Learning Visual Features from Large Weakly Supervised Data

Armand Joulin*, Laurens van der Maaten*, Allan Jabri, and Nicolas Vasilache

Facebook AI Research
{ajoulin,lvdmaaten,ajabri,ntv}@fb.com

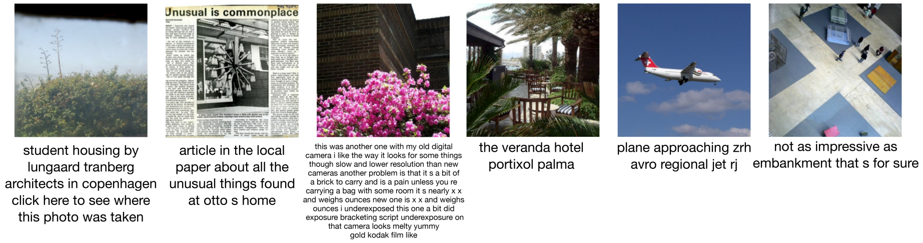
Abstract. Convolutional networks trained on large supervised datasets produce visual features which form the basis for the state-of-the-art in many computer-vision problems. Further improvements of these visual features will likely require even larger manually labeled data sets, which severely limits the pace at which progress can be made. In this paper, we explore the potential of leveraging massive, weakly-labeled image collections for learning good visual features. We train convolutional networks on a dataset of 100 million Flickr photos and comments, and show that these networks produce features that perform well in a range of vision problems. We also show that the networks appropriately capture word similarity and learn correspondences between different languages.

1 Introduction

Recent studies have shown that using visual features extracted from convolutional networks trained on large object recognition datasets [22, 33, 53, 56] can lead to state-of-the-art results on many vision problems including fine-grained classification [27, 50], object detection [17], and segmentation [47]. The success of these networks has been largely fueled by the development of large, manually annotated datasets such as Imagenet [9]. This suggests that to further improve the quality of visual features, convolutional networks should be trained on even larger datasets. This begs the question whether fully supervised approaches are the right way forward to learning better vision models. In particular, the manual annotation of ever larger image datasets is very time-consuming[1], which makes it a non-scalable solution to improving recognition performances. Moreover, manually selecting and annotating images often introduces a strong bias towards a specific task [48, 58]. Another problem of fully supervised approaches is that they appear rather inefficient compared to how humans learn to recognize objects: unsupervised and weakly supervised learning plays an important role in human vision [11], as a result of which humans do not need to see thousands of images of, say, chairs to obtain a good grasp of what a chair looks like.

[*] Both authors contributed equally.

[1] For instance, the development of the COCO dataset [36] took more than 20,000 annotator hours spread out over two years.



student housing by lungaard tranberg architects in copenhagen click here to see where this photo was taken

article in the local paper about all the unusual things found at otto s home

this was another one with my old digital camera i like the way it looks for some things though slow and lower resolution than new cameras another problem is that it s a bit of a brick to carry and is a pain unless you re carrying a bag with some room it s nearly x x and weighs ounces new one is x x and weighs ounces i underexposed this one a bit did exposure bracketing script underexposure on that camera looks melty yummy gold kodak film like

the veranda hotel portixol palma

plane approaching zrh avro regional jet rj

not as impressive as embankment that s for sure

Fig. 1. Six randomly picked photos from the YFCC100M dataset and the corresponding comments we used as targets for training.

In this paper, we depart from the fully supervised learning paradigm and ask the question: *can we learn high-quality visual features from scratch without using any fully supervised data?* We perform a series of experiments in which we train models on a large collection of photos and comments associated with those photos. This type of data is available in great abundance on photo-sharing websites: specifically, we use the publicly available YFCC100M dataset that contains 100 million Flickr photos and comments [57]. Figure 1 displays six randomly picked Flickr photos and corresponding comments. Indeed, many of the comments do not describe the contents of the photos (that is, the comments are not captions or descriptions), but the comments do carry weak information on the image content. Learning visual representations from such weakly supervised data has three potential advantages: (1) there is a near-infinite amount of weakly supervised data available[2], (2) the training data is not biased towards solving a specific task, and (3) it is more similar to how humans learn to solve vision.

We present experiments showing that convolutional networks can learn to identify words that are relevant to a particular image, despite being trained on the very noisy targets of Figure 1. In particular, our experiments show that the visual features learned by weakly-supervised models are as good as those learned by models that were trained on Imagenet, which shows that *good visual representations can be learned without manual supervision.* Our experiments also reveal several benefits of training convolutional networks on datasets such as the YFCC100M dataset: our models learn word embeddings that capture semantic information on analogies whilst being grounded in vision. Although they are not trained for translation, our models can also relate words from different languages by observing that they tend to be assigned to similar visual inputs.

2 Related Work

This study is not the first to explore alternatives to training convolutional networks on manually annotated datasets [8, 12, 51, 69]. In particular, Chen and

[2] The combined number of photo uploads via various platforms was estimated to be 1.8 billion photos per day in 2014 [39].

Gupta [8] propose a curriculum-learning approach that trains convolutional networks on "easy" examples retrieved from Google Images, and then finetune the models on weakly labeled image-hashtag pairs. Their results suggest that such a two-stage approach outperforms models trained on solely image-hashtag data. This result is most likely due to the limited size of the dataset that was used for training (∼1.2 million images): our results show substantial performance improvements can be obtained by training on much larger image-word datasets. Izadinia *et al.* [26] finetune pretrained convolutional networks on a dataset of Flickr images using a vocabulary of $5,000$ words. By contrast, this study trains convolutional networks *from scratch* on 100 million images associated with $100,000$ words. Ni *et al.* [43] also train convolutional networks on tens of millions of image-word pairs, but their study does not report recognition performances. Xiao *et al.* [64] train convolutional networks on noisy targets, but they only consider a very restricted domain and their targets are much less noisy.

Several studies have used weakly supervised data in image-recognition pipelines that use pre-defined visual features. In particular, Li and Fei-Fei [34] present a model that performs simultaneous dataset construction and incremental learning of object recognition models. Li *et al.* [35] learn mid-level representations by training a multiple-instance learning SVMs on low-level features extracted from images from Google Image search. Denton *et al.* [10] learn embeddings of images and hashtags on a large set of Instagram photos and hashtags. Torresani *et al.* [59] train weak object classifiers and use the classifier outputs as additional image features. In contrast to these studies, we backpropagate the learning signal through the entire vision pipeline, allowing us to learn visual features.

In contrast to our work, many prior studies also attempt to explicitly discard low-quality labels by developing algorithms that identify relevant image-hashtag pairs from a weakly labeled dataset [14, 46, 62]. These studies solely aim to create a "clean" dataset and do not explore the training of recognition pipelines on noisy data. By contrast, we study the training of a full image-recognition pipeline; our results suggest that "label cleansing" may not be necessary to learn good visual features if the amount of weakly supervised training data is sufficiently large.

Our work is also related to prior studies on multimodal embedding [54, 65] that explore approaches such as kernel canonical component analysis [18, 24], restricted Boltzmann machines [55], topic models [28], and log-bilinear models [32]. Some works co-embed images and words [16], whereas others co-embed images and sentences or n-grams [15, 30, 61]. Frome *et al.* [16] show that convolutional networks trained jointly on annotated image data and a large corpus of unannotated texts can be used for zero-shot learning. Our work differs from those prior studies in that we train convolutional networks without any manual supervision.

3 Weakly Supervised Learning of Convnets

We train our models on the publicly available YFCC100M dataset [57]. The dataset contains approximately 99.2 million photos with associated titles, hashtags, and comments. Our models are publicly available online.

Preprocessing. We preprocessed the text by removing all numbers and punctuation (*e.g.*, the # character for hashtags), removing all accents and special characters, and lower-casing. We then used the Penn Treebank tokenizer to tokenize the titles and captions into words, and used all hashtags and words as targets for the photos. We remove the 500 most common words (*e.g.*, "the", "of", and "and") and because the tail of the word distribution is very long [1], we restrict ourselves to predicting only the $K = \{1,000; 10,000; 100,000\}$ most common words. For these dictionary sizes, the average number of targets per photo is 3.72, 5.62, and 6.81, respectively. The target for each image is a bag of all the words in the dictionary associated with that image, *i.e.*, a multi-label vector $\mathbf{y} \in \{0,1\}^K$. The images were preprocessed by rescaling them to 256×256 pixels, cropping a central region of 224×224 pixels, subtracting the mean pixel value of each image, and dividing by the standard deviation of its pixel values.

Network architecture. We experimented with two convolutional network architectures, *viz.*, the AlexNet architecture [33] and the GoogLeNet architecture [56]. The AlexNet architecture is a seven-layer architecture that uses max-pooling and rectified linear units at each layer; it has between 15M and 415M parameters depending on the vocabulary size. The GoogLeNet architecture is a narrower, twelve-layer architecture that has a shallow auxiliary classifier to help learning. Our GoogLeNet models had between 4M and 404M parameters depending on vocabulary size. For exact details on both architectures, we refer the reader to [33] and [56], respectively—our architectures only deviate from the architectures described there in the size of their final output layer.

Loss functions. We denote the training set by $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1,\ldots,N}$ with the D-dimensional observation $\mathbf{x} \in \mathbb{R}^D$ and the multi-label vector $\mathbf{y} \in \{0,1\}^K$. We parametrize the mapping $f(\mathbf{x};\theta)$ from observation $\mathbf{x} \in \mathbb{R}^D$ to some intermediate embedding $\mathbf{e} \in \mathbb{R}^E$ by a convolutional network with parameters θ; and the mapping from that embedding $\mathbf{e}$ to a label $\mathbf{y} \in \{0,1\}^K$ by $\operatorname{sign}(\mathbf{W}^\top \mathbf{e})$, where $\mathbf{W}$ is an $E \times K$ matrix. The parameters θ and $\mathbf{W}$ are optimized jointly to minimize a one-versus-all or multi-class logistic loss. We considered two loss functions. The one-versus-all logistic loss sums binary classifier losses over all classes:

$$\ell(\theta, \mathbf{W}; \mathcal{D}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{y_{nk}}{N_k} \log \sigma(\mathbf{W}^\top f(\mathbf{x}_n;\theta)) + \frac{1-y_{nk}}{N-N_k} \log(1-\sigma(\mathbf{W}^\top f(\mathbf{x}_n,\theta))),$$

where $\sigma(x) = 1/(1+\exp(-x))$ and N_k is the number of positive examples for the class k. The multi-class logistic loss minimizes the negative sum of the log-probabilities, which are computed using a softmax layer, over all positive labels:

$$\ell(\theta, \mathbf{W}; \mathcal{D}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk} \log \left[\frac{\exp(\mathbf{w}_k^\top f(\mathbf{x}_n;\theta))}{\sum_{k'=1}^{K} \exp(\mathbf{w}_{k'}^\top f(\mathbf{x}_n;\theta))} \right].$$

In preliminary experiments, we also considered a pairwise ranking loss [60, 61]. Such losses only update two columns of $\mathbf{W}$ per training example (corresponding to a positive and a negative label). We found that when training convolutional networks end-to-end, these sparse updates significantly slowed down training, which is why we did not consider ranking loss further in this study.

Class balancing. The distribution of words in our dataset follows a Zipf distribution [1]: much of its probability mass is accounted for by a few classes. We carefully sample training instances to prevent these classes from dominating the learning, which may lead to poor general-purpose visual features [2]. We follow Mikolov *et al.* [40] and sample instances *uniformly per class*. Specifically, we select a training example by picking a word uniformly at random and select an image associated with that word randomly. When using multi-class logistic loss, all the other words are considered negative for the corresponding image, *even words that are also associated with that image.* This procedure potentially leads to noisier gradients but it works well in practice. (The comments miss relevant words anyway, so our procedure only slightly exacerbates an existing problem.)

Training. We trained our models with elastic averaging stochastic gradient descent (EA-SGD; [68]) on batches of size 128. In all experiments, we set the initial learning rate to 0.1 and after every sweep through a million images (an "epoch"), we compute the prediction error on a held-out validation set. When the validation error has increased after an "epoch", we divide the learning rate by 2 and continue training; but we use each learning rate for at least 10 epochs. We stopped training when the learning rate became smaller than 10^{-6}.

Large dictionary. Training a network on $100,000$ classes is computationally expensive: a full forward-backward pass through the last linear layer with a single batch takes roughly $1,600$ms (compared to 400ms for the rest of the network). This scaling issue commonly occurs in language modeling [7], and can be addressed using approaches such as importance sampling [4], noise-contrastive estimation [21, 41], and the hierachical softmax [19, 42]. Similar to Jozefowicz *et al.* [29], we found importance sampling to be quite effective: we only update the weights that correspond to classes present in a training batch. This means we update at most 128 columns of $\mathbf{W}$ per batch instead of all $100,000$ columns. This reduced the training time of our largest models from months to weeks. Whilst our approximation is consistent for the one-versus-all loss, it is not for the multi-class logistic loss: in the worst-case scenario, the "approximate" logistic loss can be arbitrarily far from the true loss. However, we observe that the approximation works well in practice. We also derived upper and lower bounds on the expected value of the approximate loss, which show that it is closely related to the true loss. Denoting $s_k = \exp\left(\mathbf{w}_k^\top f(\mathbf{x}_n;\theta)\right)$ and the set of sampled classes by $\mathcal{C}$ (with $|\mathcal{C}| \leq K$) and leaving out constant terms, a trivial upper bound shows that the expected approximate loss never overestimates the true loss:

$$\mathbb{E}\left[\log \sum_{c \in \mathcal{C}} s_c\right] \leq \log \sum_{k=1}^{K} s_k = \log Z.$$

Assuming that $\forall k : s_k \geq 1$[3], Markov's inequality provides a lower bound, too:

$$\mathbb{E}\left[\log \sum_{c \in \mathcal{C}} s_c\right] \geq P\left(\frac{1}{|\mathcal{C}|}\sum_{c \in \mathcal{C}} s_c \geq \frac{1}{K} Z\right)\left(\log \frac{|\mathcal{C}|}{K} + \log Z\right).$$

[3] This assumption can always be satisfied by adding a constant inside the exponentials of both the numerator and the denominator of the softmax.

This bound relates the sample average of s_c to its expected value, and is exact when $|\mathcal{C}| \rightarrow K$. The lower bound only contains an additive constant $\log(|\mathcal{C}|/K)$, which shows that the approximate loss is closely related to the true loss.

4 Experiments

To assess the quality of our weakly-supervised convolutional networks, we performed three sets of experiments: (1) experiments measuring the ability of the models to predict words given an image, (2) transfer-learning experiments measuring the quality of the visual features learned by our models in a range of computer-vision tasks, and (3) experiments evaluating the quality of the word embeddings learned by the networks.

4.1 Experiment 1: Associated Word Prediction

Experimental setup. We measure the ability of our models to predict words that are associated with an image using the precision@k on a test set of 1 million YFCC100M images, which we held out until after all our models were trained. Precision@k is a suitable measure for assessing word prediction performance because it is robust to the fact that targets are noisy, *i.e.*, that images may have words assigned to them that do not describe their visual content.

As a baseline, we train L2-regularized logistic regressors on features produced by convolutional networks trained on the Imagenet dataset. The Imagenet models were trained on 224×224 crops that where randomly selected from 256×256 input images. We applied photometric jittering on the input images [25], and trained using EA-SGD with batches of 128 images. Our pretrained networks perform on par with the state-of-the-art on ImageNet: a single AlexNet obtains a top-5 test error of 24.0% on a single crop; our GoogLeNet has top-5 error of 10.7%. The L2 regularization parameter of the logistic regressor was tuned on a held-out validation set.

Table 1. Word prediction precision@10 on the YFCC100M test data for three dictionary sizes K obtained by: (1) logistic regressors trained on features extracted from convolutional networks that were *pretrained* on Imagenet and (2) convolutional networks trained *end-to-end* using multiclass logistic loss. Higher values are better.

		Dictionary size K		
Type	**Network**	**1,000**	**10,000**	**100,000**
Pretrained	AlexNet	8.27	4.01	1.61
	GoogLeNet	13.20	4.76	1.54
End-to-end	AlexNet	17.98	6.27	2.56
	GoogLeNet	20.21	6.47	–

Results. Table 1 presents the precision@10 of word prediction models trained using multi-class logistic loss on the YFCC100M dataset, using dictionaries with $K = 1,000$, $K = 10,000$, and $K = 100,000$ words. The results of this experiment show that end-to-end training of convolutional networks on the YFCC-100M dataset works substantially better than training a classifier on features extracted from an Imagenet-pretrained network: end-to-end training leads to a relative gain of 45 to 110% in precision@10. This suggests that the features learned by networks on the Imagenet dataset are too tailored to the specific set of classes in that dataset. The results also show that the relative differences between GoogLeNet and AlexNet are smaller on the YFCC100M than on the Imagenet dataset, possibly, because GoogLeNet has less capacity than AlexNet.

In preliminary experiments, we also trained models using one-versus-all logistic loss: using a dictionary of $K = 1,000$ words, such a model achieves a precision@10 of 16.43 (compared to 17.98 for multiclass logistic loss). We surmise this is due to the problems one-versus-all logistic loss has in dealing with class imbalance: because the number of negative examples is much higher than the number of positive examples (for the most frequent class, more than 95.0% of the data is negative), the rebalancing weight in front of the positive term is very high, which leads to spikes in the gradient magnitude that hamper training. We tried various reweighting schemes to counter this effect, but nevertheless, multi-class logistic loss consistently outperformed one-versus-all logistic loss.

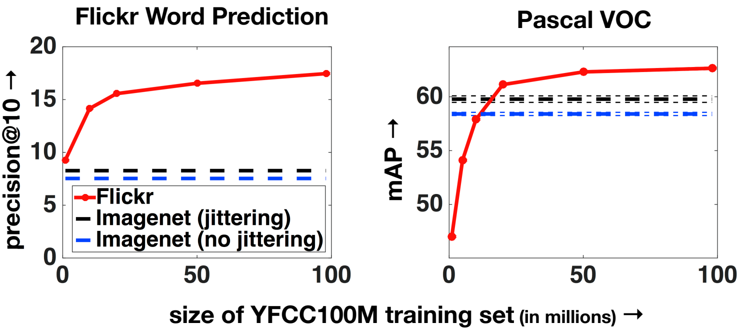


Fig. 2. *Left:* Word prediction precision@10 of AlexNets trained on YFCC100M training sets of different sizes using $K = 1,000$ and a single crop (in red); and precision@10 of logistic regressors trained on features from convolutional networks trained on ImageNet with and without jittering (in blue and black). *Right:* Mean average precision on the Pascal VOC 2007 image classification task obtained by logistic regressors trained on features extracted by an AlexNet trained on YFCC100M (in red) and ImageNet (in blue and black).

To investigate the performance of our models as a function of the amount of training data, we also performed experiments in which we varied the training set size. Figure 2 presents the resulting learning curves for the AlexNet architecture with $K = 1,000$. The figure shows that there is a clear benefit of training on larger datasets: the word prediction performance of the networks increases substantially when the training set is increased beyond 1 million images (which is roughly the size of Imagenet); for our networks, it only levels out after ∼50 million images.

To illustrate the kinds of words for which our models learn good representations, we show a high-scoring test image for six different words in Figure 3. To obtain more insight into the features learned by the models, we applied t-SNE [37, 38] to features extracted from the penultimate layer of an AlexNet trained on 1,000 words. This produces maps in which images with similar visual features are close together; Figure 4 shows such a map of 20,000 test images. The inset shows a "sports" cluster that was formed by the visual features; interestingly, it contains visually very dissimilar sports ranging from baseball to field hockey, ice hockey and rollerskating. Whilst all sports are grouped together, the individual sports are still clearly separable: the model can capture this multi-level structure because the images sometimes occur with the word "sports" and sometimes with the name of the individual sport itself. A model trained on classification datasets such as Pascal VOC is unlikely to learn similar structure unless an explicit target taxonomy is defined (as in the Imagenet dataset) and exploited via a hierarchical loss. Our results suggest that class taxonomies can be learned directly from photo comments instead.

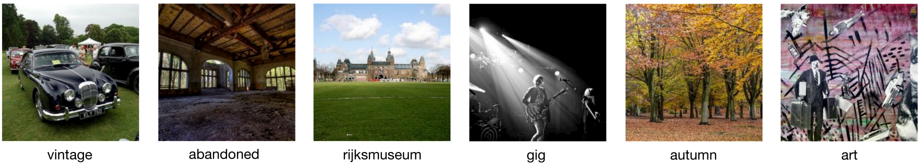


Fig. 3. Six test images with high scores for different words. The scores were computed by an AlexNet trained on the YFCC100M dataset using $K = 100,000$ words.

4.2 Experiment 2: Transfer Learning

Experimental setup. To assess the quality of the visual features learned by our models, we performed transfer-learning experiments on seven test datasets comprising a range of computer-vision tasks: (1) the MIT Indoor dataset [49], (2) the MIT SUN dataset [63], (3) the Stanford 40 Actions dataset [66], (4) the Oxford Flowers dataset [44], (5) the Sports dataset [20], (6) the ImageNet ILSVRC 2014 dataset [52], and (7) the Pascal VOC 2007 dataset [13]. We applied the same preprocessing on all datasets: we resized the images to 224×224 pixels, subtracted their mean pixel value, and divided by their standard deviation.

Following [50], we compute the output of the penultimate layer for an input image and use this output as a feature representation for the corresponding image. We evaluate features obtained from YFCC100M-trained networks as well as Imagenet-trained networks, and we also perform experiments where we combine both features by concatenating them. We train L2-regularized logistic regressors on the features to predict the classes corresponding to each of the datasets. For all datasets except the Imagenet and Pascal VOC datasets, we report classification accuracies on a separate, held-out test set. For Imagenet, we report classification errors on the validation set. For Pascal VOC, we report average precisions on the test set as is customary for that dataset. Again, we use convolutional networks trained on Imagenet as a baseline. Additional details on the setup of the transfer-learning experiments are in the supplemental material.

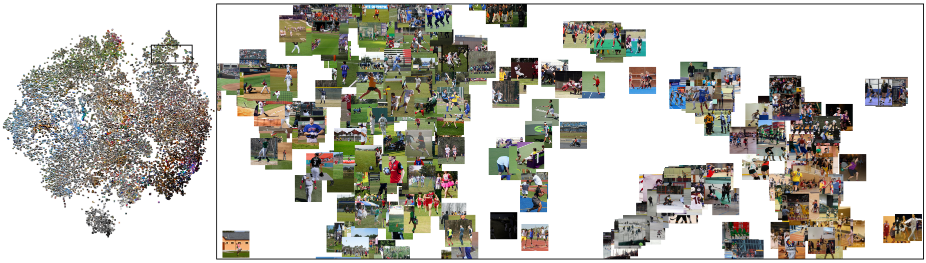


Fig. 4. t-SNE map of 20, 000 YFCC100M test images based on features extracted from the last layer of an AlexNet trained with $K = 1,000$. A full-resolution map is presented in the supplemental material. The inset shows a cluster of sports.

Results. Table 4.2 presents the classification accuracies—averaged over 10 runs—of logistic regressors on six datasets for both fully supervised and weakly supervised feature-production networks, as well as for a combination of both networks. Table 2 presents the average precision on the Pascal VOC 2007 dataset. Our weakly supervised models were trained on a dictionary of $K = 1,000$ words. The results in the tables show that using the AlexNet architecture, weakly supervised networks learn visual features of similar quality as fully supervised networks. This is quite remarkable because the networks learned these features *without any strong supervision*. Using more complex classifiers and ensembling, the classification accuracies can be improved substantially: for instance, we obtain an mAP of 82.01 on the Pascal VOC 2007 dataset using a neural-network classifier and multiple crops, using the same features (see supplemental material).

Table 2. Pascal VOC 2007 dataset: Average precision (AP) per class and mean average precision (mAP) of classifiers trained on features extracted with networks trained on the Imagenet and the YFCC100M dataset (using $K = 1,000$ words). Using more complex classifiers and multiple crops, we obtain an mAP of 82.01 on the Pascal VOC dataset (see supplemental material). Higher values are better.

Dataset	**Model**	✈	🚲	🐦	⛵	🍾	🚌	🚗	🐈	🪑	🐄	table	🐕	🐎	🏍	🧍	🪴	🐑	🛋	🚂	📺	**mAP**
Imagenet	AlexNet	75.7	61.9	66.9	66.5	29.3	56.1	73.5	68.0	47.1	40.9	57.4	60.0	74.0	63.2	86.2	38.8	57.9	45.5	75.7	51.1	59.8
	GoogLeNet	91.3	84.0	88.4	87.2	42.4	79.6	87.3	85.0	59.1	66.5	69.5	83.3	86.6	82.9	88.4	57.5	75.8	64.6	89.5	73.8	77.1
YFCC100M	AlexNet	84.0	72.2	70.2	77.0	29.5	60.8	79.3	69.5	49.2	40.5	54.0	57.1	79.2	64.6	90.2	43.0	47.5	44.1	85.0	50.7	62.4
	GoogLeNet	91.5	83.7	84.1	88.5	41.7	78.0	86.8	84.0	54.7	55.5	63.3	78.5.	86.0	77.4	91.1	51.3	60.8	52.7	91.9	60.9	73.2
Combined	AlexNet	82.96	70.32	73.28	76.29	32.21	61.84	79.81	72.91	51.56	43.82	60.77	63.32	78.63	67.72	90.26	45.45	53.15	49.14	84.8	55.8	64.7
	GoogLeNet	94.09	85.03	89.71	88.47	49.35	81.47	88.1	85.2	60.51	68.37	71.65	85.81	88.87	85.22	88.69	60.45	77.26	66.61	90.71	74.49	79.0

Admittedly, weakly supervised networks perform poorly on the flowers dataset: Imagenet-trained networks produce better features for that dataset, presumably,

Table 3. Classification accuracies on held-out test data of logistic regressors obtained on six datasets (MIT Indoor, MIT SUN, Stanford 40 Actions, Oxford Flowers, Sports, and ImageNet) using feature representations obtained from convolutional networks trained on the Imagenet and the YFCC100M dataset (using $K = 1,000$ words and a single crop). Errors are averaged over 10 runs. Higher values are better.

Dataset	**Model**	**Indoor**	**SUN**	**Action**	**Flower**	**Sports**	**ImNet**
Imagenet	AlexNet	53.82	41.40	51.27	80.28	86.07	53.63
	GoogLeNet	64.00	48.76	67.10	79.05	95.91	69.89
YFCC100M	AlexNet	55.82	42.67	53.02	74.24	90.78	35.71
	GoogLeNet	55.56	44.43	52.84	65.80	87.40	35.61
Combined	AlexNet	58.76	47.27	56.35	83.28	87.50	–
	GoogLeNet	67.87	55.04	69.19	83.74	95.79	–

because the Imagenet dataset itself focuses strongly on fine-grained classification. Interestingly, fully supervised networks do learn better features than weakly supervised networks when a GoogLeNet architecture is used: this result is in line with the results from 4.1, which suggest that GoogLeNet has too little capacity to learn optimal models on the Flickr data. The substantial performance improvements we observe in experiments in which features from both networks are combined suggest that the features learned by both models complement each other. We note that achieving state-of-the-art results [6, 45, 50, 70] on these datasets requires the development of tailored pipelines, *e.g.*, using many image transformations and model ensembles, which is outside the scope of this paper. We also measured the transfer-learning performance as a function of the YFCC100M training set size. The results of these experiments with the AlexNet architecture and $K = 1,000$ are presented in Figure 5 for four of the datasets (Indoor, MIT SUN, Stanford 40 Actions, and Oxford Flowers) and the Pascal VOC dataset. The results show that good feature-production networks can be learned from tens of millions of weakly supervised images.

4.3 Experiment 3: Assessing Word Embeddings

The weights in the last layer of our networks can be viewed as an embedding of the words. This word embedding is, however, different from those learned by language models such as word2vec [40] that learn embeddings based on word co-occurrence: it is constructed *without explicitly modeling words co-occurrence* (recall that during training, we use a single, randomly selected word as target for an image). This means that structure in the word embedding can only be learned when the network notices that two words are assigned to images with similar visual content. We perform two sets of experiments to assess the quality of the word embeddings learned by our networks: (1) experiments investigating how well the word embeddings represent semantic information and (2) experiments investigating the ability of the embeddings to learn correspondences between different languages.

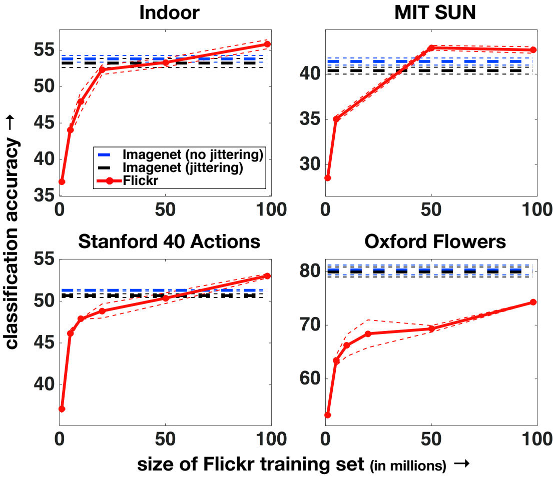


Fig. 5. Average classification accuracy (averaged over ten runs) of logistic regressors trained on features produced by YFCC100M-trained AlexNets trained on four datasets (in red). For reference, we also show the classification accuracy of classifiers trained on features from networks trained on ImageNet without jittering (in black) and with jittering (in blue). Dashed lines indicate the standard deviation across runs. Higher values are better.

Semantic information. We evaluate our word embeddings on two datasets that capture different types of semantic information: (1) a syntactic-semantic questions dataset [40] and (2) the MEN word similarity dataset [5]. The syntactic-semantic dataset contains $8,869$ semantic and $10,675$ syntactic questions of the form "A is to B as C is to D". Following [40], we predict D by finding the word embedding vector $\mathbf{w}_D$ that has the highest cosine similarity with $\mathbf{w}_B - \mathbf{w}_A + \mathbf{w}_C$ (excluding A, B, and C from the search), and measure the number of times we predict the correct word D. The MEN dataset contains $3,000$ word pairs spanning 751 unique words—all of which appear in the ESP Game image dataset—with an associated similarity rating. The similarity ratings are averages of ratings provided by a dozen human annotators. Following [31] and others, we measure the quality of word embeddings by the Spearman's rank correlation of the cosine similarity of the word pairs and the human-provided similarity rating for those pairs. In all experiments, we excluded word quadruples / pairs that contained words that are not in our dictionary. We repeated the experiments for three dictionary sizes. For reference, we also measured the performance of word2vec models that were trained on all comments in the YFCC100M dataset (using only the words in the dictionary).

The prediction accuracies of our experiments on the syntactic-semantic dataset for three dictionary sizes are presented in the lefthand side of Table 4. The right-hand side of Table 4 presents the rank correlations for our word embeddings on the MEN dataset (for three vocabulary sizes). As before, we only included word pairs for which both words appeared in the vocabulary. The results of these experiments show that our weakly supervised models learned meaningful semantic structure. For small dictionary sizes, our models even perform on par with word2vec, even though our models had no access to language like word2vec: our models were trained only on image-word pairs and, unlike word2vec, do not explicitly model word co-occurrences. All semantic structure in the word embedding of our weakly supervised convolutional network was learned by observing that certain words co-occur with particular visual inputs.

Table 4. *Lefthand side:* Prediction accuracy of predicting D in questions "A is to B like C is to D" using convolutional-network word embeddings and word2vec on the syntactic-semantic dataset, using three dictionary sizes. Questions containing words not in the dictionary were removed. Higher values are better. *Righthand side:* Spearman's rank correlation of cosine similarities between convolutional-network (and word2vec) word embeddings and human similarity judgements on the MEN dataset. Word pairs containing words not in the dictionary were removed. Higher values are better.

Model	**Syntactic-Semantic Dataset**			**MEN dataset**		
	$\mathbf{K=1,000}$	$\mathbf{K=10,000}$	$\mathbf{K=100,000}$	$\mathbf{K=1,000}$	$\mathbf{K=10,000}$	$\mathbf{K=100,000}$
AlexNet	67.91	29.29	0.85	73.77	75.73	67.35
GoogLeNet	71.92	24.06	–	75.72	75.89	–
word2vec	71.92	61.35	47.24	75.25	77.53	77.91
AlexNet + word2vec	74.79	57.26	44.35	78.17	79.24	78.57
GoogLeNet + word2vec	75.36	56.05	–	78.75	79.11	–

We also made t-SNE maps of the embedding of $10,000$ words in Figure 6. The insets highlight five "topics": (1) musical performance, (2) female and male first names, (3) sunsets, (4) photography, and (5) gardening. These topics were identified by the model solely based on the fact that the words in the are associated with images that have a similar visual content: for instance, first names are often assigned to photos of individuals or small groups of people. Interestingly, the "sunset" and "gardening" topics show examples of grouping of words from different languages. For instance, "sonne", "soleil", "sole" mean "sun" in German, French, and Italian, respectively; and "garten" and "giardino" are the German and Italian words for garden. Our model learns multi-lingual word correspondences because the words are assigned to similarly looking images.

Table 5. Precision@k of identifying the French counterpart of an English word (and vice-versa) for two dictionary sizes. Chance level (with $k=1$) is 0.0032 for $K=10,000$ words and 0.00033 for $K=100,000$ words. Higher values are better.

K	**Query → Response**	$\mathbf{k=1}$	$\mathbf{k=5}$	$\mathbf{k=10}$
$10,000$	English → French	33.01	50.16	55.34
	French → English	23.95	50.16	56.63
$100,000$	English → French	12.30	22.24	26.50
	French → English	10.11	18.78	23.44

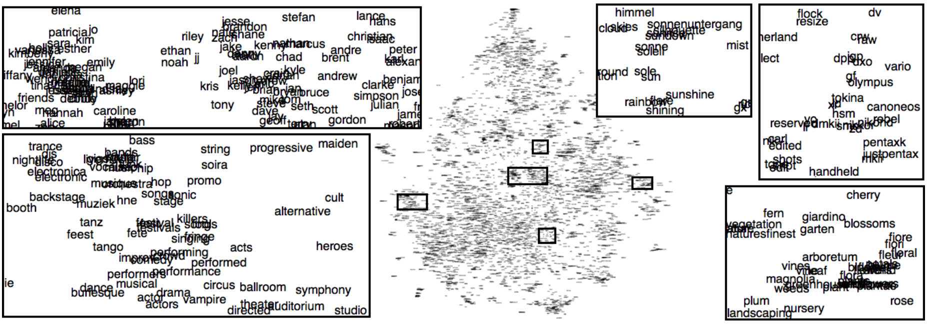


Fig. 6. t-SNE map of 10,000 words based on their embeddings as learned by a weakly supervised convolutional network trained on the YFCC100M dataset. Note that all the semantic information represented in the word embeddings is the result of observing that these words are assigned to images with similar visual content (the model did not observe word co-occurrences during training). A full-resolution version of the map is provided in the supplemental material.

Multi-lingual correspondences. To quantitatively investigate the ability of our models to find correspondences between words from different languages, we selected pairs of words from an English-French dictionary[4] for which: (1) both the English and the French word are in the dictionary and (2) the English and the French word are different. This produced 309 English-French word pairs for models trained on $K = 10,000$ words, and 3,008 English-French word pairs for models trained on $K = 100,000$ words. We measured the quality of the multi-lingual word correspondences in the embeddings by taking a word in one language and ranking the words in the other language according to their cosine similarity with the query word. We measure the precision@k of the predicted word ranking, using both English and French words as query words.

Table 5 presents the results of this experiment: for a non-trivial number of words, our procedure correctly identified the French translation of an English word, and vice versa. Finding the English counterpart of a French word is harder than the other way around, presumably, because there are more English than French words in the dictionary: this implies that the English word embeddings are better optimized than the French ones. In Table 6, we show the ten most similar word pairs, measured by the cosine similarity between their word embeddings. These word pairs suggest that models trained on YFCC100M find correspondences between words that have clear visual representations, such as "tomatoes" or "bookshop". Interestingly, the identified English-French matches appear to span a broad set of domains, including objects such as "pencils", locations such as "mauritania", and concepts such as "infrared".

[4] http://www-lium.univ-lemans.fr/~schwenk/nnmt-shared-task/

Table 6. Twelve highest-scoring pairs of words, as measured by the cosine similarity between the corresponding word embeddings. Correct pairs of words are colored green, and incorrect pairs are colored red according to the dictionary. The word "oas" is an abbreviation for the Organization of American States.

English	French	English	French	English	French
oas	oea	server	apocalyptique	mauritania	mauritanie
infrared	infrarouge	uzbekistan	ouzbekistan	pencils	crayons
tomatoes	tomates	mushroom	champignons	fog	brouillard
bookshop	librairie	filmed	serveur	jetliner	avion

5 Discussion and Future Work

This study demonstrates that convolutional networks can be trained *from scratch* without any manual annotation and shows that good vision features can be learned from weakly supervised data such as Flickr photos and associated comments. Indeed, our models learn visual features that are roughly on par with those learned from an image collection with over a million manually defined labels, and achieve competitive results on a variety of datasets. This result paves the way for interesting new approaches to the training of large computer-vision models, and over time, may render the manual annotation of large training sets unnecessary. In this study, we have not focused on beating the state-of-the-art performance on an individual vision benchmark: obtaining state-of-the-art results generally requires averaging predictions over many crops and models, which is not the goal of this paper. In the supplemental material, however, we do show that it is straightforward to obtain a mAP of 82.01 on the Pascal VOC 2007 classification dataset using the features learned by our models.

The results presented in this paper lead to three main recommendations for future work in learning models from weakly supervised data. First, our results suggest that the best-performing models on the Imagenet dataset are not optimal for weakly supervised learning. We surmise that current models have insufficient capacity for learning from the complex Flickr dataset. Second, multi-class logistic loss performs remarkably well in our experiments even though it is not tailored to multi-label settings. Presumably, our approximate multi-class loss works very well on large dictionaries because it shares properties with losses known to work well in that setting [40, 60, 61]. Third, it is essential to sample data *uniformly per class* to learn good visual features [2]. Uniform sampling per class ensures that frequent classes in the training data do not dominate the learned features, which makes the features better suited for transfer learning.

In future work, we aim to combine our weakly supervised vision models with a language model such as word2vec [40] to perform, for instance, visual question answering [3, 67]. We also intend to extend our model to do language modeling, *e.g.*, by using an LSTM as output [23]. We also intend to further investigate the ability of our models to learn visual hierarchies, such as the "sports" example of Section 4.2.

References

1. Adamic, L., Huberman, B.: Zipf's law and the internet. Glottometrics 3, 143–150 (2002)
2. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Good practice in large-scale learning for image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(3), 507–520 (2014)
3. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C., Parikh, D.: VQA: Visual question answering. In: Proceedings of the International Conference on Computer Vision (2015)
4. Bengio, Y., Senecal, J.S.: Quick training of probabilistic neural nets by importance sampling. In: Proceedings of AI-STATS (2003)
5. Bruni, E., Boleda, G., Baroni, M., Tran, N.: Distributional semantics in technicolor. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics. pp. 136–145 (2012)
6. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: An evaluation of recent feature encoding methods. In: Proceedings of the British Machine Vision Conference (2011)
7. Chen, W., Grangier, D., Auli, M.: Strategies for training large vocabulary neural language models. arXiv:1512.04906 (2015)
8. Chen, X., Gupta, A.: Webly supervised learning of convolutional networks. In: Proceedings of the International Conference on Computer Vision (2015)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, (CVPR) (2009)
10. Denton, E., Weston, J., Paluri, M., Bourdev, L., Fergus, R.: User conditional hashtag prediction for images. In: Proceedings of the SIGKDD Conference on Knowledge Discovery and Data Mining (2015)
11. DiCarlo, J., Zoccolan, D., NC, N.R.: How does the brain solve visual object recognition? Neuron 73(3), 415–434 (2012)
12. Divvala, S.K., Farhadi, A., Guestrin, C.: Learning everything about anything: Webly-supervised visual concept learning. In: Computer Vision and Pattern Recognition (CVPR) (2014)
13. Everingham, M., Eslami, S., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The Pascal visual object classes challenge — a retrospective. International Journal on Computer Vision 111(1), 98–136 (2015)
14. Fan, J., Shen, Y., Zhou, N., Gao, Y.: Harvesting large-scale weakly tagged image databases from the web. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 802–809 (2010)
15. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: Generating sentences from images. In: ECCV. pp. 15–29. Springer (2010)
16. Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: Advances in Neural Information Processing Systems. pp. 2121–2129 (2013)
17. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Computer Vision and Pattern Recognition (CVPR). pp. 580–587. IEEE (2014)
18. Gong, Y., Ke, Q., Isard, M., Lazebnik, S.: A multi-view embedding space for modeling internet images, tags, and their semantics. International journal of computer vision 106(2), 210–233 (2014)

19. Goodman, J.: Classes for fast maximum entropy training. In: ICASSP. vol. 1, pp. 561–564 (2001)
20. Gupta, A., Kembhavi, A., Davis, L.: Observing human-object interactions: Using spatial and functional compatibility for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(10) (2009)
21. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: International Conference on Artificial Intelligence and Statistics. pp. 297–304 (2010)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)
24. Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research pp. 853–899 (2013)
25. Howard, A.: Some improvements on deep convolutional neural network based image classification. In: arXiv 1312.5402 (2013)
26. Izadinia, H., Russell, B., Farhadi, A., Hoffman, M., Hertzmann, A.: Deep classifiers from image tags in the wild. In: Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions. pp. 13–18. ACM (2015)
27. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: arXiv 1506.02025 (2015)
28. Jia, Y., Salzmann, M., Darrell, T.: Learning cross-modality similarity for multinomial data. In: ICCV. pp. 2407–2414. IEEE (2011)
29. Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y.: Exploring the limits of language modeling. In: arXiv 1602.02410 (2016)
30. Karpathy, A., Joulin, A., Fei Fei, L.: Deep fragment embeddings for bidirectional image sentence mapping. In: Advances in neural information processing systems. pp. 1889–1897 (2014)
31. Kiela, D., Bottou, L.: Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2014)
32. Kiros, R., Salakhutdinov, R., Zemel, R.: Multimodal neural language models. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). pp. 595–603 (2014)
33. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (2012)
34. Li, L.J., Fei-Fei, L.: Optimol: Automatic online picture collection via incremental model learning. Internation Journal of Computer Vision (2010)
35. Li, Q., Wu, J., Tu, Z.: Harvesting mid-level visual concepts from large-scale internet images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2013)
36. Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, L.: Microsoft COCO: Common objects in context. In: ECCV 2014 (2014)
37. van der Maaten, L.: Accelerating t-SNE using tree-based algorithms. Journal of Machine Learning Research 15(Oct), 3221–3245 (2014)
38. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research 9(Nov), 2579–2605 (2008)

39. Meeker, M.: Internet trends 2014. Tech. rep., Kleiner, Perkins, Caufield & Byers (2014)
40. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: arXiv 1301.3781 (2013)
41. Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: Advances in Neural Information Processing Systems. pp. 2265–2273 (2013)
42. Morin, F., Bengio, Y.: Hierarchical probabilistic neural network language model. In: AI-STATS. vol. 5, pp. 246–252 (2005)
43. Ni, K., Pearce, R., Wang, E., Boakye, K., Essen, B.V., Borth, D., Chen, B.: Large-scale deep learning on the YFCC100M dataset. In: arXiv 1502.03409 (2015)
44. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (2008)
45. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. pp. 1717–1724. IEEE (2014)
46. Ordonez, V., Kulkarni, G., Berg, T.: Im2text: Describing images using 1 million captioned photographs. In: Advances in Neural Information Processing Systems. pp. 1143–1151 (2011)
47. Pinheiro, P., Collobert, R., Dollár, P.: Learning to segment object candidates. In: Advances in Neural Image Processing (2016)
48. Ponce, J., Berg, T., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B., Torralba, A., Williams, C., Zhang, J., Zisserman, A.: Dataset issues in object recognition. In: Lecture Notes in Computer Science 4170. pp. 29–48 (2006)
49. Quattoni, A., A.Torralba: Recognizing indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
50. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: arXiv 1403.6382 (2014)
51. Rubinstein, M., Joulin, A., Kopf, J., Liu, C.: Unsupervised joint object discovery and segmentation in internet images. In: Computer Vision and Pattern Recognition (CVPR) (2013)
52. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision pp. 1–42 (2015)
53. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations (2015)
54. Socher, R., Ganjoo, M., Manning, C., Ng, A.: Zero-shot learning through cross-modal transfer. In: Advances in Neural Information Processing Systems. pp. 935–943 (2013)
55. Srivastava, N., Salakhutdinov, R.: Multimodal learning with deep boltzmann machines. In: Advances in neural information processing systems. pp. 2222–2230 (2012)
56. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)

57. Thomee, B., Shamma, D., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Communications of the ACM 59(2), 64–73 (2016)
58. Torralba, A., Efros, A.: Unbiased look at dataset bias. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1521–1528 (2011)
59. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classemes. In: Proceedings of the European Conference on Computer Vision (2010)
60. Usunier, N., Buffoni, D., Gallinari, P.: Ranking with ordered weighted pairwise classification. In: Proceedings of the International Conference on Machine Learning. pp. 1057–1064 (2009)
61. Weston, J., Bengio, S., Usunier, N.: Wsabie: Scaling up to large vocabulary image annotation. In: Proceedings of the International Joint Conference on Artificial Intelligence (2011)
62. Xia, Y., Cao, X., Wen, F., Sun, J.: Well begun is half done: Generating high-quality seeds for automatic image dataset construction from web. In: Proceedings of the European Conference on Computer Vision (2014)
63. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2010)
64. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
65. Yang, Y., Teo, C., Daumé III, H., Aloimonos, Y.: Corpus-guided sentence generation of natural images. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 444–454. Association for Computational Linguistics (2011)
66. Yao, B., Jiang, X., Khosla, A., Lin, A., Guibas, L., Fei-Fei, L.: Human action recognition by learning bases of action attributes and parts. In: International Conference on Computer Vision (2011)
67. Yu, L., Park, E., Berg, A., Berg, T.: Visual madlibs: Fill in the blank description generation and question answering. In: Proceedings of the International Conference on Computer Vision (2015)
68. Zhang, S., Choromanska, A., LeCun, Y.: Deep learning with elastic averaging sgd. In: Advances in Neural Information Processing Systems (2015)
69. Zhou, B., Jagadeesh, V., Piramuthu, R.: Conceptlearner: Discovering visual concepts from weakly labeled image collections. arXiv:1411.5328 (2014)
70. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems. pp. 487–495 (2014)