Development of an M-Health Application with Seven-Segment Digit Recognition Feature for Reading Values on Digital Sphygmomanometer

Hansel Valentino Tanoto, Rinaldi Munir, and Nur Ahmadi, *Member, IEEE*

***Abstract*—Blood pressure is a crucial metric in health assessments as it reflects an individual's general health status and could help the diagnosis of cardiovascular diseases, which are the leading cause of death worldwide. Currently, blood pressure data recording is still performed manually, making it prone to human error and inefficiency. Therefore, there is a need for a solution that enables accurate, fast, and practical reading and recording of blood pressure measurement data. This paper proposes the development of a mobile health (m-health) application prototype equipped with 4 features: blood pressure measurement reading using deep learning, data visualization, user management, and authentication. The model was trained using 3,649 images of sphygmomanometer with objective to recognize seven-segment digits representing 3 blood pressure metrics. Various YOLOv8 model variants—small, medium, and large—were utilized in the training. Each variant also underwent model compression techniques such as quantization and pruning. The evaluation result indicated that the small variant of the YOLOv8 model, that quantized to INT8, proved to be the most suitable model. This is attributed to its compact size (11 MB) and short inference time (641.4 ms). The model has achieved a seven-segment digit detection accuracy of 99.28% and an f1-score of 96.48%. The model was successfully deployed in the m-health application, with a slight increase in average inference time to 1867.6 ms. Furthermore, direct testing of the model on 40 images within the m-health application yielded a seven-segment digit grouping accuracy of 96.67% and an overall image reading accuracy of 95%.**

***Index Terms*—Seven-segment digit, m-health, YOLOv8, blood pressure, digital blood pressure monitor.**

I. Introduction

HEALTH is one of the most important aspects of human life. Poor health conditions can indeed disturb and impede an individual's daily activities. It's important to note that, health encompasses not only curative treatment but also preventive measures aimed at sustaining bodily well-being. An example of preventive action involves conducting regular health checkups. These activities play a role in the early detection of health problems that may go unnoticed.

One of the most commonly used health metrics is blood pressure, which measures the pressure exerted on the walls of blood vessels due to the heart's pumping activity [1]. Blood pressure is composed of two values: systolic and diastolic pressure. Systolic pressure is the blood pressure value when the heart is contracting, while diastolic pressure is the blood pressure value when the heart is relaxing. These two values are critical for understanding cardiovascular health conditions. Moreover, they are relatively easy to obtain and can provide a general overview of an individual's vital signs and fitness, making them frequently used by doctors as a basis for diagnosing patients [2], [3].

Hansel Valentino Tanoto is with the School of Electrical Engineering and Informatics, Bandung Institute of Technology, 40132, Indonesia (e-mail: 13520046@std.stei.itb.ac.id).

Rinaldi Munir is with the School of Electrical Engineering and Informatics, Bandung Institute of Technology, 40132, Indonesia (e-mail: rinaldi@staff.itb.ac.id).

Nur Ahmadi is with the Center for Artificial Intelligence (U-CoE AI-VLB), School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, 40132, Indonesia (e-mail: nahmadi@itb.ac.id).

TABLE I
General blood pressure classification [4]

Category	Systolic		Diastolic
Optimal	< 120	and	< 80
Normal	$120 - 129$	and/or	$80 - 84$
High Normal	$130 - 139$	and/or	$85 - 89$
Class I Hypertension	$140 - 159$	and/or	$90 - 99$
Class I Hypertension	$160 - 179$	and/or	$100 - 109$
Class I Hypertension	≥ 180	and/or	≥ 110
Isolated Systolic Hypertension	≥ 140	and	< 90

Regular and consistent blood pressure measurements can help in the early detection of cardiovascular issues such as hypertension, heart attacks, or strokes. Elevated blood pressure is often dubbed a "silent killer" because it frequently shows no early symptoms yet has the potential to cause cardiovascular organ damage or failure [1]. This is why cardiovascular diseases rank at the top as the leading cause of death worldwide [5]. Therefore, for individuals with cardiovascular diseases, regular blood pressure measurement is essential for monitoring their health condition to prevent the emergence of more severe illnesses or complications. The blood pressure general classification is shown in Table I

The rapid advancement of technology has caused disruptive effects in various aspects of life, including healthcare. One significant development is the growing use of telecommunication technology, particularly mobile devices, for health monitoring services known as m-health (mobile health). The existence of m-health enables individuals to manage their health data and/or connect with medical professionals without having to meet in person [6]. The seamless integration of technology into healthcare not only enhances access to crucial health insights but also encourages proactive health management. Ultimately, this contributes to improved well-being and fitness levels by empowering users to monitor their health condition and implement preventive measures based on their medical data.

Typically, when conducting health checkups at healthcare facilities, medical personnel still manually collect and record patients' blood pressure data. This recording is usually done on paper and then manually entered into a system or database [7]. The same applies to self-recording of blood pressure through m-health applications by individuals, which is also generally typed manually. This activity is not practical, tends to be slow, and requires additional effort. Additionally, this process is prone to errors, whether due to data entry mistakes (human error) or the loss of records because the papers are misplaced.

One alternative solution is to utilize wireless communication technologies such as Bluetooth or Wi-Fi. With this technology, blood pressure measurement data can be directly sent to the m-health application or other data recording system. However, not all currently available blood pressure monitors are equipped with this technology. According to data from the British & Irish Hypertension Society, of the 176 recommended digital blood pressure monitors, less than 3% use Bluetooth technology to send measurement data to users' phones [7]. Implementing a widespread replacement of blood pressure monitors with models that include wireless communication technology would be highly challenging and inefficient in terms of both cost and resources. Therefore, another solution is needed that can be easily implemented on a large scale without burdening individuals to buy new blood pressure monitors with wireless communication technology.

Another viable solution is to use image recognition methods to detect and extract the values (digits) displayed on measurement devices quickly and accurately. In this approach, what patients need to do is simply measure their blood pressure and then photograph the results using their mobile phones. The m-health application will then use image recognition algorithms to process (detect) and save the blood pressure data. The image recognition model will be designed to identify seven-segment digits, given their widespread use in various medical devices, including digital blood pressure monitors, for displaying measurement outcomes. The extensive adoption of seven-segment displays is due to their ability to present numerical data in a simple, economical, and energy-efficient manner, ensuring clarity and precision. Their straightforward design, consisting of seven individual segments that can be illuminated in various combinations, makes them highly reliable and easy to read, even in low-light conditions. Furthermore, their widespread adoption across various medical devices ensures consistency and familiarity for healthcare professionals and patients alike, facilitating ease of use and reducing the likelihood of misinterpretation.

The paper is organized as follows. Section II presents the related works, research gap, and summary of our contribution to the present work. Section III describes the proposed solution design and analysis. Section IV details the implementation process of the proposed solution including dataset preparation, model training, model compression, and application development. The evaluation of the resulting model and m-health application are presented and discussed in Section V. Finally, conclusions are drawn in Section VI.

II. Related Works

The intersection of technology and healthcare has seen remarkable advancements, particularly in the realm of health monitoring systems. Extensive research has explored various methods for accurately detecting and interpreting seven-segment digits, which are widely used in medical devices for displaying numerical data. Finnegan *et al.* conducted research on seven-segment digits detection and reading in blood pressure monitors and blood glucose meters using conventional image processing techniques [7]. Their algorithm achieved an overall accuracy of 93% in recognizing these digits, with a digit localization F1-score of 80% and classification accuracy exceeding 89%. However, the algorithm's blob extraction component showed occasional misclassifications, particularly between digits 0 and 8, and digits 1 and 7. Additionally, this study's scope was solely dedicated to model development, with no involvement in the creation of a mobile health (m-health) application. Such an application is useful for integrating the model into user-friendly platforms that facilitate real-time health monitoring and data management.

Shenoy and Aalami, in their research, implemented a smartphone-based seven-segment digit detection system (mobile application) to read numeric values on medical devices such as blood pressure monitors, blood glucose meters, and weight scales [8]. The study used computer vision to extract features (digits) and then applied a random forest algorithm to interpret the readings. The resulting mobile application can identify seven-segment digits with an overall accuracy of 98.2%. However, similar to the previous study conducted by Finnegan *et al.*, there remains an issue of misclassification between digits 0 and 8 due to an insufficient volume of training data. Additionally, the model exhibits limited performance in recognizing digits in images that are blurry. Furthermore, the model has yet to be directly integrated into the mobile application but remains hosted on a server. Consequently, the application necessitates internet connectivity and experiences added delay in the detection process due to internet latency.

Another study, conducted by Wannachai *et al.* focused on developing a system for recognizing seven-segment digits using CNN (Convolutional Neural Network) algorithms [9]. Their research aimed to create a system capable of reading numerical statuses on manufacturing machine monitors in factories, eliminating the need for manual data collection by operators. The resulting model was deployed on a server and integrated with cameras placed at various points in the factory as edge devices. Based on the experiment results, the developed system achieved an accuracy rate of 91.1%. However, the weakness of the model is that it was not designed to be deployed directly on devices with limited computational capabilities (edge devices). That results in the model having to be deployed on a server, which leads to additional latency during the detection process.

In this paper, an m-health application prototype is developed with the capability to detect and identify blood pressure measurement values represented in seven-segment digits using image recognition. The model utilizes deep learning, specifically YOLOv8, for image recognition. It also incorporates data augmentation techniques to enhance model robustness, improve performance in handling variations in digit representation, and mitigate misclassification issues in digit recognition. Furthermore, the model is designed to be deployed on edge devices, specifically smartphones, by leveraging model compression techniques to ensure efficient usage of resources and optimal performance. This approach not only enhances the accessibility of blood pressure monitoring but also addresses the challenges of limited computational power and storage capacity typically encountered in edge computing environments. By harnessing the capabilities of YOLOv8 and employing model compression strategies, the proposed m-health application aims to provide accurate and real-time blood pressure measurements conveniently and effectively for improved healthcare management.

III. Solution Design & Analysis

As mentioned earlier, this paper aims to develop an image recognition system to read the measurement results of a digital blood pressure monitor (sphygmomanometer) using a smartphone's camera. The system will detect and identify seven-segment digit objects representing systolic pressure, diastolic pressure, and heart rate per minute displayed on the sphygmomanometer. Based on literature studies and related research, there are several common alternatives for reading seven-segment digit values, namely Optical Character Recognition (OCR) and object detection. Research by Hjelm and Andersson (2022) indicates that object detection methods outperform OCR for detecting seven-segment digits on odometers, especially in images with poor quality due to noise, blur, lighting, viewing angles, etc [10]. This finding is particularly relevant as both odometers and sphygmomanometers utilize seven-segment digits for displaying their measurements. Therefore, object detection is selected as an image recognition method for the m-health application.

In general, object detection can be performed using either image processing techniques or deep learning techniques. This paper chooses to use deep learning due to its typically superior accuracy. Additionally, the deep learning technique does not require manual feature extraction; it can automatically extract relevant features from data and improve the accuracy over time with training. This capability allows deep learning techniques to be more adaptable in managing data variability, such as blur, perspective, rotation, and lighting factors. A deep learning model trained with a sufficiently large dataset will have good generalization capabilities, enabling it to recognize objects, specifically seven-segment digits in this context, in unseen situations during training [11]. Achieving large and diverse dataset can be done through data augmentation techniques, which expand the dataset by creating variations in existing data, such as rotating, cropping, or adding blur, noise, and other visual effects to the images.

The deep learning architecture chosen for this paper is the YOLOv8 neural network. YOLO was selected due to its typically superior performance in AP (Average Precision) and inference time compared to other object detection models [13]. YOLO, known as "You Only Look Once," has been developed since 2015 as an open-source algorithm, thereby providing accessibility to researchers and developers globally. Its popularity has made it gain a large community, resulting in thorough documentation and facilitating easier troubleshooting as well as knowledge sharing among users [12]. The frequent updates to the YOLO architecture, with nearly annual new versions, demonstrate its ongoing enhancement and ability to address new challenges in object detection. YOLOv8, introduced as the eighth version of the YOLO architecture, represented the most recent version in the series at the start of this study. As stated in its official repository, YOLOv8 exhibits enhancements in mAP over its predecessor [14]. Given its recent release in January 2023, YOLOv8 is still relatively new and not widely adopted, prompting this study to explore its implementation.

Meanwhile, the prototype of the m-health application will be developed on the Android platform. The main reason is that Android dominates the market share of smartphone operating systems, especially in Indonesia. According to data published by Statcounter, Android holds approximately 88.33% of the market share in Indonesia as of January 2024. The reason for this is primarily because most smartphone brands that utilize the Android operating system provide lower prices in comparison to other brands like the iPhone with its iOS. This high popularity of Android provides significant potential for the m-health application to reach more users and enhance the accessibility of digital health services.

IV. Implementation

The implementation flow of the m-health application involves two primary stages: model implementation and application development. Model implementation includes dataset preparation, model training & evaluation, and model compression processes. The details of each process will be explained later in the following subsections. Before that, here are the specifications of the working environment used in this work.

- **Google Colab**
  - CPU: Intel® Xeon® CPU @ 2.20GHz
  - GPU: NVIDIA Tesla T4
  - System RAM: 12.7 GB
  - GPU RAM: 15 GB
  - Operating System: Ubuntu 22.04.3 LTS
- **Computer (Asus A409JB)**
  - CPU: Intel® Core™ i5-1035G1 CPU @ 1.00GHz
  - RAM: 12 GB
  - Operating System: Windows 10
- **Smartphone (Samsung A30s)**
  - CPU: Samsung Exynos 7904
  - RAM: 4 GB
  - Operating System: Android 11

The Google Colab environment was used for implementing the model, while the computer was used for developing the m-health application. The mobile phone, on the other hand, was used as the platform to deploy and run the m-health application.

TABLE II
DATASET SOURCES DISTRIBUTION

Source	Amount	Percentage
Data Scraping	441	20.5%
Public Dataset 1: AH (2024) [15]	225	10.5%
Public Dataset 2: Ega (2022) [16]	187	8.7%
Public Dataset 2: Finnegan *et al.* (2018) [7]	1294	60.3%
Total	2147	100%

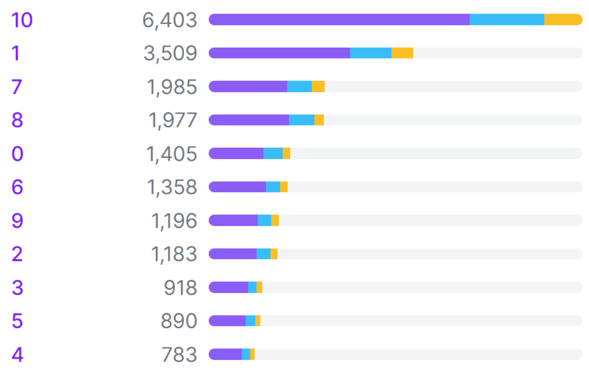


Fig. 1. Class distribution of the dataset.

A. Dataset Preparation

Generally, the data preparation process consists of data collection, data labeling, dataset splitting, and data augmentation stages. The data used in this study are images of blood pressure monitor measurement results that were collected through a combination of public datasets and image scraping algorithms. Details about the sources of the datasets used can be found in Table II. Subsequently, data sorting is performed to eliminate irrelevant or poor-quality data. Once the data is filtered, the labeling process begins, which involves applying bounding boxes to the input data (images) covering a total of 11 classes that consist of 10 classes for digits (0 to 9) and 1 class for indicating systolic pressure, diastolic pressure, or heart rate. Figure 1 displays the class distribution of the whole dataset. In the dataset, Class 10 has the highest occurrence as each image typically contains three objects from Class 10, representing the three blood pressure metrics: systolic, diastolic, and heart rate. Following Class 10, Class 1 has the second-highest occurrence because systolic pressure values generally fall in the range of hundreds. The relatively high frequencies of Class 7 and 8 can be attributed to diastolic pressure and heart rate values, which commonly fall within the 70s to 80s range.

The labeled data is then divided into three groups: training set, validation set, and test set, with percentages of 70%, 20%, and 10%, respectively. The final stage is data augmentation, which aims to increase the volume and variety of training data by twofold. This is intended to improve the model's accuracy and flexibility in real-world conditions. The data augmentation process includes rotations of $\pm15°$, shearing of $\pm15°$, brightness adjustments of $\pm15\%$, blurring up to 3.5 pixels, and adding noise up to 5% of the total pixels. The dataset splits before and after the augmentation process can be seen in Table III.

TABLE III
DATASET SPLIT

Split	Amount	
	Before Augmentation	After Augmentation
Train	1502 (70%)	3004 (82%)
Validation	430 (20%)	430 (12%)
Test	215 (10%)	215 (6%)
Total	2147(100%)	3649(100%)

B. Model Training

In this process, training is conducted on several variants of the YOLOv8 model. YOLOv8 itself has five model variants: nano, small, medium, large, and extra-large. A comparison of these five variants can be found in Table III. As the model size increases, its network complexity and accuracy also improve, but its inference time will be longer. In this study, the training process will be conducted only for the small, medium, and large variants. The nano variant was not selected due to its relatively lower accuracy compared to other variants. Meanwhile, the extra-large variant was not chosen because of its complexity and large size, which is not suitable for use on smartphones with limited computational capabilities.

The three model variants will be trained using the following parameters.

- Epochs: 500
- Patience: 50
- Batch Size: 8, 16, and 32
- Image Size: 640
- IoU: 0.5
- Optimizer: auto

The automatic option for optimizer parameters results in the use of the best optimizer, which is SGD with a learning rate of 0.01 and momentum of 0.9. Meanwhile, other parameters that are not mentioned will use the default configuration. The training results of these three model variants will be evaluated based on accuracy, precision, recall, f1-score, size, and inference time metrics.

C. Model Compression

The model compression includes techniques such as model pruning and quantization. This process aims to enhance detection speed and reduce the computational load of the model, considering it will be deployed on a smartphone which typically has lower computational capabilities compared to computers. The pruning process is implemented using PyTorch by selectively removing weight values in convolutional layers (Conv2d) with a pruning parameter of 0.1. Consequently, the pruned model will have 10% fewer weight values compared to the original model. In other words, 10% of these weight values will be set to 0.

Meanwhile, quantization techniques were applied to convert model weights into 16-bit floating point (FP16) and 8-bit integer (INT8) precision. The FP16 quantization process was conducted using the export feature of the Ultralytics YOLOv8 library, by specifying the format parameter to 'tflite' and also enabling the 'optimize' and 'half' options. The 'optimize' parameter will optimize the model for mobile devices, while the 'half' parameter implements FP16 half-precision quantization. In addition to producing models in the FP16 format, the export results also yielded models in the 32-bit floating point (FP32) TensorFlow format, which will be used for quantization to INT8 precision using the TensorFlow library. During INT8 quantization, a calibration process is necessary to estimate the range of all floating-point values in the model. Constant values such as weights and biases were easily calibrated. However, variable values like inputs, activations, and outputs required cycles of the inference process to be calibrated. Consequently, a dataset of approximately 100 to 500 samples is needed for this calibration process. In this Final Project, a validation dataset consisting of 430 images was used as the calibration dataset.

D. M-Health Development

As previously outlined, this m-health application will feature three main functionalities: blood pressure reading using the camera, recording historical blood pressure data, and visualizing this data in graphical form. Additionally, there will be an authentication feature to bind these data to a user account. The deep learning models developed in the previous stages will be integrated into the m-health application for the blood pressure reading feature.

This m-health application will encompass a total of 9 use cases, as depicted in Figure 2. The first three use cases—login, register, and logout—are part of the authentication feature. Subsequently, the blood pressure data visualization feature is represented as the display measurement history use case. With this use case, users can see the blood pressure measurement history in the form of a line chart and a list of measurement logs. Furthermore, users can record their blood pressure measurement through the use case to add new blood pressure measurement results. Recording blood pressure data can be done in two ways: by directly capturing measurement results or by uploading photos of the results from the gallery. Meanwhile, user management features can be accessed through the last 4 use cases: view profile, add new user, delete user, and edit user information.

The workflow of the blood pressure reading feature is illustrated in Figure 3. Generally, this process is divided into two main parts: digit detection and digit grouping to derive blood pressure values. In the digit detection process, a deep learning model is utilized to detect and classify the digits present in the photo. Subsequently, in the grouping process, these digits are grouped and categorized into systolic pressure, diastolic pressure, and heart rate values. Then, the resulting data is stored in a database to be used for the visualization feature. The data storage is implemented using the Cloud Firestore, a non-relational database owned by Google. This choice ensures that the data is not only stored locally, allowing users to access it even if their local data is deleted. Additionally, Cloud Firestore supports caching, eliminating the need for constant internet access to perform data reading and writing operations. Data on the user's phone is automatically synchronized with the data stored in the cloud once the phone is connected to the internet.

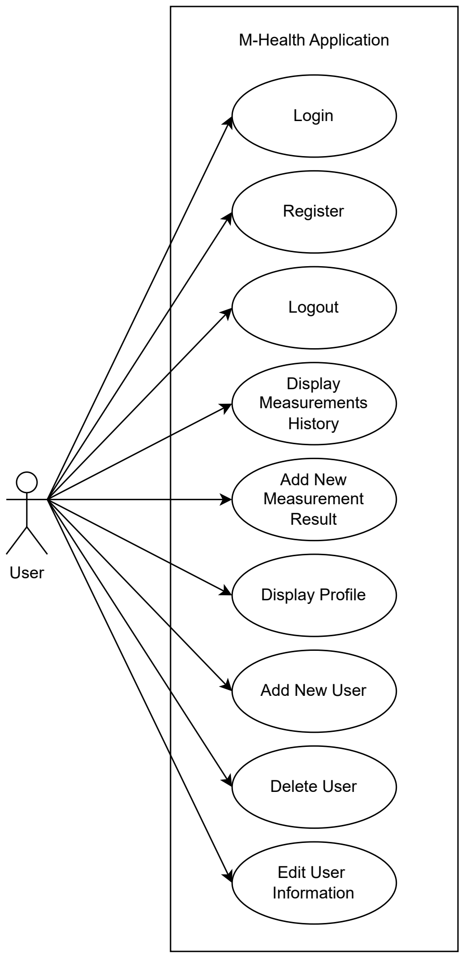


Fig. 2. Use case diagram of the m-health application.

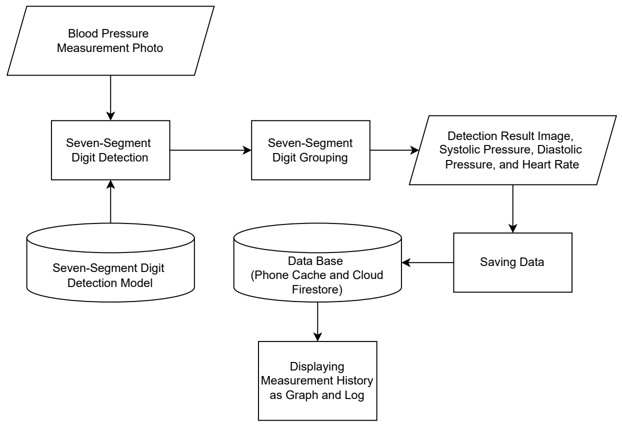

Blood Pressure Measurement Photo
Seven-Segment Digit Detection
Seven-Segment Digit Grouping
Detection Result Image, Systolic Pressure, Diastolic Pressure, and Heart Rate
Seven-Segment Digit Detection Model
Data Base (Phone Cache and Cloud Firestore)
Saving Data
Displaying Measurement History as Graph and Log

Fig. 3. Workflow of m-health's blood pressure measurement reading.

The grouping process of the detected digits into meaningful values is performed by considering the intersection percentage of a digit's bounding box (represented by classes '0' to '9') with a value's bounding box (represented by class '10'). An

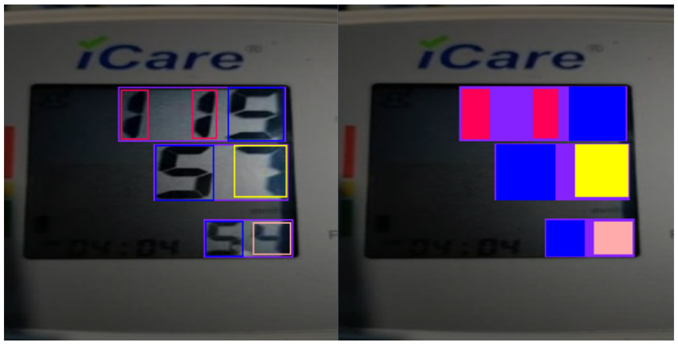


Fig. 4. Illustration of bounding boxes intersection.

illustration of these classes and their corresponding bounding boxes is depicted in Figure 4. Value classes are displayed as purple boxes, while digit classes are displayed as boxes of other colors. A digit class is grouped into a value class if its bounding box intersects with at least 75% of the area of the value's bounding box. The 75% threshold is applied to handle some special cases like a digit bounding box may exceed the boundary of a value class bounding box. Once grouped, the digits within each group are sorted based on their positions from left to right. Then, concatenating these digits results in a meaningful value. As a result, three values representing the three blood pressure metrics are obtained. Subsequently, categorization into systolic pressure, diastolic pressure, or heart rate values is determined by sorting the value's position in the image from top to bottom.

Overall, this m-health application consists of 5 pages: login page, register page, home page, camera page, and profile page. The user interface of the home, camera, and profile page is displayed in Figure 5. In the implementation, those pages are divided into 3 activities: main activity, login activity, and register activity, with the main activity handling 3 fragments: home, camera, and profile.

On the login page, users are required to enter their registered email and password, while on the registration page, users are asked to provide personal information, including their name, email, password, weight, height, and date of birth. The name, email, and password are used for authentication purposes, while the weight, height, and date of birth are used to determine the categorization of blood pressure values.

The homepage serves the primary function of visualizing recorded blood pressure data within the application. On the homepage, there is a dropdown menu for selecting the user whose blood pressure data will be displayed. Meanwhile, on the profile page, users can view and modify their account details. Additionally, the profile page displays a list of other user profiles registered under the same account, allowing for user management operations such as adding, modifying, and deleting user data.

On the camera page, there is a camera view displaying live frames captured by the camera. This section also includes an overlay to draw bounding boxes surrounding the detected seven-segment digits. The detection results, such as inference time, systolic pressure, diastolic pressure, heart rate, and blood pressure category, are displayed below the camera view. After capturing an image, users can confirm the detection results or retake the picture. In addition to using the camera, users have the option to detect images from the gallery by pressing the gallery button.

V. Results and Discussion

A. Evaluation of Model Training

The evaluation metrics for the 3 variants of the digit detection model (YOLOv8) training results can be seen in Table IV. Based on these results, the training variation with a batch size of 8 was found to be the optimal configuration for each model variant. This is evidenced by the highest accuracy, precision, recall, and F1-score values for this configuration.

Among the three trained YOLOv8 variants, the medium variant achieved the highest accuracy, recall, and F1-score even though there is no significant difference in their metrics. The significant differences are observed only in the model size and inference time. The size and inference time of the YOLOv8 small variant is approximately half of the medium variant and one-fourth of the large variant. This is due to the simpler network structure of the YOLOv8 small variant compared to the YOLOv8 medium and YOLOv8 large variants.

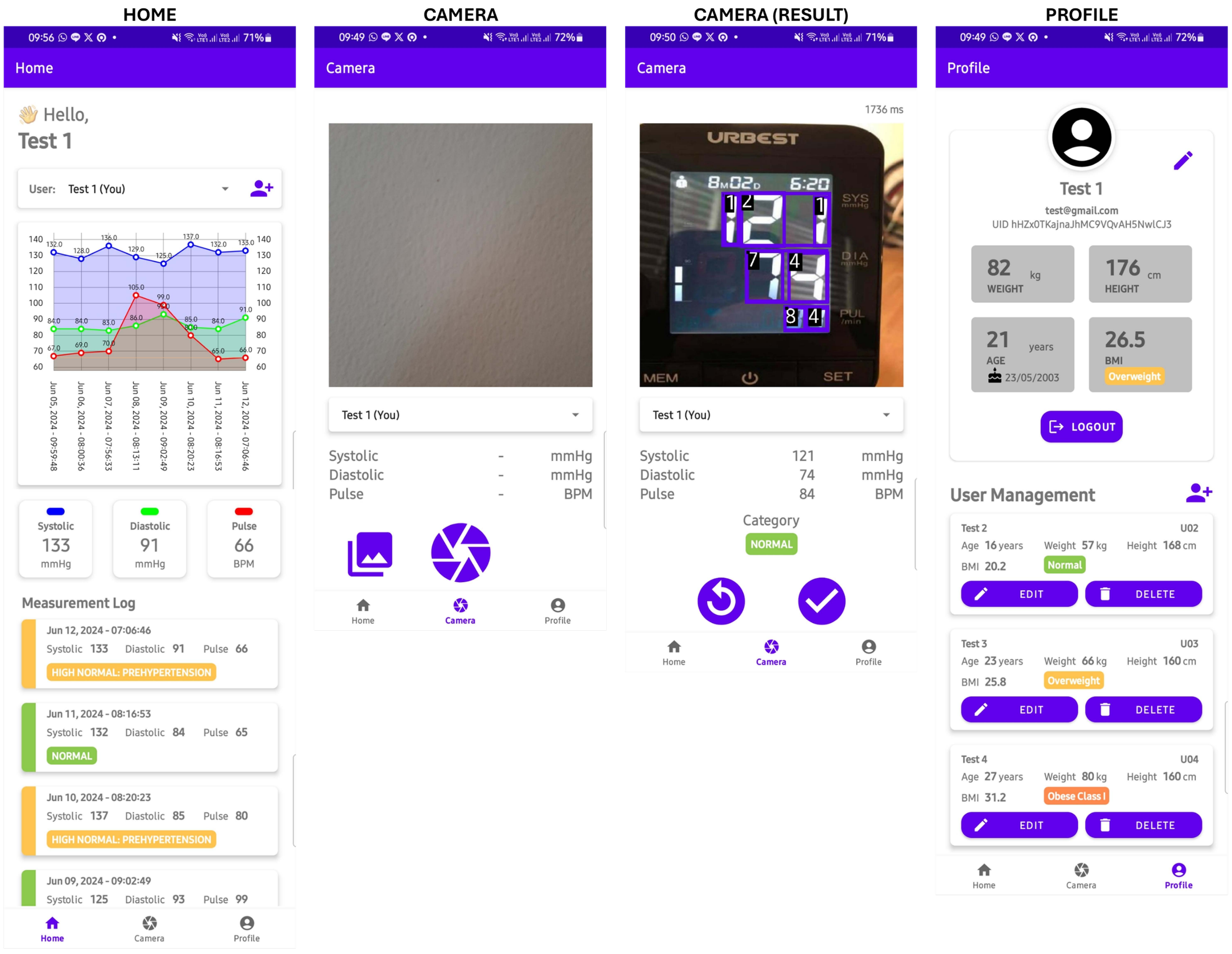
B. Evaluation of Model Pruning

Table V shows the comparison of evaluation metrics for the digit detection process among each variant of the YOLOv8 model and their pruned versions. For ease of reference, the original models will be referred to as ”base models”.

Based on these evaluation metrics, the pruned models did not experience a significant reduction in inference time, except for the large variant, which saw a decrease of about 500 ms (12.7%). However, all pruned models experienced a substantial decline in F1-score and precision, ranging from 6.7% to 12% and 11.7% to 20.5%, respectively. Additionally, the model size increased by up to 2 times (±99%) for each variant. This increase in size occurred because the base models (YOLOv8), as PyTorch models, are stored in 16-bit floating point precision by default. In contrast, the pruned models are stored in 32-bit floating point precision, resulting in a doubling of the model size.

Overall, the pruned model of YOLOv8l had the best accuracy, precision, recall, and F1-score metrics among the pruned variants. However, the base model of YOLOv8m still had the best accuracy, precision, recall, and F1-score when considering all models. Meanwhile, the small variant remained the model with the smallest size and inference time as it is the simplest model compared to other tested variants.

Based on these evaluation metrics, it can be concluded that pruning is not effective enough in compressing the YOLOv8 model because the pruned models have significantly worse performance compared to the base models, despite only being pruned by 10%. Therefore, the decision was made not to proceed with the pruned models to the next process, which is quantization. Instead, quantization will be applied directly to the base models.



Hello,
Test 1

User: Test 1 (You)

Systolic	Diastolic	Pulse
133	91	66
mmHg	mmHg	BPM

Measurement Log

Jun 12, 2024 - 07:06:46
Systolic 133 Diastolic 91 Pulse 66
HIGH NORMAL: PREHYPERTENSION

Jun 11, 2024 - 08:16:53
Systolic 132 Diastolic 84 Pulse 65
NORMAL

Jun 10, 2024 - 08:20:23
Systolic 137 Diastolic 85 Pulse 80
HIGH NORMAL: PREHYPERTENSION

Jun 09, 2024 - 09:02:49
Systolic 125 Diastolic 93 Pulse 99

Test 1 (You)

Systolic	-	mmHg
Diastolic	-	mmHg
Pulse	-	BPM

1736 ms

Test 1 (You)

Systolic	121	mmHg
Diastolic	74	mmHg
Pulse	84	BPM

Category
NORMAL

Test 1
test@gmail.com
UID hHZx0TKajnaJhMC9VQvAH5NwlCJ3

82 kg WEIGHT
176 cm HEIGHT
21 years AGE 23/05/2003
26.5 BMI Overweight

LOGOUT

User Management

Test 2 U02
Age 16 years Weight 57 kg Height 168 cm
BMI 20.2 Normal
EDIT DELETE

Test 3 U03
Age 23 years Weight 66 kg Height 160 cm
BMI 25.8 Overweight
EDIT DELETE

Test 4 U04
Age 27 years Weight 80 kg Height 160 cm
BMI 31.2 Obese Class I
EDIT DELETE

Fig. 5. The UI of the m-health application.

TABLE IV
EVALUATION METRICS FOR TRAINING RESULTS ON THREE YOLOV8 MODEL VARIANTS

Metrics	YOLOv8s			YOLOv8m			YOLOv8l		
	8 batch	16 batch	32 batch	8 batch	16 batch	32 batch	8 batch	16 batch	32 batch
Accuracy (%)	99.70	99.64	99.62	**99.73**	99.66	99.69	99.70	99.71	99.65
Precision (%)	97.77	97.85	97.94	98.14	97.45	97.89	**98.19**	98.16	97.87
Recall (%)	98.40	98.01	97.84	**98.58**	98.13	98.40	98.22	98.13	97.98
F1-Score (%)	98.08	97.93	97.89	**98.36**	97.79	98.14	98.21	98.06	97.83
Size (MB)	21.51	**21.50**	21.51	49.65	49.65	49.65	83.61	83.61	83.61
Inference (ms)	974.50	1032.80	**722.10**	1996.60	1762.20	1740.60	4299.50	3408.20	3411.60

C. Evaluation of Model Quantization

The evaluation metrics for the quantization process of the digit detection model can be seen in Table VI. Based on the results of 16-bit quantization, there were no significant changes in accuracy, precision, recall, F1-score, and model size. In fact, the changes in these five metrics were almost negligible, approaching 0%. However, there was a significant reduction in inference time, particularly for the small model variant, which saw a decrease of 36.6%. The model size remained relatively unchanged because, as stated before, the base model is a PyTorch model, so they are already stored in 16-bit floating point precision. Overall, the FP16 version of the YOLOv8s variant proved to have the smallest size and the fastest inference speed.

In contrast, with 8-bit quantization, there was a significant reduction in both model size and inference time. The model size was successfully compressed to half of the base model. The INT8 version of the YOLOv8s variant had the smallest size among others, at only 11 MB. Generally, there was a slight

TABLE V
EVALUATION METRICS FOR PRUNING RESULTS ON THREE YOLOV8 MODEL VARIANTS

Metrics	YOLOv8s			YOLOv8m			YOLOv8l		
	Base	Pruned	Delta (%)	Base	Pruned	Delta (%)	Base	Pruned	Delta (%)
Accuracy (%)	99.70	96.67	-3.00	99.73	97.43	-2.30	99.70	**97.77**	-1.90
Precision (%)	97.77	77.71	-20.50	98.14	83.93	-14.50	98.19	**86.71**	-11.70
Recall (%)	98.40	96.88	-1.60	98.58	96.31	-2.30	98.22	**97.20**	-1.00
F1-Score (%)	98.08	86.24	-12.00	98.36	89.69	-8.80	98.21	**91.65**	-6.70
Size (MB)	21.51	**42.84**	+99.20	49.65	99.10	+99.10	83.61	167.01	+99.80
Inference (ms)	974.50	**938.90**	-3.70	1996.60	1915.70	-4.00	4299.50	3753.30	-12.70

decrease in precision metrics for models quantized to INT8 precision, ranging from 2.5% to 4%. The YOLOv8 medium (YOLOv8m) variant became the best-performing INT8 model in terms of accuracy, precision, and F1-score, although the differences compared to other INT8 models were not substantial.

So, based on the results, it can be concluded that quantization is quite effective in compressing the model. This is demonstrated by the fact that quantization generally reduces model size and inference time without significant drops in accuracy, precision, recall, and f1-score. When considering all quantized variants, the FP16 version of the YOLOv8 medium (YOLOv8m) variant had the best accuracy, precision, recall, and f1-score. However, the differences in these four metrics were not significant compared to other quantized models. Since the model will be deployed in a mobile (m-health) application, model size and inference time are the primary considerations in selecting the final model. Based on the data, the INT8 version of the YOLOv8 small (YOLOv8s) variant had the smallest size compared to other quantized models. Additionally, its inference time was not significantly different from the FP16 version of the YOLOv8s variant, which had the fastest inference time. Therefore, considering these factors, the INT8 quantized version of the YOLOv8 small (YOLOv8s) variant was selected as the model to be used in the m-health application.

A comparison of the proposed seven-segment digit detection model with related studies can be seen in Table VII. Generally, the developed model has successfully detected seven-segment digits accurately. The model presented in this Final Project achieved higher accuracy and F1-score compared to related research. Additionally, it can detect blood pressure measurement images that include shadow reflections and illumination effects by leveraging data variability and augmentation techniques. This demonstrates that the developed model has effectively addressed challenges faced by Finnegan *et al*. and Shenoy & Aalami. Furthermore, the model has been compressed to INT8 precision, resulting in a smaller model size and inference time. Therefore, this model can run smoothly with low latency on edge devices such as mobile phones. Consequently, there is no need to deploy the model on servers like the implementation by Wannachai *et al*. in their research.

D. Evaluation of M-Health Application

The outcomes of the model's detection process include bounding boxes for each defined class. Hence, further processing within the m-health application is required to categorize and identify the detected digits as blood pressure values: systolic pressure, diastolic pressure, and heart rate per minute. Subsequently, an evaluation of these processes is also necessary to assess their effectiveness and accuracy.

To verify its performance, the model was tested using readings from 40 data samples of blood pressure measurements. These samples included 30 images from a test dataset and 10 images directly captured from a digital sphygmomanometer. The inclusion of 10 directly captured images was aimed to validate that the model did not overfit. Based on the test results, the accuracy of digit grouping was found to be 96.67%, indicating only 4 out of 120 blood pressure metrics were incorrectly identified. Overall, the model successfully interpreted 38 sphygmomanometer images out of 40, achieving an image reading accuracy of 95%.

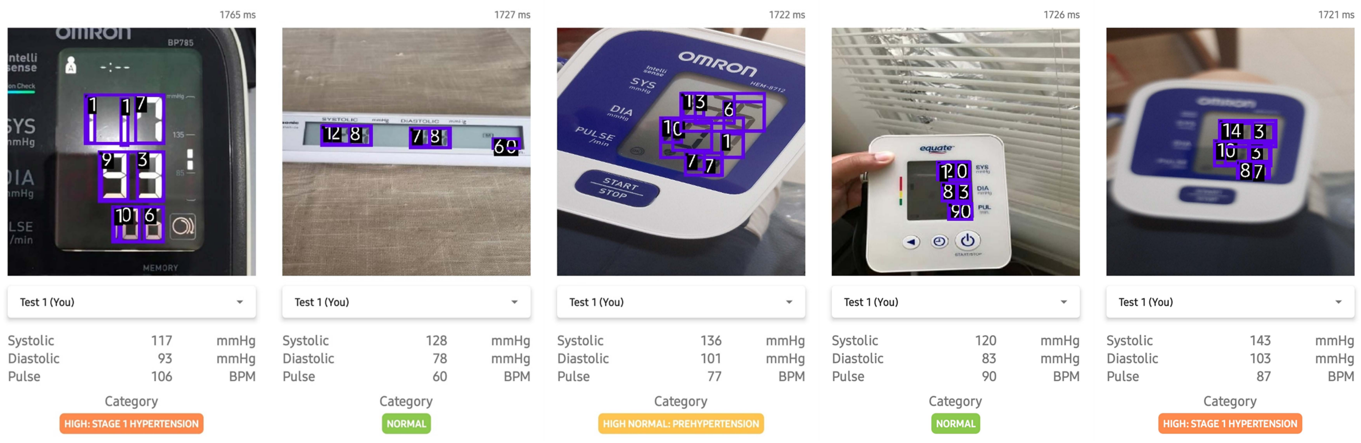
Meanwhile, the average inference time obtained was 1867.6 ms, a threefold increase than the measurement in Google Colab (CPU). This increase is attributed to differences in device specifications between Google Colab as the model development platform and the mobile phone as the m-health deployment platform. Mobile phones typically have limited computational capabilities compared to the hardware of Google Colab, thus resulting in longer inference times for model execution. The hardware specifications of Google Colab and the mobile phone used are detailed in Section IV. Additionally, the m-health application's model is also tested on 5 general cases, including LCD color variations, sphygmomanometer value layout variations, distance and angle of image capture, as well as image disturbances such as blur and light reflections. A sample of the test results is shown in Figure 6.

Based on the results in the table, the model successfully detected blood pressure values across these defined cases with satisfactory performance. For cases 1 and 2, the model accurately recognized blood pressure values across various digital sphygmomanometer layouts. In cases 4 and 5, the model also successfully detected images taken from a distance and slightly blurred, as long as they remained within reasonable limits for manual readability on normal people. The model only failed to detect digits in one image from the test case involving an extreme angle of image capture, where the angle was too steep for the model to accurately recognize the seven-segment digits.

Overall, the m-health application has proven capable of accurately reading blood pressure measurement results. However, during live camera readings, users still need to keep the

TABLE VI
EVALUATION METRICS FOR QUANTIZATION RESULTS ON THREE YOLOV8 MODEL VARIANTS

Metrics	YOLOv8s					YOLOv8m					YOLOv8l				
	Base	FP16	Delta (%)	INT8	Delta (%)	Base	FP16	Delta (%)	INT8	Delta (%)	Base	FP16	Delta (%)	INT8	Delta (%)
Accuracy (%)	99.70	99.70	0.00	99.28	-0.40	99.73	**99.74**	+0.01	99.47	-0.30	99.70	99.70	0.00	99.27	-0.40
Precision (%)	97.77	97.77	0.00	94.89	-2.90	98.14	**98.21**	+0.07	96.69	-1.50	98.19	98.19	0.00	94.31	-4.00
Recall (%)	98.40	98.40	0.00	98.11	-0.30	98.58	**98.65**	+0.07	98.55	-0.04	98.22	98.22	0.00	98.60	+0.40
F1-Score (%)	98.08	98.09	+0.01	96.48	-1.60	98.36	**98.43**	+0.07	97.61	-0.80	98.21	98.21	0.00	96.41	-1.80
Size (MB)	21.51	21.40	-0.50	**11.00**	-48.90	49.65	49.49	-0.30	25.22	-49.20	83.61	83.40	-0.30	42.36	-49.30
Inference (ms)	974.50	**617.90**	-36.60	661.40	-32.10	1996.60	1866.40	-6.50	1926.50	-3.50	4299.50	3794.40	-11.80	3918.70	-8.90



Test 1 (You)
Systolic 117 mmHg
Diastolic 93 mmHg
Pulse 106 BPM
Category
HIGH: STAGE 1 HYPERTENSION

Test 1 (You)
Systolic 128 mmHg
Diastolic 78 mmHg
Pulse 60 BPM
Category
NORMAL

Test 1 (You)
Systolic 136 mmHg
Diastolic 101 mmHg
Pulse 77 BPM
Category
HIGH NORMAL: PREHYPERTENSION

Test 1 (You)
Systolic 120 mmHg
Diastolic 83 mmHg
Pulse 90 BPM
Category
NORMAL

Test 1 (You)
Systolic 143 mmHg
Diastolic 103 mmHg
Pulse 87 BPM
Category
HIGH: STAGE 1 HYPERTENSION

Fig. 6. Sample of m-health application's test results.

TABLE VII
MODEL PERFORMANCE COMPARISON WITH RELATED STUDIES

Research	Model/Algorithm	Accuracy (%)	F1-Score (%)
Finnegan *et al.* (2019)	Image Processing	93.00	80.00
Wannachai *et al.* (2020)	CNN	90.00	-
Shenoy & Aalami (2018)	Random Forest	98.20	-
Proposed Model	YOLOv8 small INT8	99.28	96.48

camera steady for approximately 1-2 seconds to ensure more accurate detection results. This is due to the model's inference time being slightly longer than the frame capture interval of the camera, causing a minor bottleneck in the generated frames.

VI. CONCLUSION

In this study, an m-health application equipped with the capability to automatically acquire blood pressure measurement results from a sphygmomanometer has been successfully designed, implemented, and tested with good accuracy. This capability was implemented through a deep learning model (YOLOv8s with INT8 precision) capable of detecting seven-segment digits. Overall, the developed m-health application features include blood pressure value reading via camera, blood pressure data logging, data visualization, and authentication. The resulting model has achieved a digit detection accuracy of 98.20%. The inference time for the model was measured at 661.4 ms on a Google Colab CPU and 1867.6 ms on a mobile phone, with a model size of 11 MB. Meanwhile, through testing with 40 image samples, the m-health application achieved a digit grouping accuracy of 96.67% and an image reading accuracy of 95%.

REFERENCES

[1] Man, P. K., Cheung, K. L., Sangsiri, N., Shek, W. J., Wong, K. L., Chin, J. W., ... & So, R. H. Y. (2022, October). Blood Pressure Measurement: From Cuff-Based to Contactless Monitoring. In *Healthcare* (Vol. 10, No. 10, p. 2113). MDPI.

[2] Marhaendra, Y. A., Basyar, E., & Adrianto, A. A. (2016). *Pengaruh letak tensimeter terhadap hasil pengukuran tekanan darah* (Doctoral dissertation, Diponegoro University).

[3] Rehman, S., Hashmi, M. F., & Nelson, V. L. (2022). Blood Pressure Measurement. In *StatPearls* [Internet]. StatPearls Publishing. Available from https://www.ncbi.nlm.nih.gov/books/NBK482189/

[4] Williams, B., Mancia, G., Spiering, W., Agabiti Rosei, E., Azizi, M., Burnier, M., ... & Desormais, I. (2018). 2018 ESC/ESH Guidelines for the management of arterial hypertension: The Task Force for the management of arterial hypertension of the European Society of Cardiology (ESC) and the European Society of Hypertension (ESH). *European heart journal*, 39(33), 3021-3104.

[5] Rana, J. S., Khan, S. S., Lloyd-Jones, D. M., & Sidney, S. (2021). Changes in mortality in top 10 causes of death from 2011 to 2018. *Journal of general internal medicine*, 36, 2517-2518.

[6] Kwee, V., & Widjojo, H. (2022). Understanding the Determinants of m-Health Adoption in Indonesia. *Jurnal Manajemen Teori dan Terapan*, 15(3).

[7] Finnegan, E., Villarroel, M., Velardo, C., & Tarassenko, L. (2019). Automated method for detecting and reading seven-segment digits from images of blood glucose metres and blood pressure monitors. *Journal of medical engineering & technology*, 43(6), 341-355.

[8] Shenoy, V. N., & Aalami, O. O. (2017). Utilizing smartphone-based machine learning in medical monitor data collection: seven segment digit recognition. In *AMIA Annual Symposium Proceedings* (Vol. 2017, p. 1564). American Medical Informatics Association.

[9] Wannachai, A., Boonyung, W., & Champrasert, P. (2020). Real-Time Seven Segment Display Detection and Recognition Online System Using CNN. In *Bio-inspired Information and Communication Technologies: 12th EAI International Conference, BICT 2020, Shanghai, China, July 7-8, 2020, Proceedings 12* (pp. 52-67). Springer International Publishing.

[10] Hjelm, M., & Andersson, E. (2022). Benchmarking Object Detection Algorithms for Optical Character Recognition of Odometer Mileage.

[11] Gushima, K., & Kashima, T. (2023, September). Automatic Generation of Seven-Segment Display Image for Machine-Learning-Based Digital Meter Reading. In *PHM Society Asia-Pacific Conference* (Vol. 4, No. 1).

[12] Setiyono, B., Amini, D. A., & Sulistyaningrum, D. R. (2021, March). Number plate recognition on vehicle using YOLO-Darknet. In *Journal of Physics: Conference Series* (Vol. 1821, No. 1, p. 012049). IOP Publishing.

[13] Boesch, G. (2024, June). *Object Detection in 2024: The Definitive Guide*. viso.ai. https://viso.ai/deep-learning/object-detection/

[14] Ultralytics. (2023). *YOLOv8.2: Unleashing Next-Gen AI Capabilities*. GitHub. https://github.com/ultralytics/ultralytics

[15] AH. (2024). *Esfigmo Computer Vision Project (Version 3)* [Data set]. Roboflow Universe. https://universe.roboflow.com/ah-kkl2g/esfigmo

[16] Ega, A. V. (2023). *Automated BP Device Digit Recognition Image Dataset (Version 4)* [Data set]. https://universe.roboflow.com/adindra-vickar-egaodhei/automated-bp-device-digit-recognition

Hansel Valentino Tanoto is currently pursuing the B.Eng. degree in informatics engineering at Bandung Institute of Technology (ITB), Indonesia. His research interests include machine learning, computer vision, and software engineering.

Rinaldi Munir received the B.Eng. degree in informatics engineering and the M.Sc. degree in digital image compression from the Bandung Institute of Technology (ITB), Indonesia, in 1992 and 1999, respectively, and the Ph.D. degree in image watermarking from the School of Electrical Engineering and Informatics, ITB, in 2010. In 1993, he started his academic career as a Lecturer with the Department of Informatics, ITB. He is currently an Associate Professor with the School of Electrical Engineering and Informatics, ITB, and the Informatics Research Group. His research interests include cryptography and steganography-related topics, digital image processing, fuzzy logic, and numerical computation.

Nur Ahmadi (Member, IEEE) received the B.Eng. degree in electrical engineering from Bandung Institute of Technology (ITB), Indonesia, in 2011 and M.Eng. degree in communication and integrated systems from Tokyo Institute of Technology, Japan, in 2013. He received his Ph.D. degree in Electrical and Electronic Engineering from Imperial College London, UK, in 2020. His Ph.D. research focused on signal processing and deep learning for intracortical brain-machine interfaces. He is now with the Center for Artificial Intelligence and School of Electrical Engineering and Informatics, Institut Teknologi Bandung. His current research interests include biomedical signal processing, artificial intelligence, machine learning, digital and embedded systems.