E3S Web of Conferences **475**, 02012 (2024) https://doi.org/10.1051/e3sconf/202447502012
InCASST 2023

Batik classification using KNN algorithm and GLCM features extraction

David Wijaya[1], and *Anastasia Rita* Widiarti[1*]

[1]Informatics Department, Sanata Dharma University, Yogyakarta, Indonesia

Abstract. Batik is one of the Indonesian cultures that has been recognized by UNESCO as an intellectual right of Indonesia. The popularity of batik internationally raises concerns about the Indonesian people's understanding of batik if Indonesian people only refer to all types of batik just as 'batik'. By utilizing K-Nearest Neighbour (KNN) algorithm which is a simple classification algorithm, then a system can be created that can classify batik types. The first step of KNN is training, which stores each training pattern. The second step is classification, whenever classifying a pattern, KNN examine all training patterns to determine the K closest patterns using certain calculations such as Euclidean Distance and Manhattan Distance. Before classification, a characteristic that represents a pattern is needed. Gray-Level Co-Occurrence Matrix (GLCM) is an algorithm that has proven to be very powerful as a feature descriptor in representing the texture characteristic of an image. This research experiments with the value of K in KNN = 1, 3, 5, and 7 with the distance calculation using Euclidean and Manhattan. The GLCM characteristic used are Entropy, Energy, Contrast, Homogeneity, Dissimilarity, Correlation, ASM, and the average of each characteristic. From the research that has been done, the system created obtained the highest accuracy of 75% with the combination of parameters; pixel distance = 7, K value = 1, 1st fold as test data and 2nd and 3rd fold as training data, and by using StandardScaler. But despite getting decent accuracy, there is still a need for further research to improve the accuracy of batik classification.

1 Introduction

Indonesian batik was designated as Masterpieces of the Oral and Intangible Heritage of Humanity, recognized by UNESCO as an intellectual right of the Indonesian people on October 2, 2009 [1]. The 6th President of Indonesia, H. Susilo Bambang Yudhoyono, M.A. issued Presidential Decree No. 33 of 2009 on the establishment of National Batik Day and signed on November 17, 2009 due to the inauguration of Indonesian batik by UNESCO in Abu Dhabi. Then on October 1, 2019, it was stipulated to wear batik clothes in the framework of National Batik Day on October 2, 2019 [2]. According to KBBI, batik is an illustrated cloth that is specially made by writing or applying wax to the cloth, then processing it in a certain way. The batik process produces diverse and high-value patterns whose designs are

* Corresponding author: rita_widiarti@usd.ac.id

inspired by nature or mythology, resulting in distinctive geometric patterns, various batik motifs with different names [3]. The popularity of batik in the international world raises concerns about the understanding of the Indonesian people about batik, the knowledge of Indonesians about batik should be more than outsiders [4], especially since there are so many advances in fashion today that batik is considered very old. In addition, due to the COVID-19 outbreak, everything is done by utilizing technology to reduce direct contact. Although now the COVID-19 outbreak is not as bad as it used to be, the habit of using such technology still exists. Therefore, the idea arose to create a system that can classify various types of batik easily. KNN is one of the simplest machine learning algorithms of many [5] that classifies an object based on the closest training data object in a feature space [6]. To measure the distance in finding the nearest neighbor, there are several approaches that can be used including Euclidean Distance and Manhattan Distance. In previous research, the batik classification process with the edge detection method using K-Nearest Neighbor concluded that Manhattan Distance obtained higher accuracy results, which amounted to 83.99% compared to using Euclidean Distance which obtained the highest accuracy of 80.78% [7]. However, another study that classified textual data using K-Nearest Neighbor concluded that Euclidean Distance produces better accuracy than Manhattan Distance with an odd number of nearest neighbors starting from 3 to 19 [8]. Therefore, it is necessary to find the best closest distance calculation method for batik classification using Euclidean Distance and Manhattan Distance. With this system, all people, young and old, are expected to be interested and easily recognize the types of Indonesian batik [9-11].

2 Research method

Research that utilizes KNN classification and GLCM feature extraction that has been done before can be seen in Table 1.

Table 1. Accuracy results of previous research

Data	K Value and Features	Highest Accuracy
Classification of Sasirangan fabrics and non-Sasirangan fabrics [12]	K = 1, ASM, contrast, IDM, entropy, and correlation	63%
Classification of Kudus batik and not Kudus batik [13]	K = 1, Energy, contrast, correlation, homogeneity, and entropy	97%
Classification of Lampung batik and not Lampung batik [14]	K = 7, ASM, contrast, homogeneity, and correlation	97,96%
Classification of Javanese, Kalimantan, Papuan, Sulawesi, and Sumatran batik [15]	K = 1, Discrete Cosine Transform on GLCM (ASM, contrast, IDM, entropy, and correlation)	77,90%
Image classification of Aseman, Kawung Baganan, Latohan, Sidomukti, and Naga [16]	K = 3, Energy, contrast, homogeneity, and correlation	77,33%
Classification of grains (rice, corn, green beans, peanuts, and soybeans) [17]	K = 1 and 3, Energy, entropy, contrast, homogeneity, correlation, IDM, sum mean, sum entropy, sum variance, difference variance, and difference entropy	100%
Classification of orchid and non-orchid flower types [18]	K = 1, Energy, entropy, contrast, and homogeneity	80%
Classification of 6 types of Cepuk cloth [19]	K = 1, Contrast, energy, entropy, homogeneity, dissimilarity, ASM, and IDM	100%

Batik Classification of Gajah Oling, Kopi Pecah, and neither of them [20]	K = 9, Contrast, dissimilarity, homogeneity, energy, and correlation	87,5%

2.1 Image preprocessing

Image preprocessing is a step taken before the image is used so that the image can be analyzed better and can speed up computation [21]. There are various processes in preprocessing, including grayscaling and normalization.

2.1.1 Grayscaling

Grayscaling is the process of converting an image into an image that only has black and white colors. Grayscaling is usually used to reduce computation time [22].

2.1.2 Normalization

Data normalization or scaling is used to convert values into a scale of values within a specific range [23].

2.1.3 StandardScaler

StandardScaler is used to resize the distribution of values so that the mean of the values is 0 and the standard deviation is 1 [24].

2.2 Classification

Classification is a method to classify an object based on the characteristics possessed by the object of classification [25]. According to KBBI, classification is a systematic arrangement in groups or classes according to established rules or standards. In the case of this image classification, we can relate that the rules or standards set come from the characteristic parameters obtained from feature extraction.

2.2.1 K-Nearest Neighbor

KNN consists of 2 steps, the first step is training which stores each training pattern and the second step is classification. Every time it classifies a pattern, KNN must examine all training patterns to determine the K closest patterns [26]. To find the closest distance between the object to be classified and all other objects, several methods can be used, including:

1. Euclidean Distance
   Euclidean Distance is one of the methods to measure the distance of two points in two, three or more dimensions [27].
   $$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$
2. Manhattan Distance
   Manhattan Distance is a method for measuring distance by calculating the absolute difference between the coordinates of a pair of objects [27].
   $$d(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{2}$$

2.3 Feature extraction

Characteristic extraction is a method to obtain the characteristics of an object to be used as a reference to distinguish it from other objects [28].

2.3.1 Gray level co-occurrence matrix

GLCM in the first order uses statistical calculations based on the original image pixel values and does not pay attention to neighboring pixels, while the second order is a feature extraction method that takes texture into account [29]. There are 14 features that can be extracted with the GLCM method, but only 4 of them are the main features, namely contrast, energy, entropy, and homogeneity [30]. Some features or characteristics resulting from feature extraction with the GLCM method are:

- Energy. Energy is a measure of the degree of uniformity of the texture [30].
- Contrast. Contrast is a representation of the size of the gray level variation in the co-occurrence matrix [30].
- Homogeneity. Homogeneity is a measure of the level of homogeneity or similarity of variation in the gray intensity of the image in the co-occurrence matrix [30].
- Entropy. Entropy is a measure of the degree of randomness of the gray degree of an image in the cooccurrence matrix [30].
- Dissimilarity. Dissimilarity is a measure of the dissimilarity of a texture in an image that will have a large value if it is random and a small value if it is uniform [31].
- Correlation. Correlation is a measure of the linearity of a number of pixel pairs [31].
- ASM. Angular Second Moment is the sum of squares of the energy values that measures the texture uniformity of the gray image, if the pixel values are similar to each other, it will be high and if the pixel values are different, the value will be high [31].

2.4 Result evaluation

Evaluation of results is used to see how well a system performs, in this case seeing how well the system is able to classify batik types accurately.

2.4.1 Confusion matrix

Confusion Matrix is a benchmark to measure how well machine learning classification performs [32] in the form of a table containing 4 different combinations of predicted and actual values, namely;

- True Positive (TN). If the prediction is positive and correct, for example, a woman is pregnant and predicted to be pregnant [32].
- False Positive (FP). If the prediction is positive and false, for example, a man is predicted to be pregnant [32].
- True Negative (TN). If predicted negative and correct, for example, a man is predicted not to be pregnant [32].
- False Negative (FN). If predicted negative and false, for example a woman is pregnant and predicted not pregnant [32].

The four combinations of predicted and actual values can be used to measure classification performance, including:

- Accuracy. Accuracy shows how accurate the classification is [33]. Accuracy can be calculated by the formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

- Precision. Precision shows what fraction of positive classes are truly positive [33]. Precision can be calculated by the formula:

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

- Recall. Recall or sensitivity indicates what fraction of all positive samples are correctly predicted [33]. Recall can be calculated by the formula:

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

2.4.2 K-Fold cross validation

Cross Validation is one of the methods to test the effectiveness of a machine learning model that divides data into 2 parts, namely test data and training data. K-Fold Cross Validation is a cross validation method that is often used because this method produces an unbiased model, because each data has the opportunity to become test data and training data [34]. In K-Fold Cross Validation, a value of K that is too large results in greater variance, causing overfit, while a value of K that is too small can trigger bias [35]. The first step in K-Fold Cross Validation is to randomize the dataset and then separate the dataset into K parts. Next, an iteration is performed where one part becomes the test data and the other part becomes the training data and then the results of the evaluation of this model are stored and continued to the next iteration where the test data moves to another part until all parts have had the opportunity to become test data. The final step is to summarizethe results of the K-Fold Cross Validation evaluation [36].

3 Results and analysis

Before testing, the GLCM feature data from 0° is divided into 3 folds which will be used to determine what is used as training data and test data. If fold 1 is used as test data, then training data is fold 2 and 3 plus the feature data from 45°, 90°, and 135°. If fold 2 is used as test data, then training data is fold 1 and 3 plus the feature data from 45°, 90°, and 135°. If fold 3 is used as test data, then training data is fold 1 and 2 plus the feature data from 45°, 90°, and 135°. Table 2, Table 3, Table 4, Table 5, Table 6, and Table 7 presents the test results using the various value variables mentioned above.

Table 2. Accuracy results with pixel distance 2 without StandardScaler

Testing Data	K Value	Accuracy	
		Euclidean	Manhattan
Fold 1	1	8%	11%
	3	8%	9%
	5	7%	6%
	7	8%	8%
Fold 2	1	11%	13%
	3	12%	12%
	5	10%	12%
	7	13%	12%
Fold 3	1	5%	7%
	3	8%	9%

	5	8%	8%
	7	7%	7%

Table 3. Accuracy results with pixel distance 2 with StandardScaler

Testing Data	K Value	Accuracy	
		Euclidean	Manhattan
Fold 1	1	51%	54%
	3	34%	38%
	5	32%	32%
	7	31%	53%
Fold 2	1	53%	38%
	3	38%	36%
	5	33%	33%
	7	30%	12%
Fold 3	1	49%	50%
	3	31%	33%
	5	28%	28%
	7	25%	28%

Table 4. Accuracy results with pixel distance 5 without StandardScaler

Testing Data	K Value	Accuracy	
		Euclidean	Manhattan
Fold 1	1	9%	12%
	3	10%	11%
	5	8%	9%
	7	10%	10%
Fold 2	1	10%	13%
	3	10%	8%
	5	9%	10%
	7	10%	10%
Fold 3	1	11%	12%
	3	10%	10%
	5	11%	13%
	7	12%	13%

Table 5. Accuracy results with pixel distance 5 with StandardScaler

Testing Data	K Value	Accuracy	
		Euclidean	Manhattan
Fold 1	1	71%	72%
	3	59%	60%
	5	50%	51%
	7	41%	43%
Fold 2	1	70%	70%
	3	62%	61%
	5	53%	52%
	7	47%	47%
Fold 3	1	70%	70%
	3	63%	61%
	5	54%	55%
	7	49%	49%

Table 6. Accuracy results with pixel distance 7 without StandardScaler

	K Value	Accuracy

Testing Data		**Euclidean**	**Manhattan**
Fold 1	1	9%	12%
	3	7%	9%
	5	8%	9%
	7	8%	9%
Fold 2	1	10%	12%
	3	9%	9%
	5	9%	9%
	7	11%	13%
Fold 3	1	7%	8%
	3	11%	11%
	5	11%	12%
	7	10%	13%

Table 7. Accuracy results with pixel distance 7 with StandardScaler

Testing Data	**K Value**	**Accuracy**	
		Euclidean	**Manhattan**
Fold 1	1	75%	75%
	3	65%	65%
	5	53%	56%
	7	44%	47%
Fold 2	1	74%	73%
	3	67%	68%
	5	60%	60%
	7	52%	50%
Fold 3	1	71%	71%
	3	62%	63%
	5	54%	52%
	7	44%	43%

From the results obtained, it can be concluded that the application of StandardScaler and pixel distance parameters has an effect in improving accuracy. The highest accuracy result is 75% obtained from Table 7, with pixel distance parameter = 7, K value = 1, with the 1st fold 0° degree feature as test data and the 2nd and 3rd fold 0° degree plus the feature data from 45°, 90°, and 135° degree features as training data. Furthermore, in an effort to improve the accuracy, the image is subjected to additional preprocessing by cropping the image and then further tested by applying StandardScaler. Table 8, Table 9, and Table 10 presents new result after applying cropping and using StandardScaler.

Table 8. Accuracy results using cropped image and pixel distance 2

Testing Data	**K Value**	**Accuracy**	
		Euclidean	**Manhattan**
Fold 1	1	44%	43%
	3	29%	30%
	5	26%	30%
	7	23%	25%
Fold 2	1	46%	46%
	3	28%	30%
	5	26%	25%
	7	22%	24%
Fold 3	1	45%	44%
	3	30%	27%
	5	25%	25%

	7	22%	22%

Table 9. Accuracy results using cropped image and pixel distance 5

Testing Data	K Value	Accuracy	
		Euclidean	Manhattan
Fold 1	1	69%	68%
	3	56%	58%
	5	48%	49%
	7	40%	42%
Fold 2	1	74%	74%
	3	61%	61%
	5	51%	54%
	7	44%	44%
Fold 3	1	73%	73%
	3	63%	62%
	5	53%	51%
	7	43%	43%

Table 10. Accuracy results using cropped image and pixel distance 7

Testing Data	K Value	Accuracy	
		Euclidean	Manhattan
Fold 1	1	69%	68%
	3	56%	58%
	5	49%	49%
	7	42%	44%
Fold 2	1	70%	72%
	3	62%	64%
	5	56%	60%
	7	46%	45%
Fold 3	1	71%	72%
	3	61%	62%
	5	53%	54%
	7	43%	45%

It turns out that using the cropped image does not produce better accuracy than 75% obtained in Table 7. Furthermore, from Table 3, Table 5, and Table 7, the average accuracy results are compared using distance calculations with Manhattan Distance and Eulclidean Distance. Table 11 presents the results that were obtained.

Table 11. Average accuracy of the two distance calculations

K Value	Accuracy	
	Euclidean	Manhattan
1	69%	68%
3	56%	58%
5	49%	49%
7	42%	44%

With a combination of parameters that got the best accuracy and because the comparison of distance accuracy between the two calculations is not too different, a single test system is made with batik images from the internet (outside the dataset) and using both distance calculations. In Fig. 2, a screenshot is shown when the batik recognition application program is run with the input data for the image of batik Betawi. The test results are correct because the output of the program is information that the image of the batik is batik Betawi.

 https://doi.org/10.1051/e3sconf/202447502012

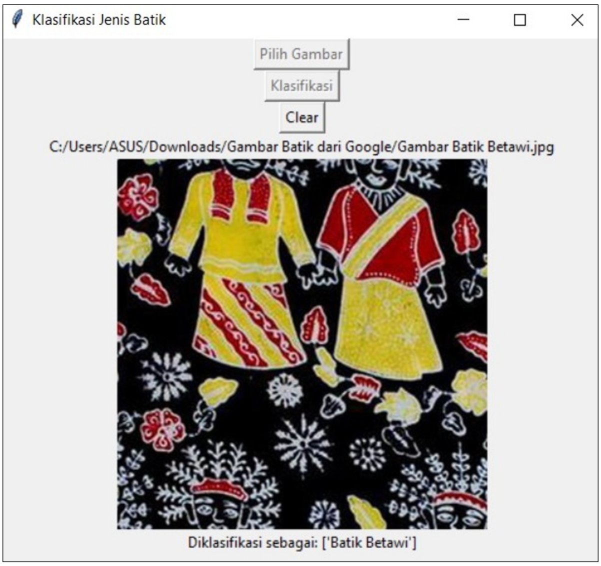

Fig. 1. GUI display after classifying the image

Tables 12 and Table 13 presents all the combination information of image input data and output results from the application during a single test.

Table 12. Single trial results with Manhattan Distance

Batik image	Classified as
Bali	Garutan
Bali	Tambal
Betawi	Betawi
Betawi	Betawi
Celup	Pekalongan
Celup	Cendrawasih
Cendrawasih	Cendrawasih
Ceplok	Keraton
Ceplok	Pekalongan
Ciamis	Lasem
Ciamis	Sidoluhur
Garutan	Gentongan
Gentongan	Sidomukti
Kawung	Kawung
Kawung	Parang
Keraton	Keraton
Keraton	Tambal
Lasem	Lasem
Megamendung	Megamendung
Megamendung	Megamendung
Parang	Cendrawasih
Pekalongan	Gentongan
Priangan	Priangan
Sekar	Gentongan
Sidoluhur	Tambal
Sidomukti	Tambal
Sogan	Ciamis
Sogan	Betawi

Tambal	Tambal
Tambal	Priangan

Table 13. Single trial results with Euclidean Distance

Batik image	Classified as
Bali	Garutan
Bali	Tambal
Betawi	Betawi
Betawi	Betawi
Celup	Sidoluhur
Celup	Cendrawasih
Cendrawasih	Cendrawasih
Ceplok	Keraton
Ceplok	Pekalongan
Ciamis	Lasem
Ciamis	Sidoluhur
Garutan	Garutan
Gentongan	Sidomukti
Kawung	Parang
Kawung	Pekalongan
Keraton	Keraton
Keraton	Garutan
Lasem	Lasem
Megamendung	Betawi
Megamendung	Megamendung
Parang	Kawung
Pekalongan	Kawung
Priangan	Priangan
Sekar	Sekar
Sidoluhur	Tambal
Sidomukti	Tambal
Sogan	Sekar
Sogan	Ciamis
Tambal	Tambal
Tambal	Priangan

Apparently, the results of a single test using the Manhattan Distance and Euclidean Distance calculations both succeeded in classifying 10 out of 30 batik images correctly.

4 Results and analysis

From a series of tests that have been carried out, the model that produces the best accuracy is obtained with pixel distance parameter = 7, K value = 1, with the 1st fold 0° degree feature as test data and the 2nd and 3rd fold 0° degree plus the feature data from 45°, 90°, and 135° degree features as training data with StandardScaler. Both Manhattan and Euclidean distance calculations does not produce much different accuracy. Although in a single test, the two distance calculations both succeeded in classifying 10 out of 30 batik images correctly, but on average Manhattan distance calculations got slightly better accuracy than Euclidean which can be seen in Table 11. Furthermore, it can be concluded that StandardScaler greatly affects the classification results, as evidenced by the significant increase in accuracy that can be seen in Table 3, Table 5, and Table 7 compared to Table 2, Table 4, and Table 6. The pixel distance parameter also affects the classification results which can be seen from the increase in accuracy obtained from Table 3, Table 5, and Table 7. Datasets are also very influential in classifying batik types. It is better if the batik images to be used as a dataset are taken with the same photo-taking technique. According to the author, the dataset used for this research is still dirty because there are still images that are not batik in the dataset and some batik images were taken with different lighting, angles, and distances so that the author has to sort the dataset manually.

5 Conclusion

KNN and GLCM can be applied to classify batik types. However, to keep pace with the development of increasingly advanced technology, further research still needs to be done to improve the accuracy of batik classification so that people can easily recognize the types of Indonesian batik with their technology.

References

1. UNESCO, Indonesian Batik (2009)
2. Kristina, Hari Batik Nasional (2021)
3. C. Irawan, E.N. Ardyastiti, D.R.I.M. Setiadi, E.H. Rachmawanto, C.A. Sari, Effect of the Number of GLCM Features on Classification Accuracy of Lasem Batik Images using K-Nearest Neighbor, in Research of Information Technology and Intelligent System, 33-36 (2019)
4. S. Alam, Batik Mendunia, Ajak Milenial Mengetahuinya (2021)
5. H.S. Khamis, K.W. Cheruiyot, S. Kimani, Application of k-Nearest Neighbour Classification in Medical Data Mining, in International Journal of Information and Communication Technology Research, **4**, 121-128 (2014)
6. S.B. Imandoust, M. Bolandraftar, Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Event, in International Seminar on Journal of Engineering Research and Applications, **3**, 605-610 (2013)
7. J.W. Yodha, A.W. Kurniawan, Perbandingan Penggunaan Deteksi Tepi Dengan Metode Laplace, Sobel, dan Prewit dan Canny Pada Pengenalan Pola, in Techno.COM, **13**, 189-197. (2014)
8. W. Wahyono, I.N.P. Trisna, S.L. Sariwening, M. Fajar, D. Wijayanto, Perbandingan penghitungan jarak pada k-nearest neighbour dalam klasifikasi data tekstual, in Jurnal Teknologi dan Sistem Komputer, **8**, 54-58 (2020)
9. Setiohardjo, N. M. dan Harjoko, A., 2014, Analisis Tekstur untuk Klasifikasi Motif Kain (Studi Kasus Kain Tenun Nusa Tenggara Timur), Indonesian Journal of Computing and Cybernetics Systems, **8**, 177-188.
10. Z. Musiafa, Perancangan Ekstraksi Fitur Motif Sasaringan Menggunakan Algoritma Naïve Bayes Berbasis Color Histogram dan GLCM, in Technologia, **8**, 108-117 (2017)
11. P. Mohanaiah, P. Sathyanarayana, L. GuruKumar, Image Texture Feature Extraction Using GLCM Approach, in Internasional Journal of Scientific and Research Publications, **3**, 2250-3153 (2013)
12. J. Wahyudi, I. Maulida, Pengenalan Pola Citra Kain Tradisional Menggunakan GLCM dan KNN, in Jurnal Teknologi Informasi Universitas Lambung Mangkurat, **4**, 43-48 (2019)

13. Y. Kusumawati, A. Susanto, I.U.W. Mulyono, D.P. Prabowo, Klasifikasi Batik Kudus Berdasarkan Pola Menggunakan KNN dan GLCM, in Prosiding Semniar Nasional Lembaga Penelitian dan Pengabdian Masyarakat Universitas Muhammadiyah Purwokerto, **2**, 464-469 (2020)
14. R. Andrian, M.A. Naufal, B. Hermanto, A. Junaidi, F.R. Lumbanraja, KNN Classification for Recognition of the Batik Lampung Motifs, in Journal of Physics: Conference Series, 1338, 1-6 (2019)
15. Z.Y. Lamasigi, DCT Untuk Ekstraksi Fitur Berbasis GLCM pada Identifikasi Batik Menggunakan KNN, in Jambura Journal of Electrical and Electronics Engineering, **3**, 1-6 (2021)
16. C. Jatmoko, D. Sinaga, Ekstraksi Fitur GLCM pada KNN Dalam Mengklasifikasi Motif Batik, in Prosiding SENDI_U, (2019)
17. Y. Ganis K., I. Santoso, R.R. Isnanto, , Klasifikasi Citra Dengan GLCM Pada Lima Kelas Biji-Bijian, in Skripsi on Fakultas Teknik, Universitas Diponegoro (2011)
18. D.P. Pamungkas, Ekstraksi Citra menggunakan Metode GLCM dan KNN untuk Indentifikasi Jenis Anggrek, in Innovation in Research of Informatics, **1**, 51-56 (2019)
19. W.K. Oktalao, I.D.M.B.A. Darmawan, I.W. Santiyasa, I.P.G.H. Suputra, I.G.N.A.C. Putra, Klasifikasi Motif Kain Tradisional Cepuk Menggunakan GLCM dan KNN, in Jurnal Elektronik Ilmu Komputer Udayana, **11**, 545-552 (2023)
20. L. Hakim, S.P. Kristanto, D. Yusuf, F.N. Afia, Pengenalan Motif Batik Banyuwangi Berdasarkan Fitur Gray Level Co-Occurrence Matrix, in Jurnal TEKNOINFO, **16**, 1-7 (2022)
21. J. Nelson, What is Image Preprocessing and Augmentation?,(2020)
22. A. Fredrick, Getting Started with Image Preprocessing in Python (2021)
23. D. Marlina, N.F. Putri, A. Fernando, A. Ramadhan, Implementasi Algoritma K-Medoids dan K-Means unuk Pengelompokkan Wilayah Sebaran Cacat pada Anak, in Jurnal CoreIT, **2**, 64-71 (2018)
24. A. Kharwal, StandardScaler in Machine Learning (2020)
25. A.P. Wibawa, M.G.A. Purnama, M.F. Akbar, F.A. Dwiyanto, Metode-metode Klasifikasi, in Prosiding Seminar Ilmu Komputer dan Teknologi Informasi, **3**, 134-138 (2018)
26. Suyanto, Machine Learning Tingkat Dasar dan Lanjut (2018)
27. M. Nishom, Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square, in Jurnal Informatika: Jurnal Pengembangan IT, **4**, 20-24 (2019)
28. A. Pamungkas, Ekstraksi Ciri Citra Grayscale (2017)
29. R.A. Surya, A. Fadlil, A. Yudhana, Ekstraksi Ciri Metode GLCM dan Filter Gabor Untuk Klasifikasi Citra Batik Pekalongan, in Jurnal Informatika: Jurnal Pengembangan IT, **2**, 23-26 (2017)
30. W.I. Praseptiyana, A.W. Widodo, M.A. Rahman, Pemanfaatan Ciri GLCM Untuk Deteksi Melasma Pada Citra Wajah, in Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, **3**, 10402-10409 (2019)
31. Softcients, Ciri Fitur dengan GLCM (2021)
32. S. Narkhede, Understanding Confusion Matrix (2018)
33. J. Mohajon, Confusion Matrix for Your Multi-Class Machine Learning Model (2020)
34. M. Sanjay, Why and how to Cross Validate a Model? (2018)

35. Y. Widyaningsih, G.P. Arum, K. Prawira, Aplikasi K-Fold Cross Validation Dalam Penentuan Model Regresi Binomial Negatif Terbaik, in Jurnal Ilmu Matematika dan Terapan, **15**, 315-322 (2021)
36. J. Brownlee, A Gentle Introduction to k-fold Cross-Validation (2018)