Attn-HybridNet: Improving Discriminability of Hybrid Features with Attention Fusion

Sunny Verma, Chen Wang, Liming Zhu, *Member, IEEE* and Wei Liu, *Senior Member, IEEE*

arXiv:2010.06096v2 [cs.CV] 14 Oct 2020

***Abstract*—The principal component analysis network (PCANet) is an unsupervised parsimonious deep network, utilizing principal components as filters in its convolution layers. Albeit powerful, the PCANet consists of basic operations such as *principal components* and spatial pooling, which suffers from two fundamental problems. First, the *principal components* obtain information by transforming it to column vectors (which we call the amalgamated view), which incurs the loss of the spatial information in the data. Second, the generalized spatial pooling utilized in the PCANet induces feature redundancy and also fails to accommodate spatial statistics of natural images. In this research, we first propose a tensor-factorization based deep network called the Tensor Factorization Network (TFNet). The TFNet extracts features from the spatial structure of the data (which we call the minutiae view). We then show that the information obtained by the PCANet and the TFNet are distinctive and non-trivial but individually insufficient. This phenomenon necessitates the development of proposed *HybridNet*, which integrates the information discovery with the two views of the data. To enhance the discriminability of hybrid features, we propose *Attn-HybridNet*, which alleviates the feature redundancy by performing attention-based feature fusion. The significance of our proposed *Attn-HybridNet* is demonstrated on multiple real-world datasets where the features obtained with *Attn-HybridNet* achieves better classification performance over other popular baseline methods, demonstrating the effectiveness of the proposed technique.**

***Index Terms*—Tensor Decomposition, Feature Extraction, Attention Networks, Feature Fusion**

I. Introduction

Feature engineering is an essential task in the development of machine learning systems and has been well-studied with substantial efforts from communities including computer vision, data mining, and signal processing [1]. In the era of deep learning, the features are extracted by processing the data through multiple stacked layers in deep neural networks. These deep neural networks sequentially perform sophisticated operations to discover critical information concealed in the data [2]. However, the training time required to obtain these superior data representations is exponentially large as these networks have an exhaustive hyper-parameter search space and usually suffer from various training difficulties [3]. Besides, the deep networks are complex models that require high computational resources for their training and deployment. Hence, this limits the usability of these networks on micro-devices such as cellphones [4], [5]. The current research trend focuses on alleviating the memory and space requirements associated with the deep networks [6].

In this regard, to produce lightweight convolution neural networks (CNNs) architecture, most of the existing solutions would 1) approximate the convolution and fully connected layers by factorization [7], [8], 2) compress the layers with quantization/hashing [5], [9], or, 3) replace the fully connected layer with a tensorized layer and optimize the weights of this layer by retraining [6]. However, all these techniques require a pre-trained CNN network and can only work as post-optimization technique. By contrast, the goal of this research is to build lightweight deep networks which are computationally inexpensive to train. In other words, this research aims to build deep networks that are independent of a) high-performance hardware and b) exhaustive hyper-parameter search space, where a PCANet represents one such promising architecture.

The PCANet is an unsupervised deep parsimonious network utilizing *principal components* as convolution filters for extracting features in its cascaded layers [10]. Due to the remarkable performance of PCANet on several benchmark face datasets, the network is recognized as a simple deep learning baseline for image classification. However, the features extracted by PCANet do not achieve competitive performance on challenging object recognition datasets such as CIFAR-10 [11]. There are two major reasons for this performance limitation: 1) the PCANet vectorizes the data while extracting *principal components* which results in loss of spatial information exhibiting in the data and, 2) the output layer (which is spatial-pooling) utilized in the PCANet induces feature redundancy and does not adapt to the structure of natural images, deteriorating the performance of classifiers [12], [13]. However, the vectorization of the data is inherent with the *principal components* and hence motivates to devise techniques that can alleviate the loss of spatial information present in the data. In other words, techniques that can extract information from the untransformed view of the data[1] which is proven to be beneficial in literature [14], [15], [16].

In this research, we first propose an unsupervised tensor factorization based deep network called Tensor Factorization Network (TFNet). *The TFNet, contrary to the PCANet, does*

Sunny Verma is with The Data Science Institute, University of Technology Sydney, Australia (email: Sunny.Verma@uts.edu.au

Wei Liu is with the School of Computer Science, University of Technology Sydney, Australia (email: Wei.Liu@uts.edu.au).

Chen Wang and Liming Zhu are with Data61, Commonwealth Scientific and Industrial Research Organization, CSIRO, Sydney, Australia (email: Chen.Wang@data61.csiro.au, Liming.Zhu@data61.csiro.au).

The source code of proposed technique and extracted features are available at https://github.com/sverma88/Attn-HybridNet—IEEE-TCYB.

[1]Throughout this paper we refer to the vectorized presentation of the data as the amalgamated view where all modes of the data (also called dimension for higher order-matrices, i.e. tensors) are collapsed to obtain a vector. The untransformed view of the data, i.e., when viewed with its multiple modes (e.g., tensors), is referred to as the minutiae view of the data.

not vectorize the data while obtaining weights for its convolution filters. Therefore, it is able to extract information associated with the spatial structure of the data or the minutiae view of the data. Besides, the information is independently obtained from each mode of the data, providing several degrees of freedom to the information extraction procedure of TFNet.

Importantly, we hypothesize that the information obtained from either the amalgamated view or the minutiae view of the data is essential but individually insufficient as they respectively conceal complementary information associated with the two views[2] of the data [17], [15]. Therefore, the integration of information from these two views can enhance the performance of classification systems. To this end, we propose the *Hybrid Network* (*HybridNet*) that integrates information discovery and feature extraction from the minutiae view and the amalgamated view of the data simultaneously in its consolidated architecture.

Although the *HybridNet* reduces the information loss by integrating information from the two views of the data, it may still suffer from feature redundancy problems arising from the generalized spatial pooling utilized in the output layer of PCANet. Therefore, we propose an attention-based fusion scheme *Attn-HybridNet* that performs feature selection and aggregation, thus enhancing the feature discriminability in hybrid features.

The superiority of feature representations obtained with the *Attn-HybridNet* is validated by performing comprehensive experiments on multiple real-world benchmark datasets. The differences and similarities between the PCANet, TFNet, *HybridNet*, and *Attn-HybridNet* from data view perspectives are summarized in Table I.

Our contributions in this paper are summarized below:

- We propose *Tensor Factorized Network* (TFNet), which extracts features from the minutiae view of the data and hence is able able to preserve the spatial information present in the data that is proven beneficial for image classification.
- We propose Left one Mode Out Orthogonal Iteration (*LoMOI*) algorithm, which optimizes convolution weights from the minutiae view of the data utilized in the proposed TFNet.
- We introduce the *Hybrid Network* (*HybridNet*), which integrates the feature extraction and information discovery procedure from two views of the data. This integration procedure reduces information loss from the data by combining the merits of the PCANet and TFNet and obtains superior features from both of the two schemes.
- We propose the *Attn-HybridNet*, which alleviates feature redundancy among hybrid features by performing feature selection and aggregation with an attention-based fusion scheme. The *Attn-HybridNet* enhances the discriminability of the feature representations, which further theoretically improves the classification performance of our scheme.

[2]Throughout this paper by two views, we mean the amalgamated view and the minutiae view.

Methods	Amalgamated View	Minutiae View	Attention Fusion
PCANet [10]	✓	×	×
TFNet [18]	×	✓	×
HybridNet [18]	✓	✓	×
Attn-HybridNet	✓	✓	✓

TABLE I: Comparison of different feature extraction models

- We perform comprehensive evaluations and case studies to demonstrate the effectiveness of features obtained by *Attn-HybridNet* and *HybridNet* on multiple benchmark real-world datasets.

The rest of the paper is organized as follows: in Sec. II we present the literature review including prior works and background on PCANet and tensor preliminaries. We then present the details of our proposed TFNet, *HybridNet*, and *Attn-HybridNet* Sec. III, Sec. IV, and Sec. V respectively. Next we describe our experimental setup, results and discussions in Sec. VI and Sec. VII. Finally, we conclude our work and specify the future directions for its improvement in Sec. VIII.

II. Literature Review

The success of utilizing CNNs for multiple computer vision tasks such as visual categorization, semantic segmentation, etc. has lead to drastic research and development in the deep learning field. At the same time, to supersede human performance on these tasks, the CNNs requires an enormous amount of computational resources. For example, the ResNet [19] achieves a top-5 error rate of 3.57% that consists of 152-layers accounting for a total of 60M parameters and requires 2.25×10^{10} flops of the data at inference. This substantial computational cost restricts the applicability of such models on devices with limited computational resources such as mobile devices. Reducing the size and time complexities of the CNNs has, therefore, become a non-trivial task for their practical applications. In this regard, researchers actively pursuit three main active research directions: 1) compression of trained CNNs weights with quantization, 2) approximating convolution layer with factorization, and 3) replacing fully connected layers with custom-built layers.

In the first category, the aim is to reduce the size of trained CNNs by compressing its layers with quantization or hashing, as in [5], [9], [20]. These quantized CNN models achieve similar recognition accuracy with significantly less requirements for computational resources during inference. Similarly, the works in [7], [8] obtain approximations of fully connected and convolution layers by utilizing factorization for compressing the CNN models. However, both the quantization and factorization based methods compress a pre-trained CNN model instead of building a smaller or faster CNN model in the first place. Therefore, these techniques inherit the limitations of the pre-trained CNN models.

In the second and the third categories, the aim is to replace fully connected layers by customized lightweight layers that substantially reduce the size of any CNN model. For example, in [6] proposes a neural tensor layer while the work in [4] proposes a BoF (Bag-of-features) model as a neural pooling layer. These techniques augment conventional CNNs layers

and produce their lightweight versions which are trainable in an end-to-end fashion. However, a major limitation of these work is that they are only capable of replacing a fully connected layer, and in order to replace a convolution layer, they usually end up functioning similarly to the work in the first category.

Different from the above research, possible solutions for obtaining lightweight CNN architecture with lower computational requirements on smaller size images are proposed in PCANet [10] and TFNN [16]. The PCANet is a deep unsupervised parsimonious feature extractor, whereas TFNN is a supervised CNN architecture utilizing neural tensor factorizations for extracting information from multiway data. Both these networks achieve very high classification performance on handwritten digits dataset but fail to obtain competitive performance on object recognition dataset. This is because the PCANet (and its later variants FANet [21]) incur information loss associated with the spatial structure of the data as it obtains weights of its convolution filters from the amalgamated view of the data. Contrarily, the TFNN extracts information by isolating each view of the multi-view data and fails to efficiently consolidate them for their utmost utilization, incurring the loss of common information present in the data.

Therefore, the information from both the amalgamated view and the minutiae view is essential for classification, and their integration can enhance the classification performance [17], [15]. In this research, we first propose *HybridNet*, which integrates the two kinds of information in its deep parsimonious feature extraction architecture. A major difference between *HybridNet* and PCANet is that the *HybridNet* obtains information from both views of the data simultaneously, whereas the PCANet is restricted to obtain information from the amalgamated view of the data. The *HybridNet* is also notably different from TFNN as the *HybridNet* is an unsupervised deep network while the TFNN is a supervised deep neural network. Moreover, the *HybridNet* extracts information from minutiae view of the data, whereas the TFNN extracts information by isolating each mode of multi-view data.

Moreover, to enhance the discriminability of the features obtained with *HybridNet*, we propose the *Attn-HybridNet*, which performs attention-based fusion on hybrid features. The *Attn-HybridNet* reduces feature redundancy by performing feature selection and obtains superior feature representations for supervised classification. We present the related background preliminaries in the next subsection.

A. Background

We briefly summarize PCANet's 2-layer architecture and provide background on tensor preliminaries in this section.

1) The First Layer: The procedure begins by extracting overlapping patches of size $k_1 \times k_2$ around each pixel in the image; where patches from image $\boldsymbol{I}_i$ are denoted as $\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{i,\tilde{m}\tilde{n}} \in \mathbb{R}^{k_1 k_2}$, $\tilde{m} = m - \lceil \frac{k_1}{2} \rceil$[3] and $\tilde{n} = n - \lceil \frac{k_2}{2} \rceil$. Next, the obtained patches are zero-centered by subtracting the mean of the image patches and *vectorized* to obtain $\boldsymbol{X}_i \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$ as the patch matrix. After repeating the same procedure for all the training images we obtain $\boldsymbol{X} \in \mathbb{R}^{k_1 k_2 \times N\tilde{m}\tilde{n}}$ as the final patch-matrix from which the pca filters are obtained. The PCA minimizes the reconstruction error with orthonormal filters known as the principal eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ calculated as in Eq. 1

$$\min_{\boldsymbol{V} \in \mathbb{R}^{k_1 k_2 \times L_1}} \|\boldsymbol{X} - \boldsymbol{V}\boldsymbol{V}^T\boldsymbol{X}\|_F, \ s.t. \quad \boldsymbol{V}^T V = \boldsymbol{I}_{L_1} \tag{1}$$

where $\boldsymbol{I}_{L_1}$ is an identity matrix of size $L_1 \times L_1$ and L_1 is the total number of obtained filters. These convolution filters can now be expressed as:

$$\boldsymbol{W}^1_{l_{PCANet}} = mat_{k_1,k_2}(ql(\boldsymbol{X}\boldsymbol{X}^T)) \in \mathbb{R}^{k_1 \times k_2} \tag{2}$$

where $mat_{k_1,k_2}(v)$ is a function that maps $v \in \mathbb{R}^{k_1 k_2}$ to a matrix $\boldsymbol{W} \in \mathbb{R}^{k_1 \times k_2}$, and $ql(\boldsymbol{X}\boldsymbol{X}^T)$ denotes the l-th principal eigenvector of $\boldsymbol{X}\boldsymbol{X}^T$. Next, each training image $\boldsymbol{I}_i$ is convolved with the L_1 filters as in Eq. 3.

$$\boldsymbol{I}^l_{i_{PCANet}} = \boldsymbol{I}_i * \boldsymbol{W}^1_{l_{PCANet}} \tag{3}$$

where $*$ denotes the 2D convolution and i, l are the image and filter indices respectively. Importantly, the boundary of image $\boldsymbol{I}_i$ is padded before convolution to obtain $\boldsymbol{I}^l_{i_{PCANet}}$ with the same dimensions as in $\boldsymbol{I}_i$. From Eq. 3 a total of $N \times L_1$ images are obtained and attributed as the output from the first layer.

2) The Second Layer: The methodology of the second layer is similar to the the first layer. We collect overlapping patches of size $k_1 \times k_2$ around each pixel from all input images in this layer i.e., from $\boldsymbol{I}^l_{i_{PCANet}}$. Next, we vectorize and zero-centre these images patches to obtain the final patch matrix denoted as $\boldsymbol{Y} \in \mathbb{R}^{k_1 k_2 \times L_1 N\tilde{m}\tilde{n}}$. This patch matrix is then utilized to obtain the convolution *pca* filters in layer 2 as in Eq. 4.

$$\boldsymbol{W}^2_{l_{PCANet}} = mat_{k_1,k_2}(ql(\boldsymbol{Y}\boldsymbol{Y}^T)) \in \mathbb{R}^{k_1 \times k_2} \tag{4}$$

where $l = [1, L_2]$ denotes the number of pca filters obtained in this layer. Next, the input images in this layer $\boldsymbol{I}^l_{i_{PCANet}}$ are convolved with the learned filters $\boldsymbol{W}^2_{l_{PCANet}}$ to obtain the output from this layer in Eq. 5. These images are then passed to the feature aggregation phase as in the next subsection.

$$\boldsymbol{O}^l_{i_{PCANet}} = \boldsymbol{I}^l_{i_{PCANet}} * \boldsymbol{W}^2_{l_{PCANet}} \tag{5}$$

3) The Output Layer: The output layer combines the output from all the convolution layers of PCANet to obtain the feature vectors. The process initiates by first binarizing each of the real-valued outputs from Eq. 5 by utilizing a Heaviside function $H(\boldsymbol{O}^l_{i_{PCANet}})$ on them, which converts the positive entries to 1 otherwise 0. Then, these L_2 outputs are assembled into L_1 batches, where all images in a batch belong to the same convolution filter in the first layer. Then, these images are combined to form a single image by applying weighted sum as in Eq. 6 whose pixel value is in the range $[0, 2^{L_2} - 1]$:

$$\boldsymbol{I}^l_{i_{PCANet}} = \sum_{l=1}^{L_2} 2^{l-1} H(\boldsymbol{O}^2_{l_{PCANet}}) \tag{6}$$

Next, these binarized images are partitioned into B blocks and a histogram with 2^{L_2} bins is obtained. Finally, the histograms from all the B blocks are concatenated to form a feature vector from the amalgamted view of the images in Eq. 7.

$$f_{i_{PCANet}} = [Bhist(\boldsymbol{I}^1_{i_{PCANet}}), ..., Bhist(\boldsymbol{I}^{L_1}_{i_{PCANet}})]^T \in \mathbb{R}^{(2^{L_2})L_1 B} \tag{7}$$

[3]The operator $\lceil z \rceil$ gives the smallest integer greater than or equal to z.

Algorithm 1 Left One Mode Out Orthogonal Iteration, *LoMOI*

Input: n-mode tensor $\mathcal{X} \in \mathbb{R}^{i_1, i_2, \ldots, i_n}$; factorization ranks for each mode of the tensor $[r_1 \ldots r_{m-1}, r_{m+1} \ldots r_n]$, where $r_k \leq i_k \forall\ k \in 1, 2, \ldots, n$ and $k \neq m$; factorization error-tolerance ε, and Maximum allowable iterations $= Maxiter$, m = mode to discard while factorizing
for $i = 1, 2, \ldots, n$ and $i \neq m$ **do**
  $\boldsymbol{X}_i \leftarrow$ unfold tensor $\mathcal{X}$ on mode-i
  $\boldsymbol{U}^{(i)} \leftarrow r_i$ left singular vectors of $\boldsymbol{X}_i$ ▷ extract leading r_i matrix factors
$\mathcal{G} \leftarrow \mathcal{X} \times_1 (\boldsymbol{U}^{(1)})^T \ldots \times_{m-1} (\boldsymbol{U}^{(m-1)})^T \times_{m+1} (\boldsymbol{U}^{(m+1)})^T \ldots \times_n (\boldsymbol{U}^{(n)})^T$ ▷ Core tensor
$\hat{\mathcal{X}} \leftarrow \mathcal{G} \times_1 \boldsymbol{U}^{(1)} \ldots \times_{m-1} \boldsymbol{U}^{(m-1)} \times_{m+1} \boldsymbol{U}^{(m+1)} \times_n \boldsymbol{U}^{(n)}$ ▷ reconstructed tensor obtained by multilinear product of the core-tensor with the factor-matrices; Eq. 8.
$loss \leftarrow \|\mathcal{X} - \hat{\mathcal{X}}\|$ ▷ decomposition loss
$count \leftarrow 0$
while $[(loss \geq \varepsilon)\ Or\ (Maxiter \leq count)]$ **do** ▷ loop until convergence
  for $i = 1, 2, \ldots, n$ and $i \neq m$ **do**
    $\mathcal{Y} \leftarrow \mathcal{X} \times_1 (\boldsymbol{U}^{(1)})^T \ldots \times_{(i-1)} (\boldsymbol{U}^{(i-1)})^T \times_{(i+1)} (\boldsymbol{U}^{(i+1)})^T \ldots \times_n (\boldsymbol{U}^{(n)})^T$ ▷ obtain the variance in mode-i
    $\boldsymbol{Y}_i \leftarrow$ unfold tensor $\mathcal{Y}$ on mode-i
    $\boldsymbol{U}^{(i)} \leftarrow \mathbf{r}_i$ left singular vectors of $\boldsymbol{Y}_i$
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 (\boldsymbol{U}^{(1)})^T \ldots \times_{(m-1)} (\boldsymbol{U}^{(m-1)})^T \times_{(m+1)} (\boldsymbol{U}^{(m+1)})^T \ldots \times_n (\boldsymbol{U}^{(n)})^T$
  $\hat{\mathcal{X}} \leftarrow \mathcal{G} \times_1 \boldsymbol{U}^{(1)} \ldots \times_{(m-1)} \boldsymbol{U}^{(m-1)} \times_{(m+1)} \boldsymbol{U}^{(m+1)} \ldots \times_n \boldsymbol{U}^{(n)}$
  $loss \leftarrow \|\mathcal{X} - \hat{\mathcal{X}}\|$
  $count \leftarrow count + 1$
Output: $\hat{\mathcal{X}}$ the reconstructed tensor and $[\boldsymbol{U}^{(1)} \ldots \boldsymbol{U}^{(m-1)}, \boldsymbol{U}^{(m+1)} \ldots \boldsymbol{U}^{(n)}]$ the factor matrices

This block-wise encoding process encapsulates the L_1 images from Eq. 6 into a single feature vector which can be utilized for any machine learning task like clustering or classification.

B. Tensor Preliminaries

Tensors are simply multi-mode arrays or higher-order[4] matrices of dimension > 2. In this paper, the vectors are denoted as $\boldsymbol{x}$ are called first-order tensors, whereas the matrices are denoted as $\boldsymbol{X}$ are called second-order tensors. Analogously, matrices of order-3 or higher are called tensors and are denoted as $\mathcal{X}$. A few important multilinear algebraic operations utilized in this paper are described below.

a) Matriziation: also known as tensor unfolding, is the operation to rearrange the elements of an n-mode tensor $\mathcal{X} \in \mathbb{R}^{i_1 \times i_2 \ldots \times i_N}$ as matrix $\boldsymbol{X}_{(n)} \in \mathbb{R}^{i_n \times j}$ on the chosen mode *n*, where $j = \big(i_1\ \ldots \times i_{n-1} \times i_{n+1} \ldots \times i_N\big)$.

b) n-mode Product: the product of an n-mode tensor $\mathcal{X} \in \mathbb{R}^{i_1 \ldots \times i_{m-1} \times i_m \times i_{m+1} \ldots \times i_n}$ and a matrix $\boldsymbol{A} \in \mathbb{R}^{j \times i_n}$ is denoted as $\mathcal{X} \times_n \boldsymbol{A}$. The resultant of this product is also a tensor $\mathcal{Y} \in \mathbb{R}^{i_1 \times i_2 \times i_{n-1} \times j \times i_{n+1} \ldots \times i_n}$ which can also be expressed through matricized tensor as $\boldsymbol{Y}_{(n)} = \boldsymbol{A}\boldsymbol{X}_{(n)}$.

c) Tensor Decomposition: Tensor decomposition is a form of generalized matrix factorization for approximating multimode tensors. The factorization an n-mode tensor $\mathcal{X} \in \mathbb{R}^{i_1 \times i_2 \ldots \times i_n}$ obtains two sub components: 1) $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \ldots \times r_n}$ which is a lower dimensional tensor called the *core-tensor* and, 2) $\boldsymbol{U}^{(j)} \in \mathbb{R}^{r_n \times i_n} \forall j = [1, n]$ which are matrix factors associated with each mode of the tensor. The entries in the *core-tensor* $\mathcal{G}$ signify the interaction level between tensor elements. The factor matrices $\boldsymbol{U}^{(n)}$ are analogous to *principal components* associated with the respective mode-n. This scheme of tensor factorization falls under the *Tucker* family of tensor decomposition [22]. The original tensor $\mathcal{X}$ can be reconstructed by taking the n-mode product of the *core-tensor* and the factor matrices as in Eq. 8.

$$\mathcal{G} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \ldots \times_N \boldsymbol{U}^{(n)} \approx \mathcal{X} \tag{8}$$

The advantages of *Tucker* based factorization methods are already studied in several domains such as computer vision, [14], data mining [23], and signal processing [24], [22]. However, in this research, we factorize tensor to obtain weights of convolution-tensorial filters for TFNet by devising our custom tensor factorization scheme which we call as Left one Mode Out Orthogonal Iteration (*LoMOI*) presented in Alg. 1.

III. The Tensor Factorization Network

The development of *Tensor Factorization Network* (TFNet) is motivated to reduce the loss of spatial information occurring in the PCANet while vectorizing image patches. However, this transformation of the data is inherent while extracting the *principal components* which destroys the geometric structure of the object encapsulated in the data which is proven beneficial in many image classification tasks [14], [15], [16]. Furthermore, the vectorization of the data results in high dimensional vectors and generally requires more computational resources. Motivated by the above shortcomings with the PCANet, we propose the TFNet. The TFNet preserves the spatial structure of the data while obtaining weights of its convolution-tensor filters. The unsupervised feature extraction procedure with minutiae view of the data is detailed in the next subsection.

A. The First Layer

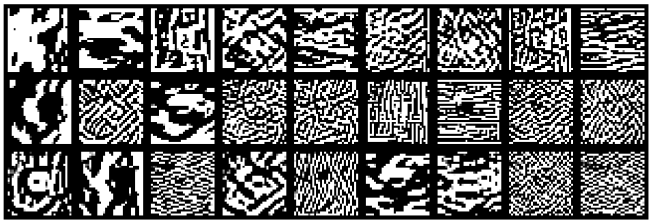
Similar to the first layer in PCANet, we begin by collecting all overlapping patches of size $k_1 \times k_2$ around each pixel from the image $\boldsymbol{I}_i$. However, contrary to PCANet the spatial structure of these patches are preserved and instead of matrix and we obtain a 3-mode tensor $\mathcal{X}_i \in \mathbb{R}^{k_1 \times k_2 \times \tilde{m}\tilde{n}}$. The mode-1 and mode-2 of this tensor represent the row-space, and the column-space spanned by the pixels in the image. Whereas the mode-3 of this tensor represents the total number of image patches obtained from the input image. Iterating this process for all the training images, we obtain $\mathcal{X} \in \mathbb{R}^{k_1 \times k_2 \times N\tilde{m}\tilde{n}}$ as our final patch-tensor. The matrix factors utilized to generate our convolution-tensorial filters for to the first two modes of $\mathcal{X}$ are obtained by utilizing our custom-designed *LoMOI* (presented in Alg. 1) in Eq. 9.

$$[\hat{\mathcal{X}}, \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}] \leftarrow LoMOI(\mathcal{X}, r_1, r_2) \tag{9}$$

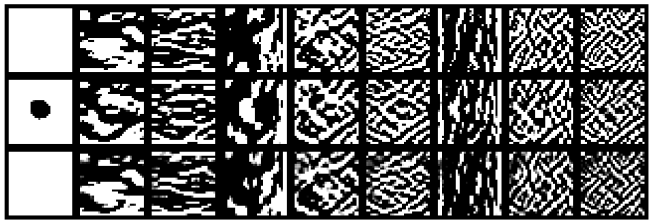
where $\hat{\mathcal{X}} \in \mathbb{R}^{r_1 \times r_2 \times N\tilde{m}\tilde{n}}$, $\boldsymbol{U}^{(1)} \in \mathbb{R}^{k_1 \times r_1}$, and $\boldsymbol{U}^{(2)} \in \mathbb{R}^{k_2 \times r_2}$. We discard obtaining the matrix factors from mode-3 of tensor $\mathcal{X}$ (which is $\boldsymbol{X}_3$) as this is equivalent to the transpose of the patches matrix $\boldsymbol{X}$ in layer 1 of the PCANet which is not factorized in the PCANet while obtaining weights for its convolution filters. Moreover, the matrix factors for this mode span the sample space of the data which is trivial. A total of $L_1 = r_1 \times r_2$ convolution-tensor filters are obtained from the factor matrices $\boldsymbol{U}^{(1)}$ and $\boldsymbol{U}^{(2)}$ as in Eq. 10.

$$\boldsymbol{W}^1_{l_{TFNet}} = \boldsymbol{U}^{(1)}_{(:,i)} \otimes \boldsymbol{U}^{(2)}_{(:,j)} \in \mathbb{R}^{k_1 \times k_2} \tag{10}$$

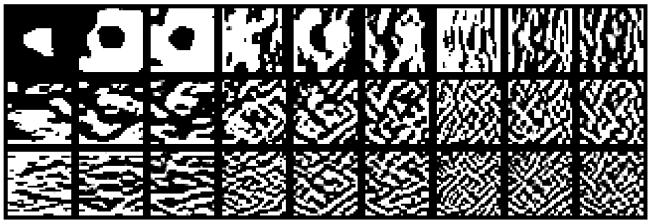
[4] Also known as modes (dimensions) of a tensor and are analogous to rows and columns of a matrix.



(a) Convolution responses from the PCANet where each convolution response is visually distinct from the rest of the responses. This illustrates extraction of unique information with the amalgamated view of the data.



(b) Convolution responses from the MPCANet. Although the output is less diversified than TFNet, visual resemblance is partially observable.



(c) Convolution responses from our TFNet where the visual resemblance is observed in a sequence of three responses. This illustrates extraction of common information with minutiae view of the data.

Fig. 1: Comparison of convolution outputs from Layer1 in PCANet, MPCANet and TFNet on CIFAR-10 dataset. These plots demonstrate the contrast between the kinds of information obtained with the amalgamated and the minutiae view of the data.

where '$\otimes$' is the *outer*-product between two vectors, $i = [1, r_1]$, $j = [1, r_2]$, $l = [1, L_1]$, and $\boldsymbol{U}^{(m)}_{(:,i)}$ represents 'i^{th}' column of the 'm^{th}' factor matrix. Importantly, our convolution-tensorial filters do not require any explicit reshaping as the *outer*-product between two vectors naturally results in a matrix. Therefore, we can straightforwardly convolve the input images with our obtained convolution-tensorial filters as described in Eq. 11 where $i = [1, N]$ and $l = [1, L_1]$ are the image and filter indices respectively.

$$\boldsymbol{I}^{l}_{i_{TFNet}} = \boldsymbol{I}_i * \boldsymbol{W}^{1}_{l_{TFNet}} \tag{11}$$

However, whenever the data is an RGB-image, each extracted patch from the image is a 3-order tensor $\mathcal{X} \in \mathbb{R}^{k_1 \times k_2 \times 3}$ (i.e., *RowPixels*×*ColPixels*×Color). After collecting patches from all the training images, we obtain a 4-mode tensor as $\mathcal{X} \in \mathbb{R}^{k_1 \times k_2 \times 3 \times N\tilde{m}\tilde{n}}$ which is decomposed by utilizing *LoMOI* ($[\hat{\mathcal{X}}, \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \boldsymbol{U}^{(3)}] \leftarrow LoMOI(\mathcal{X}, r_1, r_2, r_3)$) for obtaining the convolution-tensorial filters in Eq. 12.

$$\boldsymbol{W}^{1}_{l_{TFNet}} = \boldsymbol{U}^{(1)}_{(:,i)} \otimes \boldsymbol{U}^{(2)}_{(:,j)} \otimes \boldsymbol{U}^{(3)}_{(:,k)} \tag{12}$$

where $i \in [1, r_1]$, $j \in [1, r_2]$, and $k \in [1, r_3]$.

B. The Second Layer

Similar to the first layer, we extract overlapping patches from the input images and zero-center them to build a 3-mode patch-tensor denoted as $\mathcal{Y} \in \mathbb{R}^{k_1 \times k_2 \times NL_1\tilde{m}\tilde{n}}$ which is decomposed as $[\hat{\mathcal{Y}}, \boldsymbol{V}^{(1)}, \boldsymbol{V}^{(2)}] \leftarrow LoMOI(\mathcal{Y}, r_1, r_2)$ to obtain the convolution-tensor filters for layer 2 in Eq. 13.

$$\boldsymbol{W}^{2}_{l_{TFNet}} = \boldsymbol{V}^{(1)}_{(:,i)} \otimes \boldsymbol{V}^{(2)}_{(:,j)} \in \mathbb{R}^{k_1 \times k_2} \tag{13}$$

where, $\hat{\mathcal{Y}} \in \mathbb{R}^{r_1 \times r_2 \times NL_1\tilde{m}\tilde{n}}$, $\boldsymbol{V}^{(1)} \in \mathbb{R}^{k_1 \times r_1}$, and $\boldsymbol{V}^{(2)} \in \mathbb{R}^{k_2 \times r_2}$, $i = [1, r_1]$, $j = [1, r_2]$, and $l = [1, L_2]$. We, now convolve each of the L_1 input images from the first layer with the convolution-tensorial filters obtained as below in Eq. 14.

$$\boldsymbol{O}^{l}_{i_{TFNet}} = \boldsymbol{I}^{l}_{i_{TFNet}} * \boldsymbol{W}^{2}_{l_{TFNet}}, \quad l = 1, 2, ..., L_2 \tag{14}$$

The number of output images obtained here is equal to $L_1 \times L_2$ which is identical to the number of images obtained at layer 2 of PCANet. Finally, we utilize the output layer of PCANet (Sec. II-A3) to obtain the feature vectors from the minutiae view of the image in Eq. 15.

$$\begin{aligned} \boldsymbol{I}^{l}_{i_{TFNet}} &= \sum_{l=1}^{L_2} 2^{l-1} H(\boldsymbol{O}^{2}_{l_{TFNet}}) \\ f_{i_{TFNet}} &= [Bhist(\boldsymbol{I}^{1}_{i_{TFNet}}), ..., Bhist(\boldsymbol{I}^{L_1}_{i_{TFNet}})]^T \in \mathbb{R}^{(2^{L_2})L_1 B} \end{aligned} \tag{15}$$

Despite having close resemblance between the feature extraction mechanism of the PCANet and the TFNet, these two networks capture visibly distinguishable features from the two view of the images as shown in Fig. 1. These plots are obtained by convolving image of a *cat* with the convolution filters obtained in the first layer of the networks.

Undoubtedly, each of the L_1 convolution responses within the PCANet is visibly distinct. Whereas the convolution responses within the TFNet shows visual similarity, i.e., the images in a triplet sequence show similarity consecutively. These plots demonstrate that the TFNet emphasize mining the *common* information from the minutiae view of the data. Whereas the PCANet emphasizes mining the *unique* information from the amalgamated view of the data. Both these kinds of information are proven beneficial for classification in [17], [15] and motivate the development of *HybridNet*.

Besides, we also present convolution responses of MPCANet [25] in the comparisons. MPCANet employs tensorized-convolution but differs from our TFNet in two

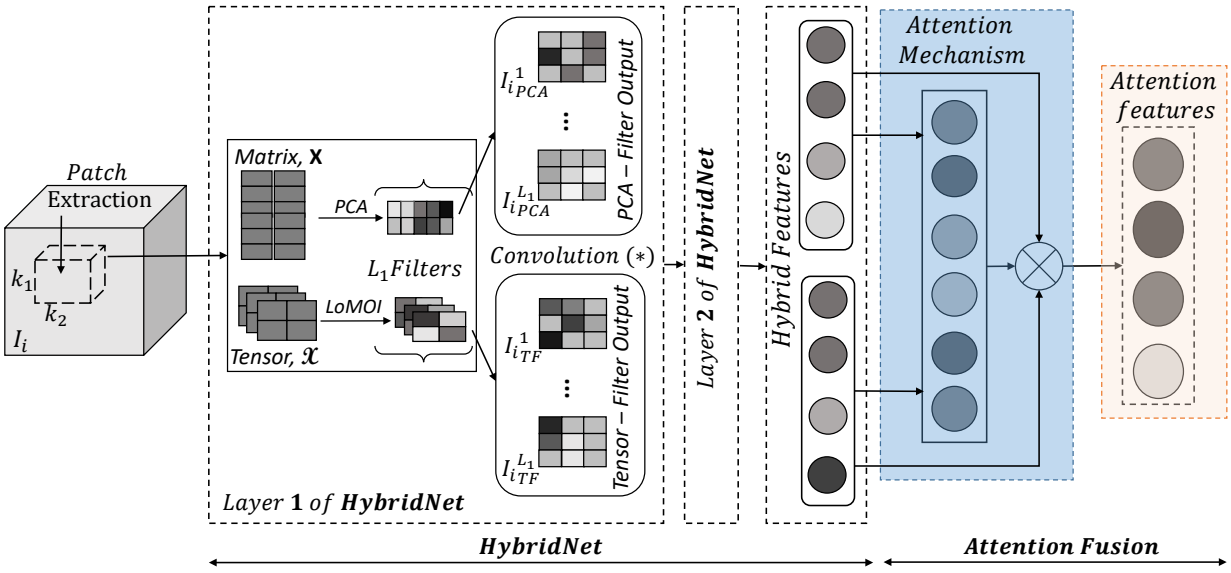


Fig. 2: Workflow of the proposed *Attn-HybridNet* model.

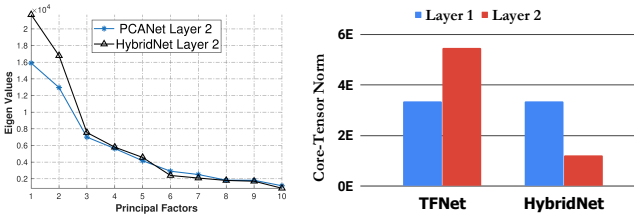
ways: 1) construction of convolution kernels and 2) convolution operation. Technically, the convolution kernel and convolution operation in MPCANet in [25] (and its predecessor in [26]) are combined together as conventional tensor operations, obtaining a) factor matrices for each mode from the patch-tensor and b) n-mode products of factor matrices with the patch-tensor. In other words, the convolution in MPCANet is a n-mode product of the patch-tensor and factor matrices, whereas in TFNet the convolution kernels are obtained by performing outer-procut of factor matrices in Eq. 10 followed by convolution in Eq. 11. From Figure 1, it is visible that the convolution responses in MPCANet is less diversified than those of TFNet although visible resemblance is still observable. We believe that this is because of how the convolution filters and convolution operation are performed in MPCANet as analyzed above.

IV. The Hybrid Network

The PCANet and the TFNet extract contrasting information from the amalgamated view and the minutiae view of the data, respectively. However, we hypothesize that the information from both these views are essential as they conceal complementary information and that their integration can enhance the performance of classification systems. Motivated by the above, we propose the *HybridNet*, which simultaneously extracts information from both views of the data and is detailed in the next subsection. However, for ease of understanding, we illustrate the complete procedure of feature extraction with *Attn-HybridNet* in Fig. 2.

A. The First Layer

Similar to the previous networks, we begin the feature extraction process by collecting all overlapping patches of size $k_1 \times k_2$ around each pixel from the image I_i. Importantly, the first layer of *HybridNet* consists of image-patches expressed both as tensors $\mathcal{X} \in \mathbb{R}^{k_1 \times k_2 \times 3 \times N\tilde{m}\tilde{n}}$ and matrices



(a) Comparison of eigenvalues between networks

(b) Comparison of core-tensor strength between networks

Fig. 3: Comparison of factorization strength in Layer 2 of the PCANet, TFNet and *HybridNet* on CIFAR-10 dataset

$\boldsymbol{X} \in \mathbb{R}^{k_1 k_2 \times N\tilde{m}\tilde{n}}$ which are utilized for obtaining weights of convolution filters in layer 1 of *HybridNet*.

This enables this layer (and the subsequent layers) of *HybridNet* to learn superior filters as they perceive more information from both views of the data. The weights for the *pca*-filters are obtained as the principal-eigenvectors as $\boldsymbol{W}_{l_{PCA}}^1 = mat_{k_1,k_2}(ql(\boldsymbol{X}\boldsymbol{X}^T))$, and the weights for convolution-tensor filters are obtained by utilizing *LoMOI* as $\boldsymbol{W}_{l_{TF}}^1 = \boldsymbol{U}_{(:,i)}^{(1)} \otimes \boldsymbol{U}_{(:,j)}^{(2)} \otimes \boldsymbol{U}_{(:,k)}^{(3)}$. Furthermore, the output from this layer is obtained by convolving input images with a) the PCA-filters and b) the convolution-tensorial filters in Eq. 16. This injects more diversity to the output in succeeding layer of *HybridNet*.

$$\begin{aligned} \boldsymbol{I}_{i_{PCA}}^l &= \boldsymbol{I}_i * \boldsymbol{W}_{l_{PCA}}^1 \\ \boldsymbol{I}_{i_{TF}}^l &= \boldsymbol{I}_i * \boldsymbol{W}_{l_{TF}}^1 \end{aligned} \tag{16}$$

Since we obtain of L_1 *pca* filters and L_1 convolution-tensor filters, a total of $2 \times L_1$ outputs are obtained in this layer.

B. The Second Layer

Similar to the first layer, we begin with collecting all overlapping patches of size $k_1 \times k_2$ around each pixel from the images. However, contrary to the above layer, the weights of the *pca*-filters $\boldsymbol{W}_{l_{PCA}}^2$ and convolution-tensor filters $\boldsymbol{W}_{l_{TF}}^2$

Algorithm 2 The *HybridNet* Algorithm

Input: $\boldsymbol{I}_i, i = 1, 2, ..., n$ n is the total number of training images, $L = [l_1, l_2, ...l_D]$ the number of filters in each layer,k_1 and k_2 the patch-size, B, D = the depth of the network.
for $i = 1, 2, ..., n$ **do** ▷ DO for each image in the first convolution layer.
  $\boldsymbol{X} \leftarrow$ extract patches of size $k_1 \times k_2$ around each pixel of I_i ▷ mean centered and vectorized.
  $\mathfrak{X} \leftarrow$ extract patches of size $k_1 \times k_2$ around each pixel of I_i ▷ mean centred but retain their spatial shape.
$\boldsymbol{W}_{PCA} \leftarrow$ obtain PCA filters by factorizing $\boldsymbol{X}$.
$\boldsymbol{W}_{TF} \leftarrow$ obtain tensor filters by factorizing $\boldsymbol{X}$ with LoMOI Algo. 1.
for $i = 1, 2, ..., n$ **do**
  $\boldsymbol{I}^1_{i_{PCA}} \leftarrow \boldsymbol{I}_i * \boldsymbol{W}_{PCA}$ ▷ store convolution with pca filters.
  $\boldsymbol{I}^1_{i_{TF}} \leftarrow \boldsymbol{I}_i * \boldsymbol{W}_{TF}$ ▷ store convolution with tensorial filters.
for $l = 2, ..., D$ **do** ▷ DO for the remaining convolution layers.
  $\boldsymbol{I} \leftarrow [\boldsymbol{I}^{(l-1)}_{PCA}, \boldsymbol{I}^{(l-1)}_{TF}]$ ▷ Utilize both views together.
  for $i = 1, 2, ..., \bar{n}$ **do** ▷ $\bar{n} = 2 \times n \times l_{l-1}$.
    $\boldsymbol{X} \leftarrow$ extract patches of size $k_1 \times k_2$ around each pixel of $\boldsymbol{I}_i$
    $\mathfrak{X} \leftarrow$ extract patches of size $k_1 \times k_2$ around each pixel of $\boldsymbol{I}_i$
  $\boldsymbol{W}^l_{PCA} \leftarrow$ obtain PCA filters by factorizing $\boldsymbol{X}$.
  $\boldsymbol{W}^l_{TF} \leftarrow$ obtain tensor filters by factorizing $\boldsymbol{X}$ with LoMOI Alg. 1.
  for $i = 1, 2, ..., \bar{n}$ **do** ▷ $\bar{n} = n \times l_{l-1}$.
    $\boldsymbol{I}^l_{i_{PCA}} = \boldsymbol{I}^{(l-1)}_{i_{PCA}} * \boldsymbol{W}^1_{l_{PCA}}$
    $\boldsymbol{I}^l_{i_{TF}} = \boldsymbol{I}^{l-1}_{i_{TF}} * \boldsymbol{W}^1_{l_{TF}}$
for $i = 1, 2, ..., n$ **do** ▷ DO for each image.
  $\boldsymbol{I}_{i_{PCA}} = \sum_{k=1}^{l_d} 2^{l-1} H(\boldsymbol{I}^d_{k_{PCA}})$ ▷ Binarize and accumulate output from all convolution layers.
  $\boldsymbol{I}_{i_{TF}} = \sum_{k=1}^{l_d} 2^{l-1} H(\boldsymbol{I}^d_{k_{TF}})$
  $f_{i_{PCA}} \leftarrow Bhist(\boldsymbol{I}_{i_{TF}})$ ▷ create block-wise histogram.
  $f_{i_{TF}} \leftarrow Bhist(\boldsymbol{I}_{i_{TF}})$
Output: features from the amalgamated mode as f_{PCA}, and the minutiae mode as f_{TF}.

are learned from the data obtained by convolving input images with the *pca* filters and the convolution-tensor filters i.e. both $\boldsymbol{I}_{i_{PCA}}$ and $\boldsymbol{I}_{i_{TF}}$. Hence both the patch-matrix $\boldsymbol{Y} \in \mathbb{R}^{k_1 k_2 \times 2L_1 N \tilde{m} \tilde{n}}$ and the patch-tensor $\mathcal{Y} \in \mathbb{R}^{k_1 \times k_2 \times 2L_1 N \tilde{m} \tilde{n}}$ contain image patches obtained from $[\boldsymbol{I}_{i_{PCA}}, \boldsymbol{I}_{i_{TF}}]$. This enables the hybrid filters to assimilate more variability present in the data while obtaining weights of their convolution filters as evident in Fig. 3.

The plot in Fig. 3(a) compares the eigenvalues obtained in layer 2 (we exclude eigenvalues from layer 1 as they completely overlap as their expected behavior). The leading eigenvalues obtained in layer 2 of the *HybridNet* by $principal\ components$ has much higher magnitude than the corresponding eigenvalues obtained by $principal\ components$ in PCANet. This demonstrates that the pca filters in the *HybridNet* capture more variability than those in the PCANet.

Similarly, Fig 3(b) compares the core-tensor strength in different layers of the *HybridNet* and the TFNet. We plot the norm of the core-tensor for both the networks as the values in the core-tensor is analogous to eigenvalues for higher-order matrices, and its norm signifies the compression strength of the factorization [27]. Again, the norm of the core-tensor in layer 2 of *HybridNet* is much lower than that of the TFNet, suggesting relatively higher factorization strength in *HybridNet*. Besides, as expected, the norm of the core-tensor in layer 1 for both the networks coincides and signifies equal factorization strength at this layer. Consequently, this leads to attainment of better-disentangled feature representations with the *HybridNet* and hence enhances its generalization performance over the PCANet and the TFNet by integrating information from the two views of the data.

In the second layer, the weights of *pca* filters are obtained by *principal components* as $\boldsymbol{W}^2_{l_{PCA}} = mat_{k_1,k_2}(ql(\boldsymbol{Y}\boldsymbol{Y}^T))$ and the weights for convolution-tensor filters are obtained as $\boldsymbol{W}^2_{l_{TF}} = \boldsymbol{V}^{(1)}_{(:,i)} \otimes \boldsymbol{V}^{(2)}_{(:,j)}$, where the matrix factors are obtained using *LoMOI* $[\hat{\mathcal{Y}}, \boldsymbol{V}^{(1)}, \boldsymbol{V}^{(2)}] \leftarrow LoMOI(\mathcal{Y}, r_1, r_2)$. Analogous to the previous networks, the output images from this layer of *HybridNet* are obtained by a) convolving the L_1 images corresponding to the output from the PCA-filters in the first layer with the L_2 pca filters obtained in the second layer (Eq. 17), and b) convolving the L_1 images corresponding to the output from the convolution-tensorial filters in the first layer with the L_2 convolution-tensorial filters obtained in the second layer (Eq. 18). This generates a total of $2 \times L_1 \times L_2$ output images in this layer.

$$\boldsymbol{O}^l_{i_{PCA}} = \boldsymbol{I}^l_{i_{PCA}} * \boldsymbol{W}^2_{l_{PCA}} \tag{17}$$

$$\boldsymbol{O}^l_{i_{TF}} = \boldsymbol{I}^l_{i_{TF}} * \boldsymbol{W}^2_{l_{TF}} \tag{18}$$

The output images obtained from the *pca*-filters ($\boldsymbol{O}^l_{i_{PCA}}$) in layer 2 are then processed with the output layer of the PCANet (Sec. II-A3) to obtain $f_{i_{PCA}}$ as the information from amalgamated view of the image. Similarly, the output images obtained from the convolution-tensor filters ($\boldsymbol{O}^l_{i_{TF}}$) are processed to obtain $f_{i_{TF}}$ as the information from minutiae view of the image. Finally, these two kinds information are concatenated to obtain the hybrid features as in Eq. 19.

$$f_{i_{hybrid}} = [f_{i_{PCA}}\ f_{i_{TF}}] \in \mathbb{R}^{(2^{L_2})2L_1 B} \tag{19}$$

The whole procedure of obtaining hybrid features is detailed in Alg. 2. Although these hybrid features bring the best of both the common and the unique information obtained respectively from the two views of the data, they still suffer from feature redundancy problems induced by the spatial pooling operation in the output layer. To alleviate this drawback, we propose the *Attn-HybridNet*, which further enhances the discriminability of the hybrid features.

V. Attention-based fusion - Attn-HybridNet

Our proposed *HybridNet* eradicates the loss of information by integrating the learning scheme of PCANet and TFNet thus obtaining superior features than either of the networks. However, the feature encoding scheme in the output layer is elementary and induces redundancy in the feature representations [12], [28]. Moreover, the generalized spatial pooling operation in the output layer is unable to accommodate the spatial structure of the natural images, i.e., it is more effective for aligned images dataset like face and handwritten digits than for object recognition dataset. Simply, the design of the output layer is ineffectual to obtain utmost feature representation on object recognition datasets resulting in performance degradation with the *HybridNet*. Moreover, efficient ways to alleviate this drawback with the output layer are not addressed in the literature, which necessitates the development of our proposed attention-based fusion scheme i.e. the *Attn-HybridNet*.

Our proposed attention-based fusion scheme is presented in Alg. 3, where $f_{hybrid} \in \mathbb{R}^{N \times (2^{L_2})L_1 B \times 2}$ are the hybrid feature vectors obtained with the *HybridNet*, $w \in \mathbb{R}^d$ is the feature level context vector of dimension $d << (2^{L_2})L_1 B$,

Algorithm 3 The *Attn-HybridNet* Algorithm

Input: $f_{hybrid} = [f_{PCA};\ f_{TF}] \in \mathbb{R}^{N\times(2^{L_2})L_1B\times 2}$ the hybrid feature vectors from the training images; $y = [0, 1, ..., C]$ ground truth of training images, dimensionality of feature level context vector $w \in \mathbb{R}^d$, where $d << \mathbb{R}^{(2^{L_2})L_1B}$.
randomly initialize $\boldsymbol{W}$, f_c, and w
$loss \leftarrow 1000$ ▷ arbitrary number to start training
do
  $[f_{batch}, y_{batch}] \leftarrow$ sample batch ($[f_{\textbf{hybrid}}, y]$)
  $\boldsymbol{P}_F \leftarrow tanh(W.f_{batch})$ ▷ get the hidden representation of the hybrid features
  $\alpha = softmax\big(w^T.\boldsymbol{P}_F\big)$ ▷ measure and normalize the importance
  $F_{attn} = f_{batch}.\alpha^T$ ▷ perform attention fusion
  $\hat{y} \leftarrow f_c(F_{attn})$ ▷ fully connected layer
  $loss \leftarrow LogLoss(y_{batch}, \hat{y}_{batch})$ ▷ compute loss for optimizing parameters
  back-propagate loss for optimizing $\boldsymbol{W}$, f_c, and w.
while $[(loss \geq \varepsilon)]$ ▷ loop until convergence
Output: parameters to perform attention fusion $\boldsymbol{W}$, f_c, and $w \in \mathbb{R}^d$

$\alpha^T \in \mathbb{R}^2$ is the normalized importance weight vector for combining the two kinds of information with attention fusion, and $F_{attn} \in \mathbb{R}^{(2^{L_2})L_1B}$ are the attention features. The fully connected layers i.e. $\boldsymbol{W} \in \mathbb{R}^{d\times(2^{L_2})L_1B}$ and f_c are utilized to obtain hidden representations of features while performing attention fusion.

A few numerical optimization based techniques proposed in [12], [13] exist for alleviating the feature redundancy from architectures utilizing generalized spatial pooling layers. However, these techniques require grid search between the dictionary size (number of convolution filters in our case) and the pooling blocks in the output layer while performing optimization. Besides, the transition to prune filters from a single-layer networks to multi-layer network is not smooth in these techniques. A major difference between our proposed *Attn-HybridNet* and the existing proposal in [12], [13] is that we reduce the feature redundancy by performing feature selection with attention-based fusion scheme, whereas the existing techniques prune the filters to eliminate the feature redundancy. Therefore, our proposed *Attn-HybridNet* is superior to these existing techniques as it decouples the two sub-processes, i.e., information discovery with convolution layers and feature aggregation in the pooling layer while alleviating the redundancy exhibiting in the feature representations.

The discriminative features obtained by *Attn-HybridNet* i.e. F_{attn} are utilized with $softmax$-layer for classification, where the parameters in the proposed fusion scheme (i.e., $\boldsymbol{W}$, f_c and w) are optimized via gradient-descent on the classification loss. This simple yet effective scheme substantially enhances the classification performance by obtaining highly discriminative features. Comprehensive experiments are conducted in this regard to demonstrate the superiority of *Attn-HybridNet* detailed in Sec. VI.

A. Computational Complexity

To calculate the computational complexity of *Attn-HybridNet*, we assume the *HybridNet* is composed of two-layers with a patch size of $k_1 = k_2 = k$ in each layer followed by our attention-based fusion scheme.

In each layer of the *HybridNet*, we have to compute the time complexities arising from learning convolution weights from the two views of the data. The formation of the zero-centered patch-matrix $\boldsymbol{X}$ and zero-centered patch-tensor $\boldsymbol{\mathcal{X}}$ have identical complexities as $k^2(1+\tilde{m}\tilde{n})$. The complexity of eigen-decomposition for patch-matrix and tensor factorization with *LoMOI* for patch-tensor are also identical and equal to $\mathcal{O}((k^2)^3)$, where k is a whole number < 7 in our experiments. Further, the complexity for convolving images with the convolution filters at stage i requires $L_i k^2 mn$ flops. The conversion of L_2 binary bits to a decimal number in the output layer costs $2L_2\tilde{m}\tilde{n}$, where $\tilde{m} = m - \lceil \frac{k}{2} \rceil, \tilde{n} = n - \lceil \frac{k}{2} \rceil$ and the naive histogram operation for this conversion results in complexity equal to $\mathcal{O}(mnBL_2log2)$.

The complexity of performing matrix multiplication in *Attn-HybridNet* is $\mathcal{O}\Big(2L_1B\big(d(1+2^{L_2})+2^{L_2}\big)\Big)$ which can be efficiently handled with modern deep learning packages like Tensorflow [29] for stochastic updates. To optimize the parameters in the attention-based fusion scheme ($\boldsymbol{W}$, f_c, and w), we back-propagate the loss through the attention network until convergence of the error on the training features.

VI. Experiments and Results

A. Experimental Setup

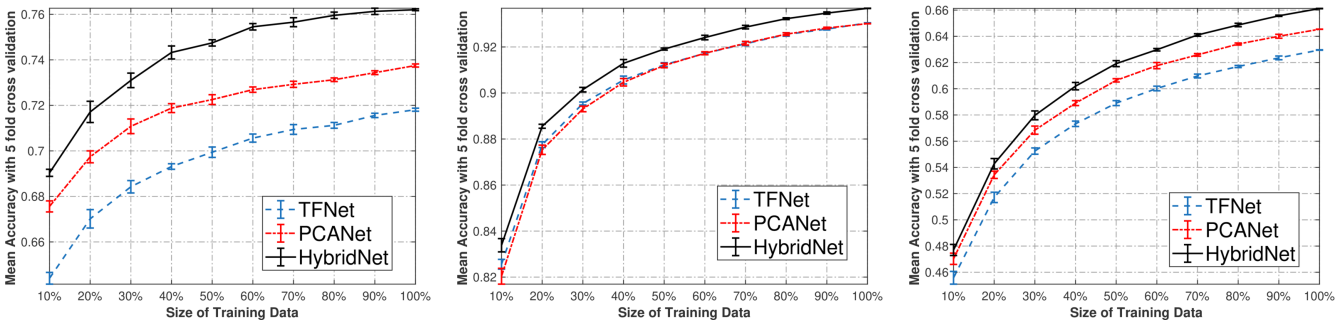
In our experiments, we utilized a two-layer architecture for each of the networks in comparison, while the number of convolution filters in the first and the second layer are optimized via cross-validation on the training datasets. The dimensionality of the feature vectors extracted from PCANet and TFNet is then $BL_1 2^{L_2}$, where L_1 and L_2 are the number of convolution filters in layer 1 and layer 2 respectively. The dimensionality of feature vector with *HybridNet* is then $2BL_1 2^{L_2}$. We utilized *Linear-SVM* [30] as the classifier incorporating features obtained with the PCANet, TFNet, and the *HybridNet*.

The attention-based fusion scheme is performed by following the procedure as described in Alg. 3, where we obtained the optimal attention dimension on training data for the context level feature vector $w \in \mathbb{R}^d$ in $[10, 50, 100, 150, 200, 400]$. The obtained attention features i.e. $F_{attn} \in R^{BL_1 2^{L_2}}$ are utilized with $softmax$-layer for classification. The parameters of attention-based fusion scheme ($\boldsymbol{W}$, f_c, and w) are optimized via back-propagation on the classification loss implemented in TensorFlow [29]. We observed that the attention-network's optimization took less than 15 epochs for convergence on all the datasets utilized in this paper.

B. Datasets

The details of datasets and hyper-parameters are as below:

1 MNIST variations [31], which consist of gray scale handwritten digits of size 28×28 with controlled factors of variations such as background noise, rotations, etc. Each variation contains $10K$ training and $50K$ testing images. We cite the results for baselines techniques like 2-stage ScatNet [32] (ScatNet-2) and 2-stage Contractive auto-encoders [33] (CAE-2) as reported in [10]. The parameters of *HybridNet* (and other networks) are set as $L_1 = 9$, $L_2 = 8$, $k_1 = k_2 = 7$, with a block size of $B = 7\times7$ keeping size of overlapping regions equal to half of the block size for feature pooling.



(a) CIFAR-10 (b) MNIST Rotation (c) MNIST bg-img-rot

Fig. 4: Performance Comparison by varying size of the training data

2 CUReT dataset [34], consists of 61 texture categories, where each category has images of the same material with different poses, illumination, specularity, shadowing, and surface normals. Following the standard procedure in [34], [10] a subset of 92 cropped images were taken from each category and randomly partitioned into train and test sets with a split ratio of 50%. The classification results are averaged over 10 different trails with hyper-parameters as $L_1 = 9$, $L_2 = 8$, $k_1 = k_2 = 5$, block size $B = 50 \times 50$ with overlapping regions equal to half of the block size. Again, we cite the results of the baselines techniques as published in [10].

3 ORL[5] and Extended Yale-B [35] datasets are utilized to investigate the performance for face recognition. The ORL dataset consists of 10 different images of 40 distinct subjects taken at different times, varying the lightning, facial expression, and facial details. The Extended Yale-B dataset consists of face images from 38 individuals under 9 poses and 64 illumination conditions. The images in both the datasets are cropped to size 64×64 pixels followed by unit length normalization. The classification results are averaged over 5 different trails by progressively increasing the number of training examples[6] with hyper-parameters as $L_1 = 9$, $L_2 = 8$, $k_1 = k_2 = 5$ with a non-overlapping block of size $B = 7 \times 7$.

4 CIFAR-10 [11] dataset consists of RGB images of dimensions 32×32 for object recognition consisting of $50K$ and $10K$ images for training and testing respectively. These images are distributed among 10 classes and vary significantly in object positions, object scales, colors, and textures within each class. We varied the number of filters in layer 1 i.e., L_1 as 9 and 27 and kept the number of filters in layer 2 i.e. $L_2 = 8$. The patch-size k_1 and k_2 are kept equal and varied as 5, 7, and 9 with block size $B = 8 \times 8$. Following [10] we also applied spatial pyramid pooling (SPP) [36] to the output layer of *HybridNet* (and similarly to the out layer of other networks). We additionally applied PCA to reduce the dimension of each pooled feature to 100[7]. These features are utilized with Linear-SVM for classification and *Attn-HybridNet* for obtaining attention features F_{attn}.

5 CIFAR-100 [11] dataset closely follows CIFAR-10 dataset and consists 50k training and 10k testing images roughly distributed among 100 categories. We use the same experimental setup as in CIFAR-10 on this dataset.

C. Baselines

On face recognition datasets, we compare the performance of *Attn-HybridNet* and *HybridNet* against three recent baselines: Deep-NMF [37], PCANet+ [38], and PCANet-II [39][8]. On MNIST variations and CuReT dataset we select the baselines as in [10]. Besides, the hyper-parameters of these baselines are set equal to those in *HybridNet*.

On CIFAR-10 dataset, we compare our schemes against multiple comparable baselines such as Tiled CNN [40], CUDA-Convnet [41], VGG style CNN (VGG-CIFAR-10 reported by [42]), K-means (tri) [43], Spatial Pyramid Pooling for CNN (SPP) [44], Convolution Bag-of-Features (CBoF) [45], and Spatial-CBoF [4]. Note that, we do not compare our schemes against schemes which aim to either compress deep neural networks or transfer pre-learned CNN filters such as in [46], [47], [48], [49], [50], [51] as these schemes do not train a deep network from scratch whereas our proposed schemes and comparable baselines do so. Besides, we also report the performances of ResNet [19] and DenseNet [52] on CIFAR datasets, as mentioned in their respective publication.

Lastly, we perform a qualitative case study on the CIFAR-10 dataset by studying the performance of baselines and our scheme by varying the size of the training data. Besides, we also studied the effect on the discriminability of hybrid features with attention-based fusion scheme.

VII. Results and Discussions

Our two main contributions in this research are - 1) the integration of information available from both the amalgamated view (i.e., the unique information) and the minutiae view (i.e., the common information), and 2) attention-based

[5]The ORL database is publicly available and can be obtained from the website: http://www.uk.research.att.com/facedatabase.html

[6]The training and test split are obtained from http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

[7]Results does not vary significantly on increasing the projection dimensions.

[8]The paper did not provide its source code and the results are based on our independent implementation of their second order pooling technique.

Parameters				PCANet [10]	TFNet [18]	*HybridNet*	*Attn-HybridNet*
L_1	L_2	k_1	k_2	Error (%)	Error (%)	Error (%)	Error (%)
8	8	5	5	34.80	32.57	31.39	**28.08**
8	8	7	7	39.92	37.19	35.24	**30.94**
8	8	9	9	43.91	39.65	38.04	**35.33**
27	8	5	5	26.43	29.25	23.84	**18.41**
27	8	7	7	30.08	32.57	28.53	**25.67**
27	8	9	9	33.94	34.79	31.36	**27.70**

TABLE II: Classification Error obtained by varying hyper-parameters on CIFAR-10 dataset.

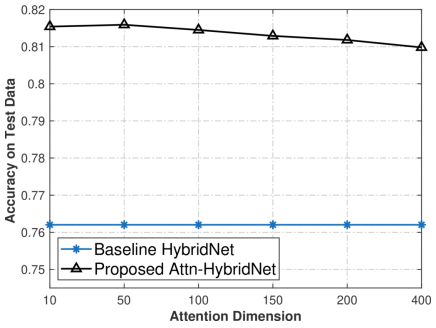


Fig. 5: Accuracy of Attn-HybridNet on CIFAR-10 dataset by varying the dimension of w in Alg. 3.

ORL - Dataset	Number of Training Instances		
	4	6	8
Deep-NMF [37]	9.50 ± 1.94	6.50 ± 2.59	1.75 ± 2.09
PCANet-II [39]	16.16 ± 1.29	7.87 ± 1.29	5.00 ± 2.05
PCANet+ [38]	1.25 ± 0.83	0.50 ± 0.52	0.25 ± 0.55
PCANet [10]	1.75 ± 0.95	0.37 ± 0.34	0.40 ± 0.68
TFNet [18]	1.98 ± 0.54	0.50 ± 0.68	0.25 ± 0.59
HybridNet (proposed)	**1.48 ± 0.72**	**0.25 ± 0.32**	**0.21 ± 0.55**
Attn-HybridNet (proposed)	5.43 ± 0.78	3.11 ± 0.27	1.13 ± 0.31

TABLE III: Classification Error on ORL dataset.

YaleB - Dataset	Number of Training Instances			
	20	30	40	50
Deep-NMF [37]	10.94±0.89	8.03±0.61	5.43±0.94	4.78±0.76
PCANet-II [39]	11.40±0.97	5.54±1.49	2.86±0.35	1.98±0.75
PCANet+ [38]	1.15±0.14	0.28±0.07	0.23±0.14	0.22±0.23
PCANet [10]	1.35±0.17	0.40±0.18	0.38±0.16	0.38±0.13
TFNet [18]	1.97±0.27	0.91±0.28	0.40±0.16	0.42±0.21
HybridNet (proposed)	**1.32±0.35**	**0.55±0.26**	**0.32±0.25**	**0.34±0.21**
Attn-HybridNet (proposed)	5.11±0.65	2.80±0.42	2.12±0.15	1.88±0.40

TABLE IV: Classification Error on Extended YaleB dataset.

fusion of information obtained from these two views for supervised classification. We evaluate the significance of these contributions under the following research questions:

Q1: Is the integration of both the minutiae view and the amalgamated view beneficial? Or, does their integration deteriorate the generalization performance of HybridNet?

In order to evaluate this, we varied the amount of training data in *HybridNet*, the PCANet, and TFNet and obtained the classification performance from their corresponding features on CIFAR-10 and MNIST variations datasets. We cross-validated the performances of these schemes for 5 times and present their mean and variances in Fig. 4.

Firstly, these plots clearly suggest that the classification accuracies obtained with the features from *HybridNet* (and also from the PCANet and the TFNet) linearly increase with respect to the size of training data. Secondly, these plots also demonstrate that the information obtained from the amalgamated view in PCANet is superior than the information obtained from the minutiae view TFNet on object-recognition dataset. However, these two kinds of information achieve competitive classification performance on variations of handwritten digits dataset which contains nearly aligned images.

Most importantly, these plots unambiguously demonstrate that integrating both kinds of information can enhance the superiority of feature representations, consequently improving the classification performance in proposed *HybridNet*.

Q2: How does the hyper-parameters affect the discriminability of feature representations? Moreover, how does these affect the performance of HybridNet and Attn-HybridNet?

To address this question, we present a detailed study on how the hyper-parameters affect the performance of *HybridNet* and *Attn-HybridNet*. In this regard, we compare the classification performance of the PCANet, TFNet, *HybridNet*, and *Attn-HybridNet* on CIFAR-10 dataset in Table II. The lowest error is highlighted in slightly larger font, while the minimum error achieved in each row is highlighted in bold font. Moreover, we also illustrate the performance of *Attn-HybridNet* by varying the dimension of context level feature vector w utilized in our attention-fusion scheme in Fig. 5.

A clear trend is visible in Table II among the performances of all the networks, where the classification error decreases with an increase in the number of filters in the first layer of the networks. This trend also demonstrates the effect of the factorization rank while obtaining the *principal-components* and the matrix factors with *LoMOI*; signifying that increasing the number filters in the first layer allows all the networks to increase the data variability that aids in obtaining better feature correspondences in the output stage. In addition, this also increases the dimensionality of the features extracted by the networks suggesting that comparatively higher dimensional features have lower intraclass variability among the feature representations of objects from the same category.

Another trend can be observed in the performance table where the classification error increases with the increase of the patch size of the image. Since the dimension of images in CIFAR-10 is 32×32, this may be due to the presence of less background with smaller image-patches as increasing the patch size gradually mount to non-stationary data [10].

Importantly, our proposed *Attn-HybridNet* substantially reduces the classification error by **22.78**% when compared to classification performance with *HybridNet* on CIFAR-10 dataset. The plot in Fig. 5 shows the effect on classification accuracy by varying dimensions of feature level context vector w in *Attn-HybridNet*.

Q3: How does the proposed Attn-HybridNet (and HybridNet) perform in comparison to the baseline techniques?

To evaluate this requirement, we compare the performance of the proposed *Attn-HybridNet* and *HybridNet* against baselines as detailed in Sec. VI-C. In this regard, we present

Methods	baisc	rot	bg-rand	bg-img	bg-img-rot	rect-image	convex
CAE-2 [33]	2.48	9.66	10.90	15.50	45.23	21.54	-
TIRBM [53]	-	**4.20**	-	-	35.50	-	-
PGBM [54]	-	-	6.08	12.25	36.76	**8.02**	-
ScatNet-2 [32]	1.27	7.48	12.30	18.40	50.48	15.94	6.50
PCANet [10]	1.07	6.88	6.99	11.16	35.46	13.59	4.15
TFNet [18]	1.07	7.15	6.96	11.44	37.02	16.87	4.98
HybridNet (proposed)	1.01	6.32	5.46	10.08	33.87	12.91	3.55
Attn-HybridNet (proposed)	**0.94**	4.31	**3.73**	**8.68**	**31.33**	10.65	**2.81**

(a) MNIST Variations Datasets

Methods	Error (%)
Textons [55]	1.50
BIF [56]	1.40
Histogram [57]	1.00
ScatNet [32]	**0.20**
PCANet [10]	0.84
TFNet [18]	0.96
HybridNet (proposed)	0.81
Attn-HybridNet (proposed)	0.72

(b) CuReT Dataset

TABLE V: Classification Error on MNIST variations and CUReT datasets.

Methods	**#Depth**	**#Params**	**Error**
Tiled CNN [40]	-	-	26.90
K-means (tri.) [43] (1600 dim.)	1	5	22.10
CUDA-Convnet [41]	4	1.06M	18.00
VGG-CIFAR-10 [58]	5	2.07M	20.04
SPP [44]	5	256.5K	19.39
CBoF [45]	5	174.6K	20.47
Spatial-CBoF [4]	5	199.1K	21.37
ResNet reported in-[59]	110	1.7M	13.63
DenseNet-BC reported in-[52]	250	15.3M	**5.2**
PCANet [10]	3	7	26.43
TFNet [18]	3	7	29.25
HybridNet (proposed)	3	7	23.84
Attn-HybridNet (proposed)	3	12.7k	18.41

TABLE VI: Classification Error on CIFAR-10 dataset with no data augmentation. The DenseNet achieves the lowest classification error but at the expense of huge depth and substantial computational cost among all techniques.

the performance comparison on face recognition datasets in Table III and Table IV.The performance comparison on handwritten digits and texture classification datasets are presented in Table V. Furthermore, classification results on CIFAR-10 and CIFAR-100 datasets are presented in Table VI and Table VII, respectively.

Besides, we visualize the discriminability of feature representation obtained from *HybridNet* and *Attn-HybridNet* with t-SNE plot [60] in Fig. 7 to perform a qualitative analysis of their discriminability. We further aid this analysis with a plot to study their classification performance obtained with increasing amount of training data on CIFAR-10 dataset in Fig. 6.

a) Performance on face recognition: A similar trend is noticeable from the classification performances on ORL and Extended YaleB datasets. First, for all schemes, the classification error decreases with the increase of the number of training examples. This is expected as by increasing the amount of training data all schemes can better estimate the variation in lighting, facial expression, and pose. Secondly, among the baselines, PCANet+ performs substantially better than Deep-NMF and PCANet-II on both the face datasets. The poor performance for Deep-NMF can be justified as it needs a large amout of data to estimate its parameters. Whereas for PCANet-II, the explicit alignment of face images as a requirement can explain its degradation in performance. Besides, the PCANet+ also performs slightly better than PCANet as the earlier enhances the latter with a better feature encoding scheme.

Methods	**#Depth**	**#Params**	**Error**
ResNet reported in-[59]	110	1.7M	37.80
DenseNet-BC reported in-[52]	250	15.3M	**19.64**
PCANet [10]	3	7	53.00
TFNet [18]	3	7	53.63
HybridNet (proposed)	3	7	49.87
Attn-HybridNet (proposed)	4	42.2k	47.44

TABLE VII: Classification Error on CIFAR-100 dataset with no data augmentation.

Lastly, our proposed *HybridNet* outperforms the baselines and individual schemes, i.e. PCANet and TFNet, on both the datasets. This validates our hypothesis that the common and unique information are both essential and their fusion can enhance the classification performance. However, the classification performance of *Attn-HybridNet* is slightly worse than *HybridNet*. Again, this might be due to less amount of data available while learning the attention parameters, similar to Deep-NMF albeit it still performs better than Deep-NMF as; the underlying features are highly discriminative, and therefore it is less strenuous to discover attention weights for the proposed fusion scheme in comparison to Deep-NMF.

b) Performance on digit recognition and texture classification: On MNIST handwritten digits variations dataset, the *Attn-HybridNet* (and also the *HybridNet*) outperforms the baselines on five out of seven variations. In particular, for *bg-rand* and *bg-img* variations, we decreased the error (compared to [18]) by **31.68**% and **13.80**% respectively. On CUReT texture classification dataset, the *Attn-HybridNet* achieves the lowest classification error among all the networks, albeit it achieves slightly higher classification error compared to state of the art. However, the difference in classification error achieved by state of the art [32] and *Attn-HybridNet* is marginal and is only 0.5%.

c) Quantitative Performance on CIFAR Datasets: We compare the performance of proposed *Attn-HybridNet* and *HybridNet* against baselines as described in Sec. VI-C on CIFAR-10 and CIFAR-100 datasets and report their respective accuracies in Table VI and Table VII respectively.

The proposed *Attn-HybridNet* achieves the best performance among all kinds of networks studied in the paper. Technically, the proposed *HybridNet* reduces the error by 9.80% on CIFAR-10 and 5.91% on CIFAR-100 dataset in comparison to min(PCANet, *HybridNet*). Additionally, the proposed *Attn-*

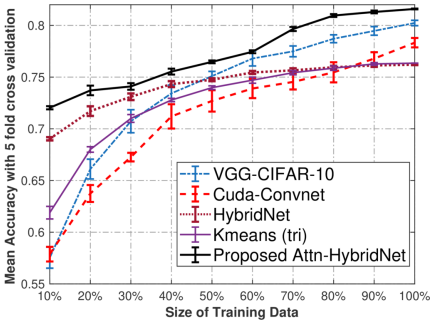
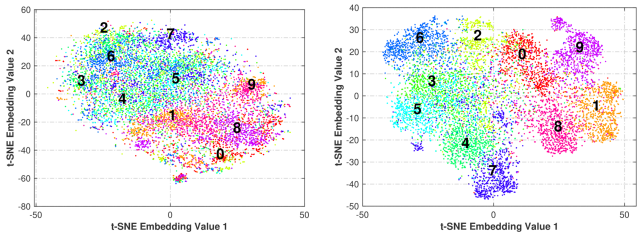


Fig. 6: (Best viewed in color) Accuracy of various methods on CIFAR-10 dataset by varying size of the training data



(a) *HybridNet* (b) *Attn-HybridNet*

Fig. 7: (Best viewed in color) t-SNE visualization of features from *HybridNet* and *Attn-HybridNet* on CIFAR-10 dataset.

HybridNet further reduces the error by 22.78% on CIFAR-10 and 4.87% on CIFAR-100 dataset in comparison to *HybridNet*. Besides, on CIFAR-10 dataset, *Attn-HybridNet* achieves substantially lower error compared to Titled CNN [40], K-means (tri), and the PCANet; particularly 16.70% lower than K-means (tri) which has $2\times$ higher feature dimensionality than our proposed *HybridNet* and utilizes L_2 regularized-SVM instead of $Linear$-SVM for classification.

The performance of our proposed *Attn-HybridNet* is still better than VGG-CIFAR-10 [42] and comparable to CUDA-Convnet [41][9], both of which have more depth than the proposed *Attn-HybridNet*. In particular, we have reduced the error by 1.63% than VGG-CIFAR-10 with 99.63% less trainable parameters. At the same time, we have performed very competitive to CUDA-Convnet achieving 0.41% higher error rate but with 88% less number of tunable parameters.

Our proposed *Attn-HybridNet* also performs marginally better in compared to deep-quantized networks such as SPP [44], CBoF [45], and Spatial-CBoF [4]. This is because the proposed scheme is decoupled as feature extraction and feature pooling schemes and hence the effort required to estimate the tuneable parameters is negligible. The quantization schemes are proposed to reduce the parameters in the fully connected layer but requiring the same efforts required to find optimal parameters of the higher layers.

Besides, the performance gap between *Attn-HybridNet* and state of the art ResNet and DenseNet are not comparable as the depth and the computational complexity of the latter networks are tremendously huge. Therefore, the main bottleneck for these schemes is requirement of high performing hardware which is opposite to the motivation of this work that is alleviation of such requirementsand hence the tradeoff.

d) Qualitative Discussion on CIFAR-10: We now present a qualitative discussion on the performances of various baselines and our proposals by varying the size of training data in Fig. 6. Although our proposed *Attn-HybridNet* consistently achieved the highest classification performance, a few interesting patterns are noticeable in the performance curves.

A paramount observation in this regard is the lower classification performance achieved by both CUDA-Convnet [41] and VGG-CIFAR-10 [58] with less amount of training dataset, particularly until 40%. It is intuitive and justifiable since less amount of the training data is not sufficient to learn the parameters of these deep networks. However, on increasing the amount of training data (above 50%), the performance of these networks increases substantially i.e., increases with a larger margin compared to the performance of SVM based schemes in *HybridNet* and K-means (tri) [43].

The second observation is regarding the classification performances of *HybridNet* and K-means (tri). Both these networks achieve higher classification accuracy compared to the deep networks with less amount of training data; particularly, the *HybridNet* has **11.56**$\%$ higher classification rate compared to the second-highest classification accuracy achieved by K-means (tri) with only 10% of the training dataset. However, the accuracy of these networks does not scale or increase substantially with an increase in the training data, as noticed by deep-network-based schemes.

Besides, the *Attn-HybridNet* achieved the highest classification performance across different sizes of the training dataset among all techniques. A possible explanation for this is the requirement of fewer parameters with proposed attention-fusion while performing feature selection with attention-based fusion to alleviate the feature redundancy. Moreover, the t-SNE plot in Fig. 7 compares the discriminability of features obtained with the *HybridNet* and *Attn-HybridNet*. The plot on the features obtained from *Attn-HybridNet* Fig. 7(b) visually achieves better clustering than the plot on features obtained from *HybridNet* Fig. 7(a) and justifies the performance improvement with our proposal.

VIII. Conclusion and Future Work

In this paper, we have introduced *HybridNet*, which integrates the information discovery and feature extraction procedure from the amalgamated view and the minutiae view of the data. The development of *HybridNet* is motivated by the fact that information obtained from the two views of the data are individually insufficient but necessary for classification. To extract features from the minutiae view of the data, we proposed the TFNet that obtains weights of its convolution-tensor filters by utilizing our custom-built *LoMOI* factorization algorithm. We then demonstrated how the information obtained with the two views of data are complementary to each other. Then, we provided details to simultaneously extract the common information from the amalgamated view and unique information with the minutiae view of the data in our proposed *HybridNet*. The significance of integrating these two kinds of

[9]We cite the accuracy as published.

information with *HybridNet* is demonstrated by performing classification on multiple real-world datasets.

Although the *HybridNet* achieves higher classification accuracy, it still suffers from the problem of feature redundancy arising from the generalized spatial pooling operation utilized to aggregate the features in the output layer. Therefore, we proposed *Attn-HybridNet* for alleviating the feature redundancy by performing attentive feature selection. Our proposed *Attn-HybridNet* enhances the discriminability of features, which further enhances their classification performance.

We performed comprehensive experiments on multiple real-world datasets to validate the significance of our proposed *Attn-HybridNet* and *HybridNet*. The features extracted using our proposed *Attn-HybridNet* achieved similar classification performance among popular baseline methods with significantly less amount of hyper-parameters and training time required for their optimization. Besides, we also conducted multiple case studies with other popular baseline methods to provide qualitative justifications for the superiority of features extracted by our proposed *Attn-HybridNet*.

Furthermore, our research can be further improved with two interesting research directions. The first direction is the design of *HybridNet* filters to accommodate various non-linearities in the data such as alignments and occlusion. A second research direction can be the design of attention-based fusion for generalized tasks such as face verification and gait recognition.

REFERENCES

[1] L. Zheng, Y. Yang, and Q. Tian, "Sift meets CNN: a decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, 2018.

[2] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[3] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[4] N. Passalis and A. Tefas, "Training lightweight deep convolutional neural networks using bag-of-features pooling," *IEEE transactions on neural networks and learning systems*, 2018.

[5] S. Han, H. Mao, and W. J. Dally, "Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding," *ICLR*, 2016.

[6] J. Kossaifi, A. Khanna, Z. Lipton, T. Furlanello, and A. Anandkumar, "Tensor contraction layers for parsimonious deep nets," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 1940–1946.

[7] X. Zhang, J. Zou, X. Ming, K. He, and J. Sun, "Efficient and accurate approximations of nonlinear convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1984–1992.

[8] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned cp-decomposition," in *ICLR*, 2015.

[9] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu, "Quantized cnn: a unified approach to accelerate and compress convolutional networks," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–14, 2017.

[10] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: a simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[11] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Master's thesis, Department of Computer Science, University of Toronto, 2009.

[12] Y. Jia, O. Vinyals, and T. Darrell, "On compact codes for spatially pooled features," in *International Conference on Machine Learning*, 2013, pp. 549–557.

[13] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3370–3377.

[14] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *European Conference on Computer Vision*. Springer, 2002, pp. 447–460.

[15] S. Verma, W. Liu, C. Wang, and L. Zhu, "Extracting highly effective features for supervised learning via simultaneous tensor factorization," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[16] J.-T. Chien and Y.-T. Bao, "Tensor-factorized neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1998–2011, 2017.

[17] W. Liu, J. Chan, J. Bailey, C. Leckie, and K. Ramamohanarao, "Mining labelled tensors by discovering both their common and discriminative subspaces," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 614–622.

[18] S. Verma, W. Liu, C. Wang, and L. Zhu, "Hybrid networks: Improving deep learning networks via integrating two views of images," in *Neural Information Processing - 25th International Conference, ICONIP , Proceedings, Part I*, 2018, pp. 46–58.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *International Conference on Machine Learning*, 2015, pp. 2285–2294.

[21] J. Huang and C. Yuan, "Fanet: factor analysis neural network," in *International Conference on Neural Information Processing*. Springer, 2015, pp. 172–181.

[22] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: from two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[23] B. Savas and L. Eldén, "Handwritten digit classification using higher order singular value decomposition," *Pattern Recognition*, vol. 40, no. 3, pp. 993–1003, 2007.

[24] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of eeg signals: a brief review," *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, 2015.

[25] J. Wu, S. Qiu, R. Zeng, Y. Kong, L. Senhadji, and H. Shu, "Multilinear principal component analysis network for tensor object classification," *IEEE Access*, vol. 5, pp. 3322–3331, 2017.

[26] R. Zeng, J. Wu, L. Senhadji, and H. Shu, "Tensor object classification via multilinear discriminant analysis network," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 1971–1975.

[27] H. Lu, K. N. Plataniotis, and A. Venetsanopoulos, *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. Chapman and Hall/CRC, 2013.

[28] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 921–928.

[29] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[30] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: a library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[31] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007, pp. 473–480.

[32] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[33] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011, pp. 833–840.

[34] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, 2009.

[35] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[36] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1458–1465.

[37] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, "A deep matrix factorization method for learning attribute representations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 417–429, 2016.

[38] C.-Y. Low, A. B.-J. Teoh, and K.-A. Toh, "Stacking pcanet+: An overly simplified convnets baseline for face recognition," *IEEE Signal Processing Letters*, vol. 24, no. 11, pp. 1581–1585, 2017.

[39] C. Fan, X. Hong, L. Tian, Y. Ming, M. Piettikäinen, and G. Zhao, "Pcanet-ii: When pcanet meets the second order pooling," *IEICE Transactions on Information and Systems*, vol. 101, no. 8, pp. 2159–2162, 2018.

[40] J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, and A. Y. Ng, "Tiled convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2010, pp. 1279–1287.

[41] A. Krizhevsky. (2012 (accessed May 20, 2019)) Cuda convnet. [Online]. Available: https://code.google.com/archive/p/cuda-convnet/

[42] S. Chintala, *VGG Style CNN on CIFAR10*, 2017 (accessed September 3, 2017), https://github.com/soumith/DeepLearningFrameworks.

[43] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 215–223.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[45] N. Passalis and A. Tefas, "Learning bag-of-features pooling for deep convolutional neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5755–5763.

[46] R. Keshari, M. Vatsa, R. Singh, and A. Noore, "Learning structure and strength of cnn filters for small sample size training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9349–9358.

[47] X. Yu, T. Liu, X. Wang, and D. Tao, "On compressing deep models by low rank and sparse decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7370–7379.

[48] S. Lin, R. Ji, C. Chen, D. Tao, and J. Luo, "Holistic cnn compression via low-rank decomposition with knowledge transfer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 2889–2905, 2018.

[49] S. Lin, R. Ji, Y. Li, Y. Wu, F. Huang, and B. Zhang, "Accelerating convolutional networks via global & dynamic filter pruning." in *IJCAI*, 2018, pp. 2425–2432.

[50] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 784–800.

[51] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5058–5066.

[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[53] K. Sohn and H. Lee, "Learning invariant representations with local transformations," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012, pp. 1339–1346.

[54] K. Sohn, G. Zhou, C. Lee, and H. Lee, "Learning and selecting features jointly with point-wise gated boltzmann machines," in *International Conference on Machine Learning*, 2013, pp. 217–225.

[55] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the significance of real-world conditions for material classification," in *European Conference on Computer Vision*. Springer, 2004, pp. 253–266.

[56] M. Crosier and L. D. Griffin, "Using basic image features for texture classification," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 447–460, 2010.

[57] R. E. Broadhurst, "Statistical estimation of histogram variation for texture classification," in *Proc. Intl. Workshop on Texture Analysis and Synthesis*, 2005, pp. 25–30.

[58] I. Karmanov. (accessed May 20, 2019) Vgg style cnn on cifar10. [Online]. Available: https://github.com/soumith/DeepLearningFrameworks

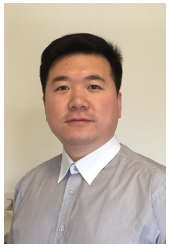
[59] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European conference on computer vision*. Springer, 2016, pp. 646–661.

[60] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

Sunny Verma Sunny Verma received his Ph.D. degree in Computer Science from University of Technology Sydney in 2020. He is currently working as Postdoctoral Research Fellow at the Data Science Institute, University of Technology Sydney, and as a visiting scientist at Data61, CSIRO. Before joining UTS, he was a Research Assistant at Department of Electrical Engineering, IITD India, and then worked as Senior Research Assistant at Hong Kong Baptist University, Hong Kong. He obtained his Ph.D. from the University of Sydney. His research interests include data mining, fairness in machine learning, interpretable deep learning, and open-set recognition systems.

Chen Wang Chen Wang is a senior research scientist with Data61, CSIRO. His research is in distributed and parallel computing with recent focus on data analytics systems and deep learning interpretability. He published more than 70 papers in major journals and conferences such as TPDS, TC, WWW, SIGMOD and HPDC. He has industrial experience. He developed a high-throughput event system and a medical image archive system used by many hospitals and medical centers in the USA.

Liming Zhu Dr/Prof. Liming Zhu is a Research Director at Data61, CSIRO. He is also a conjoint full professor at University of New South Wales (UNSW). He is the chairperson of Standards Australia's blockchain and distributed ledger committee. His research program has more than 300 people innovating in the area of big data platforms, computational science, blockchain, regulation technology, privacy and cybersecurity. He has published more than 200 academic papers on software architecture, secure systems and data analytics infrastructure and blockchain.



Wei Liu Wei Liu (M'15-SM'-20) is a Senior Lecturer and the Data Science Research Leader at the Advanced Analytics Institute, School of Computer Science, University of Technology Sydney. Before joining UTS, he was a Research Fellow at the University of Melbourne and then a Machine Learning Researcher at NICTA. He obtained his PhD from the University of Sydney. He works in the areas of machine learning and data mining and has published more than 80 papers in research topics of tensor factorization, game theory, adversarial learning, graph mining, causal inference, and anomaly detection. He has won three best paper awards.