Factorized Convolutional Networks: Unsupervised Fine-Tuning for Image Clustering

Liang-Yan Gui* Liangke Gui* Yu-Xiong Wang Louis-Philippe Morency José M. F. Moura
Carnegie Mellon University
{lgui,liangkeg,yuxiongw,morency,moura}@andrew.cmu.edu

Abstract

Deep convolutional neural networks (CNNs) have recognized promise as universal representations for various image recognition tasks. One of their properties is the ability to transfer knowledge from a large annotated source dataset (e.g., ImageNet) to a (typically smaller) target dataset. This is usually accomplished through supervised fine-tuning on labeled new target data. In this work, we address "unsupervised fine-tuning" that transfers a pre-trained network to target tasks with unlabeled data such as image clustering tasks. To this end, we introduce group-sparse non-negative matrix factorization (GSNMF), a variant of NMF, to identify a rich set of high-level latent variables that are informative on the target task. The resulting "factorized convolutional network" (FCN) can itself be seen as a feed-forward model that combines CNN and two-layer structured NMF. We empirically validate our approach and demonstrate state-of-the-art image clustering performance on challenging scene (MIT-67) and fine-grained (Birds-200, Flowers-102) benchmarks. We further show that, when used as unsupervised initialization, our approach improves image classification performance as well.

1. Motivation

Advances in computer vision and machine learning, especially deep convolutional neural networks (CNNs), have relied on supervised learning and availability of large-scale annotated data. In practice, however, collecting such massively annotated training data for new categories or tasks of interest is typically unrealistic. Fortunately, when trained on a large enough, diverse "base" set of data (*e.g.*, ImageNet), CNNs exhibit certain attractive transferability properties for a broad range of tasks [38, 53]. This suggests that CNNs could serve as universal representations for novel categories and tasks.

*The first two authors contributed equally.

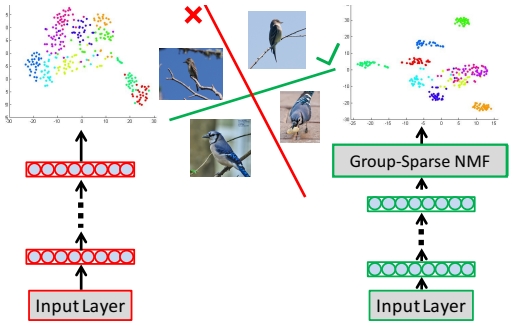


Figure 1: Unsupervised transfer of pre-trained CNN representations to novel target tasks with *unlabeled data* via a factorized convolutional network (FCN). Off-the-shelf features that are extracted from CNNs pre-trained on ImageNet are limited to describing subtle differences among novel fine-grained categories, as visualized by embedding the features in a 2-dim space via t-SNE [44] (**left**). By leveraging group-sparse non-negative matrix factorization (GSNMF), *unsupervised fine-tuning* is accomplished and features extracted from the resulting GSNMF-FCN model lead to more discriminative clusters (**right**). We thus learn a better representation with enhanced transferability for target tasks with unlabeled data, in which conventional supervised fine-tuning with back-propagation is inapplicable.

Unsupervised fine-tuning: Fine-tuning is by far the dominant strategy for transfer learning with neural networks [38, 34, 1, 46]. This approach was pioneered in [20] by transferring knowledge from a generative to a discriminative model, and has since been generalized with great success [15, 54]. The basic pipeline involves replacing the last "classifier" layer of a pre-trained network with a new randomly initialized layer for the target task. The modified network is then fine-tuned with additional passes of appropriately tuned gradient descent on the target training set. Even though its use is widespread, fine-tuning requires *annotated* target data, and we use the term "supervised fine-

tuning" to refer to this conventional paradigm. However, in scenarios where there are no labeled images for novel categories or tasks (*e.g.*, in image clustering applications), such supervised fine-tuning is inapplicable and how to best adapt a pre-trained CNN still remains an open challenge. Hence, we propose "unsupervised fine-tuning" *as a new paradigm* to address this issue.

Factorized convolutional networks: To this end, we transfer knowledge *from a discriminative to a generative model* and explore "factorized convolutional networks" (FCNs) that *fine-tune the pre-trained CNN representations in an unsupervised manner*. Given unlabeled target images, a factorization of the CNN representations is learned using low-rank and group-sparsity constraints. Inspired by the success of non-negative matrix factorization (NMF) [26, 47] in clustering applications, we introduce a novel NMF based adaptation module with a generative loss that can be plugged into any standard CNN to facilitate the desired unsupervised transfer. As a classic multi-variate analysis technique, the appeal of NMF is the ability to disentangle exploratory factors of variations underlying unlabeled, non-negative data samples as well as the inherent clustering property. Intuitively, the CNN activations of interest are those after the rectified linear units (ReLUs), which consistently show better recognition performance for various tasks and which are also non-negative. It is thus natural to investigate NMF techniques on top of CNN activations for image clustering, as shown in Figure 1.

Group-sparse non-negative matrix factorization: More precisely, our key insight is to effectively adapt between the source and target tasks by both utilizing generic statistics learned from a large corpus of labeled source images through CNNs and separating out the current underlying factors of variation relevant to the observed, unlabeled target data via NMF. To better select a group of correlated CNN activations, we propose a variant of NMF — group-sparse NMF (GSNMF), which identifies a rich set of informative and discriminative latent variables across tasks. Given that NMF/GSNMF could also be interpreted as a two-layer neural network [26], our GSNMF based FCN is then regarded as a principled feed-forward model. This allows to fine-tune the resulting augmented architecture (*i.e.*, modifying the CNN parameters as well) on the target task with respect to a NMF based objective using stochastic gradient descent and back-propagation.

Contributions: Our contributions are four-fold. (1) Different from the conventional strategy that transfers knowledge from a generative to a discriminative model [20], we propose a novel way of CNN transfer — supervised, discriminative pre-training and then unsupervised, generative fine-tuning. (2) Based on this general principle, we show how factorized convolutional networks (FCNs), which combine NMF and pre-trained CNN, learn a more generic feature representation across tasks. (3) We show how to explicitly enforce group-sparsity on FCN to better leverage the correlation of CNN activations by introducing elastic net regularization into NMF. (4) Our unsupervised fine-tuning is general; it could be used in image clustering tasks and also used as unsupervised initialization to further improve classification tasks. Finally, to the best of our knowledge, we are the first to evaluate the performance of image clustering on challenging large-scale scene and fine-grained recognition datasets, producing state-of-the-art results.

2. Related Work

Unsupervised feature learning: Unsupervised feature learning focuses on discovering low-dimensional features that capture some structure underlying the high-dimensional unlabeled data. Classic approaches include principal component analysis (PCA) [23], independent component analysis (ICA) [22], and locally linear embedding (LLE) [39]. Inspired by the hierarchical architecture of the neural system, many new schemes that stack multiple layers of simple learning blocks, such as sparse coding [27], restricted Boltzmann machines (RBMs) [19], auto-encoders [16], and NMF [26], have been proposed to build deep representations [48, 28]. One similar work is the deep semi-NMF model, which stacks semi-NMF together to learn low-dimensional feature representations [43]. Another similar work is the deep linear discriminant analysis model, which projects high-dimensional observations to linearly separable representations [10].

Different from the previous work, we combine a NMF layer with a pre-trained deep CNN and use the reconstruction error as the objective function to fine-tune the network for unsupervised learning. We extend work [17] reported in the previous workshop in three important ways. (1) [17] is pipelined, while ours is end-to-end. [17] simply applies NMF on top of *off-the-shelf* CNN features, in which NMF reduces the dimension of *fixed* CNN features. In contrast, ours is more general and introduces NMF (and its variant) as a feature reconstruction (generative) loss for unsupervised CNN fine-tuning. Ours thus integrates NMF and CNN as a principled feed-forward network, and allows for fine-tuning the full network with back-propagation. As shown in the following sections, we not only learn the NMF adaptation layers, but also *modify (a portion of) the pre-trained CNN weights using the generative loss* towards the target task. Due to the end-to-end nature, ours is more flexible and achieves better performance. (2) [17] can only deal with image clustering. In contrast, due to the end-to-end nature, ours could be also used as *unsupervised initialization* and *improves image classification* on target tasks. (3) We have substantially extended experimental results, including more datasets, more baselines, different clustering techniques (k-means and spectral clustering), additional hyper-parameter

analysis, and ablation analysis.

Domain adaptation and transfer learning: Another related line of work focuses on standard domain adaptation with the assumption that the data from source and target datasets share the same set of categories but have shifted distributions [13, 14]. Our work, however, does not have this assumption and addresses a more general, challenging task (*i.e.*, different but relevant source and target categories/tasks). The learning processes are different as well. References [13, 14] explicitly use the source labels to infer the target labels. In contrast, we transfer a pre-trained (ImageNet) network to unsupervised target tasks and do not use the source labels in this process. Besides, [13, 14] are evaluated on image classification tasks while our work is mainly evaluated on image clustering tasks. The reconstruction loss used in [14] is also different: [14] simply reconstructs target raw images with the mean squared error loss, whereas we reconstruct the learned CNN features and leverage their non-negativity. More recently, a recurrent network with a single loss function is proposed to guide the agglomerative clustering [52]. While this work uses different network architectures for different datasets to train a dataset specific model, it fails to address the subtle difference among fine-grained categories. Different from this work, our model uses the same parameters and network architecture for all datasets, leading to a more universal feature representation.

In transfer learning, the target task is different from but related to the source task [35], such as transfer from object-centric source categories to scene-centric target categories or from coarse source categories to fine-grained target categories. The standard fine-tuning strategy [1] and its variants [29, 46] in supervised transfer are inapplicable here since they require a significant amount of labeled target data, which is simply not available. For novel categories, effective unsupervised transfer of CNN representations remains an open challenge [45].

Image clustering: Different from previous work [12, 42, 7], we use CNN features and evaluate our model on both standard image clustering datasets and large-scale image classification datasets. The latter datasets still remain challenging even for (supervised) image classification tasks. In related work, an ensemble of image prototype sets is sampled from the available data to represent a rich set of visual categories, and images are projected onto these prototypes as new feature representations [6]. Unlike [6], which takes advantage that the test data is used as unlabeled data for training (*i.e.*, transductive learning), we follow strict train/test splits for each dataset to ensure the generalization of our approach. Direct clustering in a pre-trained, fixed supervised CNN feature space [6] is simple but sub-optimal due to domain shift. Performing clustering through unsupervised deep feature learning provides an attractive option [50]. However, the performance of the unsupervised deep models is still not on par with that of their supervised counterparts. On the contrary, we leverage both supervised CNN feature learning and unsupervised transfer learning.

3. Factorized Convolutional Networks

Let us consider a CNN architecture pre-trained on a source domain with abundant data, for example the vanilla VGGNet [41] pre-trained on ImageNet (ILSVRC) $1{,}000$ categories [40]. The CNN is composed of a feature representation module $\mathcal{F}$ (*e.g.*, the 13 convolutional layers $C1$-$C13$ and two fully-connected layers $fc6$, $fc7$ for VGGNet) and a classifier module $\mathcal{C}$ (*e.g.*, the last fully-connected layer $fc8$ with $1{,}000$ units and the $1{,}000$-way softmax for ImageNet classification) [46].

We now transfer this CNN for representation learning on unlabeled target images in tasks such as image clustering. The transfer is accomplished through our unsupervised fine-tuning and the target CNN is instantiated and initialized in the following way, as shown in Figure 2: (1) the representation module $\mathcal{F}_T$ is copied from $\mathcal{F}_S$ of the source CNN with the parameters $\Theta_T^{\mathcal{F}} = \Theta_S^{\mathcal{F}}$, and (2) the classifier module $\mathcal{C}$ is removed and a new "adaptation module" $\mathcal{A}$ is introduced that consists of a group-sparse non-negative matrix factorization (GSNMF) on top of $fc7$ activations.

Note that the unsupervised fine-tuning is different from conventional supervised fine-tuning; in the latter case, a new classifier module $\mathcal{C}_T$ (*e.g.*, a new $fc8$ and softmax) is introduced with the parameters $\Theta_T^{\mathcal{C}}$ randomly initialized.

As a complex non-linear function of all input pixels, the $fc7$ representation may capture mid-level object parts as well as their high-level configurations [34]. Our GSNMF module then reduces feature dimension and enlarges sparsity among different groups simultaneously, thus identifying a rich set of informative latent variables useful for unsupervised adaptation. The GSNMF module is trained while a portion of the parameters $\Theta_T^{\mathcal{F}}$ are optionally fine-tuned (depending on the amount of available data) by continuing the back-propagation.

3.1. NMF/GSNMF Module

We consider an M dimensional random vector $\mathbf{x}$ with non-negative elements, *e.g.*, the CNN $fc7$ activations in our case. Its *N* observations are denoted as $\mathbf{x}_i, i = 1, 2, \ldots, N$. N is the batch size in our stochastic optimization. Let the data matrix be $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}_{\geq 0}^{M \times N}$. NMF seeks a non-negative basis matrix $\mathbf{W} \in \mathbb{R}_{\geq 0}^{M \times L}$ and a coefficient matrix $\mathbf{H} \in \mathbb{R}_{\geq 0}^{L \times N}$ such that

$$\mathbf{X} \approx \mathbf{W}\mathbf{H}. \tag{1}$$

Usually *L* $\ll$ *min(M,N)*.

While classic NMF is able to identify informative latent variables, it is commonly known that large deep neural networks typically are comprised of many redundant and

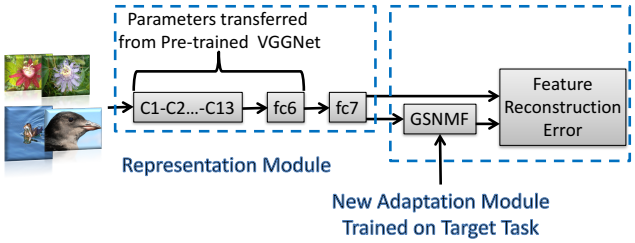


Figure 2: Illustration of unsupervised fine-tuning and factorized convolutional networks. A network (*e.g.*, VGGNet) is trained on the source task (*e.g.*, ImageNet classification) with a large amount of labeled images. The pre-trained parameters of its feature representation module ($C1$-$C13$ and $fc6$, $fc7$) are then transferred to the target task with unlabeled data (*e.g.*, image clustering). In such unsupervised scenario, we introduce a new "adaptation module" that consists of a group-sparse non-negative matrix factorization (GSNMF) on top of $fc7$ activations to compensate for the different image dataset statistics (*e.g.*, type of objects, typical viewpoints) between the source and target data. We then train the GSNMF module while fine-tuning the representation module based on the GSNMF reconstruction (generative) loss using the unlabeled target image data.

highly correlated units [21]. Hence, to better select a group of correlated CNN activations, we enforce additional group-sparsity constraints on NMF. The joint ℓ_1 and ℓ_2 norm penalty, *i.e.*, elastic net regularization, has been widely used as a group-sparse regularization technique [55]. The ℓ_1 part generates a sparse model while the ℓ_2 part encourages a smoothing, grouping effect [30]. Such group-sparsity property is beneficial when transferring a pre-trained CNN to a novel task, since it allows to select the correlated features suitable to the target data while discarding those uncorrelated ones. We thus impose a weighted mixture of ℓ_1 and squared ℓ_2 penalties on the coefficient matrix $\mathbf{H}$ to achieve the desired group-sparse representations. The resulting GSNMF objective function is defined as

$$f(\mathbf{W},\mathbf{H}) = \frac{1}{2} \parallel \mathbf{X} - \mathbf{WH} \parallel_F^2 + \frac{\lambda_1}{2} \parallel \mathbf{H} \parallel_2^2 + \lambda_2 \parallel \mathbf{H} \parallel_1, \quad s.t. \mathbf{W},\mathbf{H} \geq 0. \tag{2}$$

Here λ_1 and λ_2 are the hyper-parameters that control the importance of the ℓ_1 and ℓ_2 regularization terms.

3.2. Optimization

We use the alternating minimization procedure and multiplicative update rule to optimize Eqn. (2) following [26]. Since we impose group-sparsity on the coefficient matrix $\mathbf{H}$, the update rule of $\mathbf{W}$ remains the same as that in the standard NMF formulation [26]. We use gradient descent to optimize $\mathbf{H}$, and the first-order update rule of $\mathbf{H}$ should be generally in the form of

$$\mathbf{H} \leftarrow \mathbf{H} - \eta * \frac{\partial \mathbf{f}(\mathbf{H})}{\partial \mathbf{H}}, \tag{3}$$

where $*$ denotes the element-wise multiplication and the matrix η is the step size. We take the derivative of $f(\mathbf{H})$ in Eqn. (2) with respect to $\mathbf{H}$, leading to

$$\frac{\partial f}{\partial \mathbf{H}} = -\mathbf{W^T X} + \mathbf{W^T WH} + \lambda_1 \mathbf{H} + \lambda_2 \mathbf{I}, \tag{4}$$

where $\mathbf{I}$ is an all-ones matrix of the same size as $\mathbf{H}$. Since the ℓ_1 norm is not differentiable at 0, Eqn. (4) is the subgradient at 0. Following a similar deriving procedure as in [26], we let the adaptive step size η to be

$$\eta = \frac{\mathbf{H}}{\mathbf{W^T WH} + \lambda_1 \mathbf{H} + \lambda_2 \mathbf{I}}, \tag{5}$$

where the division is element-wise division, and we then have the following update rule

$$\begin{cases} \mathbf{W} \leftarrow \mathbf{W} * \frac{\mathbf{XH^T}}{\mathbf{WHH^T}}, \\ \mathbf{H} \leftarrow \mathbf{H} * \frac{\mathbf{W^T X}}{\mathbf{W^T WH} + \lambda_1 \mathbf{H} + \lambda_2 \mathbf{I}}. \end{cases} \tag{6}$$

Here the coefficient $\mathbf{H}$ is the new feature representation. Eqn. (6) is a straightforward modification to the multiplicative update rule in the standard NMF optimization [26]. Following a similar proof to that of [26] which uses an auxiliary function analogous to that used for proving convergence of the Expectation Maximization algorithm [8], we can show that the process converges. Since the update rules are multiplicative, when $\mathbf{W}$ and $\mathbf{H}$ are initialized as non-negative, they will remain non-negative during the optimization.

3.3. Unsupervised Fine-Tuning of the Network

As shown in Figure 2, we use the feature reconstruction (generative) loss in Eqn. (2) for our unsupervised fine-tuning, in contrast to the cross-entropy loss in conventional supervised fine-tuning. In the off-the-shelf (OTS) scenario, we only train the GSNMF module while freezing the pre-trained representation module. In the fine-tuning (FT) scenario, we train the GSNMF module while fine-tuning the representation module. Following the standard NMF practice, in our implementation, we introduce additional ℓ_2 normalization layers to $\mathbf{X}$ and the basis matrix $\mathbf{W}$ before the factorization layer. Regularizing the feature vector norm has been a staple of unsupervised learning approaches to prevent degenerate solutions and collapsed networks [37].

During each iteration, after forward propagation, we obtain the $fc7$ activations (*i.e.*, $\mathbf{X}$) on the mini-batch. The mini-batch size is $N = 256$. We learn $\mathbf{W}$ and $\mathbf{H}$ using the update rule in Eqn. (6) . We then fix $\mathbf{W}$ and $\mathbf{H}$, and the loss in Eqn. (2) reduces to the standard Euclidean loss $\parallel \mathbf{X} - \mathbf{WH} \parallel_F^2$. We back-propagate the Euclidean error to update the parameters in the CNN representation module.

This alternating fine-tuning strategy using generative connections could also be seen broadly relevant to the wake-sleep algorithm [18]. In our evaluation with limited target data, we froze the remaining layers underneath $fc7$ and did not fine-tune them due to over-fitting concerns. With more training data available, additional layers could be further fine-tuned.

Algorithm complexity: The time complexity of our approach is polynomial time $O(NLT)$, where N is the number of samples, L is the feature dimension, and T is the iteration number. In our experiments, a forward-backward pass took less than 0.5 second on a single Titan GPU.

4. Experimental Evaluation

In this section, we evaluate the representation transferability of our factorized convolutional networks (FCNs) on both standard image clustering datasets and multiple much more challenging benchmarks for image clustering, in which no labeled data is provided. We first introduce the datasets and the implementation details, and then present quantitative results by comparing with several state-of-the-art methods and validating across tasks the generality of FCN. In absolute terms, we achieve the best performance on all these benchmarks. We also show that FCN can be used as unsupervised initialization to further improve the performance of classification tasks. Our approach is general as it can be applied to different CNN architectures. Here we focus on VGGNet [41] and evaluate variants of our model:

NMF-FCN: All layers of VGGNet are frozen, and we feed the $fc7$ activations to an NMF module. The coefficient matrix $\mathbf{H}$ is used as the new feature representation.

GSNMF-FCN-OTS: All layers of VGGNet are frozen, and we feed the $fc7$ activations to a GSNMF module, in which the group-sparsity constraints are imposed on the standard NMF layer.

GSNMF-FCN-FT: All layers except the $fc7$ layer are frozen, and we combine VGGNet with a GSNMF module and fine-tune $fc7$ as well. The coefficient matrix $\mathbf{H}$ is used as the new feature representation.

4.1. Datasets

Our model is evaluated on diverse datasets including standard image clustering datasets and large-scale image classification datasets (used for the image clustering tasks):

MNIST [25]: MNIST consists of 28×28 gray scale images of handwritten digits ranging from 0 to 9. The dataset contains $50{,}000$ training samples, $10{,}000$ validation samples, and $10{,}000$ test samples.

COIL-20 [31]: COIL-20 consists of 1440 32×32 gray scale images of 20 objects. The images of each object were taken 5 degree apart.

As there is no standard large-scale image clustering dataset, we evaluate our model on large-scale image classification datasets whose labels are not used during training:

MIT-67 [36]: MIT-67 consists of 15K images spanning 67 indoor scene classes, which makes it a challenging test case for feature representations. The provided train/test split for this dataset includes 80 training and 20 test images per class.

Caltech-UCSD Birds (CUB) 200-2011 [49]: Birds-200 contains $11{,}788$ images of 200 birds species. $5{,}994$ images are used for training and $5{,}794$ for testing.

Oxford 102 Flowers [33]: Flowers-102 contains 102 flower categories, and each class consists of between 40 and 258 images. 10 images are used as training data and the rest are used as test data.

These are very challenging tasks because of the following reasons. (1) There are strong domain shifts between the source and target datasets. Compared to the object-centric ILSVRC dataset where the CNN features are pre-trained, the target MIT-67 dataset is more scene-centric and consists of similar objects presented in different indoor scenes [36], and the target Birds-200 and Flowers-102 datasets involve very subtle differences between examples of a visual category [1]. Importantly, the transferability of a CNN decreases when the target task is far from the CNN source task [1]. (2) The datasets used for evaluation are standard classification benchmarks, and they are still very challenging even for supervised image classification. However, we tackle a more difficult scenario here by testing the representations for unsupervised image clustering, without having access to the label information on these datasets. We will show that with limited amount of unlabeled training data from distinct target tasks, our FCN model is capable of discovering informative and discriminative latent variables from CNN representations.

4.2. Baseline Models

In order to evaluate the performance of our FCN model, we compared it against not only the state-of-the-art algorithm, but also other linear and nonlinear dimension reduction algorithms that could be useful in learning effective feature representations. These baselines include:

CNN: All layers of VGGNet are frozen, and the $fc7$ activations are used as the feature representation with dimension $4{,}096$.

PCA-CNN: We perform PCA over the CNN representation and use the coefficient as the new feature representation. The number of principal components is set as $1{,}024$.

LLE-CNN [9]: Locally linear embedding uses an eigenvector based optimization technique to find the low-dimensional embedding of points, such that each point is still described with the same linear combination of its neighbors. The number of nearest neighbors is set as 12 and the feature dimension is set as $1{,}024$.

Autoencoder-CNN (AE-CNN): After careful preliminary experiments, we choose linear activations as the transition functions of the encoder and decoder. The autoencoder is trained in 200 epochs with a batch size of 128. To avoid over-fitting, we use 10% of training data as validation data.

Non-Negative AutoEncoder-CNN (NNAE-CNN): Due to limited training data, we use linear activations as the transition functions as above. During each iteration, we force the weights of the encoder and decoder to be non-negative.

EP-CNN [5, 6]: Ensemble projection samples from the available training data as an ensemble of image prototype sets and learns discriminative functions over these prototype sets. We follow the same parameter setting in [5, 6], and the feature dimension is 3,000.

4.3. Implementation Details

Our FCN model includes two modules and is implemented in Keras [4]. For the CNN layers, we use the VGGNet pre-trained on ILSVRC where all the layers except $fc7$ are frozen to those learned on ILSVRC without fine-tuning [41]. In our preliminary experiment, we fine-tuned $fc6$ as well. Compared with only fine-tuning $fc7$, the performance dropped due to limited target data in our case. This is consistent with the observation in standard supervised fine-tuning. With more training data, fine-tuning more layers should further improve the performance. For each image, we resize the image to 224×224, and extract a 4,096-dim feature vector from the entire image.

For the GSNMF module, to speed up the convergence rate of NMF, we use the non-negative double singular value decomposition (NNDSVD) [2]. NNDSVD is a method based on two SVD processes: one approximates the initial data matrix, and the other approximates the positive components of the resulting partial SVD factors. We use the unlabeled training data on the target task to learn the bases and coefficients. L is set as 1,024. The test images are then fed forward to the learned FCN model, producing a final 1,024-dim feature representation.

Note that our main purpose is to validate whether the proposed approach is able to boost the transferability of CNN features for image clustering and is not to propose a better clustering approach. Hence, we use two standard clustering methods, which are spectral clustering (SC) [32] and k-means. Choosing the k number of clusters is typically difficult for clustering algorithms without any prior knowledge of the data. We then chose k as the number of classes for each target dataset. In our preliminary experiments, we found that ours consistently outperformed baselines with different values of k, due to our improved feature representations. For a fair comparison, we perform ℓ_2 normalization on the feature representations for both our models and baselines. To reduce the influence of randomness introduced by different initializations of k-means, the k-means grouping stages in SC and k-means are repeated 10 times. The result with the minimum distortion is selected. Euclidean distance is used for both methods.

4.4. Methodology

Hyper-parameter settings: For the regularization parameters λ_1 and λ_2 in GSNMF, in a preliminary experiment, we tested image clustering on the Scene-15 dataset [11], which is a relatively small dataset. After searching λ_1 and λ_2 on a 2D grid $10^{[-4:1:1]} \times 10^{[-4:1:1]}$, we observed that the best performance was achieved when $\lambda_1 = 0.02$ and $\lambda_2 = 0.05$. In all our experiments, we then simply set λ_1 as 0.02 and λ_2 as 0.05.

Choosing the optimal representation dimension L remains challenging in dimensionality reduction. Similarly, in our preliminary experiment, we tested image clustering on Scene-15 and MIT-67, and found $L = 1{,}024$ usually achieved the best performance. In all our experiments, we then simply set $L = 1{,}024$. Even better performance could be obtained by further tuning these hyper-parameters. We also conduct hyper-parameter sensitivity analysis to test how λ_1, λ_2 and L affect the clustering accuracy.

Evaluation metrics: Consistent with the previous work, accuracy [51] and normalized mutual information (NMI) [3] are used as the evaluation criterion. We assume that the clustering algorithm is tested on *N* samples. For a sample x_i, the cluster label is denoted as r_i, and its ground truth label is t_i. Accuracy is defined as

$$\text{accuracy} = \frac{\sum_{i=1}^{N} \delta(t_i, \text{map}(r_i))}{N}, \tag{7}$$

where $\delta(x, y)$ equals to 1 if x is equal to y, and 0 otherwise. The function $\text{map}(x)$ is the best permutation mapping function, which maps a cluster to its corresponding predicted label. Hence, a higher accuracy indicates that more samples are predicted correctly.

Now let C denote the cluster centers of the ground truth, and $C^{'}$ denote the cluster centers predicted by the clustering algorithm. NMI is then defined as

$$\text{NMI}(C, C^{'}) = \frac{\text{MI}(C, C^{'})}{\max(H(C), H(C^{'}))}, \tag{8}$$

where $H(C)$ and $H(C^{'})$ are the entropies of C and $C^{'}$, respectively. $\text{MI}(C, C^{'})$ is the mutual information of C and $C^{'}$. NMI measures the dependency of two distributions. A higher NMI means that two distributions are more similar.

5. Results and Discussion

In our FCN model, we aim to learn better high-level representations by introducing the GSNMF module on top of the original CNN representation. Furthermore, by fine-tuning the CNN representation module (the last fully-connected layer in our case), the additional degree of freedom allows the FCN model to represent the target data more

	Method	k-means				spectral clustering			
		MNIST		COIL-20		MNIST		COIL-20	
		Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI
Baselines	CNN	46.7	38.4	74.0	89.2	51.1	41.3	84.3	92.0
	PCA-CNN	55.0	47.2	76.9	89.9	48.4	39.6	84.4	91.9
	LLE-CNN [9]	29.0	23.0	34.2	34.5	28.9	17.4	22.6	24.7
	AE-CNN	45.9	37.3	81.7	90.4	54.9	44.6	81.7	92.1
	NNAE-CNN	46.0	37.4	77.1	90.0	49.2	40.6	85.7	92.3
	EP-CNN [5, 6]	60.0	58.0	80.5	92.2	53.2	47.6	86.7	93.8
Ours	NMF-FCN	61.7	57.8	82.3	93.0	56.4	47.6	88.6	93.9
	GSNMF-FCN-OTS	62.2	58.0	**84.3**	**94.4**	57.6	47.9	**89.2**	**94.0**
	GSNMF-FCN-FT	**63.3**	**58.6**	79.6	89.4	**58.4**	**48.0**	86.5	92.7

Table 1: Accuracy (%) and normalized mutual information (NMI) (%) of image clustering on two standard benchmark datasets. k-means and spectral clustering are used on top of the feature representations of our FCN model and baseline models. The best results are in bold.

effectively. To validate this, we evaluate our model on standard image clustering datasets and large-scale benchmarks.

Evaluation on standard image clustering datasets: Table 1 summarizes the comparison between our model and the baseline models. Our FCN model outperforms the original CNN representation by a large margin on these standard image clustering datasets. For example, in terms of accuracy, our FCN model outperforms the original CNN representation by 16.6% on MNIST and 10.3% on COIL-20.

Evaluation on large-scale benchmark datasets: Table 2 summarizes the *k*-means and spectral clustering (SC) performance of our FCN representation and related baseline features on the scene and fine-grained recognition datasets. We can see that our FCN representation outperforms the original CNN feature and other representations by a considerable margin using the same clustering method. For instance, in terms of clustering accuracy, FCN outperforms CNN by 3.2% on MIT-67, 2.6% on Birds-200 (where chance is 0.5%), and 5.3% on Flowers-102 (where chance is 1%) when k-means is used.

Given that there are only 10 images available for each class on Flowers-102 to fine-tune our network, GSNMF-FCN-FT performs slightly worse than GSNMF-FCN-OTS, while GSNMF-FCN-OTS significantly outperforms CNN. Consistent with the standard supervised fine-tuning, for target tasks with medium sized data, GSNMF-FCN-FT consistently outperforms GSNMF-FCN-OTS (*e.g.*, by 1% on Birds-200 with 200 classes and $5{,}994$ images). With more data, GSNMF-FCN-FT will further improve the performance.

Moreover, EP-CNN reported improved performance over CNN in transductive learning, where the EP representation (ensemble of classifiers) was learned using both the training and test datasets [5]; however, in our case of learning representation on the training dataset and conducting clustering on the test dataset, EP-CNN shows inferior performance to CNN. This means that having access to the distribution of the test data is advantageous for EP-CNN.

The superior performance of our GSNMF-FCN reveals that it learns a more generic and transferable representation to capture the subtlety of differences across different categories and tasks. In particular, these results show that our approach, pre-trained on ILSVRC, is effective on a broad range of target domains, ranging from low source-target distance (*e.g.*, MIT-67), to medium distance (*e.g.*, Birds-200, Flowers-102), and to large distance (*e.g.*, MNIST) [1].

Hyper-parameter sensitivity analysis: We now examine the influence of the hyper-parameters of our model on its clustering performance. They are the regularization coefficients λ_1, λ_2 and the feature dimension L. We first evaluate L on MIT-67, and Table 3 shows that FCN consistently outperforms PCA-CNN and AE-CNN in different settings of L. We then evaluate λ_1, λ_2 on Flowers-102. Each time we change the value of one hyper-parameter with the others fixed to the values described in the experimental settings.

Figure 3 summarizes the hyper-parameter sensitivity analysis. The performance of our model increases with λ_2 at first and then stabilizes quickly given a certain λ_1. It validates that our model benefits from imposing the regularization terms. After λ_2 increases above some threshold (*e.g.*, 0.05), the accuracy and NMI become stable; as λ_2 increases further, the performance drops accordingly, implying that a larger ℓ_2 regularization coefficient will hurt the performance. Similar trend is observed for the ℓ_1 regularization (*e.g.*, for a fixed $\lambda_2 = 0.05$, our model achieves best when λ_1 lies in the range from 0.02 to 0.05).

Evaluation of group-sparsity formulations: Our GSNMF-FCN uses the joint ℓ_1 and ℓ_2 norm penalty to impose group-sparsity. The mixed $\ell_{2,1}$ norm penalty is an alternative [24]. In our preliminary experiment, we compared the two formulations, and found that introducing group sparsity helped and the joint ℓ_1 and ℓ_2 norm worked better, as shown in Table 4.

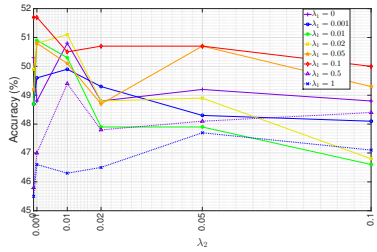
Unsupervised fine-tuning as initialization for image clas-

	Method	k-means						spectral clustering					
		MIT-67		Birds-200		Flowers-102		MIT-67		Birds-200		Flowers-102	
		Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI
Baselines	CNN	45.0	63.2	32.5	61.6	45.7	63.7	37.3	57.6	26.5	57.0	44.7	63.6
	PCA-CNN	45.9	63.8	32.1	62.0	46.3	63.9	37.3	57.5	29.1	58.7	46.0	63.0
	LLE-CNN [9]	17.4	32.8	21.3	49.4	26.6	45.9	20.7	42.3	17.3	33.4	18.0	35.3
	AE-CNN	46.8	64.3	32.6	62.2	46.8	64.5	35.4	57.1	28.0	58.0	43.8	62.1
	NNAE-CNN	43.0	63.0	32.4	61.8	45.6	64.5	35.3	58.8	29.0	58.7	**46.6**	63.5
	EP-CNN [5, 6]	47.2	64.6	31.0	61.1	43.9	60.0	43.7	61.9	29.0	59.6	43.0	61.3
Ours	NMF-FCN	46.8	64.0	34.1	62.2	50.2	65.7	42.8	61.4	28.7	58.9	45.3	**64.7**
	GSNMF-FCN-OTS	47.7	64.5	34.4	62.1	**51.0**	**65.9**	43.2	62.3	31.0	59.8	45.7	63.9
	GSNMF-FCN-FT	**48.2**	**64.9**	**35.1**	**62.5**	50.1	64.9	**44.2**	**62.5**	**32.2**	**60.0**	44.7	62.0

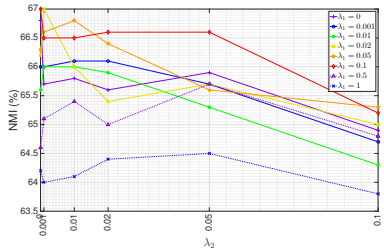
Table 2: Accuracy (%) and normalized mutual information (NMI) (%) of scene and fine-grained image clustering on three large-scale benchmark datasets. k-means and spectral clustering are used on top of the feature representations of our FCN model and baseline models. The best results are in bold.

	Method		
Dimension L	PCA-CNN	AE-CNN	GSNMF-FCN (Ours)
256	40.3	42.7	**43.2**
512	42.8	43.9	**45.1**
1024	45.9	46.8	**48.2**
2048	43.6	45.2	**46.0**

Table 3: Hyper-parameter sensitivity analysis on MIT-67: Accuracy (%) comparison between our FCN and PCA-CNN, AE-CNN as functions of the feature dimension L.



(a) Accuracy



(b) NMI

Figure 3: Hyper-parameter sensitivity analysis on Flowers-102: Accuracy (%) and NMI (%) of our FCN as functions of its regularization coefficients λ_1 and λ_2.

sification: Our FCN can be used to improve the classification performance as well. We evaluate this point by first learning GSNMF embedding on the target training data without using the labels and then training SVM classifiers on top of the learned embedding using the training labels. Table 5 shows that our approach outperforms SVM directly trained with the original CNN feature representation.

Method	MNITS	Birds-200
mixed $\ell_{2,1}$	56.5	29.1
joint ℓ_1 and ℓ_2 (Ours)	**57.6**	**31.0**

Table 4: Performance comparison of clustering accuracy (%) between different group-sparsity formulations. The joint ℓ_1 and ℓ_2 norm penalty outperforms the mixed $\ell_{2,1}$ norm.

Method	MIT-67	Birds-200	Flowers-102
CNN	70.78	68.51	82.44
GSNMF-FCN (Ours)	**72.75**	**70.89**	**84.70**

Table 5: Performance comparison of classification accuracy (%) between GSNMF-FCN and CNN. Learning SVM classifiers on top of the unsupervised GSNMF-FCN embedding outperforms training SVMs in the original CNN space.

6. Conclusion

In this paper, we showed how to improve the transferability of a deep CNN representation for other visual recognition tasks with *unlabeled* training data, where conventional fine-tuning with back-propagation is inapplicable. By introducing group-sparse non-negative matrix factorization (GSNMF) on top of CNN activations to constitute a unified feed-forward factorized convolutional network (FCN), we discovered a rich set of informative and discriminative latent variables. Extensive large-scale image clustering (and classification) experiments demonstrate that the new feature representations are significantly suitable for scene and fine-grained recognition tasks.

References

[1] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 36–45, Boston, MA, USA, June 2015.

[2] C. Boutsidis and E. Gallopoulos. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362, April 2008.

[3] D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, August 2011.

[4] F. Chollet et al. Keras, 2015.

[5] D. Dai, M. Prasad, C. Leistner, and L. Van Gool. Ensemble partitioning for unsupervised image categorization. In *European Conference on Computer Vision (ECCV)*, pages 483–496. Firenze, Italy, October 2012.

[6] D. Dai and L. Van Gool. Unsupervised high-level feature learning by ensemble projection for semi-supervised image classification and image clustering. *arXiv preprint arXiv:1602.00955*, 2016.

[7] D. Dai, T. Wu, and S.-C. Zhu. Discovering scene categories by information projection and cluster sampling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3455–3462, San Francisco, CA, USA, June 2010.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, 39(1):1–38, 1977.

[9] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *National Academy Sciences*, 100(10):5591–5596, May 2003.

[10] M. Dorfer, R. Kelz, and G. Widmer. Deep linear discriminant analysis. In *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, USA, May 2016.

[11] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 524–531, San Diego, CA, USA, June 2005.

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–264–II–271, Madison, WI, USA, June 2003.

[13] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of International Conference on International Conference on Machine Learning (ICML)*, volume July, pages 1180–1189, Lille, France, 2015.

[14] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision (ECCV)*, pages 597–613, Amsterdam, the Netherlands, October 2016.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, Columbus, OH, USA, June 2014.

[16] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. Measuring invariances in deep networks. In *Conference on Neural Information Processing Systems (NIPS)*, pages 646–654, Vancouver, B.C., Canada, December 2009.

[17] L. Gui and L.-P. Morency. Learning and transferring deep Convnet representations with group-sparse factorization. In *International Conference on Computer Vision Workshops (ICCVW)*, 2015.

[18] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158, May 1995.

[19] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.

[20] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[21] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[22] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 46. John Wiley & Sons, April 2004.

[23] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[24] J. Kim, R. D. Monteiro, and H. Park. Group sparsity in nonnegative matrix factorization. In *Proceedings of SIAM International Conference on Data Mining (ICDM)*, pages 851–862, Anaheim, CA, USA, April 2012.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[26] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, August 1999.

[27] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Conference on Neural Information Processing Systems (NIPS)*, pages 801–808, Vancouver, B.C., Canada, December 2007.

[28] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of International Conference on International Conference on Machine Learning (ICML)*, pages 609–616, Montreal, QC, Canada, June 2009.

[29] Z. Li and D. Hoiem. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*, volume PP, pages 1–13, November 2016.

[30] W. Liu, S. Zheng, S. Jia, L. Shen, and X. Fu. Sparse nonnegative matrix factorization with the elastic net. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 265–268, Hong Kong, China, December 2010.

[31] S. A. Nene, S. K. Nayar, H. Murase, et al. Columbia object image library (COIL-20). Technical report, technical report CUCS-005-96, 1996.

[32] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Conference on Neural Information Processing Systems (NIPS)*, pages 849–856, December 2002.

[33] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics Image Processing (ICCVGIP)*, pages 722–729, Bhubaneswar, India, December.

[34] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1717–1724, Columbus, OH, USA, June 2014.

[35] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10):1345–1359, 2010.

[36] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 413–420, Miami, FL, USA, June 2009.

[37] M. Ranzato. *Unsupervised learning of feature hierarchies*. PhD thesis, 2009.

[38] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 806–813, Columbus, OH, USA, June 2014.

[39] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, December 2015.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[42] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 370–377, Beijing, China, October 2005.

[43] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller. A deep semi-nmf model for learning hidden representations. In *Proceedings of International Conference on International Conference on Machine Learning (ICML)*, pages 1692–1700, Beijing, China, June 2014.

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

[45] Y.-X. Wang and M. Hebert. Learning from small sample sets by combining unsupervised meta-training with CNNs. In *Conference on Neural Information Processing Systems (NIPS)*, 2016.

[46] Y.-X. Wang, D. Ramanan, and M. Hebert. Growing a brain: Fine-tuning by increasing model capacity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017.

[47] Y.-X. Wang and Y.-J. Zhang. Nonnegative matrix factorization: A comprehensive review. *TKDE*, 25(6):1336–1353, 2013.

[48] Z. Wang, S. Chang, J. Zhou, M. Wang, and T. S. Huang. Learning a task-specific deep architecture for clustering. pages 369–377, May.

[49] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[50] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of International Conference on International Conference on Machine Learning (ICML)*, pages 478–487, Lille, France, July 2015.

[51] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of International ACM SIGIR Conference on Research and Development in Informaion Retrieval (ACM SIGIR)*, pages 267–273, Toronto, Canada, July-August 2003.

[52] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5147–5156, Las Vegas, NV, USA, June-July 2016.

[53] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Conference on Neural Information Processing Systems (NIPS)*, pages 3320–3328, Montreal, Canada, December 2014.

[54] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833, Zurich, September 2014.

[55] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, March 2005.