Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/103/20241079

CNN-Based Distracted Driver Recognition: Enhancing Accuracy Through Optimized Training Techniques

Chuanzhi Ma

School of advanced manufacturing, Guangdong University of Technology, Guangdong, China

1812010707@stu.hrbust.edu.cn

Abstract. Distracted driving is a significant contributor to vehicle accidents worldwide, underscoring the need for an effective recognition model that can identify risky driver behaviors in real-time and provide timely alerts. This study proposes a Convolutional Neural Network (CNN)-based model designed to accurately detect distracted drivers. To optimize the model's performance, this paper implemented several strategic enhancements. During the preprocessing phase, 90% of the dataset was allocated for training, ensuring a comprehensive learning process, while the remaining 10% was used for validation to assess the model's accuracy. The Adam optimizer was chosen for its ability to dynamically adjust the learning rate, facilitating faster convergence to an optimal solution. Additionally, the Cross-Entropy loss function was employed to amplify errors during training, driving the model to correct inaccuracies more effectively. The model was trained over 25 epochs, resulting in an accuracy rate of nearly 80%. This level of performance demonstrates the model's viability for real-world applications, where it can play a critical role in reducing accidents caused by distracted driving. Future research may focus on further refining the model by exploring advanced loss functions, optimizers, and CNN architectures, as well as incorporating more sophisticated data preprocessing techniques.

Keywords: Distracted Driving, Convolutional Neural Network (CNN), Cross-Entropy loss function.

1. Introduction

The history of automobiles is so ancient that can be dated back to 1678, when a artifice of a 3-wheel steam car was invented. Then in early 1800s, an American inventor called Oliver Evans built the first American vehicle. And after 80 years, Karl Benz, a Germany inventor, invented the first 4-wheel car using combustion engine to run, which is thought as the origin of modern car [1]. Although cars are becoming faster and faster with the developments of technology, such as the engines, steering type, etc. There are still almost 94 percent of accidents caused by human mistakes. Therefore, identifying dangerous human behaviors while driving the vehicle becomes an important task nowadays.

The main objective of this study is to recognize human distracting behaviors in their cars by using Convolutional Neural Network (CNN). For the pre-processing part, the activity-map is used to classify the drivers behaves, which benefits for saving the pictures and add the labels. For the model part, the CNN model does play an important role in extracting features of the pictures put into the model, it is meanwhile a vital component for predicting the pictures which aren't labeled. Besides, with the help of

DOI: 10.54254/2755-2721/103/20241079

convolutional layer and the pooling layer, CNN can extract the main traits of the picture with less variables, which reduces the waste of resources and save training time. Additionally, Adaptive-Moment-Estimation (Adam) optimizer is chosen to be the optimizer instead of adaptive-gradient (AdaGrad) or Momentum, because Adam combines the advantages of Ada-grad as well as Momentum. Meanwhile, the model is to be trained for several epochs, so that it can become more accurate. And the Cross-entropy is used to describe the extent of how close the two probability distributions are. In addition, the cosine annealing strategy is adopted to shorten the training time by adjusting the learning steps. Therefore, the experimental results demonstrate that the results are more accurate when the batch size is 64, and the epoch size is 10. Through using CNN to classify what the driver is doing while driving, not only can it improve the application of CNN, but also it can save the recognition cost and reduce the possibility of car incidents which is caused by some dangerous behaviors of the drivers.

2. Literature Review

When it comes to behavior recognition, there have been already a few trials before the appearance of CNN. One of the trials is a methodology based on feature engineering, which need some experienced people to design and train the feature extractors, such as support vector machine (SVM), decision tree etc., which relies on how a programmer experienced. Another method is based on Spatio-Temporal Interest Points [2-4].

The method focuses on extracting some description icons by detecting the Spatio-Temporal Interest Points, then use these signs like Scale-invariant Feature Transform (SIFT), and Speed up Robust Features (SURF) to identify the actions [5,6]. In conclusion, though it can recognize the local actions in the video, it cannot capture the features when most of objects are moving in the video. Besides, some researchers use Dense Trajectories to get the motions [7]. They generate the trajectories by following the pixels in the video, then extract the traits such as Histogram of Oriented Gradient (HOF), Motion Boundary Histograms (MBH), along the trajectories [8].

However, there is a drawback that the accuracy is hugely affected when the perspective changes or the light changes. Then in 2004, researchers find a method to identify the motion called Hidden Markov Model (HMM) [9]. HMM can be used to model the motion pattern, and infer what kind of the motion it is. Nevertheless, its performance is unsatisfied when the action becomes more complicated and the video becomes longer. Other interesting method called Dynamic Time Warping is also used to identify the behaves before CNN comes out, which recognize the motion with comparing the action patterns in different videos [10]. After deep learning comes out, there is no doubt that researchers use it to recognize the motions. For example, some of them use Two-stream CNN to identify the facial expression, others use 3D Convolutional Networks [11,12]. Therefore, it shows that CNN gets famous for its accuracy in behavior identification field.

3. Methodology

3.1. Dataset description and preprocessing

In this research, there are totally 102,150 driving pictures used as dataset which comes from Kaggle, including safe-driving, texting on the right hand, talking on the phone with the right hand, talking on the phone with the left hand, texting on the left hand, drinking, operating the radio, reaching behind, hair and makeup, and talking to the passenger [13]. Besides, the 22,424 of them are used as training set, and the rest of them are used as test set. During the preprocessing, the pictures is separated into 64 groups to train the model and optimize the hyper-parameters. Besides, the number of epochs are set to be 25, aiming to acquire the most accurate model.

3.2. Proposed method

The present study focuses on recognizing the distracted drivers with CNN model, which consists of convolutional layers and fully connected layers. But except for CNN model, the Adam optimizer also is used to adjust the learning steps while training. Besides, the data loader splits the data in 9:1 scale, which

DOI: 10.54254/2755-2721/103/20241079

means that if there are 10 datasets, 9 of them are used as training set, and the rest of them is used as verification set. But it is worth to catch attention that in this study the augmentation of the image is improper because the total accuracy is lower than normal situation if the image augmentation is utilized. So, in conclusion, firstly the training dataset (images) is put into the model, then the CNN adjusts its parameters with loss function and Adam optimizer, finally, some pictures are put into the model to check the accuracy of the model prediction. And the Figure 1 as below is the general structure of the model.

The study introduces a CNN-based model for distracted driver recognition, emphasizing its innovation over traditional models like Fully Connected Neural Networks (FNNs). The CNN model stands out by extracting key features from images, leading to higher efficiency during training compared to methods that process all image details. Additionally, the integration of the Adam optimizer accelerates convergence, further enhancing the model's performance and training speed. This combination of CNN's feature extraction capabilities and Adam's optimization highlights the model's superior efficiency and effectiveness in comparison to other approaches.

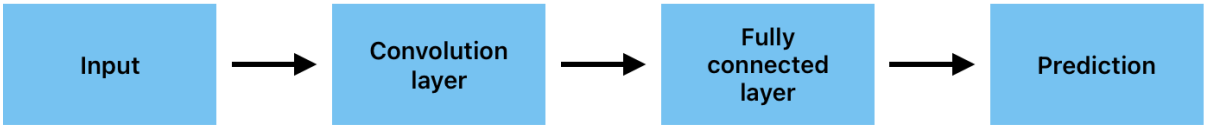


Figure 1. The pipeline of the model.

3.2.1. CNN. CNN is one of the most popular models in deep learning. Besides, there are totally compositions in CNN (see in Figure 2), they are respectively feature extraction, classification, and probability distribution. It is obvious that the feature extraction is used to extract the features of the picture. And it is composed with input layer, pooling layer and flatten layer and some kernels. The input layer is used to reserve the original picture. And the pooling layer is made up of several layers produced by different kernels in the last layers. Additionally, with being scan by the kernels, the pooling layer becomes smaller and smaller, and finally the features are saved in the flatten layer which is used as the input of the classification which is actually consisted with a fully connected layer. Next is about the classification part. It is composed with a whole fully connected layer, whose input is the flatten layer and the output is probability distribution. Besides, the fully connected layer is composed with a intact neuron network, and with the help of the neuron network, the features of the picture turns into a few discrete values which is the input the probability distribution. And the probability distribution multiplies each value with a different random weight, and produces a unique value to recognize what kind is the picture.

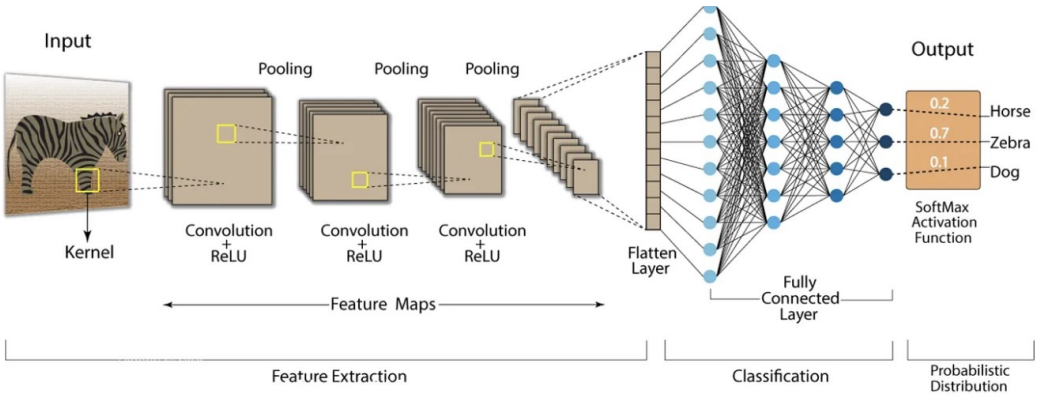


Figure 2. CNN architecture [14].

After learning about how the CNN works, the size of the pooling layers in feature extraction and the size of nodes in each layer of the fully connected layer can be reduced hugely with the CNN, therefore,

DOI: 10.54254/2755-2721/103/20241079

one of the most obvious advantages of the CNN model is that it can compress the data size, which make the training more available for the computer. Besides, it can also make the classification easier with the help of the decrease of the parameters. Therefore, the CNN model is the best candidate for distracted driver recognition. In this study, the data loader is applied to put the driver images into the feature extraction part of the CNN model, and train it for 25 times to make the model as accurate as possible.

3.2.2. Adam optimizer. Before diving into Adam optimizer, some basic information of AdaGrad optimizer and the Momentum optimizer must be mentioned. First, it's about the AdaGrad. AdaGrad is an adaptive gradient optimizer, it gives different parameters different learning steps. If the update frequency of the parameter is too few, the AdaGrad optimizer gives a higher learning step to the parameter, otherwise, the AdaGrad gives a lower learning step to the parameter if its update frequency is too high. Next, it is about Momentum. The Momentum optimizer is used to accelerate the convergence of the model by introducing an exponential weighted average of a cumulative gradient so that it can take the past gradient information into consideration of updating the parameters. And the Adam optimizer is a combination of the AdaGrad and the Momentum. It can calculate the adaptive learning step independently for each parameter and it doesn't need to adjust the size of learning steps. What's more, the Adam optimizer can accelerate the convergence quicker and reduce the training time. So, in this study, the Adam optimizer is chosen to update the weights of parameters in the neuron network of fully connected layer.

3.2.3. Cross-Entropy loss function. Choosing a right loss function also plays an important role in the training of the CNN model. In order to make the model more accurate, the cross-entropy function is chosen to be the loss function.

$$L(y, y) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(y_i) + (1 - y_i)\log(1 - y_i)] \tag{1}$$

where y means the real value set of the whole dataset, and the other input of the function means the prediction value set of intact datasets, N is the number of the dataset, y_i means the real value of the NO. i sample, and y_i means the prediction value of the NO. i sample. With the help of this loss function, the value of the hyper-parameters of the CNN model can be revised more quickly and identify what the driver is doing on his car.

3.3. Implementation details

The research based on python3 uses "sklearn" library to split the dataset, and uses "torch" library for realizing the cross-entropy loss function. Besides, the study implements the Adam optimizer with the help of CFG class. Additionally, the size of the batches is recommended to 32 for the training set, 64 for validation set and test set respectively, and the size of epochs is recommended to 25. What is worth to pay attention to is that the whole program can be run on the Google code-lab. What's more, the picture augmentation is not recommended because the model accuracy reduces while using the augmentation.

4. Result and Discussion

In order to show the process of training CNN model, some results such as loss value, accuracy, and some figures are presented as the followings, which is convenient to indicate the efficiency of the CNN and its accuracy.

4.1. Analysis of train loss

From the table 1 the author can find out that the train loss is decreasing from 0.0597 to 0.0511, with the advance of the epochs. In other words, according to the decreasing of the train loss, it is obvious that after training for 25 times, the CNN model can output the correct answer. Besides, it indicates that the

DOI: 10.54254/2755-2721/103/20241079

CNN model spends less time on training and adjusting the parameters, benefiting from its neuron network structure. Besides, according to the table 1, the train accuracy increases from 0.5704 to 0.8239, which indicates that with the advance of 25 epochs based on the CNN model, the model accuracy can be improved obviously.

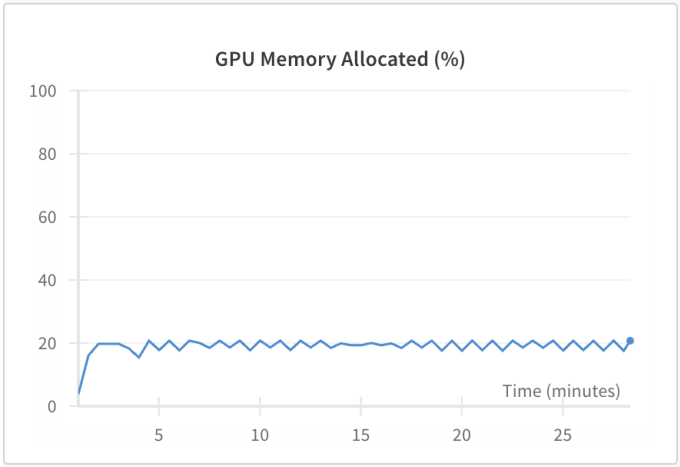
Table 1. Train loss value caused by different epoch.

Epoch	Train Loss	Train Accuracy
1	0.0597	0.5704
2	0.0544	0.7205
3	0.0534	0.7502
4	0.0530	0.7624
5	0.0528	0.7659
6	0.0527	0.7711
7	0.0525	0.7719
8	0.0523	0.7810
9	0.0523	0.7844
10	0.0522	0.7904
11	0.0520	0.7964
12	0.0520	0.7933
13	0.0519	0.7987
14	0.0518	0.8001
15	0.0517	0.8031
16	0.0515	0.8134
17	0.0516	0.8059
18	0.0515	0.8113
19	0.0514	0.8151
20	0.0513	0.8154
21	0.0513	0.8209
22	0.0512	0.8188
23	0.0511	0.8238
24	0.0511	0.8234
25	0.0511	0.8239

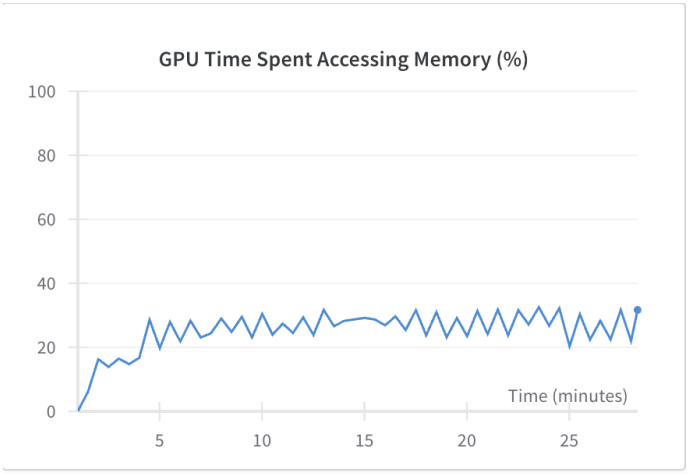
4.2. Analysis of GPU usage

According to Figure 3, the percentage of the Graphics Processing Unit (GPU) memory usage is only about 20%, the time spend of GPU accessing the memory also less than 40%. In other words, because the kernel sliding on the convolutional layers reduces the number of parameters which means that the kernel can extract the main characteristics and throws some useless features away, the input of the fully connected layer can be minimized as much as possible, so the usage of GPU memory while running CNN is less than regular neuron network. In addition, what is different from the normal neuron network is that each neuron only connects to the neurons on the last convolutional layer, however, in a normal neuron network, every neuron connects to all of the neurons on the last layers, which makes the computer have to access the memory to calculate all of the neurons in the last layer, and it takes too much time versus to the CNN model. Therefore, the CNN model is more efficient than traditional neuron network in terms of accessing the memory.

DOI: 10.54254/2755-2721/103/20241079



(a) memory usage while running CNN



(b) the percentage of GPU time spent accessing memory

Figure 3. The GPU usage while running CNN.

4.3. Evaluation

From table 1, it can be seen that although the original training accuracy is only 57%, this phenomenon is caused by imperfect parameter values. Then in the second period, the training accuracy soared to 72%, indicating that the proposed CNN model can quickly study the main photographic features. Therefore, the proposed solution in this article can save some training time and costs for automotive companies. Due to the fact that CNN models can spend less time accessing memory while on the move and save GPU usage, computers can handle more processes simultaneously and improve computational stability to serve long-term tasks. In addition, after training 25 epochs within 27 minutes and 49 seconds, the optimal model accuracy was close to 0.9960. Thanks to the architecture of CNN and Adam optimizer, this model can automatically study image features, so users do not need to design their own feature extractors. Therefore, the proposed model has excellent generalization ability after training on multiple databases, and can therefore solve input data from different angles and dimensions.

5. Conclusion

This research presents a novel application of a CNN model for recognizing distracted drivers. To enhance the model's performance, the Adam optimizer was selected due to its ability to dynamically adjust the learning rate of the model's parameters, thereby accelerating convergence to a global optimum. The Cross-Entropy loss function was employed to quantify the model's errors, effectively amplifying

DOI: 10.54254/2755-2721/103/20241079

training discrepancies and compelling the model to correct its mistakes more efficiently. The dataset was divided using a 9:1 ratio, with 90% allocated for training and 10% reserved for testing. This partitioning ensured a robust evaluation of the model's performance. After training for 25 epochs, the model achieved an accuracy exceeding 82%, demonstrating its potential for practical application in real-world scenarios. Looking forward, there is considerable scope for further improvement. Future work could involve developing more advanced loss functions and optimizers, refining the CNN architecture, and exploring enhanced data preprocessing techniques. These advancements could lead to even greater accuracy and reliability in distracted driver recognition systems.

References

[1] Lucendo J 2019 Cars of Legend: First Cars of History Jorge Lucendo
[2] Huang S J et al. 2018 Applications of support vector machine learning in cancer genomics. Cancer genomics & proteomics vol 15 no 1 pp 41-51
[3] Su J and Harry Z 2006 A fast decision tree learning algorithm Aaai. vol 6
[4] Willems G Tinne T and Luc V G 2008 An efficient dense and scale-invariant spatio-temporal interest point detector Computer Vision–ECCV European Conference on Computer Vision
[5] Wu J et al. 2013 A Comparative Study of SIFT and its Variants Measurement science review vol 3 no 3 pp 122-131
[6] Oyallon E and Julien R 2015 An analysis of the SURF method Image Processing On Line 5 pp176-218
[7] Gu J Chen S and Hang Z 2021 Densetnt: End-to-end trajectory prediction from dense goal sets Proceedings of the IEEE/CVF International Conference on Computer Vision
[8] Uijlings J R R, et al. 2014 Realtime video classification using dense hof/hog Proceedings of international conference on multimedia retrieval
[9] Blunsom P 2004 Hidden markov models Lecture notes vol 15 no 18-19 p 48
[10] Senin P 2008 Dynamic time warping algorithm review Information and Computer Science Department University of Hawaii at Manoa Honolulu vol 855 no1-23 p 40
[11] Peng X J and Cordelia S 2016 Multi-region two-stream R-CNN for action detectio Computer Vision–ECCV
[12] Yu C Y et al. 2020 A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing vol 13 pp 2485-2501
[13] ANNA M 2023 State Farm Distracted Driver Detection Retrieved on 2024 Retrieved from: https://www.kaggle.com/competitions/state-farm-distracted-driver-detection/data
[14] Shahriar N 2023 What is convolutional neural network–CNN (Deep Learning) E. Retrieved on 2024 Retrieved from: https://nafizshahriar. medium. com/what-is-convolutional-neural-network-cnndeep-learning-b3921bdd82d5