

# Ear Disease Clasification Using Deep Learning with Xception and MobileNet-V2 Architecture

## *Klasifikasi Penyakit Telinga Menggunakan Deep Learning dengan Arsitektur Xception dan MobileNet-V2*

Lalu Rudi Setiawan<sup>[1]\*</sup>, I Gede Pasek Suta Wijaya<sup>[1]</sup>, Fitri Bimantoro<sup>[1]</sup>

<sup>[1]</sup>Dept Informatics Engineering, Mataram University  
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: rudistiawannn@gmail.com, gpsutawijaya@unram.ac.id, bimo@unram.ac.id

### **Abstract**

Hearing loss is a significant global health problem, with a high prevalence in Indonesia. Limited access to ENT specialists, especially in remote areas, causes delays in diagnosis and treatment of ear diseases. This research aims to develop an early diagnosis system for ear diseases using deep learning. The proposed method applies Xception and MobileNet-V2 Convolutional Neural Network (CNN) architecture with hyperparameter optimization using Bayesian Optimizer. The dataset consists of 1,101 images covering 20 types of ear diseases, collected using an endoscope ear cleaning kit at Mataram University Hospital. The dataset was divided into 60% training data, 20% validation data, and 20% test data. Xception recorded the best performance with accuracy, precision, recall, and f1-score of 0.911, 0.166, 0.166, and 0.151, respectively. The best model performance was obtained on MobileNet-V2 with the application of Bayesian Optimizer, resulting in the best hyperparameters at Unit Dense 174, Dropout Rate 0.2, and Learning Rate 0.003. This scenario resulted in an increase in accuracy, precision, recall, and f1-score compared to the scenario without hyperparameter search of 0.004, 0.010, 0.018, and 0.012, respectively. This research demonstrates the potential of deep learning in improving early diagnosis of ear diseases.

**Keywords:** Ear Disease, Convolutional Neural Network, MobileNet-V2, Xception, Bayesian Optimizer

\*Corresponding Author

## **1. INTRODUCTION**

According to the World Health Organization (WHO) report on the World Report on Hearing that in 2021 there are around 430 million people in the world who need rehabilitation services for their hearing loss and 109.4 million are in Southeast Asia. WHO estimates that the number of people with hearing loss will increase by 2050 to nearly 2.5 billion and at least 700 million of them will require rehabilitation services. If left untreated, hearing loss can negatively impact many aspects of life, such as communication, language and speech development in children, cognition, education, employment, mental health and interpersonal relationships[1].

Ranking 4th in the prevalence of hearing loss in Southeast Asia shows that people in Indonesia lack awareness of the adverse effects of ear diseases that are not diagnosed early. This phenomenon is caused by several factors, among which is the access of the public to diagnose ear diseases to a specialist doctor

which is fairly expensive and difficult. The distribution of ENT (Ear, Nose and Throat) specialists in Indonesia shows significant inequality, with higher concentrations in major cities and provincial capitals. This leads to disparities in access to healthcare, especially in rural and remote areas[2].

The development and satisfactory results of CNN performance, there are several architectural models of CNN including Xception and MobileNet-V2. Both architectures also show satisfactory performance in performing image classification for medical needs. Research has shown that the Xception and MobileNet-V2 architectures can perform a variety of tasks including automatic classification of benign and malignant gastric ulcers in general digestive endoscopy images[3], detection and classification of skin cancer[4], classification of otitis media ear disease[5], classification of skin diseases[6], detection of COVID-19[7], detection of pneumonia[8], and many others.

Hyperparameter optimizer in its application is generally related to training methods on network

architecture [9]. Based on research that has been done, finding hyperparameter values using a Bayesian optimizer results in optimal model performance in the case of medical images so that the proposed research will apply a Bayesian optimizer that is able to find hyperparameters efficiently and can improve model performance in detection and classification[10].

This research will apply the Xception architecture and MobileNet-V2 which applies a bayesian optimizer to improve the performance of the model in the classification of 20 types of ear diseases totaling 1,101 images collected by the Mataram University Hospital using an endoscope ear cleaning kit. The purpose of this examination is to accurately diagnose the type of ear disease experienced by the patient, so that further action can be taken to prevent or treat the ear disease experienced in accordance with the diagnostic results. Diagnosis using medical devices that produce images requires a specialist and time to study the image so that it can recognize the type of disease experienced. Thus, this research is expected to help specialists and general practitioners to perform early diagnosis to recognize the type of ear disease experienced by the patient without requiring a lot of time. With the help of early diagnosis by the system proposed in this study, people can perform ear disease diagnosis not only through specialists.

## 2. LITERATURE REVIEW

Previous research on ear disease detection was conducted by Zebin Wu et al. [11] has developed a method for automatic classification of ear diseases using Xception and MobileNet-V2 with a dataset of 3 types of disease classes, namely acute otitis media, otitis media effusion, and normal. Je Yeon Lee et al. [12]. Developed a method for automatic classification of tympanic membrane ear diseases using CNN.

Some studies have shown that the application of hyperparameter optimizer using Bayesian optimizer can improve the performance of model performance conducted by Shankar et al. [13]. Developed the Inception-V4 architecture model in detecting and classifying diabetic retinopathy (DR) from color fundus images. The hyperparameter optimizer used is a bayesian optimizer with maximum accuracy, sensitivity, and specificity values higher than the architectural model without hyperparameter optimizer. Research in applying bayesian optimizer was also conducted by Afify et al. [14]. Developed a CNN architecture for automatic diagnosis of ear diseases using otoscopic images. Hyperparameter search using Bayesian optimizer gets the maximum value of

accuracy, sensitivity, specificity, and Positive Predictive Value (PPV) which is higher than the architecture model without hyperparameter optimizer. Research in applying the Bayesian optimizer was also conducted by Amou et al. [10]. Developing CNN architecture in tumor disease classification using MRI images. The application of CNN with hyperparameter search using Bayesian optimizer in the study obtained maximum precision, recall, F1-score, accuracy, and loss values compared to architectural models without hyperparameter optimizer.

Based on the research that has been done, finding hyperparameter values using a Bayesian optimizer results in optimal model performance in the case of medical images so that the proposed research will apply a Bayesian optimizer that is able to find hyperparameters efficiently and can improve model performance in performing detection and classification.

### 2.1. Basic Theory

In this study, the authors used several basic theories to support the research to be conducted:

#### a. Ear Disease

The definition of disease diagnosis is a term that is defined as an effort to establish, determine, and identify a type of disease or health problem suffered or experienced by a patient/sufferer or community. Meanwhile, the result of disease diagnosis is a diagnosis/diagnosis of disease. Proper diagnosis is necessary for effective treatment and avoiding more serious complications [15].

Ear diseases are medical conditions that are usually characterized by symptoms such as pain, discharge such as blood or pus, hearing loss, fullness or buzzing sensation, itching inside the ear, and dizziness or vertigo [16].

#### b. Disease Diagnose

The diagnostic process is a combination of intellectual and manipulative activities. Diagnosis itself is defined as an important process of naming and classifying a patient's illness, which indicates the patient's likely fate and which leads to specific treatment[17].

#### c. Deep Learning

Deep Learning is a branch of machine learning that uses algorithms to model data through a series of multi-layered non-linear transformation functions. Deep Learning offers a powerful architecture, allowing the model to better represent the data by adding more layers. To solve problems with large-scale data, deep artificial neural network concepts are needed, so that

computers can learn quickly and accurately. This principle is called Deep Learning and is often used in research and industry[18].

d. MobileNet-V2

MobileNet-V2 introduces the inverse residual structure and linear bottleneck structure in the network. The inverse residual structure is different from the traditional residual structure. The traditional residual structure reduces the dimension first and then increases it. The inverse residual structure performs the reverse order. First,  $1 \times 1$  point-by-point convolution is used to increase the dimension, then  $3 \times 3$  deep convolution is used to replace the standard  $3 \times 3$  convolution, which can significantly reduce the number of calculations and improve the effectiveness of the network model. The linear bottleneck structure is designed with  $1 \times 1$  point-by-point convolution to reduce the dimension and add it to the input, removing the ReLU activation function in the last layer and replacing it with a linear activation function. This method can solve the problem of serious information loss. In addition, expansion coefficients are introduced to control the size of the network [19].

e. Xception

The Xception network mainly consists of entry flow, middle flow, exit flow, and depthwise separable convolution, with the core being the depthwise separable convolution structure. Xception performs convolution and pooling through three flows, there is depthwise separable convolution to reduce network complexity, ensure maximum information transmission between layers in the network and widen the network. The improved Xception is based on the original Xception model, incorporating an enhanced residual attention mechanism to improve the network's ability to extract global information. The model also uses  $1 \times 1$  convolution to reduce the data dimension and the number of parameter calculations[3].

f. Transfer Learning

Transfer learning is a method that uses a pretrained model on a dataset to solve other related problems. Transfer learning is used by making the pretrained model a starting point to be modified, and updating its parameters so that it fits the new dataset [20]. So, transfer learning is a learning method with a simple idea, which is to take the knowledge learned in some models in some domains and apply it to the required tasks and domains [11].

g. Hyperparameter Optimizer

Current modeling methods often involve various parameters in the data training process. In contrast to other parameters that can change during the training process, there are immutable parameters known as hyperparameters. Hyperparameters are important to define as they can affect the reliability and performance of modeling techniques. Hyperparameters are parameters that are used to adjust and control the modeling method to produce better prediction performance [21].

h. Bayesian Optimizer

Deep learning optimization includes a black box optimization problem where the objective function,  $f(y)$ , is a black box function. It is important to reduce the number of samples over multiple steps across layers. Bayesian optimization is very useful in this domain where human expertise cannot provide better accuracy. Bayesian optimization includes prior information about the function  $f$  and updates the posterior information, which helps reduce losses and maximize model accuracy [22].

### 3. RESEARCH METHODOLOGY

#### 3.1. Materials

The tools and materials used in this research are software and hardware as well as the data needed during the research. The tools used in this research are the followings, Acer Aspire A515-45 processor AMD Ryzen 5 5500U with Radeon Graphics, Windows 11 Home Operation System, Python Programming Language, Google Collab Software, and Microsoft Word. The research materials used for the needs of this research are images of ear diseases totaling 20 disease classes with JPG format.

#### 3.2. Research Flow

The research flow process that will be conducted is described using the flowchart in Figure 1.

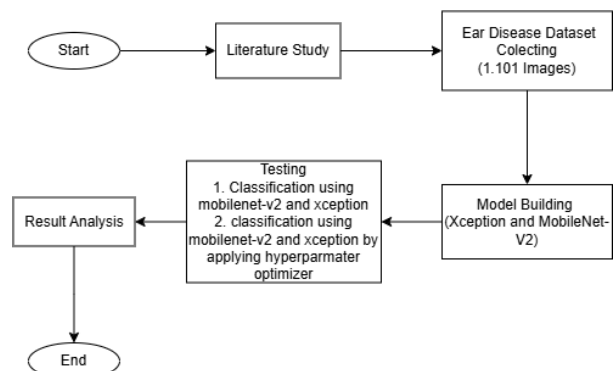


Figure 1. Research Flow System

The initial stage of this research is looking for literature studies to compare the advantages and disadvantages of methods from previous studies. Next is dataset retrieval. The dataset used is an image of ear disease taken using an endoscope ear cleaning tool kit in '.jpg' format consisting of 20 classes of disease types. The dataset is then subjected to preprocessing and data augmentation to multiply and increase the number of data variations. The next stage is model building in accordance with the design that has been made based on the results of a review of some of the literature that has been studied. The next stage is to evaluate the performance of the model with several tests to ensure that the model that has been built is in accordance with the research objectives. After that, documentation related to the research is carried out by making a research report.

### 3.3. Model Building

Architectural model building is a stage that is conducted starting from processing the data used to the testing stage to ensure the research is in accordance with the research objectives.

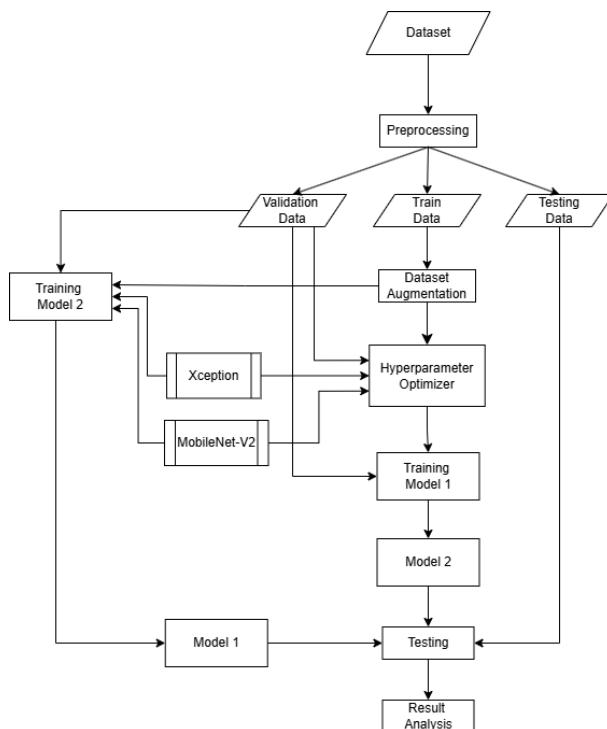


Figure 2. Model Building

Figure II explains that this research begins with the dataset collection stage then preprocessing is carried out to equalize the image size. Next, the dataset is divided into training data, test data and validation data. Next, augmentation is performed on the training data, then initialize the architecture that uses a pretrained model from the Keras library, namely

Xception and MobileNet-V2 as the base model in ear disease classification research. After that, initialize the hyperparameter search using Bayesian optimizer to improve model performance and perform training. Finally, testing the performance of the architecture model that has built.

### 3.4. Dataset

This study uses an image dataset totaling 1,101 ear diseases collected by Mataram University Hospital using an endoscope ear cleaning tool kit. The types of diseases contained in this research dataset and their details are in TABLE I.

TABLE I. Dataset Distribution

Class	Total
Aerotitis Barotrauma	10
Cerumen	55
<i>Corpus Alienum</i>	6
Membran Timpani Normal	161
Miringitis Bulosa	10
Normal	132
Otomikosis	31
Otitis Eksterna Difusa	95
Otitis Eksterna Furunkulosa	14
Otitis Media Akut Hiperemis	79
Otitis Media Akut Oklusi Tuba	58
Otitis Media Akut Perforasi	43
Otitis Media Akut Resolusi	12
Otitis Media Akut Supurasi	55
Otitis Media Efusi	26
Otitis Media Supuratif Kronik Resolusi	23
Otitis Media Supuratif Kronik Tipe Aman	132
Otitis Media Supuratif Kronik Tipe Bahaya	94
Perforasi Membran Timpani	10
Tympanosklerotik	55

Previously, the dataset was divided into 3 subsets with a division ratio of 60% training data, 20% validation data and 20% test data. Model training uses 60% training data with a total of 660 images, then initial testing is carried out using 20% validation data with a total of 220 images. Testing using validation data is needed to ensure that the model not only learns to memorize training data but can also generalize well to new data that has never been learned before. Finally, data testing uses 20% test data with a total of 220 images to evaluate the final performance of the model after training.

### 3.5. Preprocessing

Preprocessing aims to improve image quality for optimal training results and ensure all images are the same size. Xception and MobileNet-V2 have a default image size of 224x224 pixels for training using MobileNet-V2 and 229x229 for Xception. The datasets in this study were also rescaled with pixel values between -1 and 1. Rescaling is done to ensure that the image to be processed matches the default input for the proposed pretrained model.

### 3.6. Data Augmentation

Data augmentation is applied to increase the amount and variety of training data. The training data used in this study has an unbalanced amount and has a very significant imbalance.

The dataset used in this study was captured using an endoscope ear cleaning kit with a horizontal image format. Therefore, the model will only learn from images with horizontal orientation. In testing the model in technical applications, it is likely that the cases that appear are not only images with horizontal orientation. This may cause bias in the model when detecting images with orientations other than horizontal. Therefore, data augmentation is applied to the training data so that the proposed model can learn from diverse training data.

Data augmentation in this study applied upsampling and downsampling. Increasing the amount of training data applying flip, rotation, zoom, and contrast techniques to create different and artificially diverse samples with balanced classes to train data can help the performance of deep learning models to be better [23]. Techniques performed for data augmentation include:

- a. Flip, the addition of data by randomly flipping the image either horizontally, vertically, or both.
- b. Rotation, the addition of data by randomly rotating the image 30% of 360 degrees i.e. up to 108 degrees.
- c. Zoom, enlarges and reduces the image in the height and width dimensions up to 30% of the original size.
- d. Contrast, the addition of data by randomly changing the contrast of the image within a range of 40% lighter or darker than the original contrast.

The application of augmentation is based on the number category of each class dataset. Based on the training data, we categorized the dataset class based on the number category into low, medium and high.

The low category is in the range of less than 20 images, the medium category is between 21 - 60 images and the high category is more than 60 images. The upsampling technique will be applied to the low category with a target average number of the medium category of 41. Downsampling is applied to the high category with the same target number of classes of 41.

### 3.7. Pre-Trained Model

The pretrained model initiation architecture uses input images with a size of 224x224 pixels for MobileNet-V2 and 229x229 for Xception. Then, adding instructions to the Keras library to not include the final layer (fully connected layer) of the original model trained on ImageNet in order to add new layers according to the classification needs in the proposed research. Finally, freezing the weights of the pretrained model so that they do not change during model training. This helps in maintaining the basic features that have been learned in the pretrained model.

### 3.8. The Classification Layer

The additional models are added in order to add layers that can process the output of the pretrained models MobileNet-V2 and Xception. The pretrained model has many layers that process input images and generate features, the added model will process the features generated by the pretrained model to perform classification tasks. Here are some of the layers that are added [4]:

- a. Adding a pretrained model as the basis of the new model.
- b. Global Average Pooling: Adds a global pooling layer that reduces the dimensionality of the data and the output of the convolution layer before the data is fed to the fully connected layer. This layer helps make the feature maps more manageable for storage and computational space and improves model generalization by reducing the risk of overfitting.
- c. Dense: Adding a dense (fully connected) layer as an important component to learn and extract complex data patterns. This layer helps improve the model's ability to understand and classify data better.
- d. ReLU activation: Uses the ReLU (Rectified Linear Unit) activation function to change the output of the dense layer and helps the network to learn more efficiently and reduce computation time [24].

- e. Dropout: Adding a dropout layer to prevent overfitting by randomly dropping certain neurons during the training process.
- f. Softmax Activation: The softmax layer converts the output into probabilities for each class[25].
- g. Dense: Adds a dense layer as an output with 20 neurons for final classification into 20 different classes.

### 3.9. Hyperparameter Optimizer

The steps in performing the Bayesian optimizer technique to find hyperparameters in this study are first, defining the pretrained model as the base model and adding classification layers. However, at this stage the value of each layer will be searched using Bayesian optimizer. The steps in applying Bayesian optimization in this research are first defining the Objective Function that wants to be optimized with the aim of assessing the performance of the model based on different hyperparameter combinations. This function returns the value of the evaluation metric, namely validation loss to maximize validation accuracy which will be used to direct the search for the best hyperparameters [22]. Second, determine the Hyperparameter search space. The hyperparameters to be optimized in this study are as follows:

- a. Units, refers to the number of units in the dense layer influencing the model to learn complex features. The range of values to be searched is 32 to 265.
- b. Dropout rate, dropout works by removing a random number of neurons during training each epoch. This helps the model to be less dependent on certain neurons, thus improving the generalization of the model on new data. The range of dropout values to search for is 0.2 to 0.5.
- c. *Learning rate*, the learning rate value determines the amount of change allowed at each iteration during the training process. The learning rate value search in this study is logarithmically distributed in the range between  $1e-4$  (0.0001) to  $1e-2$  (0.01).

### 3.10. Testing

The test scenario of the proposed research aims to measure the performance of the Xception and MobileNet-V2 architecture models in recognizing ear disease types through images and evaluate the performance of the proposed architecture model using and not implementing the hyperparameter optimizer.

- a. Classification Using Xception and MobileNet-V2

This stage is conducted to determine the performance of the Xception and MobileNet-V2 architecture models. After the pretrained model is initiated, the next step is to add additional models. In this scenario, the value of the classification layer in the additional model is determined with 256 dense units, learning rate  $1e-4$  (0.0001) and *dropout rate* 0,5. Then the results of the ear disease classification scenario using MobileNet-V2 and Xception will be evaluated.

- b. Classification Using Xception and MobileNet-V2 with Hyperparameter Optimizer

This stage is carried out to determine the performance of the Xception and MobileNet-V2 architecture models by searching for hyperparameters using the Bayesian optimizer. The steps in performing the Bayesian optimizer technique to find hyperparameters in this study are first, defining the pretrained model as the base model and adding additional models that are the same as the previous scenario, namely GlobalAveragePooling2D, dense, dropout, and output dense. The test in this scenario is to find the value of the dense unit, learning rate and dropout rate.

## 4. RESULT AND DISCUSSION

### 4.1. Data Processing

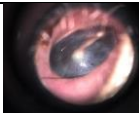
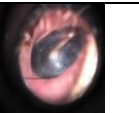


#### 4.1.1. Preprocessing

Preprocessing in this study includes resizing and rescaling the image to ensure that the image to be processed is in accordance with the input default of the pretrained model to be used.

- a. Resize

The datasets used in ear disease classification research have different image sizes so that the entire image is resized to equalize the size of the dataset image to match the input of the pretrained model used. MobileNet-V2 has an input default with an image size of 224x224 and Xception with an image size of 229x229.

TABLE II. Resize

Resize	Before	After
224x224		
229x229		

b. Rescaling

The dataset is rescaled with pixel values between - 1 to 1. Rescaling is done to ensure that the image to be processed matches the default input for the proposed pretrained model.

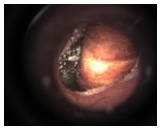
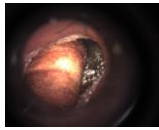


4.1.2. Augmentation

At this stage, data augmentation is performed to overcome the imbalance of the dataset in the training data and to create different and diverse dataset samples. Data augmentation in this study applied upsampling and downsampling. Increasing the amount of training data applies flip, rotation, zoom, and contrast techniques.

a. Flip

The addition of data by randomly flipping the image either horizontally, vertically, or both.

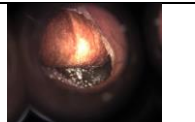
TABLE III. Flip

Flip	Before	After
Vertikal		
Horizontal		

b. Rotation

Addition of data by randomly rotating the image 30% of 360 degrees i.e. up to 108 degrees.



TABLE IV. Rotation

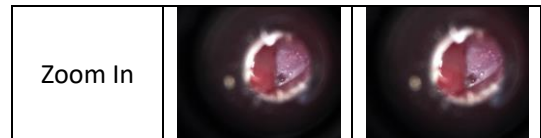
Before	After
	

c. Zoom

Zoom in and out of images in the height and width dimensions up to 30% of the original size.

TABLE V. Zoom



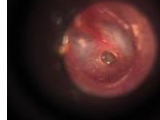
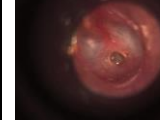
Zoom	Before	After
Zoom Out		



d. Contrast

Data augmentation by randomly changing the contrast of an image within a range of 40% lighter or darker than the original contrast.

TABLE VI. Contrast

Contrast	Before	After
Light		
Dark		

4.2. Model Testing

The evaluation metric to be applied is AUC so that researchers can compare the performance results of Xception and MobileNet-V2 models with and without the application of hyperparameter optimizer more objectively on training and validation sets so as to reveal the gap between the two to show the level of generalization of the model. Confusion matrix is applied to evaluate the performance of the proposed model in predicting disease types between classes in the dataset. Hence testing is conducted to measure the performance of the model under various conditions and to compare the results obtained from each scenario.

4.2.1. Classification using Xception and MobileNet-V2

In this scenario, the value of the classification layer in the additional model is determined with 256 dense units, a learning rate of 1e-4 (0.0001) and dropout rate of 0,5. Testing using a pre-trained model is done by compiling the model using Adam's optimizer. Then, defining a Keras callback to save the model based on the validation AUC performance ensures that the model has the ability to generalize to new data that has never been seen before. Thus, whenever the AUC on the validation set increases, a new model is saved. The training used 100 epochs.

The model stopped improving at the 93rd epoch by saving the model performance against AUC of 0.9875, loss of 0.7713 and validation AUC of 0.9120, validation loss of 1.7422. The model performed well in learning

on the training data however, showed a large gap on the validation data, especially on the loss. This is an indication of overfitting when the model performs very well on training data but poorly on validation data. High loss in validation data shows how far the model's ability to generalize with validation data, a large gap in validation data, especially in loss occurs due to the unbalanced distribution of datasets used for validation data, thus showing high loss.

To determine the performance of the model in each class of ear disease, the performance evaluation of the model on test data is carried out to determine the performance of the model in predicting each class of ear disease types. MobileNet-V2 performance on test data shows the average accuracy, precision, recall, and f1-score results of 0.907, 0.122, 0.118 and 0.119 respectively.

Meanwhile, in Xception, the model stops improving at the 96th epoch by saving the model's best performance against AUC of 0.9726, loss of 1.118 and validation AUC of 0.923, validation loss of 1.6924. The performance of the Xception model is better than MobileNet-V2 with increased accuracy, precision, recall and f1-score values of 0.911, 0.166, 0.166, and 0.151 respectively, although the model still tends to predict the majority class.

#### **4.2.2. Classification Using Xception and MobileNet-V2 with Hyperparameter Optimizer**

This stage is conducted to determine the performance of the proposed architecture models, namely Xception and MobileNet-V2 by searching for hyperparameters using a Bayesian optimizer. Search space is defined for three hyperparameters namely the number of units in the dense layer (32 to 256), dropout rate (0.2 to 0.5), and learning rate  $1e-4$  (0.0001) to  $1e-2$  (0.01). After that, defining the objective function is the process of training and evaluating the model by training for 10 epochs and calculating the validation accuracy. The targeted objective function is to minimize the validation loss value so as to maximize the validation accuracy. The optimization process is carried out using the Gaussian Process for the Bayesian Optimizer. The optimization process applies 50 evaluations to find the best hyperparameters. For the first 10 parameter sets, the algorithm will randomly select values from a predefined range in the search space. The results of these 10 evaluations will be used to build the initial Gaussian Process model, which is used as the search base for the next 40 iterations. The ultimate goal of this optimization is to find the combination of hyperparameters in the search space

(unit dense, dropout rate, learning rate) that results in the highest validation accuracy.

##### **a. MobileNet-V2 Hyperparameter Search**

Hyperparameter search on MobileNet-V2 was found by producing the best hyperparameters of dense unit 174, dropout rate 0.2, and learning rate 00.003103753471420176. The hyperparameters found the best validation accuracy with a value of 0.4859 at the 48th iteration with a training accuracy (categorical accuracy) of 0.349, training loss 0.5247, and validation loss (val loss) 1.577.

The model saved the best performance at the 17th epoch with AUC 0.991, loss 0.597 and validation AUC 0.922, validation loss 1.6710. MobileNet-V2 by applying hyperparameter search using Bayesian optimizer to the unit dense, dropout rate, and learning rate values shows the ability of the model to capture complex features from the data and improve the generalization of the model to new data better than the previous scenario, this can be seen from the higher validation AUC value and lower validation loss. This scenario also shows that the learning rate was chosen appropriately to help ensure model training stability and efficient computation time. Improvement in average prediction value against test data on MobileNet-V2 with hyperparameter search. BO MobileNet-V2 recorded an average accuracy of 0.911, precision of 0.132, recall of 0.136 and f1-score of 0.131.

##### **b. Xception Hyperparameter Search**

The best hyperparameter is found in Xception with dense 209 units, dropout rate 0.2, and learning rate 0.0018277161121728472. The hyperparameter produces the best validation accuracy with a value of 0.4720 at the 31st iteration with a training accuracy (categorical accuracy) of 0.706, training loss of 0.932, and validation loss (val loss) of 1.732.

The model saved the best performance at the 4th epoch with training AUC 0.990, loss 0.618 and validation AUC 0.924, validation loss 1.687. Similar to the scenario using BO MobileNet-V2, training the Xception model by applying hyperparameter search using Bayesian optimizer (BO Xception) obtained the optimal value faster than the previous scenario using Xception by setting hyperparameters (NON-BO Xception). The difference in model convergence in obtaining the optimal value above shows that hyperparameter search using Bayesian Optimizer on the value of unit dense, dropout rate, and learning rate on Xception shows improvisation only on the training set, while on the previous model validation data NON-BO Xception is better. Hyperparameter search using Xception recorded an average value of accuracy 0.907,



precision 0.110, recall 0.106 and f1-score 0.098. These results show that training the model using Xception with hyperparameter search using Bayesian optimizer does not improve the average value of accuracy, precision, recall, and f1-score than the previous scenario by setting hyperparameters (NON-BO Xception).

#### 4.2.3. Classification using Xception and MobileNet-V2 with Dataset Split Scenarios

This stage is conducted to determine the performance of the proposed architecture models, namely Xception and MobileNet-V2, by applying dataset division to the training and test data directories with a division ratio of 80:20. Furthermore, to perform model training, dataset division is carried out on training data with two subsets, namely train data and validation data with a division ratio of 80:20.

Increasing the number of data trains applies the flip, rotation, zoom, and contrast techniques as in the previous scenario.

Model development using MobileNet-V2 with a dataset sharing scenario using the same configuration and model training as the previous training. After 100 epochs running the model saved its best performance at the 96th epoch with AUC 0.989 loss 0.706 and validation AUC 0.931 validation loss 1.550.

The average accuracy, precision, recall, and f1-score were 0.907, 0.112, 0.109, and 0.108, respectively. This scenario produces lower average accuracy, precision, recall, and f1-score values than the previous scenario using MobileNet-V2 with a dataset sharing ratio of 60:20:20.

Testing with a dataset division scenario of 80% training data (80% train data, 20% validation data) and 20% test data is also applied to the Xception model. The model saved its best performance at the 100th epoch with the results of AUC training 0.9820, loss training 0.9253 and AUC Validation 0.9139, loss validation 1.7290. These results show that the convergence of the model in achieving the optimal value using the dataset sharing scenario with a ratio of 60:20:20 is faster than the dataset sharing scenario with a ratio of 80:20. The optimal value achieved on training data shows that this model is better than the previous scenario but worse on validation data. This scenario produces average accuracy, precision, recall, and f1-score values of 0.911, 0.126, 0.157, 0.136 respectively which are lower than the previous scenario using Xception with a dataset sharing ratio of 60:20:20.

This study examines the performance of MobileNet-V2 and Xception models in medical image

classification through three different test scenarios. The first scenario uses a 60:20:20 dataset ratio with manual hyperparameter setting, the second scenario applies automatic hyperparameter search, and the third scenario uses an 80:20 dataset ratio. The model achieved its best performance summarized in Figure III.

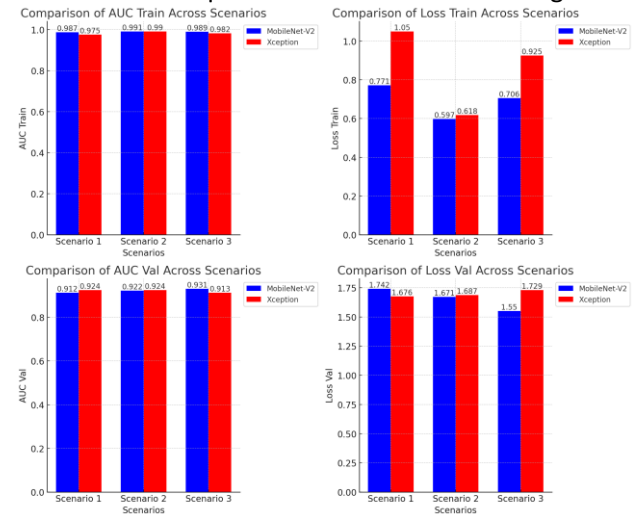


Figure 3. AUC Review.

The test results show that the hyperparameter search in the second scenario effectively improves the performance and efficiency of both models, especially in terms of faster convergence. The third scenario with 80:20 ratio generally outperforms the first scenario, except for the Xception model on the validation data. Nonetheless, all scenarios still show a significant gap between training and validation results, especially on the validation loss. This indicates that the model is still not optimal in generalizing to the validation data, possibly due to the imbalance in the dataset distribution.

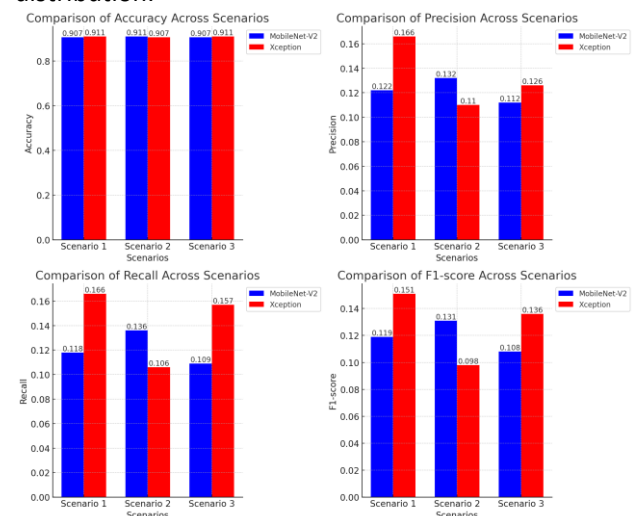


Figure 4. Test Data Prediction Summary.

These average results show that the performance of the model in detecting each ear disease class is extremely poor. High accuracy cannot be taken as a good evaluation performance in this case, because the value is caused by the predominant negative class prediction, so the majority of models tend to predict negative for almost all cases but fail to identify positive cases that actually exist in some cases. Precision gives the proportion of positive predictions that are actually positive. The average of the three model tests obtained a value for precision of 0.127. This means that out of the 20 cases of ear disease predicted as that type of ear disease, only 0.127 were correctly predicted. Recall provides the proportion of actual positive cases that were successfully identified, in the summary of the results above, the average of the three model tests obtained a recall value of 0.132. This means that the model only managed to identify 0.132 of all cases of ear disease types that actually exist. The performance is due to the unbalanced distribution of the dataset which affects the performance of the model.

Evaluation of the test data shows that hyperparameter search improves the performance of MobileNet-V2, but not Xception. For MobileNet-V2, a dataset ratio of 60:20:20 proved more effective than 80:20. However, both models tend to predict negative classes more often, indicating a bias towards the dominant class in the dataset. Overall, this study highlights the importance of proper dataset ratio selection and hyperparameter optimization in improving the performance of medical image classification models. However, a major challenge that still needs to be addressed is the generalization ability of the model, especially in the face of unbalanced datasets.

## 5. CONCLUSION

This study analyzes the effectiveness of MobileNet-V2 and Xception models in ear disease classification, using various test scenarios. Initially, Xception performed better with a dataset ratio of 60:20:20, but after hyperparameter optimization using Bayesian optimizer, MobileNet-V2 surpassed Xception. Consistently, the 60:20:20 dataset sharing ratio proved more effective than 80:20 for both models. However, all scenarios faced significant challenges in the form of large gaps between training and validation performance, indicating generalization issues. Despite the high accuracy achieved, both models showed weaknesses in detecting specific disease classes, with a tendency to predict negatively for the majority of cases. The main factor behind this

problem is dataset imbalance, where the models are only able to detect true positives in classes with a sufficiently large number of samples (more than 31 images).

Based on the results of this study, future suggestions are to address dataset imbalance in model training by collecting additional data for minority classes or through a hierarchical approach by grouping similar classes for stepwise classification, for example, binary normal/abnormal classification followed by multi-class classification for specific disease types. Implement a sampling strategy using proven methods to divide the dataset into training, validation and testing sets to ensure balanced representation in each set. Continue the exploration of hyperparameter search on pre-trained models for more significant improvement in classification performance.

## ACKNOWLEDGMENT

The research can be completed with the help of several parties who have contributed greatly. The author would like to thank several parties who have contributed to this research. Thanks also go to the supervisor who has given his time and attention to discussions about this research so that this research can be completed well.

## REFERENCES

- [1] *World report on hearing: executive summary*. Geneva: World Health Organization: Licence: CC BY-NC-SA 3.0 IGO, 2021. [Online]. Available: <http://apps.who.int/bookorders>.
- [2] S. Herlina, S. Herlina Dalimunthe, and P. Oktamianti Fakultas Kesehatan Masyarakat, "Analisis Kebijakan Pemerataan Dokter Spesialis Di Indonesia: Sebuah Tinjauan Naratif," vol. 10, no. 7, 2022, doi: 10.36418/syntax-literate.v7i10.13365.
- [3] Y. Liu *et al.*, "An xception model based on residual attention mechanism for the classification of benign and malignant gastric ulcers," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-19639-x.
- [4] R. O. Ogundokun *et al.*, "Enhancing Skin Cancer Detection and Classification in Dermoscopic Images through Concatenated MobileNetV2 and Xception Models," *Bioengineering*, vol. 10, no. 8, Aug. 2023, doi: 10.3390/bioengineering10080979.
- [5] X. Lu and Y. A. Firoozeh Abolhasani Zadeh, "Deep Learning-Based Classification for Melanoma Detection Using XceptionNet," *J*

- Healthc Eng*, vol. 2022, 2022, doi: 10.1155/2022/2196096.
- [6] K. A. Muhaba, K. Dese, T. M. Aga, F. T. Zewdu, and G. L. Simegn, "Automatic skin disease diagnosis using deep learning from clinical image and patient information," *Skin Health and Disease*, vol. 2, no. 1, Mar. 2022, doi: 10.1002/ski2.81.
- [7] Y. Kaya and E. Gürsoy, "A MobileNet-based CNN model with a novel fine-tuning mechanism for COVID-19 infection detection," *Soft comput*, vol. 27, no. 9, pp. 5521–5535, May 2023, doi: 10.1007/s00500-022-07798-y.
- [8] E. Ayan, B. Karabulut, and H. M. Ünver, "Diagnosis of Pediatric Pneumonia with Ensemble of Deep Convolutional Neural Networks in Chest X-Ray Images," *Arab J Sci Eng*, vol. 47, no. 2, pp. 2123–2139, Feb. 2022, doi: 10.1007/s13369-021-06127-z.
- [9] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, and A. Gertych, "Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization," *Heliyon*, vol. 10, no. 5, Mar. 2024, doi: 10.1016/j.heliyon.2024.e26586.
- [10] M. A. Amou, K. Xia, S. Kamhi, and M. Mouhafid, "A Novel MRI Diagnosis Method for Brain Tumor Classification Based on CNN and Bayesian Optimization," *Healthcare (Switzerland)*, vol. 10, no. 3, Mar. 2022, doi: 10.3390/healthcare10030494.
- [11] Z. Wu *et al.*, "Deep Learning for Classification of Pediatric Otitis Media," *Laryngoscope*, vol. 131, no. 7, pp. E2344–E2351, Jul. 2021, doi: 10.1002/lary.29302.
- [12] J. Y. Lee, S. H. Choi, and J. W. Chung, "Automated classification of the tympanic membrane using a convolutional neural network," *Applied Sciences (Switzerland)*, vol. 9, no. 9, May 2019, doi: 10.3390/app9091827.
- [13] K. Shankar, Y. Zhang, Y. Liu, L. Wu, and C. H. Chen, "Hyperparameter Tuning Deep Learning for Diabetic Retinopathy Fundus Image Classification," *IEEE Access*, vol. 8, pp. 118164–118173, 2020, doi: 10.1109/ACCESS.2020.3005152.
- [14] H. M. Afify, K. K. Mohammed, and A. E. Hassanien, "Insight into Automatic Image Diagnosis of Ear Conditions Based on Optimized Deep Learning Approach," *Ann Biomed Eng*, vol. 52, no. 4, pp. 865–876, Apr. 2024, doi: 10.1007/s10439-023-03422-8.
- [15] D. Mauli, "Tanggung Jawab Hukum Dokter Terhadap Kesalahan Diagnosis Penyakit Kepada Pasien," *Cepalo*, vol. 2, no. 1, p. 33, Sep. 2019, doi: 10.25041/cepalo.v2no1.1760.
- [16] A. A. Ahmadiham, E. R. D. Leluni, R. Priskila, and V. H. Pranatawijaya, "Sistem Pakar Diagnosa Penyakit Telinga Berbasis WEB Menggunakan Forward Chaining," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, Jun. 2024.
- [17] M. Turnip, "Sistem Pakar Diagnosa Penyakit THT Menggunakan Metode Backward Chaining," *Riau Journal Of Computer Science*, vol. 1, no. 1, pp. 1–8, 2015.
- [18] Hendriyana and Y. Hilman Maulana, "Identifikasi Jenis Kayu menggunakan Convolutional Neural Network dengan Arsitektur Mobilenet," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 1, pp. 70–76, 2020.
- [19] P. Li *et al.*, "A novel CapsNet neural network based on MobileNetV2 structure for robot image classification," Sep. 2022, doi: <https://doi.org/10.3389/fnbot.2022.1007939>.
- [20] P. R. Aningtiyas, A. Sumin, and S. Wirawan, "Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih," *Jurnal Ilmiah Komputasi*, vol. 19, no. 3, Mar. 2020, doi: 10.32409/jikstik.19.3.68.
- [21] M. Y. Aristyanto and R. Kurniawan, "Pengembangan Metode Neural Machine Translation Berdasarkan Hyperparameter Neural Network," *Seminar Nasional Official Statistics*, 2021.
- [22] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using Bayesian optimization," *Evolving Systems*, vol. 12, no. 1, pp. 217–223, Mar. 2021, doi: 10.1007/s12530-020-09345-2.
- [23] T. Kaisyarendika Mazdavilaya, F. Yanto, E. Budianita, S. Sanjaya, and F. Syafria, "KLIK: Kajian Ilmiah Informatika dan Komputer Implementasi VGG 16 dan Augmentasi Zoom Untuk Klasifikasi Kematangan Sawit," *Media Online*, vol. 4, no. 6, 2024, doi: 10.30865/klik.v4i6.1940.
- [24] E. Setia Budi, A. Nofriyaldi Chan, P. Priscillia Alda, and M. Arif Fauzi Idris, "RESOLUSI: Rekayasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan

- Algoritma Convolutional Neural Network,” *Media Online*, vol. 4, no. 5, p. 509, 2024, [Online]. Available: <https://djournals.com/resolusi>
- [25] M. Faizal Nazili, A. B. Firmansyah, and R. Purbaningtyas, “Klasifikasi Keperahan Demensia Alzheimer Menggunakan Metode Convolutional Neural Network pada Citra MRI Otak,” vol. 3, pp. 1–7, Apr. 2023, Accessed: Aug. 26, 2024. [Online]. Available: <https://journal.irpi.or.id/index.php/malcom>