Journal of Information Technology and Computer Science
Volume 5, Number 1, April 2020, pp. 23-31
Journal Homepage: www.jitecs.ub.ac.id

K-Value Effect Based on Combination GLCM Angle and KNN for Detecting Smart Wheelchair

Ahmad Wali Satria Bahari Johan[1], Fitri Utaminingrum[2], Agung Setia Budi[3]

[1,2]Computer Vision Research Group, Faculty of Computer Science, Brawijaya University, Malang, Indonesia
[3]Embeded System Group, Faculty of Computer Science, Brawijaya University, Malang, Indonesia
{[1]ahmadsatria27@gmail.com, [2]f3_ningrum@ub.ac.id, [3]agungsetiabudi@ub.ac.id}

Received 11 September 2019; accepted 10 October 2020

Abstract. This study aims to analyze the k-value on K nearest neighbor classification. k-value is the distance used to find the closest data to label the class from the testing data. Each k-value can produce a different class label against the same testing data. The variants of k-value that we use are k=3, k=5 and k=7 to find the best k-value. There are 2 classes that are used in this research. Both classes are stairs descent and floor classes. The GLCM method is used to extract features. The data we use comes from videos obtained from the camera on the smart wheelchair taken by the frame. Refer to the results of our tests, the best k-value is obtained when using *k=7* and angle *0°* with accuracy is 92.5%. The stairs descent detection system will be implemented in a smart wheelchair.

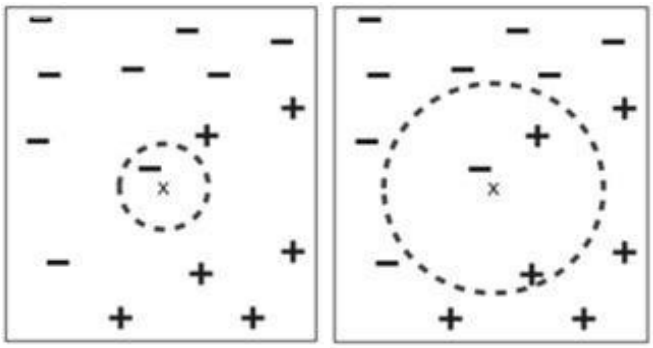
Keyword. K nearest neighbor, GLCM, smart wheelchair.

1 Introduction

Classification is a categorization process where objects are recognized, distinguished and understood based on training data sets. In the classification of supervised learning techniques where a set of properly defined training and observations is available. Algorithms that implement classification are often known as classifiers, and observations are often known as instances. There are 2 types of training in classification. Both are eager learners and lazy learners. The eager learner will build a model from the major training tuples before receiving a test tuple. The lazy learner is a learner that simply stores it (or does only a little minor processing) and waits until it is given a test tuple when given a training tuple[1]. K nearest neighbor classifiers are examples of lazy learners..

K nearest neighbor algorithm, tuples are predicted based on the class of its nearest neighbors[2]. Figure 1 shows the K nearest neighbors of an *x* record, data points that have *k* with the closest distance to *x*. In Fig. 1(a) shows when using *k=1* negative data is the closest data to *x*, so *x* is labeled as negative class. In Fig. 1(b) there is 1 negative data and 2 positive data closest to *x* when using *k=3*, so the majority of data is taken that is positive data to label *x*. Refer to Fig. 1, it can be seen that the use of *k-value* is

very influential on the classification results obtained. This certainly affects the accuracy of the classification performed.



(a) (b)

Fig. 1 *k-value* variation

The previous research by Abdul Rohman who used 5 types of k values (*k=1*, *k=2*, *k=3*, *k=4*, and *k=5*) in the KNN classification to predict student graduation. The analysis results obtained the best accuracy with *k=5* of 85.15% while the lowest accuracy when using *k=2* [3]. In another study conducted by M.Akhil Jabbar in classifying heart disease using KNN. The study uses 3 variations of *k-values* in finding the closest data. The *k-values* used are *k=1*, *k=5,* and *k=7*. Referring to the results of the testing that have been obtained the best accuracy when using *k=1*[4].Refer to several previous research, it can be concluded that the *k* value can affect the high accuracy obtained. This study will use several k values (*k=3*, *k=5*, *k=7*). From these *k* values, we will find the best accuracy in classifying stairs down and floors.

Research on the stairs descent detection system has been carried out by Rai Munoz[5]. In this study using the RGB-D method based on the image to detect stairs going down. The accuracy obtained is 88.89%. This research will analyze to get the best k value in classifying 2 classes. The two classes are stairs descent and floor classes. The research aims to detect stairs descent by using a camera on a smart wheelchair. The *k-value* with the best accuracy will be used in the stairs detection system for the smart wheelchair.

Proposed Method

2.1 System Design

In this study, we analyzed the effect of *k-value* on *K* nearest neighbor classification process. The *k-value* that we used in our study is *k=3, k=5,* and *k=7*. The angle of GLCM that we use to extract are *0°*, *45°*, *90°*, and *135°*. By using 3 kinds of *k-value* and four angles of GLCM, the results of the accuracy will be compared to get the best *k-value* and angle in identifying the stairs descent. Fig. 2 is a general description of our system. Here is our step of the experiment.

1. The system gets captured image. The image resolution is 480x640.
2. Cropped on the bottom third of the image section.
3. Convert the RGB image into grayscale image. Fig. 3 explains the cropping process.
4. Feature extraction. In this stage we use 5 features of GLCM.

5. Choose the *k value* to classify.
6. Classify the testing data with K nearest neighbor algorithm.
7. The system will give the notification of classification between floor or stairs descent.

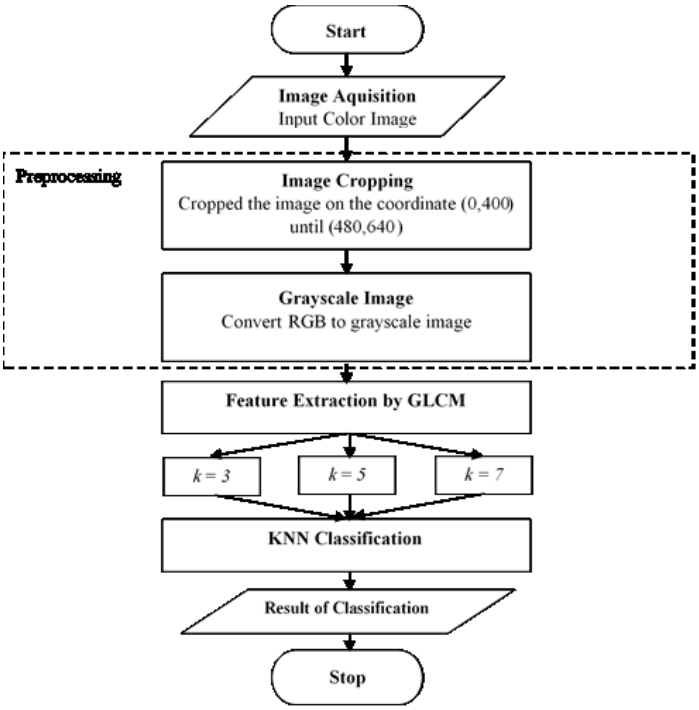


Fig. 2 System Design

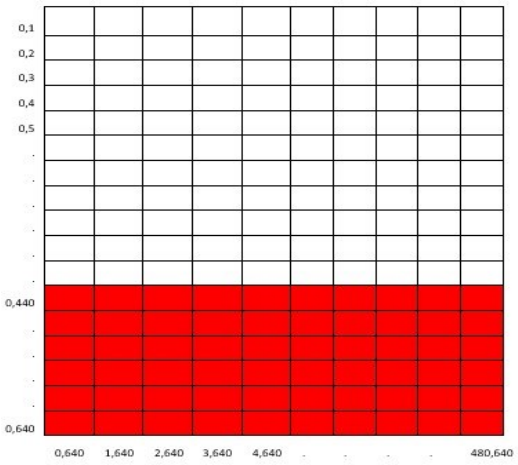


Fig. 3 Cropping area

2.2 Converting RGB to Grayscale Image

A grayscale image is an image that only has a gray level color. The use of grayscale images because it requires less information provided at each pixel compared

to color images. Gray in the grayscale image are the colors R, G, and B which have the same intensity. So that in grayscale images only require a single intensity value compared to color images it requires three intensities for each pixel. Figure 4 shows the intensity of the grayscale image is stored in 8 integer bits provides 256 possibilities ranging from level 0 to 255. Fig. 4 is the grayscale level[6].

Fig. 4 Grayscale level

In this study, images were obtained from frames from videos recorded on a smart wheelchair. The image is in the form of an RBG image. Then convert from RGB to grayscale image using equation 1[7].

$$Grayscale = \frac{red+green+blue}{3} \quad (1)$$

2.3 Feature Extraction by Gray Level Co-occurrence Matrix

Gray Level Co-occurrence Matrix (GLCM) is a method used for texture analysis or feature extraction. GLCM is a matrix that describes the frequency of occurrence of pairs of two pixels with a certain intensity in a certain distance and direction in the image[8]. The direction of the orientation of the angle formed in four directions with an interval of *45°*, which is *0°*, *45°*, *90°*, and *135°*[9][10]. While the distance usually used are *1, 2, 3,* and *etc*.

1. The steps are used to calculate the Gray Level Co-occurrence Matrix features of the grayscale image used, among others[11]: 1. Create a matrix work area from the image input.
2. The formation of the initial GLCM matrix of two-pixel pairs which are aligned according to the direction of *0°*, *45°*, *90°*, or *135°*. Fig. 5 is the GLCM angle visualization.
3. A symmetrical matrix by summing the initial GLCM matrix with the transpose value.
4. Normalize the GLCM matrix by dividing each matrix element by the number of pixel pairs.

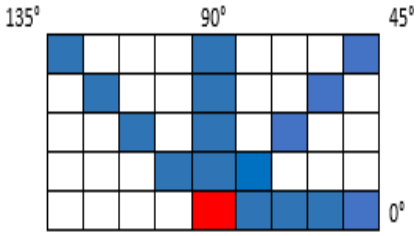


Fig. 5 GLCM angle

```
im = skimage.img_as_ubyte(im)
g = skimage.feature.greycomatrix(im, [1], [0], 256, symmetric=True, normed=True)
a1 = skimage.feature.greycoprops(g, 'contrast')[0][0] a2 = skimage.feature.greycoprops(g, 'energy')[0][0] a3 = skimage.feature.greycoprops(g, 'homogeneity')[0][0] a4 = skimage.feature.greycoprops(g, 'correlation')[0][0] a5 = skimage.feature.greycoprops(g, 'dissimilarity')[0][0] a6 = skimage.feature.greycoprops(g, 'ASM')[0][0]
data = [a1, a2, a3, a4, a5, a6] def greycoprops :
- 'contrast': :math:`\\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2`
- 'dissimilarity': :math:`\\sum_{i,j=0}^{levels-1}P_{i,j}|i-j|`
- 'homogeneity': :math:`\\sum_{i,j=0}^{levels-1}\\frac{P_{i,j}}{1+(i-j)^2}` - 'energy': :math:`\\sqrt{ASM}`
'correlation':: math:: \\sum_{i,j=0}^{levels-1} P_{i,j}\\left[\\frac{(i-\\mu_i) \\ (j-\\mu_j)}{\\sqrt{(\\sigma_i^2)(\\sigma_j^2)}}\\right]
```

Fig. 6 Pseudo-code of calculating GLCM features

5. Extracting 5 feature of GLCM. That are contrast, homogeneity, correlation, energy, and dissimilarity. Fig. 6 shows the pseudo-code of calculating each feature of GLCM.

2.2 K Nearest Neighbor

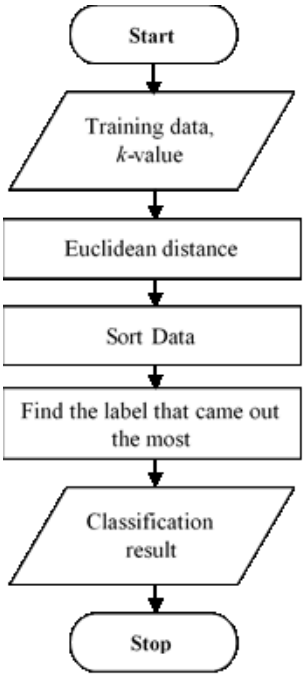


Fig. 7 Flowchart of KNN

The K nearest neighbor algorithm (KNN) is a method that uses the supervised algorithm[12]. The difference between supervised learning and unsupervised learning is that supervised learning aims to find new patterns in data by connecting existing data patterns with new data. Whereas in unsupervised learning, data does not yet have any pattern, and the purpose of unsupervised learning is to find patterns in a data[13]. KNN is included as an integrated learning group. This algorithm is also one of the lazy learning techniques. KNN is done by finding groups of *k-value* in the closest training data to objects on new data or testing data[14]. Fig. 7 is the flowchart of K nearest neighbor classification. The following are the steps of the KNN method carried out in this study:

1. Determine the value of *k*-value.
2. Calculate the euclidean distance between the test data and all training data. The euclidean formula is used, as shown in equation 2.

$$D(a,b) = \sum_{i=0}^{n}(X_i - Y_i)^2 \quad (2)$$

(a, b) : closest euclidean distance
X : data 1
Y : data 2
i : feature to – n
n : number of features

3. Sort data that has the smallest distance.
4. Determine the group of test results based on the majority label of the K nearest neighbor.

3. Result

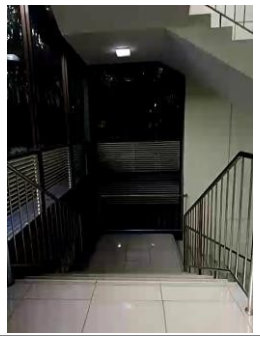
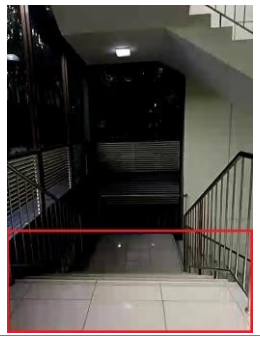
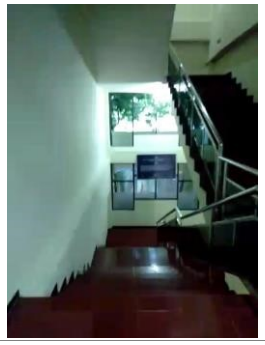
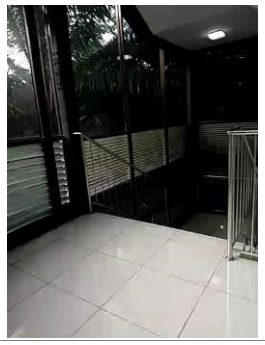
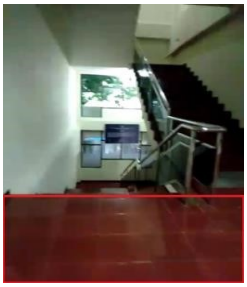
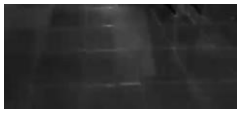
3.1 Input Color Image

In the initial stage, the image input is done. Image is obtained from the retrieval process from the frame on the video. The initial image obtained is an RGB image. The original size of the image is 640x480 pixels. Table 1 shows the RGB image obtained. Image number 1 and 2 are the stairs descent image. Image number 3 and 4 are the floor image.

3.2 Image Cropping and Convert to Grayscale

The original image is cropped into the fixed coordinate. This process will take on the coordinate (0,440) until (480,640). The cropping image resolution will be 480x200. The crooping coordinate is fixed on that area. After cropping image, the next step is to convert from an RGB image to a grayscale image. The image that is converted to grayscale is a cropped image. Table 1 shows the converting results from RGB images to grayscale images. The grayscale image will be calculated by GLCM and get the value of 5 GLCM features.

Table 1 Visual ilustration of preprocessing process

No	Image Accuisition	Cropped Image	Grayscale Image
1			
2			
3			
4			

3.3 Accruration and Result Analysis

At this stage, the accuracy of the system is tested in classifying input images. There are 2 classes used. The two classes are the stairs descent class and the floor class. To

get the accuracy value, a measurement of a new sample that is correctly classified is carried out.

$$Accuracy = \frac{TN+TP}{TP+FP+TN+FN} \quad (3)$$

From equation 3 it can be seen that True Positive (TP) and True Negative (TN) are the numbers of positive and negative classes that are correctly classified. While False Positive (FP) and False Negative (FN) are incorrectly classified samples[15].

This study used 100 photos of stairs descent and 100 photos of the floor used as training data. While the test sample uses as many as 20 photos of stairs descent and 20 floor photos. Experiments were carried out using angel *0°*, *45°*, *90°*, and *135°*. While the distance used is 1. In the K nearest neighbor classification process will be *k=3*, *k=5* and *k=7*. That k-value are tested to determine the most optimum *k-value* in clasifying between stairs descent and floor.

Table 2 shows the classification using *k=3*. Classification was done using several angles. The classification results that get the best accuracy are using angel *0°* and *135°* that get true classify 19 data of stairs descent and 17 data of floor. The accuracy value was 90%. While the worst accuracy is obtained when using angle *90°* with accuracy is 82.5% that has shown in Fig. 8. Futhermore the testing accuracy using *k=5* is shown in table 2. The best accuracy value is obtained with a value of 90%. The best angels are *0°* and *135°* that get 19 true data in classify stairs descent and 17 true data in classify floor. While the lowest accuracy is with a value of 82.5% when the angle that is used is *90°*. The last testing was using *k=7*. The best accuracy was 92.5% that has shown in Fig. 8. Table 2 shows the result of our testing. Angel *0°* got 20 true data of stairs descent and 17 true data in classifying floor. And the worst accuracy is 82.5% when we used angles *45°* and *90°*.

Table 2 Classification with *k=3*

KNN *k*-value	Classes	GLCM Angles							
		0°		45°		90°		135°	
		T	F	T	F	T	F	T	F
3	Stairs Descent	19	1	20	0	20	0	19	1
	Floor	17	3	14	6	12	8	17	3
5	Stairs Descent	19	1	20	0	20	0	19	1
	Floor	17	3	14	6	13	7	17	3
7	Stairs Descent	20	0	20	0	20	0	20	0
	Floor	17	3	13	7	13	7	16	4

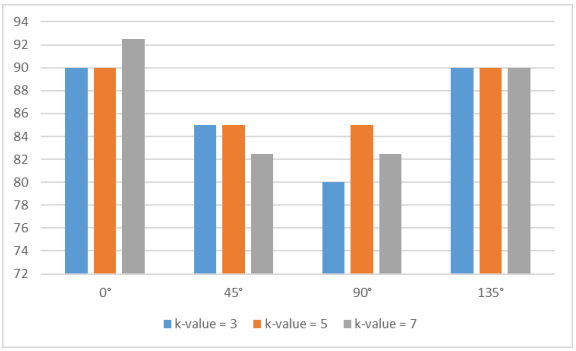


Fig. 8 Accuration Rate

4 Conclusion

This study aims to find out the best *k value* from K nearest neighbor classification in detecting stairs descent which will be implemented in the smart wheelchair. We also compare the best angle of GLCM with angle *0°*, *45°*, *90°*, and *135°* is tried one by one to get the value of its features. Then the classification process is done by using the K Nearest Neighbor method for each angel. This research used *k=3*, *k=5*, and *k=7* to classify. Refer to the results of tests that have been done, it was found that *k=7* get the best level of accuracy with an angle *0°*. The accuracy that we get is 92.5%. While the *k value* is *k=3* and angle *90°* gets the lowest level of accuracy, which is 80%.

References

1. D. D. Bhavani, A. Vasavi, and P. T. Keshava.: Machine Learning : A Critical Review of Classification Techniques. pp. 22–28, (2016).
2. P. Mulak and N. Talhar.: Analysis of Distance Measures Using K-Nearest Neighbor Algorithm on KDD Dataset. vol. 4, no. 7, pp. 2101–2104, (2015).
3. A. Rohman.: k-nearest neighbor (k-nn) algorithm model for student graduation prediction. (2012).
4. M. Akhil, B. L. Deekshatulu, and P. Chandra.: Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm. *Procedia Technol.*, vol. 10, pp. 85–94, (2013).
5. Rai Munoz.: Depth-Aware Indoor Staircase Detection And Recognition For The Visually Impaired. Dept . of Electrical Engineering The City College of New York , CUNY New York, NY 10031.
6. A.Wali Satria Bahari Johan and Fitri Utaminingrum.: Stairs Descent Identification for Smart Wheelchair by Using GLCM and Learning Vector Quantization. pp. 64–68, (2019).
7. S. Macˆ and J. Kelner.: A comparative study of grayscale conversion techniques applied to SIFT descriptors. vol. 6, no. 2, pp. 30–36, (2015).
8. S. J. A. Sarosa.: Mammogram Breast Cancer Classification Using Gray-Level CoOccurrence Matrix and Support Vector Machine. *2018 Int. Conf. Sustain. Inf. Eng. Technol.*, pp. 54–59, (2018).
9. C. Dewi and S. Sundari.: Texture Feature On Determining Quantity of Soil Organic Matter For Patchouli Plant Using Backpropagation Neural Network. vol. 4, no. 1, pp. 1–14, (2019).
10. M. A. Ben Atitallah, R. Kachouri, M. Kammoun and H. Mnif.: An efficient implementation of GLCM algorithm in FPGA. *2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*, Hamammet, Tunisia, 2018, pp. 147-152. 11. M. Hall-beyer.: Glcm Texture : A Tutorial. no. March, (2017).
12. M. S. Sarma, Y. Srinivas, M. Abhiram, L. Ullala, M. S. Prasanthi and J. R. Rao.: Insider Threat Detection with Face Recognition and KNN User Classification. 2017 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Bangalore, 2017, pp. (39-44).
13. S. S. Tabrizi and N. Cavus.: A Hybrid KNN-SVM Model for Iranian License Plate Recognition. *Procedia Comput. Sci.*, vol. 102, no. September, pp. 588–594, (2016).
14. P. Sonar, U. Bhosle, and C. Choudhury.: Mammography classification using modified hybrid SVM-KNN. *Proc. IEEE Int. Conf. Signal Process. Commun. ICSPC 2017*, vol. 2018–Janua, no. July, pp. 305–311, (2018).
15. A. Ahirwar.: Study of Techniques used for Medical Image Segmentation and Computation of Statistical Test for Region Classification of Brain MRI. no. April, pp. 44–53, (2013).