ELINVO (Electronics, Informatics, and Vocational Education), Mei 2023; vol 8(1):44-54
ISSN 2580-6424 (printed), ISSN 2477-2399 (online) DOI: https://doi.org/10.21831/elinvo.v8i1.52104

Comparison of Convolutional Neural Network Architecture on Detection of Helmet Use by Humans

Hartatik[1*], Muhammad Khoirul Anam [2]
[1,2]Ilmu Komputer Universitas Amikom Yogyakarta
[*]E-mail: hartatik@amikom.ac.id

ABSTRACT

The helmet is one of the protective equipment for the head when driving. Although it is a protector, some criminals misuse helmets to disguise their identities, such as robbery at ATMs. Some places have put a sticker not to wear a helmet in the ATM room. However, this advice is often violated. The research adopted the Convolutional Neural Network (CNN) algorithm to identify humans who use helmets and do not use helmets based on digital images. Several CNN models, such as MobileNet-V2, ResNet-50, and VGG-16, were compared in performance. The experiment was carried out using a dataset consisting of 3,207 images which were divided into two classes. The first class is used for classifying human images using helmets with 1,603 images. At the same time, the second class is for images of humans who do not use helmets, with a total of 1,604 images. The test results show that the architecture with the highest accuracy value is ResNet-50, 97.81%. At the same time, the mobileNet-V2 architecture obtained a lower accuracy value of 96.36% and the VGG-16 architecture of 52.25%.

Keywords*:* CNN, MobileNet-V2, ResNet-50, VGG-16

INTRODUCTION

The need for security becomes very important for everyone, offices, homes, shopping centers, and other vital objects such as Automated Teller Machines (ATMs). ATM is one of the places that are very prone to theft. To prevent robbery and theft in ATMs, every bank applies rules or regulations in ATM rooms prohibiting users from entering ATMs with head coverings such as helmets and hats and installing CCTV. The installation of CCTV has a weakness because the supervisor must constantly monitor and monitor the screen to find irregularities. The use of Computer Vision technology or intelligent software can be one solution that can assist supervisors in giving firmness to the CCTV monitor screen, making it easier for supervisors to identify users who wear helmets.

Several techniques have been used to identify an image object, including the Convolutional Neural Network (CNN). In detecting an object, CNN has a function to perform feature extraction. Conventional methods, such as machine learning, perform feature extraction manually by first processing the extracted image features [1]. Unlike the CNN method, which automatically performs feature extraction in the convolutional layer, the pooling layer, and the activation section, which uses the Rectified Linear Unit (ReLU) function. Furthermore, the classification process is carried out on the Fully Connected layer (FCL) section with a softmax activation function [2].

The CNN algorithm has been widely used by researchers in analyzing an object with a reasonably high accuracy value [3]. Based on the description of the problems described above, the research conducted by the researcher is to build a classification system for someone who uses a helmet and does not use a helmet using the CNN method. Because various types of architecture can be used in the CNN method, this study will compare several CNN architectures to find out the highest CNN architecture accuracy for use in helmet user object detection. The architectures to be compared are ResNet-50, VGG-16, and MobileNet-V2.

CNN is often used to recognize objects or objects and detect and segment objects. To eliminate manual feature extraction, CNN learns through image data [4]. Technically, CNN is an architecture that can be trained and consists of several stages. The input and output of each stage consist of several arrays commonly called feature maps. CNN has four primary steps: the convolution layer, pooling layer, activation function, and fully connected layer. The following is an overview of the Convolutional Neural Network architecture network [5].

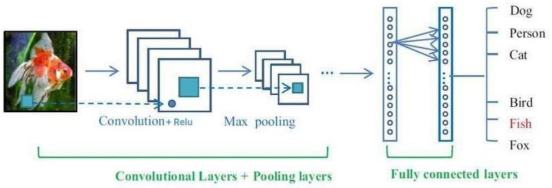


Figure 1. Convolutional Neural Network Architecture

Based on Figure 1, the first stage in the CNN architecture is convolution. This step is done by using a kernel of a specific size. Calculating the number of kernels used depends on the number of features generated. Then proceed to the activation function, usually using the ReLU (Rectifier Linear Unit) activation function. Then after exiting the activation function, proceed through the pooling process. This process is repeated several times until a sufficient feature map is obtained to proceed to the fully connected layer, and from the fully connected layer is the output class.

The introduction of motorcycle riders wearing helmets or not using CNN was studied by [6]. They developed a system based on Tensorflow and hard in the field of Computer Vision that can detect whether a motorcycle rider is wearing a helmet or not in real-time. There are 50 test classes, with the results of 49 successfully recognized correctly and one incorrectly. Another study was conducted by [7]. CNN is used to perform visual helmet identification automatically. The dataset used in this study is a pedestrian dataset in the form of video by dividing it into 20,504 segments of human-like images measuring 320 × 160 pixels. The dataset contains 4646 human samples and 15858 non-human samples. The average accuracy of the CNN method in the study reached 99.6%.

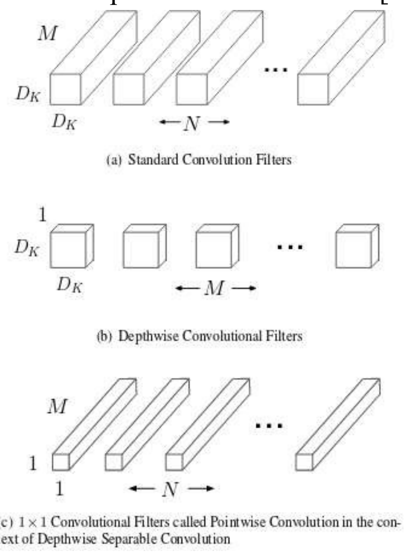
With the help of the OpenCV library for multi-face detection, the CNN method was researched by [8]. The 5MP M-Tech, Web Cam device performs facial recognition with an average accuracy of more than 89%. Salsabila [9] applies the CNN method in the classification of Punakawan wayang images using 4 classes, namely wayang gareng, wayang semar, wayang petruk, and wayang bagong. The resulting accuracy is relatively high, which is 91.6%.

T. Waris et al. [10] suggest a mechanism to automatically identify helmet infractions from surveillance footage shot by cameras set on the side of the road. The proposed method is based on a quicker region-based convolutional neural network (R-CNN) deep learning model that uses video as an input to detect helmet violations and take appropriate action against offenders. According to experimental research, the proposed system provides an accuracy of 97.69%.

B. RaviKrishna et al. [11] presented a solution for YOLOv2 (You Only Look Once version 2) and LeNet-based helmet detection and recognition. All types of helmets can be found and identified in various situations and circumstances. Instances of this misconduct are considered when a rider is visible in the source files in image or video format without a helmet, and an output is produced by analyzing the data. When used with a CPU, the trained model yields accurate results at a rate of 95% in most scenarios.

In some studies, the accuracy of the CNN method depends on the architecture used. Several architectures that are used quite often and will be compared for accuracy in this study are MobileNet V2, Residual Network (ResNet), and VGG-16.

Researchers from Google built MobileNet on the need for a CNN architecture that can be used for mobile phones and embedded systems. Figure 2 shows that the primary difference between the MobileNet architecture and the CNN architecture, in general, is the use of a convolution layer or layer with a filter thickness that matches the thickness of the input image. MobileNet divides convolution into depthwise convolution and pointwise convolution [12].



(a) Standard Convolution Filters

(b) Depthwise Convolutional Filters

(c) 1 × 1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 2. (a) Standard convolution divided into two layers: (b) depthwise convolution and (c) pointwise convolution to create separate depthwise filters [13]

The MobileNet architecture utilizes Batch Normalization (BN) and Rectified-Linear units (ReLU) for depthwise and pointwise convolution, as shown in Figure 3.

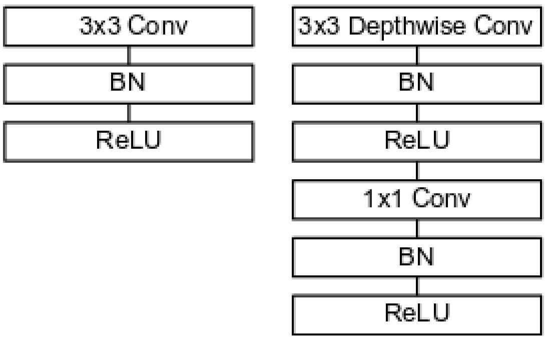


Figure 3. Left: standard convolution layer with the batch norm and ReLU. Right: Depthwise convolution and Pointwise convolution with the batch norm and ReLU [13]

MobileNet released its second version in April 2017. Like MobilenetV1, MobileNetV2 still uses depthwise and pointwise convolution [13]. MobileNetV2 adds two new features: linear bottlenecks and shortcut connections between bottlenecks. The basic structure of this architecture is shown in Figure 4.

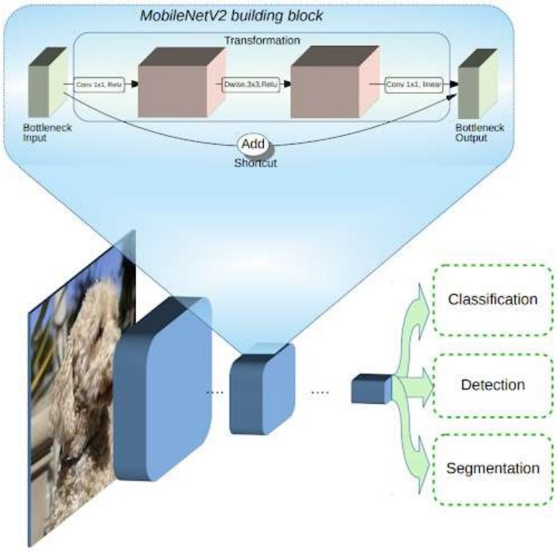


Figure 4. Architecture of MobilenetV2 [13]

In contrast to MobileNetV2, Residual Network or ResNet is a residual network with a deep network [14]. The deepest network of ResNet consists of 152 layers. This network is eight times deeper than the VGG network, but the complexity is still lower than the VGG network. The ResNet architecture model proposes a residual learning framework to simplify network training. ResNet is a CNN network architecture that can use hundreds or even thousands of convolution layers [1]. The deeper the neural network, the problem will arise, namely the loss of gradient values or vanishing gradient. ResNet stacks the identity mapping (a layer that does nothing at first), then goes through it and reuses the activation of the previous layer. Skip the initial compression of the network into multiple layers, speeding up learning. When the network is retrained, all layers are expanded, and the rest of the network will explore more image features.

Due to the increased network depth, the ResNet network is easy to optimize and obtains higher accuracy and better results than the previous network. However, in a network that only accumulates layers (counterpart plain), a higher training error occurs as the depth increases [15].

There are two types of VGG, namely VGG-16 with a depth of 16 layers and VGG-19 with a depth of 19. This network consists of 3x3 convolutional layers and the max pooling layer to reduce the size. The last layer is fully connected with 4096 neurons, followed by the softmax layer. Preprocessing is only done on the input, namely the average RGB value calculated on each pixel's training data set. A max-pooling layer follows some convolutional layers in the pooling process, but not all convolutional layers are followed by a max-pooling layer. Max pooling is done with a 2x2 pixel window and two strides. ReLu activation is used for each hidden layer. The number of filters increases according to the depth of the VGG variance [16].

VGG-16 is a model that consists of 16 convolution layers and is fully connected, which is usually used to recognize and classify images. The VGG-16 architecture has 13 convolution layers using a 3x3 convolution filter with a max pooling layer for downsampling [16]. In addition, VGG-16 also has two fully connected layers in the hidden layer with a total of 4096 layer units. Followed by a dense layer of 1000 units, where each unit represents one of the image categories in the imageNet database. The VGG-16 architecture is illustrated as shown below:

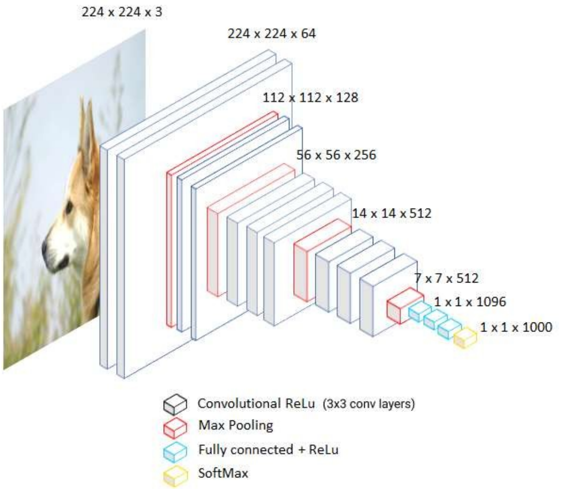


Figure 5. VGG-16 Architecture [16]

METHODS

The dataset in this study amounted to 3207 images obtained via the internet, taken directly through cellphone and notebook cameras. Of the 3207 images, as many as 1603 images of humans wearing helmets and 1604 images of humans not wearing helmets. The existing data then goes to the preprocessing stage. The preprocessing step resizes the dataset to 244 x 244 pixels. This data is then segmented to determine edge detection and identify objects. The augmentation process is carried out using the Python programming language, which utilizes the ImageDataGenerator library in the hard module. The stages of image augmentation carried out in this study are:

1. Rotation. At this stage, the generated data is rotated randomly with an angle of 20 degrees.
2. Zoom. The zoom used is in the zoom_range = 0.15 size. This number means that the image that enters the augmentation stage is enlarged by 1+0.15 of the image area.
3. Shift width. Width_shift_range or floating point numbers used in the study were assigned values between 0.2.
4. Shift height. The image is shifted 0.2 vertically.
5. Shear Intensity. This step will create a kind of 'stretch' in the image, which is not visible in rotation. shear_range is set to 0.150 angles.
6. Flip Horizontal. At this stage, the generator will generate an image randomly flipped horizontally.
7. Brightness. Brightness is set by specifying a range to select the brightness shift value randomly
8. Channel Shift. Channel shift randomly shifts the channel value by a selected value from the range defined by channel_shift_range.

After the preprocessing process is run, the next step is to classify using the CNN method. The description of the stages of the CNN algorithm in classifying can be seen in Figure 6.

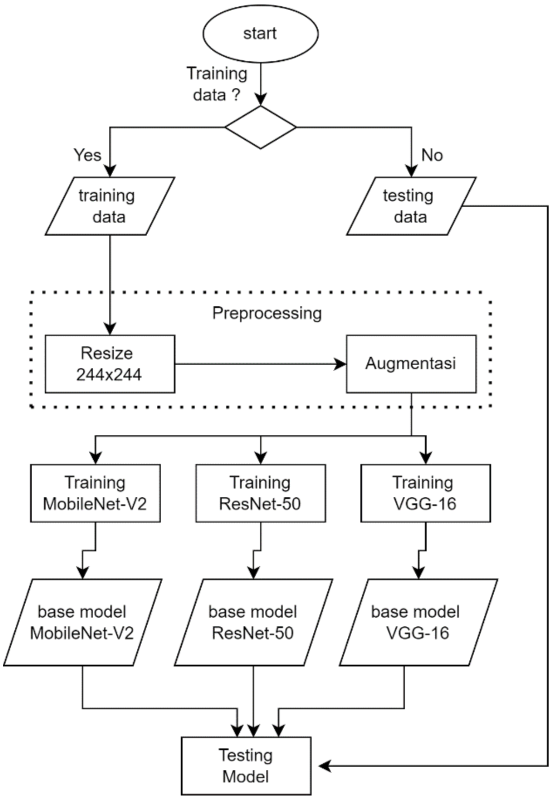


Figure 7. CNN method classification process

The training and testing process uses three architectures ResNet50, VGG16, and MobileNet. After the preprocessing process, the dataset is divided into training and test data. The datasets are trained independently using three architectures. Loss values and model accuracy resulting from each architecture on the test data are compared to get the best model conclusion. In addition to architecture, the epoch value is also tested to get the smallest error or error.

Furthermore, the experimental stage is carried out by trying the model to work in streaming conditions. Images of people wearing helmets and not wearing helmets in jpg format were tested under various conditions to get the model's accuracy.

RESULT AND DISCUSSION

The data training uses the transfer learning method from the model provided by Tensorflow. The training process uses a Batch Size of 32, epochs of 30, and the learning rate is assigned a value of 0.0001. The model compilation process uses 'binary cross entropy as a loss function, and the model optimizer uses 'adam.'

The training data results on the three architectures are presented in Table 1. The MobileNet-V2 architecture takes 3.329 seconds, or an average of 111 seconds per epoch. In comparison, the ResNet50 architecture takes 8,395 seconds or an average of 279 seconds per epoch. Finally, the VGG16 architecture takes the longest time, 52.380 seconds, or an average of 1746 seconds per epoch.

Classification using 60 images (30 people are wearing helmets and 30 images are not wearing helmets) resulted in varying accuracy for these three architectures. These images can be seen in Figures 7, 8, 9, 10, 11, and 12.

The ResNet50 architecture produces the best accuracy compared to the other two architectural models. ResNet50 has an accuracy of 99.50% for classifying human images wearing helmets. As for images of humans who do not use helmets, the accuracy produced is the same at 99.50%. In contrast to ResNet50, the MobileNet-V2 architecture can have 97.73% accuracy in classifying human images using helmets. For human images without helmets, the MobileNet-V2 architecture produces the same accuracy as ResNet50, which is 99.50%. Finally, the VGG16 architecture has the worst accuracy in classifying human images using helmets, with an accuracy of 10%. While the accuracy of the classification of human images that do not use a helmet, the VGG16 architecture has the same accuracy value as the ResNet50 and MobileNet-V2 architectures, namely 99.50%.

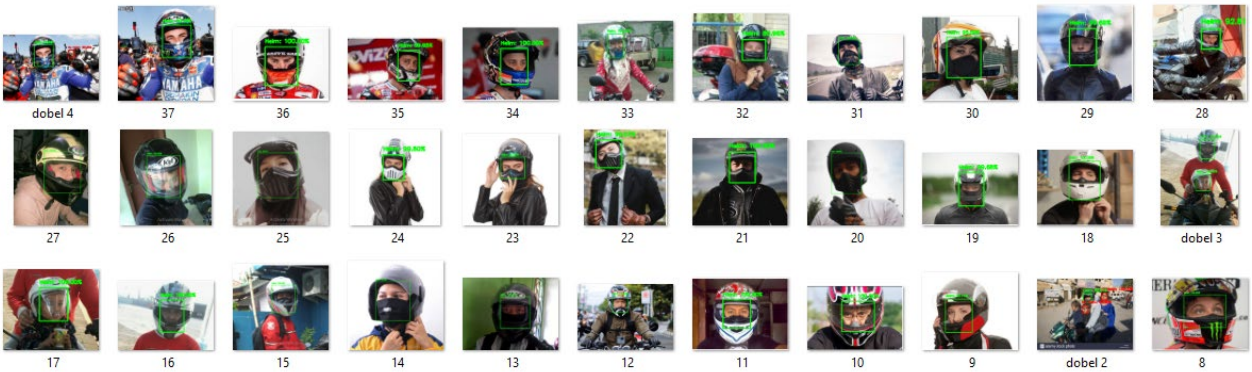


Figure 8. Classification images of people wearing helmets using MobileNet-V2 Architecture

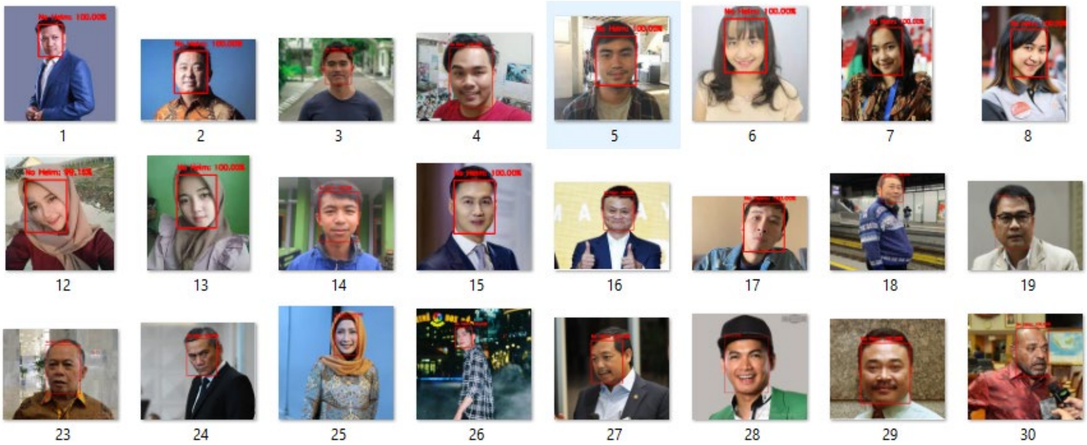


Figure 9. Classification images of people not wearing helmets using MobileNet-V2 Architecture

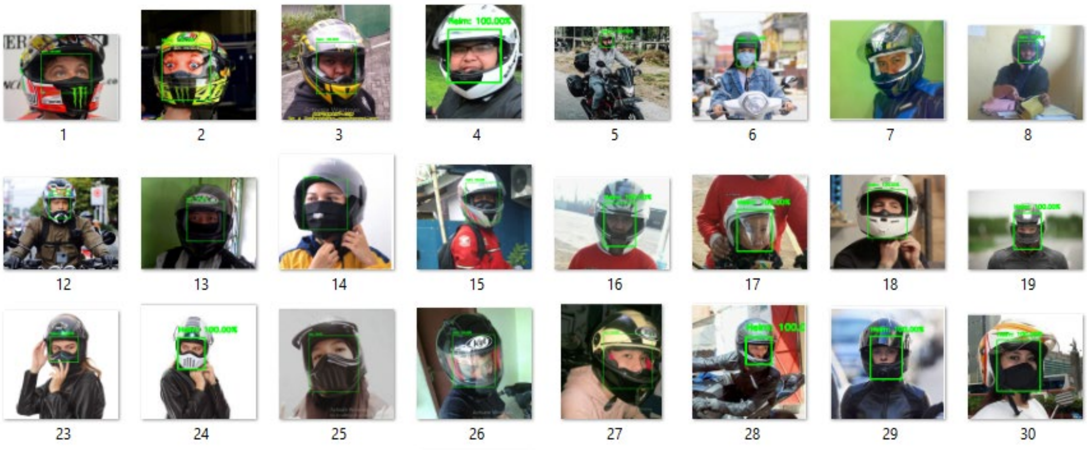


Figure 10. Classification images of people wearing helmets using ResNet50 Architecture

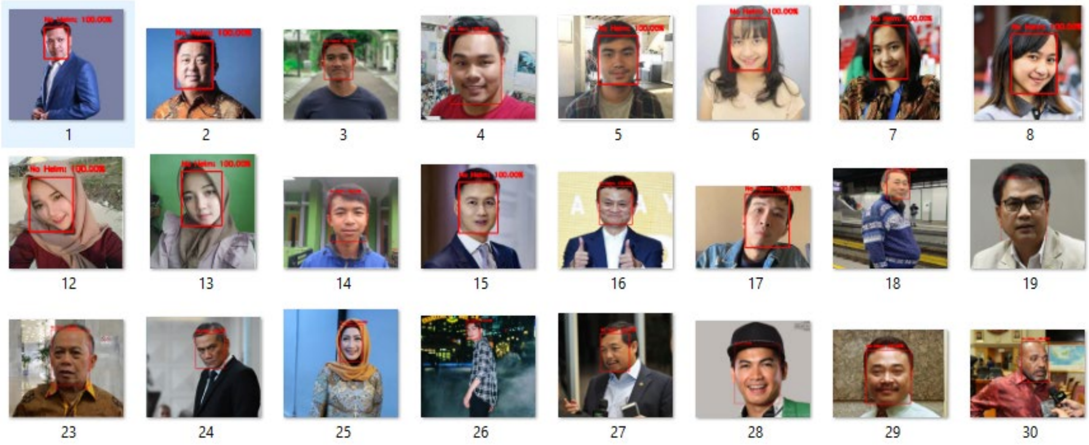


Figure 11. Classification images of people not wearing helmets using ResNet50 Architecture

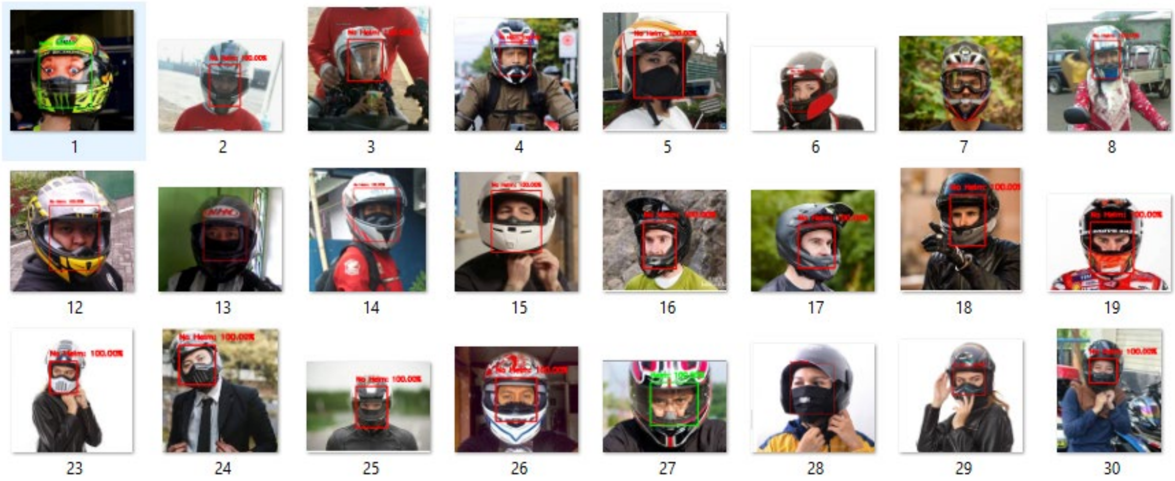


Figure 12. Classification images of people wearing helmets using VGG-16 Architecture

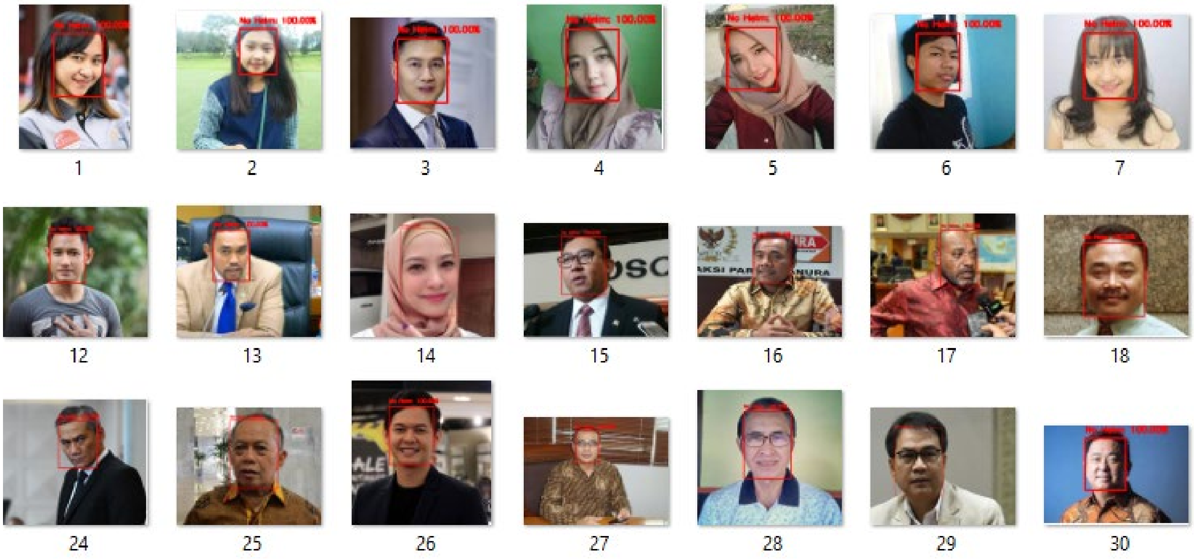


Figure 12. Classification images of people not wearing helmets using VGG-16 Architecture

Table 1. Classification results on 30 test images

Architecture	Image of Human Using Helmet	Image of people not wearing helmets
MobileNet-V2	97.73%	99.50%
ResNet50	99.50%	99.50%
VGG16	10%	99.50%

The second testing mechanism is done by streaming using a WebCam camera directly. The output is a video recording in .mp4 format. An example of an image for streaming testing can be seen in Figures 9, 10, 11, and 12.

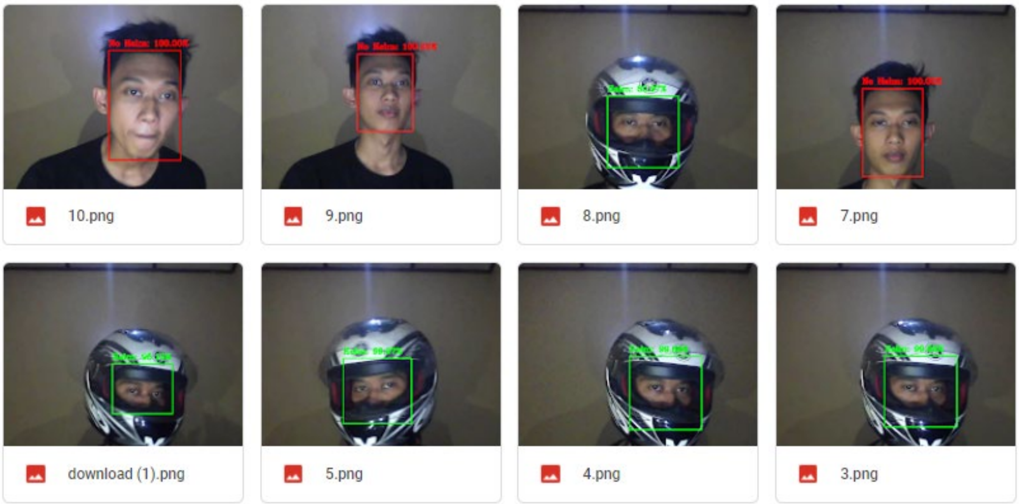


Figure 9. Example video for streaming data classification using MobileNet-V2 Architecture

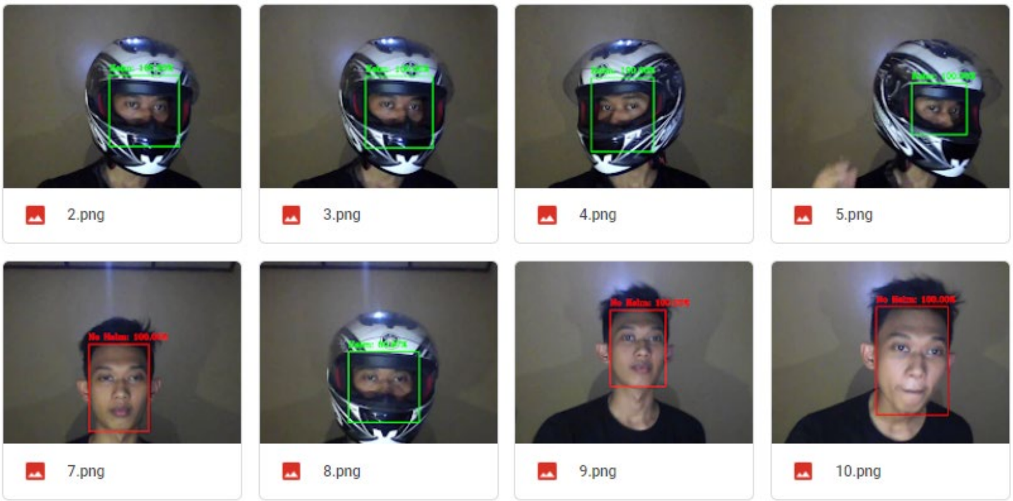


Figure 10. Example video for streaming data classification using ResNet50

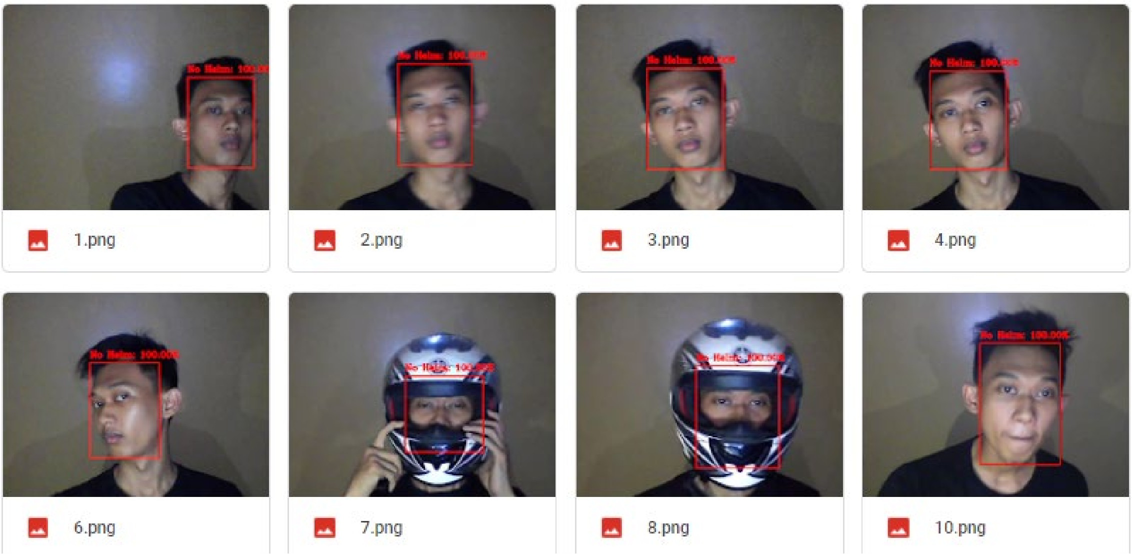


Figure 10. Example video for streaming data classification using VGG-16

The accuracy of the three architectures in classifying video streaming produces a value that is not much different from the classification in the image. A summary of the accuracy of the test results on video streaming can be seen in Table 2. The Resnet50 architecture has 92.76% accuracy in images of people wearing helmets. At the same time, images of people without helmets produce the same accuracy of 99.50%, in contrast to the MobileNet-V2 architecture, which only gets an accuracy value of 88.73% on images of people wearing helmets. The MobileNet V2 architecture provides the same accuracy as Resnet 50, 99.50% for human images without helmets. The VGG16 architecture is the least accurate in classification using streaming video data. The accuracy of VGG16 produces an accuracy value of 0% in classifying human images using a helmet. The small accuracy value in the detection of helmet use is due to the fact that the VGG16 architecture focuses a lot of attention on face detection. This is reasonable, considering that the Vgg16 architecture has a simpler layer than the other two architectures. While the human image without using a helmet, VGG16 Architecture has an accuracy value of 99.50%.

Table 2. Classification results on 30 test video streaming

Architecture	Video of Human Using Helmet	Video of people not wearing helmets
MobileNet-V2	88.73%	99.50%
ResNet50	92.76%	99.50%
VGG16	0%	99.50%

CONCLUSION

The research has succeeded in classifying images of people wearing helmets and not using helmets using three different architectures in the Convolutional Neural Network algorithm. The model is built using 3207 image datasets with image dimensions measuring 244x244 pixels. The images are divided into two classes, the first class contains images of people wearing helmets, and the second class contains images of people not wearing helmets. The number of each image class is 1603 and 1604 datasets, respectively. The training begins with data augmentation and freezes the base model.

The learning rate used in training is 0.0001, with a dropout value of 0.5. The three architectures compared are MobileNet-V2, Resnet-50, and VGG-16. Experiments are carried out by setting different times when training the dataset. The time required for each architecture to conduct training data is: MobileNet-V2 takes an average of 111 seconds per epoch, Resnet50 takes 279 seconds per epoch, and VGG16 takes the longest time of 1,746 seconds per epoch. Tests using image data obtained different results for the three architectures. Resnet-50 architecture got the highest accuracy value, with a value of 99.50% for images of people wearing helmets and 99.50% for images of people not wearing helmets.

Meanwhile, the mobileNet-V2 and VGG-16 architectures obtained accuracy rates of 97.73% and 10%, respectively, for images of people wearing helmets and the same accuracy value of 99.50% for images of people not wearing helmets. The test uses video streaming data with the position of the person wearing a helmet; the ResNet50 architecture has the largest accuracy value of 92.76%. Meanwhile, the MobileNet-V2 architecture obtained an accuracy value of 88.73%. The VGG-16 architecture owns the lowest accuracy with a value of 0%. As for videos with people not wearing helmets, the three architectures have the same accuracy value of 99.50%.

REFERENCES

[1] W. Setiawan, "Perbandingan Arsitektur Convolutional Neural Network Untuk Klasifikasi Fundus," *J. Simantec*, vol. 7, no. 2, pp. 48–53, 2020.

[2] H. Hartatik, H. Al Fatta, and U. Fajar, "Captioning Image Using Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM)," *2019 2nd Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2019*, no. December 2020, pp. 263–268, 2019.

[3] K. Kusrini, A. Setyanto, I. MADE ARTHA AGASTYA, H. Hartatik, K. Chandramouli, and E. Izquierdo, "A Deep-Learning Framework for Accurate and Robust Detection of Adult Content," *J. Eng. Sci. Technol.*, vol. 17, no. 3, pp. 2104–2119, 2022.

[4] R. Fachmi, A. Hidayatno, and A. Adi, "Sistem Identifikasi Ukuran Tubuh Menggunakan Metode Convolutional Neural Network (CNN)," *TRANSIENT*, vol. 9, no. 1, pp. 1–7, 2020, [Online]. Available: https://ejournal3.undip.ac.id/index.php/transient/article/view/25299

[5] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[6] A. Soni and A. P. Singh, "Automatic Motorcyclist Helmet Rule Violation Detection using Tensorflow Keras in OpenCV," *2020 IEEE Int. Students' Conf. Electr. Electron. Comput. Sci. SCEECS 2020*, no. November, 2020.

[7] R. Cao, H. Li, B. Yang, A. Feng, J. Yang, and J. Mu, "Helmet wear detection based on neural network algorithm," 2020.

[8] M. Zufar and B. Setiyono, "Convolutional Neural Networks untuk Pengenalan Wajah Secara Real - Time," *J. SAINS DAN SENI ITS*, vol. 5, no. 2, pp. 72–77, 2016.

[9] Salsabila, "Penerapan Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Wayang Punakawan," Universitas Islam Indonesia, 2018.

[10] T. Waris *et al.*, "CNN-Based Automatic Helmet Violation Detection of Motorcyclists for an Intelligent Transportation System," *Math. Probl. Eng.*, vol. 2022, 2022.

[11] B. RaviKrishna, K. S. Priya, J. Harika, M. Pranathi, and N L Apoorva, "Comprehensive CNN-Based Approach for Helmet Use Detection of Tracked Motor Cycles," in *2021 4th International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE)*, 2021, vol. 3, pp. 5–9.

[12] A. R. Putri, "Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 1, no. 01, pp. 1–6, 2016.

[13] M. Sandler, M. Zhu, A. Zhmoginov, and C. V Mar, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv:1801.04381v4*, 2019.

[14] S. N. A. F. Akbar, Hendra, and Supri Bin Hj. Amir, "Perbandingan Kinerja Arsitektur Inception-V4 Dan Resnet-50 Dalam

Mengklasifikasikan Citra Paru-Paru Terinfeksi Covid-19 Siti," 2020.

[15] K. He, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385v1*, 2015.

[16] R. Rismiyati and A. Luthfiarta, "VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification," *Telematika*, vol. 18, no. 1, p. 37, 2021.