MOMENTSNET A SIMPLE LEARNING-FREE METHOD FOR BINARY IMAGE RECOGNITION

Jiasong Wu, Shijie Qiu, Youyong Kong, Yang Chen, Lotfi Senhadji, Huazhong Shu

MOMENTSNET: A SIMPLE LEARNING-FREE METHOD FOR BINARY IMAGE RECOGNITION

Jiasong Wu[1, 2, 3, 4], *Shijie Qiu*[1, 4], *Youyong Kong*[1, 4], *Yang Chen*[1, 4], *Lotfi Senhadji*[2, 3, 4], *Huazhong Shu*[1, 4]

[1]LIST, the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education,
210096 Nanjing, China
[2]INSERM, U1099, Rennes, F-35000, France
[3]Université de Rennes 1, LTSI, Rennes, F-35042, France
[4]Centre de Recherche en Information Médicale Sino-français (CRIBs)

ABSTRACT

In this paper, we propose a new simple and learning-free deep learning network named MomentsNet, whose convolution layer, nonlinear processing layer and pooling layer are constructed by Moments kernels, binary hashing and block-wise histogram, respectively. Twelve typical moments (including geometrical moment, Zernike moment, Tchebichef moment, etc.) are used to construct the MomentsNet whose recognition performance for binary image is studied. The results reveal that MomentsNet has better recognition performance than its corresponding moments in almost all cases and ZernikeNet achieves the best recognition performance among MomentsNet constructed by twelve moments. ZernikeNet also shows better recognition performance on a binary image database than that of PCANet, which is a learning-based deep learning network.

Index Terms**—**Deep learning, convolutional neural network, PCANet, MomentsNet, Zernike moment

1. INTRODUCTION

Convolutional neural networks (CNNs) with backpropagation process have shown its great success in various image classification tasks [1-6]. However, backpropagation process has very high computational complexity. To address this issue, convolutional networks without backpropagation have been proposed by many researchers. Chan *et al*. [7] proposed a lightweight and learning-based convolutional network named principle component analysis network (PCANet), who extracts the feature of the input images by using cascaded PCA filter, binary hashing, and block-wise histogram successively. PCANet does not use the backpropagation process but works unexpectedly well in most image classification tasks. Due to the success of PCANet, many researchers try to modify and extend this network in recent years [8-12]. Ng and Teoh [8] proposed a discrete cosine transform network (DCTNet) for face recognition. Feng et al. [9] proposed a discriminative locality alignment network (DLANet) for scene classification. Yang et al. [10] proposed a canonical correlation analysis network (CCANet) for two-view image recognition.

On the other hand, image moments [13-23] are widely used in binary image recognition due to their invariant representations of images. The image moments have a wide range of orthogonal and nonorthogonal basis functions and are simple to compute whatever the order requires. The image sampling can be either rectangular or polar, based on uniform or nonuniform lattices. The researchers conducted on various moment families which showed the efficiency of the moments. Geometric moments are the simplest moments but they are not orthogonal. Teague [13] introduced the continuous orthogonal moments (COTs) for which Legendre moments and Zernike moments are two typical representatives. To overcome the discretization error problem of the COTs, researchers proposed various discrete orthogonal moments, including Tchebichef moments [14], Krawtchouk moments [15] and dual Hahn moments [16], etc. In order to get a more invariant image representation, Yap et al. [17] produced a set of polar harmonic transforms (PHTs), including polar complex exponential transform (PCET), polar cosine transform (PCT), and polar sine transform (PST). Hoang and Tabbone [18] then presented a set of generic PHTs (GPHTs), including generic PCET (GPCET), generic PCT (GPCT), and generic PST (GPST). Note that many fast algorithms for example, [19-21] are proposed for moments due to their learning-free (or independence of input data) properties, while deriving the fast algorithms for learning-based PCA is very difficult. For more references on moments, we refer to [24-26].

In this paper, we propose a novel simple and learning-free deep learning network named MomentsNet for extracting the features of the input images. Learning-free deep learning network means that all the three layers (the convolution layer, nonlinear processing layer and pooling layer) of the network are independent of input data.

The rest of the paper is organized as follows. In Section 2, various moments are briefly introduced. The architecture of MomentsNet is described in Section 3. The recognition performance of MomentsNet is analyzed with respect to various parameters and also compared to moments, and PCANet in Section 4. Section 5 concludes the paper.

2. PRELIMINARIES

The general definition of two-dimensional (*n*+*m*)th order moment is given by

$$\begin{aligned}\Psi_{nm} &= \iint_{R^2} \varphi_{nm}(x,y) f(x,y)dxdy \\ &= \int_0^{2\pi}\int_0^1 \varphi_{nm}(r,\theta) f(r,\theta) rdrd\theta, \quad n,m=0,1,2,...\end{aligned} \tag{1}$$

, where $\varphi_{nm}(x,y)$ and $\varphi_{nm}(r,\theta)$ are moments kernels in Cartesian coordinates and in polar coordinates, respectively. $f(x,y)$ and $f(r,\theta)$ are image values in Cartesian coordinates and in polar coordinates, respectively. A summary of the moments used in this paper is given in Table 1.

3. THE ARCHITECTURE OF MOMENTSNET

The architecture of the proposed MomentsNet is summarized in Fig. 1. We then take the MomentsNet-2, in which the first two convolutional layers are constructed by moments, as an example to describe in details.

Table 1. The definitions of various moments

Moments	Definition	Moments	Definition
Geometric	$G_{nm} = \iint_{\mathbb{R}^2} x^n y^m f(x,y)dxdy$	Legendre [13]	$L_{nm} = \frac{(2n+1)(2m+1)}{4}\int_{-1}^{1}\int_{-1}^{1} P_n(x)P_m(y)f(x,y)dxdy,$ $P_n(x) = \frac{1}{2^n}\sum_{k=0}^{n/2}(-1)^k \frac{(2n-2k)!}{k!(n-k)!(n-2k)!}x^{n-2k}.$
Zernike [13]	$Z_{nm} = \frac{n+1}{\pi}\int_0^1\int_0^{2\pi} E_{nm}(r)e^{-jm\theta}f(r,\theta)rdrd\theta,$ $0 \le \lvert m \rvert \le n, n-\lvert m \rvert \; is \; even,$ $E_{nm}(r) = \sum_{k=0}^{(n-\lvert m \rvert)/2}(-1)^k \frac{(n-k)!}{k![(n-2k+\lvert m \rvert)/2]![(n-2k-\lvert m \rvert)/2]!}r^{n-2k}$	Tchebichef [14]	$D_{nm} = [\rho(n,N)\rho(m,M)]^{-1}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} t_n(x)t_m(y)f(x,y),$ $t_n(x) = n!\sum_{k=0}^{n}(-1)^{n-k}\binom{N-1-k}{n-k}\binom{n+k}{n}\binom{x}{k},$ $\rho(n,N) = \left[N(N^2-1)(N^2-2^2)\cdots(N^2-n^2)\right]/(2n+1).$
Krawtchouk [15]	$Q_{nm} = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\bar{K}_n(x;p_1,N-1)\bar{K}_m(y;p_2,M-1)f(x,y),$ $\bar{K}_n(x;p,N) = K_n(x;p,N)\sqrt{w(x;p,N)/\rho(n;p,N)},$ $w(x;p,N) = \binom{N}{x}p^x(1-p)^{N-x}, \; \rho(n;p,N) = \left(\frac{p-1}{p}\right)^n \frac{n!}{(-N)_n},$ $K_n(x;p,N) = {}_2F_1\left(-n,-x;-N;1/p\right),$ ${}_2F_1\left(a,b;c;z\right) = \sum_{k=0}^{\infty}\frac{(a)_k(b)_k}{(c)_k}\frac{z^k}{k!},$ $(a)_k = a(a+1)(a+2)...(a+k-1) = \Gamma(a+k)/\Gamma(a).$	Dual Hahn [16]	$W_{nm} = \sum_{x=a}^{b-1}\sum_{y=a}^{b-1}\hat{w}_n^{(c)}(x,a,b)\hat{w}_m^{(c)}(y,a,b)f(x,y), m,n = 0,1,2,...,N-1$ $\hat{w}_n^{(c)}(x,a,b) = w_n^{(c)}(x,a,b)\sqrt{\frac{\rho(x)}{d_n^2}[\Delta x(x-\frac{1}{2})]},$ $w_n^{(c)}(x,a,b) = \frac{(a-b+1)_n(a+c+1)_n}{n!}$ $\times {}_3F_2(-n,a-x,a+x+1;a-b+1,a+c+1;1),$ $\rho(x) = \frac{\Gamma(a+x+1)\Gamma(c+x+1)}{\Gamma(x-a+1)\Gamma(b-x)\Gamma(b+x+1)\Gamma(x-c+1)},$ $d_n^2 = \frac{\Gamma(a+c+n+1)}{n!(b-a-n-1)!\Gamma(b-c-n)},$ ${}_3F_2(a_1,a_2,a_3;b_1,b_2;z) = \sum_{k=0}^{\infty}\frac{(a_1)_k(a_2)_k(a_3)_k}{(b_1)_k(b_2)_k}\frac{z^k}{k!},$ $-1/2 < a < b, \; \lvert c \rvert < 1+a, \; b = a+N.$
PCET [17]	$C_{nm} = \int_0^{2\pi}\int_0^1 [H_{nl}(r,\theta)]^* f(r,\theta)rdrd\theta, \quad \lvert n \rvert = \lvert m \rvert = 0,1,\cdots,\infty,$ $H_{nm}(r,\theta) = R_n(r)e^{im\theta}, \; R_n(r) = e^{i2\pi nr^2},$ where $[.]^*$denotes the complex conjugate.	GPCET [18]	$O_{nms} = \int_0^{2\pi}\int_0^1 [V_{nm}(r,\theta)]^* f(r,\theta)rdrd\theta,$ $V_{nm}(r,\theta) = R_{ns}(r)e^{im\theta}, \; R_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}e^{i2\pi nr^s}$
PCT [17]	$C_{nm}^C = \int_0^{2\pi}\int_0^1 [H_{nm}^C(r,\theta)]^* f(r,\theta)rdrd\theta, \; n, \lvert m \rvert = 0,1,\cdots,\infty,$ $H_{nm}^C(r,\theta) = R_n^C(r)e^{im\theta}, \quad R_n^C(r) = \begin{cases} 1, n=0 \\ \sqrt{2}\cos(\pi nr^2), n>0 \end{cases}$	GPCT [18]	$O_{nms}^C = \int_0^{2\pi}\int_0^1 [V_{nm}^C(r,\theta)]^* f(r,\theta)rdrd\theta$ $V_{nm}^C(r,\theta) = R_{ns}^C(r)e^{im\theta}, \; R_{ns}^C(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\begin{cases} 1, \quad n=0 \\ \sqrt{2}\cos\left(\pi nr^s\right), n>0 \end{cases}$
PST [17]	$C_{nm}^S = \int_0^{2\pi}\int_0^1 [H_{nm}^S(r,\theta)]^* f(r,\theta)rdrd\theta, \; n = 1,\cdots,\infty, \lvert m \rvert = 0,1,\cdots,\infty$ $H_{nm}^S(r,\theta) = R_n^S(r)e^{im\theta} = \sin(\pi nr^2)e^{im\theta},$	GPST [18]	$O_{nms}^S = \int_0^{2\pi}\int_0^1 [V_{nm}^S(r,\theta)]^* f(r,\theta)rdrd\theta$ $V_{nm}^S(r,\theta) = R_{ns}^S(r)e^{im\theta}, \; R_{ns}^S(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\sqrt{2}\sin\left(\pi nr^s\right), \; n>0$

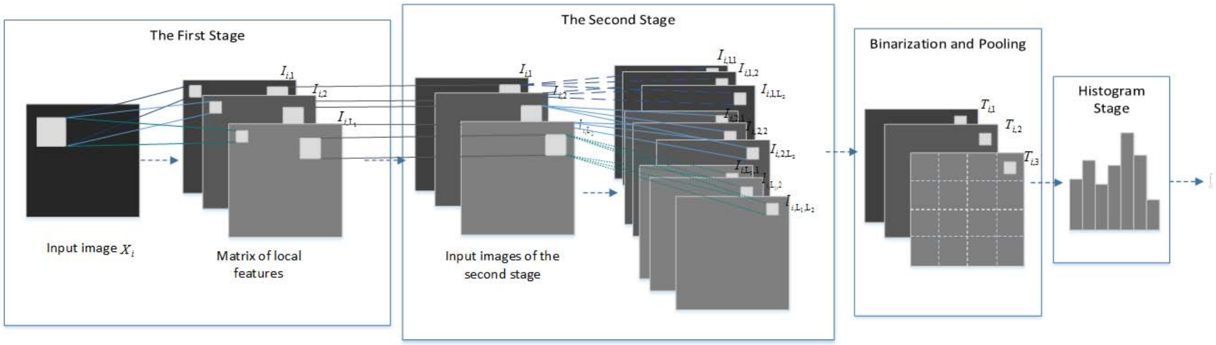


Fig. 1 The Architecture of MomentsNet-2

3.1. The first stage of MomentsNet

Suppose that we have a set of N_1 image samples $X_i \in \mathbb{R}^{M\times N}$, i=1,2,…,N_1, for training. We slide these images by using a patch, whose size is $k_1\times k_2$ and $1<k_1<M$, $1<k_2<N$. We collect all the overlapping patches around each pixel of each image and then center each patch by subtracting its mean and obtain $I_i = \left[\overline{x}_{i,1} \quad \overline{x}_{i,2} \quad \cdots \quad \overline{x}_{i,mn}\right] \in \mathbb{R}^{k_1 k_2 \times MN}$ for the ith image sample. Then we reshape all the vector $\overline{x}_{i,j} \in \mathbb{R}^{k_1 k_2}$, j=1, 2, …, MN, in I_i to the matrix $U_{i,j} \in \mathbb{R}^{k_1\times k_2}$.

By applying Eq. (1) to $U_{i,j}$, we can get n_k moments coefficients for each patch, where n_k denotes the total number of moments values of orders 0, 1, ..., k. Then combining moments coefficients of each patch in the ith image sample to form a matrix $A_i \in \mathbb{R}^{n_k \times MN}$. Let L_l be the desired number of filters in the lth layer, we get the first L_1 rows of matrix A_i and reshape all the L_1 rows to matrices $I_{i,j} \in \mathbb{R}^{M\times N}$, j=1, 2, … , L_1. Therefore, {X_i, i=1,2,…,N_1} is transformed into N_1L_1 moments feature maps {$I_{i,j}$, i=1,2,…,N_1; j=1,2,…,L_1} according to the number of filters L_1.

It turns out that each element in {$I_{i,j}$, j=1,2,…,L_1} can also be used as an input pattern. By repeating the above process, high-level features can be derived.

3.2. The second stage of MomentsNet

Assume that we have L_2 filters in the second stage. By repeating the same process as the first stage, for each $I_{i,j} \in \mathbb{R}^{M\times N}$, we can get L_2 moments feature maps { $I_{i,j,h} \in \mathbb{R}^{M\times N}$, h=1,2,…,L_2}. Therefore, {$I_{i,j}$, i=1,2,…,N_1; j=1,2,…,L_1} is transformed into $N_1L_1L_2$ moments feature maps {$I_{i,j,h}$, i=1,2,…,N_1; j=1,2,…,L_1; h=1,2,…,L_2} after the second stage of MomentsNet.

3.3. The output stage of MomentsNet

The moments feature maps extracted from the second stage are binarized, weighted, and summed to reduce the redundancy features.

Firstly, each element in {$I_{i,j,h}$, i=1,2,…,N_1; j=1,2,…,L_1; h=1,2,…,L_2} is binarized by using a modified Heaviside function defined as $H(x) = \begin{cases} 0, & x < t \\ 1, & x \ge t \end{cases}$, where t is a threshold. Note that MomentsNet is different from PCANet in this step since PCANet uses the original Heaviside function, that is, t=0. Why we use the modified Heaviside function? Because the values of some moments (for example, Zernike moments) are always larger than zero which makes t=0 is unsuitable. After this step, we get the binarized features {$J_{i,j,h} \in \mathbb{R}^{M\times N}$, i=1,2,…,N_1; j=1,2,…,L_1; h=1,2,…,L_2}.

Since different moment features can capture different variations of the original images, the binarized features should be weighted to form new single moments features as: $T_{i,j} = \sum_{k=1}^{L_2} 2^{k-1} J_{i,j,k}$, i=1,2,…,N_1; j=1,2,…,L_1. The pixel values of $T_{i,j}$ are integers in the range of [0, 2^{L_2-1}].

Next, we use a block of size $h_1\times h_2$ to slide each of the L_1 images $T_{i,j}$, j=1,2,…,L_1, with overlap ratio R. Then, the features are divided into B boxes and we compute the histogram of the decimal values for each box and denote it as $hist(B)_d$, d=1,2,…,B. After this pooling process, we concatenate all the histograms of B boxes into one vector and obtain

$$f_i = [Bhist(T_{i,1}), Bhist(T_{i,2}), ..., Bhist(T_{i,L_1})]^T \in \mathbb{R}^{(2^{L_2})L_1 B}$$

The feature vector is then sent to a SVM classifier [27] to get the final recognition results.

4. EXPERIMENTAL RESULTS

4.1. Experimental database

The database used in the following experiments is a set of binary images of 9 classes, including bird, camel, children, elephant, fork, hammer, key, ray and turtle. Each class contains 144 images and all the images are rotated from 0˚ to 330˚ with an increment of 30˚. Some image samples are shown in Fig. 2. All the images are then scaled to 32×32 to reduce the computational complexity in the experiments.

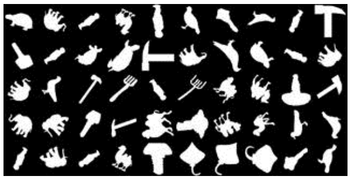
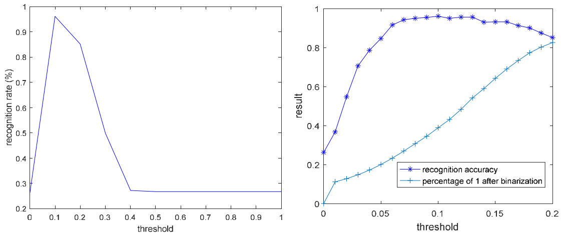


Fig. 2 Some Samples of Binary Image Database

We randomly pick up half of the images in each class as the training set and the others left as the testing set.

4.2. The impact of various parameters

In this subsection, we focus on the selection of various parameters. We take ZernikeNet as an example to show the impact of the parameters in the following.



(a) (b)

Fig. 3 The impact of threshold t. (a) the recognition rate versus the threshold t. (b) the "result" versus the threshold t. Note that the "result" denotes the recognition rate for the curve above and the percentage of 1 after binarization for the curve below.

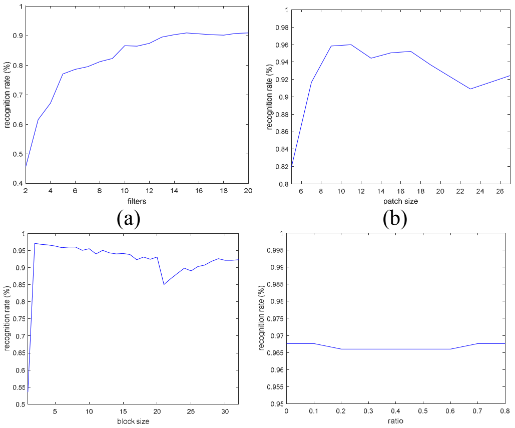
The impact of threshold t:

From our experiments, the threshold t in modified Heaviside function has a major effect on the final recognition result. To discuss the effect of threshold t, we firstly set the number of filters, patch size, block size and overlapping ratio to 9, 11×11, 8×8, and 0.5, respectively. The experiments are carried out on the one-layer Zernike Moments networks (ZernikeNet-1) for simplicity. Fig. 3(a) shows the recognition rates vary with the change of threshold t. We can see that there is a dramatic change in the recognition rate as the threshold t changes and the recognition rate is relatively good when 0.1≤ t ≤0.2. A question is raised: *Is there any other parameters that help to choose the threshold value t*? Since t

decides the proportion of the number of 1 and 0 in the image after the binarization, we thus additionally record the percentages of the number 1. Fig. 3(b) is the recognition result or the percentage of 1 after binarization vary with the threshold changes from 0 to 0.2 with step 0.01. From the figure, we can find that when the percentage of 1 is in the range of [0.3, 0.5], the thresholds correspond to good recognition results. This is in fact a good guidance for the choice of threshold value t not only for ZernikeNet-1 but also for ZernikeNet-2. For ZernikeNet-1, we set the threshold t to 0.1.

The impact of the number of filters L_l, patch size $k_1 \times k_2$, block size $h_1 \times h_2$, and overlapping ratio R:

Fig. 4 shows the recognition results as the parameters change. It is apparent that the recognition rate tends to be better with the increase of the number of filters L_1, but the performance does not improve greatly when $L_1 > 13$. The recognition rate tends to first rise and then fall with the increase of the patch size $k_1 \times k_2$ and the block size $h_1 \times h_2$. ZenikeNet-1 can get the best performance when the patch size and the block size are 11×11 and 4×4, respectively. Comparing with other parameters, the overlapping ratio R has little impact on the recognition rate. We simply set R=0.5. ZenikeNet-1 achieves the recognition accuracy of 96.72% with the aforementioned parameter setting.



(a) (b)

(c) (d)

Fig. 4 The recognition rates versus various other parameters. (a) the recognition rate versus the number of filters; (b) the recognition rate versus the patch size; (c) the recognition rate versus the block size; (d) the recognition rate versus the overlap ratio.

4.3. Experimental results

We compare the performance of the proposed MomentsNet with the moments and also the PCANet [7]. The best recognition rates and their corresponding parameter settings are shown in Table 2. Note that the comparison results are averaged over 5 independent experiments. Note also that the accuracy of moments is obtained by extracting the features by using various moments and then feed the features into the SVM classifier [27].

From Table 2, we can see that MomentsNet has better recognition performance than its corresponding moments in almost all cases and ZernikeNet performs the best and GeometryNet performs the worst in terms of the recognition performance among MomentsNet constructed by twelve moments. Most learning-free MomentsNet shows better recognition performance than that of PCANet [7], which is a learning-based deep learning network. We also find that the recognition performance of some MomentsNet-1 outperforms its corresponding MomentsNet-2, for example, TchebichefNet. We think the reason is that the descriptive powers of these moments tend to its upper limit when they are used in the construction of one-stage network, therefore, adding more stage is not helpful for improving the recognition performance.

Table 2. The best recognition rates of different methods. Quintets (L_1, k_1, h_1, R, t) whose elements denote the number of filters, patch size, block size, overlap ratio, threshold, respectively. Note that we always set L_2=L_1, k_2=k_1, and h_2=h_1, in this paper.

Moments	The accuracy of Moments (%)	The accuracy of MomentsNet-1 (%)	The accuracy of MomentsNet-2 (%)
Geometry	12.11±1.01 (n+m=4)	68.09±3.51 (9, 13, 3, 0.5, 0)	67.32±3.98 (9, 13, 3, 0.5, 0)
Legendre	35.03±1.54 (n+m=16)	90.97±2.55 (11,11,3,0.6,-0.02)	91.98±2.62 (11, 11, 3, 0.5, 0)
Zernike	80.48±2.39 (n+m=16)	**96.72**±1.97 (10, 11, 2, 0.5, 0)	**97.60**±1.92 (10, 11, 2, 0.5, 0)
Tchebichef	57.60±1.20 (n+m=17)	91.86±1.35 (12, 13, 3, 0.4, 0.1)	88.08±2.43 (12, 13, 3, 0.5, 0)
Krawtchouk	57.37±1.12 (n+m=15)	88.51±2.24 (8, 13, 4, 0.5, 0)	92.13±1.70 (8, 13, 4, 0.5, 0.1)
Dual Hahn	56.94±2.79 (n+m=15)	91.32±2.51 (12, 13, 3, 0.5, 0)	91.48±2.04 (12, 15, 4, 0.5, 0)
PCT	94.10±2.04 (n+m=15)	93.25±0.50 (13,15,4, 0.7, 0.01)	95.64±2.28 (13, 15, 4, 0.5, 0.006)
PCET	85.15±3.12 (n+m=16)	95.49±1.58 (8, 13, 5, 0.7, 0.04)	97.30±0.39 (8, 13, 5, 0.5, 0.002)
PST	90.78±1.66 (n+m=14)	92.01±1.27 (8, 13, 3, 0.6, 0.02)	88.12±3.09 (8, 13, 3, 0.5, 0.0003)
GPCT	95.26±2.82 (n+m=17)	95.03±1.20 (8, 11, 6, 0.6, 0.1)	95.18±1.20 (8, 11, 6, 0.5, 0.01)
GPCET	91.94±2.81 (n+m=15)	95.26±1.04 (8, 15, 6, 0.8, 0.07)	96.68±0.70 (8, 15, 6, 0.5, 0.01)
GPST	95.56 ± 1.35 (n+m=15)	96.61±0.46 (12, 15, 3, 0.6, 0.1)	96.34±0.20 (12,15,3, 0.5, 0.012)
PCANet [7]	57.80±2.39 (325 principal components)	86.62±1.98 (8, 7, 7, 0.5, 0)	90.28±2.01 (8, 9, 8, 0.4, 0)

5. CONCLUSION

In this paper, we propose a simple and learning-free deep learning network named MomentsNet in which 12 different moments are explored in detail. The impact of various parameters on the recognition performance is analyzed in MomentsNet. MomentsNet has better recognition performance than its corresponding moments in almost all cases and also shows better recognition performance than that of PCANet in a typical binary image recognition task.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (No. 61201344, 61271312, 61401085, 31640028, 61572258, 11301074).

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[2] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1-9.

[3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv preprint arXiv:1312.6229, 2013, pp. 1-16.

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

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. CVPR*, 2015, pp. 1-9.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv preprint arXiv:1512.03385, 2015.

[7] T.H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017-5032, 2015.

[8] C.J. Ng and A.B.J. Teoh, "DCTNet: A simple learning-free approach for face recognition," in *Proc. APSIPA*, 2015, pp. 761-768.

[9] Z. Feng, L. Jin, D. Tao, and S. Huang, "DLANet: A manifold-learning-based discriminative feature learning network for scene classification," *Neurocomput.*, vol. 157, pp. 11-21, 2015.

[10] X. Yang, W. Liu, D. Tao, and J. Cheng. "Canonical correlation analysis networks for two-view image recognition," *Inf. Sci.*, vol. 385–386, pp. 338-352, 2017.

[11] R. Zeng, J.S. Wu, Z.H. Shao, Y. Chen, B.J. Chen, L. Senhadji and H.Z. Shu, "Color image classification via quaternion principal component analysis network," *Neurocomput.*, vol. 216, pp. 416-428, 2016.

[12] H. Qin, X. Li, J. Liang, Y. Peng, and C. Zhang, "DeepFish: Accurate underwater live fish recognition with a deep architecture," *Neurocomput.*, vol. 187, pp. 49-58, 2016.

[13] M.R. Teague, "Image analysis via the general theory of moments," *J. Opt. Soc. Am.*, vol. 69, no. 8, pp. 920–930, 1980.

[14] R. Mukundan, S. H. Ong and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1357–1364, 2001.

[15] P.T. Yap, R. Paramesran and S.H. Ong, "Image analysis by Krawtchouk moments," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1367–1377, 2003.

[16] H. Zhu, H. Shu, J. Zhou, L. Luo and J. L. Coatrieux, "Image analysis by discrete orthogonal dual Hahn moments," *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1688–1704, 2007.

[17] P.T. Yap, X. Jiang and A.C. Kot, "Two-dimensional polar Harmonic transforms for invariant image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1259-1270, 2009.

[18] T.H. Hoang and S. Tabbone, "Generic polar harmonic transforms for invariant image representation," *Image Vision Comput.*, vol. 32, no. 8, pp. 497-509, 2014.

[19] R. Mukundan and K.R. Ramakrishnan, "Fast computation of Legendre and Zernike moments," *Pattern Recognit.*, vol. 28, no. 9, pp. 1433-1442, 1995.

[20] H.Z. Shu, H. Zhang, B.J. Chen, P. Haigron, and L.M. Luo, "Fast computation of Tchebichef moments for binary and grayscale images," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3171-3180, 2010.

[21] B. Asli and J. Flusser, "Fast computation of Krawtchouk moments," *Inf. Sci.*, vol. 288, pp. 73-86, 2014.

[22] C.H. Teh and R.T. Chin, "On image analysis by the methods of moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 4, pp. 496–513, 1988.

[23] S.X. Liao and M. Pawlak, "On image analysis by moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 3, pp. 254–266, 1996.

[24] H.Z. Shu, L.M. Luo, and J.L. Coatrieux, "Moment-based approaches in imaging Part 1: basic features," *IEEE Eng. Med. Biol. Mag.*, vol. 26, no. 5, pp. 70-75, 2007.

[25] R. Mukundan and K.R. Ramakrishnan. *Moment Functions in Image Analysis: Theory and Application*. Singapore, World Scientific, 1998.

[26] J. Flusser, B. Zitova, and T. Suk. *Moments and Moment Invariants in Pattern Recognition*. John Wiley & Sons, 2009.

[27] R.E. Fan, K.W. Chang, C.J. Hsieh, X. R. Wang, and C. J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871-1874, 2008.