ICIC Express Letters ICIC International ©2023 ISSN 1881-803X
Volume **17**, Number **6**, June **2023** pp. **717–724**

ANALYSIS TRANSFER DATA IMAGE PROCESSING AND FACE RECOGNITION USING CAMERA ESP32CAM WEB BROWSER IOT

Tukino[1,4,*], Rizki Aulia Nanda[2], Gunawan[3], Sutarto Wijono[4]
Sri Yulianto Joko Prasetyo[4] and Suryasatriya Trihandaru[4]

[1]Department of Information System
Faculty of Computer Science
[2]Department of Mechanical Engineering
Faculty of Engineering
University of Buana Perjuangan Karawang
HS.Ronggo Waluyo Street, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361, Indonesia
rizki.auliananda@ubpkarawang.ac.id
*Corresponding author: tukino@ubpkarawang.ac.id

[3]Department of Technical Information
STMIK YMI Tegal
Pendidikan Street No. 1, Pesurungan Lor, Margadana, Tegal City, Jawa Tengah 52122, Indonesia
gunawan@stmik-tegal.ac.id

[4]Department of Computer Science
University of Kristen Satya Wacana
Salatiga, Central Java 50711, Indonesia
{ sutarto.wijono; sri.yulianto; suryasatriya }@uksw.edu
*Corresponding author: 982021005@student.uksw.edu

Received July 2022; accepted October 2022

Abstract. *For the current attendance needs of employees, the use of face detectors and face recorders has been widely used. However, attendance with the use of the face recorder is taking on a larger and larger shape, but the development of the ESP32CAM component is one of the main solutions for the use of face recognition and face recorder with small and easy-to-obtain components without incurring high costs. However, mistakes frequently occur when the face cannot be recognized quickly and the influence of light is not favorable. As a result of these issues, the goal of this research is to examine the data transfer and visual transfer of the ESP32CAM using the IoT web browser. Some data were obtained from the test. Data transfer speed, visual transfer speed, face recognition, and face detection are all factors to consider. The data generated during the morning test is 1400 b/s and 5 fps. The maximum that has occurred up to that point is the 70-second test. The data obtained during the night test is a data transfer speed of 300 b/s and a visual transfer speed of 3 fps for 90 seconds. When the test is finished, the morning test has an 80 percent presentation and the night test has a 15 percent presentation at a frame rate of 1500 b/s fps. The test was successful because it is capable of performing face recognition and detection. It has limitations in nighttime testing due to the lack of light assistance on the camera sensor. It is hoped that the output of the mechatronic and electrical systems will be used in everyday life as a result of this test.*
Keywords: Image processing, ESP32CAM, Transfer data, Face recognition, Face detection

1. **Introduction.** The development of visual detection devices on webcams has been adopted in the workplace with the aim of obtaining special face recording devices for data retrieval, face recording data intends to conduct surveys and other purposes, but there are several obstacles encountered in recording faces using webcams, including IoT networks problem, camera exposure contrast, and transfer speed. In this study, we will

DOI: 10.24507/icicel.17.06.717

look at how image processing data transfer problems occur and how to fix them to create customized face recorders with ESP32CAM and Arduino Uno as data storage components. Several sources have also explained how image processing can take place. Image processing involves several stages, including determining the object to be studied, distinguishing objects, image analysis, color segmentation, and classification. Using this approach, we generate an image processing algorithm method that detects, normalizes colors, selects different colors, and establishes detection of objects with different colors [1]. See Figure 1, which shows the results of color segmentation and color selection.

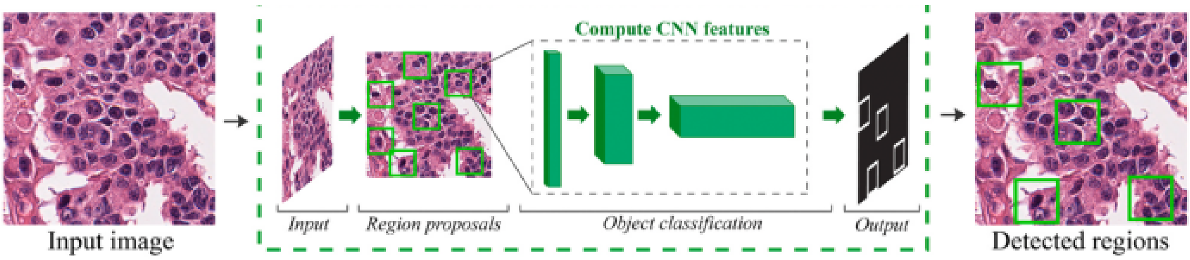


FIGURE 1. Detection regions

Image processing is a low-level component of a more comprehensive image analysis or computer vision system. The results of image processing can have a major impact on the high-level part that performs the recognition and interpretation of image data. Deep learning has recently become popular for solving low-level vision problems such as super image resolution, painting, decolorization, and coloring. Given the interconnected nature of many image processing tasks, it is normal to assume that a model that has been trained previously on one data set will be useful for another [2]. However, there are several studies that have applied pretraining to various image processing tasks. Image processing can run smoothly and produce color images without defects when using the transformer technique, as shown in Figure 2.

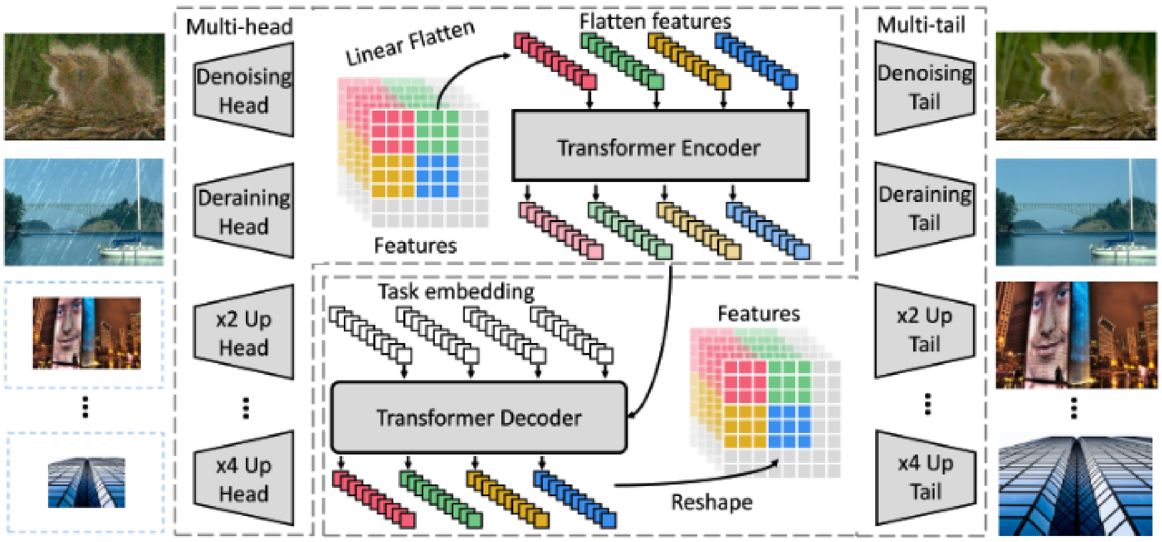


FIGURE 2. Image processing and transformer image

To calculate the transformer value, the following equation is used. $N = \frac{HW}{P^2}$, where N is the value of the transformer, and H is head corresponding to different image processing. P is the patch size and W is the input color (R, G, B). Previous research has also tested the function of the ESP32 camera, which is a function that not only detects faces but can also detect vehicle numbers. So this ESP32CAM can continue to be developed for detection of all forms with IoT-based data transfer mechanisms [3]. Basically, the determination of the colors R, G and B shows a positive combination of red, green and blue colors [4]. However, this cannot be separated from how fast the data transfer is to obtain real-time and accurate face detection data, so that this development continues to be carried out until the face detector can work quickly and data is received quickly, too. The face recognition process is aimed at markers using a facial reconstruction development approach for both 2D and 3D images. With the facial reconstruction, it produces a new marker that will be used for both attendance and other things [5]. Finding ways to automatically detect objects in images without human assistance is an important problem in computer vision. Face detection was introduced to solve this problem. Face detection can be used to detect human faces in object images. Every human face is slightly different, but it can be said that there are some characteristics that all human faces share. Face detection is the first step towards face-related technologies such as facial recognition or face verification. Face detection, on the other hand, has various applications [6]. Figure 3 shows the results of face recognition and face detection on ESP32CAM.

Figure 3. Face detection and recognition

The Arduino development environment is used to program the ESP32CAM board. A program was created to enable the use of video cameras and image capture, and all Wi-Fi functions have been integrated to connect to the Internet via WAN networks and API systems. A face detection algorithm based on the "Model Zoo" model has been added, enabling real-time detection of human faces with the MTCNN (Multi-Task Cascaded Convolutional Networks) pre-trained neural network for its English abbreviation. We create the main code for the program, which integrates all technologies for automatic facial recognition and transmission to the cloud [7]. Figure 4 shows processing data transfer with ESP32CAM to IoT, and the system design will be used in this study as facial recognition. From the source explaining that face recognition through the camera can have an impact

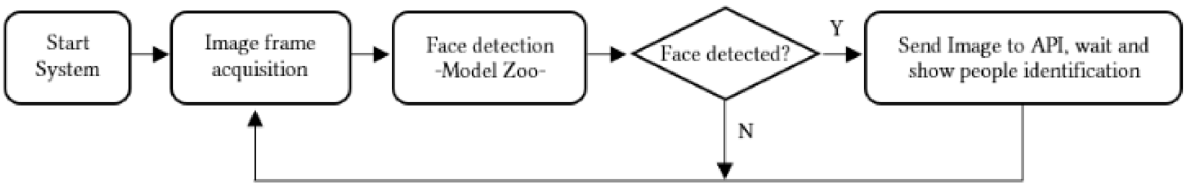


Figure 4. Face detection process and image sending to API for identity identification [7]

on the speed of Wi-Fi and received color image processing, this effect results in perfect face detection with real-time data receivers. Future applications of IoT technology will grow as a result of its ability to make sending data, both visual and non-visual, simple.

The literature study that has been collected makes this research reference to conclude that image processing, face detection and face recognition will work online through an IoT basis. However, color imagery is also related to face capture on the camera screen. After doing research on the literature review, the next step is to determine the components and research methodology.

2. **Material and Methods.** In this section, the face detection method refers to Figure 4. Send the image to the API to identify the face detector. The update in this study adds several methods from Figure 4, namely programming to display the speed of data transfer from the ESP32CAM to a web browser in order to see how much influence the data transfer speed has on the face detection and face recognition process. To support the success of this research, the components used for image processing, data transfer and facial recognition can be shown in Table 1. The components in Table 1 will be designed and connected to the circuit system as shown in Figure 5.

TABLE 1. Name and function of components

Name	Function
Arduino	Components of processing data
ESP32CAM	Image capture face detection and recognition
MicroSD	Save of data

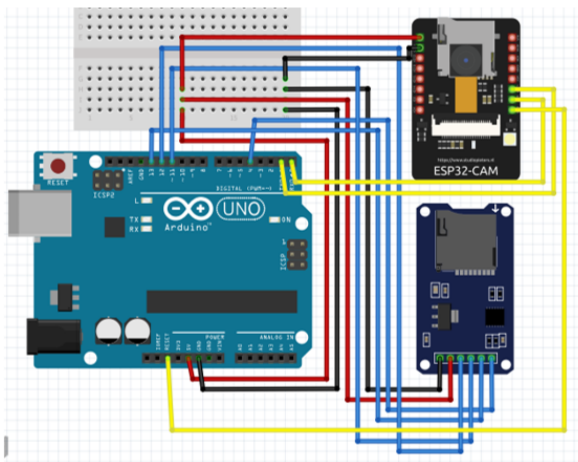


FIGURE 5. Schematic system

The components used to test the use of image processing and face recognition are detailed in Table 1. Figure 5 depicts a component.

The research flow in Figure 6 shows the steps needed to obtain visual data transfer rates and data transfer rates when the camera detects faces and the camera recognizes faces. IoT-based facial recognition can be viewed on a web browser.

The flow chart shown in Figure 6 can explain that this research refers to the literature study from previous research. Next, prepare the components needed in this study, and the components that are already available will be assembled using the system diagram schema shown in Figure 5. After designing the diagram, programming input will be carried out on the Arduino and ESP32CAM. Then the process of image capture and face recognition is done. The data that has been taken will be stored and processed to draw conclusions about

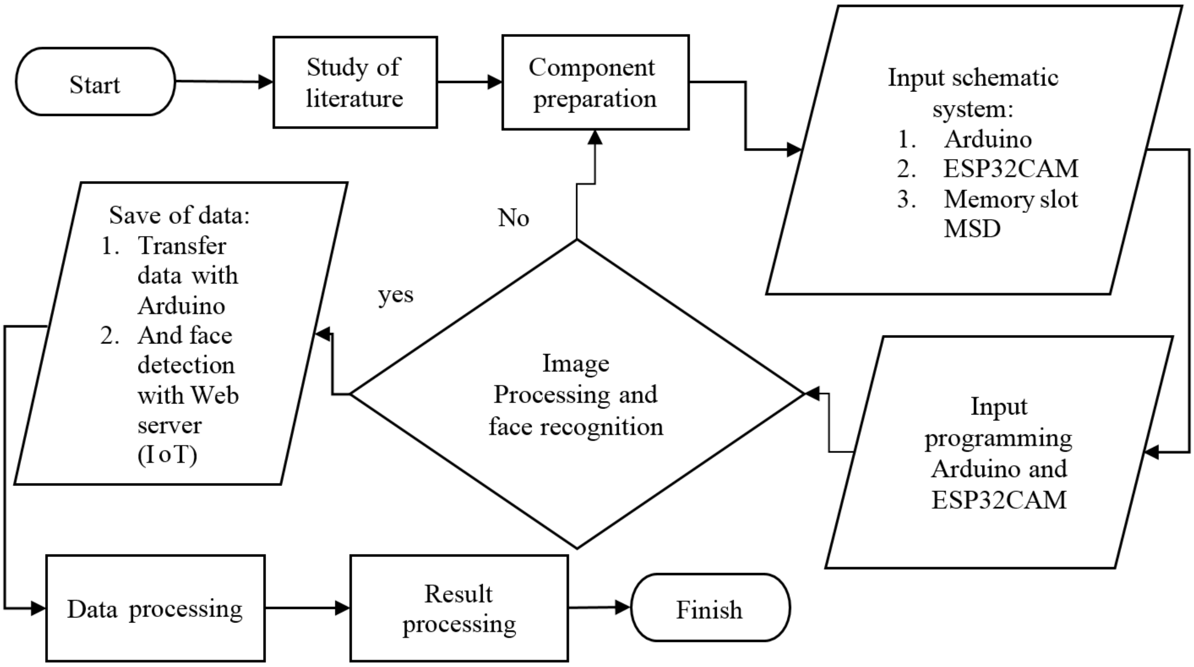


Figure 6. Flow chart processing

the data transfer process for face recognition and image processing. Image processing is done using computer programming [9].

3. **Result and Discussion.** The first step is to examine the required components, which are the Arduino ESP32CAM Micro SD. The ESP32CAM captures images that have been detected by face recognition and face detection. Following that, the data received in the form of transfer speed and face recognition will be stored and processed by Arduino using a Micro SD card. Figure 7 depicts the design that has been created.

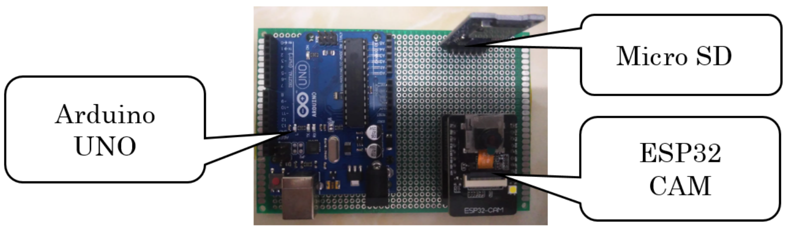
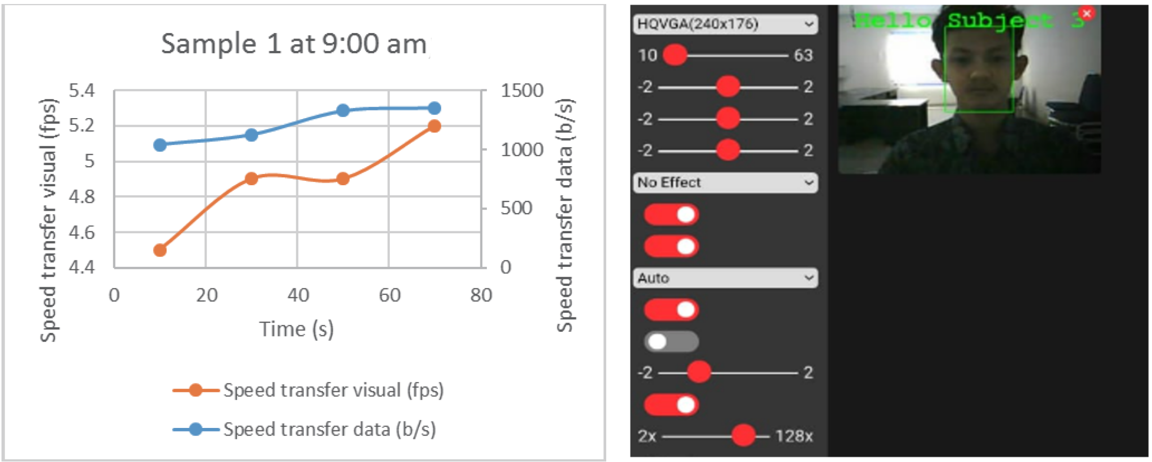


Figure 7. Component of camera recognition

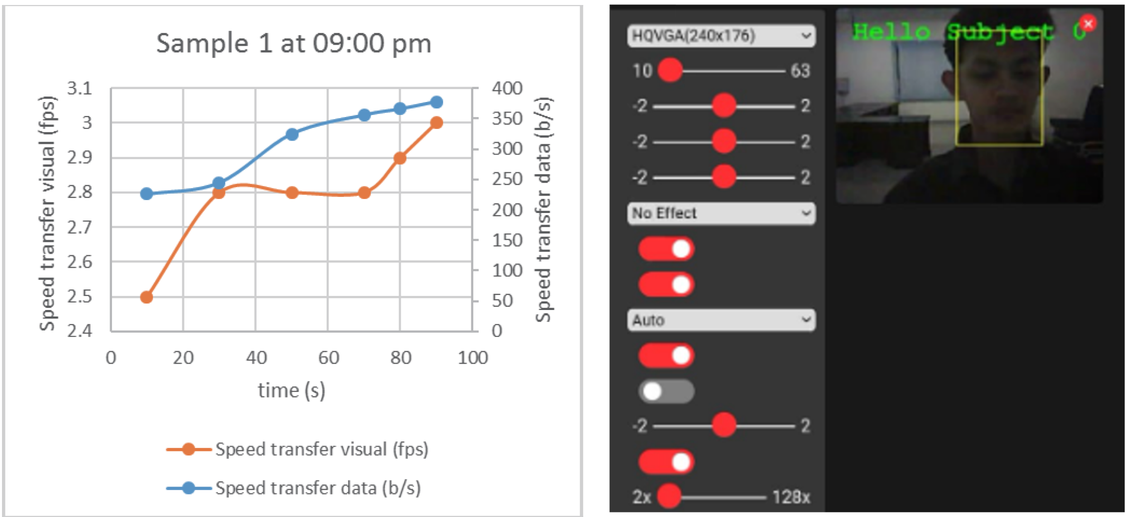
The schematic diagram employs a fritzing application to depict the position and location of each leader, each of which serves a specific purpose. After these components have been assembled, programming input is used to receive facial recognition and face detection data. The programming is done with the Arduino IDE application, which is linked to the ESP32CAM library and port. The test results are shown in Figure 8, which is one of five samples taken with results that are nearly identical.

The recognition process is depicted in Figure 8. From the results of these conclusions, the data obtained by the speed transfer visual and speed transfer data are directly proportional to the increase, resulting in a long time of 70 seconds. Face recognition is successful when it reaches 70 seconds, and facial recognition is then recorded. Figure 8 shows face recognition by marking a square shape in green based on face size, but the fifth image shows the recording recognition process. Due to delayed visual transfer, it appears yellow with blue writing. However, the confession was successfully recorded and displayed



(a) (b)

FIGURE 8. (a) Chart speed transfer; (b) face detection and recognition sample 1 at 09:00 am



(a) (b)

FIGURE 9. (a) Chart speed transfer; (b) face detection and recognition sample 1 at 09:00 pm

graphically. Face recording to detect face shape with sample 5 was successfully performed with a visual transfer range of 4.5 fps to 5.5 fps and data transfer with Vulnerable 500 bytes seconds to 1800 bytes seconds, as shown in Figure 8. After running the test in the morning, run again at night to compare the effects of face recognition and face detection in the morning and evening. Figure 9 shows data using 5 samples at point (a) showing the test graph and image (b) showing the results of face recognition.

The sampling procedure was carried out at 09:00 pm night test. The samples used for the night test included 5 samples of face readings and detected faces. The average visual transfer rate and data transfer rate differed significantly during the nighttime test. According to Figure 9, the average student transfer speed is 3 fps with a data transfer rate of 400 b/s. This is due to the influence of light, which cannot capture the face perfectly. The sampling time to the face recording point can be as long as 90 seconds. The face recording process is very long because the influence of indoor light which cannot recognize faces quickly is inversely proportional to 09:00 am where visual transfer and data transfer have high values because the significant influence of light can help the camera sensor. It can be seen in the graph below for the comparison process with data that has been averaged from samples 1 to 5, so that the efficiency of its use can be seen in Figure 10.

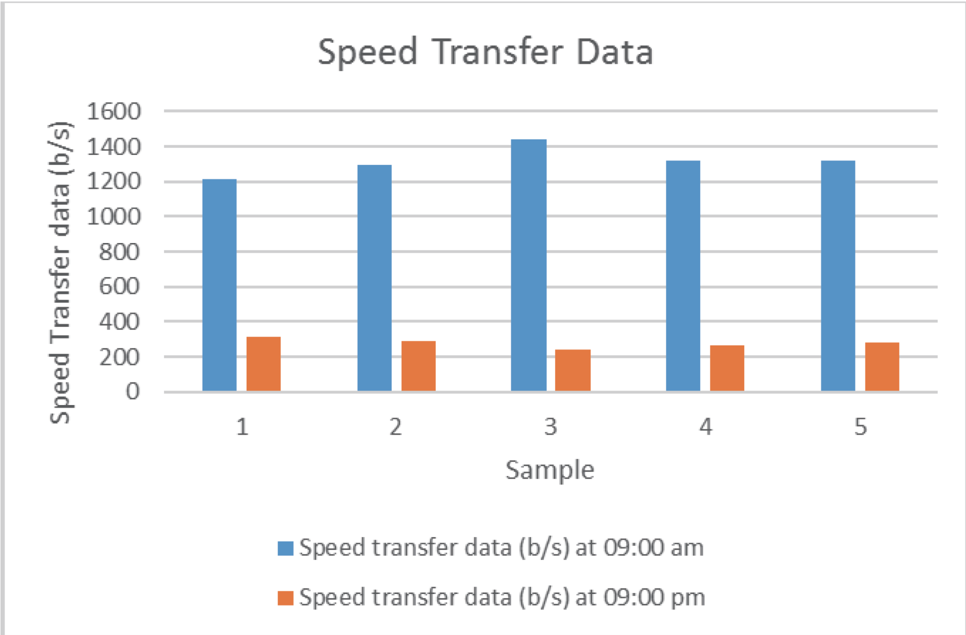


Figure 10. Data transfer rate comparison

Figure 10 explains itself. Describe the significant difference that occurred when testing speed transfer data at 09:00 am and 09:00 pm where at 09:00 am it has a high b/s data transfer influenced by light so that data size transfer is faster so that speed affects face recognition or face detection time is so fast. In contrast to data collection at night, the effect of light blurs the image, requiring a longer recording time for faces. Figure 11 depicts a comparison of speed transfer visual or frames per second in the 5 samples test.

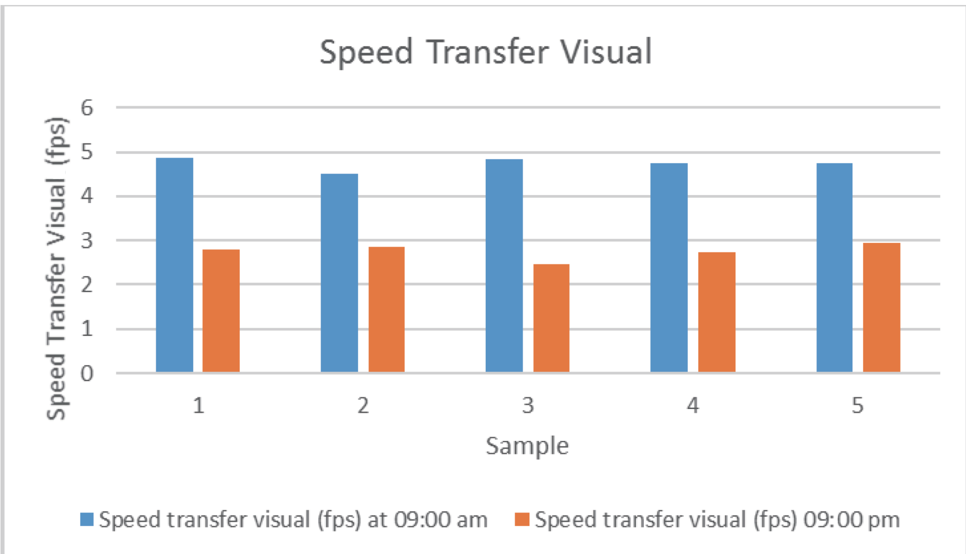


Figure 11. Speed transfer visual rate comparison

The fps level of the received image is better during the morning test than during the night test, as shown in Figure 8 and Figure 9 visual transfer test, where the average fps test yields results ranging from 4.4 to 5 fps inversely proportional to the test, and producing a minimum of 2.2 fps to 3 fps in the morning. As a result of the influence of light, testing in the morning produces more vivid facial appearance than testing in the evening.

4. **Conclusion.** The data transfer and visual transfer tests of the ESP32CAM camera concluded that the face recording test is best done in the morning, but can also be done at night if more light is available. Test results can be obtained as attendance, as facial recognition, for CCTV, and automatically, and there are still many studies that can be continued using this camera. Many studies have been conducted on the concentration of ESP32. However, this paper only explains how to use a good camera and examines the effects of using a camera in the morning and at night. The solution that can be given in this study is if you want to take advantage of data transfer, both image processing, face detection and IoT-based face recognition, network speed, lens contrast and morning and night atmosphere are very influential, so make adjustments to these problems. This research is expected to be used as a reference for further research to examine how to transfer data on facial recognition cameras so that they can receive data in real time without any problems due to Wi-Fi signal speed and lighting contrast using an ESP32CAM camera.

REFERENCES

[1] M. Salvi, U. R. Acharya, F. Molinari and K. M. Meiburger, The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis, *Comput. Biol. Med.*, vol.128, 104129, DOI: 10.1016/j.compbiomed.2020.104129, 2021.

[2] H. Chen et al., Pre-trained image processing transformer, *Proc. of IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp.12294-12305, DOI: 10.1109/CVPR46437.2021.01212, 2021.

[3] I. F. Ashari, M. D. Satria and M. Idris, Parking system optimization based on IoT using face and vehicle plat recognition via Amazon web service and ESP-32 CAM, *Computer Engineering and Applications Journal*, vol.11, no.2, pp.137-153, 2022.

[4] A. R. Putri, Image processing using a web cam on moving vehicles on the highway, *JIPI (Journal of Science Research and Informal Learning)*, vol.1, no.1, pp.1-6, DOI: 10.29100/jipi.v1i01.18, 2016.

[5] E. Winarno, I. H. Al Amin, H. Februariyanti, P. W. Adi, W. Hadikurniawati and M. T. Anwar, Attendance system based on face recognition system using CNN-PCA method and real-time camera, *2019 2nd Int. Semin. Res. Inf. Technol. Intell. Syst. (ISRITI2019)*, pp.301-304, DOI: 10.1109/ISRITI48646.2019.9034596, 2019.

[6] V. Mekala, V. M. Vinod, M. Manimegalai and K. Nandhini, Face recognition based attendance system, *Int. J. Innov. Technol. Explor. Eng.*, vol.8, no.12, pp.520-525, DOI: 10.35940/ijitee.L3406.1081219, 2019.

[7] T. L. E. Dorantes, D. B. Hernandez, A. L. Reyes and C. E. M. Medina, Development of a powerful facial recognition system through an API using ESP32-Cam and Amazon Rekognition service as tools offered by Industry 5.0, *International Conference on Machine Vision and Applications*, pp.76-81, DOI: 10.1145/3523111.3523122, 2022.

[8] R. Hassan, F. Qamar, M. K. Hasan, A. Hafizah, M. Aman and A. S. Ahmed, Internet of Things and its applications: A comprehensive survey rosilah, *Symmetry (Basel)*, vol.12, pp.1-29, 1647, 2020.

[9] D. A. Prabowo and D. Abdullah, Detection and calculation based on color objects using color object tracking, *Pseudocode*, vol.5, no.2, pp.85-91, DOI: 10.33369/pseudocode.5.2.85-91, 2018.