International Journal of Applied Mathematics and Computing
Vol. 1, No.3 October 2024
e-ISSN: 3047-146X; Page 41-52
DOI ; https://doi.org/10.62951/ijamc.v1i3.29
Available online at: https://international.arimsi.or.id/index.php/IJAMC

Image Classification Comparison Using Neural Network and Support Vector Machine Algorithm With VGG16 As Feature Extraction Method

Aulia Wicaksono [1*], I Putu Eka Nila Kencana [2], I Wayan Sumarjaya [3]
[1-3]Study Program Of Mathematics, Faculty Of Mathematics And Science, Udayana University, Indonesia

Address: Jl. Raya Kampus Unud No.9, Jimbaran, South Kuta, Badung Regency, Bali 80361
*Author Correspondence: wicaksono.aulia2@gmail.com**

Abstract*.Image classification is widely used in everyday life such as in car steering, closed-circuit television (CCTV), traffic cameras, etc. The implementation of image classification can be done using several methods, including neural network and support vector machine models. The neural network method is able to find the right weights that allow the network to show the desired behaviour while the support vector machine method has many dimensions and can overcome linear and non-linear data. In this research, feature extraction was carried out using VGG16 to increase accuracy. This research aims to find out how to implement the neural network and SVM algorithms to classify images and determine the results of analyzing the performance of the two methods. The data used in this study is secondary data consisting of 10 types of large wild cats with a total of 2339 training image datasets and 50 testing image datasets. The research stages consist of data augmentation, model design, model training, and model evaluation. Classification with the neural network model produced an accuracy of 96% and the support vector machine model produced an accuracy of 96%, which means that in a consistent training environment, the two models have the same performance.*

Keywords: *Image classification,Neural network,Support vector machine,VGG16*

1. INTRODUCTION

Classification is a method that aims to allow a machine (computer) to detect a given object and categorize it appropriately. Adult humans can generally classify what they see, hear, or feel, but making machines (computers) that can perform classification methods has led to a lot of research and technological advances in the development of this method. In recent years, many applications of image classification methods for daily activities have emerged. For example, the use of image classification for automatic steering in Tesla cars the implementation of image classification on closed-circuit television (CCTV) cameras and highways to detect traffic violations. These activities have in common the need for large data sources and appropriate data grouping. This has triggered rapid development in the field of deep neural networks as one of the options for implementing classification methods.

Just as humans can evaluate and develop themselves from experience, artificial intelligence has developed a similar concept called machine learning, where machines can learn information from given data based on certain patterns Machine learning has a sub-topic called deep-learning where machines also learn from the given data but with the help of neural network architecture that replicates how neural systems in the human brain work

Received: September 16,2024; Revised: September 30,2024; Accepted:October 12,2024; Published: October 14, 2024

A standard neural network consists of many simple systems connected and learning or called neurons The essence of the neural network learning process is to find the right weights that allow the network to exhibit the desired behavior, such as recognizing patterns or driving a car .The simplest form of neural network system is the feed-forward neural network (FNN) .In an FNN, the information received by the neural network only moves in one direction, namely forward from where the data (input) is received. FNN does not have a cycle (loop) and is commonly used for applications where direct mapping of input data is required, such as image classification.

Along with the times, neural network technology is also growing. Convolutional neural network (CNN) is one of the neural network architectures specifically developed to process image data .CNN consists of several convolution layers, followed by pooling layers, and fully connected layers. The key to CNN is the convolution layers which are useful for processing and capturing spatial features of the image data used, this is also an advantage of CNN when compared to FNN in performing computer vision problems.

Besides CNN, support vector machine (SVM) is also one of the options for image classification.SVM is a model commonly used to solve classification and regression problems in machine learning, this model is proven effective on data that has many dimensions and can overcome linear and non-linear data.In the study “Image Classification using Support Vector Machine and Artificial Neural Network”,SVM used to classify Roman numeral images obtained 86% precision.

As part of the development in the field of neural networks, Simoyan and Zisserman designed a CNN-based neural network model named visual geometry group (VGG). VGG is a convolutional neural network that has 16 or 19 layers (VGG16/VGG19). In ImageNet, the VGG16 model ranked 1st and 2nd on localization and classification test consecutively.The VGG model can perform image classification on 1,000 objects with different categories. This study compares the performance of neural networks and SVM models in image classification. The data used are images of 10 species of wild big cats with a total of 2442 images. To avoid the imbalance of data obtained in the feature extraction process, the author uses the VGG16 model that has been trained using ImageNet data for the feature extraction process.

2. RESEARCH METHODS

The type of data used in this research is secondary data. The image data obtained has JPG format. Data obtained from Kaggle "10 Big Cats of the Wild - Image Classification" (https://www.kaggle.com/datasets/gpiosenka/cats-in-the-wild-image-classification). The data consists of 10 types of big wild cats with a total of 2339 training image data and 50 test image data. The image distribution table of the 10 stray cats can be seen in Table 1 below.

Table 1. Image Distribution of 10 Wild Cats

Jaguar	238
Tiger	237
African Leopard	236
Caracal	236
Puma	236
Cheetah	235
Ocelot	233
Snow Leopard	231
Clouded Leopard	229
Lions	228

Data Augmentation

According to,as an effective way to improve the adequacy and diversity of training data, data augmentation has become an important part of the Convolutional Neural Network model. Data augmentation is a technique to modify data so it fulfils the requirements that the model needed for training.One way to perform data augmentation is by performing image manipulation, this focuses on image transformation, such as rotation, flipping, cropping, and others. Most of these techniques manipulate images directly and are easy to implement.The following is Table 2 which shows the data augmentation method.

Table 2. Data Augmentation Method

Method	Description
Flipping	Flip the image horizontally or vertically
Rotation	Rotate the image at a specific angle
Scaling Ratio	Enlarge or reduce the size of the image
Noise Injection	Add noise into the image
Color Space	Change the color channel of the image with Red Green Blue (RGB), Red Green Blue Alpha (RGBA), Hue Saturation Brightness (HSB), or Hue Saturation Lightness (HSL) options
Contrast	Change the contrast of the image
Sharpening	Change the sharpness of the image
Translation	Move the image horizontally or vertically
Cropping	Cut the image into pieces

Model Design

The model design in this study uses neural network and support vector machine models in the classification process. To prevent data imbalance during model training, researchers also use the VGG16 model to perform feature extraction on the data used by researchers.

Model Training

Model training is the initial stage that aims to process the available data. In this process, the data will go through a feature extraction process using the VGG16 model. After that, the results of feature extraction will go through the neural network and support vector machine models, which respectively perform training and classification of the given data.

Model Evaluation

At this stage, the trained model can be tested for performance with previously prepared test data. The final result of this process is the accuracy and compatibility of the test data with the classification produced by the trained model.

3. RESULTS AND DISCUSSION

a. Data Analysis and Exploration

This stage is carried out to find out an overview of the research data that will be used. The data in this study consists of 2,339 training image data and 50 test image data. The data is divided into 10 big cat classifications with the distribution as in Table 1. First, the resolution of all image data is changed to a size with a height of 224 pixels and a width of 224 pixels. Next, the image augmentation process is carried out on the data owned, the image augmentation performed can be seen in Table 3 below.

Table 3. Image Augmentation Application

Augmentation Type	*Value*	Description
rescale	1/255	Convert the color channel in each image data to grayscale
zoom_range	0.2	Perform size amplification by 20% on each image data
horizontal_flip	True	Change the perspective on each image data horizontally

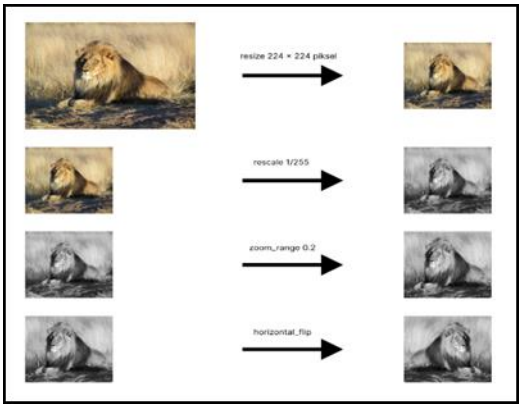

Figure 1. Visualization of image augmentation application

Feature Extraction with VGG16 Model

This research combines two types of models in the training stage, the VGG16 model is used to perform feature extraction on the image data owned and then continue the classification process using two types of models, neural networks and support vector machines, each of which will be compared in performance. The VGG16 model used in this research is obtained from the Tensorflow tf.keras.applications. There are several parameters that need to be considered so that the research can run as intended.

Table 4. Parameter of VGG16

Parameter	***Value***
include_top	False
weights	‘imagenet’
input_shape	224, 224, 3

The feature extraction process performed by the VGG16 model utilizes convolution layers to find a representation of the input image. Convolution layers transform the image into a series of matrix-shaped inputs, after which element wise multiplication is performed on the kernel of the matrix with the filter used.

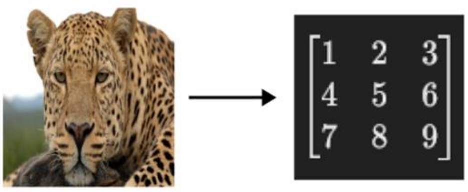

Figure 2. Simulation of image representation against its matrix

This operation can be simulated with the example of a cat image that has a matrix representation of *Input* = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, with *Filter* = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, with a kernel shift (padding) of 1 row/column, the scalar multiplication operation is obtained as follows:

Kernel 1,1	$\sum \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
	$(1 \times 1) + (2 \times 0) + (4 \times 0) + (5 \times 1) = 6$
Kernel 1,2	$\sum \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
	$(2 \times 1) + (3 \times 0) + (5 \times 0) + (6 \times 1) = 8$
Kernel 2,1	$\sum \begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
	$(4 \times 1) + (5 \times 0) + (7 \times 0) + (8 \times 1) = 12$
Kernel 2,2	$\sum \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
	$(5 \times 1) + (6 \times 0) + (8 \times 0) + (9 \times 1) = 14$
Output	$\begin{bmatrix} 6 & 8 \\ 12 & 14 \end{bmatrix}$.

This operation is performed as many layers of the VGG16 model on each input image data. When visualized, the feature extraction process makes the input image have low-level features to high-level features produced by each convolution layers. The output result is a feature map that is ready to be classified by SVM and neural network models.

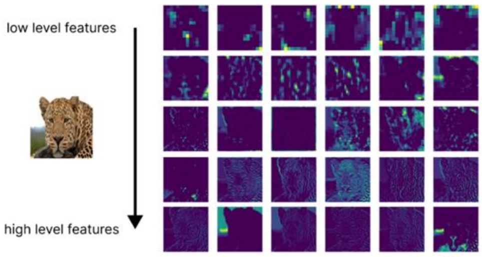

Figure 3. Visualization of feature extraction hierarchy on research data

Classification with Neural Network Model

The model used in this classification training uses VGG16 by disabling the include_top section which uses fully-connected layers with a multilayer perceptron type to fully-connected layers with a singlelayer perceptron type. After going through the training process, the accuracy and loss results of the model that has been designed are obtained. To simulate the classification calculation on the neural network, the author will use the following example:

1. For simulation purposes, we will assume a neural network classification calculation with the following layers:
   a. Input layers: 2 neuron
   b. Hidden layers: 3 neuron with ReLU activation function

e-ISSN: 3047-146X; Page 41-52

c. Output layers: 3 neuron, with softmax activation function to predict 3 different classes

2. Also assume we have two input data in the form of x_1 and x_2 with values:
   a. $x_1 = 0,5$
   b. $x_2 = 1,2$
3. Also assume we have weights and biases for the input layers and output layers with values:
   a. Neuron input 1: $w_{11} = 0{,}2$, $w_{12} = 0{,}5$, $bias = 0{,}1$
   b. Neuron input 2: $w_{21} = 0{,}4$, $w_{22} = 0{,}3$, $bias = -0{,}2$
   c. Neuron input 3: $w_{31} = 0{,}3$, $w_{32} = 0{,}8$, $bias = 0{,}05$
   d. Neuron output class 0: $w_{01} = 0{,}6$, $w_{02} = -0{,}1$, $w_{02} = 0{,}2$, $bias = -0{,}05$
   e. Neuron output class 1: $w_{11} = -0{,}3$, $w_{12} = 0{,}2$, $w_{13} = 0{,}4$, $bias = 0{,}1$
   f. Neuron output class 2: $w_{21} = 0{,}1$, $w_{22} = 0{,}5$, $w_{23} = -0{,}2$, $bias = 0$
4. With this we can calculate the classification of the neural network:

Input layers

$$X = \begin{bmatrix} 0{,}5 \\ 1{,}2 \end{bmatrix}$$

Hidden layers neuron 1

$$z_1 = (0{,}2 \times 0{,}5) + (-0{,}5 \times 1{,}2) + 0{,}1 = 0{,}1 - 0{,}6 + 0{,}1 = -0{,}4$$

ReLU function application

$$a_1 = max(0, -0{,}4) = 0$$

Hidden layers neuron 2

$$z_2 = (0{,}4 \times 0{,}5) + (0{,}3 \times 1{,}2) + (-0{,}2) = 0{,}2 + 0{,}36 - 0{,}2 = 0{,}36$$

ReLU function application

$$a_2 = max(0, 0{,}36) = 0{,}36$$

Hidden layers neuron 3

$$z_3 = (-0{,}3 \times 0{,}5) + (0{,}8 \times 1{,}2) + 0{,}05 = -0{,}15 + 0{,}96 + 0{,}05 = 0{,}86$$

ReLU function application

$$a_3 = max(0, 0{,}86) = 0{,}86$$

Output hidden layers result

$$X = \begin{bmatrix} 0 \\ 0{,}36 \\ 0{,}86 \end{bmatrix}$$

Output layers class 0

$$z_0 = (0{,}6 \times 0) + (-0{,}1 \times 0{,}36) + (0{,}2 \times 0{,}86) + (-0{,}05)$$
$$= 0 - 0{,}036 + 0{,}172 - 0{,}05 = 0{,}086$$

Output layers class 1

$$z_1 = (-0{,}3 \times 0) + (0{,}2 \times 0{,}36) + (0{,}4 \times 0{,}86) + 0{,}1$$
$$= 0 + 0{,}072 + 0{,}344 + 0{,}1 = 0{,}516$$

Output layers class 2

$$z_2 = (0{,}1 \times 0) + (0{,}5 \times 0{,}36) + (-0{,}2 \times 0{,}86) + 0{,}0$$
$$= 0 + 0{,}18 - 0{,}172 = 0{,}008$$

Output before softmax activation function

$$z = \begin{bmatrix} 0{,}086 \\ 0{,}516 \\ 0{,}008 \end{bmatrix}$$

Softmax activation function

$$softmax(z_j) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Convert the output into exponential form

$e^{0,086} = 1,0899$, $e^{0,516} = 1,675$, $e^{0,008} = 1,008$

Exponential total

$$sum = 1,0899 + 1,675 + 1,008 = 3,7729$$

Probability class 0

$$P(class\ 0) = \frac{1,0899}{3,7729} = 0,289$$

Probability class 1

$$P(class\ 1) = \frac{1,675}{3,7729} = 0,444$$

Probability class 2

$$P(class\ 2) = \frac{1,008}{3,7729} = 0,267$$

Output classification probability

$$P = \begin{bmatrix} 0,289 \\ 0,444 \\ 0,267 \end{bmatrix}$$

The following process is carried out on 512 neurons of the neural network model used in this study, with input from feature extraction in the previous VGG16 process with 10 different classes. After going through the classification process, the confusion matrix, accuracy, and loss results of the designed model were obtained.

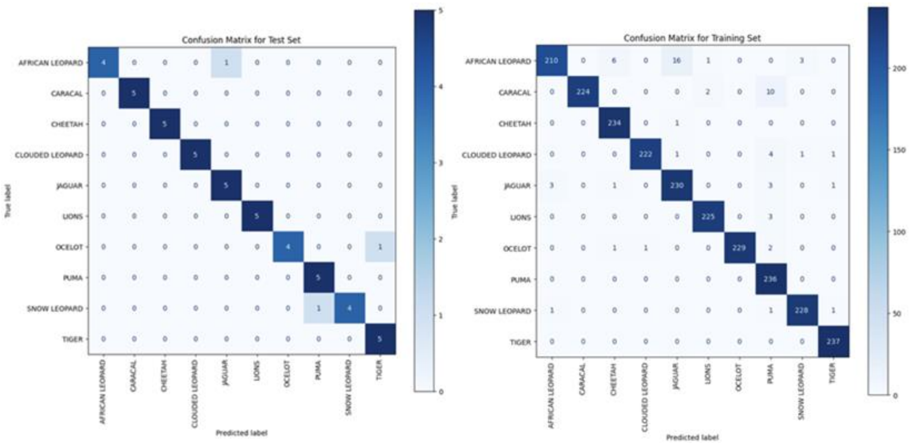

Figure 4. Confusion matrix of artificial neural network model training results

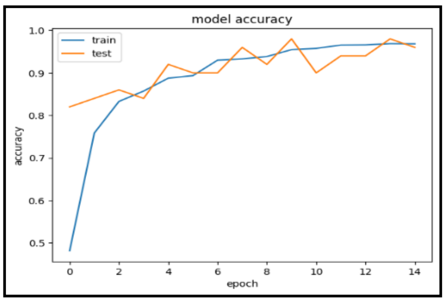

Figure 5. Neural network model training accuracy

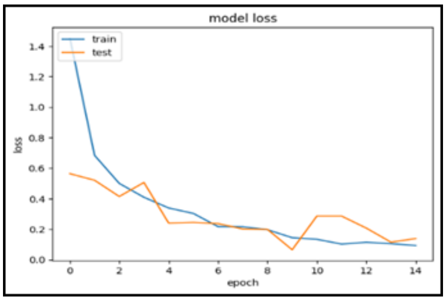

Figure 6. Neural network model validation accuracy

The final results of the classification using the neural network model are as follows:

Table 5. Neural Network Model Evaluation

Loss	Accuracy	Validation Loss	Validation Accuracy
9,27%	96,84%	13,84%	96%

Classification with Support Vector Machine Model

The model used in this classification training uses VGG16 by disabling the include_top section which uses fully-connected layers with a multilayer perceptron type into a support vector machine with a linear type. To simulate the classification calculation in SVM, the author will use the following example:

1. Assume we have a dataset with the following example:

Table 6. Data Points Simulation for SVM Model

Feature 1 (x_1)	Feature 2 (x_2)	Class (y)
1.0	2.0	0
2.0	3.0	0
2.0	0.0	1
3.0	1.0	1
3.0	4.0	2
4.0	2.0	2

1. With the OvR method, we will perform the following classification:
   a. Classification 1: Class 0 vs Rest
   b. Classification 2: Class 1 vs Rest
   c. Classification 3: Class 3 vs Rest
2. Classification equation

   For the class 0 vs Rest, we will relabel it as follows:

   Class 0 (positive): $y = 1$

   Class 1 dan 2 (negative): $y = -1$

   Dataset will be transformed into the following:

Table 7. Data Points Simulation for SVM Model after processing

Feature 1 (x_1)	Feature 2 (x_2)	Class (y)
1.0	2.0	1
2.0	3.0	1
2.0	0.0	-1
3.0	1.0	-1
3.0	4.0	-1
4.0	2.0	-1

With linear SVM, the hyperplane decision boundary is as follows:

$$w_1x_1 + w_2x_2 + b = 0$$

After the data is trained, assume the decision boundary results as follows:

$$-0,5x_1 + 2,5 = x_2$$

Do the same for Class 1 vs Rest and Class 2 vs Rest, and assume the following decision limits for each as follows:

$$1,5x_1 - 2 = x_2$$

$$-x_1 + 5 = x_2$$

3. Assume we will classify a new data point with point (2.5 , 2.5). We will use the three classification equations to determine the class of the data point from the corresponding values.

   Classification 1

$$2.5 = -0.5(2.5) + 2.5$$
$$2.5 = 1.25$$

Classification 2

$$2.5 = 1.5(2.5) - 2$$
$$2.5 = 1.75$$

Classification 3

$$2.5 = -(2.5) + 5$$
$$2.5 = 2.5$$

The classification result closest to the hyperplane decision boundary is Classification 2 in Class 1.

This method was performed repeatedly with 10 classes and 2339 training data and 50 test data after the feature extraction process on VGG16 was completed. After going through the training process, the accuracy result of the designed model validation data is 96%.

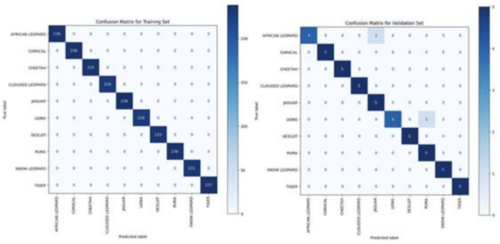

Figure 7. Confusion matrix of artificial SVM model training results

4. CONCLUSIONS

Based on the results of the research that has been carried out in the previous chapter, several conclusions are obtained, namely:

1. Classification with neural network model uses the VGG16 model for the feature extraction method and replaces the multilayer perceptron with a single layer perceptron for the classification process. The trained model produces 96% accuracy.
2. Classification with support vector machine model using VGG16 model for feature extraction method and replacing multilayer perceptron to support vector machine for classification process. The trained model produced an accuracy of 96%.
3. With a consistent training environment, it can be concluded that linear neural network and support vector machine models for classification have similar performance.

5. REFERENCES

Aggarwal, C. C. (2018). Neural networks and deep learning. Springer International Publishing. https://doi.org/10.1007/978-3-319-94463-0

Aremu, T., Zhiyuan, L., Alameeri, R., Khan, M., & El Saddik, A. (2023). SSIVD-Net: A novel salient super image classification & detection technique for weaponized violence. arXiv. https://arxiv.org/abs/2207.12850

Balestriero, R., Misra, I., & LeCun, Y. (2022). A data-augmentation is worth a thousand samples: Exact quantification from analytical augmented sample moments. arXiv. https://arxiv.org/abs/2202.08325

Buhot, A., & Gordon, M. B. (1998). Learning properties of support vector machines.

Chib, P. S., & Singh, P. (2023). Recent advancements in end-to-end autonomous driving using deep learning: A survey. arXiv. https://arxiv.org/abs/2307.04370

Dawson-Howe, K. (2014). A practical introduction to computer vision with OpenCV. Wiley.

Dede, D., Sarsıl, M. A., Shaker, A., Altıntaş, O., & Ergen, O. (2023). Next-gen traffic surveillance: AI-assisted mobile traffic violation detection system. arXiv. https://arxiv.org/abs/2311.16179

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Kumar, T., Mileo, A., Brennan, R., & Bendechache, M. (2023). Image data augmentation approaches: A comprehensive survey and future directions. arXiv. https://arxiv.org/abs/2301.02830

Murat, H. S. (2006). A brief review of feed-forward neural networks. Communications Faculty of Science University of Ankara, 50(1), 11–17. https://doi.org/10.1501/commua1-2_0000000026

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks.

Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. Engineering Applications of Artificial Intelligence, 60, 97–116. https://doi.org/10.1016/j.engappai.2017.01.013

Riera, I., Rana, A., & Lorente, Ò. (2021). Image classification with classic and deep learning techniques.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

Sultana, F., Sufian, A., & Dutta, P. (2019). Advancements in image classification using convolutional neural network. Proceedings of the 2018 International Conference on Robotics, Control, and Information Technology (ICRCICN). https://doi.org/10.1109/ICRCICN.2018.8718718

Thai, L. H., Hai, T. S., & Thuy, N. T. (2012). Image classification using support vector machine and artificial neural network. International Journal of Information Technology and Computer Science, 4(5), 32–38. https://doi.org/10.5815/ijitcs.2012.05.05

Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., & Shen, F. (2022). Image data augmentation for deep learning: A survey.