Bulletin of Electrical Engineering and Informatics
Vol. 9, No. 4, August 2020, pp. 1446~1453
ISSN: 2302-9285, DOI: 10.11591/eei.v9i4.2348

Road surface classification based on LBP and GLCM features using kNN classifier

Arthur Ahmad Fauzi[1], Fitri Utaminingrum[2], Fatwa Ramdani[3]
[1,2]Computer Vision Research Group, Faculty of Computer Science, Brawijaya University, Indonesia
[3]Geoinformatic Research Group, Faculty of Computer Science, Brawijaya University, Indonesia

Article Info

Article history:

Received Oct 30, 2019
Revised Feb 20, 2020
Accepted Mar 23, 2020

Keywords:

Classification
GLCM
kNN
LBP
Unmanned ground vehicle

ABSTRACT

Autonomous ground vehicle (UGV) technology has shown a fast development this past year and proven to be useful. The use of UGV technology is restricted on a particular road condition. Classification of the road is an essential process in UGV, especially to control the autonomous vehicle. For example, the speed could be adjusted by referring to the road type, these processes require a fast-computational time. This research focuses on finding the most discriminant feature while keeping the number of features into a minimum to obtain fast computational time and accurate classification result. One can experiences difficulties because the condition of the road varies, this research proposes a combination of gray level co-occurrence matrix (GLCM) a statistical method to extract feature and local binary pattern (LBP) feature to improve the robustness of the features. The kNN classifier is used to do the classification with the accuracy of 98% and 12 picture processed per second.

Corresponding Author:

Arthur Ahmad Fauzi,
Faculty of Computer Science,
Brawijaya University,
Jl. Veteran, Ketawanggede, Lowokwaru, Malang, Jawa Timur, Indonesia.
Email: arthurahmadf@gmail.com

1. INTRODUCTION

The number of four-wheeled vehicle users in Indonesia is increasing every year and the number of accidents increases accordingly, these accidents are caused by three factors, inadequate infrastructure, inadequate vehicles, and human error, human error contribute 61% to the incident counts. This needs to be a concern in order to improve road user safety. The use of autonomous technology can reduce accident rates caused by human error [1]. Autonomous Technologies has received attention over the last years because of the rapid advancement in this technology.

One of the technologies is the unmanned ground vehicles (UGV), it is a self-driving vehicle that can automatically capture much different information in the driving environment. One of the features in autonomous technology is the classification of road surface types, which can be used to regulate autonomous vehicle behavior, such as regulating vehicle speed, giving an early warning system to road conditions in front of it and keeping the vehicle stay in the track. Another thing that is no less important is the method proposed to carry out the road classification process needs to have fast computational time because it will be implemented in real-time conditions.

Visual information about the road surface is one of the crucial information that can be captured to obtain the safety of UGV [2]. Visual information of the road surface can be obtained using camera devices, visual information on the road is not always free of noise, sometimes there is a shadow on the road that causes the color of the road surface is uneven, this will affect the results of the classification. Many

researches about processing visual information on the road have been carried out. K-min et al. [3] use visual information to classify the road sign using the Support Vector Machine classifier. Hee Chang et al. [4] uses not only visual information and adds a laser scanner to improve the detection of obstacles on the road. Slavkovikj et al. [5] use GLCM and eight other features to classify the road surface using their proposed classification method. The latest research by Marianingsih et al. [6], uses 4 features from GLCM and kNN to classify the road surface.

Selecting the most discriminant feature of the object will lead to more precise classification [7, 8], the discriminant features need to be extracted from the road surface image and as the number of a feature need to be as minimum as possible. Gray level co-occurrence matrix (GLCM) is a statistical method that describes the spatial distribution of gray values [9], which have been used in many different fields of study that use texture as features, such as bamboo strips defect detection [10] and a road distress detection by M. Gavilan et al. [5]. We propose a new feature combination method in order to overcome the problem of classification of road surfaces polluted by shadows by combining two features in the GLCM method and one feature in the LBP method. Where the GLCM method is tasked with detecting the surface of the road that has texture as its identifier and LBP is used to overcome the problem of shadows on the surface of the road. In the GLCM method, Energy and Contrast features will be taken while LBP uses the Mean-LBP feature; the three features are used to get fast computing time, compared with the same approach by using more features.

2. RESEARCH METHOD

In our proposed research shown in Figure 1. We combine between GLCM feature those are energy and contrast that have obtained from the extraction of the GLCM and the mean-LBP feature which is resulted from the LBP extraction process. The combination of these feature is proposed to produce a minimum number of discriminant feature to minimalize the computational time in the classification [11]. The classification process in our proposed method uses kNN with Euclidean distance to measure the closest value between data testing and training, the classification result will be analyzed to measure the performance of the features used. All of the steps are discussed in each sub-section.

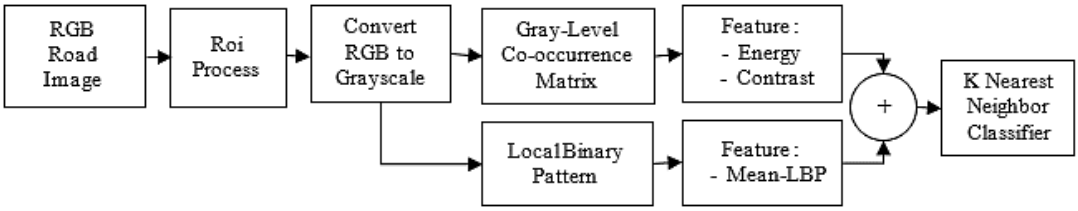

Figure 1. Research method

2.1. Region of interest (ROI) process

This research will classify between road surface image and non-road surface image. The resolution of the images is 1242x375 shown in Figure 2. Then the images are cropped into 64x64 pixel size as shown in Figure 3, the cropped image is taken from fixed coordinates of (*x* 600:*y* 300). Sample of the cropped ROI is shown in Figure 4.

Figure 2. Sample data of the original road image

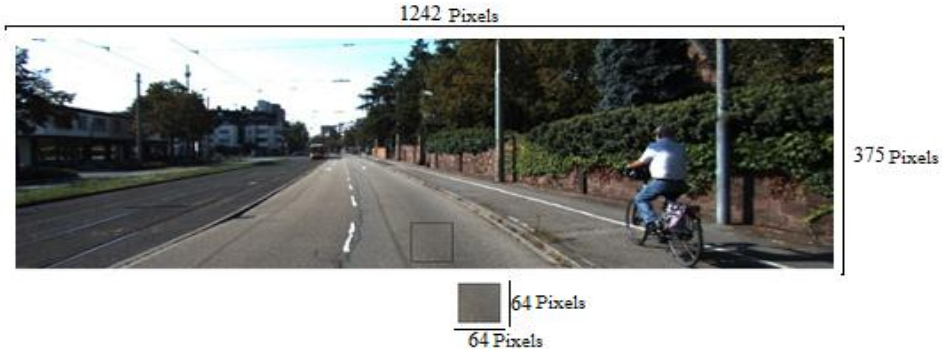

Figure 3. ROI Extraction

(a) (b)

Figure 4. The samples of (a) cropped image from the road surface from Figure 3, (b) sample image of non-road surface from Figure 4

2.2. Convert RGB to grayscale

The conversion from color images or red green blue (RGB) images to grayscale is a dimensional reduction from three dimensional to a single dimension, providing a single dimension to manipulate thus reducing the computational load on image processing [12, 13]. Image involving only intensity are called grayscale images. Gray levels represent the interval number of quantization in grayscale image processing. The other purpose of converting the color images to grayscale is to integrate the data to be used on the GLCM algorithm. The intensity of each pixel can have from 0 to 255, with 0 being black and 255 being white [14] the formula for Gray image is shown in (1);

$$G(i,j) = \frac{B(i,j)+G(i,j)+R(i,j)}{3} \quad (1)$$

denotes *B, G, R* the digital data in blue, green, and red color channels respectively, *i* represent the row and *j* represent the column, illustrated in Figure 5.

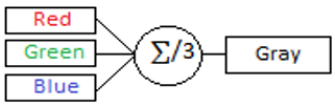

Figure 5. Illustration of grayscale conversion

2.3. Gray level co-occurrence matrix (GLCM)

Gray level co-occurrence matrix (GLCM) is categorized as texture analysis and it is considered the most common and convenient algorithm [15], which process an image and reflect its second-order conditional probability value of pixel combination (*i,j*) and has a specific angle (θ), distance (*d*), and with different intensity [16]. Usually the intensity is 8x8 or 16x16 as it will not produce a lot of redundant information [17]. The value of *d* is set to 1. The GLCM matrix is produced by calculating how often a pixel with certain gray values *i* occurs horizontally (θ=0) adjacent with j or occurs vertically (θ=90). The GLCM matrix can produce 36 different kinds of features representing the grayscale image depending on the *d* or θ,

using all of that features may result in a slow or inaccurate prediction, because not all of the features serve the purpose of our analysis. In our research, we chose only energy and contrast as it most represents the characteristic of the road surface. The value of θ is usually 0,45,90, 135, or 185 as shown in Figure 6, the occurrence search determined by the selected angle, in Figure 7 we use 0 and the search will become horizontal, while $\theta=90$ will make the search go vertical. The value of $d=1$ means that the occurrence search of certain pixel combination is 1 pixel away, while $d=2$ will move the pixel by 2, visual illustration is provided in Figure 7. After the GLCM matrix is formed, the Energy and Contrast feature can be extracted.

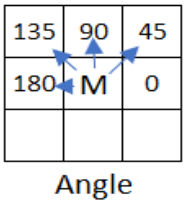

Figure 6. Illustration of different angle (θ) affect GLCM

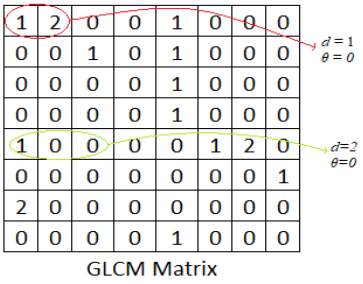

Figure 7. Illustration of how d works in GLCM

2.3.1. Energy feature

Energy is obtained from the square root of the angular second movement (ASM) value [18]. ASM measures the homogeneity of an image from a symmetrical GLCM matrix as shown in Figure 8, to obtain the symmetrical matrix we use (2);

$$G_s(i,j) = G(i,j) + G^T(i,j) \quad (2)$$

$\boldsymbol{G(i,j)}$ denotes the GLCM matrix, (i,j) denotes the pixel value, and $\boldsymbol{G^T(i,j)}$ is the transpose of the GLCM matrix. Then the ASM values can be calculated using (3);

$$ASM = \sum_{i=0}^{S-1} \sum_{j=0}^{S-1} G_d\,(i,j)^2 \quad (3)$$

$\boldsymbol{G_d(i,j)^2}$ denotes pixel values of the symmetrical matrix while S is the size of the GLCM matrix. When the value of Energy is equal to 1 that means the image is constant [19]. The Energy feature can be calculated using (4);

$$Energy = \sqrt{ASM} \quad (4)$$

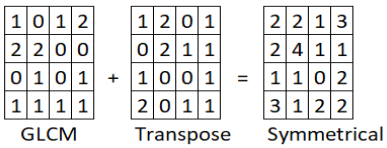

Figure 8. Symetrical GLCM matrix calculation

2.3.2. Contrast feature

Contrast is the variety of the gray values between the pixel and its referencing neighbor, the formula for contrast is written in (5);

$$Contrast = \sum_{n=0}^{S-1} n^2 \{\sum_{i=0}^{S-1} \sum_{j=0}^{S-1} \boldsymbol{G_d(i,j)}\} \quad (5)$$

where $n=|i-j|$ and $\boldsymbol{G_d(i,j)}$ denotes the pixel values.

2.4. Local binary pattern (LBP)

The LBP Operator proposed by Ojala et al. [20] characterizes the spatial structure of a local image patch by thresholding the differences between the pixel value of the central point and those of its neighbors [21], the condition for the threshold is shown in (6), considering only the signs to form a binary pattern. The Figure 9 is the illustration of LBP steps, a sample of original image with 3x3 patches is shown in Figure 9(a), and the resulting decimal value of the generated binary pattern is shown in Figure 9(b) then the label multiplied with the weight shown in Figure 9(c), the result is used to label the given pixel. The steps of how a local binary pattern result is shown in Figure 10.

$$if\{C>N=0,\ C<N=1\} \tag{6}$$

C denotes the center pixel value and *N* is the neighboring pixel value. The formula for the LBP calculation is written in (7);

$$LBP = \sum_{i=0}^{b-1} p(i) \,.\, w(i) \tag{7}$$

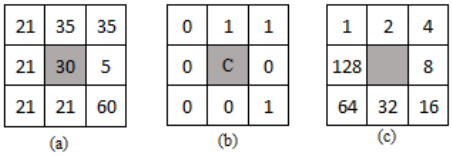

Figure 9. Illustration of LBP steps, (a) the original image, (b) thresholded image, (c) is the weight of the LBP

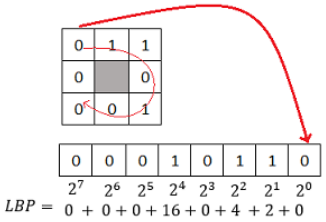

Figure 10. Sorting process of Binary Matrix in LBP

2.4.1. Mean local binary pattern (mean-LBP)

The features produced by GLCM and LBP are not in the same form, the GLCM produce a single-valued feature while the LBP produce a matrix feature. to integrate these two different features, the Mean-LBP is used to transform the Matrix values from LBP into a single-valued feature. This process is needed for classification because the features must be uniform. The formula for the mean LBP is shown in (8);

$$Mean\ LBP = \frac{\sum_{x=0}^{x+1}\sum_{y=0}^{y+1} f(x,y)}{p} \tag{8}$$

where *x, y* is the coordinates for the processed image, and *p* is the total size of the image in pixel.

2.5. kNN classifier

The k Nearest Neighbor Classifier is algorithm that classifies the object based on the distance to the training examples [22]. Training process for kNN classifier consists of only storing the feature vectors and the label of the training images, the computational time will depend on how much the training examples (feature) selected. The classification process is to pick the nearest with the predefined number of *k* (neighbor). Euclidean distance is commonly used to classify the object based on its *k*-nearest neighbor [23]. *k* must be an odd number to avoid ambiguity. Determining the closest vector can be done using the Euclidean distance formula as shown in (9),

$$d(x,y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \tag{9}$$

d(x,y) is the closest vector, *x* denotes the training and *y* denotes the testing, and *i* is the number of feature used.

3. RESULTS AND DISCUSSION

Referring to research method described in section 2, we carried an experiment to evaluate the performance of the features combined to perform on two road condition "shadowed" and "non-shadowed", the pseudo code for this experiment is shown in Figure 11. In this experiment Python programming language version 3.7 is used, the system specification used is Intel Core i7 laptop with 8 GB of ram. We use library for this experiment, the GLCM Extraction is using 'skimage-feature' and for the basic image manipulation we used 'OpenCV-python' version 4.1. The source of the dataset used in this research is taken from the kitti road data set [24], consist of 200 training images and 200 testing images. The images are categorized into two road scenes shadowed and non-shadowed. First, we do a feature extraction on two different sets of data with

the categories "shadowed" and "non-shadowed", containing 100 image each set. The second step is the process of classification using kNN, the neighboring value used for the kNN classification is 3. After the classification results are obtained, an evaluation of the results will be carried out and explained in the following sub-section.

```
Data : RGB Road Images[Array]
Result : Prediction of the road surface type
Initialization : Generate model for kNN(Feature Extraction), Index

Do
    ROI = ROI Extraction of Data[Index];
    Grayscale Conversion of ROI;
    GLCM Feature Extraction(Energy and Contrast) of ROI;
    Mean-LBP Feature Extraction of ROI
    Result = Prediction from kNN Classifier(Input : Energy, Contrast, Mean-LBP);
    If Result is 'Road' Then
        Write "road" on Data[Index];
    Else
        Write "non-road" on Data[Index];
    End
    Display window with Data[Index];
While not at the end of Data;
End
```

Figure 11. Pseudo code for the experiment

3.1. Performance evaluation

To evaluate the performance of GLCM + Mean-LBP, we compare between GLCM with two features and the proposed method, which is GLCM with two features combined with mean-LBP feature. The neighboring value used for kNN is 1. The classification will be carried out twice, according to the number of conditions being evaluated, which is shadow and non-shadowed. The differences of these conditions are shown in Figure 12. To calculate the average result of the classification, we used (10) and the result is shown in Table 1. Referring to the data from Table 2, the proposed method is able to provide better average accuracy on shadowed and non-shadowed condition.

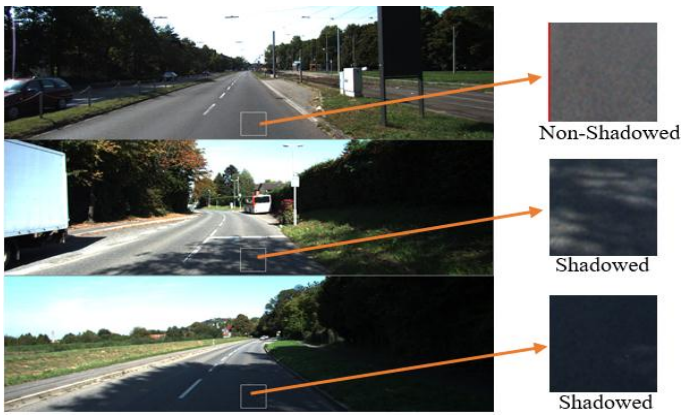

Figure 12. Roi extraction samples

Table 1. Quantitative analysis result

Method	Accuracy (%)
GLCM (5 features)+Color feature+ANN [25]	97%
GLCM (4 features)+kNN[6]	89%
GLCM (2 features)+Mean-LBP+kNN	**98%**

Table 2. Classification result

Method	Road condition	Number of data	Correct classification	Average result	Time consumed (second)
GLCM (2 features)+Mean-LBP	Shadowed	100	97	**98%**	7,017
	Non-shadowed	100	100		
GLCM (2 features)	Shadowed	100	85	87.5%	5,023
	Non-shadowed	100	90		

3.2. Visual result

The visual result is needed to provide an example of how the proposed method will solve the problems. The features which are energy, contrast, and mean-LBP is used to classify the "road" or "non-road" images and are shown in Figure 12. The proposed method is able to detect the shadowed on non-shadowed road surface using the features mentioned. Referring to Figure 13, the proposed method is able to detect the road surface with different conditions such as darkened road, road covered by shade, and road with low light. The Table 1 shown that the combination of algorithm affects the result of the classification, the combination between GLCM and LBP is producing a better accuracy, and the result from Table 2 shown that combination made but slower computational time, while the GLCM with more feature produces lower accuracy with better computational time.

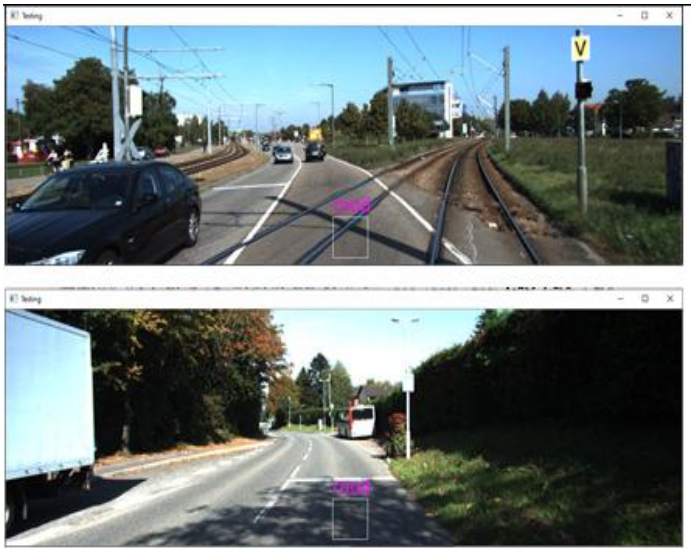

Figure 13. The visual result of the classification

3.3. Quantitative analysis

The proposed method is able to detect and classify the road or non-road surface images, at this stage, computational time and accuracy need to be analyzed. The computational time is tested using 100 images, consist of 50 road images and 50 non-road images, and there are 2 different schemes. The accuracy is obtained from (10). To understand the effect of each combination used, the number of processed image rates per second/frame per second needs to be analyzed using (11) and the result for the analysis is shown in Table 2.

$$Accuracy = \frac{TP}{N} * 100 \quad (10)$$

TP denotes true positive detection; the true positive is taken from the classification with the correct detection where the prediction of the system match the visual condition and N denotes the number of data testing.

$$PCs = \frac{N}{TC} \quad (11)$$

$$TC = ST - EDT \quad (12)$$

PCs denotes the processed image, while *N* denotes the number of data testing, and *TC* denotes the time consumed, and the formula for the time consumed is written in (12). *ST* denotes the start time of the program execution and *EDT* is the end time of the program execution.

The results in Table 1 show that in the classification process in the study using the GLCM with five features (entropy, energy, contrast, correlation, and local homogeneity) combined with color features yield an accuracy of 97%, the data processed is a video with 30 fps and the classification is using ANN with varying iteration of 150 to 700. Our proposed method uses GLCM with two features (energy and contrast) combined with Mean-LBP to produce better results as much as 1% in image data with a resolution of 1242x375 pixels. This means that the proposed method can provide a better result by using fewer features thus maintaining lower computational load for the system.

4 CONCLUSION

This research aims to find the most discriminant and minimum number of features to be used in road classification, by comparing a combination of two different feature sets and one study with an approach that uses GLCM, the best results are obtained, where the proposed method can produce better average accuracy results in shadowed and non-shadowed road conditions. Referring to the result in section 3, it is shown that the mean-LBP feature is slowing down the classification process but produces a better accuracy which is 98%. In future research, we believe that the other type of LBP can be tested to improve the accuracy or the computational time for this approach.

REFERENCES

[1] E. Altendorf and F. Flemisch, "Prediction of driving behavior in cooperative guidance and control: A first game-theoretic," *Kognitive Systeme*, vol. 2, 2017.

[2] Y. Liu, W. Xu, A. M. Dobaie, and Y. Zhuang, "Autonomous road detection and modeling for UGVs using vision-laser data fusion," *Neurocomputing*, vol. 275, pp. 2752-2761, 2018.

[3] K. I. Min, J. S. Oh, and B. W. Kim, "Traffic sign extract and recognition on unmanned vehicle using image processing based on support vector machine," *Int. Conf. on Control, Automation and Systems*, pp. 750-753, 2011.

[4] H. C. Moon, J. H. Kim, J. C. Lee, and D. M. Lee, "Development of unmanned ground vehicles available of urban drive," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pp. 786-790, 2009.

[5] V. Slavkovikj, S. Verstockt, W. De Neve, S. Van Hoecke, and R. Van De Walle, "Image-based road type classification," *Proceedings-International Conference on Pattern Recognition*, 2014.

[6] S. Marianingsih, F. Utaminingrum, and F. Abdurrachman, "Road surface types classification using combination of K-nearest neighbor and naïve bayes based on GLCM," *Int. J. of Adv. in Soft Comp. & Its Appl.*, vol. 11, no. 2, pp. 1-13, 2019.

[7] S. Verstockt, V. Slavkovikj, P. De Potter, J. Slowack, and R. Van De Walle, "Multi-modal bike sensing for automatic geo-annotation geo-annotation of road/terrain type by participatory bike-sensing," *2013 International Conference on Signal Processing and Multimedia Applications (SIGMAP)*, pp. 39-49, 2013.

[8] M. Pratiwi, Alexander, J. Harefa, and S. Nanda, "Mammograms classification using gray-level co-occurrence matrix and radial basis function neural network," *Procedia Comput. Sci.*, vol. 59, no. Iccsci, pp. 83-91, 2015.

[9] C. Fern and I. Garc, "Surface classification for road distress," *Springer Berlin Heidelberg*, pp. 600-607, 2012.

[10] H. Kuang, Y. Ding, R. Li, and X. Liu, "Defect detection of bamboo strips based on LBP and GLCM features by using SVM classifier," *Proc. 30th Chinese Control Decis. Conf.*, pp. 3341-3345, 2018.

[11] S. Zhang, "Cost-sensitive KNN classification," *Neurocomputing*, 2019.

[12] D. Nofriansyah and H. Freizello, "Python application: Visual approach of hopfield discrete method for hiragana images recognition," *Bull. Electr. Eng. Informatics*, vol. 7, no. 4, pp. 609-614, 2018.

[13] F. Utaminingrum, R. Primaswara, and Y. Arum Sari, "Image processing for rapidly eye detection based on robust haar sliding window," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 2, pp. 823-830, 2017.

[14] T. Kumar and K. Verma, "A theory based on conversion of RGB image to gray image," *Int. J. Comput. Appl.*, vol. 7, no. 2, pp. 5-12, 2010.

[15] C. Di Ruberto, L. Putzu, and G. Rodriguez, "Fast and accurate computation of orthogonal moments for texture analysis," *Pattern Recognit.*, vol. 83, pp. 498-510, 2018.

[16] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features," *Syst. Man Cybern.*, vol. 3, no. 6, pp. 610-621, 1973.

[17] L. Wei and D. Hong-ying, "Real-time road congestion detection based on image texture analysis," *Procedia Engineering*, vol. 137, pp. 196-201, 2016.

[18] R. Tao, C. Jie, C. Biao, and C. Liu, "GLCM and fuzzy clustering for ocean features classification," *2010 Int. Conf. Mach. Vis. Human-Machine Interface, MVHI 2010*, pp. 538-540, 2010.

[19] K. N. N. Hlaing and A. K. Gopalakrishnan, "Myanmar paper currency recognition using GLCM and k-NN," *2016 2nd Asian Conf. Def. Technol. ACDT 2016*, pp. 67-72, 2016.

[20] T. Ojala, M. Pietikäinen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern analysis and machine intelligence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971-987, 2002.

[21] Y. Lei, X. Zhao, G. Wang, K. Yu, and W. Guo, "A novel approach for cirrhosis recognition via improved LBP algorithm and dictionary learning," *Biomed. Signal Process. Control*, vol. 38, pp. 281-292, 2017.

[22] J. Huang, Y. Wei, J. Yi, and M. Liu, "An improved knn based on class contribution and feature weighting," *Proc-10th Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2018*, pp. 313-316, 2018.

[23] J. Li and B. L. Lu, "An adaptive image Euclidean distance," *Pattern Recognit.*, vol. 42, no. 3, pp. 349-357, 2009.

[24] Andreas G. et al., "Road/lane detection evaluation 2013," *Karlsruhe Institute of Technology*, 2013. [Online]. Available at: http://cvlibs.net/datasets/kitti/eval_road.php.

[25] I. Tang and T. P. Breckon, "Automatic road environment classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 476-484, 2011.