Automatic Emotion Recognition for the Calibration of Autonomous Driving Functions

electronics

MDPI

Article

Automatic Emotion Recognition for the Calibration of Autonomous Driving Functions

Jacopo Sini *, Antonio Costantino Marceddu and Massimo Violante

Department of Control and Computer Engineering, Politecnico di Torino, 10129 Turin, Italy; antonio.marceddu@polito.it (A.C.M.); massimo.violante@polito.it (M.V.)

* Correspondence: jacopo.sini@polito.it

Received: 22 February 2020; Accepted: 18 March 2020; Published: 21 March 2020

Abstract: The development of autonomous driving cars is a complex activity, which poses challenges about ethics, safety, cybersecurity, and social acceptance. The latter, in particular, poses new problems since passengers are used to manually driven vehicles; hence, they need to move their trust from a person to a computer. To smooth the transition towards autonomous vehicles, a delicate calibration of the driving functions should be performed, making the automation decision closest to the passengers' expectations. The complexity of this calibration lies in the presence of a person in the loop: different settings of a given algorithm should be evaluated by assessing the human reaction to the vehicle decisions. With this work, we for an objective method to classify the people's reaction to vehicle decisions. By adopting machine learning techniques, it is possible to analyze the passengers' emotions while driving with alternative vehicle calibrations. Through the analysis of these emotions, it is possible to obtain an objective metric about the comfort feeling of the passengers. As a result, we developed a proof-of-concept implementation of a simple, yet effective, emotions recognition system. It can be deployed either into real vehicles or simulators, during the driving functions calibration.

Keywords: artificial neural networks; automotive applications; autonomous vehicles; emotion recognition; machine learning

1. Introduction

The development of Autonomous Vehicles (AVs) poses novel problems regarding ethics, safety, cybersecurity, and social acceptance. It is expected that these vehicles will be safer with respect to the human-driven ones and, thanks to the new connectivity capabilities in terms of both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, more able to reduce the traffic inside cities. It is a disruptive technology that puts on the table issues about safety, security, ethics, and social acceptance. In particular, the latter is an important point to be taken into account, since, if the users do not trust those vehicles, all these advantages will be lost.

We claim that an improvement in the trustiness on these vehicles can also improve their social acceptance. Of course, acceptance is a complex and multi-faceted phenomenon [1]. Acceptance studies are a novel field but, among authors, the idea that technological improvements can be assessed only when considered as part of a social, economic, and usage-related context is widespread. Considering that the development of the AVs is in the prototype stage, there are many activities aimed at improving these vehicles. The first, of course, are those related to the development of the driving algorithms. Such algorithms, other than the instruction sequences, need also a huge set of calibration parameters that can be equivalent from the safety and vehicle stressing point of view, but that can have different effects from the passengers' perspective. As an example, the way in which an autonomous vehicle decides to approach a road bend, either moving toward the center of the lane or towards the side of

the road, can be perceived in different ways by people depending on the surrounding conditions, such as weather or other traffic, and their emotional states.

Similarly, the AV may react to a fault according to many different strategies to preserve functional safety [2]. This is possible thanks to the mitigation strategies added by the manufacturers to isolate the effects of dangerous faults, making them safe. Unfortunately, it is not always possible to maintain the same level of performance. In this case, we are in the presence of *graceful degradation* mitigation techniques that guarantee the same safety level of the original functions but with less comfort. On a prototype vehicle, it is possible to inject the fault we want to assess, engaging its failure effect mitigation algorithm. At this point, thanks to the proposed emotions recognition system, it is possible to evaluate the effect of the algorithm on the passengers' feelings, obtaining an objective metric on how the performance degradation can be acceptable by the users' perspective.

The main contribution of this paper is the implementation of a proof-of-concept system to improve calibrations of AVs driving algorithms. The idea is to provide objective classifications of the passengers' reactions to autonomous vehicle decisions, thus helping the driving functions calibration on the basis of an analysis that is less sensitive to the subjectivity and variability of post-drive responses to questionnaires. To achieve this result, we developed a system to recognize the emotions that the passengers are feeling with different calibrations in an objective and automatic manner. We adopted known techniques, such as emotions recognition through neural networks and 3D ambient reconstruction. To improve the emotions recognition accuracy, we chose and trained state-of-the-art neural networks on databases available in the literature. These databases are often used individually for training and test phases, but we thought that merging them would improve networks generalization capabilities. We largely described the training results to allow readers to evaluate the accuracy of the system. Moreover, we prepared an environment to perform the emotion detection, able to work off-line or in real-time. The off-line environment is intended for post-processing videos, recorded during driving sessions, for classifying human emotions, while providing to observers the ability to manually validate the neural network results. Conversely, the real-time environment is intended for use during test drive sessions, as it performs on-the-fly classification of human emotions.

The rest of the paper is organized as follows. Section 2 presents the state of the art on emotion recognition through neural networks and the development of autonomous vehicles. Section 3 describes the proposed approach. Section 4 describes the training results obtained with the chosen neural networks. Section 5 describes the experimental results obtained with the proof-of-concept system. Section 6 draws some conclusions.

2. State of the Art

As humans, we are instinctively able to determine the emotions that our fellows are feeling. It is well known that facial expressions are a fundamental part of this social ability. In the 1970s, the American psychologist Paul Ekman scientifically studied this phenomenon. In 1972 [3], he published the list of the six primal emotions shared among all human groups, independently from their culture: anger, disgust, fear, happiness, sadness, and surprise. In the following years, he and other researchers added to this list other emotions. For our purposes, and considering n the labels available in the chosen facial expressions databases, we considered only eight basic emotions: the six proposed in 1972 plus contempt and neutrality. For our automotive application, however, we were interested in recognizing only five of them: fear, happiness, neutrality, sadness, and surprise. We chose to keep recognition of all of them to make the obtained results interesting for a wider audience.

Ekman developed also the *Facial Action Coding System* (FACS) [4]. Facial expressions are performed thanks to facial muscles; hence, they are, from the physical point of view, possible configurations of those that are moved one by one or in groups. These groups of muscular movements are called *Action Units* (AUs). Thus, it is possible to classify a facial expression resorting to a weighted evaluation of those AUs. Thanks to these evaluations, it is possible to make the facial expressions determination more objective. However, to make things even more complex, the same emotion can be shown with

different groups of AUs, thus there is a huge intraclass variability. If the labeling of the considered facial expression has been performed by analyzing the AUs, the picture is marked as FACS encoded. Furthermore, facial expressions can be posed or spontaneous: while the latter are more common to see in everyday life, the former are a more caricatural, exaggerated version of the same.

Various scientists worked on this topic in the years, hence, nowadays, many pictures of posed and spontaneous facial expressions, organized in databases, are available in the literature. The databases selected for this work are:

- The *Extended Cohn–Kanade (CK+) database* [5,6] contains 593 sequences from 123 subjects portrayed in all eight emotional states considered in this document. Each sequence starts from a neutral state and then gradually reaches the peak of the considered emotion. Overall, 327 of the 593 sequences are FACS coded.
- The *Facial Expression Recognition 2013 (FER2013) Database* [7] is composed of 35,887 pictures of 48×48 pixels retrieved from the Internet. Since the original labeling method has demonstrated itself erroneous in some cases, a newer set of annotations named FER+ [8] was released in 2016. It contains labels for 35,488 images since the remaining 399 do not represent human faces, and it also adds the contempt emotion.
- The *Japanese Female Facial Expression (JAFFE) database* [9] contains 213 grayscale photos of posed facial expressions performed by 10 Japanese women. Each image has been rated on six emotional adjectives by 60 Japanese subjects.
- The *Multimedia Understanding Group (MUG) database* [10] contains photos of 86 models posing six emotional states: anger, disgust, fear, happiness, sadness, and surprise. The images of this database are taken inside a photographic studio, thus in controlled illumination conditions.
- The *Radboud Faces Database (RaFD)* [11] is a collection of photos of 67 models, posing all eight emotional states considered in this paper. Each picture was taken from five different angles simultaneously.
- The *Static Facial Expression in the Wild (SFEW 2.0) database* [12] is composed of frames extracted from different movies depicting people having seven different emotional states: anger, disgust, fear, happiness, neutrality, sadness, and surprise. For our purposes, we decided to use only the 1694 labeled aligned images.
- The *FACES database* [13] is a collection of 2052 images taken from 171 actors. They acted two times the following six facial expressions: anger, disgust, fear, happiness, neutrality, and sadness. The actors are further divided into three different age classes.

To the best of our knowledge, in the literature results obtained by merging various facial expressions databases to train a neural network are not available. We thought that this merging operation could be very useful to augment the image variability in terms of the number of portrayed people, light conditions, backgrounds in which the photos were taken, etc. We called these *database ensembles* and we developed an open-source tool to simplify their creation, as described in Section 3.1.

The Society of Automotive Engineers (SAE) defined six levels [14] of driving automation, starting from 0 when the driving is completely in charge of the driver, up to level 5, where the vehicle drives by itself in any condition. Various authors studied the interactions between these automations and humans, focusing especially on how the Advanced Driver Assistance Systems (ADAS) integrated into the car should interact with the driver [15] and about the adaptation of the digital cockpit to different driving situations [16]. Other devices installed inside cars are driver fatigue and drowsiness sensors. They work thanks to a sensor for detecting the steering wheel angle, electrocardiogram performed on the steering wheel surface [17], and cameras that, thanks to a computer vision algorithm, can detect the frequency at which the driver blinks [18].

While these applications are applied during the driving, we are interested in the algorithm calibration phase, before the vehicle is shipped, especially for the trajectory planning (examples of the involved coefficients can be found in [19]). This can help carmakers to choose algorithms and respective calibrations that best suite their customer expectations. To the best of our knowledge,

no author has yet proposed the use of emotion recognition through computer vision to calibrate autonomous driving algorithms.

3. Proposed Approach

As described in the previous section, it is possible to determine people's emotions by their facial expressions. It is not possible to write "by hand" a software function to analyze the pictures of the passengers' faces and determine their emotions with a good performance so we adopted a machine learning approach. We expect that, thanks to a properly trained neural network, it will be possible to solve this challenge. From the operative point of view, we decided to divide the development of the proof-of-concept calibration system into three different phases:

1. We developed a tool, called Facial Expressions Databases Classifier (FEDC) [20], able to perform different operations on the selected databases images in order to prepare them for the training of the neural networks. FEDC can also be used to make the supported databases homogeneous so that they can be merged. We called these derived datasets database ensembles (DE).
2. We chose the most suitable neural networks available from the literature, and trained them with single databases as well as with some database ensembles to compare them by means of objective metrics that we define below.
3. We create 3D graphics reconstructed scenarios depicting some driving situations with different calibrations of the autonomous driving algorithm. By showing them to testers, and analyzing their facial expressions during the representations, we determined what calibrations are preferred by passengers.

3.1. Facial Expressions Databases Classifier

It provides an easy to use Graphical User Interface (GUI) that allows the operator to select the database he/she wants to classify, the output directory, and some post-processing options he/she wants to apply to the images, displaying the current operation with a progress bar and an informative label. More technically, the tool takes the images from the database file provided by the databases' editors, creates a number of folders equal to the number of emotions present in the chosen database, and moves the images. After that, the selected post-processing operations can be applied, in the relative folder, using the cataloging system adopted by databases' editors. This tool has been released as open source under the MIT license on GitHub [20] and it is constantly updated.

3.1.1. Partitioning of the Dataset

To properly train a neural network, it is a good practice to divide the databases into three smaller datasets:

- The training dataset is used to effectively train the network.
- The validation dataset is used to evaluate the neural network performances during the training.
- The test dataset is used to test the capability of the neural network to generalize by using different samples from the ones involved in the training.

If the subdivision option is enabled, FEDC creates the train, test, and optionally the validation folder, each one containing a subfolder containing the related images of every emotion of the selected database. The user can choose how to subdivide the database images for the datasets as a percentage, making using of two sliders in the case that the validation subdivision is disabled, or three.

3.1.2. Performance Enhancement Features

The most recent version of FEDC (4.0.3) can also perform the following operations on the images:

- change image format;
- conversion in grayscale color space;

- crop the images to face only;
- histogram equalization (normal or CLAHE);
- scaling of horizontal and vertical resolutions; and
- transformation of rectangular images into square ones.

3.2. Choice of the Neural Networks

To obtain effective results, we searched for the best neural networks made specifically for facial expression recognition. Our choice fell on the following two state-of-the-art networks, designed with different approaches. Ferreira [21] published in 2018 a deep network that is relatively complex and has a default image size of 120x120 pixels. Miao [22] published in 2019 a shallow neural network that is much simpler and has a default image resolution of 48x48 pixels. Both, in their default configuration, have about 9 million parameters, but, by setting a resolution of images of 48×48 pixels for the network in [21], it is possible to reduce its parameters to about 2 million. This reduction allows performing emotion recognition on single board computers, opening the door to a reduction of the cost of these tests. In this way, it is possible to run the tests in real-time on autonomous vehicle prototypes. This is important since running the tests without storing face images allows increasing the number of testers.

3.3. Calibration Benchmark Applications

After we obtained some suitable neural networks, we used them to assess the effects of different *calibrations* on the passengers' feelings considering different *situations* within common *scenarios*. To perform emotion recognition, we developed a utility software, called *Emotions Detector*, using Java and the OpenCV, DeepLearning4J (DL4J) [23], and Apache Maven libraries. It can acquire both images from a webcam or frames of a prerecorded video, crop them to the face only, apply the post-processing algorithms needed by the neural network, and run the network on them. At the end of the process, the images themselves and their emotions probability distributions are saved automatically to obtain a test report. We defined:

- *Calibration*: A set of parameters that determine the behavior, in terms of trajectory (acceleration ramps, lateral distances from obstacles, and preferred lateral accelerations) and, in general, all the numerical parameter (not considered in this paper) needed to properly develop an AV driving algorithm.
- *Sscenario*: The environment (real or virtual) in which the vehicle's behavior is shown with different calibrations and traffic conditions.
- *Situation*: A combination composed of a calibration, a scenario, and a traffic conditions set, to be shown to testers.

The *situations* can be represented both in simulators and real vehicles. Of course, the use of a real vehicle can give better results, but ensuring the repeatability of the tests requires the use of a closed track and other vehicles for the traffic, making the tests extremely expensive.

4. Neural Networks Training

We decided to focus on neural networks training since the evaluation of their accuracies is fundamental to achieve an objective emotions detection. Section 4.1 describes the training set-up. Section 4.2 describes the metrics to assess the performances of the networks, and ways to improve them performing operation such as cross validation, data augmentation, and normalization. Section 4.3.1 describes the results obtained training the network [21] on the CK+ database. Section 4.3.2 describes the results obtained from the networks trained on the FER2013 database. Section 4.3.3 describes the results obtained training the networks on the database ensembles. For the reader convenience, these results are summarized in Section 4.4.

For the training of the aforementioned neural networks (see Section 3.2), we chose the following databases in order to be able to compare the results of our implementations with those obtained by neural networks' authors:

- CK+, which was used only for the network in [21] because it was not used by the authors of [22]; and
- FER2013.

We also prepared the following two *database ensembles* recurring to FEDC:

- *Ensemble 1*, composed of all the labeled images from all the databases supported by FEDC; and
- *Ensemble 2*, composed of all the posed facial expressions images from the databases CK+, FACES, JAFFE, MUG, and RaFD.

We performed training in 23 different configurations. Table 1 indicates the number of pictures for each emotion that can be found in the chosen databases.

Table 1. Picture available for each emotion in the chosen databases.

Emotion	*Ensemble 1*	*Ensemble 2*	CK+	FER2013
Anger	4328	981	45	4953 3111 [a]
Contempt	788	572	18	0 216 [a]
Disgust	1340	1022	59	547 248 [a]
Fear	1887	950	25	5121 819 [a]
Happiness	10,676	1089	69	8989 9355 [a]
Neutrality	14,196	1070	123	6198 12,906 [a]
Sadness	5524	953	28	6077 4371 [a]
Surprise	5254	656	83	4002 4462 [a]

[a] FER+ annotations.

4.1. Training Environment Set-Up

We chose to use Keras [24] as a high-level abstraction API because it is simple to use and, for some years now, it has been one of the most widely used solutions for neural networks training. It can abstract three different frameworks for machine learning: TensorFlow [25], Microsoft Cognitive Toolkit (CNTK) [26], and Theano [27]. All three proposed solutions adopt an open source-like license. For our purposes, we chose to use TensorFlow. Other utility libraries adopted to speed-up the code writing and to improve the presentation of the experimental results are:

- *Matplotlib* [28], a 2D plotting library for Python;
- *NumPy* [29], a package for scientific computing for Python;
- *Open Source Computer Vision Library* (OpenCV) [30], a computer vision and machine learning software library for C++, Java, MATLAB, and Python;
- *Pandas* [31], which provides high-performance, easy-to-use data structures and data analysis tools for Python; and
- *Scikit-learn* [32], a library for data mining and data analysis.

4.2. Performance Assessment Metrics

An (artificial) neural network is a mathematical system. The name "neural networks" comes from the conceptual similarity to the biological neural system. From the mathematical point of view, a "neuron" is a mathematical function with a certain number q of inputs, $u_1, u_2, ..., u_q$ and one output, y. Those inputs are linearly combined to determine the activation signal s, with the equation $s = \Theta_0 + \Theta_1 u_1 + ... + \Theta_q u_q$. Θ_0 is usually called the bias parameter. After the sum node, a non-linear

function is applied to s, obtaining the output signal $y = \sigma(s)$. $\sigma(s)$ is commonly called activation function. Popular activation functions are historically the sigmoidal function and, nowadays, the ELU, ReLU, and LeakyReLU functions.

Various layers of this kind compose a neural network. In the literature, it is possible to find various neural networks designed for various purposes.

4.2.1. Underfitting and Overfitting

The primary objective of a neural network is to create a model that is able to generalize. This implies that a good model can work in the same way with both already seen and new unseen data. There are two different ways in which the system is unable to achieve this ideal behavior:

- If a model has not learned sufficient characteristics from the input data, it will not be able to generalize towards new data, therefore it will *underfit*.
- Conversely, if it has learned too many features from the training samples, it will limit its ability to generalize towards new data: in this case, the model will *overfit*.

Not all the network parameters are chosen during the training. Some of them have to be set before the training or are determined by the neural network structures. The former are called *hyperparameters*. Before describing the experimental results, it is better to define some terms:

- *Learning rate* defines the update "speed" of the parameters during the training. If it is lower with respect to the ideal one, the learning is slowed down but become smoother; on the contrary, if its value is too high, the network can diverge or underfit.
- *Sample* is an element of a database. In our case, it is a picture of a human face with a facial expression properly labeled with the represented emotion.
- *Batch* is a set of N samples processed independently and in parallel. During the training process, a batch corresponds to a single update of the network parameters.
- *Epoch* is usually a passage on the entire dataset and corresponds to a single phase of the training.

For each experiment, we computed these metrics:

- *Accuracy* is defined as
$$\alpha = \frac{P_r}{P} \tag{1}$$
where P_r is the number of correct predictions and P is the number of total predictions. For this metric, the higher is the better.
- *Loss* represents how bad the model prediction is with respect to a single sample. For this metric, the lower is the better. In the literature, there are l many different methods to compute this parameter, such as binary cross-entropy, categorical cross-entropy, mean absolute deviation, mean absolute error, mean squared error, Poisson, squared hinge, and so on. For our purposes, we chose to compute this metric as a categorical cross-entropy, defined as:
$$L(y, \hat{y}) = \Sigma_{j=0}^{M} \Sigma_{i=0}^{N} (y_{ij} \cdot \log(\hat{y}_{ij})) \tag{2}$$
This loss function must be used for single label categorization, i.e. when only one category is applicable for each data point. It is perfectly suited to our cases, since we formulated the hypothesis that each image (sample) can represent only one of the considered emotions (category).

  In particular, the curve composed by the various losses computed in each epoch, called loss curve in the literature, is important to determine if the model underfits or overfits. If the training dataset loss curve is much greater than the one of the validation dataset, we are in underfitting conditions. If the loss curves are near, we probably obtained a good model. Finally, if the loss curve of the training dataset is instead much lower than that of the validation dataset, it indicates the presence of overfitting [33].

- *Confusion matrix*: Considering that the classification system has been trained to distinguish between eight different emotions, the confusion matrix summarizes the result of the testing of the neural network. It is a particular contingency table in which emotions are listed on both sides. In the top row, there are the labels of the pictures (ground truths), while in the left column there are the predicted categories (emotions).

4.2.2. Cross Validation

To reduce overfitting, it is possible to adopt the cross-validation technique. It consists in partitioning the dataset into multiple subsets, some of which are used for training and the remaining for validation/testing purposes. In the literature are described various kinds of techniques, such as Leave-One-Out Cross-Validation (LOOCV), k-Fold, Stratified, and Time-Series. Stratified is used when we are dealing with binary classification problems, while Time-Series is used when the dataset is composed of observation made at different times; hence, these two are not suitable for our purposes. For this work, LOOCV or k-Fold can be chosen. We chose the latter, putting $k = 9$. In the k-Fold, the dataset is split into k folds (subsets) of approximately the same size: $k - 1$ folds are used for training, while the remaining one is used for validation or test. Using the FEDC database subdivision function, we divided the database into two subsets: one containing 90% of the images, which was used for training and validation, while the remaining 10% was used to perform the test. Before the training, we further split the first subset into nine smaller subsets: eight of them were used for training, while the remaining one was used for validation. Changing the validation subset after each training, it was possible to perform nine different training of the neural network, in order to pick the one that performed better.

4.2.3. Data Augmentation

Data augmentation is adopted when a low number of samples is available. The idea is to modify them in different ways in order to artificially increase their number. For example, in the case of images, augmented ones can be obtained by rotating, reflecting, applying translations, and so on. In this way, it is possible to improve the generalization capability of the network without modifying the model. For our purposes, we applied different transformations on the images. In all the training, we applied these data augmentations:

- brightness range from 50% to 100%;
- random horizontal flip enabled;
- rotation interval between $\pm$2.5 deg;
- shear range of $\pm$2.5%;
- width and height shift range of 2.5%; and
- zoom transformation interval between $\pm$2.5%.

4.2.4. Normalization

To improve the training process, we applied, alternately, two different normalizations to the grayscale space of the images: *[0,1]* and *z-score normalization*. The [0,1] normalization is a particular case of the *scaling to range normalization*, defined generally by the formula:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

in which x_{min} is set to 0 and x_{max} is set to 1. The *z-score normalization*, sometimes called *standardization*, is used to obtain a distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$. The applied formula is:

$$x_z = \frac{x - \mu}{\sigma} \tag{4}$$

in which x represent the 8-bit brightness value of the images, x_{min} and x_{max} are, respectively, the minimum and the maximum brightness within the images, μ is the arithmetic average of the brightness of all the pixels of the images, and σ its standard deviation.

4.3. Training Results

We implemented the networks within the Keras environment, as described elsewhere. The code to train the network can be found at [34]. For the network in [22], we did not encounter any problems, while, for the network in [21], we faced an ambiguity in the "e-block" layer because, in the paper, it is not clearly described how to implement the relative "crop and resize" operation. We decided to implement it as a single convolutional layer, in which the kernel size is defined according to the resolution of the input images. For 120 × 120 pixels images, which is the default input size for the network, the kernel size is 106 × 106, while,for 48 × 48 pixels images, which is the size of the picture of the FER2013 database, the kernel size is 43 × 43. In both cases, we have set the number of output filter to 64, in order to make the next multiplication operation possible. We trained the networks with these datasets:

- CK+ database [5,6] (only for the network in [21]);
- FER2013 [7];
- *Ensemble 1*; and
- *Ensemble 2*.

For each training, we used the "EarlyStopping" callback to stop the training if, after 18 consecutive epochs, there was no improvement in the loss curve computed on the validation dataset. In some trainings, we also set the "ReduceLROnPlateau" callback to multiply the learning rate by 0.99 or 0.95 in every epoch.

To avoid being excessively long and boring, we only report the most interesting cases. The other cases can be found in [35]. The cases we selected are in bold in Table 3: for each of them, we report its accuracy graph, its loss graph, and its confusion matrix.

As hyperparameters, we set:

- batch size: 100 (except for the network in [21] trained on the CK+ database, where it was set as 50);
- maximum number of epochs: 1000; and
- learning rate: 0.001.

4.3.1. CK+ Database

The first training of the network in [21] was performed on the CK+ database, resized to a resolution of 120 × 120 pixels. With the previously said division, it was split in this way: 364 images were used for training, 46 images were used for validation, and 40 images were used for testing.

The obtained results are shown in Figures 1–3.

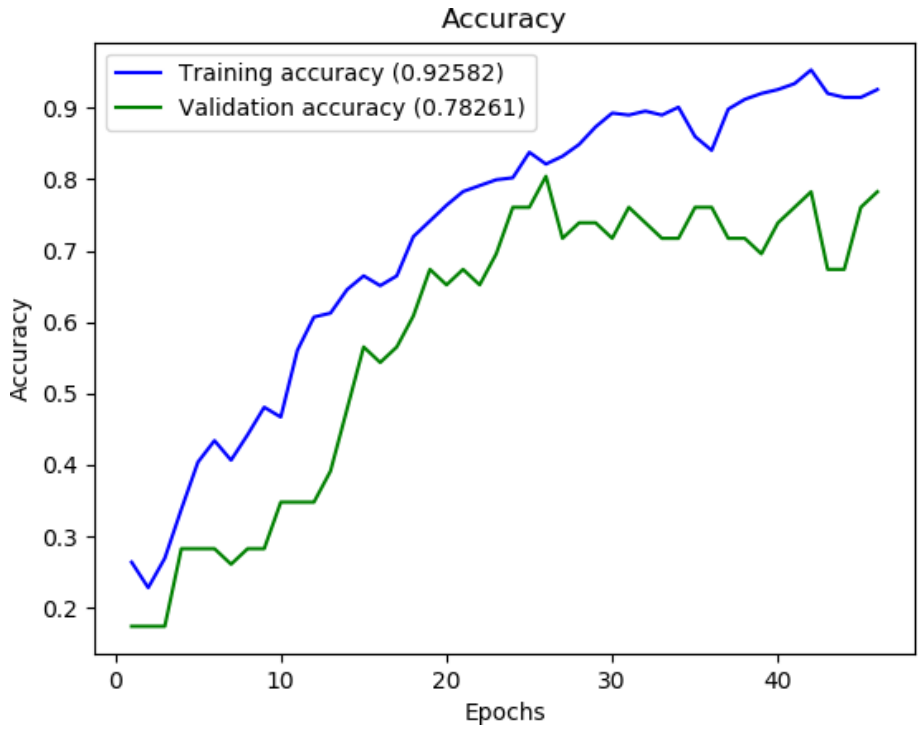

Figure 1. Accuracy graph of the network in [21], trained with the CK+ database using the data augmentation and the z-score normalization. Figure from [35].

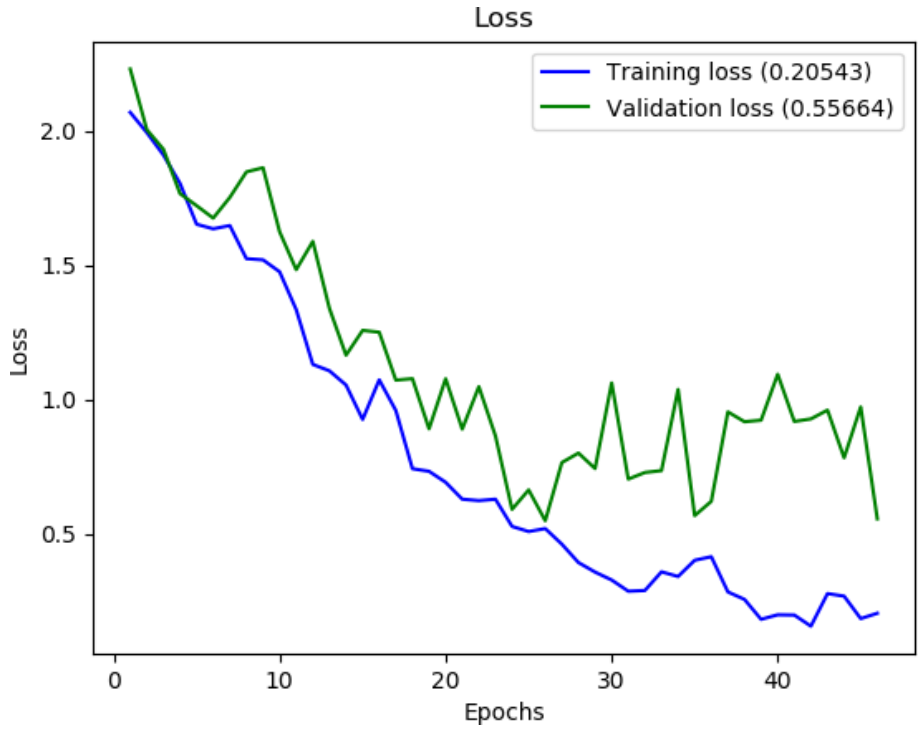

Figure 2. Loss graph of the network in [21], trained with the CK+ database using the data augmentation and the z-score normalization. Figure from [35].

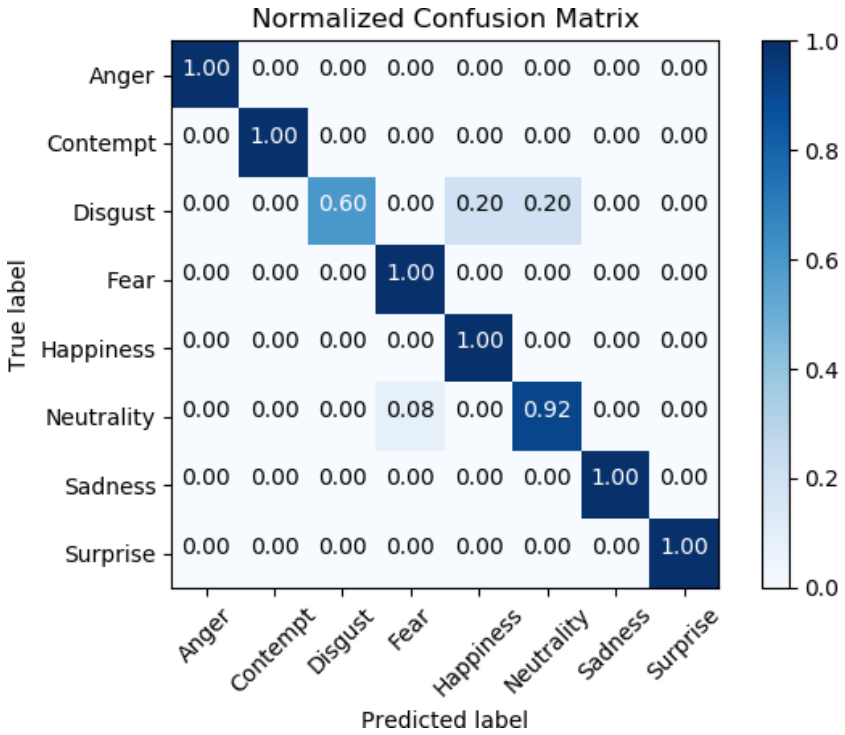

Figure 3. Normalized confusion matrix of the network in [21], trained with the CK+ database using the data augmentation and the z-score normalization. Figure from [35].

The results obtained are in line with the one presented in [21], thus our implementation seems to work properly.

4.3.2. FER2013 Database

The FER2013 [7] database has a huge number of pictures, but the resolution of the images is only 48x48 pixels. Instead of performing an upscaling of these pictures, we decided to modify the network in [21] to work, as described previously, with these low-resolution images.

We used the same settings and hyperparameters adopted from the CK+ database, increasing only the batch size to 100. We therefore obtained 28,712 images for training, 3590 for validation, and 3585 for testing.

With this database, we obtained accuracies around 60%: a not impressive result, surely improvable but also undermined from the sometimes dubious labels and to the presence, in the database, of some images that do not represent human faces. Thus, we decided to use the FER+ [8] annotations, which allowed us to remove erroneous images and to improve ground truth.

The best results in terms of test accuracy on this database were obtained from the network in [21], and are shown in Figures 4–6.

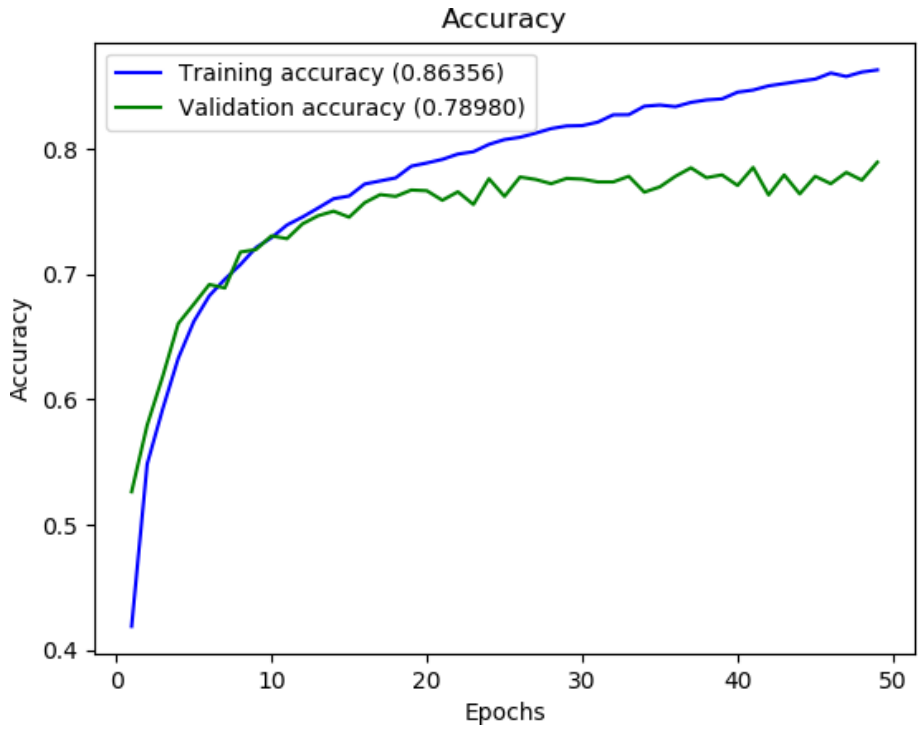

Figure 4. Accuracy graph of the network in [21], trained with the FER2013 database with the FER+ annotations using the data augmentation and the z-score normalization. Figure from [35].

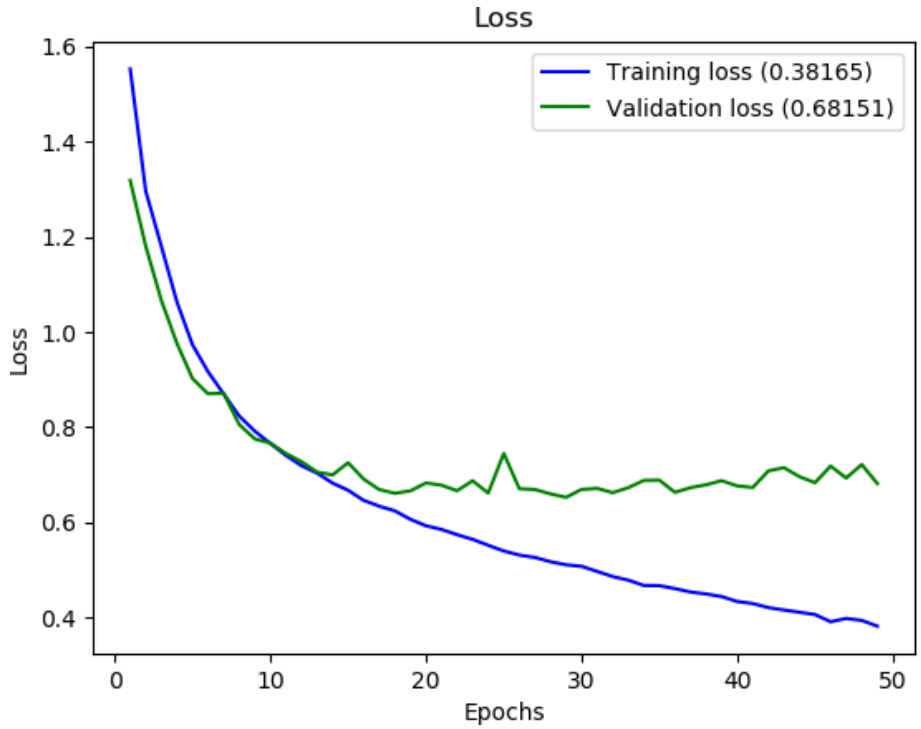

Figure 5. Loss graph of the network in [21], trained with the FER2013 database with the FER+ annotations using the data augmentation and the z-score normalization. Figure from [35].

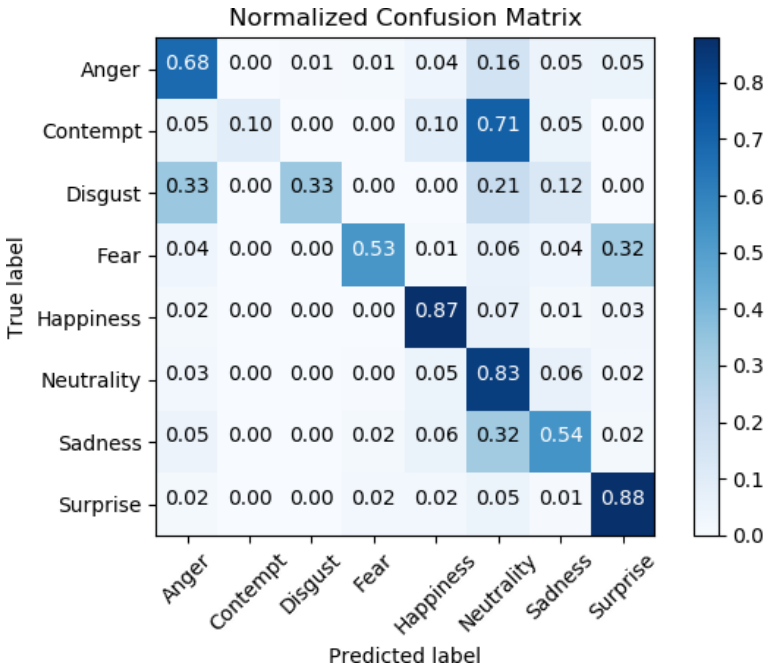

Figure 6. Normalized confusion matrix of the network in [21], trained with the FER2013 database with the FER+ annotations using the data augmentation and the z-score normalization. Figure from [35].

As shown by the confusion matrix (see Figure 6), the trained network is quite good at detecting happiness, neutrality, and surprise, while it is weak at detecting fear and sadness. We also have poor performance in the recognition of contempt and disgust, but these emotions are not important for our purposes. Since FER2013 is known to be a not well-balanced database, and considering that also the network in [22], trained with the same settings and on the same databases, presents a similar confusion matrix (see Figure 7), our hypothesis is that the FER2013 database does not provide sufficient examples for contempt, disgust, and, more important for our application, fear and sadness classes.

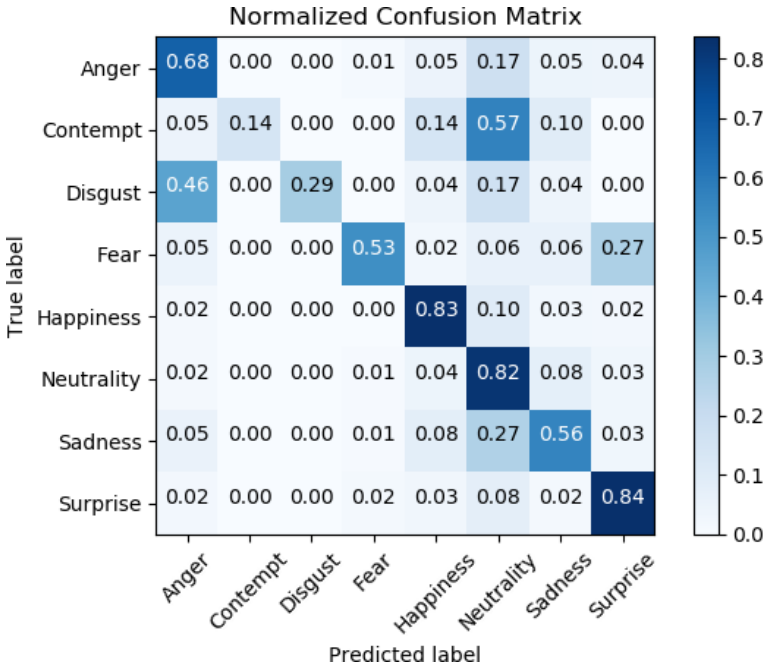

Figure 7. Normalized confusion matrix of the network in [22], trained with the FER2013 database with the FER+ annotations using the data augmentation and the z-score normalization. Figure from [35].

4.3.3. Database Ensembles

We decided to train again the neural networks using two different *database ensembles*: one containing the images, posed and spontaneous, of all the databases supported by FEDC and one containing only the posed ones. These were obtained using FEDC, applying a conversion to the grayscale color space and a face detection algorithm in order to crop the images on human faces. Both were created downscaling all the images to 48 × 48 pixels, in order to adapt them to those of the FER2013 database and to be able to compare the results of the two databases placed under the same conditions. For the FER2013 database, we chose to also use the FER+ annotations, because the improvement in accuracy due to their use is relevant.

The *Ensemble 1* database is composed of all the available images from the database supported, for now, by FEDC. Making use of the same subdivision procedure used in the previous examples, we obtained 35,212 images for training, 4402 for validation, and 4379 for testing. Results are shown in Figures 8–10.

The obtained results are better, in terms of classification errors, than those obtained using the databases individually, especially for the contempt and disgust classes, which had accuracies similar to random ones.

The *Ensemble 2* database is a subset of *Ensemble 1* composed only of posed images. Thanks to the FEDC subdivision procedure, we obtained 5847 images for training, 731 for validation, and 715 for testing.

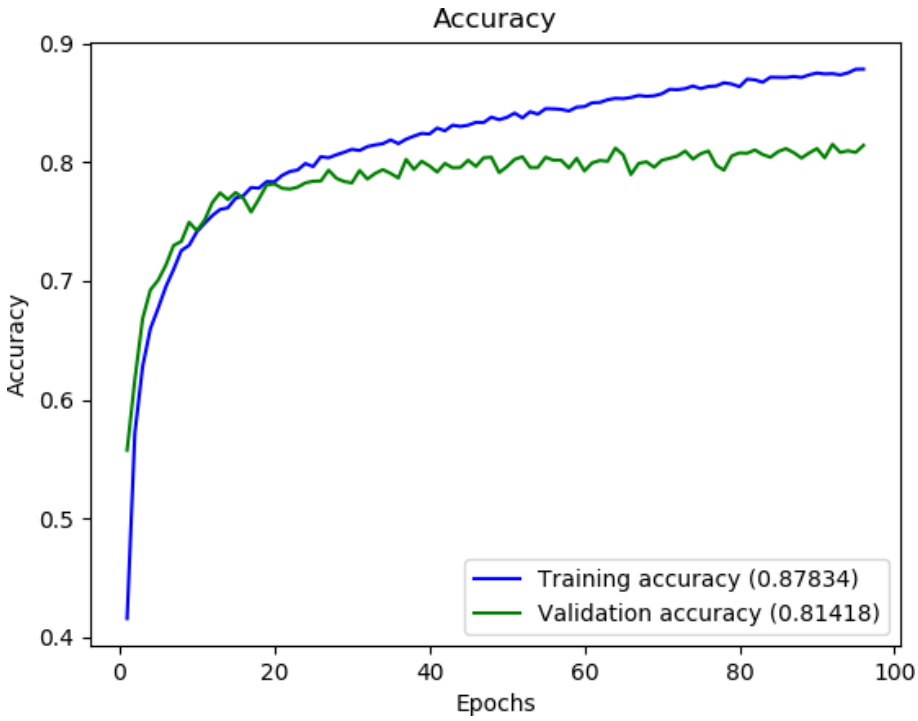

Figure 8. Accuracy graph of the network in [21], trained with the *Ensemble 1* database using the data augmentation and the z-score normalization. Figure from [35].

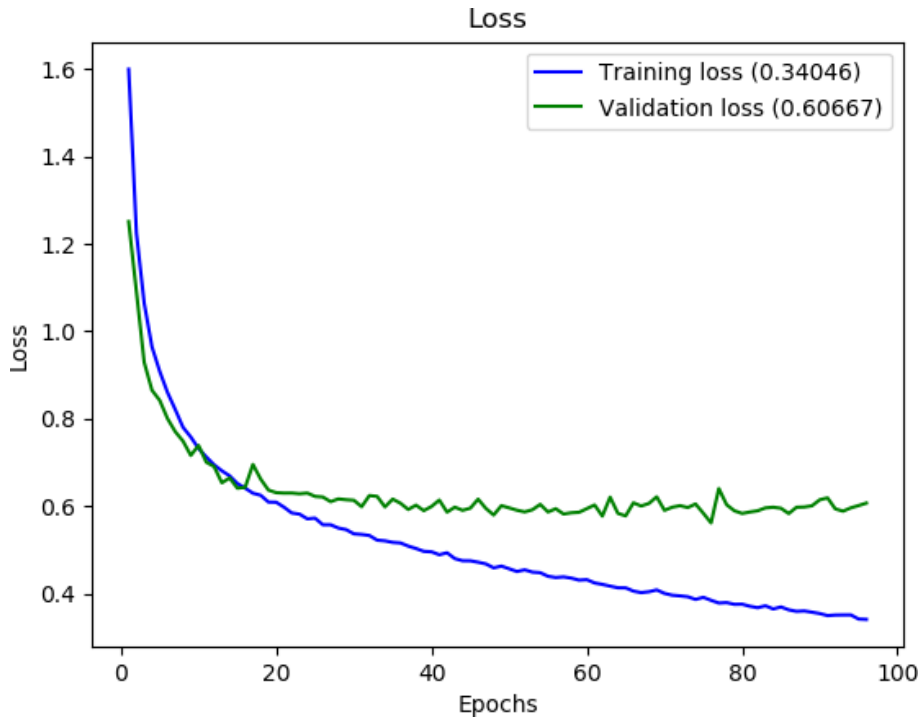

Figure 9. Loss graph of the network in [21], trained with the *Ensemble 1* using the data augmentation and the z-score normalization. Figure from [35].

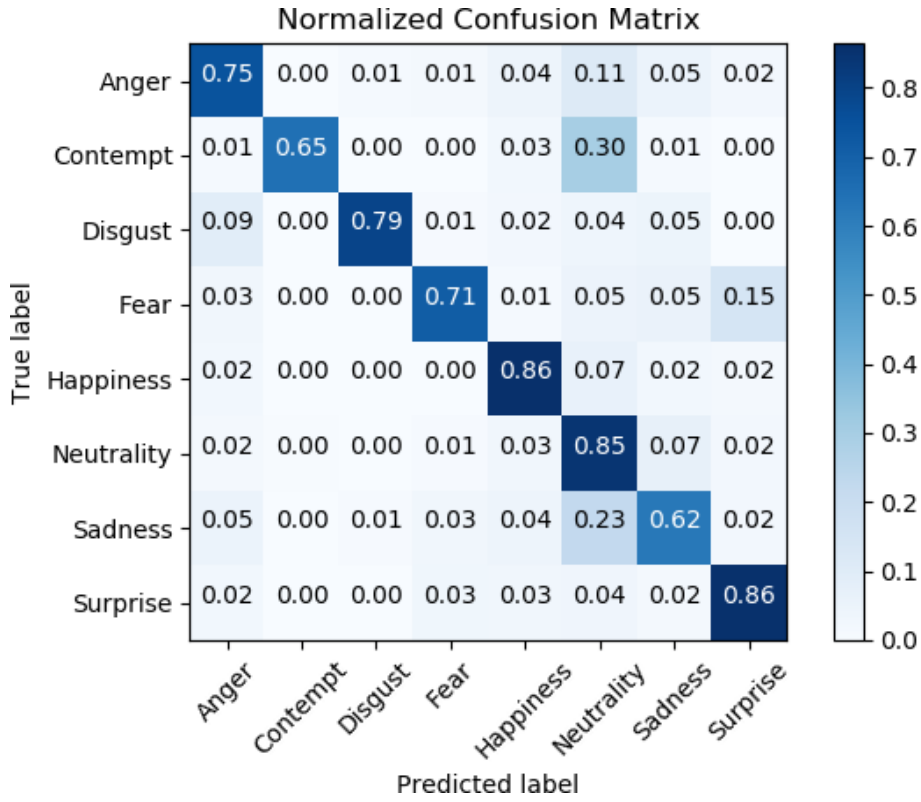

Figure 10. Normalized confusion matrix of the network in [21], trained with the *Ensemble 1* database using the data augmentation and the z-score normalization. Figure from [35].

4.4. Summary of Training Results

For reader convenience, we summarized the obtained results into two tables. Table 2 contains the numbers of photos for training, validation, and test datasets, while Table 3 contains the best training accuracies obtained per-database and neural network. The test accuracies of the cases shown in detail in the paper are in bold. In general, the network in [21] requires more time for training, but has slightly better performance and, with the same image size, requires fewer parameters than the one in [22]. Thus, if we had to choose a network, we would certainly pick the first one. Before continuing, it is important to make an observation: even if the test accuracies of the trainings made with the *Ensemble 2* database are better with respect to the ones obtained with *Ensemble 1*, we expect better result on the field from the networks trained with the latter. This is because the spontaneous expressions are those that we can observe most commonly in everyday life, while those posed are more caricatural and deliberately exaggerated: this makes the interclass difference greater, but, at the same time, a network that is trained with these images will inevitably experience a bias between the images on which it is trained and those in which a prediction will actually be requested.

Table 2. Subdivision of the databases.

	Ensemble 1	***Ensemble 2***	**CK+**	**FER2013**
Train	35212	5847	364	28712
Validation	4402	731	46	3590
Test	4379	715	40	3585

Table 3. Test accuracies summary table (best values).

	Ensemble 1	***Ensemble 2***	**CK+**	**FER2013**
[21]	74.79 % 78.85 [b] % **80.38 [c] %**	94.69 % 97.20 [b] % 97.34 [c] %	82.50 % 92.50 [b] % **92.50 [c] %**	56.57 % 61.40 [b] % 62.46 [c] % **78.36 [a,c] %**
[22]	71.39 % 76.91 [b] % 78.47 [c] %	92.59 % 94.83 [b] % **96.78 [c] %**	N.A.	54.03 % 61.67 [b] % 62.71 [c] % 76.67 [a,c] %

[a] FER+ annotations. [b] [0,1] normalization. [c] z-score normalization.

5. Experimental Results

5.1. Situations Preparation

We proposed five different *calibrations* of the autonomous driving algorithm in two different *scenarios*. By combining those *calibrations* and *scenarios*, we prepared 11 benchmark *situations*. The first six of them (identified in the following as Cn) involve as *scenario* a curve to the right in a suburban environment. The car can face it with three different *calibrations*, hence following three different trajectories: strictly keeping the right side (*situations* C1 and C4), keeping the center (*situations* C2 and C5) of the lane, or widening at the entrance of the curve to decrease the lateral accelerations (*situations* C3 and C6). Since in all these cases the vehicle remains within its lane, all these behaviors are allowed by the majority of road regulations.

The other five, instead (identified in the following as Tn), have as *scenario* a right turn inside an urban environment. In the road just taken, there is an obstacle that obstructs the rightmost lane. The road has two lanes for each direction of travel. With the first *calibration* (*situations* T1, T2, and T3), the car tries to stay at the right with a lot of decision, therefore it suddenly discards the obstacle. With the

second *calibration* (*situations* T4 and T5), instead, the car decides to widen the turn in advance and to move to the right lane only after it passed the obstacle.

5.2. Criteria for Emotion Analysis

As described in Section 3.3, for each of the considered *situations*, we prepared a 3D representation. The most relevant emotion, when different from neutrality or sadness, was taken into consideration. We considered fear and surprise as negative emotions, while happiness as positive ones. Sadness and neutrality have been considered as middle values since the network appears to little appreciate the differences between these moods. In any case, if there was no other emotion than neutrality or sadness, the one with a greater number of sadness outcomes was considered worse with respect to ones that score more neutrality outcomes. Since the neural networks can recognize also anger, contempt, and disgust, we considered those outcomes as *experiment failures* because these moods are not the ones we expected to obtain in our tests.

5.3. Experimental Campaign

We asked eight people, six males and two females, average ages 25 years, interval 23–31 years, to watch the *situations*, starting from a black screen and without describing what they would see to not interfere with their moods. We detected their emotions every 2 s.

We represented the *situations* in the order: *T2-T4-T3-T1-T5-C1-C5-C2-C6-C3-C4*. We chose to not mix the urban (T) and suburban (C) *scenarios* to not break the environments immersion. In the urban *scenario*, we placed the *situations* that we expected to provoke greater emotional reactions in the middle of the representation, while, in the suburban one, we started from the softer one moving to the most critical at the end.

For the tests, we used a flat projection screen to be able to choose the point of view, avoiding, in this way, that the tester could not be able to see the critical moments represented. The use of a virtual reality set could improve the environment immersion, but, since we were using an emotion recognition technique that requires to see the entire face, the use of a device of this kind is not possible.

5.4. Results Discussion

The experimental results in Table 4 show that *situations* T2 and C6 are the most stressful from the passengers' points of view. In the urban *scenario*, there are some positive reactions to the *situation* T3, probably due to the capability of the vehicle to make the safest decision by keeping in the right lane and stopping in front of the obstacle. In addition, the *situation* T4, which is the one that minimizes the lateral movement of the car, is appreciated. With traffic, the *calibrations* shown in the *situations* T1 and T5 appears to be equivalent. Regarding the curve *scenario*, the *calibration* shown in C3 and C6 is preferred when there is no traffic from the other direction (*situation* C3). Oppositely, for the one where the car stays at the right side of its lane (C1 and C4), it is preferred the *situation* C4 in which there is traffic in the other direction. The *calibration* shown in C2 and C5 are not appreciated: in our opinion, this is due to the unnatural path that follows the centerline of the lane.

These preliminary results agree with the experiences reported by the testers when they were interviewed after the tests. In particular, asking about the *situations* C3 and C6, it emerged that the C3 one, in which the curve is traveled keeping the left side of the lane, is more appreciated without traffic in the opposite direction. Instead, following the same trajectory with traffic, as in the *situation* C6, causes inconveniences to the passengers.

Table 4. Emotional effects of the benchmark tests. In the columns are indicated the number of people that reacted to the considered *situation* with the emotion on the left. Data obtained by the network in [21], trained with the *Ensemble 1* database using the data augmentation and the z-score normalization.

	T1	***T2***	***T3***	***T4***	***T5***	***C1***	***C2***	***C3***	***C4***	***C5***	***C6***
Fear	0	0	0	0	0	0	0	0	0	0	0
Sadness	5	7	2	3	4	1	4	3	4	4	5
Surprise	0	0	1	1	0	0	0	0	0	0	0
Happiness	0	0	2	2	0	3	0	1	3	1	0
Neutrality	3	1	3	2	4	4	4	4	1	3	3
Experiment failure	0	0	0	0	0	0	0	0	0	0	0

T1: The vehicle takes the right-turn curve in a step way, staying at as possible at the right of the street. The car discards the obstacle when it is really close to keep the right rigorously, then re-enters the rightmost lane immediately after it. No other traffic. *T2*: Same *situation* as in *T1*, but with incoming traffic from the opposite direction. *T3*: Same *situation* as in *T1*, but traffic from the same direction of the passenger's vehicle supersede it, preventing the algorithm to move around the obstacle. The vehicle stops in front of it, then starts again moving around the obstacle. *T4*: The vehicles take the right-turn curve entering in the left lane, super-seed the obstacle then move to the right lane. *T5*: Same *situation* as *T4*, but with incoming traffic from the opposite direction. *C1*: The vehicle runs through the curve keeping strictly along the right edge of the road. No other vehicle comes from the other direction of travel. *C2*: The vehicle travels the curve keeping in the center of its lane. No other vehicle comes from the other direction of travel. *C3*: The vehicle travels the curve widening to the left at the entrance, then closing it to the right, to reduce lateral accelerations. *C4*: Same *situation* as in *C1*, but with traffic from the opposite direction. *C5*: Same *situation* as in *C2*, but with traffic from the opposite direction. *C6*: Same *situation* as in *C3*, but with traffic from the opposite direction.

6. Conclusions

This paper proposes a proof-of-concept way to smooth the transition towards autonomous vehicles. To improve the passengers' trustiness on these vehicles, a delicate calibration of the driving functions should be performed, making the AV decisions closest to the ones expected by the passengers. We adopted machine learning techniques to recognize passengers' emotions, making it possible to obtain an objective comparison between various driving algorithm *calibrations*. To achieve this result, we chose two state-of-the-art neural networks, implemented, trained, and tested in different conditions. We developed two software tools, called Facial Expressions Databases Classifier and *Emotions Detector*. The first, designed to generate large facial expressions pictures databases by merging and processing images from various databases, has been released under the MIT open-source license on GitHub [20]. The second has been developed for internal use to analyze the testers' emotions during the *situations* representations. The proposed methodology has demonstrated itself able to help designers to choose between different calibrations of the trajectory planner when applied considering two different conditions.

As future work, we would like to improve our results by using an improved car simulator, with motion capabilities and a curved screen, to improve the immersion in the simulated environment, and by increasing the number of testers to obtain analysis with statistically-relevant results.

Author Contributions: Conceptualization, J.S. and M.V.; project administration J.S., methodology, J.S. and A.C.M.; software, A.C.M.; validation, J.S. and A.C.M.; visualization A.C.M.; writing—original draft preparation, J.S. and A.C.M.; writing—review and editing, J.S. and A.C.M.; supervision, M.V.; and funding acquisition, M.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AV	Autonomous Vehicles
ADAS	Advanced Driver Assistance Systems
V2V	Vehicle-to-vehicle
V2I	Vehicle-to-infrastructure
FACS	Facial Action Coding System
AU	Action Units
FER2013	Facial Expression Recognition database
JAFFE	Japanese Female Facial Expression database
MUG	Multimedia Understanding Group database
RaFD	Radboud Faces Database
SFEW 2.0	Static Facial Expression in the Wild database
ECU	Electronic Control Unit
FEDC	Facial Expression Databases Classifier
DE	Database Ensembles
GUI	Graphical User Interface
LOOCV	Leave-One-Out Cross-Validation
SAE	Society of Automotive Engineers
ZCA	Zero-phase Component Analysis

References

1. Fraedrich, E.; Lenz, B. Societal and Individual Acceptance of Autonomous Driving. In *Autonomous Driving*; Maurer, M., Gerdes, J.C., Lenz, B., Winner, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2016. [CrossRef]
2. ISO 26262-1:2018. *Road Vehicles—Functional Safety*; International Organization for Standardization: Geneva, Switzerland, 2018.
3. Ekman, P. Basic emotions. In *Handbook of Cognition and Emotion*; University of California: San Francisco, CA, USA, 1999; pp. 45–60.
4. Ekman, P.; Friesen, W. *Facial Action Coding System (FACS): A Technique for the Measurement of Facial Action*; Consulting: Palo Alto, CA, USA, 1978.
5. Kanade, T.; Cohn, J.F.; Tian, Y. Comprehensive database for facial expression analysis. In Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), Grenoble, France, 28–30 March 2000; pp. 46–53. [CrossRef]
6. Lucey, P.; Cohn, J.F.; Kanade, T.; Saragih, J.; Ambadar, Z.; Matthews, I. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, San Francisco, CA, USA, 13–18 June 2010; pp. 94–101. [CrossRef]
7. Goodfellow, I.J.; Erhan, D.; Carrier, P.L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.H.; et al. Challenges in Repre- Sentation Learning: A Report on Three Machine Learning Contests. 2013. Available online: https://www.sciencedirect.com/science/article/pii/S0893608014002159 (accessed on 7 October 2019).
8. Barsoum, E.; Zhang, C.; Ferrer, C.; Zhang, Z. Training deep networks for facial expression recognition with crowd-sourced label distri-bution. *arXiv* **2016**, arXiv:1608.01041.
9. Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding facial expressions with gabor wavelets. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 200–205. [CrossRef]
10. Aifanti, N.; Papachristou, C.; Delopoulos, A. The mug facial expression database. In Proceedings of the 11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10, Desenzano del Garda, Italy, 12–14 April 2010; pp. 1–4.
11. Langner, O.; Dotsch, R.; Bijlstra, G.; Wigboldus,D.; Hawk, S.; Knippenberg, A. Presentation and validation of the radboud face database. *Cogn. Emot.* **2010**, *24*, 1377–1388. [CrossRef]

12. Dhall, A.; Goecke, R.; Lucey, S.; Gedeon, T. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 2106–2112. [CrossRef]
13. Ebner, N.; Riediger, M.; Lindenberger, U. Faces a database of facial expressions in young, middle-aged, and older women and men: Development and validation. *Behav. Res. Methods* **2010**, *42*, 351–362. [CrossRef] [PubMed]
14. SAE. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-Road Motor Vehicles (Surface Vehicle Recommended Practice: Superseding J3016 Jan 2014)*; SAE International: Warrendale, PA, USA, 2016.
15. Hasenjäger, M.; Wersing, H. Personalization in Advanced Driver Assistance Systems and Autonomous Vehicles: A Review. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017. [CrossRef]
16. Choi, J.K.; Kim, K.; Kim, D.; Choi, H.; Jang, B. Driver-Adaptive Vehicle Interaction System for the Advanced Digital Cockpit. In Proceedings of the International Conference on Advanced Communications Technology(ICACT), Chuncheon-si Gangwon-do, Korea, 11–14 February 2018. [CrossRef]
17. Jung, S.J.; Shin, H.S.; Chung, W.Y. Driver Fatigue and Drowsiness Monitoring System with Embedded Electrocardiogram Sensor on Steering Wheel. *IET Intell. Transp. Syst.* **2014**, *8*, 43–50. [CrossRef]
18. Kusuma, B.M.; Sampada, S.; Ramakanth, P.; Nishant, K.; Atulit, S. Detection of Driver Drowsiness using Eye Blink Sensor. *Int. J. Eng. Technol.* **2018**, *7*, 498. [CrossRef]
19. Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J.Z.; Langer, D.; Pink, O.; Pratt, V.; et al. Towards fully autonomous driving: Systems and algorithms. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 163–168. [CrossRef]
20. Facial Expressions Databases Classifier. Available online: https://github.com/AntonioMarceddu/Facial_Expressions_Databases_Classifier (accessed on 20 March 2020).
21. Ferreira, P.M.; Marques, F.; Cardoso, J.S.; Rebelo, A. Physiological inspired deep neural networks for emotion recognition. *IEEE Access* **2018**, *6*, 53930–53943. [CrossRef]
22. Miao, S.; Xu, H.; Han, Z.; Zhu, Y. Recognizing facial expressions using a shallow convolutional neural network. *IEEE Access* **2019**, *7*, 78000–78011. [CrossRef]
23. Eclipse Deeplearning4j Development Team. Deeplearning4j: Open-sOurce Distributed Deep Learning for the JVM, Apache Software Foundation License 2.0. 2019. Available online: http://deeplearning4j.org/ (accessed on 20 March 2020).
24. Keras. Available online: https://keras.io/ (accessed on 20 March 2020).
25. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 20 March 2020).
26. The Microsoft Cognitive Toolkit. Available online: https://docs.microsoft.com/en-us/cognitive-toolkit/ (accessed on 20 March 2020).
27. Theano. Available online: http://deeplearning.net/software/theano/ (accessed on 20 March 2020).
28. Hunter, J.D. Matplotlib: A 2d graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]
29. NumPy. Available online: https://numpy.org/ (accessed on 20 March 2020).
30. Bradski, G. The OpenCV Library. Dr Dobb's Journal of Software Tools, 2000.
31. McKinne, W. Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference, Austin, Texas; van der Walt, S., Millman, J., Eds.; pp. 51–56. Available online: https://www.researchgate.net/publication/265001241_Data_Structures_for_Statistical_Computing_in_Python (accessed on 20 March 2020)
32. Scikit-Learn. Available online: https://scikit-learn.org/stable/ (accessed on 20 March 2020).
33. Classification Loss Metric. Available online: https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics (accessed on 20 March 2020).
34. Available online: https://github.com/AntonioMarceddu/Facial_Expressions_Recognition_With_Keras (accessed on 20 March 2020).
35. Marceddu, A.C. Automatic Recognition And Classification Of Passengers' Emotions In Autonomous Driving Vehicles. Available online: https://webthesis.biblio.polito.it/12423/ (accessed on 20 March 2020).