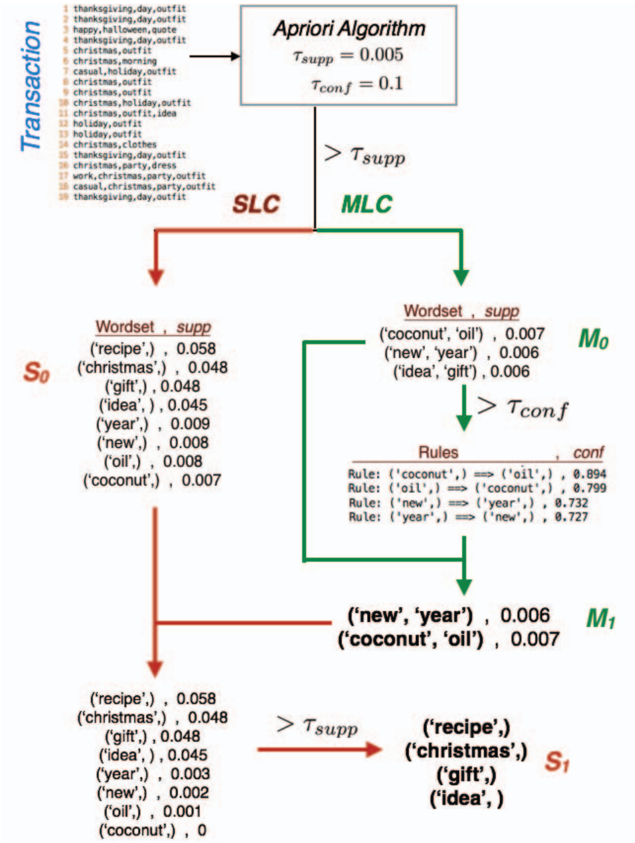

Figure 3. The procedure of the proposed label generation algorithm, illustrated with an example.

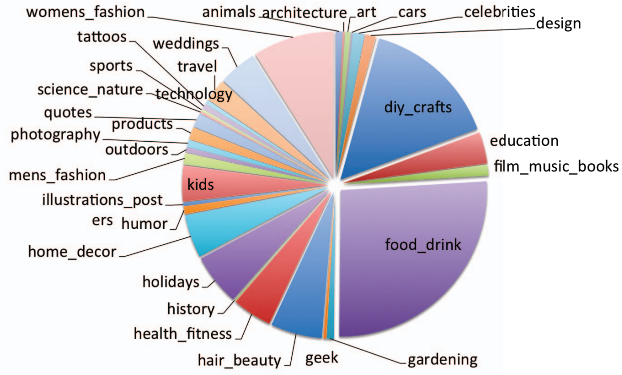

Figure 4. Vertical distribution in PinterNet.

Class labels within each vertical were generated automatically, from the search terms that were used to query those images. On inspection of the *food* vertical, we found 5,750 unique search terms used to acquire it. The distribution of a number of images acquired by each of the search terms is

[5] http://www.pinternet.org

[6] Pinterest categories are explained in the section "Get ideas from categories" at: https://help.pinterest.com/en/guide/discovering-things

Table I
LABEL INFORMATION OF EACH VERTICAL (TOP 10) IN PINTERNET.

Vertical name	#Images	#Classes
Food and drink	24762	287
DIY crafts	14223	126
Women's fashion	8438	87
Holidays	5441	52
Hair beauty	5361	53
Kids	5201	77
Home decor	4450	28
Health and fitness	4141	30
Weddings	4004	43
Education	3292	44

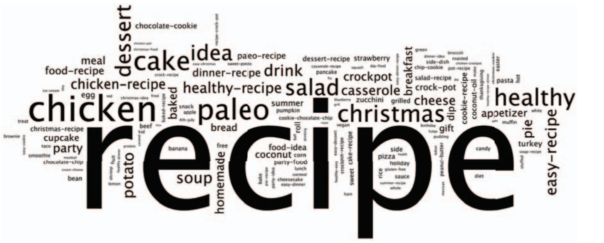

Figure 6. Word cloud presenting the ratio of occurrence for terms in the Food and Drink Vertical. The size of the term indicates the number of images in the PinterNet dataset.

shown in Figure 5 in descending order. We can see how it is a heavy-tailed problem. If we were to directly use search terms as class labels, there would be too many classes and image assignment over classes would get skewed. The top 20 search terms are zoomed in to show the exact phrases used. After label curation, however, we obtain 287 classes, which is a much more reasonable number. Figure 6 shows all the labels in a word cloud. The size of the label indicates the number of images in the corresponding class. We can see "recipe" is the most used single word, even after subtracting its occurrence in multi-word phrases such as "chicken recipe", "health recipe", "cookie recipe", "paleo recipe", etc., all can be seen in the cloud (multi-word labels use '-' in between words for better visualization).

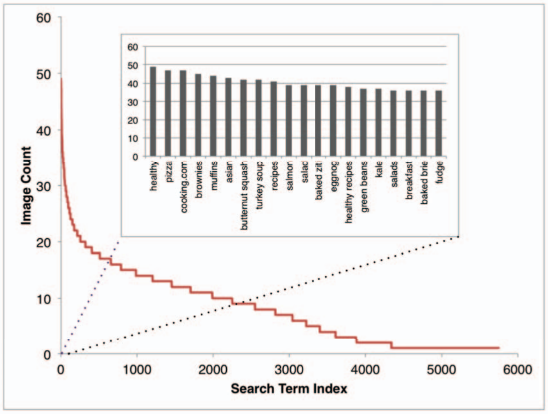

Figure 5. Number of images returned from each distinct search term in *food* vertical. Top 20 search terms in terms of returning size are shown in detail.

The image set we publicize is different from existing datasets in three-folds. It is a dataset of thematic labels, multi-labeled images, and of hierarchies. These special characteristics are explained below.

Thematic labels. Due to the unique characteristics of Pinterest as an idea-provoking platform, the labels that got assigned to images are not names of objects like in ImageNet. Instead, people search for ideas and the images returned are thematic portrayals of those ideas. Theme descriptions can be vague, generic, obscure, and therefore harder for classification systems to recognize, but are a lot closer to how human interprets images. Four thematic labels generated from curating textual search terms of all obtained images (as opposed to curating under a certain vertical), along with several example images, are shown in Figure 7. In the label *first day*, some are posters that contain words like "school", "begins". There are some scenes which suggest summer is over, while others suggest outfits to wear on first day of school (or work). More variations are seen in the label *graduation*. As imagined, most are seen with a scholar cap, but some are about a more derived concept like earning money, moving, or instructions of how to make a graduation cake. In *hair color* the actual position of colored hair can be anywhere, big or small. An image about nail color also appears, possibly due to a user's remark of what hair color goes with this. The theme *hair color* is the most obscure, a concept that is so widely applicable that can basically contain anything. Themes pose a much harder job for machine classifiers, however, they are a closer description of human's needs when searching for an image. The identification of themes in images specifically will be helpful for business applications, but also helpful to semantic understanding of images, and is closer to human cognition.

Multi-labeled images. In this dataset, each image is described by more than one labels. The most number of labels an image has in the current set is 47. The histogram of the number of labels each image is associated with is shown in Figure 8. Over 65% of images got assigned to more than one labels. In a multi-label classification problem, the challenge comes from not only intra-class variation but also inter-class similarity. For example the four labels of the image in Figure 1 are not semantically independent, making the classification difficult.

Hierarchies. Images and labels in PinterNet dataset are organized in a hierarchy. Instead of all labels on one flat level, there is first a separation of verticals. The same label, However, can exist in multiple verticals. For example "gift ideas" can be a label in "fashion" and in "holidays". Another example refers back to Table I. The top 10 verticals each

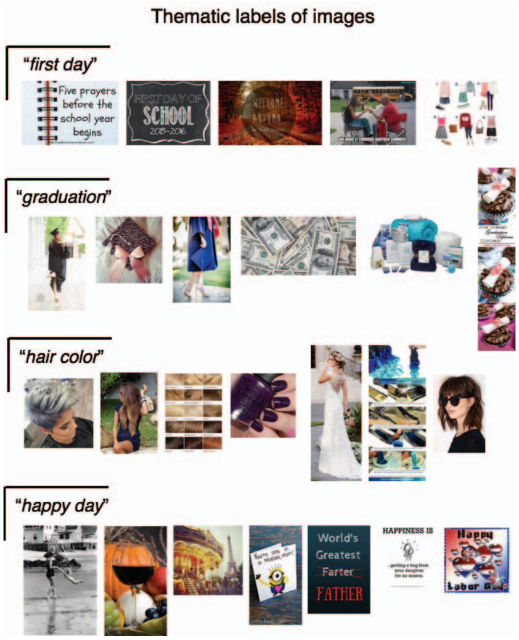
Thematic labels of images
“first day”
Five prayers before the school year begins
FIRST DAY OF SCHOOL 2015-2016
“graduation”
“hair color”
“happy day”
World's Greatest Farter FATHER
HAPPINESS IS
Happy Labor Day

Figure 7. Four thematic labels, “first day”, “graduation”, “hair color”, and “happy day”, each with a couple of image examples. In each case some obscurity is seen. Sometimes it is even required to read the texts in images to be able to classify.

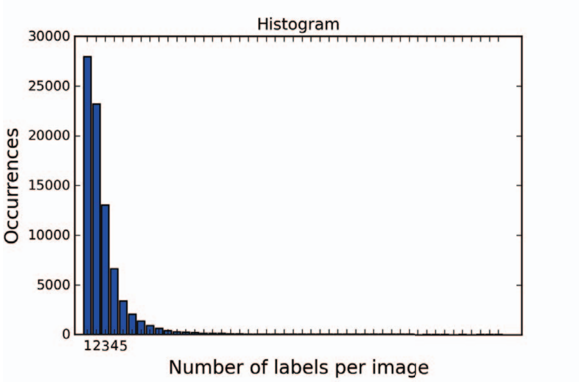

Figure 8. Histogram distribution of number of labels per image. Over 65% of images have more than one labels. This is generated using 80K training data. Test data has similar distribution.

has their own set of labels. The arithmetic sum of labels is 827. However, the actual number of distinct labels are 536, making the overlapping rate as high as 35%. In Figure 9, we show many images with the same label *4th of July* but in different verticals. A classifier is required to generalize that it is actually the color scheme that makes the major determinant.

IV. Training Neural Networks for Theme Classification

This section provides benchmark results using classic CNN architectures on the PinterNet dataset. We perform and present results of the following experiments.

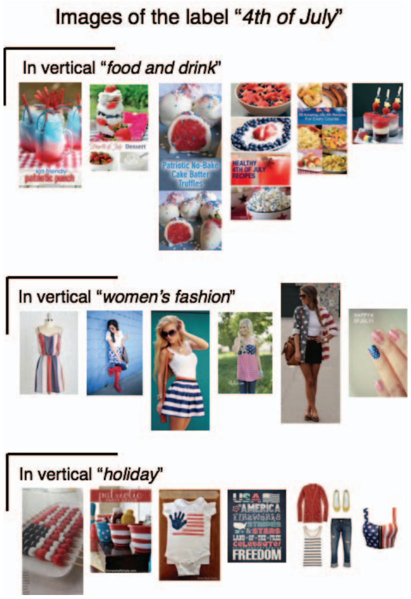
Images of the label “4th of July”
In vertical “food and drink”
In vertical “women's fashion”
In vertical “holiday”
USA AMERICA FREEDOM

Figure 9. Examples of images of the same label, “4th of July”, but under different verticals.

(a) Take all images and treat all 536 classes as on one flat level. No pre-training.
(b) Take all images, first train a binary support vector machine (SVM) classifier that classifies an image as whether *food* or *not food*. Train CNN with images in the *food* vertical with 287 classes. No pre-training.
(c) Same as (b), but with pre-training from all PinterNet image classes.
(d) Same as (b), but with pre-training from ImageNet classes.

Experiment (a) gives the benchmark of using no hierarchical information in images. The total number of theme classes is 536. We use existing popular CNN architectures: AlexNet [1], AlexNet-with-one-weird-trick [11], VGG [2], Overfeat [12], and GoogLeNet [3]. The implementation is based on a multi-GPU Torch package [7]. Parameters used are unchanged from this package. We found AlexNet-with-one-weird-trick to perform the best, beating deeper structures like VGG, Overfeat and GoogLeNet. Results shown in this section are all AlexNet-with-one-weird-trick(referred to AlexNet-OWT for simplicity).

Figure 10 shows the top 1, top 5 and top 10 classification accuracies. With no vertical information to narrow down the image category and no pretraining scheme, the top 1 accuracy can barely reach 5% after 20 epochs (of 5000 iterations of 128 batch size).

Then we move to a specific vertical, *food and drinks* (simplified as *food* from here on). We consolidate the three experiment results on *food* vertical data, with different

[7] https://github.com/soumith/imagenet-multiGPU.torch

schemes of pre-training, into Figure 11. We can see that pretraining from all themes in PinterNet helps the most, even more than pre-training with ImageNet, a much larger object dataset. CNNs are trained on the training dataset (80% of the entire PinterNet dataset). Results shown are all on test data (20% of all data maintaining same image distribution among labels).

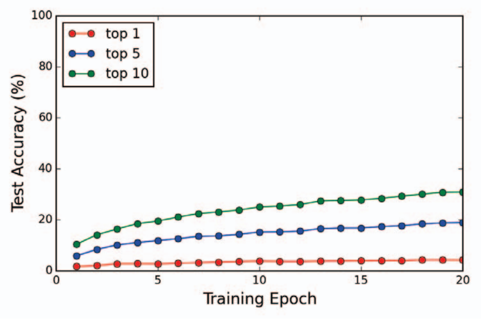

Figure 10. Testing results for Experiment (a): flat 536 classes, no pretraining, using AlexNet for 20 epochs.

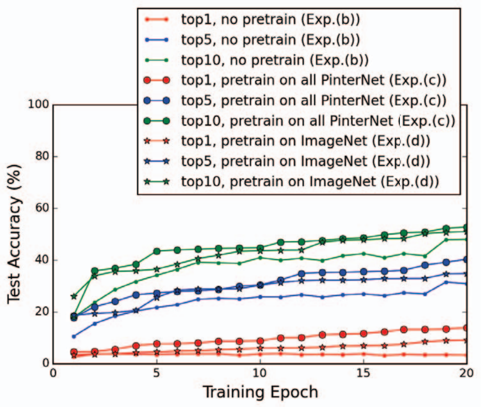

Figure 11. Consolidated testing results for Experiment (b), (c) and (d), all using AlexNet for 20 epochs.

We found that AlexNet-with-one-weird-trick (AlexNet-OWT) performed best among all the top four models for all our verticals with(out) pre-training. In the results presented for Food vertical, we had 287 labels (larger than many other existing Image datasets). Without using any pre-training, we achieved an accuracy of around 50% for label in top 10 output labels, after 30 epoch (of 5000 iterations of 128 batch size). After using pre-training (using Pinternet data and ImageNet data separately), we found around 10% increase in accuracy for label in top 10 output labels. Overfeat had 10% less accuracy compared to AlexNet-OWT. VGG and AlexNet didn't work well. We found similar result across other verticals.

To predict theme of a new image/pin, we first predict its vertical using SVM and then use our best deep CNN model to predict the theme within that vertical. This hierarchical model helps in achieving better accuracy in predicting theme without increasing complexity (depth) of existing top CNN models.

V. Image Collection from Pinterest

This section describes the procedure of image collection from Pinterest using a set of search phrases. The image collection is described by three steps. First, we create search terms which we query images with. Then, we crawl the Pinterest website and collect image URLs for each search term. Next, for each search term, we download a random set of images to build the PinterNet image catalog.

A. *Search term creation*

We generate a search term (e.g. "stocking stuffer ideas for men") by traversing the suggested terms by the Pinterest website for several levels. Once logged in, the Pinterest homepage provides 33 classification categories – the same that we used as verticals in the resulting dataset. Our program first selects one vertical at a time, and visits the vertical homepage. On each vertical page, Pinterest recommends a set of search categories. The second step is visiting each one of these search category pages in that vertical. Again, for each search category page, there are lower level recommended search categories. This way, we traverse the search category tree. It is more appropriate to call it a search category graph as the hierarchy of search terms are not uniquely classified. For example in the vertical "Animals and pets", we have suggested terms to follow as "dogs", "cute", "mammals", etc., mentioned as search categories. However, both "dogs" and "cute" search pages have each other (i.e. "cute" and "dogs" respectively) mentioned as the following level search categories.

In our current work, we traverse up to 6th level categories and also, restrict ourselves to a limited set of search categories in each level. This is because the breadth of the search tree explodes very quickly. Our crawler collects images at every level, not just at the bottom (sixth) level. An example of three-level search term crawling is displayed in Figure 13.

We eventually build 20K unique search terms containing 1 to 6 words. Figure 12 presents the top 20 words, or search categories, and their position at which they occur in the final concatenated search phrase, regardless of verticals. We see that the words "Christmas" and "women" occur mostly at the first position of search phrases while "ideas" and "recipes" occur mostly at the second and third in search phrases.

B. *Image Crawling*

We create an image crawler to obtain images from Pinterest with a certain search term, comprised of either a word or a sequence of words.

We use an open-source web automation tool, Selenium WebDriver [13] for generating the image crawler. Selenium

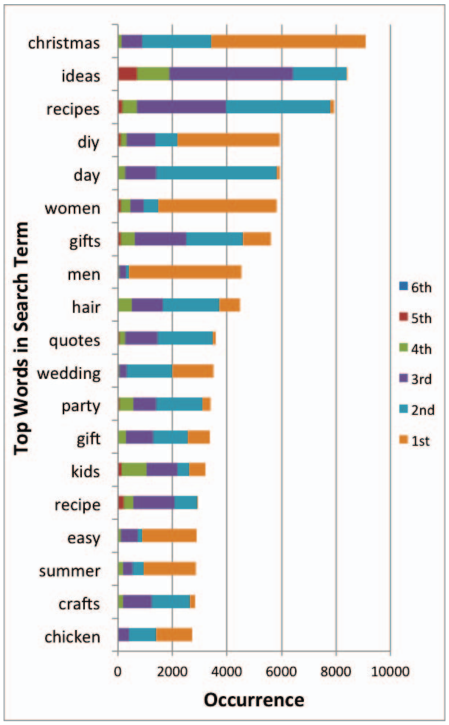

Figure 12. Top 20 words represented as a stacked bar graph. We present the positions at which these words occur.

is one of the most popular suites for automating web application testing and it is employed in many industrial projects. Selenium WebDriver provides a comprehensive programming interface used to control a browser. It offers different ways to locate the UI elements composing a web page i.e. by name, id, xpath, class of the corresponding web element.

First, our code logs into Pinterest by identifying the email address and password feeding location on the login page, and then passing the corresponding values. Our code iterates over the list of created search terms, one at a time. For search term ”dogs”, our query looks like http://www.pinterest.com/search/?q=dogs. q refers to the query containing the search term. For each term, we let the Pinterest page load for a minute. We do this to make sure a sufficient number of images have loaded for that search term. Once the page has loaded, we download the page source. In Table II, we list example search terms for three levels.

Table II
EXAMPLE SEARCH QUERIES FOR THREE SEARCH LEVELS

Category Level	Search Query
Top/First	q=dogs
Second	q=dogs+cute
Third	q=dogs+cute+funny

For every aggregated search term (i.e. second or third level term), the order of the terms do not matter. For e.g. the query q=dogs+cute would give us the same set of images as q=cute+dogs. However, Pinterest keeps updating their image catalog frequently i.e. the set of images for a search term changes with time. Hence, the exact set of images collected by our crawler for both queries may not be same.

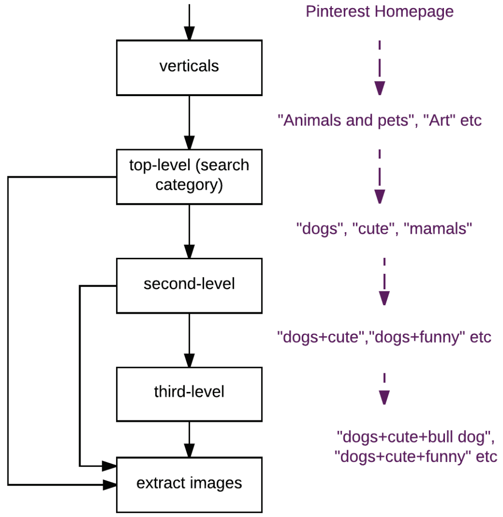
Pinterest Homepage
verticals
"Animals and pets", "Art" etc
top-level (search category)
"dogs", "cute", "mamals"
second-level
"dogs+cute","dogs+funny" etc
third-level
"dogs+cute+bull dog", "dogs+cute+funny" etc
extract images

Figure 13. Steps of crawling from verticals to top and lower level search categories

To ensure a properly curated dataset, we use search terms which are listed under verticals or a higher level search term. This ensures that our image curation database is not arbitrary. Rather, it contains a representative set of pins.

C. Data Cleaning and Downloading Pins

During our image crawling step, we store source pages for 20,408 search phrases. This includes top, second and up to sixth level search terms. Each source page contains a list of pins (Pinterest images) corresponding to that search term.

We use a python library called BeautifulSoup to parse the HTML of the source page. It helps format and organize the HTML structure into an easily traversed Python object. Using BeautifulSoup, we generate a list of pin URLs for each search term from the HTML of the page source. For each pin, we collect the image size, verification information of the pin URL domain, and the name of the vertical it is listed under.

The name of the vertical might seem like redundant information as the search term was constructed from the vertical page. However, we find that it is possible that Pinterest lists a pin image without a vertical. Therefore to make sure our data remains consistent and clean, we ignored images with no vertical information, or images which were not posted by verified user.

The dictionary containing the image details is stored as

MongoDB tables. MongoDB is a free and open source document oriented database and it adheres to the NoSQL paradigm. The database structure resembles a JSON file structure. Finally, we traverse the pin URLs for each search term and download the pins.

VI. Related Work

We discuss research works in three related areas. First of all, we review existing datasets that have been released in computer vision community for various purposes. Then, we explore existing work that tackles data labeling in an automatic, unsupervised fashion. Lastly, we give a brief overview of the literature in perception tasks, including image classification, object detection, image captioning, etc.

A. Datasets

Probably the most widely used large-scale dataset, ImageNet [5] provides large-scale image classification and object localization annotations. It contains over 1.4 million images, with in average 1000 images per class (a total of 1000 object classes). The main focus of ImageNet is object recognition, which means the class labels are only nouns. There is no information regarding what the object is doing, or what the photographer tries to depict, other than the fact that the object exists. Each image is assigned one label, corresponding to only the most salient object in the scene.

PASCAL VOC [14] provides classification (whether an object is there), detection (where is the object) and segmentation (which exact pixels belong to the object) labels for 20 classes of objects. The size of dataset is about 20K. Compared to ImageNet whose images are mostly a clear, centered shot of one object, PASCAL VOC images are less handpicked, more real-world like. It is closer to what we offer in PinterNet, a set of unfiltered natural images that are labeled by themes.

Microsoft COCO [15] contains 300K images for multiple object segmentation – each pixel in image has an assigned label in one of 80 object categories.

Visual Genome [16] is a rich dataset with over 100K images, each associated with a large number of objects, attributes, relationships, and question and answers. Each object is grounded by a bounding box and each image is annotated densely with a number of object descriptions. It provides a platform for tasks that are closer to human perception; rather than recognizing apparent objects in image, can you describe the events (by, say, adjectives)? Can you answer questions about the scene?

The idea of describing images from a different angle than objects was experimented by the Places [17] dataset. As the name suggests, the classes in Places database are about scenes of places, such as “bedroom”, “kitchen”, “forest path”. Images within one scene class can be composed of totally different objects which may pose a greater challenge for object-oriented classification systems.

B. Automatic Labeling

A highlight of our tool is to generate class labels automatically, without human manual annotation. Tools of this kind have been very scarce, although some work are related.

In [18]. a Bayesian Network based interactive system for facial expression labeling is used. Initial labeling is produced automatically, but human has to examine the initial result and make corrections. In [19], Chen et al. presented an automatic segmentation approach given annotated 3D bounding boxes. However there is still human supervision involved.

C. Perception Tasks

With the increasing availability of large datasets, researchers in computer vision, machine learning and data mining society have begun to tackle increasingly complicated problems.

Object detection and classification are mostly performed using ImageNet, with ever improving results shown by AlexNet, VGG, Overfeat, GoogLeNet, ResNet and a recent FractalNet [20]. Object segmentation requires a more fine-grained labeling of objects in pixel level. The usual way to achieve it, is by running a sliding window over each pixel and make local object recognition. Tasks in this category often involve less number of objects compared to object classification.

Image captioning is the task of describing images with natural language, with a freedom of using any words in the vocabulary. Recent approaches [6], [21]–[23] have adopted Recurrent Neural Networks (RNNs) for generating captions, conditioned on image information.

Visual question and answering is an interesting task that has been proposed as a proxy task for evaluating a vision systems capacity for deeper image understanding [24]. Given an image and a question, the system is required to give answers either in free form or from multiple choices.

VII. Future Work and Conclusion

PinterNet is a combination of an automatic label curation tool for web crawled images, and the resulting thematic dataset of 110K Pinterest images. The tool takes free-form, noisy and repetitive textual descriptions of each image, and produce a concise set of meaningful, representative labels of various number of words.

The label generation tool performs association rule mining on the search terms used in image query. Further studies can apply the same strategy on image annotations, user comments, or other noisy textual information associated with online images on social cites. Other future directions to improve the current work include: (1) incorporate image content information (in the form of image features extracted by CNNs) when determining labels; (2) consider word affinities with synonyms grouped together; and (3) extend such practice to all types of data beyond images. Moreover, this tool can be adapted to curate search phrases for commercial product pages in order to achieve more accurate responses for queries.

The novelty of the PinterNet dataset is its thematic labels. Identification of themes requires not only recognizing objects, but also capturing salient information from different perspectives such as color, tone, and arrangement of objects. It is a recognition challenge much closer to true human cognition. This work lays down the ground-work for better theme-based classification in the future.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[6] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[7] A. Karpathy, A. Joulin, and F. F. F. Li, "Deep fragment embeddings for bidirectional image sentence mapping," in *Advances in neural information processing systems*, 2014, pp. 1889–1897.

[8] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *Advances in Neural Information Processing Systems*, 2015, pp. 2935–2943.

[9] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pp. 69–72.

[10] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215, 1994, pp. 487–499.

[11] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv preprint arXiv:1404.5997*, 2014.

[12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[13] M. Leotta, D. Clerissi, F. Ricca, and C. Spadaro, "Improving test suites maintainability with the page object pattern: An industrial case study," in *Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on*. IEEE, 2013, pp. 108–113.

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 740–755.

[16] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: http://arxiv.org/abs/1602.07332

[17] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[18] L. Zhang, Y. Tong, and Q. Ji, "Interactive labeling of facial action units," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[19] L.-C. Chen, S. Fidler, A. Yuille, and R. Urtasun, "Beat the mturkers: Automatic image labeling from weak 3d supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3198–3205.

[20] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648*, 2016.

[21] X. Chen and C. L. Zitnick, "Learning a recurrent visual representation for image caption generation," *arXiv preprint arXiv:1411.5654*, 2014.

[22] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[23] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[24] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7w: Grounded question answering in images," *arXiv preprint arXiv:1511.03416*, 2015.