DoubleClick: Journal of Computer and Information Technology E-ISSN: 2579-5317
P-ISSN: 2685-2152
Vol. 7, No. 1, August 2023, Pages 9-15 http://e-journal.unipma.ac.id/index.php/doubleclick

Detection of Motorcycle Tire Endurance based on Tire Load Index using CNN

Birawa Kaca Buana Gora[1], Nugroho Tegar Maulana[2], Muhamad Novan Aulia Zam Zami[3], Anan Nugroho[4], Alfa Faridh Suni[5]
[1,2,3]S1 Informatics and Computer Engineering Education, Semarang State University
[1]birawakaca01@students.unnes.ac.id; [2]exokai.bg@students.unnes.ac.id;
[3]novanauliazami@students.unnes.ac.id; [4]anannugroho@mail.unnes.ac.id;
[5]alfafs@mail.unnes.ac.id

Abstract: *With increasingly rapid technological developments, the production of motorized vehicles will increase with the use of robotic power in production. The increasing number of motorized vehicles in big cities does not escape the rise of traffic accidents that occur. One aspect of accidents that we usually underestimate is the resistance of our vehicle tires to support the load on the vehicle. Therefore, we need a system to detect the resistance of a tire in supporting the load on the vehicle. For this reason, this study was conducted to detect the durability of motorcycle tires based on tire load index using a convolutional neural network. A 70% result was found in classifying tire resistance based on tire load index.*

Keywords*: Tire load index, Convolutional Neural Network, Motorcycle, Endurance*

Introduction

With increasingly rapid technological developments, the production of motorized vehicles will increase with the use of robotic power in production. In the book Potret Lalu Lintas in Indonesia 2019 Edition 3, the population of motorized vehicles throughout Indonesia in 2018 was 141,428.052 units, and 81.58% of them were motorcycles. The rest were 16,440,987 units of four-wheeled vehicles or passenger cars, which means motorcycles are the type of vehicle that dominates on the highway (Badan Pusat Statistik, 2018).

Reported to the Asosiasi Industri Sepeda Motor Indonesia (AISI), based on the last three years, in 2019 there were 6,487,460 units of motorcycles sold, then in 2020 there were 3,660,616 motorcycles, and in 2021 there were 5,057, 516 (Asosiasi Industri Sepeda Motor Indonesia, 2022). There was a decline in 2019 to 2020 due to the COVID-19 pandemic and an increase in 2020 to 2021 after the COVID-19 pandemic subsided. This means that the motorcycle population in Indonesia is increasing again as the economy recovers. The increasing number of motorized vehicles in big cities does not escape the rise of traffic accidents that occur. One aspect of accidents that we usually underestimate is the resistance of our vehicle tires to support the load on the vehicle. In an article written by Aprida Mega Nanda (2021), it is stated that drivers will find it difficult to maneuver when driving on the highway because of the heavy load of the vehicle, which makes the vehicle difficult to control. Not only that, an overloaded load can also cause damage to the motor engine. This is because the engine rotation is high but not proportional to the speed of movement. Therefore, we need a system to detect the resistance of a tire in supporting the load on the vehicle.

The authenticity of this study is based on several previous studies that have relatively the same characteristics in terms of the theme of the study, although they differ in terms of subject criteria, number of subjects, and the analytical method used. Research will be conducted on the resistance of a tire in supporting the load on the vehicle. A convolutional neural network is used to train the program to detect the load

index of tires 40 and 46 using the load index image of the existing tires. Pulung Adi Nugroho et al. (2020) conduct research on objects of human expression.

The similarity between the research conducted by Pulung Adi Nugroho and the research conducted by the researcher is that they both use the convolutional neural network method in the program training process. While the difference lies in the object under study.

Through a search conducted by researchers, no research was found that aims to detect tire load index using a convolutional neural network to determine the resistance of a tire in supporting the load on the vehicle. So it can be said that this research is the first research in this regard.

Theoretical Basis

Deep Learning

Deep learning is some kind of artificial neural network algorithm that utilizes metadata as input and analyzes it using several hidden layers of non-linear transformations of the input data to get the output value. The ability to autonomously extract data is a special characteristic of deep learning algorithms. This indicates that, while addressing a problem, its algorithm can automatically identify the relevant characteristics. The neural network owned by deep learning is formed from a simple hierarchy with several layers to a high level or many layers (multi layer). Based on this, deep learning can be used to solve complex problems that are more complicated and consist of a large number of non-linear transformation layers (LeCun, Bengio and Hinton 2015).

Keras Framework

One of the packages used to solve neural network-related problems is Keras. With a focus on optimizing experiments in convolutional and recurrent processes in neural networks, or between the two, Keras supports practically all neural network models. It is not necessary to write the code line by line in order to verbally convey the mathematical computations involved in creating a neural network model. This is due to the fact that Big Brother has offered various fundamental CNN models that have been enhanced to support deep learning research. The CPU or GPU may effectively execute the hard package's computing operation (Shafira, 2018).

Tensorflow

TensorFlow is a Google-developed open-source machine learning library that supports many programming languages (Yao, et al. 2019). Tensorflow is used in the Transfer Learning process to process the Inception-v3 Model for retraining with fresh data and then build a classifier with rapid computation and high accuracy. Tensorflow is compatible with various systems.

Optimizer

The optimization algorithm attempts to minimize errors, identify the ideal weights, and increase accuracy. The model's parameters (weights) are modified during training in an effort to minimize the loss function and provide highly accurate predictions. The optimizer is used in this situation since it is still uncertain when, how much, and exactly to adjust. By modifying the model in response to the loss function's output, they combine the model parameters and the loss function. To put it simply, the optimizer uses our model's weight to shape it into the most realistic form (Sumardi, 2019).

RMSprop

RMSprop is an optimizer that takes advantage of the latest gradient magnitude to normalize the gradient,

which keeps the moving average above the root mean square gradient hence the name Rms. $f'(\theta_t)$ becomes the derivative of loss with respect to the parameter in the time step t. In its basic form, given the step rate α and the decay term γ, the following updates are made:

$$r_t = (1-\gamma)\, f'(\theta_t)^2 + \gamma r_{t-1}$$

$$v_{t+1} = \frac{\alpha}{\sqrt{r_t}}\, f'(\theta_t)$$

$$\theta_{t+1} = \theta_t - v_{t+1}$$

Information:
α : learning rate
r_t : exponential mean of gradient box
θ_t : gradient at time t along v^j

The benefit of RMSProp is that it has pseudo-curvature information and is a very strong optimizer. It also applies to learning in tiny batches since it can effectively overcome stochastic objectives (RMSProp, 2013).

Adam

Adam (Adaptive moment estimation) is a frequently used and well-liked optimization algorithm in deep learning. Instead of using the conventional stochastic gradient descent method to iteratively update the network weights based on the training data, Adam is an optimization approach that may be employed. To create an optimization technique that can handle diffuse gradients, Adam's optimization combines the best elements of the RMSProp and AdaGrad algorithms. The Adam Optimizer goes through the following steps:

1. Fixed first and second moment bias

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

2. Calculating first and second moment bias

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)\,.\, g$$

$$v_t \leftarrow \beta_1 v_{t-1} + (1-\beta_2)\,.\, g^2$$

3. Fixed parameters

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}$$

Information:
α : *learning rate*
g : *gradient*
m : *first moment*
v : *second moment*
β_1, β_2 : *Exponential decay rates*
θ : parameters to be fixed

Up until all epochs have been run, the Adam optimizer step is repeated as many times as the number of datasets selected at random. Because Adam has a bias correction throughout the calculation stage, the difference between RMSProp and Adam is in the step size change at the beginning of the parameter change (Karim, 2018).

Method

Convolutional Neural Network (CNN)

One of the most popular and widely used DL networks is the convolutional neural network (CNN) (Yao, et al. 2019). CNN has made DL more popular in recent years. CNN's key benefit over its predecessors is that it automatically recognizes significant characteristics without the need for human intervention, making it the most widely utilized. Because of the high and numerous network depths applied to picture data, CNN is included in this type of Deep Neural Network. CNN is a dimensionless vector high that will require some parameters to describe the network. It is used to study visual pictures, and identify and recognize items on images. In terms of structure, CNN is similar to traditional neural networks. CNN is made up of neurons with different weights, biases, and activation functions. The kernel in CNN is used to combine spatial

features with a spatial form that resembles the input medium. Then CNN uses various parameters to reduce the number of variables to make it easier to study (Khan, et al. 2018). The name "Convolutional Neural Network" indicates that the network uses a mathematical operation called convolution (Goodfellow, Bengio, and Courville 2016). The CNN is then trained to study the characteristics of the object to predict it. The CNN illustration in this study can be seen in Figure 1.

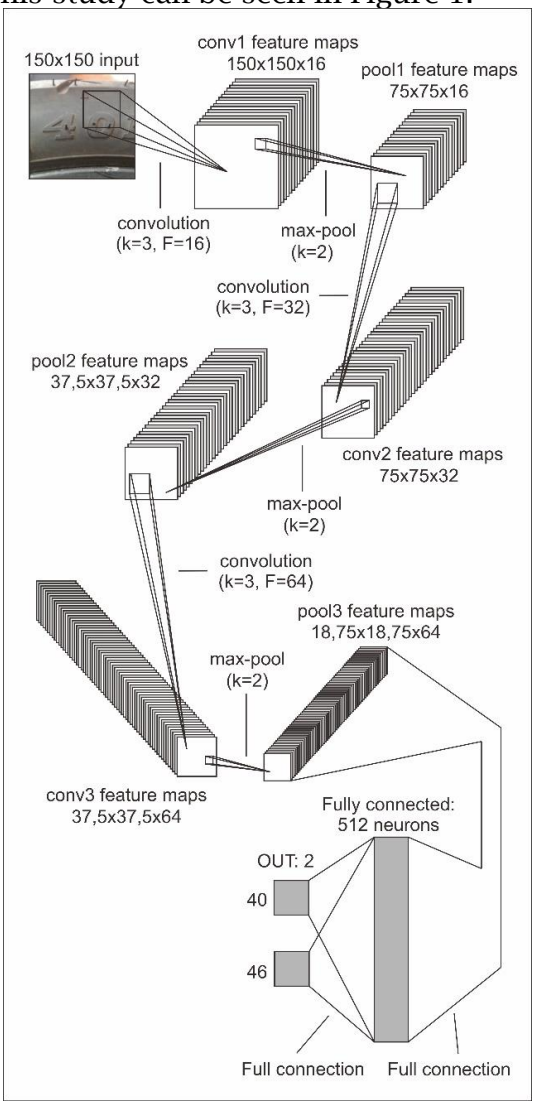

Figure 1. CNN Architecture used for load index tires detection

Activation Function ReLU ((Rectified Linier Unit)

The most typical and fundamental method of incorporating non-linearity into neural networks is through rectified linear units, or ReLU. The ReLU activation function is commonly stated in the equation below.

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

According to this equation, if the input is negative, the neuron's output value can be declared to be 0. The output of the neuron is the value of the activation input itself if the input value of the activation function is positive. When compared to the sigmoid and tanh functions, this activation function has the benefit of being able to accelerate the Stochastic Gradient Descent (SGD) configuration process. However, this activation also has a disadvantage, which is that it might grow fragile through training and cause the unit to death (Nurhikmat, 2018). The picture below shows how ReLu's activation function works.

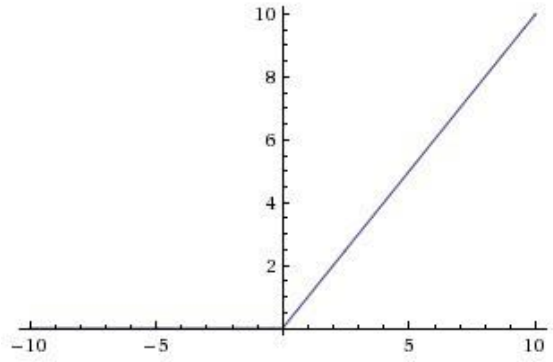

Figure 2. ReLU Activation Function

There are various phases involved in carrying out the steps of this research, including data gathering, selecting a parameter, data preprocessing, data sharing, data training, and testing.

Data Collection

The method of gathering datasets includes capturing the research object. The tire load index is captured as a whole, with the tire load

index number facing the camera, to obtain the object data. Each photograph has a varying distance between the camera and the subject.

Selecting a Parameter

The parameter values for this phase were determined throughout the CNN modeling process. The number of epochs, batches, picture pixels, learning rates, filters, convolutional layer size, and pooling layer size is among the factors employed.

Data Preprocessing

The picture must first be preprocessed before the data training procedure can begin. The first step in the preprocessing procedure is to reduce the image's original pixel size to 150 by 150 pixels. The Jupyter Notebook program and the OpenCV Python package were used in this study to analyze processes.

Data Sharing

Following the preprocessing step, all tire load index picture data is divided into three groups: training data, validation data, and test data. The validation and training data are separated into 20 classes, each comprising 20 tire load index images. Meanwhile, 10 tire load index images are included in the test data.

Data Training and Testing

Following the data sharing phase, data training is performed with the specified number of epochs on the training and validation datasets. The test phase is completed by putting an image from the test data into the CNN network model that has been developed. The testing method is used to evaluate the tire's load index detection level.

Result

This study uses image data derived from the documentation of researchers on motorcycle tires around the research test environment, with a total of 40 images. For training data, there are 15 images of the tire load index with a value of 40 and 15 images of the tire load index with a value of 46. Then the other 10 images are used as image tests. The 10 images can be shown in Figure 1.

Figure 3. Test Dataset

The test was carried out with 10 test images with batch sizes of 1 and 5, learning rate = 0.001, output layer = 2, and epochs of 50 and 100. The optimizer tested was Adam. The test results on each tire image can be shown in Table 1.

Table 1. Tire image test results

Data ke -	Epoch 50 & batch 1		Epoch 50 & batch 5		Epoch 100 & batch 1		Epoch 100 & batch 5	
	40	46	40	46	40	46	40	46
1		F		F		F		F
2	T		T		T			F
3		F		F		F		F
4	T		T		T		T	
5	T			F		F		F
6	F			T		T	F	
7		T		T		T		T
8	F			T		T		T
9		T		T		T		T
10		T		T		T		T

From the test results above, a bar graph was made to determine the percentage of success for each condition, which can be shown in Figure 4.

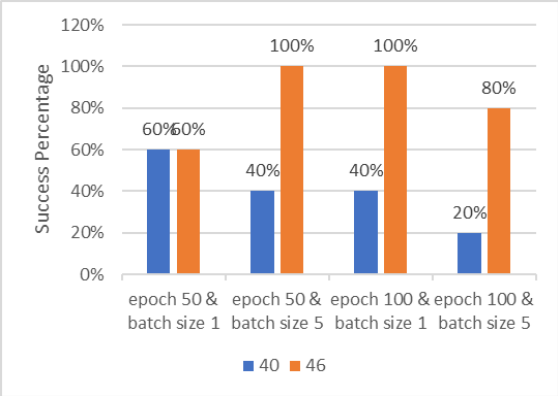

Figure 4. Success percentage

Based on the test results above, it can be seen that the best learning accuracy results are shown in the process using epochs of 50 and batch size 5, and epochs of 100 and batch size 1. An epoch of 50 and batch size of 5 can detect 2 out of 5 or 40% of images. which has a tire load index of 40 and can detect 5 out of 5 or 100% of images that have a tire load index of 46. Then the epoch of 100 and batch size 1 can detect 2 of 5 or 40% of images that have a tire load index of 40 and can detect 5 out of 5 or 100% of images that have a tire load index of 46. Both conditions have accuracy and loss, which can be shown in Figure 5.

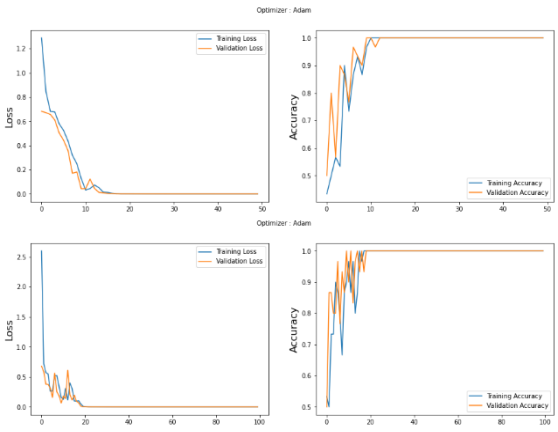

Figure 5. Accuracy and Loss epochs of 50 and batch 4, and epochs of 100 and batch size 1

Discussion

Based on the results of the research that has been done, it is known that the convolutional neural network program in the Python programming language runs quite well if it has enough training data. The dataset used by the researcher consisted of 40 images, with 15 images representing the tire load index value of 40 and the other 15 images representing the tire load index value of 46, while the remaining 10 images were used for testing. For this study, researchers used the Adam optimizer for testing and ran tests with different configurations such as different batch sizes, learning rates, output layer sizes, and epochs.

To analyze the performance of the model, the test results for each tire image were categorized by load index scores of 40 and 46 and displayed in table chart. In addition, researchers created a bar chart showing the success rate for each test condition which can be seen in Figure 4. The chart shows that the best learning accuracy is achieved with the following two configurations: 50 epochs for batch size 5, 100 epochs for batch size 1.

The first configuration using epoch of 50 and batch size of 5 recognized 2 out of 5 or 40% of the images at a tire load index of 40, while achieving a 100% success rate in identifying all 5 out of 5 images with a tire load index of 46. Similarly, in the second configuration (epochs of 100, batch size of 1), 2 out of 5 or 40% of the images with a tire load index of 40 were recognized, and all of 5 images with a tire load index of 46 were correctly identified. The results also state that both configurations come with accuracy and loss epoch metrics, which are shown in Figure 5.

In summary, the research's results indicate that the best learning accuracy was achieved at epoch 50 for a batch size of 5, and batch size 1 at epoch 100. These configurations show promising results in detecting motorcycle tire load index values, with

an average percentage of 70% for both types (40 and 46) and are considered good tire load index detection. For future applications, it is recommended to have at least 30 images for each type of training data so that the tire load index can be determined accurately. The other settings utilized a learning rate of 0.001 and the Adam Optimizer.

Conclusion

Through the results of research that has been carried out, the percentage of 70% for the best learning accuracy results is with epoch 50 and batch size 5, and epoch 100 and batch size 1. A tire load index of 40 can only be detected by 20% and a tire load index of 46 can be detected by 100%. As for conducting training programs using a learning rate of 0.001 and the optimizer tested was Adam.

In future implementations, there will be a minimum of 30 images for each type of training data so that the tire load index identification process can produce the right results.

Reference

Asosiasi Industri Sepeda Motor Indonesia. (2020). *Statistic Distribution*. Retrieved June 23, 2022, from www.aisi.or.id: https://www.aisi.or.id/statistic/

Badan Pusat Statistik. (2018). *Statistik Transportasi Darat 2018*. Jakarta: Badan Pusat Statistik.

Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies, and applications to object detection. Progress in Artificial Intelligence, 9(2), 85-112.

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

Karim, H., Niakan, S. R., & Safdari, R. (2018). Comparison of neural network training algorithms for classification of heart diseases. IAES International Journal of Artificial Intelligence, 7(4), 185.

Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A guide to convolutional neural networks for computer vision. *Synthesis lectures on computer vision*, 8(1), 1-207.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Nanda, A. M. (2021, March). Masih Disepelekan, Bahaya Muatan Berlebih Bagi Pengendara Motor. Retrieved June 23, 2022, from https://otomotif.kompas.com/read/2021/03/29/112200315/masih-disepelekan-bahaya-muatan-berlebih-bagi-pengendara-motor

Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia. *Algor*, *2*(1), 12-20.

Nurhikmat, T. (2018). Implementasi deep learning untuk image classification menggunakan algoritma Convolutional Neural Network (CNN) pada citra wayang golek.

RMSProp. (2013). Retrieved June 21, 2022, from climin.readthedocs.io: https://climin.readthedocs.io/en/latest/rmsprop.html

Shafira, T. (2018). Implementasi Convolutional Neural Networks Untuk Klasifikasi Citra Tomat Menggunakan Keras. Yogyakarta: Skripsi UII.

Sumardi, D. G. (2019). Implementasi Algoritma CNN Dalam Klasifikasi Gangguan Mata Menggunakan Pendekatan Image Processing. Yogyakarta: Skripsi UII.

Yao, G., Lei, T., & Zhong, J. (2019). A review of convolutional-neural-network-based action recognition. Pattern Recognition Letters, 118, 14-22.