Implement Probabilistic Generative Classifier and Edge Histogram Descriptor to Class Different Type of Bricks

Hao-Yu Liao, *LOL*, Yixi Shen, *LOL*, Angelina Liu, *LOL*, Jingxi Weng, *LOL*

Abstract—In this work, an algorithm that can classify five types of patterns, including four brick patterns such as Flemish Stretch, English Bond, Stretcher Bond, and Other Brick Patterns, and other non-brick objects, is trained based on a dataset with over 11,000 images. The Edge Histogram Descriptor (EHD) is selected for extracting the features for each image of brick patterns, and the Probabilistic Generative Classifier (PGC) is applied to identify images based on the extracted features. The image retrieval methods applied in this work are based only on shape and texture features and do not collect color information. The filter action is implemented first to eliminate images with noise information, such as windows, doors, grassland, and other non-brick information. This action can improve the performance of the training model. The filtered images are then transformed from images with RGB channels into images with only one channel. Then the Edge Histogram Descriptor is implemented to gather the shape information, including Vertical Edge, Horizontal Edge, Diagonal 45 degrees, Diagonal 135 degrees, and Non-Orientation Type Edge of each image.

Index Terms— Probabilistic Generative Classifier, Edge Histogram Descriptor, Pixel classification, Edge and feature detection

1 Introduction

Digital images are a convenient media for storing, describing, and sharing information at low cost. Thus, digital images are now playing an important role in depicting and storing pictorial information. Large image databases with various kinds of image are being built and used in recent years. These databases usually consist of tens of thousands of images, which are difficult for a user to browse and organize through the entire database. Methods that can efficiently identify and classify photos with different features are now necessary [1]. In this work, the Edge Histogram Descriptor (EHD) and Probabilistic Generative Classifier (PGC) would be applied to identify images with different type of bricks from a database containing over 11,000 images.

Histogram is a widely used characteristic to describe features of an image. The translation or rotation of the images would not change the features represented by histogram. Furthermore, normalizing the histogram makes the scale invariant [2]. Considering the properties above, the histogram is a very useful tool for indexing and retrieving images [3], [4]. Edge in the image represents the content of the image; human eyes are known to be sensitive to edge features. Thus, the edge histogram descriptor is one of the most widely used descriptors. Agarwal et. al. (2013) applied EHD for image retrieval based on shape and texture feature only [5]. They analyze and extract the actual contents of the image rather than the stored metadata, like tags associated with the images.

After data selection, pre-processing, and feature extraction by EHD in this work, it is essential to be able to classify figures with different types of bricks into corresponding groups based of the extracted features. In this work, probabilistic generative classifiers (PGC) would be implemented to identify images. This classifier is based on

probabilistic mixture models [6]–[8], which can be used in many fields like data mining [9].

2 Implementation

In this study, the Edge Histogram Descriptor (EHD) will be used for extracting the features for each image, and the Probabilistic Generative Classifier (PGC) will be used to identify images. The EHD will find five features, including vertical edge, horizontal edge, diagonal 45 degrees, diagonal 135 degrees, and non-orientation type edge for each image. The details of the operator mask of EHD are shown in Table 1. Each mask is a 2 by 2 matrix, and each number in the matrix corresponds to 1 pix in the image. When extracting features from the images, each image will be transformed from three channels RGB to one channel grey image. For example, if the size of the image is 3 by 3, the operator will extract the image for four rounds. In each round, five operators will be computed. The largest value of edge type will be dominant in the round. The final result of extracting features is the cumulative number of each dominant type. The number of rounds can be decided as expressed:

$$r = (\text{image row size - 1}) \times (\text{image column size - 1}) \tag{1}$$

where r is the number of rounds for an image. The value of the five edge types will be computed in each round. The equation of computing the value of each edge type is indicated as:

$$\mathrm{OV} = sub_{1,1} \times om_{1,1} + sub_{1,2} \times om_{1,2} + sub_{2,1} \times om_{2,1} + sub_{2,2} \times om_{2,2} \tag{2}$$

where OV is the value of edge type, $sub_{1,1}$ is the pix values of image at row 1 and column 1 at subblock of image, and $om_{1,1}$ is the number of operator masks at row 1 and column 1. Each round there are five types of OV based on the five edge types; only one edge type will be dominant as expressed by Equation (3) to Equation (7):

$$V_i = \begin{cases} 1\ if\ the\ largest\ value\ is\ vetical\ edge \\ 0\ if\ the\ largest\ value\ is\ the\ other\ edge \end{cases} \tag{3}$$

$$H_i = \begin{cases} 1\ if\ the\ largest\ value\ is\ horizontal\ edge \\ 0\ if\ the\ largest\ value\ is\ the\ other\ edge \end{cases} \tag{4}$$

$$D45_i = \begin{cases} 1\ if\ the\ largest\ value\ is\ diagonal\ 45\ degree \\ 0\ if\ the\ largest\ value\ is\ the\ other\ edge \end{cases} \tag{5}$$

$$D135_i = \begin{cases} 1\ if\ the\ largest\ value\ is\ diagonal\ 135\ degree \\ 0\ if\ the\ largest\ value\ is\ the\ other\ edge \end{cases} \tag{6}$$

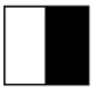
$$NOT_i = \begin{cases} 1\ if\ the\ largest\ value\ is\ non-orientation\ edge \\ 0\ if\ the\ largest\ value\ is\ the\ other\ edge \end{cases} \tag{7}$$

V_i is the value of vertical edge in round i, if the vertical edge is the largest value than other edge types, the value will be 1 which means the feature dominates in round i. The H_i is the value of horizontal edge in round i, $D45_i$ is the value of diagonal 45 degrees in round i, $D135_i$ is the value of diagonal 135 degrees in round i, and NOT_i is the value of non − orientation edge in round i. After computing dominated values at each round, the features of each image are expressed as:

$$(FV, FH, FD45, FD135, FNOT) = \left(\sum_{i=1}^{r} V_i, \sum_{i=1}^{r} H_i, \sum_{i=1}^{r} D45_i, \sum_{i=1}^{r} D135_i, \sum_{i=1}^{r} NOT_i\right) \tag{8}$$

TABLE 1

THE OPERATOR MASK OF EHD

Edge Type	Visual Representation	Operator Mask
Vertical Edge		$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$
Horizontal Edge		$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$
Diagonal 45 degrees		$\begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix}$
Diagonal 135 degrees		$\begin{bmatrix} 0 & \sqrt{2} \\ -\sqrt{2} & 0 \end{bmatrix}$
Non-Orientation Type Edge		$\begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$

where FV, FH, FD45, FD135, and FNOT are the feature values of the vertical edge, horizontal edge, diagonal 45 degrees, diagonal 135 degrees , and non − orientation edge for an image, respectively. After extracting the features, the features will be inputted into the PGC model for training and testing. In PGC model, the Gaussian distributions will be used as shown as:

$$P(\mathrm{x}|C_k) = \frac{1}{(2\pi)^{d/2}|\tau_k|^{1/2}} exp\left\{-\frac{1}{2}(\mathrm{x}-u_k)^T\tau_k^{-1}(\mathrm{x}-u_k)\right\} \tag{9}$$

where $P(\mathrm{x}|C_k)$ is probability in Gaussian distributions for class C_k with feature data x, τ_k is the covariance matrix, and u_k is the mean of class C_k. u_k and τ_k will be computed by maximum likelihood estimate as indicated as:

$$u_{k,MLE} = \frac{1}{N_K}\sum_{n\in C_k} \mathrm{x}_n \tag{10}$$

$$\tau_{k,MLE} = \frac{1}{N_K}\sum_{n\in C_k} (\mathrm{X}-u_{k,MLE})(\mathrm{X}-u_{k,MLE})^T \tag{11}$$

where $\mathbf{x}_n$ is the vector of each image with 5 by 1 based on five features with $(FV, FH, FD45, FD135, FNOT)$, and N_K is the number of training data which belongs to class k. Also, the probability of each class is expressed as:

$$P(C_k) = \frac{N_K}{N} \tag{12}$$

After computing the above parameters, the posterior probability of class k is shown as:

$$P(C_k|\mathrm{x}) = \frac{P(\mathrm{x}|C_k)P(C_k)}{\sum_{k=0}^{4} P(\mathrm{x}|C_k)P(C_k)} \tag{13}$$

The largest $P(C_k|\mathrm{x})$ means the image belongs to class k. Table 2 shows the nomenclature of this study.

TABLE 2

THE NOMENCLATURE OF THIS STUDY

Symbol	Content
r	The number of rounds for an image
OV	The value of edge type
$sub_{1,1}$	The pix values of image at row 1 and column 1 at subblock of image
$om_{1,1}$	The number of operator mask at row 1 and column 1
V_i	The value of vertical edge in round i
H_i	The value of horizontal edge in round i
$D45_i$	The value of diagonal 45 degrees in round i
$D135_i$	The value of diagonal 135 degrees in round i
NOT_i	The value of non − orientation edge in round i
FV	The feature values of vertical edge for an image
FH	The feature values of horizontal edge for an image
$FD45$	The feature values of diagonal 45 degrees for an image
$FD135$	The feature values of diagonal 135 degrees for an image
$FNOT$	The feature values of non − orientation edge for an image
$\mathbf{x}$	The features of an image
$\mathbf{x}_n$	The features of an image that belong to C_k
$\boldsymbol{\tau_k}$	The covariance matrix
$\boldsymbol{u_k}$	The mean vector
$P(\mathbf{x}|C_k)$	The probability of Gaussian distributions for class C_k with feature data $\mathbf{x}$
$P(C_k)$	The probability of class k
$P(C_k|\mathbf{x})$	The posterior probability of class k

3 Experiments

In this study, we have 11,383 images, and the types of images include Class 0 - Other [Images that are NOT of brick patterns.], Class 1 - Flemish Stretch, Class 2 - English Bond, Class 3 - Stretcher Bond, and Class 4 - Other Brick Patterns [Images of brick patterns that are NOT one of the three above.]. Before training the model, the filter action will be implemented first. Some brick pattern images contain irrelevant information such as windows, doors, and other unrelated patterns. These unwanted information will influence the performance of the training model's accuracy. After filtering the images, Class 0 has 1,671 images, Class 1 has 448 images, Class 2 has 818 images, Class 3 has 745 images, and Class 4 has 391 images. Table 3 shows the details of the dataset in this study. To maintain the proportion of the images, this study randomly selects 500 images in Class 0.

TABLE 3
THE DATASET OF THIS STUDY

Class	Name	Number
Class 0	Other [Images that are NOT of brick patterns.]	1,671 (randomly selecting 500 to model)
Class 1	Flemish Stretch	448
Class 2	English Bond	818
Class 3	Stretcher Bond	745
Class 4	Other Brick Patterns [Images of brick patterns that are NOT one of the three above.]	391

After filtering the dataset, images with correct pattern information will be inputted into the model. Fig. 1 shows the flowchart of the experimental design in this study, and Table 4 shows the parameter set for the grid search to find the best parameters. Five features from the pattern will be selected for training the model. The feature includes vertical edge, horizontal edge, diagonal with 45 degrees, diagonal with 135 degrees, and non-orientation type edge.

TABLE 4
THE PARAMETER SET OF THIS STUDY

Parameter	Values
K Fold	2~10
Feature	(Vertical Edge, Horizontal Edge, Diagonal 45 degree, Diagonal 135 degree, Non-Orientation Type Edge) with $C_5^5+C_4^5+C_3^5+C_2^5+C_1^5$
Total combinations	9* $C_5^5+C_4^5+C_3^5+C_2^5+C_1^5$ =279

In Fig. 1, 80% of data will be used for training and 20% of data for testing. The study uses cross-validation to find the best parameters. The K fold will be tried from 2 to 10. The model will try 279 different parameters, including different K folds and feature selections, to find the best model by grid search. Each result with a different parameter will be compared by accuracy, precision, and recall. The best training model has the best performance with the above indications. After selecting the best model, the test data will be put into the model to calculate its test performance. After computing the performance of the test results, this study will discuss its performance.

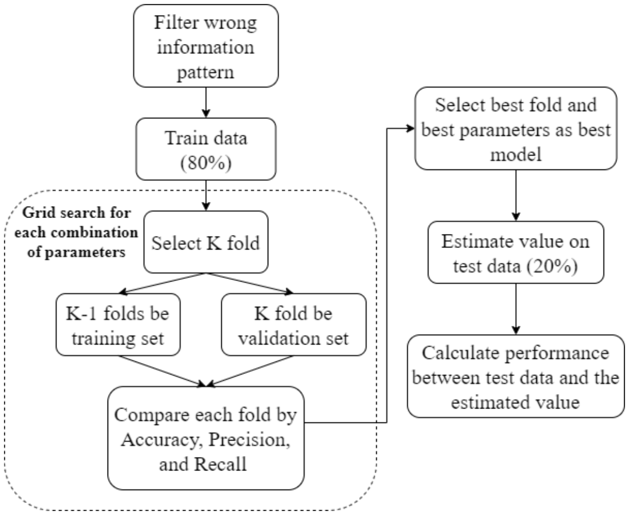


Fig. 1. Flowchart of experimental design

4 Conclusions

An algorithm that can classify the four types of brick patterns is trained based on a dataset with over 11,000 images. The Edge Histogram Descriptor is applied for extracting the features for each image of brick patterns, and the Probabilistic Generative Classifier is applied to identify images based on the extracted features. 80% of data will be used for training and 20% of data for testing. The cross-validation is also applied to find the best parameters and increase the accuracy.

A total of 279 different combination of parameters which includes different K folds and feature selections are tested by grid search in this work. The train data accuracies from this combination of parameters ranges from 0.378 to 0.662 and the test accuracies are between 0.382

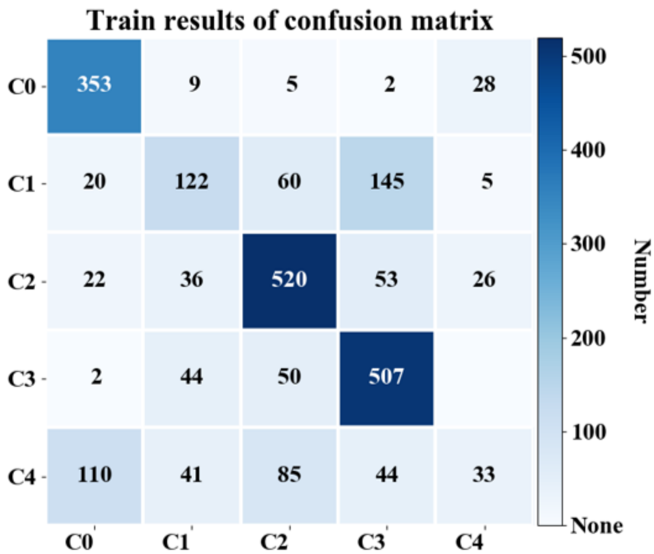


Fig. 2. Train results of confusion matrix

and 0.648. The set of features that have the highest accuracy is vertical edge, horizontal edge, and diagonal 135 degrees with K fold is 3. The observation that these three features would provide the highest accuracy is reasonable, because the most significant difference between different brick patterns images are the fraction of vertical and horizontal edge. However, it is challenging for the EHD to extract features from the image consist of oblique view, like images of Class 0 in this work. As shown in Figure 2, out of all the training images, 28 Class 0 images were misclassified into Class 4, and 110 Class 4 images were misclassified into C0. This means that even in the training process, it is hard to differentiate between Class 4 and Class 0, as there were no standard criteria in defining Class 0. Therefore, more paramters are needed to improve the performance of the classifier. Possible methods include edge detection with more directions, corner detection, image shape, and image color.

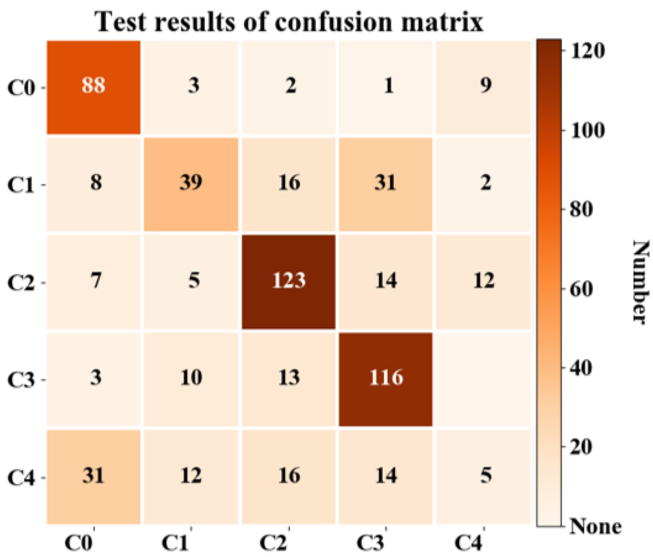


Fig. 3. Test results of confusion matrix

TABLE 5

TRAIN MODEL ACCURACY

Train model accuracy	0.662	
Train Model	Precision	Recall
Class 0	0. 699	0.892
Class 1	0.472	0.328
Class 2	0.728	0.793
Class 3	0.671	0.847
Class 4	0.403	0.119

TABLE 6

TEST MODEL ACCURACY

Test model accuracy	0.648	
Test Model	Precision	Recall
Class 0	0.676	0.866
Class 1	0.467	0.324
Class 2	0.720	0.787
Class 3	0.665	0.836
Class 4	0.286	0.083

The confusion matrix of the test results is shown in Fig. 3. The confusion matrix of test results has similar features as train results of confusion matrix in Fig. 2. The algorithm shows relatively lower accuracy when classifying Class 4 from Class 0. The train and test model accuracy in Table 5 and Table 6 prove this argument.

Overall, the algorithm trained in this work is effective in identifyinh different patterns of bricks in the given dataset. The accuracy of the system is relatively high; there was a training accuracy of approximately 0.66 and a testing accuracy at about 0.648. The comparable accuracies between the two models indicate that there was no overfitting during the training. The feature selection model (EHD) used and Probabilistic Generative Classifier are popular tools used in the field; it was simple to implement and possessed a sufficient classification ability. With the training and testing combined, the whole process took less than three minutes.

REFERENCE

[1] S. Liapis and G. Tziritas, “Color and texture image retrieval using chromaticity histograms and wavelet frames,” IEEE Trans. Multimed., vol. 6, no. 5, pp. 676–686, 2004, doi: 10.1109/TMM.2004.834858.

[2] C. S. Won, D. K. Park, and S. J. Park, “Efficient use of MPEG-7 edge histogram descriptor,” ETRI J., vol. 24, no. 1, pp. 23–30, 2002, doi: 10.4218/etrij.02.0102.0103.

[3] A. K. Jain and A. Vailaya, “Image retrieval using color and shape,” Pattern Recognit., vol. 29, no. 8, pp. 1233–1244, 1996, doi: 10.1016/0031-3203(95)00160-3.

[4] M. . Swain and D. . Ballard, “Color indexing,” iNT. j. Comput. Vis., vol. 7–1, 1991.

[5] S. Agarwal, A. K. Verma, and P. Singh, “Content based image retrieval using discrete wavelet transform and edge histogram descriptor,” Proc. 2013 Int. Conf. Inf. Syst. Comput. Networks, ISCON 2013, pp. 19–23, 2013, doi: 10.1109/ICISCON.2013.6524166.

[6] D. Fisch and B. Sick, “Training of radial basis function classifiers with resilient propagation and variational Bayesian inference,” Proc. Int. Jt. Conf. Neural Networks, vol. 1, pp. 838–847, 2009.

[7] D. Fisch, C. Gruhl, E. Kalkowski, B. Sick, and S. J. Ovaska, “Towards automation of knowledge understanding: An approach for probabilistic generative classifiers,” Inf. Sci. (Ny)., vol. 370–371, pp. 476–496, 2016, doi: 10.1016/j.ins.2016.08.016.

[8] C. M. Bishop, Pattern Recognition and machine learning. 2006.

[9] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “Knowledge discovery and data mining: towards a unifying framework,” Proc. 2nd Inter- Natl. Conf. Knowl. Discov. Data Min., pp. 82–88, 1996.