25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-1039
CONVOLUTIONAL NEURAL NETWORK APPROACH APPLIED TO DISTRACTED DRIVER CLASSIFICATION

Luiz Eduardo Soares Emidio da Silva
Computer Engineering Undergraduate Program, Pontifical Catholic University of Parana (PUCPR), Curitiba, PR, Brazil
LuizE.PUC@outlook.com

Rodrigo Negri de Azeredo
Electrical Engineering Undergraduate Program, Federal University of Parana (UFPR), Curitiba, PR, Brazil
rodrigonegria@gmail.com

Luis Gustavo Tomal Ribas
Electrical Engineering Graduate Program (PPGEE), Federal University of Parana (UFPR), Curitiba, PR, Brazil
tomalribas@hotmail.com

Roberto Zanetti Freire
Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR), PR, Brazil
roberto.freire@pucpr.br

Viviana Cocco Mariani
Department of Mechanical Engineering (PPGEM), Pontifical Catholic University of Parana (PUCPR), Curitiba, PR, Brazil
viviana.mariani@pucpr.br

Leandro dos Santos Coelho
Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR), and
Electrical Engineering Graduate Program (PPGEE), Federal University of Parana (UFPR), Curitiba, PR, Brazil
leandro.coelho@pucpr.br

Abstract. *The number of passenger vehicle accidents has increased over the last decade, making thousands of victims worldwide. Among the different causes of accidents, it is possible to highlight problems originated by situations of distracted drivers. Therefore, a survey was carried out by the State Farm company for the production of images by on-board cameras in vehicles, in a controlled environment, with the objective of identifying risk situations. The situations could be recognized through pose estimation, evaluating hands movements and face position for the driver. The application of Convolutional Neural Network (CNN) is an efficient method in this type of classification task, since they already presented interesting results in the classification of images, showing as an efficient classifier option in terms of accuracy for this case. The obtained results showed promising performance in terms of classification accuracy of the proposed CNN setup to the Kaggle's State Farm distracted driver detection contest. The best results were obtained through a CNN with stochastic gradient descent (SGD) optimizer and ReLU (rectified linear unit) and an ELU (exponential linear unit) activation functions, with an accuracy of 98.91% and 98.89%, respectively.*

Keywords: *Distracted driver, machine learning, classification, convolutional neural network, pose estimation.*

1. INTRODUCTION

The National Highway Traffic Safety Administration (NHTSA) released a report about distracted driving which informs that 3,477 people were killed and 391,000 were injured in motor vehicle crashes involving distracted drivers in 2015 (NHTSA, 2017). According to NHTSA, "distraction is a specific type of inattention that occurs when drivers divert their attention from the driving task to focus on some other activity instead", that are classified into manual, visual and/or cognitive distraction. Manual distractions are defined as taking the hands off the driving wheel; visual occurs when the eyes take off the road; and cognitive when the driver takes one's mind off driving (CDC, 2017).

Therefore, a distraction can be anything that takes attention from the driver. Center for Disease Control and Prevention (CDC) cites several examples, such as: sending a text message, using a navigation system and talking on a cell phone. These distractions could endanger the driver and other people in transit. Texting while driving is especially dangerous because it combines all three types of distraction cited, mainly mental distraction. Sending or reading a text message takes the eyes off the road for about 5 seconds, enough period to cover a football field while driving at 55 mph (about 90 km/h) (CDC, 2017).

For this reason, a research-focused in this theme was developed. The State Farm, a group of insurance companies and financial services, developed a project in a controlled environment to produce images from distracted driver situations. The idea behind it consists of being able to use deep learning techniques to know whether or not a car dashboard camera is able to detect a driver in some distracting behavior. The distracted driver problem was proposed as competition in the Kaggle Challenge in 2016.

In this task, State Farm made available a large collection of RGB images (640x480 pixels) as a dataset split into in two parts – one with more than twenty-two thousand training samples and other with almost eighty thousand test samples. The driver in each picture (see details in Fig. 1) represent one of ten possible classes, such as: c_0: safe driving; c_1: texting – right; c_2: talking on the phone – right; c_3: texting – left; c_4: talking on the phone – left; c_5: operating the radio; c_6: drinking; c_7: reaching behind; c_8: hair and makeup; and c_9: talking to passenger (Kaggle, 2016). There are 26 different drivers inside the training phase of the classifier.

Wherefore, we are focused on detecting manual distractions, once the images released for State Farm contest allow to identify what kind of activity the driver is doing and recognize the cause, which is classified in 10 different instances. In this context, Convolutional Neural Network (CNN) is a possible classifier approach of the deep learning field that provides a framework to solve this computer vision problem. By iteratively feeding images into a CNN, it is able to learn features of different hierarchical natures.

In recent years, CNN models have been increasingly used in several computer vision tasks, like image classification (Seo and Shin, 2019), medical applications supporting diagnosis by signals or images (Diamantis et al., 2019), action recognition (Yao et al., 2019), head pose estimation (Xu et al., 2019) and brain-computer interface (Zhang et al., 2019).

The contribution of this paper is related to the design and validation of a CNN to the multi-class problem proposed by Kaggle’s State Farm distracted driver detection contest. The distractions included in the mentioned context are composed mainly for hand position modifications and face orientation. Thus, they could be estimated by posture recognition.

The remainder of this paper is organized into five sections. Section 2 presents a description of the distracted driver detection case study. After, Sections 3 and 4 present the fundamentals of CNNs and results analysis, respectively. Finally, Section 5 is focused on the conclusion and future research.

2. DESCRIPTION OF THE CASE STUDY

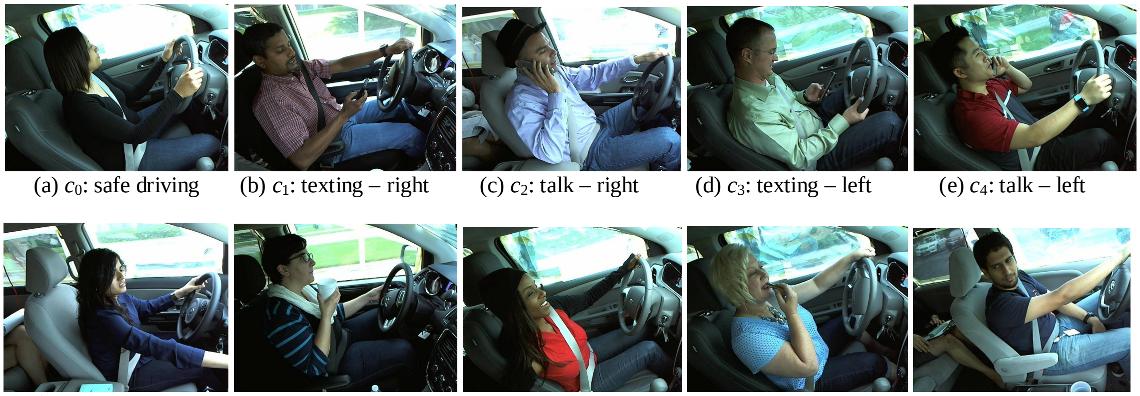
The database provided for the competition presents images of drivers as shown in Fig.1, where it is possible to detect ten varying postures. It was the first publicly dataset with a goal of posture classification and for this reason, the idea was to use a CNN to classify those images, distinguishing the different situations presented in the problem. CNNs are presenting incredible progress during the last years in image classification, object detection and natural language processing. The system is generated including convolutional filters/layers, activation functions, pooling layer and Fully Connected (FC) layer (Baheti et al*.*, 2018).

One solution related to the distracted driver issue is to identify the distraction type, and possibly to alert the individual to potential hazards, so that a safe driving condition is restored reducing accident probability. That concern is in the fact that one in every four fleet crashes occurs because of distracted driving (State Farm, 2016). The survey also showed that other services with cell phones (texting and social media) and activities are performed frequently while driving, which increases the interest in combating distracted driving.

Following the definitions established by the NHTSA (NHTSA, 2017), and in the CDC (CDC, 2017), distractions are classified in manual, visual, or cognitive distractions. Cognitive distractions are related to the mental distraction of the driver, making it impossible to detect them through the images of a camera installed on the vehicle's dashboard. Visual distractions depend on the detection and facial tracking, being possible to identify through images. In the case of manual distractions, which are related to the positioning of the driver's limbs, tracing is performed through the individual's posture.

The applied approach is composed by a CNN to detect manual distractions, analyzing the behavior of the driver and classifying their state based on posture and performed gestures. As technique lacking was used a dropout, a regularization technique that aid to reduce overfitting. It was also applied a batch normalization, which normalizes the output of previous activation layer and helps to accelerate the learning and increase the accuracy.

The competition ended in August 2016. After that, a collection of solutions was made available at the Kaggle’s forum, being the majority of them written in Python programming language version 3. These solutions could be used as benchmarks and, in our case, the structure itself was composed of sixteen convolutional layers, each one considering batch normalization, and a final output dense layer, as shown in Fig. 2.



(a) c_0: safe driving (b) c_1: texting – right (c) c_2: talk – right (d) c_3: texting – left (e) c_4: talk – left

(f) c_5: operate radio (g) c_6: drinking (h) c_7: reaching behind (i) c_8: hair and makeup (j) c_9: talk passenger

Figure 1. Possible classes from State Farm database to distracted driving situations (KAGGLE, 2016).

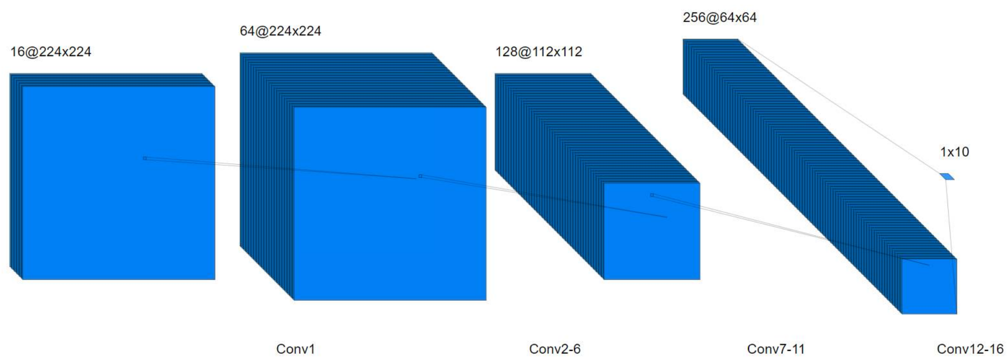


Figure 2. CNN model

To facilitate training and testing Google Colaboratory (Colab) was used as a platform to develop this research due to the large dataset and the CNN format. Colab is a Google's free cloud service for AI (Artificial Intelligence) developers and with it is possible to carry out deep learning applications on the Graphics Processing Unit (GPU). The GPU model available is a Tesla K80, allowing the use of 12.72 GB RAM (Random Access Memory) maximum and 358.27 GB disk memory. Google Colab was originally part of the Jupiter Project, and it is an environment that allows running Python scripts in the cloud and stored it in Google Drive.

3. FUNDAMENTALS OF CONVOLUTIONAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) are part of a bigger ground of machine learning techniques known as inductors. A Neural Network learns by judging an input, verifying, and then adjusting weights based on real answers. This procedure is named supervised learning. A simple ANN is composed by a series of neurons separated in layers, these layers can be classified within 3 main groups: the input layer, the hidden layers, and the output layer. The previously mentioned structure is presented in Fig. 3.

The input layer is where data enter in the Network. In the case of an MLP, first data must be separated according to its characteristics and each neuron receives one of them. The Hidden layer represents where learning occurs and based on previews experiences of the Network data are classified. Finally, the conclusion of the classification can be found in the output layer, where each neuron corresponds to one of the possible answers and the perceptron with higher values representing the network final response.

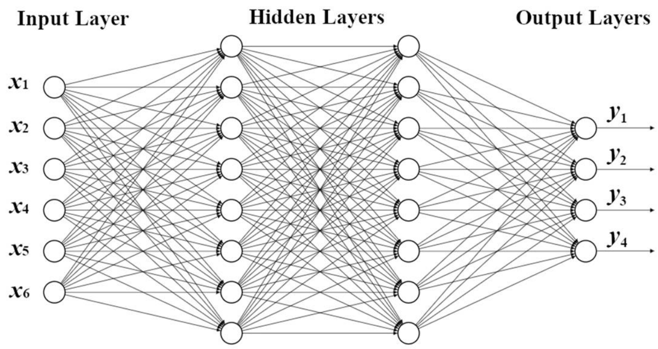


Figure 3. Multi-layer perceptron model.

A neuron represents the atomic structure of a Neural Network, it works as both information transformer and transmitter and, outside the input layer, every single neuron is fully connected by a link called synapses. However, not all neurons are connected equally, the synapses among neurons can have strong or weak relationships, changing the signal received. For that reason, synapses can also be called weights, additionally, a neuron also contains a bias term (b), which contributes as a signal (see details in Fig. 4).

In order to create an output, the neuron sums its bias and all the outputs from the previous layer, multiplying these values by their respective weights. Finally, they pass the answer into an activation function, such as Relu (Eq.7), that will transform the sum into the output of the neuron.

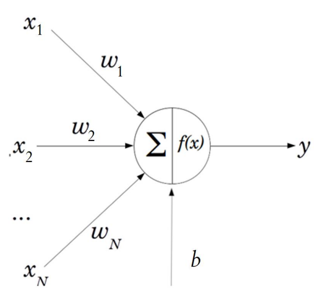


Figure 4. Neuron model.

CNNs have been applied to image recognition and classification, such as a tool to solve problems in analyzing medical images (Litjens et al., 2017; Tustison et al. 2019), video recognition (Yao et al., 2019), lip reading (Mesbah et al., 2019), sentimental analysis (Abid et al., 2019), image and video semantic segmentation (Garcia-Garcia et al., 2018) and breast cancer classification (Ting et al., 2019). This specific type of network contains several layers that could be classified into three different types: convolutional, max-pooling and fully connected.

Convolutional layers comprehend a grid of neurons that works as a learnable filter. These filters are moved throughout the input image performing a dot product considering the weights of the filter and a piece of the image, as showed in Eq. (1), where x and y represent the pixel position, n_1 and n_2 are iterative variables (see Fig. 5). In this project, we resized the input image to 224×224 and applied 16 filters for the first layer, 64 filters for four hidden layers, and 128 filters for the next five hidden layers, and 256 filters for the last five layers. All these filters have a 3×3 dimension.

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2] \quad (1)$$

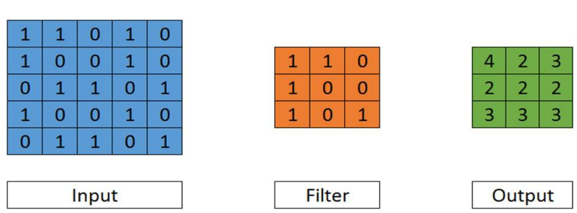


Figure 5. Convolutional Grids example

After each convolutional layer, a batch normalization was considered. Batch normalization is one of many techniques available as a tool to improve the CNN performance; it is applied and operates on top of the output from the previous layer. Similar to normalization, its objective is to transform the data to the same scale with a mean activation close to zero and a standard deviation around one. This operation occurs at a mini-batch base.

The process happens by calculating the mini-batch mean (μ_{B}) in Eq. (2), and its standard deviation(σ_B^2) Eq. (3), normalizing and adding arbitrary parameters as presented in Eq. (5), where p represents a specific example, m is the size of mini-batch, i is the index of the neuron and γ and β are arbitrary parameters. By using the batch normalization technique, it is possible to increase the learning speed of a CNN helping to prevent the exploding gradient descent problem.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} p_i \tag{2}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(p_i - \mu_B)^2 \tag{3}$$

$$P_i = \frac{p_i - \mu_B}{\sqrt{\sigma_B^2}} \tag{4}$$

$$q_i = \gamma P_i + \beta \tag{5}$$

The pooling layer generates small rectangular blocks originated from the convolutional layer. The main idea is to reduce the size of the input image for faster execution and to prevent overfitting. There are many possibilities to perform this pooling, such as considering the maximum grid value or the average value, or a learned linear combination of the neurons. We used a pooling layer that acquires the average value of the block to solve these problems, as represented in Fig. 6.

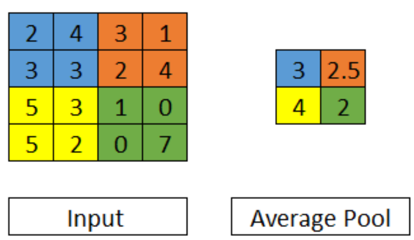


Figure 6. Average Pooling representation

Finally, after transmitting information through all these layers, CNN learning can be concluded via fully connected layers. This structure is used to predict each class of the problem and a flatten layer was applied to convert a two-dimensional matrix in a one-dimensional vector. A SoftMax activation function was considered to give a probabilistic answer for the output categories and it is represented in Eq. (6), where i is the index of the neuron and p its value for $i = 0, 1, \ldots, k$.

$$h(p) = \frac{e^{p_i}}{\sum_{j=0}^{k} e^{p_j}} \tag{6}$$

The complete CNN architecture used can be visualized in Fig. 7.

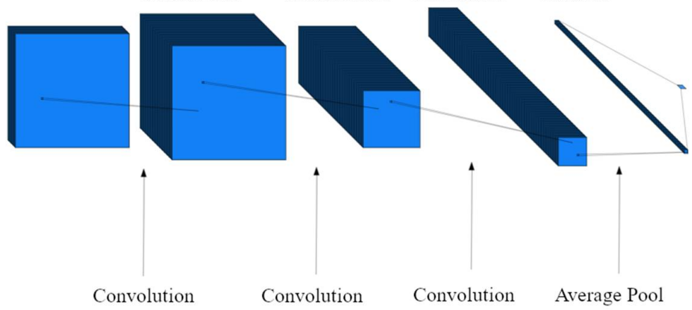


Figure 7. Complete CNN architecture

The performance measures are accomplished through multi-class logarithmic loss (logloss), also known as cross-entropy loss. This function defines the classification model, where the prediction input is a probability value between 0 and 1. The idea is that the algorithm minimizes this value, where a perfect model would have a log loss result equal to zero. Equation (7) shows the $logloss$ function, where M represents the number of classes (in the State Farm problem there are 10 classes); $y_{0,c}$ is an integer 0 or 1 that consists in the real label and the value that the neuron should achieve; and $P_{0,c}$ is the value attributed by the SoftMax activation (Eq. (6)) to a neuron in the output layer.

$$logloss = -\sum_{c=1}^{M} y_{0,c} \log(P_{0,c}) \quad (7)$$

The CNN training procedures considers the same algorithms applied in other neural networks, being among these the back-propagation algorithm, that is a popular training method based on gradient descent (Gu et al., 2018). In a supervised learning problem, the weights are selected so that the output loss function is minimized, also called the network error (see Eq. (6)). The weights are updated by some negative scalar reduction of the derivative of the error in relation to this weight. In the case of the gradient descent, the criterion for algorithm optimization is the convergence speed considering adequate stability. Therefore, the objective of the back-propagation is to optimize the weights, allowing the neural networks to learn how to correctly map the inputs to the outputs.

There are numerous variants of CNN architectures in the literature. However, their basic components are very similar. The main aspects of a CNN design were linked in order to define: convolutional layer, pooling layer, activation function, loss function, regularization, and optimization.

Some variations of the network were tested using different activation functions, for instance, the Rectified Linear Unit (ReLU) that is represented in Eq. (8) and the Exponential Linear Unit (ELU) described in Eq. (9), where r represents the input and a an arbitrary parameter. The representation of these functions is shown in Fig. 8.

$$f(r) = \max(0, r) \quad (8)$$

$$f(r) = \begin{cases} r & , \text{ if } r > 0 \\ a(e^r - 1) & , \text{ otherwise} \end{cases} \quad (9)$$

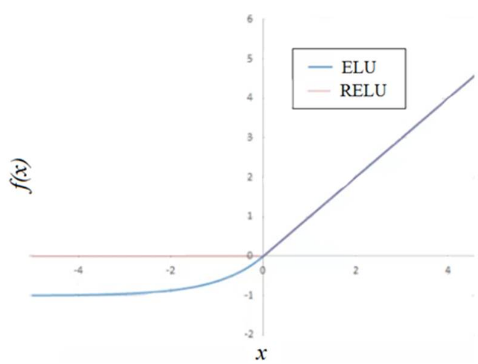


Figure 8. ReLU and ELU activation functions.

4. RESULTS AND DISCUSSION

An optimization procedure is necessary during the learning process to define how exactly the Backpropagation algorithm will affect the weights and biases of the Neural Network. The algorithm consists of finding the gradient vector, which represents the direction of the greater ascension of the error, and then inverting it. With that in mind, it is possible to define how each weight and bias should be modified to reduce the error improving the CNN accuracy.

However, updating the Neural Network at each training sample would demand considerable computational power and time, for that reason, different techniques to soften the computational effort still improving the CNN performance were considered.

The Stochastic Gradient Descent (SGD), Adam (Adaptive Moment Estimation), Nadam (Incorporating Nesterov Momentum into Adam) and Root Mean Square prop (RMSprop) are optimizers required as an argument for compiling a model in Keras Application Programming Interface (API). They were used for optimization purpose during the learning process and their results are represented in Table 1. All experiments were run with a rate split of 33% for the test base considering 5 epochs as a stopping criterion.

The parameters of each optimization method used with the CNN model were described in the sequence. For the SGD a *learning rate* of 0.001, the *decay* of 10^{-6}, *momentum* (a parameter that accelerates SGD in the relevant direction and dampens oscillations) equal 0.9, and *Nesterov Momentum* (momentum update). For the Adam algorithm, it was assumed a *learning rate* of 0.001 and a *decay* of 10^{-6}. Additionally, for Nadam method, it was used a *learning rate* of 0.002, *beta1* equal 0.9 and *beta2* equal to 0.999, none *epsilon* (fuzz factor) and a *schedule decay* of 0.004. Finally, form RMSprop algorithm, a *learning rate* of 0.001, *decay* of 10^{-6}, *rho* equal to 0.9 and none *epsilon*.

Table 1. Results of the CNN models test to the case study.

CNN optimizer	Activation Functions	Accuracy (%)
SGD	ELU	**98.91**
Adam	ELU	94.80
Nadam	ELU	93.39
RMSprop	ELU	97.72
SGD	ReLU	**98.89**
Adam	ReLU	96.43
Nadam	ReLU	94.57
RMSprop	ReLU	92.05

Figure 9 shows the convergence of the model that presented the best result per epochs that is a CNN constructed with SGD optimizer and an ELU activation function. The accuracy of the problem increased with the number of epochs, both for training and testing bases. The loss function showed the opposite behavior, indicating good performance of the model and the absence of overfitting. On the other hand, it showed a disadvantage of the CNNs in long training times.

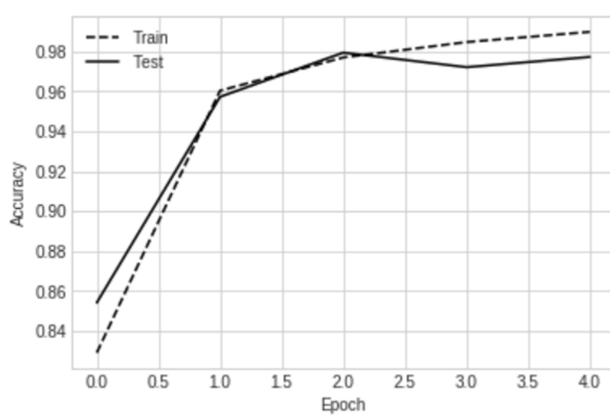
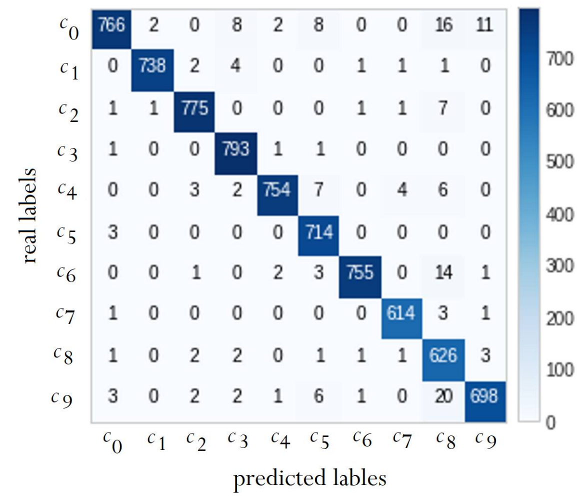


Figure 9. Train and test convergence curves from CNN with SGD optimizer and ELU activation functions.

A confusion matrix is represented in Table 2. It is possible to notice that the most confusing posture for the CNNs is the “hair and makeup”. In static images, it is difficult to differentiate among some positions due to the lack of temporal information. Consequently, in some cases, it is hard to define both the correct motion and state. According to the results presented in Table 2, some distractions could be confused with other activity often by the proximity of the movements and positions of the limbs.

Table 2. Confusion matrix for basic CNN model



5. CONCLUSION AND FUTURE RESEARCH

Distracted driving problem is the reason for many accidents worldwide, causing a large number of victims every year. Therefore, to previously detect distractions is an important attempt to reduce risk situation during driving. As showed in this paper, an efficient way to identify the driver's state can be performed by analyzing their behavior in order to classify their state based on posture and gestures. In this case, in order to recognize manual distractions, a possible approach is to use Convolutional Neural-Networks to classify images of driver state.

A dataset based on a survey accomplished for an insurance company called State Farm with more than a hundred thousand images of twenty-six different drivers was assumed in this study. The dataset was split in ten different states during the direction activity. The CNN applied in this research consisted of sixteen convolutional layers with two different activation functions and optimizers. The better results achieved almost 99% for the classification accuracy. Based on these results, it is possible to conclude that the use of CNNs is a viable option to detect distraction while driving.

However, in some cases, it is difficult to identify the correct state considering static images, as it is not possible to know what the temporal response is and the action of the driver. This situation can generate false "safety drive" classifications. To improve the classification method, more features can be acquired, such as skin detection (color and texture identification) and hand points.

In future work, it could be incorporated the temporal context to reduce misclassification errors. Additionally, the developing of a system that can detect visual distractions (mainly eyes information) as well as with manual distractions (positioning of the driver's limbs) in order to identify new distracted conditions in addition to the cases already presented by the database. Furthermore, an ensemble of CNNs can increase the classification accuracy through the combination of several architectures, such as AlexNet, VGG-16, VGG-19, InceptionV3 and others, such as pose estimation (Kawana et al., 2018) and bioimage classification (Nanni et al., 2019).

6. ACKNOWLEDGEMENTS

The authors would like to thank National Council of Scientific and Technologic Development of Brazil - CNPq (Grants: 303908/2015-7-PQ, 304783/2017-0, and 404659/2016-0-Univ) and Fundação Araucária (PRONEX-FA/CNPq 042/2018) for its financial support of this work.

7. REFERENCES

Abid, F., Alam, M., Yasir, M. and Li, C., 2019. "Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter". *Future Generation Computer Systems*, Vol. 95, pp. 292-308.

Baheti, B.; Gajre, S. and Talbar, S., 2018a. "Detection of distracted driver using convolutional neural network". *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018. pp. 1032-1038. Salt Lake City, USA.

CDC, 2017. “Distracted Driving”. Center for Disease Control and Prevention. 25 Feb. 2019 <https: //www.cdc.gov/motorvehiclesafety/distracted_driving/>.

Diamantis, D. E., Iakovidis, D. K. and Koulaouzidis, A., 2019. “Look-behind fully convolutional neural network for computer-aided endoscopy”. *Biomedical Signal Processing and Control*, Vol. 49, p. 192-201.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P. and Garcia-Rodriguez, J., 2018. “A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, Vol. 70, pp. 41-65.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, L., Wang, G., Cai, J., and Chen, T., 2018. “Recent advances in convolutional neural networks”. *Pattern Recognition*, Vol. 77, pp. 354-377.

Kaggle, 2016. “State farm distracted driver detection”. Kaggle. 20 Feb. 2019 <https://www.kaggle.com/c/state-farm-distracted-driver-detection>.

Kawana, Y., Ukita, N., Huang, J. B. and Yang, M. H., 2018. “Ensemble convolutional neural networks for pose estimation”. *Computer Vision and Image Understanding*, Vol. 169, pp. 62-74.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A. and Sánchez, C. I., 2019. “A survey on deep learning in medical image analysis”. *Medical Image Analysis*, Vol. 42, pp. 60-88.

Mesbah, A., Berrahou, A., Hammouchi, H., Berbia, H. and Qjidaa, H., and Daoudi, M., 2019. “Lip reading with Hahn Convolutional Neural Networks”. *Image and Vision Computing*, Vol. 88, pp. 76-83.

Nanni, L., Ghidoni, S. and Brahnam, S., 2019. “Ensemble of convolutional neural networks for bioimage classification”. *Applied Computing and Informatics* (in press).

NHTSA, 2017. National Center for Statistics and Analysis. “Distracted Driving: 2015, in Traffic Safety Research Notes”. DOT HS 812 381. National Highway Traffic Safety Administration: Washington, D.C. 20 Feb. 2019 <https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/812_381_distracteddriving2015.pdf>.

NHTSA, 2017. National Center for Statistics and Analysis. Distracted Driving. Traffic Safety Facts. 16 May 2019 <https://www.nhtsa.gov/risky-riving/distracted-driving>.

Seo, Y. and Shin, K., 2019. “Hierarchical convolutional neural networks for fashion image classification”. *Expert Systems with Applications*, Vol. 116, pp. 328-339.

State Farm, 2016. Distracted Driving. 04 May 2019 <https://www.statefarm.com/simple-insights/auto-and-vehicles/distracted-driving-includes-growing-mobile-web-use>.

Ting, F. F., Tan, Y. J. and Sim, K. S., 2019, “Convolutional neural network improvement for breast cancer classification”. *Expert Systems with Application*, Vol. 120, pp. 103-115.

Tustison, N. J., Avants, B. B. and Gee, J. C., 2019. “Learning image-based spatial transformations via convolutional neural networks: a review”. *Magnetic Resonance Imaging* (in press).

Xu, L.; Chen, J. and Gan, Y., 2019. “Head pose estimation with soft labels using regularized convolutional neural network”. *Neurocomputing*, Vol. 337, pp. 339-353.

Yao, G., Lei, T. and Zhong, J., 2019. “A review of convolutional-neural-network-based action recognition”. *Pattern Recognition Letters*, Vol. 118, pp. 14-22.

Zhang, Y., Zhang, X., Sun, H., Fan, Z. and Zhong, X., 2019. “Portable brain-computer interface based on novel convolutional neural network”. *Computers in Biology and Medicine*, Vol. 107, pp. 248-256.

8. NOTICE

The authors are the only responsible for the printed material included in this paper.