International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I6P239

Volume 71 Issue 6, 396-405, June 2023

Original Article

Improving OCR Performance on Low-Quality Image Using Pre-processing and Post-processing Methods

Ivan Christian[1], Gede Putra Kusuma[2]

[1,2]*Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia.*

[1]*Corresponding Author : ivan.christian004@binus.ac.id*

Received: 03 April 2023 Revised: 09 June 2023 Accepted: 14 June 2023 Published: 25 June 2023

Abstract *- Optical Character Recognition (OCR) is a technology to recognize text inside images. One of the factors affecting the success rate of OCR is image quality. Therefore, it is necessary to improve the image quality before OCR processing. In addition to pre-processing, post-processing was also carried out. This was done to improve the success rate of the OCR. In the pre-processing stage, what is done is to resize the image using bicubic interpolation, which is then followed by deleting the background image. Bicubic interpolation was chosen because it can result in a smoother, enlarged image and has fewer interpolation artifacts. A grayscale conversion using luminance algorithm was also carried out to optimize the process. OCR processing is done using a tesseract. As for post-processing, what is done after OCR is done is to use the N-gram language model and the Levenshtein distance algorithm. The performance of the proposed model is assessed by comparing the success rate of the usual OCR and one of the existing OCR pre-processing or post-processing models with the developed OCR method. The best pre-processing method in this study is to use a combination of the shadow removal method and custom grayscale conversion with a total error rate of 14.56%. Then the post-processing method using a lookup table can also improve the final OCR performance with a total error rate of 13.94%. So, it can be concluded that combining the pre-processing shadow removal method, custom grayscale conversion, and post-processing lookup table method can improve the accuracy of OCR performance.*

Keywords *- Luminance algorithm, n-gram language, Optical Character Recognition, Post-processing, Pre-processing.*

1. Introduction

Finance Technology (FinTech) is a company engaged in the finance sector that utilizes modern technology. The main objective of FinTech itself is to minimize infrastructure costs while still providing the best service for consumers. Therefore, FinTech offers a fast and efficient process. Consumers can immediately take care of their finances by using a smartphone or computer connected to the internet.

To apply for or register for FinTech requires KTP (Kartu Tanda Penduduk/Indonesia Identity Card) identification from the consumer concerned. Consumers only need to take a KTP photo using a consumer smartphone camera, and then consumer data will be read using OCR (Optical Character Recognition). The KTP is to identify the authenticity of the identity of the consumer. The problem faced is that not all KTPs from consumers are in good condition (damaged). In addition, smartphone cameras have low-resolution quality; therefore, when OCR is done, some data will not be read.

This research will focus on increasing OCR by combining several pre-processing and post-processing methods. Optical Character Recognition (OCR) is the process of translating images in the form of text, capturing text, typing or handwriting into a format that is understood by machines for the purposes of indexing, searching, editing, and reducing storage space [1]. Simply put, OCR is a technology for converting images in the form of text and hardcopy text into machine-readable digital text files. OCR begins with scanning the document. The next stage of OCR will tidy up the position of the image so that it is aligned and not tilted so that text and characters can be more easily recognized. OCR will detect the contents of the image and then analyze each character in the document. The final step is to change the scan results in the form of a digital text file.

Mande and Lei conducted research to improve the accuracy of OCR on low-quality images. In their research [2], they overcome low-quality images by doing some processing before OCR is carried out, namely by removing the background image. The first thing to do is brightness distortion and chromaticity distortion. Apart from using brightness and chromaticity, they also used image enhancement to make the text clearer and suppress the background RGB value. This is done by using a non-linear transformation to change the contrast of each image.

Another study was carried out by Matteo, Ratko, Matija, and Tihomir to improve the accuracy of their OCR using resizing, sharpening, and blurring. Resizing is done

because when the image to be done on OCR does not meet the minimum height required, it will decrease the accuracy of the OCR. Tesseract OCR has a minimum height of 20 pixels, therefore resizing up to 100 pixels in that study. Apart from resizing, sharpening is also done to increase the contrast between the text and the background. This was also done by Di and Gady in research [3]. The difference is in research [4] using unsharp masking techniques to increase the contrast between text and background. Unsharp masking is done to get a smoothed image. In this study, they used a Gaussian low-pass filter to obtain a smoothed image. The last method is blurring, which reduces high-frequency information and removes noise. This method is done by applying a low-pass filter to analyze the image for each pixel and replace it with the average value. Using these noise filters, you can remove background images commonly known as diacritic characters.

In this study, [5] used the grayscale conversion method on OCR pre-processing. They mainly use grayscale conversion because grayscale can simplify the algorithm and reduce computational requirements. Many algorithms can be used to perform a grayscale conversion. Kanan and Cotrell said the Luminance algorithm could perform better than other algorithms in terms of texture-based image processing [6].

In addition to using pre-processing methods to improve OCR accuracy, there is also a post-processing method, namely, using the N-gram language model. The use of N-gram language model itself is a statistical model that can calculate the probability of a given word sequence. In research [7] using Google Books N-gram Viewer (dataset) to train the language model used. If the language model fails, a selection will be made using the Levenshtein distance algorithm to choose the best word.

There has been a lot of research to improve the accuracy of OCR on low-quality images. There are various ways to improve the accuracy of OCR, namely by doing pre-processing and post-processing. Therefore, in this study, the authors propose to combine methods to improve OCR results on low-quality KTP photos. The proposed method by doing pre-processing before the OCR is done and, after that, does post-processing to minimize the errors that occur in the resulting OCR. The pre-processing that will be done is resizing, deleting the background image, and grayscale conversion. The first step is to perform resizing so that it will be easier for further processing. In addition, resizing is also carried out to enlarge the letters in the KTP so that it can be easier to recognise letters during OCR. After resizing, the background image will be removed. On the KTP, there is a background image that can decrease OCR accuracy. At the last stage in pre-processing, a grayscale conversion is carried out. Grayscale conversion is done to minimize if the background image is not completely erased. This can happen because the background image contained in the KTP has characters/letters that can be recognized as text. The method of grayscale conversion is carried out using the Luminance algorithm calculation. After the pre-processing is done, OCR will be carried out using Tesseract 4.1. The results from the OCR will be reprocessed using the N-gram language model and the Levenshtein distance algorithm to optimize OCR accuracy. The N-gram language model is responsible for correcting words if the word is found in the dataset used. If the word is not found in the dataset, the best word will be taken based on the levenshtein distance algorithm.

2. Related Works

There have been several studies aimed at improving the accuracy of OCR. This increase was carried out by increasing the pre-processing and the OCR process. One of them is research [2]. In their research, Mande and Hansheng said that removing the background image before the OCR process is carried out can reduce the results of errors in the OCR. In this study, decomposing the difference in brightness can make it easier to erase the background. The background image has a richer texture compared to plain text characters. In addition, the background image has a large difference in the value of the RGB value for each pixel. In this study, a comparison of OCR results was produced without deleting and by deleting the background image. The results of this study prove that the removal of the background can improve the process of OCR results. The background removal method is proven to be effective. It affects the OCR output results by comparing the different RGB values in the background and using brightness distortion and chromaticity to increase the contrast in the image.

Research [5] conducted optimization at the OCR pre-processing stage. The optimization will be done in the form of a grayscale conversion. This is done because the grayscale conversion is the simplest image optimization technique. Grayscale can also make algorithms simpler and reduce computational requirements. The grayscale algorithm used in this research is the Luminance algorithm because [6] said the Luminance algorithm is the best choice for texture recognition. The next step is to increase the detail of the existing text so that it can be read more easily before the next steps are taken. The enhancement of the text and edge details is by un-sharp masking of the filter.

Research [4] applies a combination of image resizing, sharpening, and blurring. The result of combining these methods can increase OCR accuracy by 33.3% in Tesseract 3.5 and 22.6% in Tesseract 4.0. In addition, [4] also said it can improve OCR performance to be more efficient when adding k-means.

In research [8] using artificial neural networks to improve the accuracy of OCR. By using basic CNN to enlarge an image from a low-quality image.

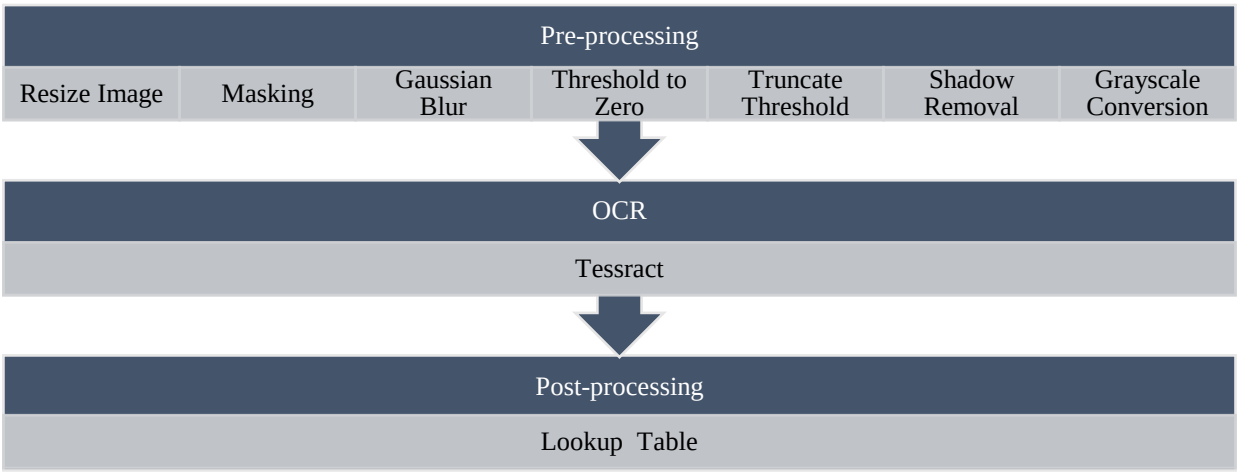

Fig. 1 Proposed methods

The development of OCR accuracy improvement in handwriting using the Generative Adversarial Networks (GANs) method was carried out by Karimi, Veni, and Dan Yu. In research [9] it was proven that using GAN-based at the pre-processing stage can improve the accuracy of character recognition in handwriting.

Apart from improving pre-processing, there are also those who have improved the OCR process. Research [3] focuses on using the X-Y Cut algorithm and K-means Clustering algorithm. [3] propose to carry out OCR processing using the X - Y Cut algorithm and K-means Clustering, where the process performs an X-Y Cut Algorithm to separate each word and also the separation of each existing character. The next step is to group using K-means Clustering by grouping similar characters. The grouping results are used to convert low-resolution characters into higher-resolution characters. The study tested 5 images with different resolutions. In the test results, there is an additional error. This happens because of an error when grouping per character. This grouping error can occur because of the ambiguous character, for example, the character "i" with the character "l", then the character "e" with the character "c". Table 3 shows the ambiguous characters generated when grouping. When not many characters are generated, the error result on the OCR will not increase. In addition, [10] also said that the X - Y cut algorithm could increase efficiency in conducting OCR.

Other studies [11] improved by doing an Enhanced Ensemble Technique (EET). EET is a technique to improve character recognition and not produce ambiguous characters. The process of EET itself is to process a character by dividing it into 3 parts based on pixels. The division process is divided into the first group, namely the group with black pixels. The second group is the white pixel group, and the last group is the gray pixel group. This research focuses on changing the gray pixels into black pixels or white pixels so as not to produce ambiguous characters later. Imad, Zeyad, and Hanan tested using and without using EET. The results of the test show that EET can reduce the character error to 7.47%.

[7] improved the OCR process using the N-gram language model and the Levenshtein distance algorithm. The N-gram language model is used to recognize words generated from OCR, whether they are in the data dictionary used or not. The data dictionary used in this study is Google Books N-gram Viewer. If the resulting word still produces several words, the best word retrieval will be carried out using the levenshtein distance algorithm. In this study, the WER decreased by 28.44%, so it can be said that using the N-gram language model and the Levenshtein distance algorithm can reduce the final result of the OCR output.

[12] improve OCR performance by using the post-correction model method. In this study, OCR testing was carried out in 3 different languages. The method used is to predict the initial results of OCR. The built model does not increase the recognition rate on the initial OCR results but improves performance on the final results.

3. Proposed Methods

In this research, a combination of methods will be carried out, as shown in (Figure 1). The method used is the pre-processing method and the post-processing method. For the pre-processing method that will be used is resizing the image, masking, gaussian blur, threshold to zero, truncating threshold, shadow removal, and grayscale conversion. After pre-processing will be processed using tesseract 4.1. For the post-processing stage, the method used is with using the lookup table method.

3.1. Pre-Processing

The resize image method used in this research is to Equalize the size of each existing image object. This is due to the size of each image are not the same, so it can affect the value of the OCR. The difference in size can be seen in Figure 2. The image resizing method is to change the pixel size in the image to 670 (width) x 431 (height).

The second method used is masking photos of people on KTPs. This matter is used to improve the performance of the OCR because the OCR can detect the image as a character to be read.

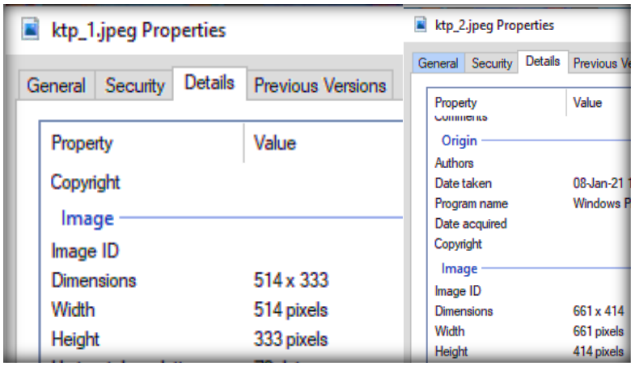

Fig. 2 Example KTP size image

Fig. 3 Masking photos of people on KTP

For the masking method, the method used is to overwrite the person's photo on the KTP with a blue color, the same as the KTP background, as seen in Figure 3.

The Gaussian blur method is an image filter method that uses the Gaussian function to create blurred images. Using the method of Gaussian blur can break down the noise in an image [13]. The processing results of Gaussian blur can be seen in Figure 4.

The next method used in this research is the method of thresholding. The global thresholding algorithm takes a value pixel of an image. It compares it with the predetermined threshold value set and then changes the pixel value of the image to a new value [14]. There are several thresholding methods for this research. The thresholding methods used are threshold to zero and truncate threshold. The threshold-to-zero method is a method that compares a pixel-value image with a predefined threshold of 127. So, if the value of a pixel is greater than the threshold value, then the pixel value will be fixed.

Fig. 4 Gaussian blur method

Fig. 5 Example of threshold to zero

Fig. 6 Example of truncate threshold

Conversely, if the value of a pixel is smaller or equal to the threshold value, then the pixel value will be changed to 0 with the sample example in Figure 5. The next threshold method is the truncate threshold. This method is hampered the same as the threshold to zero methods. The difference is for truncating the threshold when the value of a pixel is greater than the threshold value, and then the pixel value will be converted to a threshold value. Whereas if the pixel value is smaller or the same as the threshold value, then the value will remain the same as the sample in Figure 6.

The Shadow Removal method is used to remove shadows or also the distortion contained in the image [15]. Shadow removal is widely used as an effective pre-processing method. In this research, the purpose of shadow removal itself is to remove the background image contained in the KTP, as can be seen in the example Figure 7.

Fig. 7 Example of shadow removal

Fig. 8 Example of custom grayscale conversion

Provinsi	Kota/Daerah
D.I Yogyakarta	Kabupaten Bantul
	Kabupaten Gunung kidul
	Kabupaten Kulon Progo
	Kabupaten Sleman
	Kota Yogyakarta
PROVINSI DKI JAKARTA	Kabupaten Kepulauan Seribu
	Jakarta Barat
	Jakarta Pusat
	Jakarta Selatan
	Jakarta Timur
	Jakarta Utara

Fig. 9 Example of provincial and regional lookup table

```
Value OCR PROVINSI JAWA BARAI
Value OCR KOTA BANDUNG
PROVINSI JAWA BARAT PROVINSI JAWA BARAT True
0.0
KOTA BANDUNG KOTA BANDUNG True
0.0
```

Fig. 10 provincial and regional lookup table method

```
Value OCR PROVINSI JAWA BARAT Lookup table PROVINSI BALI Avg CER 52.17391304347826
Value OCR PROVINSI JAWA BARAT Lookup table PROVINSI BANTEN Avg CER 40.74074074074074
Value OCR PROVINSI JAWA BARAT Lookup table PROVINSI BENGKULU Avg CER 32.25806451612903
Value OCR PROVINSI JAWA BARAT Lookup table PROVINSI DKI JAKARTA Avg CER 22.857142857142858
Value OCR PROVINSI JAWA BARAT Lookup table PROVINSI JAWA BARAT Avg CER 0.0
```

Fig. 11 Example of lookup table

The last pre-processing method applied in this research is grayscale conversion. The algorithm used for grayscale conversion is the luminance algorithm in the formula Equation 1. The luminance algorithm makes adjustments by adjusting the dominant colour of the KTP itself. The first step is to change the blue pixel value because the dominant colour of the KTP is blue. For the range of colour changes carried out by trial-and-error stages of several values. For values changes taken from 0.1B – 0.5B. The final result of adjusting the formula luminance algorithm as in the formula Equation 2. The result of the adjustment grayscale conversion can be seen in Figure 8.

$$\text{Luminance} = 0.3R + 0.59G + 0.11B \quad (1)$$

$$\text{CustomGrayscale} = 0.299R + 0.587G + 0.5B \quad (2)$$

3.2. Post-Processing

The post-processing method used in this research is by using lookup tables. This method refers to the concept of N-gram language and the Levenshtein algorithm by making a lookup table from the values that have value can be ascertained. Lookup tables include provinces, regions, religion, occupation, citizenship status, and marital status.

Figure 9. is an example of a provincial and regional lookup table. The lookup table method will compare the OCR results to all values in existing lookup tables. If the value from the lookup table is close to the OCR results, then the value from the lookup table will be the new value from the OCR. As can be seen in Figure 10., the initial value of OCR is "JAWA BARAI", which is done lookup table against all available lookup table values by doing character error rate (CER) calculation as shown in Figure 11. Got value “JAWA BARAT” is the most likely value because that value is the lowest CER. So "JAWA BARAT" was taken to replace the value of “JAWA BARAI” as the latest OCR result value.

4. Experiments

4.1. Dataset

The case study that will be tested in this research is the data of the Indonesian National Identity Card (Kartu Tanda Penduduk). The image dataset will be collected manually by the researcher with permission of the KTP owner to avoid misappropriation of the dataset. Because the proposed system does not require a training stage, the KTP data obtained will be fully used for analysis purposes.

The analysis was carried out by comparing several OCR applications and OCR applications from researchers to take and compare their performance values and the resulting percentage error rate. The target KTP data to be collected is 100 KTP photos (including blurry and non-blurred photos). Examples of KTP photos to be used can be seen in Figures 12 and 13.

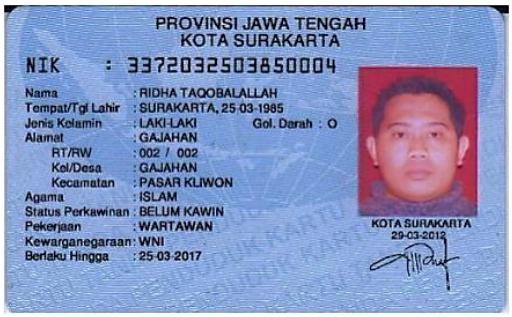

Fig. 12 Example for high resolution KTP

Fig. 13 Example for low resolution KTP

The data on the KTP whose value will be taken as the success value of this research is province data, region, NIK/ ID Number, name, place/date of birth, type gender, blood type, address, RT/RW, Kel/Village, District, religion, status marriage, occupation, citizenship, valid until. For the data, as previously mentioned, labelling is done manually as a result of comparison for performance measurement, as can be seen in Figure 14.

{"0": "PROVINSI DKI JAKARTA", "1": "JAKARTA BARAT", "NIK": "3171234567890123", "nama": "MIRA SETIAWAN", "tempat_tanggal_lahir": "JAKARTA, 18-02-1986", "jenis_kelamin": "PEREMPUAN", "gol_darah": "B", "alamat": "JL.PASTI CEPAT A7/66", "rt_rw": "007/008", "kel_desa": "PEGADUNGAN", "kecamatan": "KALIDERES", "agama": "ISLAM", "status_perkawinan": "KAWIN", "pekerjaan": "PEGAWAI SWASTA", "kewarganegaraan": "WNI", "berlaku_hingga": "22-02-2017"}

Fig. 14 Example labeling KTP

4.2. Experimental Design

The experiment was carried out by making an OCR prototype using the method proposed in section 3. In addition to conducting experiments using prototypes, comparisons of each method will also be carried out. The dataset used is in the form of an Indonesian national identity card with various photo qualities. In this experiment, the error rate calculation for each character and each word will be calculated. The experiments carried out focused on data on name, place/date of birth, address (address, RT/RW, Kel/Desa, Kecamatan), marital status, and occupation. An experiment can be said to be successful when the data can be read using the proposed OCR prototype with the lowest error rate. After testing / testing, performance measurements will be carried out. Then the results obtained from these performance measurements will be compared with several other methods.

Performance measurement is done by comparing the error rate of each existing method. The formula for calculating the error rate used is the calculation of the character error rate (CER). Calculations to calculate the character error rate will use the calculation Equation 3. The CER calculation is obtained by calculating the wrong characters divided by the total characters and multiplying by 100%. This calculation will be used to measure the performance of each value on each dataset, as can be seen in Figure 15. After the measurement of each value is carried out, measuring the average CER value of each dataset uses the calculation formula Equation 4 for AvgCER measurements used to calculate the average CER from each KTP. Measurement of total error rate performance is done to do the total CER average of all available datasets using the calculation formula Equation 5. The Total Error Rate is used to calculate the average – overall average on the total existing dataset. In addition, there are calculations lowest avg CER used to find the lowest avgCER value in the entire dataset using calculation Equation 6. As for the highest avg, CER is used to find the highest avgCER value in all datasets using calculation Equation 7.

$$\text{CER} = \frac{\text{Wrong character}}{\text{Total character}} * 100\% \quad (3)$$

$$\text{AvgCER} = \frac{\text{Total CER}}{\text{Total Field}} \quad (4)$$

$$\text{Total Error Rate} = \frac{\text{Total AvgCER}}{\text{Total dataset}} \quad (5)$$

$$\text{Lowest Avg CER} = \min(\text{avgCER}) \quad (6)$$

$$\text{Highest Avg CER} = \max(\text{avgCER}) \quad (7)$$

4.3. Experimental Result

The experimental results that have been carried out after testing the dataset using OCR without any pre-processing and post-processing stages can be seen in Table 1. Table 1 shows that the results of OCR performance measurements are absent; the pre-processing and post-processing methods get the lowest avg CER value at 0.43%, while the highest avg CER value is at 97.15%, and the resulting total error rate is 37.50%. Figure 16 is an example of a KTP with the lowest avg CER of 0.43% where the identity card is clearly legible. Figure 17 is an example of a KTP with the highest avg CER. The KTP looks shaded, making the OCR unable to read properly.

```
PROVINSI DKI JAKARTA PROVINSI DKI JAKARTA True
0.0
JAKARTA BARAT JAKARTA BARAT True
0.0
3171234567890123 13171234567890123 False
5.063291139240507
MIRA SETIAWAN MIRA SETIAWAN True
0.0
JAKARTA, 18-02-1986 JAKARTA, 18-02-1986 True
0.0
PEREMPUAN PEREMPUAN True
0.0
B B True
0.0
```

Fig. 15 CER measurement of each value

Table 1. Testing results without pre-processing and post-processing

Lowest Avg CER	Highest Avg CER	Total Error Rate
0.43 %	97.15 %	37.50 %

Fig. 16 Example KTP lowest avg CER

Fig. 17 Example of KTP's highest avg CER

The experimental results were carried out after carrying out several combinations of methods pre-processing with the resulting values , as can be seen in Table 3, with legend listed in Table 2. The results from Table 3 with the most effective pre-processing method are using shadow removal and custom grayscale conversion with a total values error rate of 14.56%. The worst method is to combine gaussian blur, threshold to zero, truncate threshold, and shadow removal methods with a total error rate exceeding the total OCR error rate without pre-processing, namely 65.72%.

Table 2. Legend pre-processing

Abbreviation	Explanation
GB	*GaussianBlur*
TZ	*ThresholdToZero*
TT	*ThresholdTrunc*
SR	*ShadowRemoval*
CGS	*CustomGrayScale*

The result of table 4 is a detailed table showing the amalgamation of Shadow removal and Custom Grayscale methods. For the lowest value, avg CER decreased compared to the value of the lowest avg CER regardless of the method pre-processing to 0.31%. An example of using the lowest avg CER method pre-processing as shown in Figure 18. As for the highest avg CER value increased on the same KTP. As shown in Figure 19, results from pre-processing make the KTP unreadable.

Table 3. OCR results using pre-processing

Pre-processing					Total Error Rate
GB	TZ	TT	SR	CGS	
V					43.63
	V				40.03
		V			25.39
			V		24.02
				V	25.5
					37.18
					37.3
					37.45
					37.16
					36.41
					37.63
V	V				37.63
V		V			33.43
V			V		23.23
V				V	29.62
	V	V			41.8
	V		V		52.29
	V			V	36.03
		V	V		21.81
		V		V	16.92
			V	V	14.56
V	V	V			45.88
V	V		V		62.82
V	V			V	40.72
	V	V	V		54.45
	V	V		V	36.74
		V	V	V	37.66
V		V	V		28.34
V		V		V	28.19
V			V	V	20.05
	V		V	V	32.78
V	V	V	V		65.72
V	V	V		V	44.75
	V	V	V	V	34.38
V	V		V	V	41.48
V		V	V	V	60.48

Table 4. Details of amalgamation Shadow Removal and Custom Grayscale

Lowest Avg CER	Highest Avg CER	Total Error Rate
0.31 %	100.0 %	14.56 %

Fig. 18 Example of removal of the lowest average CER Shadow and Custom Grayscale

Fig. 19 Highest avg CER Shadow removal and Custom grayscale examples

In this section, an experiment is carried out by adding the post-processing method from the results of previous experiments with the best method. The results of previous experiments are used to use the method of shadow removal and custom grayscale conversion, which will be added methods lookup table. The results of these experiments can be seen in Table 6, with a legend in Table 5. The results in Table 6 show the value of the total error rate is nothing higher than the previous total error rate, which was 14.56%. For total value, the highest error rate after the combination of pre-processing and post-processing methods was 14.54%, while the lowest value was 13.94%.

Table 5. Legend post-processing

Abbreviation	Explanation
MPr	*Mapping Provinsi*
MD	*Mapping Daerah*
MA	*Mapping Agama*
MP	*Mapping Pekerjaan*
AD	*add default WNI*
MS	*Mapping Status Perkawinan*

Table 6. OCR results using pre-processing and post-processing

Pre-processing		Post Processing						Total Error Rate
SR	CGS	MPr	MD	MA	MP	AD	MS	
V	V	V						14.54
V	V	V	V					14.48
V	V	V	V	V				14.32
V	V	V	V	V	V			14.19
V	V	V	V	V	V	V		13.94
V	V	V	V	V	V	V	V	14.12

Fig. 20 Example of lowest avg CER pre-processing and post-processing

Fig. 21 Example of the highest avg CER pre-processing and post-processing

Table 7 is a detail of the combination of pre-processing and post-processing with the lowest total error rate. For lowest avg CER and highest avg CER has not changed compared to using a pre-processing method only. KTP with the lowest avg CER and highest avg CER too unchanged, as can be seen in Figures 20 and 21.

Based on the experimental results, as shown in Table 8, the pre-processing method and post-processing can improve OCR performance. A decrease in the value of the total error rate can be evidence of it. The resulting total error rate is 37.5% without using the pre-processing and post-processing methods. Whereas after adding the pre-processing method, the value of the total error rate produced decreased to more than half, namely 14.12%. The method best pre-processing combination based on the total error rate is Shadow Removal and Custom Grayscale. The value of the lowest avg CER also suffers decreased from 0.43% to 0.31%. The post-processing method uses a lookup table also makes OCR performance better. The total error rate shows this decreased from 14.12% to 13.94%. Therefore, the recommendation of the best pre-processing method is to use Shadow Removal and Custom Grayscale, and the post-processing method that can be recommended is a lookup table.

Table 7. Detail pre-processing and post-processing have the lowest total error rate

Lowest Avg CER	Highest Avg CER	Total Error Rate
0.31 %	100.0 %	13.94 %

Table 8. Detailed summary of evaluation results

Methods	Lowest Avg CER	Highest Avg CER	Total Error Rate
OCR	0.43 %	97.15 %	37.50 %
Pre-processing and OCR	0.31 %	100.0 %	13.94 %
Pre-processing, OCR and post-processing	0.31 %	100.0 %	13.94 %

5. Conclusion and Future Work

In this research, implementation and evaluation have been carried out methods - pre-processing and post-processing methods. Several pre-processing methods that have been carried out are gaussian blur, threshold to zero, truncate threshold, shadow removal, and custom grayscale conversion. Based on the results of the data evaluation shows that the best combination of pre-processing methods is to perform shadow removal and custom grayscale conversion with a total error value rate of 14.56%. Post-processing methods using lookup tables can also be used to improve OCR performance results. Based on the results of existing data, there is no value in the total post-processing error rate that exceeds the OCR total error rate using the best pre-processing method. The lookup table method cannot be used in all lookup table values created. It can be seen based on the evaluation results. The value from the lookup table when doing the lookup table on marital status, the value of the total error rate of 13.94% becomes higher than the previous 14.12%. So, the best lookup table method is by doing a lookup table on province, region, religion, occupation, and nationality with a total error rate of 13.94%. It can be concluded that using a combination of pre-processing shadow removal and custom grayscale methods, as well as post-processing methods lookup table on province, region, religion, occupation, and nationality values, can improve the performance of the accuracy of values generated from OCR.

This study only focuses on pre-processing and post-processing methods. Meanwhile, OCR itself is still using the existing library without any changes. OCR used in this research is Tesseract 4.1. If, in the future, OCR is found to perform better, then it can be done for further research. In the future, the model can also be evaluated using the pre-processing method based on machine learning, while the post-processing method can use an error correction algorithm.

References

[1] Sameeksha Barve, "Optical Character Recognition Using Artificial Neural Network," *International Journal of Advance Technology And Engineering Research (IJATER)*, vol. 2, no. 2, pp. 139-142, 2012. [Publisher Link]

[2] Mande Shen, and Hansheng Lei, "Improving OCR Performance with Background Image Elimination," *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1566 - 1570, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[3] Di Ma, and Gady Agam, "A Super Resolution Framework for Low Resolution Document image OCR," *SPIE-IS&T Electronic Imaging*, vol. 8658, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[4] Matteo Brisinello et al., "Improving Optical Character Recognition Performance for Low Quality Images," *International Symposium ELMAR*, pp. 167 - 171, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[5] Abdeslam El Harraj, and Naoufal Raissouni, "OCR Accuracy Improvement on Document Image through a Novel Pre-Processing Approach," *Signal & Image Processing : An International Journal (SIPIJ)*, vol. 6, no. 4, pp. 1 - 18, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[6] Christopher Kanan, and Garrison W. Cottrell, "Color-to-Grayscale: Does the Method Matter in Image Recognition?," *PLoS ONE*, pp. 1-8, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[7] Habeeb Imad Qasim, Al-zaydi Zeyad Qasim Habeeb, and Abdulkhudur Hanan Najm, "Selection Technique for Multiple Outputs of Optical Character Recognition," *Eurasian Journal of Mathematical and Computer Applications*, vol. 8, no. 2, pp. 41-51, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Ram Krishna Pandey et al., "Binary Document Image Super Resolution for Improved Readability and OCR Performance," *arXiv, Computer Vision and Pattern Recognition*, pp. 1-13, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Karimi Mostafa, Veni Gopalkrishna, and Yu Yen-Yun, "Illegible Text to Readable Text: An Image-to-Image Transformation using Conditional Sliced Wasserstein Adversarial Networks*," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-10, 2019.[Google Scholar] [Publisher Link]

[10] Simple Batra, "Word Extraction Using X-Y Cut Algorithm," *Journal of Engineering Research and Application*, vol. 8, no. 12, pp. 60 - 63, 2018. [CrossRef] [Publisher Link]

[11] Imad Qasim Habeeb, Zeyad Qasim Al-Zaydi, and Hanan Najm Abdulkhudhur, "Enhanced Ensemble Technique for Optical Character Recognition," *New Trends in Information and Communications Technology Applications*, pp. 213-225, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Shruti Rijhwani, Antonios Anastasopoulos, and Graham Neubig, "OCR Post Correction for Endangered Language Texts," *arXiv, Computation and Language*, pp. 5931-5942, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Kusnantoro Kusnantoro, Tatang Rohana, and Dwi Sulistya Kusumaningrum, "Implementasi Metode Tesseract OCR (Optical Character Recognition) untuk Deteksi Plat Nomor Kendaraan Pada Sistem Parkir," *Scientific Student Journal for Information, Technology and Science*, vol. 3, no. 1, pp. 59-67, 2022. [Google Scholar] [Publisher Link]

[14] Satya Mallick, Image Thresholding in OpenCV, 2015. [Online]. Available: https://learnopencv.com/opencv-threshold-python-cpp/

[15] Yun-Hsuan Lin, Wen-Chin Chen, and Yung-Yu Chuang, "BEDSR-Net: A Deep Shadow Removal Network from a Single Document Image," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12905-12914, 2020. [Google Scholar] [Publisher Link]

[16] Saikiran Subbagari, "Leveraging Optical Character Recognition Technology for Enhanced Anti-Money Laundering (AML) Compliance," *SSRG International Journal of Computer Science and Engineering*, vol. 10, no. 5, pp. 1-7, 2023. [CrossRef] [Publisher Link]

[17] Vaibhav Kumar, "Recurrent Neural Network based Language Modeling for Punjabi ASR," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 9, pp. 7-13, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Piyush Kiran Redgaonkar et al., "Imageprocessing Based Pincode Recognizing and Sectionwise Courier Sorting System," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 3, no. 3, pp. 16-18, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[19] Pooja Goyal, Sushil Kumar, and Komal Kumar Bhatia, "Hashing and Clustering Based Novelty Detection," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 6, pp. 1-9, 2019. [CrossRef] [Publisher Link]

[20] Asif Ansari, and NM. Sreenarayanan, "Analysis of Text Classification of Dataset Using NB-Classifier," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 6, pp. 24-28, 2020. [CrossRef] [Google Scholar] [Publisher Link]