Construction of Diverse Image Datasets from Web Collections with Limited Labeling

Niluthpol Chowdhury Mithun, *Member, IEEE*, Rameswar Panda, *Member, IEEE*, and Amit K. Roy-Chowdhury, *Fellow, IEEE*

***Abstract*—Image datasets play a pivotal role in advancing computer vision and multimedia research. However, most of the datasets are created by extensive human effort, and are extremely expensive to scale up. To address these issues, several automatic and semi-automatic approaches have been proposed for creating datasets by refining web images. However, these approaches either include significant redundant images in the dataset or fail to provide a diverse enough set to train a robust classifier. Ideally, a representative subset should be both semantically and visually diverse so as to provide the maximum amount of information under the current budget. Most current approaches are entirely based on analysis of visual features, which may not well correlate with image semantics and hence, collected images may not be enough to give a detailed understanding of a category. In this paper, we propose a system for creating diverse image dataset collections from the web with limited manual labeling effort. It is based upon a semi-supervised sparse coding framework that employs a joint visual-semantic space to simultaneously utilize both images and associated textual information from the web for dataset construction. Additionally, the proposed system is online that is capable of collecting more discriminative images continuously as new data becomes available, which is also suitable for enriching existing datasets. Experiments demonstrate that our system can create and enrich datasets with limited manual labeling, with better cross-dataset generalization capability and diversity compared to the state-of-the-art datasets.**

***Index Terms*—Image Dataset, Active Learning, Sparse Optimization, Joint Image-Text Analysis.**

I. Introduction

The efficiency of several visual recognition tasks depends upon the ability to identify suitable training examples to learn initial models. The majority of the success in this regard has been achieved by models trained on large-scale hand-labeled image datasets (e.g., SUN [73], ImageNet[50]). Although, these datasets cover large numbers of categories, expanding them to new categories or providing new examples to an existing category, is extremely costly and labor-intensive [32]. Moreover, there exist various types of bias in the popular image datasets and hence, they do not demonstrate satisfactory cross-dataset generalization (training on a dataset, testing on a different dataset) capability [29], [65]. Future multimedia and image analysis research requires examining even a greater number of visual categories and adapting to higher intra-class variation present within a category [7]. The complexity of the models will increase over time to cope with this. Hence, creating high-quality image datasets and continuously updating existing datasets with new diverse examples is becoming more important over time. Complete human labeling based solution is unlikely to keep pace with this growing need.

• Niluthpol Chowdhury Mithun, and Amit K. Roy-Chowdhury are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA. Rameswar Panda was with Department of Electrical and Computer Engineering, University of California, Riverside. Currently, he is with IBM Research AI (MIT-IBM Watson AI Lab). E-mails: (nmithun@ece.ucr.edu, rpand002@ucr.edu, amitrc@ece.ucr.edu)

(a) Google (b) Twitter (c) Performance

Fig. 1. The significance and challenges in collecting a diverse and informative set of images for dataset construction. (a),(b) Few top-ranked images from Google and Twitter for query 'Bike'. (c) Accuracy of a classifier (10 classes and 400 per class) for different combinations of train and test data.

Motivated by the above, the main goal of this work is to develop a system for constructing high-quality image datasets with a limited labeling budget. The images in each category of the dataset should be relevant and diverse both visually and semantically so as to provide the maximum amount of information for a category under the current budget. The secondary goal is to develop an online framework, that is capable of collecting more diverse images continuously as new data becomes available, which is suitable for enriching existing image datasets. In order to achieve these goals, we propose a sparse coding based framework with human in the loop for dataset creation from web images. Our system is capable of concurrently utilizing images and associated text, by learning a joint latent space upon the image-text association.

Motivation: To address the issues of creating large-scale hand-labeled image datasets, and inspired by streams of images available on the web, there has been lot of recent interest in developing systems for curating web images for creating a dataset with no or minimal human labeling [71], [34], [4], [2], [75], [76], [7]. However, most of these approaches rely heavily on a reliable search engine (e.g. Google [32], [19]), for image collection or initial selection of seed images. This may bring up issues of bias and lack of diversity. Moreover, most of the approaches primarily aim at collecting as many relevant images as possible. Hence, in spite of causing serious wastage of space, the dataset loses quality, and training with these images may not provide expected performance gain (See Fig. 1 for an illustrative example). We observe that search-engines usually provide relevant but archetypal images [40], and hardly represent the diversity of real-world scenarios. On the contrary,

social media provides diverse real-world images, but there is a high chance of getting irrelevant images. A high-quality dataset should cover both relevance and diversity aspects of a query. Hence, it is important to efficiently leverage a relatively clean but less diverse set of images, in conjunction with a diverse but noisy annotated image set in creating datasets.

Recently, many dataset construction methods have been proposed to utilize freely available images with user-generated tags or captions from social media websites like Flickr. However, most existing methods use only one of the modalities (i.e., image or text) for refining web images [76], [81], [81], [52], [35], [4], [81], [80], [62], [58], [59]. Text-based approaches [4], [52], [81] suffer from the ambiguous nature of textual description and fail to connect relevant images under different keyword indices. Moreover, many images indexed by the same tag may be irrelevant [81]. Visual-only methods [76] are effective for images with similar visual content but often fail to find relevant images having the same semantic meaning but a moderate difference in visual content.

In this regard, several methods have been proposed that use both modalities sequentially for dataset construction [52], [75], [74], [35], [49], [25], [82], [71], [11]. To increase diversity in a created dataset and to overcome the download restrictions of the image search engines, some previous works [11], [75], [74], [44] use query expansion to collect many candidate images from the search engines. Then visual feature based analysis is applied to the collected images to select the final set. There are also many works to clean up web images utilizing both visual information and associated text [71], [82], which utilizes rationale of visual consistency to clean up images [35]. These methods leverage visually close images to assign tags to a new image [71], [82], utilize relationships between images labeled with the same tag [82], and learn visual classifiers from socially tagged examples [34], [52]. The above works exploit the image modality and the text modality in a sequential way. By contrast, our approach focuses on concurrently exploiting both the image and all the associated tags (if available) in creating a semantically and visually diverse dataset. This allows us to better model the correlations between the visual data and tags.

The significance of using images and associated text concurrently for dataset construction can be identified from Fig. 2. The tags in 2(a) and 2(b) are similar and represent the same event, but they look very different. The same can be said of 2(c) and 2(d). On the contrary, the tags in 2(a) and 2(c) represent different events but the images look similar. Selecting a subset of these images based on text/image only methods or sequential methods is likely to be sub-optimal. Jointly utilizing both image and text modality has been shown to be effective in several applications, e.g., cross-modal retrieval [31], [43], image captioning [26], image search[21], video summarization [5]. We believe that the ability to simultaneously use image and text will also be a catalyst for web-based dataset construction methods to reach full potential.

A few works have also explored user information along with visual information and tag information for refining tags of social images [51], [60], [61]. The key idea is that user information is likely to reveal important cues regarding the correctness of the tags. Different tensor completion based approaches have been explored in these works for tag refinement [51], [60], [61]. In contrast, we only use visual information and associated tags in our work. However, an interesting future work would be to investigate how user information can be integrated into the current framework for resolving ambiguities in collecting web images in many cases.

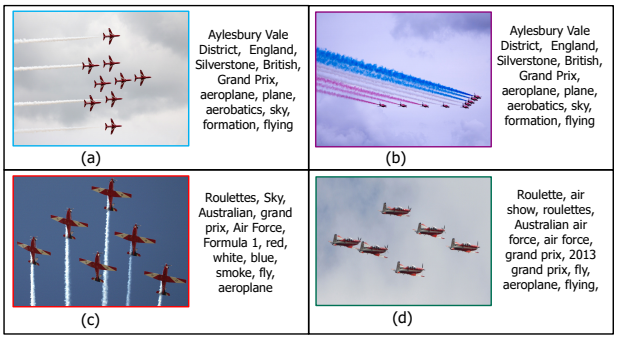


Fig. 2. Example Web images from Flickr for query 'aeroplane grand prix' to illustrate the significance of jointly using images and associated text for dataset construction.

Contributions: The main contributions of this work can be summarized as follows.

- This paper presents a novel system to construct image datasets that are diverse given limited labeling. The proposed framework is online, i.e., capable of collecting diverse images continuously as new data becomes available. Hence, it can be utilized for both creating a high-precision image dataset from scratch and/or efficiently updating an already created dataset with diverse examples when new data becomes available.
- The proposed approach explores both visual information and associated tags from noisy web image collection for dataset construction. To the best of our knowledge, this is the first proposal for image dataset construction using joint image-text embedding. Our system not only allows to select visually diverse images but also gives a way to select images that are semantically more representative and diverse.
- We develop a diversity-aware sparse representative selection based active learning approach, which provides flexibility that permits filtering out irrelevant images and obtaining a reliable set of diverse images based on the budget available.
- Experimental results demonstrate that our system is not only useful in reducing the manual annotation efforts, but also successful in collecting images with high precision, scalability, and diversity, and robust image classifiers can be trained from these images which shows better cross-dataset generalization compared to other methods.

This paper is an extended version of our work [44]. The main extension is taking both visual modality and text modality (when available) into account concurrently using a joint embedding space in the process of diverse representative selection, instead of using only visual modality. Ideally, a good representative subset will be both semantically and visually diverse. The second difference is, using SVM based active learning scheme, where the SVM classifier model is used for actively selecting samples, that can be incrementally updated with new labeled samples. A sparse reconstruction based

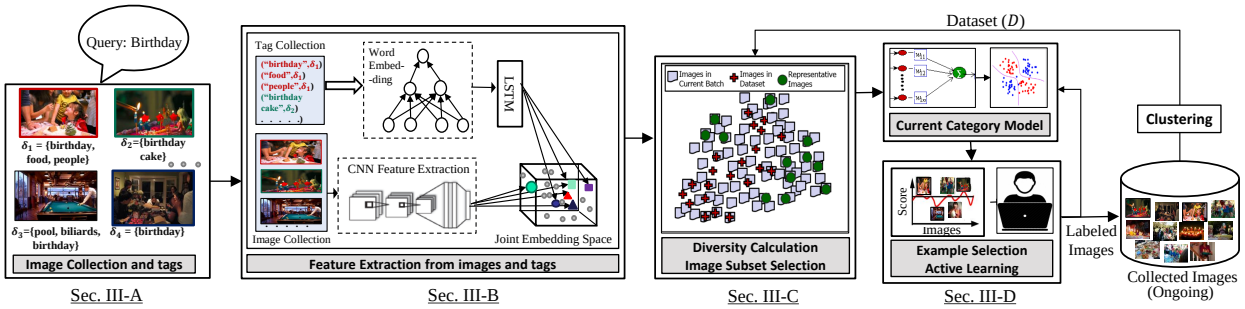


Fig. 3. A brief illustration of our proposed framework for collecting images. Please see the text in Sec. III for details.

method was used in our previous work [44], which utilizes all previously selected positive labeled dataset images of that category for labeling new samples. The incremental SVM-based active learning scheme not only has a less computational load but also utilizes negatively labeled samples in updating the model. Experiments show that these changes result in an improvement over the previous work.

II. Related Works

Our method builds upon several machine learning tools, e.g., visual-semantic embedding [30], [69], active learning [53], [64], sparse coding [14], [70].

Active learning. Despite the advances in machine learning, human in the loop remains a popular concept for achieving desired performance and adapting learned models[16]. Active learning with human in the loop, in which a limited number of unlabeled examples are selected to be labeled by a human [67], [68], is an iterative and interactive process, which is a natural fit for our scenario of refining web images with limited labeling budget. In active learning [53], the system selects a few instances that are assumed to be the most informative and then asks queries about these instances to a human, who labels only instances that are assumed to be the most informative. Recently, a few semi-supervised approaches have been developed for large-scale dataset creation that minimizes human effort for dataset creation using active learning framework [76], [9], [7]. However, these methods generally fail to select a diverse set of examples to train a robust classifier. Moreover, these approaches [7], [76] may select many samples for human labeling that have significant information overlap [15].

Sparse Coding. Recently, there has been a growing interest in applying sparse coding techniques in many computer vision tasks, such as image restoration [37], activity recognition [57], face recognition [70] and classification [79], [63]. Sparse coding based techniques have been highly successful in finding an informative representative subset of a large number of data points [12], [8], [14], which fits well in the scenario of selecting informative examples for dataset construction. The representatives can be used to obtain high precision classifiers using few selected samples and annotated from a large pool of unlabeled samples [15], [13].

Visual-Semantic Embedding. Recently, joint image-text models have shown impressive performance on several computer vision tasks, such as cross-modal retrieval [78], [69], image captioning [38], [26], image classification [20] video summarization [48]. Motivated by these applications, we build on top of a joint image-text embedding for dataset creation. The embedding is given by transformation functions trained to project visual and textual features into a common space where similar image and text are mapped nearby. We follow [30] to learn a joint embedding utilizing image-text pairs and the learned embedding is used as the representation of image and associated text for selecting a diverse image subset.

III. Proposed System

We start by giving notations and overview of our system and then present the detailed system of dataset creation.

Notation: Throughout this paper, we use uppercase letters to denote matrices and lowercase letters to denote vectors. For matrix $X = (x_{ij})$, its i-th row and j-th column are denoted by $x_{i,.}$ and x_j respectively. $||X||_F$ is Frobenius norm of X. The l_p-norm of the vector $x \in \mathbb{R}^n$ is defined as $||x||_p = \sum_{j=1}^{N}(|x_j|^p)^{1/p}$ and l_0-norm is defined as $||x||_0 = \sum_{j=1}^{N}\|x_j\|_0$. The Frobenius norm of $X \in \mathbb{R}^{n\times m}$ is defined as $||X||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}^2}$. Generalized $l_{r,p}$ norm is defined as $||X||_{r,p} = \sum_{i=1}^{N}(|x_{i,.}|_r^p)^{1/p}$. When $r \geq 1$ and $p \geq 1$, the $l_{r,p}$ norm is a valid norm since it satisfies the three basic conditions of a norm. However, when $r < 1$ or $p < 1$, $l_{r,p}$ norm is not valid, but we also call them norms for convenience. The operator $diag(.)$ puts a vector on the main diagonal of a matrix.

Overview of the system: In our system, collecting images of one category is independent of other categories. Hence, images for different categories can be collected in parallel. Fig 3 summarizes our incremental image collection framework for a category 'birthday' from a web source (e.g., Flickr). Initially, we collect images and associated tags related to the category from several web-sources. Our framework for selecting images for dataset is an incremental one, so that information learned about a category from previous batches can be utilized in estimating the relevance of next batch of images. Hence, our system is suitable for both creating a new dataset and enriching an existing dataset with new examples. During each run of incremental update, we process a batch from the collected images. First, we employ a diversity-aware sparse representative selection approach to choose a smaller set of representative images that not only best represents this batch, but also is distinctive to the images in current dataset. Then, we calculate relevance of each representative image based on previously labeled images using a SVM classifier

model. Initially, we learn the prior SVM classifier model from a few positive (few top ranked images from a reliable search engine) and negative images (few images related to other query words) corresponding to the concept. Based on the relevance score, we employ active learning to decide whether to label an image manually or not. As collecting images for one category is independent of others, we ask only binary questions to annotator. When we have sufficient new images in a buffer labeled by the active learning system, we update the SVM model with new images labeled by the system and add positive examples to the dataset. We continuously update the dataset with images labeled by the system and update the model.

A. Image Collection and Pre-processing

Since the number of returned images from a web search based on a query is limited, we use a query expansion scheme to increase the amount of data. The expansion is done using ConceptNet [56]. We use only synonyms and derived phrases as expanded queries, as these are highly relevant. For example, given a query 'Bike', we expand to queries such as 'Bicycle', 'Ride Bike', 'Mountain Bike' etc. The expanded queries are used to collect images from different web sources (e.g. Google, Bing, Flickr). We also collect associated tags from web sources (e.g. Flickr) when available. We filter out meaningless tags. We also filter out the images having very low quality, e.g., out-of-focus or blurred, too white or black, empty or too small.

We divide the training data into batches of size 200. The first batch is used to train the prior SVM model. Rest of the batches are processed using our framework sequentially to select diverse examples from the batch that are also dissimilar to the previously selected examples and update the SVM models. We considered no manually labelled image is available beforehand. We take advantage of the high-precision of few top-ranked image from Google search-engine by utilizing them as the first batch and initially collected images. For updating this model, we obtain the newly labeled instances from the active learner and store them in a buffer. We use the obtained labels from a batch for incremental training of the models. We also use the obtained labels to update the dataset.

B. Feature Extraction

In this work, we process relatively clean set of images from web search engines (Google or Bing) initially. After these images are processed, we process images collected from social media (e.g., Flickr). In order to better utilize, the relationship between image and corresponding tags from social media images, we extract visual and textual features utilizing a joint image-text embedding space. If no text is available for a Flickr image, we consider the query as the associated tag. In this work, we employ a pre-trained joint embedding model learned using two-branch network utilizing image-text pairs from MSCOCO dataset [36] with pairwise ranking loss following [30]. However, our method does not depend on specific joint embedding methods and any image-text embedding method can be used [66], [42]. Here, initially a deep pre-trained CNN is used to produce visual feature representation [55], denoted by $\overline{v} \in \mathbb{R}^{B^{(v)}}$ and word2vec model is used [41] to produce the representation of words, denoted by $\overline{t} \in \mathbb{R}^{B^{(t)}}$. One of the branches of this network takes in visual features and the another takes in text features. We briefly describe the method for training the embedding below.

Learning joint embedding of image and Text. Given visual feature representation, the projections for images can be derived as $x^{(v)} = W^{(v)}\overline{v}$ ($x^{(v)} \in \mathbb{R}^{B}$), where $W^{(v)} \in \mathbb{R}^{B \times B^{(v)}}$ is the transformation matrix that projects the visual content into the joint embedding. On the other hand, given representation for tags, the projection of text in the joint embedding $x^{(t)}$ is found from the hidden state of the LSTM. Here, given the feature representation of both images and corresponding text, our goal is to learn a joint embedding characterized by θ (i.e., $W^{(v)}$ and LSTM weights) such that the visual and semantic content are projected into the joint space. The network is trained by minimizing a pairwise ranking loss combining bi-directional ranking terms in order to learn to maximize the similarity between a image embedding and its corresponding text embedding and minimize similarity to all other non-matching ones. The optimization problem can be written as,

$$\min_{\theta} \sum_{x^{(v)}} \sum_{x^{(t)^-}} \max\{0, \alpha - \mathrm{S}(x^{(v)}, x^{(t)}) + \mathrm{S}(x^{(v)}, x^{(t)^-})\} + \sum_{x^{(t)}} \sum_{x^{(v)^-}} \max\{0, \alpha - \mathrm{S}(x^{(t)}, x^{(v)}) + \mathrm{S}(x^{(t)}, x^{(v)^-})\} \tag{1}$$

Here, $x^{(t)}$ is a matching text embedding for image embedding $x^{(v)}$ and $x^{(t)^-}$ is non-matching text embedding. α is the margin value for the pairwise ranking loss. The scoring function $\mathrm{S}(x^{(v)}, x^{(t)})$ is defined as cosine similarity to measure the similarity between the embedded images and text. In (1), the first term ensures that for each visual feature, matching text features should be closer than non-matching ones, and similarly, the second term ensures text features that correspond to the image should be closer to each other than non-matching image features. The joint embedding is trained on the combination of image-text pairs from MS COCO dataset [36], which contains multiple annotations for a large number of images. The embedding was trained using stochastic gradient descent for 30 epoch with an initial learning rate of 0.001 and decreased by a factor of 10 after every 10 epoch. The margin α was set as 0.2 following [30].

After learning the embedding, we map visual and text features from our Flickr image collection to the shared semantic space and use them to compute our objective of selecting a small representative set, that will be both semantically and visually diverse so as to provide the maximum amount of information under the current budget. As the joint space exhibit multimodal linguistic regularity phenomenon [30], when multiple tags are available for an image, we generate the feature representation by summing over the embeddings of all associated tags and then normalizing it by the number of tags. Averaged word vectors has been shown to be a strong feature for text in several tasks [77], [28], [27], especially when the order is unknown.

C. Diverse Representative Set Selection

The goal of this step is to find a small set of images from the input batch that conveys the most important details of the

batch. Since importance is a subjective notion, we define a good representative set as one that has the following properties.

Representativeness. The input set of web images should be reconstructed with high accuracy using the extracted subset. We extend this notion of representative as finding a subset that simultaneously minimizes reconstruction error of collected images of the category, as well as the set of semantic information associated with the images.

Sparsity. Although the subset should be representative and diverse, the total number of images in the subset should be as small as possible.

Diversity. The selected subset should be diverse, as much as possible, capturing different aspects of the input collection. Moreover, the amount of redundancy with respect to previously collected images should be minimal.

Based on the above, we design the subset selection objective, enforcing representativeness, sparsity and diversity, as explained below.

1) Representative and Sparse Subset Selection: The goal of this step is to find an optimal subset of the current batch of images. In particular, we are trying to represent the current batch of images by selecting only a few representative images, which are also dissimilar to the images in the current dataset. Our representative selection algorithm is based on sparse representative selection (SRS) [8], [13], [14] approach. The basic idea behind SRS is to utilize the self-expressiveness property, which states that each point in the dataset can be described as a linear combination of a few of the selected representative points [8], [13], [14]. The natural goal for SRS is to establish a image level sparsity which can be induced by performing l_1 regularization on rows of the selection matrix $Z \in \mathbb{R}^{N\times N}$. By introducing the row sparsity regularizer, the problem can now be succinctly formulated as

$$\min_{Z} \; ||X^{(v)} - X^{(v)}Z||_F^2 + \lambda||X^{(t)} - X^{(t)}Z||_F^2; \quad \text{s.t. } ||Z||_{2,0} \leq \tau, \tag{2}$$

$\|Z\|_{2,0}$ gives the number of nonzero rows of the matrix Z. λ and τ are tradeoff parameters and N denotes the number of images in the batch. $X^{(v)} \in \mathbb{R}^{B\times N}$ is the feature matrix for all images in the current batch, where $X^{(v)} = \{x_j^{(v)} \in \mathbb{R}^B, j = 1, \cdots, N\}$. Each column of $X^{(v)}$ i.e., $x_j^{(v)}$ represents the feature descriptor of an image in current batch in joint embedding space. Similarly, $X^{(t)} \in \mathbb{R}^{B\times N}$ is feature matrix for tags corresponding to images in current batch.

(2) is a NP-hard problem since it requires searching over every subset of the τ columns of X. A standard relaxation to the problem (2) is given by

$$\min_{Z} \; ||X^{(v)} - X^{(v)}Z||_F^2 + \lambda||X^{(t)} - X^{(t)}Z||_F^2; \quad \text{s.t. } ||Z||_{2,1} \leq \tau, \tag{3}$$

$\|Z\|_{2,1} \triangleq \sum_{i=1}^{N} \|z_{i,.}\|_2$ is the row sparsity regularizer, i.e., sum of l_2 norms of the rows of Z.

In (3), the objective is motivated by the fact that the representatives of the set should come from images in current batch. The constraint i.e., $l_{2,1}$ regularizer is to induce row level sparsity in a matrix, which is very common in representative selection [8], [14]. However, this formulation only characterizes the reconstruction capability and sparsity but does not account for the fact that the selected images should be dissimilar to the previously selected images. As a result, it may leave out some crucial images and select redundant ones.

2) Adding Diversity in Subset Selection : To leverage diversity along with representativeness, we propose a simple extension to (3) as follows:

$$\min_{Z} \; ||X^{(v)} - X^{(v)}Z||_F^2 + \lambda||X^{(t)} - X^{(t)}Z||_F^2; \quad \text{s.t. } ||KZ||_{2,1} \leq \tau, \tag{4}$$

where, $K = [diag(k)]^{-1}$ and $k_j \in \mathbb{R}^N$ represent the diversity score of jth image. It is easy to see that, (4) favors selection of diverse images by assigning a lower score prior via K. So, optimization of (4) attempts to obtain a sparse set of images non-redundant with previously selected images.

To estimate the dissimilarity/diversity of each image in current set $X^{(v)} \in \mathbb{R}^{B\times N}$ to the previously selected images of the same category in the dataset, we propose a dissimilarity estimation approach, which is similar to sparse representative based classification methods [70], [79]. These methods aim at finding the class distribution of a sample over a learned dictionary of multiple classes. In contrast, our goal here is to find how diverse is a sample to a particular class, given some examples of the same class. Let, $D \in \mathbb{R}^{B\times M}$ is the feature matrix of the previously selected samples of currently considered category in the dataset, where $D = \{d_j \in \mathbb{R}^B, j = 1, \cdots, M\}$ and M denotes the number of images of the category.

In this regard, given a sample x_j, we compute its sparse representation c_j based on D. Then, we select the samples as diverse based on how the nonzero entries in the estimate c_j are associated with the columns of D. Given the above stated goals, the optimization problem can be written as,

$$\min_{C} \; ||X^{(v)} - DC||_F^2 + \lambda_D||X^{(t)} - DC||_F^2; \quad \text{s.t. } ||c_j||_1 \leq \kappa \tag{5}$$

Here, $C \in \mathbb{R}^{M\times N}$ is the sparse coefficient matrix, where $C = \{c_j \in \mathbb{R}^M, j = 1, \cdots, N\}$. λ_D and κ are tradeoff parameters.

In (5), the constraint, i.e., l_1 regularizer is to induce element-wise sparsity in a column. The objective is logical as any new sample of the same category will approximately lie in the linear span of some samples in dataset associated with the category. We require the coefficient matrix C to be sparse by solving the optimization program in (5).

After getting C, the diversity score for every image is calculated as follows,

$$k_j = \beta \frac{\|x_j^{(v)} - Dc_j\|_2}{\|x_j^{(v)}\|_2} + (1-\beta)\frac{\|x_j^{(t)} - Dc_j\|_2}{\|x_j^{(t)}\|_2} \tag{6}$$

Here, k_j is diversity score of jth sample. $\|x_j^{(v)} - Dc_j\|_2$ indicates the residual between $x_j^{(v)}$ and Dc_j, which is reconstruction of $x_j^{(v)}$ using samples of the same category from the dataset. β ($0 < \beta < 1$) determines the contribution of visual diversity and textual diversity in diversity score calculation.

Optimization. Here, we briefly describe the strategy to solve the optimization problem in (4) and (5). Using Lagrange multiplier, the optimization problem in (4) can be written as follows, where, λ_k is regularization parameter.

$$\min_{Z} \; ||X^{(v)} - X^{(v)}Z||_F^2 + \lambda||X^{(t)} - X^{(t)}Z||_F^2 + \lambda_k||KZ||_{2,1} \tag{7}$$

The problem (7) is equivalent to the following problem (8),

$$\min_{Z} \quad ||X^{(v)} - X^{(v)}Z||_F^2 + \lambda||X^{(t)} - X^{(t)}Z||_F^2 + \lambda_k||Z||_{K,2,1} \quad (8)$$

where, $||Z||_{K,2,1}$ denotes the weighted $l_{2,1}$ norm of Z and is defined as $||Z||_{K,2,1} = ||KZ||_{2,1}$

The objective function (8) is a convex weighted $l_{2,1}$ norm minimization problem which can be efficiently solved using Alternating Direction Method of Multipliers (ADMM) framework [3]. The ADMM procedure to solve (8) is summarized in Algo. 1.

For convergence, we compute the max norm of the difference between the approximation U and Z every iteration and when it is below a threshold (e.g., 10^{-7}), the optimization is considered to be converged. The approximation of Z on the last step is returned as the result of the computation. We also consider it converged if the algorithm run for the specified number of iterations (e.g., 1000).

Algorithm 1 An ADMM solver for (8)

1: **Input:** Feature Matrix $X^{(t)}$ and $X^{(v)}$, K, λ and $\mu > 0$, $Th = 10^{-7}$, $MaxIter$= 1000
2: **Initialization:** Initialize U, Z, Λ to zero.
3: **while** $(||U - Z||_{max} < Th)$ OR $(iter \leq MaxIter)$ **do**
4: $U \leftarrow (X^{(v)^T}X^{(v)} + \lambda X^{(t)^T}X^{(t)} + \mu I)^{-1}(X^{(v)^T}X^{(v)} + \lambda X^{(t)^T}X^{(t)} + \mu(Z - \Lambda/\mu))$;
5: $Z \leftarrow \max\{||U + \Lambda/\mu||_2 - \frac{\lambda_k K}{\mu}, 0\}\frac{U + \Lambda/\mu}{||U + \Lambda/\mu||_2}$;
6: $\Lambda \leftarrow \Lambda + \mu(U - Z)$;
7: $iter \leftarrow iter + 1$;
8: **end while**
9: **Output:** Sparse coefficient matrix Z.

We choose columns of $X^{(v)}$ corresponding to the nonzero rows of final Z and denote the feature matrix of the representative set as Y^*. Here, $Y^* \in \mathbb{R}^{B \times L}$ is the feature matrix for all images in representative set, where $Y^* = \{y_i^* \in \mathbb{R}^B, i = 1, \cdots, L\}$. y_i^* represents the feature descriptor of the ith representative sample in B-dimensional feature space. L denotes the number of images in the representative set.

We also use similar ADMM procedure stated above to solve the optimization problem in (5).

D. Image Labeling and Dataset Update

Active Learning for Image Labeling. After we have a diverse representative set Y^*, the next goal is to label each image in Y^* as relevant or not, as the selected images should represent the category. In other words, we want to remove irrelevant images and keep the relevant ones. In our proposed framework, we take the advantage of pool-based active learning scheme to label images, where given a pool of unlabeled examples $Y^* = \{y_1^*, y_2^*, ...\}$ and a fixed human labeling budget b, the learner chooses best samples from the pool to be labeled by a human. Our active learning algorithm is built upon SVM classifier and exploits the structure of the SVM to determine which images to label [64].

Now, the following questions remain: when should we ask a human and when the decision from classifier is reliable? For an instance $y_i^* \in Y^*$ when the score $f(y_i^*)$ is greater than a threshold δ, we assume that current model is highly confident about the instance. Here, $f(y_i^*)$ is the decision score function of the SVM classifier, which represents the distance of the instance y_i^* on to the separating hyperplane of SVM and the sign of $f(y_i^*)$ indicates the instance y_i^* belongs to the positive or the negative class. Hence, we label the instance y_i^* as positive or negative based on the sign of $f(y_i^*)$ and retain it for the incremental update. Number of instances obtained from the classifier is not fixed and depends on the value of δ, which we set sufficiently large so that irrelevant instances are less likely to be added to the dataset and used to update the current classifier model.

Among the remaining samples in Y^*, we follow an uncertainty sampling scheme [53] to find the samples $Y = \{y_j\}_{j=1}^b$ that the model is most uncertain about. Say, during a iteration, we can choose a set of samples Y to be labeled by a human, which is a subset of Y^*. This involves evaluating the informativeness of unlabeled instances, which are sampled from a given distribution. The most informative instance or the best query for human labeling $\hat{y_j}$ can be chosen based on the following condition:

$$\hat{y_j} = \min_{y_i^* \in Y^*} |f(y_i^*)| \quad (9)$$

Here, $f(y_i^*)$ represents the distance from the sample y_i^* to the separating hyperplane of SVM. The method is called minimum marginal hyperplane method, which assumes that the data with the smallest $f(y_i^*)$ value are those that the SVM is most uncertain about and hence, provide the greatest insight into the underlying data distribution. We remove at most b instances from Y^* using (9) and place in Y to be labeled by a human. We only ask a binary question to the human annotator : "Does this image belong to the category?".

To further decrease human labeling, We introduce an additional vision-language guidance for most uncertain samples Y from (9) based on the inter-modal similarity score between each image and its corresponding query text. More precisely, let, $y_j^{(v)}$ denote the image embedding and $y_j^{(t)}$ is the embedding of corresponding query text and ζ_j represent cosine similarity between them. If the similarity score ζ_j is greater than a threshold, we believe these examples are relevant and we don't need human labeling for these samples.

Incremental Update of SVM Models. Each newly labeled image instance y_n from the active learner are stored in a buffer with corresponding label q_n. When the buffer is full with a pool of labeled instances, $N_\ell = \{(y_n, q_n)\}_{n=1}^{\ell}$ from the active learner, the positive labeled samples from the buffer are added to the dataset. However, all of the ℓ instances in the buffer are used to incrementally update the SVM model. For updating the model, we need an incremental SVM solver, where the learning can be done in a batch framework. We use pegasos SVM solver [54] for this, which suits the requirement and is also effective in large-scale setting [54], [39]. The SVM model is updated after f iterations. At iteration $r+1$, the current SVM normal vector w_r is updated to w_{r+1} as follows,

$$w_{r+1} = (1 - \eta_r \nu)w_r + \frac{\eta_r}{\ell}\sum_{n \in N_\ell} \mathbb{1}[q_n\langle w_r, y_n\rangle < 1]q_n y_n \quad (10)$$

Here, $\mathbb{1}$[argument] is indicator function that returns a value of one, when the argument is true, and zero otherwise. ν is the regularization parameter that scales SVM primal objective function and step size, $\eta_r = 1/\nu t$. Finally, w_f is used to update the SVM model. We set the regularization parameter as 0.001 in SVM following [54].

Image Clustering. Due to increasing nature of our dataset, the number of selected images for a category may be very high over time. Comparing all of the previously selected images of a category for diversity calculation may be no longer possible, as it will induce significant computational load [1]. In such a case, we use a clustering algorithm to limit the maximum size of D to be used for diversity calculation. We use the sequential k-means [1] algorithm, as it is simple and efficient [47]. We select the maximum number of clusters to be 1000. This step is optional and may not be required based on available resources and number of images to be collected for a category.

E. Near-Duplicate Image Removal Across categories

Our approach ensures diverse/dissimilar images are collected for a category. The number of near duplicates in a category is likely to be very low. However, there may be duplicates across categories as we collected images for each category independently. Here, we apply a PCA based duplicate removal algorithm (Shown in Algo. 2) following [72], which can be applied to remove both intra-class and inter-class duplicates. We apply this duplicate image removal algorithm on the image dataset created by our approach. We use gist [45] feature and set p_t, g_t to 0.0005 and 0.02 respectively. Note that the algorithm does not depend on the choice of feature.

Algorithm 2 Near Duplicate Image Removal Across Category

Data: Feature Matrix for all images G, threshold for first principal component p_t, threshold for feature distance g_t;
Result: Index of images to delete;
Step 0: PCA on G to find 1st principal components ρ;
Step 1: Find candidate groups for near-duplicates:
for $i \in \{1, ..., \text{total no. of images}\}$ **do**
  for $j \in \{i+1, ..., \text{total no. of images}\}$ **do**
    if $(\rho(j) - \rho(i)) < p_t)$ **then**
      if $(\sqrt{(G(j,:) - G(i,:))^2} < g_t)$ **then**
        i and j is a candidate group of duplicate.
      end if
    end if
  end for
end for
Step 2: Find all members in a group of duplicates by a connected component algorithm [23];
Step 3: Keep the image with highest resolution from each group of duplicates, and select other indexes to delete.

IV. Experiments

To evaluate the effectiveness of our system, we follow [44], [75] and construct a dataset with 20 categories. We compare the image classification performance (Sec. IV-C), cross-dataset generalization ability (Sec. IV-D) and diversity (Sec. IV-E) of our dataset with several manually labeled and automated dataset construction methods. We also verify the performance of our system in enriching datasets with diverse examples (Sec. IV-F) and scalability in labeling (Sec. IV-G).

A. Image Dataset Construction

To evaluate our system, we constructed a dataset by collecting images from Google, Bing, and Flickr, which we name as Div20. Since many existing web-supervised dataset construction systems [44], [75] were evaluated on the PASCAL VOC 2012 categories [17], we focus on these categories as the target categories for the construction of Div20. The same categories have been utilized in prior dataset construction works [75], [44] and these basic categories also exist in most hand-labeled datasets (e.g., ImageNet). Hence, it allows us to compare against both existing hand-labeled datasets and dataset construction methods. The compared datasets have around 1000-1500 images per category. The number of collected images per category in Div20 ranges from 1201 to 1804, with an average of 1504. We considered no image related to the concept word

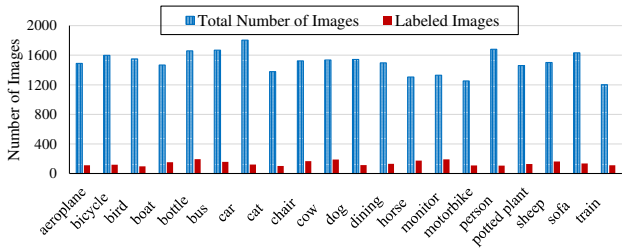


Fig. 4. Labeling for collecting images for different categories in Div20. The labeling effort was reduced more for categories like bird, bicycle, potted plant, whereas the labor was comparatively higher in monitor, cow, and bottle

is available beforehand in the construction of Div20. We take advantage of the high-precision of few top returned images for the query from Google search by utilizing them as the initial dataset. Our framework also allows enriching existing datasets with new examples, as shown in Section IV-F. In case of dataset enrichment, we start with images from a particular dataset as the initial set and collect 1K more images using our method. The ratio of human labeling used, compared to the total number of images in Div20 dataset is 9.28%. The average accuracy of the labels in Div20 is estimated by manually inspecting 1K images (50 random images per concept) from the entire dataset. The average accuracy has been found to be 96.8%, which is slightly lower than 99.7%, reported in ImageNet. However, for collecting the same number of images for any category, the manual labeling is more than 10 times lower in our case. Fig.4 shows human labeling statistics of 20 categories from our dataset for collecting images.

NUS-WIDE. We also conduct experiments on NUS-WIDE dataset [6] to evaluate label accuracies before and after applying our dataset construction system. NUS-WIDE has about 270K images and associated tags. Moreover, the dataset provides ground-truth on 81 labels, which allows us to evaluate label accuracies before and after dataset construction. The dataset is divided into two sets, i.e., development set (161,789

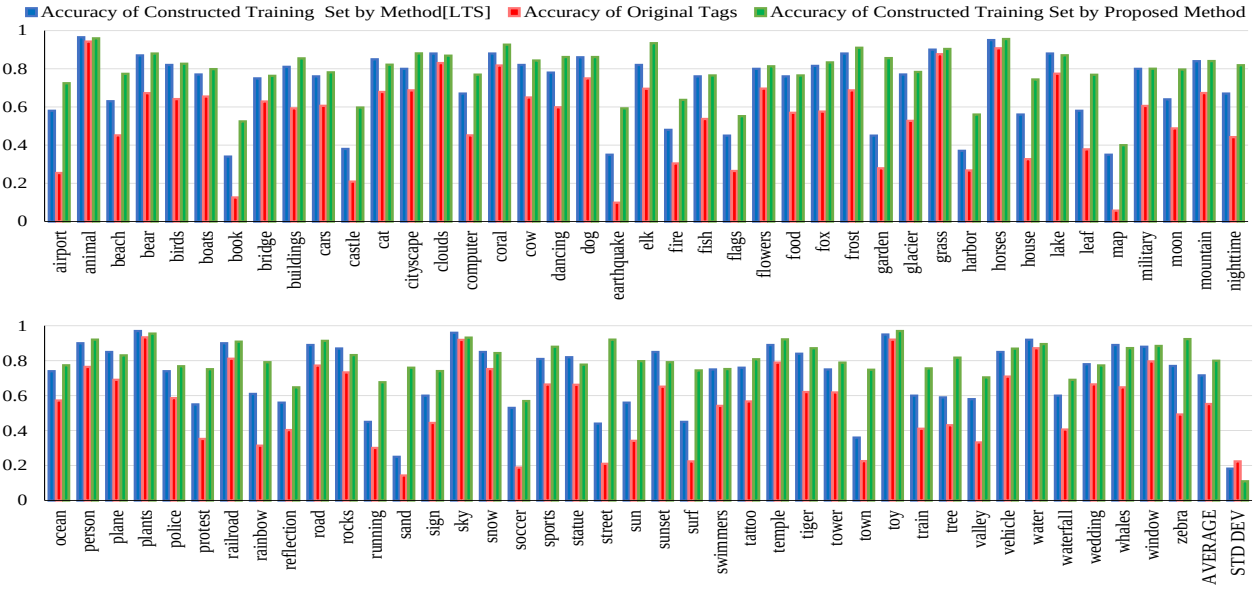


Fig. 5. Comparisons of the label precision before and after the dataset construction from NUS-WIDE. Proposed method improves average precision significantly.

images), and testing set (107,859 images). We follow the experimental setting of label-specific training set construction approach (LTS) [62] for a fair comparison, which also showed training set construction performance on NUS-WIDE. Similar to [62], we construct the training set from the development part utilizing our framework. We create the set with about 8% labeling similar to [62]. In Fig. 5, we compare accuracy of training set constructed by our framework against accuracy of tags in NUS-WIDE [6] and LTS [62].

We observe from the figure that after the training set construction, the accuracies of the labels improve significantly compared to the initial tags of NUS-WIDE and training set construction method LTS [62]. Specifically, our method improves average precision by about 34% from the average precision of tags [6] and by about 12% from LTS [62].

B. Experimental Setup.

For fair comparison across datasets, we have used the same pre-trained AlexNet CNN [33] for feature extraction from all the compared datasets in the experiments. For all the experiments other than image classification performance on trained CNN in Sec. IV-C, we trained one versus all SVM classifiers and we set the same options for all the datasets. The type of kernel for SVM was set as a radial basis function. We used all images for a category as positive examples and 200 randomly sampled image per category from all other categories as negative examples for training SVM models. When we train and test on the same dataset (e.g., train on VOC and test on VOC), we used five-fold cross-validation to test the classifier performance (80% training, 20% testing). When we train on one dataset and test on another dataset (e.g., train on ImageNet, test on VOC), we used all images from the dataset to train the classifier model and test on all images of the other dataset. When we test SVM classifier models, we consider a sample as correctly classified if the score (signed distance from the sample to the decision boundary) is more than 0.25.

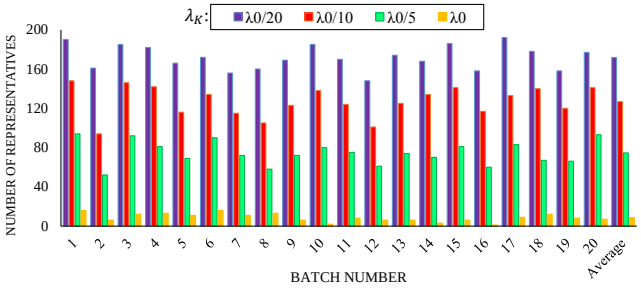
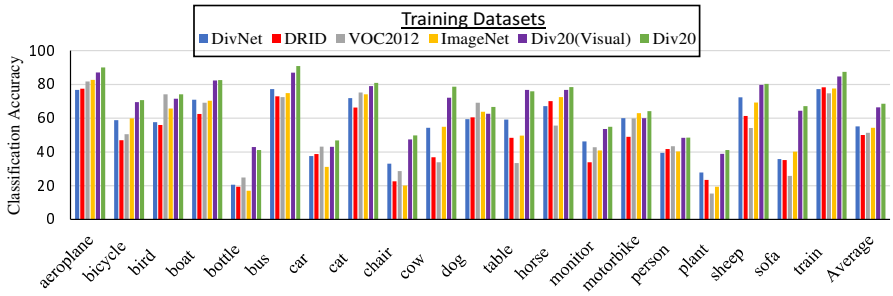


Fig. 6. Number of diverse representatives selected in 20 batches (batch size: 200) of aeroplane class for different value of regularization parameter λ_k ($\lambda_k = \lambda_0/\mu_k$, where $\mu_k > 1$ and λ_0 is computed from the input data [14]). We observe that the number of selected representatives increases with decreasing value of λ_k, or increasing value of μ_k.

Parameters: There are two main parameters in solving ADMM for representative subset selection, i.e., λ and λ_k. $\lambda(0 \leq \lambda \leq 1)$ determines the weight of textual information in subset selection. When λ is set as zero, visual feature is only considered in subset selection. When λ is set as one, textual feature has equal weight to visual feature in subset selection. In Div20 construction, initially, we process batches from a search engine (Google and Bing). As there is no tag associated with these images, we set λ as zero. Then we set $\lambda = 1$ as we process Flickr images which usually has associated tags. λ_k is the regularization parameter. We set the Lagrangian multipliers $\lambda_k = \lambda_0/\mu_k$, where $\mu_k > 1$ and λ_0 is computed from the input data [14]. The sensitivity of λ_k on selecting the number of representatives from first 20 batches of aeroplane class is shown in Fig. 6. From Fig. 6, we see that when μ_k is set as 10, about 60% images are retained as representative. When we set μ_k as 20, about 85% images are selected and about 35% images are selected for setting μ_k as 20. We empirically found setting $\mu_k = 10$ is suitable for collecting diverse representative images in our case.

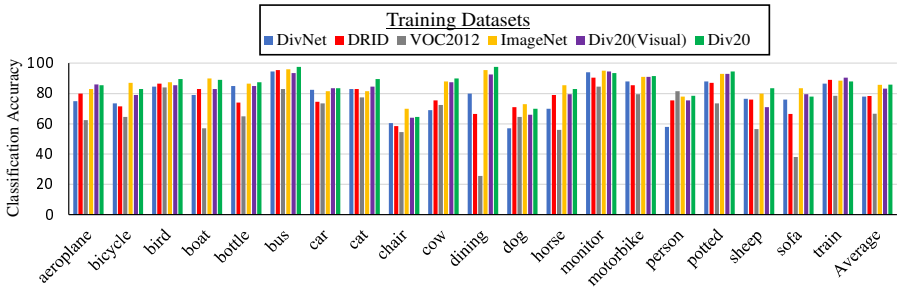
There are three main parameters in diversity score calculation. λ_D determines the weight of textual information



Average Accuracy on VOC2012

Training Dataset	Average Accuracy
DivNet	55.20
DRID	50.13
VOC2012	51.43
ImageNet	54.41
Div20(Visual)	66.42
Div20	68.54

Fig. 7. Image Classification Performance on VOC 2012 dataset, for classifiers trained on different datasets: (a) ImageNet, (b) VOC2012, (c) DRID, (d) DivNet (e) Div20(Visual), (f) Div20. Here, the same categories between datasets are compared by training one versus all SVM classifiers (as discussed in Sec IV-B). When we train on VOC, we used five-fold cross-validation to test the classifier performance (80% training, 20% testing). When we train on any other datasets (e.g., ImageNet, DRID, DivNet, Div20(Visual), Div20), we use all images from a category to train and test on all images of the same category in VOC. The accuracy per category is shown on the left plot and the average accuracy is shown in the right table. Best viewed in color.



Average Accuracy on ImageNet

CNN Training Dataset	Average Accuracy
DivNet	78.03
DRID	78.43
VOC2012	66.60
ImageNet	85.70
Div20(Visual)	83.23
Div20	85.88

Fig. 8. Image Classification Performance of Trained CNNs on ImageNet dataset. We randomly choose 80% images per category from ImageNet as training image set for training ImageNet CNN and remaining 20% as testing image set for this experiment. For other datasets (e.g., VOC2012, Div20, DRID, DivNet), we have used all images from the datasets to train CNN models and tested on testing image set of ImageNet dataset. Best viewed in color.

in computing sparse representation C based on previously selected examples D. We set λ_D as 1 as we provide equal weight to the textual information. In case of processing batches from Google and Bing, we set $\lambda = 0$ as we do not collect any text associated with these images. The regularization parameter of this experiment is calculated from data [14]. The regularization parameter affects the number of examples from D will be used to compute the sparse reconstruction. We set this regularization parameter as 5. A higher value will allow reconstructing a sample from a higher number of previously selected samples in the dataset, whereas a lower value will force to reconstruct from a smaller number of previously selected samples. The diversity score parameter β ($0 < \beta < 1$) determines the contribution of visual diversity and textual diversity in diversity score calculation. We set β as 0.5 so that the contribution is the same for textual and visual information.

Baselines: In order to validate the performance of our dataset, we compare with several baselines that fall into two main categories: (1) manually labeled datasets such as VOC2012 [17], Caltech-256 [22] and ImageNet [50], (2) dataset constructed using two recent methods, i.e., DRID [75] and our previous work, Divnet [44]. The VOC2012 [17] dataset has 11,530 images of 20 object categories. Each training image has an annotation file giving a bounding box and object class label for each object in one of the twenty classes present in the image. The Caltech-256 [22] dataset consists of 30,607 images covering 256 categories, where a minimum number of images per category is 80. ImageNet [50] is an image dataset organized according to the WordNet hierarchy. It provides an average of 1000 images to illustrate each category. DRID is a dataset construction method, which uses Multiple Instance Learning to filter noisy images and select representative images for the dataset. DivNet is our previous work for construction of diverse image dataset, which uses a sparse coding based approach to incrementally select representative and diverse images and filters irrelevant images. In order to quantify the role of tags associated with web images contributing to the final results, we also build Div20(Visual) dataset considering only visual features. Div20(Visual) resembles our previous system [44] with incremental SVM-based active learning scheme, instead of the sparse reconstruction based active learning scheme utilized in the previous work.

C. Image Classification

The goal of this experiment is to compare the performance of classifiers trained on Div20 images with classifiers trained on other baseline datasets. We select VOC 2012 as the testing benchmark dataset for this experiment. In this experiment, we compare our dataset with hand-labeled image datasets, e.g., VOC2012 [17] and ImageNet[50] and automated datasets, e.g. DivNet [44], DRID [75] and Div20(Visual). The performance of classifiers trained on a dataset created by the proposed method and other datasets are shown in Fig. 7. Div20 outperforms the second best baseline Div20(Visual) in terms of accuracy by an average of 3.2% across categories, with a maximum of 9% in the cow category. Moreover, Div20 shows significant performance improvement (average of 19.8%) over our previous work DivNet. We believe this is due to the fact that Div20 dataset, being constructed by

using both visual and textual features, has more visually and semantically diverse images than other datasets. Div20 shows better image classification performance in other datasets too. The results of image classification in other datasets can be found in the supplementary material.

Classification Performance of Trained CNNs: We trained one versus all SVM classifiers to evaluate image classification performance following previous works [75], [44]. We also believe that it is important to understand the performance of a dataset by training convolutional neural networks. We trained deep CNNs utilizing our dataset and other compared datasets. The total number of images for 20 categories in these datasets are not enough to train a deep CNN from scratch. Hence, we start with pre-trained CaffeNet CNN [24] and fine-tune the layers. We start training with a learning rate of 0.001 and decrease it when the training loss had reached a plateau. We used stochastic gradient descent in training the model. The classification performance of the CNN models on ImageNet dataset is shown below in Fig. 8. We randomly choose 80% images per category from ImageNet as training image set for training ImageNet CNN and remaining 20% as the testing image set for this experiment.

It is evident from Fig. 8 that the average accuracy of the CNN trained on Div20 is higher than CNNs trained on other datasets. CNN trained on Div20 achieves slightly higher accuracy than CNN trained on ImageNet CNN and significantly higher accuracy than CNNs trained on other datasets. It is also evident from Fig. 8 that the Div20 CNN consistently shows high performance across categories. We also observe a similar performance trend in other datasets. The classification performance on VOC2012 dataset is provided in the supplementary material.

Ablation Study: In this experiment, we focus on observing the effects of the three most significant parts (described in Sec.III-B, Sec.III-C, and Sec.III-D) of our proposed system on the performance. In this regard, we perform an ablation study of our system in Fig. 9 by removing one part of the system at a time and constructing a dataset with average 1500 images per category. We observe how that impacts classification performance evaluation on VOC2012 and ImageNet datasets in several categories. The main observations from Fig. 9 can be summarized as follows.

• Effect of Image-Text Embedding- In order to quantify the role of jointly utilizing both image and text modality for dataset construction (Sec.III-B), we can compare purple columns and green columns in Fig. 9. Div20(Visual Only) dataset is created considering only visual features. From Fig. 9, it is evident that jointly utilizing both image and text in Div20(Proposed) contributes to overall performance improvement over Div20(Visual Only) in almost all categories across datasets. In ImageNet evaluation, the classifier trained on Div20 (Proposed) outperforms Div20 (Visual) in terms of average accuracy by a relative improvement of about 4.65%, with a maximum improvement of about 31% in the chair category. We observe similar improvement in VOC2012 evaluation, where the average accuracy improved from 66.42% in Div20(Visual Only) to 68.54% in Div20(Proposed) and the maximum relative improvement is 9% in the cow category.

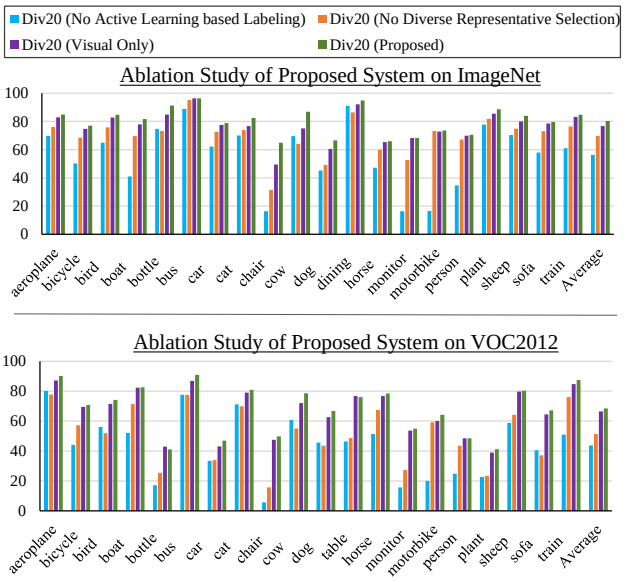
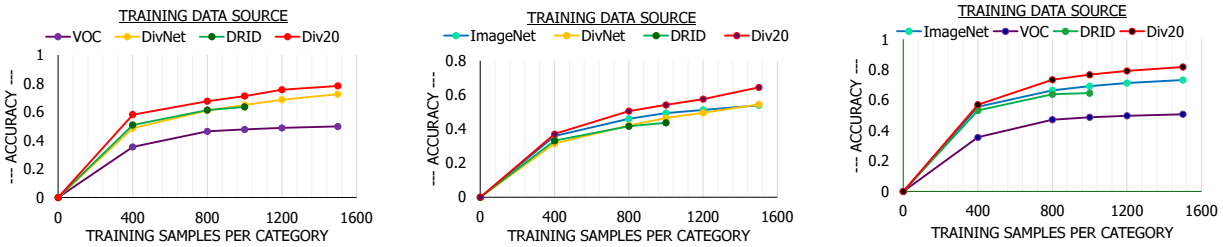


Fig. 9. Ablation analysis of proposed system to evaluate relative importance of different parts of the system

• Effect of Diverse Representative Selection- We can compare orange columns and green columns in Fig. 9 in order to quantify the role of diverse representative selection step shown in Sec.III-C. Div20(No Diverse Representative Selection) represents constructed dataset, where Sec.III-C has been omitted and Sec.III-D (Image Labeling and Dataset Update) step is performed after Sec.III-B (feature extraction). From Fig. 9, we observe Div20 shows significant performance improvement over Div20(No Diverse Representative Selection) as expected. Diverse representative selection step effectively minimizes the chance of including redundant instances in the dataset and allows to better utilize human effort in labeling distinctive instances. When diverse representative selection step is not utilized, the average performance drops significantly from 68.54% to 51.28% in VOC2012, and from 80.32% to 69.79% in ImageNet evaluation. We also observe large performance drop in most categories (e.g., drop from 41.20% to 23.42% in plant category in VOC2012).

• Effect of Active Learning based labeling- We compare blue columns and green columns in Fig. 9 to analyze the effect of incremental SVM based active learning scheme for actively selecting samples (Sec.III-D) in our system. Div20(No Active Learning based Labeling) represents constructed dataset where Sec.III-D has been omitted in the system. Without this part, the system mainly loses its ability to select correct examples to update the dataset. We again observe that classifiers trained using images from Div20 performs consistently better across categories in both datasets, with average improvement about 56% in VOC2012 and 43% in ImageNet. Div20(No Active Learning based Labeling) shows high variance in performance across categories as it relies mostly on the accuracy of tags.

We observe from the above analysis that all three parts help to achieve consistent and significant improvement in classification performance across categories on different datasets. We do not report the impact of near-duplicate image removal step



(a) Test on ImageNet (b) Test on VOC2012 (c) Test on DivNet

Fig. 10. Cross-dataset performance of classifier trained on different datasets with a different number of samples per category. Cross-dataset generalization ability of classifiers learned from ImageNet, VOC2012, DivNet, DRID and Div20, and then tested on: (a) ImageNet, (b) VOC2012, (c) DivNet. We sequentially select [400, 800, 1200, 1500] images per category from different datasets as positive training examples. In this experiment, we calculate accuracy for each category by training one versus all SVM classifiers. In the figure, the average classification accuracy is reported, which represents the cross-dataset generalization ability of one dataset on another dataset. Best viewed in color.

(Sec.III-E) in this experiment, as the number of near duplicates in a category is very low in our dataset construction and hence, Sec.III-E has almost no impact on the classification performance. The main purpose of Sec.III-E is removing duplicate images across categories as we collect images for each category independently.

D. Cross-Dataset Generalization

To evaluate the generalization ability of our constructed dataset, we compare the dataset with VOC2012 [17], ImageNet [50], DivNet [44], DRID [75] and DivNet. We select all twenty categories from VOC2012 dataset [17] for this experiment. The result for different training and testing data combinations is shown in Fig. 10. As seen from the figures, training with Div20 shows the best generalization among datasets, as the average accuracy is high and cross-dataset performance drop is minimum for Div20. Initially, with few training samples, the performance of Div20 may be lower as it has very few labeled samples. However, as we iteratively select more diverse images to be labeled by our system, the performance improves at a higher rate. We can compare the performance of the datasets at the point of 1000 training samples (since state of the art ImageNet has on average 1000 images per category) and see the generalization ability of Div20 is better. The comparison at the same number of training samples shows the average cross-dataset performance of Div20 is significantly higher than other baselines. Moreover, Div20 can achieve even better performance because of its ability to scale up with limited labeling effort.

E. Diversity

In order to illustrate the diversity of images in our collected dataset, we follow [10], [7], which computes the average image of each category and measure lossless JPG file size, which reflects the amount of information in an image. A diverse image set should result in a blurrier average image, and the JPG file size of the average image should be smaller. We re-size all images to 256×256, and create average images for each category from all images of the category. Fig. 11 shows the average images and the corresponding JPG image size comparison of four categories: person, dog, monitor, and aeroplane. The average image of Div20 is blurrier and harder to recognize the object than the average image of other datasets, which are comparatively more structured and sharper than Div20. Div20 has smaller JPG file size than ImageNet, VOC, DRID, DivNet, and Caltech-256. This phenomenon is common for almost all of the categories. For randomly picked 8 categories, the average loss-less JPG size has been found to be 2.2 KB in Div20, 2.8 KB in DRID, 2.6 kB in DivNet, 2.5 KB in ImageNet and 3.5 KB in Caltech-256.

F. Dataset Enrichment

We enrich ImageNet, VOC2012, and Caltech-256 by proposed system to evaluate the performance of our approach in dataset enrichment. For this experiment, we pick eight categories that are common in these datasets: *airplane, bike, bird, car, dog, horse, monitor, and person*. For each category, we start our dataset construction method with images from a particular dataset as the initial dataset and collect 1000 more images using our framework automatically with no labeling (Enrich Dataset-Auto) and also with a manual labeling budget of 50 (Enrich Dataset-50). We train one versus all SVM classifiers (as described in Section IV-B) for each category with initial dataset images and enriched dataset images, and test on Div20. The result in Fig. 12 shows the performance of classifier improves significantly after enriching with our framework. Hence, the proposed method is suitable for extending existing image collections with diverse examples.

G. Scalability

Runtime. A major advantage of our system is creating a high-quality dataset with low cost. Here, we compare the runtime of our approach to complete manual labeling based approaches. Our initial pool consists of around 6000 candidate images per category for 20 categories. According to [18], the human labeling speed is 2 images per second and on average 3 annotators verify one image. Hence, the approximate time for creating a dataset from our collected images with compete manual labeling can be calculated as follows:

$$\text{Total Time} = \frac{6000 \times 20 \times 3}{2}\text{sec} = 3{,}000 \text{ min} = 50 \text{ hours}$$

On the other hand, our approach considers both images in the current batch and previously selected images to select the best subset for further processing. Thus, we can utilize human effort in labeling most informative and distinctive instances based on our labeling budget. In our pipeline, we used a batch

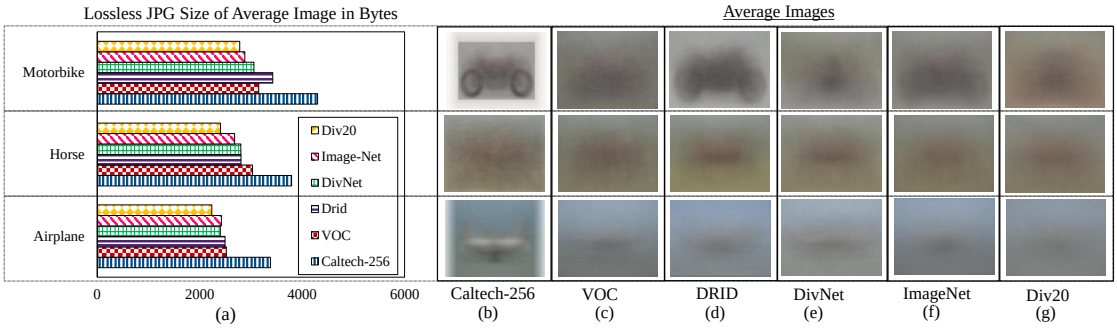


Fig. 11. (a) Lossless JPG file sizes of average images in Bytes for four different categories in Div20, ImageNet, DivNet, DRID, VOC and Caltech-256. (b)-(g) shows average images for each category in different datasets indicated in (a). We resize all images to 256×256, and create average images for each category from all images of the category. We measure the lossless JPG file size of the average image, which reflects the amount of information in an image. A diverse image set should result in a blurrier average image, and the JPG file size of the average image should be smaller.

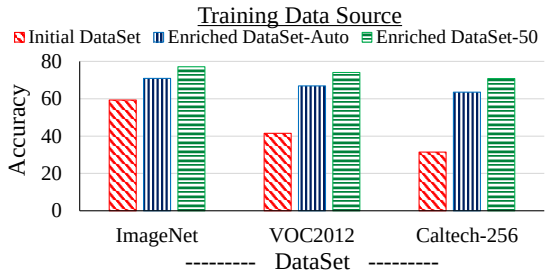


Fig. 12. Plot shows the change in image classifier performance after enriching datasets with our framework. We utilized the proposed approach to collect 1000 more images per category and created Enriched Dataset-Auto (automatic, no manual labeling) and Enriched Dataset-50 (with 50 manual labeling budget). We compare the image classification performance of enriched datasets with that of initial datasets on Div20 dataset.

size of 200 and set the maximum manual labeling budget for each batch as 10, which we found as sufficient to achieve good performance. Considering the same average human labeling speed and number of annotators as the previous step, the time required for labeling in our approach can be calculated as:

$$\text{Labeling Time} = \frac{6000 \times 20 \times 10 \times 3}{200 \times 2}\text{sec} = 150 \text{ min} = 2.5 \text{ hours}$$

Note that, this is the upper bound of our required manual labeling time. We select a smaller diverse representative set from each batch (Sec. III-C) and then utilize active learning (Sec. III-D) to select the samples for labeling. Hence, we find that many times the number of samples to be labeled is less than the labeling budget, which happens regularly after we process a number of batches. We ran our experiment on a core i7 CPU with 16 GB RAM. The total time required for the algorithm without manual labeling was about 93 minutes (the average time per batch for diversity calculation optimization was about 2.71 seconds, the subset selection optimization was about 2.189 seconds and SVM optimization was 3.92 seconds). The time for feature extraction (with a k40 GPU) was around 156 minutes. Hence, the total time in our approach is :

$$\text{Total Time} = (150 + 93 + 156) = 399 \text{ minutes} = 6.65 \text{ hours}$$

Our approach decreases the total runtime by around 7.5 times and labeling cost by around 20 times compared to the completely manual approaches. Note that, category-wise parallelization is not considered in this experiment. As collecting images per category is independent of other categories, the required time for dataset creation can be reduced significantly. It can be further reduced using ADMM optimization parallelization [46]. We leave this as one of our future work.

Performance Change with Human Labeling. Different from static dataset construction, our method can be used to dynamically update datasets. It is possible to collect images based on desired dataset size and labeling budget. Such property makes sense as one user may be interested in collecting more image for a category, compared to others. It is also likely that a user may want to spend more time labeling images from a particular category than other categories. We investigate the scalability in labeling by collecting a fixed number of images with different labeling budget. The accuracy of the classifier per category increases by 3.4% on average initially, as we increase labeling budget by 25. However, the performance improvement usually saturates after labeling around 100-200 examples(the actual number varies by category). The classifier performance in car, chair, bottle, and bicycle category for different labeling budget, with a fixed number of images collected, are shown in Fig.13. The accuracy of the classifier increases with increasing labeling. However, it is apparent from Fig.13 that, the performance improvement saturates after a limited labeling effort.

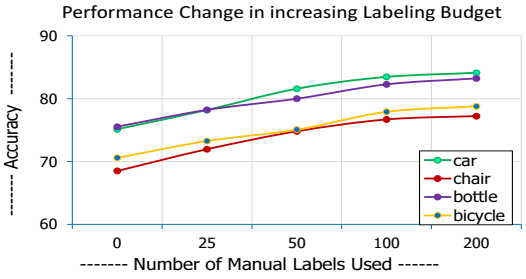


Fig. 13. The plot shows the performance change in image classifier performance with an increase in manual labeling, with a fixed number of images collected. We utilize the proposed approach with different labeling budget and collect images for 4 different categories. We again train one versus all SVM classifiers for each category (as described in Section IV-B)

V. Conclusions

In this paper, we propose a sparse coding based framework that employs a joint visual semantic space to simultaneously utilize both images and associated textual information from

web collections for continuously collecting diverse images from the web, which is suitable for dataset construction, or enriching existing datasets with new examples. Our system provides a flexibility that permits filtering out irrelevant images and obtains a reliable set of diverse images based on resource and labeling budget available so that a high-precision large-scale image classifier can be trained. The experimental results demonstrate that our system is not only useful in reducing the manual annotation efforts, but also successful in collecting images with high precision and diversity, and robust image classifiers can be trained from these images.

ACKNOWLEDGMENT

This work was partially supported by NSF grants IIS-1746031 and CNS-1544969. We gratefully acknowledge the support of NVIDIA with the donation of a Tesla K40 GPU used for this research.

REFERENCES

[1] M. Ackerman and S. Dasgupta. Incremental clustering: The case for extra clusters. In *NIPS*, 2014.
[2] Y. Bai, K. Yang, W. Yu, C. Xu, W.-Y. Ma, and T. Zhao. Automatic image dataset construction from click-through logs using deep neural network. In *ACM Multimedia*, 2015.
[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 2011.
[4] D. S. Cheng, F. Setti, N. Zeni, R. Ferrario, and M. Cristani. Semantically-driven automatic creation of training sets for object recognition. *CVIU*, 2015.
[5] J. Choi, T.-H. Oh, and I. S. Kweon. Textually customized video summaries. *arXiv preprint arXiv:1702.01528*, 2017.
[6] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *ACM CIVR*, 2009.
[7] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. In *ECCV*. 2008.
[8] Y. Cong, J. Yuan, and J. Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *TMM*, 2012.
[9] Y. Cui, F. Zhou, Y. Lin, and S. Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. *arXiv preprint arXiv:1512.05227*, 2015.
[10] 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.
[11] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014.
[12] F. Dornaika and I. K. Aldine. Instance selection using non-linear sparse modeling. *TCSVT*, 2017.
[13] E. Elhamifar, G. Sapiro, and S. S. Sastry. Dissimilarity-based sparse subset selection. *TPAMI*, 2016.
[14] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *CVPR*, 2012.
[15] E. Elhamifar, G. Sapiro, A. Yang, and S. Sasrty. A convex optimization framework for active learning. In *ICCV*, 2013.
[16] B. Emond. Multimedia and human-in-the-loop: interaction as content enrichment. In *Int. Workshop Human-Centered Multimedia*, 2007.
[17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.
[18] L. Fei-Fei. Imagenet: crowdsourcing, benchmarking & other cool things. In *CMU VASC Seminar*, 2010.
[19] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, 2005.
[20] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.
[21] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu. Visual-textual joint relevance learning for tag-based social image search. *TIP*, 2013.
[22] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.
[23] J. Hopcroft and R. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM*, 1973.
[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014.
[25] J. Johnson, L. Ballan, and L. Fei-Fei. Love thy neighbors: Image annotation by exploiting image metadata. In *ICCV*, 2015.
[26] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.
[27] T. Kenter, A. Borisov, and M. de Rijke. Siamese cbow: Optimizing word embeddings for sentence representations. *arXiv preprint arXiv:1606.04640*, 2016.
[28] T. Kenter and M. de Rijke. Short text similarity with word embeddings. In *CIKM*, 2015.
[29] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*. 2012.
[30] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.
[31] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015.
[32] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. *arXiv preprint arXiv:1511.06789*, 2015.
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
[34] L.-J. Li and L. Fei-Fei. Optimol: automatic online picture collection via incremental model learning. *IJCV*, 2010.
[35] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. Snoek, and A. D. Bimbo. Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. *ACM Computing Surveys*, 2016.
[36] 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 *ECCV*, 2014.
[37] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *TIP*, 2008.
[38] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.
[39] A. K. Menon. Large-scale support vector machines: algorithms and theory. *University of California, San Diego*, 2009.
[40] E. Mezuman and Y. Weiss. Learning about canonical views from internet image collections. In *NIPS*, 2012.
[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint:1301.3781*, 2013.
[42] N. C. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ACM ICMR*, 2018.
[43] N. C. Mithun, R. Panda, E. E. Papalexakis, and A. K. Roy-Chowdhury. Webly supervised joint embedding for cross-modal image-text retrieval. In *ACM Multimedia*, 2018.
[44] N. C. Mithun, R. Panda, and A. K. Roy-Chowdhury. Generating diverse image datasets with limited labeling. In *ACM Multimedia*, 2016.
[45] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.
[46] Z. Peng, M. Yan, and W. Yin. Parallel and distributed sparse optimization. In *Asilomar Conf. Signals Systems Computers*, 2013.
[47] D. T. Pham, S. S. Dimov, and C. Nguyen. An incremental k-means algorithm. *Proc. the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2004.
[48] B. Plummer, M. Brown, and S. Lazebnik. Enhancing video summarization via vision-language embedding. In *CVPR*, 2017.
[49] Z. Qian, P. Zhong, and R. Wang. Tag refinement for user-contributed images via graph learning and nonnegative tensor factorization. *IEEE Signal Processing Letters*, 2015.
[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.
[51] J. Sang, J. Liu, and C. Xu. Exploiting user information for image tag refinement. In *ACM Multimedia*, 2011.
[52] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. *TPAMI*, 2011.
[53] B. Settles. Active learning literature survey. *Univ. of Wisconsin, Madison*, 2010.
[54] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 2011.
[55] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[56] R. Speer and C. Havasi. Conceptnet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*. 2013.
[57] T.-F. Su, C.-K. Chiang, and S.-H. Lai. A multiattribute sparse coding approach for action recognition from a single unknown viewpoint.

TCSVT, 2016.
[58] J. Tang, Q. Chen, M. Wang, S. Yan, T.-S. Chua, and R. Jain. Towards optimizing human labeling for interactive image tagging. *ACM TOMM-CAP*, 2013.
[59] J. Tang, X.-S. Hua, Y. Song, T. Mei, and X. Wu. Optimizing training set construction for video semantic classification. *EURASIP Journal Adv. Signal Processing*, 2008.
[60] J. Tang, X. Shu, Z. Li, Y.-G. Jiang, and Q. Tian. Social anchor-unit graph regularized tensor completion for large-scale image retagging. *arXiv preprint arXiv:1804.04397*, 2018.
[61] J. Tang, X. Shu, G.-J. Qi, Z. Li, M. Wang, S. Yan, and R. Jain. Tri-clustered tensor completion for social-aware image tag refinement. *TPAMI*, 2017.
[62] J. Tang, S. Yan, C. Zhao, T.-S. Chua, and R. Jain. Label-specific training set construction from web resource for image annotation. *Signal Processing*, 2013.
[63] D. Tao, J. Cheng, X. Gao, X. Li, and C. Deng. Robust sparse coding for mobile image labeling on the cloud. *TCSVT*, 2017.
[64] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *JMLR*, 2002.
[65] A. Torralba, A. Efros, et al. Unbiased look at dataset bias. In *CVPR*, 2011.
[66] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *ICLR*, 2016.
[67] S. Vijayanarasimhan, P. Jain, and K. Grauman. Far-sighted active learning on a budget for image and video recognition. In *CVPR*, 2010.
[68] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. Cost-effective active learning for deep image classification. *TCSVT*, 2017.
[69] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016.
[70] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 2009.
[71] Y. Xia, X. Cao, F. Wen, and J. Sun. Well begun is half done: Generating high-quality seeds for automatic image dataset construction from web. In *ECCV*. 2014.
[72] J. Xiao. Professor x toolkit. https://github.com/jianxiongxiao/ProfXkit, 2016.
[73] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, et al. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.
[74] Y. Yao, J. Zhang, F. Shen, X. Hua, J. Xu, and Z. Tang. A new web-supervised method for image dataset constructions. *Neurocomputing*, 2017.
[75] Y. Yao, J. Zhang, F. Shen, X.-S. Hua, J. Xu, and Z. Tang. Exploiting web images for dataset construction: A domain robust approach. *TMM*, 2017.
[76] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
[77] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632*, 2014.
[78] X. Zhai, Y. Peng, and J. Xiao. Learning cross-media joint representation with sparse and semisupervised regularization. *TCSVT*, 2014.
[79] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai. Graph regularized sparse coding for image representation. *TIP*, 2011.
[80] G. Zhu, S. Yan, and Y. Ma. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *ACM Multimedia*, 2010.
[81] S. Zhu, C.-W. Ngo, and Y.-G. Jiang. Sampling and ontologically pooling web images for visual concept learning. *TMM*, 2012.
[82] X. Zhu, W. Nejdl, and M. Georgescu. An adaptive teleportation random walk model for learning social tag relevance. In *ACM SIGIR*, 2014.

Niluthpol Chowdhury Mithun received his Bachelors and Masters degree in 2011 and 2014 respectively from Bangladesh University of Engineering and Technology. He is currently pursuing the Ph.D. degree in the department of Electrical and Computer Engineering at University of California, Riverside. His broad research interest includes computer vision and machine learning with more focus on weakly supervised learning, multimodal data analysis, vision-based localization, deep learning etc.

Rameswar Panda graduated from University of California, Riverside with a Ph.D. in Electrical and Computer Engineering in 2018. Previously, he received his Bachelors and Masters degree from Biju Patanaik University of Technology, India and Jadavpur University, India. He is currently a researcher at IBM Research AI (MIT-IBM Watson AI Lab). His main research interests include computer vision, machine learning, video summarization, person re-identification and multimedia.

Amit K. Roy-Chowdhury received the Bachelors degree in Electrical Engineering from Jadavpur University, Calcutta, India, the Masters degree in Systems Science and Automation from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree in Electrical and Computer Engineering from the University of Maryland, College Park. He is a Professor of Electrical and Computer Engineering and a Cooperating Faculty in the Department of Computer Science and Engineering, University of California, Riverside. His broad research interests include computer vision, image processing, and vision-based statistical learning, with applications in cyber-physical, autonomous and intelligent systems. He is a coauthor of two books: Camera Networks: The Acquisition and Analysis of Videos over Wide Areas, and Recognition of Humans and Their Activities Using Video. He is the editor of the book Distributed Video Sensor Networks. He has been on the organizing and program committees of multiple computer vision and image processing conferences and is serving on the editorial boards of multiple journals. He is a Fellow of the IEEE and IAPR