JOIV : Int. J. Inform. Visualization, 7(1) - March 2023 139-145

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

# Design of Livestream Video System and Classification of Rice Disease

Maria Agustin [a], Indra Hermawan [a,*], Defiana Arnaldy [a], Asep Taufik Muharram [a], Bambang Warsuta [a]

[a] *Informatics and Computer Engineering, Jakarta State Polytechnic, 16425, Indonesia*

*Corresponding author: *indra.hermawan@tik.pnj.ac.id*

***Abstract*—One of the agricultural products which is an important aspect of the life of Indonesian people is rice. Rice disease has a devastating effect on rice production, while detecting rice diseases in real-time is still difficult. Therefore, this study designed a Livestream video system that is equipped with a rice disease Classification system. The Livestream system utilizes 4G network communication and is assisted by the WebSocket protocol to communicate in real-time and for the rice disease Classification system using YOLO algorithm. In addition, Livestream uses the raspberry pi camera V2 to take video stream data. In analyzing the performance of the Livestream system, four tests were carried out, namely: functionality test, connectivity test, classification performance test, and implementation performance test. The test was carried out using the *wireshark* and *conky* tools, while the classification training used 5447 images from the Huy Minh do dataset that he provided on the Kaggle website. The results show that all programs run well and get a good QoS value according to the index of the parameter results, it is also found that sending non-base64 can reduce the size of the data to approximately 200,000 bytes/s and the performance of the classification system is good because it has an average accuracy of 80% even though it is quite burdening the raspberry pi. This system can still be optimized and developed further to support research in the field of data transmission and the performance of machine learning in a microcontroller.**

***Keywords*— Livestream; unmanned aerial vehicles; WebSocket protocol; agricultural; rice disease; YOLO.**

*Manuscript received 1 Nov. 2022; revised 17 Nov. 2022; accepted 13 Dec. 2022. Date of publication 31 Mar. 2023.*


## I. INTRODUCTION

Indonesia is a country that has vast agricultural land. In addition, Indonesia is an agricultural country whose agricultural activities and products significantly affect the lives of its people. One of the agricultural products which is an essential aspect of the life of Indonesian people is rice. Rice is a basic need for most Indonesians, and rice is also the largest source of income for farmers. Rice disease has such a devastating impact on rice production that it poses a major threat to food security. Thus, the diagnosis and identification of rice diseases play a significant role in ensuring high yield, high quality, and high efficiency. Classification of rice leaf disease is one of the challenges in developing agriculture, so there is a tendency to decrease productivity. This can be prevented by doing one of the artificial intelligence technologies. For farming to be sustainable, the application of technology is very influential [1].

Over the past few decades, the demand for video services has grown significantly. The increasing role of unmanned aerial vehicles (UAVs) in providing video services is confirmed by the fact that the used inventory has increased by 50% over the last few years [2]. The various uses of this drone are part of smart farming 4.0. Smart farming 4.0 encourages the work of agricultural farmers to be efficient, agrarian and integrated. Farmers can grow crops regardless of the season and can be overcome through mechanization arrangements. The process of planting to harvesting can be done accurately, starting from labor, planting time, and harvesting process [3]. So the author designed a Livestream video system and disease detection for a drone case study using raspberry pi 4 as a control for photos and Livestream video equipped with a rice classification system. In this study, there is an update regarding comparing base64 and non-base64 data transmission on Livestream and a Livestream system equipped with rice disease classification using the YOLO algorithm implemented on the raspberry pi 4 device.

Streaming is a service that directly processes the received data without waiting for all the data to be sent. Current streaming services are audio and video services. Streaming technology, also known as streaming media, is a technology for running files (audio or video) from a streaming server (web page), either live or recorded, where the file must first be encoded using a specific suitable data rate for transmission over a network. Or network according to the user's bandwidth capacity. For this reason, it is necessary to encode audio and video files with varying data rates, which can then be adjusted to the speed of the network and the speed of the data access system. Users can directly view audio and video files from the streaming server by playing them live, and this avoids taking a long time to view huge files. The quality of the streaming file depends on the amount of bandwidth, the content of the file, and the amount of data that can be streamed per second as it traverses the network [4].

So, to build a livestream system that can meet the QoS criteria, the author uses the WebSocket protocol. The WebSocket protocol is said to have the best performance in livestreaming or transmitting streaming data via HTTP (Hypertext Transfer Protocol) [5]–[7].

WebSocket is an HTTP web data exchange method where the WebSocket method uses the typical HTTP request and response method but also requires a request from the client side to open an open connection state (handshake) with the server so that the server and client can communicate and exchange data when new data is available. The delivery process can run in real-time without repeating the entire HTTP protocol [8-10]. WebSocket workflow can be seen in Fig 1.

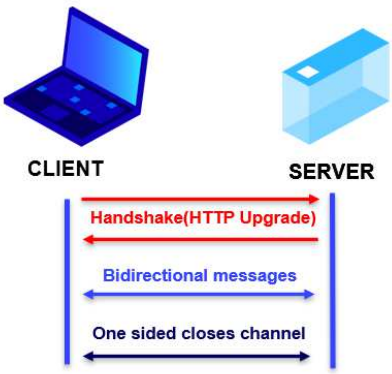


Fig. 1 WebSocket Workflow

Fig 1 shows that WebSocket has a technology that provides its path to support two-way (full-duplex) communication in a single TCP (Transmission Control Protocol) protocol, but WebSocket requires compatibility support from the server side and also the client browser to the WebSocket procedure. The advantages of WebSocket itself are increasing bandwidth efficiency and the disadvantage that it requires a stable TCP to maintain the connection and is not supported by all browsers and web servers [11].

The You Only Look Once (YOLO) algorithm is used to identify rice diseases on the full system. YOLO is an algorithm developed to detect an object in real-time. In one evaluation, a single neural network predicts bounding box and class probabilities directly from the full image. A smaller network version can process 155 frames per second and achieve twice the mAP (mean Average Precision) of other real-time detectors [12]. YOLO uses an Artificial Neural Network (ANN) approach to detect objects in an image. This network divides the image into regions and predicts each bounding box and the probability for each region. These bounding boxes are then compared with each predicted probability. YOLO implements an architecture like the Convolutional Neural Network (CNN). YOLO only uses a convolution layer and a pooling layer. The last convolution layer is adjusted to the number of classes and the number of prediction boxes desired. YOLO has undergone several iterations of its development, from v1 to the latest version being researched, now v5. However, what has been officially published has only reached the fourth version [13]. This research tested each version to classify rice disease, but the main system's implementation used v4-tiny. The reason for choosing the four models is the availability of several sources that can facilitate the application of the model into a microcontroller device, and the reason the author uses a lightweight model is to apply it to a microcontroller device that has limited memory and Central Processing Unit (CPU) capacity [14],[15].

Because of their good data transmission capabilities, many Livestream video stream communication developments are currently more focused on improving performance on the latest network devices such as WiFi and WiMax. Therefore, the author designed a livestream video that utilizes the WebSocket communication protocol and 4G cellular network, which is said to have wide coverage in all cellular areas and at an affordable cost with good data transmission [16].

The use of live video streaming on UAVs has grown rapidly to increase its cost and mobility [17], one of which is in the agricultural sector. The development of Livestream video in the agricultural sector is mainly researched for increasing productivity on getting a good harvest by controlling the needs of the crops on the fields or detecting the crop's disease and curing them by utilizing Livestream video technology that combined with AI technology to detect and classify the disease on the farmer's crop. Some previous studies revealed a comprehensive review of this technology and also used AI method that utilizes a spectrum of color [18]–[21].

This study focuses on designing both a Livestream video data transmission system using the WebSocket protocol on a 4G cellular network and a rice disease classification system using Yolov4-Tiny. This live-streaming video system is built on raspberry pi and sends data to the web, which is built using Flask as a web framework. There is also a similar study where they implement Yolo v3 and v5 in a livestream video for the detection of rice disease in a high-end computer [22],[23]. The novelty of this research is the use and testing Yolo v4-Tiny capability to recognize rice disease and implement it into a stand-alone Livestream system that uses WebSocket and 4G as its network communication in raspberry pi 4 devices.

Furthermore, the authors explain how the software development process, system planning, and testing methods were used in the research methods. In the results and discussion, the authors explain the results of research and development that have been carried out, as well as the results of tests leading to the software's feasibility. In conclusions, the authors provide and state the conclusions based on the testing of the software developed, and the suggestion provides expectations to help improve the device's quality.

## II. Materials and Method

In this section, the authors describe the livestream video design using WebSocket and 4G and the introduction of rice disease on the Raspberry Pi 4. The system design uses the experimental method as the basis for the stages of the research method process. This system is designed and built to transmit, process, and display livestream video data.

### A. System Planning

The overall system of the tool in this study is a livestream video delivery communication system equipped with a rice disease classification system via a 4G cellular network which can be forwarded to the internet so that it can be received by the server and accessed by end users. The working scheme of the overall system is shown in Fig 2. In this study, the system scheme was divided into two parts: raspberry pi as a client and flask web server as a server.

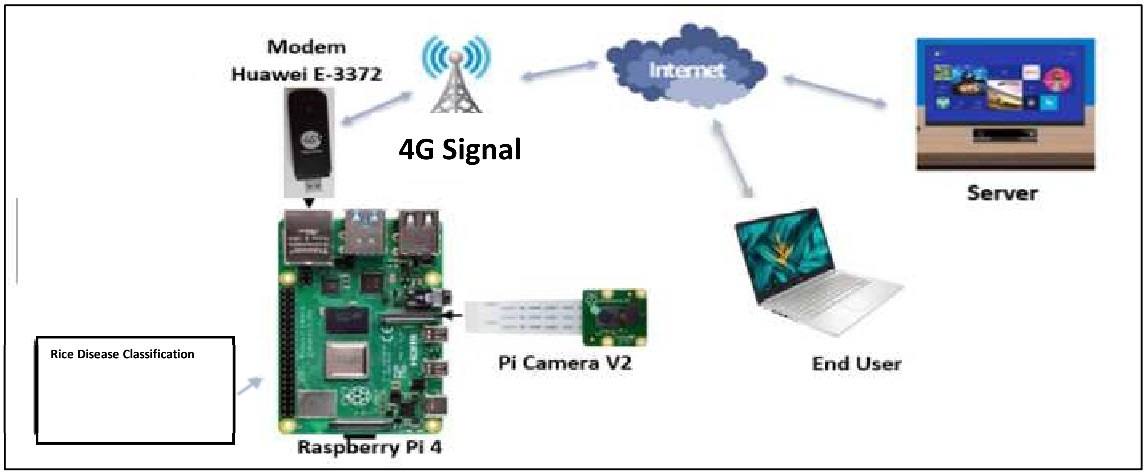


Fig. 2 System Workflow Scheme

### B. System Design

The full system designed in this research was made into three parts. The first is the webserver as a server that is created with the flask framework receiving the stream with the help of WebSocket and showing its result into one of the web browser's HTML pages. The second is raspberry pi, which works as the client that live stream video with or without a classification function. The last is the classification system which functions using Yolov4 Tiny algorithm that embeds its program into the raspberry pi.

*1) Webserver design:* The webserver is built using the flask framework based on the Python programming language. It is equipped with a simple website code and WebSocket server code that receives the results of stream submissions and photos of rice disease classification. It is from the WebSocket client, which makes a new thread process. The results of sending stream video frames can be obtained from the threading process. Alternatively, photos introduced according to the frame taken according to the stream display can be displayed on the website page. The webserver design that is carried out can be seen in Fig 3.

*2) Raspberry Pi Client Design:* Livestream system and rice disease classification on Raspberry pi are made using the OpenCV module in the process of capturing and processing video streams. The classification encompasses encoding the frames obtained by OpenCV into bytes to be sent via a WebSocket client to a WebSocket server in the form of a webserver with the windows ten operating system in performing image viewing. It reassembles the frame and displays a video livestream. The design of the raspberry pi 4 client can be seen in Fig 4.

*3) Rice Disease Classification Design:* In designing which Yolo algorithm to be implemented for the classification system in the main system, the writer tests each Yolo algorithm with rice disease datasets first, and with the result writer decides to use Yolov4 Tiny for the classification implementation on raspberry pi 4.

- Yolo classification design. The model of Yolov4 Tiny is trained using the datasets that the authors have already prepared and to be trained. The training process is a way of processing images and labels so that the patterns or characteristics found in each class are formed to be considered by the computer in reaching a decision or prediction using the Transfer Learning technique. Transfer Learning is a technique that uses a pre-trained model that can be used to classify new datasets and must be retrained so that the model stores the desired dataset. The writer carried out the transfer learning process by using Darknet-53 model using data files, cfg files, and pre-trained weights. In addition, this algorithm was implemented on the Raspberry Pi 4 which was affected by the capabilities of the Raspberry Pi [24].
- Training Dataset. In preparing the datasets to be trained using the yolo algorithm, the collection, and pre-processing of image data was carried out first. The authors enter 5447 image datasets from Huy Minh do on the Kaggle website, which has already been divided by class [25]. Next, object labeling was divided into 4 classes: Brown spots, Healthy, Leaf blast, and Bacterial blast. The objects labeled are only unit rice leaves contained in the dataset.

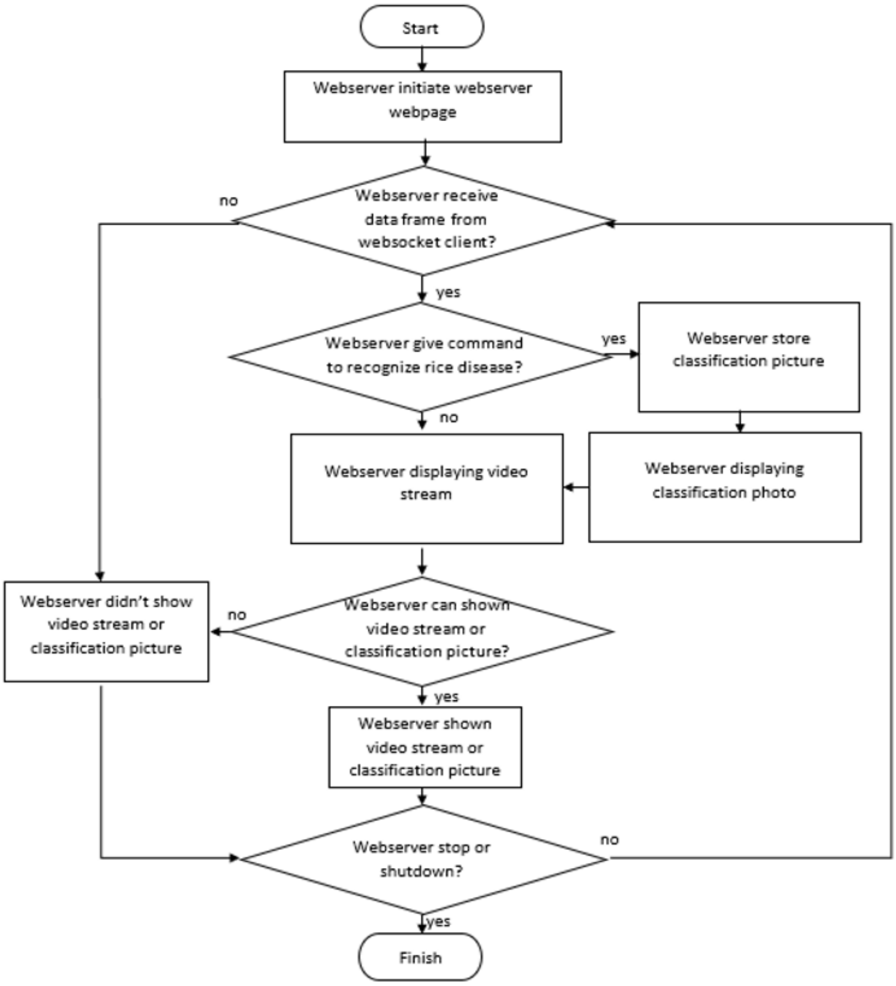


Fig. 3 Webserver Flowchart

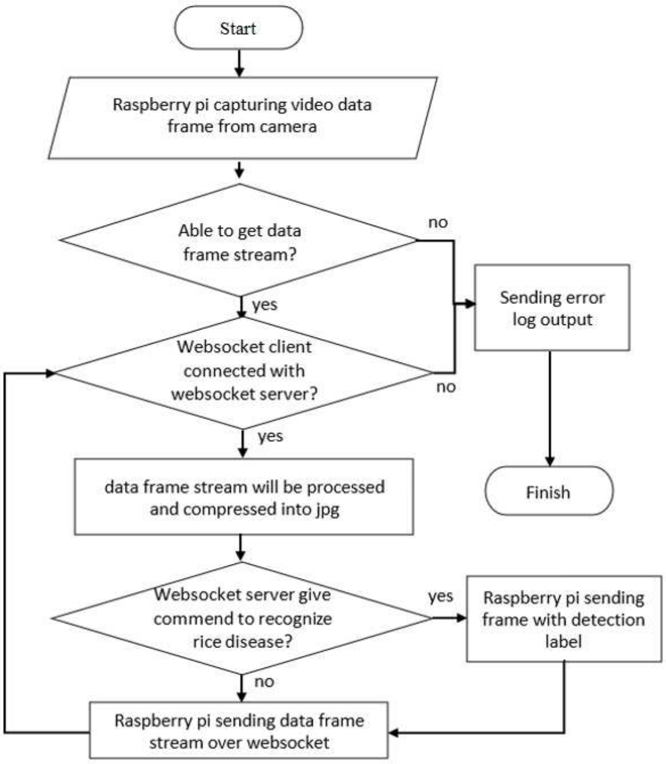


Fig. 4 Raspberry Pi Flowchart

*C. Testing*

The process of testing and analyzing data is carried out with four test methods: functionality testing, connectivity testing, classification performance testing, and implementation performance testing. Tests in this study focus on the performance of rice disease classification by measurement of the accuracy and performance of the CPU running process on the Raspberry Pi device. It is carried out by observing CPU usage using Conky software. Thus, FPS (Frame Per Second) and interferences time obtained network quality in terms of QoS (Quality of Service) on the livestream system in sending data in several circumstances. Therefore, it greatly affects video QoE (Quality of Experience) [26] and the performance in carrying out its functions or functionality tests using a black box [27]. QoS quality testing uses the Wireshark tool which functions to analyze data packets sent during the streaming process. In addition, using conky in analyzing system performance tests.

*1) Functionality Test:* Functionality testing is carried out to see whether the program that the author has made can be used from the user's side by using the black box method.

*2) Connectivity Test:* In developing streaming services, it is very important to meet the index of good QoE criteria. This good QoE criterion is highly dependent on the index of QoS parameters such as throughput, delay, jitter, and packet loss [28], proposed by Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) [29],[30]. Therefore, Connectivity testing was carried out by reading network QoS in the form of delay, throughput, jitter, and packet loss using the Wireshark tool to read packets. Connectivity testing was carried out by sending livestream data locally in the form of base64 and non-base64 data. Both were tested with a video frame data size (resolution) of 640*480 and 1280*720 by sending 2400 data packets using jpeg compression. The formulation of each QoS parameter [31] to see its good in the index can be seen as follows:

The throughput parameter is the effective data transfer rate, which is usually measured in bps. Throughput is the total number of observed packet arrivals at the receiving location during a certain time interval divided by the duration of that time interval. The method for calculating packet loss is formulated in the form of equation 1.

$$throughput = \frac{total\ received\ packets}{total\ delivery\ time} \quad (1)$$

Packet Loss parameter is a parameter used to determine the amount of data that failed to reach the destination. The packet loss value is obtained by counting the packets sent minus the packets that reach their destination. The value of the previous reduction is then divided by the number of packets sent and then multiplied by 100% to be expressed as a percentage. The method for calculating packet loss is formulated in the form of equation 2.

$$packet\ loss = \frac{amount\ send - amount\ received}{total\ packet\ data}\ x100\% \quad (2)$$

The next test parameter used is a delay. Delay is a parameter used to determine the time of sending data to reach the destination. The delay value is obtained by calculating the time the data is successful until the destination is reduced by the time of data transmission. The method for calculating the delay of a packet is formulated in the form of equation 3.

$$Delay = received\ time - time\ send \quad (3)$$

Based on equation 3, when a lot of data is tested, it can be seen that the average delay value of sending a lot of data is by adding up all the delays that occur and then divided by the amount of data tested. To simplify the equation in calculating the average delay value, it can be seen in equation 4.

$$Average\ delay = \frac{received\ time - time\ send}{total\ packet\ data} \quad (4)$$

The next parameter used to determine network performance can also be seen from the jitter parameter generated from data transmission. Jitter is the variation in the package arrival time at the delivery destination address. To calculate the jitter, a delay variation value is needed, formulated by reducing the delay from sending data with the previous delay. After the delay variation value is obtained, the jitter can be determined by dividing the delay variation value by the number of packets received minus one. For convenience, the formula for calculating jitter can be seen in equation 5.

$$jitter = \frac{Total\ delay\ variation}{Total\ received\ packet\ -1} \quad (5)$$

*3) Performance Test:* This test was conducted with two performance testing. The first test tests the rice leaf disease classification ability on each Yolo algorithm with CPU processing to test classifying each healthy rice leaf. Three types of rice diseases, namely brown spot, leaf blast, and bacterial blight, are divided into 200 to 300 images with a total of 1300 datasets. It aims to see which version has the best Yolo algorithm performance on classifying rice disease by calculating the performance parameter of each version of Yolo deep learning. The authors use the performance parameters of loss, precision, recall, f1-score, and mAP. While the second test was carried out the monitoring CPU usage on the raspberry pi using Conky and testing the accuracy of implemented rice Classification using the Yolov4 Tiny algorithm. It aims to see its performance while the program is implemented on raspberry pi 4 by observing CPU usage, FPS, and interferences time. It is subject to the classification system's time and resource usage.

To see the performance of the classification, some performance test was conducted, some parameters in this research have been used for checking classification performance on machine learning in general, namely: accuracy, precision, recall, F-1 score, and mAP (mean Average Precision) where there also some research that is researching to found the performance of Yolo v3 algorithm to classify some items [32], [33].

Accuracy Is the ratio of correct predictions (success and failure) of the overall data obtained, and a formula can be made, namely the number of correct classifications divided by the total number of classifications that should be as shown in equation 6.

$$Accuracy = TP + TNTP + TB + FP + FN \quad (6)$$

Precision is the predicted ratio of the correct identifier to the number of correctly predicted identifiers for a specific class. The precision value is calculated by dividing the total positive samples that are classified correctly by the total predicted positive samples as in the equation below; high precision shows that the positive labeled samples are indeed positive (a little FP) can be calculated with equation 7.

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

Recall is the ratio of the prediction of correct identification to the number of identifications that are correct. A high recall indicates a small number of samples that were identified incorrectly. Recall can be represented as equation 8.

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

The F1-score quantifies the harmonic average between precision and accuracy. This metric usually represents the robustness of the classification task and can be calculated using the equation 9.

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Recall+Precision} \quad (9)$$

The mean average precision is the average value of the average precision (AP), which forms an evaluation metric that measures the performance of an object. The AP value is obtained from precision and recall calculations, which are calculated in the equation 10 below.

$$AP = \sum(\text{recalln}+1-\text{recalln}) \times \text{precisioninterp} \times (\text{recalln}+1) \quad (10)$$

## III. Results and Discussion

From this research, the authors have finished implementing a webserver and client with a WebSocket network communication system and tested the livestream video transmission process and rice disease classification.

### A. Functionality Test

In testing the data obtained that the system has been running according to the expectation, the test uses black box testing method to provide system functionality. The results of the functionality test can be seen in Table 1.

TABLE I
FUNCTIONALITY TEST RESULT

| No | Test | Result |
|---|---|---|
| 1 | Capture camera frame data | Success |
| 2 | Sending stream frames via WebSocket | Success |
| 3 | Displaying data frames on web pages | Success |
| 4 | Taking photos and doing rice disease Classification | Success |
| 5 | Store data on database | Success |

### B. Connectivity Test

Connectivity testing is done by calculating the network QoS parameters that support QoE video, namely: throughput (bps), delay (ms), jitter (ms), and packet loss (%) under the same conditions by sending 2400 packets of video stream data with jpeg compression with measuring QoS parameters with different resolution in different data states of base64 and non-base64. The results of the connectivity test can be seen in Table II.

TABLE II
CONNECTIVITY TEST RESULT

| Data Type | Resolution | Throughput (byte/s) | Packet loss (%) | Rata-rata delay (ms) | Rata-rata jitter (ms) |
|---|---|---|---|---|---|
| Base64 | 640* 480 | 1.110.677 | 0% | 1,145 | $5{,}25\text{x}10^{-5}$ |
| Base64 | 1280*720 | 1.208.242 | 0% | 1,476 | $6{,}40\text{x}10^{-3}$ |
| Non- Base64 | 640* 480 | 927.029 | 0% | 1,118 | $5{,}04\text{x}10^{-5}$ |
| Non- Base64 | 1280*720 | 1.001.311 | 0% | 1,338 | $8{,}79\text{x}10^{-5}$ |

Based on the result of the connectivity test shown in Table 2, sending base64 data had a bigger size than sending non-base64 data. The result can be seen that sending video frames with resolution 640*480 or 1280*720 with base64 format had a higher delay caused by the higher size that can be seen in the throughput that can represent the size of the video frame data.

### C. Classification Performance Test

The classification performance test calculated some of the general performance parameters of machine learning: loss, mAP, f1-score, recall, and precision. Datasets tested each Yolo algorithm with a total of 1300 pictures, each containing a healthy rice leaf and 3 types of rice disease, namely brown spot, leaf blast, and bacterial blight, divided into 200 to 300 images. Table 3 shows the Yolo performance test result. It shows that Yolov4 Tiny has the best performance on classifying rice disease with the most less loss, which is 40%, compared to other Yolo models.

TABLE III
YOLO PERFORMANCE TEST RESULT

| No | Model | Loss | mAP | F1 Score | Recall | Precision |
|---|---|---|---|---|---|---|
| 1 | Yolov3 Tiny | 79% | 65% | 55% | 62% | 49% |
| 2 | Yolov4 Tiny | 40% | 80% | 69% | 70% | 68% |
| 3 | Yolov5n | 57% | 69% | 68% | 69% | 68% |
| 4 | Yolov5s | 47% | 70% | 68% | 70% | 68% |

### D. Implementation Classification Performance Test

Implementation classification performance testing is carried out by looking at the CPU usage performed by the raspberry pi 4 device in running the entire livestream video and rice disease classification system with Yolov4 Tiny. Moreover, observing each classification's average performance and accuracy in five tests. The performance results can be seen in Table 4.

TABLE IV
RICE DISEASE CLASSIFICATION PERFORMANCE TEST RESULT

| Class | Accuracy | Average Processing Time (second) | FPS (second) | Inference Time (second) | CPU Usage |
|---|---|---|---|---|---|
| Healthy | 80% | 5 second | 0.3 - 5.06 | 0.63 - 2.44 | 81% - 90% |
| Brownspot | 60% | 5 second | | | |
| Leafblast | 100% | 5 second | | | |
| Bacterial-Blight | 80% | 5 second | | | |

The result of rice disease classification on raspberry in Table IV shows the accuracy in recognizing each rice disease and the healthy rice with Yolov4 Tiny algorithm. It can work properly to detect each class that has been trained. However, each classification process with Yolov4 Tiny still takes much time and resources in raspberry pi 4 as seen from the FPS and CPU usage of the implemented classification system.

## IV. CONCLUSION

This research found that the livestream video and rice classification system can run well. It works according to the authors expectations, and as a surveillance medium. The QoS value obtained from the connectivity test has stated, according to the TIPHON index, that the quality of the video livestream is running well and meets the good quality of the QoS index, so the results of the video QoE are also good. Using non-base64 data in WebSocket delivery has reduced yield but can reduce the data size by up to 200,000 bytes/s, reducing the burden on livestream video delivery. This research offers research on other's deep learning algorithm performance on the limited device such as raspberry pi. This research suggests better compression or codec and data transmission methods to increase Livestream's real-time capability further. Ultimately, this research also suggests using hybrid post-processing on the Yolo algorithm to increase processing capabilities [34].

## ACKNOWLEDGMENT

Jakarta State Polytechnic supports this research through *Penelitian Produk Vokasi Unggulan Perguruan Tinggi* research program 2022 with grant number B.378/PL3.18/PT.00.06/2022.

## REFERENCES

[1] "Analisis Biofisik Tanaman Padi dengan Citra Drone (UAV) Menggunakan Software Agisoft Photoscan," pp. 2–3, 2017.

[2] V. Barannik, S. Podlesny, D. Tarasenko, D. Barannik, and O. Kulitsa, "The video stream encoding method in infocommunication systems," in *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)*, 2018, pp. 538–541. doi: 10.1109/TCSET.2018.8336259.

[3] R. R. Rachmawati, "Smart Farming 4.0 Untuk Mewujudkan Pertanian Indonesia Maju, Mandiri, Dan Modern," *Forum Penelit. Agro Ekon.*, vol. 38, no. 2, p. 137, 2021, doi: 10.21082/fae.v38n2.2020.137-154.

[4] A. Wongkitrungrueng and N. Assarut, "The role of live streaming in building consumer trust and engagement with social commerce sellers," *J. Bus. Res.*, vol. 117, no. November 2017, pp. 543–556, 2020, doi: 10.1016/j.jbusres.2018.08.032.

[5] G. M. B. Oliveira *et al.*, "Comparison between MQTT and WebSocket Protocols for IoT Applications Using ESP8266," *2018 Work. Metrol. Ind. 4.0 IoT, MetroInd 4.0 IoT 2018 - Proc.*, pp. 236–241, 2018, doi: 10.1109/METROI4.2018.8428348.

[6] Z. A. Achmad, "Penerapan Websocket Untuk Transmisi Data Pada Iot (Internet of Things) Guna Mendukung Era Industri 4.0," 2019.

[7] M. Lutfi, P. H. Trisnawan, and R. Primananda, "Implementasi Routing Statis menggunakan Media Komunikasi LoRa dan Websocket untuk Pengiriman Data dari Sensor ke Cloud pada IoT," vol. 5, no. 12, pp. 5339–5348, 2021.

[8] B. Soewito, Christian, F. E. Gunawan, Diana, and I. Gede Putra Kusuma, "Websocket to support real time smart home applications," *Procedia Comput. Sci.*, vol. 157, pp. 560–566, 2019, doi: 10.1016/j.procs.2019.09.014.

[9] F. P. Uditama, R. Primananda, and M. Data, "Perancangan Aplikasi Pemantauan Pendaki Gunung Menggunakan Wireless Network Dengan Protokol MQTT," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya e-ISSN 2548-964X*, vol. 2 No 5, no. 5, pp. 2102–2108, 2018.

[10] Juslenius, S. WebSocket vs WebRTC in the stream overlays of the Streamr Network. University of Helsinki. 2021.

[11] K. E. Ogundeyi and C. Yinka-Banjo, "WebSocket in real time application," *Niger. J. Technol.*, vol. 38, no. 4, p. 1010, 2019, doi: 10.4314/njt.v38i4.26.

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.

[13] V. Gupta Pola, A. Bhavya Vaishnavi, and S. Suraj Karra, "Comparison of YOLOv3, YOLOv4 and YOLOv5 Performance for Detection of Blood Cells," *Int. Res. J. Eng. Technol.*, pp. 4225–4229, 2021.

[14] G. Plastiras, C. Kyrkou, and T. Theocharides, "You Only Look Once: Unified, Real-Time Object Detection," *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3243394.3243692.

[15] Jiang, P. et al. "A Review of Yolo Algorithm Developments," Procedia Computer Science, 199, pp. 1066–1073. 2021 doi: 10.1016/j.procs.2022.01.135.

[16] P. P. Ray, "Internet of things for smart agriculture: Technologies, practices and future direction," *J. Ambient Intell. Smart Environ.*, vol. 9, no. 4, pp. 395–420, 2017, doi: 10.3233/AIS-170440.

[17] Hermawan, I., Arnaldy, D., Agustin, M., Widyono, M. F., Nathanael, D., & Mulyani, M. T. "Low-cost surveillance system using smartphone and raspberry Pi 4 based on real time streaming Protocol". In 2022 5th International Conference on Computer and Informatics Engineering (IC2IE). IEEE. 2022.

[18] A. D. Boursianis *et al.*, "Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review," *Internet of Things (Netherlands)*, vol. 18, p. 100187, 2022, doi: 10.1016/j.iot.2020.100187.

[19] S. K. Bhoi *et al.*, "An Internet of Things assisted Unmanned Aerial Vehicle based artificial intelligence model for rice pest detection," *Microprocess. Microsyst.*, vol. 80, no. December 2020, p. 103607, 2021, doi: 10.1016/j.micpro.2020.103607.

[20] D. Zhang, X. Zhou, J. Zhang, Y. Lan, C. Xu, and D. Liang, "Detection of rice sheath blight using an unmanned aerial system with high-resolution color and multispectral imaging," *PLoS One*, vol. 13, no. 5, pp. 1–14, 2018, doi: 10.1371/journal.pone.0187470.

[21] Sari, M. Y. A. et al. "Monitoring Rice Crop and Paddy Field Condition Using UAV RGB Imagery," International Journal on Informatics Visualization, 5(4), pp. 469–474, 2021. doi: 10.30630/JOIV.5.4.742.

[22] Haque, M. E. et al. "Rice Leaf Disease Classification and Detection Using YOLOv5," arXiv 2209.01579, pp. 1–16, 2022.

[23] Li, D. et al. "A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network," Sensors (Switzerland), 20(3), 2020. doi: 10.3390/s20030578.

[24] Hermawan, I., Arnaldy, D., Agustin, M., Widyono, M. F., Nathanael, D., & Mulyani, M. T."Sistem Pengenalan Benih Padi Menggunakan Metode Light Convolutional Neural Network Pada Raspberry Pi 4 B". Jurnal Teknologi Terpadu. Jurnal Teknologi Terpadu Vol, 7(2), 120-126, 2021.

[25] "Rice Diseases Image Dataset | Kaggle." https://www.kaggle.com/datasets/minhhuy2810/rice-diseases-image-dataset (accessed Oct. 31, 2022).

[26] N. Rao *et al.*, "Analysis of the effect of QoS on video conferencing QoE," *2019 15th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2019*, pp. 1267–1272, 2019, doi: 10.1109/IWCMC.2019.8766591.

[27] A. Eko, S. Putro, H. Tolle, and A. P. Kharisma, "Rancang Bangun Aplikasi Penawaran dan Pencarian Kerja Paruh Waktu (Part Time) Berbasis Lokasi," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 8, pp. 2548–964, 2018.

[28] G. Ardiansa and R. Primananda, "Manajemen Bandwidth dan Manajemen Pengguna pada Jaringan Wireless Mesh Network dengan Mikrotik," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 11, p. 47, 2017.

[29] Subektiningsih, Renaldi dan Ferdiansyah, P. (2022) "Analisis Perbandingan Parameter QoS Standar TIPHON Pada Jaringan Nirkabel Dalam Penerapan Metode PCQ," Explore, 12(1), hal. 57–63.

[30] Aprianto Budiman, M. Ficky Duskarnaen, and Hamidillah Ajie, "Analisis Quality of Service (Qos) Pada Jaringan Internet Smk Negeri 7 Jakarta," *PINTER J. Pendidik. Tek. Inform. dan Komput.*, vol. 4, no. 2, pp. 32–36, 2020, doi: 10.21009/pinter.4.2.6.

[31] R. D. Christian, "Sistem monitoring tingkat pencemaran air sungai menggunakan protokol websocket dan modul komunikasi lora," 2021.

[32] N. Eka Budiyanta, M. Mulyadi, and H. Tanudjaja, "Sistem Deteksi Kemurnian Beras berbasis Computer Vision dengan Pendekatan Algoritma YOLO," *J. Inform. J. Pengemb. IT*, vol. 6, no. 1, pp. 51–55, 2021.

[33] E. Tirtana, K. Gunadi, and I. Sugiarto, "Penerapan Metode YOLO dan Tesseract-OCR untuk Pendataan Plat Nomor Kendaraan Bermotor Umum di Indonesia Menggunakan Raspberry Pi," *J. Infra*, vol. 9, no. 2, p. 7, 2021.

[34] Fajrianti, E. D. et al. "High-Performance Computing on Agriculture: Analysis of Corn Leaf Disease," International Journal on Informatics Visualization, 6(2), pp. 411–417, 2021. doi: 10.30630/joiv.6.2.793.