APST

Asia-Pacific Journal of Science and Technology

https://www.tci-thaijo.org/index.php/APST/index

Published by the Research and Technology Transfer Affairs Division,
Khon Kaen University, Thailand

Rotation invariant binary gradient contour for geographic object-based image analysis

Sarun Apichontrakul[1,2,*] and Rasamee Suwanweerakamtorn[1,2]

[1]Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand
[2]Geo-Informatics Centre for Development of Northeast Thailand, Khon Kaen University, Khon Kaen, Thailand
[*]Corresponding author: sarunap@kku.ac.th

Received 11 July 2021
Revised 28 December 2021
Accepted 13 February 2022

Abstract

This study proposed a modified rotation invariant texture descriptor based on Binary Gradient Contour (BGC1) for land cover classification under Geographic Object-based Image Analysis (GEOBIA). The modified texture descriptor's performance was tested with 6 machine learning algorithms and a high-resolution Theos satellite image of the area of the city of Khon Kaen. The satellite image was segmented into 9,929 homogeneous land cover objects, of which, 5,417 objects were labeled as one of the ten land cover classes and validated using the 5-Fold cross validation method. The overall accuracy, the individual class F1-Scores, and the computational efficiency of the classification models, which used rotation invariant $\mathrm{BGC1}_{\mathrm{Rot}}$, were compared with models, which had used GLCM, LBP variations, and the original BGC1. The results showed that among the 6 classifiers, Random Forest (RF) had produced the best overall accuracy. The model with RF and $\mathrm{BGC1}_{\mathrm{Rot}}$ had produced the best overall accuracy at 84.863%, which was significantly higher than the original BGC1, and was the highest F1-score for 6 out of 10 investigated land cover classes. During the feature extraction step, the more computationally efficient $\mathrm{BGC1}_{\mathrm{Rot}}$ was also found to process 4.48 times faster than GLCM. When compared to the widely accepted Uniform $\mathrm{LBP}_{\mathrm{Uni}}$, $\mathrm{BGC1}_{\mathrm{Rot}}$ provided an overall accuracy and an average F1-Score that were slightly better with a similar computation time. Thus, the proposed $\mathrm{BGC1}_{\mathrm{Rot}}$ has been proven to be an effective texture descriptor for GEOBIA based on overall and individual class accuracy, as well as on computational efficiency.

Keywords: BGC, LBP, GEOBIA, HEP, Land cover classification, Pattern recognition, Remote sensing, Satellite image analysis, Texture descriptor, Theos

1. Introduction

In recent years, Geographic Object-based Image Analysis (GEOBIA) or Object-based image classification has gained significant popularity for use in the land cover classification of high resolution remote sensing data [1,2]. Many studies have reported improved classification performance when using GEOBIA compared to the traditional pixel-based classification [3-6]. GEOBIA imitates how humans analyze visual information by first partitioning an image into objects and then by using the distinct features or attributes of the objects to identify their classes [6]. In contrast to the pixel-based method, which mainly utilizes spectral features to classify the individual pixels, GEOBIA incorporates the sizes, shapes, textures, patterns, and contexts of objects to help distinguish land cover types [4,5]. The most widely cited texture descriptor in remote sensing has been derived from Haralick's gray level co-occurrence matrix (GLCM) [7]. However, since the conception of GLCM in 1973, numerous texture descriptors have been developed in the field of computer vision, but most have not been sufficiently explored in remote sensing applications; such as the Local Binary Pattern (LBP) [8] and the Binary Gradient Contour (BGC) [9], both of which belong to the Histogram of Equivalent Pattern (HEP) texture descriptor family [10].

There have already been a few studies, which have utilized HEP for satellite image analysis [11,12]. For example, Aguilar et al. demonstrated in their study that GEOBIA classification accuracy could be improved by incorporating HEP as input features [12]. However, no studies have applied the aspect of the rotation invariant HEP on the arbitrary shapes of GEOBIA's segmented objects. An effective texture descriptor for remote sensing data should be rotation invariant because land cover on the Earth's surface can be in any arbitrary direction. If a texture descriptor is not rotation invariant, an object with a pattern running in an East-West direction could have a different texture signature from an object with the same pattern that is running in a North-South direction. Although GLCM has proven its effectiveness in image analysis for over 40 years since its inception, it was not initially designed to support rotation invariants. In addition, GLCM is not gray-scale invariant, and is much more computationally expensive when compared to the more robust gray-scale invariants of LBP and BGC. The Gray-scale invariant is an important characteristic of a texture descriptor for remote sensing data because the descriptor's performance should not be affected by the brightness or the contrast of the image. While there have been many applications of rotation invariant LBP in computer vision, such as the acknowledged Uniform $\mathrm{LBP_{Uni}}$ [13], BGC does not yet have its own rotational invariant version despite its pattern recognition superiority, especially the single-loop version - BGC1 [9,12]. Moreover, [9] has shown that BGC1 outperformed LBP when tested with benchmark datasets. Hence, similar results could be expected for variations of their rotation invariants. Therefore, we have proposed a modified rotation invariant version of BGC1 for GEOBIA, which from now on will be referred to as $\mathrm{BGC1_{Rot}}$. This study examined and compared the performance of the rotation invariant $\mathrm{BGC_{Rot}}$ against GLCM and $\mathrm{LBP_{Uni}}$ for GEOBIA land cover classification.

1.1 Background on the investigated texture descriptors

1.1.1 Gray-level co-occurrence matrix (GLCM)

GLCM was first proposed by Haralick and Shanmugam in 1973 [7] and is still being applied in various image analysis applications, including remote sensing. GLCM characterizes texture by using the second order statistics of an image pixel and its neighbors to depict the occurrence frequency of the different patterns in an image. GLCM is created by comparing the value of a pixel with its immediate neighbor. The neighbor can be specified by the users, but the default neighbor is the pixel's adjacent neighbor to the right. GLCM can be viewed as a mapping table, which keeps track of the frequency of each unique pixel pair, in which the number of rows and columns of GLCM are the number of the gray scale level of the original image. However, by directly mapping gray level intensity to a co-occurrence matrix means that the algorithm is not gray-scale invariant, and by specifying the fixed direction of a neighbor, GLCM is not rotation invariant. The rotation invariant problem is partially solved by summing the results of at least four non-parallel directions to create the isotropic GLCM. However, it is not considered to fully be a rotation invariant because the same sum value could be derived from different patterns. From the matrices created by GLCM and its variations, one can extract various second order statistics to represent patterns of texture, such as Entropy, Contrast, Homogeneity (Inverse Different Moment and Angular Second Moment), Dissimilarity, Correlation, and Variance.

1.1.2 Local Binary Pattern (LBP)

LBP was first introduced by Ojala, Pietikäinen, and Harwood (1994) [8], and since then, it has become one of the most popular texture descriptors in computer vision and in the pattern recognition community. It was inspired by Wang and He's Texture Spectrum [14]. Similar to GLCM, LBP functions by analyzing the relationship between a pixel's value and its neighbors' values. Usually, a 3x3 kernel of 8 immediate neighbor pixels are used to describe a local pattern. A larger kernel size could be used in case of patterns that are more spread out, but it increases quantization level (dimensions) and computational costs. LBP uses the binary thresholding function (Eq. 1), in which the binary encoded value can be calculated by Equation 2. As the equations suggest, LBP is based on binary code for characterizing an image pattern. A 3x3 kernel moves along each pixel in the image, and an 8-bit pattern code is derived, which can have a possible value that ranges from 0-255 (Figure 1A, B and Figure 2). A 256-bin histogram is created to keep track of the occurrence frequency of each 8-bit value. To determine if two images have the same pattern, the LBP histograms are simply compared and examined to determine if they match. This is the main trait of the HEP family of texture descriptors.

$$\xi(\mathrm{i}) = \begin{cases} 0, & \text{if } \mathrm{I_n} < \mathrm{I_c} \\ 1, & \text{Otherwise} \end{cases} \tag{1}$$

$$\mathrm{LBP}_{3\mathrm{x}3} = \sum_{\mathrm{n}=0}^{7} \xi(\mathrm{I}_\mathrm{n}, \mathrm{I}_\mathrm{c}) \times 2^\mathrm{n} \tag{2}$$

in which I_c is the center pixel value of a 3x3 kernel and I_n is the pixel value of neighborhood pixel value at location n (Figure 1 A)

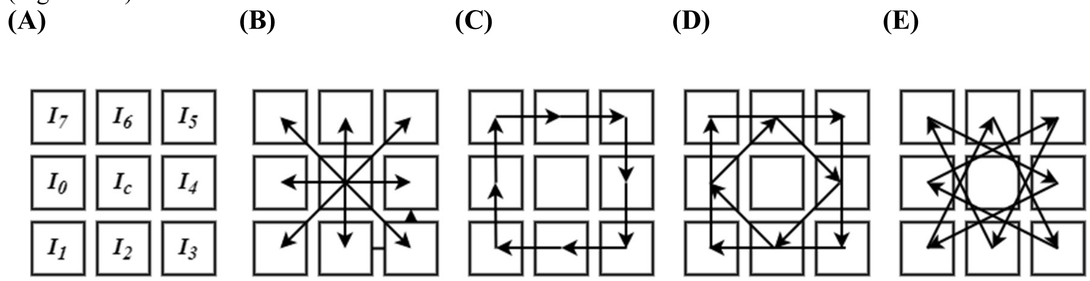


Figure 1 How LBP and BGC traverse a 3 x 3 kernel to compute an 8-bit binary code. (A) A 3x3 grayscale kernel, (B) LBP (C) a single-loop - BGC1, (D) a double-loop - BGC2, and (E) a triple-loop versions - BGC3 [9].

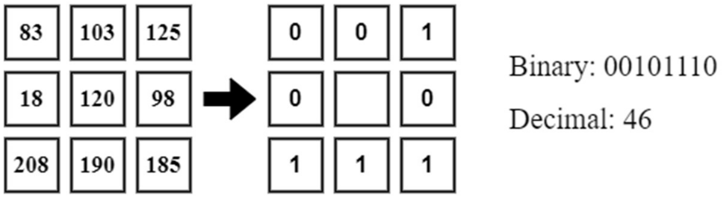


Figure 2 An Example of how LBP encodes a 3 x 3 pattern into an 8-bit number.

The original LBP is gray-scale invariant, since the intensity of value does not matter when it is encoded into binary. It is, however, not rotation invariant, and the 256 bins of the histogram means that there are 256 dimensions of feature space that must be analyzed. In their later works, Ojala et al. proposed improved versions of LBP so that by using Uniform LBP, the texture descriptor could be robust and could cope with changes in spatial scale and in the reduction of dimension, as well as with the rotation invariant [13]. A local binary is considered to be uniform if there are 2 or less than 2 bit-wise transitions in the binary string. For example, 00000000, 00000111 and 00111000 are uniform, but 10001011 (4 transitions) is not. According to Ojala et al., the uniform patterns are usually made up of approximately 90% of 3x3 patterns found in any image, and they often represent the important characteristics of the objects, such as the edges, corners, spots, and the flat areas, etc. [13]. The dimension of LBP was reduced by discarding the patterns that were not uniform. In addition, LBP becomes rotation-invariant by using a circular bit-wise right shift on a string of LBP binary code until it reaches the minimum value (i.e., 10000011 or 00111000 will become 00000111). This operation, in turn, makes LBP rotational invariant because all uniform patterns are rotated to their minimum state. By solely considering the uniform patterns in the analysis, the histogram's dimension could be reduced from 256 to 58. Subsequently, by performing the minimum circular bit-wise right shift, the dimension can be further reduced to 9 dimensions (36 dimensions if including the non-uniform lbp). This dimension reduction significantly decreases the computation, while keeping the loss of information to a minimum. Ojala et al. also proposed multiresolution analysis for LBP, in case the 3x3 kernel is not sufficient to represent a more complex or spread-out pattern. Users can combine patterns from a different kernel radius to produce a larger quantization of binary numbers. However, unlike fabric patterns, remote sensing data rarely displays symmetric or repeating patterns. Moreover, Tobler's law states that nearer objects (pixels) are more related to the point of interest than further objects. Thus, a 3x3 kernel is sufficient for remote sensing data, especially for high resolution satellite images. Furthermore, if pixel contrast is imperative, users can calculate the local variance of pixel values in each kernel to compliment LBP [13].

There have been many variations of, such as the Improved LBP (ILBP), which uses an average gray value of local 3x3 as the threshold value, instead of the center pixel value. Sign LBP (CLBP_S) uses -1, 1 instead of 0, 1 for coding. Magnitude LBP (CLBP_M) compares the averaged absolute differences between neighbors and the center pixel value against each neighbor's own difference. The Center LBP (CLBP_C) uses the average gray value of the whole image as the threshold; while the Completed LBP (CLBP_S_MxC) uses a Combination of CLBP_S, CLBP_M, and CLBP_C. Aguilar et al. (2016) [12] compared the performance of 26 non-parametric texture descriptors of the HEP family, which included the mentioned LBP and its variations, against GLCM textures (entropy, correlation, standard deviation, contrast, and mean) for urban area classification of GeoEye-1 imagery. The top five performers were CLBP_S_MxC, CLBP_MxC, ILBP, LBP, and BGC1, all of which produced significantly better accuracy and less computational time than GLCM, even without using rotation invariant versions of the mentioned texture descriptors.

1.1.3 Binary Gradient Contour (BGC)

BGC was first proposed by Fernandez, Alvarez and Bianconi [9] to improve upon the foundation of LBP by using binary encoding from 3x3 local patterns to differentiate the texture patterns. Three versions of BGC were proposed [single loop (BGC1), double loop (BGC2), and triple loop (BGC3) (Figure 1 C, D, E). The mathematical equations for each BGC version are shown in Equation 3-5 demonstrates how BGC can encode a pattern into an 8-bit value. From the equations, it can be observed that BGC was designed to have a minimum value of 0 and a maximum value of 254 for BGC1 and BGC 3, while BGC2 has a maximum value of 224. The subtraction of 1 in Eq.3, however, prevents the encoded binary of BGC1 from being rotation invariant by using minimum bit-wise rotation, which led to its modification for this study. Details of the modification are discussed in Section 3.4.3 Texture feature extraction. According to [9], BGC1 provided the best performance among the three versions, and in most cases, gave better accuracy than LBP, despite the algorithm discarding the central pixel value. The authors argued that by removing the central value, the algorithm with the lesser dimension and smaller number of traversing angles led to a more balanced histogram partition and was theoretically more efficient than LBP.

$$BGC1_{3x3} = \sum_{n=0}^{7} \xi\left(I_n, I_{(n+1)mod8}\right) \times 2^n - 1 \tag{3}$$

$$BGC2_{3x3} = 15 \times \sum_{n=0}^{3} \xi\left(I_{2n}, I_{2(n+1)mod8}\right) \times 2^n + \sum_{n=0}^{3} \xi\left(I_{2n+1}, I_{2(n+3)mod8}\right) \times 2^n - 16 \tag{4}$$

$$BGC3_{3x3} = \sum_{n=0}^{7} \xi\left(I_{3n\ mod8}, I_{3(n+1)mod8}\right) \times 2^n - 1 \tag{5}$$

in which $\xi(x,y) = \begin{cases} 0, & if \quad x < y \\ 1, & if \quad x \geq y \end{cases}$

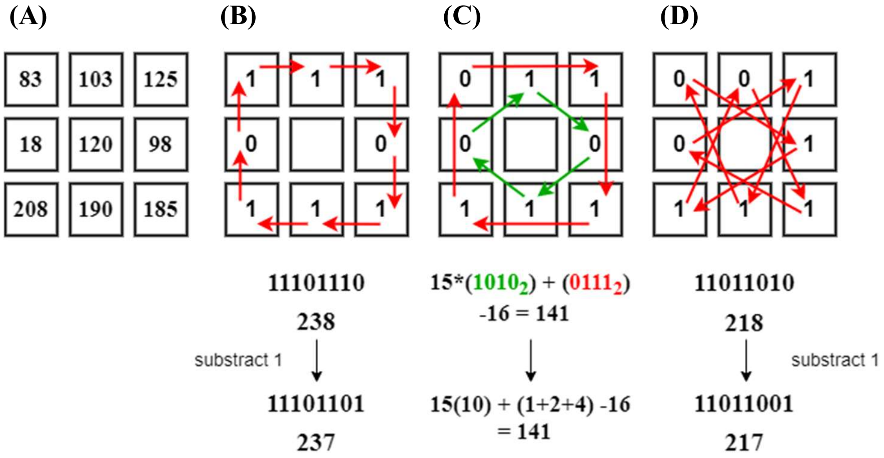


Figure 3 Examples of how the 8-bit numbers are generated from BGC (Equation 3-5) by visualizing the close-loop gradient directions of (A) a Sample 3x3 grayscale kernel to compute for (B) BGC1, (C) BGC2, and (D) BGC3.

2. Materials and methods

The workflow of this study is shown in Figure 4. For ease of reference, each important step has been marked with a number corresponding to the subsection numbers in this section.

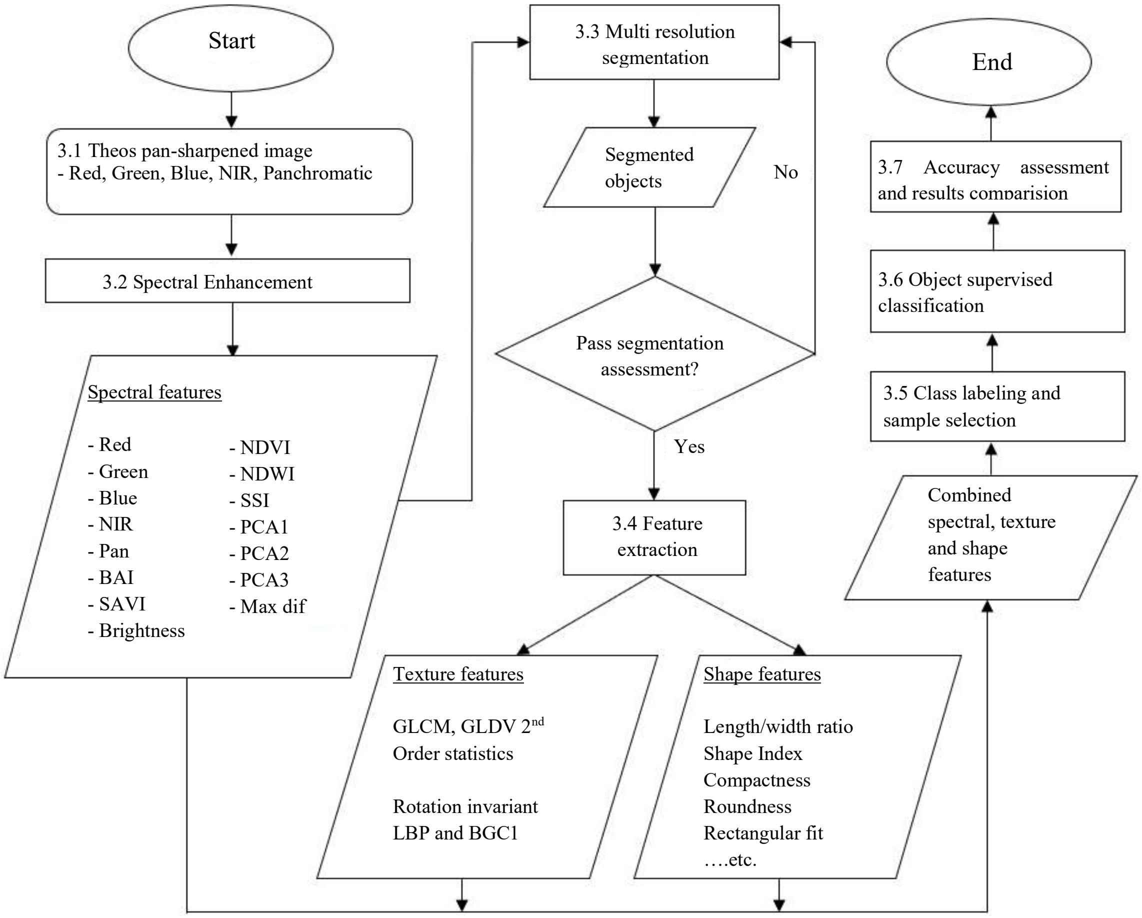

Start
3.1 Theos pan-sharpened image
- Red, Green, Blue, NIR, Panchromatic
3.2 Spectral Enhancement
Spectral features
- Red
- Green
- Blue
- NIR
- Pan
- BAI
- SAVI
- Brightness
- NDVI
- NDWI
- SSI
- PCA1
- PCA2
- PCA3
- Max dif
3.3 Multi resolution segmentation
Segmented objects
No
Pass segmentation assessment?
Yes
3.4 Feature extraction
Texture features
GLCM, GLDV 2nd Order statistics
Rotation invariant LBP and BGC1
Shape features
Length/width ratio
Shape Index
Compactness
Roundness
Rectangular fit
….etc.
Combined spectral, texture and shape features
3.5 Class labeling and sample selection
3.6 Object supervised classification
3.7 Accuracy assessment and results comparision
End

Figure 4 The workflow of this study.

2.1 Study area and satellite image

This study used a geometrically corrected pansharpened Theos satellite image of the city of Khon Kaen in Khon Kaen Province, which had been provided by the Geo-Informatics and Space Technology Development Agency (GISTDA). The image was acquired on 2-May-2015. The specification of the dataset is shown in Table 1. Similar to other provinces in Thailand, agriculture is the leading economic sector found in Khon Kaen. Rice paddy fields and farmlands account for most of the province's area. The city of Khon Kaen is the capital of Khon Kaen Province and is located in the middle of the Northeastern region between latitude of 16.398° - 16.483° North and longitude of 102.802° - 102.876° East. Khon Kaen is a fast-growing city with an international airport and several regional government offices. The studied area (satellite image coverage) covered roughly 350 km2 (18 x 19.44 km^2) or 35,000 ha (Figure 5). The blue highlighted subdistricts in Figure 5 have a combined population of 278,397, with an average population density of 816 per km2. The area is relatively flat with elevations that vary from between 96 to 196 meters above the mean sea level. In the area, land covers range from high-rise condominiums, shopping malls, residential areas, housing estate developments, industrial areas, government complexes, and university campuses to rice paddy fields, farmlands, golf courses, solar farms, perennial plantations, lakes, rivers, and ponds. The rooftops of buildings and houses vary in types and colors, such as sheet metal, zinc, concrete, and straw, plus various colors of rooftop tiles and slates. Major roads are made of asphalt and concrete, while dirt roads can be seen throughout the rural areas along crop fields. Mittraphap Road, which is a highway connecting Bangkok and the Northeastern region up to Thai-Laos Friendship bridge, cuts through the middle of Khon Kaen city. The city also has a ring road for bypass with a diameter of about 14 km. Most of the urban growth is occurring within and along this ring. The diverse land cover classes presented in the area made the satellite image ideal for validating the performance of the classification algorithms and texture descriptors.

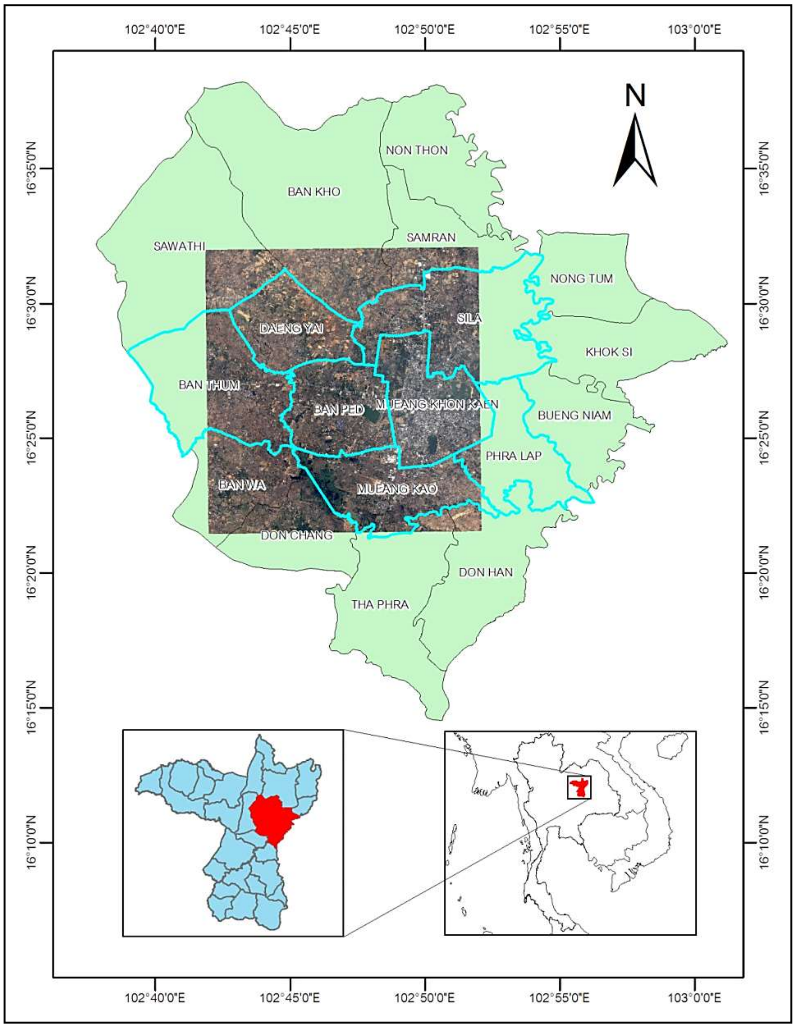


Figure 5 A satellite image of the study area, the Khon Kaen metropolitan area, and the nearby sub-districts in the Mueang Khon Kaen District of Khon Kaen Province.

Table 1 The Theos satellite image data used in this study.

Band	Spectral Ranges (nm)	Spatial Resolutions (m)	Radiometric Resolutions	Gain Values
0. Panchromatic	450-900	2	8-bit	1.52253
1. Red	620-690	15	8-bit	1.71019
2. Green	530-600	15	8-bit	1.50071
3. Blue	450-520	15	8-bit	1.46853
4. NIR	770-900	15	8-bit	1.67119

2.2 Spectral enhancement

Principal Component Analysis (PCA) and 5 spectral indices were calculated from the pansharpened image. According to Villa (2007) [15], the best index for detecting impervious surface is their proposed Soil and Vegetation Index (SVI). However, similar to other high performance impervious surface indices, SVI requires Short-Wave Infrared (SWIR) band, which is lacking in the Theos sensor. Therefore, this study was limited to those spectral indices, which only use the four traditional multispectral bands of Red, Green, Blue, and NIR. These consisted of the Normalized Difference Vegetation Index (NDVI, Equation 6) [16], the Normalized

Difference Water Index (Green) ($\mathrm{NDWI_{Green}}$, Equation 7) [17], the Spectral Shape Index (SSI, Equation 8) [18], the Soil Adjusted Vegetation Index (SAVI, Equation 9) [19], and the Built-up Area Index (BAI, Equation 10) [20].

$$\mathrm{NDVI} = \frac{\mathrm{NIR-Red}}{\mathrm{NIR+Red}} \tag{6}$$

$$\mathrm{NDWI_{Green}} = \frac{\mathrm{Green-NIR}}{\mathrm{Green+NIR}} \tag{7}$$

$$\mathrm{SSI} = |\mathrm{Red} + \mathrm{Blue} - 2\ \mathrm{x}\ \mathrm{Green}| \tag{8}$$

$$SAVI = \frac{(1+L)(NIR-RED)}{(NIR+RED+L)} \tag{9}$$

where L is a soil brightness adjustment factor

$$\mathrm{BAI} = \frac{(\mathrm{Blue} - \ \mathrm{NIR})}{(\mathrm{Blue} + \mathrm{NIR})} \tag{10}$$

NDVI has been the standard index for detecting vegetation, or the lack thereof. $\mathrm{NDWI_{Green}}$ is a substitute for $\mathrm{NDWI_{SWIR}}$ when seeking to detect water bodies. SSI is used to detect water and shadows. SAVI is an improved version of NDVI which can be used in areas with sparse vegetation (less than 40%) to remove errors caused by soil reflectance, which can potentially affect the NDVI values by up to 20% [19]. The ‘L’ value in Equation 9 is a correction factor which ranges from 0 for very high vegetation cover to 1 for very low vegetation cover. SAVI has the same value as NDVI when ‘L’ approaches zero. The adjustment factor of L is usually found by trial and error until a factor that gives equal vegetation index results for the dark and light soils is found. In most cases, when ‘L’ = 0.5, it is sufficient to minimize the variations in soil brightness, thus an ‘L’ value of 0.5 was used for this study. BAI is useful in detecting asphalt and concrete surfaces. According to [21], BAI was found to be very robust and had even performed well with blurred or noisy images.

2.3 Image segmentation

The first step of GEOBIA is the image segmentation process. The eCognition Developer software was used to perform multiresolution image segmentation of the 5-band pan-sharpened satellite image plus 8 additional bands from 3 PCAs and 5 indices from the step of spectral enhancement. Some studies have developed an algorithm to automatically generate the optimal segmentation parameters and scale levels [22], but others have argued that there is no image analysis algorithm that is comparable to human interpretation, and that segmentation errors, which have been assessed by the human eye, provide the best results [3,23]. Since there is no rule of thumb or standard algorithm that can be utilized to define the scale numbers and other segmentation parameters, a trial-and-error method was used to obtain the best set of segmentation parameters. Developing an algorithm, which can optimize the segmentation parameters, is out of the scope of this study.

The image was segmented with 10 different scale levels starting from 25 to 250 in increments of 25. The unitless scale level value indicates the heterogeneity threshold that a segmented object could have. Usually, the larger the scale level number, the larger the segmented objects are. The scale level of the multiresolution segmentation algorithm in eCognition is an arbitrary integer, starting from 1, and is calculated based on the values of the input bands. After manually assessing the results of each segment scale level by eye, the best scale level was selected for this study. The main criteria, which was used to determine the appropriate scale level, was how well the objects represented the types of homogenous land cover. The largest scale level, which could partition the image into the highest number of homogenous land cover objects, was selected. In addition to the scale level, different values for weight, shape, color, and compactness, which were used to measure the objects’ heterogeneity, were also tested.

2.4 Object feature extraction

After the image was segmented into homogenous land cover objects, each object’s attributes or features were extracted. The eCognition software was used to calculate the primary features to the segmented objects, which were then exported as a vector shape file for further analysis. This study extracted 3 types of object features, spectral features, shape features, and texture features. This study aimed to only assess different texture features.

Therefore, we used the same set of spectral and shape features with each individual texture descriptor for all classification models.

2.4.1 Spectral features

Spectral Features were the original satellite image band values, PCA bands, the indices calculated in the spectral enhancement step, brightness, and the maximum difference. Some of the features had already been used as inputs for the segmentation process. The 15 spectral features used in this study are shown in Table 2.

Table 2 A list of Spectral Features used for this study.

Names	Full names	Ranges	Descriptions
Blue	Mean blue band value	[0,255]	The Mean Blue band value of an object
Green	Mean green band value	[0,255]	The Mean Green band value of an object
Red	Mean red band value	[0,255]	The Mean Red band value of an object
NIR	Mean NIR band value	[0,255]	The Mean Near Infrared band value of an object
Pan	Mean panchromatic band value	[0,255]	The Mean Panchromatic band value of an object
BAI	Mean BAI value	[-1,1]	The Built-up Area Index is useful in detecting asphalt and concrete surfaces
NDVI	Mean NDVI value	[-1,1]	The Normalized Difference Vegetation Index is the standard index for detecting vegetation, or the lack thereof.
$\text{NDWI}_{\text{Green}}$	Mean NDWI value	[-1,1]	The Normalized Difference Water Index is used to detect water using Green and Near Infrared bands.
SAVI	Mean SAVI value	[-1,1]	The Soil Adjusted Vegetation Index detects vegetation and helps to remove potential errors caused by soil surface reflectance.
SSI	Mean SSI	[-2,2]	The Spectral Shape Index is useful in distinguishing between pixels of shadows and non-shadows
PCA1	Mean PCA1 band value	[0,1]	PCA1 value is the eigen value of the first eigen vector. Calculated from the Principal Component Analysis of Red, Green, Blue, and NIR Bands
PCA2	Mean PCA2 band value	[0,1]	PCA2 value is the eigen value of the second eigen vector. Calculated from the Principal Component Analysis of Red, Green, Blue, and NIR Bands
PCA3	Mean PCA3 band value	[0,1]	PCA3 value is the eigen value of the third eigen vector. Calculated from the Principal Component Analysis of Red, Green, Blue, and NIR Band
Brightness	Brightness	[0,255]	The Mean Brightness of all the spectral bands
Max_diff	Maximum Difference	[0,1]	Ratio of maximum difference between a pair of band values with the highest difference

2.4.2 Shape features

Shape features are the geometric attributes of an object (i.e., the area, perimeter, and the length to width ratio, etc.). All shape features were calculated using the eCognition software. Irrelevant shape features were screened out. For example, Area, Perimeter, and Main direction were excluded because they generally do not offer meaningful information for the classification process. The 10 shape features chosen for the initial consideration for this study are listed in Table 3.

Table 3 A list of the Shape Features used for this study (see the detailed description in the eCognition Reference book [24]).

Full name	Range	Description
Asymmetry	[0,1]	The relative length of an image object, compared to a regular polygon.
Border Index	[0,∞)	Determines how jagged an image object is by comparing the object's border to a rectangle. [1 = ideal]
Compactness	[0,∞)	Describes how compact an image object is by calculating the product of the length and the width, divided by the number of pixels. [1 = ideal]
Density	[0,∞)	Squares are considered to be the densest shape, while the thinner (or more elongated) an object is, the lower its density. The density is the ration of the number of object's pixels and its approximated radius is determined by using the covariance matrix. [1 = ideal]
Elliptic Fit	[0, 1]	Describes how well an image object fits into an ellipse of similar size and proportions. A zero (0) indicates no fit, while 1 indicates a perfect fit. [1 = complete fitting, 0 = <50% fit.]
Length-Width Ratio	[0,∞)	The smaller number between the ratio of the eigenvalues of the covariance matrix and the ratio of the length to the width of the object's bounding box.
Radius of Smallest Enclosing Ellipse	[0,∞)	The ratio of the radius of the smallest enclosing ellipse to the radius of the original ellipse, which is created with the same number of pixels as the object.
Rectangular Fit	[0,1]	Determines how similar an object is to a rectangle of similar size and proportions. A zero (0) indicates no fit, while 1 indicates a complete fitting. [1 = ideal]
Roundness	[0,∞)	Describes how similar an image object is to an ellipse. It is calculated by the difference of the enclosing ellipse and the enclosed ellipse.
Shape Index	[1,∞)	Describes the smoothness of an image's object border, in which a square is the smoothest object. The smoother the border, the lower its shape index.

2.5 Texture features

In this step, GLCM, the original LBP, the rotation invariant LBPRot, the Uniform LBPUni, the original BGC1, and the proposed modified rotation invariant BGC1Rot texture descriptors were extracted from each segmented object using the panchromatic band of the image. The GLCM matrix and its best five second order statistics were calculated with the eCognition Developer software, while the LBP and BGC1 were calculated in Matlab. The best five second order statistics of GLCM were selected using the Random Forest's feature importance, which ranks the features based on their weights. The five statistics were Homogeneity, Mean, Correlation, Dissimilarity, and Angular 2nd Moment. The derived GLCM statistics were the average values from all 4 directions for the rotation invariant.

The original LBP and BGC1 were calculated for each segmented object by masking each image pixel with the object's id. The minimum bitwise rotation was then performed on the original LBP 8-bit code, lowering the number of feature dimensions from 256 to 36 to create the rotation invariant $\mathrm{LBP_{Rot}}$. The dimension of the $\mathrm{LBP_{Rot}}$ was then further lowered to 9 dimensions by only selecting those dimensions with uniform patterns to create Uniform $\mathrm{LBP_{Uni}}$.[13]. Conversely, BGC1 required a slight modification to the equation (Equation 3) for it to become a rotation invariant descriptor. The equation of the proposed modified BGC1 is shown in Equation 11, which produced values between 1-255 instead of the original values of 0-254 by simply removing the subtraction from Eq.3. The binary codes received from the modified BGC1 could then be minimum bit-wise rotated in the same way as the $\mathrm{LBP_{Rot}}$ so that the dimension could be reduced from 255 to 35 and BGC1 could be turned into the rotation invariant descriptor $\mathrm{BGC1_{Rot}}$.

$$\mathrm{BGC1_{3x3\_mod}} = \sum_{\mathrm{n}=0}^{7} \xi\left(\mathrm{I_n}, \mathrm{I_{(n+1)mod8}}\right) \times 2^{\mathrm{n}} \qquad (11)$$

in which $$\xi(x, y) = \begin{cases} 0, & if \quad x < y \\ 1, & if \quad x \geq y \end{cases}$$

Please note that not all HEP texture descriptors could be bit-wise rotated, and some HEP texture descriptors could have more than 1,000 dimensions. This study chose to investigate LBP and the modified BGC1 because of their performance, low dimensionality, rotation invariant capability, and their computational simplicity. Once the GLCM, LBP, and BGC1 texture features had been calculated, they were simply concatenated to spectral and shape features to form a set of feature values for each segmented object, which could be used in the classification process. Table 4 shows a list of the different texture descriptors which were tested, along with their number of dimensions and the total number of dimensions after including the 15 spectral features and 10 shape features.

Table 4 A list of the Texture Features tested in this study and their dimensions.

Texture	Details	Number of texture features (dimensions)	Total Number of features (dimensions)
No texture	Only use Spectral and Shape features	0	25
GLCM	Using 5 selected GLCM 2nd order statistics	5	30
BGC1	The original BGC1	255	280
BGC1Rot	The proposed rotation invariant BGC1Rot	35	60
LBP	The original LBP	256	281
LBPRot	The rotation invariant LBPRot	36	61
LBPUni	The rotation invariant Uniform LBPUni	9	34

2.6 Class labeling and sample selection

One of the most important, but seldom mentioned, processes, which can determine the accuracy or performance of a GEOBIA classification model, is the class labeling process [1]. Not only do inappropriate labels affect the sample selection process, but they can also lead to a flawed accuracy assessment due to differences in the remote sensing platform's capabilities, spatial resolutions, and band limitations. In addition, segmentation results could also hinder the performance of the accuracy assessment process if objects with mixed landcover classes were used to train and to test models. The labeling and sample selection process was a long and tedious task requiring multiple revisions. The class labels were initially based on the CORINE Land Cover (CLC) nomenclature [25]. Some labels and codes were added and modified to become applicable for the land use / land cover of the area of Khon Kaen city and the 2-meter resolution of Theos pansharpened image. The 10 land cover classes and their numbers of objects used for this study are in shown in Table 5. Please note that in order to avoid errors from mixed-class objects, only objects with homogenous classes were selected.

Table 5 A list of 10 class labels and number of samples.

ID	Class labels	Number of samples
2	Farmed vegetation, crops, and grass	1,011
4	Inland Marsh and water plants	125
5	Water	458
111	Residential and mixed buildings	639
121	Industrial or commercial units	523
122	Road	151
310	Forest (Dense large tree)	280
320	Shrubs and sparse vegetation	435
2110	Dark soil	453
2111	Plowed and bare land	1,342
	Total	5,417

2.7 Object classification

In addition to validating texture descriptors, this study also compared 6 basic machine learning supervised classifiers. Each classifier has its own parameters, whichcan be adjusted to optimize the model. Table 6 displays a list of classifiers and their parameters used for this study. The classification step was performed using Matlab's machine learning toolbox developed by Jing Wei Too [26].

Table 6 The classifiers and their adjustable parameters.

Supervised classifiers	Parameters	Selected
K-Nearest Neighbor (KNN)	K number of nearest neighbors: [1 - 20]	10
Discriminant Analysis (DA)	Methods: [Linear, Quadratic, Pseudoquadratic, Pseudolinear, Diaglinear, and Diagquadratic]	Linear
Naïve Bayes (NB)	Distributions: [Normal, Kernel]	Kernel
Decision Tree (DT)	Numbers of Splits: [10, 20, 30, … ,200]	80
Support Vector Machine (SVM)	Kernel functions: [Radial Basis Function (RBF), Linear, Polynomial]	RBF
Random Forest (RF)	Numbers of Trees: [100,200,300,500,800,1000]	100

(1) K-Nearest Neighbor (KNN) – Nearest Neighbor, which is one of the most frequently used classification algorithms, is known for its simple computation [27]. The KNN algorithm assigns a class to an unclassified object by calculating the distance between the object and the trained k nearest neighbors in the feature space, in which the majority of the classes among the k neighbors is assigned to the object. This study used Euclidean distance and tested K numbers from 1 to 20 to find the most suitable K for the model, from which, 10 was selected.

(2) Discriminant Analysis (DA) – DA is also called the Fisher Discriminant, which is based on Fisher's score optimization [28]. DA is a statistical based classifier, which calculates the statistical properties of each class from the sample dataset. It uses means and the covariance matrix to distinguish different classes. The algorithm was designed for continuous independent predictors and multi-categorical dependent targets, which made it an ideal classifier for this study. The drawback of DA is that it is a parametric method, which assumes a normal distribution of data. Linear DA was used in this study.

(3) Naïve Bayes (NB) – NB algorithm is a probabilistic classifier which was developed based on Bayes' theorem by Thomas Bayes. It is computationally simple and is able to handle large datasets well. NB predicts class labels by calculating the posterior probability for each class using *a priori* probability from the training data. Since the data for this study is not normal distribution, kernel distribution was used for the NB model.

(4) Decision Tree (DT) or CART – DT is one of the most widely used non-parametric classification algorithms and is known for its intuitive simplicity. The algorithm recursively splits datasets into homogenous groups based on conditions, rule sets or threshold numbers, and thereby, creates an upside-down tree-like structure. This study tested different number of splits ranging from 10 to 200 (in increments of 10) to choose the most optimal number, which was found to be 80.

(5) Support Vector Machine (SVM) – The SVM algorithm calculates for hyperplanes, which best separate the classes in the feature space. The chosen hyperplanes are the ones that have the largest margins between the planes and the closest samples for each class. These samples are the support vectors. For complex tasks, linear hyperplanes may not be suitable [29]. Therefore, the Radial Basis Function (RBF) kernel function is used to calculate a non-linear hyperplane for the model.

(6) Random Forest (RF) – RF algorithm is an ensemble classification of decision trees [30], and is also a non-parametric algorithm. Each tree is trained with random subsets of datasets and features (called bagging). Any subsets, which have not been used to train the trees, are called out-of-bag data sets, and are used to test and fine tune the model. The majority votes from all trees are used to assign a class to an unclassified sample. Different numbers of trees [100, 200, 300, 500, 800, 1000] were tested to determine the most optimal model, of which, no significant improvement was shown with a number of trees higher than 100. In addition, the RF algorithm utilizes the GINI index to measure the entropy of each feature split, which was used to select the five best second order statistics of GLCM.

2.8 Accuracy assessment

Ma et al [1] suggested that for the GEOBIA accuracy assessment, an area-based approach should be used over a point-based or pixel-based. Using the area-based approach means the whole individual segmented object is utilized as a unit for the validation process in place of the random pixels within an object. The advantage of area-based is that it investigates all features and spatial distributions of the segmented objects, which correspond to the principles of OBIA. All labeled samples from the class labeling process were used for model accuracy assessment.

The K-fold cross validation method [31,32] was used to evaluate the performance of each model. The main advantage of the cross validation approach versus a normal random split of training and testing data is the reduction of any bias and the chance of selecting underfit or overfit training data by using all the samples as the testing and training data [33]. This study used a stratified 5-fold cross validation method, in which each fold contained 20% of the stratified random samples of the 5,417 objects from 10 land cover classes (Table 5) in order to validate the performance of each classification model. Each fold retained the same ratio of samples from each class as the overall class ratio of all samples. The results of the 5-fold cross validation were merged into one confusion matrix, from which the Overall Accuracy (OA)(Equation 12), the User's Accuracy (UA) (Equation13), the Producer's Accuracy (PA) (Equation 14), the Kappa Coefficient (Equation.15), and the F1-Score (Equation18) were calculated. It is important to note that the sample datasets were significantly imbalanced among classes, which indicated that the OA did not represent the actual performance of a model for different target classes. In addition, since it was possible that different classifiers and texture descriptors could provide better results for different classes, this study evaluated the model's performance of each class by using the class F1-Score. The F1-Score is based on the F-Measure proposed by Chinchor [34] to calculate harmonic mean of the UA (or recall) and the PA (or precision), which is more intuitive and more robust than a simple arithmetic mean [35].

$$\text{Overall Accuracy (OA)} = \frac{\text{TP} - \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \qquad (12)$$

$$\text{User's Accuracy (UA)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (13)$$

$$\text{Producer's Accuracy (PA)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (14)$$

$$\text{Kappa} = \frac{\text{p}_0 - \text{p}_\text{e}}{1 - \text{p}_\text{e}} \qquad (15)$$

$$\text{p}_0 = \frac{\text{TP}-\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}} \qquad (16)$$

$$\text{p}_\text{e} = \frac{(\text{TP} + \text{FN})\text{x}(\text{TP} + \text{FP}) + (\text{FP} + \text{TN})\text{x}(\text{FN} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})^2} \qquad (17)$$

$$\text{F1}_\text{Score} = \frac{2\text{xTP}}{2\text{xTP} + \text{FP} + \text{FN}} = \frac{2\text{xUAxPA}}{\text{UA} + \text{PA}} \qquad (18)$$

in which TP is True Positive
TN is True Negative
FP is False Positive (Commission error)
FN is False Negative (Omission error)

3. Results and discussion

3.1 Segmentation result

The segmentation results from the trial-and-error processes were subjectively assessed by eye to determine the best combination of parameters of spectral features, scale levels, weights of shapes (and colors), weights of compactness (and smoothness), and the features' weights. The best scale levels, shape weights, and weights of compactness were found to be 150, 0.2, and 0.5, respectively. Red, green, blue, NDVI, and PCA2 were given parameter weights of 1, while both NIR and PCA1 had the weight of 2. BAI, NDWI, SSI, and PCA3 were excluded (weight of zero) because these features introduced noises within image objects, which diminished the segmentation quality. The segmentation process resulted in 9,929 objects, which are shown in Figure 6.

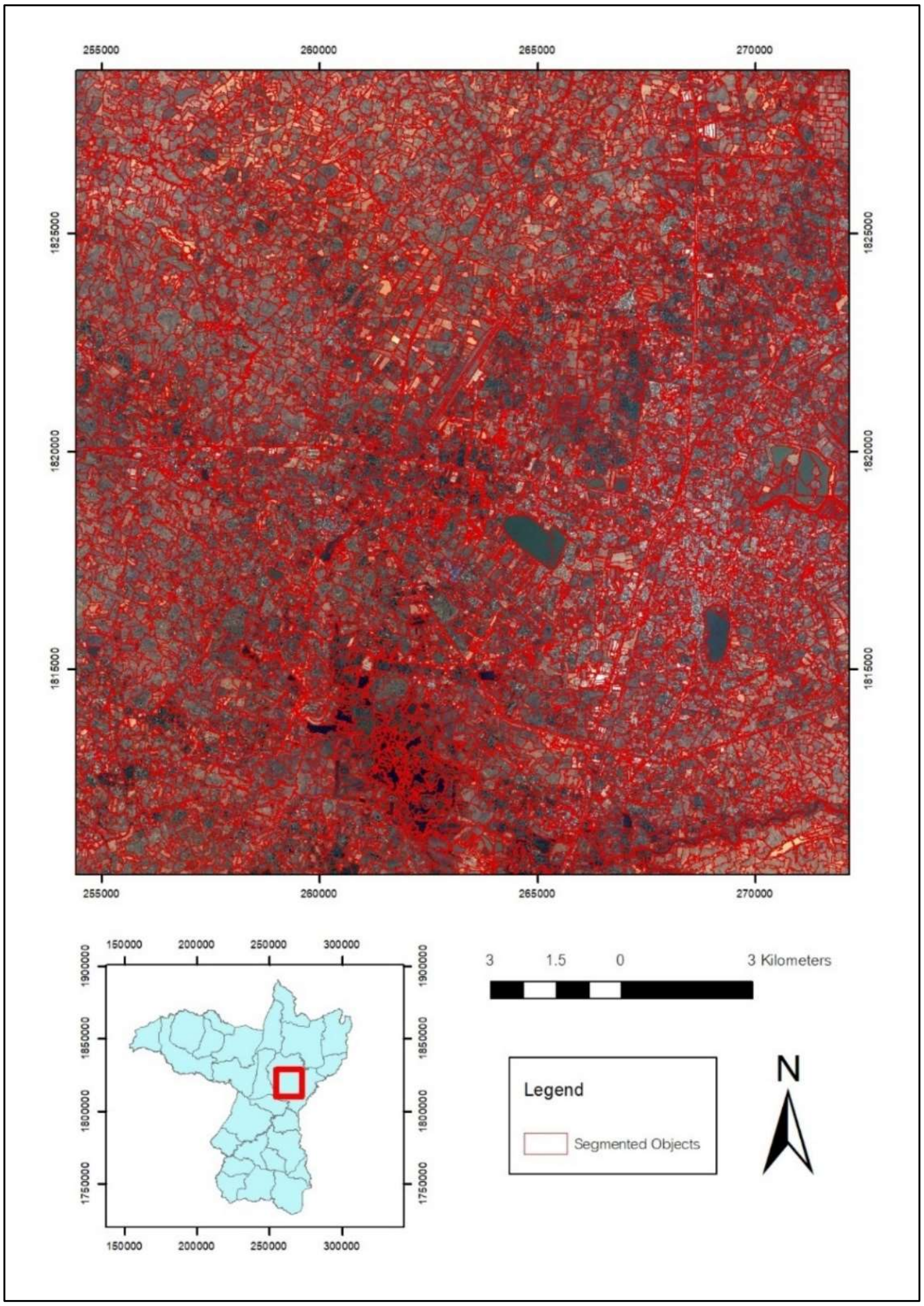
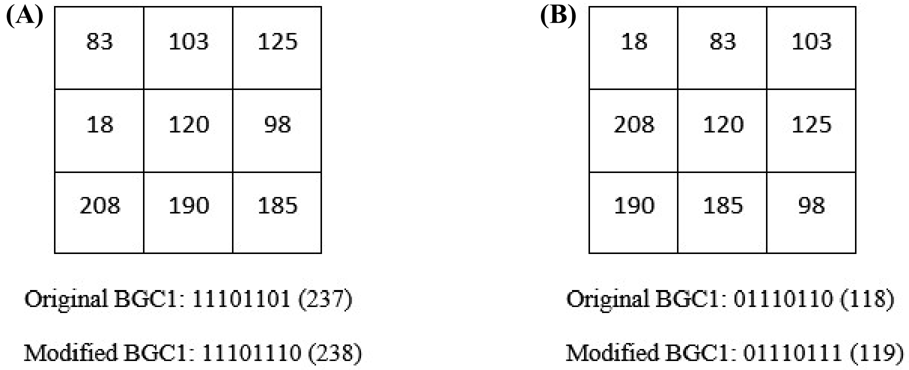


Figure 6 The multiresolution image segmentation results from 9,929 objects (Scale level: 150, Shape: 0.2, Color: 0.8, Compactness: 0.5, and Smoothness: 0.5).

3.2 BGC1 texture features extraction results

Figure 7 exhibits the rotation invariant capability of the modified BGC1 from Equation 11. As mentioned earlier, the only difference between the original BGC1 and the modified BGC1 was that the original, which has a value range of between 0-254, while the modified version has a value range of between 1-255. Nevertheless, this slight difference was determined to be extremely crucial for the modified BGC1 so that it could become a rotation invariant descriptor. For example, in Figure 7 (A) and (B) below, consider the two identical patterns, which have different orientations. Using Eq.11, the encoded binary number of the modified BGC1 would be 11101110 (238) and 01110111 (119) for pattern (a) and for pattern (b), respectively, allowing them to be minimum bit-wise rotated to the same value of 01110111 (119). However, the original BGC1 from Equation 3 would encode the 2 patterns as 11101101 (237) and 01110110 (118) (each has one value less than the modified BGC1) which would not allow them to be rotated in any way that could signify the same pattern of different orientation. Figure 8 shows the result of the original BGC1, while Figure 9 shows the result of the minimum bit-wise rotation of the modified BGC1Rot. From visual inspection, objects from BGC1Rot had clearer edges, showed better contrast, and had less noise. For LBP and BGC1, each histogram bin was converted into a rate of occurrence and concatenated to the values of the other features of each object. For GLCM, the aforementioned five second order statistics were concatenated.



(A)

83	103	125
18	120	98
208	190	185

Original BGC1: 11101101 (237)

Modified BGC1: 11101110 (238)

(B)

18	83	103
208	120	125
190	185	98

Original BGC1: 01110110 (118)

Modified BGC1: 01110111 (119)

Figure 7 Example of the same pattern (a) and (b), but with a rotated orientation. The modified BGC1 value of pattern (A) could be minimum bit-wise rotated to become the same value as pattern (B), 01110111 (119), but the original BGC1 could not.

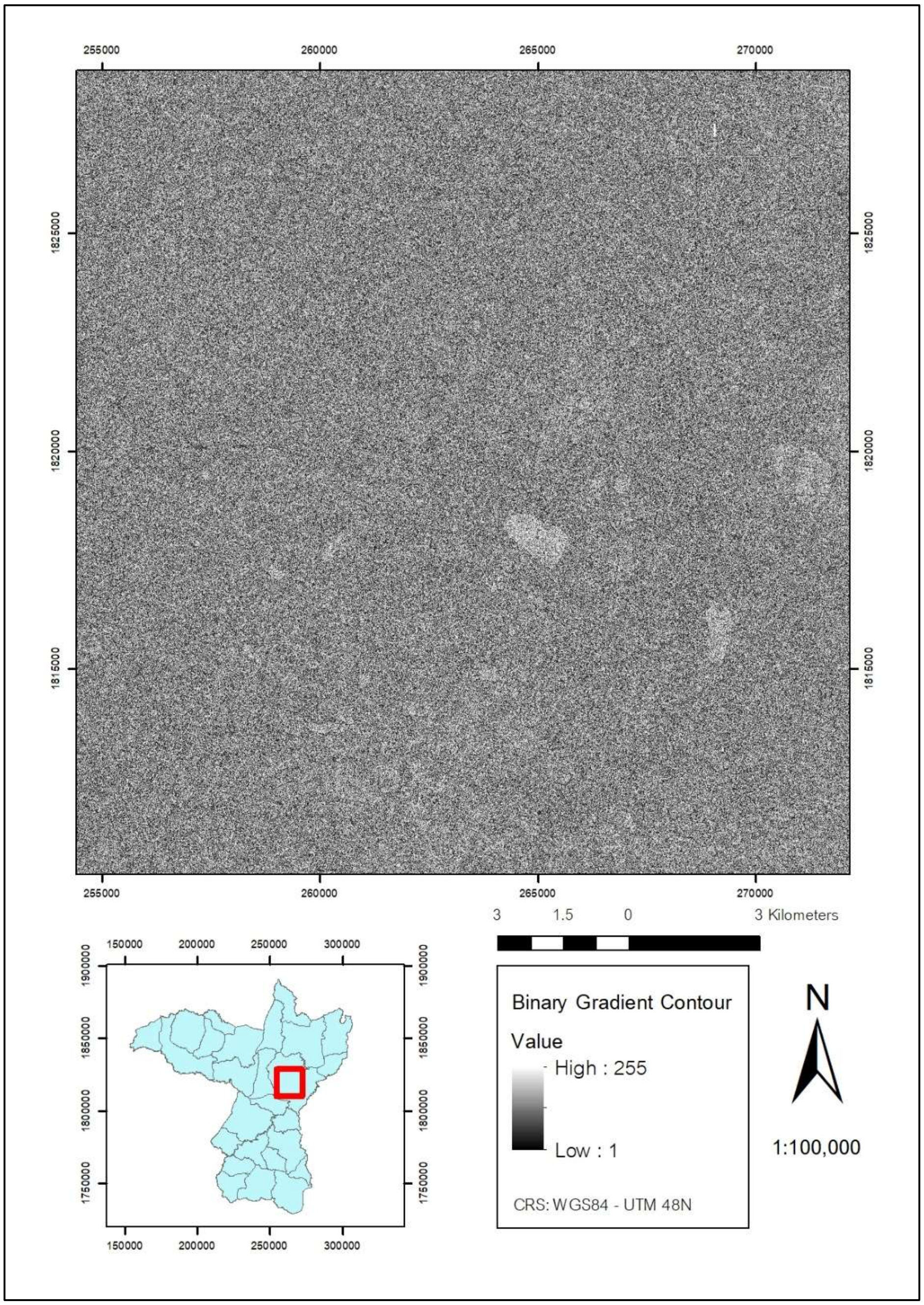


Figure 8 The result of the original Binary Gradient Contour 1 (1-255).

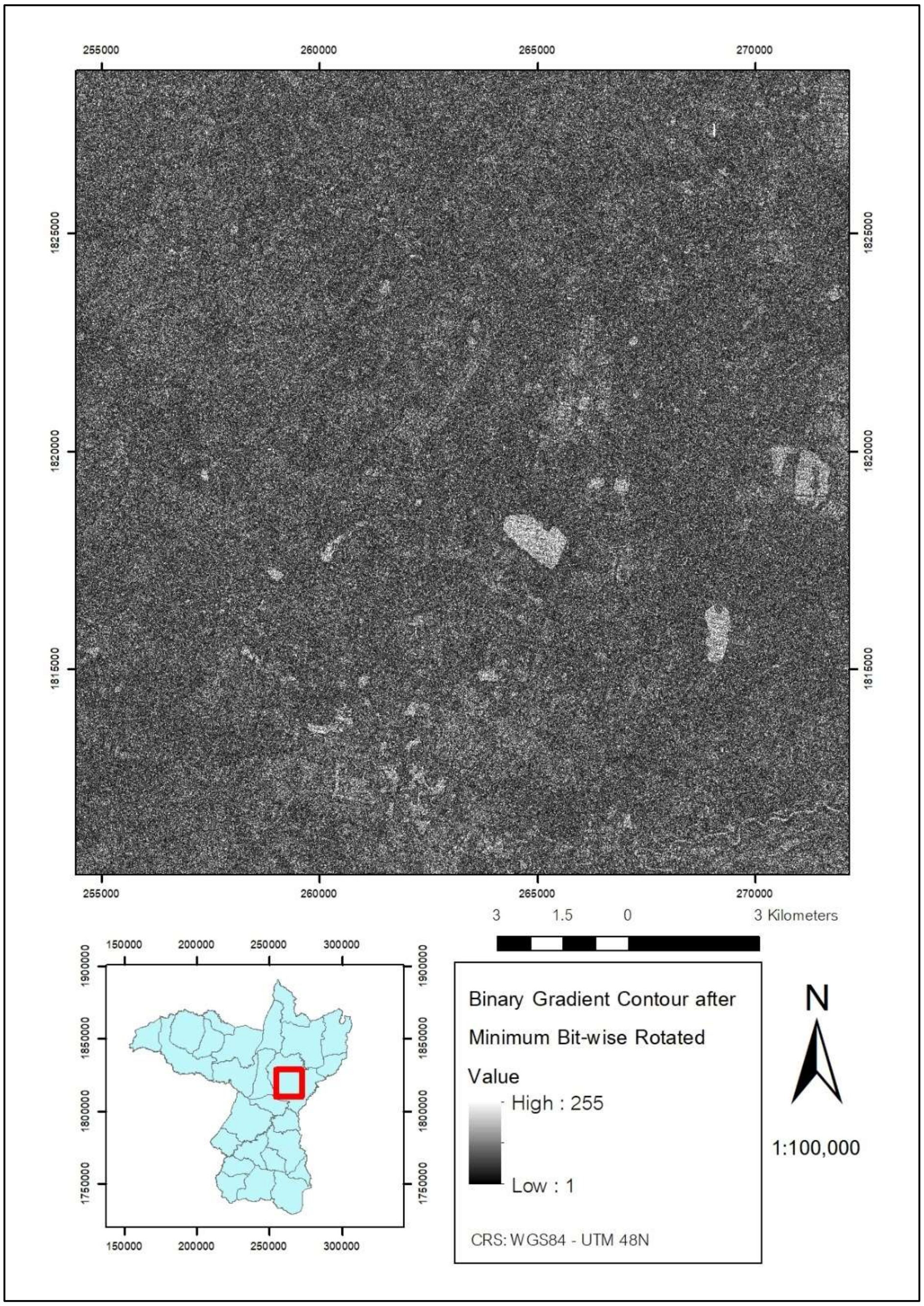


Figure 9 The result of the modified Binary Gradient Contour after the minimum bit-wise rotation ($BGC1_{Rot}$).

3.3 Overall accuracy and computation efficiency

Although the Overall Accuracy (OA) does not indicate the true performance of each classifier for each individual class label when the input data is imbalanced, it could still be used to provide a quick performance estimation. Please note that this study did not perform any post-classification steps, such as merging, splitting, or dissolving objects, which should be performed when creating a land cover map. Table 7 displays the overall accuracy of the 10-class scenario from the 5-Fold cross validation with different combinations of texture features and classifiers. As expected, all classifiers produced better results when the processing had been carried out with one of the texture descriptors than when no texture features had been used. Evidently, the best classifier for this study was the Random Forest (RF), which had provided the best results with all texture descriptors. The best model was the combination of RF classifier and the proposed $BGC1_{Rot}$ (including the spectral and shape features), which had an overall accuracy of 84.863% and a Kappa coefficient of 0.823, suggesting the strong reliability of the results. The Z statistic at 95% confidence of pairwise comparison between RF and other classifiers had been 11.853, 3.433, 18.158, 8.979, 3.591 when compared to KNN, DA, NB, DT, and SVM, respectively, all of which were well above the 1.96 significant threshold. Even though RF was a significantly more superior classifier, the performance margins between each texture were very small. Table 8 shows the Z statistic at 95% confidence of the pairwise comparison among texture descriptors under the RF classifier. Even though BGC_{Rot} had produced the highest OA, it was only statistically significant when compared to NoTexture and to the original BGC1 and LBP. Despite its age, GLCM also proved to be an effective texture descriptor. However, the strength of BGC1 and LBP is their computational efficiency. When choosing a descriptor or a model, the computation time should also be considered. Later, the individual performance of each class label will also be discussed in detail.

Table 7 The overall accuracy (%) of 6 classifiers with spectral and shape features combined with different texture descriptors.

Classifiers	NoTexture	GLCM	BGC1	$BGC1_{Rot}$	LBP	LBP_{Rot}	LBP_{Uni}
KNN	74.248	77.700	73.966	74.580	73.984	74.746	74.617
DA	74.267	82.297	0.000*	82.038	0.000*	80.635	79.786
NB	69.632	76.629	66.993	73.731	67.195	76.297	76.445
DT	73.011	80.727	75.227	78.770	75.429	77.054	77.589
SVM	77.330	82.794	78.124	79.842	77.700	79.306	79.177
RF	78.438	84.475	82.832	84.863	82.832	83.921	84.069

* The Discriminant Analysis function in Matlab failed to produce results due to exceeding the limit of the number of features.

Table 8 The Z statistic at 95% confidence of pairwise comparisons between different texture features using RF.

Texture features	NoTexture	GLCM	BGC1	$BGC1_{Rot}$	LBP	LBP_{Rot}	LBP_{Uni}
No texture	0	9.361	6.674	9.944	6.629	8.465	8.697
GLCM	9.361	0	2.629	0.593	2.650	0.881	0.649
BGC1	6.674	2.629	0	3.216	0.028	1.749	1.980
$BGC1_{Rot}$	9.944	0.593	3.216	0	3.236	1.473	1.241
LBP	6.629	2.650	0.028	3.236	0	1.773	2.003
LBP_{Rot}	8.465	0.881	1.749	1.473	1.773	0	0.232
LBP_{Uni}	8.697	0.649	1.980	1.241	2.003	0.232	0

In this study, the computation efficiency assessment was split into 2 phases: the feature extraction phase and the classification phase. The majority of the computation time was in the feature extraction phase, in which the texture descriptors were calculated and extracted from each object. Using a general laptop with 2.30 GHz cpu speed and 16 gigabytes of ram, the processing times for GLCM, LBP, and BGC1, which are shown in Table 9, were determined. As expected, LBP and BGC1 had similar processing times of 466.286 and 450.938 seconds, respectively, and were found to be more than 4.4 times faster than GLCM. This could have been because, in order for GLCM to be a semi-rotation invariant, four different second order matrices (for the directions of 0°, 45°, 90°, and 135°) had to be generated, which had a size of 256 x 256 each and which resulted in a large histogram size of 65,536 bins for each of the 9,929 objects. Please note that this processing time did not include the calculation time for the second order GLCM statistics.

Table 9 The feature extraction computation time of the 3 texture descriptors (seconds).

Texture descriptors	Computation time for 9,929 objects
GLCM	2020.200
LBP	466.286
BGC1	450.938

For the classification phase, the results were as expected. With a smaller number of features, there was less computation time for all classifiers (Table 10). Since RF provided the best accuracy, this report will, from here on, focus on the results of the RF classifier. The original BGC1 and LBP had 255 and 256 dimensions, respectively. Therefore, the computations involving these features took an exceptionally long time (485.644 and

516.23 seconds for the original BGC1 and LBP, respectively). The modified rotation invariant $BGC1_{Rot}$ reduced the dimension down to 35 dimensions, which cut the computation time down to one-third of the original, 163.148 seconds. LBP_{Rot} and LBP_{Uni} reduced the dimension from the original LBP to 36 and 9 dimensions, respectively. As a result, LBP_{Uni} showed a significant improvement in computation time as compared to the original LBP of 65.355 seconds (about 8 times faster than the original). GLCM displayed the quickest computation time of 57.598 seconds, but it should be noted that only 5 second order statistics, which were derived from GLCM, were used classifications (5 dimensions). Moreover, this time did not include the feature extraction processing time, of which GLCM was the worst at 2,020 seconds (about 33 minutes) when compared to the times of 466 and 451 seconds by LBP and BGC1, respectively. The total computation time is shown in

Table 11 11. From the results, although $BGC1_{Rot}$ had about the same overall accuracy as GLCM, it clearly outperformed GLCM in computational efficiency, along with the gray-scale and rotation invariant properties.

Table 10 The classification computation time for each combination of classifiers and texture descriptors (seconds).

Classifiers	NoTexture	GLCM	BGC1	$BGC1_{Rot}$	LBP	LBP_{Rot}	LBP_{Uni}
KNN	0.495	0.373	2.040	0.584	1.973	0.545	0.346
DA	0.232	0.226	0.000	0.461	0.000	0.343	0.228
NB	30.321	35.656	377.405	89.424	380.577	76.937	40.612
DT	0.404	0.382	2.907	0.725	2.943	0.706	0.434
SVM	6.209	6.252	12.789	9.671	12.131	6.878	6.260
RF	56.838	57.598	485.644	163.148	516.230	115.997	65.355

Table 11 The total computation time (seconds).

Classifiers	NoTexture	GLCM	BGC1	$BGC1_{Rot}$	LBP	LBP_{Rot}	LBP_{Uni}
KNN	0.50	2,020.57	452.98	451.52	468.26	466.83	466.63
DA	0.23	2,020.43	0.00	451.40	0.00	466.63	466.51
NB	30.32	2,055.86	828.34	540.36	846.86	543.22	506.90
DT	0.40	2,020.58	453.85	451.66	469.23	466.99	466.72
SVM	6.21	2,026.45	463.73	460.61	478.42	473.16	472.55
RF	56.84	2,077.80	936.58	614.09	982.52	582.28	531.64

3.4 Class F1-score

As mentioned earlier, the overall accuracy does not indicate the true performance of a model when there is an imbalance of input classes. Therefore, it was necessary to individually investigate the accuracy of each class to verify the models' performance. The F1-Score (Equation. 18), which calculates the harmonic mean from the User's Accuracy (UA) and the Producer's Accuracy (PA) of each class, was used to test the individual class performance. The F1-Score has a range of between 0 and 1, in which 1 represents a perfect score (no commission or omission errors). Table 12 shows a summary of the highest F1-scores from all 10 classes and from all possible combinations of the texture descriptors and classifiers. Again, RF provided the highest F1-Score for all classes. The RF classifier and $BGC1_{Rot}$ produced the highest average F1-score of 0.843, and the highest F1-scores for 6 out of 10 classes (i.e., crops, marshes, water, forests, shrubs, and bare land), as well as a tied with LBP_{Uni} for water class (Table 13). The proposed $BGC1_{Rot}$ outperformed the original BGC1 in every class.

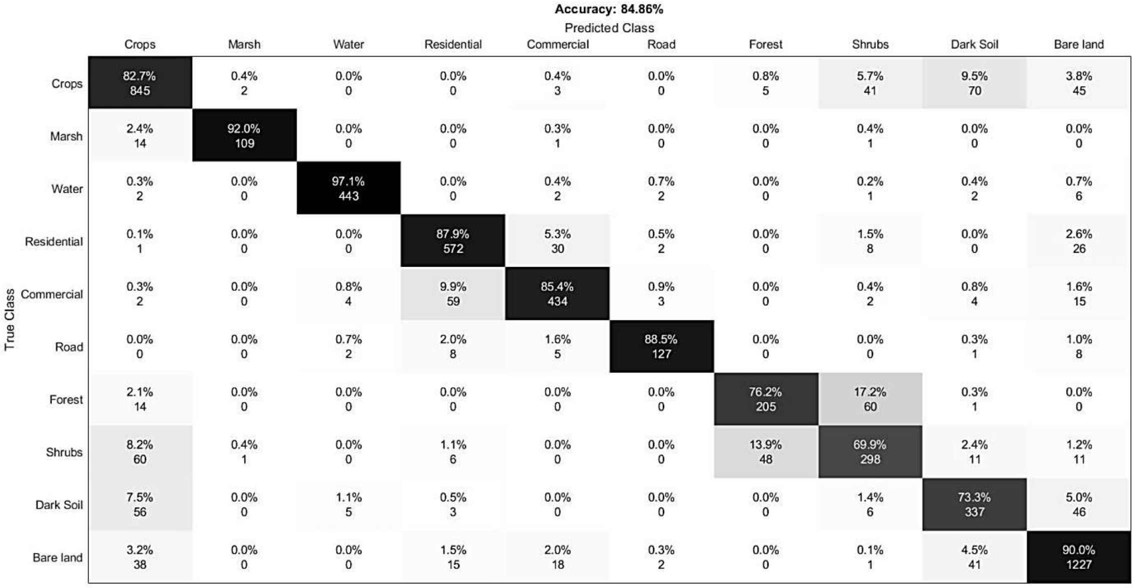
Table 12 A summary of the best combinations of texture descriptors and classifiers for each class.

Classes	Highest F1-scores	Texture descriptor combinations	Classifiers
Crops	0.827	$BGC1_{Rot}$	RF
Marsh	0.920	$BGC1_{Rot}$	RF
Water	0.971	LBP_{Uni} , $BGC1_{Rot}$	RF
Residential	0.904	GLCM	RF
Commercial	0.866	LBP_{Uni}	RF
Road	0.917	LBP_{Rot}	RF
Forest	0.784	$BGC1_{Rot}$	RF
Shrubs	0.729	$BGC1_{Rot}$	RF
Dark Soil	0.744	LBP_{Uni}	RF
Bare land	0.906	$BGC1_{Rot}$	RF

Table 13 The texture descriptors' F1-Scores for all classes from the Random Forest classifier.

Classifiers	NoTexture	GLCM	BGC1	$BGC1_{Rot}$	LBP	LBP_{Rot}	LBP_{Uni}
Crops	0.733	0.817	0.815	0.827	0.797	0.812	0.818
Marsh	0.897	0.906	0.894	0.920	0.879	0.911	0.889
Water	0.961	0.969	0.956	0.971	0.958	0.966	0.971
Residential	0.787	0.896	0.840	0.879	0.868	0.861	0.860
Commercial	0.842	0.856	0.848	0.854	0.855	0.861	0.863
Road	0.846	0.900	0.812	0.885	0.848	0.900	0.896
Forest	0.686	0.740	0.702	0.762	0.748	0.746	0.736
Shrubs	0.442	0.687	0.626	0.699	0.563	0.654	0.656
Dark Soil	0.665	0.716	0.721	0.733	0.709	0.729	0.744
Bare land	0.881	0.897	0.896	0.900	0.898	0.896	0.895
Avg.	0.774	0.839	0.811	0.843	0.812	0.834	0.833

Figure 10 displays the confusion matrix of the RF classifications with $BGC1_{Rot}$ texture descriptors. Water had the highest F1-Score of 0.971, followed by Road, Marsh, and Bare land. The classes with lowest F1-Scores were Dark Soil and Shrubs. Dark soil was confused with crops and Bare land, while Shrubs were confused with Crops and Forest. Although, the proposed $BGC1_{Rot}$ improved the classification accuracy for classes of Forest and Shrubs, they had a relatively low accuracy when compared to other classes. These two classes had the highest commission and omission errors with each, which could have resulted from poor input samples during the labeling process and from their similar spectral signatures. Similarly, during the labeling process, the Crops and Dark soil classes were mixed up since it was difficult to distinguish them by eye. Another notable area of class confusion was the Residential and Commercial areas in which about 10% of the predicted Residential objects, in fact, belonged to the Commercial class. Depending upon the applications, these confusing pairs of classes can be merged into a single class, such as Vegetation, Crop field, or the Built-up class. Nevertheless, the accuracy of Forest, Shrubs, Crops, Dark soil, and Residential improved significantly when the texture features were used in the analysis and were compared to those land cover classes for which texture had little effect: Commercial, Bare land, Marsh, Water, and Road (Table 13).



	Crops	Marsh	Water	Residential	Commercial	Road	Forest	Shrubs	Dark Soil	Bare land
Crops	82.7% 845	0.4% 2	0.0% 0	0.0% 0	0.4% 3	0.0% 0	0.8% 5	5.7% 41	9.5% 70	3.8% 45
Marsh	2.4% 14	92.0% 109	0.0% 0	0.0% 0	0.3% 1	0.0% 0	0.0% 0	0.4% 1	0.0% 0	0.0% 0
Water	0.3% 2	0.0% 0	97.1% 443	0.0% 0	0.4% 2	0.7% 2	0.0% 0	0.2% 1	0.4% 2	0.7% 6
Residential	0.1% 1	0.0% 0	0.0% 0	87.9% 572	5.3% 30	0.5% 2	0.0% 0	1.5% 8	0.0% 0	2.6% 26
Commercial	0.3% 2	0.0% 0	0.8% 4	9.9% 59	85.4% 434	0.9% 3	0.0% 0	0.4% 2	0.8% 4	1.6% 15
Road	0.0% 0	0.0% 0	0.7% 2	2.0% 8	1.6% 5	88.5% 127	0.0% 0	0.0% 0	0.3% 1	1.0% 8
Forest	2.1% 14	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	76.2% 205	17.2% 60	0.3% 1	0.0% 0
Shrubs	8.2% 60	0.4% 1	0.0% 0	1.1% 6	0.0% 0	0.0% 0	13.9% 48	69.9% 298	2.4% 11	1.2% 11
Dark Soil	7.5% 56	0.0% 0	1.1% 5	0.5% 3	0.0% 0	0.0% 0	0.0% 0	1.4% 6	73.3% 337	5.0% 46
Bare land	3.2% 38	0.0% 0	0.0% 0	1.5% 15	2.0% 18	0.3% 2	0.0% 0	0.1% 1	4.5% 41	90.0% 1227

Figure 10 The Confusion Matrix and F1-Score (in percentages) from the RF classifier and BGC1Rot texture descriptor.

In summary, the popular HEP texture descriptors, LBP and BGC, have been accepted and widely used in the field of pattern recognition and computer vision. However, they have rarely been applied to remote sensing data, especially with regard to their rotation invariant capability. Therefore, this study examined the capability of the proposed rotation invariant BGC1Rot for land use and for land cover classification under GEOBIA. LBP, Uniform LBP_{Uni} and GLCM, were also tested for comparisons. Six machine learning algorithms (KNN, DA, NB, DT, SVM, and RF) were investigated to determine the most suitable model combination of texture descriptors and classifiers. Overall accuracy, Kappa statistics, F1-Score, and computation time were the evaluating factors under the 5-fold cross validation method. For the input, a Theos image of Khon Kaen city area was segmented into objects, which were then labeled into one of the ten classes. The spectral, shape, and texture features for each object were then extracted, which resulted in 15 spectral features, 10 shape features, and 6 texture features as shown in Table 2, Table 3 and Table 4. GLCM and its second order statistics were calculated in eCognition, while Matlab was used to compute the texture features for LBP, BGC1, and for their rotation invariant variations.

Based on overall accuracy, the best model was the combination of the RF algorithm and the input from 15 spectral features, 10 shape features, and the proposed BGC1_{Rot} texture descriptor, which had the overall accuracy of 84.863%. As expected, GLCM presented the worst computation efficiency among the investigated texture descriptors, which took 2,020.2 seconds in the feature extraction process. This was more than quadruple the time with LBP and BGC1. When considering the overall accuracy, Kappa statistics, and the computation efficiency, the RF classifier with the proposed BGC1_{Rot} texture descriptor was the most efficient model. In addition, BGC1_{Rot} provided rotation invariance and gray-scale invariance, which enabled the descriptor to work with the arbitrary shapes and directions of the land objects and the temporal data in which there may be variances in the sun's angle or in the atmospheric conditions. However, it is important to note that the datasets used in this study were heavily imbalanced among class samples. Therefore, the overall accuracy might not translate to the performance for individual land cover class. Thus, F1-Score was used to rank the model's performance in each class. Based on the average of the F1-Scores, the best model was, again, the combination of RF algorithm with input from spectral features, shape features, and BGC1_{Rot}. This study used sample image objects, which had been segmented from a 2-meter pan-sharpened Theos satellite imagery, with only four basic multispectral bands. Depending upon class labeling, the accuracy of the models should improve when images from remote sensing platforms with superior spatial and radiometric resolution are used, as well as when temporal data and vegetation phenology signatures are used.

4. Conclusion

This study proved that the proposed rotation invariant texture descriptor, BGC1Rot, is an effective texture descriptor for satellite image land cover classification for GEOBIA. The proposed BGC1Rot showed significant improvements in both accuracy and computational efficiency when compared to GLCM and the original BGC1. The results also confirmed the superiority of BGC1 over LBP, supporting the results of [9], in which the original non-rotation invariant BGC1 outperformed the original LBP. Therefore, employing the BGC1Rot texture descriptor for future land use / land cover classification of Theos and other similar high resolution remote sensing platform is recommended.

5. Acknowledgements

We would like to thank the Geo-Informatics Centre for Development of Northeast Thailand at Khon Kaen University in Thailand for providing computers, tools, and a great atmosphere in which to carry out the research. In addition, we express our gratitude to the Geo-Informatics and Space Technology Development Agency of Thailand (GISTDA), which provided the satellite images for this research. Finally, we would like to also thank the editors and reviewers for their time and efforts in reviewing the manuscript and sharing their helpful comments.

6. References

[1] Ma L, Li M, Ma X, Cheng L, Du P, Liu Y. A review of supervised object-based land-cover image classification. ISPRS J Photogramm Remote Sens. 2017;130:277-293.

[2] Hay GJ, Castilla G. Geographic object-based image analysis (GEOBIA): a new name for a new discipline. In: Blaschke T, Lang S, Hay GJ, editors. Object-based image analysis. 1st ed. New York: Springer; 2008. p.75-89.

[3] Momeni R, Aplin P, Boyd DS. Mapping complex urban land cover from spaceborne imagery: the influence of spatial resolution, spectral band set and classification approach. Remote Sens. 2016;8(2):88.

[4] Bhaskaran S, Paramananda S, Ramnarayan M. Per-pixel and object-oriented classification methods for mapping urban features using Ikonos satellite data. Appl Geogr. 2010;30(4):650-65.

[5] Azofeifa SGA, Sinkwich JD, Kang KK. Object-based and pixel-based classification comparison of high-resolution quickbird data in forested alberta [Internet]. Alberta: Alberta Biodiversity Monitoring Institute; 2014 [cited 2021 June 1]. Availale from: http://www.abmi.ca/dam/jcr:f61e944d-9be3-48da-bf2a-25bf982dcc19/20056_ABMI_2004_Quickbird_Report.pdf.

[6] Blaschke T, Hay GJ, Kelly M, Lang S, Hofmann P, Addink E, et al. Geographic object-based image analysis-towards a new paradigm. ISPRS J Photogramm Remote Sens. 2014;87(100):180-191.

[7] Haralick RM, Shanmugam K, Dinstein IH. Textural features for image classification. Media Commum. 1973;3(6):610-621.

[8] Ojala T, Pietikainen M, Harwood D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. The 12th IAPR International Conference on Pattern Recognition; 1994; Oct 9-13, Jerusalem, Israel. California: IEEE; 1994. p. 582-585.

[9] Fernández A, Álvarez MX, Bianconi F. Image classification with binary gradient contours. Opt Lasers Eng. 2011;49(9-10):1177-1184.

[10] Fernández A, Álvarez MX, Bianconi F. Texture description through histograms of equivalent patterns. J Math Imaging Vis. 2013;45(1):76-102.

[11] Song C, Yang F, Li P. Rotation invariant texture measured by local binary pattern for remote sensing image classification. 2010 Second International Workshop on Education Technology and Computer Science; 2010 Mar 6-7; Wuhan, China. California: IEEE; 2010. p.3-6.

[12] Aguilar MA, Fernández A, Aguilar FJ, Bianconi F, Lorca AG. Classification of urban areas from geoeye-1 imagery through texture features based on histograms of equivalent patterns. Eur J Remote Sens. 2016;49(1):93-120.

[13] Ojala T, Pietikäinen M, Mäenpää T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE PAMI. 2002;24(7):971-987.

[14] Wang L, He DC. Texture classification using texture spectrum. Pattern Recognit. 1990;23(8):905-910.

[15] Villa P. Imperviousness indexes performance evaluation for mapping urban areas using remote sensing data. 2007 Urban Remote Sensing Joint Event; 2007 Apr 11-13; Paris, France. California: IEEE; 2007. p. 1-5.

[16] Rouse JW, Haas R, Schell J, Deering D. Monitoring vegetation systems in the great plains with ERTS. 1974;1:1-9.

[17] McFeeters SK. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. Int J Remote Sens. 1996;17(7):1425-1432.

[18] Chen Y, Su W, Li J, Sun Z. Hierarchical object oriented classification using very high resolution imagery and LIDAR data over urban areas. Adv Space Res. 2009;43(7):1101-1110.

[19] Huete AR. A soil-adjusted vegetation index (SAVI). Remote Sens Environ. 1988;25(3):295-309.

[20] Mhangara P, Odindi J, Kleyn L, Remas H. Road extraction using object oriented classification [Internet]. California: WorldPress; 2005 [cited 2021 Jun 15]. Available from: http://www.ee.co.za/wp -content/uploads/Legacy/posit11/PositionIT_nov-dec11-Vis_45-50.pdf.

[21] Kumar M, Singh R, Raju P, Krishnamurthy Y. Road network extraction from high resolution multispectral satellite imagery based on object oriented techniques. ISPRS Ann Photogramm Remote Sens Spat Inf Sci. 2014;2(8):107-110.

[22] Johnson B, Bragais M, Endo I, Macandog MD, Macandog P. Image segmentation parameter optimization considering within- and between-segment heterogeneity at multiple scale levels: test case for mapping residential areas using landsat imagery. ISPRS Int J Geo Inf. 2015;4(4):2292.

[23] Myint SW, Gober P, Brazel A, Clarke GS, Weng Q. Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. Remote Sens Environ. 2011;115(5):1145-1161.

[24] Trimble Germany GmbH [Internet]. Munich: The Incorporation; 1978 [cited 2021 Jun 15]. eCognition 9.0 Available from: http://www.gisat.cz/images/upload/9c230_releasenotes.pdf.

[25] Bossard M, Feranec J, Otahel J. CORINE land cover technical guide: Addendum 2000. Copenhagen: European Environment Agency; 2000 May. Technical report No 40.

[26] Too JW. Machine Learning Toolbox [Internet]. California: GitHub, Inc.;2022. [cited 2020 Jun 16]. Available from https://github.com/JingweiToo/Machine-Learning-Toolbox.

[27] Lopez FH, Ek AR, Bauer ME. Estimation and mapping of forest stand density, volume, and cover type using the k-nearest neighbors method. Remote Sens Environ. 2001;77(3):251-274.

[28] Fisher RA. The use of multiple measurements in taxonomic problems. Ann Eugen. 1936;7(2):179-188.

[29] Kavzoglu T, Colkesen I. A kernel functions analysis for support vector machines for land cover classification. Int J Appl Earth Obs Geoinf. 2009;11(5):352-359.

[30] Breiman L. Random forests. Mach Learn. 2001;45(1):5-32.

[31] Stone M. Cross-validatory choice and assessment of statistical predictions. J R Stat Soc Ser B Methodol. 1974;36(2):111-133.

[32] Refaeilzadeh P, Tang L, Liu H. Cross-validation. In: Liu L, ÖZsu MT, editors. Encyclopedia of database systems. 1st ed. Massachusetts: Springer US; 2009. p. 532-538.

[33] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York: Springer; 2009.

[34] Chinchor N. MUC-4 evaluation metrics. The 4th Conference on Message understanding (MUC-4); 1992 Jun 16-18; Virginia, United States. California: Morgan Kaufmann Publishers, Inc.; 1992. p. 22-29.

[35] Sasaki Y. The truth of the F-measure [Internet]. [accessed 2021 May 26] Available from: https://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf.