De-Correlating CNN Features for Generative Classification

Chaitanya Desai Jayan Eledath Harpreet Sawhney Mayank Bansal
SRI International

Abstract

The problem of training a classifier from a handful of positive examples, without having to supply class specific negatives is of great practical importance. The proposed approach to solving this problem builds on the idea of training LDA classifiers using only class specific foreground images and a large collection of unlabelled images, as described in [11]. While we adopt the LDA training methodology of [11], we depart from HOG features and work with those extracted from a Convolutional Neural Network (CNN) pre-trained on ImageNet (Overfeat). We combine Overfeat features with the LDA training methodology to derive generative classifiers. When evaluated on a K-way classification problem, these classifiers are almost as good as those trained discriminatively using the same features. Unlike the HOG based approach of [11], our classifiers do not need any post-processing step of calibration, a step that requires positives and negatives. Finally, we show that in an instance retrieval setup, we can employ these generative classifiers to derive a novel query-expansion framework that achieves a significant performance boost by utilizing only the top ranked positive examples from an initial nearest-neighbor list.

1. Introduction

A traditional binary classification setup typically requires two disjoint sets of data: a set of positive training examples and a set of negative training examples. Discriminative training approaches (and SVMs in particular) benefit from having a large set of negative examples. Indeed, the merits of "hard mining" negative examples have been well documented in vision [7, 10]. The availability of a large set of class specific negatives, however, is not always practical. In fact, there are important real-world use cases for which *positive* examples for a class might be readily available, while negative examples might be hard to collect. For example, in surveillance applications, it is natural for a domain expert to acquire a handful of images for a class of interest (an *airport terminal* for instance) with the intent of readily finding instances of this class on a new set of images. It would seem natural to treat this as a classification problem. Each time a batch of positives for a new class arrives, it would be ideal to train a classifier for future use, wherein the input from the end user in only positive images for the class of interest. Armed with a collection of such binary classifiers trained over time, the surveillance expert might now be interested in mapping a new image to one of the "known" classes (i.e. the ones for which trained classifiers exist) – a multi-class classification problem that would be amenable to a one-vs-all setting.

Another use case where positives are readily available but negatives might be hard to obtain is an end user of a vision application who is interested in searching a personal photo collection for images containing a specific scene or category of interest. For example, consider a situation where you take some pictures while out on a kayaking trip with friends. Upon returning from the trip, you are interested in searching all your existing digital photo albums for other kayaking pictures you've taken in the past (to either tag the newly taken pictures or share previously taken pictures with friends). It would be advantageous for the search aplication to "learn" what you are searching for, using only the pictures you took on that day as a source of "training" data.

Unfortunately, the traditional route to training a classifier would require the end user to provide classs-specific negative data. This seems like an unreasonable expectation from someone who is mererly a consumer of a vision application and has little knowledge about training classifiers. We propose a solution to this practical problem that borrows the learning methodology of [11], so that our "online" classifier training does not depend on any class specific negative images. Fig. 1 gives an overview of our approach. Our background model is computed offline from a large collection of unlabeled data, and can be packaged as part of a vision system that trains classifiers online.

Supported by the Air Force Research Laboratory (AFRL) through contract number FA8750-12-C-0103. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). Distribution statement "A" :Approved for Public Release, Distribution Unlimited.

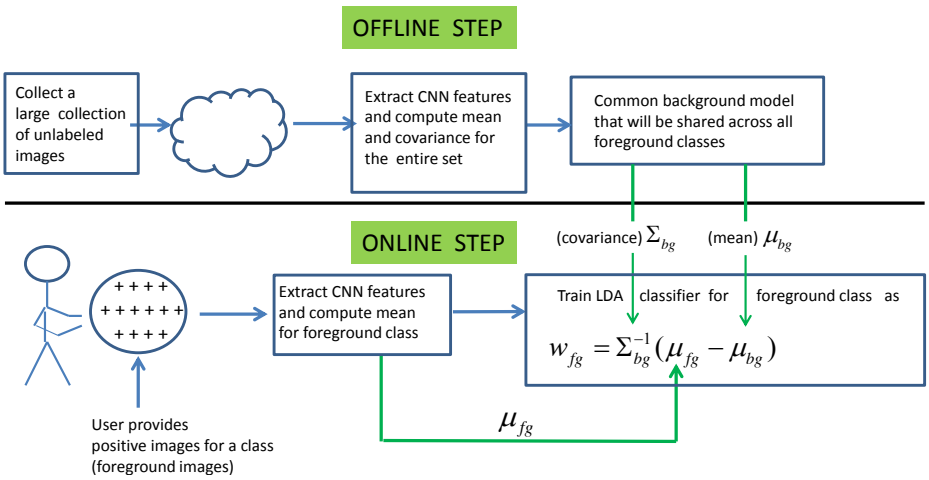


Figure 1. Overview of our approach for training classifiers without class specific negatives. The green arrows indicate the outputs of various steps.

2. Related Work

The problem of training classifiers from positive instances and a common pool of unlabeled instances has received more attention in the data mining and machine learning communities than in Vision. Notable contributions include [8, 13, 21, 20, 18]. As summarized in [8], these approaches can broadly be grouped into two categories:

1. Approaches that iterate between i) Identifying a good set of negatives from the unlabeled examples and ii) Training a binary classifier using positives and the (new) set of negatives.

2. Approaches that assign weights to unlabeled examples and train a classifier with the unlabeled examples interpreted as weighted negative examples.

We believe that the modeling choices involved in such systems can be fairly nuanced and crucial for achieving good test performance. Our approach is simpler to understand, faster to train, has no "knobs" to tweak and results in very similar performance compared to traditional discriminative classifiers trained using positives and negatives. Therefore, it is appealing from the perspective of integration into a real-world engineering system.

Our learning methodology is very similar to the framework of [11]. Interestingly, the core contribution of [11] is unrelated to the problem we are interested in solving in this paper. Their motivation was to develop a way of quickly training linear classifiers in HOG space without resorting to the more conventional and expensive SVM training (expensive due to the often time consuming step of hard mining negatives). They achieve this by training a linear discriminant for each foreground class of interest in a generative fashion (LDA). Instead of building a background model for each class, each of the foreground classes trained in [11] relies on a common background model that is shared across all classes. Importantly, the background model is a relatively simple to train Gaussian distribution trained on a large collection of **unlabeled** images. It is this LDA training methodology with a common background model that we adopt to solving our problem.

However, HOG as a feature does not lend itself well to our problem. It is well known that different viewpoints for the same category induce very different appearances in HOG space. This necessitates the use of mixture models. In fact, [11] train LDA classifiers for different mixtures of a single category, where mixtures correspond to clusters in whitened HOG space. However, when it comes to aggregating these mixtures into a single category level classifier, they find it necessary to calibrate these mixtures in order to compete them against one another at test time. The calibration (similar to Platt's scaling) is done by two levels of discriminative training (SVM + logistic regression) on a set of mixture specific positives and a set of negatives. Our setting is not amenable to such forms of calibration, because we **do not** have any class specific negatives to begin with. Ideally, we would like to operate in a feature space that obviates the need for mixture models; a feature space that *implicitly* captures changes in appearance without the need to *explicitly* model the modes in distribution. The argument for calibrating within-class LDA mixtures in HOG space can be extrapolated to LDA models across classes. Therefore, we look beyond HOG as a features. Our experimental results demonstrate that using the LDA framework of [11] with HOG does poorly in the absence of calibration.

One of the biggest breakthroughs in vision in the last couple of years has been the staggering success of multi-layered Convolutional Neural Networks on various benchmarks, a wave that was initiated by the phenomenal suc-

cess shown by these models on the ImageNet competition [12]. Judging by the performance of CNNs on ImageNet, they seem to exhibit several desirable invariance properties from a classification standpoint (scale, color, viewpoint, 3D pose, etc.). This leads us to believe that these ImageNet trained CNNs satisfy our requirement of *implicitly* capturing changes in appearance without the need to *explicitly* model the modes in distribution. Curiously, researchers have shown that the features generated by a CNN pre-trained on ImageNet achieve "astounding" results [16] when applied on a novel dataset and a novel classification problem. Others have also shown clever ways of adapting the ImageNet CNN to a new domain by swapping out the last two layers of the Krizhevsky architecture with domain-specific layers [14]. Unfortunately, a bottleneck associated with such forms of adaptation is that it requires examples from an explicit "negative" class (i.e. images that do not belong to any of the positive classes in the new domain), precisely the sort of bottleneck we want to overcome.

Therefore, instead of resorting to any fine-tuning, we follow [16] and use features that come from a pre-trained CNN [17]. We combine these features and train LDA classifiers for each positive class using a common background model. Like [11], our background model is built from a large collection of unlabeled images. Note that our unlabeled dataset can contain images of several classes, **including** the positive class. As shown in the figure, our classifier is based on LDA, wherein the only parameter computed online is the mean of the foreground class. The covariance and the mean of the unlabeled set are computed offline and are re-used each time a classifier for a new foreground class needs to be trained.

In the context of using CNN features off-the-shelf, [16] showed promising results on various classification and retrieval benchmarks using Overfeat [17]. Their classifiers were discriminatively trained SVMs using positives and class-specific negatives. To the best of our knowledge, we are the first to show competitive classification performance using CNN features trained generatively and without any class specific negatives. As part of our contribution, we show that using Overfeat's 1000-dimensional output layer (corresponding to the 1000 ImageNet categories) as a feature vector works favorably compared to using the 4096-dimensional fully connected layer.

3. Generative Classifiers using LDA

LDA classifiers have been a widely used tool in Vision, especially for the problem of Face Recognition. Examples include [2, 3, 4, 9]. The core mathematical idea used in such examples is an extension of the Fisher linear discriminant to K classes. The Fisher linear discriminant is defined as the linear function $\mathbf{w}^T\mathbf{x}$ (parameterized by $\mathbf{w}$) that maximizes the criterion $J(\mathbf{w}) = \frac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$, where the matrix S_w is the so-called "within" class scatter and the matrix S_b is the "between" class scatter matrix. Finding the optimal solution $\mathbf{w}^*$ that maximizes $J(\mathbf{w})$ amounts to solving a generalized eigenvalue problem, and when the data comes from two classes, the optimal solution is given by $\mathbf{w}^* = S_w^{-1}(\mu_{pos} - \mu_{neg})$ where μ_{pos} and μ_{neg} are the means of the positive and negative class respectively.

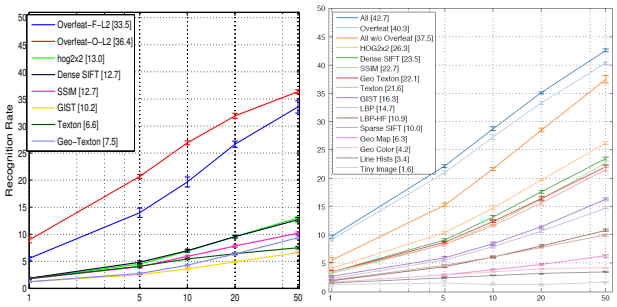
The extension to K classes is an optimal projection matrix $W^* = \left[\mathbf{w}_1^* | \mathbf{w}_2^* | \dots | \mathbf{w}_{K-1}^*\right]$. The columns of W^* are the eigenvectors of the $K-1$ largest eigenvalues of the matrix $S_w^{-1}S_b$, where the between and within class matrices S_b and S_w now decompose over the K classes.

Therefore, multi-class LDA assumes that K disjoint sets of positive examples for each of the K classes are available, an assumption that may not always be practical. Instead, we choose to train K binary classifiers, which is more consistent with our use case wherein positive examples arrive in batches (one class at a time).

For binary classification, when the class conditional distributions are modeled as Gaussian and the two classes share a common co-variance matrix, the optimal LDA classifier is given by $\mathbf{w}^* = \Sigma^{-1}(\mu_{pos} - \mu_{neg})$, where Σ is the shared covariance across the two classes. Note that even in the binary setup, the negatives are chosen so as to not contain instances of the positive class. However, as pointed earlier, one of the core insights of [11] was to show that competitive binary LDA classifiers can indeed be trained without class specific negatives (as long as some form of calibration using positives and negatives is done as a post-processing step). In particular, they train an optimal binary classifier as $\mathbf{w}_{pos}^* = \Sigma_{bg}^{-1}(\mu_{pos} - \mu_{bg})$, where bg denotes a large common pool of unlabeled images. This is the regime we operate in as well, as shown in Fig. 1, except that our classifiers are trained in the space of CNN features, and require no extra calibration.

4. De-correlated CNN for Scene Classification

To evaluate the central hypothesis of this paper, we compare the performance of our generative classifiers (trained using LDA) on Overfeat features with those that are trained discriminatively (using SVMs). Additionally, we are also interested in comparing Overfeat features to a host of other features that have been traditionally used for classification, under both, the discriminative as well as generative settings. Our experiments are conducted on the SUN-397 dataset that was introduced by Xiao et al. [19]. This dataset contains 108,754 images from 397 well-sampled scene categories with at least 100 images per category. Importantly, there is a sufficient disconnect between the 1000 ImageNet categories on which Overfeat was trained and the 397 SUN categories. The dataset is divided into 10 overlapping partitions with each partition containing 50 training and 50 test images for each of the 397 categories. The evaluation pro-



(a) (b)

Figure 2. Recognition performance on the SUN-397 scene dataset using (a) generatively and (b) discriminatively (*reproduced from Fig.17(b) in Xiao et al. [19]*) trained features.

tocol demands that the average recognition performance on the test sets across the 10 partitions be reported. Other details can be found in [19] as well as the authors' web-page.

Generative vs. Discriminative: Fig. 2 compares the performance of our generatively trained classifiers (Fig. 2(a)) to the results reported by [19] based on discriminatively trained classifiers (Fig. 2(b)). Each plot in both, Figs. 2(a)&2(b) corresponds to a particular feature. Fig. 2(b) has been reproduced from a recent journal submission of [19] that is publicly available on the primary author's web-page. We briefly explain how to interpret the plots. The x-axis represents varying number of training examples per category, keeping the test set constant, and the y-axis represents the average accuracy and standard deviation across the 10 splits. For instance, for the point $n = 20$ on the x-axis, the following experiment will be repeated for all the 10 splits for any given feature:

- Pick the first 20 examples from each of the 397 categories for training, while all 50 test examples per category in the given split are picked for testing. Train 397 binary classifiers and evaluate them in a one-vs-all fashion on the test set. Compute the average K-way classification accuracy from the class-confusion matrix (call it split-specific-accuracy).

The point and the error-bar on the y-axis for $n = 20$ represents the average accuracy and the standard deviation over the 10 split-specific-accuracies. To be consistent with [19], our generative classifiers are evaluated in a one-vs-all fashion i.e. if $\mathbf{x}^*$ represents a test instance, then its class label y^* is given as $y^* = \arg_k \max(\mathbf{w}_k^T \mathbf{x}^*)$ where

$$\mathbf{w}_k = \Sigma_{bg}^{-1}(\mu_{fg,k} - \mu_{bg}) \quad (1)$$

is the LDA classifier for the k^{th} class and $\mu_{fg,k}$ is the mean of the k^{th} foreground class.

http://vision.princeton.edu/projects/2010/SUN/

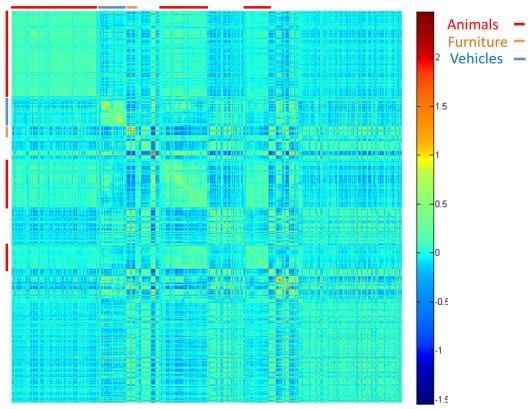


Figure 3. Correlations learned from the final layer of the Overfeat features on the SUN-397 scene dataset.

Discriminative Features: We refer the reader to [19] for a detailed description of the features used in Fig. 2(b). Note that for Overfeat, the authors work with the 4096 dimensional vector produced by the fully connected layer of the CNN. For every plot shown in Fig. 2(b), a one-vs-all SVM was trained by combining histogram intersection kernels on appropriately normalized features.

Generative Features: We build and evaluate generative classifiers for two variants of Overfeat features in addition to the six best performing features from Fig. 2(b). For these six other features, we use the pre-computed descriptors made publicly available by the authors of [19] on their web-page. Our training of LDA classifiers on these six other features is done based on (1) where the various terms of the equation are computed in the feature space of interest. The background model (Σ_{bg} and μ_{bg}) for all features is built using all the 108,754 images from the SUN dataset.

We briefly describe the features plotted in Fig. 2(a):

- **Overfeat-F-L2:** The 4096 dimensional vector produced by Overfeat followed by L_2 normalization. L_2 normalization was also suggested in [16].

- **Overfeat-O-L2:** The 1000 dimensional vector produced by the **output** layer of Overfeat followed by L_2 normalization. We believe we are the first to publicly report results using the **output** layer from an off-the-shelf CNN trained on ImageNet.

- **Rest of the features:** For reference, we note that the discriminative training of the six other features we consider (as reported in Fig. 2(b)) is based on training SVMs using histogram intersection kernels (details in [19]). In some cases, the histograms at multiple spatial levels are combined using a weighted combination of intersection kernels. For building generative classifiers, we simply concatenate the histograms at possibly

multiple levels of an image into a single vector. Note that the plot for HoG in Fig.3 a is a proxy for how a direct application of [11] would perform on this problem without any calibration. In fact, the plot corresponds to a BoW model, whereas [11] uses rigid templates. So in some sense, the plot is an upper bound on what the rigid templates of [11] would have done, showing the value of replacing HOG with CNN features.

One of the benefits of using the 1000-D output layer of the CNN as a feature is that it allows a direct mapping of the correlations captured by Σ_{bg} to the 1000 semantic ImageNet categories that the network was trained on. Fig. 3 is a visualization of Σ_{bg}. We note that the 1000 ImageNet categories exhibit a "banded" structure. For example, the first 200 or so categories are all animals, the next few categories are all vehicles, and so on. From Fig. 3, we see that the animal categories are strongly correlated to each other (the big block on the top left), and so are the vehicle categories (the next small block around the diagonal). Similarly, the second red "band" of animal categories is strongly decorrelated with the furniture categories. Note that the figure is **not** capturing ground truth correlations in ImageNet data. Instead, it is characterizing the behaviour of a pre-trained CNN on a novel set of images (the SUN dataset in this case), a dataset whose class labels are reasonably different from the ImageNet labels. It is these correlations (both, positive and negative) that guide the LDA classifier trained on CNN features.

To extract features from Overfeat, the input image is resized to 221×221 as mentioned in [16]. The six best performing features from Fig. 2(b) show a dramatic drop in performance when trained and evaluated using LDA with a common background. Overfeat features on the other hand, show little drop. Overfeat was pre-trained in a supervised manner on more than a million images from ImageNet. We feel this makes them more resilient to training without class-specific negatives compared to other hand engineered features. Note that while the discriminative SVM classifiers would have required explicit class specific negatives and possibly a few hours to train, the generative classifiers are instantaneous to train. Unlike [11], our CNN classifiers report competitive performance without any post-processing involving a calibration across the K classes (we suspect that such additional calibration using positives and negatives can only improve our results).

Role of Correlations: How important are the correlations captured by Σ_{bg} for overall recognition accuracy? To address this question, we begin by rewriting the LDA classifier for class k as $\mathbf{w}_k = (\Sigma_{bg} + \lambda I)^{-1}(\mu_{fg,k} - \mu_{bg})$ where λ is a scalar and I is the Identity matrix. When $\lambda = 0$ we get back the original equation (1). However, as λ increases, $\Sigma_{bg} + \lambda I$ comes close to becoming an isotropic covariance matrix, and the decision boundary captured by $\mathbf{w}_k$ is simply a hyperplane located halfway along the vector $\mu_{fg,k} - \mu_{bg}$ and perpendicular to it. Fig. 4 shows the adverse impact of reducing the LDA model's ability to capture these correlations on the overall accuracy.

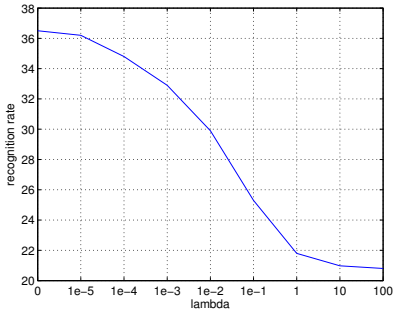
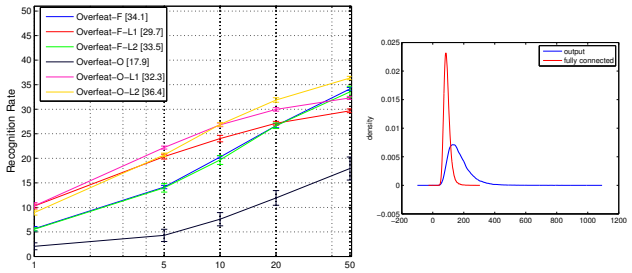


Figure 4. Importance of correlations captured by Σ_{bg}. As λ increases, $\Sigma_{bg} + \lambda I$ comes close to becoming an isotropic covariance matrix, reducing the ability of the LDA classifier to benefit from the correlations captured by Σ_{bg}, reducing the overall recognition accuracy. The recognition rate shown corresponds to the Overfeat-O-L2 features from Fig. 2.

Role of Normalization: We show the effect of normalizing the CNN features (both the output layer and the fully connected layer) prior to training the generative classifiers in Fig. 5(a). We see that normalizing the 1000-D output layer substantially improves the results compared to normalizing the 4096-D fully connected layer. We believe this is due to the fact that the norm values of the 4096-D feature exhibit a much smaller range compared to the norm values of the 1000-D feature (see Fig. 5(b)) so the latter benefits more from normalization. Note that from a practical perspective, although normalizing features introduces extra computation, it is trivial in comparison to the time spent in extracting the CNN features (for instance, Overfeat takes about 2-3 seconds on a 221×221 image on a modern CPU).

Choice of Background Model: Fig. 2(a) uses all of SUN data to build the background model. To understand the role of dataset bias in building the background model and its impact on classification performance, we repeated the experiments conducted in generating Fig. 2(a) based on background models created under two additional settings: (a) Use the entire PASCAL 2011 dataset (14000 images), and (b) Use data from the same split in SUN that contained the 50 training examples for each class (19850 images). The foreground model training and the evaluation protocol was kept the same as the setup used in generating Fig. 2(a).

Fig. 6 plots the classification accuracy w.r.t. these different background models. We notice that when the background model is built from PASCAL, there is indeed some drop in performance. This is not all that surprising, given



(a) (b)

Figure 5. Effect of normalization (L_1 and L_2) of CNN features for LDA classifiers: (a) Recognition performance on the SUN-397 scene dataset. While normalization adds little to classifiers built on the fully connected layer, when using the 1000-D output layer for classification, L_2 normalization doubles the accuracy. See text for details. (b) Kernel density estimates computed on the L_2 norms of CNN features extracted from the fully connected layer (4096-D) vs. the final output layer (1000-D) on all of SUN dataset. The range of norm values for the output layer is much greater than the range of values for the fully connected layer.

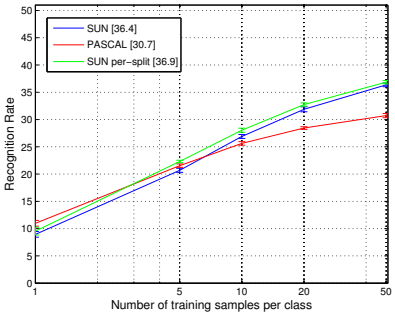


Figure 6. Recognition performance on the SUN-397 scene dataset using LDA-Overfeat features trained with different background models.

Table 1. Average Binary Classification Results

CNN	95.7%	Geo-Textons	83.8%
HOG2x2	90%	SelfSim	89%
Dense SIFT	89.6%	GIST	83.4%
Texton	88.5%		

that the PASCAL dataset has 20 classes and is more of a detection benchmark, whereas the SUN dataset has 397 scene categories. Therefore, the behavior of Overfeat on the two datasets as captured by the respective covariance matrices is not the same.

Binary Classification Performance: We also evaluate our LDA classifiers in a purely binary classification regime. For this experiment, we randomly picked one of the ten SUN splits. Recall that a single split has 19850 images (397 SUN classes $\times$ 50 images per class) for both training and testing. We used the 50 training images for each class to build 397 binary classifiers using LDA. These binary classifiers were independently evaluated on the test set, i.e. they were not competing against each other, but producing a classification score on each test image. The area under the ROC curve on the 19850 test images (50 true positives and 19800 false positives) was computed for each of the 397 classes. Table-1 reports the average AUC across the 397 classes for various features. For binary classification, the LDA classifiers show less variation in performance across the features, even though CNN features still perform the best. Given that binary classification is a much simpler problem, LDA classifiers designed using any reasonably well engineered feature results in decent performance. However, when classifiers have to compete (as is the case with K-way classification), LDA classifiers for non CNN features suffer as a result of no proper calibration between them, while the LDA classifier built using CNN features appears to be robust to such issues.

Figure 7. Qualitative Results of applying LDA-CNN classifiers for two categories from the SUN dataset: *Rafting* (top row) and *Railway Train* (bottom row). The left column shows 20 out of the 50 training images that went into training the classifier. The right column shows the top 20 out of the 19850 test images (397 SUN classes $\times$ 50 images per class) ranked left to right, top to bottom by the classification score. Green boxes indicate true positives, red boxes indicate false positives.

In Fig. 7, we show qualitative results that depict the ranked list of images obtained by applying the LDA-CNN classifier trained on 50 positive images from two of the SUN categories viz. Rafting and Railway Train. Notice that some of the false positives are an artifact of how images in SUN have been labeled. For instance, *Railroad Track* is a separate category from *Railway Train*.

5. De-correlated CNN for Instance Retrieval

In addition to classification, CNN features have shown promise for the instance retrieval task as well. In particular, Razavian et al. [16] demonstrated a performance improvement over other low memory footprint methods by using the Overfeat-F-L2 features. While they discussed specific tricks like spatial-search and feature-augmentation to improve on the overall performance, we will focus on com-

plementary query expansion techniques [6, 5, 1] to demonstrate that a decorrelated version of the same CNN features can lead to a significant performance boost without relying on negative data typically used for the discriminative version of query expansion.

Let $\mathbf{f}_q$ be the CNN feature extracted from the query (region-of-interest) and $\mathbf{f}_i$ be the corresponding feature extracted from reference image i for $i \in \{1, \ldots, N\}$ in the database. A ranked list $\mathcal{N}_q = \{i_1, i_2, \ldots, i_N\}$ can be generated by simply ordering the database images in increasing order of the L2 distance $||\mathbf{f}_q - \mathbf{f}_i||_2$. We will refer to this scheme as the nearest-neighbor (NN) ranking method. Given a NN ranked list of images, two query expansion approaches have been suggested in the literature to improve retrieval accuracy:

Average Query Expansion (AQE): In this approach [6, 5], the query feature $\mathbf{f}_q$ is averaged with the features from the top-K retrieved images in the NN-set $\mathcal{N}_q$ to generate a new query feature, $\mathbf{f}'_q = \frac{1}{K+1}\left(\mathbf{f}_q + \sum_{k=1}^{K} \mathbf{f}_{i_k}\right)$. This feature is then used to re-query the database using the NN scheme to generate the output ranked list.

Discriminative Query Expansion (DQE): In this approach [1], instead of simply averaging the features from the top-K retrieved images, a discriminative linear-SVM classifier is trained by using these features $\mathcal{F}_+ = \{\mathbf{f}_q, \mathbf{f}_{i_1}, \ldots, \mathbf{f}_{i_K}\}$ as positive samples and features from M images at the bottom of the NN-list $\mathcal{F}_- = \{\mathbf{f}_{i_{N-M+1}}, \ldots, \mathbf{f}_{i_N}\}$ as negative samples. This classifier $\mathbf{w_D}$ is then applied to the database image features and the output ranked list is generated by sorting on the value $\mathbf{w_D}^T\mathbf{f}_i$.

Generative Query Expansion (GQE): In this paper, we propose a novel query expansion approach that does not rely on the negative set $\mathcal{F}_-$. Instead, we use the LDA-based approach to directly learn a generative classifier from the positive set $\mathcal{F}_+$ by first computing the foreground mean μ_{fg} as the mean of the features in $\mathcal{F}_+$ and then applying (1) with the pre-computed background model μ_{bg} to compute the weight vector $\mathbf{w_G}$. This classifier $\mathbf{w_G}$ is then applied to the database image features like in the DQE case, and a ranked list is generated by sorting on the value $\mathbf{w_G}^T\mathbf{f}_i$.

We demonstrate the advantage of using de-correlated CNN features in the generative query expansion framework on the widely-used Paris6k buildings dataset [15]. This dataset consists of 6412 images of various buildings and monuments from Paris and has a pre-defined set of 55 queries for evaluation. To baseline the performance of CNN features for instance retrieval on this dataset, we follow the same convention as [16] and extract Overfeat-F-L2 features from the smallest square containing the query region-of-interest. No region-of-interest or spatial-gridding is employed for the reference database images.

Fig. 8 plots the mean average precision (mAP) achieved

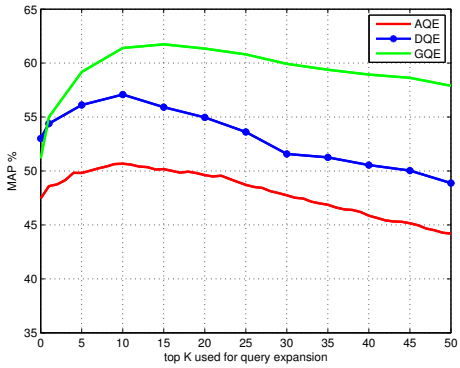


Figure 8. Performance evaluation of generative query expansion using de-correlated CNN features for instance retrieval on the Paris6k dataset.

using each of the above query-expansion techniques with varying number K of the positive samples used to create the positive set $\mathcal{F}_+$. The number of negative samples M was kept fixed at $M = 200$ to be consistent with the DQE settings suggested in [1]. We tried different values for the parameter C (the regularization parameter for SVM training) for DQE, and plot the performance for C=1, which we found to be the best. We can observe that the DQE approach leads to a higher mAP initially when using only the top 2 positive samples but the GQE quickly surpasses this performance as more positive samples are added to the training data. Additionally, GQE is more robust to outliers in the positive set as it retains a higher mAP for a higher K while the performance of DQE quickly degrades.

In Fig. 9, we show qualitative results on a query image (Notre Dame) from the Paris6k dataset. In each row, we show the top-20 images retrieved from the dataset by using one of the methods described above for the same query. The first row shows the results with the NN ranking method and exhibits a number of false-positives (shown as red boxes). From this list, we pick the first 10 results (thus $K = 10$ per the above description) to apply the different query expansion methods. In the second row, application of AQE leads to a slight improvement in the NN results. The third and fourth rows show that the results of DQE and GQE are significantly better, with the GQE method removing all the false-positives for this query. Note that this is the case even when the top-10 NN list (i.e. the positive training set $\mathcal{F}_+$) has a number of outliers showing the robustness of our generative approach.

6. Conclusion

We propose a practically appealing solution to the problem of training a classifier from a collection of positive and unlabelled data by combining the LDA classifier training method of [11] with off the shelf CNN features. Our approach shows promising results on a large and challeng-

Figure 9. Results for a Notre Dame query on the Paris6k dataset. In each row, the left-most image is the query followed by retrieved results ranked left-to-right. Green boxes indicate true positives and red boxes indicate false positives.

ing Scene Classification benchmark as well as on the problem of Query Expansion for Instance Retrieval. On Scene Classification, our method does nearly as well as discriminatively trained methods using the same positive examples and a large collection of negative examples. In the space of CNN features, our approach to Query Expansion shows significant improvement over Average Query Expansion (AQE) and Discriminative Query Expansion (DQE). We believe that the proposed approach to Query Expansion presents an interesting avenue that needs to be explored in more detail. For example, this technique can be applied even on Bag of Words models that operate in other feature spaces. Understanding the relative merits of various features on which these generative models are trained, as well the charactersistics of the datasets on which they shine compared to AQE and DQE are questions worth exploring in their own right, and we see them as future work.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918. IEEE, 2012. 7

[2] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE TPAMI*, 1997. 3

[3] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In *CVPR*, 2005. 3

[4] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new lda-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 2000. 3

[5] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall ii: Query expansion revisited. In *CVPR*, pages 889–896. IEEE, 2011. 7

[6] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, pages 1–8. IEEE, 2007. 7

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, June 2005. 1

[8] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2008. 2

[9] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *Journal of Optical Society of America A*, 1997. 3

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE TPAMI*, 32, 2010. 1

[11] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012. 1, 2, 3, 5, 7

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, page 2012. 3

[13] W. S. Lee and B. Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *International Conference on Machine Learning (ICML)*, 2003. 2

[14] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 3

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, pages 1–8. IEEE, 2007. 7

[16] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014. 3, 4, 5, 6, 7

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR 2014)*, 2014. 3

[18] C. Wang, C. H. Q. Ding, R. F. Meraz, and S. R. Holbrook. Psol: a positive sample only learning algorithm for finding non-coding rna genes. *Bioinformatics*, 2006. 2

[19] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 3, 4

[20] H. Yu. Single-class classification with mapping convergence. *Machine Learning*, 2005. 2

[21] H. Yu, J. Han, and K. C.-C. Chang. Pebl: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 2004. 2