Classification of Original and Fake Images Using Deep Learning- Resnet50

Novita Rajagukguk[1]*, I Putu Eka Nila Kencana[2]
I GN Lanang Wijaya Kusuma[3]

[1,2,3]Udayana University, Bali, Indonesia
*novitarajagukguk920@gmail.com

Abstract.
Computer Vision has developed and become a necessity for human life. Computer Vision is widely used in image classification tasks, object detection, semantic segmentation, video understanding, and so on. In this research, Computer Vision is used to perform classification tasks to distinguish between real and fake images of human faces. To carry out classification tasks, this research will use a CNN model with the ResNet50 architecture which is known to be good at carrying out image classification tasks. ResNet50 is well known to solve the vanishing gradient problem with fewer stacked layers and minimizes time in learning with ever-increasing accuracy. This research uses 589 real human face image data and 700 fake human face data. By using image augmentation in the pre-processing stage, the data is divided by 80% for training data and 20% for validation data. The results of model training on training data show an accuracy of 76,07% and the model performance in testing data shows an accuracy of 53%.

Keywords: Classification Image, Convolution Neural Network, ResNet50

1 Introduction

Technology has developed very rapidly and is used in various areas of life. The technological advancement that is widely used today is computer vision. Computer vision is a computer system that imitates aspects of the human visual system to perform various visual tasks such as image classification, object detection, semantic segmentation, video understanding, image generation, 3D vision, multimodal tasks, and self-directed learning [1]. The use of computer vision to create fake images for personal gain is becoming increasingly prevalent, and this can have both positive and negative consequences. To address this issue, we need a program that can accurately distinguish between real and fake images. One effective method of achieving this is through image classification.

Image classification is one of the tasks of computer vision which is tasked with grouping objects based on the characteristics of the object [2]. One well-known method that can perform image classification tasks well is the Convolutional Neural Network (CNN) method. CNN is a learning neural network that consists of convolutional layers and subsampling layers and ends with one or more layers connected to a Multilayer

K. Dharmawan and N. A. Sanjaya ER (eds.), *Proceedings of the First International Conference on Applied Mathematics, Statistics, and Computing (ICAMSAC 2023)*, Advances in Computer Science Research 110,
https://doi.org/10.2991/978-94-6463-413-6_6

Perceptron (MLP). The advantage of using CNN is that CNN can extract features simultaneously, reduce data dimensions, and classify images in one network structure [3].

The accuracy of deep learning in computer vision tasks has been improved by the development of CNNs. Several architectures such as LeNet-5, AlexNet, VGGNet, and ResNet have been designed by researchers to enhance network performance [4]. ResNet in this case has been developed to overcome the vanishing gradient problem with fewer layer stacks and minimize learning time with ever-increasing accuracy [5].

Several previous studies have carried out computer vision tasks in image classification, for example, Diyasa and Romadhon carried out the classification of Javanese script which is unique in each letter and has 20 letter characters with different consonant pronunciations. This research uses the Convolutional Neural Network (CNN) algorithm to classify Javanese script images into 20 classes. The research results show that the highest accuracy is obtained with the contour and canny segmentation models with an accuracy value of 0,5307 and a loss of 5,1892 [6].

In 2022, Alzamily et al. perform image classification for protection from data attacks using the CNN algorithm with ResNet50 architecture. By training on encrypted images, the research results show that encoded image classification can classify encrypted images without decoding the images again. The model performance in image classification in his research showed an accuracy of 99,75%, Recall of 94,12%, Precision of 94,23%, and F1-Score of 94,70% [7].

Further research was carried out by Latif and Khalifa to differentiate normal images, pneumonia viruses, and COVID-19 using three Deep Transfer Learning (DLT) algorithms. The research results show higher training accuracy using ResNet50 with an accuracy of 94,72%. Furthermore, the ResNet50 model in his research showed good performance in testing with an accuracy value of 80,66% [8].

Furthermore, Liew et al. classify gender using a Convolutional Neural Network approach. By using a CNN architecture that is optimized by combining convolutional and subsampling layers, as well as cross-correlation, the research results show superior classification performance with accuracy of 98,75% and 99,38%.[3].

Then Natsir et al. Classifying tuna to see whether tuna is suitable for export or not. This research used 128 data which was divided into 80% for training data and 20% for test data. By using the VGG-16 architecture which is a CNN model an accuracy of 81,9%, a precision value of 79,4%, and a recall value of 90% were obtained.[9].

From these studies, many previous researchers have used the CNN algorithm to classify images. Therefore, this research will use the CNN algorithm with ResNet architecture to classify real images and fake images.

2 Methods

2.1 Dataset

The data used in this research is image data obtained from the Kaggle discussion forum. Data was then collected using the Kaggle library available in Python. Using the Kaggle API, the data will then be downloaded and stored in the prepared directory.

2.2 Pre-processing Image

The data that has been obtained will then go through a pre-processing stage to extract useful information and discard useless information. Image pre-processing aims to improve the quality of the input data so that the data will have a good influence on the prediction results. In image data pre-processing several techniques can be used to improve data quality such as erosion, dilation, opening, closing, and super-resolution [10].

In data pre-processing, there is the term data augmentation which is used to improve the quality and quantity of data. Data augmentation is one method that can be used to reduce validation errors and prevent overfitting. In augmentation, data is extracted from the original data set so that the data will be more comprehensive and can minimize the distance between training data and validation and testing data [11].

2.3 Convolution Neural Network

Convolutional Neural Network (CNN) is an algorithm formed by several layers of convolutional filters and interspersed with subsampling filters followed by layers of fully connected layers [3]. CNN is an algorithm built on three types of layers: Convolution, Pooling, and Fully Connected, all of which are based on mathematics [9]. Convolution Layers consist of kernels or can be called convolution filters. In the Convolution layer, convolution operations will be carried out, namely capturing image features in the input data [12]. The pooling layer is an operation to prevent over-fitting classification results by reducing feature dimensions [13]. Then the Fully Connected layer is a layer for carrying out feature dimension transformations so that they can be classified as linear functions [2]. Then when building the model, the activation function will be used to introduce non-linearity in building a good model. There are many common activation functions used today such as ReLu, tanh, sigmoid activation functions, and so on [14].

Convolution Layer

A convolution layer is a layer that captures small, low-level features in the first hidden layer and then builds the features into higher-level features in the next hidden layer. Convolution operations on input data use filters or kernels to capture specific features in a plane such as edges, corners, or other patterns. In the context of convolution layers, each filter will produce one output, namely a feature map. For each feature map, the convolution layer has one neuron per pixel and all neurons in the feature map have the same parameters, namely weight and bias [15]. In mathematics, the output of a particular neuron in a convolution layer can be calculated by equation (1).

$$z_{i,j,k} = b_k - \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'}.w_{u,v,k',k} \quad with \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases} \qquad (1)$$

Pooling Layer

The pooling layer is a layer that aims to reduce computational load, memory usage, and number of parameters by subsampling or reducing pixels. Just like the convolution layer, each neuron in the pooling layer is also connected to the output neuron in the previous layer. In a pooling layer, the size, stride, and padding type are determined independently. Neurons in the pooling layer have no weights, so simply aggregate input using an aggregation function known as MaxPooling or Average Pooling [15].

Fully Conected Layer

Fully Connected Layer is a layer that will produce output in the form of a probability distribution for all classes [16]. A fully Connected Layer is a layer that is connected with an activation function with the aim that the model being built can understand complex relationships in the data. Calculations for accuracy and loss results can be written in the mathematical equation (2) and equation (3), where N is the number of observations and K is the number of classes.

$$Accuracy = \frac{Images\ Validation}{Number\ of\ Images\ Validation} \qquad (2)$$

$$Loss = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{K1} (\text{Images Train} - \text{Images Validation})^2 \qquad (3)$$

2.4 ResNet50

Residual Network (ResNet) is an architecture developed to overcome training errors that cause low accuracy degradation [5]. The ResNet model was created to show that models with deeper training should produce higher accuracy. In the ResNet architecture, a residual network is introduced that can be optimized for deeper training so that with deeper training, accuracy will increase. ResNet 50 is a variant of ResNet which consists of 50 layers with a residual network [5]. The construction of the ResNet50 architecture is shown in Table. 1.

Table 1. ResNet50 architecture construction

Layer name	Output Size	Arsitektur ResNet
conv1	112×112	7×7, 64, stride 2
conv2_x	56×56	3×3 max pool, stride 2
		$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$

conv3_x	28×28	$\begin{bmatrix}1\times 1,128\\3\times 3,128\\1\times 1,512\end{bmatrix}\times 4$
conv4_x	14×14	$\begin{bmatrix}1\times 1,256\\3\times 3,256\\1\times 1,1024\end{bmatrix}\times 6$
conv5_x	7×7	$\begin{bmatrix}1\times 1,512\\3\times 3,512\\1\times 1,2048\end{bmatrix}\times 3$
	1×1	Average pool, 1000-d fc, softmax
FLOPs		3.8×10^9

2.5 Evaluation

When evaluating experimental results, performance models can be assessed using accuracy, precision, recall, and F1-score calculations. [17].

Accuracy measures the value of predictions that are classified as correct compared to the data as a whole. In mathematics, accuracy can be written in equation (4)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (4)$$

Recall to measure much of the data that is classified as correct versus all of the data that is essentially correct. In mathematics, recall can be written in equation (5)

$$Recall = \frac{TP}{TP+FN} \qquad (5)$$

Precision measures the amount of data that is predicted to be correct compared to the data that is predicted to be correct without considering the truth value of the prediction in reality. In mathematics, precision can be written in equation (6)

$$Precision = \frac{TP}{TP+FP} \qquad (6)$$

The f-1 score is a calculation of precision and recall. F-1 score describes the FP and FN as the best value for viewing model performance. In mathematics, f-1 score can be written in equation (7)

$$\text{F-1 Score}= \frac{Precision * \text{Recall}}{Precision + \text{Recall}} \qquad (7)$$

3 Road Map

3.1 Dataset

The data used in this research is image data obtained from the Kaggle discussion forum which can be accessed via the website page https://www.kaggle.com/datasets/hamzaboulahia/hardfakevsrealfaces. The data used consists of two classes, namely image data containing real photos and fake images of human faces. The original human face image data is 589 images, while the fake human face data is 700 images.

3.2 Pre-processing Data

The data that has been collected will then be pre-processed to improve the quality and quantity of data using data augmentation. In the pre-processing stage, the data will be rotated randomly with a predetermined distance of 20°. Then the data will be shifted horizontally and vertically. Image shifts carried out horizontally and vertically are set with a maximum shift of 10%. Next, the data is enlarged and reduced by a random factor between 1-0.2 and 1+0.2. The image pixel range will be converted into normal form into a 1-0 range by dividing each pixel value by 255 and empty pixels will be filled with the closest pixel value. The data will be divided into 80% for training data and 20% for validation data. Each image will be resized to 224×224 size.

3.3 Classification

The CNN architecture used to build the model in this research is ResNet50. The ResNet50 architecture follows the general design of the main CNN architecture, namely by building a model on convolution layers, pooling layers, and fully connected layers. Meanwhile, in the ResNet50 architecture, there is a deep residual network that is used in deep learning training. Next, using the program code provided by Tensorflow, the model is built with the argument weights = 'imagenet' which states that the model uses pre-trained weights from the ImageNet dataset, input_shape with size (256,256, 3) which states the dimensions of the input image with image resolution, namely 256×256 with 3 channels (RGB), and include='False' which states that the model will remove fully connected layers at the end of the model. The fully connected layer in ResNet50 will be removed because, in the next program, a separate fully connected layer will be added. Next, the ResNet50 model will be inserted into the Sequential model to create a model in sequential conditions. The ResNet50 model will also freeze all weights of all layers during the training of the new model. In the next program code, the model will add a Flatten layer to equalize the output into a one-dimensional vector, add a Dense Layer with 512 units with the ReLu activation function, and add a Dense Layer with 1 unit by including sigmoid activation function parameters which are suitable for binary classification tasks. After the model is built, the next model will be compiled with the Adam optimizer.

3.4 Evaluation

Evaluation is performed by utilizing accuracy, precision, recall, and f1-score metrics based on a confusion matrix to evaluate prediction results.

4 Result

4.1 Dataset

The dataset used in this research is public data available on the Kaggle platform which consists of 1289 real and fake images of human faces. Some real and fake image data is shown in Figure 1.

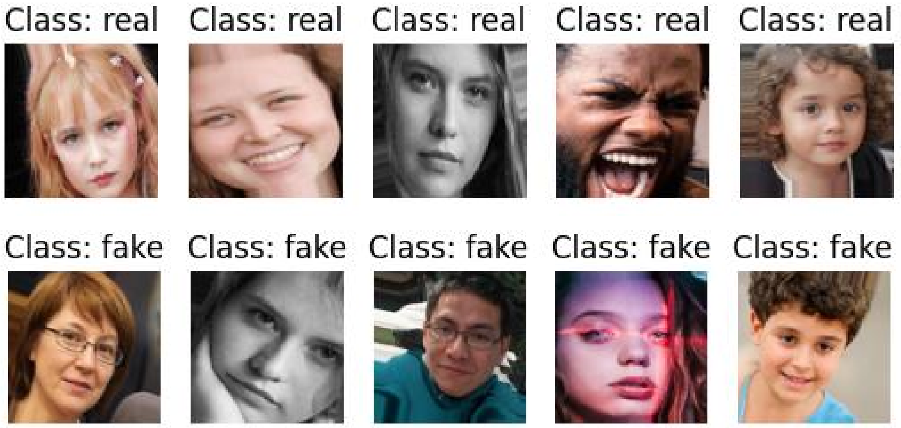


Fig. 1. Real and fake image datasets

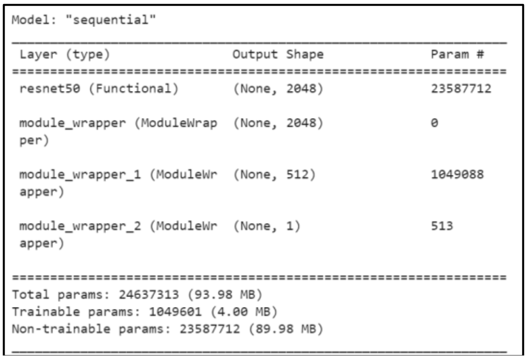
4.2 Pre-processing

The pre-processing stage is carried out to produce good image data before the training and validation process is carried out. This research uses data augmentation with libraries provided in Python to improve the quality and diversity of training data. One example of increasing the diversity of image data is by shifting the image vertically as shown in Figure 2.

Fig. 2. Vertical pre-processing of images

4.3 Model

The model is built with the Resnet50 architecture which refers to the use of residual blocks. A summary of the model architecture built with ResNet50 can be seen in Figure 3. Next, the data that has been divided into training data and validation data will be trained with an epoch value of 30 and a batch size of 32.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 2048)              23587712

 module_wrapper (ModuleWrap  (None, 2048)              0
 per)

 module_wrapper_1 (ModuleWr  (None, 512)               1049088
 apper)

 module_wrapper_2 (ModuleWr  (None, 1)                 513
 apper)

=================================================================
Total params: 24637313 (93.98 MB)
Trainable params: 1049601 (4.00 MB)
Non-trainable params: 23587712 (89.98 MB)
_________________________________________________________________
```

Fig. 3. Summary of ResNet50 model architecture

4.4 Evaluation Model

The training data results show an accuracy value of 76.07% with a validation accuracy of 74.71%. The results of accuracy, loss, accuracy validation, and loss validation from training data using the ResNet model can be seen in the plot shown in Figure 4.

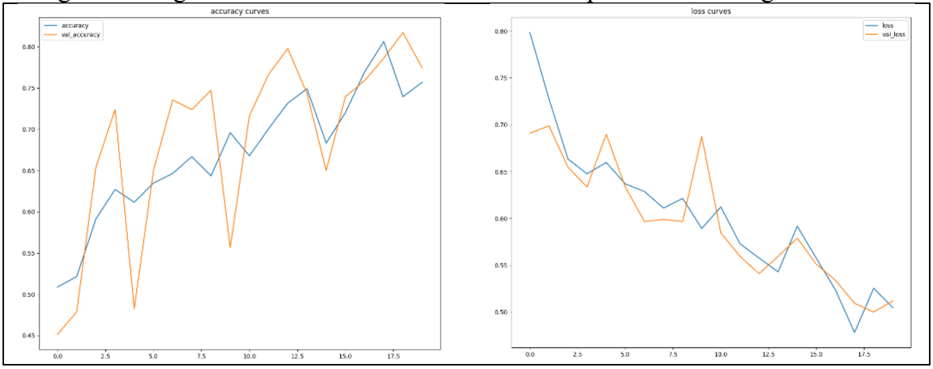
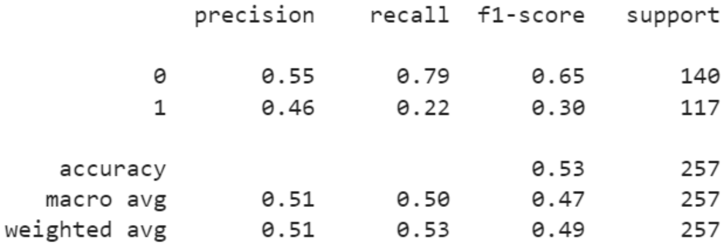


Fig. 4. Plot accuracy, validation accuracy, loss and validation loss

The accuracy and validation accuracy plot above shows that training gets better as learning progresses, this can be seen from the increasing accuracy and validation accuracy values. On the other hand, the loss value appears to be getting smaller as the learning process progresses. This can be seen in the plot which shows that the loss and validation loss lines are decreasing as learning is carried out. Next, the trained model will be used to carry out testing on data validation. The performance of the model in carrying out classification in testing will be shown in Figure 5.



```
              precision    recall  f1-score   support

           0       0.55      0.79      0.65       140
           1       0.46      0.22      0.30       117

    accuracy                           0.53       257
   macro avg       0.51      0.50      0.47       257
weighted avg       0.51      0.53      0.49       257
```

Fig. 5. Classification Report

From the classification report, it can be seen that the model's performance in classifying test data using the Resnet50 algorithm shows an accuracy value of 53%.

5 Conclusion

Research on classifying images of fake faces and real faces was carried out through the Augmentation pre-processing stage. After the data went through the pre-processing stage, the model was built using a CNN model with the ResNet50 architecture, and learning was carried out in 30 epochs. The results of training using ResNet50 obtained an accuracy value of 76.07%. Furthermore, the model performance in data testing shows an accuracy of 53%. The next research proposal is to carry out classification using architectures, activation functions, and optimizers that are different from this research.

References

[1] M. H. Guo *et al.*, “Attention mechanisms in computer vision: A survey,” *Comput. Vis. Media*, vol. 8, no. 3, pp. 331–368, 2022, doi: 10.1007/s41095-022-0271-y.

[2] O. Orlando and M. E. Al Rivan, “Klasifikasi Jenis kanker Kulit Manusia Menggunakan Convolution Neural Network,” *MDP Student Conf.*, vol. 2, no. 1, pp. 144–150, 2023, doi: 10.35957/mdp-sc.v2i1.4335.

[3] S. S. Liew, M. Khalil-Hani, S. Ahmad Radzi, and R. Bakhteri, “Gender classification: A convolutional neural network approach,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 24, no. 3, pp. 1248–1264, 2016, doi: 10.3906/elk-1311-58.

[4] X. Lei, H. Pan, and X. Huang, “A Dilated CNN Model for Image

Classification," *IEEE Access*, vol. 7, pp. 124087–124095, 2019, doi: 10.1109/ACCESS.2019.2927169.

[5] K. He, "Deep Residual Learning for Image Recognition".

[6] I. G. S. M. Diyasa and R. Romadhon, "Klasifikasi Karakter Tulisan Aksara Jawa Menggunakan Algoritma Convolutional Neural Network," *Semin. Keinsinyuran Progr. Stud. Progr. Profesi Ins.*, vol. 3, no. 1, pp. 927–936, 2023, doi: 10.22219/skpsppi.v3i1.7720.

[7] J. Yousef, I. Alzamily, S. B. Ariffin, and S. S. A. B. U. Naser, "CLASSIFICATION OF ENCRYPTED IMAGES USING DEEP LEARNING – RESNET50," vol. 100, no. 21, 2022.

[8] E. I. A. El-latif and N. E. Khalifa, "COVID-19 digital x-rays forgery classification model using deep learning," vol. 12, no. 4, pp. 1821–1827, 2023, doi: 10.11591/ijai.v12.i4.pp1821-1827.

[9] A. M. F. M. Natsir, A. Achmad, and H. Hazriani, "Klasifikasi Ikan Tuna Layak Ekspor Menggunakan Metode Convolutional Neural Network," *J. Ilm. Sist. Inf. dan Tek. Inform.*, vol. 6, no. 2, pp. 172–183, 2023, doi: 10.57093/jisti.v6i2.173.

[10] D. Stojnev and A. Stojnev Ilić, "Preprocessing Image Data for Deep Learning," pp. 312–317, 2020, doi: 10.15308/sinteza-2020-312-317.

[11] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[12] M. F. Nazil, A. B. Firmansyah, and R. Purbaningtyas, "Klasifikasi Keparahan Demensia Alzheimer Menggunakan Metode Convolutional Neural Network pada Citra MRI Otak," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 1–7, 2023, doi: 10.57152/malcom.v3i1.200.

[13] M. A. T. Ahmed Ali Mohammed Al-Saffar, Hai Tao, "2017 ICRAMET proceeding : 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET) : Jakarta, Indonesia, October 23-24, 2017," *Int. Conf. Radar, Antenna, Microwave, Electron. Telecommun.*, pp. 26–31, 2017.

[14] A. F. Agarap, "An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification," 2017, [Online]. Available: http://arxiv.org/abs/1712.03541

[15] A. Geron, *Hands-on Machine Learning With Scikit-Learn, Keras, and TensorFlow*, 2nd ed. O'Reilly Media, 2019. [Online]. Available: http://oreilly.com/catalog/errata.csp?isbn=9781492032649 f

[16] W. L. Mao, H. I. K. Fathurrahman, Y. Lee, and T. W. Chang, "EEG dataset classification using CNN method," *J. Phys. Conf. Ser.*, vol. 1456, no. 1, 2020, doi: 10.1088/1742-6596/1456/1/012017.

[17] C. J. Ejiyi *et al.*, "ResfEANet: ResNet-fused External Attention Network for Tuberculosis Diagnosis using Chest X-ray Images," *Comput. Methods Programs Biomed. Updat.*, p. 100133, 2023, doi: 10.1016/j.cmpbup.2023.100133.