IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 2, 2019, accepted March 12, 2019, date of publication March 15, 2019, date of current version April 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2905304

Building Recognition Based on Sparse Representation of Spatial Texture and Color Features

BIN LI, FUQIANG SUN, AND YONGHAN ZHANG
School of Computer Science, Northeast Electric Power University, Jilin 132012, China

Corresponding author: Bin Li (libinjlu5765114@163.com)

This work was supported in part by the Science and Technology Development Plan Project of Jilin Province under Grant 20180520017JH, and in part by the Science and Technology Project of the Jilin Provincial Education Department of China under Grant JJKH20170107KJ.

ABSTRACT In this paper, we presented a novel building recognition method based on a sparse representation of spatial texture and color features. At present, the most popular methods are based on gist features, which can only roughly reflect the spatial information of building images. The proposed method, in contrast, uses multi-scale neighborhood sensitive histograms of oriented gradient (MNSHOGs) and color auto-correlogram (CA) to extract texture and color features of building images. Both the MNSHOG and the CA take spatial information of building images into account while calculating texture and color features. Then, color and texture features are combined to form joint features whose sparse representation can be dimensionally reduced by an autoencoder. Finally, an extreme learning machine is used to classify the combined features after dimensionality reduction into different classes. Several experiments were conducted on the benchmark Sheffield building dataset. The mean recognition rate of our method is much higher than that of the existing methods, which shows the effectiveness of our method.

INDEX TERMS Building recognition, texture and color features, extreme learning machine, autoencoder, Sheffield buildings database, sparse representation.

I. INTRODUCTION

Building recognition is a crucial step for many applications, such as mobile device localization [1], [2], architectural design, building images retrieval [3], [4], robot vision and navigation [5]. The changes of shooting distance, non-uniform illumination, partial occlusion and other factors make building recognition a challenging task. The existing building recognition methods can be classified into two categories: hand-crafted feature based and gist feature based methods.

The hand-crafted feature based methods are mainly rely on feature points, feature line segments, histograms to depict features of buildings. Zhang and Košecḱa [6] proposed a two-step building recognition approach. In the first step, localized color histograms are used to index model views. In the second step, SIFT descriptors are used to refine the building recognition results. In [7], SIFT descriptors are used to extract rotation invariant building features which is robust to different views of buildings. Hutchings and Mayol-Cuevas [8] use Harris corner detector to find key points for building recognition.

The gist feature based methods are proposed base on the following two reasons [9]: these hand-crafted feature based methods rely on only one or two low-level features, which make it difficult to express the deep semantic concepts of building images. The high-dimensional hand-crafted features may lead to high computational costs. Li and Allinson [9] proposed a building recognition method based on gist features and subspace learning algorithms. The gist features combine the three features of intensity, color and texture to address the limitation of single hand-crafted feature in expressing deep semantic concepts of building images. The gist feature extraction method used by Li and Allinson was proposed by Siagian and Itti. Siagian and Itti calculated 34 saliency maps from three building image feature channels: color, intensity and texture. Then, each saliency map is divided into 4 × 4 sub-regions, and a 16-dimensional feature vector is obtained by averaging each sub-region. A total of 34 feature vectors can be obtained from 34 saliency maps. This 544-dimensional

The associate editor coordinating the review of this manuscript and approving it for publication was Jinming Wen.

vector composed of 34 feature vectors is the gist feature proposed by Siagian and Itti. In order to reduce the computational costs of the 544-dimensional gist feature vector, some subspace learning algorithms such as principal component analysis (PCA) [10], locality preserving projections (LPP) [11] and margin fisher analysis (MFA) [12] are used to reduce the dimension of the gist feature vector. The following two studies are the improvement of Li and Allinson's research. In [13], a multi-scale gist feature which can reflect the structure information of building images was proposed. Then, an enhanced fuzzy local maximal marginal embedding subspace learning algorithm was proposed to reduce the dimension of the multi-scale gist features. Li *et al.* [14] used histogram of oriented gradient [15] instead of the gabor filter to extract texture features of buildings to enhance the texture description ability of gist features.

Gist feature vectors are obtained by partitioning saliency maps and averaging each sub-region. Through the feature extraction method, we can see that the gist feature is a rough feature. Gist features can only roughly reflect the spatial information of images by dividing them into 4 × 4 sub-regions. Therefore, gist features can only describe the color and texture distribution of building images very roughly. In this paper, we presented a novel building recognition method based on spatial texture and color features. We proposed multi-scale neighborhood sensitive histograms of oriented gradient (MNSHOG) which can well describe the spatial distribution information of image texture. The neighborhood sensitive histograms of oriented gradient (NSHOG) [16] is an image texture description method proposed in our previous work. The MNSHOG is the extended version of NSHOG, which can better adapt to building size scaling in images caused by different shooting distances. Huang *et al.* [17] proposed the color correlogram which can be used to describe the color distribution of an image. The color auto-correlogram (CA) is the simplified version of the color correlogram. Both MNSHOG and CA take spatial information of building images into account while calculating texture and color features. Due to sparse representation is robust to partial occlusion, the combined features of MNSHOG and CA are first raised to 1000 dimensions by the sparse layer of an four layer autoencoder [18], [19] to achieve the purpose of sparse representation, and then reduced by the under-complete layer of the autoencoder to capture the most prominent features. Finally, a robust and fast extreme learning machine (ELM) [20], [21] is used to classify the features after dimensionality reduction.

Compared with existing hand-crafted feature based and gist feature based methods, our proposed method has the following advantages:

1) The proposed method combines texture and color features, and uses autoencoder to reduce dimension and extract most prominent features. Our method compensates for the shortcomings of hand-crafted feature based method, such as weak expression ability of single hand-crafted feature and high feature dimension.
2) MNSHOG and CA can reflect the distribution of texture and color in images very fine, which makes up for the shortage of rough gist feature.
3) Multi-scale NSHOG can adapt to building size scaling in images caused by different shooting distances.

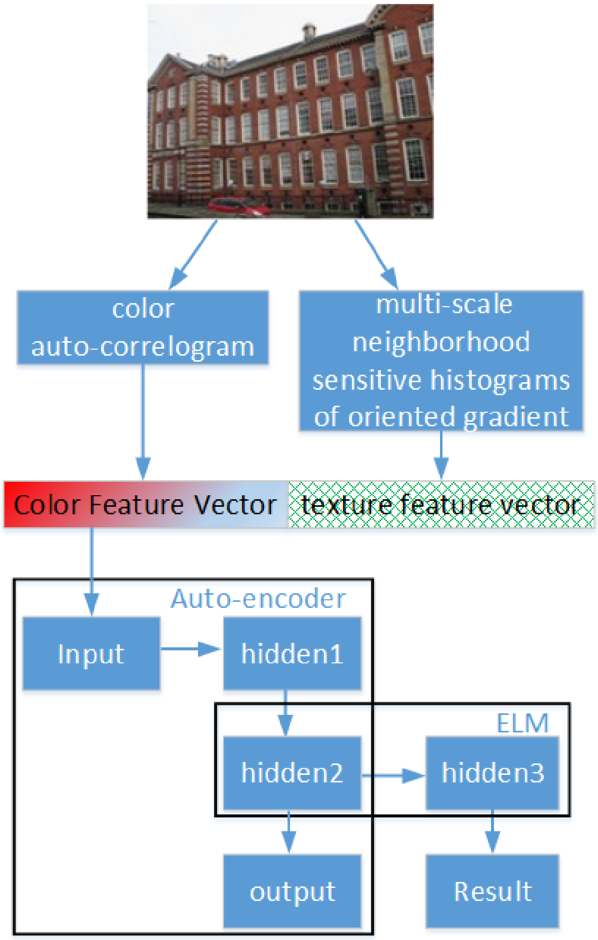
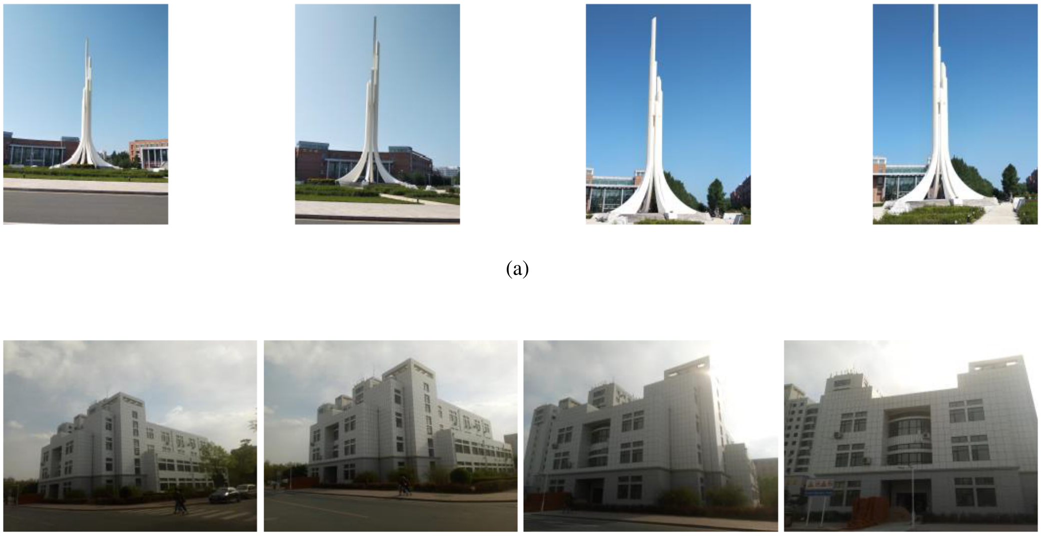


FIGURE 1. The steps of the proposed building recognition approach.

The rest of this paper is organized as follows: Section 2 describes the proposed multi-scale neighborhood sensitive histograms of oriented gradient and our building recognition approach in detail; Section 3 evaluates the proposed building recognition approach by conduct experiments on Sheffield building database; Section 4 concludes this paper.

II. THE PROPOSED APPROACH FOR BUILDING RECOGNITION

The main steps of the proposed method are shown in Figure 1. The proposed method consists of three steps: extracting spatial texture and color features using MNSHOG and CA respectively. The extracted color and texture feature vectors are combined to form a joint feature vector. A four layer autoencoder is used to elevate the dimension of the joint feature and then reduce its dimension. Finally, the ELM



(a)

(b)

FIGURE 2. **Building size scaling due to different shooting distances.**

obtains recognition results based on these reduced dimension feature vectors. The hidden1 and hidden2 in Figure 1 represent the first and second hidden layers of the autoencoder, and hidden2 is also the input layer of the ELM. The hidden3 in Figure 1 is the hidden layer of the ELM. Each step of the proposed method is described in the following sections.

A. NEIGHBORHOOD SENSITIVE HISTOGRAMS OF ORIENTED GRADIENT

In our previous work, we proposed neighborhood sensitive histograms of oriented gradient (NSHOG) [16] which has three steps.

1) Gradient templates [-1, 0, 1] and $[-1, 0, 1]^T$ are used to calculate the horizontal $G_x(x, y)$ and vertical $G_y(x, y)$ gradients of an image respectively.

2) According to (1) and (2), the magnitude $|G(x, y)|$ and orientation angle $\theta(x, y)$ of gradient at each pixel location are calculated respectively.

$$|G(x, y)| = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{1}$$

$$\theta(x, y) = \arctan \frac{G_y(x, y)}{G_x(x, y)} \tag{2}$$

3) For each pixel position p calculate a histogram of oriented gradient H_p which cover the whole image. The H_p contains 9 bin in the range of 0° $\sim$ 180°. For each bin b in H_p can be calculate by Equation (3).

$$H_p(b) = \sum_{q=1}^{m\times n} \alpha \cdot V_b(\theta_q, b), \quad b = 1 \ldots 9 \tag{3}$$

where $m \times n$ is the number of pixels of the image, and $V_b(\theta_q, b)$ can be define as (4). In Equation (3), the α is a sensitive factor which role is to reduce the contribution of the pixel q far from the current pixel p to the histogram H_p at p.

$$V_b(\theta_q, b) = \begin{cases} |G_q|, & if \quad \theta_q \in bin \quad b \\ 0, & otherwise \end{cases} \tag{4}$$

where $|G_q|$ and θ_q are the magnitude and orientation angle of gradient of pixel q, respectively. The NSHOG was originally used in face recognition, but building recognition may encounter building size scaling due to different shooting distances as shown in Figure 2. Therefore, we proposed multi-scale neighborhood sensitive histograms of oriented gradient (MNSHOG) to solve the problem of building size scaling in images.

B. MULTI-SCALE NEIGHBORHOOD SENSITIVE HISTOGRAMS OF ORIENTED GRADIENT

The calculation steps of MNSHOG are shown in Figure 3. For each building image, an image pyramid with three spatial scales is created. The first layer of the pyramid is the original image, the second layer is 90% of the original image, and the third layer is 80% of the original image. For each layer of the Pyramid, magnitude and orientation angle of gradient are calculated at each pixel location, and then a NSHOG about the whole image is calculated at each pixel location. Some uniformly distributed sampling points are generated at each layer of the pyramid, and the number of sampling points decreases with the decrease of image size. Finally, the histograms corresponding to all sampling points are linked together to form a feature vector, that is, the texture feature vector of a building.

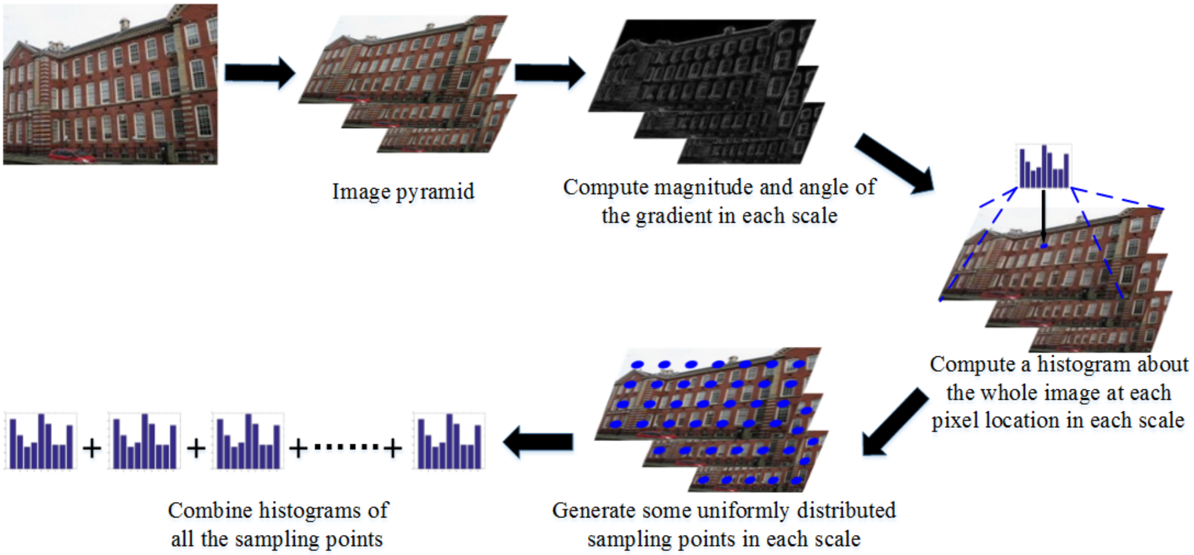


Image pyramid
Compute magnitude and angle of the gradient in each scale
Compute a histogram about the whole image at each pixel location in each scale
Generate some uniformly distributed sampling points in each scale
Combine histograms of all the sampling points
+ + +·······+

FIGURE 3. The steps of MNSHOG.

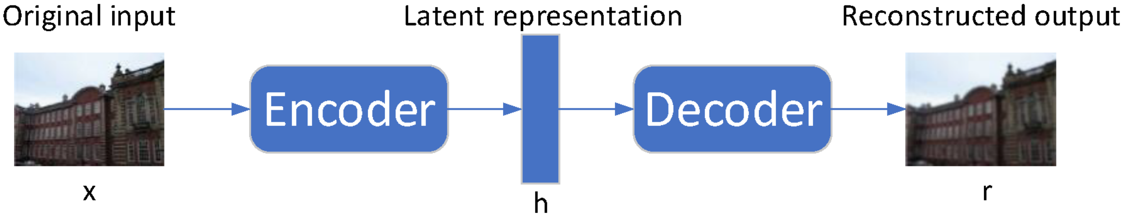


FIGURE 4. The structure of an autoencoder.

C. FEATURE EXTRACTION OF BUILDING IMAGES BASED ON MNSHOG AND CA

The NSHOG calculates a histogram of oriented gradient above a whole image at each pixel position, so NSHOG can well describe the spatial distribution of gradients in an image. As an extended version of NSHOG, MNSHOG has the additional ability to adapt to building size scaling in images. We set the number of sampling points of MNSHOG's three layers to 28, 18 and 15, respectively, so the dimension of MNSHOG vector is 549 ($(28 + 18 + 15) \times 9 = 549$).

The distribution of color in color image can be counted by color correlogram [17]. Suppose that the coordinates of any two pixels p_a and p_b in the image are (x_a, y_a) and (x_b, y_b), and the quantized image color space is $\{ c_1, c_2, \cdots c_m\}$. The distance between the pixels p_a and p_b is:

$$|p_a - p_b| = max\{|x_a - x_b|, |y_a - y_b|\} \tag{5}$$

The color correlogram of image I is defined as:

$$V^d_{c_ic_j}(I) = Pr_{p_a \in I_{c_i}, p_b \in I}\left|p_b \in c_j \middle| |p_a - p_b| = d\right| \tag{6}$$

where $V^d_{c_ic_j}$ denotes the number of pixels in image I whose distance from pixel p_a is d and color value is c_j. The time complexity of color correlogram is: $O(m^2d)$, which is relatively high. Therefore, the color auto-correlogram (CA) with lower time complexity ($O(md)$) is chosen to describe the color distribution of building images. The CA of image I is defined as: $a^d_{cc}(I) = V^d_{cc}(I)$. The CA uniformly quantized the RGB color space into 64 colors from which the 64-dimensional auto-correlogram feature vector can be calculated.

After extracting the texture and color feature vectors of a building image, we normalized the texture feature vector and the color feature vector into [0-1] intervals respectively. Then, we constructed a 613-dimensional combination vector consisting of a 549-dimensional texture feature vector and a 64-dimensional color feature vector.

D. FEATURE DIMENSIONALITY REDUCTION AND CLASSIFICATION

Autoencoder is a self-supervised and feedforward neural network [18], [19]. As shown in Figure 4, the autoencoder network consists of two parts: an encoder and a decoder. In the training process, the input data x is compressed into a latent-space representation by the encoder, which can be expressed as $h = f(x)$, and $f(x)$ is the encoding function. Then, the decoder reconstructs latent-space representation h, and the reconstructed result r is expressed as $r = g(h)$, $g(h)$

is the decoding function. The autoencoder can be described by function $g(f(x)) = r$. The training objective is to make the reconstructed output r close to the original input h. The training process of the autoencoder is to copy the input x to the output y and constantly modify the weight in the network. The autoencoder whose dimension of latent-space representation h is more than the dimension of input x is called sparse autoencoder. The autoencoder whose dimension of latent-space representation h is less than the dimension of input x is called under-complete autoencoder. By training a sparse representation h, we can get a sparse representation which is robust to partial occlusion. By training an under-complete representation h, we force the autoencoder to learn the most prominent features of training data while reducing the dimension of input x.

We use a four layer autoencoder with two hidden layers to elevate the dimension of the joint feature and then reduce its dimension. The output of the first hidden layer is the sparse representation of the input data x, and the output of the second hidden layer is the low-dimensional representation of the input data x. The second hidden layer is the output layer. In order to compare fairly with the gist feature based methods, in all the experiments in Section 3, the proposed method and all gist feature based methods reduce the feature dimension to 100 dimensional. Therefore, the number of nodes in the second hidden layer (output layer) of the autoencoder is set to 100.

Extreme Learning Machine (ELM) is an algorithm for solving single hidden layer neural networks proposed by Huang *et al.* [20], [21]. The connection weights between input layer and hidden layer and the threshold of hidden layer can be set at random, and they need not be adjusted after setting. The connection weights between hidden layer and output layer β does not need to be adjusted iteratively, but is determined at one time by solving a set of equations. Compared with BP [22] network and SVM [23], ELM has the following salient features: no parameters need to be manually tuned except the number of hidden layer nodes, faster learning speed, and higher generalization performance.

Suppose there are N samples (X_i, t_i), where $X_i = [x_{i1}, x_{i2}, \dots x_{in}]$ and $t_i = [t_{i1}, t_{i2}, \dots t_{im}]^T$. A single hidden layer neural network with L hidden layer nodes can be expressed as:

$$o_j = \sum_{i=1}^{L} \beta_i \cdot g(W_i \cdot X_j + b_i), \quad j = 1 \dots N \tag{7}$$

where g(x) is the activation function, $W_i = [w_{i1}, w_{i2}, \dots w_{in}]^T$ is connection weights between input layer and hidden layer, β_i is connection weights between hidden layer and output layer, b_i is the threshold of hidden layer node. The learning objectives of ELM is:

$$\sum_{j=1}^{N} ||o_j - t_j|| = 0 \tag{8}$$

where t_j can be considered as building ID. Equation 7 can be expressed by matrix multiplication as follows:

$$H \cdot \beta = T \tag{9}$$

where H is the output matrix of hidden layer nodes and T is the expected output.

$$H\{W_1, \cdots, W_L, b_1, \cdots, b_2, X_1, \cdots, X_L\} = \begin{bmatrix} g(W_1 \cdot X_1 + b_1) & \cdots & g(W_L \cdot X_1 + b_L) \\ \vdots & \vdots & \vdots \\ g(W_1 \cdot X_N + b_1) & \cdots & g(W_L \cdot X_N + b_L) \end{bmatrix} \tag{10}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix} \qquad T = \begin{bmatrix} T_1^T \\ \vdots \\ T_L^T \end{bmatrix} \tag{11}$$

The connection weights $\hat{\beta}$ between hidden layer and output layer can be calculated by Equation 12.

$$\hat{\beta} = H^+ T \tag{12}$$

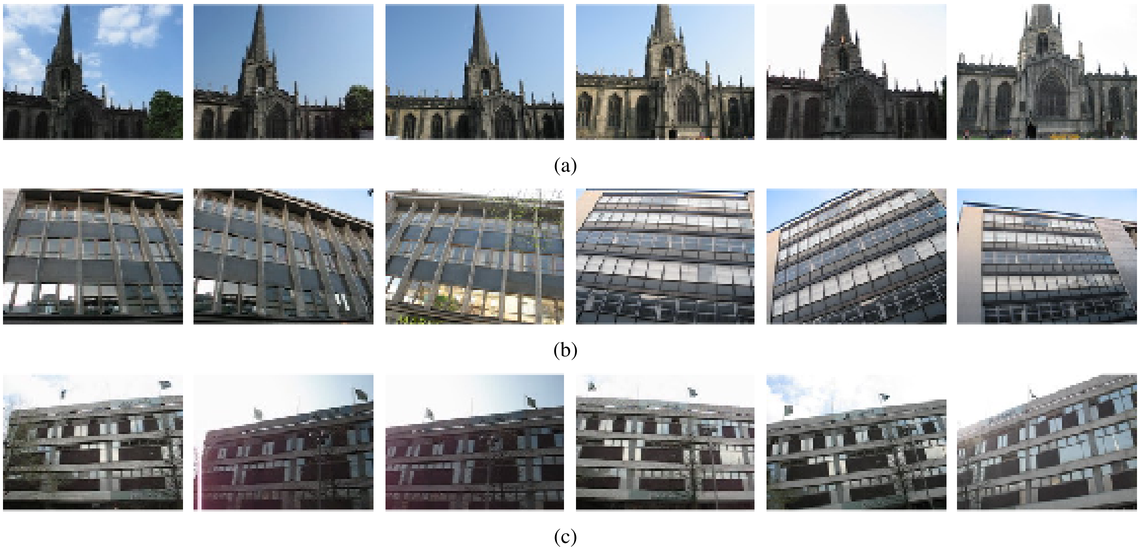
where H^+ is the generalized inverse matrix of H. In the classification step, we can get the classification result by $Y = H^+ T$, and we can get the result building ID by Equation 13.

$$t_{ID} = \arg\max_j (Y) \tag{13}$$

III. EXPERIMENTS AND ANALYSIS

In order to evaluate the proposed building recognition method, three experiments were carried out. In section 3.1, we shown the effect of the number of ELM's hidden layer nodes on the recognition rate, and selected the appropriate number of ELM's hidden layer nodes according to the experimental results. In Section 3.2, we compared the performance of the proposed method and existing building recognition methods by conducting building recognition experiments on the expanded Sheffield building dataset [9]. In Section 3.3, we selected images with partial occlusion or non-uniform illumination from Sheffield building dataset and formed a subset of Sheffield building database. Building recognition experiments were carried out on this subset.

Figure 5 are sample images from the Sheffield building dataset. As can be seen from Figure 5, building pictures are taken from different viewpoints or under different lighting conditions. The buildings in Figure 5(a) have different scaling due to different shooting distances. The Sheffield building dataset contains a total of 3192 images from 40 different buildings, ranging in number from 100 to 400. The size of images in Sheffield building dataset is 120 × 160. We expanded the Sheffield building dataset to include 8000 images by flipping all the images in the building dataset horizontally and rotating some of them $5° - 10°$. Denote the dataset containing 8000 images as D8000. We selected 1600 images with occlusion and non-uniform illumination from D8000, and formed a partial occlusion and non-uniform illumination image dataset. In following experiments, $train_x$

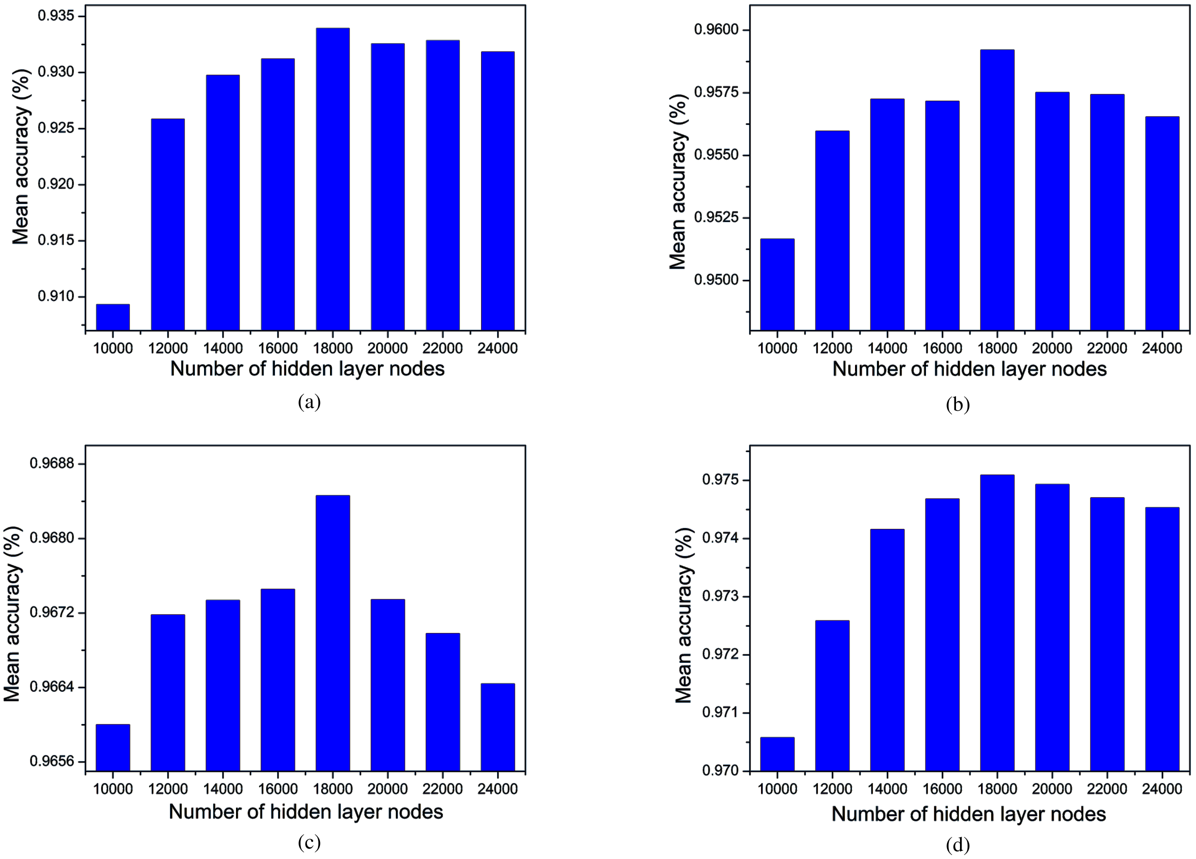


(a)

(b)

(c)

FIGURE 5. **(a)-(c) are sample images from Sheffield buildings database.**



(a)

(b)

(c)

(d)

FIGURE 6. **The mean recognition accuracy of each number of hidden layer nodes.**

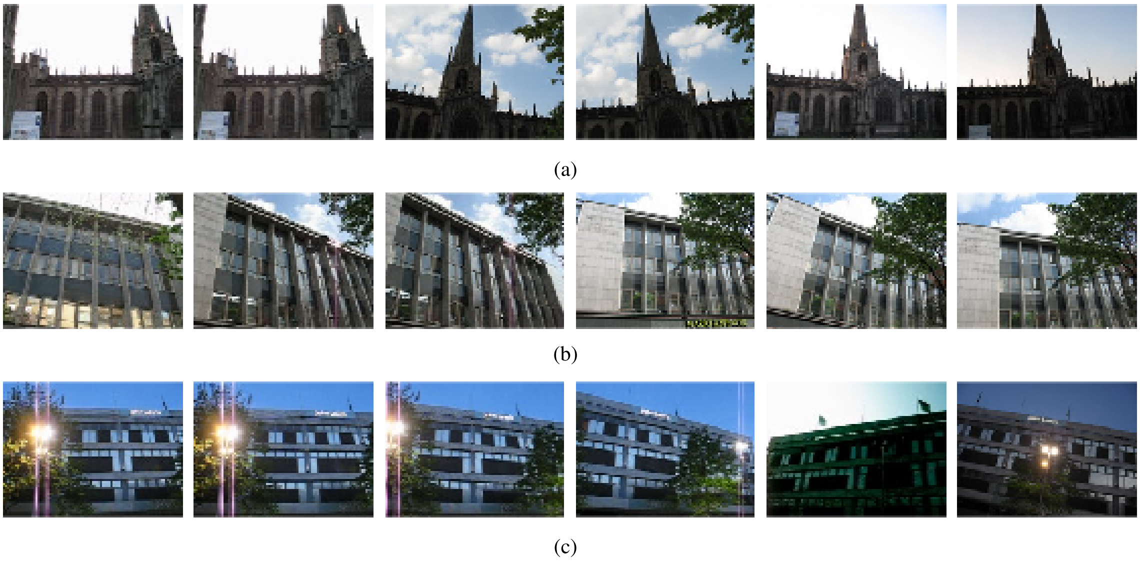
represents $x\%$ of all images are randomly selected for training and other $(1-x)\%$ images for testing.

A. EXPERIMENTS FOR THE NUMBER OF ELM'S HIDDEN NODES

The purpose of this experiment is to select the appropriate number of ELM's hidden layer nodes. Parameter selection experiments were conducted on following groups: $train_{10}$, $train_{15}$, $train_{20}$ and $train_{25}$, and experiments were repeated 20 times in each group. The average of 20 experimental results were taken as the final result of each group. Figure 6 shows the relationship between mean recognition rate and the number of ELM's hidden layer nodes. In Figure 6 (a)-(d), the four horizontal axes are the number of nodes in

TABLE 1. The mean recognition accuracy on the D8000 image dataset.

	$train_{10}$	$train_{15}$	$train_{20}$	$train_{25}$	$train_{30}$	$train_{35}$	$train_{40}$
ours	93.39	95.92	96.73	97.50	97.82	98.25	98.34
Gist+PCA	78.12	84.55	88.16	90.23	91.76	92.89	93.44
Gist+LPP	75.33	82.66	87.00	89.49	90.92	92.35	93.26
SM-gist+PCA	78.85	85.10	88.39	90.60	92.02	92.97	93.93
SM-gist+LPP	76.34	83.41	87.15	89.84	91.34	92.49	93.60



(a)

(b)

(c)

FIGURE 7. (a)-(c) are sample images from the partial occlusion and non-uniform illumination image dataset.

TABLE 2. The mean recognition accuracy on partial occlusion and non-uniform illumination image dataset.

	$train_{10}$	$train_{15}$	$train_{20}$	$train_{25}$	$train_{30}$	$train_{35}$	$train_{40}$
ours	82.31	85.61	87.70	88.74	89.63	90.36	90.90
Gist+PCA	58.18	64.64	69.43	71.74	73.69	75.28	76.40
Gist+LPP	54.63	62.06	67.38	70.54	72.94	74.58	75.63
SM-gist+PCA	59.12	64.77	69.39	72.42	74.57	76.02	77.28
SM-gist+LPP	56.26	62.80	67.46	71.26	73.48	75.16	76.60

ELM's hidden layer which range are [1000-2400]. From Figure 6 (a)-(d), it can be seen that when the number of hidden layer nodes is in [1400-2200], the mean recognition rate corresponding to each number is almost the same, but when the number of hidden layer nodes is 1800, the mean recognition rate is the highest. Therefore, the number of ELM's hidden layer nodes is set to 1800 in the following experiments.

B. BUILDING RECOGNITION ON THE D8000 IMAGE DATASET

In this experiment, we compared our method with methods in [9] and [14]. The Gist+PCA and the Gist+LPP in Table 1 are the methods in [9]. The two methods using the same gist feature extraction algorithm but different dimension reduction algorithms. The SM-gist+PCA and the SM-gist+LPP in Table 1 are the methods in [14]. In order to be fair, all methods reduce building features to 100-dimensional. From Table 1 we can see that the performance of our building recognition method is superior to all the contrast methods.

C. BUILDING RECOGNITION ON PARTIAL OCCLUSION AND NON-UNIFORM ILLUMINATION IMAGE DATASET

Some samples of partially occlusion and non-uniform illumination image dataset are shown in Figure 7. All the images in this image set may have partial occlusion, non-uniform illumination and building tilt as shown in Figure 7.

Due to the interference of partial occlusion or non-uniform illumination, the mean recognition rate of each method in Table 2 is much lower than that of the corresponding method in Table 1. In Table 2, the average recognition rate of our method is much higher than that of other methods. For example, in $train_{10}$ group, the mean recognition rate of our method is more than 20% higher than that of other methods. This is because MNSHOG and CA respectively reflect the global spatial texture and color distribution of a building image, so they are insensitive to local interference.

IV. CONCLUSION

In this paper, we presented a novel building recognition method based on spatial texture and color features. Firstly, we proposed multi-scale neighborhood sensitive histograms of oriented gradient (MNSHOG) which can well describe the spatial distribution information of image texture. The proposed MNSHOG combines CA to describe the spatial distribution of texture and color in building images. After that, a four layer autoencoder is used to elevate the dimension of the joint feature and then reduce its dimension. Finally, the ELM obtains recognition results based on these reduced dimension feature vectors. The experimental results are much higher than of the existing methods, which shows the effectiveness of our method.

REFERENCES

[1] N. J. C. Groeneweg, B. D. Groot, A. H. R. Halma, B. R. Quiroga, M. Tromp, and F. C. A. Groen, "A fast offline building recognition application on a mobile telephone," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.*, Sep. 2006, pp. 1122–1132.

[2] H. Ali, G. Paar, and L. Paletta, "Semantic indexing for visual recognition of buildings," in *Proc. 5th Int. Symp. Mobile Mapping Technol.*, 2007, pp. 28–31.

[3] Q. Iqbal and J. K. Aggarwall, "Applying perceptual grouping to content-based image retrieval: Building images," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 42–48.

[4] Y. Zhang, M. A. Nascimento, and O. R. Zaiane, "Building image mosaics: An application of content-based image retrieval," in *Proc. Int. Conf. Multimedia Expo*, Jul. 2003, p. III-317.

[5] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, P. Jensfelt, and H. I. Christensen, "Towards robust place recognition for robot localization," in *Proc. IEEE Int. Conf. Robot. Automat.*, Pasadena, CA, USA, May 2008, pp. 530–537.

[6] W. Zhang and J. Košecká, "Hierarchical building recognition," *Image Vis. Comput.*, vol. 25, no. 5, pp. 704–716, 2007.

[7] Y. Li and L. G. Shapiro, "Consistent line clusters for building recognition in CBIR," in *Proc. Object Recognit. Supported Interact. Service Robots*, vol. 3, Aug. 2002, pp. 952–956.

[8] R. Hutchings and W. Mayol-Cuevas, "Building recognition for mobile devices: Incorporating positional information with visual features," Dept. Comput. Sci., Univ. Bristol, Bristol, U.K, Tech. Rep. CSTR-06-017, 2005.

[9] J. Li and N. M. Allinson, "Subspace learning-based dimensionality reduction in building recognition," *Neurocomput.*, vol. 73, nos. 1–3, pp. 324–330, 2009.

[10] I. T. Jolliffe, *Principal Component Analysis* (Springer Series in Statistics). Berlin, Germany: Springer, 2002.

[11] X. He, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, vol. 45, no. 1, pp. 186–197.

[12] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[13] C. Zhao, C. Liu, and Z. Lai, "Multi-scale gist feature manifold for building recognition," *Neurocomputing*, vol. 74, no. 17, pp. 2929–2940, 2011.

[14] B. Li, C. Kaili, and Y. Zhezhou, "Histogram of oriented gradient based gist feature for building recognition," in *Proc. Comput. Intell. Neurosci.*, Oct. 2016, Art. no. 6749325.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[16] B. Li, and G. Huo, "Face recognition using locality sensitive histograms of oriented gradients," *Optik*, vol. 127, no. 6, pp. 3489–3494, 2015.

[17] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 762–768.

[18] C.-Y. Liou, J.-C. Huang, and W.-C. Yang, "Modeling word perception using the Elman network," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3150–3157, 2008.

[19] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, Sep. 2014.

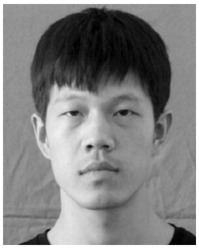
[20] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, pp. 985–990, Jul. 2004.

[21] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[22] X. Yang, and P. Xue, "Behaviors of transform domain backpropagation (BP) algorithm," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Nov. 1991, pp. 349–354.

[23] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

BIN LI was born in Changchun, Jilin, China, in 1982. He received the M.S. and Ph.D. degrees from the School of Computer Science and Technology, Jilin University, China, in 2011 and 2015, respectively. He is currently an Associate Professor with the School of Computer Science, Northeast Electric Power University. His research interests include image processing, computer vision, and pattern recognition.



FUQIANG SUN received the bachelor's degree from Yantai Nanshan University, in 2017. He is currently pursuing the degree with the School of Computer Science, Northeast Electric Power University. His research interests include computer vision and image processing.

YONGHAN ZHANG received the bachelor's degree from the Faculty of Software Engineering, Harbin University of Science and Technology, in 2017. He is currently pursuing the degree with the School of Computer Science, Northeast Electric Power University. His research interests include computer vision, image processing, and deep learning.