Real-Time Detections of Opened-Closed Eyes Using Convolutional Neural Network

Firman Matiinu Sigit[1], Rahmad Syaifudin[2], Dwi Ari Suryaningrum[3]
Firman.matiinu@gmail.com
Electrical Engineering Study Program
Faculty of Engineering,
Tulungagung University

Abstract

The sleepy condition can affect changing behaviors in the human body, and one part of the human body that gets this effect is the eye; eyes are narrower than in normal conditions, and the frequency of blinking eyes is going to increase when people are sleepy. In this study, we will study the behavior of eyes, opened and closed eyes that the camera can capture in real-time, and tools of image processing that can capture and track eyes. Data images from this treatment are fed into Convolutional Neural Network (CNN) as data learning, so CNN can recognize opened and closed eyes from those eyes. In this study, we will characterize tools of image processing (Haar cascade Method) combined with CNN and their performance to detect opened-closed eyes in real-time detections.

In this study, we use two CNN models as a comparison; the first CNN model uses 1 layer with 2 nodes, and the second CNN model uses 2 layers, with the first layer with 500 nodes and the second layer with 2 nodes; the output of each CNN has two targets namely 'open-label eyes and 'close' label eyes. The image dataset contains 20000 eye images, i.e., 10000 'open' eye images and 10000 'close' eye images. The image dataset is trained into two CNNs so that we have two CNN models: the one-layer CNN model and the two-layer CNN model. Each of those models has a pre-trained network.

Each pre-trained model CNN is tested to detect opened-eyes and closed-eyes in real time. There are ten different people. For example, in this experiment, each person was subjected to ten trials of 'opening' and 'closing' eye detection and counted successfully detecting and failing to detect; from all the sample people tested, it can be concluded that the percentage was successful in detecting and percentage failed to detect. The Two-layers CNN model has a 55 % success rate in this experiment.

***Keywords**: Convolutional Neural Network (CNN), Dataset, Haar cascade Method, Eyes.*

I. INTRODUCTION

Studies about face recognition and detect changing behavior of faces are massive growth nowadays; before data abundance like today, studies about those are still limited to image processing techniques; we cannot learn from data even though the data has its characteristics, so research on facial recognition becomes difficult to do. Now, along with the development of the amount of data, knowledge is also developing that studies data classification. One of the data classification tools is machine learning. For classifying large amounts of data using deep learning CNN is one type of deep learning.

The human facial image dataset has certain information that can be classified, for example, recognizing a person's face and facial expressions, one of which is detecting the eyes opening and closing. Here, the haar cascade method combined with CNN is studied for its ability to detect changes in eye-opening and closing in real-time to sleep detection research in someone driving a car.

We all know that many countries struggle to reduce traffic accidents; one person is killed every 25 seconds in traffic accidents [1]. In 2015, WHO released that every year around the world, more than 1.25 million people died and 50 million severe injuries because of traffic accidents [2]. Most traffic accident survivors be poor and need mental rehabilitation to heal their traumatic [3]. The most common factors that cause traffic accidents are human errors such as inexperience, skill levels, and novice young drivers [4]. In Indonesia, according to data from the police, every hour, three people die from traffic accidents [5]. Currently, Indonesia is building much infrastructure, one of which is toll roads; in 2024 Indonesian Government has a target to build toll roads along 5000 km [6]. This will take less time to travel from city to city, but it should be remembered that 'less time' needs more speed, which means getting a higher risk, so to avoid this situation, the Government suggests taking a rest after 4 hours long driving [7]. Studies about technology safety cars to avoid traffic accidents have always been developed, such as pedestrian assist, lane keeping assist, and camera yawning detections, and one of those technologies that are still going to develop is a safety car based on data. One of the upstanding technology based on data we can get peculiarities from that data. One data compared with another data has its peculiarities. For example, my facial image data compared to another person's facial image data has its peculiarities, so never be the same. This will get the advantages if we want to study technology that needs authority; one of these examples of technology is driver drowsiness detection.

II. RESEARCH METHODS

The research method that is used in this study is shown in the following diagram block:

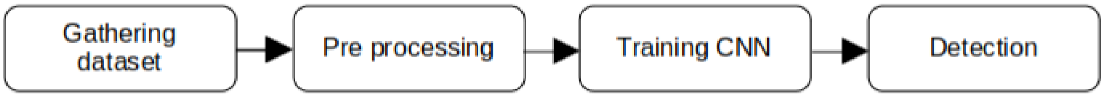

Figure 1. The diagram blocks this research

A. Gathering Dataset

In gathering datasets in this research, we use an image processing tool called the Haar cascade method. The ability of Haar cascade method can detect and track faces or eyes, like the following picture:

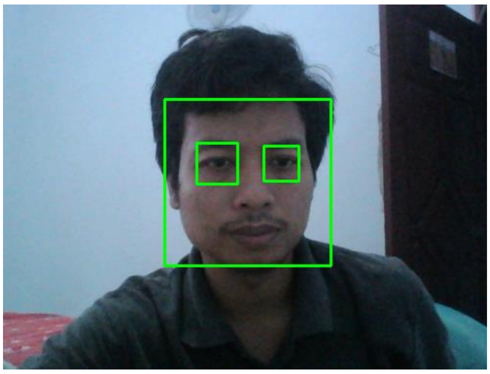

Figure 2. Haar cascade method to track face and eye [8]

Data images from real-time video tracked by haar cascade are saved as image datasets and divided by two class labels, which are 'open' and 'close' eye labels. Here, we use 20000 eye images gathered from people of different ages, gender, and race.

For example, picture datasets that are used for training CNN in this research are like the following picture:

Figure 3. Opened eyes picture example of a dataset

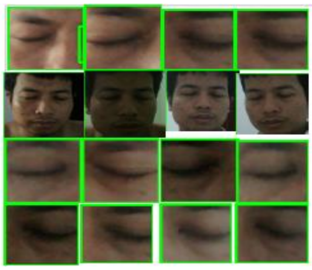

Figure 4. Closed eyes image dataset [8]

B. Pre-processing

Now, we have a dataset after compiling an amount of image data from the haar cascade; before we train this dataset into CNN, we need to pre-process this image dataset. We all know that this image dataset has many shapes of images, and their shape is not similar, so we want to resize all of those images to have one similar shape of images. Hence, we need the function *resize* from the open CV library and convert this image to a grayscale. Finally, we can get an image dataset with grayscale and similar shape (here, we used a shape image with width = 32, length = 32, depth = 1) and depth = 1 because of the grayscale image. Furthermore, we are ready to feed this image dataset into CNN.

C. Training CNN

In this study, we are using two kinds of CNN architecture to be compared; first, CNN is a very shallow CNN that has only one layer with 2 nodes, the architecture of that CNN is like the following picture:

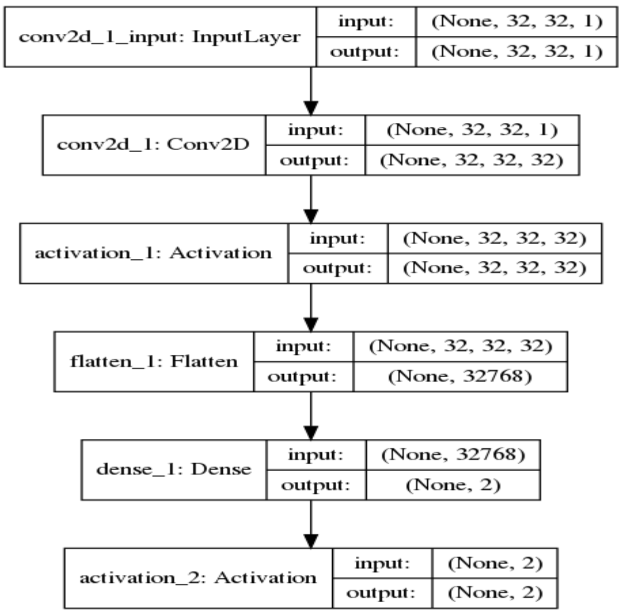

Figure 5. One layer 2 nodes shallow CNN architecture [8]

The second CNN model has two layers, with 500 nodes in the first layer and 2 in the second. Here is the picture architecture of this CNN:

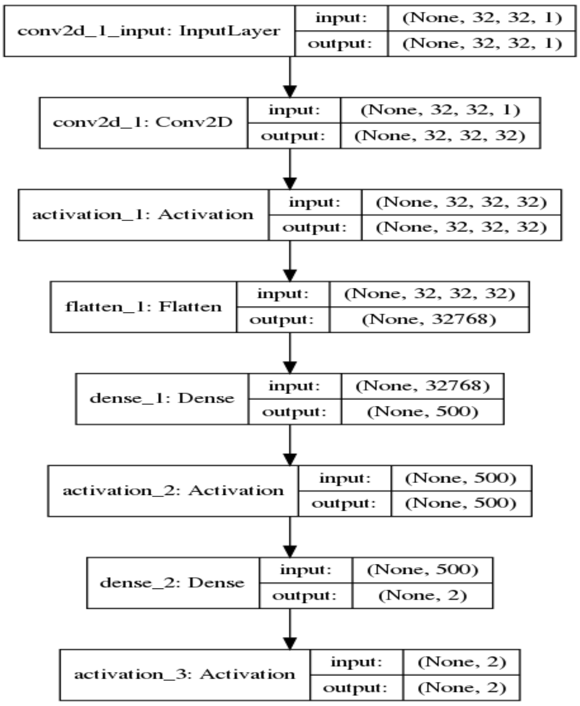

Figure 6. CNN Architecture with 2 layers, first layer (500 nodes), second layer (2 nodes)

The process training in this CNN beginning with data grayscale image from pre-processor (32 x 32 x 1) is convolved with (3 x3) kernels as many as 32 kernels, so we get one data image with shape (width = 32, height = 32, and depth = 32 pixels) followed by RELU activation, after that our image data (32 x 32 x 32) is ready to be fully connected to layer that has a node (in this case 2 nodes) but before this treatment, our data image should be flattened so we get (32 x 32 x 32 = 32768) feature vector with 1 Dimensional data. After we get 1-dimensional data with 1 row and 32768 columns data, our data is ready to connect with 2 nodes layer followed by softmax activation for two target outputs with two label classifications, namely 'membuka' and 'menutup'.

Training data in the second model CNN architecture has the same principle as the first model. However, one layer consisting of 500 nodes is only added before being connected with the second output layer with 2 nodes (see Figure 6).

D. Detections

After successfully training our dataset to CNN, we can get a CNN model with the pre-trained network. This pre-trained network is valuable stuff that will be used to predict our video data running in real-time whether our CNN model can detect an 'opened' eye or 'closed' eye. This is a diagram blocking our CNN model in duty to detect an 'opened' eye or 'closed' eye running in real-time.

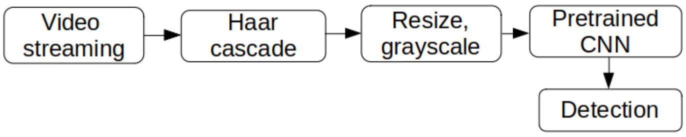

Figure 7. Block Diagram detection running in real-time [8]

Real-time video streaming from a camera containing the face and eye that is tracked by the Haar cascade method, we can get ROI from the face and eye. ROI face and eye image, then entering pre-processor, at this state image resizes to a similar shape and then converts to grayscale to have an image with shape (32 x 32 x 1). This image is similar to the image shape trained on CNN before.

III. RESULTS

This training uses a dataset split into 75 percent training and 25 percent testing datasets, so we get 15000 image data for training and 5000 image data for testing. This treatment has the purpose of observing our performance CNN model.

A. Training the First CNN model

The first architecture CNN model is like in Figure. 5, we use epoch = 100; this training yields a Loss and Accuracy Graph like in Figure 8,

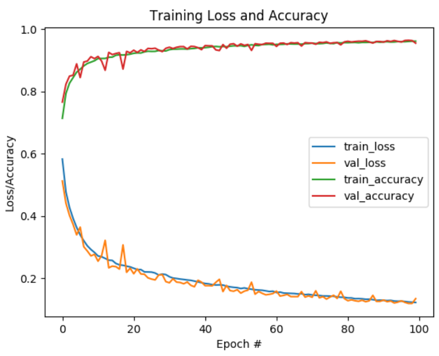

Figure 8. Loss and Accuracy Graph from the first CNN model (20000 datasets, epoch = 100)

From that graph, we get information that accuracy increases rapidly close to 1 along with epoch toward 100. Likewise, Loss decreases to near zero as the epoch increases toward 100.

We must pay special attention to epoch 100 because the pre-trained network is formed when epoch = 100. Training accuracy, validation accuracy, training loss, and validation loss at epoch = 100 can be seen in the following table,

Table 1. Loss and accuracy CNN model at epoch = 100

Epoch	Loss	Accuracy	validation_loss	validation_accuracy
100	0.1229	0.9617	0.1351	0.9552

From that table, we can see that our training model has a validation accuracy of around 0.9. This information means that out of 10 trials, the probability of success for detecting 'opened' and 'closed' eyes is 9 times. Does this prediction match reality when this pre-trained CNN detects the eyes opening and closing of any sample of people in real time?

B. Training second CNN model

The architecture of this CNN model is like in Figure. 6, this architecture is almost the same as the first CNN model but added one layer consisting of 500 nodes, with epoch = 100; training graph that shows the performance of this model (training Loss, training Accuracy, validation Loss, and validation Accuracy) are shown with the following graph,

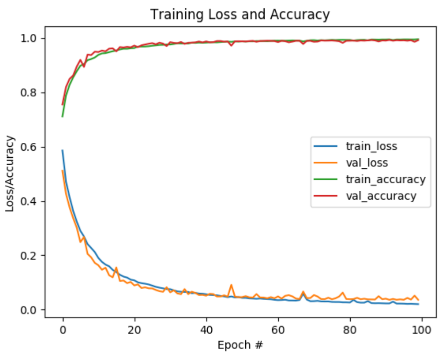

Figure 9. Loss and Accuracy Graph of CNN model with 2 layers (2 nodes and 500 nodes), epoch = 100

Loss and accuracy of training and validation at epoch 100 can be shown in the following table,

Table 2. Loss and validation at epoch 100-second model CNN (2 layers)

Epoch	Loss	Accuracy	Validation_loss	Validation_accuracy
100	0.0202	0.9952	0.0361	0.9916

At epoch 100, validation loss is close to 0, and validation accuracy is close to 1, so we can call this model of CNN has very well performance if this model is tested with that dataset, but is this performance also good if the CNN model is tested to recognize eye opening and closing in real-time?

In this experiment, we used 10 samples from people aged between 3 - 60 years. Each person was tested on CNN as many as 100 tests, whether CNN was able to detect the eyes opening and closing from the sample and produce a table like the following,

Table 3. Result of Detection second pre-trained CNN model to 10 samples running in real-time

Sample names	Successful rate/100
Adisty	0.55
Agus	0.56
Yain	0.50
Febri	0.6
Bayu	0.56
Rian	0.53
Roni	0.57
Ibu	0.55
Bapak	0.55
Wulan	0.57
Mean	0.554

From the table, the success rate of our CNN modeling is around 0.55 in a hundred trials.

IV. DISCUSSION AND CONCLUSION

We use two model CNNs to compare their performances between those two model CNNs. From Table 1 and Table 2, we can get information that the performance of the second CNN model is better than the first CNN model because the second CNN model has more layers than the first layer; scrutinizing validation accuracy, validation accuracy in Table 1 has 0.9552 but validation accuracy at Table 2 has 0.9916, this is clear information that the second model has performance better than the first model in predictions using that dataset (2000 images) but in reality, the performance of our CNN model can only detect with successful rate about a half in 100 trial detections, it is not so poor performance but is not so good. Many factors can affect this CNN giving a poor performance in reality. However, the Haar cascade method performs poorly when detected in real-time. We should try our detection tools that perform better than Haar cascade Method for further experiments.

V. REFERENCES

Wikipedia.(2019) en.wikipedia.org. [Online]. Available:https://en.wikipedia.org/wiki/List of countries by traffic-related death rate.

WHO (2015). Global Status Report on Road Safety 2015. WHO Press.

Donaldson, L.H., Brooke, K., Orthopaedic trauma from road crashes: is enough being done?.pdf. [Online]. Available: http:scopus.com/record.

J. Rolison, S. Regev, S. Moutari, and A. Feeney, What factors contribute to road accidents? An assessment of law enforcement views, ordinary drivers' opinions, and road accident records, Accident Analysis and Prevention., vol. 115, pp. 11-24, February. 2018.

Kemenkominfo. Every hour three people died caused by traffic accidents. [Online]. Available: https://kominfo.go.id//.

Jawa Pos. Indonesian Government targets 5000 km toll roads across Indonesia in 2024. (2019). [Online]. Available: jawapos.com.

CNN. Setiap empat jam perjalanan seseorang dianjurkan untuk beristirahat selama setengah jam.Available:https://www.cnnindonesia.com/otomotif/20230404114029-592-933294/hapus-pikiran- cepat-sampai-pemudik-mobil-wajib-per-4-jam-istirahat.

Firman M Sigit, Blinking Eyes Detection using Convolutional Neural Network on Video Data. International Conference on Information and Communication Technology. Vol. 3 pp. 67, 24 – 25 November 2020, Indonesia.