Learning Structure and Strength of CNN Filters for Small Sample Size Training

Rohit Keshari, Mayank Vatsa, Richa Singh
IIIT-Delhi, India
{rohitk, mayank, rsingh}@iiitd.ac.in

Afzel Noore
Texas A&M University-Kingsville, USA
Afzel.Noore@tamuk.edu

Abstract

Convolutional Neural Networks have provided state-of-the-art results in several computer vision problems. However, due to a large number of parameters in CNNs, they require a large number of training samples which is a limiting factor for small sample size problems. To address this limitation, we propose SSF-CNN which focuses on learning the "structure" and "strength" of filters. The structure of the filter is initialized using a dictionary based filter learning algorithm and the strength of the filter is learned using the small sample training data. The architecture provides the flexibility of training with both small and large training databases, and yields good accuracies even with small size training data. The effectiveness of the algorithm is first demonstrated on MNIST, CIFAR10, and NORB databases, with varying number of training samples. The results show that SSF-CNN significantly reduces the number of parameters required for training while providing high accuracies on the test databases. On small sample size problems such as newborn face recognition and Omniglot, it yields state-of-the-art results. Specifically, on the IIITD Newborn Face Database, the results demonstrate improvement in rank-1 identification accuracy by at least 10%.

1. Introduction

Convolutional Neural Network (CNN) is a multilayer representation learning architecture which has received immense success in multiple applications such as object classification, image segmentation, and natural language processing. From LeNet [25] to AlexNet [23], GoogleNet [39], VGG-Net [38], ResNet [17], and now DenseNet [18], given large training data, CNNs have shown state-of-the-art performance for several applications. However, large training data is also a limiting requirement for applications with small sample size and many of these architectures easily overfit on small training samples. For example, as shown in Figure 1, a face recognition model trained on large training data of adult faces (e.g. CelebA or LFW databases) may not provide good performance when tested for newborn face recognition [2, 3]. In newborn face recognition, the available training data may be small and therefore, even after fine-tuning, standard deep learning based face recognition models may not yield high performance.

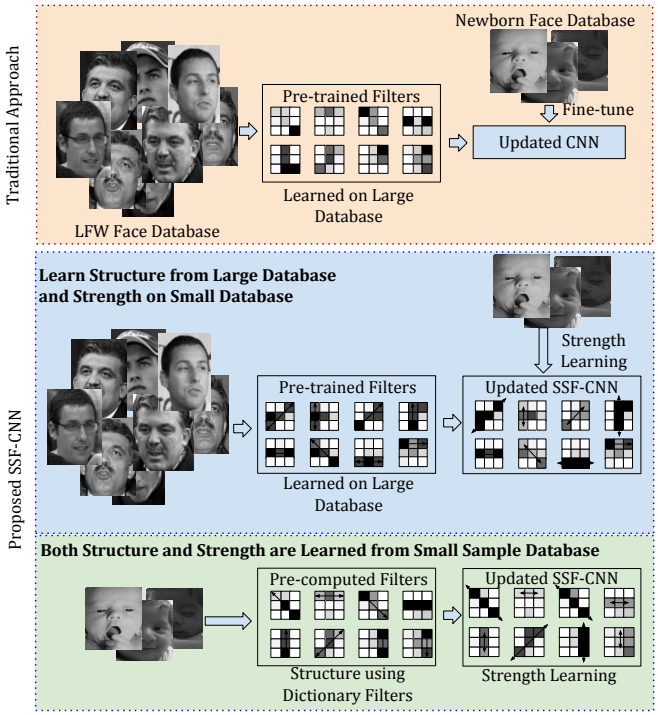

Figure 1. Face recognition models trained on adult face images may not provide good performance for newborn face recognition. SSF-CNN proposes to learn structure and strength of the filters for improving the classification performance for small sample databases.

To address the challenge of small sample size, researchers have proposed algorithms focusing on CNN initialization tricks and modifications to CNN architecture. Erhan *et al.* [9] have investigated the importance of unsupervised pre-training of deep architecture and empirically shown that pre-trained weights of the network generalize better than randomly initialized weights. Similarly, Mishkin and Matas [32] have proposed Layer-Sequential

Unit-Variance (LSUV) initialization that utilizes the orthonormal matrices to initialize the weights of each convolutional layer and normalize the weight to the unit variance. Along the same lines, pre-defined handcrafted filters are also proposed to handle the small sample size problem. For example, Andén and Mallat [1] propose Scattering network (ScatNet) which is a CNN like architecture where pre-defined Morlet filter bank is utilized to extract features. However, these handcrafted filters may not represent the true distribution of the data and hence extract not-so-meaningful features. To overcome this limitation, Oyallon *et. al.* [35] have proposed hybrid network, where they have utilized ScatNet feature followed by CNN architecture. Similarly, Chan *et. al.* [4] propose PCANet architecture that utilizes Principal Component Analysis (PCA) to learn the filter banks. They also present an extension, termed as LDANet, in which the selection of the cascade filters are trained from Linear Discriminant Analysis (LDA). Gan *et al.* [12] propose a PCA-based Convolutional Network (PCN) which has the influence of both CNN [20] and PCANet [4]. Dan *et al.* [45] utilize the concept of kernel PCA to further improve the PCANet architecture. Zeng *et al.* [49] propose a multilinear discriminant analysis network (MLDANet) which is a variant of PCANet and LDANet. Feng *et al.* [11] propose Discriminative Locality Alignment Network (DLANet) which is based on manifold learning. These architectures learn filters in stack-wise manner, and once the network (filters) is trained, generally, it is not allowed to fine-tune the filters on other databases.

In other research directions for small sample size training, Mao *et al.* [31] propose a neural network learning method based on posterior probability (PPNN) to improve the accuracy. Ngiam *et al.* [34] propose tied weights in a filter using tiling parameter which handles the total number of learning parameters. In another work, Indian Buffet Process (IBP) priors are utilized to propose semi-supervised ibpCNN which shows better generalizability [10]. Xiong *et al.* [47] propose Structured Decorrelation Constrained (SDC) for hidden layers. The authors have also proposed a novel approach termed as Regularized Convolutional Layers (Reg-Conv) that can help SDC to regularize the complex convolutional layers. Similarly, Cogswell *et al.* [5] propose DeConv loss for CNN architecture that helps in training small databases.

One of the major problems with adapting pre-trained CNN models for small sample size problems, as mentioned previously, is large amount of parameters; therefore, insufficient training samples may cause overfitting. If we reduce these parameters to a significantly small number, then the problem can be addressed in a better way. This paper focuses on two novel ways to develop CNN based feature representation algorithm for small sample size problems: (i) associating "strength" parameter to control the effect of each pre-trained filter, and (ii) utilizing a generalizable approach that pre-learns the "structure" of the filters using small training samples. The proposed architecture is motivated from ScatNet but in place of pre-defined filters, we utilize dictionary learning model to *pre-learn* the filters. Further, unlike CNN approaches where we update the weights in every iteration, we introduce strength of the filter and update only the strength parameter not the filters. The introduction of "*strength*" of filters significantly reduces the number of parameters to learn (detailed calculations shown later) and therefore avoids overfitting with limited training. Experiments are performed on object classification databases, MNIST [26], CIFAR-10 [22], NORB [27], Omniglot [24], and a challenging small sample size database of newborn faces [3]. Comparison with existing algorithms show that the proposed approach achieves state-of-the-art performance for small sample size problems and significantly reduces the number of parameters to learn/fine-tune.

2. Proposed SSF-CNN

It is difficult to learn the entire network from scratch while training with small size databases. Existing approaches with pre-defined or handcrafted filters [1], and pre-trained filters [4, 12, 49], may not allow fine-tuning the filters and therefore, the learned model may not represent the true data distribution for small sample size problems. To mitigate these challenges, we propose a novel approach, termed as Structure and Strength Filtered CNN (SSF-CNN), which has two components: (i) structure of the filter and (ii) strength of the filter. It is our hypothesis that structure of the CNN filters can be learned from either domain specific larger databases or from other representation learning paradigms that require less training data for instance, dictionary learning [40, 41]. It is well known that matrix factorization or dictionary learning allows us to learn the *dictionary* that helps encoding the representative features. If we represent CNN filters using dictionary, it can provide the "structure"; however, it may not be well optimized for the classification task. Therefore, the next part of the framework is computing "strength" of every filter to adapt the weights of these filters according to the data characteristics. Strength can be interpreted as the attuning parameter to update or adapt the filters based on the small size training data. For illustration, columns (a) to (d) in Figure 2 represent the samples from trained dictionary filters for the MNIST database and columns (e) to (h) represent the updated filters where changes are due to the strength parameter.

Formally, in the proposed approach, first the hierarchical dictionary filters are learned to initialize the CNN, followed by learning the strength parameter to train the CNN model. We introduce strength parameter '$\mathbf{t}$' for the CNN filters '$\mathbf{W}$' which allows the network to assign *weight* for each filter based on its structural importance. In CNN model,

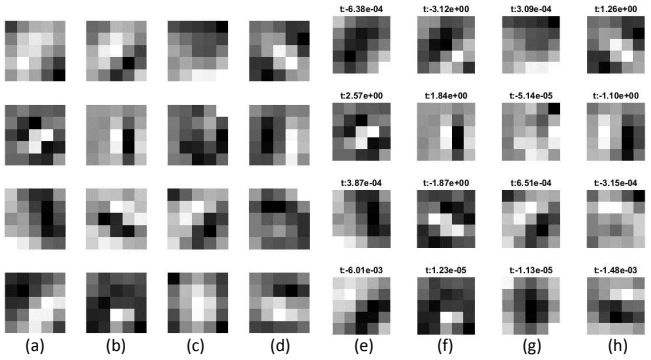

Figure 2. Filters (a) to (d) are dictionary trained filters. Filters (e) to (h) illustrate the change due to the proposed strength parameter in CNN architecture. These filters are trained on MNIST database.

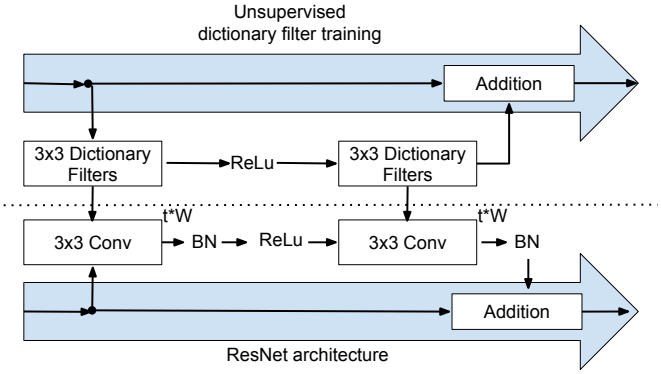

Figure 3. The proposed SSF-CNN architecture for initializing the ResNet architecture with the filters learned from dictionary.

strength and structural parameters $\mathbf{t}$ and $\mathbf{W}$ can be learned in two ways: 1) pre-train $\mathbf{W}$, use it in CNN by freezing the values of $\mathbf{W}$ followed by learning the strength $\mathbf{t}$, and 2) pre-train $\mathbf{W}$ which is used to initialize the CNN model followed by learning $\mathbf{t}$ and $\mathbf{W}$ iteratively. While the second approach which simultaneously learns both structure and strength may be desirable, the first approach requires very few parameters to be trained in CNN model. We next describe the approach to hierarchically learn $\mathbf{W}$, filters of CNN model, using dictionary learning followed by learning the strength parameter $\mathbf{t}$.

2.1. Learning Structure of Filters

In this research, we propose to use dictionary learning algorithm for learning the structure of the filters. The algorithm can be divided into two steps: 1) learn hierarchical dictionary filters and utilize trained dictionary filters to initialize the CNN, and (2) train CNN with dictionary initialized filters.

Hierarchical Dictionary Filter Learning: Dictionary learning focuses on learning a sparse representation of the input data in the form of a linear combination of basic elements or atoms [8, 29, 30, 40, 41]. For a given input $\mathbf{Y}$, a dictionary $\mathbf{D}$ is learned along with the coefficients α:

$$\min_{\mathbf{D},\,\alpha} \|\mathbf{Y} - \mathbf{D}\alpha\|_F^2 \,,\ such\ that\ \ \|\alpha\|_0 \leq \tau \qquad (1)$$

where, the ℓ_0-norm imposes a constraint of sparsity on the learned coefficients and τ corresponds to the maximum number of non-zero elements. Often, the ℓ_0-norm is relaxed and the updated dictionary learning formulation can be written as:

$$\min_{\mathbf{D},\,\alpha} \|\mathbf{Y} - \mathbf{D}\alpha\|_F^2 + \lambda||\alpha||_1 \qquad (2)$$

where, λ is a regularization parameter which controls the sparsity promoting ℓ_1-norm. In this research, we utilize dictionary learning to pre-train the filters of CNN in a hierarchical manner. As shown in Algorithm 1, a hierarchical dictionary learning technique is utilized to initialize the CNN model (ResNet [17]). The trained dictionary atoms are used to convolve over the input image. After convolution, feature maps are normalized according to the activation function (e.g. ReLu) used in CNN models. Figure 3 presents the structure of a block of the SSF-ResNet architecture. The extracted feature map is an input for the next level of the hierarchical dictionary. In this manner, the number of dictionary layers is same as the number of convolutional layers in CNN models. In Algorithm 1 $extractPatch$ function is used to tessellate the input image into small patches. The trained dictionary is organized in the two-dimensional array where each filter is arranged in one column. These learned filters are reshaped and convolved over the input image to produce the feature maps for the next level of the dictionary.

Algorithm 1 Hierarchical Dictionary Filter Learning

Notation: N is a number of training samples, n number of extracted patches, y is a patch from Y
Input: X_N
Output: D
for each layer $l := 1$ to $numLayer$ **do**
  $[x^n]^N \leftarrow extractPatch(X_N)$
  Y $\leftarrow$ reshape($[x^n]^N$)
  $\min_{D\in\mathbb{R}^{m\times k}} \frac{1}{n}\sum_{i=1}^{n} \min_{\alpha^i}(\frac{1}{2}||y^i - D^l\alpha^i||_2^2 + \lambda||\alpha^i||_1)$
  W $\leftarrow$ reshape(D^l)
  for $j := 1$ to N **do**
    $fmap_j = X_j * W$
  end for
  $X_N \leftarrow ReLu(fmap)$
end for

Training CNN with Dictionary Initialized Filters: Typically, CNN has multiple convolutional layers, each layer has multiple filters, and these filters are trained using stochastic

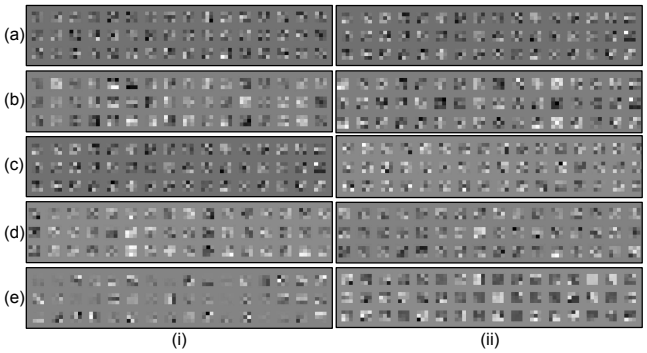

Figure 4. Filter visualization of the (i) 1^{st} layer and (ii) 2^{nd} layer of the ResNet architecture on CIFAR10 dataset. (a) Xavier [13] initialized filters at zero epoch, (b) Xavier [13] initialized filters are trained on 1000 training samples, (c) MSRA [16] initialized filters at zero epoch, (d) MSRA [16] initialized filters are trained on 1000 training samples, and (e) Dictionary initialized filters at zero epoch. For better visualization, only 16 filters are used from the 2^{nd} layer.

gradient descent (SGD) [28]. For input $\mathbf{X}$ and convolutional filter $\mathbf{W}$, the convolutional function of the CNN can be defined as $f(\mathbf{X}, \mathbf{W}, b) = \mathbf{X} * \mathbf{W} + b$, where $*$ is the convolutional operation and b is the bias. A CNN architecture is designed by stacking multiple convolutional and pooling layers. These deep CNN architectures are trained in two passes: 1) forward pass and 2) backward pass. In the forward pass, network propagates the input signal to the last classification layer. In backward pass, the error δ_j^l for each layer l on node j is computed with respect to the cost and the weights of the CNN filters are updated accordingly.

Let $\mathbf{a}^l$ be the output feature map at l^{th} layer of the CNN with a cost function C. The weights are updated as per the gradient direction, i.e. $\Delta\mathbf{W}^l = \frac{\partial C}{\partial \mathbf{W}^l}$. Using chain rule, $\Delta\mathbf{W}^l = \mathbf{a}^{l-1}\delta^l$. In traditional CNN learning, the weights are initialized in different ways such as Xavier [13], or MSRA [16] approach and even randomly. In this research, we propose initialization of the CNN filters using dictionary learned filters as discussed above. As shown in Figure 4, filters learned from the dictionary learning technique show more "structure" than traditional approaches, particularly with small training data. While dictionary initialization helps in finding improved features, updating the filters in a traditional manner still requires large parameter space, which is not conducive for small training data. In the next subsection, we present the proposed approach of incorporating strength of the filters and not update the filters using SGD which reduces the number of learning parameters significantly.

2.2. Learning Filter Strength

The proposed concept of learning strength of the filter is illustrated in Figure 5. Here, we freeze the values of

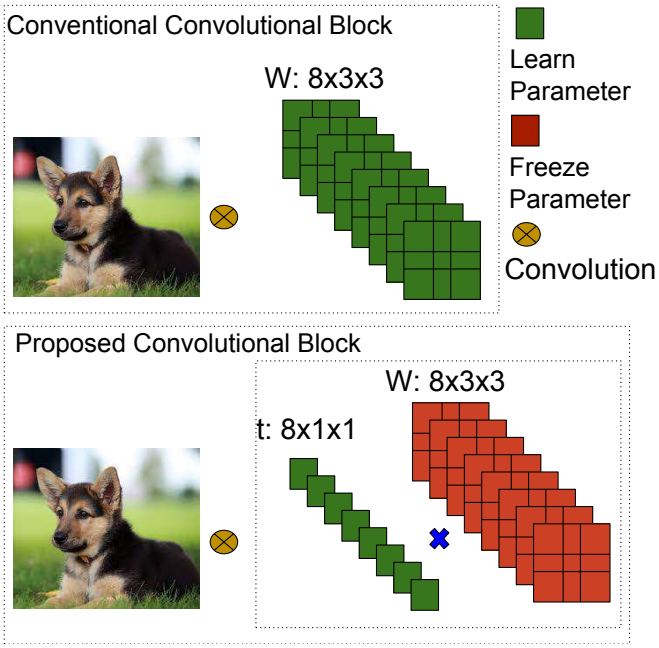

Figure 5. Illustrating the concept of learning the strength of a filter which significantly reduces the number of training parameters.

filters obtained from dictionary learning technique and update only the strength of the filter. As shown in Figure 5, this significantly reduces the number of learning parameters. For l^{th} layer, the strength parameter '$\mathbf{t}^l$' is learned using stochastic gradient descent method; i.e. a scalar value is learned rather than learning the complete filter. The proposed process can be written as,

$$f(\mathbf{X}, \mathbf{W}, b, \mathbf{t}) = \mathbf{X} * (\mathbf{t} \odot \mathbf{W}) + b \tag{3}$$

where, $(\mathbf{t} \odot \mathbf{W})$ represents element-wise multiplication. The pre-trained filters learned from dictionary learning or pre-trained model are selected and the only variable to be learned is $\mathbf{t}$ which can be learned using SGD. Since $|\mathbf{W}| >> |\mathbf{t}|$, even small training data can be used to train the network. In literature, various regularization techniques have been utilized for better convergence. Existing regularization techniques such as dropconnect and ℓ_1 regularization can also be used while learning $\mathbf{t}$.

3. Experimental Results

The effectiveness of the proposed algorithm is evaluated on multiple databases with state-of-the-art CNN architectures including ResNet [17] and DenseNet [18]. The details of experiments and results are described below.

3.1. Database and Experimental Protocol

Since the proposed architecture is for small size training data, the experiments are performed with varying train-

Table 1. Experimental protocols for MNIST, CIFAR-10 and NORB databases.

Databases	Small Training Data	Standard Training	Standard Testing
MNIST	$100:100:1k;1k:1k:5k$	50k	10k
CIFAR-10	$100:100:1k;1k:1k:5k$	40k	10k
NORB	$100:100:1k;1k:1k:5k$	20k	24.3k

ing sizes on three databases: MNIST [26], CIFAR10 [22], and NORB [27]. More specifically, as shown in Table 1, the experiments are performed with 14 data sizes, $100, 200, \cdots, 1000, 2000, \cdots, 5000$. The proposed algorithm is also tested with the complete/standard training set. Further, experiments are performed on an interesting and small sample size problem of newborn face recognition [3]. The newborn database has images from 96 babies and as per the predefined protocol [3], training data consists of images from 10 newborns and the remaining images, corresponding to 86 newborns, are used for testing (with 1, 2, 3, and 4 images per subject in the gallery). Finally, experiments are also performed on the Omniglot database [24] which comprises 1623 handwritten characters pertaining to 50 different alphabets. The background database has 30 alphabets and evaluation set has 20 alphabets. All the experiments are performed with five fold cross validation and average accuracies are reported in next subsections.

3.2. Implementation Details

To demonstrate the results of the proposed SSF-CNN, a popular ResNet [17] architecture is used. Figure 6 illustrates the ResNet architecture which has 1 input layer, 31 convolutional layers, 1 global pooling layer, and 1 softmax layer. The strength parameter is regularized with both ElasticNet [50] ($\lambda_1|\mathbf{t}|_1 + \lambda_2|\mathbf{t}|_2$) and DropConnect [43]. It is experimentally observed that in the first 20 epochs, λ_2 is 0.0001 and λ_1 is 0. After 20 epochs both the regularization constants are set to 0.0001. ℓ_1 regularization introduces sparsity in $\mathbf{t}$ parameters and helps to fadeout the less contributing filters thus improving the strength of filters with large contribution. Further, at every epoch, dropconnect parameter is randomly initialized by $Bernoulli(pr)$ where pr has 0.8 and 0.2 probability for generating $1s$ and $0s$ respectively.

The proposed model utilizes a dictionary and pre-trained model to initialize and train the CNN filters. Specifically, dictionary filters are learned using K-SVD algorithm 1. These dictionaries are layered in a similar manner as CNN layers and are referred to as hierarchical dictionary. The parameter values for K-SVD such as sparsity parameter, the total number of iteration, and batch size for dictionary have been initialized with 0.1, 1000, and 100 respectively. The input signal for dictionary are patches extracted from randomly selected N number of balanced training samples. The value of N varies from

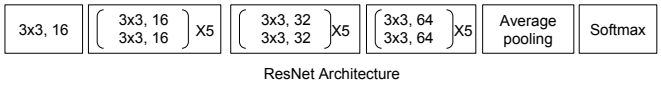

Figure 6. Illustrating the ResNet architecture used in the experiments.

$100, 200, \cdots, 1000, 2000, \cdots, 5000$, as shown in Table 1.

3.3. Parameter Learning

In traditional ResNet architecture, total number of parameters to be learned in convolutional layers for the CIFAR-10 dataset is $242,352$. On the other hand, in the proposed SSF-CNN, total number of strength parameters to be learned for the same database is $26,928$. This shows that the proposed architecture reduces the total number of parameters to be learned by $1/9^{th}$ factor in each convolutional layer. Similarly, for other databases and architectures, we observe reduced number of parameters to train.

3.4. Results on Limited Training Data - MNIST, CIFAR-10, and NORB

The main focus of the proposed SSF-CNN is to learn the deep neural network models with a small number of training samples. Since the proposed initialization is performed using dictionary learning, we also compute the results of shallow dictionary which serves as the baseline for all the experiments. We have also compared the proposed algorithm with PCANet [4], Deep Hybrid Network [35], ScatNet [1], ResNet initialized with Xavier [13], and ResNet initialized with MSRA [16]. For the proposed SSF-CNN, two sets of results are computed based on the manner in which the parameters $\mathbf{W}$ and $\mathbf{t}$ are learned.

- **Experiment 1** - *Learn* $\mathbf{W}$: Initialized filters are fine-tuned while doing backpropagation.
- **Experiment 2** - *Learn* $\mathbf{t}$, *Freeze* $\mathbf{W}$: Only the strength parameter $\mathbf{t}$ is learned while the initialized filters are not updated.

Filter Visualization: We first analyze the filters learned from the proposed method and CNN. Figure 4 shows the first and second layer filters trained on CIFAR-10 database: (a) & (c) showcase filters with two existing initialization techniques in CNN architecture, (b) & (d) trained CNN filters on 1000 training samples, and (e) trained dictionary filters on 1000 training samples. In Figure 4, it can be observed that dictionary trained filters have less noisy patterns compare to CNN trained filters on small data. In literature, Zeiler and Fergus [48] have also suggested that the filters that have structural properties are good while the ones with noisy, correlated, and unstructured pattern are bad. This visualization illustrates that the proposed SSF-CNN utilizes

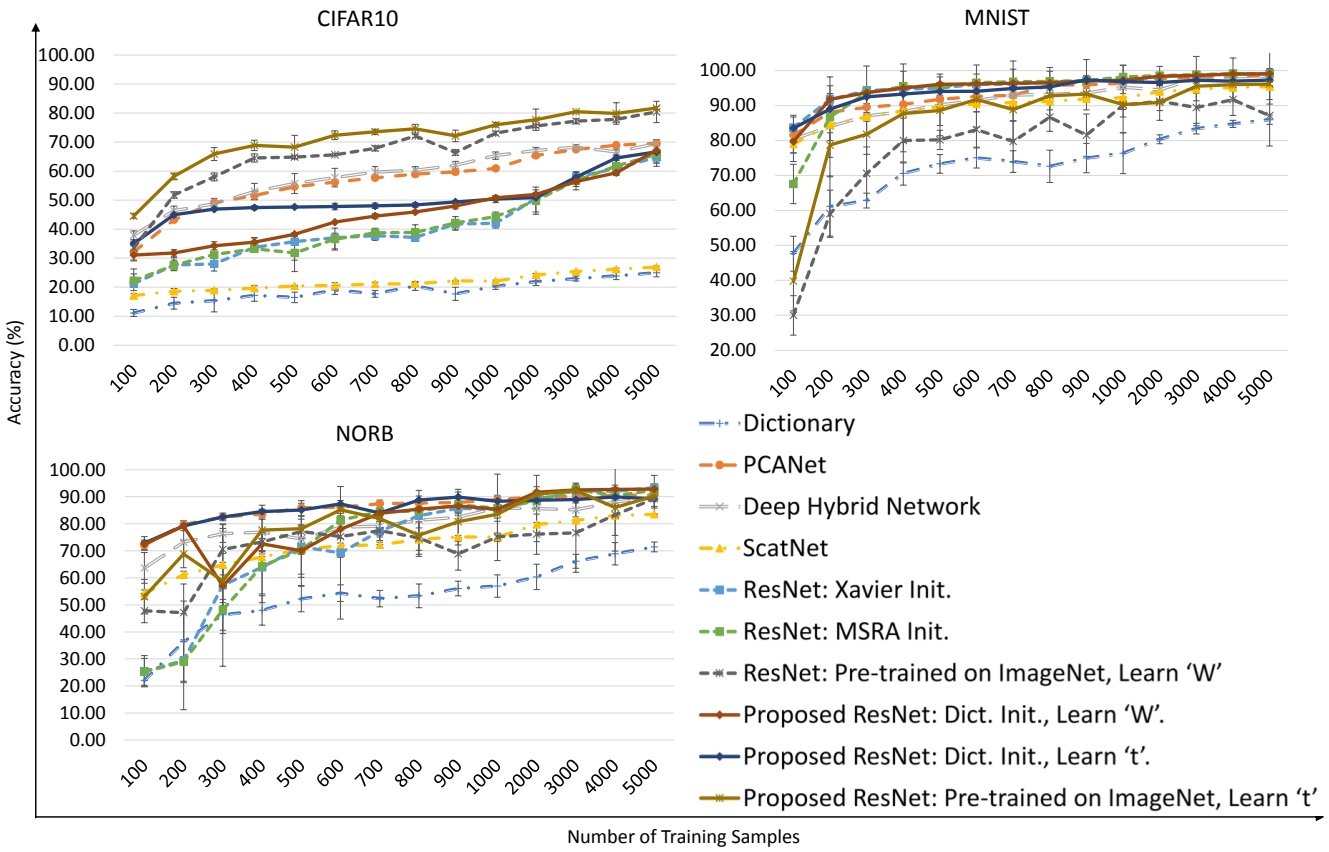

Figure 7. Classification accuracies (%) for CIFAR-10, MNIST, and NORB databases with varying the number of training samples.

good filters. We next support these assertions with experimental results.

Performance with Shallow Dictionary: To analyze the performance of the proposed method with varying training data sizes, 14 subsets of the training data of size $100, 200, \cdots, 1000, 2000, \cdots, 5000$, are created. These sets are used to train the dictionary and SSF-CNN on each of the three databases individually. To train shallow dictionary for each database, 50 atoms are initialized and trained with varying number of training samples. The trained dictionary is then utilized to compute sparse features for training and testing samples. These features are input to a 3 layer neural network with 2 hidden layers of size $\{40, 20\}$. The results of shallow dictionary learning on three object classification databases are reported in Figure 7. From these results, it can be inferred that shallow dictionary learning might not require large training data and increasing data may not lead to large improvement in classification results. This figure also shows that shallow dictionary learning may not be able to yield high classification accuracy and deep CNN architectures may further help.

Performance with SSF-CNN and Comparison with Existing Algorithms: We next evaluate the performance of the proposed SSF-CNN on three object classification databases by varying the training data size. The results in Figure 7 show that, in general, Xavier and MSRA initialization yield lower performance compared to the proposed dictionary initialization for very small training data. It can be consistently observed that the differences in results are more profound when the strength parameter $\mathbf{t}$ is learned with fixed $\mathbf{W}$. The results further show that the performance of the proposed SSF-CNN increases with increase in training database size. It can be inferred that unlike shallow dictionary, where the performance does not improve significantly with increase in training database size, the parameters learned by the proposed SSF-CNN evolves with large data.

We also observe that the proposed algorithm, in general yields higher performance compared to three existing algorithms, PCANet [4], Deep Hybrid Network [35], and ScatNet [1]. We next perform the experiments when the structure of the filters are obtained from training on ImageNet data and then strength parameter is used to adapt to small sample size problem (i.e. Proposed ResNet: Pretrained on ImageNet, Learn $\mathbf{t}$). Results in Figure 7 show that our hy-

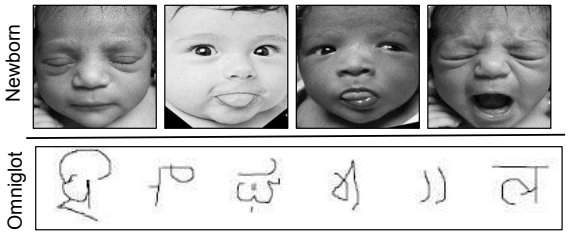

Figure 8. Samples images from the Omniglot and Newborn Faces databases.

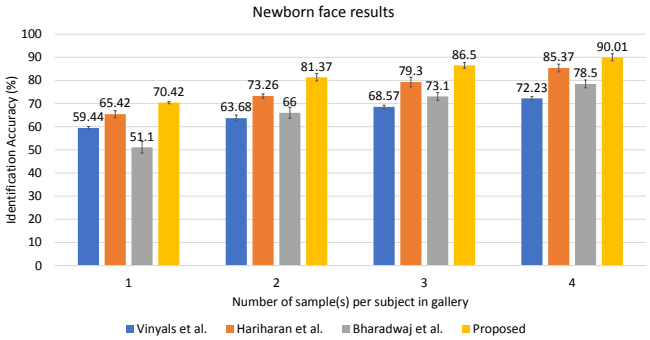

Figure 9. Summarizing the results on the newborn face database.

pothesis that the structure of filters can be learned from training on large databases and knowledge can be *adapted* with small training data using the strength $\mathbf{t}$ is valid.

Results on Complete Training Data: We have also evaluated the proposed dictionary learning based initialization method on the standard training protocols of all three databases, i.e., using the complete training data. Similar to small training data size, the experiments are performed with multiple methods of initializations and two ways of learning $\mathbf{W}$ and $\mathbf{t}$, i.e., (i) *learn* $\mathbf{W}$ and (ii) *learn* $\mathbf{t}$, *freeze* $\mathbf{W}$. In this experiment, the proposed dictionary learning based initialization for ResNet is compared with Xavier and MSRA initialization. On the MNIST database, the proposed initialization yields an accuracy of 99.70% which is comparable with 99.71% achieved by standard initialization. On the NORB database, the proposed approach yields at least 3.8% higher classification accuracy compared to existing initialization approaches. It is also observed that even if the filters have random values, learning strength produces considerably high accuracies. Once the filters are trained, optimizing the strength of those filters can further improve the performance.

3.5. Small Sample Size Case Studies

To showcase the effectiveness of the proposed *structure* and *strength* concept on small sample size databases, we present two case studies (i) newborn face database [3] and (ii) Omniglot database [24]. Figure 8 shows sample images from both the databases.

Newborn Face Recognition: Bharadwaj *et al.* [3] have shown that newborn face recognition is a challenging small sample size application. The publicly available IIITD Newborn database [3] contains face images from 96 newborns. The pre-defined protocol limits us to use training samples from only 10 newborns and testing is performed with 86 newborns. We compute the performance of ResNet architecture where the proposed dictionary based initialization helps in estimating the structure using images from 10 newborns and then strength parameter is used to attune the filters. The observed rank-1 accuracy in this case is 36.32% which is at least 0.5% better than pre-trained ResNet architecture (which is traditionally fine-tuned with newborn training data). Also, when we use training images of only 10 newborns to train filter of CNN models from scratch, the test accuracies are extremely low.

As discussed before, we can learn "structure" from large domain-specific data and then the proposed "strength" can help attune the filters for problem-specific data. Therefore, we perform experiments with pre-trained networks (pre-trained filters are obtained after learning from either ImageNet or Labeled Faces in the Wild dataset (LFW) [19] and YouTube Faces (YTF) [44] databases) and use strength parameter to attune it for newborn face recognition based on training data of 10 newborns. For this experiment, as shown in Table 2, we use variants of ResNet [17], VGG [38], VGGFace [36], LightCNN [46], and DenseNet [18] architectures, and the performance is compared with standard fine-tuning approaches using same images from 10 newborns. As shown in Table 2, we have observed that learning strength of the filters improves the performance of CNN models compared to conventional fine-tuning approach. With single gallery image per subject, the best rank-1 accuracy of over 70% is obtained when the proposed strength parameter is used with pre-trained VGG-Face [36] which is at least 10% better than the conventional fine-tuning based approach. This shows that in real-world applications, the concept of learning structure and strength helps in achieving improved performance.

The performance of the proposed approach is also compared with deep hybrid network [35] and ScatNet [1]. For one gallery per subject, the rank-1 accuracies of these two algorithms are $25.18\% \pm 1.33\%$ and $31.04\% \pm 1.94\%$ respectively, which are at least 39% less than the best results reported in Table 2. Finally, we also compare the performance of the proposed algorithm with the Vinyals *et al.* [42], Hariharan *et al.* [15], and Bharadwaj *et al.* [3] on newborn face database. Using the same protocol, Figure 9 illustrates the comparison between the proposed method (best reported result in Table 2) with existing methods. The proposed method improves the rank-1 accuracies by

Table 2. Rank-1 identification accuracies (%) on the newborn face database [3]. The results are reported for fine-tuned pre-trained models and with learning the strength of pre-trained filters. The last three models are trained on face databases and the remaining models are trained on ImageNet [6] database.

Pre-trained Model	Number of Gallery Images							
	Fine-tuning				Proposed Strength Learning			
	1	2	3	4	1	2	3	4
ResNet 50	35.77 ± 2.34	43.59± 0.92	49.90 ± 2.57	52.14 ± 3.31	37.80 ± 2.01	46.77± 1.79	52.61 ± 1.89	56.73 ± 1.79
ResNet 101	35.86 ± 2.78	45.90± 2.54	51.17± 2.10	54.59 ± 3.41	36.62 ± 4.06	46.71 ± 3.73	52.79 ± 1.72	56.16 ± 3.07
ResNet152	36.30 ± 3.19	46.74 ± 2.42	51.99 ± 2.24	55.47 ± 2.34	38.30 ± 3.57	47.92 ± 2.29	53.71 ± 2.62	59.57 ± 2.46
VGG13	56.34 ± 2.46	68.49 ± 3.07	73.37 ± 2.53	76.47 ± 2.33	65.54 ± 3.20	78.14 ±1.97	84.05 ± 1.40	87.76 ±1.88
VGG16	57.07 ± 2.85	67.84 ± 2.61	73.21± 3.10	76.21 ± 2.86	65.29 ± 1.99	79.18 ± 2.85	84.24 ± 2.82	87.50 ± 1.47
VGG19	53.87 ± 4.49	66.95 ± 2.15	72.33± 1.25	75.75 ± 1.77	62.29 ±1.70	75.36 ± 2.03	80.90 ±0.77	84.20 ± 0.75
DenseNet161	50.64 ± 3.27	63.65 ± 2.95	68.98± 1.79	72.86 ±1.82	58.39 ±5.59	72.14 ±1.82	77.36 ± 1.57	81.04 ± 1.40
DenseNet169	54.15± 4.33	68.91± 2.99	73.31± 1.72	72.97 ± 2.05	58.25 ± 1.68	73.10 ±0.99	78.91 ± 1.02	83.31 ± 1.12
DenseNet201	60.78 ±2.00	71.19 ± 0.84	71.48±2.17	73.64 ±1.39	61.45 ± 5.09	74.58 ±2.40	80.75 ± 3.86	85.02 ±3.98
LightCNN-9	55.72 ±2.90	66.09 ±2.27	67.65 ±2.29	71.81 ±1.64	56.48 ± 4.60	69.82 ±4.49	76.91±3.69	81.87± 3.93
LightCNN-29	53.10 ±3.75	65.28 ±2.47	71.91±1.99	75.85 ±2.02	62.67 ±2.59	76.19 ±1.15	82.55 ± 0.87	86.00 ± 1.03
VGG-Face	60.77 ±1.28	72.93 ±1.40	77.19 ±1.27	79.66 ±1.97	**70.42 ±0.50**	**81.37 ±1.59**	**86.50 ±1.20**	**90.01±1.53**

$11 - 19\%$ for varying number of sample(s) per subject. However, the proposed algorithm consistently yields improved accuracies and is approximately 4.5% better than the second best performing approach [15].

Omniglot Database: On the Omniglot database [24], SSF-CNN yields classification accuracies of $97.6\% \pm 0.84\%$ and $98.3\% \pm 1.03\%$ for 1-shot, 5-way and 5-shot, 5-way, respectively which are comparable to state of the art results. Table 3 summarizes the results of the proposed algorithm and compares them with existing algorithms. The results show that SSF-CNN is among the top performing algorithms for both the protocols.

Table 3. Classification results (%) on the Omniglot database [24].

Algorithm	1-shot, 5-way	5-shot, 5-way
Santoro *et al.* [37]	82.8	94.9
Koch *et al.* [21]	97.3	98.4
Vinyals *et al.* [42]	98.1	98.9
Proposed	97.6	98.3

4. Discussion and Conclusion

Large training database is a key requirement for training convolutional neural networks. However, there are several applications and problem statements that do not have the luxury of large training databases. In this research, we propose Structure and Strength Filtered CNN as a framework for learning a CNN model with small training databases. We propose to initialize the filters of CNN using dictionary filters which can be trained with small training samples. Since the dictionary atoms are learned for reconstruction, they may not be optimal for classification. Therefore, we next suggest to learn the strength of the filters with the given training data. The effectiveness of the proposed model has been demonstrated on multiple object classification databases and a real-world newborn face recognition problem. Using different architectures and experiments, we demonstrate the efficacy of the proposed approach. Specifically, in case of newborn face recognition, remarkable improvement in accuracy is achieved with the proposed approach. The proposed CNN has the flexibility to work for small as well as large databases. The current model incorporates unsupervised dictionary filters to initialize the CNN network. As a future work, other trained filters such as supervised dictionary filters can also be used. They can also be used to adapt the filters from one task to another task while learning only the strength of the filters. The proposed algorithm can also be extended to other applications such as face recognition with variations in disguise [7], matching faces in videos [14], and sketch to photo matching [33].

5. Acknowledgment

This research is partially supported by Ministry of Electronics and Information Technology, India. Rohit Keshari is partially supported by Visvesvaraya Ph.D. fellowship. Richa Singh and Mayank Vatsa are partly supported by the Infosys Center of Artificial Intelligence, IIIT Delhi, India.

References

[1] J. Andén and S. Mallat. Multiscale scattering for audio classification. In *ISMIR*, pages 657–662, 2011. 2, 5, 6, 7

[2] S. Bharadwaj, H. S. Bhatt, R. Singh, M. Vatsa, and S. K. Singh. Face recognition for newborns: A preliminary study. In *IEEE BTAS*, pages 1–6, 2010. 1

[3] S. Bharadwaj, H. S. Bhatt, M. Vatsa, and R. Singh. Domain specific learning for newborn face recognition. *IEEE TIFS*, 11(7):1630–1641, 2016. 1, 2, 5, 7, 8

[4] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. PCANet: A simple deep learning baseline for image classification? *IEEE TIP*, 24(12):5017–5032, 2015. 2, 5, 6

[5] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 8

[7] T. I. Dhamecha, A. Nigam, R. Singh, and M. Vatsa. Disguise detection and face recognition in visible and thermal spectrums. In *ICB*, pages 1–8, 2013. 8

[8] K. Engan, S. O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *ICASSP*, pages 2443–2446. IEEE Computer Society, 1999. 3

[9] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *JMLR*, 11(Feb):625–660, 2010. 1

[10] J. Feng and T. Darrell. Learning the structure of deep convolutional networks. In *ICCV*, pages 2749–2757, 2015. 2

[11] Z. Feng, L. Jin, D. Tao, and S. Huang. Dlanet: A manifold-learning-based discriminative feature learning network for scene classification. *Neurocomputing*, 157:11–21, 2015. 2

[12] Y. Gan, J. Liu, J. Dong, and G. Zhong. A PCA-based convolutional network. *arXiv preprint arXiv:1505.03703*, 2015. 2

[13] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9, pages 249–256, 2010. 4, 5

[14] G. Goswami, R. Bhardwaj, R. Singh, and M. Vatsa. MDLFace: Memorability augmented deep learning for video face recognition. In *IEEE IJCB*, pages 1–7, 2014. 8

[15] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. 7, 8

[16] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 4, 5

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016. 1, 3, 4, 5, 7

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 1, 4, 7

[19] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, 07-49, University of Massachusetts, Amherst, 2007. 7

[20] K. Jarrett, K. Kavukcuoglu, Y. LeCun, et al. What is the best multi-stage architecture for object recognition? In *IEEE ICCV*, pages 2146–2153, 2009. 2

[21] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 8

[22] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, 2009. 2, 5

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[24] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33, 2011. 2, 5, 7, 8

[25] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 1

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2, 5

[27] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE CVPR*, volume 2, pages II–104, 2004. 2, 5

[28] Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the connectionist models summer school*, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988. 4

[29] D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999. 3

[30] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11(Jan):19–60, 2010. 3

[31] R. Mao, H. Zhu, L. Zhang, and A. Chen. A new method to assist small data set neural network learning. In *ISDA*, volume 1, pages 17–22. IEEE, 2006. 2

[32] D. Mishkin and J. Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015. 1

[33] S. Nagpal, M. Singh, R. Singh, M. Vatsa, A. Noore, and A. Majumdar. Face sketch matching via coupled deep transform learning. In *IEEE ICCV*, pages 5429–5438, 2017. 8

[34] J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, and A. Y. Ng. Tiled convolutional neural networks. In *NIPS*, pages 1279–1287, 2010. 2

[35] E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. *arXiv preprint arXiv:1703.08961*, 2017. 2, 5, 6, 7

[36] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015. 7

[37] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016. 8

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 7

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1

[40] S. Tariyal, A. Majumdar, R. Singh, and M. Vatsa. Deep dictionary learning. *IEEE Access*, 4:10096–10109, 2016. 2, 3

[41] I. Tosic and P. Frossard. Dictionary learning. *IEEE SPM*, 28(2):27–38, 2011. 2, 3

[42] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016. 7, 8

[43] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *ICML*, pages 1058–1066, 2013. 5

[44] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, pages 529–534. IEEE, 2011. 7

[45] D. Wu, J. Wu, R. Zeng, L. Jiang, L. Senhadji, and H. Shu. Kernel principal component analysis network for image classification. *arXiv preprint arXiv:1512.06337*, 2015. 2

[46] X. Wu, R. He, Z. Sun, and T. Tan. A light CNN for deep face representation with noisy labels. *arXiv preprint arXiv:1511.02683*, 2015. 7

[47] W. Xiong, B. Du, L. Zhang, R. Hu, and D. Tao. Regularizing deep convolutional neural networks with a structured decorrelation constraint. In *ICDM, 519–528*. IEEE, 2016. 2

[48] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV, 818–833*. Springer, 2014. 5

[49] R. Zeng, J. Wu, L. Senhadji, and H. Shu. Tensor object classification via multilinear discriminant analysis network. In *IEEE ICASSP*, 2015. 2

[50] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. 5