Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

Implementation of ColorSpace, GrabCut, and Watershed Methods on Digital Image Segmentation of Coral and Fish Objects

Aditya Ahmad Fauzi[1)*], Adisuputra[2)]
[1)2)]STIE Pertiba Pangkalpinang, Indonesia
[1)]aditya.a.fauzi23@gmail.com, [2)]saputra.as@gmail.com

ABSTRACT

Poor coral reefs rising in eastern Indonesia. The colors have disappeared from the seafloor instead, bleached branches are visible. To recognize the differences between dead and healthy coral reefs, an identification system has been created using the image processing method. Object segmentation is a step in digital image processing to separate one object from another based on specific characteristics. In this study, coral objects with various colored backgrounds and things became a problem to separate, so this study aimed to separate these various colors. This research uses color space segmentation to visualize RGB and HSV colors, Grabcut segmentation to separate the largest corals, and watershed segmentation to separate dead corals. Therefore, from this study, the RGB and HSV color visualizations were clearly visible. From Grabcut segmentation, it is found that the largest fish is detected and can be displayed in the segmentation results. At the same time, the watershed segmentation displays dead coral taken by gray segmentation with otsu.

Keywords: Image processing, color space segmentation, grabcut segmentation, watershed segmentation, coral.

1. INTRODUCTION

Not everyone realizes that life under the sea, especially coral reefs which are home to fish, is currently slowly dying, the results of LIPI's research, the category of bad coral reefs has increased in the eastern Indonesian region, the colors have disappeared from the sea floor instead , visible whitened branches (Dewi, 2007). This is the author's rationale for building a system that can identify living and dead corals. The more advanced the development of the world of IT (Information Technology) makes our daily lives easier, in this case, the author utilizes image processing methods to build a system that can identify dead corals and healthy corals.

Segmentation is the process of separating one object from another in an image into objects based on certain characteristics. The segmentation process stops when the object being sought has been found. Based on its understanding, segmentation has the goal of finding the special characteristics of an image. Therefore, segmentation is needed in the pattern recognition process. The better the segmentation quality, the better the pattern recognition quality.

The image segmentation process aims to separate the object (foreground) from the background. In general, the output of image segmentation results is in the form of a binary image where the object (foreground) you want is white (1), while the background you want to remove is black (0). As with the process of improving image quality, the image segmentation process is also experimental, subjective, and depends on the goals to be achieved (Yunus, 2019).

Image segmentation algorithms are generally based on one of two intensity value properties, namely discontinuity and similarity. Discontinuity has an approach of breaking or sorting out images based on sudden or large changes in intensity. The discontinuity segmentation process includes point detection, line detection, and edge detection. Meanwhile, the similarity is based on splitting the image into the same region according to several predetermined criteria, such as processes: thresholding region growing and region splitting and merging.

On the basis of this research, the authors have reviewed various related journals and articles, and research that has been conducted under the title Segmentation of Fish Digital Image Using Thresholding Method has succeeded in separating fish eye objects with a value of T3 = 61. Eye objects are in matrix coordinates [274:295,152:173]. (Rindengan & Mananohas, 2017).

Whereas the previous research entitled Object Segmentation in Digital Image Using the Otsu Thresholding

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

Method was successful in conducting trials, of the three trials each one was taken with good image segmentation results, namely when using the values L=265 and i=0, 1, 2, …, L-1. In the process of inverting images, the remover is better. The last is to use a threshold value of 150 in the noise removal process (Syafi'i et al., 2016).

Digital Image Segmentation of Leaf Shapes in Plants at Samarinda Politani Using the Thresholding Method, based on the results of the research that has been done that papaya leaves are classified as a type of finger leaf bone shape. These results are obtained from the results of LoG and Histogram operations which have in common with the example of the finger leaf shape image (Maria et al., 2018).

In previous research conducted under the title Digital Image Segmentation Using Otsu Thresholding for Comparative Analysis of Edge Detection, it can be concluded that it can give thin and smooth edges that cannot be detected by other methods. In addition, canny does not omit important information in the image. Meanwhile, Roberts requires very fast computing time compared to the others (Ambarwati et al., 2016).

In the previous research conducted under the title Object Detection Using Image Processing, it can be concluded that I concluded my project report by pointing out its use in surveillance and obstacle detection processes. In the future, this program can be used to control the cameras in a UAV and navigate through obstacles effectively. This program can be upgraded to reduce the processing time of the controller so that a different methodology can be tried and implemented. We also believe that visualizations can be a powerful tool for understanding object detection systems and advancing research in computer vision. To this end, we hope that more intuitive visualizations will prove useful for the community (Voronkov, n.d.).

2. LITERATURE REVIEW

2.1 Color Space Segmentation

In the process of analyzing images, sometimes we need processing only on a certain object in one frame, therefore we need a separation between the object we want and other objects in the frame. Image segmentation is useful for separating objects (foreground) and background. One method for performing image segmentation is to use Color Space.

The most common color space is RGB (Red Green Blue), where colors are represented by 3 main colors, namely red, green and blue. In more technical terms, RGB describes colors as tuples of three components. Each component can take values between 0 and 255, where the tuples (0, 0, 0) represent black and (255, 255, 255) represent white.

RGB is considered an "additive" color space, and color can be imagined as the result of bright amounts of red, blue, and green light hitting a black background. Here is an example of the colors and their RGB values:

Color	**RGB Value**
Red	255, 0, 0
Orange	255, 128, 0
Pink	255, 153, 255

RGB is one of the 5 most widely used color space types. For example, CMYK is widely used in printing, HED is widely used in the medical world to analyze glass slides which are mounted with stained tissue samples scanned and saved as images.

HSV and HSL are descriptions of hue, saturation, and brightness/luminance, which are very useful for identifying contrast in an image. This color space is often used in color selection tools in software and for web design. In the real world, color is a continuous phenomenon, meaning that there are an infinite number of colors. The color space, however, represents color through a discrete structure (a number of integer values), which is acceptable because the human eye and perception are also limited. The color space is fully representative of all the colors we can distinguish.

In the real world, color is a continuous phenomenon, meaning that there are an infinite number of colors. The color space, however, represents color through a discrete structure (a number of integer values), which is acceptable because the human eye and perception are also limited. The color space is fully representative of all the colors we can distinguish. The following is an RGB color space segmentation technique with the python repository, before doing this segmentation technique, make sure you have also installed the OpenCV python libraries, Matplotlib and Numpy (Kadir, 2019)

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

2.2 Grabcut Segmentation

In the segmentation process, it turns out that there is a GrabCut digital image processing method, which is a technique that separates an image into several regions based on certain characteristics. This method depends on the box that is considered as the background, while what is inside the box is unknown and the information outside the box will be used as a reference to delete anything in the box that has parts outside the box.

Grabcut was designed by Carsten Rother, Vladimir Kolmogorov & Andrew Blake, who worked at Microsoft Research Cambridge, UK. The implementation takes a long time to segment; longer than the watershed. The advantage is that the results are more accurate. Since it takes a long time, this method is only suitable for still images (not videos).

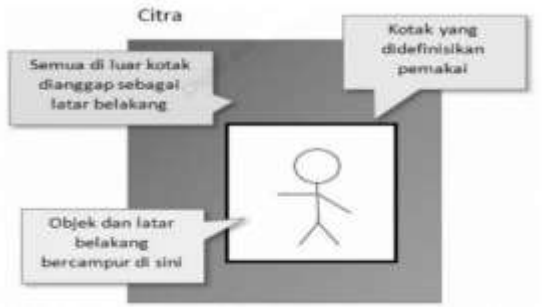

Figure 1. Grabcut Segmentation (Kadir, 2019)

GrabCut works on the basis of the "Graph Cut" algorithm and uses the Gaussian Mixture Model (GMM) to estimate the target object through color distribution. In this case, each pixel will be grouped based on its color distribution. The color distribution of each pixel is arranged in one graph. Based on this graph, the background and foreground determination is executed. OpenCV provides cv2. grabcut() to perform GrabCut operation. It looks like the following: cv2.grabcut(citra, cadar, kotak, modelLB, modelLD, jumlahiterasi, mode).

The first argument is an image in RGB format to be processed. The second argument is an array that will be filled with veils for object segmentation. The third argument specifies the box area for object segmentation processing. The fourth argument is an array used for background processing. The fifth argument is an array used for foreground processing. The sixth argument determines the number of iterations to get the object. The seventh argument contains the processing mode. In this case, it's cv2. GC-INIT-WITH-RECT for box processing (Achanya, T, & Ray, 2005).

2.3 Watershed Segmentation

Watershed is one of the algorithms used to separate a number of objects that are in the image. This algorithm works on the basis of markers on objects, which allows two objects that are stacked on top of each other to be separated. First of all, a marker for each object must be given. This can be done manually or through a computer. Then, based on each marker, which can be considered as a basin, flooding is carried out (towards the basin) so that the two boundaries of two objects that overlap or touch meet.

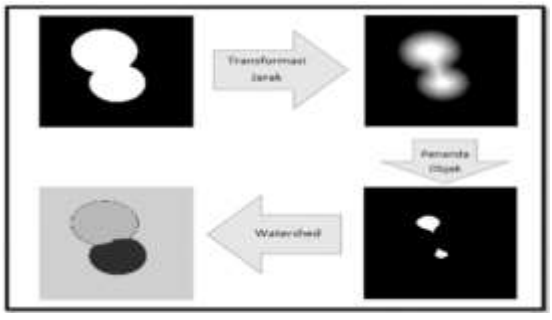

Figure 2. Separation process with watershed (Kadir, 2019)

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

The image above on the top-left shows a binary image of two overlapping circles. By using the distance transformation, the results are obtained as shown in the top-right. In order to mark both objects, a marker needs to be established. This can be done by applying a certain threshold value so that the results are obtained as shown in the image on the right below. Each of these markers is used as a starting point for the application of watersheds. The end result is visualized in the lower left image. It appears that the two objects can be separated.

Distance transformation is a process applied to binary images to calculate the distance of each pixel in the object area based on the closest boundary to the object. The image below shows an example of the distance transformation applied to a box object. The result is a grayish periodic image.

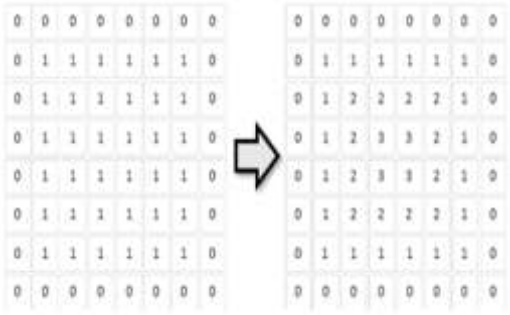

Figure 3. Distance transformation to binary image (Kadir, 2019)

The first argument is the source of the processed image. The second argument is the type of operator used to perform the distance transformation. The third argument is the size of the veil used to perform the distance transformation. The value of n represents the size of the veil of n x n. cv2 constant. DIST_MASK_3 and cv2. DIST_MASK_5 is provided in OpenCV. The return value is a two-dimensional array (Agoston, 2005).

2.4 OpenCV

Open Computer Vision (OpenCV) is an open-source library whose purpose is dedicated to performing image processing. The point is that computers have capabilities similar to the way visual processing in humans. OpenCV has provided many basic computers vision algorithms. OpenCV also provides an object detection module that uses the computer vision method (Zulkhaidi et al., 2020)

2.5 Python

Python is a high-level programming language that is interpreter, interactive, object-oriented, and can operate on almost all platforms: Mac, Linux, and Windows. Python is a programming language that is easy to learn because of its clear syntax, can be combined with the use of ready-to-use modules, and efficient high-level data structures. The Python distribution comes with a shell-like facility on Linux. The usual Python installation location is "/usr/bin/Python", and may vary. Running Python, simply by typing "Python", wait a moment then the ">>>" display appears, meaning Python is ready to receive commands. There is also a "…" sign that means the next line in a '>>>' prompt block. The text editor is used for script mode. To build this research used Python v.3.10 which is the Python programming language. And added some tools, like NumPy, Matplotlib, Scikit-image (Kadir, 2018).

2.6 Image Processing

Image processing or Image Processing is a system where the process is carried out with the input in the form of an image and the result (output) is also in the form of an image. Initially, image processing was carried out to improve image quality, but with the development of the computing world which is marked by the increasing capacity and speed of computer processing, as well as the emergence of computer sciences that allow humans to retrieve information from an image, image processing cannot be separated from the field of image processing. computer vision (Abdul Kadir, 2013).

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

3. METHOD

This research was carried out using the Visual Studio Code text editor with a python repository added to the OpenCV, Numpy, and Matplotlib libraries. As well as hardware in the form of an Intel i5 Processor, 8GB RAM, HDD, 500GB, 14 Inch Monitor, and 4 GB VGA. The data processed in this study are in the form of images that will be used, which are images of coral and fish objects as follows:

Figure 4. Coral object (Luthfi, O. M. i, 2020)

In the image Grabcut segmentation in the box determines the object to be obtained, namely coral as shown in the image below.

Figure 5. GrapCut object

In the watershed segmentation, images of dead corals will be separated or overlapping in various areas as shown in the image below..

Figure 6. Separating Corals

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

The next stage is the stage of making a design to process object data starting from data input to obtaining segmentation results from three image processing methods namely Colorspace, GrabCut, and Watershed using Python 3.10

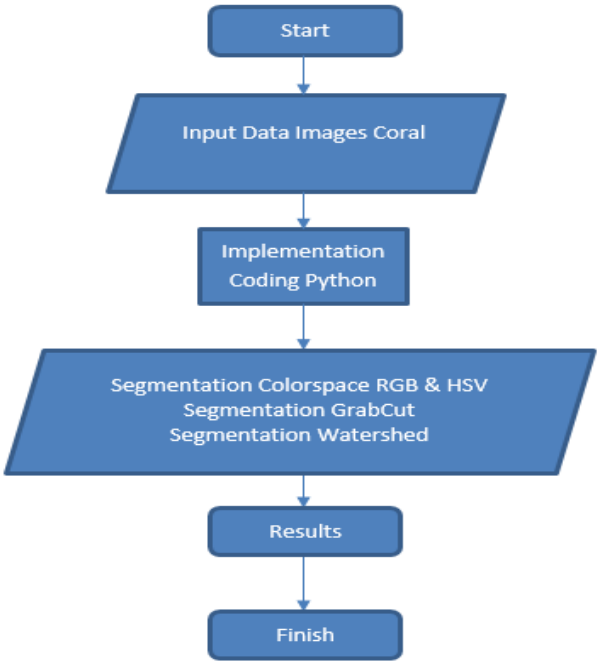

Figure 7. Research design

4. RESULT

4.1 Color Space RGB Segmentation

To process RGB color space segmentation, HSV color space, GrabCut, and Watershed, the author made it using the Python 3.10 programming language with the Visual Studio Code text editor by adding tools in the form of NumPy, Matplotlib, Scikit-image. The image below is the result of segmentation using the RGB color space method, showing various conditions before and after the segmentation of objects is carried out.

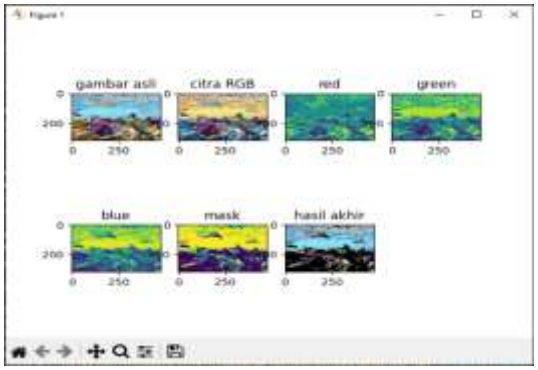

Figure 8. RGB color space segmentation results

4.2 Color Space HSV Segmentation

The image below is the result of segmentation using the HSV color space method which shows various conditions before and after segmentation of objects is carried out.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

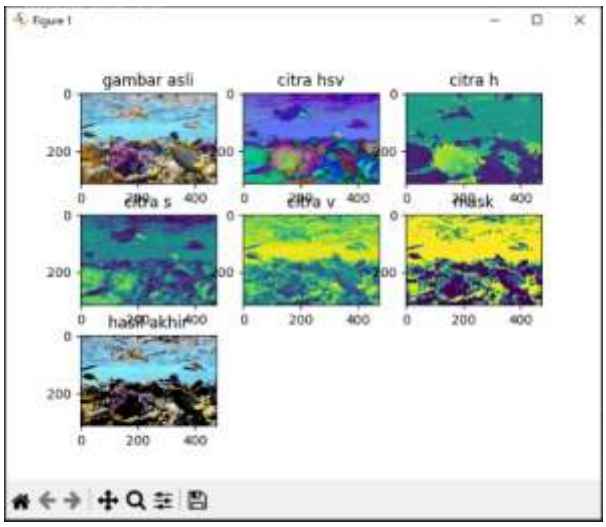

Figure 9. HSV Color space segmentation results

4.3 Grabcut Segmentation

The image below is the result of segmentation with the GrabCut method which shows various conditions before and after the object segmentation was carried out:

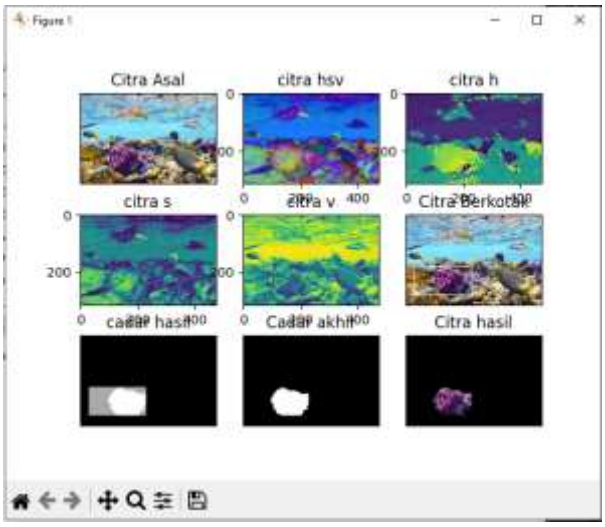

Figure 10. GrabCut Segmentation Results

4.4 Watershed Segmentation

In OpenCV, distance transformation can be obtained using cv2. distanceTransform(). The summon form is distance transform (image, distance type, Veil size). If segmentation is performed using the Otsu method, the results obtained are only one object as shown in the image below: (Tri Utami, 2017)

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

Figure 11. Otsu's method

Below are the results of his execution stating the number of people who died:

```
python.exe "d:/ADT/jurnal/jurnal image processing/watershed/segmentasi dengan
watershed.py"
jumlah area karang yang mati: 2 area
```

Figure 12. Watershed execution results

The following is an image of the results of segmentation with the Watershed method that is formed, indicating object markers, on each object a different color.

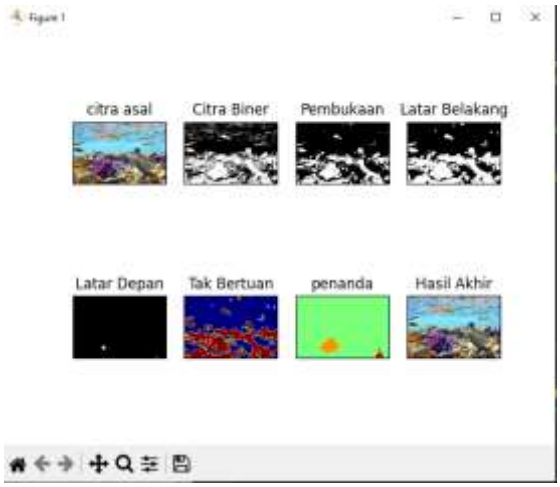

Figure 13. Watershed Segmentation Results

DISCUSSIONS

This stage contains an explanation of the steps taken to process the image. Here's the flow of explanation:

ColorSpace RGB Segmentation

1. Statement used to import OpenCV, NumPy and Matplotlib:

```
import cv2
```

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

```
import numpy as np
import matplotlib.pyplot as plt
```

2. The statement used to read the object:

```
#baca gambar
ikan = cv2.imread(r"D:\ikan2.jpg")
```

In this case, the object is stored in partition D with the name object ikan2.jpg

3. The statement used to convert BGR to RGB:

```
ikan = cv2.cvtColor(ikan,cv2.COLOR_BGR2RGB)
plt.subplot(2,4,1)
plt.imshow(ikan)
plt.title('gambar asli')
```

The first character after COLOR_ indicates the default color space, and the character after 2 is the target color space. this command shows the conversion from BGR (Blue, Green, Red) to RGB (Red, Green, Blue)

4. The statement used for the lower bound and upper bound declarations:

```
#deklarasi batas bawah
light_blue = (2,5,150)
#deklarasi batas atas
dark_blue = (280, 255, 255)
```

The statement is a tuple which is a lower bound for bright blue and an upper bound for dark blue, where the higher the level of the matrix value, the brighter the resulting image will be.

5. Statement to represent the colors red, green, blue:

```
red=ikan[:,:,0]
plt.subplot(2,4,3)
plt.imshow(red)
plt.title('red')

green=ikan[:,:,1]
plt.subplot(2,4,4)
plt.imshow(green)
plt.title('green')

blue=ikan[:,:,2]
plt.subplot(2,4,5)
plt.imshow(blue)
plt.title('blue')
```

6. The thresholding statement which is the separation between the object and the background based on the level of brightness or darkness is done with a statement:

```
#thresolding
mask = cv2.inRange(ikan, light_blue, dark_blue)
plt.subplot(2,4,6)
plt.imshow(mask)
plt.title('mask')
```

7. The statement used to impose the original object with a mask:

```
#impose gambar asli dengan mask
result = cv2.bitwise_and(ikan, ikan, mask = mask)
plt.subplot(2,4,7)
plt.imshow(result)
plt.title('hasil akhir')
```

Color Space HSV Segmentation

Explanation:

Read() is used to open the image while write() is used to save the image in the same folder. Import cv2 means that Python calls the cv2 library ie. OpenCV. ikan= cv2.imread (r"D:\ikan2.jpg") that is to read (load) the file 'ikan2.jpg' which is then saved.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

Convert from RGB image to HSV:

```
#convert RGB ke HSV
hsv_ikan = cv2.cvtColor(ikan, cv2.COLOR_RGB2HSV)
plt.subplot(3,3,2)
plt.imshow(hsv_ikan)
plt.title('citra hsv')
```

The declaration is entered by calling (HSV coin, light blue, dark blue) with the command:

```
#impose gambar asli dengan mask
result = cv2.bitwise_and(ikan, ikan, mask = mask)
plt.subplot(3,3,7)
plt.imshow(result)
plt.title('hasil akhir')
```

GrabCut Segmentation

1. The statement used to read the object to be processed

```
citra = cv2.imread(r"D:\ikan2.jpg")
if citra is None :
    print ("berkas tidak ditemukan")
    exit()
```

2. The statement to convert BGR image to RGB

```
#  Konversi BGR ke RGB
rgb = cv2.cvtColor(citra, cv2.COLOR_BGR2RGB)
```

3. The statement to convert RGB image to HSV

```
# konversi ke HSV
hsv_ikan = cv2.cvtColor(citra, cv2.COLOR_RGB2HSV)
plt.subplot(332)
plt.imshow(hsv_ikan)
plt.title('citra hsv')
```

4. The statement below is used to declare HSV images one by one, namely H, S, and V images

```
hue_image=hsv_ikan[:,:,0
]
plt.subplot(333)
plt.imshow(hue_image)
plt.title('citra h')
saturation_image=hsv_ika
n[:,:,1]
plt.subplot(334)
plt.imshow(saturation_im
age)
plt.title('citra s')
value_image=hsv_ikan[:,:
,2]
plt.subplot(335)
plt.imshow(value_image)
plt.title('citra v')
```

5. The statement to form a square tuple. The first and second values represent the (x,y) coordinate which is the top-left corner of the box. The third element specifies the width of the box and the fourth element specifies the height of the box.

```
# kotak untuk objek yang akan diambil
kotak = (30, 180, 200, 100)
```

6. The statement to create a two-dimensional array

```
#  Buat cadar awal
cadar = np.zeros(rgb.shape[:2], np.uint8)
```

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

7. The statement to insert an image into a Box image.

```
#  citra dengan kotak
citraKotak = citra.copy()
```

This image is intended to be supplemented with a grid image for visualization purposes only. With roses box build command like below:

```
cv2.rectangle(citraKotak, (kotak[0], kotak[1]),
              (kotak[0] + kotak[2], kotak[1] + kotak[3]),
              (255, 255, 255), 2)
```

8. The veil used by grabCut() is supplied with all zero-value elements. The statement is in the form of:

```
cadarAkhir = np.where((cadar == 0) | (cadar == 2), 0, 1).astype('uint8')
```

9. The arrays used to process the background and foreground by grabCut() are provided using the following two statements:

```
#  Buat larik untuk Grabcut
modelLB = np.zeros((1, 65), np.float64)
modelLD = np.zeros((1, 65), np.float64)
```

Each is a 1x65 array with an element type of 64-bit real numbers. In this case, modelLB handles the background and modelLD handles the foreground. All elements are assigned a value of 0.

10. The statement to call grabCut( ) is done with the command below:

```
#  Eksekusi Grabcut
cv2.grabCut(rgb, cadar, kotak,
            modelLB, modelLD, 5,
            cv2.GC_INIT_WITH_RECT)
```

The result after calling grabCut(), the veil contains three parts, namely the background outside the box, the background inside the box, and the object inside the box. The image below shows the values for each sequentially in the form of 0, 2, and 1.

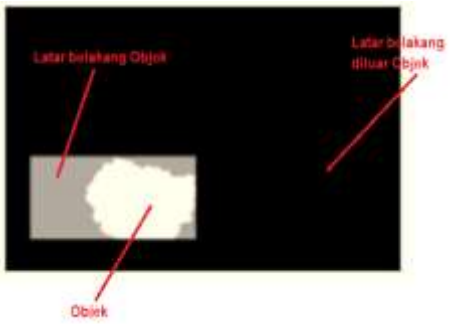

Figure 18. GrabCut background

Therefore, so that only the object has a value of 1 and the background value is 0, we need the command:

```
cadarAkhir = np.where((cadar == 0) | (cadar == 2), 0, 1).astype('uint8')
```

This command simultaneously creates the End veil of type unsigned 8-bit integer so that it is the same as the image element type. This final veil is used to remove the background in the original image.

11. The background earning process is carried out using:

```
#  Proses untuk mendapatkan citra tanpa latar belakang
citraAkhir = rgb * cadarAkhir[:, :, np.newaxis]
```

Watershed Segmentation

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

Explanation of the code on watershed segmentation, to display an image, the code command:

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
```

Statement on `abuabu= cv2.cvtColor(citra, cv2.COLOR_BGR2GRAY)`, used to convert color images into grayscale images, and the results are processed again into binary images through the following command:

```
ambang, citraBiner = cv2.threshold(abuabu, 0, 255,
         cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
```

Otsu is used to convert to binary images and constants `cv2.THRESH_BINARY_INV` to make the object white and the background black. In the binary image results above, if you pay close attention there are spots that act as noise, and if you want to remove the noise you can use a morphological transformation called an opening, for this purpose the command is:

```
# Hapus derau
kernel = np.ones((3, 3), np.uint8)
pembukaan = cv2.morphologyEx (citraBiner,
        cv2.MORPH_OPEN, kernel, iterations = 2)
```

First, create a kernel with size 3x3 and all elements value 1, then opening transform is done 2x. Furthermore, there are operations to enlarge the results of the opening transformation, namely:

```
# Untuk memastikan daerah latar belakang
latarBelakang = cv2.dilate(pembukaan, kernel,
                           iterations = 2)
```

This is to ensure that the image becomes the background. To get a watershed object, the distance transformation is calculated by:

```
transformJarak = cv2.distanceTransform(pembukaan,
                     cv2.DIST_L2, cv2.DIST_MASK_5)
```

With this, the Distance transform is an array containing the distance of each pixel in the object to the object's boundaries. In order for the touching objects to be separated, segmentation is implemented via:

```
ambang, latarDepan = cv2.threshold(transformJarak,
    0.7 * transformJarak.max(), 255, cv2.THRESH_BINARY)
```

The results on the Home screen will display small objects. And this object will be used as the initial watershed marker. Need to know `0.7 * transformJarak.max` means "0.7 x highest value on transformJarak". The difference between the Background and Foreground values is a temporary no-man's area. This area is calculated through:

```
daerahTakBertuan = cv2.subtract(latarBelakang,
                                latarDepan)
```

Actually, this area is part of the object that will be formed after the watershed is applied. Objects in the Peril Foreground are labeled (order numbers) labeling is done in order:

```
jumObjek, penanda = cv2.connectedComponents(latarDepan)
```

In this way, the marker contains every object used to perform the watershed operation. The result is that the back is labeled with the number 0. In order for the background to be given 1, all labels need to be increased by 1. This is done through a process: `penanda = penanda + 1.` Furthermore, label 0 is used for unclaimed areas. This is done by A process: `penanda[daerahTakBertuan == 255] = 0`
After this, it is this marker that is left to the watershed() with the image. Then the result is saved to the bookmark again. You can see in the picture above the results of "PENANDA" This is the last one, the boundary marks for each object are added to the original image through the process:

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 5, Number 1, January 2023
https://doi.org/10.47709/cnahpc.v5i1.2012

Submitted : January 18, 2023
Accepted : January 21, 2023
Published : January 24, 2023

```
citra[penanda == -1] = [255, 0, 0]
```
In this case, the blue color ([255,0,0]) is used.

5. CONCLUSION

Based on the results of image processing research using the color space, Grabcut, and watershed methods, the following conclusions can be drawn: By using color space segmentation we can analyze the color of coral objects with the basic colors red, green, blue, H , S, and V, then you can see the basic color difference of the fish object. By using Grabcut segmentation the output can vary, depending on the segmented area and the input tuple that is created so that segmentation can be done repeatedly to get the desired result.

Watershed segmentation is used to separate a number of objects that are in overlapping images so that they can be separated through a marking process for each given object. Then, based on each marker, which can be considered as a basin, flooding is carried out (towards the basin) so that the two boundaries of two objects that overlap or touch meet. Although in practice when it is found that the color in the object image resembles the background color it will be difficult to meet when it is segmented. In this study, the authors only implemented the image processing method on coral reefs and fish objects, for the future this image processing method can be developed and used on other objects that can be utilized for various needs..

6. REFERENCES

Abdul Kadir, & A. S. (2013). Teori dan Aplikasi Pengolahan Citra. Andi.

Achanya, T, & Ray, K. (2005). Image Processing Principles and Applications. Jhon Wiley & Sons, Inc.

Agoston, M. K. (2005). Computer Graphics and Geometric Modeling Implementation and Algorithms. Springer-Verlag.

Ambarwati, A., Passarella, R., & Sutarno. (2016). Segmentasi Citra Digital Menggunakan Thresholding Otsu untuk Analisa Perbandingan Deteksi Tepi. Annual Research Seminar 2016, 2(1), 216–226.

Dewi, T. (2007). Matinya Terumbu Karang. LIPI. http://lipi.go.id/berita/matinya-terumbu-karang/1338

Kadir, A. (2018). Dasar pemrograman python 3 : panduan untuk mempelajari python dengan cepat dan mudah bagi pemula (Maya (ed.)). Andi.

Kadir, A. (2019). Langkah Mudah Pemrograman OpenCV & Python. Gramedia. https://doi.org/719051502

Luthfi, O. M., Isdianto, A., Sirait, A. P. R., Putranto, T. W., & Affandi, M. (2020). Ekologi Terumbu Karang Buatan Pantai Damas. Envirobiotechjournals. https://news.unair.ac.id/2021/12/29/ekologi-terumbu-karang-buatan-pantai-damas/?lang=id

Maria, E., Yulianto, Y., Arinda, Y. P., Jumiaty, J., & Nobel, P. (2018). Segmentasi Citra Digital Bentuk Daun Pada Tanaman Di Politani Samarinda Menggunakan Metode Thresholding. Jurnal Rekayasa Teknologi Informasi (JURTI), 2(1), 37. https://doi.org/10.30872/jurti.v2i1.1377

Rindengan, A. J., & Mananohas, M. (2017). Perancangan Sistem Penentuan Tingkat Kesegaran Ikan Cakalang Menggunakan Metode Curve Fitting Berbasis Citra Digital Mata Ikan. Jurnal Ilmiah Sains, 17(2), 161. https://doi.org/10.35799/jis.17.2.2017.18128

Syafi'i, S. I., Wahyuningrum, R. T., & Muntasa, A. (2016). Segmentasi Obyek Pada Citra Digital Menggunakan Metode Otsu Thresholding. Jurnal Informatika, 13(1), 1–8. https://doi.org/10.9744/informatika.13.1.1-8

Tri Utami, A. (2017). Implementasi Metode Otsu Thresholding untuk Segmentasi Citra Daun. Fakultas Komunikasi Dan Informatika Universitas Muhammadiyah Surakarta.

Voronkov, I. (n.d.). 【2016-11-23】Object Detection Using Image Processing.pdf. 1–6.

Yunus, M. (2019). Perbandingan Metode-Metode Edge Detection Untuk Proses Segementasi Citra Digital. Jurnal Teknologi Informasi, Vol. 3(02), 146–160.

Zulkhaidi, T. C. A.-S., Maria, E., & Yulianto, Y. (2020). Pengenalan Pola Bentuk Wajah dengan OpenCV. Jurnal Rekayasa Teknologi Informasi (JURTI), 3(2), 181. https://doi.org/10.30872/jurti.v3i2.4033

* Corresponding author