Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 4, August 2024, pp. 1043-1050

DOI: https://doi.org/10.52436/1.jutif.2024.5.4.1489
p-ISSN: 2723-3863
e-ISSN: 2723-3871

INTEGRATION OF ESP32-CAM WITH ANDROID AND IOT BASED ENGLISH-INDONESIAN TRANSLATION APPLICATION USING OCR TECHNOLOGY

Siti Nurhaliza[*1], Kirana Alyssa Putri[2], Iqlimah Attyyatullatifah[3], I'zaaz Akhdan Muhadzdzab[4], Atiqah Meutia Hilda[5]

[1,2,3,4,5]Informatics, Faculty of Industrial Technology and Informatics, Universitas Muhammadiyah Prof. DR. Hamka, Indonesia
Email: [1]lizzaaxo46@gmail.com, [2]kiranaalyssa.putri@gmail.com, [3]Iqlimahattyyatullatifah0372@gmail.com, [4]jazzdev01@gmail.com, [5]atiqahmeutiahilda@uhamka.ac.id

(Article received: October 25, 2023; Revision: December 28, 2023; published: July 29, 2024)

Abstract

Language is a constant element in global human interaction, particularly English. This research presents the design and development of an innovative Android and IoT-based translation application, which facilitates seamless English-Indonesian translation. By utilizing Optical Character Recognition (OCR) technology for text input, the app is enhanced by the integration of ESP32-CAM, a versatile microcontroller with a camera module. This unique combination promises accurate and efficient translation, bridging language barriers while exploring the potential of the Internet of Things (IoT) in linguistic applications. This research reveals the intricate process of creating this translator tool, using the Dart programming language and Flutter framework in Android app development, with the support of Visual Studio Code as the software development environment, as well as the Arduino IDE for the ESP32-CAM microcontroller. It shows how OCR technology and ESP32-CAM significantly enhance the translation experience in an increasingly connected world.

Keywords: *ESP32-CAM, Internet of Things, Optical Character Recognition, Translator tool.*

1. INTRODUCTION

Communication through language is a key foundation of everyday life, allowing individuals to engage in activities and interactions that transcend international borders. At the forefront of these global interactions is English, an essential element in human communication. Its status as an international lingua franca emphasizes its role in bridging language differences, encouraging cross-cultural exchange, and facilitating access to diverse knowledge. As a result, many educational materials and scientific publications are more readily available in English. Therefore, learners, especially students, need to understand English to access the world of learning and discovery. Consequently, learners, especially students, are compelled to understand the intricacies of the English language in order to access the world of learning and discovery [1], [2].

In an increasingly advanced and connected world, cross-language communication has become a major barrier of concern, especially for modern society. Translation tools play a vital role in bridging the linguistic gap. Thus, translators allow people with different languages to stay connected and exchange information smoothly. With developments in the world of technology, especially in the field of Internet of Things, there is an increasing demand for innovative and efficient translator applications [3], [4].

The development of Android and IoT-based translator applications that provide English to Indonesian translation facilities and vice versa is a form of progress in the field of technology and language. This application, which uses Optical Character Recognition (OCR), allows for text recognition and detection through optical mechanisms. This can be a form of innovation for users who want to understand English documents. The integration of ESP32-CAM further enhances the ability of this application to provide accurate and efficient translation results [5]. Moreover, the integration of ESP32-CAM, an adaptable microcontroller equipped with a camera module, enhances the application's capabilities, assuring highly accuracy and efficient translations. ESP32-CAM facilitates image capture, which is then swiftly processed by OCR technology to deliver quick and accurate translations [6], [7].

OCR, or also known as Optical Character Recognition, is a technology or a computer system that changes or converts images from various source, including scanned, printed, handwritten text, PDFs, or images that captured by camera, into an editable text. The camera, for example digital camera can be problematic due to some issues like edge distortion, and inadequate lighting. Translation accuracy depends on image's clarity and the chosen method. Therefore, tesseract selected due to its extensive use, flexibility, and active developer community [8], [9].

Dart programing language was chosen to develop this translation tool. Beside that, the dart programming language also integrated with the flutter framework. The reason why dart was selected to develop this app is beacause of its efficiency and also its compability to cross-platform mobile app development. By integrating dart programing language and flutter, the app developed as a flexible and highly responsive android application, guaranteeing a seamless user experiences [10], [11], [12]. The development occurred using the software development environment of Visual Studio Code, recognized for its strong features and capabilities in code editing. Visual Studio Code provides an efficient platform for coding, testing, and also debugging. Beside software, hardware integration is also an important component. Arduino IDE also utilized for the ESP32-CAM microcontroller, an important element of the application. The strategic combination of programming language, development environment, also the hardware integration work collaborativelly to create an efficient translation application. Users can also enjoy a seamless experience when using this app.

A similar study was carried out by Aulia and his colleagues, which discussed the creation of an android-based OCR Japanese-Indonesian language translator application. They developed an Android application that utilizes a smartphone camera to detect Japanese kanji characters, display their meanings, and provide pronunciation guidance on the screen. The app, developed using Java and XML in Android Studio, along with the Tesseract OCR engine and Yandex Translate API, achieved an 87.5% accuracy rate in black box testing. However, Aulia's app is different from this study. Aulia's research focuses on Japanese language learning and application development to help read kanji characters and understand Japanese. Meanwhile this research is focused on the development of an English-Indonesian translation application using OCR technology and the ESP32-CAM [13].

Another similar study also carried out by Andrew and his colleagues, titled "Automatic Number Plate Recognition (ANPR)", which exploring the challenge of recognizing vehicle license plates. Algorithms designed for license plate recognition rely on these distinct features, making it challenging to achieve a universal solution, as image analysis techniques themselves cannot guarantee 100% accuracy. Andrew's research focuses on proposing an algorithm optimized for Ghanaian vehicle license plates. The algorithm, developed in C++ with the OpenCV library, utilizes edge detection, Feature Detection techniques, and mathematical morphology for plate localization. Meanwhile this research leverages OCR technology and the ESP32-CAM microcontroller to achieve efficient and accurate translation, according to the language of interest [14].

There are several differences between the translation application designed in this study and previous studies. This study explains the design process in building a translator tool with the aim of improving language accessbility and facilitating cross-lingual communication. By exploring the potential of OCR Tesseract and integrating it with the IoT devices, this study utilizes various technologies to overcome the challenges of language barriers effectively. This kind of innovation not only enrich the communication experience, but also opens up opportunities for cultural exchange and profound collaboration [15], [16].

2. METODE PENELITIAN

2.1. System Architecture

Figure 1 below is the system architecture for an Android and IoT translation application that utilizes ESP32-CAM as its device and integrates tesseract OCR technology for application development.

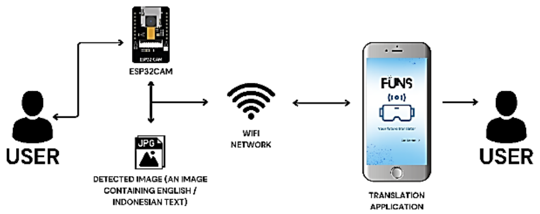

Figure 1. System Architecture Design

Users who use the application will initiate it by first capturing an image using ESP32-CAM. Then, the captured image will be processed by the system using OCR technology in order to transform it into textual data. Afterwards, by utilizing the same wifi network, ESP32-CAM and the android device that used for the application, both will communicate to display the textual data that has been processed by OCR and the translation results to the application. Thus, users can see the translation result of words or sentences they want to translate in the application.

2.2. System Architecture

The use case diagram from the translator application shown in Figure 2 explain that if user wants to do translation in application by capturing the image first in the system, they need to login first. If they have logged in, sebsequently, they can capture the image of the text that they wanted to detect and translate. Once the text detected, the detected text will be translated. Afterward, either the detected text or the translated text will be displayed on the dashboard of the application.

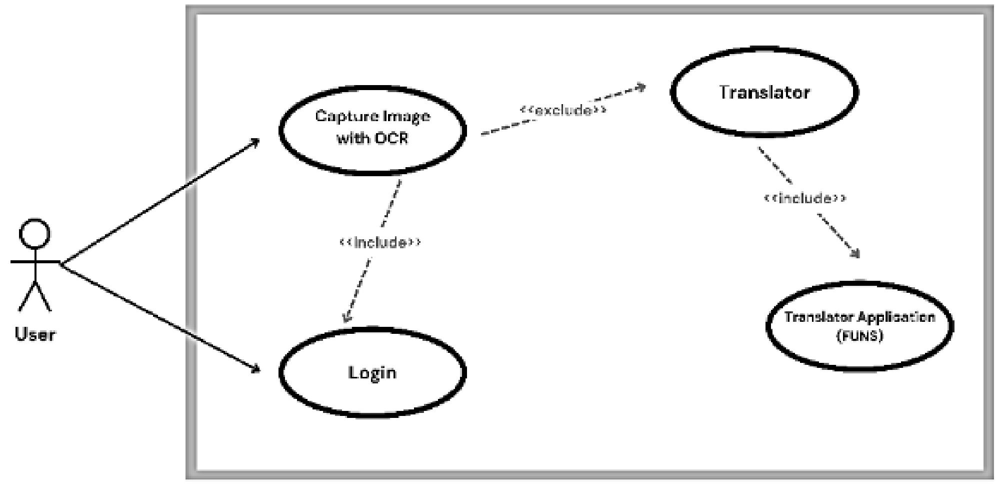

Figure 2. Use Case Diagram

2.3. Specification Requirements

At this stage, the author will mention the data requirements as well as connections, software to create applications, also hardware to create applications.

A. Hardware

The hardware used to design, and develop the application is an Asus M409D notebook with the following specifications:

- Processor : Ryzen 3 3200u
- OS : Microsoft Windows 11 (64-bit)
- Memory : 12 GB RAM
- Storage : 128 GB SSD, 1 TB HDD
- Graphics : AMD Vega3

B. Software

Supporting software used in the design and development of this English and Indonesian translator applications with OCR technology and Android-based that integrated with ESP32-CAM are:

- Visual Studio Code
- Arduino IDE
- Flutter

C. Data Requirement and Connection

When using this translator application, users must first ensure that the android device and ESP32-CAM are connected to the same wifi network. The data that used to test the application is an English and Indonesian language module printed on paper. There are two objects in the module, i.e. text and images. This aims to ensure that when the test is done, OCR will only detect the text. Regarding how well the writing object will be detected and translated, it all depends on how the capturing technique and light intensity when taking the pictures.

2.4. Application Testing

The testing of the application is conducted by using Black-Box method and calculating the value of OCR in recognizing characters from text that printed on paper, where some of the papers contain words, sentences, even images. The purpose of using image as on of object in this testing is to assess whether objects other than text will be detected or not. The equation used to calculate the accuracy value is:

$$Accuracy = \frac{All\ detected\ words}{All\ input\ words} \times 100\% \qquad (1)$$

2.5. User Interface Design

The step of designing the user interface of this application is using Figma. The application consists of 3 pages, with one main page as the dashboard page, which will be used as a page to display the results of detected words or sentences as well as the translation results. In addition, there are two other pages, namely splash screen page which will display logo of the application and serves as the initial page that appears when the user opens this translation application, and the other one is the login page. The UI design of this translation application shown in Figure 3 below.

Figure 3 is the user interface design of the splash screen page. Splash screen page is the first page that will appear when the user opens this translation application. On splash screen page, there is “get started” button, which, when usser presses that button, it will immediately redirect to the next page, namely the login page. On login page, user will enter email or ID along with the password that provided by the application’s creator. Furthermore, when the user presses the login button, it will redirect them to the dashboard page that also shown in Figure 3. On dashboard page, user can capture or detect the text object they want to translate and bring up the translation results at the same time. In Figure 3 on dashboard page, to perform text detection and translation is done by pressing the "Detect and Translate" button. Then if you want to change the language translation format, from English to Indonesian or Indonesian to English, you only need to press the direction button located just below the logo on the dashboard page.

Figure 3. UI Design (Splash Screen Page, Login Page, and Dashboard Page)

3. RESEARCH RESULTS

3.1. System Building Phase

In developing the system, it started by first analyzing the requirements for the design and development of the application. The main components needed to develop this application are hardware along with software to write code. In this case, the IDE used is Visual Studio Code with the Flutter framework, which will be used to write the code for the main part of the application, namely text detection with OCR technology and translating text. In this context, the translation library used is "package:translator/translator.dart". As for the installation of ESP32-CAM as the main camera of the application, Arduino IDE is used.

The very first step for building this application is to start by installing ESP32-CAM using Arduino IDE. Once installed, and showing the camera device on the ESP32-CAM is functioning properly, the next step is to write the code for the main application. Beginning with ensuring that all libraries that will be used for building this application have been downloaded, then the next step is to write the application creation code for three pages, namely the splash screen page, login page, and dashboard page. The code files for Splash Screen, Login, and Dashboard page are organized as shown in Figure 4 below.

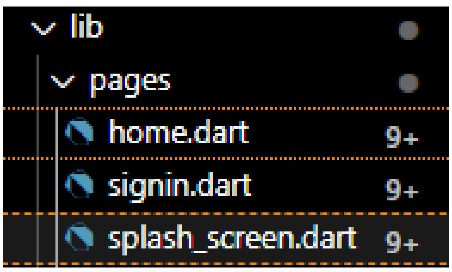

Figure 4. Coding Files for Splash Screen, Login, and Dashboard

After completed writing all the codes for the application, the next phase is to integrate the ESP32-CAM with the application. This is in order to make the camera device on ESP32-CAM used as the main camera to detect text on application.

3.2. Testing Results

Application testing is carried out using the Black-Box method, by using papers that have been printed as the test media. On the test media there are not only text objects. Instead, there is also an image object. This aims to test whether OCR in the application will successfully detect text objects only or not. In addition, testing is also done on text objects in the form of long sentences.

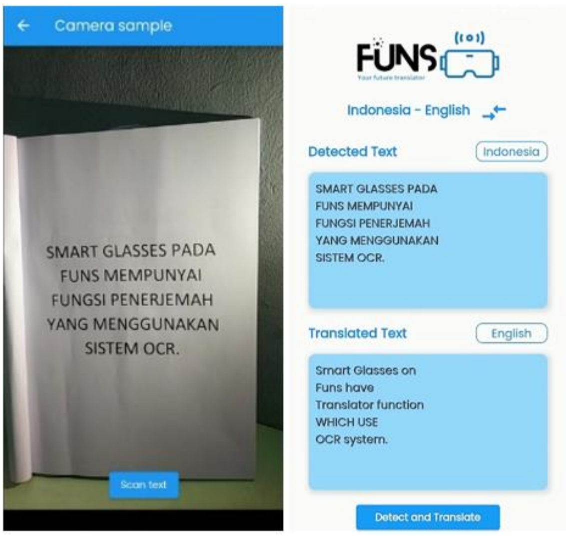

Figure 5. Testing of Sentence-shaped Written Objects Detection and Translation from Indonesian to English

In Figure 5, there is an example of an application trial using test media in the form of sentences on printed paper. On the left side of Figure 5, the ESP32-CAM that has been integrated and connected to the same Wi-Fi network as the android device used to test the application capture the pictures. Afterward, on the right side of Figure 5, it is shown that the OCR has successfully worked, and each word on the test media is detected and displayed in the "Detected Text" box. Then, the words forming the sentence are translated by the system, and the translation result is displayed on the same page, in the "Translated Text" box. In this case, what is being done is the translation from Indonesian to English.

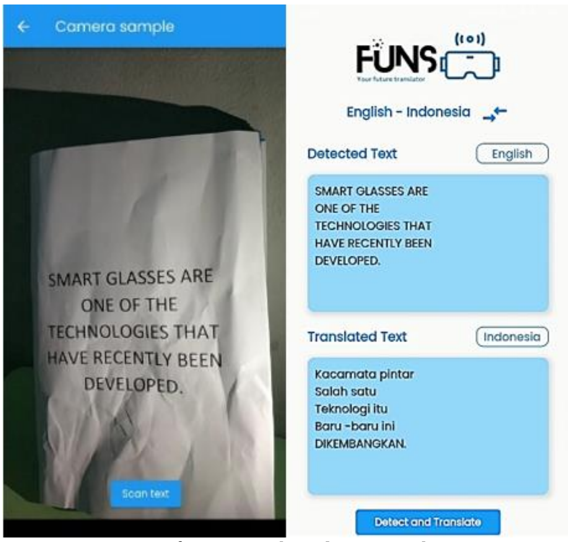

Figure 6. Testing of Sentence-shaped Written Objects Detection and Translation from English to Indonesian

In Figure 6, there is also an example of application trial using test media in the form of sentences on printed paper. But, in this case what is being done is the translation from English to Indonesian. To switch the translation format from English-Indonesian to Indonesian-English, or vice versa, simply press the two-way button located just below the app logo on the dashboard page.

In Figure 7, there is another example of an application trial using test media in the form of text and image object that printed on paper. As shown in Figure 7, it has been proven that OCR has succeeded by only detecting text objects, so that the other objects such as images or the others kind of objects will not be detected and displayed in the application.

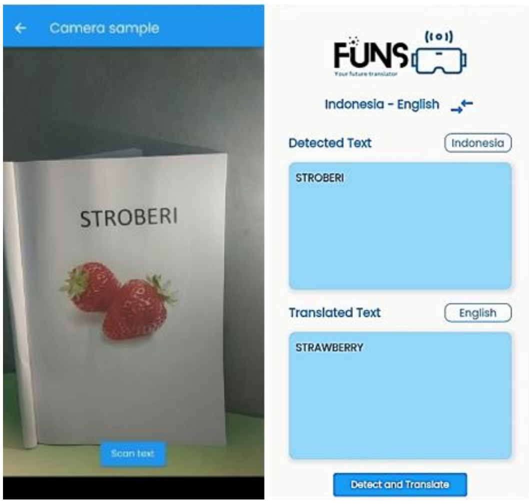

Figure 7. Testing of Word and Image Objects Detection and Translation from Indonesian to English

In Figure 8, the trial was conducted by taking an image in landscape mode. Even though the image was taken in landscape format, OCR was still able to successfully detecting the text that contained in the test media.

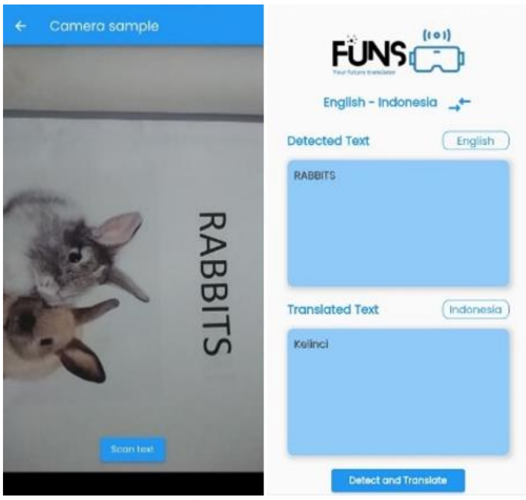

Figure 8. Testing of Word and Image Objects Detection (Landscape Version) and Translation from English to Indonesian

From some of the experiments carried out above and some other experiments, it can be tested using the Black-Box method. The first test is testing the word. Then, the next is testing the sentence, and the image capture mode (landscape and portrait). The following is the conducted test:

Table 1. Black-box Testing Method on Words

Input	Sample Test	OCR Result	Translation Result
One Word and an Image	STROBERI (id)	Text detected successfully: STROBERI	Translation successful: STRAWBERRY (en)
One Word and an Image	RABBITS (en)	Text detected successfully: RABBITS	Translation successful: KELINCI (id)
Two Words	Translator Glasses (id)	Text detected successfully: Translator Glasses	Translation failed: Translator Glasses (id)
Two Words and an Image	Malaikat Pelindung (id)	Text detected successfully: Malaikat Pelindung	Translation failed: Guardian angel (id)
Some Words and Symbol	Hello, how are you? (en)	Text detected successfully: Hello, how are you?	Translation successful: Halo, apa kabarmu? (id)

In the test as shown in Table 1, there was one test that failed. The reason of this failed test is because at the time of text detection, the translation language format had not been changed. In the failed test, the translation language format should have been English-Indonesian, but because it had not been changed, the translation process failed.

Table 2. Black-box Testing Method on Sentences

Input	Sample Test	OCR Result	Translation Result
One Sentence	SMART GLASSES PADA FUNS MEMPUNYAI FUNGSI PENERJEMAH YANG MENGGUNAKA SISTEM OCR (id)	Text detected successfully: SMART GLASSES PADA FUNS MEMPUNYAI FUNGSI PENERJEMAH YANG MENGGUNAKA SISTEM OCR	Translation successful: Smart Glasses on Funs have Translator function which use OCR system (en)
One Sentence	SMART GLASSES ARE ON OF THE TECHNOLOGIES THAT HAVE RECENTLY BEEN DEVELOPED (en)	Text detected successfully: SMART GLASSES ARE ON OF THE TECHNOLOGIES THAT HAVE RECENTLY BEEN DEVELOPED	Translation successful: Kacamata pintar salah satu teknologi itu baru-baru ini dikembangkan (id)
Some Words	Hello, world. My name is Liza. My dream is going to United Kingdom and become an Oxford's student. I wish I can realize my dream as soon as possible. See you soon, UK! (en)	Text detected successfully: Hello, world. My name is Liza. My dream is going to United Kingdom and become an Oxford's student. I wish I can realize my dream as soon as possible. See you soon, UK!	Translation successful: Halo, dunia. Nama saya Liza. Mimpiku adalah pergi ke Inggris dan menjadi siswa Oxford. Saya berharap saya bisa mewujudkan impian saya secepat mungkin. Sampai jumpa, Inggris! (id)

Table 3. Black-box Testing Method on Potrait and Landscape Mode

Input	Sample Test	OCR Result	Translation Result
Potrait	STROBERI (id)	Text detected successfully: STROBERI	Translation successful: STRAWBERRY (en)
Landscape	RABBITS (en)	Text detected successfully: RABBITS	Translation successful: KELINCI (id)

The results of black-box testing method in testing sentences shown in Table 2. Table 2 explains that all of the testing is successfully done. Meanwhile, Table 3 presents the results of black-box testing method in testing potrait and landscape mode. Same as Table 2, all of the results in Table 3 also successfully done.

Next is the stage of testing the accuracy of OCR in detecting text. From the experiments conducted in Table 1, Table 2, and Table 3, we can calculate the accuracy value of OCR in detecting text in these experiments. In the experiment, 8 experiments were conducted with a total of 66 words. From 66 samples test, the number of words that detected properly was 66 words. So that the accuracy value of character recognition performed by OCR is calculated by Formula (1) as follows:

$$Accuracy = \frac{66}{66} \times 100\% = 100\%$$

Thus, the accuracy rate of OCR in detecting text in the application is 100% based on the experiments and calculations that have been carried out. In terms of detection, the system did not experience any failure in recognizing the text. However, in translation, one failure was experienced due to a mistaken format of the translated language.

4. DISCUSSION

This research introduces a robust Android and IoT-based translation application, utilizing OCR technology and ESP32-CAM integration to facilitate seamless English-Indonesian translation. Developed through the Dart programming language and Flutter framework, the system features a user-friendly interface that ensures a streamlined and efficient translation experience. The success of this application is underscored by OCR's flawless 100% accuracy in detecting text, even in challenging scenarios such as processing long sentences and diverse content types.

Taking previous works as reference on OCR technology, this research puts a distinct emphasis on the innovative contribution of integrating ESP32-CAM for image capture and translation enhancement. The structured design of the user interface enhances accessibility, with special attention paid to language settings to reduce potential translation errors.

This research not only highlights the consistency of the application in producing accurate translations, demonstrated through the successful translation of all tested data samples into the desired language, but also underlines the flawless performance of the OCR component. The ability of this technology to detect text with 100% accuracy within the app signifies a remarkable achievement.

The broader implications of this research also suggest a comprehensive solution to overcome language barriers, emphasizing the potential of OCR and IoT integration in linguistic applications. By seamlessly combining these technologies, this research paves the way for advanced language translation tools that can navigate diverse types of content and deliver precise results, contributing significantly to the field of language processing and communication.

5. CONCLUSION

In the development of this system, the first step is to analyze the needs to design and develop the application. The main components required include hardware and software for writing code. The development of this application uses Visual Studio Code with the Flutter framework and Arduino IDE to install the ESP32-CAM.

After the ESP32-CAM integrated with the application to be used as the main camera in detecting text, the next step is to test the application by using Black-box method and calculating the accuracy value of OCR in detecting text. The results show that OCR can detect text well, even on long sentences. From those experiments, it can be proven that OCR can only detects text and symbols like (.,!@#$%^&*) and will not detect images or other objects. Text translation was successfully performed from Indonesian to English. Although there was one failed test due to a translation language format error, the accuracy rate of OCR in detecting text was 100% based on the experiments conducted.

Therefore, the system was successful in detecting and translating the text, although special attention needs to be paid to the language format settings. So the conclusion is, OCR has successfully detected the text, and the translator application has also successfully provided translation results from the detected objects.

REFERENCES

[1] A. Kevin Wiguna, N. Suciati, and W. N. Khotimah, "Aplikasi Penerjemah Gambar Teks Berbahasa Inggris Menggunakan Teknologi Realitas Tertambah pada Perangkat Berbasis Android," *J. Tek. ITS*, vol. 8, no. 1, 2019, doi: 10.12962/j23373539.v8i1.40070.

[2] M. Patasik, N. Nirwana, and ..., "Penerapan Algoritma Brute Force Pada Aplikasi Penerjemah Bahasa Inggris-Toraja Berbasis Android," *SISITI Semin. Ilm. ...*, vol. X, no. 1, pp. 40–49, 2021, [Online]. Available: https://www.ejurnal.dipanegara.ac.id/index.php/sisiti/article/view/786.

[3] A. S. Putera and D. K. S. Putra, "KOMUNIKASI LINTAS BUDAYA DALAM PROSES BELAJAR BAHASA INGGRIS DI KAMPUNG INGGRIS PARE KEDIRI," *Communicology*, vol. 7, no. 1, pp. 1–31, 2019, doi: https://doi.org/10.21009/COMMUNICOLOGY.14.01.

[4] M. Arif Suryawan, N. Syahkintan, and A. Bahasa Wakatobi, "Rancang Bangun Aplikasi Penerjemah Bahasa Daerah Wakatobi Berbasis Android Android-Based Design of Wakatobi Regional Language Translator," *J. Inform.*, vol. 11, no. 1, pp. 1–9, 2022, [Online]. Available: http://ejournal.unidayan.ac.id/index.php/JIU.

[5] T. E. Prassetiyo, "Rancang Bangun Aplikasi Penerjemah Bahasa Indonesia-Jawa Menggunakan Optical Character Recognition," *Ubiquitous Comput. its Appl. J.*, vol. 4, no. 1, pp. 37–40, 2021, [Online]. Available: https://www.nesabamedia.com/pengertian-.

[6] M. K. Dhanunjaya and A. S. Kumar, "Laser Based Security System," *Laser*, no. 4, pp. 2299–2303, 2023, [Online]. Available: http://www.journal-iiie-india.com/1_apr_23/273_online.pdf.

[7] A. Isrofi, S. N. Utama, and O. V. Putra, "RANCANG BANGUN ROBOT PEMOTONG RUMPUT OTOMATIS MENGGUNAKAN WIRELESS KONTROLER MODUL ESP32-CAM BERBASIS INTERNET of THINGS (IoT)," *J. Teknoinfo*, vol. 15, no. 1, p. 45, 2021, doi: 10.33365/jti.v15i1.675.

[8] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[9] G. Abdul Robby, A. Tandra, I. Susanto, J. Harefa, and A. Chowanda, "Implementation of optical character recognition using tesseract with the javanese script target in android application," *Procedia Comput. Sci.*, vol. 157, pp. 499–505, 2019, doi: 10.1016/j.procs.2019.09.006.

[10] Nelly Sofi and Riza Dharmawan, "Perancangan Aplikasi Bengkel Csm Berbasis Android Menggunakan Framework Flutter (Bahasa Dart)," *J. Tek. dan Sci.*, vol. 1, no. 2, pp. 53–64, 2022, doi: 10.56127/jts.v1i2.125.

[11] S. Tjandra and G. S. Chandra, "Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang," *J. Inf. Syst. Hosp. Technol.*, vol. 2, no. 02, pp. 76–81, 2020, doi: 10.37823/insight.v2i02.109.

[12] D. R. Poetra, "Performa Algoritma Bubble Sort dan Quick Sort pada Framework Flutter dan Dart SDK(Studi Kasus Aplikasi E-Commerce)," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 2, pp. 806–816, 2022, doi: 10.35957/jatisi.v9i2.1886.

[13] A. Rizqi, N. Kasan, and Widianto, "Rancang Bangun Aplikasi Penerjemah Bahasa Jepang – Indonesia Menggunakan OCR Berbasis Android," *Artik. Ilm. Tek. Elektro*, vol. 1, no. 3, pp. 231–241, 2019.

[14] A. S., J. Yankey, and E. O., "An Automatic Number Plate Recognition System using OpenCV and Tesseract OCR Engine," *Int. J.*

Comput. Appl., vol. 180, no. 43, pp. 1–5, 2018, doi: 10.5120/ijca2018917150.

[15] I. N. T. Lestari and D. I. Mulyana, "Implementation of Ocr (Optical Character Recognition) Using Tesseract in Detecting Character in Quotes Text Images," *J. Appl. Eng. Technol. Sci.*, vol. 4, no. 1, pp. 58–63, 2022, doi: 10.37385/jaets.v4i1.905.

[16] G. Poorani, B. R. K. Krishnna, T. Raahul, and P. P. Kumar, "Number Plate Detection Using YOLOV4 and Tesseract OCR," *J. Pharm. Negat. Results*, vol. 13, no. SO3, pp. 130–136, 2022, doi: 10.47750/pnr.2022.13.s03.021.