Automatic Image Dataset Construction from Click-through Logs Using Deep Neural Network

Yalong Bai
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, 150001, P. R. China
ylbai@mtlab.hit.edu.cn

Kuiyuan Yang
Microsoft Research
13F, Bldg 2, No. 5, Danling St
Beijing, 100080, P. R. China
kuyang@microsoft.com

Wei Yu
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, 150001, P. R. China
w.yu@hit.edu.cn

Chang Xu
College of Computer and Control Engineering
Nankai University
Tianjin, 300071, P. R. China
changxu@nbjl.nankai.edu.cn

Wei-Ying Ma
Microsoft Research
13F, Bldg 2, No. 5, Danling St
Beijing, 100080, P. R. China
wyma@microsoft.com

Tiejun Zhao
School of Computer Science and Technology
Harbin Institute of Technology
Harbin, 150001, P. R. China
tjzhao@hit.edu.cn

ABSTRACT

Labelled image datasets are the backbone for high-level image understanding tasks with wide application scenarios, and continuously drive and evaluate the progress of feature designing and supervised learning models. Recently, the million scale labelled image dataset further contributes to the rebirth of deep convolutional neural network and bypass manual designing handcraft features. However, the construction process of image dataset is mainly manual-based and quite labor intensive, which often take years' efforts to construct a million scale dataset with high quality. In this paper, we propose a deep learning based method to construct large scale image dataset in an automatic way. Specifically, word representation and image representation are learned in a deep neural network from large amount of click-through logs, and further used to define word-word similarity and image-word similarity. These two similarities are used to automatize the two labor intensive steps in manual-based image dataset construction: query formation and noisy image removal. With a new proposed cross convolutional filter regularizer, we can construct a million scale image dataset in one week. Finally, two image datasets are constructed to verify the effectiveness of the method. In addition to scale, the automatically constructed dataset has comparable accuracy, diversity and cross-dataset generalization with manually labelled image datasets.

Categories and Subject Descriptors

H.2.4 [**Information Systems**]: Systems—*Multimedia databases*

MM '15, Brisbane, Australia

DOI: http://dx.doi.org/10.1145/2733373.2806243.

General Terms

Algorithms, Experimentation, Performance

Keywords

Automatic Image Dataset Construction, Image Representation, Word Representation, Deep Learning

1. INTRODUCTION

Image datasets with category labels serve as the backbone for high-level image understanding tasks, and typically used by splitting into training set for model learning and testing set for performance evaluation. With the help of crowdsourcing, ImageNet constructed by manually labeling has reached million scale [6] and acts as one important factor for the great success of deep convolutional neural network by automatically learning visual features [16, 22, 23].

However, the process of manual-based image dataset construction is both time consuming and labor intensive. For example, ImageNet has taken years' efforts to label 21841 categories (nouns from WordNet [18]), which is still far from their goal to label all WordNet nouns, not to mention the continuously emerging categories such as new products and cartoon characters. Even worse, the time cost will cause long latency for application scenarios which require constructing datasets with their own category lists.

To find ways to shorten the construction process, we analyze the whole process by breaking down it into three steps:

1. *Category list generation*, which is often pre-defined according to specific task;
2. *Query formation*, where each category is expanded to a set of queries by referring synonyms in existing ontology such as WordNet, then submitted to image search engines to collect candidate images.
3. *Noisy image removal*, manually verifying each candidate image to removal noisy images.

Obviously, the most time-consuming part comes from step 2 and 3 especially when the number of categories is large.

Step 2 can be bypassed if categories already exist in some predefined ontology, but efforts are required for new emerging categories. While step 3 is extremely time-consuming since there are thousands of images need to be manually verified for each category.

In this work, we propose to relieve the cost by automatizing both step 2 and 3, and construct high quality image datasets in a scalable and timely manner. The proposed method relies on click-through logs from image search engine where lots interactions between queries and images have been associated, and learns both image representation and word representation in a deep learning framework. With the learned representations, word-word similarity and image-word similarity can be effectively calculated. The word-word similarity is used to automatize step 2 by finding similar words for each category. Meanwhile, image-word similarity is used to automatically verify candidate images by removing noisy images that are of low similarity to the category. In the used deep neural network, image representation is modeled by convolutional layers and a fully-connect layer (also some layers without weights such as max-pooling layer), and word representation is formulated as a weight matrix of a fully-connected layer following the fully-connect layer of image representation. To resist heavy tail distribution in click-through logs, we further introduce a regularizer called "Cross Filters Regularization" to speed up the training process.

To verify the effectiveness of the proposed automatic image dataset construction method, we first construct a small scale dataset with 10 categories, and show its cross-dataset generalization ability by comparing with other two manually constructed datasets. Furthermore, to verify the scalability of our method, we construct another large scale dataset with 1000 categories and demonstrate its comparable accuracy and diversity with other human labelled high quality datasets.

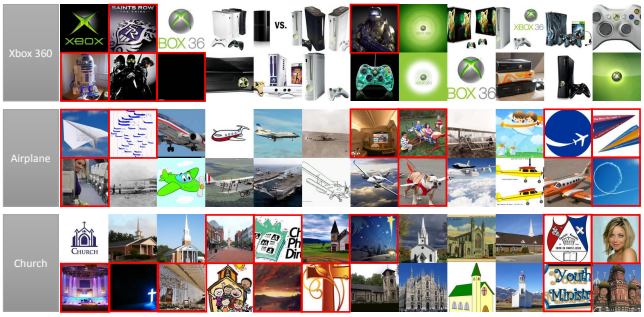

Figure 1: A snapshot of click-through logs from Bing image search. Images marked with red boxes are noisy images.

The rest of the paper is organized as follow. We cover related work in Section 2 and then present our automatic method for image dataset construction in Section 3. In Section 4, we apply the proposed method to construct datasets, and analysis characteristics of the constructed datasets in detail. Finally, we discussion conclusions and future work in Section 5.

2. RELATED WORK

Considering the importance of image datasets in the area of image content understanding, lots of efforts have been involved in constructing image datasets. Most works are manually based, while some works explore automatic ways as ours.

2.1 Manual based image dataset construction

The classical way to build an image dataset is manually based (e.g., ImageNet [6], CIFAR-10/100 [15], Pascal VOC series [8], Caltech-101/256 [9, 11], LabelMe [20], SUN [28], etc.). Most of these datasets are built by sending category names to image search engines and aggregating returned images as candidate images, then cleaning candidate images by human judgement. Here, we briefly discuss these works along the steps involved in image dataset construction:

Generating Category List The generation of category list depends on specific tasks. For example, SUN [28] targets on scene recognition task by defining 899 scene categories. Borth *et al.* [1] proposed to detect visual sentiment by constructing a dataset around a category list with strong sentiment. Datasets such as TinyImage [25] and ImageNet [6] directly adopt nouns of WordNet as category list, which cover a large amount of objects but are still far from complete.

Query formation Since most image search engines restrict the number of images returned for each query (in the order of hundreds to one thousand) and only top ranked images are with acceptable precision. To overcome the restriction, synonyms are often used to expand a category into a query set. Moreover, methods such as appending category with popular adjectives and words from its parent category, even translating category to different languages are further used to enrich the query set. All expanded queries will submit to several popular image search engines to collect candidate images from Internet. The method only works for categories defined from existing ontology such as WordNet [18], and cannot generalize to categories that have not been compiled into existing ontology. Recently, word embedding [5, 19] provides a learning based method to compute similarity between words and can be used to bypass the manual compilation of ontology. In this paper, we use learning based method to obtain word representations, and automatically expand a category to a query set.

Noisy image removal The candidate images contains lots noisy images with average accuracy around 10% [6]. Human efforts are involved to remove noisy images by checking candidate images one by one. As this step is quite time consuming and labor intensive, NUS-WIDE only partially labelled the whole dataset [3], while TinyImage [25] and visual sentiment dataset [1] keep all raw candidate images without manual labelling. We are interested in generating high quality image dataset without manual labelling. By leveraging the power of deep neural network, noisy images are automatically removed by calculating image-category similarity with the learned image representation and word representation.

2.2 Learning based image dataset construction

To save labelling cost, some works also explored in the direction of learning based image dataset construction. Collins *et al.* [4] proposed to construct image dataset in a semi-automatic manner by using active learning. Several randomly selected images are firstly labelled by human as seed training set for classifier learning, then the learned classifier

is used to examine the unlabelled images to find out unconfident images for manual labelling. The process is iterated until sufficient classification accuracy is obtained.

To further reduce labelling cost, semi-supervised learning is applied to learn classifiers from small amount of labelled images and large amount of unlabelled images. These labelled images can be manually labelled [21] or top-ranked images from image search engines [2, 7, 17]. Schroff *et al.* proposed a multi-modal approach by employing both text, metadata and visual features to remove noisy images. Hua *et al.* [12] proposed clustering based method and propagation based method to remove noisy images, where clustering based method removes large irrelevant image "groups" and propagation based method further removes relatively smaller noises. The human cost of this method lies in labelling clusters to learn cluster level classifier for irrelevant cluster removal.

There are lots of work related to the step of *noisy image removal* though not aiming for dataset construction. To name a few, Weston *et al.* [27] proposed to represent images and annotations jointly in a low dimensional embedding space for relevance estimation, the method is limited by the used low level visual representations. Frome *et al.* [10] proposed to map images into a semantic space learned via word embedding from large scale text corpus. However, the semantic space constructed from text corpus is suitable for NLP tasks while not necessarily reflects visual similarity.

In summary, exiting learning based methods save human cost by leveraging the generalization ability of machine learning models. However, the generalization ability is affected by both quantity and quality of manually labelled images, also models' capability where shallow models are limited. To the best of our knowledge, we are the first to use deep neural network for fully-automatic image dataset construction by gaining generalization ability from large scale click-through logs.

3. AUTOMATIC APPROACH

We are targeting at constructing image dataset in a scalable way while ensuring both accuracy and diversity. The basic idea is to automatize the two most labor cost steps: query formation and noisy image removal. In a computational perspective, the core components for these two steps are two similarity metrics, one is word-word similarity to expand each category to a set of similar words, the other is image-word similarity to remove images not relevant to a category. Furthermore, these two similarity metrics are well defined if word representation and image representation can be effectively obtained. Inspired by the success of deep learning in learning word representation for NLP tasks [5] and image representation for image classification [16], we learn these two representations simultaneously in a deep neural network. To train such a complex model with millions parameters, we resort to large scale click-through logs from image search engine. With the trained model, word representation and image representation are naturally defined, and word-word similarity and image-word similarity are further obtained which are used for automatizing the two labor intensive steps in image dataset construction.

3.1 Representation Learning via DNN

We will first introduce how to model the word representation, image representation and their associations. All these

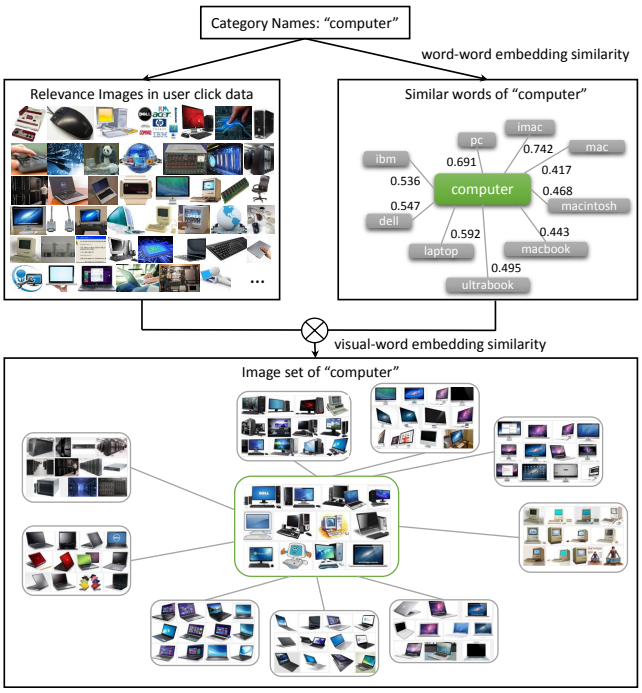

Figure 2: Illustration of our process for automatic image dataset construction. For a given category such as "computer", we first find its similar words with learned word representation, then collect the relevant images from click-through logs according to the similarity between image and similar word set.

computation are then carried by introducing a deep neural network. The learning process is performed on the deep neural network through standard method using click-through logs.

3.1.1 Word Representation

Similar to word embedding technique which has successfully used in NLP tasks [5], we also represent each word w in vocabulary $\mathcal{V}$ [1] as a vector $\mathbf{e}_w \in \mathbb{R}^{D_w}$ in a continuous space, and denote all word embedding vectors as a matrix E. Instead of learning the representation using text corpus with context constraints which often results only syntactic-level similarity, the word representation will be directly learned using click-through logs together with image representation.

3.1.2 Image Representation

Convolutional neural network has demonstrated its superiority in learning image representation from low level to middle level until high level [29]. The network used for image representation learning contains four convolutional layers and one fully-connected layer, three max-pooling layers are used following the first, second and fourth convolutional layers, two local contrast normalization layers are used following the first and second convolutional layers. For image I, its representation is obtained from the output vector $\mathbf{f}(I;\theta_{image}) \in \mathbb{R}^{D_i}$ of the last fully-connected layer, where θ_{image} denotes all parameters in the network for image rep-

[1]In this paper, we use top 50,000 most frequent words in click-through logs as vocabulary

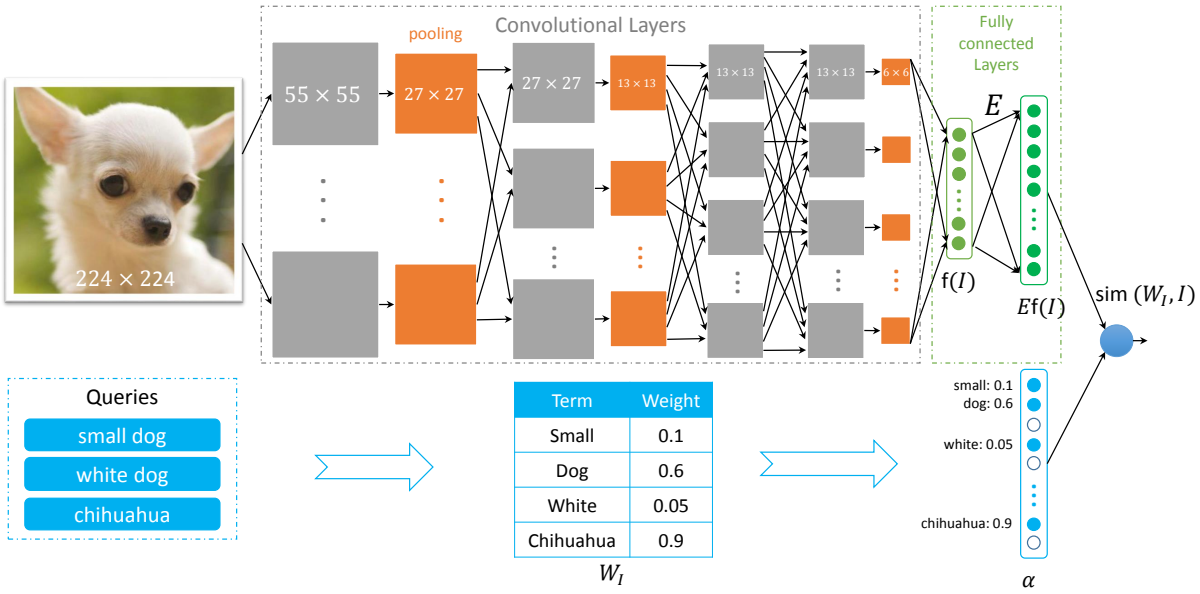

Figure 3: Illustration of the deep neural network for image-query association. The output vector $\mathbf{f}(I)$ of the first fully-connected layer gives the image representation. The weight matrix E encodes word representations in the whole vocabulary. $\boldsymbol{\alpha}$ is the bag-of-textual-words vector generated based on queries.

resentation learning and will be omitted for clarity. Here, we set the dimension of image vector D_i equal to the dimension of word vector D_w for easing image-word similarity caculation.

3.1.3 Image-Query Association

The word representation and image representation are determined by E and θ_{image}, respectively. We follow the standard machine learning pipeline to learn all parameters using supervision from large scale click-through logs. In click-through logs, large number associations are established between queries and images through massive user interactions with image search engine. All queries clicked to image I are merged into a document W_I to form the image's word based representation. Through average composition, document W_I can also be represented as a vector in the same space of word

$$\mathbf{e}_{W_I} = \sum_{w_i \in W_I} \alpha_i \mathbf{e}_{w_i} \tag{1}$$

where α_i is the tf-idf weight for word w_i.

Image I and its associated document W_I is similar based on the judgement of user clicks, should also be similar measured by image representation and word representation, i.e.,

$$\begin{aligned}
\text{sim}(W_I, I) &= \langle \mathbf{e}_{W_I}, \mathbf{f}(I) \rangle & (2)\\
&= \Big\langle \sum_{w_i \in W_I} \alpha_i \mathbf{e}_{w_i}, \mathbf{f}(I) \Big\rangle & (3)\\
&= \sum_{w_i \in W_I} \alpha_i \langle \mathbf{e}_{w_i}, \mathbf{f}(I) \rangle & (4)\\
&= \sum_{w_i \in \mathcal{V}} \alpha_i \langle \mathbf{e}_{w_i}, \mathbf{f}(I) \rangle & (5)\\
&= \langle \boldsymbol{\alpha}, E\mathbf{f}(I) \rangle & (6)
\end{aligned}$$

Eq(5) summarizes over all words in vocabulary $\mathcal{V}$ where the weights of words not in W_I are zero. $E\mathbf{f}(I)$ can be computed as the output of a fully-connected layer with weight matrix E and input $\mathbf{f}(I)$. Then all computations can be carried in a deep neural network as illustrated in Figure 3.

To avoid trivial solution that maximizes $\text{sim}(W_I, I)$ by simply scaling E or $\mathbf{f}(I)$, we further modify the dot product into cosine similarity

$$\text{sim}(W_I, I) = \frac{\langle \boldsymbol{\alpha}, E\mathbf{f}(I) \rangle}{\|\boldsymbol{\alpha}\| \cdot \|E\mathbf{f}(I)\|} \tag{7}$$

Then the objective function on the whole click-through dataset is defined as

$$J_0(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \text{sim}(W_{I_i}, I_i) + \frac{\beta}{2} \|\theta\| \tag{8}$$

where θ summarizes all parameters in E and θ_{image}.

3.1.4 Cross Convolutional Filters Regularization

The query distribution in click-through dataset has a very long tail. That is only a few of words are frequently appeared, while most words are with low frequency. It also means large amount of similar inputs are fed into DNN frequently during the training process, which leads detection of high-frequency patterns with large neuron responses, and filters in convolutional layers are prone to activate for those highly frequent inputs. Thus, there usually are lots of highly similar filters in convolutional layers at the beginning of training. These highly similar filters not only waste the capacity of the model, but also cause the neural network converge slowly.

To speed up the training process, we add a new regularizer called Cross Convolutional Filters Regularization (CFR) to penalize highly similar filters by adding a regularization term to the objective function. Then the final objective function

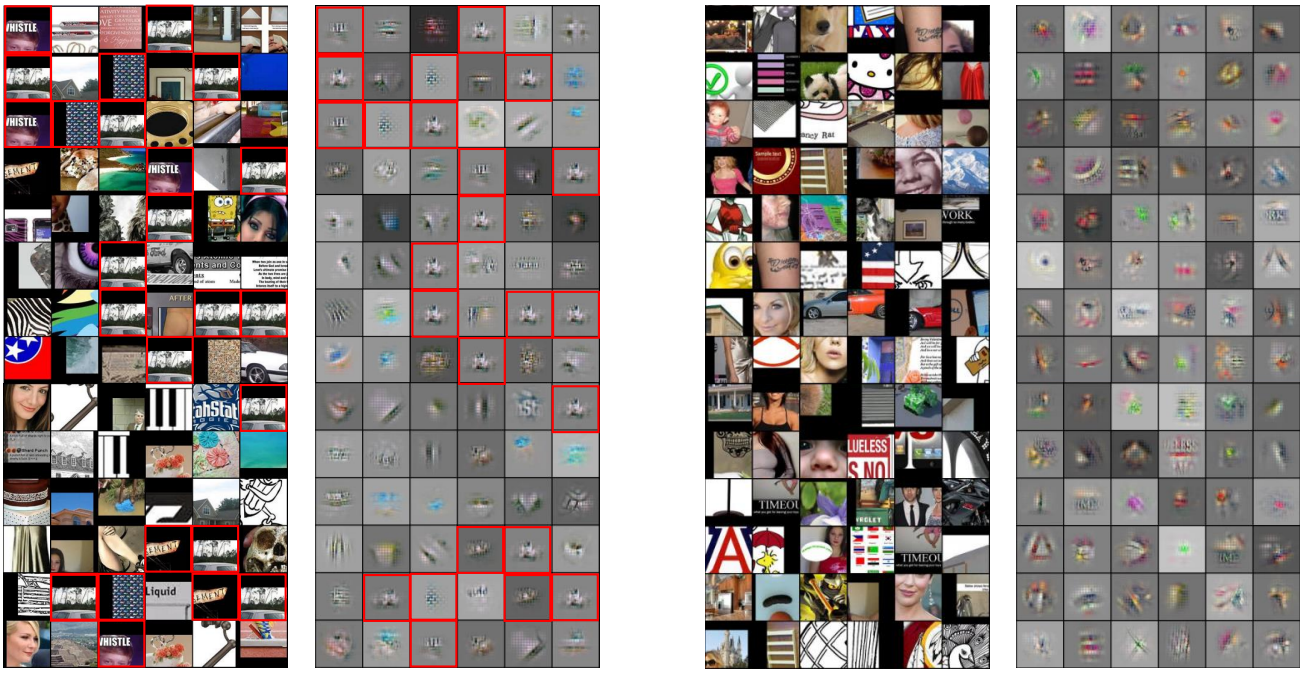

Figure 4: Visualization of filters at the 19th epoch of training. we show the strongest activations in a random subset of the last convolutional layer's feature maps across the click image dataset. The corresponding image patches for each feature map are also shown in the figure: a) DNN, b) DNN+CFR.

is defined as

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\text{sim}(W_{I_i}, I_i) + \frac{\beta}{2}\|\theta\| + \gamma\sum_{l=1}^{L}\sum_{i=1}^{s_l}\sum_{j=i+1}^{s_l}\langle\theta_i^l, \theta_j^l\rangle \quad (9)$$

where L is the total number of convolutional layers in the neural network, s_l is the number of filters in layer l, and θ_i^l is the parameter vector of the ith filter in layer l. The first term measures the fitness of the neural network on the click-through dataset. The second term is a weight decay term that penalizes weights with large magnitude and used to avoid overfitting. The third term is cross convolutional filters regularization, which aims to penalize highly similar filters.

Figure 4a and 4b visualize the learned filters before and after adding CFR at the early stage of training. The filters are from the last convolutional layer and visualized by image patches with the strongest activations and their deconvolution versions as proposed by [29]. Obviously, CFR can help the network learn more diverse filters, while without CFR, capacity is wasted with lots of similar filters. Figure 5 compares the learning curve of average cosine similarity in the training process, CFR can significantly speed up the training process.

3.1.5 *Training Details*

The training process of the whole model follows standard training setting of deep neural network. Specifically, we trained our model using stochastic gradient descent with a

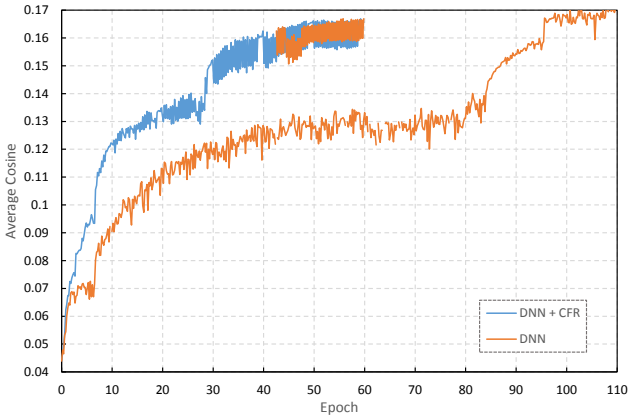

Figure 5: Learning curves of DNN and DNN with CFR. DNN with CFR reaches 0.13 average cosine similarity three times faster than without CFR.

mini-batch size of 128 samples, weight decay of 5×10^{-4}, cross convolutional filters regularization of 5×10^{-4}, and dropout rate of 0.5 for the first fully-connected layer. The learning rate was initially set to 1, and then decreased by a factor of 10 when average cosine similarity score on validation set stops improving. In total, the learning rate was decreased 4 times.

We initialized the weights in each layers from a normal distribution with the zero mean and 0.01 variance. The neuron biases in the first full-connected hidden layer and

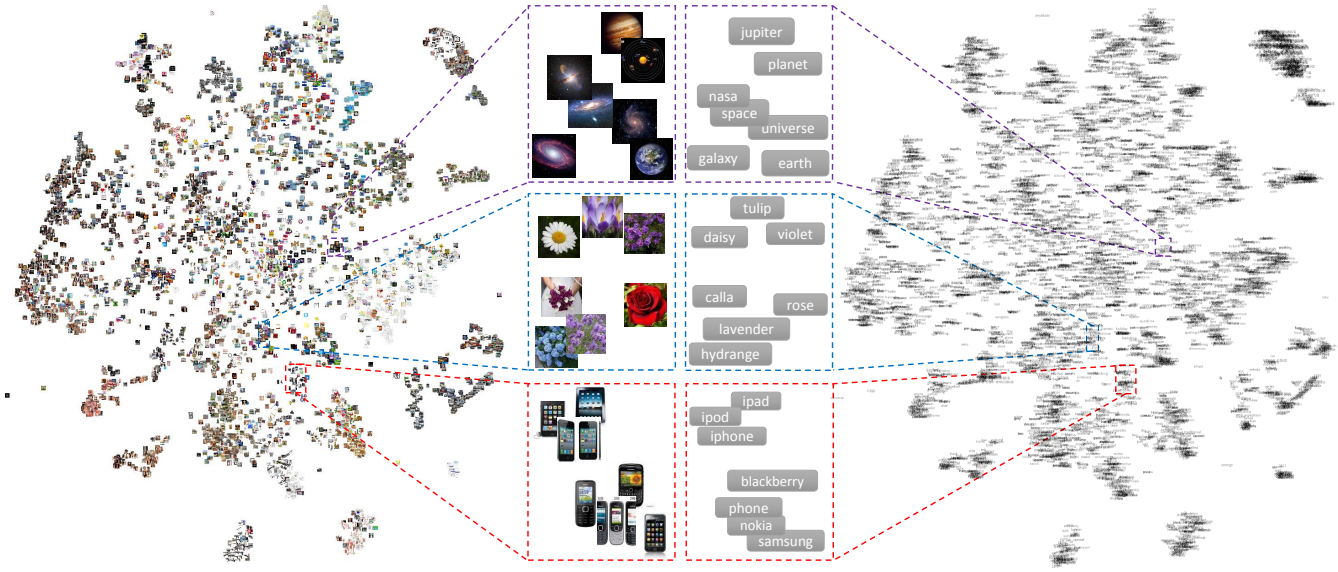

Figure 6: t-SNE visualization of learned word representations of 5,000 randomly selected words, each word is projected as a point in the two dimensional space. Left: each point is pasted with an image clicked by the word. Right: each point is pasted with the string of the word. Middle: Zoom-in to three regions.

the second, fourth convolutional layers were initialized with the constant 1, while the neuron biases in the remaining layers were initialized with zero.

It should be noted that CFR is only used in the early stage of training and removed when learning rate decreased to 0.01. After warm start with less highly similar filters, CFR is removed to let give the objective function more flexility to fit training data.

3.2 Constructing Image Dataset

In this section, we present the automatic way to construct image dataset from click-through logs based on the learned word representation $\mathbf{e}_w$ and image representation $\mathbf{f}(I)$. The process starts with a set of category words $\mathcal{C} = \{c_k\}_{k=1}^{K}$ predefined according specific task on hand, and outputs a set of images for each category. Since the process is the same for every category, we will only detail the construction process for category c as an example.

3.2.1 *Query Formation*

For category c, we first expand it to a set of similar words from vocabulary $\mathcal{V}$. The learned word representation is used for measuring word-word similarity

$$\text{sim}(c, w_i) = \frac{\langle \mathbf{e}_c, \mathbf{e}_{w_i} \rangle}{\|\mathbf{e}_c\| \cdot \|\mathbf{e}_{w_i}\|}. \tag{10}$$

Accordingly, similar words for category c is defined by

$$\mathcal{S}_c = \{w_i | \text{sim}(c, w_i) > \xi_w, w_i \in \mathcal{V}\} \tag{11}$$

With the generated similar word set $\mathcal{S}_c$, candidate image set $\mathcal{I}_c$ for category c are collected by aggregating images clicked to queries that contain words in $\mathcal{S}_c$.

3.2.2 *Noisy Image Removal*

Candidate images still contain many noises, this step is to remove noisy images by leveraging both image representation $\mathbf{f}(I)$ and word representation $\mathbf{e}_w$. Given similar word set $\mathcal{S}_c$ and candidate image set $\mathcal{I}_c$, image set $\mathcal{D}_c$ for category c after noisy removal is obtained by

$$\mathcal{D}_c = \{I | \text{sim}(I, \mathcal{S}_c) > \xi_i, I \in \mathcal{I}_c\} \tag{12}$$

where similarity between image $I \in \mathcal{I}_c$ and similar word set $\mathcal{S}_c$ is measured as

$$\text{sim}(I, \mathcal{S}_c) = \max_{w \in \mathcal{S}_c} \text{sim}(w, c) \cdot \text{sim}(w, I). \tag{13}$$

The similarity prefers images that are similar to some word in $\mathcal{S}_c$ and the word is similar to category c. From another perspective, words close to the category have the capacity to keep more images which is consistent with intuition.

The threshold ξ_i for noisy image removal in Eq (12) is set to 0.35 considering the number of images can be collected for each category and ξ_i can be set higher when more click-through logs are available.

4. EXPERIMENTAL RESULTS

We use the click-through logs publicly available from Bing image search [2]. There are two sets of click-through logs: Clickture-Lite and Clickture-Full[13]. Clickture-Lite contains 11.7 million queries and 1 million images, while Clicture-Full is much larger which contains 73.6 million queries and 40 million images. In our experiments, we use Clickture-Lite to train the DNN, and apply the trained DNN on Clickture-Full to construct image dataset. We construct two image datasets to verify our proposed method. The first image dataset named as AutoSet-10 is constructed for 10 categories from CIFAR-10. The second image dataset named as AutoSet-1K, is constructed for 1000 popular categories which were frequently searched by users.

[2]http://research.microsoft.com/en-us/projects/clickture/

Table 1: For six query words, we show their top-10 similar words based on our method and context based word embedding [19]. Words with italic font are visually dissimilar. Our results contains some typos from user queries, but no plurals as we merged plurals in $\mathcal{V}$.

Category	KNN based on visual based word embedding	KNN based on context based word embedding
dog	puppy, dogss, *breed*, hound, spaniel boxer, cocker, retriever, beagle, mastiff	dogs, puppy, *cat*, *pet*, pup canine, puppies, *cats*, *kitten*, terrier
airplane	airplan, aeroplane, boeing, 747, dreamliner lockheed, sukhoi, beechcraft, bomber, fighter	airplanes, aeroplane, plane, *flying*, aircraft planes, takeoff, airliner, helicopter, jet
iphone	ipone, 4s, iphone4s, iphone5, iphon *4gs*, iphone4, *5*, phone5, *otterbox*	*ipad*, *ipod*, 3gs, iphone4, *android* *itouch*, *3g*, ios, iphones, *itunes*
egyptian	pharaoh, egyption, mummification, *horu*, thoth tutankhamun, *heiroglyphic*, bastet, egiptian, mummy.	egypt, *aztec*, egyptians, *arabian*, *greek* egyption, *mayan*, *arabic*, pharaohs, sphinx
universe	hubble, galaxy, supernova, astronomy, *telescope* milkyway, nebula, comet, protostar, andromeda	universes, cosmos, worlds, *existence*, cosmic planet, *infinite*, *realm*, humanity, exist
pokemon	pokeman, *legendary*, victini, bulbasaur, zekrom charizard, pokemoncard, *shiny*, emboar, arceu	*gameboy*, pikachu, *gba*, *nintendo*, *naruto* mew, *yugioh*, *mario*, *zelda*, *digimon*

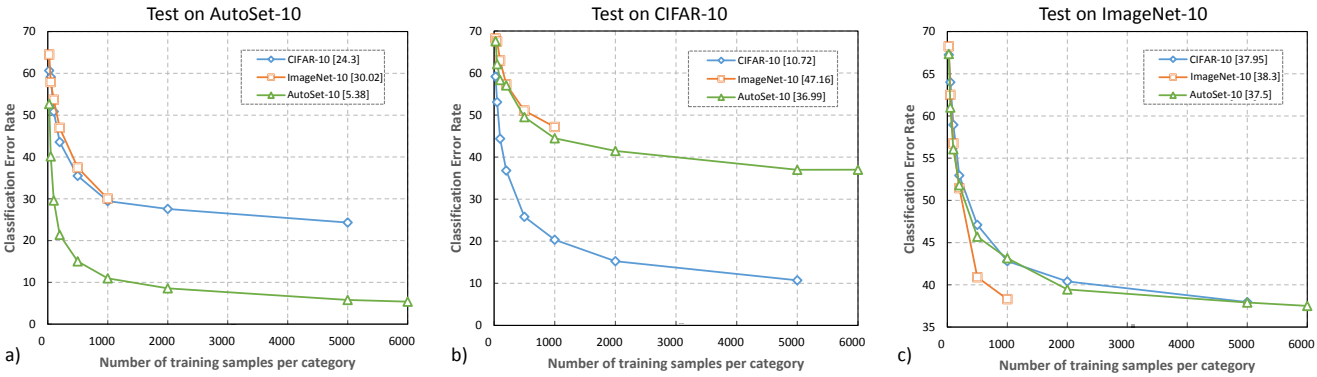

Figure 7: Cross dataset generalization of CNNs trained on AutoSet-10, CIFAR-10 and ImageNet-10, then tested on: a) AutoSet-10, b) CIFAR-10 and c) ImageNet-10.

4.1 Learned Representations

With the learned image representation and word representation, images and words are representated as points in high dimensional space. In Figure 6, we use dimension reduction tool t-SNE [26] to map a set of 5,000 randomly selected words as points in two dimensional space, each point is pasted with an image clicked to the word or string of the word. It can be observed that visually similar words are closely distributed in a meaningful way. For example, words related to handheld are close in the space.

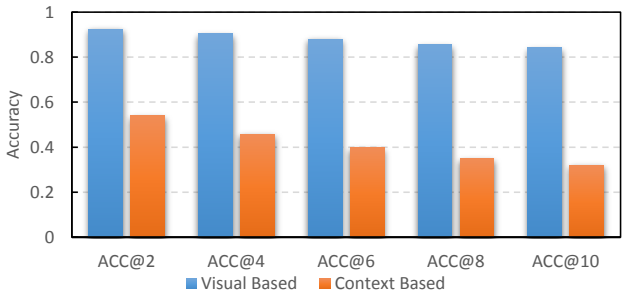

Figure 8: Average accuracy of top-K similar words on 100 randomly words based on our method and context based word embedding [19].

4.2 AutoSet-10

Moreover, we carry a quantitative evaluation for the learned word representation and compare it with state-of-the-art word representation learned from large scale text corpus (including 840 billion tokens) with context constraints [19]. For a set of 100 randomly selected words, we use different word representations to find their top similar words from the vocabulary. The groundtruth of these similar word pairs are manually judged, a pair is labelled as true if the two words are describing the same visual pattern otherwise false. Some exemplar words and their similar words calculated by different word representations are present in Table 1, words with italic font are not visually similar. Obviously, our word representation reflects more about visually similar. Quantitative comparison is based on the following performance metric

$$acc@K = \frac{\sum_{k=1}^{K} 1\{(w_k, w)\}}{K} \tag{14}$$

where $1\{\cdot\}$ is an indicator function, so that it equals to 1 if (w_k, w) is true visually similar word pair and 0 otherwise. Figure 8 shows the average $acc@K$ on the 100 randomly selected words based on the two word representations, and our word representation learned together with image achieved higher accuracy than word representation learned from context based word embedding, thus our word representation

Figure 9: A snapshot of six categories and their sample images in AutoSet-1K.

is more suitable to expand category for image dataset construction.

We first construct a small scale dataset named as AutoSet-10 using the same 10 categories of CIFAR-10 [15]. AutoSet-10 consists of 70000 images in 10 categories, and randomly split into 60000 images as training set and 10000 images as testing set. CIFAR-10 contains 60000 images in total, 50000 images as training set and 10000 images as testing set, which are pre-split. In addition, we construct a subset of ImageNet named as ImageNet-10 using the same 10 categories, and randomly split into 10000 images as training set and 2000 images as testing set.

To compare these three datasets, we conduct a set of image classification experiments to verify their cross-dataset generalization ability [24]. Cross-dataset generalization measures the performance of classifiers learned from one dataset on the other dataset.

The same configuration of convolution neural network is used to build the classifiers, which is modified from [14])(layers-conv-local-11pct.cfg) by replacing local-connected layers with convolutional layers. All images in AutoSet-10 and ImageNet-10 are resized and central cropped into same size as CIFAR-10, i.e., 32×32. Randomly cropped 24×24 image patches with horizontal flips are used as data augmentation for training. Initial learning rate is set to 0.001 and decreased by a factor 10 twice when the validation error stops descreasing.

Figure 7 shows the classification error rates. Each dataset produces one CNN using its training set, and all three CNNs are compared on test set of AutoSet-10 (a), CIFAR-10 (b) and ImageNet-10 (c). In all three cases, at the same number of training samples, the best performance is achieved by training and testing on the same dataset. Since the smallest dataset ImageNet-10 only have 1000 training images per category, we compare the performance of three different dataset at the point of 1000 training samples, it shows that the generalization ability of the three dataset is very close and AutoSet-10 performs slightly better than ImageNet-10 on CIFAR-10. Since AutoSet-10 is larger than the other two datasets, it achieved the best performance on two testing sets when all training samples are used. The comparison at the same number of training samples shows the comparable generalization ability of AuotSet-10 with other two manually constructed datasets, and AutoSet-10 can further get better generalization benefits from its scale up ability.

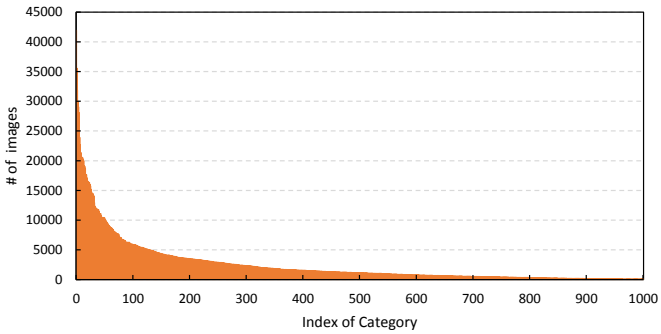

Figure 10: Number of images per category in AutoSet-1K.

4.3 AutoSet-1K

To show the scalability in number of categories, we build a large scale image dataset named as AuotSet-1K consists of 2.5 millions of images for 1000 popular categories that are frequently searched by users. The 1000 categories not only include the long existing categories such as "dog" or other natural objects, but also include many new categories such as products, TV dramas and digital games. Some examplar

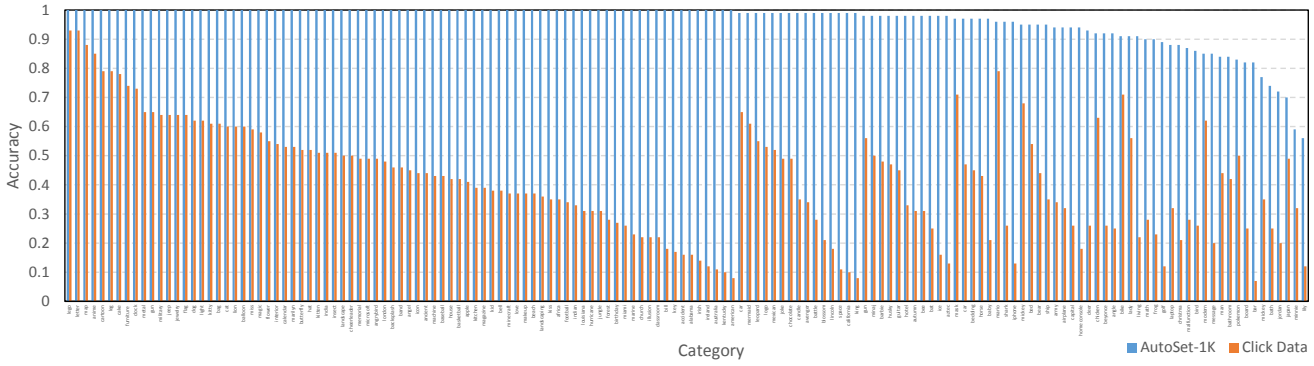

Figure 11: Accuracy of AutoSet-1K and candidate images collected from click-through logs. Randomly sampled 15000 images from AutoSet-1K and 15000 images from click-through logs are manually verified.

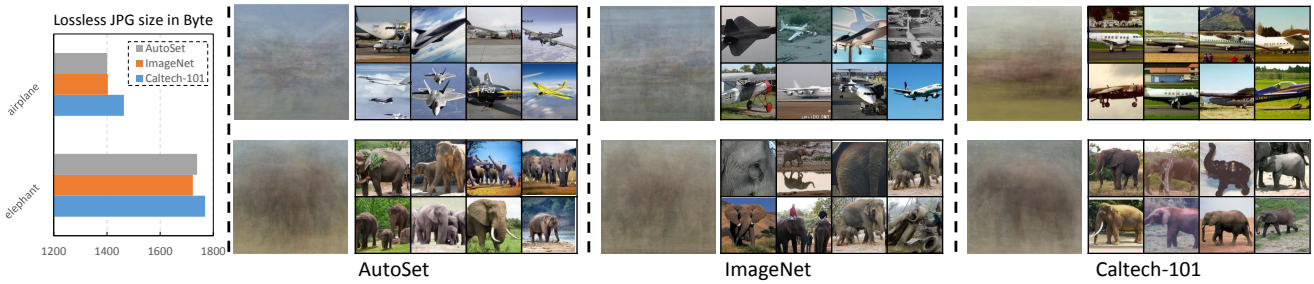

Figure 12: Average images and image samples of AutoSet-1K, ImageNet and Caltech-101 from two categories: "elephant" and "airplane". Left chart shows the comparison of the lossless JPG file sizes of average images, we downsampled the average image to 32×32 and sizes are measured in byte. Diverse image set results in a smaller lossless JPG file size.

categories with randomly sampled images are showed in Figure 9. The whole construction process of AutoSet-1K takes a week include the training of deep neural network, which is also an advantage over manually based methods that often need years' efforts to reach datasets with such scale.

In addition to short latency for constructing large scale dataset, we further analysis the quality of AutoSet-1K from three aspects as following.

Scale AutoSet-1K contains 1000 popular categories, and on average over 2500 images for each category. Figure 10 shows the numbers of images per category in AutoSet-1K. Different with static image datasets, AutoSet-1K can be dynamically increased with few efforts.

Accuracy To evaluate accuracy of AutoSet-1K, we randomly sampled 150 categories and 100 images per category. Each category and its images are manually judged, the accuracy on each category are showed in Figure 11. The average accuracy of AutoSet-1K is 95.2% which is much higher than 42.3% of candidate images directly collected from click-through logs, but still little lower than 99.7% achieved by ImageNet. To be noted that, there are several failure categories in AutoSet-1K such as "jordan" with low accuracy 72%, as lots of images about "jordan shoes" instead of "Michael Jordan" exist in click-through logs and mistreated as "jordan" too.

Diversity In order to illustrate the diversity of images belong to a category, we followed the method in [6], which compare the average image of each category and measure lossless JPG file size which reflects the amount of information in an image. Diverse image set should result in a blurrier average image, and the JPG file size of average image of diverse images should be smaller. We resize all images to 256×256, and create average images with randomly selected 60 images for each category. Figure 12 shows the average images of two categories: "elephant" and "airplane", similar to ImageNet, the average image of AutoSet-1K is also very blurrier and hard to recognize out the object, while the average image of Caltech-101 is more structured and sharper. AutoSet-1K has comparable JPG file size with ImageNet, but significantly smaller than Caltech-101.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a deep learning based method to automatically construct image dataset from large scale user click-through logs. DNN is used to learn image representation and word representation for calculating word-word similarity and image-word similarity, and further used to automatize two labor costing steps in image dataset construction: query formation and noisy image removal. Moreover, cross convolutional filter regularization is proposed to speed up the training process. The auto-constructed image dataset has no scale up issue, and with high accuracy, diversity and good cross dataset generalization.

One limitation of our method is the unigram based vocabulary, while some categories cannot be represented as unigrams such as "hot dog". In the future, we will replace the unigram based vocabulary with phrase based vocabu-

lary, which will greatly increase the size of vocabulary and require better design of the training system.

6. REFERENCES

[1] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 223–232. ACM, 2013.

[2] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1409–1416. IEEE, 2013.

[3] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. NUS-WIDE: A Real-World Web Image Database from National University of Singapore. In *CIVR*, Santorini, Greece., 2009.

[4] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. In *ECCV*. 2008.

[5] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3270–3277. IEEE, 2014.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):594–611, 2006.

[10] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.

[11] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[12] X.-S. Hua and J. Li. Prajna: Towards recognizing whatever you want from images without image labeling. AAAI - Association for the Advancement of Artificial Intelligence, January 2015.

[13] X.-S. Hua, L. Yang, J. Wang, J. Wang, M. Ye, K. Wang, Y. Rui, and J. Li. Clickage: Towards bridging semantic and intent gaps via mining click logs of search engines. In *ACM Multimedia*, 2013.

[14] A. Krizhevsky. cuda-convnet2, 2014. `http://code.google.com/p/cuda-convnet2/`.

[15] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 2009.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[17] L.-J. Li and L. Fei-Fei. Optimol: automatic online picture collection via incremental model learning. *International journal of computer vision*, 88(2):147–168, 2010.

[18] G. A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[19] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12, 2014.

[20] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[21] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*, pages 369–383. 2012.

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

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[24] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[25] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008.

[26] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[27] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, volume 11, pages 2764–2770, 2011.

[28] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[29] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*. 2014.