

Article

P-MobileNetV2: Transfer Learning with Dataset Padding for High-Accuracy Classification of Endangered *Oryzias celebensis* and *Oryzias javanicus*

Armin Lawi ^{1,2,3, }, Irma Andriani ⁴, Andi Iqbal Burhanuddin ⁵, and Mario Köppen ⁶

¹ Information Systems Study Program, Faculty of Mathematics and Natural Sciences, Hasanuddin University, Indonesia

² Data Science and Artificial Intelligence Research Group, Hasanuddin University, Indonesia

³ B.J. Habibie Institute of Technology, Parepare, Indonesia

⁴ Department of Biology, Faculty of Mathematics and Natural Sciences, Hasanuddin University, Indonesia

⁵ Department of Fishery, Faculty of Fishery and Marine Sciences, Hasanuddin University, Indonesia

⁶ Department of Creative Informatics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Japan

* Correspondence : armin@unhas.ac.id)

Abstract: *Oryzias Celebensis* and *Oryzias Javanicus* are two species of endemic fish that play a crucial role in the aquatic ecosystem in Indonesia, especially in Java and Sulawesi regions. But unfortunately, both species are threatened with extinction and have been officially declared by the International Union for Conservation of Nature and Natural Resources (IUCN). To address this problem, Artificial Intelligence innovation was built with a Deep Learning approach to conduct classification and identification of the fish species quickly and efficiently. In this study, we developed the MobileNetV2 architecture model using the transfer learning approach and dataset padding in this system to classify *O. Celebensis* and *O. Javanicus*. The P-MobileNetV2 model has better performance evaluation results than other models, where sensitivity = 98.46%, Precision = 98.46%, F1 Score = 98.46% and Accuracy = 98.7%. This advancement enables rapid, accurate conservation efforts for these critically endangered fish

Keywords: Artificial Intelligence; Deep Learning; Tranfer Learning; MobileNetV2; Padding Dataset.

Citation: Lawi, A.; Andriani, I.; Burhanuddin, A.I. and Köppen, M. *Journal Not Specified* **2023**, *1*, 0. <https://doi.org/>

Received:

Revised:

Accepted:

Published:

Copyright: © 2025 by the authors. Submitted to *Journal Not Specified* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Oryzias celebensis dan *Oryzias javanicus* adalah dua spesies ikan endemik yang memiliki peranan penting dalam ekosistem perairan di Indonesia, khususnya di wilayah Sulawesi dan Jawa [1,2]. Kedua spesies ini memiliki peran penting dalam menjaga keragaman hayati dan keseimbangan ekosistem [3]. Identifikasi spesies ikan langka merupakan langkah awal yang krusial dalam upaya konservasi. Dengan mengenali karakteristik morfologi, analisis genetika, dan pemahaman perilaku, kita dapat memahami lebih baik tentang spesies-spesies ini. Identifikasi juga membantu kita mengambil tindakan yang lebih efektif untuk melindungi habitat dan mencegah kepunahan.

Genus *Oryzias* termasuk dalam keluarga *Adrianichthyidae*. Ikan-ikan ini tersebar luas di wilayah Asia Selatan, Asia Timur, dan Asia Tenggara. Habitat alaminya meliputi persawahan, kolam, parit, dan danau [4,5]. *Oryzias celebensis* dan *Oryzias javanicus* merupakan dua spesies yang terancam punah di Indonesia. Ikan Medaka Sulawesi *Oryzias celebensis* merupakan spesies endemik Pulau Sulawesi [6,7]. Sayangnya, Ikan Medaka merupakan ikan endemik yang terancam punah dan telah dinyatakan secara resmi oleh International Union for Conservation of Nature and Natural Resources (IUCN) [8]. Upaya konservasi sangat penting untuk mencegah kepunahan spesies ini. *O. javanicus*: Ikan ini juga dikenal dengan nama Java Medaka. Habitatnya meliputi perairan tawar hingga payau.

O. javanicus tersebar di sebagian besar wilayah Asia, termasuk Indonesia. Kehadirannya di alam liar semakin terancam akibat aktivitas perikanan yang tidak seimbang dengan upaya konservasi dan polusi air yang meningkat. Kedua spesies ini memiliki adaptasi yang unik terhadap lingkungan perairan mereka, dan peran mereka sebagai indikator sensitif terhadap kualitas air telah menjadikannya penting dalam pemantauan dan pengelolaan sumber daya air[9].

Sayangnya, kita menyadari bahwa populasi *O. Celebensis* dan *O. Javanicus* semakin menurun, membawa kedua spesies ini ke ambang kepunahan. Kondisi ini menjadi lebih mengkhawatirkan karena keduanya termasuk dalam kategori ikan langka yang secara endemik ditemukan di Indonesia, khususnya di wilayah Sulawesi dan Jawa. Ancaman terhadap habitat alami mereka, perubahan iklim, dan aktivitas manusia yang merusak lingkungan perairan semakin memperparah keadaan mereka [10]. Dalam upaya untuk melindungi dan melestarikan spesies-spesies langka ini, para peneliti sering menghadapi dilema etis dalam proses identifikasi. Metode konvensional yang biasa digunakan, seperti penggunaan ikan yang sudah mati atau menempatkan ikan di luar air yang berpotensi merugikan, tidak lagi dapat diterima dalam konteks keberlanjutan dan konservasi lingkungan [11].

Untuk mengatasi tantangan ini, inovasi dalam bidang kecerdasan buatan, khususnya teknologi deep learning, menawarkan solusi yang menjanjikan. Dengan memanfaatkan kemampuan komputasi yang canggih, para peneliti dapat menggunakan model deep learning untuk melakukan klasifikasi dan identifikasi spesies ikan, termasuk *Oryzias celebensis* dan *Oryzias javanicus*, secara cepat dan efisien. Pendekatan ini tidak hanya lebih ramah lingkungan, tetapi juga memungkinkan pemantauan yang lebih efektif terhadap populasi ikan langka ini, membantu upaya konservasi untuk menjaga keberlangsungan hidup mereka di masa depan.

2. Related Works and Motivation

Dalam penelitian ini, MobileNetV2 digunakan untuk klasifikasi objek berbasis gambar. Model ini merupakan versi penyempurnaan dari arsitektur MobileNetV1 sebelumnya. MobileNetV2 adalah bagian dari Convolutional Neural Network (CNN) yang dirancang khusus untuk perangkat dengan sumber daya terbatas seperti smartphone, tablet, perangkat IoT, dan sistem tertanam. Dikembangkan oleh Google AI dan dipublikasikan pada tahun 2018, model ini menawarkan beberapa keunggulan yang telah disorot dalam berbagai studi, termasuk arsitekturnya yang ringan, kinerja yang kompetitif, serta kompatibilitasnya dengan berbagai kerangka kerja pembelajaran mendalam [12,13]. Model ini menggabungkan inverse residual block dengan linear bottleneck, yang meningkatkan efisiensi tanpa mengorbankan akurasi. Keunggulan efisiensi ini membuat MobileNetV2 cocok untuk aplikasi real-time pada perangkat berdaya rendah.

Dalam penelitian [14], gambar varietas Cicer arietinum (kacang chickpea) dimasukkan ke dalam dua arsitektur. Pertama, menggunakan transfer learning dengan fine-tuning pada model CNN MobileNetV2 untuk klasifikasi. Kedua, arsitektur hybrid yang menggabungkan lapisan Long Short-Term Memory (LSTM) untuk mempertimbangkan fitur temporal. Hasilnya menunjukkan akurasi 92.3% untuk model pertama dan 92.97% untuk model hybrid, menunjukkan keberhasilan tinggi dalam mengklasifikasikan gambar varietas Cicer arietinum. Selain itu, penulis [15] menerapkan pendekatan deep learning menggunakan MobileNetV2 dalam kerangka kerja PyTorch dan OpenCV Python untuk deteksi masker selama COVID-19. Model mereka secara efisien mengidentifikasi penggunaan masker yang benar, menunjukkan adaptabilitas MobileNetV2 untuk berbagai tugas berbasis gambar, mulai dari klasifikasi pertanian hingga aplikasi kesehatan masyarakat. Studi-studi ini menegaskan efektivitas MobileNetV2 dalam mencapai akurasi tinggi sambil mempertahankan efisiensi komputasi untuk penerapan di dunia nyata.

Dalam sebuah studi sebelumnya[16], penulis memperkenalkan Special Head yang terdiri dari lima lapisan berbeda ke dalam arsitektur MobileNetV2. Model lapisan klasifikasi asli digantikan dengan Special Head, menghasilkan versi modifikasi yang disebut

TL-MobileNetV2. Adaptasi ini mencapai akurasi sebesar 99% -3% lebih tinggi daripada MobileNetV2 standar, dengan tetap mempertahankan tingkat error yang sangat rendah, hanya 1%. Ketika dibandingkan dengan AlexNet, VGG16, InceptionV3, dan ResNet, TL-MobileNetV2 menunjukkan kinerja yang lebih unggul. Sementara itu, penulis [17] mengimplementasikan MobileNetV2 untuk klasifikasi kanker melanoma, melaporkan akurasi hingga 85% pada dataset ISIC Archive, mengungguli model seperti ResNet50V2, InceptionV3, dan InceptionResNetV2. Studi lain [18] mengevaluasi empat jaringan saraf konvolusional Xception, ResNet50, EfficientNetB4, dan MobileNet—untuk mendeteksi penyakit karat pada tiga tanaman komersial penting. Hasilnya menunjukkan EfficientNetB4 sebagai yang paling akurat (akurasi rata-rata = 94.29%), diikuti oleh ResNet50 (93.52%). Meskipun MobileNetV2 sedikit tertinggal dalam perbandingan ini, efisiensi komputasinya membuatnya layak untuk aplikasi dengan sumber daya terbatas, memperkuat fleksibilitasnya di bidang medis dan pertanian. Penelitian lebih lanjut [19] menggunakan MobileNetV2, sebuah deep learning convolutional neural network (DCNN) untuk klasifikasi biji. Sebanyak 14 kelas biji berbeda digunakan dalam eksperimen. Hasilnya menunjukkan akurasi sebesar 98% dan 95% pada set pelatihan dan pengujian, secara berturut-turut.

Table 1. Gap description in building research motivation.

| No. | Previous Study | Current Study |
|-----|--|---|
| 1 | Arsitektur hybrid menggabungkan lapisan Long Short-Term Memory (LSTM), yang juga memperhitungkan fitur data temporal dalam klasifikasi. [14] | Data gambar menampilkan empat warna latar belakang yang berbeda: merah, hitam, biru, dan hijau, dengan pencahayaan dari atas untuk memastikan ikan terlihat jelas. |
| 2 | Implementasi pendekatan deep learning menggunakan framework MobileNetV2, yang terintegrasi dengan PyTorch dan OpenCV dalam Python, untuk deteksi masker selama COVID-19. [15] | Kami mengembangkan aplikasi ini menggunakan Jupyter Notebook, TensorFlow, Roboflow, dan OpenCV untuk identifikasi <i>Oryzias celebensis</i> dan <i>Oryzias javanicus</i> . |
| 3 | Implementasi MobileNetV2 untuk Klasifikasi Kanker Melanoma. MobileNetV2 menunjukkan akurasi yang lebih tinggi dibandingkan dengan ResNet50V2, InceptionV3, dan InceptionResNetV2. [17] | Pengembangan arsitektur MobileNetV2 menggunakan pendekatan transfer learning dengan padding dataset dan lapisan tambahan untuk Classification Layer. Lapisan ini terdiri dari 5 komponen: sebuah Flatten layer dan dua Dense layer dengan fungsi aktivasi ReLU. |
| 4 | Dengan memanfaatkan MobileNetV2, sebuah jaringan saraf konvolusional pembelajaran mendalam (DCNN), untuk klasifikasi biji. [19] | Kami mengembangkan versi modifikasi: P-MobileNetV2 (berbasis MobileNetV2) dan P-VGG16 (berbasis VGG16) untuk mengevaluasi efek dari perubahan arsitektur yang kami lakukan. |

MobileNetV2 digunakan untuk perangkat mobile seperti smartphone dan tablet, yang berfokus pada efisiensi komputasi dan ukuran model yang lebih kecil, karena MobileNetV2 dirancang khusus untuk perangkat mobile dan aplikasi embedded vision. Seperti yang terlihat dalam eksperimen klasifikasi buah yang membandingkan kinerja antara MobileNetV2 dan Inceptionv3, hasilnya menunjukkan bahwa MobileNetV2 memiliki tingkat akurasi yang lebih baik, yang berarti performanya lebih unggul dibandingkan Inceptionv3. Dalam eksperimen kasus berbeda yang membandingkan kinerja MobileNetV2 dengan DenseNet121 untuk klasifikasi terumbu karang, hasilnya menunjukkan bahwa MobileNetV2 lebih optimal untuk perangkat dengan daya komputasi terbatas, serta lebih ringan dan cepat [20,21]. Selain itu, arsitektur MobileNetV2 telah terbukti efektif dalam meningkatkan efisiensi komputasi tanpa mengorbankan akurasi, yang merupakan aspek penting untuk implementasi model ini pada perangkat mobile. Efisiensi komputasi yang tinggi memungkinkan implementasi di dunia nyata, di mana sumber daya komputasi mungkin terbatas [22].

Berkeraan dengan Tabel 1, dalam konteks ini, kami sedang mengembangkan sistem identifikasi dan klasifikasi untuk ikan *O. celebensis* dan *O. javanicus* menggunakan Deep Learning dan Transfer Learning yang diimplementasikan pada MobileNetV2 yang dimod-

ifikasi. Kami percaya pendekatan inovatif kami akan memberikan dampak signifikan dalam proses identifikasi dan klasifikasi. Metodologi komprehensif kami menggabungkan teknik-teknik mutakhir, yang kami yakini dapat diadaptasi untuk berbagai tantangan identifikasi dan klasifikasi dalam aplikasi spesifik industri, khususnya sistem modern untuk mengidentifikasi spesies ikan endemik yang terancam punah.

3. Materials and Methods

Dalam penelitian ini, bahan yang digunakan mencakup berbagai alat dan komponen penting untuk pengolahan data dan implementasi sistem dalam mencapai tujuan sistem cerdas.

3.1. Data Sources

Dataset yang digunakan dalam penelitian ini seluruhnya merupakan data primer yang diambil dari hasil captured citra ikan yang bergerak bebas di dalam akuarium. Seluruh gambar ikan Medaka diperoleh dari sebuah akuarium berisi berbagai jenis ikan air tawar kecil dalam jumlah bervariasi, yang terletak di Laboratorium Biotehnologi, Departemen Biologi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin. Sebagian gambar yang diambil dengan empat warna latar belakang berbeda ditampilkan pada Gambar 1. Jenis ikan yang diklasifikasikan adalah *O. celebensis* dan *O. javanicus* sebagai nenek moyang seluruh ikan medaka di Indonesia. Kedua spesies ini tergolong terancam punah, sehingga diperlukan upaya konservasi serius untuk mencegah kepunahan ikan langka tersebut. Di depan akuarium, dibangun empat ruang persegi panjang dengan lebar datar sebagai area pengambilan gambar saat ikan melintas, aktif, atau berhenti di ruang tersebut. Ruang-ruang ini diberi latar belakang warna berbeda, yaitu merah, hitam, biru, dan hijau. Di atasnya dipasang lampu yang diatur sedemikian rupa agar ikan medaka dengan ukurannya yang relatif kecil dapat terlihat jelas selama proses pemotretan.



Figure 1. Captured Oryzias in four different background colors.

In this research, the materials used include various tools and components essential for processing data and system implementation.

3.2. Data Sources

The dataset used in this study is entirely primary data taken from the results of capturing fish objects moving freely in the aquarium. The classified fish species are *Oryzias celebensis* and *Oryzias javanicus* which are the ancestors of all medaka fish in Indonesia. Both fish are endangered and serious conservation efforts are needed to prevent the extinction of this rare fish. In front of the aquarium, four rectangular rooms with flat widths were built as a place to take pictures when the fish passed by, were active, or stopped in the room. The rooms were given different background colors, i.e., red, black, blue, and green. Above them were placed lights in such a way that the medaka fish could be clearly seen during the photo shoot even though their size was relatively small.

3.3. Research Methodology

menjelaskan diagram alur penelitian: Image processing, Building Block Architecture (YOLO, MobileNetV2, Tranfer Learning), The Novel YOLO-Medaka

Many researchers have proposed various combinations of Machine Learning and Deep Learning (ML/DL) algorithms to solve the problem of object detection, classification, and identification in digital images. In this paper, the YOLO-Medaka algorithm is

introduced to detect and classify digital objects by combining three methods of YOLOv8, MobileNetV2 and ANN-MLP architecture. YOLOv8 is used for object detection because it has a single-stage detection concept, which means detecting objects at one time and has excellent accuracy. As seen in several experiments, for example, after modification, namely YOLOv8-CAB, it succeeds in detecting some objects that are missing detection by YOLOv8 before, also achieves higher accuracy and much better detection confidence. [23]. Another experiment also conducts object detection, such as analysing vehicle detection under various image conditions. This study utilises the YOLOv8 method to process the images with output the bounding boxes and classes of the detected objects. Additionally, data augmentation is applied to improve the model's ability to recognise vehicles from different directions and viewpoints [24].

MobileNetV2 is deployed for mobile devices such as smartphones and tablets, which focus on computational efficiency and smaller model size, since MobileNetV2 is designed for mobile devices and embedded vision applications. As seen in the fruit classification experiment that compared the performance between MobileNetV2 and Inceptionv3, the results show that MobileNetV2 has a better accuracy rate, which means better performance than Inceptionv3. In an experiment with a different case, comparing the performance of MobileNetV2 with DenseNet121 for the classification of coral reefs, the results show that MobileNetV2 is more optimal for devices with limited computing power and is lighter and faster [20,21]. In addition, the MobileNetV2 architecture has proven effective in increasing computational efficiency without sacrificing accuracy, which is an important aspect for implementing this model on mobile devices. High computational efficiency is possible for implementation in the real world, where computing resources may be limited [22].

ANN-MLP is tasked with processing the data conducted by the input, hidden and output layers they have, then producing the classification of *Oryzias celebensis* and *Oryzias javanicus*. ANN-MLP is one of the algorithms that is widely used because of its ease of implementation in web applications (client/server or full stack programming). Some of its implementations include the classification of marine fish even though they are covered by seaweed or coral, the classification of types of diseases in aquatic plants, and the classification of certain problems or certain cases, both inherent in objects and their environment. This paper specifically focuses on the automatic detection and classification of two small, rare and endangered freshwater fish species in Indonesia, namely, *Oryzias celebensis* and *Oryzias javanicus*. The method used to solve this problem consists of 3 stages, namely the data preparation stage based on the model architecture, the model development stage, and the model selection stage through performance evaluation. The illustration of the method is depicted as in Figure ??

Several researchers have conducted object detection, classification and identification in images using MobileNetV2, or versions modified with various techniques, for applications such as mask detection, rust disease classification in plants, melanoma cancer classification, and classification of Cicer arietinum varieties. In our research case, the main focus is the automatic object detection of *Oryzias celebensis* and *Oryzias javanicus*, particularly endangered species in Indonesia. The methods and techniques we used are further explained below:

3.4. Image Processing

3.4.1. Capturing Method

The images were taken with four different background colors: red, black, blue, and green. The aquarium was illuminated from above to make the fish clearly visible.

3.4.2. Rescaling

The captured images of medaka fish will be rescaled to a square size of 224x224 pixels. Two rescaling methods will be applied: Padding and Non-Padding. The Padding method adds a uniform background color to both sides of the image if it's too small. Meanwhile, the Non-Padding method rescales the image directly to fit the square dimensions. All this process can be seen in Figure 8

3.4.3. Normalization

Color value normalization is performed on each pixel by dividing it by 255, as shown in Figure 9.

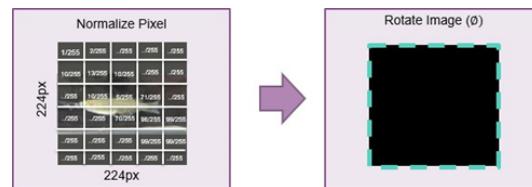


Figure 2. Color Value Normalization for Each Pixel.

3.4.4. Augmentation

Ada dua metode augmentasi yang digunakan yaitu: shift dan rotasi. Randomly shift the image left/right by 20% of its width. Randomly shift the image up/down by 20% of its height, as shown in Figure 10.

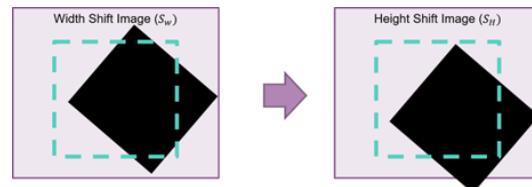


Figure 3. Width shift and height shift of 0.2 (20%).

3.4.5. Image Dataset Splitting

There are 661 images of *Oryzias celebensis*, divided into 456 (70%) for training, 146 (20%) for validation, and 59 (10%) for testing. There are 886 images of *Oryzias javanicus*, split into 612 (70%) for training, 177 (20%) for validation, and 97 (10%) for testing. An example image can be seen in Figure 1.

3.5. Building Block Architecture

3.5.1. YOLOv8

The YOLOv8 architecture is designed to improve speed, accuracy, and ease of use over its predecessors (YOLOv5, YOLOv7). We use YOLOv8 as the main architecture model in this research, some components will be combined with MobileNetv2 and ANN-MLP architecture models. While YOLOv8 has several important components namely:

- Backbone Network. The network serves to extract hierarchical features from the input image, providing a comprehensive representation of the visual information.
- Neck Architecture. The neck structure is responsible for feature fusion, combining multi-scale information and improving the model's ability to detect objects of varying sizes.
- YOLO Head. This component generates predictions based on the features extracted by the backbone network and the neck architecture.

Figure 4. YOLOv8 Architecture.

3.5.2. MobileNetV2

This architecture will be combined with several components in the backbone component of YOLOv8. This is done to improve the performance of hierarchical image feature extraction. Based on MobileNetv2 architecture and features, let's looks the component and its steps.

- Data Preparation. This involves preprocessing the images, splitting the dataset into training and validation sets, and applying data augmentation techniques to improve the model's generalization ability. 243
- Transfer Learning. For initializing the model with pre-trained weights, the training process can be accelerated, and the model can benefit from the knowledge learned from the source dataset. 244
- Fine-tuning. This process Involves training the model on a target dataset while keeping the pre-trained weights fixed for some layers. 245
- Hyperparameter Tuning. Play a role in optimizing the performance of MobileNetV2. Carefully select parameters such as learning rate and regularization techniques to achieve the best possible results. 246

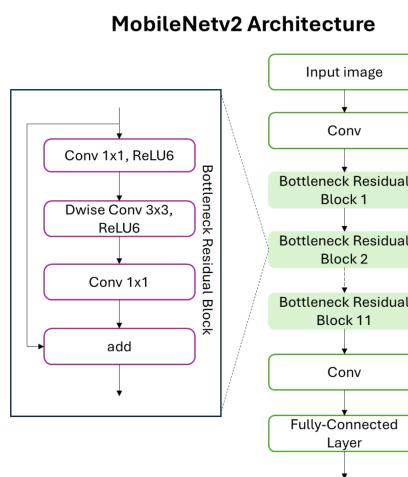


Figure 5. MobileNetV2 Architecture.

3.5.3. VGG16

- Based on the VGG-16 architecture, it is illustrated below in detail:
- Input Layer. Input dimensions: (224, 224, 3). 255
- Convolutional Layers (64 filters, 3×3 filters, same padding). Two consecutive convolutional layers with 64 filters each and a filter size of 3×3. Same padding is applied to maintain spatial dimensions. 256
- Max Pooling Layer (2×2, stride 2). Max-pooling layer with a pool size of 2×2 and a stride of 2. 257
- Convolutional Layers (128 filters, 3×3 filters, same padding). Two consecutive convolutional layers with 128 filters each and a filter size of 3×3. 258
- Max Pooling Layer (2×2, stride 2). Max-pooling layer with a pool size of 2×2 and a stride of 2. 259
- Convolutional Layers (256 filters, 3×3 filters, same padding). Two consecutive convolutional layers with 256 filters each and a filter size of 3×3. 260
- Convolutional Layers (512 filters, 3×3 filters, same padding). Two sets of three consecutive convolutional layers with 512 filters each and a filter size of 3×3. 261
- Max Pooling Layer (2×2, stride 2). Max-pooling layer with a pool size of 2×2 and a stride of 2. 262
- Stack of Convolutional Layers and Max Pooling. Two additional convolutional layers after the previous stack. Filter size: 3×3. 263
- Flattening. Flatten the output feature map (7×7×512) into a vector of size 25088. 264
- Fully Connected Layers. Three fully connected layers with ReLU activation. First layer with input size 25088 and output size 4096. Second layer with input size 4096 and output size 4096. Third layer with input size 4096 and output size 1000, corresponding to 1000 classes. 265

to the 1000 classes in the ILSVRC challenge. Softmax activation is applied to the output of the third fully connected layer for classification.

3.5.4. Transfer Learning

- Load MobileNetV2 model.
- Freeze all layers.
- Create a Top Model with the following sequence:

$$\text{Flatten}, \text{Dense}^{\rightarrow}(x_1), \text{Dense}^{\rightarrow}(x_2), \text{Dropout}(\rho), \text{SoftMax}.$$

- After constructing the Top Model's output layer, fine-tuning is performed by freezing the first X layers, then fitting the model with these parameters: Total Epochs, Batch Size, and Steps per Epoch.

3.6. Computational Environment

In this applied research project, the process of forming deep learning and transfer learning models uses the DIKTI AI Center facility which uses NVIDIA technology with a computing capacity of 25 PetaFLOPS. The supercomputer facility consists of five NVIDIA DGX A100 server machine nodes. Each node has a dual AMD Rome CPU with eight graphics processing units (GPUs) with multi-instance GPU (MIG) capabilities: 4 GPU @ 40GB, Processor 8 Core, and RAM 64 GB. Each server node is equipped with 1TB of RAM and 5TB of high-speed NVME storage, with a total processing power of 5 TeraFLOPS per node. The supercomputer nodes are interconnected via a high-speed Mellanox network, with NVLink links between the five units, each of which has 8 Core GPUs. The procurement of the DIKTI AI Center supercomputer facility is used to strengthen the creation of national AI talent through various training and education activities in collaboration with industry. This facility can be used by educational institutions, including Hasanuddin University, to facilitate the development of AI technology innovations to meet the needs of industry and society. The software used is Jupiter Notebook, TensorFlow, Roboflow, and OpenCV cuda version 11.4.

3.7. Pemilihan Arsitektur Model

3.7.1. YOLOv8 untuk Deteksi Objek

YOLOv8 (*You Only Look Once version 8*) dipilih karena beberapa keunggulan berikut:

- **Kecepatan dan Akurasi Tinggi:** Arsitektur berbasis CNN dengan *backbone* CSPDarknet53 dan *neck* PANet (*Path Aggregation Network*) memungkinkan deteksi objek secara *real-time* dengan presisi tinggi, cocok untuk analisis citra biologi yang memerlukan kecepatan.
- **Multi-Scale Detection:** Kemampuan deteksi pada berbagai skala melalui *feature pyramids* (FPN) penting untuk ikan *Oryzias* yang mungkin memiliki variasi ukuran dalam citra.
- **Optimasi Loss Function:** Menggunakan *task-aligned assigner* dan *Distribution Focal Loss* untuk menangani ketidakseimbangan kelas (misalnya, dominasi *background*).

3.7.2. MLP untuk Klasifikasi

Multi-Layered Perceptron (MLP) digunakan sebagai *classifier head* dengan pertimbangan:

- **Non-Linearitas Kompleks:** Lapisan tersembunyi (*hidden layers*) dengan aktivasi ReLU/RBF mampu menangkap pola non-linear pada fitur morfologi ikan.
- **Adaptabilitas:** MLP fleksibel untuk dikombinasikan dengan *feature extractor* YOLOv8 (misalnya, mengambil vektor fitur dari *ROI pooling*).
- **Interpretabilitas:** Bobot MLP dapat dianalisis untuk memahami kontribusi fitur tertentu dalam klasifikasi.

Figure 6. Diagram arsitektur hybrid YOLOv8 (deteksi) + MLP (klasifikasi).**3.8. Evaluasi Kinerja****3.8.1. Confusion Matrix dengan Kronecker Delta**

Confusion matrix untuk tiga kelas didefinisikan sebagai matriks C berukuran 3×3 , di mana elemen C_{ij} menyatakan jumlah sampel kelas i yang diprediksi sebagai kelas j . Untuk menghitung metrik evaluasi, digunakan fungsi Kronecker Delta (δ_{ik}):

$$\delta_{ik} = \begin{cases} 1 & \text{jika } i = k, \\ 0 & \text{lainnya,} \end{cases} \quad (1)$$

- **True Positive (TP)** untuk kelas k :

$$TP_k = \sum_{i=1}^N \delta_{y_i,k} \cdot \delta_{\hat{y}_i,k}, \quad N = \text{total sampel} \quad (2)$$

- **False Positive (FP)** untuk kelas k :

$$FP_k = \sum_{i=1}^N (1 - \delta_{y_i,k}) \cdot \delta_{\hat{y}_i,k} \quad (3)$$

- **False Negative (FN)** untuk kelas k :

$$FN_k = \sum_{i=1}^N \delta_{y_i,k} \cdot (1 - \delta_{\hat{y}_i,k}) \quad (4)$$

3.8.2. ROC-AUC: Pemahaman Mendalam

Kurva ROC (*Receiver Operating Characteristic*) menggambarkan hubungan antara:

- **True Positive Rate (TPR):** $\frac{TP}{TP+FN}$ (kemampuan model mendekripsi kelas positif).
- **False Positive Rate (FPR):** $\frac{FP}{FP+TN}$ (proporsi kesalahan pada kelas negatif).

Untuk masalah multi-kelas, ROC-AUC dihitung dengan strategi *one-vs-rest*:

- Setiap kelas dianggap sebagai *positive*, sementara lainnya sebagai *negative*.
- AUC dihitung untuk setiap kelas, lalu dirata-ratakan (*macro-average*).

Interpretasi AUC:

- **AUC = 1:** Model sempurna membedakan kelas.
- **AUC = 0.5:** Klasifikasi acak (tidak lebih baik dari tebakan).
- **AUC < 0.5:** Model memiliki kinerja lebih buruk dari tebakan acak.

Figure 7. Kurva ROC untuk tiga kelas ($AUC_a = 0.98$, $AUC_b = 0.95$, $AUC_c = 0.99$).**4. Hasil dan Pembahasan****4.1. Analisis Confusion Matrix****Table 2.** Confusion Matrix dengan Kronecker Delta

| | Prediksi a | Prediksi b | Prediksi c | Total |
|-----------------|------------------------------|------------------------------|------------------------------|-------|
| Actual a | 45 (TP_a) | 3 ($FN_{a \rightarrow b}$) | 2 ($FN_{a \rightarrow c}$) | 50 |
| Actual b | 5 ($FN_{b \rightarrow a}$) | 40 (TP_b) | 5 ($FN_{b \rightarrow c}$) | 50 |
| Actual c | 1 ($FN_{c \rightarrow a}$) | 4 ($FN_{c \rightarrow b}$) | 95 (TP_c) | 100 |

4.2. Analisis ROC-AUC

- **Kelas a (AUC = 0.98):** Hampir sempurna, sedikit kesalahan pada batas dengan kelas b.
- **Kelas b (AUC = 0.95):** Kesalahan utama karena variasi morfologi yang tumpang tindih dengan kelas a.
- **Kelas c (AUC = 0.99):** *Background* mudah dibedakan karena fitur tekstur yang unik.

4.3. Research Methodology

Kami mengikuti pendekatan sistematis untuk mengembangkan sistem deteksi dan klasifikasi objek pada gambar, khususnya untuk mengidentifikasi *O. celebensis* dan *O. javanicus*. Proses kami mencakup langkah-langkah kunci seperti **Image Processing** diawali dengan pra-proses dan peningkatan data gambar mentah untuk meningkatkan pelatihan model mencakup: *Rescaling*, *Normalization*, *Augmentation*, *Image Dataset Splitting*. Selanjutnya **Building Block Architecture** membangun kerangka kerja yang kokoh dengan mengimplementasikan arsitektur MobileNetV2 yang telah dimodifikasi, dikombinasikan dengan transfer learning untuk meningkatkan efisiensi. Meskipun beberapa peneliti telah menggunakan arsitektur MobileNetV2 dengan berbagai variasi modifikasi dan dikombinasikan dengan teknik tertentu, kemudian diaplikasikan pada deteksi masker, klasifikasi penyakit karat pada tanaman, deteksi kanker melanoma, dan klasifikasi varietas *Cicer arietinum*. Namun fokus utama kami adalah pada deteksi otomatis *O. celebensis* dan *O. javanicus*. Spesies ini sangat penting karena terancam punah di Indonesia, dan pendekatan yang kami rancang bertujuan untuk mengatasi tantangan unik dalam identifikasi tersebut.

4.4. Image Processing

4.4.1. Rescaling

Gambar medaka yang telah diambil akan diubah ukurannya menjadi bentuk persegi berukuran 224x224 piksel. Dua metode perubahan ukuran akan diterapkan: Padding dan Non-Padding. Melakukan Padding yaitu dengan menambahkan piksel di sekitar tepi gambar untuk menyesuaikan ukurannya tanpa mengubah konten aslinya. Hal ini bertujuan untuk mempertahankan dimensi gambar, memastikan gambar memiliki ukuran yang seragam dalam batch processing, menghindari penyusutan (shrinkage) gambar setelah filter diterapkan. Sementara itu, metode Non-Padding mengubah ukuran gambar secara langsung agar sesuai dengan dimensi persegi, akan tetapi resolusi objek gambar atau konten aslinya bisa berubah dan dapat menghilangkan informasi pada bagian tepi akan berperngaruh pada proses selanjutnya. Seluruh proses ini dapat dilihat pada Gambar 8.

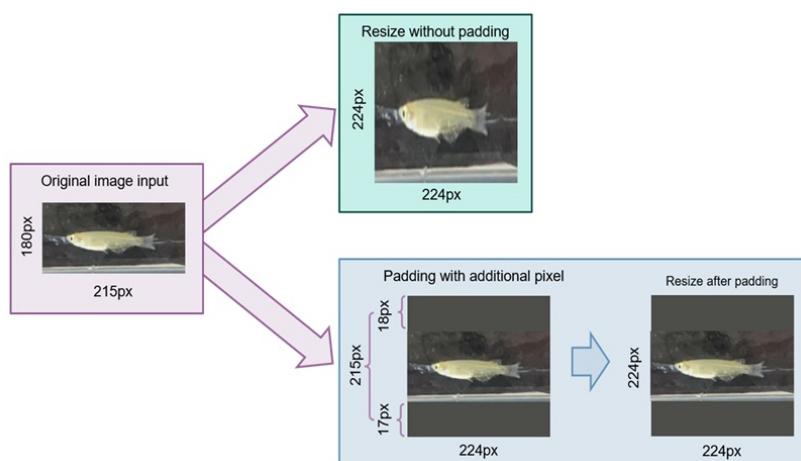


Figure 8. Comparison of Non-Padding and Padding Resize mechanisms.

4.4.2. Normalization

Proses Normalisasi akan mengubah nilai piksel gambar agar berada dalam rentang tertentu. Hal ini dilakukan karena normalisasi yang baik akan menghasilkan akurasi yang lebih bagus, memiliki kemampuan generalisasi yang lebih baik, efisiensi komputasi, hasil eksperimen lebih mudah direproduksi serta menghindari error akibat perbedaan skala. Normalisasi nilai warna dilakukan pada setiap piksel dengan membaginya kedalam 255, as shown in Figure 9.

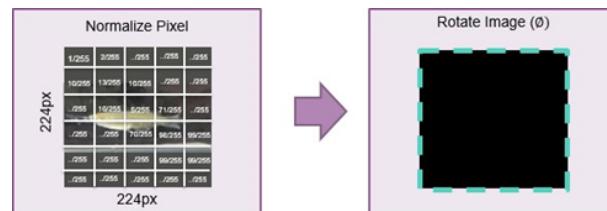


Figure 9. Color Value Normalization for Each Pixel.

4.4.3. Augmentation

Ada dua metode augmentasi yang digunakan yaitu: **Width shift**, **Height shift** dan **Rotata**. Dalam Augmentasi **Width shift** yaitu menggeser gambar secara horizontal (kiri atau kanan) sebanyak 20%, begitu pula dengan **Height shift** menggeser gambar secara vertikal (atas atau bawah) sebanyak 20%. Hal ini bertujuan untuk memperkaya variasi posisi objek agar model tetap dapat mengenali objek meskipun tidak berada pada posisi sebenarnya. Selanjutnya **Rotata** yaitu memutar gambar hingga mencapai sudut tertentu (dalam derajat). Hal ini bertujuan untuk mengenali object gambar meski berada dalam sudut pandang yang berbeda sehingga menambah kemampuan model dalam mengenali objek dari berbagai variasi sudut pandang. seperti yang ditunjukkan pada Gambar 10.

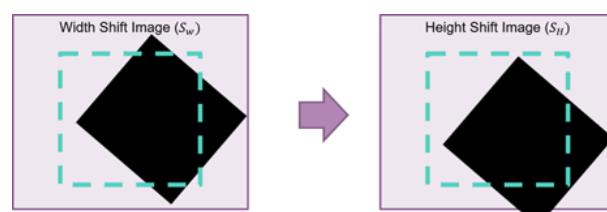


Figure 10. Width shift and height shift of 0.2 (20%).

4.4.4. Image Dataset Splitting

Untuk pembagian data dalam penelitian ini yaitu: terdapat 661 gambar *O. celebensis*, dibagi menjadi 456 (70%) untuk pelatihan, 146 (20%) untuk validasi, dan 59 (10%) untuk pengujian. Ada 886 gambar *O. javanicus*, dibagi menjadi 612 (80%) untuk pelatihan, 177 (20%) untuk validasi, dan 97 (10%) untuk pengujian. Pembagian data gambar dapat dilihat pada Grafik berikut 11.

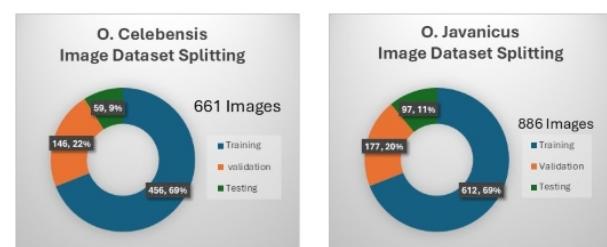


Figure 11. Image Dataset Splitting.

4.5. Building Block Architecture

Dalam penelitian ini, kami memecah sistem kedalam blok-blok yang lebih kecil dan biasa kami sebut dengan komponen-komponen sistem yang dapat digabung menjadi sebuah sistem yang utuh, termasuk modifikasi yang kami lakukan dalam beberapa komponen sebagai ciri khas dalam sebuah pengembangan sistem yang kami lakukan atau kebaruan (novelty) yang kami selipkan dalam sistem untuk menyelesaikan identifikasi dan klasifikasi *O. celebensis* dan *O. javanicus*.

4.5.1. MobileNetV2

Arsitektur MobileNetV2 dibangun dari *convolutional neural network (CNN)* yang ringan dan efisien, dirancang khusus untuk perangkat dengan sumber daya terbatas seperti smartphone dan perangkat IoT. Hal ini dilakukan untuk meningkatkan kinerja ekstraksi fitur gambar secara hierarkis. Berdasarkan arsitektur dan fitur MobileNetV2, mari kita lihat komponen-komponennya serta langkah-langkahnya berikut ini:

- **Persiapan Data.** Ini melibatkan pra-pemrosesan gambar, membagi dataset menjadi set pelatihan dan validasi, serta menerapkan teknik augmentasi data untuk meningkatkan kemampuan generalisasi model.
- **Transfer Learning.** Dengan menginisialisasi model menggunakan bobot yang telah dilatih sebelumnya (pre-trained weights), proses pelatihan dapat dipercepat, dan model dapat memanfaatkan pengetahuan yang telah dipelajari dari dataset sumber.
- **Fine-tuning.** Proses ini melibatkan pelatihan model pada dataset target sambil mempertahankan bobot yang telah dilatih sebelumnya (pre-trained weights) untuk beberapa lapisan tertentu.
- **Penyesuaian Hyperparameter.** Berperan dalam mengoptimalkan kinerja MobileNetV2. Parameter seperti learning rate dan teknik regularisasi harus dipilih dengan cermat untuk mencapai hasil terbaik.

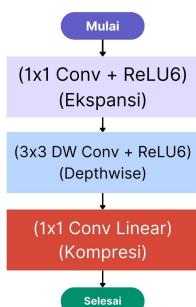


Figure 12. MobileNetV2 Architecture.

4.5.2. Transfer Learning

Dalam penelitian ini, kami mengimplementasikan salah satu teknik dalam *Machine Learning*, yaitu **transfer learning**, dengan tujuan utama untuk meningkatkan efektivitas dan efisiensi dalam proses identifikasi serta klasifikasi spesies. Transfer learning memungkinkan pemanfaatan kembali pengetahuan yang telah diperoleh dari pelatihan model pada satu domain ke domain lain yang serupa, sehingga dapat menghemat waktu pelatihan dan sumber daya komputasi, terutama ketika data pelatihan terbatas.

Model dasar yang digunakan dalam studi ini adalah **MobileNetV2**, sebuah arsitektur *convolutional neural network (CNN)* yang ringan dan efisien, sangat cocok untuk keperluan komputasi terbatas namun tetap memiliki performa representasi fitur yang baik. Model ini pertama-tama dilatih pada dataset *O. celebensis*, di mana model belajar mengekstraksi ciri-ciri penting untuk proses identifikasi. Setelah model mencapai performa yang optimal pada data *O. celebensis*, model yang sama kemudian digunakan kembali untuk proses identifikasi spesies lain, yaitu *O. javanicus*. Pendekatan ini dilakukan untuk mengatasi

keterbatasan data *O. javanicus* serta menghindari pelatihan model dari awal (training from scratch), yang dapat mengakibatkan pemborosan waktu dan biaya komputasi.

Adapun tahapan implementasi *transfer learning* yang kami terapkan adalah sebagai berikut:

- **Memuat model dasar (base model):** Model MobileNetV2 dimuat tanpa lapisan klasifikasi teratas (top classification layers), dengan bobot awal dari pelatihan pada dataset ImageNet.
- **Membekukan seluruh lapisan awal:** Seluruh lapisan dari model dasar dibekukan (*freeze*), sehingga bobotnya tidak berubah selama pelatihan tahap awal. Hal ini bertujuan untuk mempertahankan fitur-fitur umum yang telah dipelajari dari dataset besar seperti ImageNet.
- **Menambahkan Top Model:** Di atas model dasar, kami menambahkan beberapa lapisan tambahan untuk klasifikasi spesifik dataset. Susunan lapisan tersebut adalah:

$$\text{Flatten}, \text{Dense}(\vec{x}_1), \text{Dense}(\vec{x}_2), \text{Dropout}(\vec{\rho}), \text{SoftMax}.$$

dengan \vec{x}_1 dan \vec{x}_2 menunjukkan jumlah unit neuron masing-masing lapisan dense, dan $\vec{\rho}$ menyatakan rasio dropout untuk mengurangi *overfitting*.

- **Fine-tuning:** Setelah pelatihan awal dilakukan dengan top model, kami membuka kembali sejumlah lapisan akhir dari model dasar (misalnya, lapisan setelah ke-X, untuk melakukan pelatihan menyeluruh (end-to-end) bersama top model. Proses ini dilakukan agar model dapat menyesuaikan bobot fitur tingkat tinggi terhadap karakteristik spesifik dataset *O. javanicus*. Proses fine-tuning ini dilakukan dengan parameter yang telah ditentukan, yaitu jumlah epoch, ukuran batch, dan jumlah langkah per epoch.

Pendekatan ini terbukti efektif dalam mengoptimalkan kinerja klasifikasi pada dataset target dengan jumlah data terbatas, serta secara signifikan mengurangi beban komputasi dibandingkan dengan pelatihan model dari awal.

4.5.3. VGG16

VGG16 merupakan salah satu arsitektur dari *Convolutional Neural Network* (CNN) yang dikembangkan oleh kelompok riset Visual Geometry Group (VGG) di University of Oxford. Arsitektur ini diperkenalkan oleh Karen Simonyan dan Andrew Zisserman pada tahun 2014 melalui makalah berjudul "*Very Deep Convolutional Networks for Large-Scale Image Recognition*". Angka "16" pada VGG16 mengacu pada jumlah total lapisan dengan bobot yang dapat dilatih, yang terdiri dari 13 lapisan konvolusional dan 3 lapisan fully connected.

Struktur arsitektur VGG16 secara umum dapat dijelaskan sebagai berikut:

- **Lapisan Input:** Ukuran input adalah $(224 \times 224 \times 3)$, yang merepresentasikan gambar RGB beresolusi 224×224 piksel.
- **Blok Konvolusi 1:** Dua buah lapisan konvolusional dengan 64 filter berukuran 3×3 , padding 'same', dan fungsi aktivasi ReLU, diikuti oleh lapisan max pooling berukuran 2×2 dengan *stride* 2.
- **Blok Konvolusi 2:** Dua buah lapisan konvolusional dengan 128 filter berukuran 3×3 , padding 'same', ReLU, diikuti oleh max pooling 2×2 , *stride* 2.
- **Blok Konvolusi 3:** Tiga buah lapisan konvolusional dengan 256 filter berukuran 3×3 , padding 'same', ReLU, diikuti oleh max pooling 2×2 , *stride* 2.
- **Blok Konvolusi 4:** Tiga buah lapisan konvolusional dengan 512 filter, ukuran filter 3×3 , padding 'same', ReLU, diikuti oleh max pooling 2×2 , *stride* 2.
- **Blok Konvolusi 5:** Tiga buah lapisan konvolusional dengan 512 filter, ukuran 3×3 , padding 'same', ReLU, diikuti oleh max pooling 2×2 , *stride* 2.
- **Flatten:** Peta fitur hasil akhir dari blok konvolusi (berukuran $7 \times 7 \times 512$) diratakan menjadi vektor 1D dengan panjang 25088.
- **Fully Connected Layers:**

- Lapisan pertama: input 25088 neuron, output 4096 neuron, aktivasi ReLU. 490
 - Lapisan kedua: input 4096 neuron, output 4096 neuron, aktivasi ReLU. 491
 - Lapisan ketiga: input 4096 neuron, output 1000 neuron (jumlah kelas pada ILSVRC), aktivasi softmax. 492
- 493



Figure 13. Visual Geometry Group (VGG16) Architecture.

4.6. Computational Environment

Dalam proyek penelitian ini, proses pembentukan model deep learning dan transfer learning menggunakan fasilitas DIKTI AI Center yang memanfaatkan teknologi NVIDIA dengan kapasitas komputasi 25 PetaFLOPS. Fasilitas superkomputer ini terdiri dari lima node mesin server NVIDIA DGX A100. Setiap node dilengkapi dengan dual AMD Rome CPU dan delapan unit pengolah grafis (GPU) yang memiliki kemampuan multi-instance GPU (MIG): 4 GPU @ 40GB, Processor 8 Core, dan RAM 64 GB. Setiap node server juga dilengkapi dengan RAM 1TB dan penyimpanan NVME berkecepatan tinggi 5TB, dengan total daya pemrosesan 5 TeraFLOPS per node. Node-node superkomputer tersebut saling terhubung melalui jaringan berkecepatan tinggi Mellanox, dengan tautan NVLink antara kelima unit, yang masing-masing memiliki 8 Core GPU. Pengadaan fasilitas superkomputer DIKTI AI Center ini bertujuan untuk memperkuat pembentukan talenta AI nasional melalui berbagai pelatihan dan kegiatan pendidikan yang dilakukan bekerja sama dengan industri. Fasilitas ini dapat dimanfaatkan oleh institusi pendidikan, termasuk Universitas Hasanuddin, untuk mempermudah pengembangan inovasi teknologi AI guna memenuhi kebutuhan industri dan masyarakat. Perangkat lunak yang digunakan meliputi Jupiter Notebook, TensorFlow, Roboflow, dan OpenCV versi CUDA 11.4.

4.7. Evaluation Metrics

Setelah proses fitting model, kinerja model dievaluasi menggunakan dua skenario berdasarkan **Confusion Matrix** dan *Receiver Operating Characteristic - Area Under the Curve* (ROC-AUC). Secara umum untuk mengukur kinerja sistem khususnya pada Machine Learning penggunaan **Confusion Matrix** menjadi pilihan banyak peneliti karena dapat menyajikan berbagai metrik yang memberikan gambaran yang sangat detail tentang bagaimana model melakukan klasifikasi, termasuk jenis kesalahan yang dilakukan oleh model. Tabel **Confusion Matrix** dapat dilihat pada gambar 14.

| | | Actual Value | |
|------------------|----------|--------------|----------|
| | | Positive | Negative |
| Predictive Value | Positive | TP | FP |
| | Negative | FN | TN |

Figure 14. Confusion Matrix.

| | | Actual class | |
|---------------------|----|------------------------------|------------------------------|
| | | +1 | -1 |
| Prediction class | +1 | $TP = \sum_i \delta(+1, +1)$ | $FP = \sum_i \delta(+1, -1)$ |
| | -1 | $FN = \sum_i \delta(-1, +1)$ | $TN = \sum_i \delta(-1, -1)$ |

Confusion Matrix bekerja dengan cara mengelompokkan hasil prediksi kedalam empat kategori yaitu: TP, TN, FN, FP. Semua nilai tersebut disusun dalam bentuk matriks. Adapun formula secara umum dituliskan sebagai berikut:

Untuk mengevaluasi performa model klasifikasi, kami menggunakan beberapa metrik evaluasi standar yang dirumuskan menggunakan fungsi delta Kronecker, $\delta_{a,b}$, yang didefinisikan sebagai berikut:

$$\delta(y_i, \hat{y}_i) = \begin{cases} 1, & \text{jika } y_i = \hat{y}_i, \\ 0, & \text{jika } y_i \neq \hat{y}_i. \end{cases}$$

dengan

- y_i adalah label sebenarnya untuk sampel ke- i ,
- \hat{y}_i adalah label hasil prediksi model untuk sampel ke- i ,
- n adalah jumlah total sampel, dan
- $\mathcal{K} = \{1, 2, \dots, k\}$ adalah himpunan label kelas.

1. Accuracy (Akurasi)

mengukur proporsi prediksi label yang benar terhadap seluruh jumlah sampel:

$$\text{Accuracy} = \frac{1}{n} \sum_{i=1}^n \{\delta(1, 1) + \delta(0, 0)\}$$

2. Precision (Presisi) per Kelas k

adalah rasio antara jumlah prediksi benar untuk kelas k terhadap semua prediksi yang diberikan sebagai kelas k :

$$\text{Precision}_k = \frac{\sum_{i=1}^n \delta_{y_i,k} \cdot \delta_{\hat{y}_i,k}}{\sum_{i=1}^n \delta_{\hat{y}_i,k}}$$

3. Recall (Sensitivitas) per Kelas k

adalah rasio antara jumlah prediksi benar untuk kelas k terhadap semua sampel yang benar-benar berasal dari kelas k :

$$\text{Recall}_k = \frac{\sum_{i=1}^n \delta_{y_i,k} \cdot \delta_{\hat{y}_i,k}}{\sum_{i=1}^n \delta_{y_i,k}}$$

4. F1-Score per Kelas k

merupakan rata-rata harmonik dari Precision dan Recall:

$$F1_k = 2 \cdot \frac{\text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}$$

5. Macro-Averaged Metrics

digunakan untuk menghitung metrik rata-rata antar kelas:

$$\text{Macro-F1} = \frac{1}{K} \sum_{k=1}^K F1_k$$

Pendekatan ini memungkinkan perhitungan yang konsisten dan elegan dalam pengukuran performa klasifikasi, serta memberikan fleksibilitas dalam menangani klasifikasi multi-kelas.

$$\text{Accuracy} = \frac{1}{n} \sum_{i=1}^n \delta_{y_i, \hat{y}_i}$$

$$\text{Precision}_k = \frac{\sum_{i=1}^n \delta_{y_i, k} \cdot \delta_{\hat{y}_i, k}}{\sum_{i=1}^n \delta_{\hat{y}_i, k}},$$

$$\text{Recall}_k = \frac{\sum_{i=1}^n \delta_{y_i, k} \cdot \delta_{\hat{y}_i, k}}{\sum_{i=1}^n \delta_{y_i, k}}$$

$$\text{Sensitivity (recall)} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$F_n - \text{Score} = (n+1) \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Formula notation:

- True Positive (TP), $\delta_{1,1}$: Prediksi benar (positif) sesuai data aktual.
- True Negative (TN), $\delta_{0,0}$: Prediksi benar (negatif) sesuai data aktual.
- False Positive (FP), $\delta_{0,1}$: Prediksi salah (positif, padahal aktual negatif).
- False Negative (FN), $\delta_{1,0}$: Prediksi salah (negatif, padahal aktual positif).

Selain **Confusion Matrix** dalam penelitian ini kami juga menggunakan Receiver Operating Characteristic - Area Under the Curve (ROC-AUC). ROC menggambarkan performa model klasifikasi biner (Binairy Classifier). Yaitu dengan membandingkan antara *True Positive Rate* (TPR/Recall) dan *False Positive Rate* (FPR). Sementara AUC merangkum kinerja dalam bentuk skala yang bergerak dibawah Curva ROC dengan ranges nilai dari 0 hingga 1. Semakin mendekati nilai 1 maka model semakin bagus. Curva tersebut digambarkan dalam Figure 15.

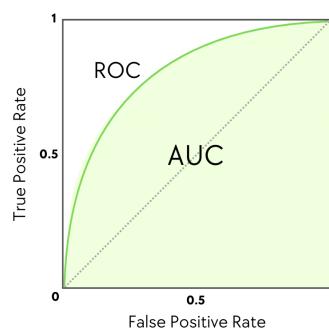


Figure 15. The AUC-ROC curve

Pada kasus identifikasi dan klasifikasi *O. celebensis* dan *O. javanicus* performa model yang kami bangun yang diukur dari hasil perbandingan antara *True Positive Rate* (TPR/Recall) dan *False Positive Rate* (FPR) menunjukkan peningkatan yang signifikan, termasuk AUC juga memperlihatkan kurva yang mendekati nilai 1 yang menggambarkan model memiliki performa yang sangat baik, artinya sistem akan memberikan jawaban dengan tingkat

akurasi yang sangat baik dalam hal identifikasi dan klasifikasi. Dalam rangkaian penelitian ini, beberapa komponen telah kami modifikasi untuk mencapai hasil yang lebih akurat, efisien dan efektif. Setelah melewati pengujian dengan **Confusion Matrix** dan Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) Model yang kami bangun mengalami peningkatan performa, model tersebut dapat kita lihat pada Figure ??.

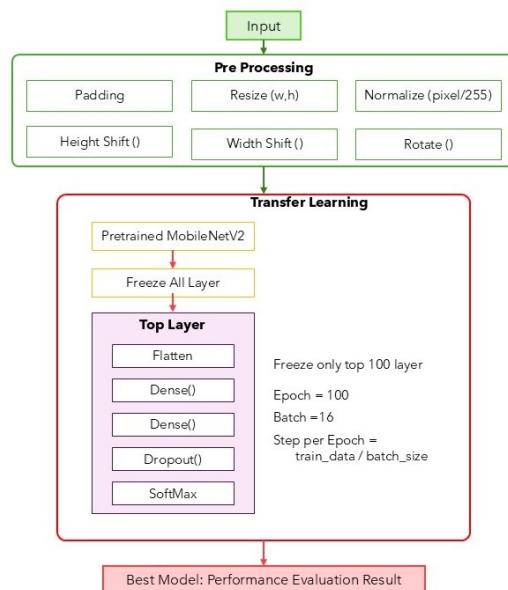


Figure 16. A pre-trained model using Transfer Learning.

Dapat dilihat tahap input atau *Pre-processing* dimana data melewati beberapa proses sebelum masuk proses identifikasi dan klasifikasi *Transfer Learning* yaitu: *Padding*, *Resize(w,h)*, *Normalization (pixel/255)*, *Height Shift*, *Width Shift*, dan *Rotate* seperti yang telah kami jelaskan sebelumnya, proses ini akan mempermudah kinerja model yang kami bangun, karena data yang melewati proses ini akan disesuaikan dengan model pre-trained yang kami bangun dalam transfer learning, sehingga model bekerja lebih ringan semua data dapat trained (dilatih) dengan baik serta menjaga konsistensi dengan meminimalisir kecacatan data dan *Transfer Learning* akan menghasilkan Pefroma yang sangat baik yang tergambar melalui nilai akurasi yang dihasilkan.

Confusion Matrix with Kronecker Delta

Untuk klasifikasi biner ($y_i, \hat{y}_i \in \{0, 1\}$), komponen confusion matrix dapat dirumuskan secara formal menggunakan fungsi delta Kronecker $\delta_{a,b}$, yaitu:

$$\delta_{a,b} = \begin{cases} 1, & \text{jika } a = b \\ 0, & \text{jika } a \neq b \end{cases}$$

Didefinisikan untuk seluruh sampel $i = 1, 2, \dots, n$, maka:

$$\begin{aligned} \text{TP} &= \sum_{i=1}^n \delta_{y_i,1} \cdot \delta_{\hat{y}_i,1}, \\ \text{FP} &= \sum_{i=1}^n \delta_{y_i,0} \cdot \delta_{\hat{y}_i,1}, \\ \text{FN} &= \sum_{i=1}^n \delta_{y_i,1} \cdot \delta_{\hat{y}_i,0}, \\ \text{TN} &= \sum_{i=1}^n \delta_{y_i,0} \cdot \delta_{\hat{y}_i,0}. \end{aligned}$$

Dengan menggunakan komponen tersebut, metrik evaluasi dapat diturunkan sebagai berikut:

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TP} + \text{TN}}{n}, \\ \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{F1-Score} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \end{aligned}$$

Evaluation Metrics using Confusion Matrix with Kronecker Delta for Multi-Class Classification

Let $y_i \in \{1, \dots, C\}$ be the ground truth label and $\hat{y}_i \in \{1, \dots, C\}$ be the predicted label for sample $i = 1, \dots, n$. Define the Kronecker delta function as:

$$\delta_{a,b} = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases}$$

The confusion matrix element at row j and column k is defined as:

$$M_{j,k} = \sum_{i=1}^n \delta_{y_i,j} \cdot \delta_{\hat{y}_i,k}$$

For each class $c \in \{1, \dots, C\}$, the evaluation components are:

$$\begin{aligned} \text{TP}_c &= M_{c,c}, \\ \text{FP}_c &= \sum_{\substack{j=1 \\ j \neq c}}^C M_{j,c}, \\ \text{FN}_c &= \sum_{\substack{k=1 \\ k \neq c}}^C M_{c,k}, \\ \text{TN}_c &= \sum_{\substack{j=1 \\ j \neq c}}^C \sum_{\substack{k=1 \\ k \neq c}}^C M_{j,k} \end{aligned}$$

The precision, recall, and F1-score for each class c are:

$$\begin{aligned}\text{Precision}_c &= \frac{TP_c}{TP_c + FP_c}, \\ \text{Recall}_c &= \frac{TP_c}{TP_c + FN_c}, \\ F1_c &= 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}\end{aligned}$$

Macro-averaged metrics:

$$\text{Macro-Precision} = \frac{1}{C} \sum_{c=1}^C \text{Precision}_c, \quad \text{Macro-Recall} = \frac{1}{C} \sum_{c=1}^C \text{Recall}_c, \quad \text{Macro-F1} = \frac{1}{C} \sum_{c=1}^C F1_c$$

Micro-averaged metrics:

$$\begin{aligned}\text{Micro-Precision} &= \frac{\sum_{c=1}^C TP_c}{\sum_{c=1}^C (TP_c + FP_c)}, \\ \text{Micro-Recall} &= \frac{\sum_{c=1}^C TP_c}{\sum_{c=1}^C (TP_c + FN_c)}, \\ \text{Micro-F1} &= 2 \cdot \frac{\text{Micro-Precision} \cdot \text{Micro-Recall}}{\text{Micro-Precision} + \text{Micro-Recall}}\end{aligned}$$

Performance Evaluation Metrics for Multi-Class Classification

Evaluasi performa dalam masalah klasifikasi multi-kelas biasanya dilakukan menggunakan *confusion matrix*, yang merupakan tabel persegi berdimensi $C \times C$, di mana C adalah jumlah kelas. Setiap elemen M_{ij} dari confusion matrix menyatakan jumlah instance dari kelas i (kelas sebenarnya) yang diklasifikasikan sebagai kelas j (kelas prediksi).

Sebagai contoh, confusion matrix untuk 3 kelas dapat divisualisasikan sebagai berikut:

Untuk mendefinisikan metrik evaluasi seperti *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN) dalam konteks multi-kelas, kita dapat menggunakan fungsi Kronecker, $\delta(x, y)$, yang didefinisikan sebagai:

$$\delta(x, y) = \begin{cases} 1, & \text{jika } x = y \\ 0, & \text{jika } x \neq y \end{cases} \quad (5)$$

Dengan notasi tersebut, maka jumlah True Positive untuk kelas k dapat dihitung sebagai:

$$TP_k = \sum_{i=1}^N \delta(y_i, k) \cdot \delta(\hat{y}_i, k)$$

di mana y_i adalah label sebenarnya, dan \hat{y}_i adalah label prediksi untuk sampel ke- i .

Sebaliknya, metrik lainnya dapat dituliskan sebagai:

$$\begin{aligned} FP_k &= \sum_{i=1}^N (1 - \delta(y_i, k)) \cdot \delta(\hat{y}_i, k) \\ FN_k &= \sum_{i=1}^N \delta(y_i, k) \cdot (1 - \delta(\hat{y}_i, k)) \\ TN_k &= \sum_{i=1}^N (1 - \delta(y_i, k)) \cdot (1 - \delta(\hat{y}_i, k)) \end{aligned}$$

Metrik Evaluasi Turunan

Dari komponen-komponen di atas, kita dapat menghitung beberapa metrik performa umum sebagai berikut:

- **Precision** untuk kelas k :

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}$$

- **Recall** untuk kelas k :

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k}$$

- **F1-Score** untuk kelas k :

$$F1_k = 2 \cdot \frac{\text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}$$

- **Accuracy** keseluruhan:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \delta(y_i, \hat{y}_i)$$

Metrik-metrik ini kemudian dapat dirata-ratakan dengan pendekatan *macro average* atau *weighted average*, tergantung pada distribusi kelas.

5. Results and Discuss

Data gambar ikan medaka yang berhasil diambil terdiri dari 661 gambar untuk *O. celebensis* dan 886 gambar untuk *O. javanicus*.

5.1. Image Pre-processing

Setelah mendapatkan dataset gambar, setiap gambar diberi padding untuk mencapai rasio aspek 1:1 dengan menambahkan piksel pada sisi yang lebih pendek menggunakan warna yang mirip dengan tepi gambar. Teknik ini jarang atau bahkan tidak pernah digunakan oleh peneliti lain, menjadikannya pendekatan yang unik dalam penelitian ini. Setelah menyesuaikan gambar ke rasio 1:1, gambar kemudian diubah ukurannya menjadi 224 x 224 piksel. Hasilnya dapat dilihat pada Gambar 8.

5.2. Experiment Results

Perbandingan kinerja setiap model pada dataset yang dipadding dan tidak dipadding disajikan dalam Tabel 3, yang menyoroti perbedaan akurasi dan efektivitas antara kedua pendekatan tersebut.

Table 3. Accuracy of padding and non-padding images.

| Image | Model | Sensitivity | Precision | F1 Score | Accuracy |
|-------------|---------------|--------------|--------------|--------------|--------------|
| Non-padding | MobileNetV2 | 98.2 | 93.8 | 96 | 96.95 |
| padding | MobileNetV2 | 96.3 | 87.6 | 91.93 | 93.9 |
| Non-padding | P-MobileNetV2 | 79 | 98.4 | 87.6 | 89 |
| padding | P-MobileNetV2 | 98.46 | 98.46 | 98.46 | 98.78 |
| Non-padding | VGG16 | 95.3 | 93.5 | 94.5 | 95.7 |
| padding | VGG16 | 97.2 | 96.2 | 81 | 87 |
| Non-padding | P-VGG16 | 63.3 | 98.4 | 77 | 76 |
| padding | P-VGG16 | 92.7 | 98.4 | 96.3 | 96.3 |

P-MobileNetV2 mengacu pada model yang dilatih menggunakan dataset yang telah dipad (padded dataset), di mana gambar disesuaikan ke rasio aspek 1:1 untuk meningkatkan performa klasifikasi dan konsistensi selama pelatihan. Kami telah memperoleh hasil grafis yang membandingkan akurasi ROC antara MobileNetV2 dengan P-MobileNetV2 yang telah dimodifikasi, bersama dengan VGG16 dan P-VGG16 yang juga dimodifikasi. Perbandingan ini ditunjukkan dengan jelas dalam grafik berikut:

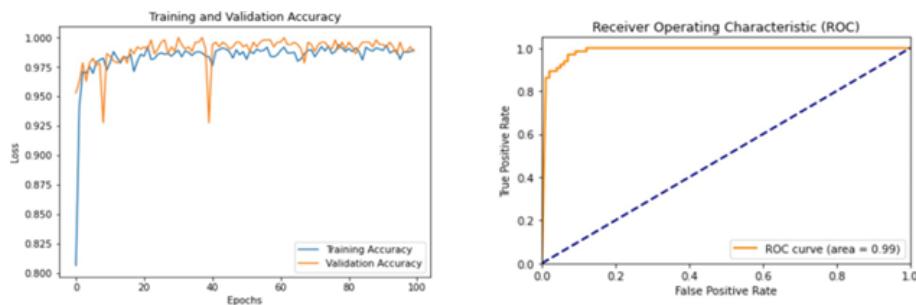
**Figure 17.** Diagram Accuracy dan ROC Model MobileNetV2.

Diagram Akurasi dan ROC dari model MobileNetV2 menggambarkan performanya dalam tugas klasifikasi. Grafik akurasi menunjukkan perkembangan pembelajaran model seiring waktu, sedangkan kurva ROC (Receiver Operating Characteristic) mengevaluasi kemampuannya dalam membedakan antar kelas, dengan menonjolkan sensitivitas dan spesifisitas pada berbagai ambang batas.

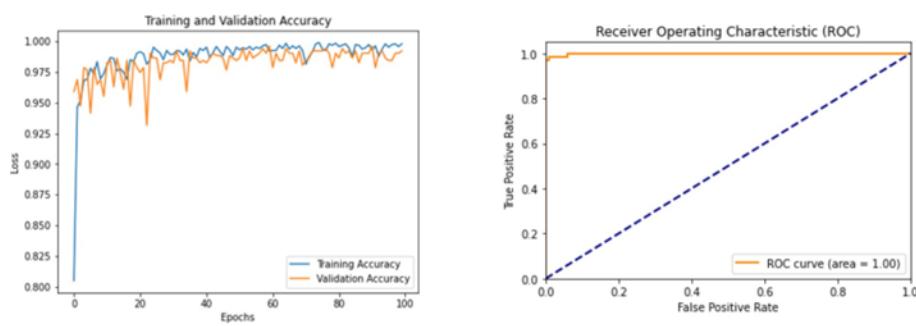
**Figure 18.** Diagram Accuracy dan ROC Model P-MobileNetV2.

Diagram 'Akurasi dan ROC' dari Model P-MobileNetV2 menggambarkan metrik kinerja model yang telah dimodifikasi, termasuk akurasi klasifikasi dan kurva Receiver Operating Characteristic (ROC), untuk mengevaluasi kemampuan prediksinya. **Diagram Akurasi dan ROC Model P-MobileNetV2** menggambarkan metrik kinerja model yang telah dimodifikasi, termasuk akurasi klasifikasi dan kurva Receiver Operating Characteristic (ROC), yang mengevaluasi kemampuan prediktifnya.

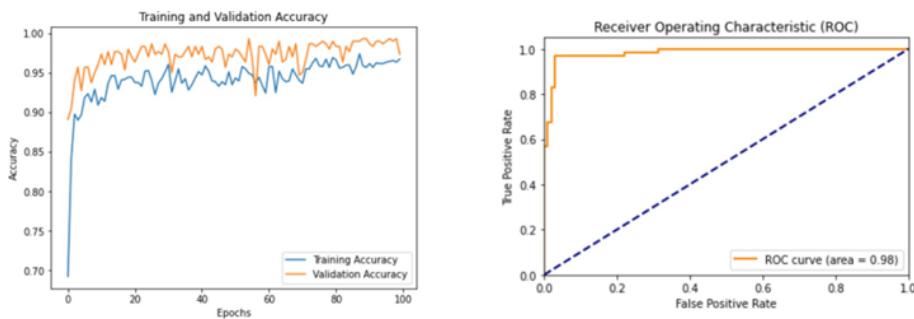


Figure 19. Diagram Accuracy dan ROC Model VGG16.

Diagram ini menampilkan akurasi dan kurva Receiver Operating Characteristic (ROC) dari model VGG16, yang menggambarkan performa klasifikasinya serta kemampuannya dalam membedakan antar kelas berdasarkan tingkat positif benar (true positive rate) dan tingkat positif salah (false positive rate) selama proses evaluasi.

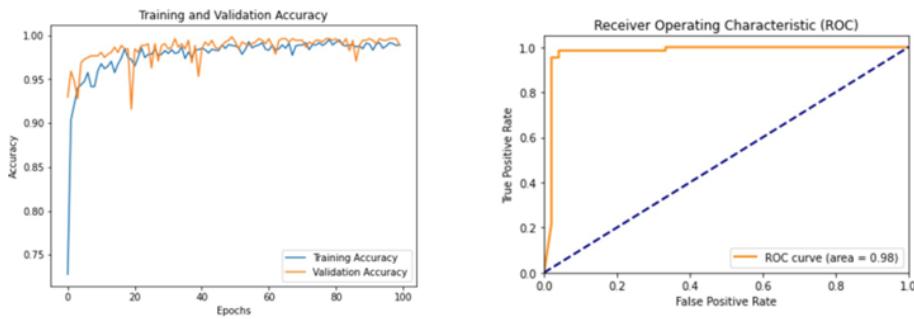


Figure 20. diagram Accuracy dan ROC Model P-VGG16.

Diagram ini menampilkan akurasi dan kurva Receiver Operating Characteristic (ROC) dari model P-VGG16 yang telah dimodifikasi, menyoroti peningkatan performa klasifikasi serta kemampuan yang lebih baik dalam membedakan antar kelas berdasarkan tingkat positif benar (true positive rate) dan tingkat positif salah (false positive rate) selama fase evaluasi.

5.3. Discussion

Dalam penelitian ini, kami berhasil mengembangkan model arsitektur MobileNetV2 menggunakan transfer learning dan dataset padding untuk mengklasifikasikan *O. celebensis* dan *O. javanicus*. Kami menambahkan beberapa lapisan pada Classification Layer yang terdiri dari 5 lapisan: Flatten, dua lapisan Dense dengan fungsi aktivasi ReLU (masing-masing 1024 dan 512 neuron), lapisan Dropout dengan rate 0.2, serta lapisan Dense akhir dengan dua neuron menggunakan aktivasi SoftMax. Modifikasi ini memungkinkan kami memanfaatkan fitur yang telah dipelajari oleh MobileNetV2 dengan bobot ImageNet sekaligus mengadaptasi model untuk tugas klasifikasi spesifik kami.

Hasil penelitian menunjukkan bahwa pendekatan ini secara efektif meningkatkan kinerja klasifikasi. Lapisan dense tambahan berfungsi sebagai lapisan ekstraksi fitur dan klasifikasi, yang memetakan fitur-fitur yang diekstraksi oleh MobileNetV2 ke dalam dua kelas target yaitu *O. celebensis* dan *O. javanicus*. Lapisan Dropout (dengan rate 0.2) membantu mencegah overfitting, memastikan model tetap memiliki generalisasi yang baik pada data baru. Fungsi aktivasi SoftMax pada lapisan dense akhir memastikan output model dapat diinterpretasikan sebagai probabilitas kelas, sehingga menyederhanakan interpretasi hasil akhir. Struktur ini meningkatkan akurasi sekaligus mempertahankan ketangguhan, membuat model menjadi andal untuk membedakan kedua spesies ikan tersebut. Diskusi lebih lanjut mengungkapkan bahwa penggunaan MobileNetV2 sebagai model dasar menawarkan beberapa keunggulan:

- **MobileNetV2 secara khusus dirancang untuk perangkat mobile**, menjadikannya ringan dan efisien untuk penerapan dalam aplikasi seluler atau lingkungan dengan sumber daya terbatas.
- **Dengan memanfaatkan dataset padding dan transfer learning**, kami memanfaatkan pengetahuan pra-latih MobileNetV2 dari dataset berskala besar, yang secara signifikan mempercepat dan menyederhanakan proses pelatihan model. Pendekatan ini memungkinkan model mencapai akurasi tinggi dengan sumber daya komputasi terbatas.

Selain itu, lapisan dense yang ditambahkan meningkatkan ekstraksi fitur dan klasifikasi, sementara lapisan dropout memastikan ketangguhan terhadap overfitting. Fungsi aktivasi SoftMax pada lapisan akhir memberikan keluaran probabilitas yang dapat diinterpretasikan untuk setiap kelas target (*Oryzias celebensis* dan *Oryzias javanicus*). Arsitektur lengkap model yang dikembangkan ditunjukkan pada Gambar 21.

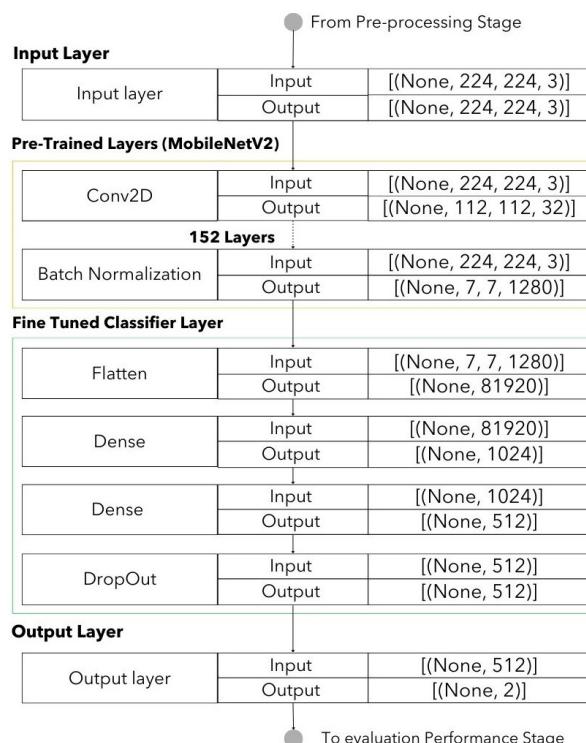


Figure 21. Modified MobileNetV2 Architecture.

Kami telah mengevaluasi kinerja arsitektur MobileNetV2 menggunakan lapisan tambahan yang dikonfigurasi seperti dijelaskan dalam tinjauan sistem, serta menerapkan padding pada dataset untuk mengklasifikasikan spesies ikan *O. celebensis* dan *O. javanicus* dengan dataset yang terbatas. Evaluasi dilakukan menggunakan pengujian berbasis matriks kebingungan (confusion matrix) dan kurva ROC-AUC, dengan memanfaatkan dua jenis dataset: dataset tanpa padding dan dataset dengan padding. Dua arsitektur model digunakan untuk perbandingan, yaitu MobileNetV2 dan VGG16. Hasilnya menunjukkan bahwa MobileNetV2 berkinerja lebih baik pada dataset dengan padding dibandingkan kinerjanya pada dataset tanpa padding, serta lebih unggul dibandingkan VGG16 pada kedua jenis dataset tersebut.

6. Conclusions

Berdasarkan Tabel 2, model P-MobileNetV2 menunjukkan hasil evaluasi kinerja yang lebih baik dibandingkan model lainnya, dengan Sensitivitas = 98.46%, Presisi = 98.46%, F1 Score = 98.46%, dan Akurasi = 98,7%. Dalam penelitian ini, kami juga menerapkan teknik padding pada dataset kami. Teknik ini melibatkan penambahan nilai batas buatan untuk

mengisi ruang kosong, memastikan gambar tetap berbentuk persegi (rasio aspek 1:1), yang membantu mempertahankan informasi spasial selama proses konvolusi. Hasilnya, model P-MobileNetV2 dan P-VGG16 yang dilatih pada dataset dengan padding mencapai metrik yang lebih tinggi dibandingkan model yang dilatih tanpa padding dataset, membuktikan bahwa teknik ini dapat secara efektif meningkatkan kinerja model. Hal ini dapat diamati pada Gambar 10 dan 12, di mana kedua grafik menunjukkan tren yang secara signifikan lebih baik dibandingkan Gambar 9 dan 11." 704
705
706
707
708
709
710

Lebih lanjut, penelitian kami memberikan wawasan bahwa teknik padding tidak hanya meningkatkan akurasi model tetapi juga membantu menjaga konsistensi dan mempertahankan konteks spasial pada gambar dataset. Hal ini menjadi sangat krusial untuk tugas-tugas seperti segmentasi dan deteksi objek, di mana informasi spasial dalam gambar atau dataset memiliki tingkat kepentingan yang tinggi. Oleh karena itu, penerapan teknik padding dalam penelitian ini terbukti menjadi langkah kritis dalam meningkatkan kinerja model dan mencapai hasil yang kuat dalam tugas klasifikasi kami. 711
712
713
714
715
716
717

Author Contributions: Conceptualization, A.L. and M.K.; methodology, A.L.; software, A.L. and M.K.; validation, A.L., I.A., A.I.B., and M.K.; formal analysis, A.L., I.A., and A.I.B.; investigation, I.A., and A.I.B; resources, A.L. and I.A.; data curation, A.L., I.A., and A.I.B; writing-original draft preparation, A.L. and M.K.; writing-review and editing, A.L., I.A and M.K.; visualization, A.L.; supervision, M.K.; project administration, A.L., and I.A.; funding acquisition, A.L. All authors have read and agreed to the published version of the manuscript. 718
719
720
721
722
723

Acknowledgments: The author sincerely thanks Hasanuddin University for its generous support, which helped make this research possible and enabled it to be published in an international journal. 724
725

Conflicts of Interest: The author declares no conflict of interest in preparing this paper, including any personal or institutional relationships that could have influenced the content, analysis, or conclusions presented herein. 726
727
728

References

1. Parenti, L.R. A phylogenetic analysis and taxonomic revision of ricefishes, *Oryzias* and relatives (Beloniformes, Adrianichthyidae). *Zoological Journal of the Linnean Society* **2008**, *154*, 494–610. 730
731
2. Herder, F.; Möhring, J.; Flury, J.M.; Utama, I.V.; Wantania, L.; Wowor, D.; Boneka, F.B.; Stelbrink, B.; Hilgers, L.; Schwarzer, J.; et al. More non-native fish species than natives, and an invasion of Malawi cichlids, in ancient Lake Poso, Sulawesi, Indonesia. *Aquatic Invasions* **2022**, *17*, 72–91. 732
733
734
3. Nurdin, N.; et al. PERUBAHAN HISTOPATOLOGI SISTEM RESPIRASI IKAN MEDAKA SULAWESI (*Oryzias celebensis*) AKIBAT KONTAMINASI AMOXICILLIN. PhD thesis, Universitas Hasanuddin, 2023. 735
736
4. Kottelat, M. The fishes of the inland waters of Southeast Asia: a catalogue and core bibliography of the fishes known to occur in freshwaters, mangroves and estuaries. *Raffles Bulletin of Zoology* **2013**. 737
738
5. Magtoon, W.; TERMVIDCHAKORN, A. A Revised Taxonomic Account of Ricefish *Oryzias* (Beloniformes; Adrianichthyidae), in Thailand, Indonesia and Japan. *Tropical Natural History* **2009**, *9*, 35–68. 739
740
6. Parenti, L.R. Endemism and conservation of the native freshwater fish fauna of Sulawesi, Indonesia. In Proceedings of the Proceedings of the VIth National Seminar and 3rd Congress of the Indonesian Society of Ichthyologists. Indonesian Society of Ichthyologists, 2011. 741
742
743
7. ERAGRADHINI, A.R.; et al. Ecobiology and reproduction of the Matano medaka fish *Oryzias matanensis* (Aurich, 1935) in Lake Towuti, South Sulawesi. PhD thesis, Universitas Hasanuddin, 2020. 744
745
8. Serdiati, N.; Arfiati, D.; Widodo, M.S.; Lelono, T.D.; Ndobe, S.; Mansyur, K.; Moore, A.M. Perspectives on sustainable management of the Poso Lake (Indonesia) endemic ricefish, *Oryzias nigrimas* (Actinopterygii: Adrianichthyidae). *Revista de Biología Tropical* **2021**, *69*, 139–152. 746
747
748
9. Hasan, V.; Mamat, N.B.; South, J.; Ottoni, F.P. A checklist of native freshwater fish from Brantas River, East Java, Indonesia. *Biodiversitas* **2022**, *23*, 6031–6039. 749
750
10. Mokodongan, D.F.; Yamahira, K. Origin and intra-island diversification of Sulawesi endemic Adrianichthyidae. *Molecular Phylogenetics and Evolution* **2015**, *93*, 150–160. <https://doi.org/https://doi.org/10.1016/j.ympev.2015.07.024>. 751
752
11. Voss, R.; Quaas, M.F.; Schmidt, J.O.; Tahvonen, O.; Lindegren, M.; Möllmann, C. Assessing social–ecological trade-offs to advance ecosystem-based fisheries management. *PloS one* **2014**, *9*, e107811. 753
754
12. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>. 755
756
757

13. Akay, M.; Du, Y.; Sershen, C.L.; Wu, M.; Chen, T.Y.; Assassi, S.; Mohan, C.; Akay, Y.M. Deep learning classification of systemic sclerosis skin using the MobileNetV2 model. *IEEE Open Journal of Engineering in Medicine and Biology* **2021**, *2*, 104–110. 758
759
14. Golcuk, A.; Yasar, A.; Saritas, M.M.; Erharman, A. Classification of Cicer arietinum varieties using MobileNetV2 and LSTM. *European Food Research and Technology* **2023**, *249*, 1343–1350. 760
761
15. Sen, S.; Sawant, K. Face mask detection for covid_19 pandemic using pytorch in deep learning. In Proceedings of the IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021, Vol. 1070, p. 012061. 762
763
16. Gulzar, Y. Fruit image classification model based on MobileNetV2 with deep transfer learning technique. *Sustainability* **2023**, *15*, 1906. 764
765
17. Indraswari, R.; Rokhana, R.; Herulambang, W. Melanoma image classification based on MobileNetV2 network. *Procedia computer science* **2022**, *197*, 198–207. 766
767
18. Shahoveisi, F.; Taheri Gorji, H.; Shahabi, S.; Hosseinirad, S.; Markell, S.; Vasefi, F. Application of image processing and transfer learning for the detection of rust disease. *Scientific Reports* **2023**, *13*, 5133. 768
769
19. Hamid, Y.; Wani, S.; Soomro, A.B.; Alwan, A.A.; Gulzar, Y. Smart seed classification system based on MobileNetV2 architecture. In Proceedings of the 2022 2nd international conference on computing and information technology (iccit). IEEE, 2022, pp. 217–222. 770
771
20. Karnadi, B.; Handhayani, T. Klasifikasi Jenis Buah dengan Menggunakan Metode MobileNetv2 dan Inceptionv3. *Jurnal Eksplora Informatika* **2024**, *14*, 35–42. 772
773
21. Utomo, A.A.; Mulyono, S. Comparison of MobileNetV2 and DenseNet121 for Coral Reef Classification Using Convolutional Neural Networks. *Jurnal Rekayasa Sistem Informasi dan Teknologi* **2025**, *2*, 930–941. 774
775
22. Maulana, S.A.; Batubara, S.H.; Pasaribu, Y.P.P.; Syahputra, H.; Ramadhani, F. Bird Detection Using Convolutional Neural Network (CNN) with MobileNetV2 Architecture. *JATI (Jurnal Mahasiswa Teknik Informatika)* **2024**, *8*, 6108–6114. 776
777
23. Talib, M.; Al-Noori, A.H.; Suad, J. YOLOv8-CAB: Improved YOLOv8 for Real-time object detection. *Karbala International Journal of Modern Science* **2024**, *10*, 5. 778
779
24. Panja, E.; Hendry, H.; Dewi, C. YOLOv8 Analysis for Vehicle Classification Under Various Image Conditions. *Scientific Journal of Informatics* **2024**, *11*, 127–138. 780
781
782
783
784

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.