2015 IEEE International Conference on Data Mining

R^{2}FP: Rich and Robust Feature Pooling for Mining Visual Data

Wei Xiong*, Bo Du*, Lefei Zhang*, Ruimin Hu*, Wei Bian‡, Jialie Shen† and Dacheng Tao‡
*School of Computer Science, Wuhan University, National Engineering Research Center for MultimediaSoftware
Luojiashan, Wuhan, China, and Collaborative Innovation Center of Geospatial Technology, China
Corresponding author: Bo Du, remoteking@whu.edu.cn
Email: wxiong@whu.edu.cn, remoteking@whu.edu.cn, zhanglefei@whu.edu.cn, hrm1964@163.com
† School of Information Systems
Singapore Management University, Singapore
Email: jlshen@smu.edu.sg
‡Centre for Quantum Computation & Intelligent Systems
University of Technology, Sydney, NSW 2007, Australia
Email: brian.weibian@gmail.com, dacheng.tao@uts.edu.au

***Abstract*—The human visual system proves smart in extracting both global and local features. Can we design a similar way for unsupervised feature learning? In this paper, we propose a novel pooling method within an unsupervised feature learning framework, named *Rich and Robust Feature Pooling (R^{2}FP)*, to better explore rich and robust representation from sparse feature maps of the input data. Both local and global pooling strategies are further considered to instantiate such a method and intensively studied. The former selects the most conductive features in the sub-region and summarizes the joint distribution of the selected features, while the latter is utilized to extract multiple resolutions of features and fuse the features with a feature balancing kernel for rich representation. Extensive experiments on several image recognition tasks demonstrate the superiority of the proposed techniques.**

***Keywords*-pooling; autoencoder; representation learning.**

I. Introduction

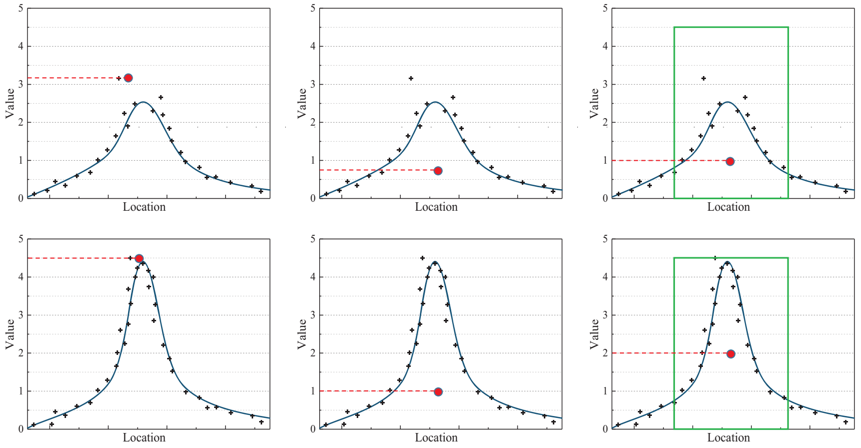
The performance of modern machine learning and data mining algorithms relies more and more on the quality of data representation, which embodies the inner structures and correlations of the data elements to make data more separable. Typical algorithms are unsupervised feature learning methods [1–3], which learn features without any supervisory information attached to the input data. The generic pipeline of the unsupervised feature learning system is composed of two main modules: the encoder module and the feature pooling module. The encoder module is established to learn the codebook/weights and encode the input data into feature maps, which are usually composed of sparse codes. Effective encoders such as sparse coding [4, 5], k-means clustering [6, 7], autoencoder [8–10] and restricted Boltzmann machine [11–13], have been proved promising to learn useful feature detectors. In this paper, the feature detectors are learned through an autoencoder. the hidden layer of the autoencoder is constrained with sparse constraint for better representation as the sparsity of the learned features is beneficial for the classification tasks [14]. In addition to sparse constraint, ReLU (Restricted Linear Unit)[15, 16] is utilized as the activation function to further increase the sparsity of the representation.

Following the encoder module, the pooling module [17–19, 19–25] is utilized to sub-sample the feature maps for compact representation. It removes the redundancy inside the features and provides invariance to small transformations of the input. Generally, two types of pooling methods have been investigated in prior work, the global pooling and the local pooling. The global methods provide efficient strategies to divide the global feature region into different resolutions of sub-regions and fuse the features of each resolution into the final representation, while the local methods calculate the statistic value of each sub-region that can summarize the joint distribution of the local features. The global pooling method has become a framework and any local pooling method can be embedded into it to pool the features in the sub-regions.

The leading example of global pooling methods is spatial pyramid matching (SPM) [26, 27]. SPM partitions an image into spatial bins of different scales and computes the histograms of each sub-region using the spatial pyramid matching kernel. Kaiming He et al. [28] utilized spatial pyramid pooling (SPP) based on SPM in convolutional neural networks (CNNs) [15] to pool the convolutional feature maps at multiple resolutions and learn rich representations in the inference stage of the network.

To generate suitable statistics of the local region, several impressive local pooling strategies have been proposed. The most universal methods are max pooling and average pooling, which adopt the maximum and the mean of the local features to represent the sub-regions, respectively. Boureau et al. [29] utilized p-norm of all the elements to represent region of the interest. The p-norm pooling introduces a factor P that can be tuned to fit data with different distributions as well as different encoders, thus better representations can

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.98



(a) Max Pooling (b) Average Pooling (c) Local R^{2}FP

Figure 1. Problems the conventional methods may face when pooling the sparse features in the local region A and B, compared with our proposed Local R^{2}FP . The top three sub-figures show the distribution of features in sub-region A, and the bottom three figures illustrate the feature distribution of sub-region B. In each sub-figure, the horizontal axis represents the location of a feature and the vertical axis stands for the value of each feature. Figure (a), (b) and (c) shows the possible pooling results of Max Pooling, Average pooling and the local R^{2}FP, respectively. The pooled feature is marked as a red point. As figure(a) shows, though feature pooled by max pooling shows outstanding separability, it loses so much information that it can't precisely summarize the distribution of the local features. Some outliers that are too large may be selected to be the representative feature. Figure (b)(top and bottom) shows when dealing with sparse codes, the pooled features of completely different regions A and B can be very close. Thus the pooled features are hard to be separated. The proposed method aims to solve these problems. As shown in figure (c), the proposed local R^{2}FP selects part of the larger features (larger features inside the green box are selected), the resulting pooled features of region A and region B are much more separated than that pooled by average pooling. Compared with max pooling, the proposed local R^{2}FP automatically reduces the negative effects of the outliers and the feature generated by local R^{2}FP fits the distribution of the local elements much better.

be extracted. Similar to p-norm pooling, other norm-form pooling methods have been proposed. In these methods, the pooled features are further normalized by the L_1, L_2 norm and the power normalization technique [21]. Zeiler et al. [30] proposed stochastic pooling, which selects the pooled features by sampling from a multinomial distribution formed by the activations of each sub-region, giving randomness to the generated representations. Boureau et al. [29] assumed the features were Bernoulli random variables and proposed smooth max pooling (SM) to combine the advantages of both average and max pooling.

All the above works have proved effective in various machine learning models and for various pattern recognition tasks. However, three main problems still exist in current feature pooling methods.

Firstly, local methods such as max and stochastic pooling select only one element of all the local features to represent the whole sub-region and discard the other features that may also have great impacts on the quality of the final representation. So the statistic generated by these methods may lose conductive information and perform poorly to describe the distribution of the sparse features in the sub-region. Furthermore, there may be some outliers and noise in the input data and they may be transformed to abnormal features which can not represent the local regions properly. If these noisy features are selected in the pooling procedure, the pooled features may be harder to be separated, as illustrated in Fig. 1(a) (top and bottom).

Secondly, local pooling methods such as average pooling, p-norm and other norm-form pooling methods, utilize all the elements in the local region to calculate the statistic value of the local features. When dealing with features with high sparsity, these methods can lead to a situation that the means or norms of features in different sub-regions are close to each other and tend to be very small, as most elements in the local areas approach zero, as illustrated in Fig. 1 (b). The resulting pooled features are then much harder to be correctly separated.

Thirdly, conventional global methods such as spatial pyramid pooling simply concatenate features of each resolution into a feature vector, without taking into consideration the importance of different levels of features. If equal importance is attached to different resolutions of features, the less vital features may take the leading place to greatly affect the fused features. There might be space for improvement that better fusion strategies can be discovered that will promote the quality of the final representation.

In this paper, we propose a novel pooling method, Rich and Robust Feature Pooling (R^{2}FP), to tackle with the three problems and make attempts to generate rich and robust representation. R^{2}FP includes global version and local version for different problems.

To solve the first and the second problem, the local R^{2}FP is formulated to generate better statistic of the local region.

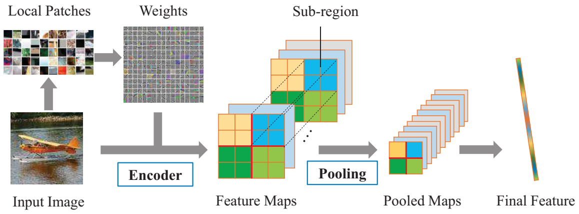


Figure 2. Illustration of unsupervised feature learning framework.

Specifically, in each sub-region, we first select the most conductive features and remove the less important ones through a feature selection principle. For sparse features in the local region, the few features that are much larger are supposed to be the major elements to describe the whole region. Then the statistic value of the selected features is calculated. The selected features are endowed with weights in order to highlight the importance of different features. In this way, more significant features are selected and utilized to generate the pooled feature that can better summarize the joint distribution of the local features. Meanwhile, the pooled features extracted from different sub-regions can be more distinguishable than the conventional methods, as illustrated in Fig. 1 (c).

To solve the third problem, the global R^{2}FP adopts a modified SPM model to extract features at multiple resolutions. The global area is firstly divided into sub-regions of different resolutions, then features extracted from each sub-region by the local methods are fused into a whole feature vector using a feature balancing (FB) kernel, which carefully balances the importance of different resolutions of features, enhancing the richness of the final representation.

The remainder of this paper is organized as follows. Details of the R^{2}FP method are provided in Section II. Experimental results and analysis on the benchmarks are presented in section III, followed by our main conclusions in Section IV.

II. Rich and Robust Feature Pooling

A. Whole Framework

Our proposed method is used in the unsupervised feature learning framework. As depicted in Fig. 2, the framework can be roughly defined as the concatenation of two main components, the encoder and the pooling module. In our work, the encoder used is a sparse autoencoder. In the training phase, patches are sampled from the image set to learn the weights of the encoder and the target response of each hidden neuron is constrained to be small. In the testing phase, the encoder module reorganizes the learned weights as filters or kernels and then transforms the input image into 3-D feature maps using the convolution operation [15] with these filters/kernels. Following the encoder, the pooling module divides the feature maps into sub-regions and calculates the statistical values of each local region using pooling methods. Finally, smaller feature maps that are more abstract and condensed representation of the input image are generated and reshaped into vectors for convenience of the subsequent classifiers.

1) Encoder module: Taking the input image as $\boldsymbol{I} \in R^{H\times W\times C}$, where C is the number of input image channels. $\boldsymbol{I}$ is transformed into feature maps with the learned kernels using the convolution operation.

$$\boldsymbol{S} = \sigma(\boldsymbol{I} * \boldsymbol{k} + \gamma_{opt}) \tag{1}$$

$\boldsymbol{S}$ denotes the learned feature maps, γ_{opt} is the optimized bias vector in the hidden layer, "*" denotes the convolution operation, $\boldsymbol{k}$ is the convolutional kernel learned by the autoencoder. σ is a non-linear activation function used in the hidden layer of the autoencoder. In the proposed work, ReLU is used as the activation function, i.e., $\sigma(\boldsymbol{x}) = \max(\boldsymbol{x}, 0)$.

2) Pooling module: If the generated feature maps are directly used as the final representation of the input image, then it will burden the subsequent classifier to correctly classify such tremendous amount of features. Besides that, the feature maps usually contain some redundant information that can mislead the classifier. In the unsupervised learning framework, the pooling module is utilized to further condense and abstract the features learned by the encoder and remove the redundancy.

The pooling module divides the spatial area of the maps into equivalent sizes of regions and calculates the statistical value of each local region, the process can be formulated as: $u = pool(\boldsymbol{v})$, $\boldsymbol{v}$ denotes features in the sub-region, $pool(\cdot)$ denotes the pooling method, u is the statistic of the sub-region. For instance, max pooling can be expressed as $u = \max\limits_{i=1}^{m}(\boldsymbol{v}_i)$ and average pooling as $q = \frac{1}{m}\sum\limits_{i=1}^{m}(\boldsymbol{v}_i)$.

In the following parts, we would detail the proposed pooling method R^{2}FP from the local version to global version in part B and part C respectively.

B. Local R^{2}FP

The local R^{2}FP deals with the local features and generates robust and separable representation from the sub-regions, to solve the first and the second problems concluded above in each sub-region. It selects K important features in each local region and removes the less conductive features. The feature selection principle is worth a careful investigation. In the view of the neural networks, the larger features are likely to be useful responses in correspondence with obvious object parts such as edges in the input image [31]. So the few features that are much larger than zero are the major clues indicating the identity of the input data and most of the features close to zero are less conductive features. From this point of view, in each local region, the maximum K elements are selected as the main features to be further investigated. This selection principle can also be analyzed from the perspective of robustness of the representation. As maximum values lead to a degree of invariance to the transformations and noise of the input, given the fixed number of selected features, in order to achieve better robustness, features close to the max feature are supposed to be selected.

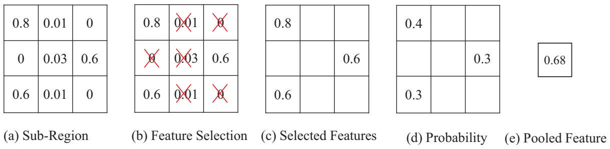
In our method, the local feature vector is firstly sorted in decreased order, the resulting vector is $\boldsymbol{v} \in \mathrm{R}^{N\times 1}$, where N is the number of features in the sub-region. Then the top K elements in $\boldsymbol{v}$ are selected according to the previous analysis, which is denoted as $\boldsymbol{r} = \boldsymbol{v}(1 : K)$. Following the feature selection process, the statistic scalar u of the selected features $\boldsymbol{r}$is calculated. In our method, an average of these features are calculated and a weighting vector $\boldsymbol{w} \in \mathrm{R}^{K\times 1}$ is utilized to determine the weights of the selected features.

$$u = f_K(\boldsymbol{v}) = \boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{r} \tag{2}$$

where $f_K(\cdot)$ denotes the local R^{2}FP method.

The weights of the selected features are vital factors that affect the final representation. In our method, two types of weights are designed. The first one is simply taking the average of the K features, i.e., w=$\frac{1}{K}\vec{\boldsymbol{I}}_{K\times 1}$. This weighting scheme is named "average weighting"(AW), which has been used in average pooling. However, average pooling takes the mean of all the features, while our method uses only part of them.

The second type of weights is to use the "probabilistic weighting" (PW). A simple example illustrating the local R^{2}FP is shown in Fig. 3.



(a) Sub-Region (b) Feature Selection (c) Selected Features (d) Probability (e) Pooled Feature

Figure 3. Example of the local R^{2}FP using the probabilistic weighting scheme. (a) is the sub-region to be pooled. (b) is the feature selection process. larger elements are retained and the smaller ones are discarded. (c) shows the selected features in the sub-region. (d) shows the probability of the retained features being chosen. (e) gives the calculated statistic of the selected features according to Eq. 2 and Eq. 3.

We regard the absolute value of each element r in the sub-region as its conditional probability $p(r|C)$ of being selected, where C is the class the sub-region belongs to. This is inspired by sampling from the multinomial distribution formed from the activations of each pooling region. Note that the value of local features may be larger than 1 as the activation function used in the autoencoder is ReLU for better sparsity. We regularize the features to be smaller than 1 by dividing the sum of all the selected features. Then the probability for each feature r to be selected is:

$$p(r_i\,|C) = r_i \Big/ \sum_{j=1}^{K} r_j \tag{3}$$

We use this probability as the weight of that feature when sub-sampling the selected features, then the weights can be formulated as: $\boldsymbol{w} = p(\boldsymbol{r}\,|C)$. This weighting scheme can be viewed as a type of normalization on the local features. It is supposed to work well with the proposed method as it can wipe off the influence of some noisy attributes that may hugely affect the representation of the region of interest when some of the features are not properly selected. Under the two designed weighting schemes, the proposed local R^{2}FP can be seen as a tradeoff between max and average pooling.

C. Global R^{2}FP

The global R^{2}FP first divides each global feature map into multiple resolutions of sub-regions. Then the local pooling method is conducted on each sub-region and multiple level of pooled features are generated. With a feature balance (FB) weighting kernel, the global R^{2}FP reallocates the importance degree of features at each resolution. At last, features at various resolutions are concatenated into a whole feature vector, which can then be classified by the subsequent classifier. Fig. 4 shows the pipeline of the global R^{2}FP. In Fig. 4, the 6×6 feature map is divided at resolution 1, resolution 2 and resolution 3, generating three pooled feature maps sized 1×1, 2×2 and 3×3 respectively. different resolutions of features are attached with different level of weights, which are expressed by the intensity of color in the figure. In the next part, we will detail the procedure of global R^{2}FP.

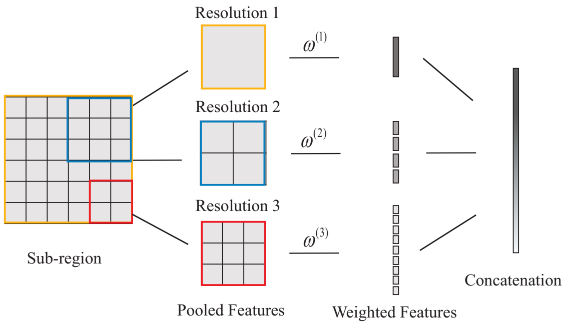


Figure 4. Pipeline of the global R^{2}FP. In the stripes representing the weighted features, the darker the stripes, the more important the features.

The idea of the global R^{2}FP is inspired from spatial pyramid matching (SPM). The SPM algorithm first divides the spatial areas of the input image into bins of different scales then utilizes the SPM kernel to calculate the number of matches or the histograms in bins of various levels to form a final representation. Specifically, considering two sets of two-dimensional vectors X and Y, which represent the feature coordinates of two feature maps. The matching kernel between X and Y is defined as follows:

$$\kappa^L(X,Y) = \frac{1}{2^L} I^0 + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}} I^L \tag{4}$$

where I^l is the number of matches at level l, $l = 1, 2, \ldots, L$, L is the number of matching levels.

We adopt the core idea of this matching method and propose a feature balance (FB) kernel to pool the convolved feature maps $\boldsymbol{S}$ that represent a single input image $\boldsymbol{I}$ for condensed representation. The FB kernel can be formulated as:

$$\kappa^l(\boldsymbol{I}) = \omega^{(l)} \cdot pool^{(l)}(\boldsymbol{S}) \tag{5}$$

where $\kappa^l(\boldsymbol{I})$ denotes to the l^{th} resolution of pooled features representing the input image $\boldsymbol{I}$, $pool^{(l)}(\cdot)$ means pooling operation at the l^{th} resolution. $\omega^{(l)}$ denotes to the weight of the l^{th} resolution of pooled features.

It should be noted that both the global R^{2}FP and the SPP method are pooling frameworks that can embed any local pooling method to summarize the statistic of the sub-regions.

Different from SPM, we define the levels of resolutions according to the size of the pooled feature maps, i.e., the number of sub-regions the input feature maps is divided. Specifically, at the l^{th} resolution, the spatial space of the convolved feature maps are divided into $l \times l$ sub-regions, and then the statistical value of elements in each sub-region is calculated, resulting in the final pooled feature maps sized $l \times l \times C$, where C is the number of channels.

We embed the local version of the proposed R^{2}FP into the novel kernel and get the full version of R^{2}FP.

$$\kappa^l(\boldsymbol{I}) = \omega^{(l)} \cdot f_K^{(l)}(\boldsymbol{S}) \tag{6}$$

where $f_K^{(l)}$ is the local R^{2}FP method conducted in the sub-regions at resolution l. Fig. 4 demonstrates the process of the proposed R^{2}FP.

In our method, arbitrary resolution can be selected to divide the global spatial region. For instance, for some datasets, resolution 1 (also denoted as the 1^{st} resolution or the 1^{st} level) and resolution 3 are preferred to form the final pooled features, while in other cases resolution 2 and resolution 4 may take the leading place to be chosen for better representation.

Different resolutions of features obey different distributions and demonstrates different statistic characteristics. Hence the final representations fused with these features by the proposed R^{2}FP can be good enough to boost the subsequent classifier compared with features pooled at a single resolution.

The conventional global method SPP fuses the features at different resolutions by a simple concatenation of the vector form of these features and discards the importance degree of different features, i.e., $\omega^{(l)} = 1$. In the proposed global R^{2}FP, the importance degree of features at different resolutions are stressed and the weights for the l^{th} resolution of features $\omega^{(l)}$ are investigated.

As features at lower resolutions are highly condensed compared to features at higher resolutions, each feature point at the lower resolutions can represent larger area back to the feature space in the convolved feature maps, hence the weights of these features are supposed to be larger than those of the higher resolutions of features. So we determine the weight of each feature by the area size it represents, i.e., the size of the sub-region, and the weights can be formulated as: $\omega^{(l)} = 1/l^2$.

D. Discussions

Firstly, we take a deep look into the reason why using only part of the features is better than using all the features in the local area. For a fair comparison, the local R^{2}FP under the Average Weighting scheme and average pooling are analyzed, which share similar operations on the selected features. We compare the distribution separability of features pooled by the two methods. We consider two feature sets $A = \{a_1, a_2, \ldots, a_N\}$ and $B = \{b_1, b_2, \ldots, b_N\}$, which represent two sub-regions that belong to two classes C_1 and C_2, respectively. Variables in both feature sets are sorted in decreased order. We examine the separation of conditional distributions $p(u_r|C_1)$ and $p(u_r|C_2)$, and $p(u_a|C_1)$ and $p(u_a|C_2)$, where u_r and u_a represent statistics of each sub-region generated by the local R^{2}FP and average pooling, respectively. Taking separability as a signal-to-noise problem, better separability can be achieved by increasing the distance between the means of the two class-conditional distributions. Hence the separation of class-conditional expectations of average pooled features φ_a can be formulated as:

$$\begin{aligned}\varphi_a &= \mathbb{E}(u_a\,|C_1\,) - \mathbb{E}(u_a\,|C_2\,) \\ &= \mathbb{E}(\frac{1}{N}\sum_{i=1}^{N} a_i) - \mathbb{E}(\frac{1}{N}\sum_{i=1}^{N} b_i)\end{aligned} \tag{7}$$

Analogously, the separation of features pooled by the local R^{2}FP can be formulated as:

$$\varphi_r = \mathbb{E}(\frac{1}{K}\sum_{i=1}^{K} a_i) - \mathbb{E}(\frac{1}{K}\sum_{i=1}^{K} b_i) \tag{8}$$

where K is the number of selected features in each sub-region in local R^{2}FP. For simplicity, we rewrite the expressions as: $\mathbb{E}(\frac{1}{N}\sum_{i=1}^{N} a_i) {=} \alpha$, $\mathbb{E}(\frac{1}{N}\sum_{i=1}^{N} b_i) = \beta$, $\mathbb{E}(\frac{1}{K}\sum_{i=1}^{K} a_i) {=} \alpha'$ and $\mathbb{E}(\frac{1}{K}\sum_{i=1}^{K} b_i) {=} \beta'$. We assume $\alpha > \beta$. Then we have:

$$\begin{aligned}\varphi_r &= \alpha' - \beta' \\ &= \mathbb{E}(\frac{1}{K}\sum_{i=1}^{K} a_i) - \mathbb{E}(\frac{1}{K}\sum_{i=1}^{K} b_i) \\ &= \frac{1}{K}\mathbb{E}(\sum_{i=1}^{K}(a_i - b_i)) \\ &= \frac{1}{K}\mathbb{E}(\sum_{i=1}^{N}(a_i - b_i) - \sum_{i=K+1}^{N}(a_i - b_i)) \\ &= \frac{N}{K}(\alpha - \beta) - \frac{1}{K}\mathbb{E}(\sum_{i=K+1}^{N}(a_i - b_i))\end{aligned} \tag{9}$$

As features in the sub-region are highly sparse, and the responses are restricted with ReLU, the smaller features are approaching 0 or equal to 0, thus for a proper K that is large enough (but K is still much smaller than N), we have $a_i \approx 0$ and $b_i \approx 0$ for $K+1 \leq i \leq N$, then we get $\frac{1}{K}\mathbb{E}(\sum_{i=K+1}^{N}(a_i - b_i)) \approx 0$. Following this equation, it can be summarized that :

$$\varphi_r = \alpha' - \beta' \approx \frac{N}{K}(\alpha - \beta) = \frac{N}{K}\varphi_a > \varphi_a \tag{10}$$

It indicates that features pooled by the local R^{2}FP have better separability than features pooled by average pooling, which exactly explains the phenomenon in Fig. 1(b) and (c) under the situation that the features to be pooled are sparse enough and most features are zero or approximate zero. For most classifiers, representations with better separability can lead to better classification performance, hence Eq. 10 proves the superiority of our method.

Secondly, we analyze the importance of the feature balancing (FB) kernel in the proposed global R^{2}FP, which is the main contribution of the proposed global R^{2}FP compared with the conventional SPP method. The FB kernel defines the weights/ importance degree $\omega^{(l)}$ of features at different resolutions in the feature concatenation process. We prove in the following analysis that when concatenating different feature vectors, the weight of each vector is the leading factor that determines the final direction of the concatenated long vector.

Without losing of generality, we consider two vectors $v_a \in R^{1\times n_a}$ and $v_b \in R^{1\times n_b}$ with weights λ_a and λ_b respectively. They are concatenated into vector $v_c \in R^{1\times(n_a+n_b)}$. We present vectors v_a and v_b in $R^{n_a+n_b}$ feature space as: $v'_a = [v_a, \vec{\boldsymbol{0}}_{1\times b}]$ and $v'_b = [\vec{\boldsymbol{0}}_{1\times a}, v_b]$. The concatenation process with weights can be viewed as an addition of vectors, i.e., v_c=$\lambda_a v'_a + \lambda_b v'_b$. It can be easily seen that the ratio of the weights λ_a/λ_b directly determines the direction of vector v_c in the new feature space. A corollary of this theorem is that different weights would lead to different distributions of the new features, which can further affect the choice of the supporting vectors and the division plane in the subsequent SVM.

III. Experiments

In the experiments, we implement and evaluate both the global and local version of the proposed R^{2}FP on three diverse datasets: MNIST [32], STL-10 [33], and Land-use [34, 35]. The local R^{2}FP is compared with some generally used pooling methods, they are max pooling, average pooling, stochastic pooling, p-norm pooling and smooth max pooling(SM). The global R^{2}FP is compared with spatial pyramid pooling (SPP). For fair comparison, all the methods are conducted on the same convolved feature maps learned by a single-hidden-layer sparse autoencoder with a KL sparse constraint. ReLU [16] is used as the activation function in the first and the second layer of the autoencoder. The pooled features representing the input images are reshaped into vectors and classified by a linear SVM implemented with LibLinear [36].

The performance of the proposed R^{2}FP is significantly affected by the main factor K, the number of selected features in each sub-region at each resolution. On each dataset, we evaluate and tune the factor K to find out the its optimal value K_{opt} on the validation set. The validation set is a 5% separation of the training set. All the experiments are conducted without data augmentation [37].

In the following parts, we first introduce the image datasets used in this paper. Then classification results of different methods on these datasets are demonstrated with complete analysis. After that, the main factor of the proposed method is evaluated on certain datasets. Conclusions and analysis of the experiment results are given in the last part.

A. Datasets Descriptions

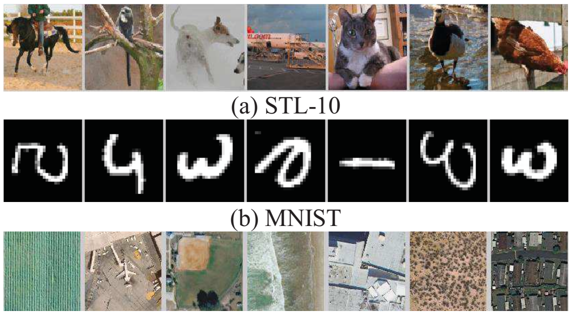
1) STL-10 dataset: The STL-10 dataset is a natural image set for developing unsupervised feature learning, deep learning and self-taught learning algorithms. STL-10 dataset contains 10 classes with a resolution of 96 × 96. Each class has 500 training images and 800 testing images. An additional 100000 unlabeled images are provided for unsupervised learning. We conduct our experiments on this dataset following the standard setting in [33, 38]. Fig. 5 (a) shows some samples of this dataset.

2) MNIST dataset: The MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been normalized in size and centered in a fixed-size image. Each image is a 28×28 gray image. Fig. 5 (b) shows some samples of this dataset.

3) Land-use dataset: The Land-use dataset contains manually extracted high-resolution aerial images downloaded from the U.S. Geological Survey (USGS) national map[1]. The resolution of the images is one foot per pixel and they

[1] Download Land-use dataset at http://vision.ucmerced.edu/datasets.

are cropped to 256×256 pixels. The dataset contains 21 scene categories with 100 samples per class. In the following experiments, we randomly take 80% of the images per category as the training samples and take the rest to test the performance the algorithms. Fig. 5 (c) shows some samples of this dataset.



(a) STL-10

(b) MNIST

(c) Land-use

Figure 5. Samples of the three image datasets.

B. Configurations of Modules

The main configurations of the modules in unsupervised feature learning framework on each dataset is firstly introduced, as shown in Table I. To compare the performance of the local pooling methods, we use a single resolution to divide the global regions of the feature maps on each dataset. To evaluate the performance of the global methods such as R^{2}FP and SPP, the global region of the feature maps are divided into multiple resolutions of sub-regions. "Multiple" indicates various resolutions used in the global methods. Table I also presents some key configurations of the autoencoder. "Map Size" denotes to the height and width of the convolutional feature maps to be pooled. "Target" provides the target response of each hidden neuron. "WD" denotes the weitht decay term used to constrain the magnitude of the weights connecting each layer. Dropout technique is utilized in all the experiments in the hidden layer of the sparse autoencoder and the probability of omitting each neural unit is set as 0.5.

Table I

CONFIGURATION OF UNSUPERVISED FEATURE LEARNING FRAMEWORK.

	STL-10	**MNIST**	**Land-use**
Kernel Size	13×13	8×8	13×13
Map Size	84×84	21×21	244×244
Hidden	800	500	1000
Multiple	1, 3	1, 3	1, 4
Target	0.01	0.1	0.01
WD	0.001	0.001	0.003

C. Evaluation of Local Method

In this part, the performance of the local methods are compared at the single resolution. The classification results on various datasets are demonstrated in Table II. Some notations should be made clear in Table II. "Level 1" denotes applying the pooling methods at the 1^{st} resolution, and "Level 3" denotes applying pooling methods at the 3^{rd} resolution. For instance, the feature maps learned from the image from STL-10 dataset sizes $84 \times 84 \times 800$, when applying the "Level 3" pooling, the size of the resulting pooled feature map will be $3 \times 3 \times 800$. "P-norm ($P = 2$)" means the parameters used in P-norm pooling is 2. For the proposed local R^{2}FP, "AW" means "Average Weighting" scheme and "PW" denotes to "Probabilistic Weighting" scheme. The presented results of the local R^{2}FP are achieved with the optimal value of K that can best boost the classification performance.

Table II shows clearly that the local R^{2}FP under the "Average Weighting" scheme outperforms the conventional pooling methods in classification tasks, indicating that better representations are learned by the proposed method. Especially, as a "tradeoff" between max pooling and average pooling (equivalent to max pooling when $K = 1$ and to average pooling when $K = N$), the local R^{2}FP achieves much better performance than the two typical pooling methods. In the challenging STL-10 dataset, the local R^{2}FP promotes the recognition accuracy by nearly 4% compared to max and average pooling. The results proves that 1) there is a considerable amount of meaningful information to be discovered from the sparse feature maps generated by the unsupervised learning framework and pooling is an effective way to conduct this information mining and representation learning work; and 2) though the proposed local R^{2}FP is a simple variant of max and average pooling, it can preserve significant features in the sub-region and remove the redundancy to generate statistic that better summarizes the joint distribution of the local features, especially for sparse codes.

Results of the proposed method under different weighting schemes in each sub-region shows that the weights of the local features greatly affect the quality of representations extracted by R^{2}FP. From Table II, though the performance of R^{2}FP under "Probabilistic Weighting" scheme is inferior to that under "Average Weighting" scheme in most cases, it equals to or is superior to the some of the conventional methods, and it outperforms max and average pooling in most experiments, demonstrating that both the two weighting schemes designed for the local features are effective.

It should be noted that max pooling performs poorly on these datasets. Boureau et al. [29] have given both theoretical and experimental supports on max pooling, showing that max pooling can generate features with very good separability. This conflict may partly show that separability is not the only criteria to judge the performance of the pooled features. The reason of this unexpected performance may be the lack of richness of features used to calculate the statistic of the local region, as max pooled features may lose useful information when other elements in a sub-region are neglected. Hence to achieve better performance, we should reach a tradeoff between the separability and the richness of features. Another assumption for this result is that the max element of a sub-region may be an outlier or noise that can not stand for the distribution of features in the whole sub-region. A toy example of this assumption is demonstrated in Fig. 1. The proposed method selects only part of the features and reduces the negative influence of the noise and outliers, thus it can get more robust representation.

Table II
CLASSIFICATION RESULTS OF EACH LOCAL POOLING METHOD.

Methods	**STL-10**	**MNIST**	**Land-use**
Max	54.5±0.6	99.12±0.08	68.6±0.8
Average	54.3±0.7	99.23±0.10	73.1±0.9
Stochastic	56.8±0.9	99.15±0.12	73.5±1.1
P-norm (P=2)	57.2±0.6	99.36±0.06	74.5±0.7
Smooth Max	57.7±0.7	99.32±0.07	74.8±0.8
Local R^{2}FP(PW)	57.1±0.8	99.19±0.11	74.0±1.0
Local R^{2}FP (AW)	**58.0±0.7**	**99.42±0.10**	**75.1±0.9**

D. Evaluation of Global Method

Note that the global pooling methods are frameworks that provide principles of dividing the spatial regions of the input feature maps into sub-regions at multiple resolutions, any local pooling method can be applied to the divided sub-regions. To evaluate the performance of the global methods, we embed different global pooling methods with the same local pooling method to see the final performance.

Table III shows the results of each local pooling method combining the global pooling methods, SPP and the global R^{2}FP at multiple resolutions.

The resolutions used on each dataset have been displayed in Table I. For local R^{2}FP embedded into the global methods, the optimal number of selected features K_{opt} at each resolution is first determined by trail and error to make the local R^{2}FP achieve the best performance. For example, on STL-10, $K_{opt} = 400$ for resolution 3 and $K_{opt} = 1000$ for resolution 1. Results in Table III shows that on each dataset, when embedded with the same local pooling method, the global R^{2}FP achieves the better performance than SPP in most cases, indicating that different resolutions of features are better fused using the proposed FB kernel in R^{2}FP than the fusion strategy used in SPP which simply concatenates different features. The main reason for the superiority of the proposed R^{2}FP lies in that our method considers the importance degree of different features resulting in better statistic characteristics of the final representation.

From Table III, it can also be seen that combining with the same global method, the local R^{2}FP still performs better than the other local methods on all the datasets. Considering the whole pooling process, results show that the global R^{2}FP embedded with the local R^{2}FP achieves the best performance in all the 14 combinations of the pooling methods and on all the datasets, demonstrating the superiority of the proposed method.

Comparing the results in Table II and Table III, classification performances of all the pooling methods are promoted when multiple resolutions of features are utilized compared to pooling features at single resolution. On Land-use dataset, there is a impressive promotion of over 5% when use R^{2}FP at two resolutions (presented in Table III) than at only single resolution (presented in Table II). And under the same settings, our proposed global R^{2}FP promotes the classification performance the most which proves that R^{2}FP has the capacity of extracting and fusing conductive features to form rich representations.

E. Parameter Sensitivity Analysis

In this section, we conduct experiments on MNIST dataset to investigate the discipline of the optimal number K_{opt} of selected features in the sub-region, which is the key factor in the proposed local R^{2}FP. As the local R^{2}FP deals with the sparse features in the sub-region and the sparseness of features is a vital start point that inspires the proposed method, it can be assumed that K_{opt} can be affected by the sparsity of the feature maps, which is determined by the target response of each hidden neuron ρ in the autoencoder. So it's necessary to study on the relationship between ρ and the number of selected features K.

Before that, we wil firstly check by experiments that the learned feature maps are sparse enough. Fig. 6 illustrates the histogram of features to be pooled under different target responses. Most of the features are less than 0.1, while only a few are much larger, proving that the feature maps learned by the encoder are highly sparse. The histograms demonstrate the leading place of ρ on the sparsity of learned feature representations. Results shown in Fig. 6 also verify the correctness of the theory given in section II. In Eq. 10, we have proved that larger separability can be achieved by the local R^{2}FP with the AW scheme than average pooling, under the assumption that a proper K can be found to make the $N - K$ smallest features in sub-region A and B small enough, thus the hypothesis $\mathbb{E}(\sum_{i=K+1}^{N}(a_i - b_i)) \approx 0$ can be true. Fig. 6 demonstrates the fact that more than 60% of features are zero when the sparsity constraint is utilized in the autoencoder combined with the ReLU activation function. Assume M features in both the sub-regions A and B are zero, then when $N - M < K < N$, it can be guaranteed that the separability of the proposed local R^{2}FP is better than that of average pooling. As we have: $\mathbb{E}(\sum_{i=K+1}^{N}(a_i-b_i)) = 0$, then according to Eq. 9 , equation $\varphi_r = \alpha' - \beta' = \frac{N}{K}(\alpha - \beta) = \frac{N}{K}\varphi_a > \varphi_a$ is true.

The classification results of local R^{2}FP as ρ changes on MNIST dataset is demonstrated in Fig. 7.

With the change of ρ, the classification performance of the unsupervised feature learning framework varies synchronously. However, the optimal number of selected fea-

Table III
RESULTS OF EACH LOCAL POOLING METHOD COMBINING THE GLOBAL POOLING METHODS AT MULTIPLE RESOLUTIONS.

Methods	STL-10	MNIST	Land-use
Max + SPP	54.7 ± 0.7	99.25 ± 0.10	71.2 ± 1.2
Max + Global R^{2}FP	**55.1 ± 0.8**	**99.35 ± 0.12**	**74.0 ± 1.3**
Average + SPP	55.0 ± 0.6	99.33 ± 0.08	73.8 ± 1.3
Average + Global R^{2}FP	**56.4 ± 0.8**	**99.37 ± 0.09**	**74.9 ± 1.2**
Stochastic + SPP	55.8 ± 1.1	**99.29 ± 0.11**	75.7 ± 1.4
Stochastic + Global R^{2}FP	**57.4 ± 1.0**	99.29 ± 0.12	**78.6 ± 1.0**
P-norm ($P = 2$) + SPP	56.0 ± 0.7	99.43 ± 0.08	76.4 ± 1.1
P-norm + Global R^{2}FP	**59.2 ± 0.7**	**99.45 ± 0.10**	**76.6 ± 0.8**
Smooth Max + SPP	57.6 ± 0.6	99.38 ± 0.08	78.8 ± 1.1
Smooth Max + Global R^{2}FP	**59.3 ± 0.9**	**99.43 ± 0.09**	**85.1 ± 0.9**
Local R^{2}FP(PW) + SPP	57.4 ± 0.8	99.30 ± 0.06	77.6 ± 1.0
Local R^{2}FP (PW) + Global R^{2}FP	**58.2 ± 0.8**	**99.31 ± 0.06**	**84.8 ± 1.2**
Local R^{2}FP (AW) + SPP	58.3 ± 0.6	99.52 ± 0.07	81.7 ± 1.1
Local R^{2}FP (AW) + Global R^{2}FP	**59.8 ± 0.7**	**99.55 ± 0.08**	**86.7 ± 1.5**

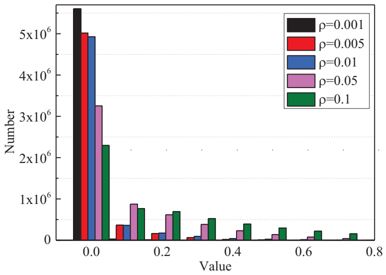


Figure 6. Histograms of the generated convolutional feature maps learned from a testing image in STL-10. The vertical axis represents the number of features belonging to a certain value range, e.g., (0.1, 0.2]. It can be concluded that under different target response (ρ = [0.1, 0.05, 0.01, 0.005, 0.001]) in the training process of autoencoder, the resulted features are highly sparse indeed. Moreover, the smaller value of ρ leads to higher sparsity of the features.

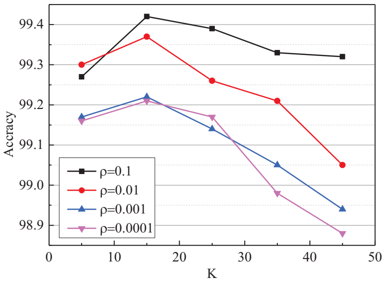


Figure 7. Classification results of local R^{2}FP as ρ changes on MNIST dataset under different sparsity target ρ. The size of sub-regions to be pooled is 21 × 21. K varies from 5 to 45 with a step of 10. Four values in different orders of magnitude of ρ = [0.1,0.01,0.001,0.0001] are utilized to test the performance of the proposed pooling method.

tures in each sub-region K_{opt} remains almost untouched ($K_{opt} = 15$). This phenomenon forces us to take a deep study on the importance of each feature in the sub-region. Which features in the local region are more important and more conductive to the final representation? Can we conclude the larger the responses, the more important the features? Combining the results of Fig. 6 and Fig. 7, this claim is not completely the truth.

When decreasing the target response, the ratio of features approaching zero increases obviously. However, the value of K_{opt} remains invariant, i.e., the number of conductive features in the sub-region stays almost the same. It can be inferred that it is the relative value of each feature compared to other elements in the local region but not the absolute value that makes the feature important. In a sparse feature region, the larger the relative value of the feature, the more important it is. It can also be inferred that there do exists a number of features in the local region that make little contribution to the final representation of the input image. From this perspective, one function of R^{2}FP is to wipe off the relatively smaller features, not the features under a fixed threshold (as the optimal number of selected features K_{opt} is usually less than the number of features N in the sub-region). In this way, the proposed pooling method can remove the less important features and retain the most important features to generate compact and invariant representations.

From Fig. 7, we can draw an empirical conclusion that $K_{opt} \approx N$. As the number of selected features in each sub-region is the only parameter to adjust in our method, and we have given an empirical value of the optimal K, then the proposed R^{2}FP is almost no parameters to tune, indicating that the proposed method is stable and practical.

IV. CONCLUSION

In this paper, we have introduced a novel pooling method R^{2}FP, together with its local and global versions, for extracting features from feature maps learned through a sparse autoencoder. Local R^{2}FP selects the most conductive features in the sub-region and summarizes the joint distribution of the selected features, which enhances the robustness of the final representation and promotes the separability of the pooled features. Global R^{2}FP is utilized to extract multiple resolutions of features and fuse the features with a feature balancing kernel for rich representation. Experiments on the benchmarks demonstrate that: 1) Both the global and local versions of R^{2}FP outperform the conventional methods. 2) Compared with pooling methods at a single resolution, the global R^{2}FP at multiple resolutions can extract rich representation which can greatly boost the performance of the classifier. On Land-use dataset, there is a 10% promotion in accuracy. 3) In the local R^{2}FP, the optimal number of selected features in the sub-region remains steady with different sparse constraints, demonstrating that the proposed algorithm is practical and stable.

Acknowledgment

The authors would like to thank the area chair and the anonymous reviewers for their careful reading and helpful remarks, which would contribute in improving the quality of this paper. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant (No. 2012CB719905), the National Natural Science Foundation of China under Grants (No. 61471274 and 61231015), the Chancellor's Postdoctoral Research Fellowship of the University of Technology, Sydney, Australian Research Council Projects (No. FT-130101457 and LP-140100569), the National High Technology Research and Development Program of China (863 Program) (No. 2015AA016306), the Natural Science Foundation of Hubei Province under Grants 2014CFB193 and the Fundamental Research Funds for the Central Universities under 2042014kf0239.

References

[1] M. Ranzato, F. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *CVPR*, 2007.

[2] Y. Bengio, A. Courville, and P. Vincent, "Unsupervised feature learning and deep learning: A review and new perspectives," *CoRR, abs/1206.5538*, 2012.

[3] L. Shi, L. Du, and Y. Shen, "Robust spectral learning for unsupervised feature selection."

[4] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision research*, 1997.

[5] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, "Learning convolutional feature hierarchies for visual recognition," in *NIPS*, 2010.

[6] A. Goswami, R. Jin, and G. Agrawal, "Fast and exact out-of-core k-means clustering," in *ICDM*, 2004.

[7] A. Coates and A. Ng, "Learning feature representations with k-means," in *Neural Networks: Tricks of the Trade*. Springer, 2012.

[8] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008.

[9] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *JMLR*, 2010.

[10] R. Min, D. A. Stanley, Z. Yuan, A. Bonner, and Z. Zhang, "A deep non-linear feature mapping for large-margin knn classification," in *ICDM*, 2009.

[11] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, 2006.

[12] M. Norouzi, M. Ranjbar, and G. Mori, "Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning," in *CVPR*, 2009.

[13] Y. Sakai and K. Yamanishi, "Data fusion using restricted boltzmann machines," in *ICDM*, 2014.

[14] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Analysis and Machine Intelligence,*, 2013.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[16] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[17] Z. Wang and X. Shang, "Spatial pooling strategies for perceptual image quality assessment," in *ICIP*, 2006.

[18] A. Moorthy and A. Bovik, "Visual importance pooling for image quality assessment," *Selected Topics in Signal Processing, IEEE Journal of*, 2009.

[19] Y.-L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition," in *ICCV*, 2011.

[20] O. Russakovsky, Y. Lin, K. Yu, and F. Li, "Object-centric spatial pooling for image classification," in *ECCV*, 2012.

[21] J. Feng, B. Ni, Q. Tian, and S. Yan, "Geometric lp-norm feature pooling for image classification," in *CVPR*, 2011.

[22] D. Lin, C. Lu, R. Liao, and J. Jia, "Learning important spatial pooling regions for scene classification," in *CVPR*, 2014.

[23] N. Murray and F. Perronnin, "Generalized max pooling," in *CVPR*, 2014.

[24] BogdanMiclut, "Committees of deep feedforward networks trained with few data," in *Pattern Recognition*, 2014.

[25] L. Shen, J. Lin, S. Wu, and S. Yu, "Hep-2 image classification using intensity order pooling based features and bag of words," *Pattern Recognition*, 2014.

[26] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[27] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *arXiv:1406.4729*, 2014.

[29] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *ICML*, 2010.

[30] M. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv:1301.3557*, 2013.

[31] ——, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, 1998.

[33] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *ICAIS*, 2011.

[34] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *ICAGIS*, 2010.

[35] A. M. Cheriyadat, "Unsupervised feature learning for aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, 2014.

[36] R. Fan, K. Chang, C. Hsieh, X. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *JMLR*, 2008.

[37] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv:1207.0580*, 2012.

[38] Q. V. Le, A. Karpenko, J. Ngiam, and A. Ng, "Ica with reconstruction cost for efficient overcomplete feature learning," in *NIPS*, 2011.