International Journal of Emerging Information Science and Technology
Vol. 6, No. 1, (2025), pp. 43-53

Implementation Of Transfer Learning In Cat Breed Detection Using Web-Based Convolutional Neural Network (CNN)

Furqon Fitrianto[1*], Erni Rouza[1], Basorudin[1]

[1] *Program Studi Teknik Informatika, Fakultas Ilmu Komputer,Universitas Pasir Pengaraian, Riau, Indonesia*

**Corresponding author: furqonfitrianto@gmail.com*

Abstract

Cats are one of the most popular pets because of their friendly and adorable nature. Along with the increasing cat population in Indonesia and the variety of breeds, cat breed identification is a challenge in itself, especially for cat lovers to the animal conservation community. This research aims to develop an image-based cat breed classification system using transfer learning method with MobileNetV2 Convolutional Neural Network (CNN) model. This model was chosen because of its ability to produce high accuracy with good computational efficiency, making it suitable for use on devices with limited resources. The dataset used consists of 13,000 training images and 3,250 testing images of 13 cat breeds. The model training process was carried out up to 50 epochs with the addition of fine-tuning for 10 epochs, after previously terminating the process at the 60th epoch, resulting in a validation accuracy of 98.67%. Model performance testing also showed high average results of evaluation metrics, namely precision of 91.38%, recall of 91.39%, and F1-score of 91.33%. Based on these results, it can be concluded that the application of MobileNetV2 transfer learning is able to classify cat breeds accurately and efficiently. The website made makes it easy for users to recognize cat breeds by simply uploading images, making it very useful for the general public, professionals, and cat enthusiasts.

Keywords: *CNN, Cat Breed Classification, Confusion Matrix, MobileNetV2, Transfer learning*

1. Introduction

Cats are carnivorous mammals that belong to the Felidae family with the scientific name Felis silvestris catus or Felis catus. These land mammals generally coexist with humans as pets, although not a few live wild in nature[1]. Cats are one of the most popular pets because they are cute, adorable, and able to create emotional bonds with their owners. It is not uncommon for cats to be considered as part of the family by most people. Based on data, the cat ownership rate in Indonesia reaches 37%, much higher than dog ownership which is only 16%[2][3]. The growth of the cat population in Indonesia has also increased significantly. According to the President Director of PT Uni-Charm Indonesia, the pet cat population increased by 129% from 2017 to 2021[4]. Cats are capable of reproducing three to four times a year. This can cause the cat population to increase rapidly[5]. Various cat breeds such as Persian, Maine Coon, Siamese, Ragdoll, and others are usually bred in authorized breeding facilities. However, purebred cats make up only about 1% of the entire cat population in the world, while the rest are mixed-breed cats such as domestic or feral

ISSN: 2722-6050

cats[1]. With such a large variety of breeds, the challenge facing society is to accurately recognize the type of cat breed, especially for those involved in the fields of breed maintenance, animal health, and preservation[4]. To overcome these challenges, Deep Learning-based image recognition technology is one potential solution. Convolutional Neural network (CNN) methods have proven to be reliable in image classification and pattern recognition[6][7]. By utilizing transfer learning models such as MobileNetV2, systems can be developed to detect cat breeds more efficiently and accurately[8][9]. Transfer learning is a method in machine learning that utilizes pre-trained models to solve new tasks with similar characteristics[10][11]. With transfer learning, the model training process becomes faster and more efficient, as the model has already "learned" common features from previous datasets[12]. There are two main approaches in transfer learning, namely Fine Tuning and Feature Extraction[13][14]. This research aims to implement transfer learning technology using MobileNetV2 to detect 13 popular cat breeds in Indonesia, namely American Shorthair, Bengal, Bombay, British Shorthair, Himalayan, Maine Coon, Manx, Persian, Ragdoll, Russian Blue, Scottish Fold, Siamese, and Sphynx. This system is designed to be web-based so that it can be easily accessed by the cat lover community, veterinarians, and related agencies. It is hoped that this technology can help in the preservation, maintenance, and automatic identification of cat breeds.

2. Method

This research uses a quantitative experimental approach with the aim of developing an image-based cat breed classification system using transfer learning with MobileNetV2 architecture integrated into a web application.

2.1 Data Collection and Processing

At this stage, data collection and processing are carried out which will be used to train the transfer learning model related to research and website development, namely:

1. Data Type: Images of 13 popular cat breeds in Indonesia, namely: American Shorthair, Bengal, Bombay, British Shorthair, Himalayan, Maine Coon, Manx, Persian, Ragdoll, Russian Blue, Scottish Fold, Siamese, and Sphynx.
2. Data Source: Dataset obtained from the Kaggle website with the keyword "Cat breeds".
3. Data Processing: The dataset that has been obtained is then divided into train and validation sets, with a ratio of 80:20.

3. Results and Discussion

3.1 Input Layer

Input layer is the first layer in modeling which functions to receive input data in the form of images. The images used in this study have standard dimensions of 224x224x3 pixels with 3 color channels (RGB). The input and output values of this layer are 224x224 with matrix values [-1,0,1]. This size matches the requirements of the MobileNetV2 model, which is at the core of the architecture[15].

3.2 MobileNetV2

MobileNetV2 MobileNetV2 is one of the CNN models available on the hard with high accuracy. MobileNetV2 is a variant of neural network architecture designed with a focus on computational efficiency and speed, especially for devices with limited resources. MobileNetV2 is very suitable for use on mobile devices due to the very small model size and low latency rate[16][17]. The convolution layer in MobileNetV2 uses a filter thickness that is adjusted to the thickness of the input image. MobileNetV2 implements depthwise convolution, pointwise convolution, linear bottlenecks, and shortcut connections between bottlenecks to improve the efficiency and performance of the model in processing data[18][19].

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Figure 1. MobileNetV2 Pretrained Transfer Learning Architecture Diagram (Source: Nafisa et al., 2023)

3.3 Convolution Layer

Convolutional layer is an image manipulation process using an external mask or subwindow to generate a new image by changing the image dimensions through convolution operation. This process aims to extract important features from the image such as edge detection, color characteristics, gradient orientation, and other features carried out through the encoding process[20][21]. After adding the MobileNetV2 model, the next step is convolution. At this stage, convolution uses 32 filters, with a 3x3 kernel size, padding same (0), ReLu activation function, and stride 1. After adding the MobileNetV2 model, the next step is convolution. At this stage, convolution uses 32 filters, with a 3x3 kernel size, padding same (0), ReLu activation function, and stride 1.

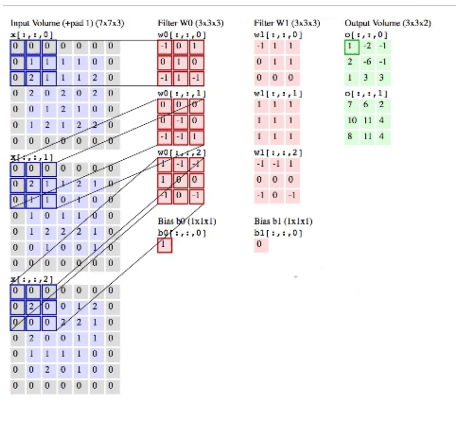

Figure 2. Convolution process (Source: Peryanto et al., 2020)

The spatial dimension of the output of the convolution process can be calculated using the following equation (1):

$$\frac{(N-F+2P)}{S}+1 \quad (1)$$

N= The spatial dimensions (height H1 and width W1) of the input image.

F= The spatial size of the filter (e.g., 3x3).

P= The amount of padding, usually zero for "valid" or can use a larger padding for "same".

S= The number of filter shifts in each calculation process (stride).

The total convolutional calculation with negative results is converted to 0 using ReLu. The ReLu (rectified linear unit) function is a non-linear function where the activation of neurons is not done simultaneously, and only when the output of the linear transformation is zero[22]. In use, the ReLu function is written using equation (2):

$$f(x) = max(0,x) \quad (2)$$

x= Input data value,
f(x)= The output of the ReLu function which produces values in the form of 0 and 1. This process is done repeatedly for each image using 32 different types of filters, so that the output of this convolution stage produces various feature maps.

3.4 Max Pooling

Max pooling is a layer that takes the feature map of the convolutional layer as input to reduce the spatial size, thus reducing the computational resources required in data processing[23]. Pooling consists of two main types, namely max pooling and average pooling[21][24]. The result of the convolution stage will be used as input to max pooling. In this stage, max pooling is used with a 2x2 kernel size, stride 2, which receives the output from the convolution stage.

3.5 Dropout Layer

Dropout serves to reduce the complexity of the model that has been built by ignoring some neurons that are not used during the training process[21].

3.6 Flatten Layer

Flattening, also known as the streamlining stage, serves to convert the output of max pooling into a vector[25][26]. Flatten is a stage that converts the matrix into a one-dimensional vector form. The output of the dropout layer will produce a new vector. The result of this stage is a one-dimensional vector which is then used as input to the fully connected layer (dense)[25][27].

3.7 Fully Connected Layer

The output of the flatten layer stage is a one-dimensional vector that will be input at this stage[21]. The final stage of the CNN model created is a dense layer with a softmax activation function. This process can be calculated using equations (3)(4)(5).

The final stage of the CNN model built is a dense layer with a SoftMax activation function. The following is the calculation of each hidden layer.

$$\sum_{i=1}^{N} I_1 * v_{ij} = J_i \quad (3)$$

$$\sum_{i=1}^{N} J_1 * w_{ij} = H_i \quad (4)$$

$$\sum_{i=1}^{N} H_1 * x_{ij} = O_i \quad (5)$$

The results of the above calculations will produce a value as a hidden layer, the next step is the calculation of softmax. The softmax function is written using equation (6).

$$\frac{e^{O_i}}{\sum_{j=1}^{n} e^{O_j}} \quad (6)$$

So the results of the above calculations produce a greater probability weight which means that the input image is successfully predicted correctly.

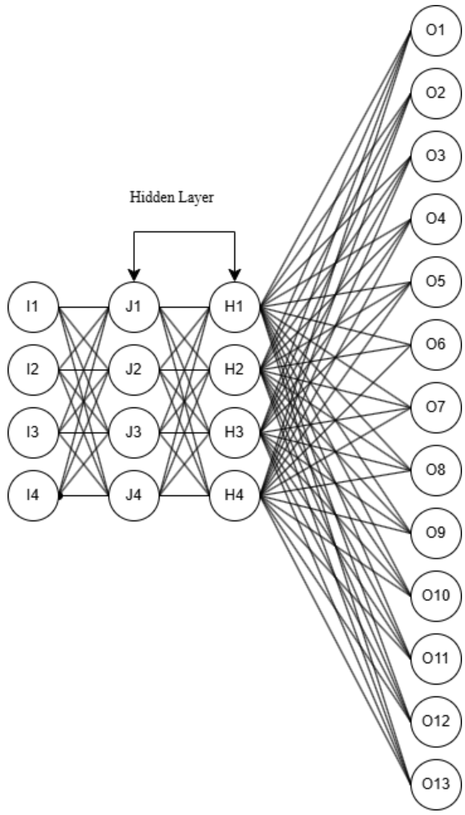

Figure 1. Illustration of Dense and Softmax Processes

3.8 Model Validation Testing

Model validation testing includes several aspects such as the number of epochs, the level of accuracy, and how long it takes (time) to train the model every 10 epochs.

Table 1. Model Validation Testing

Epoch	*Accuracy*	*Time*
10	79,19%	*25 minutes and 13 seconds*
20	82,42%	*49 minutes and 12 seconds*
30	85,06%	*77 minutes and 14 seconds*
40	86,22%	*105 minutes and 53 seconds*
50	87,31%	*127 minutes and 34 seconds*
60	*RUNTIME DISCONNECT*	

In Table 1 above, it can be concluded that there is an increase in the amount of accuracy and the amount of time required for every 10 epochs and at epoch 60 there is a runtime disconnection which results in stopping the model training process, so that the successful model at epoch 50 can be used for the fine-tuning stage using epoch 10.

3.9 Model Validation Testing Results

The model that has been trained in the previous stage is then tested to evaluate its performance. Measurement of model performance after the training stage uses Confusion Matrix. Confusion Matrix is a commonly used technique to calculate the level of accuracy in data mining[28]. This technique serves as a visual assessment instrument utilized in machine learning. Model performance measurement using Confusion Matrix is as follows[29]: *Accuracy* represents the proportion of correct predictions made by the model relative to the total number of test samples.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \tag{7}$$

Precision refers to the proportion of true positive predictions compared to the total number of positive predictions produced by the model.

$$Precision = \frac{True\ Positive}{False\ Positive + True\ Positif} \tag{8}$$

Recall is defined as the proportion of true positive predictions to the total number of actual positive instances in the dataset.

$$Recall = \frac{True\ Positive}{False\ Negative + True\ Positive} \tag{9}$$

Furthermore, the F1-Score value is calculated as a weighted average between Precision and Recall[25][29][30].

$$F1\text{-}Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

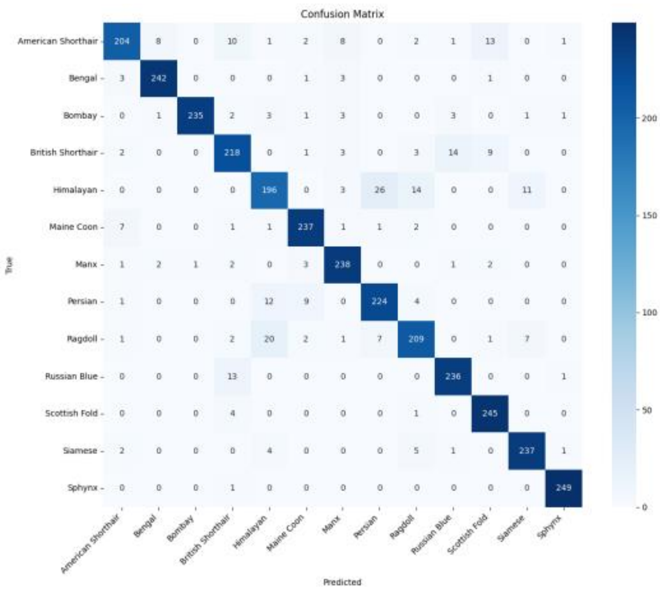

Figure 2. Confusion Matrix

Table 2. Confusion Matrix Calculation Results

No	Name	*Accuracy*	*Precision*	*Recall*	*F1-Score*
1	*American Shorthair*	98,06%	81,60%	92,31%	86,62%
2	*Bengal*	99,42%	96,80%	95,65%	96,22%
3	*Bombay*	99,51%	94,00%	99,58%	96,71%
4	*British Shorthair*	97,94%	87,20%	86,17%	86,68%
5	*Himalayan*	97,08%	78,40%	82,70%	80,49%
6	*Maine Coon*	99,02%	94,80%	92,58%	93,68%
7	*Manx*	98,95%	95,20%	91,54%	93,33%
8	*Persian*	98,15%	89,60%	86,82%	88,19%
9	*Ragdoll*	97,78%	83,60%	87,08%	85,31%
10	*Russian Blue*	98,95%	94,40%	92,19%	93,28%
11	*Scottish Fold*	99,05%	98,00%	90,41%	94,05%
12	*Siamese*	99,02%	94,80%	92,58%	93,68%
13	*Sphynx*	99,85%	99,60%	98,42%	99,01%
	Average	98,67%	91,38%	91,39%	91,33%

Based on the data presented in Table 2, it can be inferred that the CNN model demonstrates a high level of performance. The model achieves an average accuracy of 98.67%, indicating its effectiveness in correctly classifying data. Precision, which reflects the accuracy of positive predictions, attains an average of 91.38%. Recall, measuring the model's ability to identify relevant instances, reaches an average of 91.39%. Lastly, the F1-Score, which balances Precision and Recall, shows an average value of 91.33%.

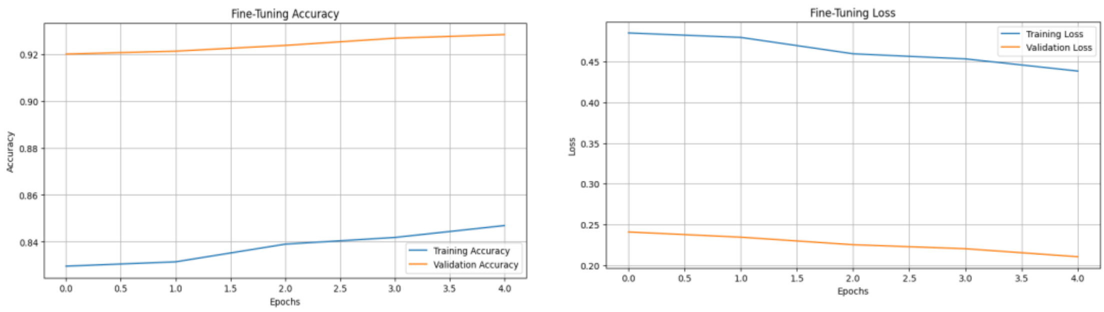

Figure 3. Fine Tuning Model Graph

Figure 5 illustrates the fine-tuning results after 10 epochs. The Fine-Tuning Accuracy graph indicates an improvement, with Training Accuracy rising from 83% and Validation Accuracy from 92%. Concurrently, the Fine-Tuning Loss graph demonstrates a reduction, as Training Loss decreased from 49% and Validation Loss from 25%.

3.10 Model Validation Testing Conclusion

Based on the results of model validation testing that has been carried out, it can be concluded that for training the MobileNetV2 transfer learning model with training data as many as 13,000 images and test data as many as 3,250 images from 13 classes can use epoch 50 and can be Fine-Tuned using epoch 10 resulting in accuracy at 98.67% and seen in the model graph shows an increase in percentage for Validation

Accuracy starting from 92% and a decrease in percentage for Validation Loss starting from 25%.

Software Implementation

The steps taken in this stage include:

1. The system was created using the python programming language and is hosted using Streamlit via Github.
2. The MobileNetV2 transfer learning model was pre-trained with a dataset consisting of 13,000 images for training data and 3,250 images for test data taken from Kaggle.

Implementation Environment

The implementation environment of this website includes two main aspects, namely hardware and software. The specifications of the implementation environment are as follows:

a. Hardware, among others:
   Prosesor : AMD Ryzen 3 3250U
   Memory : 12288MB RAM
b. Software, including:
   Operating System : Windows 11 Home
   Tools : Visual Studio Code, Web Browser Google Chrome, Github, Streamlit

1. Implementation Results

The main feature of this system is the application of the (CNN) method to detect cat breeds. By using CNN, the system can accurately detect cat breeds provided or uploaded by users. The system is supported with various image formats ranging from PNG, JPEG, JPG, to WEBP which allows users to upload cat images with various formats. By utilizing the CNN method, the system can provide cat breed classification results, showing both the detected breed types and the accuracy level of the provided images compared to what the system identifies. This is useful for determining the breed of a cat you own or one you'd like to identify.

4. Conclusion

Based on the research and testing conducted, it can be concluded that the application of transfer learning using the MobileNetV2 architecture for cat breed classification through a web-based CNN system has been successfully implemented. The validation results from training the model with 13.000 training images and 3.250 testing images across 13 cat breed categories demonstrated a consistent improvement in accuracy and training duration every 10 epochs. However, the training process was interrupted at epoch 60 due to a runtime disconnection. The model that was successfully trained up to epoch 50 was then fine-tuned for an additional 10 epochs, resulting in a validation accuracy of 98.67%. The performance graph indicated an upward trend in validation accuracy starting from 92%, along with a reduction in validation loss beginning at 25%. The MobileNetV2-based CNN model proved effective in recognizing cat breeds from images accurately and efficiently. Through this web-based application, users such as cat enthusiasts, veterinarians, or the general public can independently identify cat breeds with a high degree of accuracy. While the implementation of MobileNetV2-based transfer learning for cat breed classification through a CNN algorithm on a web platform has produced promising outcomes, several areas for improvement remain to enhance its overall performance and functionality. One key recommendation is to expand the variety of cat breeds recognized by the model, allowing for a broader and more comprehensive classification capability. Additionally, enriching the website with supplementary features—such as detailed information and representative images for each breed—would improve the user experience and educational value. Lastly, migrating the system to a premium hosting service could offer better storage capacity and performance scalability. By incorporating these improvements, the cat breed detection platform can become more robust, informative, and capable of serving a wider audience.

References

[1] A. N. Ramadhayani and V. Lusiana, "klasifikasi jenis kucing menggunakkan algoritma principal component analysis dan k-nearest neighbor," *Gener. J.*, vol. 10, no. 2, pp. 65–72, 2022, doi: 10.29407/gj.v6i2.17777.

[2] M. I. Rahayu, Faiqunisa, and Nugraha, "klasifikasi ras kucing menggunakan metadata dataset kaggle dengan framework yolo v5," no. August, 2023, doi: 10.58761/jurtikstmikbandung.v12i1.179.

[3] I. Sukma and M. Petrus, "Sistem Pakar Penyakit Kucing Menggunakan Metode Forward Chaining Berbasis Web," *Sist. Pakar Penyakit Kucing Menggunakan Metod. Forw. Chain. Berbas. Web*, vol. 5, no. 1, 2020.

[4] D. F. Ramadhoni, L. P. Abadi, and S. Suaedah, "implementasi metode forward chaining pada sistem pakar dalam mendiagnosa penyakit kucing," *JRKT (Jurnal Rekayasa Komputasi Ter.*, vol. 3, no. 03, pp. 111–117, 2023, doi: 10.30998/jrkt.v3i03.9374.

[5] H. Kurniati, Marnita, and A. Apriliany, "Upah Jasa Sterilisasi pada Kucing dalam Rangka Menekan Jumlah Populasi Kucing Prespektif Hukum Islam," *Asas J. Huk. Ekon. Syariah*, vol. 12, no. 2, pp. 97–112, 2020.

[6] R. Gunawan, D. M. I. Hanafie, and A. Elanda, "klasifikasi jenis ras kucing dengan gambar menggunakan convolutional neural network (CNN)," *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 18, no. 4, pp. 1–8, 2024, doi: 10.35969/interkom.v18i4.318.

[7] M. A. A. Fawwaz, K. N. Ramadhani, and F. Sthevanie, "klasifikasi ras pada kucing menggunakan algoritma convolutional neural network," vol. 8, no. 1, pp. 715–730, 2020.

[8] E. Rouza, F. Arifin, and Suprapto, "Essential Advances in Soil-Transmitted Helminth Detection Using Machine Learning and Deep Learning: A Systematic Review," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 6, pp. 2001–2007, 2024, doi: 10.18517/ijaseit.14.6.20691.

[9] B. Falakhi, E. F. Achmal, M. Rizaldi, R. R. R. Athallah, and N. Yudistira, "Perbandingan Model AlexNet dan ResNet dalam Klasifikasi Citra Bunga Memanfaatkan Transfer Learning," *J. Ilmu Komput. dan Agri-Informatika*, vol. 9, no. 1, pp. 70–78, 2022, doi: 10.29244/jika.9.1.70-78.

[10] A. Lumini and L. Nanni, "deep learning and transfer learning features for plankton

classification," vol. 3, no. April 2024, pp. 109–114, 2019.

[11] A. E. Putra, M. F. Naufal, and V. R. Prasetyo, "Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 1, p. 12, 2023, doi: 10.26418/jp.v9i1.58186.

[12] D. M. Wonohadidjojo, "perbandingan convolutional neural network pada transfer learning method untuk mengklasifikasikan sel darah putih," *Ultim. J. Tek. Inform.*, vol. 13, no. 1, pp. 51–57, 2021, doi: 10.31937/ti.v13i1.2040.

[13] M. F. Gunardi, "pengembangan aplikasi mobile untuk mengestimasi rentang usia berdasarkan foto wajah dan fine tuning pretrained CNN," 2022.

[14] Y. Religia, "feature extraction untuk klasifikasi pengenalan wajah menggunakan support vector machine dan k-nearest neighbor," *Pelita Teknol. J. Ilm. Inform. Arsit. dan Lingkung.*, vol. 14, no. 2, pp. 85–92, 2019.

[15] R. Indraswari, R. Rokhana, and W. Herulambang, "melanoma image classification based on mobilenetv2 network," *Procedia Comput. Sci.*, vol. 197, pp. 198–207, 2021, doi: 10.1016/j.procs.2021.12.132.

[16] Y. Gulzar, "fruit image classification model based on mobilenetv2 with deep transfer learning technique," *Sustain.*, vol. 15, no. 3, 2023, doi: 10.3390/su15031906.

[17] F. I. Eyiokur, H. K. Ekenel, and A. Waibel, "a computer vision system to help prevent the transmission of covid-19," *Signal, Image Video Process.*, vol. 17, no. 4, pp. 1027–1034, 2021, doi: 10.1007/s11760-022-02308-x.

[18] A. N. Nafisa, E. N. D. B. Purba, F. A. A. Harahap, and N. A. Putri, "implementasi algoritma convolutional neural network arsitektur model mobilenetv2 dalam klasifikasi penyakit tumor otak glioma, pituitary dan meningioma," *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA)*, vol. 5, no. 1, pp. 53–61, 2023, doi: 10.29303/jtika.v5i1.234.

[19] I. F. Annur, J. Umami, M. N. Annafii, N. Trisnaningrum, and O. V. Putra, "Klasifikasi Tingkat Keparahan Penyakit Leafblast Tanaman Padi Menggunakan MobileNetv2," *Fountain Informatics J.*, vol. 8, no. 1, pp. 7–14, 2023, doi: 10.21111/fij.v8i1.9419.

[20] Radikto, D. I. Mulyana, M. A. Rofik, and Mo. Z. Z. Zakaria, "klasifikasi kendaraan pada jalan raya menggunakan algoritma convolutional neural network (CNN)," *J. Pendidik. Tambusai*, vol. 6, no. 1, pp. 1668–1679, 2022.

[21] A. Peryanto, A. Yudhana, and R. Umar, "klasifikasi citra menggunakan convolutional neural network dan k fold cross validation," *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 45–51, 2020, doi: 10.30871/jaic.v4i1.2017.

[22] I. Firmansyah and B. H. Hayadi, "komparasi fungsi aktivasi relu dan tanh pada multilayer perceptron," *JIKO (Jurnal Inform. dan Komputer)*, vol. 6, no. 2, p. 200, 2022, doi: 10.26798/jiko.v6i2.600.

[23] U. S. Rahmadhani and N. L. Marpaung, "klasifikasi jamur berdasarkan genus dengan menggunakan metode CNN," *J. Inform. J. Pengemb. IT*, vol. 8, no. 2, pp.

169–173, 2023, doi: 10.30591/jpit.v8i2.5229.

[24] F. I. Kurniadi, V. K. Putri, and Y. E. Wibawa, “Klasifikasi Topeng Cirebon menggunakan Metode Convolutional Neural Network,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 1, pp. 163–169, 2021, doi: 10.35957/jatisi.v8i1.568.

[25] M. A. Hanin, R. Patmasari, and R. Y. Nur, “sistem klasifikasi penyakit kulit menggunakan convolutional neural network (CNN) skin disease classification system using convolutional neural network (CNN),” *e-Proceeding Eng.*, vol. 8, no. 1, pp. 273–281, 2021.

[26] T. A. Bowo, H. Syaputra, and M. Akbar, “penerapan algoritma convolutional neural network untuk klasifikasi motif citra batik solo,” *J. Softw. Eng. Ampera*, vol. 1, no. 2, pp. 82–96, 2020, doi: 10.51519/journalsea.v1i2.47.

[27] R. Permana, H. Saldu, and D. I. Maulana, “optimasi image classification pada jenis sampah dengan data augmentation dan convolutional neural network,” *J. Sist. Inf. dan Inform.*, vol. 5, no. 2, pp. 111–120, 2022, doi: 10.47080/simika.v5i2.1913.

[28] M. F. Rahman, D. Alamsah, M. I. Darmawidjadja, and I. Nurma, “klasifikasi untuk diagnosa diabetes menggunakan metode bayesian regularization neural network (RBNN),” *J. Inform.*, vol. 11, no. 1, p. 36, 2017, doi: 10.26555/jifo.v11i1.a5452.

[29] A. Ridhovan and A. Suharso, “penerapan metode residual network (resnet) dalam klasifikasi penyakit pada daun gandum,” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 1, pp. 58–65, 2022, doi: 10.29100/jipi.v7i1.2410.

[30] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, “analisis sentimen ulasan aplikasi dana dengan metode random forest,” ... *Teknol. Inf. dan ...*, vol. 6, no. 9, pp. 4305–4313, 2022, [Online]. Available: http://j-ptiik.ub.ac.id