International Journal of Application on Sciences, Technology and Engineering
(IJASTE)
Volume 1, Issue 4, 2023. ISSN:2987-2499

CLASSIFICATION OF SIBERIAN HUSKY AND GOLDEN RETRIEVER DOGS USING CONVOLUTIONAL NEURAL NETWORK METHOD

Christie Redja[1], Kelvin[2], Meirista Wulandari[3]

[1] Faculty of Engineering, Electrical and Electronics Engineering, Tarumanagara University, *Email : christie.525210009@stu.untar.ac.id*
[2] Faculty of Engineering, Electrical and Electronics Engineering, Tarumanagara University
Email : kelvin.525210001@stu.untar.ac.id
[3]Faculty of Engineering, Electrical and Electronics Engineering, Tarumanagara University
Email : meiristaw@ft.untar.ac.id

Submitted: 26-09-2023, Revised: 26-10-2023, Accepted: 07-12-2023

ABSTRACT

Classification of a dog is interesting because it allows owners to provide specialised care, detect health issues, and track relevant genetic history, ultimately improving the well-being and lives of dogs. Dog classification based on images is an interesting and crucial task in the fields of pattern recognition and computer vision. Convolutional Neural Network (CNN) methods have become one of the most effective and popular approaches for classifying objects in images. CNN is a type of artificial neural network architecture inspired by the way human vision works. In this research, the author has developed a Python program using CNN to recognize Golden Retrievers and Huskies. The program is designed to classify these specific dog breeds based on provided images. The research is conducted through several stages to achieve the desired goal. These stages include data collection, data preprocessing, CNN architecture design, model training, model evaluation, validation and testing, as well as result analysis. Based on the executed program, the CNN method successfully classifies Huskies and Golden Retrievers effectively. The accuracy that is displayed by the graphs shows an improvement of accuracy from time to time. The model loss also decreases from time to time. In conclusion, the CNN method can achieve high accuracy levels up to 100% in object classification.

1. Introduction

Dog classification is an interesting activity because it supports owners to do special care, identify health issues, and track relevant genetic history, thereby improving the well-being and lives of dogs [1]. Dog classification based on images is an essential task in the field of pattern recognition and computer vision. Convolutional Neural Network (CNN) methods have become one of the most effective and popular approaches for object classification in images [2].

CNN is a type of artificial neural network architecture inspired by how human vision works. The CNN architecture consists of convolutional layers, pooling layers, and fully connected layers that allow the network to automatically extract important features from images and recognize complex patterns [3].

Classifying dogs based on images using CNN methods involves two main stages. The first stage involves training the CNN network using a large amount of pre-categorized dog image data. This data is used to train the neural network to recognize patterns and distinctive features associated with various dog breeds.

During the training process, the CNN network will learn the visual features in images and its label. In this process, the network's weights and parameters are iteratively adjusted to optimise the network's performance in classifying dogs.
Once the training process is completed, the second process is the testing process. In this process, the CNN network is tested by new dog images, and the network's ability to recognize and classify dogs based on those images is measured [4].
CNN methods have proven to be highly effective in dog classification based on images. The main advantage of CNN is its ability to automatically extract important features from images and recognize complex patterns without the need for manually specified features beforehand. This makes this method well-suited for large-scale object classification tasks such as dog classification based on images.With the availability of this method, dog classification based on images can be used in various applications, including finding lost pets, classifying breeds in contests, or even classifying dogs in security systems [5]. In this research, the authors created a CNN Model based on Python programming to classify Golden Retrievers and Huskies. The program classified Golden Retrievers and Huskies based on the images.

2. Theory

2.1. Method

The block diagram of dog classification is shown in Figure 1. The block includes input, process, and output. The input is a dog image. The process would be started by dog image detection. The method to be used is: Convolution Neural Network (CNN). The output of the system is the accuracy value of its detection and classification results of dog classification as the output. The process was simulated by using Visual Studio Code and AMD Ryzen Series 7.

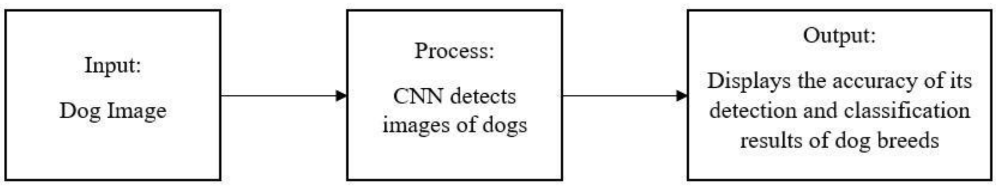

Figure 1. Block Diagram of The Classification of a Dog Process

In the first stage, a dataset of dog images, specifically including Husky and Golden Retriever breeds, is collected. Data is gathered from google a specific quantity for training and testing. The images of Golden Retriever are shown in Figure 2. The images of Siberian Husky are shown in Figure 3.

Figure 2. The images of Golden Retriever

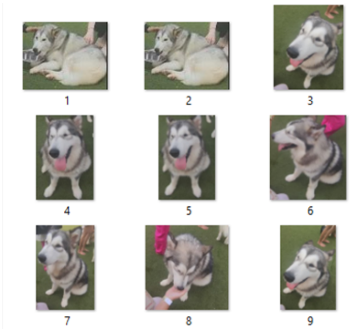

Figure 3. The images of Siberian Husky

2.2. Convolutional neural network (CNN)

A Convolutional Neural Network (CNN) is based on the fundamental principles of artificial neural networks and convolution in signal processing [6]. CNN is built upon a mathematical model inspired by biological neural networks. Neurons in artificial neural networks consist of inputs, weights, activation functions, and outputs. These neurons are connected in a structure that forms a network, where information is passed through a cascading process. Convolution is a mathematical operation that combines two functions to produce a new output. In the context of CNNs, convolution is used to apply filters to input data, such as images. These filters are useful for extracting important features from the input by performing convolution operations on the pixel matrix. The example of CNN structure is shown in Figure 4.

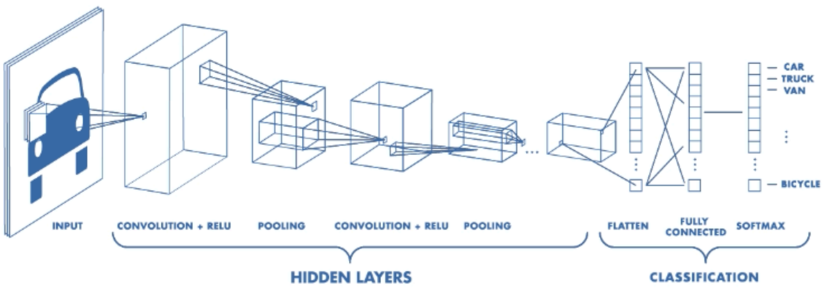

Figure 4. CNN Layers [6]

A Convolutional Neural Network (CNN) consists of several layers that play a role in processing and recognizing features in image data. Here are the main layers in CNN [6]:

1. Convolutional Layer: This layer uses filters or kernels to perform convolution operations on the input. Each filter moves gradually across the image, multiplying pixels by corresponding weights and producing feature maps that highlight important patterns in the image.
2. Activation Layer: After the convolutional layer, the activation layer applies an activation function to each element of the feature maps. Activation functions like

Rectified Linear Unit (ReLU) are used to introduce non-linearity into the network, allowing the model to learn complex patterns.

3. Pooling Layer: The pooling layer is used to reduce the spatial dimensions of the feature maps. Pooling operations, such as max pooling or average pooling, take the maximum or average values from a group of pixels in the feature map. This helps reduce the data size while preserving relevant information.
4. Batch Normalisation Layer: This layer is used to improve network stability and convergence speed. Batch normalisation normalises each batch of input data by adjusting its mean and variance. This helps maintain consistent data distribution and speeds up network training.
5. Dropout Layer: The dropout layer is used to prevent overfitting in the network. In this layer, some random units or nodes are deactivated during training. This forces the network to learn using information from the remaining nodes, promoting better generalisation.
6. Fully Connected Layer: This layer is located at the end of the network and is fully connected to the previous layers. It is responsible for taking the features generated by the previous layers and producing the final output in the form of class probabilities or the desired feature vector.

In addition to these layers, CNNs can also use other layers such as merging layers to combine multiple branches or paths in the network, as well as transpose convolution layers to perform deconvolution or upsampling operations. The combination of these layers allows CNNs to effectively extract important features from image data and recognize complex image patterns. The input size of the image is 150×150×3. The structure that is used in this paper is shown Figure 5.

```
model = keras.Sequential([
    keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    keras.layers.MaxPooling2D((2, 2)),
    keras.layers.Conv2D(64, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D((2, 2)),
    keras.layers.Conv2D(128, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D((2, 2)),
    keras.layers.Conv2D(128, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D((2, 2)),
    keras.layers.Flatten(),
    keras.layers.Dense(512, activation='relu'),
    keras.layers.Dense(2, activation='sigmoid')
])
```

Figure 5. Defines the CNN model used

2.3 Confusion Matrix

A confusion matrix is a fundamental tool in machine learning and classification tasks. It is used to evaluate the performance of a classification algorithm by providing a clear breakdown of the model's predictions compared to the actual ground truth. It is especially useful when dealing with binary classification problems [7] especially in the relation of actual and predicted.

A confusion matrix is organised into four categories :

1. True Positive (TP)
   These are cases where the model correctly predicted the positive class. In other words, the model predicted “yes”, and the actual answer is “yes”
2. True Negatives (TN)

These are cases where the model correctly predicted the negative class. The model predicted “no”, and the actual answer is “no”.

3. False Positives (FP): These are cases where the model incorrectly predicted the positive class when it should have been negative. The model predicted "yes," but the actual answer is "no." This is also known as a Type I error.
4. False Negatives (FN): These are cases where the model incorrectly predicted the negative class when it should have been positive. The model predicted "no," but the actual answer is "yes." This is also known as a Type II error.

Table 1 shows the confusion matrix correlation among the four categories.

Table 1. Confusion Matrix [7]

	Actual Time	**Actual False**
True Predicted	True Positive (TP)	False Positive (FP)
False Predicted	False Negative (FN)	True Negative (TN)

Accuracy is a common metric used to evaluate the overall performance of a classification model. It is calculated as the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances in the dataset. The formula for accuracy is [7]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

3. Result and Discussion

In this paper, we used 300 images that consist of 150 husky images and 150 golden images. Training, validation, and testing were simulated using Python. The training process was optimised using early stopping. The comparison of training, validation, and testing data were conducted for various values of percentage to obtain the optimum result. The result would be evaluated by using training accuracy, validation accuracy, testing accuracy, and accuracy value based on confusion matrix. The result can be seen in Table 2.

Table 2 showed the comparison of training data and testing data in percentage. It started from 90% to 70% as the training data and it started from 10% to 25% as the testing data. The training accuracy value shows that the accuracy is up to 90% and more. The validation accuracy value shows that the accuracy is up to 80% and more. The testing accuracy value shows that the accuracy is up to 76% and more. The accuracy value of the confusion matrix shows that the accuracy is up to 50% and more. The accuracy value trend showed that the less of the training data, obtained less accuracy.

Table 2. Comparison of Training Accuracy, Validation Accuracy, Testing Accuracy, and Accuracy Value

Data Train : Data Test (%)	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Accuracy Value (%)
90:10	92.00	90.00	76.66	75.00
85:15	95.00	100.00	76.66	64.28

80:20	92.71	80.00	76.66	57.40
75:25	89.00	80.00	80.00	58.82
70:30	92.00	90.00	76.66	50.00

The best accuracy value based on the confusion matrix was obtained 75%. The result of the confusion matrix could be shown on Figure 7.

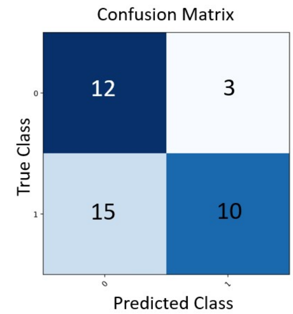

Figure 7. Confusion Matrix Heatmap : True Class and Predicted Class

4. Conclusion

The convolutional neural network (CNN) method has proven to be an effective approach in classifying dog breeds based on images. In this research, the use of CNN for dog breed classification can yield accurate results. Here are some conclusions that can be drawn:

1. CNN has a powerful ability to represent spatial features in dog images.
2. In this study, CNN achieved a testing accuracy value up to 80%.

The development of the CNN method can be expanded to recognize more dog breeds or even other objects. With the addition of more training data and appropriate CNN architecture adjustments, this method can be used to classify various types of animals or objects with a high level of accuracy.

5. References

[1] Heidi G, Lisa V, Nathan B, Scott C, Travis, D, Tiffany B, Gary, S, Hawkins B, Elaine, A, and Leonid, K 2004 Genetic Structure of the Purebred Domestic Dog (Science vol 304 issue 5674) pp. 1160-1164 doi: 10.1126/science.1097406.

[2] Jiuxiang G, Zhenhua W, Jason K, Lianyang M, Amir S, Bing S, Ting L, Xingxing W, Gang W, Jianfei C, and Tsuhan C 2017 Recent Advances in Convolutional Neural Networks (Pattern Recognition vol 77) pp.354 – 377 doi: 10.1016/j.patcog.2017.10.013

[3] Shadman S, Ahmed, Ahmed J, Jawad K, and Hridon A 2019 An Overview of Convolutional Neural Network: Its Architecture and Application (Pre prints.org) pp. 1 – 6 doi: 10.20944/preprints201811.0546.v1

[4] Alex K, Ilya S, and Geoffrey E 2012) ImageNet Classification with Deep Convolutional Neural Networks (Advances in Neural Information Processing Systems) pp.1097–1105

[5] Omkar M, Andrea V, Andrew Z, and C.V. Jawahar 2012 Cats and Dogs (IEEE Conference on Computer Vision and Pattern Recognition) pp. 3498 – 3505 doi: 10.1109/CVPR.2012.6248092

[6] Jianxin W 2015 Introduction to Convolutional Neural Networks (National Key Lab for Novel Software Technology vol 5 issue 23) pp.1 - 31.

[7] Majid R, Afrig A, Ferian F, and Rizky A 2022 Sentiment Analysis of Covid-19 Vaccination using Support Vector Machine in Indonesia (*International Journal of Advanced Computer Science and Applications* vol 13 issue 6) pp.534 - 539.