Extraction of object image features with gradation contour

Fachruddin[1], Saparudin[2], Errissya Rasywir[3], Yovi Pratama[4], Beni Irawan[5]
[1]Department of Informatics Engineering, Faculty of Engineering, Universitas Sriwijaya, Palembang, Indonesia
[2]Department Informatics, Telkom University, Bandung, Indonesia
[3,4,5]Department of Informatics Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia

Article Info

Article history:

Received Jan 13, 2021
Revised Apr 23, 2021
Accepted May 2, 2021

Keywords:

Contour
Extraction
Feature
Gradation
Image

ABSTRACT

Image retrieval using features has been used in previous studies including shape, color, texture, but these features are lagging. With the selection of high-level features with contours, this research is done with the hypothesis that images on objects can also be subjected to representations that are commonly used in natural images. Considering the above matters, we need to research the feature extraction of object images using gradation contour. From the results of the gradation contour test results, there is linearity between the results of accuracy with the large number of images tested. Therefore, it can be said that the influence of the number of images will affect the accuracy of classification. The use of contour gradation can be accepted and treated equally in all image types, so there is no more differentiation between image features. The complexity of the image does not affect the method of extracting features that are only used uniquely by an image. From the results of testing the polynomial coefficient savings data as a result of the gradation contour, the highest result is 81.40% with the highest number of categories and the number of images tested in the category is also higher.

Corresponding Author:

Errissya Rasywir
Department of Informatics Engineering
Universitas Dinamika Bangsa
Jambi, Indonesia
Email: errissya.rasywir@gmail.com

1. INTRODUCTION

The digital image technology is currently growing rapidly along with the need for information acquisition in it such as the forensic field [1], remote sensing [2], and medicine [3]. Image recovery is an important part of image retrieval or content based image retrieval (CBIR) technology. Some CBIR approaches that have been taken include semantic [4], [5], relevance feedback [6], [7], machine learning or algorithmic approaches [8]–[11], hierarchical categories [12] and, multimodal queries (image retrieval with various types of queries) [13]. The task of detecting the number of objects contained in an image can be utilized in a variety of life applications, including detecting the number of objects in the fetal ultrasound image, detecting the number of vehicles passing a crossroads, counting the number of visitors in a shopping center as a recommendation for building repairs, counting the number of passengers in a fleet, count the number of audiences, count the number of oil palm bunches in each cubic truck and many other types of applications [14].

Previous research has stated that digital image retrieval using query text input is inadequate [15], [16]. Therefore, image retrieval developed into a search based on the features of the digital image. Search using features has been used in several previous studies including form features [17] where the study states that form features alone are not robust as features, but the formation of a shape feature descriptor can produce effective features, texture features [18] the experimental results show that by using fractional differential operators can extract finer features and detect more strongly [19]. The contour feature is a feature commonly used in representing face images and has not been tested in object images [20], [21]. And texture features are features that are commonly represented in natural images (such as medical images and fingerprints) [22]. The selection of high-level feature types with contours and textures is done with the hypothesis that images on objects can also be subjected to representations commonly used in natural images [17], [23]. Based on this, the study features contours [14], [24]–[26], these features are commonly used to represent features in biometric and natural images. Such a contour is a feature commonly used in representing face images and has not been tested in CBIR [21], [27], [28]. The texture is features commonly represented in natural images.

The gradation contour method is a new technique that has been used to recognize face images based on gradations of face color [17]. The problem of facial biometrics recognition that is quite heavy such as about lighting, expressions, and poses can be handled by this feature very successfully and quite economically, but this approach is very sensitive to light, perspective and poses, even in 3D model images [21], [29]. In this research, texture and gradation features will be combined to extract object image features. We use the gradation contour method for contour features. The selection of high-level features with contours done with the hypothesis that the image on the object can also be subjected to representations that are commonly used in natural images. Considering the above matters, the researcher feels the need to research the feature extraction of object images with gradation contour.

2. RESEARCH METHOD

The research framework describes the stages of the process carried out so that the research can run well and the objectives set can be achieved. In this study, the authors doing the identification of problems. At this stage, the authors formulate the problem. The problem was raised in the selection of methods in the contour feature extraction process using the gradation contour method. The selection of high-level features with contours is conducted with the hypothesis that the image on the object can also be subjected to representations that are commonly used in natural images.

The source of object image data used is Caltech 101. Based on this, this research tries to conduct experiments on the object image. The data used in the form of a photo image location with objects that are different from each other. In this research, we also create a system modeling plan for feature image feature extraction with gradation contour. The system modeling method used is the object-oriented approach method using unified modeling language (UML) tools. For system research development and experimentation, we developed a system using a gradation contour algorithm. This process includes designing the user interface, coding and testing. The system flow illustrated in the Figure 1.

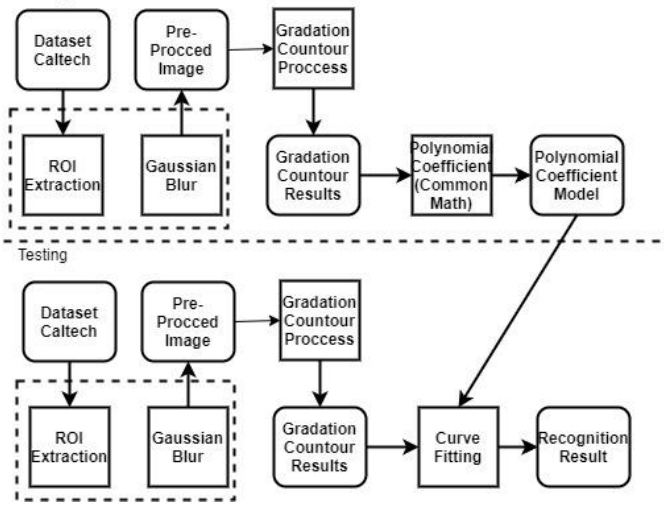


Figure 1. The flowchart in object image feature extraction system with gradation contour

In Figure 1 there is a flow chart of the feature image extraction system with gradation contour. There are two main stages, namely the training phase of dataset images and image testing. At each stage there is an image data input process an image region of interest (ROI) extraction process a Gaussian blur preprocess. After obtaining a contour gradation process is carried out on the preprocess results so that the image obtained from the contour gradation results. Then the resulting image is recalculated by the value of its image frequency as a model to be used again for the testing process. In the testing process module, the same thing is done except that the image used is the test image.

Gradation contour was previously only used for face recognition, by extracting contours of color gradation curves on human skin color [30]. Then, from the curve of polynomial equation coefficient, it is searched to find the contour curve match of the other faces. Hopefully, this approach can withstand the differences in light and pose. In this study, the gradation contour method is used to extract features in object images. The contour is a feature commonly used in representing face images and has not been tested on objects. The texture is feature commonly represented in natural images. The selection of high-level feature types with contours and textures conducted in the hypothesis that object images can also be recognized [31]. The gradation contour method is used to recognize face images based on gradations of face color. Changing an red, green, and blue (RGB) image to a grayscale image is an example of image processing using point operations. To convert an RGB image into a grayscale image is to calculate the average RGB intensity value of each pixel making up the image. The mathematical formula used is as (1) [13].

$$f_0(x,y) = \frac{f_i^R(x,y) + f_i^G(x,y) + f_i^B(x,y)}{3} \tag{1}$$

The value f_0 (x, y) is the frequency of the color intensity of each pixel in an image. The variable f^{iR} (x, y) is the red channel (Red (R)), f^{iG} (x, y) is the green channel (Green (G)), and f^{iB} (x, y) is the blue channel (Blue (B)). The grayscale image will be normalized (noise removal). This normalization process is used to standardize the intensity value of an image by adjusting the range of grayscale on pixel i, j. Normal image values are defined as (2).

$$N(i,j) = \begin{cases} M_0 + \sqrt{\frac{V_0(I(i,j)-M)^2}{V}} & if I(i,j) > M \\ M_0 - \sqrt{\frac{V_0(I(i,j)-M)^2}{V}} & Otherwise, \end{cases} \tag{2}$$

With a value of M: mean value; V: variance of the image (i, j); Mo: the desired mean value; Vo: the desired variance value. However, in this study, the value of Mo and Vo is worth 100 [21]. From the calculation results above, we get an image that has undergone normalization. A normalized image will not change the object's ridge structure because this normalization process is carried out to provide a dynamic standard of gray value variation of the image while also facilitating the next enhancement process.

2.1. Contour extraction based on color gradation

The technique for extracting contours is by utilizing the gradations of color found in each pixel in the image. Gradation taken is grayscale gradation. Because the image used in this study is a color image, it is necessary to convert from a color image to a grayscale image [32]. Figure 2 is the example of grayscale gradation. The Y-axis shows the number of pixel intencity on the matrice column in image and the X-axis shows the number of pixel intencity on the matrice row in image. In the grayscale image, the channel is just 1, where the number of color intencity depend on the amount of image binary digit. Following is an example of 1-pixel row in a grayscale image that will be extracted with a color contour.

Figure 3 is the example of 2-dimensional digital images. The number in the table is the pixel intencity on in 2-dimensional digital images. The value of intencity represents the value of grayscale image. The number of color intencity depend on the amount of image binary digit. In general, 2-dimensional digital images have 2 axes, namely X-axis and Y-axis. X-axis is for the horizontal line, and Y-axis for the vertical axis. We will add 1 more Z-axis, which is the value of each pixel at position (X, Y), so in Table 1 as the example, we will get the 3-dimensional points.

Table 1 is a 3-dimensional digital image representation that we used in this study. There are three axes namely X-axis, Y-axis and Z-axis. The number on the X-axis is data that represents the color intensity at the x-coordinate point of the image pixel. Likewise, the Y-axis is data that represents the color intensity at the Y-coordinate point in the image pixels. While the Z-axis is the intensity value which is represented as the contour of the image. To get the contour, we will use the X-axis and Z-axis, so that, if the points above are drawn as contours it will look like Figure 4. Dividing the contour into a down-up and up-down section is one of the works we do in doing gradation contour extraction. Once obtained, the next technique is to divide 1 contour line into several lines based on changes in direction. The contour lines in Figure 4 will be divided into several parts like.

Figure 4 is an example of a curve visualization from the results of the contour features performed. Figure 4 (a) visualizes the contour features in 3 dimensions. Figure 4 (b) is a visualization of the contour features taken from the curve of the equation that is used to be ready for segmentation so that the feature values can be stored. Figure 4 (c) is an example of a curve that has been segmented at each degree of slope of the curve line that forms a contour representation. The purpose of dividing it into sections is to make it easier to find the coefficient squared equation of the contour line. This contour collection is used as a feature.

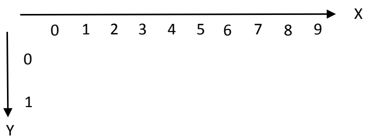


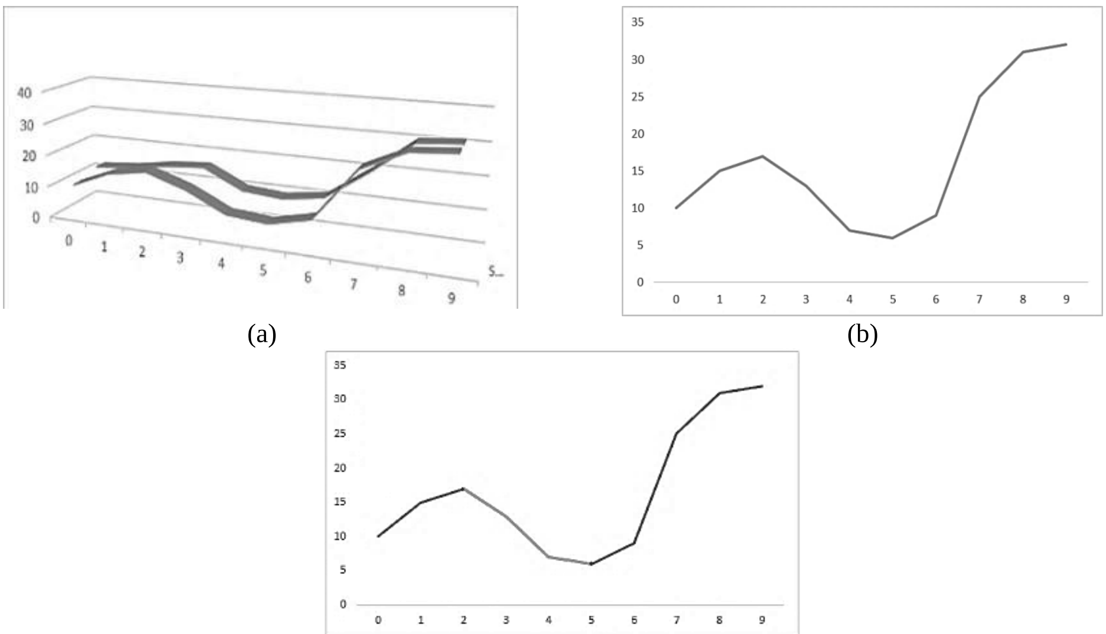
Figure 2. Example of grayscale gradation

10	15	17	13	7	6	9	25	31	32
12	13	15	16	10	9	11	20	30	31

Figure 3. Example of 2-dimensional digital images

Table 1. Example 3-dimensional digital image representation

X	Y	Z	X	Y	Z
0	0	10	0	1	12
1	0	15	1	1	13
2	0	17	2	1	15
3	0	13	3	1	16
4	0	7	4	1	10
5	0	6	5	1	9
6	0	9	6	1	11
7	0	25	7	1	20
8	0	31	8	1	30
9	0	32	9	1	31



(a) (b)

(c)

Figure 4. The example of visualization of color gradation contour for (a) 2-pixel rows, (b) before divided, and (c) after divided

2.2. Image recognition

In this section, the further process of segmentation results from the curve that has been formed in the previous section will be explained. The curves of the equations generated by strokes as contour features in the image will be stored their feature values. The feature value is derived from the equation coefficient value of the segmented curve. Furthermore, the feature values generated and stored as image contour features will be used subsequently to perform image recognition. To check whether two images are the same or not, a comparison between the contours that have been extracted in curve fitting between one contour and another.

Figure 5 is a visualization of the shape of the contour line that has been segmented and then analyzed for its slope. Segmented contour lines. In Figure 5 (a) is an example of a contour feature that contains a feature line that is skewed straight. In Figure 5 (b) is an example of a contour feature that contains feature lines that tend to be bent. Figure 5 has the following variables: X=(0, 1, 2) and Y=(10, 15, 17) and X=(0, 1, 2) and Y=(8, 14, 16). The first step taken is to find the polynomial coefficient of contour 1, the polynomial order used in this study from order 1 to 20. But the search will stop if the error has reached 0. The polynomial coefficient extraction process utilizes the Common-Math3-35 library using functions fit in "PolynomialCurveFitter" class. But before finding the coefficient, we must normalize the position of the two contours, so that both are at point 0, by the way, all points (X and Y) are reduced by the first point (X and Y).

$X_i = X_i - X_1$, $Y_i = Y_i - Y_1$,

contour 1 $\rightarrow$ X = (0, 1, 2) $\rightarrow$ X = (0, 1, 2)
Y = (10, 15, 17) $\rightarrow$ Y = (0, 5, 7)

contour 2 $\rightarrow$ X = (0, 1, 2) $\rightarrow$ X = (0, 1, 2)
Y = (8, 14, 16) $\rightarrow$ Y = (0, 6, 8)

Then look for the Contour coefficient 1. Here the results coefficient for each order:

X = (0, 1, 2), Y = (0, 5, 7)
Ordo 1 $\rightarrow$ 3.5x + 0.5, error=0.6661
Ordo 2 $\rightarrow$ $-1.5x^2 + 6.5x + -7.39E-32$, error=2.46E-32;
Ordo 3 $\rightarrow$ $-0.5x^2 + 0.0x^2 + 5.5x + -2.95E-31$, error=9.86E-32
Ordo 4 $\rightarrow$ $-0.214x^4 + 0.0x^3 + 0.0x^2 + 5.21x + 0.0$, error= 0.0

So, the best fit polynomial with contour 1 is -0.214x4 + 0.0x3 + 0.0x2 + 5.21x + 0.0. Then we want to prove how similar Contour 2 and Contour 1. Then, enter the value of X in contour 2 into the polynomial equation for contour 1.

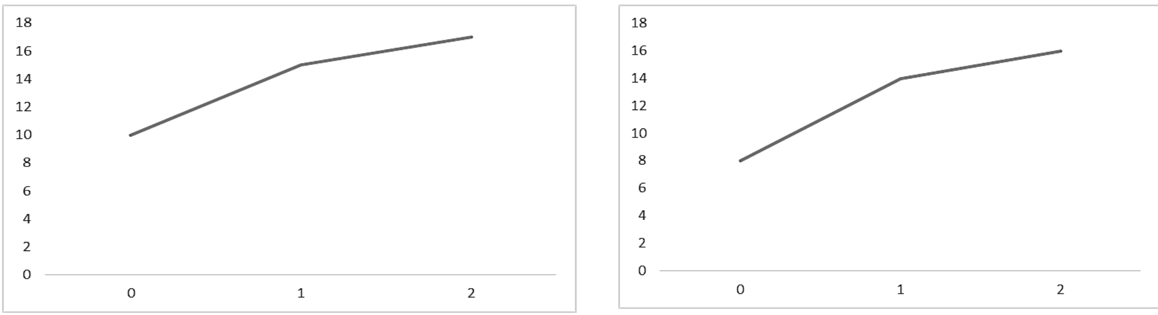
Contour Coefficient 2
X = (0,1,2), Y = (0,6,8)
f(X), X = 0 $\rightarrow$ $-0.214(0)^4 + 0.0(0)^3 + 0.0(0)^2 + 5.21(0) + 0.0 = 0$.
f(X), X = 1 $\rightarrow$ $-0.214(1)^4 + 0.0(1)^3 + 0.0(1)^2 + 5.21(1) + 0.0 = 4.9996$.
f(X), X = 2 $\rightarrow$ $-0.214(2)^4 + 0.0(2)^3 + 0.0(2)^2 + 5.21(2) + 0.0 = 8.708$;

Then we will look for errors using the formula (3).

$$err = \frac{\sum_{i=1}^{n} |y_i - f(x_i)|}{n} \tag{3}$$

Err = (|0 – 0| + |6-4.9996| + |8-8.708|)/3=0.0974

These steps are performed on all contours that have been successfully extracted, then voted for all contours that have errors below the threshold error. In the field of checking the accuracy of the results two terms commonly appear namely precision and recall. These two terms are the terms that emerge when the system created has been able to display the results (retrieve) a result in the form of classification, precision, or recall [33], [34]. From this value, it will be stored as a feature value of the image contour that can be used for pattern recognition or image classification.



(a) (b)

Figure 5. The example of contour line in gradation contour for (a) tend to straight, (b) tend to turn

3. RESULTS AND ANALYSIS
3.1. Code analysis of gradation contour algorithm program

This section describes the implementation of contour feature extraction on the image. The description of the feature extraction process with contour features in the image is explained through the program code lines. By using the program line code, an explanation of the contour features at once is included with the algorithm that we propose based on our image architecture. In the following section, we present the results and analysis of the gradation contour algorithm program code:

```
import imageio
import matplotlib.pyplot as plt
import os
from skimage import data
from skimage.color import rgb2gray
from numpy import *
import numpy as np
import warnings
import pylab
warnings.simplefilter('ignore',
np.RankWarning)
np.seterr(divide='ignore',
invalid='ignore')
dir = "./drive/My Drive/caltech/images"
for label in os.listdir(dir):
  dirimg = dir + "/" + label
  for img in os.listdir(dirimg):
    f, ext = os.path.splitext(dirimg +
"/"+img)
    if ext == ".poly": continue
    pic   =   imageio.imread(dirimg   +
"/"+img)
    height = pic.shape[0]
    width = pic.shape[1]
    pic = rgb2gray(pic)
poly = "";
    length = int(width/20)
    for i in range(height):
      for j in range(length):
        y = pic[i,j*length:(j*length)+length]
        x = [k for k in range(y.size)]
        p10 = polyfit(x,y,5).round(7)
        # plot = poly1d(p10)
        # xx = linspace(0, 100, 100)
        #  pylab.plot(x,  y,  'o',  xx,  plot(xx),'-
b')
        # pylab.legend(['data to fit', str(5)+'th
degree poly'])
        # pylab.axis([0,10,-1,2])
        # pylab.show()
        poly += " ".join(map(str,p10))+" "
      poly += "\n"
    with open(dirimg+"/"+img+".poly", 'w') as f:
      print(dirimg+"/"+img+".poly")
      f.write(poly)
```

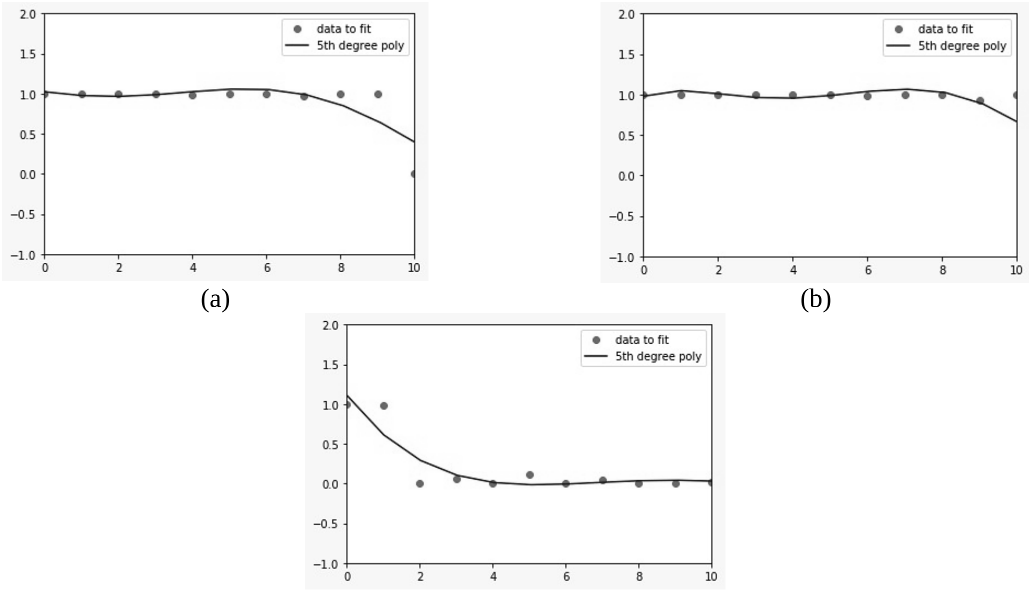
The part above is the program code gradation contour algorithm. The bar code is the result of a copy of the aggregate to perform contour gradation extraction as image feature data used in the object detection process. In the algorithm, there is a process of reading line sketches on images using polynomial function equations. The resulting polynomial degree cannot be determined, while the polynomial coefficient is stored as image contour reading data. A command with a code such as "import imageio" is used as a library call to input image output in the program. The command "import matplotlib.pyplot as plt" is used as a library calling the graphical user interface (GUI) that displays graphical results from polynomials that are searched for image data.

The command "import os from skimage import data" is used to call the library that extracts image pixel data, so that it can be calculated according to the similarity of values in the pixel image. The command "from skimage.color import rgb2gray" is the process of changing the value of the image intensity from a color image to an RGB image. The commands "from numpy import*", "import numpy as np", "import warnings" and "import pylab" are used for the reading of data numbers generated as contour extraction to be copied into polynomial equations or polynomial functions so that they can be visualized in curve ploting. From the contents of the program code is also declared dynamic variables that can adjust the needs of an image to accommodate pixel data to then be transformed into polynomial functions with varying degrees as well. The "poly" function is used as a command to construct polynomial functions from coefficient data that have been obtained from an image.

3.2. Image analysis of polynomial gradation contour image

In this section, we describe the analysis of curves generated by polynomials applied to an image. This polynomial is analyzed based on the coefficient values contained in each equation generated from the image. Each polynomial coefficient value becomes an indicator of the slope of the line curve.The following sections we present the results and analysis of the image polynomial gradation image analysis graph.

In Figure 6, a sample curve is used for one of the test images in this study. The curve of the polynomial produced in an image file under test can produce up to hundreds of polynomial curves. Every polynomial curve that has degrees up to tens with coefficient values also reaches tens of variations. So, from the results of the polynomial curve, the coefficient data of an image is generated and stored as data used for extraction features.



(a) (b)

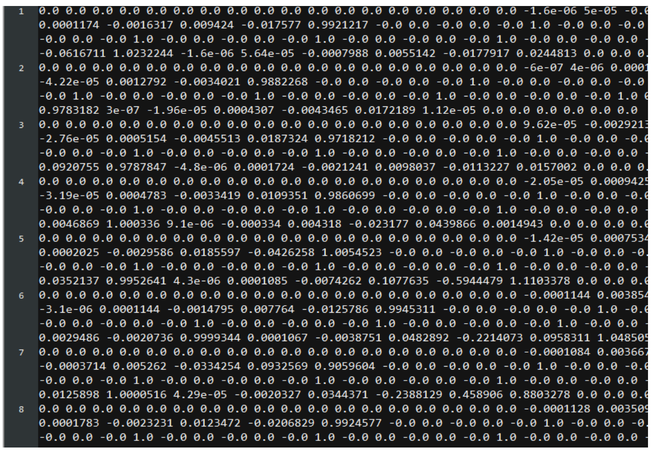
(c)

Figure 6. The polynomial curve results from image with the degrees 5 for (a) Type 1, (b) Type 2, and (c) Type 3

3.3. Image count polynomial gradation degree analysis

Analysis of the degree of the curve generated by the feature on the contour gradation is needed to show as a weight for the stored value of an image feature. In this case, the stored value used as the feature weight is the degree of the curve of the polynomial which is taken from the values of the coefficients in the polynomial equation applied to the image. This coefficient value is obtained from all images in the dataset. This value is then used as a dataset on the contour gradation feature. Here is a snippet of matrice containing the polynomial degree of an image.

Figure 7 is a sample of deposit results of polynomial coefficient data produced in a test image using a gradation contour image. It can be seen that the saved polynomial coefficient value data generated by an image, the result is a very large polynomial coefficient value. Therefore, this shows that image processing is very large and requires a high-performance computer. Meanwhile, the polynomial coefficient data structure is not limited in number and dynamic by the quality of the image used.



```
1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -1.6e-06 5e-05
  0.0001174 -0.0016317 0.009424 -0.017577 0.9921217 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0
  -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0
  -0.0616711 1.0232244 -1.6e-06 5.64e-05 -0.0007988 0.0055142 -0.0177917 0.0244813 0.0 0.0
2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -6e-07 4e-06
  -4.22e-05 0.0012792 -0.0034021 0.9882268 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0
  -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0
  0.9783182 3e-07 -1.96e-05 0.0004307 -0.0043465 0.0172189 1.12e-05 0.0 0.0 0.0 0.0 0.0 0.0
3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 9.62e-05
  -2.76e-05 0.0005154 -0.0045513 0.0187324 0.9718212 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0
  -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0
  0.0920755 0.9787847 -4.8e-06 0.0001724 -0.0021241 0.0098037 -0.0113227 0.0157002 0.0 0.0
4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -2.05e-05
  -3.19e-05 0.0004783 -0.0033419 0.0109351 0.9860699 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0
  -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0
  0.0046869 1.000336 9.1e-06 -0.000334 0.004318 -0.023177 0.0439866 0.0014943 0.0 0.0 0.0
5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -1.42e-05
  0.0002025 -0.0029586 0.0185597 -0.0426258 1.0054523 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0
  -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0
  0.0352137 0.9952641 4.3e-06 0.0001085 -0.0074262 0.1077635 -0.5944479 1.1103378 0.0 0.0
6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.0001144
  -3.1e-06 0.0001144 -0.0014795 0.007764 -0.0125786 0.9945311 -0.0 0.0 -0.0 0.0 -0.0 1.0
  -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0
  0.0029486 -0.0020736 0.9999344 0.0001067 -0.0038751 0.0482892 -0.2214073 0.0958311
7 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.0001084
  -0.0003714 0.005262 -0.0334254 0.0932569 0.9059604 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0
  -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0
  0.0125898 1.0000516 4.29e-05 -0.0020327 0.0344371 -0.2388129 0.458906 0.8803278 0.0 0.0
8 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -0.0001128
  0.0001783 -0.0023231 0.0123472 -0.0206829 0.9924577 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0
  -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0 -0.0 1.0 -0.0 0.0 -0.0 0.0
```

Figure 7. Samples of polynomial coefficient data deposit results produced in a test image

3.4. Analysis of image classification results with gradation contour

In this part, we analyze the image classification using a gradation contour. The number of images used amounts to thousands of 101 categories of images consisting of living and inanimate objects. All image objects that are used are very complex in features, there are also variations in the background that help increase the complexity of the image object to be detected. In general, each object image has a very unique image representation representing the uniqueness of the object's image. Like, in general, face images that usually use the golden ratio feature, the distance between eyes, eye position, nose distance, and other typical facial image calculations. However, in this Caltech 101 image dataset, there is no more differentiation between image features. With the use of contour gradation, all types of images can be received and treated equally. In the following section, we present the results and analysis of the degree polynomial gradation contour image.

Figure 8 is the sample distribution of image polynomial coefficients with their class. The class of each instance is the category label of 101 categories provided by Caltech 101 data. Each instance of the data is the coefficient value of the polynomial in an image. The value of polynomial forms a curved or graph with various angles and slopes. The data coefficient value of zero or 0 shows that the curve contained by the polynomial of an image is a linear or straight-line curve. However, if the value passes or is below zero, it means that the curve has a slope. Here are the log run results from dataset with coefficient content on polynomials.

1	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	class
2	0	0	0	0	0	0	0	0	0	0	0	cellphone
3	0	0	0	0	0	0	0	0	0	0	0	cellphone
4	0	0	0	0	0	0	0	0	0	0	0	cellphone
5	0	-1E-06	1.7E-05	-0.00026	0.00252	-0.01494	0.05465	-0.11681	0.12954	-0.05508	0.00004	cellphone
6	0	1E-06	-0.00004	0.0007	-0.00756	0.05167	-0.22042	0.55537	-0.72792	0.35887	0.98835	cellphone
7	0	0	0	0	0	0	0	0	0	0	1	cellphone
8	0	0	0	0	0	0	0	0	0	0	1	cellphone
9	0	0	0	0	0	0	0	0	0	0	1	cellphone
10	0	0	0	0	0	0	0	0	0	0	1	cellphone
11	0	0	0	0	0	0	0	0	0	0	1	cellphone
12	0	0	0	0	0	0	0	0	0	0	1	cellphone
13	0	0	0	0	0	0	0	0	0	0	1	cellphone
14	0	-1.3E-05	0.00037	-0.00575	0.05359	-0.30935	1.09299	-2.24692	2.39257	-0.979	1.00009	cellphone
15	0	0	-4E-06	4.2E-05	-0.00013	-0.00169	0.01931	-0.08226	0.1589	-0.1203	0.02731	cellphone
16	0	0	0	0	0	0	0	0	0	0	0	cellphone
17	0	0	0	0	0	0	0	0	0	0	0	cellphone
18	0	0	0	0	0	0	0	0	0	0	0	cellphone
19	0	0	0	0	0	0	0	0	0	0	0	cellphone
20	0	0	0	0	0	0	0	0	0	0	0	cellphone
21	0	0	0	0	0	0	0	0	0	0	0	cellphone

Figure 8. Sample distribution of image polynomial coefficients with their class

```
1 === Run information ===
2 Scheme:weka.classifiers.bayes.NaiveBayes
3 Relation:     filteredimagescaltech (1)
4 Instances:    150960
5 Attributes:   12
6             f1, f2, f3, f4, f5, f6, f7, f8, f9,
f10, f11, class
7 Test mode:evaluate on training data
8 === Classifier model (full training set) ===
9 Naive Bayes Classifier;                  Class
10 Attribute      cellphone     lotus  mandolin
platypus     chair
11                    (0.17)     (0.19)     (0.28)
(0.16)      (0.2)
12 f1:   mean                  0          0          0
0          0
13   std. dev.            0          0          0
0          0
14  weight sum        25760      28280      42080
24840      30000
15   precision            0          0          0
0          0
16  f2:   mean                  0          0
0          0          0
17  std. dev.        0.0001          0          0
0.0001    0.0001
18  weight sum        25760      28280      42080
24840      30000
19  precision            0          0          0
0          0
44 f9:   mean              -0.026
0.0024    0.0488    0.0699   -0.0022
45  std. dev.         5.0184
2.4544    1.4347    8.9157    3.7127
46  weight sum         25760
28280     42080     24840     30000
47  precision         0.0014
0.0014    0.0014    0.0014    0.0014
48 f10:   mean              0.0224
0.0094   -0.0209   -0.0275    0.0123
49   std. dev.           1.934
0.9987    0.6721    3.5542     1.475
50  weight sum         25760
28280     42080     24840     30000
51  precision         0.0006
0.0006    0.0006    0.0006    0.0006
52 f11:   mean              0.5214
0.4354    0.5135    0.4212    0.6012
53  std. dev.         0.3563
0.3183    0.3751    0.2941    0.3453
54  weight sum         25760
28280     42080     24840     30000
55  precision              0
0         0         0         0
56 Time taken to build model: 2.05
seconds
57
58 === Evaluation on training set
===
59 === Summary ===
```

```
  f3:   mean                 0         0
0         0         0
 std. dev.        0.0024    0.0007    0.0001
0.0027     0.001
  weight sum        25760     28280     42080
24840     30000
  precision            0         0         0
0         0
  f4:   mean            0.0001   -0.0001   -
0.0001   -0.0001         0
  std. dev.       0.0286    0.0084     0.002
0.0355    0.0133
  weight sum        25760     28280     42080
24840     30000
  precision            0         0         0
0         0
  f5:   mean          -0.0011    0.0002    0.0008
0.0008   -0.0003
   std. dev.        0.2118    0.0692    0.0207
0.2865    0.1091
   weight sum        25760     28280     42080
24840     30000
  precision       0.0001    0.0001    0.0001
0.0001    0.0001
 f6:   mean            0.0051   -0.0007   -0.0048
-0.0056   0.0012
  std. dev.       0.9948    0.3674    0.1317
1.461    0.5751
  weight sum        25760     28280     42080
24840     30000
  precision       0.0004    0.0004    0.0004
0.0004    0.0004
 f7 :   mean            -0.015    0.0013
0.0188    0.0227   -0.0025
   std. dev.        2.9434    1.2169    0.5172
4.6592    1.8883
 weight sum       25760     28280     42080
24840     30000
  precision       0.0009    0.0009    0.0009
0.0009    0.0009
 f8 :  mean            0.0267    -0.002   -0.0424
-0.0542    0.0024
  std. dev.       5.2387    2.3813    1.1948
8.8463    3.6547
  weight sum        25760     28280     42080
24840     30000
  precision       0.0016    0.0016    0.0016
0.0016    0.0016
```

```

 Correctly Classified Instances
48048                78.8283 %
 Incorrectly Classified Instances
102912               22.1717 %
 Kappa statistic
0.0811
 Mean absolute error
0.2724
 Root mean squared error
0.5118
 Relative absolute error
86.0479 %
 Root relative squared error
128.6221 %

 Total Number of Instances
150960
 === Detailed Accuracy By Class
===
 TP Rate   FP Rate   Precision
Recall  F-M   ROC Area  Class:
 0.019     0.034      0.101
0.019     0.032      0.444
cellphone
 0.121     0.103      0.213
0.121     0.154      0.588    lotus
 0.916     0.745      0.322
0.916     0.476      0.607
mandolin
 0.204     0.029      0.585
0.204     0.303      0.692
platypus
 0.018     0.01       0.305
0.018     0.034      0.564    chair
 Weighted Avg.    0.318     0.24
0.304     0.318     0.224      78
0.581
 === Confusion Matrix ===
     a     b     c     d     e
<-- classified as
   483  2184 21896   952   245 |
a = cellphone
   846  3422 22932   767   313 |
b = lotus
   228  2962 38529   116   245 |
c = mandolin
  1819  3159 14356  5069   437 |
d = platypus
  1396  4373 21924  1762   545 |
e = chair
```

The result of the log run from the dataset with the coefficient content on the polynomial which was tested using one of the algorithms as an evaluation of the results of the gradation contour feature extraction. The log shows the results of the calculation of the probability of the main class of all classes and the probability of each category of each attribute to the various types of main classes. The accuracy and confusion matrix results of the test are also displayed. The following Table 2 and Table 3 are the results of an evaluation of the tests conducted on the results of feature extractions performed on the Caltech 101 dataset.

Figure 9 is a graphical visualization of Table 2. The data is the result of a comparison of correctly classified with the same number of images testing that is equal to 10 images from each variation of testing categories. Good when tested with 10 categories, which means that 10x10 images have 100 images, then 30x10 images, 300 images, and 50 categories with 50x10 images, there are 500 images. From the results of the gradation contour testing results obtained are not yet stable. It cannot be ascertained the influence of the number of images will affect the accuracy of classification.

Figure 10 is a graphical visualization from Table 3. The data is the result of a comparison of correctly classified by testing the number of different images namely variations of 10 images, 30 images, and 50 images according to the number of categories tested. Based on the results of variations in category testing, both when tested with 10 categories, which means that 10x10 images have 100 images, then 30x30 images, 900 images, and 50 categories with 50x50 images, there are 2,500 images. From the results of the gradation contour test results obtained results that indicate the linearity between the results of accuracy with the large number of images tested. Therefore, it can be said that the influence of the number of images will affect the accuracy of classification.

Table 2. Correctly classified by testing the same number of imagery

Category	Image/Category	Correctly Classified
10 Categories	10	78.98%
30 Categories	10	79.03%
50 Categories	10	78.98%

Table 3. Correctly classified by testing the different number of imagery

Category	Image/Category	Correctly Classified
10 Categories	10	78.98%
30 Categories	10	80.21%
50 Categories	10	81.40 %

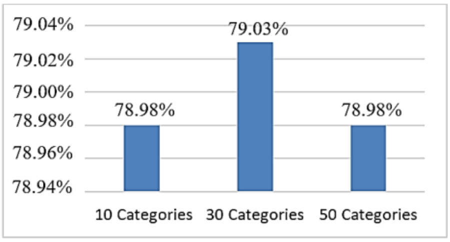


Figure 9. Correctly classified by testing the same image amount

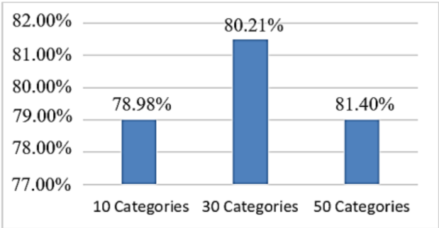


Figure 10. Correctly classified by testing the amount of different imagery

4. CONCLUSION

With the gradation contour algorithm, the coefficient value of the polynomial in an image forms a curve or curve graph with various angles and slopes. The data coefficient value of zero or 0 shows that the curve contained by the polynomial of an image is a linear or straight-line curve. However, if the value passes or is below zero, it means that the curve has a slope. From the results of the gradation contour test results obtained results that indicate the linearity between the results of accuracy with the large number of images tested. Therefore it can be said that the influence of the number of images will affect the accuracy of classification.

The use of contour gradation can be accepted and treated equally in all image types so there is no more differentiation between image features. The complexity of the image does not affect the method of extracting features that are only used uniquely by an image. From the results of testing the polynomial coefficient savings data as a result of gradation contour, the highest result is 81.40% with the highest number of categories and the number of images tested in the category is also higher. As a suggestion for the future, this research needs to be developed by testing the dataset and other types of algorithms. Also, this research requires an increase in computational performance, because the results are now still costly computing costs and the need for algorithmic experiments so that results increase.

ACKNOWLEDGEMENTS

Thank you to the Dinamika Bangsa Foundation for material and moral support in conducting this research so that it should run smoothly.

REFERENCES

[1] F. Mayer and M. Steinebach, "Forensic Image Inspection Assisted by Deep Learning," *Proc. 12th Int. Conf. Availability, Reliab. Secur. - ARES '17*, 2017, pp. 1–9, doi: 10.1145/3098954.3104051.

[2] Y. Wang *et al.*, "EV-gait: Event-based robust gait recognition using dynamic vision sensors," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, 2019, pp. 6351–6360, doi: 10.1109/CVPR.2019.00652.

[3] H. A. Jalab and R. W. Ibrahim, "Texture enhancement for medical images based on fractional differential masks," *Discret. Dyn. Nat. Soc.*, vol. 2013, 2013, doi: 10.1155/2013/618536.

[4] P. Nalini, "Review on Content Based Image Retrieval : From Its Origin to the New Age," *Int. J. Res. Stud. Sci. Eng. Technol.*, vol. 3, no. 2, pp. 18–41, 2016, [Online]. Available: http://www.ijrsset.org/pdfs/v3-i2/4.pdf

[5] A. M. U. Ahamed, C. Eswaran, and R. Kannan, "CBIR system based on prediction errors," *J. Inf. Sci. Eng.*, vol. 33, no. 2, pp. 347–365, 2017, doi: 10.1688/JISE.2017.33.2.5.

[6] B. P. S. S. Ashish Mohan Yadav, "A Survey On: 'Content Based Image Retrieval Systems,'" *Int. J. Emerg. Technol. Adv. Eng.*, vol. 4, no. 6, pp. 22–26, 2014.

[7] N. Goyal and N. Singh, "A Review on Different Content Based Image Retrieval Techniques Using High Level Semantic Features," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 7, pp. 4933–4938, 2014.

[8] Y. Pratama and E. Rasywir, "Automatic Cost Estimation Analysis on Datawarehouse Project with Modified Analogy Based Method," *Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018*, 2019, pp. 171–176, doi: 10.1109/ICECOS.2018.8605195.
[9] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, no. December 2016, pp. 11–26, 2017, doi: 10.1016/j.neucom.2016.12.038.
[10] N. Rusk, "Deep learning," *Nat. Methods*, vol. 13, no. 1, p. 35, 2015, doi: 10.1038/nmeth.3707.
[11] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and J. Frederick C. Harris, "Survey of image denoising techniques," *Life Sci. J.*, vol. 10, no. 1, pp. 753–755, 2013, doi: 10.1.1.107.1823.
[12] M. Jain and S. K. Singh, "A Survey On: Content Based Image Retrieval Systems Using Clustering Techniques For Large Data sets," *Int. J. Manag. Inf. Technol.*, vol. 3, no. 4, pp. 23–39, 2011, doi: 10.5121/ijmit.2011.3403.
[13] K. T. Ahmed, S. A. H. Naqvi, A. Rehman, and T. Saba, "Convolution, approximation and spatial information based object and color signatures for content based image retrieval," *2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019*, 2019, pp. 1–6, doi: 10.1109/ICCISci.2019.8716437.
[14] S. Fadaei, R. Amirfattahi, and M. R. Ahmadzadeh, "New content-based image retrieval system based on optimised integration of DCD, wavelet and curvelet features," *IET Image Process.*, vol. 11, no. 2, pp. 89–98, 2017, doi: 10.1049/iet-ipr.2016.0542.
[15] B. Demir and L. Bruzzone, "A Novel Active Learning Method in Relevance Feedback for Content-Based Remote Sensing Image Retrieval," *Geosci. Remote Sensing, IEEE Trans.*, vol. 53, no. 5, pp. 2323–2334, 2015, doi: 10.1109/TGRS.2014.2358804.
[16] A. Kumar, F. Nette, K. Klein, M. Fulham, and J. Kim, "A Visual Analytics Approach Using the Exploration of Multidimensional Feature Spaces for Content-Based Medical Image Retrieval," *IEEE J. Biomed. Heal. Informatics*, vol. 19, no. 5, pp. 1734–1746, 2015, doi: 10.1109/JBHI.2014.2361318.
[17] I. Supriana and Y. Pratama, "Face Recognition New Approach Based on Gradation Contour of Face Color," *International Journal on Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 125–138, 2017, doi: 10.15676/ijeei.2017.9.1.8.
[18] H. A. Jalab and R. W. Ibrahim, "Texture enhancement based on the Savitzky-Golay fractional differential operator," *Math. Probl. Eng.*, vol. 2013, 2013, doi: 10.1155/2013/149289.
[19] R. S. Patil and A. J. Agrawal, "Content-Based Image Retrieval Systems: A Survey," *Adv. Comput. Sci. Technol.*, vol. 10, no. 9, pp. 2773–2788, 2017.
[20] G. Ghazaei, A. Alameer, P. Degenaar, G. Morgan, and K. Nazarpour, "Deep learning-based artificial vision for grasp classification in myoelectric hands," *J. Neural Eng.*, vol. 14, no. 3, 2017, doi: 10.1088/1741-2552/aa6802.
[21] Fachruddin, E. Rasywir, Hendrawan, Y. Pratama, D. Kisbianty, and M. R. Borroek, "Real Time Detection on Face Side Image with Ear Biometric Imaging Using Integral Image and Haar- Like Feature," *2018 Int. Conf. Electr. Eng. Comput. Sci.*, 2018, pp. 165–170, doi: 10.1109/ICECOS.2018.8605218.
[22] K. T. Ahmed, S. Ummesafi, and A. Iqbal, "Content based image retrieval using image features information fusion," *Information Fusion*, vol. 51, pp. 76–99, 2019, doi: 10.1016/j.inffus.2018.11.004.
[23] F. Battistone and A. Petrosino, "TGLSTM: A time based graph deep learning approach to gait recognition," *Pattern Recognit. Lett.*, vol. 126, pp. 132–138, 2019, doi: 10.1016/j.patrec.2018.05.004.
[24] Y. Zhang, D. Xiao, W. Wen, and Y. Tian, "Edge-based lightweight image encryption using chaos-based reversible hidden transform and multiple-order discrete fractional cosine transform," *Opt. Laser Technol.*, vol. 54, pp. 1–6, 2013, doi: 10.1016/j.optlastec.2013.04.029.
[25] S. J. McKenna and J. Han, "Query-dependent metric learning for adaptive, content-based image browsing and retrieval," *IET Image Process.*, vol. 8, no. 10, pp. 610–618, 2014, doi: 10.1049/iet-ipr.2013.0514.
[26] A. K. Dhara, S. Mukhopadhyay, A. Dutta, M. Garg, and N. Khandelwal, "Content-Based Image Retrieval System for Pulmonary Nodules: Assisting Radiologists in Self-Learning and Diagnosis of Lung Cancer," *J. Digit. Imaging*, vol. 30, no. 1, pp. 63–77, 2017, doi: 10.1007/s10278-016-9904-y.
[27] A. Kamilaris and F. X. P. Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, no. July 2017, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.
[28] A. D. Egorov, A. N. Shtanko, and P. E. Minin, "Selection of Viola–Jones algorithm parameters for specific conditions," *Bull. Lebedev Phys. Inst.*, vol. 42, no. 8, pp. 244–248, 2015, doi: 10.3103/s1068335615080060.
[29] A. Borji, S. Izadi, and L. Itti, "iLab-20M: A Large-Scale Controlled Object Dataset to Investigate Deep Learning," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2221–2230, doi: 10.1109/CVPR.2016.244.
[30] Erwin, Saparudin, M. Fachrurrozi, A. Wijaya, and M. N. Rachmatullah, "New optimization technique to extract facial features," *IAENG Int. J. Comput. Sci.*, vol. 45, no. 4, pp. 523–530, 2018.
[31] Y. Zhang and Z. Mu, "Ear detection under uncontrolled conditions with multiple scale faster Region-based convolutional neural networks," *Symmetry (Basel).*, vol. 9, no. 4, 2017, doi: 10.3390/sym9040053.
[32] L. Bruno, G. Parla, and C. Celauro, "Image analysis for detecting aggregate gradation in asphalt mixture from planar images," *Constr. Build. Mater.*, vol. 28, no. 1, pp. 21–30, 2012, doi: 10.1016/j.conbuildmat.2011.08.007.
[33] P. Liu, J. M. Guo, C. Y. Wu, and D. Cai, "Fusion of deep learning and compressed domain features for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5706–5717, 2017, doi: 10.1109/TIP.2017.2736343.
[34] J. Ramos *et al.*, "Content-Based Image Retrieval by Metric Learning From Radiology Reports: Application to Interstitial Lung Diseases," *IEEE J. Biomed. Heal. Informatics*, vol. 20, no. 1, pp. 281–292, 2016, doi: 10.1109/JBHI.2014.2375491.