Cascading of K-Means and K-NN Methods for Texture Image Classification

B. Ari Kuncoro
Master of Information Technology
Bina Nusantara University
Jakarta, Indonesia
Email: b.kuncoro [at] binus.ac.id

Filo Limantara
Master of Information Technology
Bina Nusantara University
Jakarta, Indonesia
Email: filo.limantara [at] gmail.com

***Abstract*—One of the most important problems in pattern recognition is texture-based image classification. It is useful in recognition and interpretation, because as image, it consists of related and interrelated elements. In this work, to extract feature, Gray Level Co-Occurrence Matrix (GLCM) methods was applied. The cascading of K-Means clustering and K-Nearast Neighbor (K-NN) classification is proposed to classify images into three-class texture image. The objective of this work is to check whether the cascading methods of K-Means and K-NN can increase the accuracy compare with the original K-NN. The dataset has original labels and they were used to get the accuracy parameter for comparison between K-NN and 'cascade K-Means and KNN'.**

Keywords*—*Texture Images; K-Nearest Neighbor; K-Means; Cascade

I. Introduction

One of the well-known low-level image features is texture [1]. It is useful in recognition and interpretation, because as image, it consists of related and interrelated elements [2]. Several steps must be done to extract features from the texture images. Many researchers has reported various methods in how to extract them. They include statistical methods of Gray Level Co-occurrence Matrix (GLCM), energy filters and edgeness factor, Gabor filters, and Wavelet transform [3], [4]. There is no strict guideline to determine what features are most suitable for specific data, thus experimental methods such as empirical and heuristic methods are commonly used.

Common classification methods were proposed such as K-NN, Adaptive Neuro Fuzzy Inference System (ANFIS), Support Vector Machne (SVM), etc. Different from those pure-classifier, Karegowda, Jayaram, and Manjunath's [5] performed classification towards diabetic cascade the K-Means clustering and K-NN. The result shows that cascading K-Means clustering and K-NN classification give best accuracy compare with other report results. In this work, we perform similar technique to Karegowda et al., in which they used cascading of K-Means clustering and K-Nearest Neighbor.

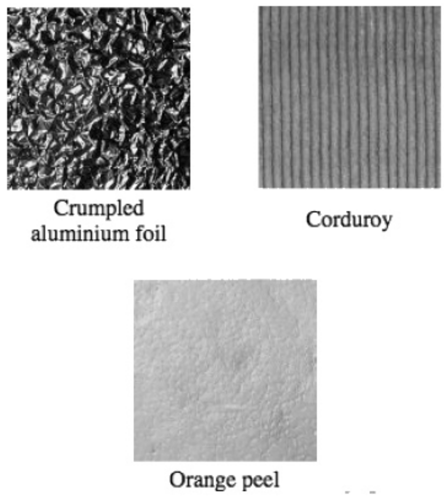


Fig. 1. Samples of Dataset

II. Dataset

The image dataset can be downloaded at http://www.nada.kth.se/cvap/databases/kth-tips/download.html [6]. Three-class texture image samples consist of alumunium foil, corduroy, and orange peel are illustrated in Fig. 1. Each class contains 80 texture images, thus there were 240 texture images collected.

III. Methodology

The methodology of this study is illustrated in Fig. 2. First, 240 gray-scale images were collected in one folder. Then the images were resized into 64×64 pixels. On each images, GLCM block was applied thus GLCM matrix was produced. From GLCM, energy and entropy were extracted as attribute 1 and attribute 2. The original labels of the data were removed first, thus the data with (240×2) size became the input of unsupervised learning K-Means clustering (in which the K equals to 3). The categorization result of K-Means clustering

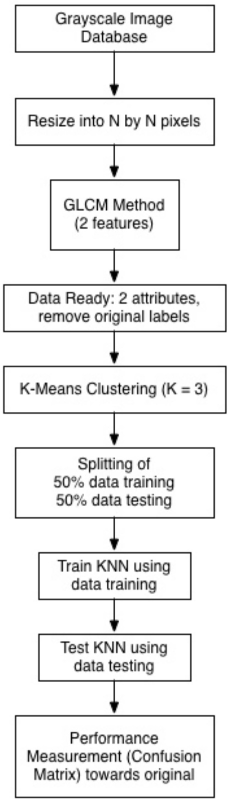
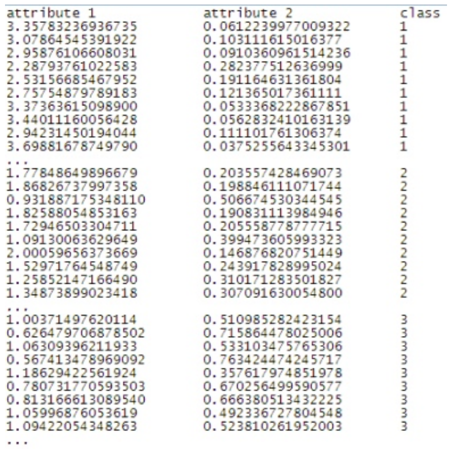


Fig. 2. Flowchart of Classification System



attribute 1	attribute 2	class
3.35783236936735	0.0612239977009322	1
3.07864545391922	0.103111615016377	1
2.95876106608031	0.0910360961514236	1
2.28793761022583	0.282377512636999	1
2.53156685467952	0.191164631361804	1
2.75754879789183	0.121365017361111	1
3.37363615098900	0.0533368222867851	1
3.44011160056428	0.0562832410163139	1
2.94231450194044	0.111101761306374	1
3.69881678749790	0.0375255643345301	1
...		
1.77848649896679	0.203557428469073	2
1.86826737997358	0.198846111071744	2
0.931887175348110	0.506674530344545	2
1.82588054853163	0.190831113984946	2
1.72946503304711	0.205558778777715	2
1.09130063629649	0.399473605993323	2
2.00059656373669	0.146876820751449	2
1.52971764548749	0.243917828995024	2
1.25852147166490	0.310171283501827	2
1.34873899023418	0.307091630054800	2
...		
1.00371497620114	0.510985282423154	3
0.626479706878502	0.715864478025006	3
1.06309396211933	0.533103475765306	3
0.567413478969092	0.763424474245717	3
1.18629422561924	0.357617974851978	3
0.780731770593503	0.670256499590577	3
0.813166613089540	0.666380513432225	3
1.05996876053619	0.492336727804548	3
1.09422054348263	0.523810261952003	3
...		

Fig. 3. Data of attribute 1, attribute 2, and Class

is considered as labels for K-NN classifier training purpose. Data was split into 50% for training and 50% for testing using interleaving. The result of testing towards the trained model is then calculated using confusion matrix, in which the predicted data is compared with the true data of original label.

A. Gray Level Co-Occurrence Matrix (GCLM)

Gray Level Co-Occurrence Matrix (GLCM) has proved to be a popular statistical method of extracting textural feature from images. According to co-occurrence matrix, Haralick defines fourteen textural features measured from the probability matrix to extract the characteristics of texture statistics of remote sensing images [7]. Mohanaiah [4] performed extraction using these important features, Angular Second Moment (energy), Correlation, Entropy, and the Inverse Difference Moment. In this work, feature of Entropy and Energy from GLCM Matrix were used as attribute 3 and attribute 4 respectively.

1) Angular Second Moment (Energy): Angular Second Moment or Energy is the sum of squares of entries in the GLCM matrix. It is also known as homogeneity. The value of which energy high when image has very good homogeneity or when pixels are very similar [4].

$$Energy = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i,j)^2 \quad (1)$$

2) Entropy: Entropy shows the amount of information of the image that is needed for the image compression. Entropy measures the loss of information or message in a transmitted signal and also measures the image information.

$$Entropy = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} -P(i,j) \times log(P(i,j)) \quad (2)$$

In this paper, each class was represented by integer. For combination 1, integer 1, 2, and 3 represent aluminium foil, corduroy, and orange peel respectively. Since the feature extraction process produced four features, each of the data contains four attributes. As a result, the used data contains four inputs and one output. The snippet of the data is as follows.

B. K-Means Clustering

K-means algorithm is a clustering algorithm that is invented by LLoyd [8].The aim of the K-means algorithm is to divide M points in N dimensions into K clusters so that the within-cluster sum of squares is minimize [9]. K-means algorithm uses an iterative technique to minimize the sum of squares.

Given an initial set of k means m1$^{(1)}$,,mk$^{(1)}$ (see below), the algorithm consist of two phase:

Assignment step: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares

(WCSS). Since the sum of squares is the squared Euclidean distance, this is intuitively the ”nearest” mean.

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \ \forall j, 1 \leq j \leq k\} \tag{3}$$

where each $x_p$is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.
Update step: Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{4}$$

C. K-Nearest Neighbor

K-nearest-neighbor (kNN) classification is one of the most fundamental and simple classification algorithm. It should the first choices for a classification study if we have no prior knowledge about the data. It was first proposed by Fix and Hodges [10].

The k-nearest-neighbor classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. Let x_i be an input sample with p features (x_{i1},x_{i2},,x_{ip}) , n be the total number of input samples (i=1,2,,n) and p the total number of features (j=1,2,,p) . The Euclidean distance between sample x_i and x_l (l=1,2,,n) is defined as

$$d(\mathbf{x}_i, \mathbf{x}_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \cdots + (x_{ip} - x_{lp})^2} \tag{5}$$

After calculating the distance of training samples to input sample x_i, select k closest sample x_m, and we finally can base the assignment of a label upon x_i by the maximum predominance of a particular class in its neighborhood [11].

IV. Results and Discussion

To visualize the data based on generated attributes, 2-D plotting data graph is required. It is depicted in Fig. 4 that color of blue, green, and red represent aluminium foil, corduroy, and orange peel textures respectively. It is seen that aluminium foil is well clustered on the right bottom of the cartesian graph. While corduroy and orange peel seems hard to be separated.

Precision-recall result and accuracy of each methods are shown in Fig. 5, 6, and 7.

The accuracy, precision, recall, and f1-score are calculated according this the following formula [12]:

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

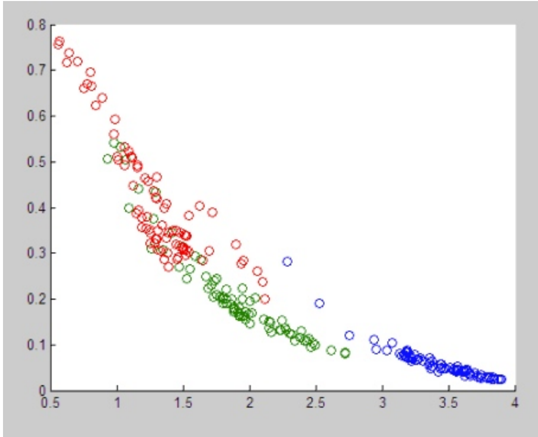


Fig. 4. Plot Input data in 2-D

```
KNN normal:

Classification report for classifier KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski',
           metric_params=None, n_neighbors=3, p=2, weights='uniform'):
             precision    recall  f1-score   support

        1.0       0.97      0.97      0.97        40
        2.0       0.80      0.80      0.80        40
        3.0       0.82      0.82      0.82        40

avg / total       0.87      0.87      0.87       120


Confusion matrix:
[[39  1  0]
 [ 1 32  7]
 [ 0  7 33]]
```

Fig. 5. Confusion matrix result of K-NN

```
K-Means normal:

Classification report for classifier K-Means:
             precision    recall  f1-score   support

        1.0       1.00      0.96      0.98        80
        2.0       0.84      0.76      0.80        80
        3.0       0.79      0.89      0.84        80

avg / total       0.87      0.87      0.87       240


Confusion matrix:
[[77  3  0]
 [ 0 61 19]
 [ 0  9 71]]
```

Fig. 6. Confusion matrix result of K-Means

$$\text{recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{8}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{9}$$

```
K-Means cascade KNN:

Classification report for classifier KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski',
        metric_params=None, n_neighbors=3, p=2, weights='uniform'):
          precision    recall  f1-score   support

      1.0     1.00      0.97     0.99       40
      2.0     0.84      0.78     0.81       40
      3.0     0.80      0.88     0.83       40

avg / total     0.88      0.88     0.88      120


Confusion matrix:
[[39  1  0]
 [ 0 31  9]
 [ 0  5 35]]
```

Fig. 7. Confusion matrix result of Cascade of K-Means and K-NN

TABLE I
ACCURACY OF K-NN VS CASCADE OF K-MEANS AND K-NN

No.	Methods	Accuracy
1	K-NN	87%
2	Cascade of K-Means and K-NN	88%

where TP are number of true positive, TN are number of true negative, FP are number of false positive, and FN are number of false negative.

The performance result of K-NN and cascade of K-Means and K-NN can be seen in Table I. It seen that the accuracy K-NN is 87%, while the accuracy of K-Means and K-NN is 88% or there is 1% increase of accuracy.

V. CONCLUSION

Entropy and energy of GLCM are the features that were extracted for texture image classification in this work. Using splitting data, in which 50% used for training and 50% for testing, the system for classification was built. The system result shows taht accuracy of pure K-NN is 87%. While the accuracy of cascade K-Means and K-NN is 88%, meaning it increases the accuracy at 1%.

ACKNOWLEDGMENT

The authors would like to thank Dr. Widodo Budiharto of Bina Nusantara University, Jakarta, Indonesia for his guidance.

REFERENCES

[1] D. Avci, M. Poyraz, and M. Leblebicioglu, "An expert system based on discrete wavelet transform - anfis for acquisition and recognition of invariant features from texture images," in *Signal Processing and Communications Applications Conference (SIU), 2015 23th*, May 2015, pp. 1070–1073.

[2] P. H. Chandankhede, "Soft Computing Based Texture Classification with MATLAB Tool," *International Journal of Soft Computing and Engineering*, no. 2, pp. 475–480, 2012.

[3] L. A. Ruiz, A. Fdez-Sarría, and J. A. Recio, "Texture Feature Extraction for Classification of Remote Sensing Data using Wavelet Decomposition: A Comparative Study," in *International Archives of Photogrammetry and Remote Sensing*, vol. XXXV, 2004, pp. 1682–1750.

[4] P. Mohanaiah, P. Sathyanarayana, and L. Gurukumar, "Image Texture Feature Extraction Using GLCM Approach," *International Journal of Scientific & Research Publication*, vol. 3, no. 5, pp. 1–5, 2013.

[5] A. G. Karegowda, M. A. Jayaram, and A. S. Manjunath, "Cascading K-means Clustering and K-Nearest Neighbor Classifier for Categorization of Diabetic Patients," *International Journal of Engineering and Advanced Technology*, vol. 1, no. 3, pp. 147–151, 2012.

[6] B. C. Mario Fritz, Eric Hayman and J.-O. Eklundh, "The KTH-TIPS and KTH-TIPS2 Image Databases," 2004. [Online]. Available: http://www.nada.kth.se/cvap/databases/kth-tips/documentation.html

[7] P. Costianes and J. Plock, "Gray-level co-occurrence matrices as features in edge enhanced images," in *Applied Imagery Pattern Recognition Workshop (AIPR), 2010 IEEE 39th*, Oct 2010, pp. 1–6.

[8] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[9] J. a. Hartigan and M. a. Wong, "A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society*, vol. 28, no. 1, pp. 100–108, 1979.

[10] P. Evelyn Fix and P. J.L. Hodges, Jr., "Discriminatory analysis, non-parametric discrimination: Consistency properties," no. May, 1951.

[11] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[12] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*, 2008.