Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 8, No. 1, February, 2024

A Hybrid Approach to Detect and Identify Text in Picture

Wydyanto Wydyanto [1, 2], Norshita Mat Nayan [1], Riza Sulaiman [1], Deshinta Arrova Dewi [3*], Tri Basuki Kurniawan [2, 4]

[1] *Institut IR.40 Universiti Kebangsaan Malaysia,43600 UKM Bangi, Selanggor, Malaysia.*

[2] *Faculty of Science and Technology, Universitas Bina Darma Palembang, 30264, UBD, Palembang, Indonesia.*

[3] *Inti International University, Nilai, Malaysia.*

[4] *Faculty of Science and Information Technology, Universiti Kebangsaan Malaysia, 43600, UKM Bangi, Selangor, Malaysia.*

Abstract

In order to create computer systems that can automatically read text from images or pictures, researchers focus on detecting and recognizing text in images. This issue is particularly difficult because images often have complicated backgrounds and a wide range of properties, including color, size, shape, orientation, and texture. Our proposed approach is based on morphology, which consists of a dilation and erosion process to extract text and recognize black-and-white text areas that contain document text or images. This suggested approach has been investigated for its ability to automatically identify text aligned with text pictures, such as store names, street names, banners, and posters. The design, application, and outcomes of the device's experiments are covered in this manuscript using Optical Character Recognition (OCR) Tesseract standards and the optimized OCR Tesseract. Our result shows that the optimized OCR Tesseract performs much better compared to the standard. Image preprocessing and text processing modules comprise this device's two modules. With an Arduino Uno and drawbot/flutter for text printing, this device was created using the Raspberry Pi and a 1.2GHz processor.

Keywords:

Text Detection;

Preprocessing; Segmentation;

Character Recognition;

Process Innovation; Product Innovation.

Article History:

Received:	08	September	2023
Revised:	07	December	2023
Accepted:	03	January	2024
Published:	01	February	2024

1- Introduction

Due to the rapid evolution in the field of digital technology, most sources of media can be found in electronic form [1], hence text recognition from still or static images is a primary extraction of letter or text data that matches the image. There are many sources that are based on paper or documents that are scanned into still images, such as book covers, magazines, pamphlets, educational materials, flashcard advertisements, street signs, and others [2]. However, there are some images that cause various problems that are challenging in the field of text extraction and recognition research, for example, text with a complex image background [3]. Image text is information embedded in or written in the form of different images [4]. Such images can be found in cameras, scanned documents, magazines, newspapers, posters, and so on [5]. Text images are very important because they represent, illustrate, and transfer information that can help communication, problem-solving, providing new job opportunities, cost-effectiveness, productivity, globalization, reducing cultural gaps, and many others [6, 7].

Computer software produces and stores images in digital storage compared to document images, digital images have more defects, such as low-resolution colored backgrounds, noise, inconsistencies, and low-quality rates. Therefore, the

[*] **CONTACT**: deshinta.ad@newinti.edu.my

DOI: http://dx.doi.org/10.28991/ESJ-2024-08-01-016

process of extracting text data is very difficult because the text has a different content background. The acquisition of information from images is more efficient and easier to understand than from images in text form. Computers identify ASCII characters by using a text extraction application during the conversion process from image form to text form [8, 9]. A detailed analysis and numerous strategies have been established for character recognition in the optical character recognition system, including the static image recognition process. With the aid of OCR (Optical Character Recognition), typed or handwritten text can be extracted from still images and converted into an editable electronic format. OCR facilitates various processes, including collecting and analyzing data from documents [10, 11].

HANWANG OCR, ABBYY, and Tesseract are just a few of the OCR tools or programs available. Because there is an integrated capability for image pretreatment before still text image extraction, Hanwang OCR and ABBYY OCR are superior to Tesseract OCR. The OCR software being utilized in this investigation, tesseract, is inferior to Hanwang OCR and ABBYY OCR [12]. Describe how textual data is essential for reporting and archiving processes in a variety of image-based applications, such as office work.

Researchers and experts have developed a variety of techniques to help with textual data extraction and detection. Introducing the edge method, the regional approach, the texture method, and the mathematical morphological method are a few of them. Jain & Gera [13] contends that the constraints of several criteria, such as the degree of picture accuracy, image extraction, and so on, result in advantages and disadvantages for all approaches and techniques employed in drawing text extraction. According to Rafi [14], it is becoming more common to digitize paper documents for archiving, indexing, or information retrieval uses. They include legal records, periodicals, commercials, and websites.

Because text is different from images in its nature, accurately and efficiently extracting text from documents is one of the most difficult difficulties. Hamzh [15] illustrates how the application's text extraction is highly reliant on the computer's operating system and picture processing. Image segmentation is a technique for breaking an image up into smaller pieces known as segments, according to Anjna & Kaur [16]. For picture compression applications or objects, especially images that don't need to be processed as complete, this technique is very helpful. Researchers will concentrate on finding a solution to the text recognition from still images challenge in this study in order to enable the extraction and rewriting of images by a print media floater or drawbot machine under Arduino control.

Human reading requires a large number of resources; therefore, there is extensive study on image processing techniques for text identification from images. The challenge of creating a reading model is complex and involves numerous facets of information. There are a lot of typographies as well, and you can find them in different legible sizes [17, 18]. Viewers typically don't want the surface where the text is printed to line up with their eyes in order to read it. Additionally, the designer employs a variety of colors for the backdrop and text [19, 20], and the lighting circumstances alter. Strong reading problems are exceedingly difficult due to all these factors. Four tiers make up a text comprehension system: text detection, text localization, text extraction, and text identification. There is no difference between these levels.

Identifying text in an image involves a process called text detection. The technique of pinpointing a text's location is known as text localization. The image's text components are separated from the background at the text extraction stage. The extracted text image can then be transformed using OCR technology into plain text [21]. Optical character recognition (OCR) and text-based search technology are used to detect and identify text in images [22, 23]. Text in an image or video is an effective informational tool. Information about the name of the relevant location, person, date, and purpose is provided in the text that is attached to the photograph. Image captions offer an overview or abstract of the picture.

2- Literature Review

Various approaches have been used in text recognition methods for still images. Among them are the initial processing method, text region extraction, and text binarization. This section will discuss the approach used in the text recognition process from still images. Casillano [24] and Nagaraja et al. [25] have introduced a new document image operation technique using contrasted adaptive images. Clustering of differences between local images and slopes is a form of image adaptation so that it can be adapted to text and environment differences caused by various types of degradation that occur on text data [26].

In the process, the PSNR value (peak signal to noise ratio) was measured to show a large improvement in the proposed method compared to other techniques. Hamad & Kaya [27] and Bansal [28] make a study on the performance of algorithms to identify segments of still images accurately and in detail. Intensity and texture-based image segmentation consist of two stages: setting the intensity level and texture-based image segmentation, which is proven to provide better results than traditional methods.

Choudhary et al. [29] have done research using OCR (optical character recognition) and character codes in text data files using a Raspberry Pi device capable of recognizing letters using the tesseract algorithm, Python, and recorded audio output. Optimum OCR and TTS methods are performed to improve more efficient results and help people with disabilities read text. Pal Singh & Khare [30] have detected and corrected text zones in document images interpreted by smartphones based on skew angle detection / correction as well as text zone detection using binarization methods.

The results of his research have shown a significant increase in text images when using the pre-processing method before using the OCR method, Kadam & Desai [31] conducted a study that included several stages, with the first stage involving a computer system that extracts the required features from the image and transforms it into the appropriate code using MATLAB software. The second stage involves transferring the MATLAB program to the robot arm in the form of coordinates that can be detected by the software application found in the robot. The application of a multi-phase algorithm is able to produce a robot that can detect, interpret, and respond to the various letters and numbers contained in the algorithm.

3- System Design

The overall flow chart based on the implemented system is shown in Figure 1. This flow chart consists of image data, a data collection module, a training module, a recognition module, and research findings.

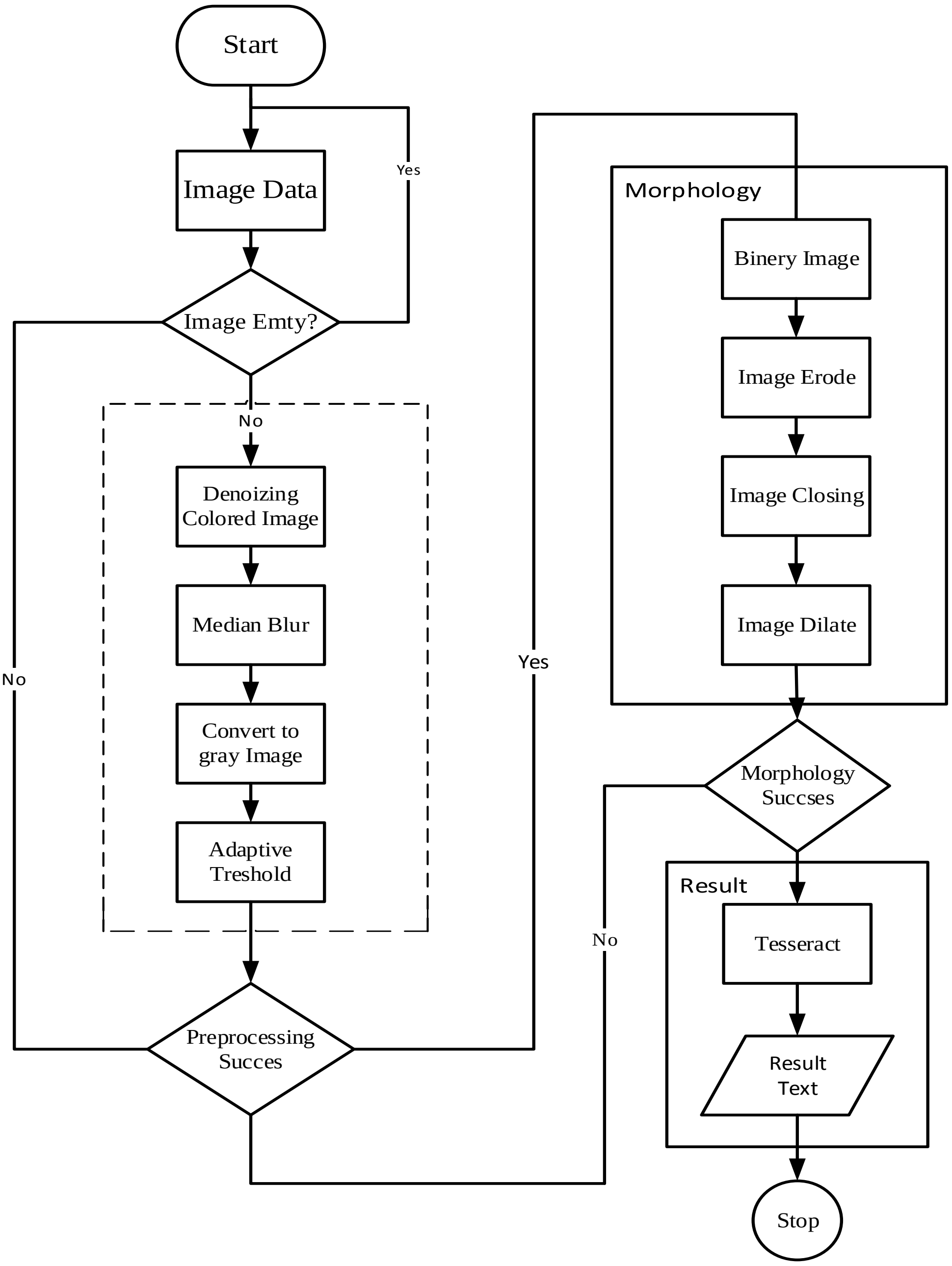


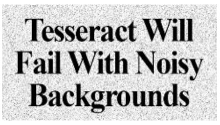
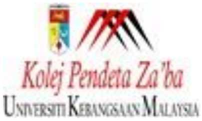
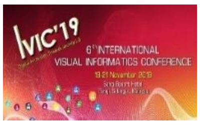
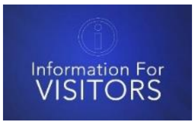
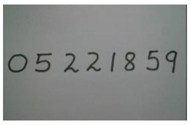
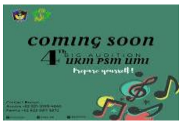
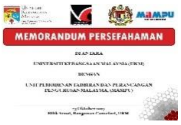
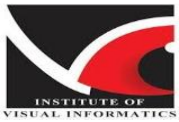
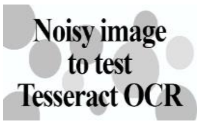
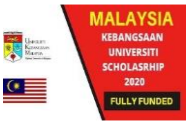
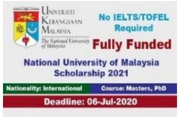
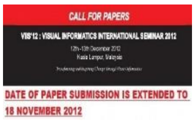
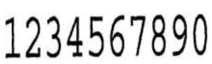
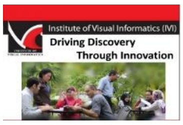
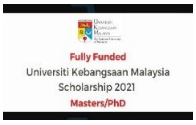
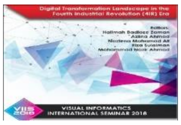
Figure 1. System Architecture

The proposed framework is used for efficient text detection and recognition. Image of the text that comes from the website of the Visual Institute of Informatics. It is an integrated framework for text detection and recognition.

4- Image Data

Text image data consists of three categories: text image with complex color background, text image with bright colors, and text image with blurred colors. These pictures are taken from the website of the Institute of Visual Informatics, National University of Malaysia (UKM), as displayed in Table 1. The text pictures that were taken are a total of 16 pictures that are considered to represent all the text pictures that will be tested in this study. Each image will be tested with Tesseract OCR to see the level of accuracy in text image processing. Then, after completing the process, the results of the study will be recorded and made in the form of a table to show the level of accuracy of text data processing. All text images will be tested using the proposed method. The results of the tests performed will be recorded and compared with the results of the Tesseract OCR process to see if there is a difference between the results of the image test before and after using the proposed method.

Table 1. **Category Image text taken for testing**

Future *grayscale*	Colored text images	Complex background image	Pictures With a bright light
 Tesseract Will Fail With Noisy Backgrounds	 Kolej Pendeta Za'ba Universiti Kebangsaan Malaysia	 IVIC'19 6th INTERNATIONAL VISUAL INFORMATICS CONFERENCE 19-21 November 2019	 Information For VISITORS
 05221859	 coming soon 4th ukm rsm iimu	 MEMORANDUM PERSEFAHAMAN	 INSTITUTE OF VISUAL INFORMATICS
 Noisy image to test Tesseract OCR	 MALAYSIA KEBANGSAAN UNIVERSITI SCHOLARSHIP 2020 FULLY FUNDED	 Universiti Kebangsaan Malaysia No IELTS/TOFEL Required Fully Funded National University of Malaysia Scholarship 2021 Nationality: International Course: Masters, PhD Deadline: 06-Jul-2020	 CALL FOR PAPERS VIS'12 : VISUAL INFORMATICS INTERNATIONAL SEMINAR 2012 12th-13th December 2012 Kuala Lumpur, Malaysia DATE OF PAPER SUBMISSION IS EXTENDED TO 18 NOVEMBER 2012
 1234567890	 Institute of Visual Informatics (IVI) Driving Discovery Through Innovation	 Fully Funded Universiti Kebangsaan Malaysia Scholarship 2021 Masters/PhD	 Digital Transformation Landscape in the Fourth Industrial Revolution (4IR) Era VISUAL INFORMATICS INTERNATIONAL SEMINAR 2018

Color images from the camera or browsing files from the internet will be converted to grayscale images based on the techniques used in the RGB approach. The flow chart of the source image processing process can be seen in Figure 2.

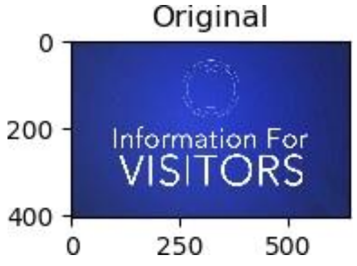


Figure 2. **Process the source Image**

The explanation of the flow chart in Figure 3 is as follows:

1. Select a file from the test data.
2. Take all the RGB pixels from the selected image file.

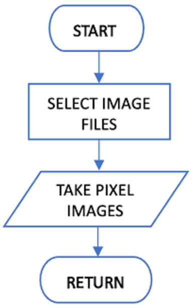


Figure 3. **Source Image processing flow chart**

After going through this process, the image will be processed:

```
img_names = "{}{}".format(path,img);
print("Load {}".format(img_names));
img = cv2.imread(img_names);
b,g,r = cv2.split(img);
rgb_img = cv2.merge([r,g,b]).
```

4-1- Preprocessing

At this stage, the raw text image is taken, and then the process of removing noise from the colored text image is performed. After that, the process of smoothing the image using the median blur technique, so that the texture of the text image becomes smoother. After the refinement process, the image is converted into a grayscale image. Then, the image is converted into a binary image, this operation is called adaptive thresholding.

4-1-1- Color Image Denoising

At this stage, the process of removing the noise in the color text image is done. The process of removing noise is done either in small amounts or in large amounts. In the process, a small area around the pixel is taken, and an average calculation process is performed with the aim of replacing the central element. Noise is considered a random variable with an average of zero. The noise equation is $p=p_0+n$, where p_0 is the actual value of the pixel and n is the noise in the pixel. In the process of removing noise, the same number of pixels, n, from different images will be taken, and then the average value will be calculated. In this process, the value $p=p_0$ should be obtained because the average noise is zero.

Once the image is grayscale, the result will be noise caused by lighting factors and source files that do not match the characteristics of the image. Therefore, the noise contained in the grayscale image will be deleted or removed using the Gaussian Fuzzy method because this technique is very suitable for removing noise from the image that has been exposed since this technique can maintain the originality of the image. The image process that still contains noise can be seen in Figure 4.

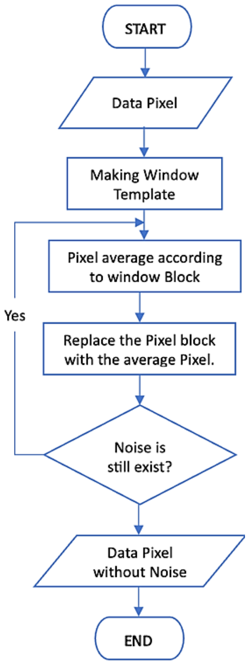


Figure 4. Flow chart of image processing that contains noise

Algorithmic process to eliminate the noise found in the picture image, can be shown in the pseudo code below:

```
INPUT: Picture
OUTPUT: Pictures without noise
For Block pixels in Pictures:
    Take the middle value from the block
    Calculating the Average of a pixel neighborhood
    Calculation of Noise Value
    Compare the mean value with the middle value
    IF Noise <> 0 Then
Replace the Middle value with the Flat value
    End IF
End For
```

Based on the process above, it can be explained that if the input from the algorithm is a colored text image, then the result of the algorithm process is a colored text image without noise. In the first stage, a repetition process will be done on each image block. Each iteration process will take the middle value from the block, then perform the process of calculating the average value of the pixel neighborhood. The result of the average calculation is then calculated as a noise value. Once the noise value is obtained, the noise value review process is carried out to ensure whether it is necessary to change the value of the center pixel or not. After the process is completed at all levels of repetition, the result is a new image without noise.

125	127	128
126	128	126
128	126	127

Figure 5. Original Image Pixels

In Figure 5, the image that has been selected will be de-noised. The first step is the value from the average will be calculated to obtain the pixel value in the middle. Based on picture 4.1 the picture that has been selected has pixels with a value of 128. Average = (125+127+128+126+126+128+126+127)/8 = 126.625 From the results of the average calculation, the average value is obtained which is 126.625, then the noise value will be calculated as follows:

$$P = P0+N \quad 126.625 = 128+N \tag{1}$$

$$N = 128 - 126.625 = 1.375 \tag{2}$$

The result of calculating the noise value obtained based on the picture is 1.375. Then the noise value produced should be made up to a value of 0. The process of removing noise can be seen in Figure 6:

125	127	128
126	127	126
128	126	127

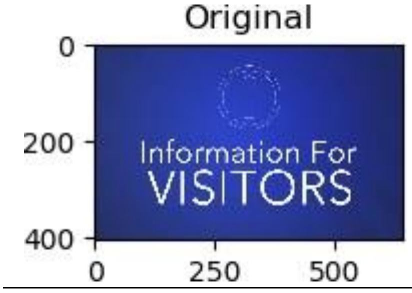
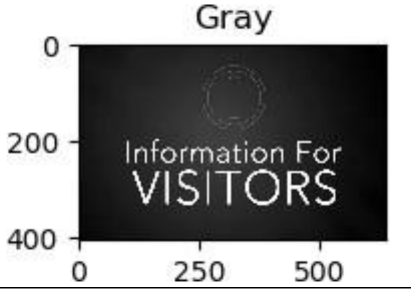
Figure 6. Image pixels of the noise removal process

The process of removing noise on colored text images implemented in Python programming code is described as follows:

```
1 img = cv2.imread(img_names)
2 b,g,r = cv2.split(img)     # get b,g,r rgb_img = cv2.merge([r,g,b]
                             # switch it to rgb
3 Denoising = cv2.fastNlMeansDenoisingColored(img,None,10,10,7,21)
```

Line 1 of the code above is the process to open the image file. The second line is the process of changing from BGR image format to RGB format. While line number three is a process to remove the noise found in the picture. The result of the implementation of the coding cut above can be shown in Table 2. which is a comparison of the original image before and after the noise removal process.

Table 2. Comparison before and after noise is removed

Before	After
	

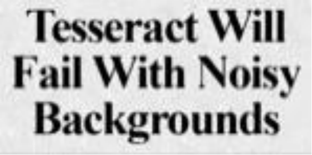
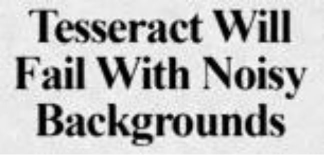
4-1-2- Image Median Blur

Median blur or median filter is a technique that is often used to remove noise from images. The central element of the image is replaced by the median of all pixels in the kernel region. Median blur is a technique that processes edges while removing noise. This technique uses the medianBlur () method from the imgproc class. The syntax of this method is as follows:

```
1 b,g,r = cv2.split(Denoising) # get b,g,r
2 rgb_dst = cv2.merge([r,g,b]) # switch it to rgb
3 rgb_dst = cv2.medianBlur(rgb_dst, 3)
```

In the median blur operation, the image to be processed is derived from an image that has had noise removed. Line one is the process of extracting BGR pixel data from the image. The second line is the process of converting from BGR pixels to RGB. The third line is the rgb_dst process that uses the input, and the number three is the number of kernels that will be used. The result of the above coding is shown in Table 3, which is a picture of the result of the median blur process.

Table 3. The result of the median blur process

Before	After
 Tesseract Will Fail With Noisy Backgrounds	 Tesseract Will Fail With Noisy Backgrounds

4-1-3- Covert to Gray Image

The group of pictures taken from the website of the Institute of Visual Informatics, Universiti Kebangsaan Malaysia, are color pictures. The process of converting a color image into a grayscale image requires more knowledge about color imagery. The color of a pixel in an image is a combination of three colors, namely red, green, and blue (RGB). RGB color values are represented by XYZ in three dimensions, which are lightness, chroma, and hue attributes. The quality of a color image depends on the color represented by the number of bits that can be supported by the digital device. Basic color images are represented by 8 bits; high color images are represented by 16 bits; native color images are represented by 24 bits; and deep color images are represented by 32 bits.

The number of bits determines the maximum number of different colors supported by a digital device. If red, green, and blue are represented by 8 bits, then the RGB combination is represented by 24 bits and supports 16,777,216 different colors. 24 bits represent the color of pixels in a color image. The grayscale image is represented by luminance using an 8-bit value. The luminance of the pixel value in the grayscale image ranges from 0 to 255. Changing the color image into a grayscale image is done by changing the RGB value that contains 24 bits into a grayscale value that only consists of 8 bits.

Various image processing techniques and software applications transform color images into grayscale images. However, image processing techniques or applications cannot deal with differences in chromaticity and illumination. In past research, several linear and non-linear techniques have been discussed to change color images into grayscale images. The latest techniques that can handle these differences are much better than the previous ones. However, the technique involves several computing procedures, such as changing RGB space to XYZ space, then approximation, then mapping, or other related techniques. Grayscale mapping from color images developed by approaching spectral uniformity is often inadequate.

The latest techniques used to convert color images into gray images will take a long time for computing purposes and use a large amount of memory. Therefore, a new algorithm is proposed to transform a color image into a grayscale image that takes less time and uses less memory space.

Based on the flow chart in Figure 7, the explanation that can be given is as follows:

1. Pixel data from the previous process.
2. Make a calculation on the Y value from RGB pixels.
3. Check whether A is less than the pixel data length?
4. If yes, continue to step 2.
5. Otherwise, replace the pixel data with Y.
6. Generate gray pixels from pixel data.

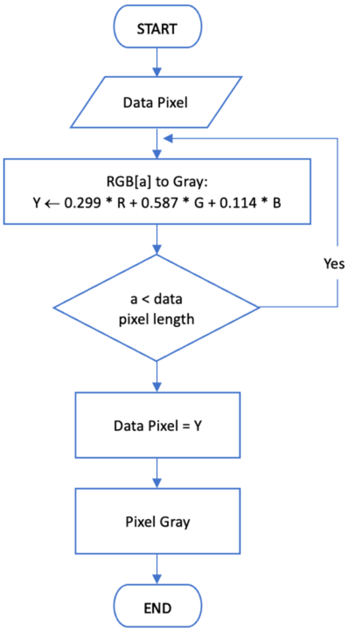


Figure 7. **Grayscale Image Processing Flow Chart**

The algorithm process is able to change the color image into a grayscale image in the picture. The process of the algorithm can be shown in pseudo code as below:

```
INPUT : Image
OUTPUT : Image Graysacle
For Blok piksel in Image:
   Piksel Grayscale = (R+G+B) /3
```

Based on the process above, it can be explained that the input from the algorithm is a color text image, then the result of the algorithm process is a grayscale text image. The first step is to perform an iterative process on each pixel of the image. In each iteration process pixels will be taken and then those pixels will be evaluated. Starting with the grayscale pixels from the RGB pixels of the image, then the process of updating the grayscale pixels on the output image is performed. After all the repetition process is completed, the result is a new grayscale image as stated in Figure 8.

R	G	B
125	127	128

Figure 8. **Native RGB Pixels**

In Figure 8, the image will undergo the pixel change process from RGB format to grayscale pixels.*Gray = (125+127+128) / 3 = 126.333*

Based on the results of the grayscale calculation, the gray pixel value was obtained with a value of 126,333. After the process of changing the image to grayscale, the gray pixel value is as shown in Figure 9:

R	G	B
126	0	0

Figure 9. **Grayscale pixels**

The process of removing and changing the pixels on the colored text image to grayscale is implemented in the python programming code as follows:

```
grayscaled = cv2.cvtColor(rgb_dst, cv2.COLOR_BGR2GRAY)
```

On the code line above, it is a process to change an image with RGB format to a grayscale image. diman rgb_dst is the input image in RGB pixel format while COLOR_BGR2GRAY is the destination pixel format in gray format. The result of the implementation of the coding program above is shown in Table 4, the result of the comparison of the image with RGB format before the process and the image in grayscale format after undergoing the grayscale process.

Table 4. Comparison Results Before and After the Grayscale Process

Before	After
VIC'19 6th INTERNATIONAL VISUAL INFORMATICS CONFERENCE 19-21 November 2019	VIC'19 6th INTERNATIONAL VISUAL INFORMATICS CONFERENCE 19-21 November 2019

4-1-4- Adaptive Thresholding

Adaptive thresholding is the simplest image segmentation method. In grayscale images, thresholding can be used to create binary images. A binary image is produced from a color image by segmentation. Segmentation is the process of assigning each pixel in the source image to two or more different classes. If there are more than two classes, then the usual result is several binary images.

In image processing, thresholding is used to divide the image into smaller segments, or chunks. A color or grayscale value is used to define the limit. The benefit of the first binary image is that it can reduce the complexity of the data and simplify the process of identification and classification. The most common way to convert a gray-level image into a binary image is to choose a single threshold value. In this matter, adaptive thresholding can help.

An algorithm will be used to determine the pixel threshold based on a small region around it. So researchers managed to get different thresholds for different regions for the same image and even gave better results for images with different lighting.

With a clean image and no more noise, thresholding will perform the Adaptive Thresholding process, especially if there are pictures that have different lighting conditions in different areas. In that case, adaptive thresholding can help. The algorithm is able to determine the pixel threshold based on a small region around it. Thresholds for different regions of the same image need to be obtained in order to provide better results for images with varied illumination. Adaptive Threshold requires three input parameters. The adaptive method decides how the threshold value can be obtained:

a. ADAPTIVE_THRESH_MEAN_C:

Threshold value is the average in the surrounding area reduced by the constant C.

b. ADAPTIVE_THRESH_GAUSSIAN_C:

The threshold value is the gaussian-weighted sum of the environmental value that is reduced by the constant C.

c. BlockSize determines the size of the neighborhood area and C is a constant that is reduced by the average or weighted number of neighborhood pixels. number of neighborhood pixels.

Based on the flow chart in Figure 10, the explanation that can be given is as follows:

1. Using the pixel data from the previous process.
2. Fill in the threshold value and fill in the src from the pixel data.
3. Calculate the equation etc.
4. Check if x is less than the width of the image?
5. If yes then step 3 needs to be repeated.
6. If not then a check should be done is y less than the height of the image?
7. If yes then step 3 needs to be repeated
8. Otherwise, fill the pixel data with T.
9. The pixel result of the binary image from the pixel data will be obtained.

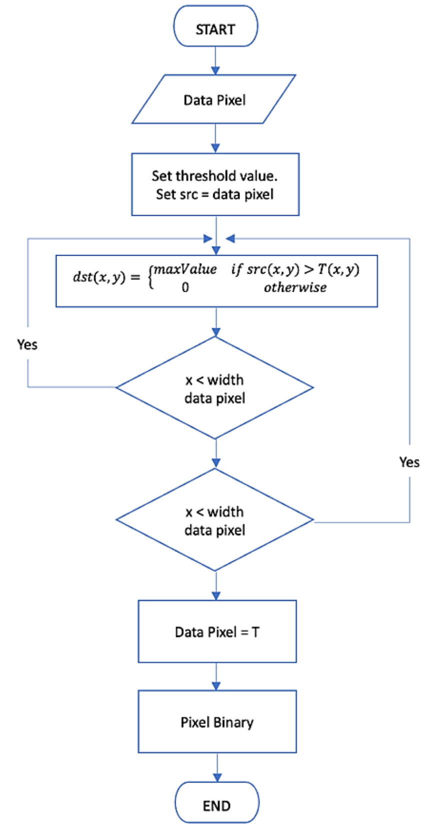


Figure 10. Flowchart of the process of Thresholding Images containing Noise

The algorithm process from the adaptive thresholding process in the picture, can be shown in the pseudo code as:

```
INPUT : Pictures Grayscale
OUTPUT : pictures Biner
For Blok piksel in pictures Biner:
     Calculate lokal threshold on each block
   For Piksel in Blok:
           If Piksel < Threshold Then
           Piksel =0
      Else
          Piksel = 1
      End If
   End For
End For
```

Based on the process above, it can be explained that the input from the algorithm is a grayscale text image. Then the result of this algortima process is a binary text image that represents the pixel values 1 and 0. Langkah pertama yang, what needs to be done is to repeat the process on each pixel block of the image, then calculate the threshold value from the pixel block. After getting the threshold results from the block, do the process to update the pixels in the binary image with a value of 1 if the pixel value is greater than the threshold and a value of 0 if the pixel value is less than the threshold. After the process is completed at all iteration levels, the result is a new binary image, as outlined in Figure 11.

125	127	128
126	127	126
128	126	127

Figure 11. Grayscale Pixel Blocks

In Figure 11, the image will be processed to change pixels from grayscale format to binary pixels.

$$Threshold = (125 + 127 + 128 + 126 + 127 + 126 + 128 + 126 + 127) \;/\; 9 = 126.666 \tag{3}$$

Based on the results of the grayscale calculation, a threshold value of 126,666 was obtained, so that the result of the pixel block from the process of converting the grayscale image to binary is shown as in Figure 12:

0	1	1
1	1	1
1	1	1

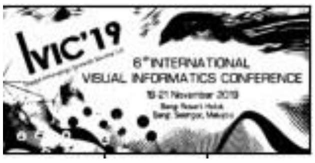
Figure 12. **Blok Pixel Biner**

The process of removing or changing pixels on a grayscale text image to a binary image is implemented in the python programming code as follows:

```
threshold = cv2.adaptiveThreshold(grayscaled, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 127, 1)
threshold = cv2.bitwise_not(threshold)
```

In the first line of code above, is the process of converting an image with grayscale format to a binary image. Grayscaled is the input image in RGB pixel format while cv2.ADAPTIVE_THRESH_GAUSSIAN_C is a threshold value which is the sum of the gaussian average of environmental values reduced by the constant C while cv2.THRESH_BINARY is for binary threshold. The result of the implementation of the coding program above is shown in Table 5 comparing the results of the grayscale image before and after the binary process.

Table 5. **Table of image comparison results before and after the adaptive threshold process**

Before	After
	

4-2- Morphology

The Morphology process is used to extract text and recognize black-and-white text areas that contain document text or images. In this process will use two processes namely:

a. **Proces *Dilation***

Returns a fast binary morphological adjustment of the image. This function returns the same result as grayscale dilation but has the ability to perform faster for binary images. Morphological dilation sets the pixel at (i, j) to the maximum over all pixels in the neighborhood centered on (i, j). Dilation enlarges the bright parts and reduces the dark parts. The environment is specified as a 2-D Array of 1 and 0. Otherwise, use structural elements with cross-connections = 1). Outside of bool, Array is an option for storing morphological results. If nothing happens, a new Array will be allocated.

b. **Proces *Erosion***

Restores image binary morphology erosion faster. This function returns the same result as grayscale erosion but works at a faster rate for binary images. Morphological erosion sets the pixel at (i, j) to the minimum over all pixels in the neighborhood centered on (i, j). Erosion shrinks in light areas and enlarges in dark areas. Selem ndarray, the environment option is specified as a 2-D array of 1 and 0. Otherwise, use intersecting structural elements (extension = 1). Out of bool, optional Array to store morphological results. If Nothing is passed, a new Array will be allocated.

At this stage, a morphological transformation process will be carried out, which is a process of change based on the shape of the image and is carried out on a binary image. In this process will require two inputs, one is the original binary image, the second is called a structuring element or kernel that determines the nature of the operation. This morphological transformation process consists of two basic morphological operators namely Erosion and Dilation. The variant form is like opening and closing.

Based on the flow chart in Figure 13, the explanation that can be given is as follows:

1. The pixel data is taken from the previous process.
2. Fill src with pixel data.
3. Calculate the erosion or dilatation value for all elements at src.

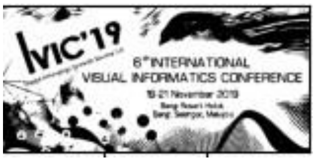
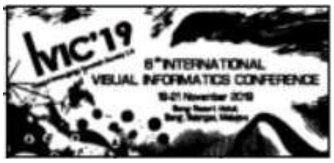
4. Check if x is less than the width of the Image?
5. If yes then step 3 needs to be repeated.
6. If not then a check should be made to ensure that y is less than the height of the image?
7. If yes, step 3 needs to be repeated.
8. otherwise, fill the pixel data with dst.
9. The pixel result of the binary image from the pixel data will be obtained.

The morphological transformation process on the text image is implemented in the python programming code as follows:

```
gradient = cv2.morphologyEx(threshold, cv2.MORPH_GRADIENT, kernel)
opening = cv2.morphologyEx(threshold, cv2.MORPH_CLOSE , kernel)
dilation = cv2.dilate(threshold,kernel,iterations = 1)
```

The first line of code above, is a process for morphological transformation on text data images. At this stage, a binary image as input. cv2.MORPH_GRADIENT is an operation type for gradient morphology while cv2.MORPH_CLOSE is an operation type for closing morphology. The result of the implementation of the coding program above is shown in Table 6 which shows the comparison results from the binary image before and after the morphological transformation process.

Table 6. Table of comparison results before and after the morphological transformation process

Before	After
	

4-2-1- Binary Image

At this level is the input process from the morphological transformation, which is the binary image input process. The image received is a processed binary image.

4-2-2- Image Erode

In this process, the boundaries of the object will be scraped with the background so that the background of the image remains white. At this level, the kernel plays an important role as an area parameter or pixel process environment. Pixels in the original binary image with values of 1 and 0 will be considered 1 only if all pixels below the kernel are 1, otherwise they will be scraped to 0 until all pixels on the border are discarded, depending on the size of the kernel. This causes the thickness or size of the foreground object to decrease, or only the white area to decrease in the picture. The small white noise present in the binary image will be removed while the two connected objects will be released.

4-2-3- Image Closing

The process of closure is the opposite of opening, which is a process of widening followed by erosion. In this case, it is useful to cover the small holes found in the foreground object or small black dots on the object, thus obtaining an object with a closed area.

4-2-4- Image Dilate

The process of dilatation is the reverse process compared to the process of erosion. This process will add to the boundaries of the object. This condition will increase the black region of the image or cause an increase in the size of the object, as in the case of noise removal, erosion, and will be followed by widening or dilation. The erosion process will remove white noise. In addition, the erosion process also shrinks the object. Therefore, the widening process needs to be done. Since the noise has disappeared, the object area needs to be added with dilation. In addition, this process is also useful for combining parts of damaged objects.

5- Results and Discussion

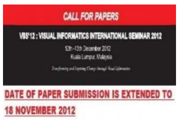
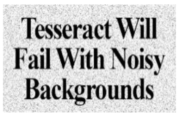
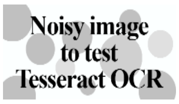
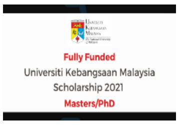
5-1- Algorithm Testing

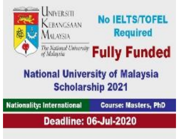
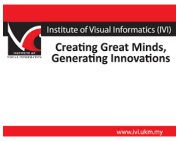
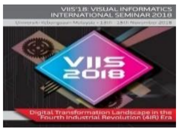
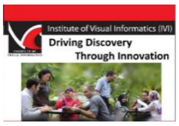
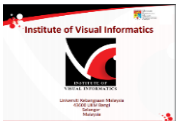
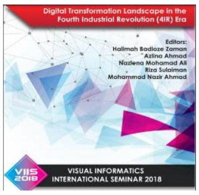
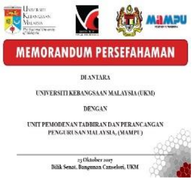
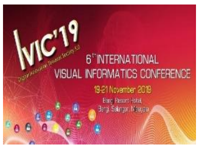
In the algorithm testing process, the test images that will be used will be selected first. Next, the Tesseract Standard OCR command will be run on the raspberry device that has been prepared. The input used for the testing process of this algorithm is in the form of test images that have been selected in advance, while the output from the results of this testing process is in the form of text or characters resulting from the recognition and interpretation process performed by the Tesseract OCR Algorithm.

The algorithm optimization process is done with the aim of getting better OCR results by using the algorithm found in OCR tesseract. The main process of Tesseract Algorithm Optimization consists of preprocessing and morphological stages. The preprocessing process involves image, denoising, median filtering, grayscale, and adaptive thresholding, while the morphological process consists of binary image, erosion, image closing, and image dilatation.

The proposed method has been implemented using the Python programming language device Raspberry Pi 3. For the character identification process, the Tesseract software will be used. At this stage, OCR Tesseract will be introduced by using the character recognition algorithm that is already available on Tesseract, which will be used as a standard parameter during algorithm testing. The main purpose to be achieved from testing this algorithm is to produce text from the test images that have been prepared. The results of Tesseract's OCR Algorithm testing are shown in Table 7:

Table 7. Tesseract OCR Algorithm Testing Results

Source Image	Process Results Using OCR Tesseract
	VISUAL INFORMATICS
	VISITORS
	CALL FOR PAPERS VIIS’12: VISUAL INFORMATICS INTERNATIONAL SEMINAR 2012 12th -13th December 2012 Kuala Lumpur, Malaysia SUN oan eR aie a Le eR od DATE OF PAPER SUBMISSION IS EXTENDED TO 18 NOVEMBER 2012
	_ Tesseract Will Fail With Noisy _ Backgrounds
	Noisyimage to test Tesseract OCR
	Fee Universit Kerpancsaan a 9 sige Fully Funded Universiti Kebangsaan Malaysia Scholarship 2021 Masters/PhD
	FULLY FUNDED
	RXR Kolej Pendeta Za’ba Unswersitt Kesancsaan Mataysia

	Jee Universi Kinaneesay No IELTS/TOFEL Mataysia Required % , A tee Fully Funded National University of Malaysia Scholarship 2021 Nationality: International Course: Masters, PhD Deadline: 06-Jul-2020
	eek net) Creating Great Minds, Generating Innovations
	VIIS'18: VISUAL INFORMATICS INTERNATIONAL SEMINAR 2018
	Institute of Visual Informatics (IVI) Driving Discovery Pee nd wsearrsronmaness Through Innovation
	"Institute of Visual Informatics VISUAL INFORMATICS Universiti Kebangsaan Malaysia 43600 UKM Bangi Selangor Malaysia -@
	E i aS . f Editors: Halimah Badioze Zaman a D¥ > * * Azlina Ahmad ow Nazlena Mohamad Ali af \ Riza Sulaiman Mohammad Nazir Ahmad VISUAL INFORMATICS NCO) URS LN aye)
	rai Cees <> om SE AGI) ee eee DIANTARA UNIVERSITI KEBANGSAAN MALAYSIA (UKM) DENGAN UNIT PEMODENAN TADBIRAN DAN PERANCANGAN PENGURUSAN MALAYSIA, (MAMPU) 23 Oktober 2017 Bilik Senat, Bangunan Canselori, UKM
	Je q Ne: N NU s Nene he cunt INFORMATICS CONFERENCE: 19-21 November 2019 Bangi Resort Hotel Bangi, Selangor, Melaysi

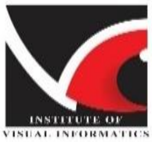
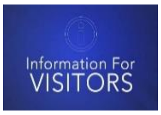
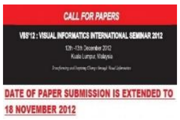
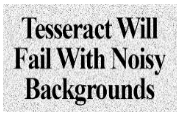
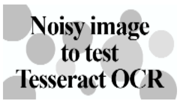
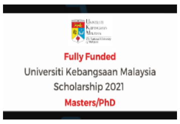
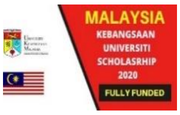
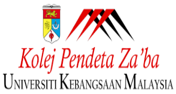
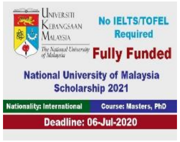
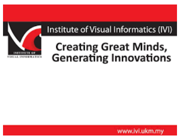
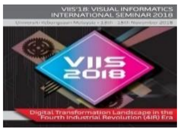
Tesseract's OCR Algorithm testing results table shows that variations from test images give different results based on Tesseract's OCR Algorithm testing. This is because the Standard Tesseract Algorithm has a weakness, especially for images that contain noise. Pictures that have motifs or non-plain content will also be noise objects for Tesseract.

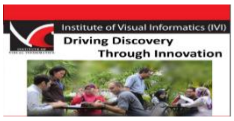
5-2- Algorithmic Optimization

Algorithm Optimization is an effort made to produce more optimal results by an algorithm. This can be done by doing additional levels or changes from the already existing algorithm by adding an insertion process or a new process so that a better algorithm result will be obtained.

The main purpose of this algorithm optimization is to get better text recognition results when compared to using the Tesseract Standard OCR Algorithm. Algorithm optimization techniques are used which consist of image processing processes such as preprocessing and morphology. The results of the Algorithm Optimization testing proposed to optimize the Tesseract OCR Algorithm are shown in Table 8:

Table 8. Tesseract OCR Algorithm Optimization Testing Results

Source Image	Results Method
	INSTITUTE OF VISUAL INFORMATICS
	VISITORS
	CALL FOR PAPERS ViIS'12: VISUAL INFORMATICS INTERNATIONAL SEMINAR 2012 12th -13th December 2012 Kuala Lumpur, Malaysia ing and Inspiring Change through Visual Informatics DATE OF PAPER SUBMISSION IS EXTENDED TO 18 NOVEMBER 2012
	Tesseract Will Fail With Noisy Backgrounds
	Noisy, image to test Tesseract OCR
	Fully Funded Universiti Kebangsaan Malaysia Scholarship 2021 Masters/PhD
	MALAYSIA KEBANGSAAN TTL rare Prva FULLY FUNDED
	Kolej Pendeta Za'ba Universiti KepancsAan MaraysiA
	0 Ni Rill ace KmANa SAAN GS i can Fully Funded National University of Malaysia Scholarship 2021 Nationality: International Course: Masters, PhD Deadline: 06-Jul-2020
	Creating Great Minds, Generating Innovations
	eee eee Landscape in the * Fourth Industrial Revolution (41R) Era

	Institute of Visual Informatics (IVI) Driving Discovery Through Innovation
	Institute of Visual Informatics VISUAL INFORMATICS Universiti Kebangsaan Malaysia 43600 UKM Bangi Selangor Malaysia
	Digital Transformation Landscape in the Fourth Industrial Revolution (41R) Era e Editors: Halimah Badioze Zaman > ere * — *Azlina Ahmad Nazlena Mohamad Ali Riza Sulaiman Mohammad Nazir Ahmad VISUAL INFORMATICS aC Nirel TUR UC een)
	poe Universi Kepancsaan VV wa Mavaysia DIANTARA UNIVERSITIT KEBANGSAAN MALAYSIA (UKM) DENGAN UNIT PEMODENAN TADBIRAN DAN PERANCANGAN PENGURUSAN MALAYSIA, (MAMPU) 23 Oktober 2017 Bilik Senat, Bangunan Canselori, UKM
	Con 6° INTERNATIONAL VISUAL INFORMATICS CONFERENCE 19-21 November 2019 ma Bang Resort Hotel * @ rei raremue esr 9, CO |

5-3- Level of Testing Accuracy

Confusion matrix is one of the predictive analytics tools that displays and compares the actual value or real value with the model's predicted value to produce a specific evaluation matrix. This means that the Confusion matrix describes the labor of an algorithm used. The evaluation structure of the results of the Confusion matrix can be seen through Table 9. The final result of the Confusion matrix will provide information about the error value of the algorithm when compared to the value of the actual text. To get an accurate value, the process of calculating the level of accuracy needs to be done.

Table 9. Confusion Matrix

		Actual Value	
		1 (Positive)	**0 (Negative)**
Predicted Value	**1 (Positive)**	TP (True Positive)	FP (False Positive)
	0 (Negative)	FN (False Negative)	TN (True Negative)

In order to find out the level of accuracy of the method used, the Confusion matrix evaluation will be carried out on the final result of OCR Tesseract Standard and also on the result of Optimizing OCR Tesseract Algorithm. The results from the evaluation will provide information about the error value of the algorithm compared to the actual text result. Confusion matrix in the form of a matrix diagram that describes the performance of the algorithm. Confusion matrix evaluation structure can be seen through Table 9. To obtain accurate results, the accuracy evaluation process needs to be carried out. In this study the accuracy equation comes from Table 9.

For each evaluation parameter, the Confusion matrix is obtained by comparing the number of characters or letters produced from the process before the Algorithm Optimization and the results of letters or characters after the Algorithm Optimization process is carried out.

True Positive (TP)

- Is positive data with correct text, the value is obtained by comparing the number of correct characters by the system before and after Algorithmic Optimization is performed.

True Negative (TN)

- It is a negative data with the correct text, the value is obtained by comparing the number of correct characters by the system before and after the Algorithm Optimization is performed.

False Positive (FP)

- Is negative data but predicted as positive data, the value is obtained by comparing the number of wrong characters but interpreted as correct by the system before and after Algorithm Optimization is performed.

False Negative (FN)

- Is positive data but predicted as negative data, the value is obtained by comparing the number of wrong characters but interpreted as wrong by the system before and after Algorithm Optimization is done.

Confusion matrix evaluation is described through the equation found in the formula 4:

$$accuracy: \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

All parameters from the Confusion matrix are obtained from the data of the results of the tests that have been performed. The final results from the Confusion matrix evaluation are shown in Table 10.

Table 10. Confusion Matrix Calculation Results Table

Number	Process Results using OCR Tesseract	Results Using the Method	Result OCR *Tesseract*		Algorithm Optimization Results	
			TP	FP	TP	FP
1	VISUAL INFORMATICS	INSTITUTE OF VISUAL INFORMATICS	20	14	34	0
2	VISITORS	VISITORS	9	0	9	0
3	CALL FOR PAPERS VIIS’12: VISUAL INFORMATICS INTERNATIONAL SEMINAR 2012 12th -13th December 2012 Kuala Lumpur, Malaysia SUN DATE OF PAPER SUBMISSION IS EXTENDED TO 18 NOVEMBER 2012	CALL FOR PAPERS ViIS'12: VISUAL INFORMATICS INTERNATIONAL SEMINAR 2012 12th -13th December 2012 Kuala Lumpur, Malaysia ing and Inspiring Change through Visual Informatics DATE OF PAPER SUBMISSION IS EXTENDED TO 18 NOVEMBER 2012	119	123	242	0
4	_ Tesseract Will Fail With Noisy _ Backgrounds	Tesseract Will Fail With Noisy Backgrounds	43	3	46	0
5	Noisyimage to test Tesseract OCR	Noisy, image to test Tesseract OCR	35	1	36	0
6	Fee Universit Kerpancsaan a 9 sige Fully Funded Universiti Kebangsaan Malaysia Scholarship 2021 Masters/PhD	Fully Funded Universiti Kebangsaan Malaysia Scholarship 2021 Masters/PhD	31	44	75	0

7	FULLY FUNDED	MALAYSIA KEBANGSAAN TTL rare Prva FULLY FUNDED	12	41	53	0
8	RXR Kolej Pendeta Za’ba Unswersitt Kesancsaan Mataysia	Kolej Pendeta Za’ba Universiti KepancsAan MaraysiA	37	24	61	0
9	Jee Universi Kinaneesay No IELTS/TOFEL Mataysia Required % , A tee Fully Funded National University of Malaysia Scholarship 2021 Nationality: International Course: Masters, PhD Deadline: 06-Jul-2020	0 Ni Rill ace KmANa SAAN GS i can Fully Funded National University of Malaysia Scholarship 2021 Nationality: International Course: Masters, PhD Deadline: 06-Jul-2020	116	70	116	38
10	eek net) Creating Great Minds, Generating Innovations	Creating Great Minds, Generating Innovations	48	9	48	0
11	VIIS‘18: VISUAL INFORMATICS INTERNATIONAL SEMINAR 2018	eee eee Landscape in the * Fourth Industrial Revolution (41R) Era	61	1	59	7
12	Institute of Visual Informatics (IVI) Driving Discovery Pee nd wsearrsronmaness Through Innovation	Institute of Visual Informatics (IVI) Driving Discovery Through Innovation	51	27	78	0
13	“Institute of Visual Informatics VISUAL INFORMATICS Universiti Kebangsaan Malaysia 43600 UKM Bangi Selangor Malaysia -@	Institute of Visual Informatics VISUAL INFORMATICS Universiti Kebangsaan Malaysia 43600 UKM Bangi Selangor Malaysia	123	3	126	0
14	e i aS . f Editors: Halimah Badioze Zaman a D¥ > * * Azlina Ahmad ow Nazlena Mohamad Ali af \ Riza Sulaiman Mohammad Nazir Ahmad VISUAL INFORMATICS NCO) URS LN aye)	Digital Transformation Landscape in the Fourth Industrial Revolution (41R) Era e Editors: Halimah Badioze Zaman > ere * — *Azlina Ahmad Nazlena Mohamad Ali Riza Sulaiman Mohammad Nazir Ahmad VISUAL INFORMATICS aC Nirel TUR UC een)	116	125	241	38
15	rai Cees <> om SE AGI) ee eee DIANTARA UNIVERSITI KEBANGSAAN MALAYSIA (UKM) DENGAN UNIT PEMODENAN TADBIRAN DAN PERANCANGAN PENGURUSAN MALAYSIA, (MAMPU) 23 Oktober 2017 Bilik Senat, Bangunan Canselori, UKM	poe Universi Kepancsaan VV wa Mavaysia DIANTARA UNIVERSITIT KEBANGSAAN MALAYSIA (UKM) DENGAN UNIT PEMODENAN TADBIRAN DAN PERANCANGAN PENGURUSAN MALAYSIA, (MAMPU) 23 Oktober 2017 Bilik Senat, Bangunan Canselori, UKM	159	31	148	42
16	Je q Ne : N NU s Nene he cunt INFORMATICS CONFERENCE : 19-21 November 2019 Bangi Resort Hotel) Bangi, Selangor, Melaysi	Con 6° INTERNATIONAL VISUAL INFORMATICS CONFERENCE 19-21 November 2019 ma Bang Resort Hotel * @ rei raremue esr 9, CO |	107	24	99	32

Based on Table 10, the accuracy level assessment is done using the Tesseract OCR Algorithm. The final result of testing is compared with the final result of optimization using the proposed algorithm. Evaluation of the level of accuracy of the Tesseract OCR Algorithm: All the following figures are obtained from Table 9 which is the final result of the testing that has been done. After that. The calculation is done on each parameter using the Confusion matrix evaluation as follows:

$$\begin{gathered} TP = 20 + 9 + 119 + 43 + 35 + 31 + 12 + 37 + 116 + 48 + 61 + 51 + 123 + 116 + 159 + 107 = 1087 \\ FP = 14 + 0 + 123 + 3 + 1 + 44 + 41 + 24 + 70 + 9 + 1 + 27 + 3 + 125 + 31 + 24 = 540 \\ \text{Accuracy: } \frac{1087+0}{1087+0+540+0} = 0.66 \end{gathered} \tag{5}$$

Tesseract OCR Algorithm Optimization Accuracy Calculation:

$$\begin{gathered} TP = 34 + 9 + 242 + 46 + 36 + 75 + 53 + 61 + 116 + 48 + 59 + 78 + 126 + 241 + 148 + 99 = 1471 \\ FP = 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 38 + 0 + 7 + 0 + 0 + 38 + 42 + 32 = 157 \\ \text{Accuracy : } \frac{1471+0}{1471+0+157+0} = 0.903 \end{gathered} \tag{6}$$

Based on the evaluation results, the TP value is the value obtained by calculating the number of correct characters in OCR identification, the FP value is the value obtained from the result of calculating the number of incorrect characters in the OCR identification process. Accuracy is a value in the form of a percentage of the success rate. The final results of the accuracy level evaluation can be seen through Table 11:

Table 11. Schedule of the Final Results of the Accuracy Stage Assessment

Method	Accuracy Value
OCR Tesseract Standard	66.6 %
Tesseract OCR optimization	90.3%

Based on Table 11, the accuracy results of the model using OCR Tesseract Standard, the accuracy value obtained is 66.6% while the accuracy result from the Tesseract OCR Algorithm Optimization model, the accuracy value obtained is as much as 90.3%.

6- Conclusion

Preprocessing is required for Tesseract OCR to be able to read text from still images. Tesseract OCR is used to recognize text from still photos. Many still images—including those from websites, scanned still photos from magazines, newspapers, and other sources, as well as posters and other visual materials—contain text and information that has been embedded in or written in various image formats. Digital photographs contain more flaws than document images, including noise, loss of compression, noise reduction, and significant texture softening. Another issue is multicolored backdrops with poor resolution. The study focuses on techniques for extracting text from still web photos. There are three stages to creating a still photograph. the first stage of preliminary processing (preprocessing), which includes morphology, noise reduction, and the grayscale technique. This stage is the initial conversion of the source image (original) into a text that Tesseract OCR can understand. The second stage involves extracting the text region using binary pictures, segmentation, and selection (region selection). Reading the blob extraction (Blob extraction) is part of the text area extraction procedure's third phase. The Raspberry Pi's second phase is its processing, and the Arduino's third phase is its data. The information from the preprocessing method will then be passed to Drawbot, which can be read and rebuilt using the Arduino control. Pre-processing and extracting the text area will produce more accurate results and aid the drawbot/Flotter in writing the text without errors if done properly. The results of this study will be used to develop a novel preprocessing technique that combines grayscale, noise removal, thresholding, and morphology to provide clear still image text outputs that the tesseract OCR engine can read reliably.

7- Declarations

7-1- Author Contributions

Conceptualization, W.W. and R.S.; methodology, W.W., N.M.N., and R.S.; software, T.B.K.; validation, W.W., N.M.N., and R.S.; formal analysis, D.A.D. and T.B.K.; investigation, W.W.; resources, D.A.D.; data curation, T.B.K.; writing—original draft preparation, W.W.; writing—review and editing, D.A.D.; visualization, D.A.D.; supervision, R.S.; project administration, N.M.N.; funding acquisition, D.A.D. All authors have read and agreed to the published version of the manuscript.

7-2- Data Availability Statement

The data presented in this study are available in the article.

7-3- Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

7-4- Institutional Review Board Statement

Not applicable.

7-5- Informed Consent Statement

Not applicable.

7-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

[1] Agrahari, A., & Ghosh, R. (2020). Multi-Oriented Text Detection in Natural Scene Images Based on the Intersection of MSER with the Locally Binarized Image. Procedia Computer Science, 171, 322–330. doi:10.1016/j.procs.2020.04.033.

[2] Inkeaw, P., Bootkrajang, J., Charoenkwan, P., Marukatat, S., Ho, S. Y., & Chaijaruwanich, J. (2018). Recognition-based character segmentation for multi-level writing style. International Journal on Document Analysis and Recognition, 21(1–2), 21–39. doi:10.1007/s10032-018-0302-5.

[3] Epshtein, B., Ofek, E., & Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, California, United States. doi:10.1109/cvpr.2010.5540041.

[4] Jung, K., Kim, K. I., & Jain, A. K. (2004). Text information extraction in images and video: A survey. Pattern Recognition, 37(5), 977–997. doi:10.1016/j.patcog.2003.10.012.

[5] Jain, R., & Gianchandani, D. (2018). A Hybrid Approach for Detection and Recognition of Traffic Text Sign using MSER and OCR. 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). doi:10.1109/i-smac.2018.8653761.

[6] Di Justo, P. (2015). Raspberry Pi or Arduino Uno? One Simple Rule to Choose the Right Board. Make Magazine. Available online: https://makezine.com/article/technology/arduino/admittedly-simplistic-guide-raspberry-pi-vs-arduino/ (accessed on January 2024).

[7] Perez-Delgado, M. L., & Roman Gallego, J. A. (2019). A hybrid color quantization algorithm that combines the greedy orthogonal bi-partitioning method with artificial ants. IEEE Access, 7, 128714–128734. doi:10.1109/ACCESS.2019.2937934.

[8] Zhao, Q. J., Cao, P., & Meng, Q. X. (2014). Image capturing and segmentation method for characters marked on hot billets. Advanced Materials Research, 945–949, 1830–1836. doi:10.4028/www.scientific.net/AMR.945-949.1830.

[9] Modi, H., & C., M. (2017). A Review on Optical Character Recognition Techniques. International Journal of Computer Applications, 160(6), 20–24. doi:10.5120/ijca2017913061.

[10] Marial, A., & Jos, J. (2017). Feature extraction of optical character recognition: Survey. International Journal of Applied Engineering Research, 12(7), 1129-1137.

[11] Zhao, M., Li, S., & Kwok, J. (2010). Text detection in images using sparse representation with discriminative dictionaries. Image and Vision Computing, 28(12), 1590–1599. doi:10.1016/j.imavis.2010.04.002.

[12] Sabu, A. M., & Das, A. S. (2018). A Survey on various Optical Character Recognition Techniques. 2018 Conference on Emerging Devices and Smart Systems (ICEDSS). doi:10.1109/icedss.2018.8544323.

[13] Jain, N. & Gera, D. (2015). Comparison of Text Extraction Techniques- A Review. International Journal of Innovative Research in Computer and Communication Engineering, 03(02), 621–626. doi:10.15680/ijircce.2015.0302003.

[14] Rafi, A. M. (2014). Text Extraction from Images Using Connected Component Method. Journal of Artificial Intelligence Research & Advances, 1(2).

[15] Hamzh, A.R.E.M. (2016). Object Recognition using Image Processing. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 4(11), 61–64. doi:10.17148/ijireeice.2016.41111.

[16] Anjna, E., & Kaur, E. R. (2017). Review of image segmentation technique. International Journal of Advanced Research in Computer Science, 8(4), 36-39.

[17] Nikam, V. S., & Yawalkar, P. M. (2015). Binarization Technique on Historical Documents using Edge Width Detection. International Journal on Recent and Innovation Trends in Computing and Communication, 3(6), 3793–3798.

[18] Taneja, A., Ranjan, P., & Ujjlayan, A. (2015). A performance study of image segmentation techniques. 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida, India. doi:10.1109/icrito.2015.7359305.

[19] Mahalakshmi, V., Bennet, M., Hemaladha, R., Jenitta, J., & Vijayabharathi, K. (2018). Implementation of OCR using raspberry PI for visually impaired person. International Journal of Pure and Applied Mathematics, 119(15), 111-117.

[20] Prum, S. (2017). Text-zone Detection and Rectification in Document Images Captured by Smartphone. Proceedings of the First EAI International Conference on Computer Science and Engineering, Penang, Malaysia. doi:10.4108/eai.27-2-2017.152342.

[21] Manwatkar, P. M., & Singh, K. R. (2015). A technical review on text recognition from images. 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India. doi:10.1109/isco.2015.7282362.

[22] Juang, J.-G., Tsai, Y.-J., & Fan, Y.-W. (2015). Visual Recognition and Its Application to Robot Arm Control. Applied Sciences, 5(4), 851–880. doi:10.3390/app5040851.

[23] Kumar, D., & Ramakrishnan, A. G. (2014). Methods for text segmentation from scene images. ELCVIA Electronic Letters on Computer Vision and Image Analysis, 13(2), 32. doi:10.5565/rev/elcvia.591.

[24] Casillano, N. F. B. (2019). Utilization of Optical Character Recognition (OCR) in the development of a Number System Converter Application. Indian Journal of Science and Technology, 12(16), 1–5. doi:10.17485/ijst/2019/v12i16/137794.

[25] Nagaraja, L., Nagarjun, R. S., Nishanth, M. A., Nithin, D., & Veena, S. M. (2015). Vision based text recognition using raspberry PI. International Journal of Computer Applications, 975, 8887.

[26] Ramesh, N., Srivastava, A., & Deeba, K. (2018). Improving optical character recognition techniques. International Journal of Engineering and Technology (UAE), 7(2), 361–364. doi:10.14419/ijet.v7i2.24.12085.

[27] Hamad, K., & Kaya, M. (2016). A Detailed Analysis of Optical Character Recognition Technology. International Journal of Applied Mathematics, Electronics and Computers, 4(Special Issue-1), 244–244. doi:10.18100/ijamec.270374.

[28] Bansal, D.S. (2018). Techniques of Text Detection and Recognition: A Survey. International Journal of Emerging Research in Management and Technology, 6(6), 83-87. doi:10.23956/ijermt.v6i6.250.

[29] Choudhary, A., Rishi, R., & Ahlawat, S. (2013). A new approach to detect and extract characters from off-line printed images and text. Procedia Computer Science, 17, 434–440. doi:10.1016/j.procs.2013.05.056.

[30] Pal Singh, D., & Khare, A. (2015). Text Region Extraction: A Morphological Based Image Analysis Using Genetic Algorithm. International Journal of Image, Graphics and Signal Processing, 7(2), 39–47. doi:10.5815/ijigsp.2015.02.06.

[31] Kadam, M. P. B., & Desai, L. R. (2014). A Hybrid Approach to Detect and Recognize Text In Images. IOSR Journal of Engineering, 4(7), 13–19. doi:10.9790/3021-04741319.