22^{nd} *Computer Vision Winter Workshop*
Nicole M. Artner, Ines Janusch, Walter G. Kropatsch (eds.)
Retz, Austria, February 6–8, 2017

Unsupervised Learning for Image Category Detection

Philipp Seeböck[1,2], René Donner[1], Thomas Schlegl[1,2], Georg Langs[1]
[1]Computational Imaging Research Lab, Medical University of Vienna
[2]Christian Doppler Laboratory for Ophthalmic Image Analysis, Medical University of Vienna
{philipp.seeboeck, rene.donner, thomas.schlegl, georg.langs}@meduniwien.ac.at

Abstract. *We propose an unsupervised object category learning approach, where the output representations improve classification performance too. The contribution is threefold: we integrate a 'Network in Network' (NIN) approach in unsupervised learning, improving the representational power of the network for local patches by replacing linear filters with micro convolutional networks. We learn receptive fields to connect layers of the network sparsely, and we propose a new encoding function that introduces sparsity in a natural way and avoids the necessity of parameter tuning. The learned model generates a feature representation of images, used for unsupervised category learning. Results demonstrate that the obtained image categories reflect true object categories well. In addition, experimental results on classification tasks show superiority of the proposed approach in comparison with an unsupervised state-of-the-art learning architecture.*

1. Introduction

The unsupervised identification of categories is a crucial aspect of machine learning when ground truth is either unknown or costly to obtain. In medical imaging both issues arise frequently, and the discovery of potential markers occurring across large image sets is of central interest. In this paper we present an unsupervised approach to learn a model for category detection. Images are mapped to a new feature representation using an unsupervisedly learned Convolutional Neural Network (CNN), and are subsequently clustered in this representation space. We demonstrate the capability of the learned model identifying meaningful categories on natural imaging data in a completely unsupervised way, and show that it improves performance in comparison with a state-of-the-art learning architecture. The key contribution of this paper is to understand if the NIN approach improves an unsupervised learning architecture in comparison with conventional unsupervised feature learning approaches. We use the term 'category detection' to stress the difference between our approach and clustering methods. In contrast to the classical approach of feeding hand-crafted features [17, 25, 23] to a clustering or classification [33, 2] algorithm, we aim at learning high level features from the training data which are both highly discriminative to the unknown categorical classes, and robust to low level visual variation.

Related work The idea of learning categories in a set of unlabeled or weakly labeled data is not new. Several approaches tackling the problem of object detection have been proposed [15, 13, 32, 28, 3, 11]. While Grauman and Darrell group images based on a dissimilarity measure between sets of unordered features [15], Faktor and Irani cluster images based on the idea that images which can be composed from each other are similar [13]. These methods can be seen as non-deep learning approaches for learning image categories without using any labels. In contrast, Wu *et al.* [32] use a weakly supervised multiple instance learning approach to train a deep CNN for image classification and annotation. Weakly supervised learning is also used by Oquab *et al.* [28] to train a deep CNN to detect and localize objects in images. In that work, labels are given at image-level, but no object location annotations are used for training. Schlegl et al. use weakly supervised learning to link image information to semantic descriptions of image content [29]. Instead of assigning labels to images, Cho *et al.* [3] find object locations based on hand-crafted features (HOG [10]) in an unsupervised way. Doersch *et al.* [11] use spatial context as supervisory signal to train a CNN, where predicting the relative position of a randomly extracted pair of

patches is used as training objective. As opposed to the work of Doersch *et al.* [11], we want to tackle the problem of identifying categories in cases where no informative context is available. Other works address the challenge of unsupervised learning beyond unsupervised object detection in a more general way [7, 5, 6, 4, 12]. Coates *et al.* [7] compare various unsupervised learning methods in a single-layer network and show that the best performance is achieved with K-means clustering as feature learning algorithm. Dosovitskiy *et al.* [12] train a network unsupervisedly with surrogate classes. These are created by applying hand-crafted transformations, which are assumed not to change the identity of the image content. In contrast, our approach does not use hand-engineered prior knowledge about the data.

An important issue in building deep neural networks is how to choose the connections between layers (i.e. how to connect the features of the actual layer with the features of the lower layer), as pointed out and investigated in [9]. In line with terminology of previous literature, we denote the grouping of filters in order to achieve sparse connections between layers as receptive field learning. Coates and Ng introduced an algorithm to learn receptive fields automatically and reduce the feature dimension for simple unsupervised training algorithms in higher layers [5]. Apart from learning connections between layers, there have also been attempts to learn spatial-pooling regions using labels [20].

Besides the choice of connections between layers, the basic network architecture plays an important role, too. Lin *et al.* [24] introduced the idea of replacing convolution filters with a "micro network", called NIN approach. The basic idea is to increase the representational power of neural networks, respectively of individual parts, by using a "micro network" as part of the overall network. This concept has been picked up and taken one step further by Szegedy *et al.* [31], who describe the "Inception Module" in their work. The "Inception Module" consists of "micro networks", while the module itself is also a part of a final, even bigger, network. Since both papers apply the concept successfully to a purely supervised learning problem, a transfer to unsupervised learning seems promising, and is evaluated in this paper.

Contribution In this paper, we use recent insights [7, 4, 24, 31] to develop a novel unsupervised architecture. Specifically, we (1) specify an algorithm that groups similar filters in order to create receptive fields, (2) introduce the basic idea of the NIN approach in unsupervised learning and (3) propose the new *mean-sparse* encoding function. First, we use these ingredients to learn image categories from the dataset (CIFAR-10, STL-10) in a purely unsupervised way. Second, we evaluate the learned feature representation on a standard supervised classification task in direct comparison with unsupervised state-of-the-art approaches [7, 5].

Receptive field learning: High input dimensionality can have a negative impact on the success of unsupervised learning, specifically on the learned features of K-means clustering [6]. Furthermore, results in [5] indicate that grouping of similar features is essential in order to enable unsupervised learning algorithms to detect useful higher-layer features. The grouping of similar features is also motivated by the Hebbian principle, 'neurons that fire together, wire together' [26]. In our experiments, the proposed receptive field learning algorithm yields better classification accuracy than the one introduced in [5].

NIN architecture: We transfer the core idea of NIN [24, 31] to unsupervised learning and replace simple filters with *"micro-conv-nets"* to improve the representational power of individual network components. In the unsupervised category detection experiment we used three external evaluation criteria to determine the performance. The proposed approach outperforms the state-of-the-art single layer architecture (Section 4.2).

Mean-sparse encoding: On the one hand, *mean-sparse* encoding is motivated by the performance of the *triangle* function, described in [7]. That encoding scheme also yields sparse outputs by subtracting a mean value across features. On the other hand, we want to avoid tuning parameters where possible. In contrast to the widely used soft-threshold encoding, the *mean-sparse* function selects the threshold value automatically. *Mean-sparse* encoding achieves comparable results on the category-learning task on both datasets as well as on the classification task on the STL-10 dataset.

2. Method

Our method consists of a CNN model that maps the input images to a new representation and a subsequent clustering or classification stage. We present an approach to train the CNN in a completely unsupervised way. Then we describe the subsequent clustering, respectively classification stage.

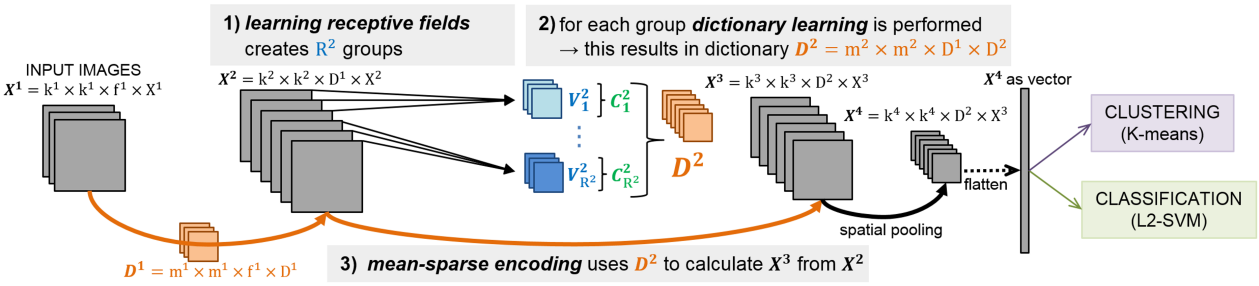

Figure 1. Illustration of the proposed architecture, revealing the interaction of the techniques described in Section 2. In general, the input images are mapped to a new feature representation $\mathbf{X}^4$ by applying two convolutional and one pooling layer, where this representation then can be used for category detection or classification tasks. The procedure to learn the dictionary is depicted for layer 2: (1) The receptive field learning algorithm is used to create R^2 receptive fields, followed by (2) learning sub-dictionaries $\mathbf{C}^2$ from the training sets $\mathbf{V}^2$ and concatenating them in $\mathbf{D}^2$. (3) $\mathbf{X}^3$ is computed by using $\mathbf{D}^2$ and mean-sparse encoding. The transition from $\mathbf{X}^1$ to $\mathbf{X}^2$ and from $\mathbf{X}^2$ to $\mathbf{X}^3$ conceptually forms an NIN approach, illustrated in Figure 2.

We work with image patches forming the input of the first layer, denoted as $\mathbf{X}^1$ of size $k^1 \times k^1 \times f^1 \times X^1$, where X^1 is the number of patches, each of size $k^1 \times k^1$ with f^1 channels (feature maps), and 1 being the layer index. Considering one single 32-by-32 RGB input image, it is then described as $32 \times 32 \times 3 \times 1$.

For clarity, we represent convolution and pooling as separate layers. Considering a convolutional layer l, the output of the unsupervised feature learning stage is a "dictionary" $\mathbf{D}^l$ of size $m^l \times m^l \times f^l \times D^l$, where D^l is the number of filters, each of size $m^l \times m^l$ with f^l feature maps. In the following we describe the procedure to learn this dictionary (illustrated in Figure 1):

1. Create R^l receptive fields (Section 2.1), containing S^l feature maps each, where $S^l \leq f^l$. R^l denotes the number of receptive fields, S^l the number of feature maps in one receptive field and f^l the total number of feature maps in layer l.
2. For each receptive field unsupervised learning is performed separately:
   (a) Extract V^l random patches within the receptive field to create a training set $\mathbf{V}^l$ of size $m^l \times m^l \times S^l \times V^l$, where $m^l * m^l * S^l$ denotes the feature dimensionality.
   (b) Apply the unsupervised learning algorithm (Section 2.2) to $\mathbf{V}^l$ in order to generate a so called sub-dictionary $\mathbf{C}^l$ of size $m^l \times m^l \times S^l \times C^l$, with C^l denoting the number of learned filters.
3. We summarize all R^l sub-dictionaries (one for each receptive field) in one final dictionary: Each $\mathbf{C}^l$ contains filters which are only connected to the feature maps contained in the corresponding receptive field. We "blow up" the filters $\mathbf{C}^l$ from size $m^l \times m^l \times S^l$ to $m^l \times m^l \times f^l$ by inserting zero values at all other positions. In terms of feature extraction, this makes no difference to applying the filters $\mathbf{C}^l$ to the corresponding receptive field. This means care has to be taken that the spatial information is preserved by linking the non-zero values to the correct feature maps.
4. All "blown up" filters build the dictionary $\mathbf{D}^l$.

If two layers are fully connected, $R^l = 1$ and $S^l = f^l$. Note that $D^l = R^l \cdot C^l$ must hold. Based on the dictionary $\mathbf{D}^l$ and a given input $\mathbf{X}^l$, we then calculate the output $\mathbf{X}^{l+1}$ by using a specific encoding, where the concrete computation steps are described in Section 2.4.

Regarding the pooling layer, no explicit learning stage is applied. We use simple average pooling to aggregate features within feature maps. Average pooling partitions a feature map by shifting a rectangle of size $p^l \times p^l$ with a stride w^l over the map and outputs the average value for each region.

2.1. Learning Receptive Fields

As dissimilarity $g_{i,j}$ between two filters $\mathbf{d}_i$ and $\mathbf{d}_j$ we use a metric defined in Coates *et al.* [4][1]:

$$g_{i,j} = \|\mathbf{d}_i - \mathbf{d}_j\|_2 = \sqrt{2 - 2\mathbf{d}_i^\top \mathbf{d}_j} \qquad (1)$$

[1]In contrast to our work, Coates *et al.* [4] use this metric to do max-pooling over similar units. Additionally, the algorithm that groups similiar filters differs from ours.

This dissimilarity is used to implement the concept of grouping together related features in the same receptive field, hence their relationship can be learned more finely by the unsupervised learning algorithm of the subsequent layer. It is important to note that the filters are all normalized to unit length.

To construct a single receptive field in layer l, we randomly select one filter from $\mathbf{D}^{l-1}$ as seed. We then compute the distance to all other filters of this dictionary and add the S^l most similar filters to the actual group. A filter can only be used once as seed. In this way we get R^l equally sized, potentially overlapping, receptive fields that determine the connections between layer l and $l+1$, as illustrated in Figure 1. Both R^l and S^l are hyper-parameters.

2.2. Learning a Dictionary

For each sample in the training set $\mathbf{V}^l$ we subtract the mean of intensities, divide by the standard deviation and apply ZCA-whitening. Then we use spherical K-means according to Coates *et al.* [6] as unsupervised learning algorithm that produces filters similar to sparse autoencoders or sparse RBMs [7]. C^l centroids $\mathbf{c}^l$ are randomly initialized from a normal distribution and normalized to unit length. Damped updates are used to compute new centroids in every iteration in oder to minimize the following objective:

$$\underset{s,\mathbf{c}^l}{minimize} \sum_i \|\mathbf{c}^l s_i - \mathbf{v}_i^l\|_2^2 \qquad (2)$$

where s_i is a vector with only one non-zero entry, for the closest centroid to sample $\mathbf{v}_i^l$. This results in a dictionary (Figure 1). A detailed explanation of the algorithm can be found in [6].

2.3. Unsupervised NIN Approach

We increase the representational power of individual parts in the neural network using a "micro network" as part of the overall network. Figure 2 illustrates the increased representational power of the proposed NIN approach. For the purpose of clarity, we explain this approach using a $6 \times 6 \times 1$ patch as input and chose concrete parameters in accordance with the network architecture explained in Section 3.1.

A conventional convolution is shown in Figure 2 (a), where one filter with $m^1 = 6$ is convolved with the input patch to produce a single activation value. Following the spirit of supervised NIN learning [24], we replace the single convolutional

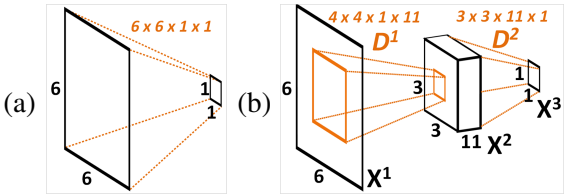

Figure 2. Illustration of two example architectures, where the filters (orange) are applied to the patch-representations (black). (a) Conventional convolution with 6-by-6 filters *"6conv"*, and (b) the proposed architecture with two convolutional layers ($m^1 = 4$ and $m^2 = 3$) *"4/3conv"*.

layer with two convolutional layers, where $m^1 = 4$, $w^1 = 1$, $D^1 = 11$, $m^2 = 3$ and $D^2 = 1$. As illustrated in Figure 2 (b), this leads to a deeper architecture and therefore to an increased abstraction capability, and can be seen as applying a "micro-conv-net" instead of a single filter.

2.4. Mean-sparse Encoding

The *mean-sparse* function is defined as follows:

$$\hat{\mathbf{x}}^{l+1} = max(0, \mathbf{D}^l * \mathbf{x}^l) \qquad (3)$$

$$\mathbf{x}^{l+1} = max(0, \hat{\mathbf{x}}^{l+1} - \mu(\hat{\mathbf{x}}^{l+1})) \qquad (4)$$

where $*$ is the operator denoting spatial convolution of all filters in $\mathbf{D}^l$ with sample $\mathbf{x}^l$, $\hat{\mathbf{x}}^{l+1}$ is an intermediate representation with D^l feature maps and $\mathbf{x}^{l+1}$ is the output representation of the sample. First, the rectified linear function is applied (Eqn. 3). Then the mean activation at each position across all feature maps $\mu(\hat{\mathbf{x}}^{l+1})$ is calculated. This leads to a "mean feature map" of size k^{l+1}-by-k^{l+1}. This map is then subtracted from each feature map in the intermediate representation $\hat{x}^{l+1}$ (Eqn. 4). With *mean-sparse* encoding, we select the threshold value automatically for each spatial position in the feature map and at the same time introduce sparsity in the activations. This can be seen as a simple form of competition between features. The use of the rectified linear function is necessary, since the negative and positive values would cancel each other out otherwise.

In contrast to previous work [5, 7], we have decided not to preprocess the "input sub-patches" at the stage of convolution during feature extraction. Though this may cause a slight loss in performance, it enables the possibility to use the fast convolution modules of Torch7 [8] for our experiments. This speed advantage is a necessity in practice when working with large-scale image data.

In our experiments, we compare the *mean-sparse* encoding function in the convolutional layers with

the widely used *soft-threshold* function:

$$\mathbf{x}^{l+1} = max(0, \mathbf{D}^l * \mathbf{x}^l - \alpha) \quad (5)$$

where α is a tunable constant.

2.5. Category Learning

Categories are identified by mapping each image $\mathbf{x}^1$ using the unsupervisedly learned CNN to a new feature representation $\mathbf{x}^L$ (in Figure 1 denoted as $\mathbf{x}^4$):

$$\mathbf{x}^L = \text{CNN}(\mathbf{x}^1) \quad (6)$$

and subsequently performing Spherical K-means clustering[2] [18] with cosine distance. This leads to cluster centers t_j, where each cluster represents a separate category. To categorize unseen images, they are mapped to $\mathbf{x}^L$ and subsequently assigned to the nearest centroid $t = \text{minInd}(\mathbf{x}^L, t_j)$. $\text{minInd}(\cdot)$ returns the index of the centroid t_j with the minimum cosine dissimilarity to sample $\mathbf{x}^L$, and t therefore represents the label of the assigned category.

2.6. Classification

The feature representation $\mathbf{x}^L$ can also be used to train a classifier (shown in Figure 1) if labels are available. Here, we train a linear Support Vector Machine (L2-SVM) in order to evaluate the applicability of the feature representation for classification.

3. Experimental Setup

Data We conducted experiments on the CIFAR-10 and the STL-10 datasets. The CIFAR-10 dataset comprises 50,000 training and 10,000 test images [21]. The 32×32 RGB images can be divided into 10 different categories, where each class consists of 5,000 training and 1,000 test samples. No preprocessing is applied to the images. The test set is only used for evaluation purposes and is not involved in any training procedure.

The STL-10 dataset has also 10 classes, but comprises only 100 labeled images per class for each training fold (10 pre-defined folds), 800 test images per class and 100,000 additional unlabeled images [7]. The unlabeled images are from a similar but broader distribution of images and are used for the unsupervised training of the CNN architecture and the clustering stage. We down-sample the $96 \times 96 \times 3$ images to $32 \times 32 \times 3$, which enables us to use the same architecture for both CIFAR-10 and STL-10.

[2]The centroids are updated according to the procedure described in [6], where all examples are normalized to unit length in advance.

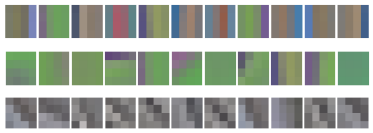

Figure 3. Examples of learned receptive fields in *4/3conv* model on CIFAR-10. Each row shows one receptive field with all 11 filter members of the first layer that are similar in terms of (a) orientation, (b) color or (c) both.

3.1. Compared Network Architectures

Our experiments are based on two network architectures, the proposed method, and a state-of-the-art reference architecture that follows [7]. This allows us both to compare receptive field learning algorithms [5], and the underlying network architectures. Following the experiments reported in [7, 5], we use the approach that achieves the best reported performance among all unsupervised single-layer convolutional networks both on the CIFAR-10 and the STL-10 dataset as reference method. For all convolutional layers in the experiments, we use stride $w = 1$.

The **reference architecture** (*6conv*) is composed of a convolutional layer, which is fully connected to the input layer, and a subsequent average pooling layer. Regarding the experiments of Coates *et al.* [7], the convolutional layer uses a filter size of $6 \times 6 \times 3$, which leads to a $27 \times 27 \times D^1$ representation after the first layer. Then average pooling with $p^2 = 14$ and $w^2 = 13$ is applied to get the final feature representation $2 \times 2 \times D^1$ of each image.

The **proposed architecture** (*4/3conv*) replaces the convolutional layer with "micro-conv-nets", as described in Section 2.3, while the average pooling layer remains the same. For better understanding, this architecture can also be described as consisting of two convolutional layers with $m^1 = 4$ and $m^2 = 3$. While the first convolutional layer is fully connected to the input layer ($R^1 = 1$ and $S^1 = f^1$), the connections between the first and second convolutional layer are sparse, since we chose $R^2 = 32$ and $S^2 = 11$ if $D^1 \leq 400$, and $S^2 = 22$ if $D^1 > 400$.

The dictionary of the first layer $\mathbf{D}^1$ exhibits a size of $4 \times 4 \times 3 \times D^1$, and the sub-dictionaries $\mathbf{C}^2$ a size of $3 \times 3 \times S^2 \times \frac{D^2}{32}$, where $D^1 = D^2$. For each receptive field we learn $\frac{D^2}{32}$ filters, which leads to D^2 feature maps in $\mathbf{D}^2$. Therefore, each receptive field in the second layer has a feature dimension of $99 = 3 \cdot 3 \cdot 11$ if $S^2 = 11$ and $198 = 3 \cdot 3 \cdot 22$ if $S^2 = 22$, which serves as input for the unsupervised feature learning algorithm in the second layer.

Considering the case where $D^1 = D^2 = 800$, we learn 800 different *"micro-conv-nets"* instead of learning 800 different simple filters. Since we learn 32 receptive fields, in each field 25 *"micro-conv-nets"* share the same first layer and differ only in the second layer. This enables the possibility to learn various relationships of the first layer filters in the second layer.

3.2. Evaluating Category Learning

We evaluate if the unsupervised learning captures meaningful categories, by comparing learned categories with ground truth classes.

In this experiment we compare the *4/3conv* with a *6conv* model (Section 3.1). While the reference architecture (*6conv*) uses only *soft-threshold* encoding, we apply both *soft-threshold* and *mean-sparse* encoding in the proposed architecture (*4/3conv*). For all experiments, we do hyperparameter tuning of the *soft-threshold* function with the following values: $\alpha = \{0.1, 0.2, 0.25, 0.3, 0.4, 0.5\}$.

Regarding CNN parameters, we only varied the number of learned filters (100, 200, 400, 800). Both feature-wise and no normalization of $\mathbf{x}^L$ is evaluated in the experiments. For Spherical K-means clustering, we used 10 clusters in all experiments and varied the initialization of the centroids (from a normal distribution or random examples as seed) as well as the re-initialization procedure in case of empty clusters (re-initialization from a normal distribution or with random examples). To ensure a fair comparison, the parameters were the same for both architectures.

For the purpose of selecting the final model, the clustering results are evaluated on the train set using three external evaluation criteria. The Adjusted Rand Index (ARI) [19], Normalized Mutual Information (NMI) [30] and Purity [34]. While $-1 \leq ARI \leq 1$ (random labels lead to values close to zero, perfect labels to 1), both NMI and Purity range between 0 (random labels) and 1 (perfect labels). For the selected category models with the best ARI values on the train set, the external measures are also calculated on the test set in order to evaluate the generalization performance.

3.3. Evaluating Unsupervised Selection of Category Models

Besides an supervised selection of the model, we also evaluate if we can choose the category model unsupervised. Instead of using the external evaluation criteria, an internal value is used to select the final model, namely the Davies-Bouldin (DB) index [16, 1], which has been calculated on the training data. A small value indicates compact and well separated clusters, hence we selected the model with the smallest DB index. The final models are evaluated with the external evaluation criteria on the test set in order to assess the quality of the chosen model.

3.4. Evaluating Classification

We evaluate how the learned features perform on a standard supervised classification task. In particular we train an L2-SVM using the LIBLINEAR library [14] on the feature representation $\mathbf{x}^L$. Both for CIFAR-10 and STL-10, 20% of the training images are used as validation set to determine the regularization parameter of the classifier. While we receive one final L2-SVM on the CIFAR-10 dataset, we obtain ten L2-SVMs for the STL-10 dataset (one for each training fold).

Again, we compare *4/3conv* with *6conv* using a varying number of filters. Additionally, in order to compare the proposed receptive field learning algorithm with a state-of-the-art method, we also train *4/3conv* architecture with the receptive field learning method introduced in [5]. For a fair comparison, we train the same number of receptive fields and select the same feature dimension for both receptive field learning algorithms.

4. Results

We report results illustrating the receptive fields, and quantitative results comparing the proposed approach with state-of-the-art methodology for unsupervised category learning as well as for classification. Furthermore, we evaluate to which extent the models can be selected based on an internal criterion.

4.1. Receptive Fields

In Figure 3 three typical examples for the learned receptive fields are illustrated (*4/3conv* model on CIFAR-10). It can be seen that the algorithm incorporates filters with similar orientation but varying color (a), similar orientation and color (b) and similar color but varying orientation (c).

4.2. Category Learning

Table 1 shows that the proposed method *4/3conv* outperforms the conventional approach *6conv* for all evaluation measures on both datasets. As expected, both outperform clustering applied directly to the input images and random clustering.

Method	CIFAR-10			STL-10		
	ARI	NMI	Pur.	ARI	NMI	Pur.
Random	0	0	0.1	0	0	0.1
SKM	0.06	0.10	0.24	0.06	0.12	0.26
6conv[7]+ SKM	0.09	0.16	0.27	0.10	0.18	0.28
4/3conv + SKM	**0.10**	**0.18**	**0.30**	**0.12**	**0.20**	**0.29**

Table 1. ARI, NMI and Purity are calculated for each model on the train data in order to select the final model. This table summarizes the values of the final models (both for CIFAR-10 and STL-10), calculated on the test set. Results for random and Spherical K-means (SKM) clustering applied directly on the images using 10 clusters are shown, too.

In terms of hyper-parameter settings, experiments showed that neither the type of initialization, the type of re-initialization nor the number of filters in the CNN has a strong influence on the external measures. Therefore, a lower number of filters seems preferable to reduce computation time. Furthermore, results indicate that feature-wise normalization helps to improve the performance. For example, the mean NMI of *4/3conv* model is $0.146(\pm 0.012)$ without and $0.170(\pm 0.006)$ with normalization on the CIFAR-10 dataset. Furthermore, both encoding functions achieve comparable performance on both datasets (e.g. on STL-10 the mean NMI is $0.194(\pm 0.012)$ for *mean-sparse* and $0.190(\pm 0.008)$ for *soft-threshold* encoding).

In-Depth Evaluation of Learned Categories Results demonstrate that the *4/3conv* model with the best external evaluation values categorizes images reflecting true object categories well, as can be seen in Figure 4. The confusion matrix of the ground truth class labels and the predicted category labels of the CIFAR-10 dataset is plotted in Figure 4 on the left-hand side. Each row of the matrix corresponds to one learned category, while every column corresponds to one ground truth class. The confusion matrix is rearranged according to the Hungarian method [22]. On the right-hand side of Figure 4 the categories are visualized. For each centroid, the ten nearest test images are plotted to illustrate the characteristics of each cluster.

When looking at the visualization of the learned categories, it is important to bear in mind that the given ground truth categorization is not the only possible reasonable one. The confusion matrix in Figure 4 shows that the category model distinguishes quite clearly between animals and non vital objects. While the clusters in row 1, 2, 3, 9, and 10 correspond to non vital objects, animals are categorized by the other centroids. As can be seen both in the confusion matrix and the nearest images, cluster "2" mainly groups white automobiles. Centroid "3" recognizes red automobiles and trucks, whereas cluster "4" mainly groups animals with white background. A big part of deer and bird images is contained in category "5". Also centroid "6" is a reasonable category, since cats and dogs have a similar appearance. While the major part of frogs is contained in cluster "7", a large part in category "8" is made up of horse images. Category "9" mainly contains ships and airplanes, where all visualized images have a clearly visible horizontal edge in the middle of the image. The major part of category "10" consists of automobiles and trucks with a bright background.

Method	CIFAR-10			STL-10		
	ARI	NMI	Pur.	ARI	NMI	Pur.
6conv[7]+ SKM	0.08	0.15	0.27	0.09	0.17	0.28
4/3conv + SKM	**0.10**	**0.18**	**0.30**	**0.12**	**0.20**	**0.29**

Table 2. This table shows ARI, NMI and Purity (calculated on the test set) for unsupervised selected models (via DB Index, calculated on the training set).

4.3. Unsupervised Model Selection

On both datasets unsupervised model selection based on the DB index chooses the *4/3conv* models that also have the best external evaluation criteria if an additional test set is used (Table 2). For the *6conv* architecture comparable models are selected for both datasets, as can be seen by comparing Table 1 and Table 2. The Pearson correlation coefficient [27] between the DB index and the external criteria ARI$(-0,55)$, NMI$(-0,37)$ and Purity$(-0,52)$, indicates that this is not a selection by chance.

4.4. Classification Results

The classification results, obtained on the test set, are illustrated in Figure 5(a) for CIFAR-10. Figure 5(b) contains the results for STL-10, where the error bars denote the standard deviation of the test accuracy, since 10 training folds are evaluated.

The proposed *4/3conv* architecture clearly outperforms the conventional *6conv* architecture on both datasets. While *mean-sparse* encoding leads to a slight performance decrease on CIFAR-10, it shows comparable results on STL-10 in comparison with

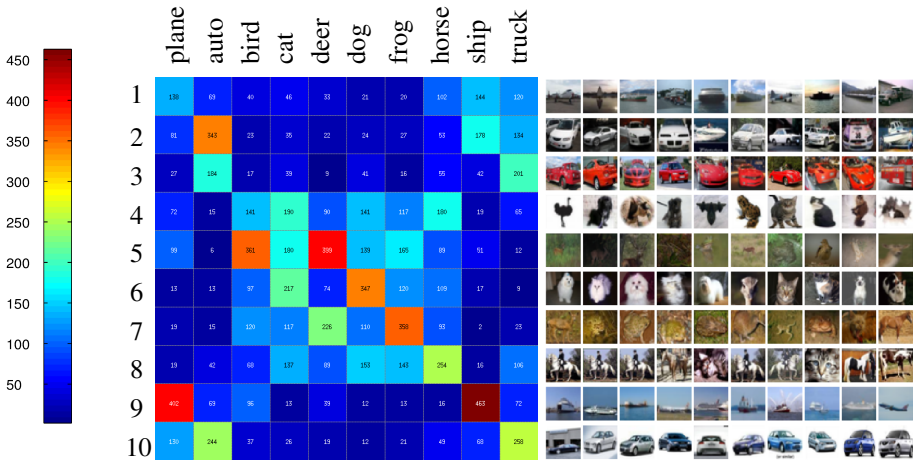

Figure 4. On the left we show the confusion matrix between clusters learned with a 4/3conv model and ground-truth classes of CIFAR-10. On the right, we show for each cluster center the nearest images in the test set.

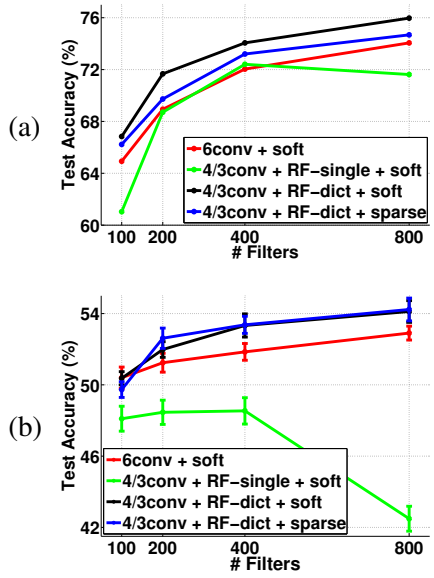

Figure 5. Test accuracy: This plot illustrates the classification performance of different approaches on the test set for a varying number of filters on the (a) CIFAR-10 and the (b) STL-10 dataset. While the proposed receptive field learning algorithm is denoted as *RF-dict*, *RF-single* refers to the state-of-the-art receptive field learning method introduced in [5].

the *soft-threshold* function.

Figure 5 also provides results for the *4/3conv* architecture using a state-of-the-art receptive field learning algorithm (*RF-single* [5]) in order to verify that the performance gains we achieve are not only the result of using a deeper architecture. As can be seen in Figure 5, the connections which are learned by our algorithm lead to a higher performance than the connections learned by the *RF-single* approach on both datasets.

5. Conclusion

In this paper we introduce a new receptive field learning algorithm, transferring the concept of the NIN approach to unsupervised learning, and propose a new encoding function, too. We evaluate how this contributes to improve unsupervised visual category learning as well as classification in comparison to unsupervised state-of-the-art algorithms. Results demonstrate superiority of the proposed method both on category learning and classification tasks. Finally, we demonstrate unsupervised category-model selection, leading to a fully unsupervised category detection method which does not lead to a performance decrease in comparison with model selection based on external criteria.

Acknowledgments

This work funded by the Austrian Federal Ministry of Science, Research and Economy, and the FWF (I2714-B31).

References

[1] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. *Signal processing*, 83(4):825–833, 2003. 6

[2] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1

[3] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[4] A. Coates, A. Karpathy, and A. Y. Ng. Emergence of object-selective features in unsupervised feature

learning. In *Advances In Neural Information Processing Systems*, volume 25, 2012. 2, 3

[5] A. Coates and A. Y. Ng. Selecting receptive fields in deep networks. In *Advances in Neural Information Processing Systems*, 2011. 2, 4, 5, 6, 8

[6] A. Coates and A. Y. Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*. Springer, 2012. 2, 4, 5

[7] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International conference on artificial intelligence and statistics*, 2011. 2, 4, 5, 7

[8] R. Collobert, C. Farabet, and K. Kavukcuoglu. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 4

[9] E. Culurciello, J. Jin, A. Dundar, and J. Bates. An analysis of the connections between layers of deep neural networks. *CoRR*, abs/1306.0152, 2013. 2

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 1

[11] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 1, 2

[12] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 766–774, 2014. 2

[13] A. Faktor and M. Irani. "clustering by composition"-unsupervised discovery of image categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1092–1106, 2014. 1

[14] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 6

[15] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 1

[16] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001. 6

[17] R. M. Haralick, K. Shanmugam, and I. H. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, (6):610–621, 1973. 1

[18] K. Hornik, I. Feinerer, M. Kober, and C. Buchta. Spherical k-means clustering. *Journal of Statistical Software*, 50(10):1–22, 2012. 5

[19] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985. 6

[20] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2

[21] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images, 2009. 5

[22] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 7

[23] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *International Conference on Image Processing*, volume 1. IEEE, 2002. 1

[24] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 2, 4

[25] D. G. Lowe. Object recognition from local scale-invariant features. In *Seventh IEEE international conference on Computer vision*, volume 2, pages 1150–1157, 1999. 1

[26] K. D. Miller. Synaptic economics: competition and cooperation in synaptic plasticity. *Neuron*, 17(3):371–374, 1996. 2

[27] A. J. Onwuegbuzie, L. Daniel, and N. L. Leech. Pearson product-moment correlation coefficient. *Encyclopedia of Measurement and Statistics*, 2007. 7

[28] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free?–weakly-supervised learning with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[29] T. Schlegl, S. M. Waldstein, W.-D. Vogl, U. Schmidt-Erfurth, and G. Langs. Predicting semantic descriptions from medical images with convolutional neural networks. In *Information Processing in Medical Imaging*. Springer, 2015. 1

[30] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003. 6

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[32] J. Wu, Y. Yu, C. Huang, and K. Yu. Deep multiple instance learning for image classification and auto-annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[33] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1

[34] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, Citeseer, 2001. 6