SOMNet: Unsupervised Feature Learning Networks for Image Classification

Richard Hankins  
School of Electrical and Electronic Eng.  
The University of Manchester  
Manchester, UK  
richard.hankins@manchester.ac.uk

Yao Peng  
School of Electrical and Electronic Eng.  
The University of Manchester  
Manchester, UK  
yao.peng@manchester.ac.uk

Hujun Yin  
School of Electrical and Electronic Eng.  
The University of Manchester  
Manchester, UK  
hujun.yin@manchester.ac.uk

Abstract—We present here an unsupervised approach to learning suitable features for a deep learning framework applied to image classification. PCANet was introduced as a simple and efficient baseline for deep learning approaches which used cascaded principle component analysis (PCA) derived filter banks, as well as other simple image processing elements such as binary hashing and blockwise histograms. This was followed by DCTNet which used discrete cosine transform (DCT) filter banks as a learning-free alternative. In this paper we propose SOMNet which uses self-organizing map (SOM) based filters offering a non-orthogonal alternative to PCANet providing comparable performance. It is well established that SOM is a non-linear version of PCA but does not suffer from the same constraints. We also show that through the use of a simple trick in the binarization process results in a dramatic reduction in the dimension of the final feature vector, thus allowing the utilization of more filters which could lead to deeper and more complex structures in further work. We also demonstrate the results of a hybrid methodology that clusters generative Markov random fields (MRF) as filters which provides more diverse features in a data driven approach to deep learning.

Index Terms—Self-organizing maps, unsupervised learning, Markov random fields, convolutional neural network, deep learning, handwritten digit recognition.

I. INTRODUCTION

The task of vision-based image classification has garnered much attention in recent years. Robust implementations have many wide ranging applications. However, the task is extremely challenging due to well known intra-class variations such as lighting, deformations, occlusions and misalignments. Up until recently low-level handcrafted features such as Gabor, local binary patterns (LBP) and scale invariant feature transform (SIFT) were used extensively to successfully overcome such problems. Yet manually designed features are very problem dependent, overly relying on domain-specific knowledge. Ergo they may not generalize well to other related data. In addition, handcrafted features can often be complex and thus expert knowledge is required to employ them on new conditions.

Using data driven representative learning techniques to learn task specific salient features has been proposed to obviate the need for complicated and costly engineered features. Deep learning has recently made numerous advances in many applications including object detection and localization [1], image recognition [2], as well as human action classification [3], [4]. Of these methods the convolutional neural network (CNN) [5] has become ubiquitous in the field of image classification [6] due to its superior performance, demonstrating above human-level accuracy in some cases [7], and its relative ease of use compared to more traditional handcrafted approaches. The cornerstone of its performance lies in its ability to leverage ever complex data-specific features through a multi-stage non-linear architecture to provide greater robustness to intra-class variations.

Despite CNN’s wide spread use, there is limited understanding of its complex learning mechanism and it is often used as a black box. In addition, the supervised backpropagation algorithm used to train CNN’s can be costly in terms of time. Furthermore, obtaining the best performance often requires fine tuning of its numerous parameters manually and the use of various expedient add-ons [8]. Scattering convolutional networks (SCATNet) [9] shone some much needed light on the underlying processes governing CNNs by using generated wavelet operators as learning-free filters. Used in a similar cascaded architecture, SCATNet outperformed both CNN and deep neural networks on two challenging datasets [9], [10].

Inspired by SCATNet, PCANet [11] was introduced as a simple deep learning framework for image classification. It utilizes multi-stage principle component analysis (PCA) derived filter banks followed by binarization and local histogramming for indexing and pooling. Despite its simple structure it performed surprisingly well on many image classification tasks, outperforming many more complex methodologies. This simple approach was further developed into a learning-free data-independent version called DCTNet [12]. DCTNet uses discrete cosine transform (DCT) derived filter banks as it has been demonstrated that DCT is a good approximation for the most important principle components. However, there are a number of inherent constraints that limit the architecture of both networks.

This paper presents a self-organizing map (SOM) based alternative to PCANet and DCTNet. We demonstrate both unsupervised data-driven and generative approaches and introduce a simple trick to the binarization process which greatly reduces the final feature vector size without significantly affecting performance. In addition SOM offers many benefits.
over the use of PCA and DCT, namely that the number and size of filters are not affected by the covariance matrix constraints.

II. RELATED WORK

Chan et al. [11] proposed PCANet that uses a filter bank derived by PCA. The resultant features are then convolved with the input to form activations. This is repeated for each subsequent layer. No activation function is used between layers, instead a non-linearity is introduced in the final layer. Their experiments with different architectures find that a two-layer network achieves the best results. The output layer produces the final feature vector by first binarizing the activation maps of the previous layer and then local histogramming. Binarization involves converting the activation maps to binary using a Heaviside function then transforming each pixels bit string to decimal to produce a single activation map. Binarized Statistical Image Features (BSIF) [13] used this binarization technique prior to PCANet, however it was only used as an image descriptor. Due to the size of the resultant output feature vectors further reduction techniques are sometimes necessary. Whilst PCANet produces excellent results for such a compact network there are some limitations. Specifically the filter number and depth are limited due to aforementioned processes used to achieve the final feature vector (all architectures experimented restrict the number of filters in the second layer to eight). In addition since PCA is used to learn the filters the maximum number achievable is limited due to the covariance matrix. The size of the filters also governs the mixture of low and high frequency components in the filter bank.

In DCTNet, DCT basis functions are chosen as an alternative to PCA due to its data independence and low complexity. Similar to characteristics that are used in DCT based JPEG compression, DCT filters are selected in a zig-zag fashion excluding the mean component. In addition to using a different basis, DCTNet also employs a further tied rank (TR) normalization step which regulates the histogram and leads to improved performance over PCANet on certain data. However, DCTNet suffers from the same inherent drawbacks as PCANet. In addition, the learning-free approach relies on the input distribution following a local high correlation assumption, which can be considered as a 2D first-order Markov process.

The self-organizing map (SOM) [14] is an unsupervised algorithm first introduced by Kohonen. It uses competitive learning in order to quantize an input space whilst preserving data topological structure on the map. A set of input vectors are reduced or mapped to a smaller set of prototype vectors, which are representative of the data distribution. In this respect SOM is similar to PCA except SOM is not restricted to finding orthogonal principle directions. In fact SOM can be considered as a non-linear version of PCA [15]. Recently SOMs have been used to learn features [16] for tasks including face recognition [17] and handwritten digit recognition [17, 18].

As an extension of the simple Markov process of DCT, Markov random fields (MRFs) have a long history in computer vision and have found applications in texture synthesis [19] and image classification [20]. Generated MRF features are also explored in this paper.

III. METHODOLOGY

A. Self-Organizing Map

Prototype vectors or neurons are commonly arranged in a 2D grid. Consider each unit i in the SOM has reference vector $w_i = [w_{1i}, w_{2i}, ..., w_{ni}]^T \in \mathbb{R}^d$ with equal dimension as the input vector. At each time step the winner or best matching unit on the map is selected by computing the maximum activation:

$$bmu(t) = \arg \max_{i \in \Omega} |w_i \ast x(t)|$$

and weights of the winner and its neighbors are updated according to

$$\Delta w_i(t) = \alpha(t) + \eta(bmu, i, t)|x(t) - w_i(t)|$$

where $\ast$ denotes convolution, $\Omega$ is the set of neuron indices, $\alpha (0 < \alpha(t) < 1)$ is the monotonically decreasing learning rate, $\eta(bmu, i, t) = \exp\left[-\frac{\|r_i - r_{bmu}\|^2}{\sigma^2}\right]$ is the neighborhood function with $r_i$ being the location of neuron $i$ on the map and $\sigma$ the effective range of the neighborhood, which decreases with time.

B. Markov Random Field

Given an undirected graph $G \triangleq (S, N)$ and a group of random variables $F = \{F_1, F_2, \cdots, F_k\}$, the random field $F$ is considered as an MRF on the set $S$ with respect to the neighborhood system $N$ if the following properties are satisfied:

1) Positivity: Any configuration is positive.

$$P(f) > 0 \quad \forall f \in \mathbb{F},$$

where $f = \{f_1, f_2, \cdots, f_k\}$ is one possible configuration and $\mathbb{F}$ denotes the set of all possible configurations of the field $F$.

2) Markovianity: The conditional probability of any site given the others only depends upon the configuration of its neighbors.

$$P(f_i \mid f_{S-\{i\}}) = P(f_i \mid f_j, j \in N_i) = P(f_i \mid N_i),$$

where $S - \{i\}$ denotes all sites excluding site $i$, $f_{S-\{i\}}$ is the set of labels at the sites in $S - \{i\}$.

An MRF is an undirected graphical model with a set of random variables satisfying Markov properties. Proposed by Hammersley and Clifford and developed by Besag [21]–[23], the Hammersley-Clifford theorem demonstrates the equivalence between MRFs and Gibbs distribution, which overcomes the difficulty of specifying an MRF from the conditional probabilities and provides a mathematically means of defining MRFs.
Let \( P(f) \) denote a Gibbs distribution on the set \( S \). Then the joint probability \( P(f) \) takes the form

\[
P(f) = \frac{1}{Z} e^{-\frac{1}{T} U(f)}
\]

(5)

where

\[
U(f) = \sum_{c \in C} V_c(f)
\]

(6)

is the energy function that sums clique potentials \( V_c(f) \) over all possible cliques \( C \) and stays positive for all possible configurations. \( T \) is a constant called the temperature, and \( Z \) is a normalizing constant defined by

\[
Z = \sum_{f \in F} e^{-\frac{1}{T} U(f)}
\]

(7)

which is also called the partition function.

While MRFs only specify the conditional dependencies, Gibbs distributions provide explicit probability function for each clique. The Hammersley-Clifford theorem establishes the equality of the MRF and Gibbs distribution, which states that the joint probability of any MRF can be written as a Gibbs distribution, and for any Gibbs distribution there exists an MRF model.

To extract image features, MRFs model the contextual relationships between pixels according to the intensity values. Auto-models and multi-level logistic models are typical models for establishing neighborhood functions and conditional probabilities. Given the contextual constraints, simulated annealing [24] is a commonly used method for approximating the global minimum.

C. Self-Organizing Map Network (SOMNet)

The architecture of SOMNet closely resembles PCANet and DCTNet except we adapt the encoding process during the binarization process. The structure is depicted in Fig. 1. The following sections detail the behavior of each layer. We refer to filter banks generated by both SOM and MRF. The SOM features are learned from randomly sampled \( s \times s \times d \) patches from the input which are mean normalized by subtracting the patch mean from each pixel (Fig. 2). When sampling from activations a single \( s \times s \) patch is sampled from a single activation map uniformly. Each SOM is trained until convergence with the second SOM being trained for twice that of the first. Given that the activation data space is larger than the input, we thought we should scale the training time of the second SOM by the same ratio. However, the results showed that this was unnecessary. Whilst the first SOM is able to reach a global minimum, due to the bootstrap sampling process and the increased input space of the activations the second layer SOM only converges to local minima. Yet this does not appear to be detrimental to the performance. When generated MRF features are reduced to a subset using SOM and used as filter banks we refer to this network as MRF-SOMNet. Given the smaller input space of the MRF-SOM, it is trained in a more systematic way to ensure convergence.

1) Convolution Layer: Given an input image \( I_d \) of size \( m \times n \) with \( d \) channels (either original input or activation from previous layer), convolution with a filter bank of filter size \( s \times s \times d \) is performed. The filter size remains constant for each layer. Prior to the convolution operation zero padding is applied with pad size \((s - 1)/2\) so that the output is the same size as the input. Convolution of the input \( I_d \) with the filter bank \( W_i \in \mathbb{R}^{s \times s \times d}, i = 1, 2, \ldots, h_l \) where \( h_l \) is the number of filters in layer \( l \), provides

\[
O^l_i = [I_j * W^l_i], i = 1, 2, \ldots, h_l, j = 1, 2, \ldots, d
\]

(8)

2) Binarization and Histogram Layer: The input to the final layer has size \( s \times s \times d \times h_l \). Binarization is performed on each set of \( d \) real-valued outputs by performing \( He(O^l_i) \), where \( He(\cdot) \) is the Heaviside step function which thresholds the values at zero. Each resultant binary string is grouped into four bit nibbles (hexadecimal), \( G^\beta_i \), and encoded using

\[
\sum_{\beta=1}^{4} 2^{\beta-1} G^\beta_i, \epsilon = 1, 2, \ldots, \frac{h_l}{4}
\]

(9)

which produces \( h_l \times d \) encoded output images where each pixel has the range \([0, 2^4 - 1]\). After which each of these images is split into \( B_{size} \times B_{size} \) blocks with overlapping ratio \( \tau \). Within each block histograms are formed and concatenated together to form the final output feature vector \( \hat{\theta} \in \mathbb{R}^{2^d dB} \), where \( B \) is the number of local histograms. Splitting the images into local histograms encodes spatial information and also some degree of invariance to translations. In comparison, PCANet has \( \hat{\theta} \in \mathbb{R}^{2^d dB} \) giving an output feature vector ratio

\[
\frac{\theta}{\hat{\theta}} = \frac{2^{h_1}}{4h_l}
\]

(10)

which is exponentially increasing. Therefore \( \theta \) will always be smaller than \( \hat{\theta} \) when \( h_l > 4 \). The same is also true for DCTNet and other PCANet derivatives which employ the same encoding strategies. Given that four filters is uncommon in state-of-the-art architectures it is safe to assume that the our binarization process offers a more compact representation. The final output feature vector is then classified using a linear support vector machine (SVM) [25] classifier.

D. Computational Complexity

The computational complexity of the proposed methods are now considered. We first consider SOMNet for which the complexity includes mean patch removal and convolution for both convolution stages as well as binary hashing and histogramming for the final stage. In addition to this there is the actual complexity of SOM itself which is \( s^2(h_1 + h_2) \). However, the overall complexity can be calculated as \( O(mns^2(h_1 + h_2)) \). Similarly, PCANet, DCTNet and MRF-SOMNet have the same network complexities however PCANet has the added operations included in eigen-decomposition as well as the increased dimensionality in the second convolutional stage.
Fig. 1. Block diagram of SOMNet.

Fig. 2. Training the SOM-based filter banks.
Table I summaries the final complexity results. For DCTNet and MRF-SOMNet we ignored the operations taken to generate the filters since this occurs only once.

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOMNet</td>
<td>$O\left(\frac{mns^2(h_1 + h_2)}{h_2}\right)$</td>
</tr>
<tr>
<td>PCANet</td>
<td>$O\left(\frac{mns^2(h_1 + h_2)}{h_1h_2}\right) + mn(h_1h_2)^2 \right)$</td>
</tr>
<tr>
<td>DCTNet</td>
<td>$O\left(\frac{mns^2(h_1 + h_2)}{h_2}\right)$</td>
</tr>
<tr>
<td>MRF-SOMNet</td>
<td>$O\left(\frac{mns^2(h_1 + h_2)}{h_2}\right)$</td>
</tr>
</tbody>
</table>

IV. Experiment and Discussion

In this section we compare the performance of SOMNet and MRF-SOMNet against PCANet, DCTNet and other state-of-the-art methods on benchmark data. Due to the high dimensionality of the output feature vector all experiments use linear SVM for classification.

A. Evaluation on MNIST

Formed from the larger NIST dataset the MNIST image dataset [26] is a collection of 70000 grayscale $28 \times 28$ images of handwritten digits 0-9. For the purposes of our experiments we use 60000 for training and 10000 for testing. With regard to the parameters of SOMNet we use a double convolution layer architecture with $h_1 = h_2 = 8$, $s_1 = s_2 = 7$, $B_{\text{size}} = 7$, and $\tau = 0.5$; which is in line with PCANet. DCTNet does not use overlapping histograms so we provide results for the default version as well as a version which has to match SOMNet and PCANet. These are referred to as DCTNet and DCTNet\(_{\text{overlap}}\) respectively. To demonstrate the benefits of our binarization process we also give results for a SOMNet with $h_1 = 16$ and $h_2 = 32$ (all other parameters remain the same as above), referred to as SOMNet\(_{\text{16-32}}\). Whilst this version uses more filters than PCANet and DCTNet methods the output feature vector is equal in size to PCANet and DCTNet\(_{\text{overlap}}\). Table II shows the results along with other state-of-the-art methods. For PCANet we show both published results and the results of our own run of the experiment using the provided code. DCTNet does not provide published results on MNIST so again we use the code provided. For both these methods we use an SVM with default parameters for a fair comparison with SOMNet (DCTNet originally uses the nearest neighbor for classification). We also show results for SOMNet after fine tuning of the SVM using five fold cross-validation. Given that SOMNet’s second layer filters converge to local minima our results are the average of three runs. For all other methods (PCANet-2 (ours), DCTNet, DCTNet\(_{\text{overlap}}\), SOMNet\(_{\text{rep}}\) and MRF-SOMNet) only a single result is necessary. In addition, to explore whether it is essential to learn filters in the second layer we give a result for when the first layer filters of the SOMNet are replicated over both filter banks (referred to as SOMNet\(_{\text{rep}}\)). As can be seen our methodology is highly competitive in terms of accuracy. The results for SOMNet\(_{\text{rep}}\) clearly show that learning filters based on the activations of the previous layer improves performance over using replicated ones. The hybrid MFR-SOMNet gives a much poorer result but the reason for this become clearer on inspection of the filters.

Figs. 3 - 6 show the filter banks for SOMNet, PCANet, DCTNet and MRF-SOMNet respectively. MRF-SOMNet filters contain a lot more high frequency components which are absent in the input space (handwritten digits), this explains the poorer results. The results for DCTNet are surprising given their simplicity and are the best in the second part of the table excluding the published PCANet result which we could not replicate here. DCTNet has already produced credible results on face recognition where the equivalence between PCA and DCT is clear. These results show that DCT filters are actually quite flexible. We note that the TR normalization does not seem to have much effect here. PCANet and SOMNet filters are, as expected, more similar, as both are data-driven. Whilst SOMNet provides slightly inferior results compared to PCANet and DCTNet it provides similar results to other more complex methods in the first half of the table. Furthermore, SOMNet achieves these results using an output feature vector that is eight times smaller than PCANet/DCTNet\(_{\text{overlap}}\) and four times smaller than DCTNet. The benefits of this more compact representation are further highlighted by the results of SOMNet\(_{16-32}\) which utilizes additional features for a more over-complete representation, providing, alongside the published PCANet result, the best performance in the second and third part of the table. This therefore demonstrates a clear advantage and a future scope for improvement. Given more filters of different sizes, which our method allows for, it would be conceivable that SOMNet could improve even further on its current performance.

B. Evaluation on CIFAR-10

The CIFAR-10 [29] object recognition dataset consists of 60000 $32 \times 32$ color images of 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse and truck. The class distribution
Fig. 3. Learned SOMNet filters on MNIST. Top row: the first layer filter bank. Bottom row: the second layer filter bank.

Fig. 4. Learned PCANet filters on MNIST. Top row: the first layer filter bank. Bottom row: the second layer filter bank.

Fig. 5. Generated DCTNet filters for MNIST. Replicated across both filter banks.

Fig. 6. Clustered MRF filters for MNIST. Replicated across both filter banks.

is uniform and the training and testing sets contain 50000 and 10000 examples respectively. As with the previous MNIST experiment we use a double convolutional architecture. Like PCANet we set SOMNet parameters $h_1 = 40$, $h_2 = 8$ and $s_1 = s_2 = 5$. However, unlike PCANet we set $B_{\text{size}} = 4$, and $\tau = 0$ to keep the final output feature size manageable (PCANet performs a further PCA step to reduce the dimension of the output feature vector by a factor of eight). Since SOM is incapable of dealing with correlations, before training, we first whiten the data so that it is uncorrelated; exaggerating the high frequency content. SOMNet layers are then trained using the whitened data. As with PCANet we also attach spatial pyramid pooling (SPP) [30] to the output so that the maximum response in each histogram bin is pooled in $4 \times 4$, $2 \times 2$, and $1 \times 1$ subregions producing 21 pooled histograms. We provide the published PCANet result as well as fine tuned SOMNet and MRF-SOMNet results alongside the results for other state-of-the-art methods in Table III. Once again due to different local minima found during training the results for SOMNet are averaged over three runs. Figs. 7 - 9 show the filter banks for SOMNet, PCANet and MRF-SOMNet respectively. As can be seen by the results our methodology does not perform as well as PCANet or other state-of-the-art methods. SOMNet achieves an accuracy of 71.81% which is 5.33% lower than PCANet. Yet, PCANet performs an extra dimensionality reduction step in producing this; the effects of which are not examined in this work. As with the MNIST experiment the naively generated MRF-SOMNet performs slightly worse than the data-driven SOMNet, however the difference is not statistically significant. This is as expected since the input space contains more high frequency content which is covered by the filters.

### Table III

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means (Triangle, 4000 features) [31]</td>
<td>79.60</td>
</tr>
<tr>
<td>Stochastic Pooling ConvNet [32]</td>
<td>84.87</td>
</tr>
<tr>
<td>NIN + Dropout [33]</td>
<td>89.59</td>
</tr>
<tr>
<td>MRF-CNN [20]</td>
<td>91.25</td>
</tr>
<tr>
<td>SOMNet (fine tune)</td>
<td>71.81 ± 0.06</td>
</tr>
<tr>
<td>MRF-SOMNet (fine tune)</td>
<td>71.1</td>
</tr>
</tbody>
</table>

Fig. 7. Learned SOMNet filters on CIFAR-10. Top five rows: the first layer filter bank. Bottom row: the second layer filter bank.

Fig. 8. Learned PCANet filters on CIFAR-10. Top five rows: the first layer filter bank. Bottom row: the second layer filter bank.

Fig. 9. Learned MRF filters on CIFAR-10. Top five rows: the first layer filter bank. Bottom row: the second layer filter bank.
In this paper we propose a simple SOM based unsupervised deep learning framework. Like its predecessors PCANet and DCTNet it offers an uncomplicated alternative to more traditional supervised convolutional based deep learning structures. We demonstrate that SOMNet is able to extract useful information and give competitive results on the MNIST dataset. We also show that the introduction of more filters enabled by our alternative binarization technique improved on the baseline MNIST result. Despite the inferiority of SOMNet on CIFAR-10 we believe that with a similar suitable extension of the filters, which our method allows for, further improvements are possible. In addition, we also explore the use of clustered MRF features replicated across both filter banks in MRF-SOMNet. Whilst the results for MNIST were unfavorable its performance on CIFAR-10 was encouraging however, an investigation of the role of learning the second layer filters from the first layer’s activations should be explored in this context as the importance of this was shown by the SOMNet investigation of the role of learning the second layer filters on CIFAR-10.  

V. CONCLUSIONS AND FUTURE WORK

We demonstrate that SOMNet is able to extract useful information and give competitive results on the MNIST dataset. We also show that the introduction of more filters enabled by our alternative binarization technique improved on the baseline MNIST result. Despite the inferiority of SOMNet on CIFAR-10 we believe that with a similar suitable extension of the filters, which our method allows for, further improvements are possible. In addition, we also explore the use of clustered MRF features replicated across both filter banks in MRF-SOMNet. Whilst the results for MNIST were unfavorable its performance on CIFAR-10 was encouraging however, an investigation of the role of learning the second layer filters from the first layer’s activations should be explored in this context as the importance of this was shown by the SOMNet result. Generally, whilst the results are promising this work is clearly still in progress. For more resolute conclusions we will need to provide more extensive testing on further data. We have outlined further avenues of study which we believe could address some of the stated limitations of PCANet and DCTNet, allow for more complex architectures, and achieve improved results, on a wider range of datasets. Whilst in this paper we have kept the parameters mostly inline with those for PCANet and DCTNet, in future, we will experiment with different filter sizes and numbers as well as incorporating further layers.

REFERENCES


