Neural Processing Letters (2020) 52:443–465
https://doi.org/10.1007/s11063-020-10244-5

Fukunaga–Koontz Convolutional Network with Applications on Character Classification

Bernardo B. Gatto[1,2] · Eulanda M. dos Santos[2] · Kazuhiro Fukui[1,3] · Waldir S. S. Júnior[2] · Kenny V. dos Santos[2]

Published online: 21 May 2020

Abstract
Several convolutional neural network architectures have been proposed for handwritten character recognition. However, most of the conventional architectures demand large scale training data and long training time to obtain satisfactory results. These requirements prevent the use of these methods in a broader range of applications. As an alternative to cope with these problems, we present a new convolutional network for handwritten character recognition based on the Fukunaga–Koontz transform (FKT). Our approach lies in the assumption that Fukunaga–Koontz convolutional kernels can be efficiently learned from subspaces and directly employed to produce high discriminant features in a shallow network architecture. When representing image classes by subspaces, the within-class separability is reduced, since the subspaces form clusters in a low-dimensional space. To increase the between-class separability, we compute a discriminative space from the training subspaces using FKT. By learning convolutional kernels from subspaces, it is possible to extract representative and discriminative features from an image with only a few parameters. Another contribution of the proposed network is the use of pooling layers, which further improves its performance. The proposed method, called Fukunaga–Koontz Network (FKNet), is suitable for solving practical problems, especially when training and processing times are constraints. Four publicly available handwritten character datasets are employed to evaluate the advantages of

✉ Bernardo B. Gatto
bernardo@cvlab.cs.tsukuba.ac.jp

Eulanda M. dos Santos
emsantos@icomp.ufam.edu.br

Kazuhiro Fukui
kfukui@cs.tsukuba.ac.jp

Waldir S. S. Júnior
waldirjr@ufam.edu.br

Kenny V. dos Santos
kennyvinente@ufam.edu.br

[1] Center for Artificial Intelligence Research (C-AIR), Tsukuba, Japan

[2] Federal University of Amazonas, Manaus, Brazil

[3] University of Tsukuba, Tsukuba, Japan

FKNet. In addition, we demonstrate the flexibility of the proposed method by experiments on LFW dataset.

Keywords Subspace method · Shallow networks · Fukunaga–Koontz transform

1 Introduction

Handwritten character classification plays an essential role in computer vision and pattern recognition areas since it is fundamental in postal sorting, bank check recognition, automatic letter recognition, industrial automation, human-computer interaction, and historical archive documents [1–6]. Considering its importance, these applications require some characteristics, such as fast training and processing times. For example, it is desirable that the model can be rapidly adjusted when new training data becomes available. Simultaneously, the model should preserve its performance.

Another challenge in handwritten character classification is related to the huge amount of data required to train a useful model. For example, most public databases [7,8] consist of samples ranging from 1000 to 1500 images per class, which is generally not enough to describe all the variability of each class. In addition, in handwritten character classification, it is expected that the characters are written legibly with smaller variations in their shape. This assumption, however, does not hold in practical scenarios where camera noise, background conditions (especially illumination), writing speed and rotations are involved during the character's acquisition process. Besides, different writing styles increase within-class variability and, as different characters may share similar structures, a high correlation between these classes increases the problem complexity [9].

Deep learning-based approaches, specially those using deep convolutional neural networks (CNN), have been widely employed in problems involving handwritten character classification. Learning through deep neural networks has received significant attention due to its improvements over hand-crafted features [10]. The central concept of deep learning is that all relevant information required to recognize image patterns are contained in hierarchical neural network models.

Despite encouraging results, the fine-tuning of deep neural networks parameters is time-consuming [11], even when using machines with GPU. To avoid this issue, many shallow networks have been proposed based on principal component analysis (PCA), independent component analysis (ICA), canonical component analysis (CCA) and discrete cosine transform (DCT), where convolutional kernels are obtained from PCA, ICA or DCT basis vectors. For instance, PCANet (PCA network) [12] employs a CNN architecture with no pooling layers, no activation functions and without using back-propagation to learn its weights. Although only PCA or Linear Discriminant analysis (LDA) basis vectors define the convolutional kernels, these networks present competitive performance when compared to the state-of-the-art results achieved in several image classification tasks.

These shallow networks assume that models generated using PCA or ICA can efficiently produce convolutional kernels in a convolutional network architecture. However, these models do not provide discriminative features in more complicated computer vision problems, such as handwritten character recognition. In this kind of application, image classes may frequently contain complex structures and, due to the enormous variability in the handwritten shapes, between-class variation increases considerably [13].

In order to cope with these problems, we propose a shallow network based on the Fukunaga–Koontz transform (FKT) [14,15] to generate discriminative features and handle complex distributions. This transformation has been employed in the mutual orthogonal subspace method for face and object recognition [16] in the context of image set representation and classification. It is worth noting, however, that to the best of the authors knowledge, there is no approach using FKT in a shallow network approach.

FKT aims to decorrelate subspaces of different image classes. Given a dataset containing several classes, the weighed eigenvectors of the sum of the correlation matrices of all classes decorrelates the distributions of these different classes. These weighed eigenvectors can be adopted to orthogonalize these distributions, making this transformation a useful tool for feature extraction. We employ FKT in a slightly different manner, since this transform is based mainly on the sum of the covariance matrices of all classes. Instead of creating the transformation matrix from the sum of the correlation matrices, we utilize the sum of the projection matrices, which might produce more stable features, since the subspaces can have their dimensions independently estimated.

In the subspace method, the training data is represented by compact clusters in a low dimensional space. This space can be efficiently obtained from a set of basis vectors produced by a singular value decomposition (SVD). The subspace method assumes that by using multiple patterns in a low-dimensional space, the performance of recognizing complicated objects is significantly improved.

Another reason for employing a subspace method is that, in practice and under certain circumstances, there exist no two identical image distributions [17,18]. Accordingly, distributions corresponding to different handwritten images generate unique clusters in high dimensional vector space. The compression of these clusters leads to subspaces, where the variability of these patterns is represented more compactly.

Therefore, instead of employing PCA or LDA to learn the convolutional kernels, we use the subspace generated by FKT. By using the FKT decorrelation subspace, we build a shallow network, FKNet, that minimizes the correlation between different handwritten image classes. In FKNet, the training images are firstly compressed as subspaces to minimize their within-class distance. Besides, the decorrelation subspace based on the compressed data is more robust to outliers. Therefore, it is expected that such convolutional kernels can reveal more discriminative information compared to PCANet and related shallow networks.

A limitation of PCANet and its variants is that their architectures do not exceed 2 convolutional layers. In previously reported experimental results [12,19], improving the number of convolutional layers do not significantly improve the classification accuracy. This observation may be a result of the unsupervised dimensionality reduction operated by PCA. Such a dimensionality reduction can discard discriminative structures, leading to a weakening of the produced features. Here, we restrict the term shallow network employed in this work. Literature shows that this term has been frequently used to describe neural networks with no more than 10 layers [20]. However, we restrict our analysis to neural networks equipped with 4 layers or less.

Concurrently, due to the lack of pooling layers, feature vectors created by PCANet-like shallow networks grow exponentially as they propagate throughout the layers. The lack of a pooling method makes it impractical to use more than two layers without compromising computational performance. For example, in a 2 layers network that supports 20×20 input grayscale images, where each layer is equipped with 4 convolutional kernels (with convolutional kernel size of 3×3), the convolution process will produce a $20 \times 20 \times 4 \times 4 = 6400$-dimensional feature vector, if zero-padding is applied during the convolutional stage. However, If another layer with just 4 convolutional kernels (with convolutional kernel size of 3×3) is added, the size of the feature vector will become 25,600, making processing unfeasible.

To tackle this problem, pooling operation is used after two convolutional layers in the introduced network. This mechanism reduces the dimensionality of the feature vectors, increasing the shallow network number of layers without compromising computational performance.

Hence, our contributions are as follows:

1. A shallow network for handwritten character classification. Through the use of FKT, we generate a discriminative subspace projection to enhance the discriminability across the handwritten images classes.
2. An average pooling layer is introduced to increase the number of layers without increasing the feature dimensionality, preserving a low computational cost.
3. We propose a new type of convolutional kernel based on orthogonalization of subspaces. We employ FKT to learn a discriminative subspace projection. We show that the basis vectors of this subspace are useful as convolutional kernels, efficiently handling supervised data, solving one of the limitations of PCANet.

This work proceeds as follows: Sect. 2 presents related work on shallow networks. Then, in Sect. 3, we describe FKNet, as well as the procedure for learning the convolution kernels through FKT. In Sect. 4, we evaluate the proposed method by using publicly available databases, precisely USPS handwritten digits [21], C-Cube handwritten digits, lowercase and uppercase letters [22], EMNIST dataset [23], Semeion handwritten digits [8] and LFW face recognition dataset [24]. Finally, conclusions and future directions are provided in Sect. 5.

2 Related Work

In this section, we outline the shallow convolutional networks based on PCA, LDA, DCT, and CCA. We also describe the optimization strategies used to train the respective convolutional kernels. This review is essential to explain the differences between the proposed network and current methods, including the advantages over the existing networks.

Learning features directly from the data, instead of designing complex techniques for feature extraction, has been recognized as a dominant trend to prevent the drawbacks of handcrafted features. For instance, the histogram of oriented gradients (HOG) [25], local binary pattern (LBP) [26] and scale-invariant feature transform (SIFT) [27] produce satisfactory results when applied in problems related to handwritten character classification.

In [26], local binary patterns of handwritten characters are extracted, and a set of clustering techniques is used to assign a label for each character image. In this approach, handwritten character datasets are used to validate the method, including MNIST. However, these methods cannot simultaneously tackle problems caused by rotation, point-of-view, different writing styles, scale, and illumination conditions, which are usually observed in handwritten characters.

Deep neural networks aim to decrease the influence of within-class variability by representing the data hierarchically. Deep CNN generally presents the following stages: convolutional layer, nonlinear processing layer, and feature pooling layer. A random schema is employed to initialize the parameter of the convolutional kernels, which is iteratively updated by stochastic gradient descent. Learning a deep network is usually time-consuming due to its multistage nature and its large number of parameters, even when using machines equipped with modern GPU.

Many shallow networks have been proposed to alleviate the high computational cost of training a CNN. For instance, PCANet [12] is an image classification framework, where its convolutional kernels are learned from the data at the local image patch level. Despite their simplicity, shallow networks perform exceptionally well in a variety of image classification benchmarks, including handwritten and face recognition. Since PCANet requires only an SVD operation, its training time is fast compared to current CNN training times.

PCANet has been employed for handwritten character classification in many works. Its incremental version has also been introduced for handwritten character classification [28]. This work takes advantage of a lifelong learning framework to accomplish the plasticity of both feature and classifier constructions, producing high discriminative features in an incremental arrangement. The main advantage of PCANet over the conventional CNN is its reduced number of parameters to be tuned during the training stage. However, PCANet and its incremental version can only be equipped with 2 or 3 stages, as reported in [12]. When the architecture is designed with more than 3 layers, the recognition rate has no significant improvement. In PCANet, the basis vectors of the local covariance matrix are employed as convolutional kernels for initial feature extraction, followed by binarization and block-wise histogram operation to create the features. Besides, LDANet was also introduced in [12], where its convolutional kernels are based on linear discriminant analysis.

DCTNet [29] is an alternative to PCANet, which employs discrete cosine transform (DCT) as convolutional kernels. DCTNet has been widely applied to several face databases benchmarks and has shown performance equivalent or superior to both PCANet and LDANet. In spite of its effectiveness, when data sparsity is not clustered around low frequencies, PCA should be a preferred model over DCT [30]. In such cases, PCANet and LDANet can benefit from the data dependency model. On the other hand, DCTNet is recommended when training data is scarce, since its convolutional kernels are data-independent. Besides, 2D DCT is also employed to decrease the computational complexity on the network training phase. Even though DCTNet has not been applied in handwritten character classification, we understand that this shallow network can achieve competitive results when dealing with such learning problem, considering that DCT features of handwritten character present satisfactory results [31,32]. Therefore, in this work, we also evaluate DCTNet.

Although PCANet and LDANet have shown high performance, these networks do not directly handle multiple-view features. In order to overcome this issue, CCANet [19] is introduced to deal with data that are not represented by single-view features. CCANet extracts two different view features of one object to generate the final pattern, which may achieve higher recognition accuracy than the accuracy attained with a single view. Experiments conducted using the ETH-80, Yale-B, and USPS databases for object classification, face classification and handwritten digit classification show that CCANet outperforms PCANet and LDANet.

Existing methods in literature employ subspaces to represent class images [33–36]. These methods address the problem of finding a so-called constraint subspace in which the projected features may provide more discriminative features. Among these methods, the orthogonal subspace method has received substantial attention due to its results in object recognition and face recognition [37,38]. Besides, the subspace method has been employed in handshape classification, protein classification and clustering, and motion recognition [39–41].

The networks investigated in this work can be examined in a subspace method perspective, since its convolutional kernels are obtained through subspace learning instead of gradient learning. PCA, LDA and CCA yield a linear projection $\mathbb{R}^N \rightarrow \mathbb{R}^M$, presenting distinct properties [42,43], where it is desirable that $M \ll N$. Therefore, we can present more

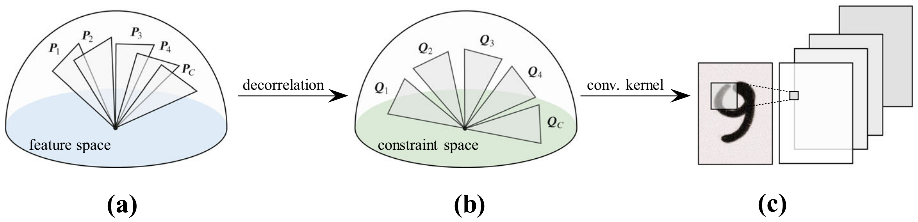

Fig. 1 The decorrelation process generated by Fukunaga–Koontz transform and its application in this work. **a** Image sets form clusters in a low-dimensional space, which can be represented by $\boldsymbol{P}_i$ subspaces. These subspaces, however, are not optimal for classification due to lack of discriminative mechanism. **b** FTK is employed to decorrelate the subspaces. **c** When subspaces $\boldsymbol{P}_1$, $\boldsymbol{P}_2 \ldots$, $\boldsymbol{P}_C$ represent image patches, the FKT transformation matrix can be used as a convolutional kernel

advanced linear projections such as those produced by FKT. The proposed shallow network is described in the next section.

3 Proposed Method

In this section, an overview of the proposed shallow network is provided, followed by the details of its building blocks. Then, the learning process employing image patches is described. After that, the procedure to compute linear subspaces is presented, as well as the method to calculate their optimal dimensions. This step is critical to maintaining the relationship between the compactness of each subspace and its representation. In our study, we understand that each class has a different compaction ratio; therefore, each class must be represented by subspaces of different dimensions. The problem of decorrelating subspaces using FKT and its application as convolutional kernels are introduced. Finally, the feature representation produced by the proposed shallow network is given. Figure 1 illustrates the procedure to construct FTK and its application as convolutional kernels.

3.1 Fukunaga–Koontz Network

The following notations are adopted in this work. Scalars are denoted by upper case letters, vectors by lowercase letters and boldface uppercase letters denote matrices. Calligraphic letters will be assigned to basis vectors. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, $\boldsymbol{A}^\top$ denotes its transpose.

Figure 2 shows the conceptual diagram of the proposed shallow network. FKNet processes images as follows. An input image is processed by a convolutional feature extraction layer, which can be followed by a mean-pooling or by other convolutional layers. Then, binary hashing is applied on the produced features in order to achieve dimensionality reduction. Finally, a block-wise histogramming is employed to achieve relative rotation invariance and create the final feature vector.

3.2 Representation by Image Patches

Given a dataset $\boldsymbol{X}$ consisting of N labeled training images of size $H \times W$, we extract patches of size $K_1 \times K_2$ from $\boldsymbol{X}$. This procedure is performed by taking a patch around each pixel from each one of the N training images. Here, we denote the set of image patches as $\boldsymbol{P}$.

Given that each image patch will have size $K_1 \times K_2$, the set $\boldsymbol{P}$ will contain $N_{\boldsymbol{P}} = HWN$ patches. It is worth noting that, after collecting the patches of all the images, FKNet does not perform the mean-removal operation on $\boldsymbol{P}$, as employed in PCANet, since this operation modifies the subspace obtained.

3.3 Computing Image Patches Subspaces

Although PCA is considered optimal for pattern representation, the subspaces created by PCA are not necessarily optimal for classification. We understand that this is an issue, since PCANet employs PCA to produce a common subspace that represents the dataset regarding its variance, but neglecting intra-class characteristics.

There are many types of supervised methods that can be employed to implement efficient convolutional kernels for our shallow network, such as LDA. FKT is suitable for the supervised problem setting since it can work well with even a small quantity of data [44]. This problem setting, well known as small sample size problem, is very challenging for LDA due to its inability to estimate the within-class scatter matrix adequately in such circumstances. In contrast, FKT avoids this issue by introducing the subspace representation, which can be stably estimated from even few samples [45].

To create subspaces, we will use the patch set $\boldsymbol{P} = \{p_i^j\}_{i,j=1}^{N_j,C}$, where C stands for the number of classes and N_j is the number of patches in the jth class. In this C class classification problem, it is required to compute C feature matrices $\{\boldsymbol{A}_j\}_{j=1}^{C}$. For each feature matrix $\boldsymbol{A}_j$, we compute the autocorrelation matrix $\boldsymbol{C}_j = \boldsymbol{A}_j{}^\top \boldsymbol{A}_j$. Equipped with all C autocorrelation matrices, we can move forward to calculate the matrix $\boldsymbol{U}_j$ of eigenvectors which diagonalizes the autocorrelation matrix $\boldsymbol{C}_j$:

$$\boldsymbol{D}_j = \boldsymbol{U}_j{}^{-1} \boldsymbol{C}_j \boldsymbol{U}_j, \quad j = 1, \ldots, C. \tag{1}$$

In Eq. (1), each $\boldsymbol{U}_j$ is a $K_1K_2 \times K_1K_2$ matrix satisfying $\boldsymbol{U}_j \boldsymbol{U}_j{}^\top = \boldsymbol{U}_j{}^\top \boldsymbol{U}_j = \boldsymbol{I}$. The columns of $\boldsymbol{U}_j$ that correspond to nonzero singular values compound a set of orthonormal basis vectors for the range of $\boldsymbol{C}_j$. $\boldsymbol{D}_j$ is the diagonal matrix of eigenvalues of $\boldsymbol{C}_j$. In our work, we use non-centering subspaces, different from the scatter matrix handled by PCANet. Since this difference produces very distinct subspaces, we follow the conventional formulation of subspace-based methods [16,34]. Unlike PCANet, FKNet creates a subspace for each class independently, exploiting its intrinsic characteristics in a more effective way.

3.4 Selecting Basis Vectors of the Image Patches Subspaces

One of the advantages of employing subspaces to represent handwritten image classes is that it is possible to compress each image set according to the basis vectors contribution in terms of variance. Specifically, the function $\mu(\cdot)$ regulates the proportion of the basis vectors employed to efficiently describe an image-set:

$$\mu(R_j) \leq \frac{\sum_{m=1}^{R_j} \lambda_m}{\sum_{m=1}^{D_j} \lambda_m}. \tag{2}$$

In this expression, R_j represents the number of selected basis vectors that spam the $\boldsymbol{P}_j$ subspace and λ_m is the m-th eigenvalue of the eigendecomposition of the scatter matrix $\boldsymbol{C}_j$.

Finally, $D_j = \text{rank}(P_j)$. Since the eigenvectors are arranged according to the eigenvalues in descent order, the R_jth eigenvector associated with the R_jth eigenvalue is selected, as well as the eigenvectors associated with the eigenvalues higher than the R_jth eigenvalue.

Here, the main idea is to set R_j that best describes the image set without redundancy and in a compact manner. This parameter depends on the complexity of the correlations inherent of each image set and is also application-dependent. As mentioned before, the eigendecomposition of the scatter matrix $\boldsymbol{C}_j$ is able to capture the vectors explaining most of its variation.

3.5 FKT for Image Patches Subspaces Decorrelation

Once equipped with all the C image patches subspaces $\boldsymbol{P}_j$ and their R_j dimensions have been computed, we can now use FKT to generate the matrix $\boldsymbol{F}$ that can decorrelate the subspaces. Then, each set of basis vectors $\boldsymbol{U}_j$ spans a reference subspace $\boldsymbol{P}_j$, where its compactness ratio is empirically defined by choosing the first R_j vectors, ordered by its accumulated energy, as shown in Eq. (2). The method to generate the matrix $\boldsymbol{F}$ that efficiently decorrelates the C R_j-dimensional classes subspaces is explained as follows. First, we compute the total projection matrix as:

$$\boldsymbol{G} = \sum_{j=1}^{C} \boldsymbol{U}_j \boldsymbol{U}_j^\top . \tag{3}$$

The eigendecomposition of the total projection matrix $\boldsymbol{G}$ produces a $K_1K_2 \times K_1K_2$ decorrelation matrix $\boldsymbol{F}$. This procedure is better described by the following equation:

$$\boldsymbol{F} = \boldsymbol{\Lambda}^{-1/2}\, \boldsymbol{B}^\top , \tag{4}$$

where $\boldsymbol{B}$ is the set of orthonormal eigenvectors corresponding to the $N_{\boldsymbol{F}}$ largest eigenvalues of $\boldsymbol{G}$, and $\boldsymbol{\Lambda}$ is the $K_1K_2 \times K_1K_2$ diagonal matrix with the m-th highest eigenvalue of the matrix $\boldsymbol{G}$ as the m-th diagonal component.

3.6 Fukunaga–Koontz Convolutional Kernels

After obtaining the image patches subspaces and the decorrelation matrix $\boldsymbol{F}$, we can now compute the FK convolutional kernel. In our formulation, each basis vector of $\boldsymbol{F} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{N_{\boldsymbol{F}}}\}$ will be a convolutional kernel in the network. According to this formulation, the definition of the Fukunaga–Koontz convolutional kernel is:

$$\boldsymbol{W}_l = \text{map}_{K_1 \times K_2}(\mathbf{w}_l), \qquad l = \{1, 2, \ldots, L_S\}, \tag{5}$$

where the operator $\text{map}_{K_1 \times K_2}(\cdot)$ maps an input vector $y \in \mathbb{R}^{K_1K_2}$ onto a matrix $\boldsymbol{Y} \in \mathbb{R}^{K_1 \times K_2}$ and L_S is the number of convolutional kernels in the Sth convolutional layer.

According to [16], FKT decorrelates all the C class subspaces by generating a matrix where the canonical angles between the projected subspaces spanned by the $\boldsymbol{Q}_j = \boldsymbol{F}^\top \boldsymbol{U}_j$ basis vectors are enlarged. Following this idea, we can conclude that the eigenvalue matrices $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ of the following products:

$$\boldsymbol{S}_1 = \boldsymbol{Q}_i^\top \boldsymbol{Q}_j, \qquad \forall\, i \neq j, \tag{6}$$

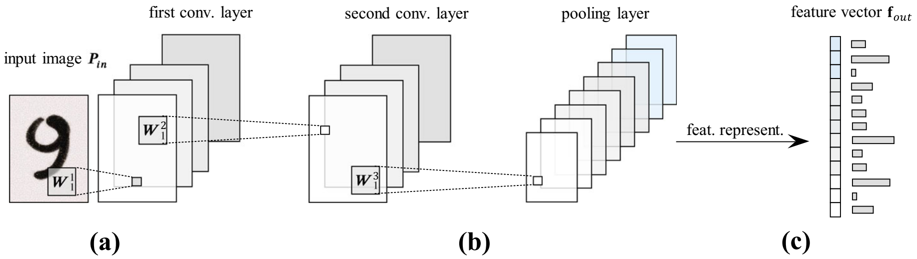

Fig. 2 The shallow network architecture introduced in this work: a convolutional feature extraction layer processes an input image based on FK convolutional layer, followed by another FK convolutional layer. Then, an average pooling layer is employed. Finally, binary hashing and a block-wise histogramming produce the final feature vector

$$\boldsymbol{S}_2 = \boldsymbol{Q}_i^\top \boldsymbol{Q}_j, \qquad \forall\, i = j, \tag{7}$$

approaches the null matrix and the identity matrix, respectively. In the proposed network, this observation enforces that the features created by the matrix $\boldsymbol{F}$ will produce a mechanism where patterns of the same class will be projected onto an adjacent space and, simultaneously, separated from the other classes.

Given an input image $\boldsymbol{P}_{in}$, the output image $\boldsymbol{Y}_l$ of a convolutional layer is obtained by the following operation:

$$\boldsymbol{Y}_l = \rho(\boldsymbol{W}_l * \boldsymbol{P}_{in}), \qquad l = \{1, 2, \ldots, L_S\}, \tag{8}$$

where $*$ refers to a convolution with zero-padding in the boundary of the image patch and $\rho(\cdot)$ is an average pooling operator, which may or may not be present in a particular layer, defined by a $B_1 \times B_2$ window, where $B_1, B_2 \in \mathbb{N}^+$.

Note that the output of one convolutional layer produces L_S images. Similar to CNN and PCANet, multiple layers can be created by feeding the produced images as input to a new layer. In general, a Z layers architecture produces $N_Z = L_1 L_2 \ldots L_Z$ images for each input image, so in total N_Z images are produced.

Moreover, the output of the first layer of the proposed network will produce L_1 images. By using $\boldsymbol{Y}_l$, more image patches subspaces can be learned to create more layers. Usually, more than one layer is employed in such shallow networks, so more features can be extracted from $\boldsymbol{P}_{in}$. For instance, for a $Z = 2$ layers network, we should learn 2 constraint subspaces, where $\boldsymbol{W}_l^1$ may be learned from $\boldsymbol{X}$, and $\boldsymbol{W}_l^2$ can be learned from $\boldsymbol{Y}_l$.

3.7 Feature Representation

Continuing with the previous Z layers system, the convolutional layers will produce $\{\boldsymbol{Y}_p\}_{p=1}^{N_Z}$. The first step of the feature representation is the binarization of all $\boldsymbol{Y}_p$ images, using a step-like function, which can be defined as follows:

$$H(a) = \begin{cases} 1, & \text{if } a > 0, \\ 0, & \text{if } a \leq 0 \end{cases}. \tag{9}$$

where the parameter a is each pixel intensity mapped in the image, in an element-wise manner. Now, each element possesses a binary value, so the set of the same pixels in the N_Z images produce digit binary words that can be seen as a decimal number. This procedure converts the images back into a single integer-valued matrix $\boldsymbol{T}_q$ whose every pixel is an integer in the range $[0, 2^{L_Z-1}]$. Formally, this operation can be expressed as:

$$\boldsymbol{T}_q = \sum_{p=1}^{L_z} 2^{p-1} H(\boldsymbol{Y}_p), \qquad q = \{1, 2, \ldots, N_{Z-1}\}, \tag{10}$$

Each $\boldsymbol{T}_q$ matrix is partitioned into B blocks, and a block histogram is computed to count the decimal values in each block. After that, we concatenate all histograms into one vector. This encoding is then stored as a $\boldsymbol{f}_{out}$ feature of the input image $\boldsymbol{P}_{in}$ in a vector of block-wise histogram.

The column vector $\boldsymbol{f}_{out}$, together with other feature vectors extracted from a training dataset are used to train a classifier. In the investigated architecture, support vector machines (SVM) is employed.

3.8 Computational Advantage

One of the advantages of the proposed network is its reduced number of parameters compared to conventional CNN. The hyper-parameters of FKNet include the filter size K_1, K_2, the pooling size B_1, B_2, the number of filters in each stage $L_1, L_2, \ldots, L_Z$, the number of stages Z, the block size for the histogram, and the class subspaces dimensions. In addition to the decorrelation process employed by FKT, FKNet has the advantage of considering each class as a subspace, which allows it to be expressed more compactly and more robustly to outliers when compared to other shallow networks.

In terms of computational complexity, FKNet inherits the low cost exhibited by PCANet. More precisely, FKNet shares all the elements employed by PCANet, whose computational complexity depends only on the autocorrelation matrix computation and the filter convolution. Different from PCANet, FKNet requires C autocorrelation matrices computation, one autocorrelation matrix for each class subspace [see Eq. (3)]. Therefore, both processes generate the cost of:

$$\mathcal{O}(HWK_1K_2(L_1 + L_2) + CHW(K_1K_2)^2).$$

Since $HW \gg K_1K_2 > C$, the computational complexity of FKNet is comparable to the computational complexity of PCANet. Similar to PCANet, this computational complexity refers to learning and testing stages, as long as $HW \gg K_1K_2$.

Next section provides, along with other experiments, experimental results of processing time measurement by each network. For instance, CNN required about 3 h to generate a model with 4 convolutional layers using the EMNIST training dataset. On the other hand, FKNet obtained a comparable model using less than 17 min on the same hardware, which is approximately one order of magnitude faster.

4 Experimental Evaluation

In this section, the effectiveness of the proposed FKNet is evaluated. Experiments are conducted using five public databases: USPS handwritten digits [21], C-Cube handwritten digits, lowercase and uppercase letters [22], Semeion handwritten digits dataset [8] and EMNIST dataset [23]. These datasets cover various unconstrained scenarios of handwriting images, as the digits were written by many different subjects, writing styles and devices, with widely varying levels of care. In addition to the handwritten datasets, we evaluate the flexibility of the proposed network by using the LFW face dataset [24].

The experiments are divided into four main series. In the first series, FKNet is compared to 5 shallow networks. In the second series of experiments, we study the impact of changing the amount of training data concerning the performance of the networks. The third series is performed by comparing the proposed shallow network to a CNN. First, however, the description of three datasets used in our experiments is presented. The fourth dataset is described in Sect. 4.4. After evaluating the proposed method using handwritten character datasets, we present a challenging task to evaluate the proposed shallow network. Therefore, in Sect. 4.5, we further evaluate our method in a face verification task using the LFW dataset.

4.1 Dataset Configuration

The US postal service dataset (USPS) is a multi-class digit dataset consisting of 9298 handwritten digit images ranging from 0 to 9. In this dataset, there are 7291 training images and 2007 test images. Each digit image is of size 16 × 16 pixels. The raw grayscale pixels are used as features for all the methods compared in this paper. We pre-processed all images to have zero-mean and to be of unit Euclidean norm and resized the images to 28 × 28 pixels. This process was performed in all other databases.

C-Cube dataset consists of 57,293 handwritten images of 52 English letters, divided into 38,160 (22,274 lowercase and 15,886 uppercase) training images and 19,133 (11,161 lowercase and 7972 uppercase) test images. This is a very realistic dataset, considering that the images were manually extracted from the Center of Excellence for Document Analysis and Recognition (CEDAR) and United States Postal Service (USPS) databases. This is a challenging dataset, since the number of images per class is very imbalanced. In addition, the handwritten images are very cursive, increasing the correlation between classes. The dataset contains upper-case and lower-case letters, which were randomly split into training and test sets.

Semeion handwritten digits dataset consists of 1593 handwritten digits from around 80 persons. The images were scanned, stretched in a 16 × 16 matrix with 256 grayscale values. Then, all pixels of each image were binarized using a fixed threshold. Each person wrote on a paper all the digits from 0 to 9. The writing was performed two times; first time trying to write each digit accurately and the second time with no accuracy, as fast as possible. In addition, as the dataset is not originally divided into training and test datasets, a 10-fold cross-validation scheme is employed to evaluate the methods.

4.2 Comparison with Related Shallow Networks

In this first series of experiments, FKNet is compared to 5 shallow networks: PCANet, LDANet, RandNet (this network follows the same architecture of PCANet, but the filter banks are replaced with totally random filters), CCANet and DCTNet.

This experiment focus on comparing the classification rates attained by FKNet and baselines, as well as on analyzing their behavior when the number of layers is increased and when pooling layers are employed. In order to accomplish these objectives, all shallow networks are evaluated according to its number of stages, which varies from 1 to 4, and with or without pooling layers.

For a fair comparison, the Coiflets and Daubechies orthogonal wavelet transform are employed to extract the low frequency sub-images of the original images to generate two view features for CCANet [19]. The TR-Normalization introduced by [29] is not applied. As in PCANet, LDANet, and DCTNet, we select linear SVM for the classification step since it is relatively less prone to overfitting than its non-linear version.

In this experiment, all the methods use the same parameters as in [12]. Previous work exhaustively analyzed these shallow network parameters, such as the convolutional kernel size. Here, we aim to verify the limits of these shallow networks by changing its number of layers and evaluating which learning strategy presents the most efficient result. In [12], the number of filters was fixed to $L_1 = L_2 = 8$, $L_3 = L_4 = 8$. The convolutional kernel size was set to $K_1 = K_2 = 7$, with block size of 7×7 and 4 pixels for overlapping ($\sim 57\%$ of overlapping ratio).

Table 1 shows the mean accuracy (%) and the standard deviation of the proposed shallow network and the different baselines investigated on USPS handwritten digits database, C-Cube dataset and Semeion handwritten digits dataset, when 1, 2, 3 and 4 layers are employed. The number of convolutional layers Z and whether the network presents or does not present pooling layers is indicated by (p) and (–), respectively.

The best results regarding the accuracy are listed in bold, while the second-best results are listed in italic. We performed a significance test using Welch's t-test (at 95% significance level) between the best-performed network on each combination with the second-best result. Underlined values in Table 1 indicate and mark the statistically significant results. According to Welch's t-test, the proposed FKNet consistently achieved significantly better results on the Semeion dataset, which is the smallest investigated dataset. This result suggests that FKNet can represent small sets robustly.

In addition, from Table 1, it is observable that most of the methods have their recognition accuracy improved as the number of convolutional layers increased, up to 3 layers. However, when the number of layers is set to 4, most of the methods present no significant improvement. We understand that increasing the number of layers higher than 3 does not boost the recognition rate because it considerably increases the feature vector dimension used by SVM.

We also compare the networks performance when pooling layers are used, which are expected to produce a certain degree of invariance with respect to translations and elastic distortions [46,47]. As a consequence, there would be a certain level of robustness to small perturbations on handwritten characters positioning. In this scenario, the shallow networks benefit from the pooling layers, improving their classification rates. Besides, FKNet demonstrated superior classification rate when compared to the other evaluated shallow networks, confirming the efficiency of the method by employing the constraint subspace as convolutional kernels equipped with pooling layers.

Table 2 lists the training time required by our proposed method and by the baselines as well. We do not list the training times of RandNet and DCTNet since these methods do not rely on data to construct their filter banks. Moreover, the testing times are not listed because it depends mostly on the network configuration. Since we compare the networks with identical configuration, the testing time is very similar for all of them. It is possible to observe that PCANet attains the fastest training time, which is reasonable, considering that PCANet requires only an eigendecomposition per layer and an autocorrelation matrix computation. LDANet and CCANet require additional computations due to their more sophisticated formulation. Finally, although FKNet requires an autocorrelation matrices computation per class, its processing time is comparable to the other networks.

Table 1 The average classification rates and standard deviation attained by our proposed method, as well as by five baselines

Datasets	Layers	PCANet [12]	LDANet [12]	RandNet [12]	CCANet [19]	DCTNet [29]	FKNet
USPS [21]	1(–)	93.83 ± 2.01	93.71 ± 1.91	92.97 ± 2.03	**94.57 ± 2.49**	93.84 ± 2.05	*93.97 ± 1.91*
	2(–)	97.51 ± 1.55	97.37 ± 1.44	93.12 ± 1.57	**97.81 ± 2.01**	96.67 ± 1.56	**97.81 ± 1.53**
	3(–)	97.63 ± 1.51	97.29 ± 1.35	93.07 ± 1.49	*97.83 ± 2.03*	96.55 ± 1.47	**97.91 ± 1.52**
	3(*p*)	97.90 ± 1.49	97.67 ± 1.32	93.44 ± 1.45	*98.14 ± 1.99*	96.54 ± 1.47	**98.30 ± 1.50**
	4(–)	97.65 ± 1.58	97.33 ± 1.47	93.02 ± 1.66	*97.81 ± 2.11*	96.52 ± 1.53	**98.07 ± 1.69**
	4(*p*)	97.76 ± 1.57	97.43 ± 1.47	93.19 ± 1.63	*97.93 ± 2.09*	95.95 ± 1.52	**98.60 ± 1.66**
C-Cube [22]	1(–)	*83.56 ± 1.92*	83.11 ± 1.71	80.72 ± 2.34	83.53 ± 2.20	82.68 ± 1.84	**84.72 ± 1.81**
	2(–)	87.39 ± 1.38	87.73 ± 1.32	83.24 ± 1.48	*88.13 ± 1.68*	85.48 ± 1.28	**88.14 ± 1.52**
	3(–)	*88.71 ± 1.24*	87.47 ± 1.29	83.03 ± 1.41	88.13 ± 1.61	85.52 ± 1.28	**89.42 ± 1.59**
	3(*p*)	*88.97 ± 1.22*	87.71 ± 1.28	83.23 ± 1.40	88.40 ± 1.65	85.76 ± 1.25	**89.69 ± 1.55**
	4(–)	*88.59 ± 1.27*	87.51 ± 1.30	83.07 ± 1.52	88.20 ± 1.92	85.23 ± 1.46	**90.26 ± 1.57**
	4(*p*)	*88.65 ± 1.29*	87.67 ± 1.30	83.18 ± 1.53	88.36 ± 1.94	85.34 ± 1.45	**90.63 ± 1.54**
Semeion [8]	1(–)	*86.28 ± 1.41*	85.43 ± 1.32	82.58 ± 1.54	83.51 ± 1.26	85.30 ± 1.21	**88.36 ± 1.41**
	2(–)	*89.58 ± 1.37*	89.43 ± 1.25	88.48 ± 1.42	88.88 ± 1.24	89.24 ± 1.17	**90.11 ± 1.48**
	3(–)	*89.63 ± 1.41*	89.45 ± 1.33	87.77 ± 1.55	88.35 ± 1.26	88.64 ± 1.22	**91.43 ± 1.44**
	3(*p*)	*89.85 ± 1.40*	89.63 ± 1.37	87.95 ± 1.51	88.58 ± 1.27	88.90 ± 1.19	**91.67 ± 1.41**
	4(–)	*89.45 ± 1.56*	89.34 ± 1.37	87.05 ± 1.60	87.51 ± 1.29	88.42 ± 1.25	**91.66 ± 1.46**
	4(*p*)	*89.60 ± 1.59*	89.43 ± 1.38	87.01 ± 1.61	87.51 ± 1.33	88.60 ± 1.22	**91.81 ± 1.43**

Table 2 The training time (in minutes) attained by the proposed method by the five baselines

Datasets	Layers	PCANet [12]	LDANet [12]	CCANet [19]	FKNet
USPS [21]	1(-)	14.67	16.22	17.77	18.90
	2(-)	15.61	17.23	18.59	20.11
	3(-)	29.11	31.78	33.45	35.51
	3(p)	20.29	22.39	24.16	26.14
	4(-)	151.57	162.26	170.47	179.17
	4(p)	22.52	24.85	26.81	29.01
C-Cube [22]	1(-)	16.31	17.51	18.95	20.14
	2(-)	17.35	18.97	20.51	21.95
	3(-)	32.35	35.09	36.98	39.53
	3(p)	22.96	24.91	26.25	28.06
	4(-)	176.21	188.61	198.70	208.71
	4(p)	25.02	27.15	28.61	30.58
Semeion [8]	1(-)	3.26	4.25	5.39	6.30
	2(-)	3.47	4.71	5.36	5.73
	3(-)	6.47	7.02	8.31	8.73
	3(p)	4.82	5.31	6.47	6.85
	4(-)	30.34	33.81	36.22	39.22
	4(p)	5.92	6.53	7.95	8.42

In this experiment, it is clear the advantage of using pooling layers. The training time of the networks is reduced by about 30% when the network is equipped with a pooling layer after the third convolutional layer. This advantage increases when another pooling layer is added after the fourth convolutional layer.

Although the convolutional kernels of RandNet are randomly generated, reasonable results are obtained, in comparison to the results achieved by PCANet and LDANet. This observation indicates the benefits that the cascading model employed by shallow networks can provide. RandNet results are competitive when 1 or 2 layers are employed. However, when more layers are added, its results do not show improvement. This can be explained by the fact that there is no function to determine the convolutional kernel, weakening the hierarchical structure of the network. On the other hand, PCANet and FKNet have well-defined functions that determine the weights of the convolutional kernels, exploiting the hierarchical model to systematically produce better features.

In C-Cube dataset, PCANet, LDANet, and CCANet demonstrated competitive performances when using 1 and 2 layers, suggesting that subspace-based methods provide efficient convolutional layers for shallow networks. On the other hand, RandNet and DCTNet achieved the worst results. Compared to USPS and Semeion datasets, C-Cube delivers a larger number of examples, which can prevent the efficiency of convolutional kernels that are not generated from the training data. The results suggest that the unsupervised networks that do not depend on training data show greater difficulty in obtaining better results. Accordingly, DCTNet is recommended when training data is scarce due to its handcraft approach that is data-independent. Nonetheless, when datasets are more comprehensive and encompass more classes with higher diversity, PCANet, LDANet, CCANet and FKNet are recommended.

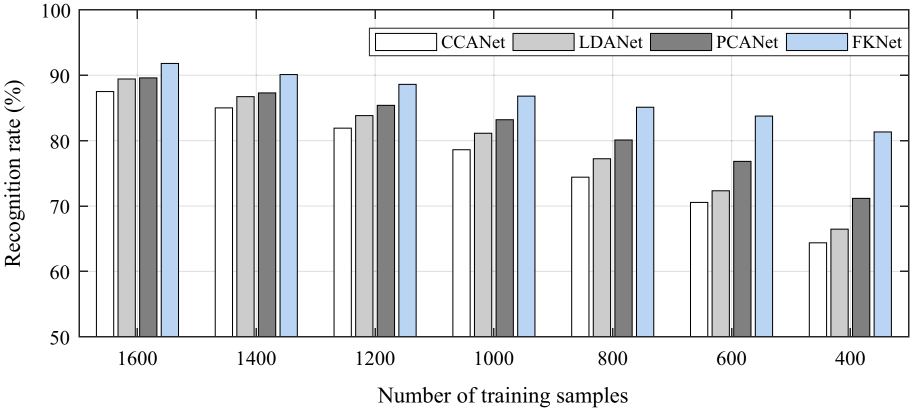

Fig. 3 Comparison of the different shallow networks when the training data is decreased

In general, FKNet attained the highest accuracy compared to the baselines. The discriminative capability of FKNet is evident when the number of layers increases. According to the experimental results, there is no significant improvement when the baselines employ 3 or 4 layers. These results may be due to the optimization model used by PCA, LDA, and CCA, which is based on dimensionality reduction. Such an approach eliminates a substantial amount of data, so that discriminative information may be lost, presenting no opportunity for the other layers to learn. In addition, after 3 layers, PCANet and DCTNet no longer improve their accuracy and CCANet even worsens its results. It is important to mention that these three methods do not make use of discriminative information among different handwritten image classes, which can be the reason for the low result when these networks are equipped with more than 3 layers.

The difference between the recognition rates achieved by FKNet and the other networks is even higher in the Semeion database, probably owing to the smaller amount of training data in this database compared to the USPS and C-cube databases. In this circumstance, FKNet benefits from the robustness inherited by FKT, which can produce efficient models with few training examples. In order to deeper analyze this aspect, the next series of experiments evaluates the impact of small-scale training datasets.

4.3 Comparing Shallow Networks Under Limited Training Data Conditions

In this experiment, we evaluate FKNet and three baselines (CCANet, LDANet and PCANet) under limited training data conditions. This experiment is essential to investigate the performance of these shallow networks under such circumstance, since many practical problems can only be solved when the learning model is appropriately designed to handle scarce training data. We evaluate FKNet, CCANet, LDANet and PCANet because the convolutional kernels of these networks are data-dependent. For this evaluation, we equip the networks with 4 convolutional layers, following the configuration 4(p) from Table 1.

We use the Semeion database, since it presents the lowest amount of data compared to the other handwritten image datasets investigated in Sect. 4.2, which is a realistic and challenging scenario. We employ a holdout strategy to evaluate the performances of the methods. The amount of training samples varies from 400 to 1400. The remaining data was used for testing. In each case, we randomly select the training data and repeat the experiment

10 times. We report the average classification rates attained in each scenario. The parameters of the networks, such as number of filters and convolutional kernel size were set as was done in the previous series of experiments. The number of convolutional layers of each network was set to 4, where a pooling layer is added after the third convolutional layer.

Figure 3 displays the average classification rates obtained by the networks in different training data scenarios. As shown in this figure, we can see that the overall performance of FKNet was better than that of the other shallow networks. In particular, FKNet works well when the training data is limited. More precisely, when only 400 samples are available for training, FKNet presents a recognition rate of 80%, which is about 10% higher than the recognition rate produced by PCANet (the best baseline). This is a challenging experiment for shallow networks, since not only the amount of training data is reduced, but also the amount of test data increases accordingly.

This experimental result suggests that FKT does not suffer from the issue known as the small sample size (SSS) problem that occurs when the number of features is larger than the number of instances [48]. Differently, LDA is sensitive to the number of training samples, which reflects the poor performance achieved by LDANet. Indeed, LDA and CCA theoretical formulations present the SSS problem, which is not present on FKT formulation. Since the CCA model depends on the correlation between a pair of training samples, the reduced training data directly affects its performance.

4.4 Comparison with Convolutional Neural Network

Motivated by the previous results, experiments are conducted using the EMNIST database, which is a dataset of segmented cursive letters and handwritten digits. The purpose of this series of experiments is to compare FKNet to a conventional CNN in terms of recognition rate, by varying the number of layers. This analysis is essential to define the advantages and limitations of the proposed method compared to CNN.

In addition to CNN, we also compare the proposed method with the state-of-the-art methods in handwritten character recognition. We compare our method with Text Capsule Networks (TextCaps) [49] and genetic Deep CNN (genetic DCNN) [50] due to their high recognition rates and novel training approaches. TextCaps is a character recognition method that provides high accuracy rates in EMNIST and is able to employ small training sets. Considering that many languages do not present handwritten character datasets with an adequate number of samples to train deep learning models, TextCaps generates augmented handwriting images, increasing the number of training samples by handling random controlled noise. Distinct from other CNN methods, genetic DCNN is an autonomous learning algorithm that automatically produces a DCNN architecture employing the data available for a specific image classification task. To this aim, genetic DCNN applies evolutionary operations, including selection, mutation and crossover to evolve a population of DCNN architectures. The performance of genetic DCNN is comparable to the state-of-the-art DCNN models.

EMNIST is derived from the NIST (National Institute of Standards and Technology) Special Database 19. In this dataset, images are normalized to 28×28 pixels. In total, it is composed of 280,000 characters divided into 62 classes, comprising 10 digit, 26 lowercase letter, and 26 uppercase letter classes. In this series of experiments, the dataset is divided into five partitions: (1) dataset A, composed of only the 26 uppercase letter classes; (2) dataset B, that includes only the 26 lowercase letter classes; (3) dataset C, composed of only the 10 digit classes (EMNIST-Digits); (4) a dataset D that includes all the 62 classes; (5) a dataset E that includes the uppercase and lowercase letter classes (EMNIST-Letters).

For comparison purposes, the employed CNN architecture is composed of 4 convolutional layers with 16, 20, 20 and 24 convolutional kernels, respectively. The convolutional kernel size is set to 7×7. The first and the third convolutional layers are followed by a 2×2 average pooling layer. Thus, the output features are provided to a fully connected layer in order to produce the final recognition score. In order to train this CNN, we employ Adam using mini-batch SGD (100 epochs) with the following hyper-parameters: learning rate $\alpha = 0.001$, and $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$. The mini-batch size was set to 20.

The hyper-parameters of FKNet are as follows: 4 convolutional layers consisting of 12, 12, 20 and 20 convolutional kernels with size 7×7. The dimension of each subspace class was set to $\mu(r_i) \leq 0.9$, according to Eq. (2).

Table 3 lists the recognition rates attained by CNN and FKNet when different numbers of layers are employed. It is observed that CNN presents the highest recognition rate with dataset C, which consists of only handwritten digits. This result may be a consequence of the reduced number of classes found in the dataset C. Because of the low complexity in terms of inter-class separability, CNN can efficiently extracts discriminatory elements.

In terms of the results attained by TextCaps and genetic DCNN (these results were obtained from the respective papers) on dataset C and dataset D, their results are 5% and 6% superior to the ones provided by the CNN and FKNet. The accuracy achieved by TextCaps is the result of a sophisticated technique based on data augmentation, which is not implemented in CNN nor in the proposed method. Besides, genetic DCNN achieves the same level of accuracy as TextCaps, with an approach that uses genetic algorithms to create its architecture. Therefore, both methods present very competitive results, but at the cost of requiring computational complexity far superior when compared to FKNet, during the training process.

Another observation is that CNN outperforms FKNet when only one layer is employed. It is noticeable that a single subspace representing the correlation among different classes may not be sufficient to resemble the complexity of the distributions. Therefore, to improve the FKNet capability, more constraint subspaces must be considered when only one layer is available.

An attempt to equip FKNet with more than 4 convolutional layers has also been established and the results show that the recognition rate achieved by the proposed network does not improve, reaching the learning limits of this shallow network. This result is directly derived from the fact that the linear subspaces employed to represent the image classes do not preserve its nonlinear relations. For instance, an image set distribution of a given class may be better represented by multiple subspaces if their distribution is multimodal. Different from FKNet, the accuracy of CNN boosts 0.7% when the network is equipped with 6 layers.

Experimental results of processing time measurement by each network are provided as follows. For reasons of reproducibility, the processing times reported in this paper were obtained using a computer equipped with an intel core i7 2.2 GHz quad-core processor, including 32GB RAM. As the other experiments, we employed Matlab to run the experiments. According to the configuration adopted, CNN required about 3 h to generate the described model with 4 convolutional layers using the EMNIST training dataset. On the other hand, FKNet obtained a comparable model using less than 17 min on the same hardware, which is approximately one order of magnitude faster.

Besides its advantage in terms of training time, the testing time on processing each example in FKNet is much faster than the time required by CNN, since the number of convolutional layers employed by FKNet is $1/3$ lower than the employed by CNN. The number of convolutional kernels employed by FKNet is very compact, precisely because of the orthogonal nature of the subspace produced by FKT. For comparison purposes, using the same hardware

Table 3 Recognition rates on comparing CNN and state-of-the-art methods, where N.A. stands for not available

Methods	Dataset A	Dataset B	Dataset C	Dataset D	Dataset E
CNN-1	90.44 ± 0.22	82.91 ± 0.26	91.25 ± 0.22	64.93 ± 0.31	88.78 ± 0.31
CNN-2	90.59 ± 0.22	83.47 ± 0.24	92.66 ± 0.21	65.58 ± 0.32	88.93 ± 0.28
CNN-3	91.61 ± 0.22	83.68 ± 0.24	93.15 ± 0.21	65.97 ± 0.30	89.71 ± 0.28
CNN-4	92.17 ± 0.19	83.93 ± 0.23	93.67 ± 0.20	66.08 ± 0.30	90.47 ± 0.27
FKNet-1	86.39 ± 0.26	77.82 ± 0.29	86.77 ± 0.24	59.29 ± 0.33	84.15 ± 0.32
FKNet-2	89.58 ± 0.25	81.43 ± 0.25	89.71 ± 0.24	63.15 ± 0.33	87.54 ± 0.29
FKNet-3	92.07 ± 0.21	84.05 ± 0.25	91.53 ± 0.21	66.17 ± 0.31	90.05 ± 0.29
FKNet-4	**92.21** ± **0.21**	**84.33** ± **0.24**	91.93 ± 0.22	**66.39** ± **0.31**	90.23 ± 0.28
TextCaps [49]	N.A.	N.A.	**99.79** ± **0.11**	N.A.	95.36 ± 0.30
Genetic DCNN [50]	N.A.	N.A.	99.75±N.A.	N.A.	**95.58** ± N.A.

scenario discussed previously, CNN needs 81 seconds to process 1000 28 × 28 handwritten images, while FKNet requires only 34 s.

The shallow networks present an advantage when there are limitations in energy consumption or computational resources. A concrete example is when a neural network should be used in an embedded device, such as an FPGA [51,52]. In this scenario, hardware limitation is evident and conventional neural networks cannot be employed, since memory and processing resources are minimal. Besides, many applications require that the algorithms not only process data but also that the learning mechanism runs on the device itself. Under these circumstances, compact networks such as FKNet have a substantial advantage over conventional networks.

It is important to note that the CNN employed in this experiment could achieve slightly higher results, in the case of some adjusts on the convolutional kernel size. Instead, we chose to employ the same parameters as FKNet, since it presents a fair comparison. These results confirm that the proposed shallow network is an attractive alternative for handwritten character classification when processing time and memory requirements are application constraints. Particularly for application in scenarios critically affected by time or hardware restrictions.

4.5 Face Verification Using LFW Dataset

Lastly, we evaluate the proposed network using the LFW (Labeled Faces in the Wild) dataset [24] for unconstrained face verification. This experiment aims to explore the limitation of FKNet on a dataset different from handwritten datasets. The LFW dataset consists of images of faces collected from the web, where the faces were detected using the Viola-Jones face detector and cropped into 150 × 80 pixels. This dataset is especially difficult for the investigated shallow networks due to the fact that the data was collected under uncontrolled scenarios.

In addition to comparing FKNet with other shallow networks, we compared the proposed method to two face verification approaches, Fisher Vector Faces (FVF) [53] and Multiscale Binarized Statistical Image Features with Overlapping Blocks (MBSIF-OB) [54]. These methods are commonly employed in this task, producing very competitive results.

Fisher Vector Faces (FVF) [53] is a representation for faces, where densely SIFT features are extracted from the image followed by dimensionality reduction and Fisher Vector to encode the features. The work [53] introduced the study of densely sampled SIFT features, which achieved a high recognition rate on the LFW dataset.

The Multiscale Binarized Statistical Image Features with Overlapping Blocks (MBSIF-OB) [54] is a face recognition framework based on a feature fusion approach and flip-free distance. This framework applies the Binarized Statistical Image Features (BSIF) [55], which learn the filters by employing statistics of natural images. After extracting the features from an image using the BSIF, a dimensionality reduction strategy is employed, where the projected vectors are scored. Finally, the scores for different scales are fused using SVM.

In this experiment, the hyper-parameters of the shallow networks are as follows: 4 convolutional layers consisting of 8, 8, 10 and 10 convolutional kernels. The first and the third convolutional layers are followed by a 2×2 average pooling layer. The dimension of each subspace class was set to $\mu(r_i) \leq 0.9$, according to Eq. (2) and 15 × 13 for the non-overlapping block size.

We report the average result of the 10-fold cross-validation. Contrasting to the experimental setup reported in [12], we do not employ the square-root operation on the final feature to maintain consistency with the other experiments provided in this work. Table 4 lists the results

Table 4 Accuracy and standard deviation of the investigated shallow networks and face recognition methods when evaluated on the LFW dataset

PCANet [12]	LDANet [12]	CCANet [19]	FKNet	Fisher VF [53]	MBSIF-OB [54]
86.11 ± 0.81	86.27 ± 0.79	85.26 ± 0.91	87.89 ± 0.67	$93.10 \pm N.A.$	93.41 ± 0.36

of the proposed method, along with the investigated shallow networks and face verification approaches.

Although the studied shallow networks were not initially designed to represent face images, these networks provided reasonable results. As expected, FVF and MBSIF-OB presented the best results, which is approximately 7% more accurate than the shallow networks.

The proposed network is 5% less accurate than the methods designed for face verification, which is a competitive result considering that its primary purpose is handwritten digits classification. It is important to take into account that both FVF and MBSIF-OB employ handcrafted features, which may be more challenging to train and more computationally expensive than the shallow networks.

5 Conclusions and Future Directions

This paper presented the Fukunaga–Koontz network for handwritten character classification. In the proposed shallow network, Fukunaga–Koontz transform is employed to create efficient convolutional kernels in a CNN architecture. Experiments conducted on USPS handwritten digits, C-Cube, Semeion and EMNIST handwritten datasets demonstrated the applicability of the proposed network. The experimental results show that by employing Fukunaga–Koontz transform for convolutional kernels, FKNet provides competitive classification results, when compared to PCANet, LDANet, RandNet, DCTNet and CCANet. To show its flexibility, FKNet was evaluated on a face verification task using the LFW dataset. In this experiment, FKNet demonstrated to be competitive, where FVF, MBSIF-OB and other shallow networks were employed as baselines.

The proposed shallow network presents the following advantages: (1) light computational resources requirements for learning, (2) small set of parameters to be tuned and, (3) fast learning and processing times. This architecture requires the choice of just a few parameters: the convolutional kernel size, the number of layers, and the class subspace dimension.

Different from the compared shallow networks, the convolutional kernels employed by FKNet are equipped with pooling layers, which provides invariance to changes in position or lighting conditions while decreasing the feature dimensionality. This improvement, coupled with the fact that the constraint subspace produced by FKT produces more discriminative features than its counterparts, allows FKNet to be an appealing method both in performance and theoretical aspects. Since FKNet is entirely based on linear algebra and can be investigated through mathematical tools, this network offers an explicit interpretable model while presenting characteristics of modern neural networks, achieving competitive results in challenging handwritten character databases.

In the third series of experiments, it is observed that one benefit of using the proposed network is that the number of convolutional kernels employed is much smaller than the ones used by a CNN. Besides, FKNet inherits the fast processing time exhibited by the shallow networks investigated in this work, which is faster than the processing time obtained by CNN, suggesting that the proposed shallow network can replace CNN when processing time is a requirement.

Experimental results have revealed that FKNet is a potential choice when there are hardware limitations. Applications where there is a limitation of energy consumption require a compact learning model, such as in autonomous vehicles applications [56,57]. There are other applications whose requirements go beyond energy consumption. For example, in remote sensing, a neural network must adjust its parameters directly on the device, which is usually very limited. In both cases, FKNet is an advantageous alternative.

For future work, we will investigate the application of a nonlinear subspaces approach to reveal more discriminative structures. Also, we will evaluate FKNet on different computer vision problems, such as real-time training and testing document image classification, which may benefit from the proposed network as well.

Another potential research direction is the application of the convolutional kernels produced by FKT as an alternative to the random initialization process of the deep neural networks. In this direction, it is expected that the training time of a deep neural network can be reduced by exploiting the discriminative weights provided by FKT. The extension of the proposed network to handle tensor data may provide a fast solution for video analysis, gesture and action recognition. Tensor subspaces exist in literature and may provide convolutional kernels to such networks.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number 19K20335 and the Foundation for Research Support of the State of Amazonas (FAPEAM). We are grateful to the anonymous reviewers for their constructive comments that contributed to improving the revised version of the manuscript.

References

1. Han Z, Liu CP, Yin XC (2005) A two-stage handwritten character segmentation approach in mail address recognition. In: Proceedings of eighth international conference on document analysis and recognition, IEEE, pp 111–115
2. Palacios R, Gupta A, Wang PS (2004) Handwritten bank check recognition of courtesy amounts. Int J Image Gr 4(02):203–222
3. Wang T, Wu DJ, Coates A, Ng AY (2012) End-to-end text recognition with convolutional neural networks. In: 2012 21st International conference on pattern recognition (ICPR), IEEE, pp 3304–3308
4. Pradeep J, Srinivasan E, Himavathi S (2012) Neural network based recognition system integrating feature extraction and classification for english handwritten. Int J Eng Trans B Appl 25(2):99
5. Wang J-S, Chuang F-C (2012) An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition. IEEE Trans Ind Electron 59(7):2998–3007
6. Richarz J, Vajda S, Grzeszick R, Fink GA (2014) Semi-supervised learning for character recognition in historical archive documents. Pattern Recognit 47(3):1011–1020
7. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
8. Buscema M (1998) Metanet*: the theory of independent judges. Subst Use Misuse 33(2):439–461
9. Impedovo S (2014) More than twenty years of advancements on frontiers in handwriting recognition. Pattern Recognit 47(3):916–928

10. Zeiler MD, Krishnan D, Taylor GW, Fergus R (2010) Deconvolutional networks. In: 2010 IEEE conference on computer vision and pattern recognition (CVPR), IEEE, pp 2528–2535
11. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
12. Chan T-H, Jia K, Gao S, Jiwen L, Zeng Z, Ma Y (2015) Pcanet: a simple deep learning baseline for image classification? IEEE Trans Image Process 24(12):5017–5032
13. Ye Q, Doermann D (2015) Text detection and recognition in imagery: a survey. IEEE Trans Pattern Anal Mach Intell 37(7):1480–1500
14. Fukunaga K, Koontz WLG (1970) Application of the karhunen-loeve expansion to feature selection and ordering. IEEE Trans Comput 100(4):311–318
15. Fukunaga K (2013) Introduction to statistical pattern recognition. Academic press, New York
16. Fukui K, Yamaguchi O (2007) The kernel orthogonal mutual subspace method and its application to 3d object recognition. In: Asian conference on computer vision, Springer, pp 467–476
17. Maeda K (2010) From the subspace methods to the mutual subspace method. In: Computer vision, Springer, pp 135–156
18. Shimomoto EK, Souza LS, Gatto BB, Fukui K (2018) Text classification based on word subspace with term-frequency. In: 2018 International joint conference on neural networks (IJCNN), IEEE, pp 1–8
19. Xinghao Y, Weifeng L, Dapeng T, Jun C (2017) Canonical correlation analysis networks for two-view image recognition. Inf Sci 385:338–352
20. Klambauer G, Unterthiner T, Mayr A, Hochreiter S (2017) Self-normalizing neural networks. In: Advances in neural information processing systems, pp 971–980
21. Hull JJ (1994) A database for handwritten text recognition research. IEEE Trans Pattern Anal Mach Intell 16(5):550–554
22. Camastra F, Spinetti M, Vinciarelli A (2006) Offline cursive character challenge: a new benchmark for machine learning and pattern recognition algorithms. In: 18th International conference on pattern recognition, 2006. ICPR 2006, vol 2, IEEE, pps 913–916
23. Cohen G, Afshar S, Tapson J, van Schaik A (2017) Emnist: extending mnist to handwritten letters. In: 2017 International joint conference on neural networks (IJCNN), IEEE, pp 2921–2926
24. Huang GB, Ramesh M, Berg T, Learned-Miller E (2007) Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst
25. Tian S, Bhattacharya U, Lu S, Su B, Wang Q, Wei X, Lu X, Tan CL (2016) Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. Pattern Recognit 51:125–134
26. Vajda S, Rangoni Y, Cecotti H (2015) Semi-automatic ground truth generation using unsupervised clustering and limited manual labeling: application to handwritten character recognition. Pattern Recognit Lett 58:23–28
27. Surinta O, Karaaba MF, Schomaker LRB, Wiering MA (2015) Recognition of handwritten characters using local gradient feature descriptors. Eng Appl Artif Intell 45:405–414
28. Hao WL, Zhang Z (2016) Incremental pcanet: a lifelong learning framework to achieve the plasticity of both feature and classifier constructions. In: Advances in brain inspired cognitive systems: 8th international conference, BICS 2016, Beijing, China, November 28–30, 2016, Proceedings 8, Springer, pp 298–309
29. Ng CJ, Teoh ABJ (2015) Dctnet: a simple learning-free approach for face recognition. In: 2015 Asia-Pacific signal and information processing association annual summit and conference (APSIPA), IEEE, pp 761–768
30. Li Y, Sankaranarayanan AC, Xu L, Baraniuk R, Kelly KF (2014) Realization of hybrid compressive imaging strategies. JOSA A 31(8):1716–1720
31. Rajput GG, Anita HB (2010) Handwritten script recognition using dct and wavelet features at block level. IJCA (Special issue on RTIPPR) 3:158–163
32. Adamek T, O'Connor NE, Smeaton AF (2007) Word matching using single closed contours for indexing handwritten historical documents. Int J Doc Anal Recognit 9(2):153–165
33. Tan H, Gao Y, Ma Z (2018) Regularized constraint subspace based method for image set classification. Pattern Recognit 76:434–448
34. Fukui K, Maki A (2015) Difference subspace and its generalization for subspace-based methods. IEEE Trans Pattern Anal Mach Intell 37(11):2164–2177
35. Gatto BB, Waldir SS, dos Santos EM (2016) Kernel two dimensional subspace for image set classification. In: 2016 IEEE 28th International conference on tools with artificial intelligence (ICTAI), IEEE, pp 1004–1011
36. Gatto BB, dos Santos EM (2016) Image-set matching by two dimensional generalized mutual subspace method. In: 2016 5th Brazilian conference on tools with artificial intelligence (ICTAI), IEEE, pp 133–138

37. Chen S, Sanderson C, Harandi MT, Lovell BC (2013) Improved image set classification via joint sparse approximated nearest subspaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 452–459
38. Wang R, Guo H, Davis LS, Dai Q (2012) Covariance discriminative learning: a natural and efficient approach to image set classification. In: 2012 IEEE conference on computer vision and pattern recognition (CVPR), IEEE, pp 2496–2503
39. Ohkawa Y, Fukui K (2012) Hand-shape recognition using the distributions of multi-viewpoint image sets. IEICE Trans Inf Syst 95(6):1619–1627
40. Suryanto CH, Saigo H, Fukui K (2016) Structural class classification of 3d protein structure based on multi-view 2d images. IEEE/ACM Trans Comput Biol Bioinform 15:286–299
41. Suryanto CH, Xue JH, Fukui K (2016) Randomized time warping for motion recognition. Image Vis Comput 54:1–11
42. Bouzalmat A, Kharroubi J, Zarghili A (2014) Comparative study of pca, ica, lda using svm classifier. J Emerg Technol Web Intell 6(1):64–68
43. Delac K, Grgic M, Grgic S (2005) Independent comparative study of pca, ica, and lda on the feret data set. Int J Imaging Syst Technol 15(5):252–260
44. Binol H, Bilgin G, Dinc S, Bal A (2015) Kernel fukunaga-koontz transform subspaces for classification of hyperspectral images with small sample sizes. IEEE Geosci Remote Sens Lett 12(6):1287–1291
45. Souza LS, Gatto BB, Xue JH, Fukui K (2020) Enhanced grassmann discriminant analysis with randomized time warping for motion recognition. Pattern Recognit 97:107028
46. Boureau YL, Ponce J, LeCun Y (2010) A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 111–118
47. Graham B (2014) Fractional max-pooling. arXiv preprint: arXiv:1412.6071
48. Duda RO, Hart PE, Stork DG (2012) Pattern classification. Wiley, New York
49. Jayasundara V, Jayasekara S, Jayasekara H, Rajasegaran J, Seneviratne S, Rodrigo R (2019) Textcaps: handwritten character recognition with very small datasets. In: 2019 IEEE winter conference on applications of computer vision (WACV), IEEE, pp 254–262
50. Ma B, Xia Y (2018) Autonomous deep learning: a genetic DCNN designer for image classification. arXiv preprint: arXiv:1807.00284
51. Baptista D, Abreu S, Travieso-González C, Morgado-Dias F (2017) Hardware implementation of an artificial neural network model to predict the energy production of a photovoltaic system. Microprocess Microsyst 49:77–86
52. Dehnavi M, Eshghi M (2017) Fpga based real-time on-road stereo vision system. J Syst Archit 81:32–43
53. Simonyan K, Parkhi OM, Vedaldi A, Zisserman A (2013) Fisher vector faces in the wild. In: BMVC, vol 2, p 4
54. Geng T, Yang M, You Z, Cai Y, Huang F (2018) Multiscale overlapping blocks binarized statistical image features descriptor with flip-free distance for face verification in the wild. Neural Comput Appl 30(10):3243–3252
55. Kannala J, Rahtu E (2012) Bsif: binarized statistical image features. In: Proceedings of the 21st international conference on pattern recognition (ICPR2012), IEEE, pp 1363–1366
56. Felipe Galindo Sanchez and Jose Nunez-Yanez (2017) Energy proportional streaming spiking neural network in a reconfigurable system. Microprocess Microsyst 53:57–67
57. Varagula J et al (2017) Object detection method in traffic by on-board computer vision with time delay neural network. Procedia Comput Sci 112:127–136