International Journal of Innovative Computing, Information and Control
Volume **19**, Number **5**, October **2023**

ICIC International ©2023 ISSN 1349-4198
pp. 1377–1391

COVID-19 AEROSOL SUCTION ROBOT TO ASSIST DENTIST SURGERY BASED ON MOUTH OPENNESS DETECTION USING DEEP LEARNING

Riyanto Sigit[1], Aditia Yuliyanto[1], Moch. Rochmad[2]
and Imam Safari Azhar[3]

[1]Department of Information and Computer Engineering
[2]Department of Electro Engineering
Electronic Engineering Polytechnic Institute of Surabaya
Jl. Raya ITS, Keputih, Sukolilo, Surabaya 60111, Indonesia
riyanto@eepis-its.edu; aditiayl@ce.student.pens.ac.id; rochmad@pens.ac.id

[3]Department of Prosthodontics
Faculty of Dental Medicine
Universitas Airlangga
Jl. Mayjen Prof. Dr. Moestopo No. 47, Tambaksari, Surabaya 60132, Indonesia
imam-safari-azhar@fkg.unair.ac.id

Received December 2022; revised April 2023

Abstract. *As of November 6, 2022, COVID-19 cases have increased in the Southeast Asia Region by approximately +28%. In dentistry, the COVID-19 infection can be transmitted quickly through aerosol particles. Extra-Oral Suction (EOS) is a mechanism for sucking an aerosol of the patient to counter the transmission of COVID-19; nevertheless the EOS suction nozzle is still manually driven. This allows aerosol particles to escape the mechanism when the patient turns his head or shifts the position of the head. In this case, Visual Servoing (VS) is required; In the realm of image processing, specifically in the context of mouth openness detection, we explore the integration of autonomous control mechanisms to dynamically adjust the orientation of a nozzle. We used a PBVS (Position-Based Visual Servoing) approach integrated with the TensorFlow deep learning models, namely EfficientDet D0, Single Shot Detector (SSD) MobileNet, and SSD ResNet50. Based on the test results of mouth openness detection, EfficientDet has the highest accuracy reaching 100%, SSD MobileNet has an accuracy of 97%, and SSD ResNet50 has an accuracy of 98%. In addition, the system can track the position of the human mouth with a response speed of 135 px/s and an accuracy of 73.1%.*

Keywords: COVID-19, Mouth tracking, Visual servoing, Deep learning, Dentist surgery

1. **Introduction.** According to World Health Organization (WHO) data, confirmed COVID-19 cases as of November 6, 2022 reached more than 629 million; more than 6.5 million deaths recorded globally. Recently, COVID-19 cases have risen in the region of Southeast Asia (+28%) moderately and in the Western Pacific region (+10%) steadily, caused by a rise in mortality rates in the Southeast Asia region (+535%). Meanwhile, in Indonesia in November 2022, there was an increase of 56%: there were 30,670 new confirmed cases of COVID-19 and a 38% rise in death cases, of 232 cases compared to the previous week's cases [1].

The COVID-19 virus is expanding quickly in several countries. Furthermore, reported by WHO, the transmission of SARS-CoV-2 can take place by contact, both indirect and direct, or close contact with a contaminated patient via secretions such as saliva and

DOI: 10.24507/ijicic.19.05.1377

respiratory secretions or breathing droplets detached when a contaminated individual coughs, sneezes, and talks. The diameter size of the breathy droplet is about > 5-10 μm. On the other hand, the ones that have a diameter size of < 5 μm, namely the droplet nucleus or aerosol [2]. In addition, aerosols can be kept in suspension in the air for up to 1 hour [3]. Next, dentists are at a higher risk of COVID-19 contact because their work area is around patients' mouths, coming into contact with saliva, rising the chance of aerosol spread of COVID-19 during treatment [4].

In response to the above problem, recommended safety protocols have been suggested to counter COVID-19 transmission in dental clinics by utilizing Extra-Oral Suction (EOS). The aerosol viruses that entered into EOS will be eliminated by filtering using the prefilter, subsequently filtrating through the medium filter, and finally refining using the High-Efficiency Particulate Air (HEPA) filter. In other words, HEPA filtration is the only solution to eliminate the spreading of COVID-19 completely [5]. The HEPA filter is capable of trapping virus-sized airborne particulates with > 99.9% accuracy [6]. The particles that have been filtrated are then illuminated by ultraviolet light, enabling such particles to assassinate the virus effectively [7]. Nevertheless, the suction nozzle device is manually moved and positioned, so that the position and distance of the Extra-Oral Suction (EOS) nozzle are directly above the head and pointed towards the patient's oral cavity. This may cause some particles of aerosol to be sucked up by the device when the patient is turning their head or shifting their head position.

The contribution of this article is to build a visual servoing robot on an Extra-Oral Suction (EOS) which includes constructing a mechanical EOS robot, designing an electrical robot, developing a mobile app, building a system control, collecting mouth openness data set and creating a mouth openness pose detection system using deep learning. In the previous article, Peng et al. [10] compared Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS) for industrial robotic assembly. Nishad [14] developed a Keras model for driver drowsiness detection. Zhang et al. [25] researched expression recognition methods on MobileNet V2-SSD-based robots that can recognize facial expressions in real time and accurately.

This article used the Position-Based Visual Servoing (PBVS) method which is a method to navigate robots by utilizing visual data information. This means the processing of input images, control theory, and robotics are integrated to steer the robot's movements according to the visual information which is captured by the camera [8]. In addition, we combined the TensorFlow model to detect mouth openness position in input images and the PBVS control method to compute the movement of the servo. Therefore, EOS robotic system was created combined with visual servoing technology. In other words, EOS can automatically track and shift the suction nozzle position following the coordinate of the patient's oral cavity.

The article is organized as follows. Section 1 discusses the introduction and Section 2 explores the proposed method and materials. The results and discussion are provided in Section 3. Finally, Section 4 summarizes this research.

2. **Material and Method.**

Figure 1 represents the overall flowchart of the previously mentioned system. First, the user will receive command input through the mobile application by the app we developed. The app includes some control buttons, for example, up, down, left, and right. Then, the command is sent to the microcontroller via Bluetooth communication. Furthermore, the command will be computed and sent to the servo, enabling the robotic arm to move or turn on/off the suction machine. Consequently, when the suction machine is on, the webcam will automatically turn on, so that image input will occur. Next, the mini PC will process the image input from the webcam; therefore, it

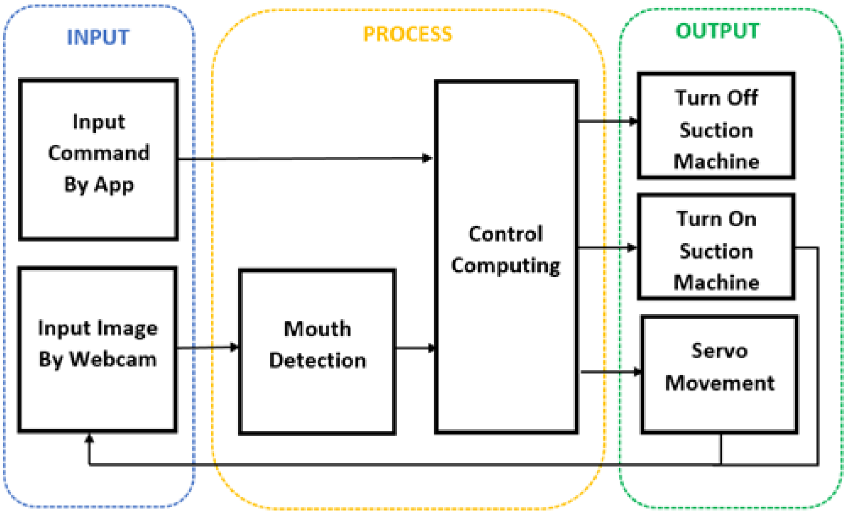


Figure 1. Flow diagram of the system

will detect the position of the mouth and it will send the result position to the microcontroller. Subsequently, the microcontroller will compute the movement of the servo based on the result of the position and will be sent to the servo. Hence, the robotic arm can move according to the patient's oral detection results and will continue to recur until the sucking machine is turned off.

2.1. **Mechanical and hardware design.** Image-Based Visual Servoing (IBVS) has greater robustness due to its camera calibration accuracy. Nevertheless, Position-Based Visual Servoing (PBVS) is steadily more accurate and faster than IBVS. This outcome is driven by the fact that the target of visual servoing in the image plane is derived from the target of a reference, which is dependent on the camera's model accuracy [10]. PBVS used a configuration of cameras on the end effector, as illustrated in Figure 2.

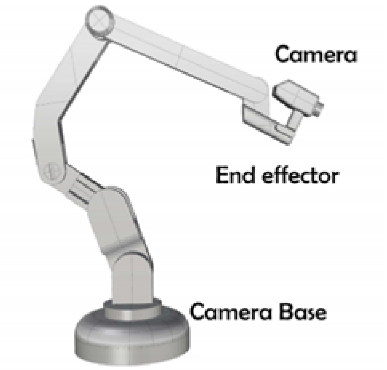


Figure 2. Configuration camera of PBVS

PBVS used the data of visual to construct the robot's 3D pose and kinematic errors that resulted in Cartesian space and actuator command mapping. In this case, calibration of the 3D camera is needed to map the feature data of the 2D image to the Cartesian space data [11]. PBVS predicted the relative object to the coordinates of the camera and the position of the system. The error of the image has to transmit to induce the rotation and movement of the camera [12].

Figure 3 shows an EOS robot display with its five important parts: the suction machine inside the robot body, the camera installed on the robot's end effector, two servos seated on the end effector and the base of the robotic arm, and the Embedded PC and microcontroller on top of the robot body.

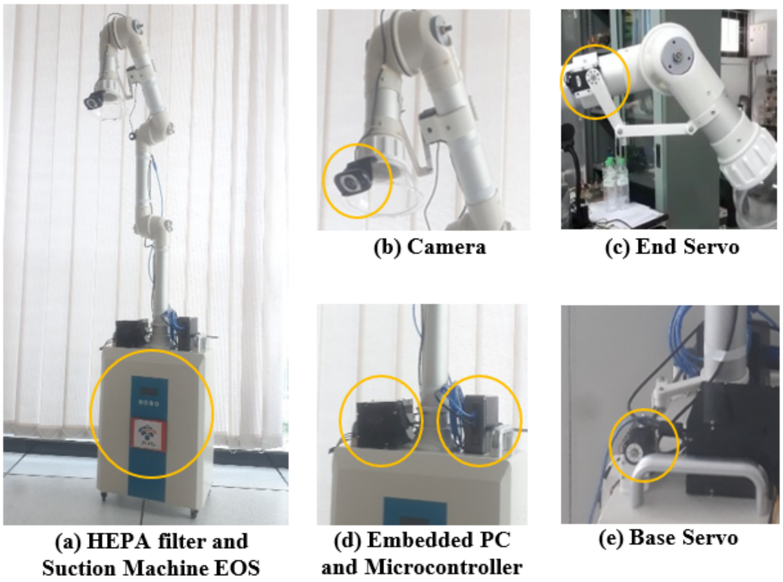

(a) HEPA filter and Suction Machine EOS
(b) Camera
(c) End Servo
(d) Embedded PC and Microcontroller
(e) Base Servo

Figure 3. Robot mechanics of PBVS

We have chosen Arduino UNO as a microcontroller because its easy-to-implement nature makes the system cost-effective and adaptable for small and medium-sized industries [9]. We used several hardware components such as Dynamixel Shield, Dynamixel Servo MX-28 as a robot arm drive, Bluetooth HC-05 as communication between Arduino UNO and Servo App, embedded PC as image processing, webcam, power supply, DC motor, and 4 Channel Relays. Arduino UNO is combined with the Dynamixel Shield so that Arduino UNO can control the dynamixel servo directly using a TTL cable. Dynamixel Shield gets voltage from a 12 V power supply connected to 220 V AC power. In addition, Dynamixel Servo regulates the input voltage entering the Dynamixel Servo 5-16 V and Arduino UNO, namely 5 V. Dynamixel Shield is connected to Intell NUC via a USB FTDI cable to communicate data from image processing obtained from Intell NUC with image input from the webcam. The Dynamixel Shield is also connected to two Dynamixel Servos, namely the Dynamixel Servo on the end effector and the Dynamixel Servo on the base arm of the robot via a TTL cable. Both Dynamixel Servos require an average of 9 V voltage and 100 mA current to run optimally. The following is a table of the Dynamixel Servo TTL PIN and Dynamixel Shield TTL PIN pair. Dynamixel Shield is also connected to the relay as a switch to disconnect or connect the HC-05 Bluetooth RX and TX PINs to set FTDI USB and Bluetooth serial communication to work alternately. In addition, the relay acts as a switch to disconnect or connect the current and voltage to the DC motor on the EOS robot. The following are tables of the Relay PIN, Dynamixel Shield PIN, HC-05 Bluetooth PIN, and DC motor PIN on the EOS robot. The hardware circuit of the robot can be viewed in Figure 4.

Dynamixel is a powerful controller package; this can bring up a manager that requires parameters such as namespace, motor port, motor ID range, baud rate, and update rate [13]. Therefore, we have chosen Dynamixel Servo to move the robot arm, which can be adjusted to rotate in continuous rotation mode or positional rotation mode. Two Dynamixel Servos are used on the EOS robot arm: the Dynamixel Servo at the end

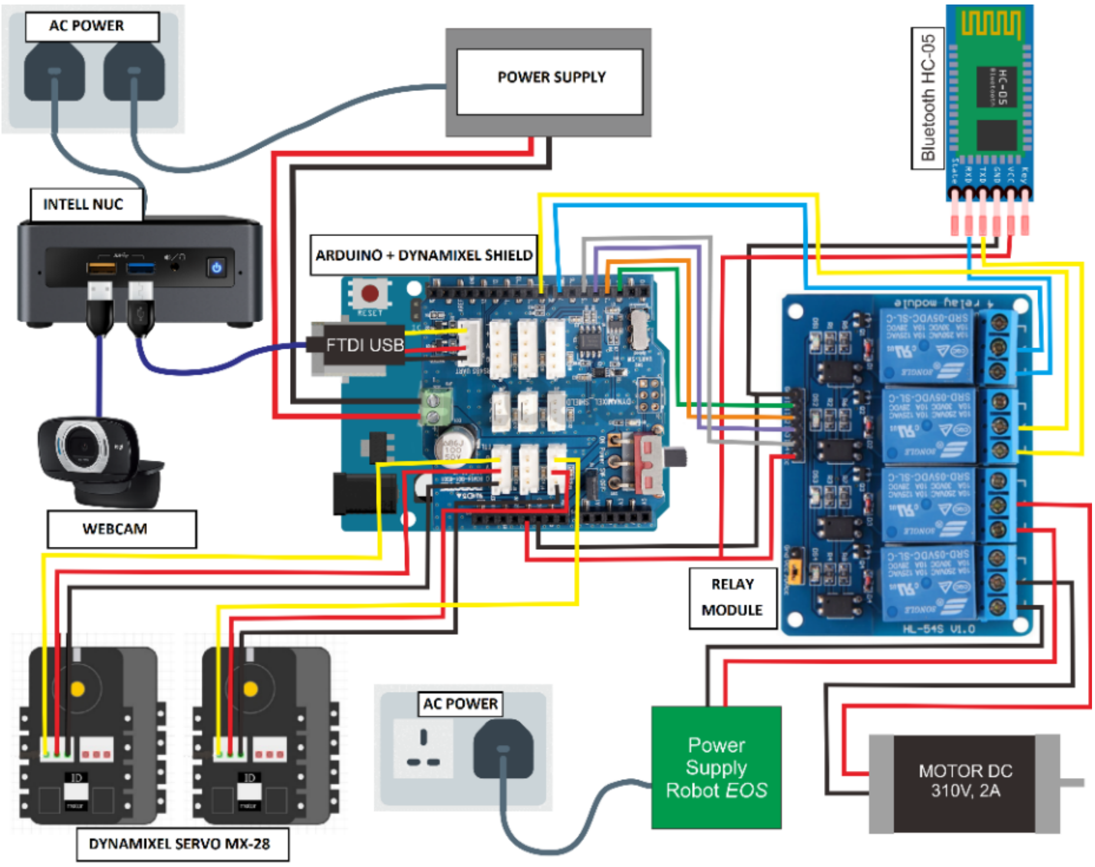


FIGURE 4. Hardware design

effector is set with the positional rotation mode and the Dynamixel Servo on the base arm of the robot is set with the continuous rotation mode. In addition, the Dynamixel Servo can also be adjusted in the direction of rotation: CCW (Counter Clockwise) and CW (Clockwise).

The top of the Dynamixel Servo (located on the end effector) can rotate by directly accessing the degree value or by incrementing the degree. The upper Dynamixel Servo is also limited, in the 90-190 degree range. Whereas, the lower Dynamixel Servo (located on the base arm of the robot) rotates 360 degrees one turn at a time and can rotate a maximum of 8 rounds without stopping. In addition, the lower Dynamixel Servo is also given a rotation limit to the left and right.

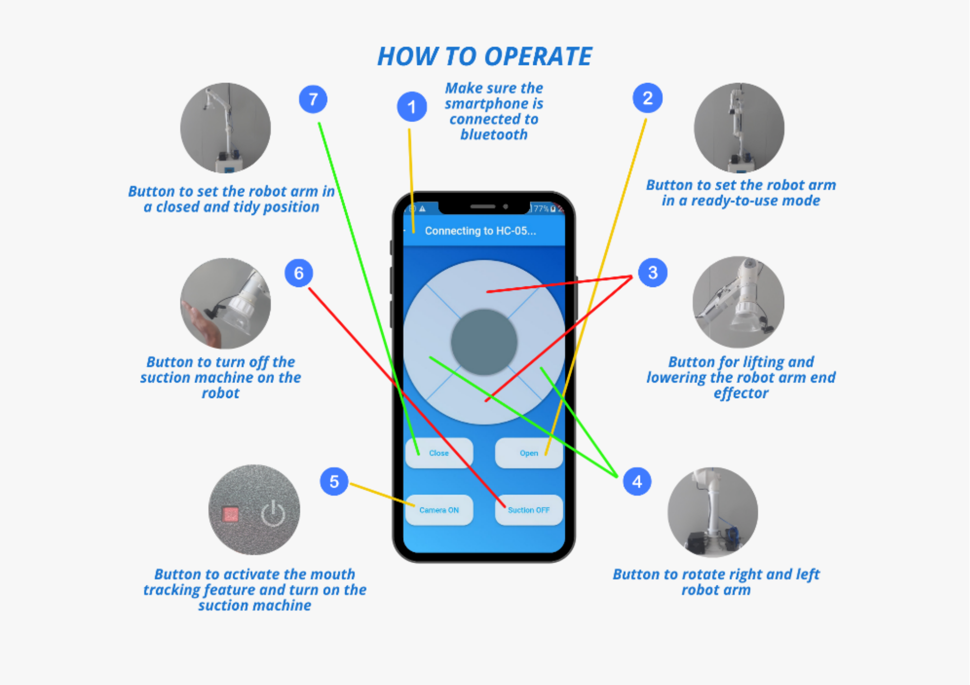
2.2. **Robot controller app.**

We have developed a Flutter-based android application that aims to control the servo rotation on the EOS robot arm. The application uses Bluetooth communication to send or receive data to the Arduino UNO.

The following Figure 5 is a more complete explanation of each button on the application, such as being able to control the EOS robot arm so that we can position the arm to open when the robot wants to be used; on the other hand, we can also close and tidy up the robot arm when the robot does not want to be used. In addition, the application can also set the relay switch so that the application's Bluetooth Communication protocol and the embedded USB FTDI PC can run alternately.

2.3. **Mouth openness detection.**

We utilized the Drowsiness Detection dataset from kaggle.com consisting of 290 images of the driver yawning or opening the mouth [14]. We used the dataset because the oral condition in the dataset is close to the patient's



HOW TO OPERATE

1 Make sure the smartphone is connected to bluetooth

2 Button to set the robot arm in a ready-to-use mode

3 Button for lifting and lowering the robot arm end effector

4 Button to rotate right and left robot arm

5 Button to activate the mouth tracking feature and turn on the suction machine

6 Button to turn off the suction machine on the robot

7 Button to set the robot arm in a closed and tidy position

FIGURE 5. How system works

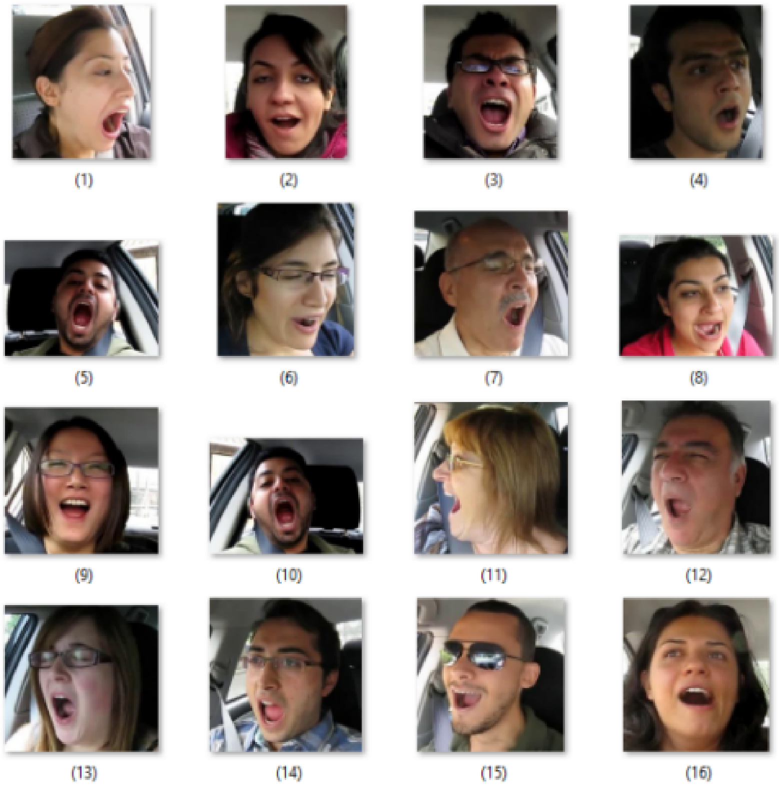


FIGURE 6. Drowsiness Detection dataset

requirement when opening the mouth in dental care. Furthermore, we have flipped every image of the dataset. Therefore, we have 580 datasets. We split the dataset into 480 images for model training and 100 images for model training validation. The Drowsiness Detection dataset is illustrated in Figure 6.

We labeled the open mouth region of each image using labelimg.py, resulting in a corresponding eXtensible Markup Language (XML) file. Furthermore, all data recorded in the XML file will be made into the Comma Separated Values (CSV) file format. We also create a label_map.pbtxt file containing the ids and annotations of all the data sets. All CSV file data and label_map.pbtxt will be sorted and recorded into a TFRecard file that contain byte strings for data payloads, plus data lengths.

TensorFlow is Google's computational framework that can effectively implement a variety of deep learning algorithms and provide excellent support for convolutional neural networks [15]. We have used TensorFlow models to troubleshoot openness mouth detection. We have used three models from TensorFlow: EfficientDet D0 512 × 512 with a speed reference of 39 ms and mean average precision 33.6 on the COCO dataset, Single Shot Detector (SSD) MobileNet V2 FPNLite 320 × 320 with a speed reference of 22 ms and mean average precision of 22.2 on the COCO dataset, and an SSD ResNet50 V1 FPN 640 × 640 with a speed reference of 46 ms and a mean average precision of 34.3 on the COCO dataset [16].

The EfficientDet is a scalable and efficient target detection algorithm that improves the fusion structure of FPN multidimensional features and borrows ideas from the EfficientNet model display method for reference. EfficientDet consists of three components: EfficientNet, Bi-directional Feature Pyramid Network (BiFPN), and Prediction Detection Network. EfficientNet was previously trained on the ImageNet backbone network. Next, go to the BiFPN section, which combines top-down and bottom-up features in EfficientNet to frequently generate Level 3-7 features, followed by the stage of introducing the prediction network, where a box is seen as a prediction indicator. Since EfficientDet is an anchor-based target detector, the initial anchor value must be adjusted correctly to get the best results [17]. EfficientNet's multi-scale feature function algorithm formula is shown below:

$$P^{tp}n = Conv\left(\frac{w1 * P^{in}n + w2 * Resize\left(P^{in}n + 1\right)}{w1 + w2 + \varepsilon}\right) \tag{1}$$

$$P^{out}n = Conv\left(\frac{w'1 * P^{in}n + w'2 * P^{tp}n + w'3 * Resize\left(P^{out}n - 1\right)}{w'1 + w'2 + w'3 + \varepsilon}\right) \tag{2}$$

$P^{in}n$ is the graph of the input features of the nth layer, $P^{tp}n$ is the intermediate feature graph of the nth layer, $P^{out}n$ is the output feature graph of the nth layer, and wi represents the weight occupied by the ith feature [18]. The system detects the opening of the target's mouth in real time. More BiFPN layers improve detection performance but decrease inference speed. EfficientDet D0 is the fastest algorithm among all EfficientDet models because it has a simple BiFPN layer for detecting objects. Therefore, small mathematical parameters are suitable for application in embedded systems.

Besides that, we used Single Shot Detector (SSD) MobileNet to detect mouth openness. The SSD network architecture was based on VGG-16 and was used to extract basic features. Convolution network series can also extract advanced features from objects (detecting targets of different scales) using a multi-scale prediction structure [19]. There are two parts of the SSD MobileNet algorithm. The first part is the MobileNet network located on the front end to extract the initial characteristics of the target (based on VGG-16). The second refers to a network that resides on the back end that can detect multi-scale features to get the distinctive features of the front-end network under different conditions [20]. The architecture of the SSD model using MobileNet to extract the feature map is shown in Figure 7 [21].

Depth-separable Convolution mainly involves two separate layers: Depthwise Convolution and Pointwise Convolution. The Depthwise layer will filter whereas the pointwise

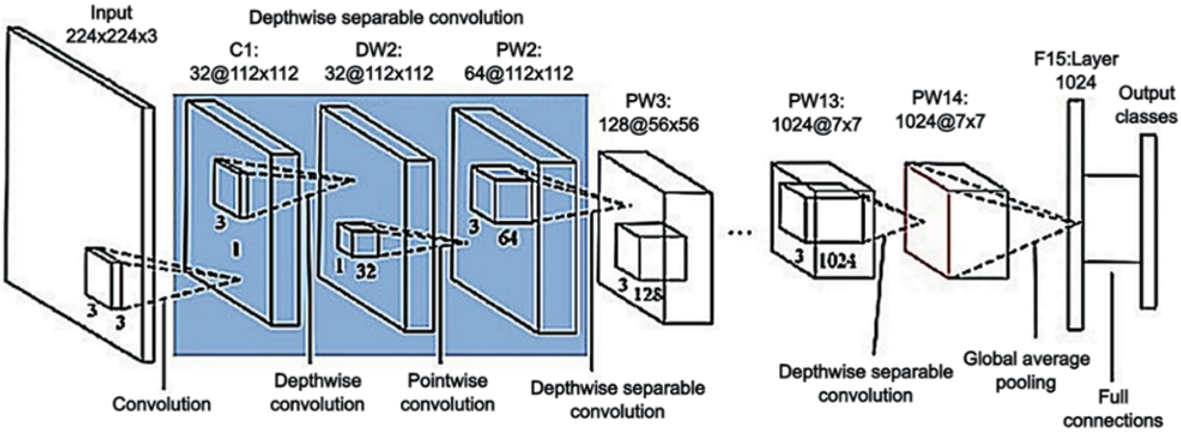


Figure 7. SSD MobileNet architecture

layer combines feature maps from different channels. With this split operation, compute costs are significantly reduced [25].

In addition, we also use the SSD-ResNet50 model for mouth openness detection. A residual module is an approach used in ResNet50 (Residual Neural Network). The module serves to unblock the potential optimization of deep networks. Identity mapping shortcut connections between network inputs and outputs can be added with a specific approach [22].

Figure 8 shows the architecture model of ResNet50, where x is the residual block input, $F(x)$ is the residual mapping, and ReLu is the activation function. In the standard advanced propagation artificial neural network, it is necessary to perform jump links to pass through i links [24]. Hence, the information of input x is directly passed to the output. Each time a layer is passed through, one remaining block will be generated [23].

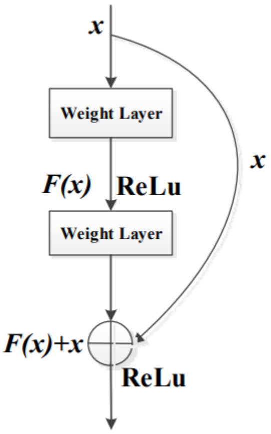


Figure 8. ResNet residual block structure diagram

2.4. **Automatic mouth tracking.** The system combines mouth openness detection image processing with servo control. In this case, the system moves the upper servo parallel to the x-axis of the camera shown in Figure 9. In this feature, the z-axis is set fixed because the fixed joint has not been paired with a servo. On the other hand, the y-axis is set fixed in this case. The y-axis is closely related to the lower servo rotating against the z-axis, and lower servo is not used in this feature because the lower servo takes 0.92

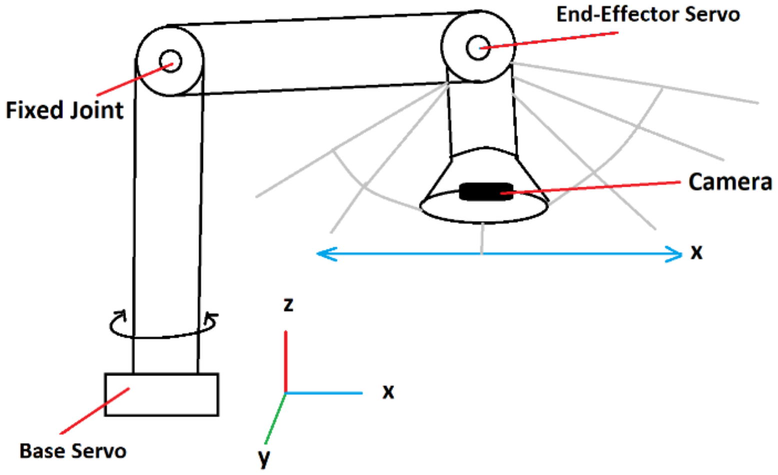


FIGURE 9. Arm robot architecture

seconds to perform one command, which is one full 360-degree turn. This causes data buffering on the microcontroller because the mouth openness detection process is carried out in real time.

We use PBVS to move the robot arm according to the position of the human mouth.

$$ZPBVS = (Z^*) - (h_{-1})(h'(Z^*)) \tag{3}$$

PBVS predicts the relative object to the camera coordinates and the position of the system. The error of the image must be transmitted to form the motion and rotation of the camera [12]. In the case of PBVS, the error is in h and $h'(Z^*)$ is the midpoint of the camera frame, where $ZPBVS$ will converge to Z^* (target pose).

We provide an error tolerance of 80 pixels. This is because, within one degree, the increase from the servo is 50-80 pixels at a distance of approximately 15-25 cm of the camera from the object. When there is an error of 80 pixels, the upper servo moves up 1 degree (increase), and vice versa; when there is an error smaller than or equal to −80, the upper servo moves down 1 degree (decrease). As long as the error is still above 80 pixels, the servo end effector will continue to increase, whereas when the difference in error is still below −80 pixels, the servo end effector will still decrease. An illustration of automatic mouth tracking can be seen in Figure 10.

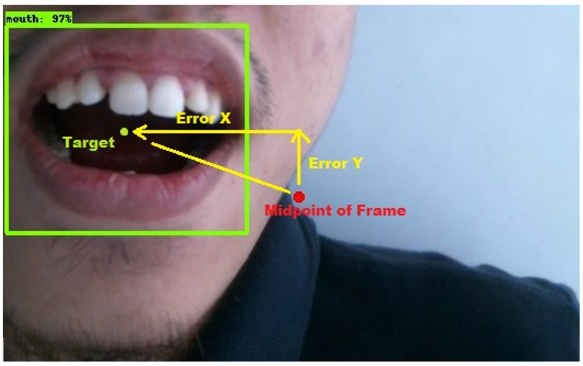


FIGURE 10. Automatic mouth tracking illustration

The accuracy calculation is based on comparing the difference between the initial distance of the camera frame and the target mouth to the difference in the final distance between the two. Below is the formula for calculating position accuracy:

$$accuracy = 100 - \left(\frac{final\ error}{initial\ error} * 100\right) \tag{4}$$

The testing of the response speed of the robot's movements is based on a predetermined set point. The speed calculation is based on the difference in distance divided by the time traveled. Here is the formula for calculating the response speed:

$$response\ speed = \frac{total\ error}{time} \tag{5}$$

3. **Results and Discussion.**

This research focuses on the construction of COVID-19 aerosol suction robots and the expansion of methods for building and choosing which methods are suitable for the automatic mouth openness tracking feature. The open mouth tracking process is integrated using visual servoing methods consisting of PBVS as a control approach and deep learning for open mouth detection. Here are the results of our research.

3.1. **Evaluation model TensorFlow.**

There are three general indicators in evaluating the training model, namely precision (what is the proportion of positive samples out of the predicted number that were correctly predicted positive class?), recall (what proportion of the actual objects did we capture?), and loss total (what is the ratio of the total evaluated samples that were incorrectly predicted?) [26]. We have assigned 100 step number of model training to each TensorFlow model. The evaluation model based on validation dataset training is shown Figure 11.

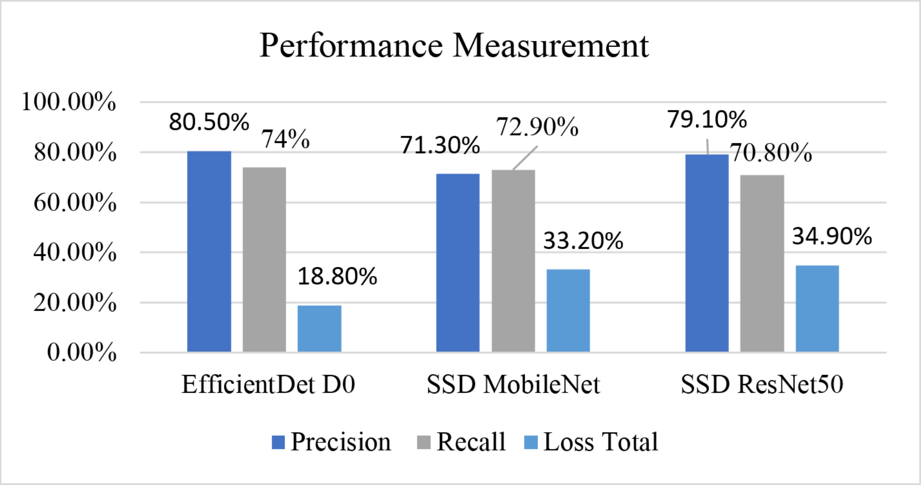


FIGURE 11. Performance measurement of TensorFlow model

3.2. **Mouth openness detection comparison results.**

Based on the evaluation results of each model, EfficientDet is the method with the highest level of accuracy, achieving the lowest loss of 18.8%. However, this model takes longer than other models at 55 ms to detect objects. On the other hand, the SSD MobileNet has a total loss of 33.2% with a detection speed of 26 ms (fastest than other models), whereas the SSD ResNet50 has a total loss slightly larger than the SSD MobileNet total loss of 34.9% with a detection speed of 48 ms.

Mouth openness detection testing was carried out using 100 images of people opening their mouths. There was a condition of turning their heads right and left and rotating the face approximately 10-30 degrees. The distance between the head and the camera is approximately 15-25 cm, as shown in Figure 12.

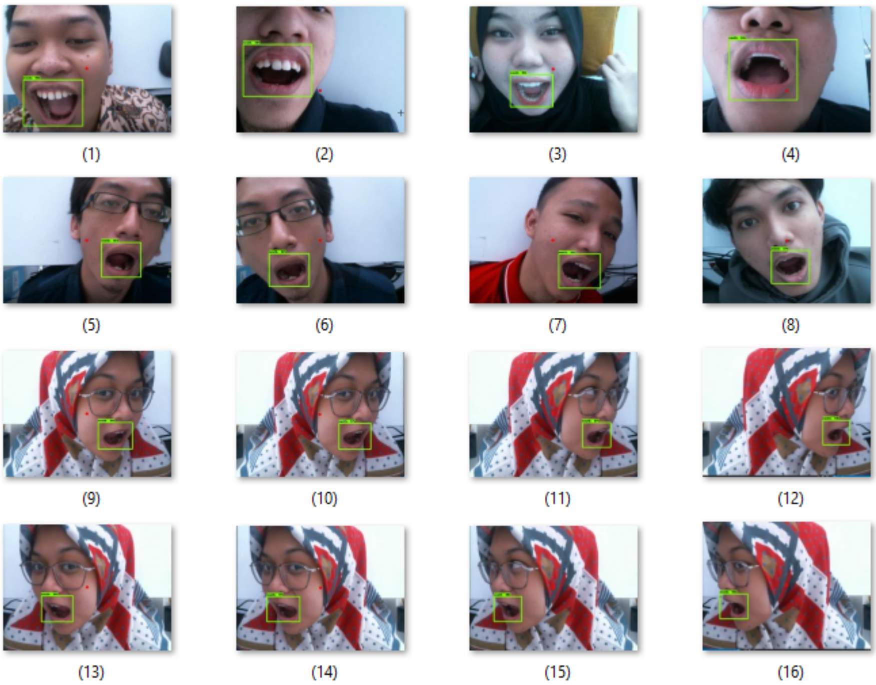


Figure 12. Mouth openness detection result

The following are the results of the mouth openness test shown in Table 1.

Table 1. Mouth openness comparison result

Method	Success detection			Not detected	Accuracy
	Rotating face	Facing forward	Turn head		
EfficientDet D0	12/12	44/44	44/44	0/100	100%
SSD MobileNet	12/12	41/44	44/44	3/100	97%
SSD ResNet50	12/12	44/44	42/44	2/100	98%

The EfficientDet D0 model has a detection accuracy of 100% based on the 100 datasets we have prepared. On the other hand, the SSD MobileNet model achieves an open mouth detection accuracy of 97%; the SSD MobileNet model cannot detect open mouths when the head turns approximately 50 degrees. Whereas the SSD ResNet50 gets an accuracy of 98%, this model has difficulty detecting an open mouth when the head is approximately 15 cm.

3.3. **Automatic mouth tracking results.** Testing the accuracy of robot movements based on tracking the human mouth area is carried out in a room with normal lighting where the object (human) is lying and facing upwards (to the EOS robot's sucking nozzle). The distance between the head and the nozzle is approximately 15-25 cm, with the robot's position on the person's left side. In this test, there were 13 different subjects of people. The subject will turn his head to the right or left to test the accuracy and response speed of the mouth-tracking feature. The human mouth tracking feature is given an error tolerance of 80 pixels. The accuracy test of the robot's movements is based on the actual condition of the mouth position by taking 10 fps test data, as shown in Figure 13.

We have evaluated 13 robotic arm movement control tests based on the camera frame. Each experiment has a different target distance. In the frame of the camera, we divide 2

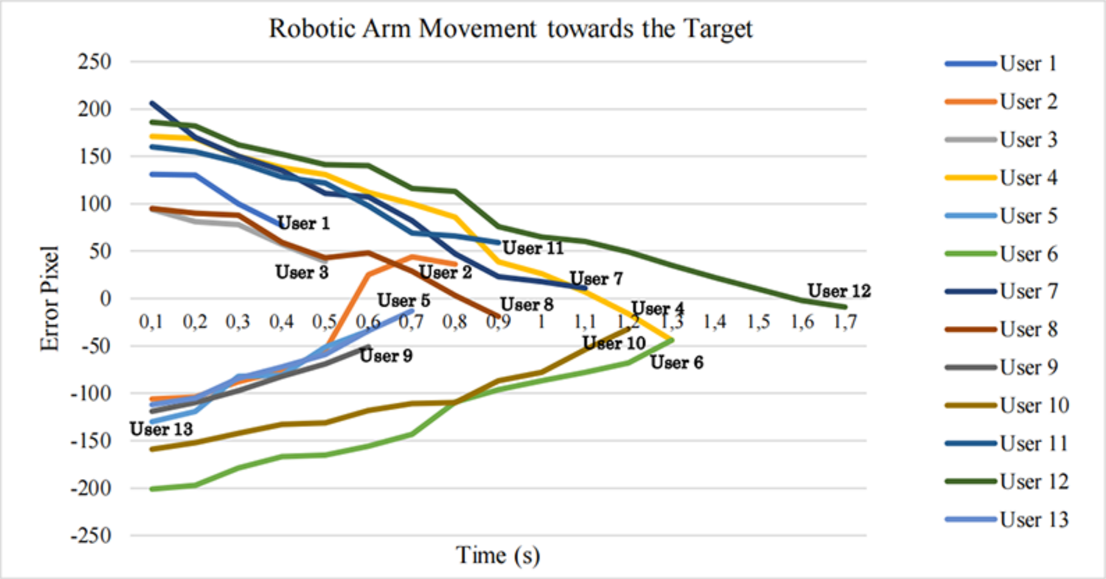


Figure 13. Mouth tracking comparison result

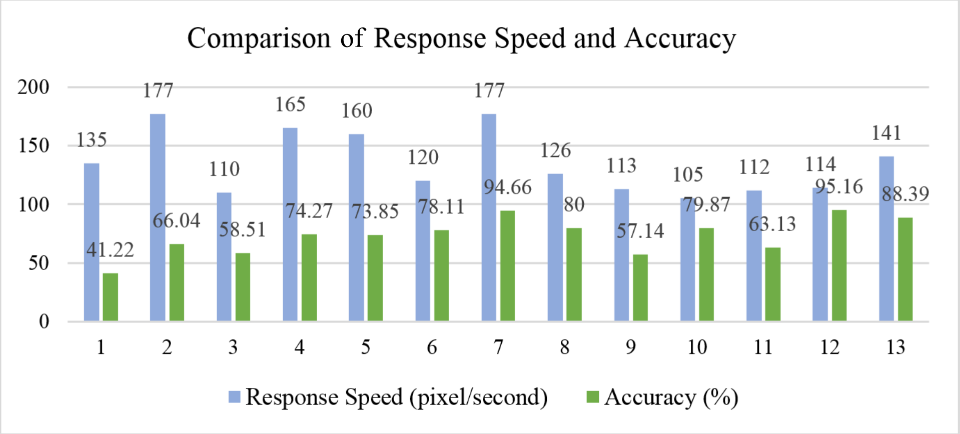


Figure 14. Mouth tracking accumulation result

parts, namely the right and left parts by drawing a vertical line. When the starting point of movement is positive, the target must be located on the right side of the camera frame, and vice versa when the point is negative, it is certain that the target is on the left side of the camera frame.

Figure 14 shows the accumulated response speed and accuracy of oral tracking tests based on pixel errors on the camera.

Based on Figure 14, the results of human mouth tracking tests have an average response speed of 135 px/s and an average accuracy of 73.1% for pixel displacement (midpoint frame camera) in the image. The accuracy result is also influenced by the pixel error tolerance of 80 pixels to the position target.

4. **Conclusions.**

We built a COVID-19 aerosol suction robot based on mouth detection. Here we used a PBVS (Position-Based Visual Servoing) approach integrated with the TensorFlow deep learning model. We have tried three tensor models: EfficientDet D0, Single Shot Detector (SSD) MobileNet, and SSD ResNet50. Based on the mouth openness detection test results, EfficientDet D0 has the highest accuracy reaching 100%. On the other hand, SSD MobileNet has an accuracy of 97%; this model has difficulty detecting an open mouth at a close distance of 15 cm, whereas the SSD ResNet50 has an accuracy of 98%; this model has difficulty detecting an open mouth when the head is turned about 50 degrees. Based on the model evaluation results, the higher the object detection accuracy is, the longer it will take to detect objects and vice versa. Moreover, the faster the object detection computation time is, the less object detection accuracy will be. In addition, based on the results of automatic mouth openness tracking, the system can track the position of the human mouth with a response speed of 135 px/s and an accuracy of 73.1% against the average pixel displacement (midpoint frame camera) with an error tolerance of 80 pixels. As future direction, the response speed and accuracy could be enhanced by adding some DoF (Degree of Freedom) of the EOS robot to move with more variety and precision. In addition, upgrading a faster microcontroller and embedded PC could be more real-time in arm robot movement.

Acknowledgment. We would like to thank the Electronic Engineering Polytechnic Institute of Surabaya through the Penelitian Unggulan Perguruan Tinggi program. We also would like to extend our sincere thanks to the researchers at the Signal, Vision, and Graphics Laboratory, Electronics Engineering Polytechnic Institute of Surabaya.

REFERENCES

[1] World Health Organization, *COVID-19 Weekly Epidemiological Update Edition 117 Published 9 November 2022*, 2022.

[2] World Health Organization, *Transmission of SARS-CoV-2: Implications for Infection Prevention Precautions*, Jakarta, 2021.

[3] J. M. Samet, K. Prather, G. Benjamin and S. Lakdawala, *Airborne Transmission of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2): What We Know*, Oxford University Press, Oxford, 2021.

[4] C. Stevanie, Effectiveness of extraoral suction in dental practice during the COVID-19 pandemic (Efektivitas extraoral suction dalam praktik kedokteran gigi di masa pandemi COVID-19), *Jurnal Kedokteran Meditek*, DOI: 10.36452/jkdoktmeditek.v26i3.1919, 2020.

[5] D. A. Christopherson, W. C. Yao, M. Lu, R. Vijayakumar and R. Sedaghat, High-efficiency particulate air filters in the era of COVID-19: Function and efficacy, *SAGE Journals*, DOI: 10.1177/01945 99820941838, 2020.

[6] K. Bull, Cabin air filtration: Helping to protect occupants from infectious diseases, *Travel Medicine and Infectious Disease*, vol.6, no.3, pp.142-144, 2008.

[7] D.-K. Kim and D.-H. Kang, UVC LED irradiation effectively inactivates aerosolized viruses, bacteria, and fungi in a chamber-type air disinfection system, *Applied and Environmental Microbiology*, vol.84, no.17, e00944-18, DOI: 10.1128/AEM.00944-18, 2018.

[8] J. Pomares, Visual servoing in robotic, *Electronics*, vol.8, no.11, 1298, DOI: 10.3390/electronics811 1298, Alicante, 2019.

[9] S. Sarkar, G. Ghosh, A. Mohanta, A. Ghosh and S. Mitra, Arduino UNO based foot pressure sensitive smart safety system for industrial robots, *2017 2nd International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Durgapur, India, DOI: 10.1109/ICECCT. 2017.8118009, 2017.

[10] Y.-C. Peng, D. Jivani, R. J. Radke and J. Wen, Comparing position- and image-based visual servoing for robotic assembly of large structures, *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, Hong Kong, China, pp.1608-1613, DOI: 10.1109/CASE 48305.2020.9217028, 2020.

[11] G. Palmieri, M. Palpacelli, M. Battistelli and M. Callegari, *A Comparison between Position-Based and Image-Based Dynamic Visual Servoings in the Control of a Translating Parallel Manipulator*, Hindawi Publishing Corporation, DOI: 10.1155/2012/103954, 2012.

[12] H. Shi, J. Chen, W. Pan, K.-S. Hwang and Y.-Y. Cho, Collision avoidance for redundant robots in position-based visual servoing, *IEEE Systems Journal*, vol.13, no.3, pp.3479-3489, DOI: 10.1109/ JSYST.2018.2865503, 2019.

[13] D. Calaver, L. Floroian and S. M. Grigorescu, Assistive rehabilitation using a 7-DoF robotic arm with self-collision and obstacle avoidance system, *2019 E-Health and Bioengineering Conference (EHB)*, Iasi, Romania, pp.1-4, DOI: 10.1109/EHB47216.2019.8970082, 2019.

[14] A. D. Nishad, *Driver Drowsiness Using Keras*, https://www.kaggle.com/code/adinishad/driverdrowsiness-using-keras/dat, 2021.

[15] Y. Ju, X. Wang and X. Chen, Research on OMR recognition based on convolutional neural networks TensorFlow platform, *2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Qiqihar, China, pp.688-691, DOI: 10.1109/ICMTMA.2019.00157, 2019.

[16] TensorFlow, *TensorFlow 2 Detection Model Zoo*, www.github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md, 2022.

[17] L. Cao, X. Zhang, J. Pu, S. Xu, X. Cai and Z. Li, The field wheat count based on the EfficientDet algorithm, *2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Dalian, China, pp.557-561, DOI: 10.1109/ICISCAE51034.2020.9236918, 2020.

[18] K. Lin, J. Chen, A. Chen and H. Huang, Application of the EfficientDet algorithm in traffic flow statistics, *2020 16th International Conference on Computational Intelligence and Security (CIS)*, Guangxi, China, pp.140-143, DOI: 10.1109/CIS52066.2020.00038, 2020.

[19] J. Zhang, Y. Zhang, Y. Yan, R. Lin and X. Wang, MobileNet-SSD with adaptive expansion of receptive field, *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, Chongqing, China, pp.177-181, DOI: 10.1109/IICSPI51290.2020.9332204, 2020.

[20] G. Yu, L. Wang, M. Hou, Y. Liang and T. He, An adaptive dead fish detection approach using SSD-MobileNet, *2020 Chinese Automation Congress (CAC)*, pp.1973-1979, 2020.

[21] W. Wang, Y. Li, T. Zou, X. Wang and J. You, A novel image classification approach via Dense-MobileNet models, *Mobile Information Systems*, DOI: 10.1155/2020/7602384, 2020.

[22] X. Hu and B. Huang, Face detection based on SSD and CamShift, *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, pp.2324-2328, DOI: 10.1109/ITAIC49862.2020.9339094, 2020.

[23] S. Chen, J. Liu, W. Niu, Y. Han, Y. Xiao, Y. Xue and X. Ye, Research on the detection method for insulation piercing connectors and bolts on the transmission line based on SSD algorithm, *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chengdu, China, pp.960-964, DOI: 10.1109/IAEAC47372.2019.8997943, 2019.

[24] A. Lin, Y. Liu and L. Zhang, Mushroom detection and positioning method based on neural network, *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, pp.1174-1178, DOI: 10.1109/IAEAC50856.2021.9390669, 2021.

[25] F. Zhang, Q. Li, Y. Ren, H. Xu, Y. Song and S. Liu, An expression recognition method on robots based on MobileNet V2-SSD, *2019 6th International Conference on Systems and Informatics (ICSAI)*, Shanghai, China, pp.118-122, DOI: 10.1109/ICSAI48974.2019.9010173, 2019.

[26] K. R. Ummah, T. Karlita, R. Sigit, E. M. Yuniarno, I K. E. Purnama and M. H. Purnomo, Effect of image pre-processing method on convolutional neural network classification of COVID-19 CT scan images, *International Journal of Innovative Computing, Information and Control*, vol.18, no.6, pp.1895-1912, 2022.

Author Biography

Riyanto Sigit received a B.S. degree (1995) and an M.S. degree (2005) in Computer Engineering and Informatics Engineering from Sepuluh Nopember Institute of Technology, and Ph.D. degree (2014) in Electrical Engineering from the National University of Malaysia. He has been a lecturer in computer engineering at the Electronic Engineering Polytechnics Institute of Surabaya (1995-2021). He has authored and co-authored more than 70 journal and conference papers in the various health and biomedical research areas including medical image processing, robotics, artificial intelligent and programming. His current interest is preprocessing, segmentation and classification in echocardiograph, CT scan images and face recognition on healthcare.

Aditia Yuliyanto received a B.A.Eng. degree in Computer Engineering from Electronic Engineering Polytechnic Institute of Surabaya, Indonesia in 2022. From August 2021 to the present, he is a Research Assistant in the Signal Vision and Graphic (SVG) Laboratory. He owns one patent in the form of a digital eye check platform and one conference papers on robotics. His research interests include computer vision, machine learning, deep learning, medical imaging, robotics, and data science. The author is the recipient of the gold medal and the Malaysia Innovation, Invention and Creativity Association (MIICA) Special Award at the Youth International Science Fair 2021.

Moch. Rochmad received a B.S. degree (1989) and an M.S. degree (2003) in Control System Engineering. He has been a lecturer in electro engineering at the Electronic Engineering Polytechnics Institute of Surabaya (1991-2023). He has conducted research in the various health and biomedical research areas including electrocardiogram signals, heart rate and blood pressure, etc. His current interest is preprocessing, segmentation and classification in echocardiograph and face recognition on healthcare.

Imam Safari Azhar received a D.D.S. degree (2012) and an M.D.S. degree (2018) in Dental Health Science from Universitas Airlangga, and Prosthodontics Spesialist (2021) from Universitas Airlangga. He has been a lecturer in Dentistry Universitas Airlangga (2014-Now). He has authored and co-authored more than 20 journal and conference papers in Prosthodontics, the various health and biomedical research areas including medical image processing, artificial intelligent, implant dentistry and bone augmentation. His current interest is preprocessing, segmentation and classification in Implant Dentristry, Bone Augmentation and Artificial Intelligent.