



# INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : [www.joiv.org/index.php/joiv](http://www.joiv.org/index.php/joiv)



## The Design of Convolutional Neural Networks Model for Classification of Ear Diseases on Android Mobile Devices

I Gede Pasek Suta Wijaya <sup>a</sup>, Heru Mulyana <sup>b</sup>, Hamsu Kadriyan <sup>c,\*</sup>, Didit Yudhanto <sup>c</sup>, Riska Yanu Fa'rifah <sup>d</sup>

<sup>a</sup> Department of Informatics Engineering, University of Mataram, 83125, Indonesia

<sup>b</sup> Department of Intelligent System, Universiti Teknologi Mara, Shah Alam, Selangor, Malaysia

<sup>c</sup> Department of ENT-HN, University of Mataram, Jl. Majapahit 62, Mataram, 83125, Indonesia

<sup>d</sup> Universitas Telkom, Bandung, Bandung, Indonesia

Corresponding author: \*hamsu@unram.ac.id

**Abstract**— An otorhinolaryngologist (ORL) or general practitioner diagnoses ear disease based on ear image information. However, general practitioners refer patients to ORL for chronic ear disease because the image of ear disease has high complexity, variety, and little difference between diseases. An artificial intelligence-based approach is needed to make it easier for doctors to diagnose ear diseases based on ear image information, such as the Convolutional Neural Network (CNN). This paper describes how CNN was designed to generate CNN models used to classify ear diseases. The model was developed using an ear image dataset from the practice of an ORL at the University of Mataram Teaching Hospital. This work aims to find the best CNN model for classifying ear diseases applicable to android mobile devices. Furthermore, the best CNN model is deployed for an Android-based application integrated with the Endoscope Ear Cleaning Tool Kit for registering patient ear images. The experimental results show 83% accuracy, 86% precision, 86% recall, and 4ms inference time. The application produces a System Usability Scale of 76.88% for testing, which shows it is easy to use. This achievement shows that the model can be developed and integrated into an ENT expert system. In the future, the ENT expert system can be operated by workers in community health centres/clinics to assist leading health them in diagnosing ENT diseases early.

**Keywords**— Artificial intelligence; convolutional neural network; ear disease; image classification; android.

Manuscript received 18 Apr. 2022; revised 22 Aug. 2022; accepted 8 Dec. 2022. Date of publication 31 Mar. 2023.  
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



### I. INTRODUCTION

The ear disease diagnosis is problematic because it can only be determined by Otorhinolaryngologists (ORL) or General practitioners and can only be done in hospital or primary health center facilities using proper Otorhinolaryngologists equipment. This condition limits people who only have access to secondary health services, especially during a covid-19 pandemic.

Nowadays, Indonesia's health service facility consists of primary (Hospital) and secondary (*Puskesmas*) health centers. Hospitals and *Puskesmas* are located across the country, even in a remote village (*Desa*). Commonly, the hospital has complete equipment for the specialist to run the daily health care. In contrast, the *Puskesmas* facility only has general practitioners and limited equipment for specialist job [1]. Additionally, general practitioners who diagnose patient ear disease under limited competencies can misdiagnose or undiagnosed illness. According to this reason, the support

system is required to assist the general practitioner in diagnosing ear disease and making better decisions.

This research develops an ANN-based system with convolutional feature extraction called Convolutional Neural Network (CNN). CNN is one of the Machine Learning (ML) methods applied to classify a digital image. In this case, the CNN model will be designed and trained using an ear disease image dataset. The CNN model of training results will consist of the built-in architecture and weights, the knowledge obtained from the learning outcomes [2]–[4]. The model will be embedded in an Android-based smartphone application as an engine for the classification process. Furthermore, integration between the app and the compact Endoscopic Ear Cleaning Kit is required to register ear images to the system.

The selection of an Android-based smartphone as a CNN model development platform for recognizing ear images is based on the popularity of Android in terms of its active users, which reaches more than 1 billion users from all over the world[5]–[7], and 124 million active internet users, and

smartphone users in Indonesia[8]. Due to its popularity and affordability, the ear disease classification system can assist general practitioners in diagnosing ear diseases.

A previous study was built to determine disease and provide solutions to prevent or treat diseases that attack tomato leaves through digital image identification using supervised classification [5], [9]–[16]. Tests were carried out with 200 samples of tomato leaf images, 160 images as training data, and 40 as test data. The test results show that the CNN method has an average accuracy percentage of 97.5%, a precision of 95.45%, a recall of 95%, and an error of 5%. Meanwhile, SVM produces an average accuracy of 95%, a precision of 90.83%, a recall of 90%, and an error of 10%. The test results show that CNN is a better classifier than SVM[9], [12].

The research used the CNN model that won the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC2014) [17] classification and localization task competition from 1000 different classes with 1.2 million datasets. This achievement proves that the Neural Network method is suitable for complex image classification tasks.

Facial recognition using the CNN method with the dataset used is The Extended Yale Face Database B, a facial photo dataset. The best results were obtained using the dropout process with recognition accuracy as high as 89.73%. Meanwhile, if the testing data is tested, the recognition accuracy results will be as high as 75.79%[18]. Ear and mastoid disease classification using 10,544 Otoendoscopy images was used to train nine public convolution-based deep neural networks and combine them (ensemble) to predict ear disease, which produces about 85% to 92% accuracy[15].

According to ORL disease classification using the forward chaining method, it produces an accuracy of 100%. The data from the expert system were obtained from ORL experts to establish the disease through the symptoms. However, the public did not understand some symptoms because they could not directly see or feel [19], [20].

In the ENT disease diagnosis system based on Android, the demand variable consists of two fuzzy sets, namely: DOWN and UP; the inventory variable consists of two fuzzy sets, namely: LITTLE, and A LOT, while the production variable consists of two fuzzy sets, namely: DECREASED and ADDED. The calculation of accuracy in this study is not separated based on each disease but accuracy for the whole disease and produces an accuracy value of 93.75% [7].

Based on previous studies, it can be seen that convolution neural networks can work well for image classification tasks, especially in recognizing many classes with high heterogeneity [2], [13], [21]–[24]. In international competitions, the CNN method succeeded in outperforming other Machine Learning methods in the case of image classification, which has high heterogeneity[17], [25]. Therefore, the author intends to classify ear disease images using the CNN method.

## II. MATERIALS AND METHOD

### A. Material and Tool

This study assembled machine learning models using a dataset of ear diseases from the Artificial Intelligence research group, Informatics Engineering Study Program, Mataram University. The dataset consists of 20 disease classes from 2032 Otoendoscopy images, consisting of 100 – 132 images

for each class. The images were provided by ORL practicing at the University of Mataram Teaching Hospital[26]. Distribution of datasets as in Table 1.

TABLE I  
DATASET DISTRIBUTION

| ID Number | Class                    | Data  |         |
|-----------|--------------------------|-------|---------|
|           |                          | Train | Testing |
| 1         | Aerotitis Barotrauma     | 80    | 20      |
| 2         | Cerumen                  | 80    | 20      |
| 3         | Corpus Alienum           | 80    | 20      |
| 4         | M Timpani normal         | 80    | 20      |
| 5         | Myringitis Bulosa        | 80    | 20      |
| 6         | Normal                   | 80    | 26      |
| 7         | OE Difusa                | 80    | 20      |
| 8         | OE Furunkulosa           | 80    | 20      |
| 9         | OMA Hiperemis            | 80    | 20      |
| 10        | OMA Oklusi Tuba          | 80    | 20      |
| 11        | OMA Perforasi            | 80    | 20      |
| 12        | OMA Resolusi             | 80    | 20      |
| 13        | OMA Supurasi             | 80    | 20      |
| 14        | OMedEfusi                | 80    | 20      |
| 15        | OMedKResolusi            | 80    | 20      |
| 16        | OMedKTipe Aman           | 80    | 20      |
| 17        | OMedKTipeBahaya          | 80    | 20      |
| 18        | Otomikosis               | 80    | 20      |
| 19        | PerforasiMembran Tympani | 80    | 20      |
| 20        | Tympanosklerotik         | 80    | 20      |

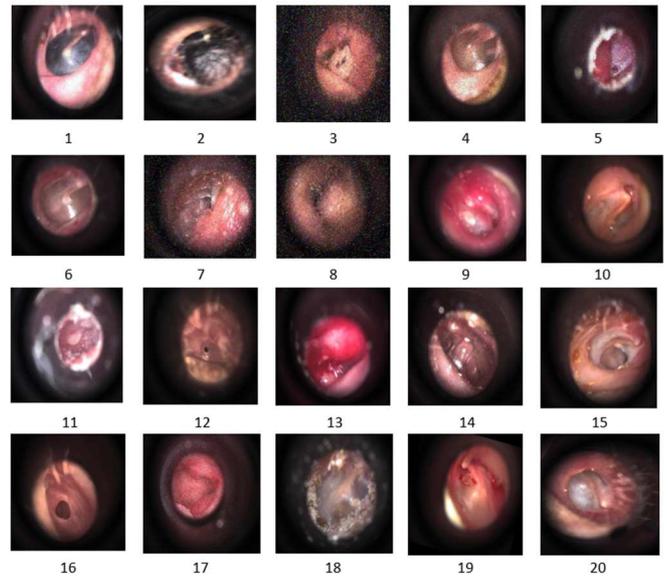


Fig. 1 The dataset class example corresponds to ID Number.

The tools used in this study process are divided into two parts, namely: hardware and software. The hardware used in this research is a computer with the following specifications in Table 2, and the software used in this study is given in Table 3.

TABLE II  
REQUIREMENT OF HARDWARE

| No. | Hardware   | Specification            |
|-----|------------|--------------------------|
| 1   | Processor  | AMD Ryzen 5 4600H        |
| 2   | GPU        | GeForce GTX 1650TI       |
| 3   | Smartphone | Android Redmi note 8 pro |

TABLE III  
SOFTWARE REQUIREMENTS

| No. | Software             | Specification                  |
|-----|----------------------|--------------------------------|
| 1   | Operating System     | Windows 10 64bit               |
| 2   | Programming Language | Python 3.8                     |
| 3   | Microsoft Office     | Office 2019                    |
| 4   | Text Editor          | Visual Studio Code, JupyterLab |
| 5   | IDE Android          | Android Studio 3.2             |

### B. Proposed Algorithm

The Ear Disease Images Classification design based on CNN for Android-Based Smartphone Devices has three main processes: training, testing, and deploying, shown in Fig. 3.

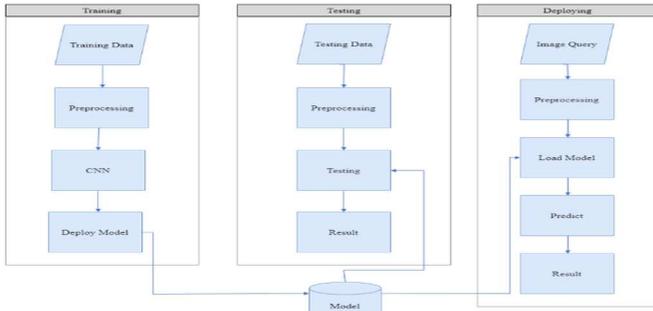


Fig. 2 The ear disease images classification algorithm design.

The three main processes are described as follows:

#### 1. Training Process

The training process includes the following stages:

- Image data in datasets will be processed in data filtering.
- Datasets will be resized to 96x96px size and then saved to be used as training data.
- The training data is entered into the neural network architecture, and the training/fitting process is carried out.
- After the training process is complete, the model formed will be saved.

#### 2. Testing Process

The testing process includes the following stages:

- The distribution of datasets that act as test data will not be used in the training process.
- The saved model will be loaded as a classifier that will be used to classify the test data.
- The overall prediction result is used for the value of the particular method.

#### 3. Deploying Process

The deploying process includes the following stages:

- The model of the training process will be embedded in the Android application.
- Integrated Endoscope Ear Cleaning Tool Kit will be used to acquire patient's ear imagery.
- The saved model will be loaded as an engine classifier that will be used to classify ear disease images.

*1. Pre-processing:* The steps taken before the dataset is used at the training stage are pre-processing, namely resizing to change the image size according to the variations in the image size used in the test, namely 96x96, 160x160, and 256x256 pixels with dimensions of 3 channels (RGB) and one channel (greyscale). Then the image is normalized to a value of 0~1 by dividing each pixel by 255. The pre-processing is

done to reduce the significantly high image variation of the image. The illustration of the pre-processing is presented in Fig. 3.

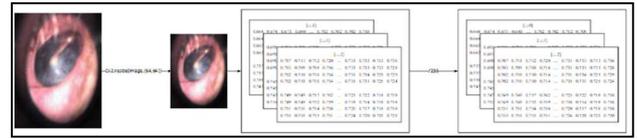


Fig. 3 Pre-processing process.

*2. CNN Architecture:* The first layer in CNN is convolution. All pixel values of the input image in a matrix will be directly inserted into the convolution layer. In this study, two variations of the number of convolutional layers were used, namely 2 and 3 layers with variations of the convolution filter 32.64, and 96. Meanwhile, two size variations were used for the kernel, namely 3x3 and 5x5 kernels. Zero padding is applied to the convolution layer to replace the missing image pixels with a value of 0. The output examples of the convolution layer is given in Fig. 4.

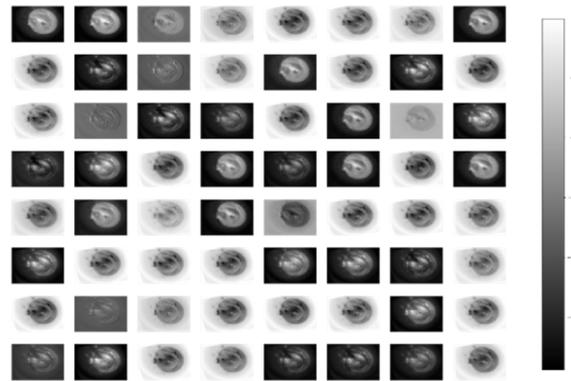


Fig. 4 Convolution layer result.

Next, the ReLU type activation function is applied to the convolution layer resulting in the first stage, which introduces non-linearity to the neural network. This activation function works by changing values less than 0 to 0 so that there are no negative pixel values (-), and values greater than 0 become the number itself. An example output of applying the ReLU activation function is presented in Fig. 5.

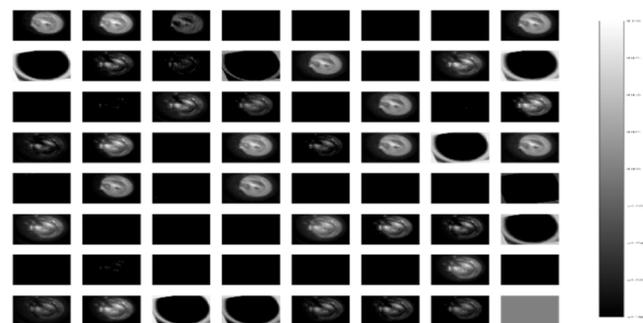


Fig. 5 ReLU activation function result.

The next stage is pooling. The type of pooling used is max pooling, which takes the largest value (max) with a pooling filter mask 2x2. The application of this technique causes the image size to decrease without significantly reducing the information, as shown in Fig. 6. This stage is useful for speeding up the computational process in training.

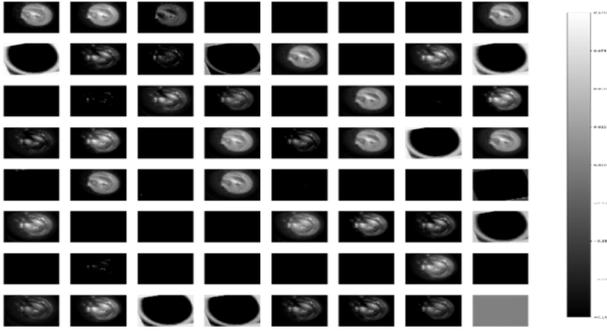


Fig. 6 Max pooling result.

After pooling, the image pixels will be transformed into vector (flattened) form, later input to fully-connected layer neurons. The fully-connected layer is the layer that is responsible for classification in CNN. In this study, the fully connected layer consisted of 64 neurons. The fully connected layer in this study also applies the Dropout technique, which means that not all the neurons involved will be used in the training process. In this case, a dropout of 20% is implemented, which means only 80% of neurons are used in the training phase. This function reduces the overfitting of the CNN model (lower test accuracy than training) by considering the very high complexity of the data.

### III. RESULT AND DISCUSSION

#### A. Testing Method

Tests were carried out to know the performance of the CNN model to classify Ear images that correspond to ENT disease. The model's performance evaluation was performed using accuracy, precision, recall, and computational time. At the same time, the android mobile application was evaluated by using the System Usability Scale (SUS). The accuracy, precision, and recall were determined based on the confusion matrix [25], [27], [28], commonly used to evaluate the CNN model as a classifier. The confusion matrix used in this research was 20x20 (see Table 4), consisting of the actual and predicted classes representing each disease class.

TABLE IV  
CONFUSION MATRIX [29]

|              |           | Predicted class |               |           |                |
|--------------|-----------|-----------------|---------------|-----------|----------------|
|              |           | Class 1         | Class 2       | Class ... | Class 20       |
| Actual class | Class 1   | True Class 1    | False Class 2 | ...       | False Class 20 |
|              | Class 2   | False Class 1   | True Class 2  | ...       | False Class 20 |
|              | Class ... | ...             | ...           | ...       | ...            |
|              | Class 20  | ...             | ...           | ...       | True Class 20  |

1) *Accuracy*: Accuracy is defined as the proportion of correct predictions (both true positive and true negative) among the total number of cases examined. The accuracy is commonly determined by Equation (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

where TP = True positive; FP = False positive; TN = True negative; FN = False negative

2) *Precision*: Precision is the level of accuracy between the information requested by the user and the answer provided by the system. For example, if the CNN model classifies an image of a healthy patient's ear as an image of an ENT diseased ear, then this result is very detrimental to the user. This case means that the precision value of the CNN model is low [30]. The precision can be determined using Equation (2)

$$Precision = \frac{TP}{TP+FP} = \frac{True\ Positive}{Total\ Predicted\ Positive} \quad \square \square \square$$

3) *Recall*: A recall is the system's success rate in recovering information. The recall is useful for calculating how many actual positives the model generates. The recall is used when the impact of false-negative predictions is high, where the sick patient image is predicted to be the healthy patient image. Recall calculation can be done using Equation (3).

$$Recall = \frac{TP}{TP+FN} = \frac{True\ Positive}{Total\ Actual\ Positive} \quad (3)$$

4) *Computation Time*: This method aims to measure the time required by the model to classify an image, starting from the image entered into the application until the classification results live obtained. In this case, the computational time in question is the computational time of the android application tested on the Samsung Galaxy Android mobile device.

5) *System Usability Scale (SUS)*: This test aims to evaluate the quality of Android applications by users. The evaluation was carried out using a questionnaire, in which the test results were defined in terms of numerical values [31]. Testing is done by asking respondents to try/run the application and then fill out a questionnaire. Respondents were asked to answer by giving a checkmark according to the data in Table 5.

TABLE V  
SUS SCALE

| MOS | Description       | Weight |
|-----|-------------------|--------|
| SA  | Strongly Agree    | 5      |
| A   | Agree             | 4      |
| MA  | Moderately Agree  | 3      |
| D   | Disagree          | 2      |
| SD  | Strongly Disagree | 1      |

The SUS can be calculated by using Equation (4).

$$SUS = (x + y) * 2.5 \quad (4)$$

Where  $x$  = Sum of the points for all odd-numbered questions – 5 and  $y$  = 25 – Sum of the points for all even-numbered questions.

#### B. Experiment Mechanism

The following parameters were tested on the model to find the CNN model with the best accuracy, recall, and precision.

- Convolutional kernel size: 3x3 kernel and 5x5
- Neural Network Convolution Layers: 2, and 3
- Convolutional filters size: 32, 64, and 128
- Input image sizes: 96x96, 160x160, and 256x256px
- Image dimensions (RGB and Grayscale)

The initial parameters used for this test are two convolution layers, a 3x3 kernel with 32 convolutional filters, 96x96 image size, and RGB input image, Trained within 20 Epoch. These parameters are determined based on the smallest value proposed.

### C. Experimental Result

In the convolution layer, the convolution filter shifts incrementally over all the image pixels taking information from the image pixels. A convolution filter (kernel) is an odd-order matrix used to obtain features from the input image. Therefore, the effect of the convolution filter size (kernel) needed to be evaluated to get the best kernel size used in the next test. The results of testing the effect of using 3x3 and 5x5 kernels on CNN's performance can be seen in Table 6. According to Table 6, the kernel size of 3x3 has the best performance. More fantastic performance could occur because it extracted less noisy detailed data fragments. Thus the 3x3 kernel size will be used for the subsequent experiments.

TABLE VI  
KERNEL SIZE TESTING PERFORMANCE

| Kernel size | Accuracy (%) | Precision (%) | Recall (%) | Computation Time (ms) |
|-------------|--------------|---------------|------------|-----------------------|
| 3x3         | 82           | 83            | 82         | 5                     |
| 5x5         | 81           | 82            | 81         | 5                     |

Multiple implementations of convolution operations in a CNN architecture will produce different performances depending on the image case being handled. Therefore, the second experiment was conducted to know how many convolution layers were applied. This study used two variations of convolution layers as test parameters, with the results as shown in Table 7. Based on the test results in Table 7, it was found that a CNN model with two convolution layer parameters gave the best performance with two convolution layer parameters. A higher number of layers may result in more noise or data variance, which is not essential to generalize the model. Thus, the kernel size of 3x3 and two convolution layers will be used as parameters for the next test.

TABLE VII  
CONVOLUTION LAYERS

| CL Number | Accuracy (%) | Precision (%) | Recall (%) | Computation Time (ms) |
|-----------|--------------|---------------|------------|-----------------------|
| 2         | 82           | 83            | 82         | 4                     |
| 3         | 80           | 82            | 80         | 5                     |

TABLE VIII  
TESTING THE PERFORMANCE OF CONVOLUTIONAL FILTERS

| Filters Size | Accuracy (%) | Precision (%) | Recall (%) | Computation Time (ms) |
|--------------|--------------|---------------|------------|-----------------------|
| 32           | 82           | 83            | 82         | 4                     |
| 64           | 83           | 85            | 84         | 4                     |
| 128          | 82           | 82            | 82         | 6                     |

Similar to the two previous tests, the third test was conducted to determine the number of filters used in the CNN model. In the third test, three filters were evaluated, namely 32, 64, and 128 filters. Based on the test results presented in Table 8, it was found that the CNN model with 64 filters had better performance than the other two filters. Larger filter dimensions can trigger overfitting by unnecessarily allowing the model to study data or noise. On the other hand, low filters cannot extract essential features from the data. Thus, the kernel size of 3x3, two convolution layers, and 64 filters will be used as parameters of CNN for the next test.

The input image size in the neural network architecture is a significant factor in influencing the model's performance. The smaller the image size is, the less information exists in the image itself. On the other hand, the larger the image size is, the more expensive the computational costs, and too much image information tends to cause the overfitting of the model. Therefore, the fourth test evaluated three variations of image sizes, namely 96x96, 160x160, and 256x256 pixels. Based on the test results presented in Table 9, it is known that the performance for the input image sizes 96x96 and 160x160 has minimal differences. However, based on the confusion matrix review, any image taller than 160x160 results in much lower performance. So it can be concluded that the best input image size parameter is 160x160 pixels with a relatively short computation time and can still be handled by mobile devices.

The effect of color on the neural network model performance depends on the complexity of the class to be classified. To avoid complexity and computational time, a grayscale image is recommended if color is not essential in classification. In this fifth test, both grayscale and RGB images were tested. As presented in Table 9, the test results show that the CNN model with the input RGB image performs better than a grayscale image. These results indicate that color is essential information distinguishing ear disease based on the middle-outer image.

TABLE IX.  
TEST PERFORMANCE OF THE INPUT IMAGE SIZE

| Image Size | Accuracy (%) | Precision (%) | Recall (%) | Computation Time (ms) |
|------------|--------------|---------------|------------|-----------------------|
| 96x96      | 83           | 85            | 84         | 4                     |
| 160x160    | 84           | 86            | 86         | 4                     |
| 256x256    | 76           | 77            | 77         | 6                     |

Based on the experimental results above, it can be seen that the CNN with the best parameters is a model with a kernel size of 3x3, two convolution layers, 64 convolution filters, 160x160 pixel input image size, and RGB input image. The best CNN model has provided accuracy, precision, and recall of about 84%, 86%, and 86%, respectively, with a fast classification time of 4 ms. The same precision and recall results indicate that the model performs exceptionally well in recognizing the value of the ear disease class while reducing the classified false class.

Based on the confusion matrix in Fig. 7, the CNN model has difficulty classifying several classes, for example, class 5 (Normal), which only produces a test accuracy of 58%, or class 16 (OMedKTypeBahaya), with an accuracy of about 55%. These results indicate that the model generalizes these classes due to the lack of variation in class data.

TABLE X  
TEST RESULTS OF IMAGE DIMENSIONS

| Image Dimension | Accuracy (%) | Precision (%) | Recall (%) | Computation Time (ms) |
|-----------------|--------------|---------------|------------|-----------------------|
| RGB             | 84           | 86            | 86         | 4                     |
| Grayscale       | 69           | 71            | 69         | 4                     |

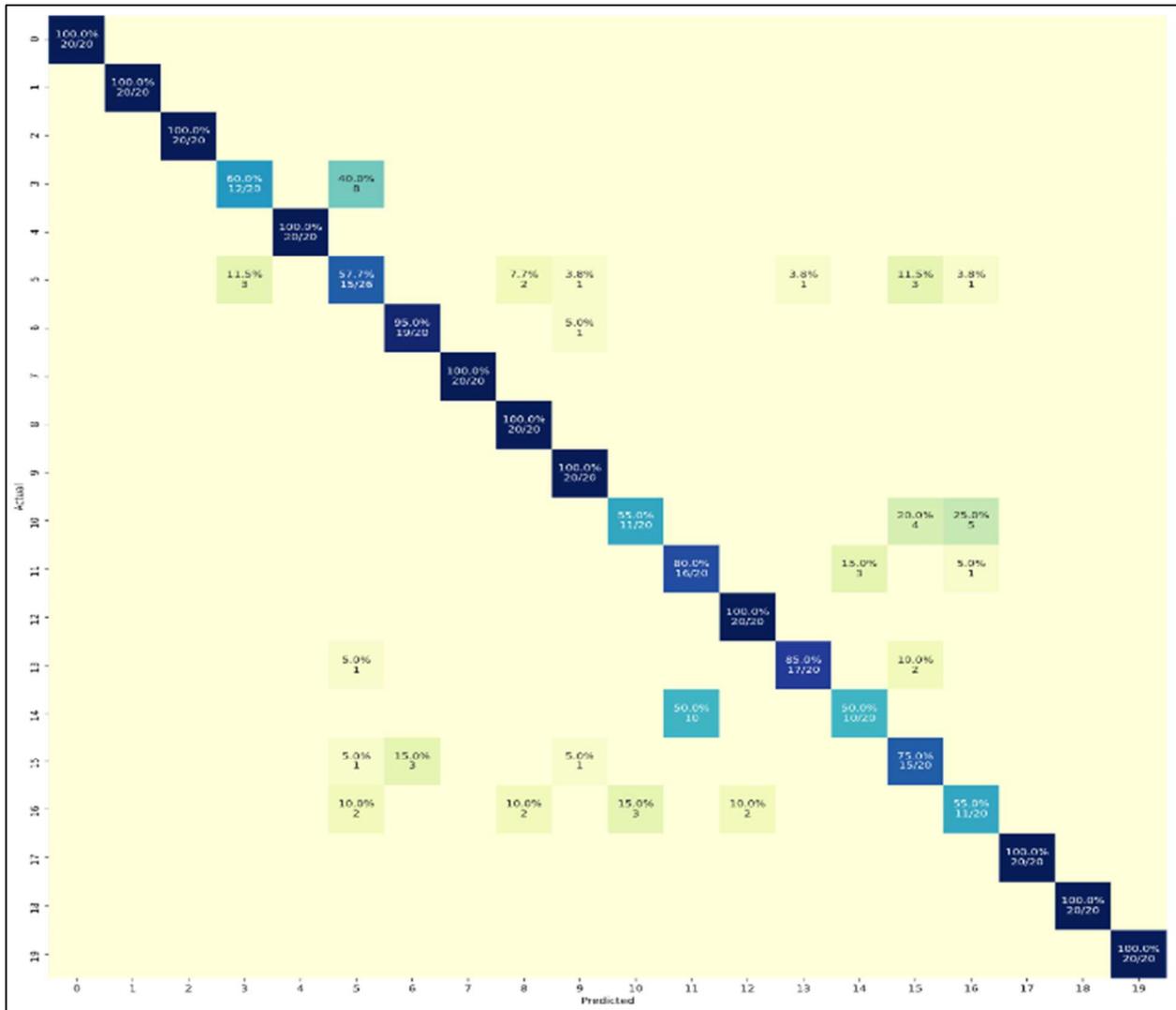


Fig. 7 Confusion matrix.

#### D. Deploying and Evaluation

At this stage, an Android-based application is developed, embedded with the model with the best performance results in the previous testing process. The application use case is presented in Fig. 8, and the main interface of the application is shown in Fig. 9. Fig. 8 shows some features as follows:

1. Diagnose by selecting a source of the image query.
2. Correct the diagnosis result to the user's desire.
3. Upload corrected diagnosis result data to the server.
4. Change the Classification engine or revert it to default.

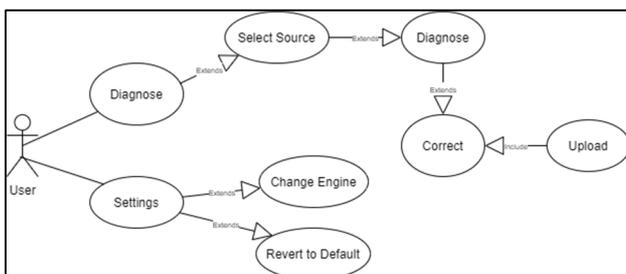


Fig. 8 Application Use case

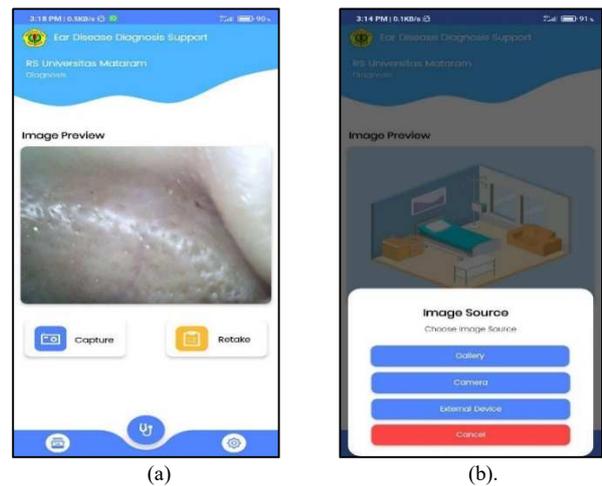


Fig. 9 Main menu and scanning process screen

In the diagnosis menu display (see Fig. 9(b)), the user can determine whether the source of the image query is from the gallery, the smartphone's internal camera, or an external device, namely the Endoscope Ear Cleaning Tool Kit.

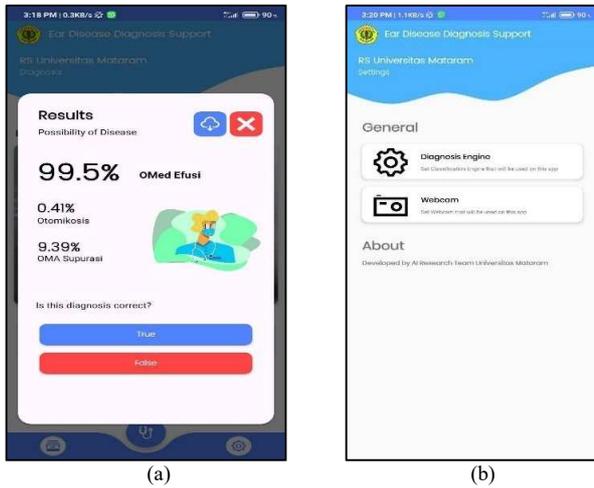


Fig. 10 Settings screen and real-time protection showcase.

After the user registers the image query and presses the diagnosis button, the top three results of the disease will appear, as shown in Fig. 10(a). In this case, the user can correct the classification results and upload the data to the server as additional data for further training. The user can update the classification engine with a new one in the settings menu (see Fig. 10(b)).

#### E. System Usability Scale (SUS)

This test was conducted to determine the feasibility of the application of ear disease classification. The test used SUS with ten questions to determine the response to the quality of the application by eight respondents consisting of medical students and ENT doctors. The ten questions of SUS and the response of the respondents presented in Table 11 and Table 12, respectively.

TABLE XI  
SUS QUESTIONS

| No  | Questions  |
|-----|--|
| Q1  | I suppose I will operate this system again.                                |
| Q2  | I discover this system complicated to use.                                 |
| Q3  | I discover this system easy to use.  |
| Q4  | I require help from someone else or a technician in operating this system. |
| Q5  | I feel the components of this system are operating properly.               |
| Q6  | I feel many items are inconsistent (incompatible with the system)          |
| Q7  | I feel others will comprehend how to operate this system quickly.          |
| Q8  | I discover this system confusing.  |
| Q9  | I feel there are no obstructions in operating this system.                 |
| Q10 | I require to get used to it first before operating this system.            |

TABLE XII  
SUS RESULT

| No | Question | NA | LA | AE | A | HA |
|----|----------|----|----|----|---|----|
| 1  | Q1       | 0  | 0  | 0  | 3 | 5  |
| 2  | Q2       | 1  | 7  | 0  | 0 | 0  |
| 3  | Q3       | 0  | 0  | 0  | 1 | 7  |
| 4  | Q4       | 1  | 1  | 3  | 1 | 2  |
| 5  | Q5       | 0  | 0  | 0  | 8 | 0  |
| 6  | Q6       | 1  | 2  | 1  | 3 | 1  |
| 7  | Q7       | 0  | 0  | 0  | 2 | 6  |
| 8  | Q8       | 4  | 2  | 2  | 0 | 0  |
| 9  | Q9       | 0  | 0  | 0  | 2 | 6  |
| 10 | Q10      | 2  | 3  | 2  | 1 | 0  |

Based on the results presented in Table 12, the average SUS score determined using Eq. (4) produces 76.88%. According to the Grade ranking of SUS score, 78.68% indicates that the application was running well and relatively easy to use.

#### IV. CONCLUSION

The highest performance of the CNN model can be achieved using a 3x3 convolutional kernel, two convolution layers, 64 convolution filters, 160x160 px input image size, and RGB input image, with 84% accuracy, 86% precision, and 86% recall. The classification time required by this best CNN model is 4 ms. The android application from the CNN data model has been running well, as shown by the SUS score of 76.88%, which indicates that the application is relatively easy to use. In order to improve the performance of the proposed model, the proposed scheme will be retrained using more disease data obtained from the application itself when carrying out diagnoses in patients annotated by otorhinolaryngologists.

#### REFERENCES

- [1] Y. Mahendradhata *et al.*, "The Republic of Indonesia Health System Review," *Health Syst. Transit.*, vol. 7, no. 1, p. 1, 2017.
- [2] I. W. A. Arimbawa, I. G. P. S. Wijaya, and I. Bintang, "Comparison of simple and stratified random sampling on porn videos recognition using CNN," in *2019 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, Nov. 2019, pp. 1–6, doi: 10.1109/CENIM48368.2019.8973305.
- [3] B. Van Ginneken, B. M. T. H. Romeny, and M. A. Viergever, "Computer-aided diagnosis in chest radiography: a survey," *IEEE Trans. Med. Imaging*, vol. 20, no. 12, pp. 1228–1241, 2001, doi: 10.1109/42.974918.
- [4] B. Alsaaidah, M. R. Al-Hadidi, H. Al-Nsour, R. Masadeh, and N. AlZubi, "Comprehensive Survey of Machine Learning Systems for COVID-19 Detection," *J. Imaging*, vol. 8, no. 10, 2022, doi: 10.3390/jimaging8100267.
- [5] T. N. Turnip, P. O. Manik, J. H. Tampubolon, and P. A. P. Siahaan, "Klasifikasi Aplikasi Android menggunakan Algoritme K-Means dan Convolutional Neural Network berdasarkan Permission," *J. Teknol. Inf. dan Ilmu Komputer; Vol 7, No 2 April 2020DO - 10.25126/jtiik.2020702641*, Feb. 2020.
- [6] Asosiasi Penyelenggara Jasa Internet (APJII), "Hasil Survei Penetrasi dan Perilaku Pengguna Internet Indonesia 2018," 2018.
- [7] F. Ekajaya, N. Hidayat, and M. T. Ananta, "Diagnosis Penyakit THT Menggunakan Metode Fuzzy Tsukamoto Berbasis Android," *J. Pengemb. Teknol. Inf. dan Ilmu Komputer; Vol 2 No 8*, 2017.
- [8] T. Verge, "Google announces over 2 billion monthly active devices on Android," 2017.
- [9] T. Shafira, *Implementasi Convolution Neural Network untuk Klasifikasi Citra Tomat Menggunakan Keras*. Yogyakarta: Universitas Islam Indonesia, 2018.
- [10] M. Pahar, M. Klopffer, R. Warren, and T. Niesler, "COVID-19 Cough Classification using Machine Learning and Global Smartphone Recordings," Dec. 2020, doi: 10.1016/j.combiomed.2021.104572.
- [11] J. Laguarta, F. Hueto, and B. Subirana, "COVID-19 Artificial Intelligence Diagnosis Using only Cough Recordings," *IEEE Open J. Eng. Med. Biol.*, vol. 1, pp. 275–281, 2020, doi: 10.1109/OJEMB.2020.3026928.
- [12] Jimmy Pujoseno, *Impelemntasi Deep Learning Menggunakan Convolution Neural Network untuk Klasifikasi Alat Tulis*. Yogyakarta: Universitas Islam Indonesia, 2018.
- [13] R. M. Pereira, D. Bertolini, L. O. Teixeira, C. N. Silla, and Y. M. G. Costa, "COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios," *Comput. Methods Programs Biomed.*, vol. 194, 2020, doi: 10.1016/j.cmpb.2020.105532.
- [14] M. F. Aslan, M. F. Unlersen, K. Sabanci, and A. Durdu, "CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection," *Appl. Soft Comput.*, vol. 98, p. 106912, 2021, doi: https://doi.org/10.1016/j.asoc.2020.106912.
- [15] D. Cha, C. Pae, S.-B. Seong, J. Y. Choi, and H.-J. Park, "Automated diagnosis of ear disease using ensemble deep learning with a big otoendoscopy image database," *EBioMedicine*, vol. 45, pp. 606–614,

- 2019, doi: <https://doi.org/10.1016/j.ebiom.2019.06.050>.
- [16] X. Zeng *et al.*, "Efficient and accurate identification of ear diseases using an ensemble deep learning model," *Sci. Rep.*, vol. 11, no. 1, p. 10839, 2021, doi: 10.1038/s41598-021-90345-w.
- [17] Imagenet, "Large Scale Visual Recognition Challenge (ILSVRC)," *ImageNet Large Scale Vis. Recognit. Chall.*, 2015.
- [18] M. P. Harini, "Pengenalan Pola Wajah Manusia menggunakan Transformasi Wavelet dan LDA (Linear Discriminant Analysis)," Institut Teknologi Telkom, 2007.
- [19] W. Verina, "Penerapan Metode Forward Chaining untuk Mendeteksi Penyakit THT," *J. Tek. Inform. Dan Sist. Inf.*, vol. 1, no. 2, pp. 123–138, 2015.
- [20] L. Lisnawita, L. L. Van FC, and E. Lianda, "Sistem Pakar Diagnosa Penyakit THT," *INOVTEK Polbeng - Seri Inform.*, vol. 1, no. 2, p. 95, 2016, doi: 10.35314/isi.v1i2.120.
- [21] C. Ding, T. Bao, S. Karmoshi, and M. Zhu, "Low-resolution face recognition via convolutional neural network," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, 2017, pp. 1157–1161.
- [22] O. Surinta and T. Khamket, "Recognizing pornographic images using deep convolutional neural networks," in *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTIDAMT-NCON)*, pp. 150–154.
- [23] C. Ouchicha, O. Ammor, and M. Mekkassi, "CVDNet: A novel deep learning architecture for detection of coronavirus ( Covid-19 ) from chest x-ray images," *Chaos , Solitons Fractals*, vol. 140, 2020, doi: 10.1016/j.chaos.2020.110245.
- [24] A.-R. Habib *et al.*, "Artificial intelligence to classify ear disease from otoscopy: A systematic review and meta-analysis," *Clin. Otolaryngol.*, vol. 47, no. 3, pp. 401–413, 2022, doi: <https://doi.org/10.1111/coa.13925>.
- [25] A. A. Hezam, S. A. Mostafa, Z. Baharum, A. Alanda, and M. Z. Salikon, "Combining Deep Learning Models for Enhancing the Detection of Botnet Attacks in Multiple Sensors Internet of Things Networks," *JOIV Int. J. Informatics Vis.*, vol. 5, no. 4, pp. 380–387, 2021.
- [26] K. Hamsu, I. G. P. S. Wijaya, D. Yudhanto, E. A. Yuliani, and H. Mulyana, "Ear Disease Determination on Computer-Assisted Outer and Middle Ear Images," in *The 3rd International Conference on Bioscience and Biotechnology*, 2020.
- [27] A. H. Azizan *et al.*, "A Machine Learning Approach for Improving the Performance of Network Intrusion Detection Systems," *Ann. Emerg. Technol. Comput.*, vol. 5, no. 5, 2021.
- [28] I. G. P. S. Wijaya, D. N. Avianty, F. Bimantoro, and R. Lestari, "Ekstraksi Fitur Citra Radiografi Thorax Menggunakan DWT dan Moment Invariant," *J. Comput. Sci. Informatics Eng.*, vol. 5, no. 2, pp. 158–166, 2021.
- [29] D. Iskandar and Y. K. Suprpto, "Perbandingan Akurasi Klasifikasi Tingkat," *J. Ilm. NERO*, vol. 2, no. 1, pp. 37–43, 2015.
- [30] D. Powers and Ailab, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation," *J. Mach. Learn. Technol.*, vol. 2, pp. 2229–3981, Jan. 2011, doi: 10.9735/2229-3981.
- [31] S. Peres, T. Pham, and R. Phillips, "Validation of the System Usability Scale (SUS)," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 57, pp. 192–196, Sep. 2013, doi: 10.1177/1541931213571043.