

SCATTERNET HYBRID DEEP LEARNING (SHDL) NETWORK FOR OBJECT CLASSIFICATION

Amarjot Singh, Nick Kingsbury

Signal Processing Group, Department of Engineering, University of Cambridge, U.K.

ABSTRACT

The paper proposes the ScatterNet Hybrid Deep Learning (SHDL) network that extracts invariant and discriminative image representations for object recognition. SHDL framework is constructed with a multi-layer ScatterNet front-end, an unsupervised learning middle, and a supervised learning back-end module. Each layer of the SHDL network is automatically designed as an explicit optimization problem leading to an optimal deep learning architecture with improved computational performance as compared to the more usual deep network architectures. SHDL network produces the state-of-the-art classification performance against unsupervised and semi-supervised learning (GANs) on two image datasets. Advantages of the SHDL network over supervised methods (NIN, VGG) are also demonstrated with experiments performed on training datasets of reduced size.

Index Terms— ScatterNet, Deep architecture design, Unsupervised learning, Convolutional neural network.

1. INTRODUCTION

Object classification is challenging due to the large intra-class variability, arising from translation and rotation of objects, lighting, deformations and occlusions. Researchers have relied on invariant and discriminative class-specific image representations to tackle this problem [1].

Numerous attempts have been made to design learning architectures that capture the necessary image representations for object classification. These methods include architectures that: (i) encode handcrafted features extracted from the input images into rich *non-hierarchical* representations [2]; (ii) learn multiple levels of *feature hierarchies*, directly from the input data [1]; (iii) make use of the ideas from both the above-mentioned categories to extract *feature hierarchies* from *hand-crafted features*.

Bag of Words (BoW) [3] models represent the first class of architectures that encode handcrafted bag-of-visual-words (BOV) descriptors into rich feature representations using unsupervised coding and pooling [2]. The pipeline was improved by encoding the local descriptor patches by a set of visual codewords with sparse coding with a linear SPM kernel [4]. This class of methods are very easy to design and

cheap to evaluate but achieve only marginally good classification performance on different benchmarks [4].

The second category includes architectures such as Convolutional Neural Networks (CNNs) [5, 6], Deep Belief Networks [7] etc that learn feature hierarchies directly from the input images. These networks have achieved state-of-the-art classification performance on various datasets [5, 6], but despite the success of these networks, their design and optimal configuration is not well understood which makes it difficult to develop them. In addition, these models produce a large number of coefficients that are learned with the help of powerful computational resources and require large training datasets which may not be available for many applications such as stock market prediction [8], medical imaging [9] etc.

The third class of models combines the concepts from both the above-mentioned models to learn feature hierarchies from low-level hand-crafted descriptors [10]. Hierarchical max (HMAX) [11] is one such model that uses an RBF kernel to learn a single layer of high-level features from descriptors captured with a battery of Gabor filters. He et al. [1] learned three layers of sparse hierarchical features from SIFT descriptors using unsupervised learning. Sivic et al. [10], discovered object class hierarchies from visual codewords using hierarchical LDA. This class of models has produced promising performance on various datasets [12, 13]. In addition, each layer of these models can be posed as an optimization problem resulting in optimal architectures [1].

The paper introduces the ScatterNet Hybrid Deep Learning (SHDL) network for object classification. This framework first extracts ScatterNet handcrafted descriptors which are used by an unsupervised learning module to learn hierarchical features that capture intricate structure between different object classes. Supervised learning then selects the features specific to each object class, from the feature hierarchies, which are finally used for classification. The term 'Hybrid' is coined because the framework uses both unsupervised as well as supervised learning. Each layer of the network is designed and optimized automatically that produces the desired computationally efficient architectures.

The contributions of the paper are as follows:

- *Hand-crafted Module*: The hand-crafted descriptors are extracted with the two-layer parametric log ScatterNet [14], instead of BOW [3] or SIFT [13] descriptors.

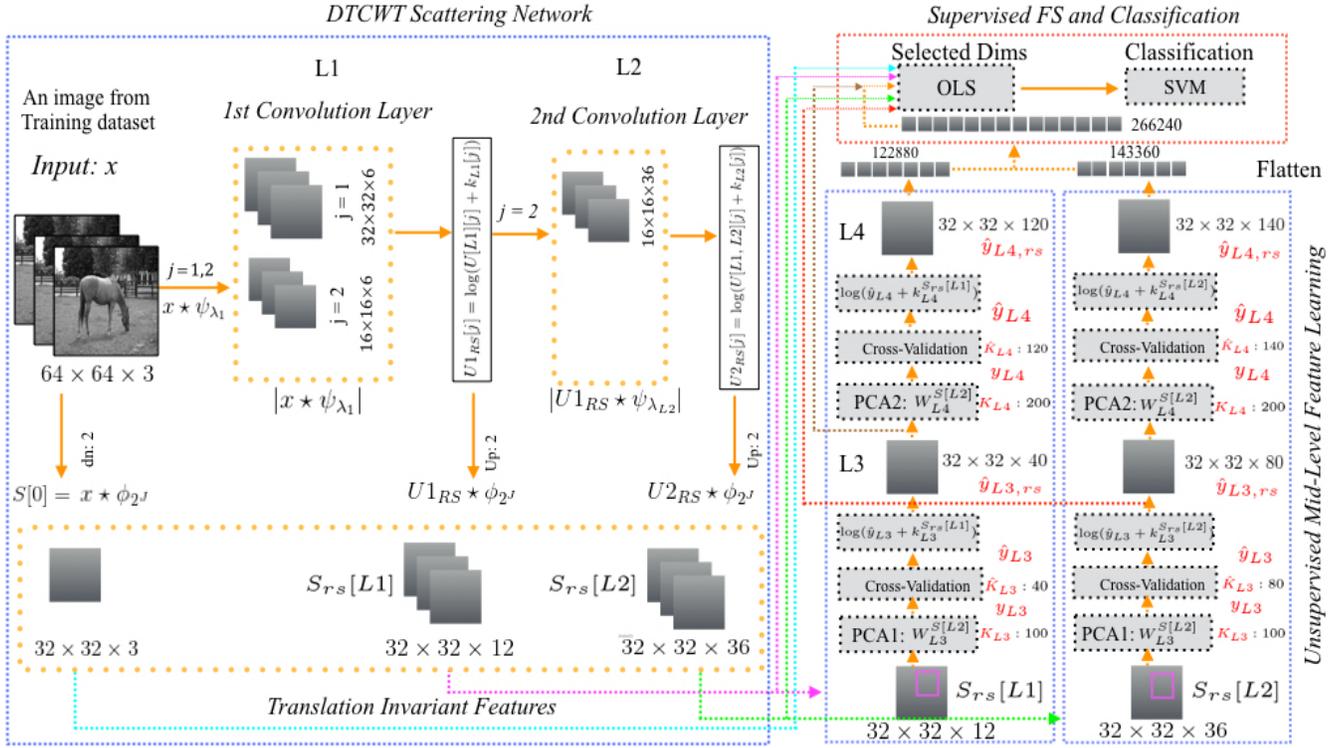


Fig. 1. SHDL: The illustration shows the input image ($64 \times 64 (x)$) from the CIFAR-10 dataset at resolution R1 decomposed to extract the translation invariant relatively symmetric coefficients at L0 ($S_{rs}[L0]$), L1 ($S_{rs}[L1]$) and L2 ($S_{rs}[L2]$). Features at the higher level of abstraction are captured at L3 and L4 layers of the PCA-Net using unsupervised learning. Parametric log transformation is applied on the output of each PCA stage to introduce relative symmetry. The representations extracts at each stage (L0, L1, L2, L3, L4) are concatenated and given to the supervised OLS layer that select the object-specific features finally used for classification using the Gaussian SVM (G-SVM).

This extracts symmetrically distributed multiscale oriented edge features at the first layer and additional discriminative sparse features at the second layer [15].

- **Unsupervised Learning Module:** This module uses two stacked PCA-Net [16] layers with parametric log non-linearity to learn robust symmetrically distributed hierarchical mid-level features across object classes. The network is fast to train as opposed to other unsupervised learning modules (autoencoders or RBMs) as the minimization of the loss function (Eq. 8) can be obtained in its simplistic form as the Eigen decomposition.
- **Supervised Learning Module:** OLS layer [17, 14] is applied to the concatenated features obtained from the layers to select a subset of object-class-specific features, without undesired bias from outliers, due to the introduced symmetry. The selected features are fed into a Gaussian-kernel support vector machine (G-SVM) to perform object classification.
- **Network Layer Optimization:** The number of filters in each layer of the unsupervised learning module are optimized as part of the automated design process. The optimization of the number of filters in a layer leads to the efficient learning of the subsequent layer as the

filters are now learned from a smaller feature space. The reduced feature space subsequently also makes the learning of the OLS and SVM efficient.

The classification performance of the proposed architecture is tested on CIFAR-10 and Caltech-101 datasets. Multiple experiments on different training dataset sizes are performed to highlight the advantages of the proposed network against supervised and unsupervised methods.

Section 2 of the paper briefly presents the proposed SHDL network. Section 3 presents the experimental results while Section 4 draws conclusions.

2. SHDL NETWORK

This section introduces the proposed ScatterNet Hybrid Deep Learning (SHDL) network (Fig. 1) with the detailed mathematical formulation of each module and a description of the representations captured by them.

2.1. Hand-crafted Module: ScatterNet

Handcrafted descriptors are extracted using the parametric log based two-level Dual-Tree Complex Wavelet Transform (DTCWT) ScatterNet [14] that extracts relatively symmetric

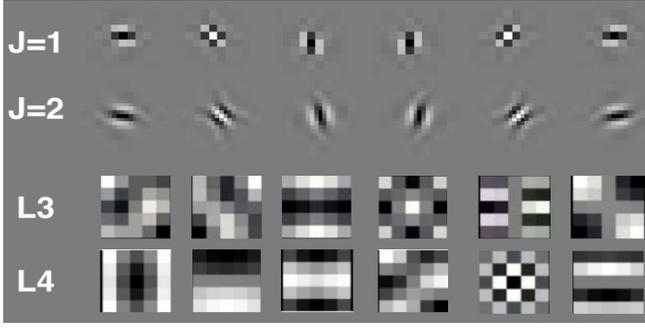


Fig. 2. Illustration shows the DTCWT real filters at two scales used at Layer L1 and L2. The filters learned by the PCA-Net at L3 and L4 stage are also shown.

translation invariant low-level features at the first layer and more discriminative sparse features at second layer [15, 18, 19]. The extracted features are also dense over the scale as they are obtained by decomposing multi-resolution images obtained at 1.5 times (R1) and twice (R2) the size of the input image. This ScatterNet [14] is chosen over Bruna and Mallat’s [15] due to its superior classification accuracy and computational efficiency. The DTCWT ScatterNet formulation is presented for an input signal (x) which may then be applied to each multi-resolution image.

The features at the first layer are obtained by filtering the input signal x with dual-tree complex wavelets $\psi_{j,r}$ at different scales (j) and six pre-defined orientations (r) fixed to $15^\circ, 45^\circ, 75^\circ, 105^\circ, 135^\circ$ and 165° , as shown in Fig. 2. A more translation invariant representation is built by applying a point-wise L_2 non-linearity (complex modulus) to the real and imaginary part of the filtered signal:

$$U[L1] = \sqrt{|x \star \psi_{\lambda_1}^a|^2 + |x \star \psi_{\lambda_1}^b|^2} \quad (1)$$

The parametric log transformation layer is applied on the oriented features, extracted at the first scale $j = 1$ with a parameter $k_{L1}[j]$, to reduce the effect of outliers by introducing relative symmetry (rs) to their amplitude distribution (as shown in Fig. 3):

$$U_{1rs}[j] = \log(U[L1][j] + k_{L1}[j]), \quad U[L1][j] = |x \star \psi_j|, \quad (2)$$

The parameter k_{L1} is selected such that it minimizes the difference between the mean and median of the distribution [14]. Next, a local average is computed on the envelope $|U[\lambda_{L1}]|$ that aggregates the coefficients to generate the desired translation-invariant representation:

$$S_{rs}[L1] = |U_{1rs}| \star \phi_{2j} \quad (3)$$

The energy (high-frequency components) lost due to smoothing is recovered by cascaded wavelet filtering applied at the second layer [15]. The recovered components are again not translation invariant so invariance is achieved by first applying the L_2 non-linearity to obtain the regular envelope:

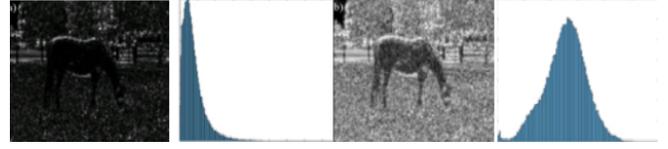


Fig. 3. Illustration shows a representation obtained using the DTCWT ($15^\circ, j=1, R=1$) at L1. The representation is affected by outliers resulting in the skewed distribution. A relatively symmetric representation is obtained using by applying the parametric log transformation that also results in contrast normalization.

$$U[L2] = |U_{1rs} \star \psi_{\lambda_{L2}}| \quad (4)$$

The parametric log transformation is applied again to produce relative symmetry:

$$U_{2rs}[j] = \log(U[L2][j] + k_{L2}[j]) \quad (5)$$

Next, a local-smoothing operator is applied to improve translation invariance::

$$S_{rs}[L2] = U_{2rs} \star \phi_{2j} \quad (6)$$

The output coefficients are typically formed from $x \star \phi$ (Layer 0), $S_{rs}[L1]$ (Layer 1) and $S_{rs}[L2]$ (Layer 2) for each of the two image resolutions R1 and R2.

2.2. Unsupervised Learning Module: PCA-Net Layers

This section details the optimization framework for the stacked PCA-Net [16] layers used to learn symmetrically distributed hierarchical mid-level features at L3 and L4, from invariant features extracted at L1 or L2, as shown in Fig. 1. The mathematics is presented for $S_{rs}[R1, L1]$ (invariant features obtained for R1 resolution image at layer L1). This formulation is also applied to $S_{rs}[R1, L2]$ (features for R1 resolution at layer L2) as well as features extracted at R2 resolution at both layers ($S_{rs}[R2, L1]$ and $S_{rs}[R2, L2]$).

The objective of the PCA layer is to minimize the reconstruction error by learning a family of multi-channel orthonormal filters. In order to learn the filters, M overlapping patches of size $z_1 \times z_2$ are collected from each channel of the input $S[R1, L1]$ i.e., $x_1, x_2, \dots, x_M \in R^{z_1 z_2 \times P}$ where x is the sampled patch, M represents the number of patches and P (12 and 36, Fig. 1) represents the number of channels of the input. After this, the patch mean is subtracted to obtain $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_M]$, where \tilde{x} is a mean-removed patch. Given N training images, we get the unified matrix:

$$X = [\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_N] \in R^{z_1 z_2 M \times PN}. \quad (7)$$

This filters are learned by minimizing the following equation,

$$\min_{W_{L3} \in R^{sL_3 \times L_3 \times P \times KL_3}} \|X - W_{L3} W_{L3}^T X\|_F^2, \text{ s.t. } W_{L3}^T W_{L3} = I_{KL_3}, \quad (8)$$

where W_{L3} are the learned filters at layer L_3 with size $sL_3 \times sL_3 \times P \times KL_3$, where KL_3 represents the number of filters.

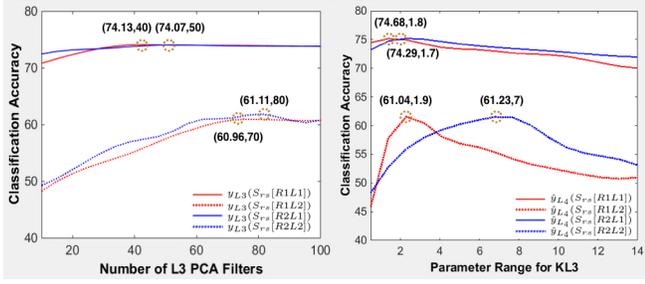


Fig. 4. Illustration shows: (a) 5-CV classification accuracy (y_{L3}) vs. the number of filters (K_{L3}) learned at layer L3 (b) optimal 5-CV classification accuracy (\hat{y}_{L3}) vs. k_{L3} , with fixed number of optimal L3 filters (\hat{K}_{L3}). The optimal filters \hat{K}_{L3} and chosen k_{L3} along with their corresponding accuracies is shown in the graphs.

The solution in its simplified form represents K_{L3} principal eigenvectors of XX^T . These learned filters (Fig. 2) capture the variance in the training dataset in the form of eigenvectors.

The output responses of the $L3$ layer can be obtained as:

$$y_{L3} = S_{rs}[R1, L1] \star W_{L3}^{sL3 \times sL3 \times P \times K_{L3}}, i = 1, 2, 3, \dots, N \quad (9)$$

$S_{rs}[R1, L1]$ is zero-padded before convolving with W_{L3} so as to make y_{L3} have the same size as $S_{rs}[R1, L1]$.

Next, G-SVM is used at the output of the $L3$ layer, with varying number of filters (10,20,..., K_{L3}), to select the optimal number (\hat{K}_{L3}) of learned filters that result in the highest five-fold cross-validation accuracy (5-CV) on the training dataset (Fig. 3). The optimum output at $L3$ layer (\hat{y}_{L3}) is computed using Eq. 10 using the optimum number of filters (\hat{K}_{L3})

As explained in the previous section, parametric log transformation is applied on \hat{y}_{L3} to introduce relative symmetry:

$$\hat{y}_{L3,rs} = \log(\hat{y}_{L3} + k_{L3}) \quad (10)$$

Next, K_{L4} filters with weights W_4 at layer $L4$ can be learned similarly:

$$\min_{W_{L4} \in \mathbb{R}^{sL4 \times sL4 \times K_{L3} \times K_{L4}}} \|X^{L3} - W_{L4}W_{L4}^T X^{L3}\|_F^2, \text{ s.t.} \\ W_{L4}^T W_{L4} = I_{K_{L4}}, \quad (11)$$

where X^{L3} represents the matrix computed by extracting patches from $\hat{y}_{L3,rs}$ ($L3$ output (relatively symmetric (rs)) obtained using the optimal (\hat{K}_{L3}) number of filters). The output response at Layer $L4$ can be computed as shown:

$$y_{L4} = \hat{y}_{L3,rs} \star W_{L4}^{sL4 \times sL4 \times K_{L3} \times K_{L4}}, i = 1, 2, 3, \dots, N \quad (12)$$

Here, $\hat{y}_{L3,rs}$ is also zero padded before applying the convolutions as described above. The optimal $L4$ output (\hat{y}_{L4}) is computed using \hat{K}_{L4} filters, obtained using five-fold cross-validation as shown in Fig. 4. Parametric log transformation is finally applied on \hat{y}_{L4} to introduce relative symmetry:

$$\hat{y}_{L4,rs} = \log(\hat{y}_{L4} + k_{L4}) \quad (13)$$

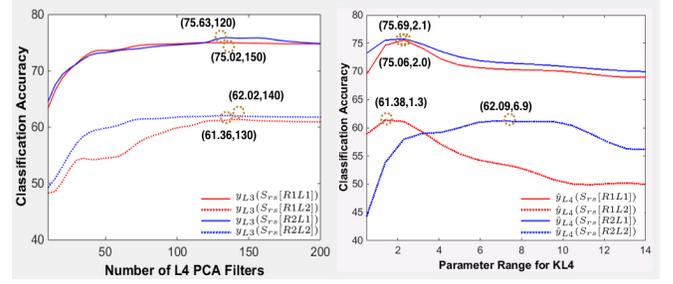


Fig. 5. Illustration shows: (a) 5-CV classification accuracy (y_{L4}) vs. the number of filters (K_{L4}) learned at layer L4 (b) optimal 5-CV classification accuracy (\hat{y}_{L4}) vs. k_{L4} , with fixed number of optimal L4 filters (\hat{K}_{L4}). The optimal filters \hat{K}_{L4} and chosen k_{L4} along with their corresponding accuracies is shown in the graphs.

2.3. Supervised Learning Module: OLS and G-SVM

The features obtained from each layer of the network ($L0$, $L1$, $L2$, $L3$, $L4$) for both R1 and R2 images are concatenated, normalized across each dimension and fed to the OLS as shown in Fig. 1. Orthogonal least square (OLS) regression [17] selects discriminative features specific to class C in a supervised way using a one-versus-all linear regression. The regression is applied to the training set of scattering features where each vector of N (Cifar: $N \approx 176000$, Caltech: $N \approx 474000$) dimensions is reduced to N' (Cifar: $N' \approx 10300$, Caltech: $N' \approx 21000$) selected dimensions. The reduced training feature dataset is utilized by the G-SVM to learn weights that best discriminate the classes in the dataset. Feature selection results in limited dimensions that lead to a more efficient training of the G-SVM and improves generalization.

3. OVERVIEW OF RESULTS

The performance of the SHDL network is evaluated on CIFAR-10 and Caltech-101 datasets. CIFAR-10 contains a total of 50000 training and 10000 test images each of size 32×32 . The Caltech-101 dataset is an unbalanced image dataset with images of different sizes. In these experiments, 30 images (resized to 128×128) per class (clutter class removed) are used for training, 10 for validation and the rest of the images in each class are used for testing. Average per class classification results are reported with an averaging over 5 random splits. A detailed comparison with unsupervised, semi-supervised, and supervised methods is also presented.

3.1. ScatterNet feature extraction

The scattering representations are extracted by first obtaining multi-resolution images of size (64×64) (R1) and (48×48) (R2) for CIFAR-10 and (256×256) (R1) and (192×192) (R2) for Caltech-101, as described in Section. 2.1. The images in the CIFAR dataset are decomposed for each colour

channel separately using DTCWT filters at 5 (for R1) and 4 (for R2) scales respectively, while the images in the Caltech dataset are decomposed with at 6 and 5 scales for R1 and R2 resolutions respectively. Next, log transformations are applied to the representations obtained (except at the coarsest scale) for both the R1 and R2 pipeline with parameters $k_{j=1} = 1.1$, $k_{j=2} = 3.8$, $k_{j=3} = 3.8$, $k_{j=4} = 7$ and $k_{j=4} = 6.8$ (selected as described in Section. 2.1), obtained by averaging the individual k value for the particular scale for all the images in the training dataset. The classification accuracies for each layer (L0, L1, L2) and the concatenated features (HC = S[L0, L1, L2]) are presented for both resolutions, using G-SVM in Table. 1. L2 features give a less good performance on their own than L1, probably due to their lower energies, but still, give a useful improvement when combined with L1.

Table 1. Accuracy (%) on CIFAR-10 for features extracted at different layers and resolutions. $S_{rs}[Layer]$, HC = S[L0,L1,L2]

	S[L1]	$S_{rs}[L1]$	S[L2]	$S_{rs}[L2]$	HC	HC_{rs}
R1	71.48	72.58	60.34	60.51	80.7	81.7
R2	72.04	73.39	60.12	60.39	80.9	81.9

3.2. PCA Layers: features and layer optimization

The L3 PCA layer of the network is trained on $S_{rs}[R1L1]$, $S_{rs}[R1L2]$, $S_{rs}[R2L1]$ and $S_{rs}[R2L2]$, to learn $K_{L3}=100$ filters of size $s_{L3}=5$. Cross-validation is used (as explained in Section. 2.2) on the L3 layer output (y_{L3}) to select the 40, 70, 50 and 80 optimal filters (\hat{K}_{L3}) for the four cases, as shown in Fig. 3(a). Next, the relatively symmetric L3 output ($\hat{y}_{L3,rs}$) is obtained by applying a log transformation with $k_{L3} = 1.8$, 1.9, 1.7 and 7.0 on \hat{y}_{L3} , respectively (Fig. 3(b)).

L4 PCA layer is trained on L3 layer outputs ($\hat{y}_{L3,rs}$) correspondingly to learn 200 (K_{L4}) filters, of size 5 (s_{L4}). Similarly, 150, 140, 120 and 130 optimal filters (\hat{K}_{L4}) are selected for the four cases ($S_{rs}[R1L1]$, $S_{rs}[R1L2]$, $S_{rs}[R2L1]$ and $S_{rs}[R2L2]$), as shown in Fig. 4(a). Next, the relatively symmetric L4 outputs ($\hat{y}_{L4,rs}$) are obtained by applying a log transformation with $k_{L4} = 2.0$, 1.3, 2.1 and 7.2 on $\hat{y}_{L4,rs}$, respectively, for the four cases, as shown in Fig. 4(b).

The five-fold cross-validation (5-CV) classification accuracies on CIFAR-10, obtained using G-SVM, at different stages of Layers L3 and L4 are presented in Table 2. There are fewer optimal filters in (K_{L3}^{op} , K_{L4}^{op}) than the originally learned filters (K_{L3} , K_{L4}) but produce an equal or higher cross-validation accuracy. This suggests that some of the filters learn redundant information which can be removed. This results in efficient learning of L4 layer (subsequently for OLS as well as SVM) as the L4 filters are learned from a smaller feature space $\hat{y}_{L3,rs}$ (obtained with $\hat{K}_{L3} \ll K_{L3}$).

3.3. Classification performance

This section evaluates the classification performance of each module of the SHDL network. The classification accuracy

Table 2. 5-CV Accuracy (%) on CIFAR-10 at L3 and L4. y_{L3} , y_{L4} output, \hat{y}_{L3} , \hat{y}_{L4} optimal output and $\hat{y}_{L3,rs}$, $y_{L4,rs}$ relatively symmetric output, at L3 and L4.

	y_{L3}	\hat{y}_{L3}	$\hat{y}_{L3,rs}$	y_{L4}	\hat{y}_{L4}	$\hat{y}_{L4,rs}$
$S_{rs}[R1L1]$	73.83	74.13	74.68	74.81	75.02	75.06
$S_{rs}[R1L2]$	60.78	60.96	61.04	60.93	61.36	61.38
$S_{rs}[R2L1]$	73.86	74.07	74.29	74.88	75.63	75.69
$S_{rs}[R2L2]$	60.81	61.11	61.23	61.78	62.02	62.67

of each module is presented by applying the supervised OLS layer on the features to select the relevant features which are then fed to the G-SVM to compute the accuracy. The accuracy of the handcrafted module (HC) is computed on the concatenated relatively symmetric features extracted at L0, L1, L2, for both resolutions (R1, R2) using OLS for feature selection and then G-SVM for classification. The hand-crafted module produced a classification accuracy of 82.4% (HC) on CIFAR-10 as shown in Table. 3. An increase of 0.4% is observed when the mid-level features, learned at L3 with $s_{L3}=5$ are concatenated with the features of the hand-crafted module (HC, $(L3)_{s_{L3}=5}$), again for both R1 and R2. A further increase of 0.7% (HC, $(L3, L4)_{s_{L3}, L4=5}$) is noticed when mid-level features from the L4 layer learned with $s_{L4}=5$ are concatenated to (HC, $(L3)_{s_{L3}=5}$) features. This suggests that the PCA layers (L3 and L4) learn useful image representations as they improve the classification performance. Finally, in order to test the optimality of the filter sizes, the L3 and L4 layers were also trained with $s_{L3}=3$ and $s_{L4}=3$. A further increase of around 0.4% (HC, $(L3, L4)_{3,5}$) is observed by concatenating the features obtained at L3 and L4 layers, with filters trained with the kernel s_{L3}, s_{L4} of size 3 and 5, with the hand-crafted module (HC). This suggests that filters of different sizes learn unique and useful image representations.

Table 3. Accuracy (%) on CIFAR-10 for each module computed with OLS and G-SVM. The increase in accuracy with the addition of each layer is also shown. HC: Hand-crafted, PCA features ($(Layer)_{filter-size}$): eg $(L3)_{s_{L3}=5}$

	HC	$HC, (L3)_5$	$HC, (L3, L4)_5$	$HC, (L3, L4)_{3,5}$
Acc.	82.4	82.8	83.5	83.9

Next, the performance of the SHDL network is evaluated on the Caltech-101 dataset. The network results in a classification accuracy of 81.46%, as shown in Table. 4.

3.4. Comparison with the state-of-the-art

The SHDL outperformed the semi-supervised and unsupervised learning methods on both datasets however the network underperformed by nearly 13% against supervised deep learning models [5, 6], as shown in Table. 4.

3.5. Advantage over supervised learning

Supervised models require large training datasets to learn which may not exist for most application. Table. 4 shows

Table 4. Accuracy (%) and comparison on both datasets. Unsup: Unsupervised, Semi: Semi-supervised and Sup: Supervised.

Dataset	SHDL	Semi	Unsup	Sup
CIFAR-10	83.90	83.3 [20]	82.9 [12]	96.2 [21]
Caltech-101	81.46	81.5 [22]	81.0 [13]	92.7 [6]

that SHDL network outperformed VGG [6] and Network in Network (NIN) [5] on the CIFAR-10 datasets with less than 2k images. The experiments were performed by dividing the training dataset of 50000 images into 8 datasets of different sizes. The images for each dataset are obtained randomly from the full 50000 training dataset. It is made sure that an equal number of images per object class are sampled from the training dataset. The full test set of 10000 images is used for all the experiments. Deeper models like NIN [5] and VGG [6] result in low classification accuracy due to their inability to train on the small training dataset.

Table 5. Comparison of SHDL network on accuracy (%) with two supervised learning methods (VGG [6] and NIN [5]) against different training dataset sizes on CIFAR-10.

Arch.	500	1K	2K	5K	10K	20K	50K
SHDL	50.3	57.9	63.4	68.6	72.3	78.4	83.9
NIN	15.6	54.5	61.1	72.9	81.2	86.7	89.6
VGG	10.3	10.7	43.4	63.4	72.0	83.1	92.7

4. CONCLUSION

The paper proposes the SHDL network that uses PCA-Net based unsupervised learning module to learn mid-level features while OLS based supervised learning is used to select features that aid the discriminative SVM learning. It is shown that a very simple PCA based network can learn useful features that can greatly improve the classification performance. The network has also shown to outperform unsupervised and semi-supervised learning methods while evidence of the advantage of SHL network over supervised learning (CNNs) methods is presented for small training datasets.

5. REFERENCES

- [1] Y. He et al., “Unsupervised feature learning by deep sparse coding,” *Proceedings of the SDM*, 2014.
- [2] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” *IEEE CVPR*, 2006.
- [3] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” *IEEE CVPR*, 2003.
- [4] J. Yang et al., “Linear spatial pyramid matching using sparse coding for image classification,” *CVPR*, 2009.
- [5] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv:1312.4400*, 2013.
- [6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ICLR*, 2015.
- [7] H. Lee R. Grosse, R. Rananth, and A. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representation,” *ICML*, 2009.
- [8] S. Jain et al., “A novel method to improve model fitting for stock market prediction,” *International Journal of Research in Business and Technology*, 2013.
- [9] Antonello Pasini, “Artificial neural networks for small dataset analysis,” *Journal of Thoracic Disease*, vol. 7, no. 11, pp. 2278–2324, 2015.
- [10] J. Sivic et al., “Unsupervised discovery of visual object class hierarchies,” *IEEE CVPR*, 2008.
- [11] T. Serre et al., “Robust object recognition with cortex-like mechanisms,” *IEEE PAMI*, 2007.
- [12] T.H. Lin et al., “Stable and efficient representation learning with non negativity constraints,” *ICML*, 2014.
- [13] S. McCann and D. Lowe, “Spatially local coding for object recognition,” *ACCV*, 2012.
- [14] A. Singh and N.G. Kingsbury, “Dual-tree wavelet scattering network with parametric log transformation for object classification,” *ICASSP*, 2017.
- [15] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” *IEEE PAMI*, vol. 35, pp. 1872–1886, 2013.
- [16] TH Chan et al., “Pcanet: A simple deep learning baseline for image classification?,” *ArXiv:1404.3606*, 2014.
- [17] T. Blumensath and M. E. Davies, “On the difference between orthogonal matching pursuit and orthogonal least squares,” 2007.
- [18] A Singh et al., “Multi-resolution dual-tree wavelet scattering network for signal classification,” *International Conference on Mathematics in Signal Processing*, 2016.
- [19] S Nadella et al., “Aerial scene understanding using deep wavelet scattering network and conditional random field,” *European Conference on Computer Vision*, 2016.
- [20] T. Salimans et al., “Improved techniques for training gans,” *ArXiv:1606.03498*, 2016.
- [21] S. Zagoruyko and N. Komodakis, “Wide residual networks,” *arXiv:1605.07146*, 2016.
- [22] Dengxin Dai et al., “Unsupervised high-level feature learning by ensemble projection for semi-supervised image classification and image clustering,” *arXiv*, 2016.