BILINGUAL PUBLISHING GROUP

Journal of Computer Science Research

https://journals.bilpubgroup.com/index.php/jcsr

ARTICLE

Enhancing Human-Machine Interaction: Real-Time Emotion Recognition through Speech Analysis

Dominik Esteves de Andrade , Rüdiger Buchkremer*

Institute of IT Management and Digitization Research (IFID), FOM University of Applied Sciences, Dusseldorf, 40476, Germany

ABSTRACT

Humans, as intricate beings driven by a multitude of emotions, possess a remarkable ability to decipher and respond to socio-affective cues. However, many individuals and machines struggle to interpret such nuanced signals, including variations in tone of voice. This paper explores the potential of intelligent technologies to bridge this gap and improve the quality of conversations. In particular, the authors propose a real-time processing method that captures and evaluates emotions in speech, utilizing a terminal device like the Raspberry Pi computer. Furthermore, the authors provide an overview of the current research landscape surrounding speech emotional recognition and delve into our methodology, which involves analyzing audio files from renowned emotional speech databases. To aid incomprehension, the authors present visualizations of these audio files in situ, employing dB-scaled Mel spectrograms generated through TensorFlow and Matplotlib. The authors use a support vector machine kernel and a Convolutional Neural Network with transfer learning to classify emotions. Notably, the classification accuracies achieved are 70% and 77%, respectively, demonstrating the efficacy of our approach when executed on an edge device rather than relying on a server. The system can evaluate pure emotion in speech and provide corresponding visualizations to depict the speaker's emotional state in less than one second on a Raspberry Pi. These findings pave the way for more effective and emotionally intelligent human-machine interactions in various domains.

Keywords: Speech emotion recognition; Edge computing; Real-time computing; Raspberry Pi

*CORRESPONDING AUTHOR:

Rüdiger Buchkremer, Institute of IT Management and Digitization Research (IFID), FOM University of Applied Sciences, Dusseldorf, 40476, Germany; Email: ruediger.buchkremer@fom.de

ARTICLE INFO

Received: 7 June 2023 | Revised: 7 July 2023 | Accepted: 10 July 2023 | Published Online: 21 July 2023

DOI: https://doi.org/10.30564/jcsr.v5i3.5768

CITATION

Esteves de Andrade, D., Buchkremer, R., 2023. Enhancing Human-Machine Interaction: Real-Time Emotion Recognition through Speech Analysis. Journal of Computer Science Research. 5(3): 22-45. DOI: https://doi.org/10.30564/jcsr.v5i3.5768

1. Introduction

A phenomenon that transcends both professional and personal domains is the growing amalgamation of machines, which aims to foster human connection. Numerous individuals cannot decipher socio-affective cues, such as nuances in tone of voice. The challenge is to ensure that gestures, facial expressions, and paralinguistic information such as volume, frequency, and intonation that make up communication are not lost. Therefore, incorporating emotions into interface design becomes indispensable, as people tend to exhibit social behaviors during interactions with machines. Nonverbal communication often carries pivotal information in a typical conversation, revealing the speaker's intentions. Apart from the semantic content conveyed through text, how words are expressed imparts significant nonverbal cues. The precise delivery of spoken words, accompanied by appropriate emotions, can bring a special message altogether.

Consequently, el Ayadi et al. [1] elucidate that humanity remains distant from achieving natural interaction with machines, particularly in comprehending the emotional states of counterparts. At this juncture, the emergence of emotion recognition technologies becomes pivotal, encompassing various methods and technologies that enable the recognition of emotions beyond human perception. The primary objective of emotion recognition is to allow a system to adapt its response when certain emotions, such as frustration or anger, are detected. In 2001, Corvie et al. [2] expound on the two channels of communication present in every human interaction: the explicit and the implicit. While the explicit channel conveys messages, the implicit channel reveals the speaker's underlying feelings and moods. They explain that extensive research has been conducted to comprehend the explicit channel, whereas the implicit channel, though less explored, holds great significance in understanding speakers and their emotional states.

Machines typically exhibit neutral behavior, which humans may perceive as indifference. Hence, devices need to recognize emotions conveyed through speech to interact effectively. This ability is commonly referred to as Speech Emotional Recognition (SER). Schuller [3] states that even animals can perceive the tonality of human speech, suggesting that the time has come for machines to possess this capability. Kraus [4] asserts that distinguishing pure voice communication from visual or audiovisual communication is vital in determining a person's empathy.

In 2020, Akçay and Oğuz [5] commented that although real-time emotion recognition through SER systems is technically feasible, it has not yet become a ubiquitous part of daily life, unlike speech recognition systems. Implementing a computer system necessitates considering both economic and ecological factors. Energy supply and usage issues are intertwined with global warming and environmental concerns [6]. Thus, an edge emotion recognition machine must consume minimal energy while operating in real time.

To achieve near real-time operation, a machine requires a highly optimized processor performance, short transmission paths, and low latency times. These three conditions constitute essential components of edge computing. Unlike cloud computing, which transfers data to a centralized location for processing, edge computing brings computational power closer to the data. Mao et al. [7] argue that cloud computing is unsuitable for latency-critical mobile applications due to the distance between the user and the data center, resulting in significant delays. Abbas et al. [8] explain that cloud computing is unsuitable for real-time applications such as augmented reality or car-to-car communication and thus supports the edge computing approach. Cao et al. [9] report that over 50 billion end devices are connected to the Internet and thus to each other, producing a data volume of 40 zettabytes. It includes mobile and ambient end devices such as smartphones, smart speakers, or Raspberry Pis. For all these network participants to act and communicate with each other in near real-time, computing power must be shifted closer to the data.

2. Related work

The introduction of cloud computing was a milestone in the early 2000s, enabling new business models and innovations. However, the era of cloud computing seems to be ending as the edge computing paradigm is increasingly replacing the cloud computing paradigm due to new requirements. Edge computing can support the new requirements for low latency, increased data security, mobility support, and real-time processing. The literature divides edge computing into the sub-areas of fog computing, cloudlet, and mobile edge computing (MEC). While the first two approaches mentioned are hardly found in practice, MEC is ubiquitous. In MEC, computationally intensive cloud servers are stationed in mobile base stations at the network's edge and thus close to the end devices, ensuring daily use of this technology. As Shi et al. [10] stated, MEC means data processing immediately to the end device and on it. In addition to MEC, mobile cloud computing (MCC) is based on the principle that end devices perform the processing and only send the result or partial result to the MEC server or the MCC server. However, none of these approaches can be found in pure form in practice. Instead, cloud and edge computing techniques are combined to cover various use cases and exploit their advantages.

The topic of speech emotion recognition (SER) and its feature extraction and pattern recognition are a constant part of current research. Thus, the recent literature review shows that in SER, especially the continuous and the spectral features of speech are used since these reflect the characteristics of emotions most appropriately. Priority is given to the course of the primary speech frequency or loudness, the temporal ratios, pauses, and spectral features such as the Mel frequency cepstral coefficient (MFCC) and the Mel spectrograms [3]. The most common classification techniques used in speech recognition in recent years are the Gaussian Mixture Model (GMM) in combination with the Hidden Markov Model (HMM), the support vector machine (SVM) (Cortes and Vapnik 1995), and more recently, neural networks [11,12]. Consequently, the successes achieved in this regard also inspired using these techniques in SER, but with a focus on neural networks, SVM, or any combination of these two. Studies reveal that even pure emotion determination by humans is not accurate in all cases, so the focus is on the use and further development of neural networks [13]. In the field of neural networks, recurrent neural networks (RNNs) such as Long Short-Term Memory Hochreiter and Schmidhuber [14] were initially used because their feedback loops make them more suitable for processing continuous inputs such as speech signals [15,16]. RNNs have been superseded by convolutional neural networks (CNNs) such as AlexNet, VGG16, ResNet, or MobileNetV2 due to their high resource and memory requirements and continued success. Furthermore, MFCC or Mel spectrograms were launched using a Convolutional Neural Network (CNN). Moreover, the everyday use of transfer learning and Multitask Learning methods makes the CNN deployment even more efficient [17].

Every pattern recognition is based on previously extracted features in considerable quantity and quality. Due to this given diversity, selecting suitable parts is relevant in classification. The method generally used in machine learning for feature extraction is the use of the framework open-source Speech and Music Interpretation by Large-space Extraction (openSMILE) [18], which in turn includes the datasets extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) and ComParE. In deep learning, recent literature has increasingly used CNN for this purpose. In this approach, the output layer is either preserved as a classifier or replaced by, for example, an SVM.

In the phase of emotion classification, diverse sets of emotions diverge, which in turn harbor a different number of emotions. The settings can vary from five to 20 other emotions. The most common set of emotions in the literature refers to the six basic emotions, according to Ekman [19], which are happiness, sadness, anger, fear, disgust, and surprise, including a seventh neutral emotion.

In the mainstream literature, descriptions of the hardware on which a neural network is trained or executed are scarce. However, Tariq et al. [15] describe that neural networks—especially deep neural networks run in cloud-like data centers. The locally collected data is transferred to these servers, deleted on the local device, processed on the servers, and only the result is sent back to the end device. Thus, applying neural networks in the context of MEC and real-time capability represents a novelty. Despite the intensive research on these topics, no everyday emotion recognition products currently exist.

3. Materials and methods

We employ labeled emotional speech data for the prototypical implementation. Audio files with a minimum length of one second but a maximum length of 20 seconds are considered for use. Most emotion databases refer to six basic emotions [20]. Considering arousal and valence dimensions is not part of our work, which is why these criteria are neglected in data acquisition. Since part of this work is the emotion recognition in speech, but human speech is divided into sentences from which the emotions emerge, the audio length of one to 20 seconds is subjectively chosen since most sentences are spoken within this period. Thus, the audio files must still contain spoken sentences without singing, noise, or the like. However, the native language is not a selection criterion since emotions are expressed in any language. Even though the speaking gender is not an immediate selection criterion, the totality of all databases must contain both male and female spoken sentences to allow for the generalization of the data. In addition, the stored channel number or sampling rate is irrelevant in data acquisition, as these are standardized in the training process. Finally, the audio files and databases must be freely accessible and identified by labels. Thus, the following audio databases are placed that meet the given quality criteria:

1) Ryerson Audiovisual Database of Emotional Speech and Song (RAVDESS) [21]

2) Berlin Database of Emotional Speech (Emo-DB) [22]

3) Toronto Emotional Speech Set (TESS) [23]

4) EMOVO [24]

5) eNTERFACE'05 [25]

The eNTERFACE'05 database, in particular, holds data in an audiovisual format, whereas the remaining databases are in pure auditory waveform format (WAV). The audio part of the database eNTERFACE'05 is extracted from the audiovisual files to use the required audio data.

In **Table 1**, a detailed overview of databases is presented. While RAVDESS, eNTERFACE'05, TESS, and EMOVO reflect the six basic emotions, Emo-DB contains only five. Except for eNTERFACE'05, all other databases have a neutral emotion. Only RAVDESS and Emo-DB include any other emotions beyond these seven listed. However, since most databases contain the six primary and neutral emotions, the prototype will classify only those.

Hence, our dataset comprises 6656 audio files from 140 different references, amounting to a cumulative playback time of 5.14 hours. On average, each file has a duration of 2.97 seconds. Within the file names, emotions are encoded either as complete textual representations, abbreviations, or numerical values. The number of files and their total length per database vary considerably. **Figure 1** illustrates the distribution of emotions across all the acquired databases, demonstrating a relatively balanced allocation of emotion labels. While the neutral emotion category is slightly underrepresented, this discrepancy is compensated for during model training by appropriately adjusting the hyperparameters.

Moreover, the distribution of audio file durations per database is depicted in **Figure 2** using boxplots that exclude outliers. Notably, the eNTERFACE'05 database exhibits six outliers surpassing the upper threshold of 20 seconds. To preserve the readability of the boxplot representation, these outliers are omitted. However, it is worth mentioning that most files in the eNTERFACE'05 database have a maximum length of less than 20 seconds. Consequently, this database remains a foundational component for the prototype.

Table 1. Overview of the speech audio databases employed.

Database	Number of files	Min. length in sec.	Max. Length in sec.	Average length in sec.	Total length in minutes	Language	Emotions
RAVDESS	1440	2.94	5.27	3.7	88.82	English	Neutral, calm, joy, sadness, anger, fear, disgust, surprise
Emo-DB	535	1.23	8.98	2.78	24.79	German	Neutral, joy, sadness, anger, fear, disgust, boredom
TESS	2800	1.25	2.98	2.06	95.91	English	Neutral, joy, sadness, anger, fear, disgust, surprise
EMOVO	588	1.29	13.99	3.12	30.59	Italian	Neutral, joy, sadness, anger, fear, disgust, surprise
eNTERFACE'05	1293	1.12	106.92	3.17	68.37	English	Joy, sadness, anger, fear, disgust, surprise

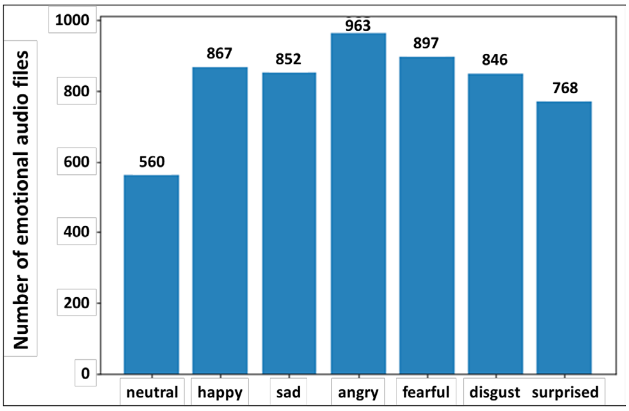


Figure 1. The overall distribution of the seven emotions across the databases was visualized with Matplotlib.

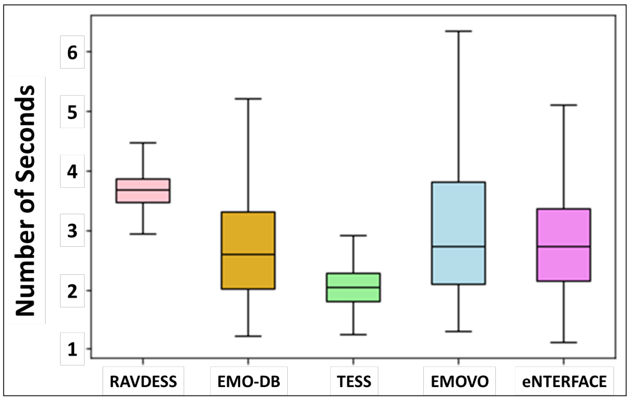


Figure 2. Boxplot representation of the databases used without outliers visualized with Matplotlib.

In recent studies, audio files with a sampling rate of 16000 hertz and mono tracks are primarily used [11,26-28]. As a result, the audio files of our data corpus are transformed to that uniform format.

A model is developed through algorithms that employ pattern recognition to classify recorded audio inputs into specific activities or emotions. Specifically, two distinct models are utilized—one using a machine learning algorithm, while the other utilizes a deep learning algorithm. It is important to note that, as indicated by Shinde and Shah [29], these algorithms are not synonymous; instead, the latter is a subtype of the former. A noteworthy distinction between the two lies in the requirement for specifying hyperparameters in the machine learning algorithm, whereas the deep learning algorithm automatically determines and optimizes these hyperparameters. Once trained, the model is deployed on ambient devices to assess the feasibility of implementing such an application using edge computing. The real-time capability of the prototype is determined by measuring its processing time.

The implementation of the prototypes in this research is carried out using Python [30]. As mentioned earlier, an essential aspect of Speech Emotion Recognition (SER) is the availability of suitable language data within the system. Consequently, implementing SER necessitates an initial filtering process that distinguishes between speech and non-speech audio. For this purpose, Hershey et al. [31] describe a neural network called YAMNet in their paper, which is specifically trained on audio classification using the AudioSet database [32]. YAMNet can distinguish between 521 audio classes from human, animal, machine, and natural sources. This publicly available Convolutional Neural Network (CNN) YAMNet serves as the upstream filter for SER and is utilized in both methods described in this work. However, since YAMNet does not impact the main SER algorithm developed in this research, further details regarding its functionality or structure are not provided here.

Distinct terminal configurations are employed during the creation and execution of the prototypes. The machine learning and deep learning models are trained on a Windows server. This step focuses on processing the five databases using the respective method, necessitating hardware utilization with appropriate performance capabilities. It should be noted that servers do not possess microphone inputs due to their general structure, broad physical localization, and clustering, which are also irrelevant for training purposes. **Table 2** lists the hardware components used in the training process.

Once the functional models are prepared, they are transferred to ambient end devices where real-time classification occurs. Typically, these end devices have lower computational power and memory than servers, rendering them unsuitable for training machine learning or deep learning models. However, these devices' internal processors and microphones are well-suited for executing such models. The performance achieved is contingent upon the specific hardware components of each device. **Table 3** presents the ambient terminals employed in this study and their respective specifications. The table encompasses two distinct types of devices chosen to represent each category.

The selection of end devices encompasses various categories, encompassing multiple operating systems, performance levels, and storage capacities. As a result, the chosen range serves as a representative cross-section of the available ambient end devices.

Due to these end devices' distinct architectures and operating systems, specific methods and requirements are necessary for utilizing the trained models. The fundamental prerequisites for deploying the ported models on the end devices include the frame-

Table 2. Server hardware used for model training.

Hardware component	Designation
Rack Server	HPE ProLiant DL380 Gen10
Operating system name	Microsoft Windows Server 2016 Standard
Processor	Intel® Xeon® Gold 6226 CPU @ 2.70GHz
Installed memory	256 GB
System type	64-bit operating system, x64-based processor

Table 3. Terminal devices and their components.

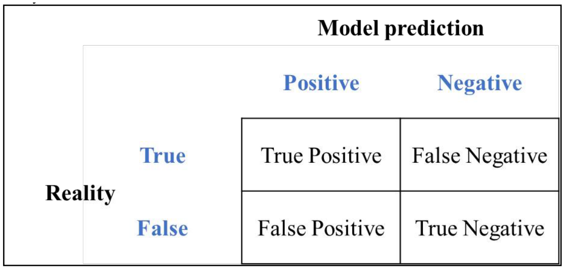
ID	Category	Terminal	Operating system	Processor	Working memory	Battery
1	Notebook	HP Envy x360 Convertible 15-cn0xxx	64-bit Windows 10 Home version 21H1	Intel® Core™ i7-8550U CPU @ 1.80 GHz	16 GB	3-cell, 52-Wh, 4.55-Ah, 11.55V, Li-ion battery
2	Raspberry PI	Raspberry Pi 4 Model B including an additional USB mini microphone	32-bit Raspbian GNU/Linux 11	Broadcom BCM2711 (Cortex-A72, ARM v8), 4-core CPU with 1.5 GHz	4 GB	External power supply

works openSMILE and TensorFlow/TensorFlow Lite alongside the Python programming language.

To measure and evaluate the performance of the prototypes, appropriate metrics are employed, focusing on real-time capability and classification success rate. Real-time capacity is assessed by measuring the response times of the prototypes in seconds. These measurements are conducted on the end devices, commencing immediately after the recording and storing of speech and concluding after classification. It is important to note that the model training and recording time are not considered during this evaluation. However, the exact time frame within which the measured response time must satisfy the criteria for real-time capability in machine processes is not explicitly defined in the literature.

In contrast, the ISO/IEC 2382: 2015 standard defines real-time as the "processing of data by a computer in connection with another process outside the computer according to time requirements imposed by the outside process" (ISO/IEC JTC 2015). Thus, it is apparent from this definition that specifying an exact time in seconds or milliseconds is not feasible. Instead, the external process defines the real-time capability, which may include human perception. Human perception is susceptible to linguistic communication, as pauses of a few milliseconds can be subjectively perceived as interruptions. Vogt et al. [33] suggest that subjective interruption is perceived after 1000 milliseconds. Zhang et al. [34] report that neural networks for image classification require a range of 15.2 to 184 milliseconds for processing, with input dimensions similar to the Deep Learning method utilized in this study (224 × 224 × 3). Furthermore, Liu et al. [35] state that compressed neural networks require only 103 to 189 milliseconds for processing on ambient devices such as smartphones. Consequently, in this prototyping without employing compressed methods, a measured duration of fewer than 1000 milliseconds is considered real-time.

Confusion matrices are commonly employed for representing and evaluating classification problems in machine learning. These matrices juxtapose the model's predictions with the actual states. **Figure 3** illustrates an example of a confusion matrix, depicting the four potential outcomes. The primary distinction lies in whether the model's prediction aligns with reality or deviates from it.



		Model prediction	
		Positive	**Negative**
Reality	**True**	True Positive	False Negative
	False	False Positive	True Negative

Figure 3. Example of a confusion matrix based on Davis and Goadrich [36].

The confusion matrix, also named the four-field matrix, does not represent a key figure in the narrower sense but provides the basis for its creation. Thus, the overall accuracy of the respective machine learning system is calculated from the confusion matrix and represented in decimal numbers, where the value 1.0 represents the maximum, and the value 0.0 is the minimum. The accuracy indicates the total number of correct predictions of the model and is determined using the following formula:

$$Accuracy = \frac{True\ Positives + True\ Negalives}{True\ Positives + False\ Pastitves + True\ Negatives + False\ Negatives}$$

For comparison, the previously mentioned CNNs are used, whereby the highest accuracy achieved in each case, as shown in **Table 4**, is deposited. The CNNs mentioned are sorted chronologically by publication date within the table. Besides MobileNetV2, the cited papers do not specify the machine used to generate results. Therefore, for the time being, it is assumed that the results of the CNNs were generated on cloud-like servers, similar to what is described by Tariq et al. (2019). Since a direct comparison of server-generated results with terminal device-generated results is not possible, the subsequent interpretation of the results of this study is limited.

Additional metrics are utilized for neural networks to measure and evaluate training results. These include training and validation accuracy and the duration of training and validation losses. Training and validation accuracy are represented as decimal values, ranging between 0.0 and 1.0, where

1.0 signifies the highest accuracy. Similarly, training and validation losses are expressed as decimal numbers, with no upper limit but a minimum value of 0.0 representing the optimal loss. Consequently, **Table 4** can also be applied to assess the accurate measurement of neural networks in this context. However, in evaluating CNNs, the focus is primarily on accuracy, rendering an evaluation or classification of training losses unnecessary. Accuracy measurement and the creation of confusion matrices occur immediately after training on the server, unlike the measure of classification time.

Table 4. Comparison of prediction accuracy of known CNN models.

CNN	Accuracy	Source
LeNet	82%	LeCun et al. 1998 [37]
AlexNet	84.6%	Krizhevsky et al. 2017 [38]
VGG	93.2%	Simonyan and Zisserman 2015 [39]
ResNet-152	96.43%	He et al. 2016 [40]
MobileNetV2	75.32%	Sandler et al. 2018 [41]

However, assessing the accuracy and elapsed time alone is insufficient to achieve the objectives. It is also essential to determine whether the described methods can be executed on ambient end devices and whether the results are comparable. While the training of the models does not occur on the end devices, the classification process does. To evaluate the performance of a general machine learning method on an end device, Liu et al. proposed criteria such as accuracy, delay, memory requirements, and power requirements. Accuracy is assessed using the confusion matrix and the resulting accuracy score, as mentioned earlier. As explained previously, the delay or temporal duration is determined by the model itself and is expressed in seconds, indicating the time required for one classification cycle. Memory requirements are measured in gigabytes, representing the average memory allocation needed for a cycle, calculated by comparing the working memory usage before and during classification. However, measuring the energy consumption of an application directly is not feasible since an application does not exclusively run on a single system. As a result, the strict identification of the energy demand for a specific application is challenging. Energy consumption can be estimated indirectly by measuring processor utilization. Thus, the difference in measured processor utilization, represented as a percentage, before and during classification is utilized as a metric in this context.

The overall accuracy metric is related to the model and, therefore, independent of the hardware utilized. However, metrics such as time, memory consumption, energy consumption, and processor utilization are hardware-dependent. The metrics mentioned above are evaluated using the hardware listed in **Table 3** in the subsequent analysis.

4. Results

To implement Speech Emotion Recognition (SER), an upstream filter is required to distinguish between speech and non-speech. Hershey et al. [31] introduced a neural network called YAMNet, trained on the AudioSet database [32], which classifies audio into 521 different audio classes, including human, animal, machine, and natural sounds. YAMNet, a freely available Convolutional Neural Network (CNN), serves as the upstream filter for SER in this work without affecting the main SER algorithm. As YAMNet's functionality and structure have been described elsewhere, further details are not provided in this study.

The traditional machine learning algorithm employed in this research follows a supervised learning approach using a data corpus of the five mentioned databases. The objective is to generate a model that can be ported to an end device for classification purposes. The openSMILE framework is utilized for feature extraction in this method, while a Support Vector Machine (SVM) is employed as the classifier. The SVM is trained on the eGeMAPS features extracted from the audio files using openSMILE. The extraction and training processes are applied to the entire data corpus rather than individual databases. After extraction, the parameter dataset is normalized by removing the mean and scaling it to unit variance. Subsequently, the normalized dataset is divided into training and test partitions in an 80:20 ratio. The training partition is used for model training, while the test partition validates the training results.

The relevant hyperparameters for training are optimized and determined by the algorithm itself. Initially, four hyperparameters with specified value ranges are provided. These include the selection of available SVM kernels (polynomial, linear, sigmoidal, and radial basis function), a regulation parameter ranging from 10^{-3} to 10^{2}, and a degree parameter ranging from zero to nine for the polynomial kernel. The algorithm optimizes and applies various combinations of hyperparameters during training on the training partition. Following the training phase, validation is performed using the training dataset.

Upon completion of training, the machine learning system can classify new, unknown data based on the learned generalization. The system is connected to a microphone, which records human speech at 1024 frames per buffer every three seconds. The recorded audio is stored locally in a 16-bit WAV format with a sampling rate of 16000 Hz and a mono channel. The stored file is then read by the machine learning system and processed using YAMNet. If the classification result from YAMNet indicates "human speech", the file is further processed using openSMILE to obtain eGeMAPS features. Similar to the training phase, this dataset is normalized and passed to the SVM for emotion classification. The classification is performed immediately, and the process continues in a continuous three-second cycle, processing the following files until manually terminated.

Figure 4 provides a schematic overview of the mentioned processing steps, indicating the sequence of individual actions. Not all processing steps are executed on the same hardware, and the figure specifies which steps are performed on the server and the end device.

The deep learning algorithm is based on the same data corpus to ensure a subsequent parity comparison of both approaches. The goal is to generate an executable model for subsequent porting to the end devices. As an alternative to machine learning, CNN acts as a feature extractor and classifier. In this context, the creation and training of the CNN are based on TensorFlow [42]. Since a CNN expects image files instead of audio files as input, it is first required to generate corresponding representative spectrograms from audio data.

Input for the CNNs is Mel spectrograms derived from the spectrogram audio representations. Therefore, speech recordings of different lengths also result in spectrograms of various sizes. However, since it is necessary to always use identically sized spectrograms for training the CNN, the audio files must be read in and processed with a fixed window. To ensure a subsequent comparison, the first three seconds of the audio files are read in, of which only two seconds are processed with an offset of half a

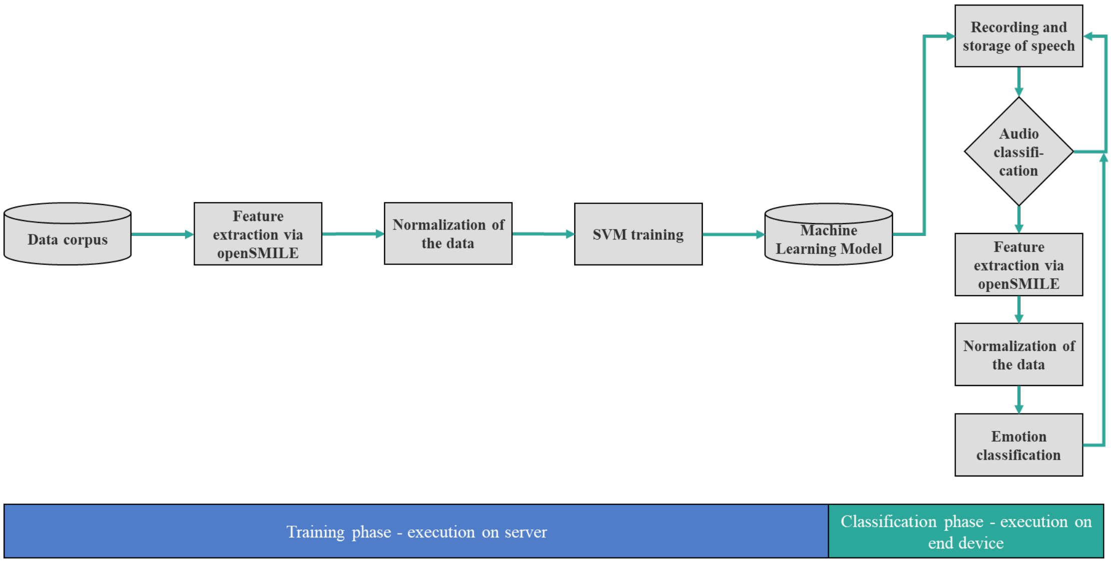

Recording and storage of speech
Audio classifi-cation
Data corpus
Feature extraction via openSMILE
Normalization of the data
SVM training
Machine Learning Model
Feature extraction via openSMILE
Normalization of the data
Emotion classification
Training phase - execution on server
Classification phase - execution on end device

Figure 4. Schematic representation of the processing steps of the machine learning method.

second. If an audio file is smaller than three seconds, the content of the file is duplicated until the minimum size is reached. To generate the spectrograms, the final two-second audio file is transformed using Fast Fourier Transform with a window size of 512 milliseconds and a jump size of 256 milliseconds between windows. From this spectrogram, the Mel spectrogram is derived with 128 Mel filters, a minimum frequency of 0 Hertz, and a maximum frequency of 8000 Hertz. Finally, this Mel spectrogram is plotted on a dB scale with 80 dB and the magma color scheme and is available for subsequent classification. The generated dB-scaled Mel spectrogram, including its intermediate stages, is visually presented in **Figure 5**. This way, the entire data corpus is preprocessed and then split again into a training and a test data set in a ratio of 80 to 20.

They must be normalized before the Mel spectrograms can serve as input data to the CNN. In this method, normalization consists of importing the image files with fixed dimensions of 224 × 224 × 3 pixels and then dividing each pixel value by a factor of 255. The dimensions of 224 × 224 × 3 pixels have been proven in image recognition by CNN since AlexNet, which is why they are also used here. The division by 255 is necessary because neural networks are known to operate from zero to one, and thus the pixel values are normalized.

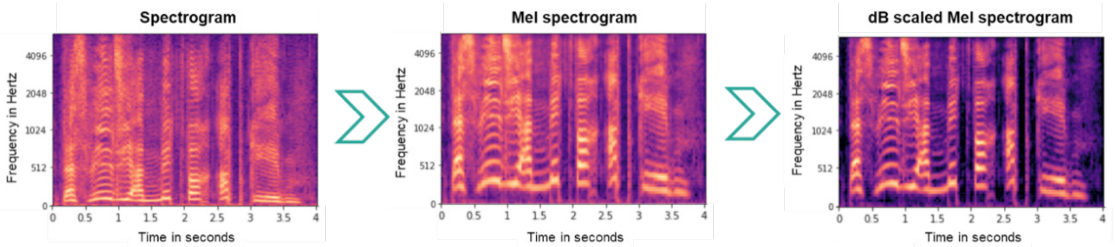


Figure 5. Mel spectrogram generation in individual steps visualized with TensorFlow and Matplotlib.

Training a neural network from scratch is computationally intensive, time-consuming, and involves significant data, so transfer learning is used now. Transfer learning for neural networks consists of removing the output layer of a pre-trained neural network and replacing it with new output layers of its own, which act as classifiers. MobileNetV2, which is pre-trained on ImageNet[①], is this method's base model for transfer learning. MobileNetV2, the training base, was initially designed for object recognition and execution on mobile devices. Three separate output layers then augment the base model with a GlobalAveragePooling2D, a dropout of 0.2, and a fully connected layer including a softmax activation function, which is used when the number of classes is more significant than two. The neural network is then optimized using the Adam optimization algorithm [43] and an initial learning rate dropout of 10^{-5}.

Furthermore, categorical cross-entropy is used as a loss function, which is used to quantify the differences between two probability distributions in prediction. Finally, the model is trained in two phases of equal size to apply transfer learning. First, the training of the new model operates with 50 initial epochs on the 154 untrainable layers and weights, which is used to transfer the experience of the base model to the task. Only the three newly added layers are trainable in this phase. Subsequently, the model training operates another time with 50 epochs, this time with 54 trainable and 100 untrainable layers, which is called fine-tuning in the corresponding literature. Each epoch is run with 100 training steps and ten validation steps. The training and validation data are read into the model training with a batch size of 16.

Figure 6 schematically visualizes the described sequence of the deep learning process. In addition to the individual processing steps and their arrangement, it is also apparent here which processing steps are executed on the server or the end device. The

[①] ImageNet is an image database consisting of over 15 million labeled and high resolution natural images with approximately 22000 categories.

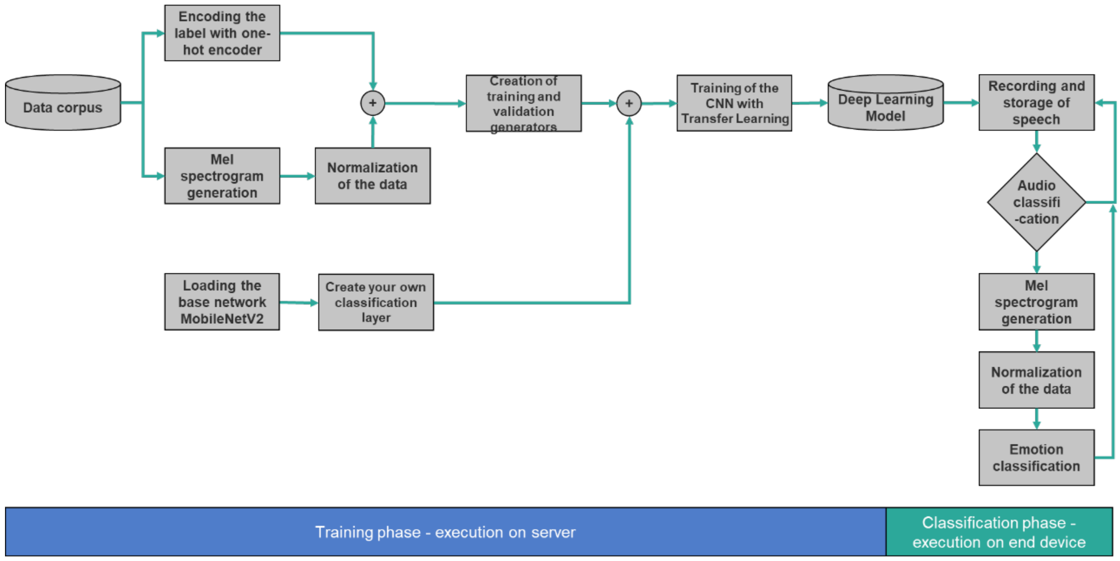

Data corpus
Encoding the label with one-hot encoder
Mel spectrogram generation
Normalization of the data
+
Creation of training and validation generators
+
Loading the base network MobileNetV2
Create your own classification layer
Training of the CNN with Transfer Learning
Deep Learning Model
Recording and storage of speech
Audio classifi-cation
Mel spectrogram generation
Normalization of the data
Emotion classification
Training phase - execution on server
Classification phase - execution on end device

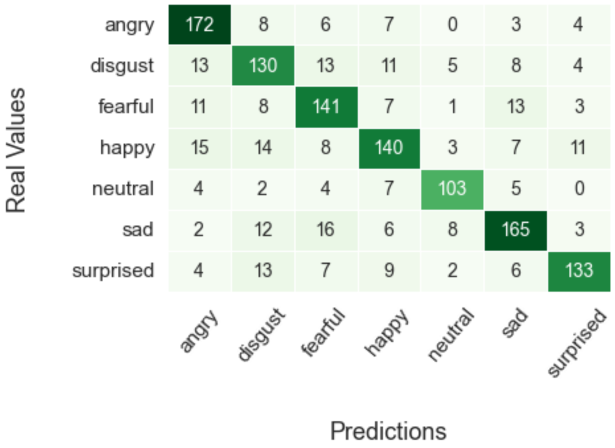
Figure 6. Processing steps of the deep learning method.

similarities and differences between this graphic and the similarities and differences in **Figure 4** become apparent. The diagrams outline the appropriate processing steps, sequence, and execution location. Furthermore, the optimization of the hyperparameters and the general parameterization of the models are part of the training and are, therefore, not listed in both diagrams. Furthermore, it can be seen from the comparison that additional work steps are necessary for the deep learning method before the neural network training is started. The effects of the extra steps on the result will be discussed later.

While describing the results of both prototypes, a distinction is made between the generation of the executable model, including its training, and the real-time classification by the same model.

The prototype is a supervised machine learning method using a support vector machine as a classifier for emotion determination. The algorithm is trained on the five databases to generate an executable and portable model. The training of this algorithm, including the optimization of the hyperparameters, is about 96 hours. The hyperparameters selected and optimized by the algorithm are the radial basis function kernel, the regulation parameter with a value of 101, the gamma with 10^{-2}, and the degree parameter with a zero value.

The accuracy after the model training is indicated by the confusion matrix, shown in **Figure 7**, for the test partition of the trained model and the classification report based on it. In the former, the list of the seven considered emotions can be lined up vertically on the left edge as absolute values on the one hand and horizontally on the bottom edge as values predicted by the model on the other. Furthermore, it can be seen from the marked green fields that the model's prediction agrees with the actual values in most cases. Those correct predictions represent the true positives. The remaining whitish areas represent the False Positives since the predicted emotion classes do not match the real-world conditions.

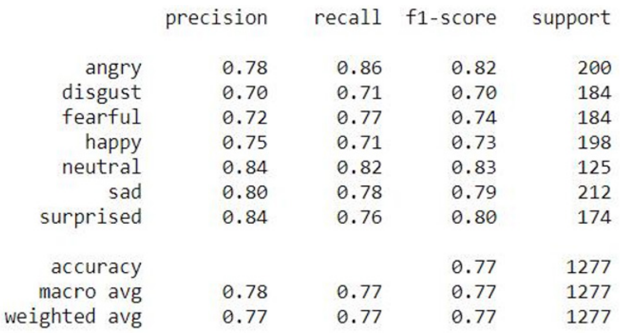


	angry	disgust	fearful	happy	neutral	sad	surprised
angry	172	8	6	7	0	3	4
disgust	13	130	13	11	5	8	4
fearful	11	8	141	7	1	13	3
happy	15	14	8	140	3	7	11
neutral	4	2	4	7	103	5	0
sad	2	12	16	6	8	165	3
surprised	4	13	7	9	2	6	133

Figure 7. Confusion matrix of the machine learning process.

The confusion matrix calculates the model's overall accuracy using the above formula, which is already included in the classification report and visualized in **Figure 8**, together with other metrics. For example, in addition to the overall accuracy rates, accuracy rates for individual emotions are also present. **Figure 8** shows that the overall accuracy of this procedure is 0.77. With an achieved value of 0.77 and 77%, respectively, the model is ranked between the CNN MobileNetV2 and LeNet based on **Table 4**.

The confusion matrix and the classification report are valid for the machine learning model and thus independent of the end device used.



```
              precision    recall  f1-score   support

       angry       0.78      0.86      0.82       200
     disgust       0.70      0.71      0.70       184
     fearful       0.72      0.77      0.74       184
       happy       0.75      0.71      0.73       198
     neutral       0.84      0.82      0.83       125
         sad       0.80      0.78      0.79       212
   surprised       0.84      0.76      0.80       174

    accuracy                           0.77      1277
   macro avg       0.78      0.77      0.77      1277
weighted avg       0.77      0.77      0.77      1277
```

Figure 8. Classification report of the machine learning method.

An exemplary metrics measurement is performed on the notebook mentioned in **Table 3**. The estimated time is measured within the model between two cycles. A cycle consists of a speech recording, an audio classification, and an emotion classification, depending on this result and its output. The average estimated time for 15 observed cycles is 0.799 seconds. For comparison purposes, processes without audio classification are also performed, where emotion classification is applied to each incoming audio signal. The average time required here is 0.114 seconds, calculated from 15 observed cycles.

In conclusion, based on the criteria set, 144 milliseconds for emotion classification alone and 799 milliseconds for emotion classification, including previous audio classification, are declared to be real-time capable. The memory requirement increases from 9.8 gigabytes to 10.1 gigabytes after starting the classification, which is derivatively an increase from 62% to 64% utilization. On the other hand, the processor utilization increases by 17% points after starting the classification, from an average of 11% to around 28%.

Furthermore, measurement is also performed on the Raspberry Pi. Here, the arithmetic means of 15 observed cycles is 4.33 seconds for audio and emotion classification. Also, at this point, the result is compared with a pure emotion classification without prior audio classification. This cycle takes an average of 0.337 seconds, calculated from 15 observed cycles. In conclusion, based on the set benchmarks, the emotion-only classification is declared real-time capable, but the combined audio and emotion classification with a time of 4330 milliseconds is not. Further memory measurement indicates an increase in memory usage after starting classification from 415 megabytes to 586 megabytes, a relative increase of 4.4% for a total availability of 3838 megabytes, from 10.8% utilization for the first time to 15.2% utilization now. On the other hand, processor utilization increases by 26.4% points during execution, from 0.7% utilization for the first time to 27.1%.

The second method described in 4.3, a CNN, is used based on the pre-trained MobileNetV2 network. The CNN developed in this method is also trained with the same intention on the five databases. The training time of the neural network with a total of 100 epochs is about six hours. As described above, the training proceeds in two identical phases of 50 epochs each, one step for initial learning and one stage for finetuning the model. Following each training epoch, the training and validation accuracy and the training and validation loss are reported. The preliminary result after the first 50 epochs is graphically visualized in **Figure 9**. On the left side is the training and validation accuracy course. The training and validation loss, each for 50 epochs, is shown on the right side. Each of these four parameters is shown as a separate curve.

The accuracy curve shows that the training accuracy starts at around 0.16 and increases to approximately 0.42 by the 50th epoch. The validation accuracy also starts at about 0.16 and reaches an accuracy of about 0.5 at the 50th epoch. It is noticeable here that the validation accuracy is always above the training accuracy. This phenomenon is due to the peculiarity of transfer learning. When training a neural

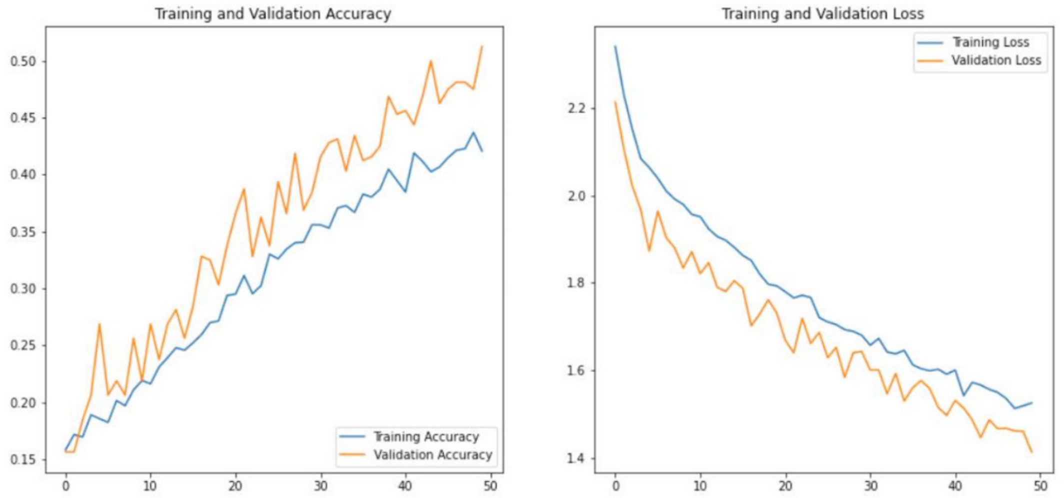


Figure 9. Training result of the CNN after initial 50 epochs with transfer learning.

network without transfer learning, the training accuracy is always above the validation accuracy.

Similar behavior can be observed in the course of the loss curve. The training loss curve starts at a loss of around 2.3 and drops to about 1.55 by completing the 50th epoch. On the other hand, the validation loss curve begins at 2.2 and drops to about 1.4 by the 50th epoch. Once again, it is characteristic of transfer learning that the validation curve always lies below the training curve.

During the subsequent fine-tuning of the CNN, more trainable layers and, thus, more trainable weights are available. The model also has more possibilities to optimize performance. The result of the fine-tuning is illustrated in **Figure 10**.

Fifty other fine-tuning epochs supplement the result in the previous **Figure 9**. A vertical straight line in the 50th epoch shows at which point the fine-tuning phase starts. Thus, after the beginning of the fine-tuning stage, the training accuracy curve drops to 0.35 but then takes a steeper course than before and reaches the maximum of 1.0 from the 90th epoch, at which point the curve stagnates until the 100th epoch. On the other hand, the validation accuracy curve initially drops to around 0.45 after the start of the finetuning phase. Still, it rises again and reaches an accuracy rate of about 0.7 by the 100th epoch.

A change can also be observed in the loss curves after the start of fine-tuning. For example, the train-

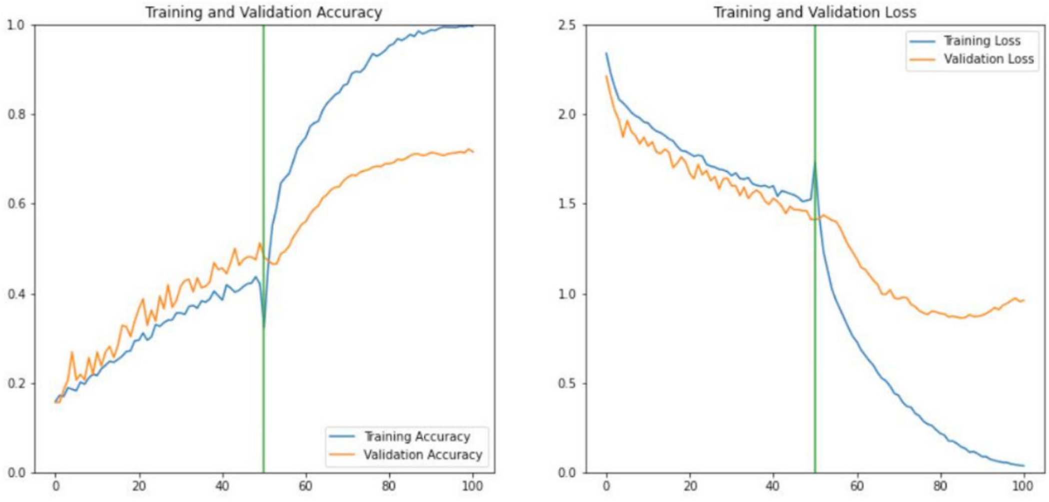


Figure 10. Training result of the CNN after 100 epochs using transfer learning.

ing loss curve rises to 1.75 after the start of fine-tuning but then drops more sharply than before, reaching a value of 0.1 at the 100th epoch. The validation loss curve does not rise after the start of fine-tuning but drops to a value of 0.8 by the 80th epoch, where the local minimum of the curve is located. By completing the 100th epoch, the curve rises to about 1.0. Ultimately, it is not the training accuracy but the validation accuracy that is decisive for the correct classification. With an accuracy of 0.7 and 70 %, respectively, this result is based on **Table 4** ranks below MobileNetV2.

Analogously, the CNN of this method is trained a second time on the five databases but without applying transfer learning. In this training, the CNN is also qualified with 100 epochs, but in only one phase and with the absolute number of trainable layers. With its 100 epochs, this training has a running time of about six hours, as before. The result of this training of 100 epochs without transfer learning is visualized in **Figure 11**. Here, it can be seen that the training accuracy curve starts at a value of 0.22 and rises to the maximum of 1.0 by the 50th epoch. There the curve remains until the 100th epoch. The validation accuracy curve also starts at a value of 0.22 and increases to 0.7 by the 50th epoch, which remains with fluctuations until the 100th epoch. The training loss curve begins at the value of 2.0 and steadily decreases until it reaches the minimum of 0.0 at about the 70th epoch and remains there.

On the other hand, the validation loss curve starts with a value of about 2.2 and falls with a fluctuating downward trend until the 35th epoch. There, the curve has reached its local minimum of 0.9. However, the curve rises again until the 100th epoch to about 1.2.

The advantages of transfer learning become apparent when comparing **Figure 10** with **Figure 11**, and the benefits of transfer learning become evident. Starting from the start of fine-tuning, it can be stated that the validation accuracy curve has already reached the value of 0.7 after 20 epochs. In contrast, the validation accuracy curve without transfer learning has only reached this value after about 50 epochs. The advantage can also be seen in that the validation accuracy curve for the method with transfer learning has a higher slope than the validation accuracy curve without transfer learning. Finally, it can be seen that the start of the validation accuracy curve for the process with transfer learning starts higher on the Y axis with a value of 0.45 than the curve without transfer learning with a value of 0.22.

In contrast to an SVM, the classification result in a neural network does not output a single value but a value range with seven entries corresponding to the number of classes present. The entries in this value range represent the probabilities with which the model predicts one class each. The individual entries

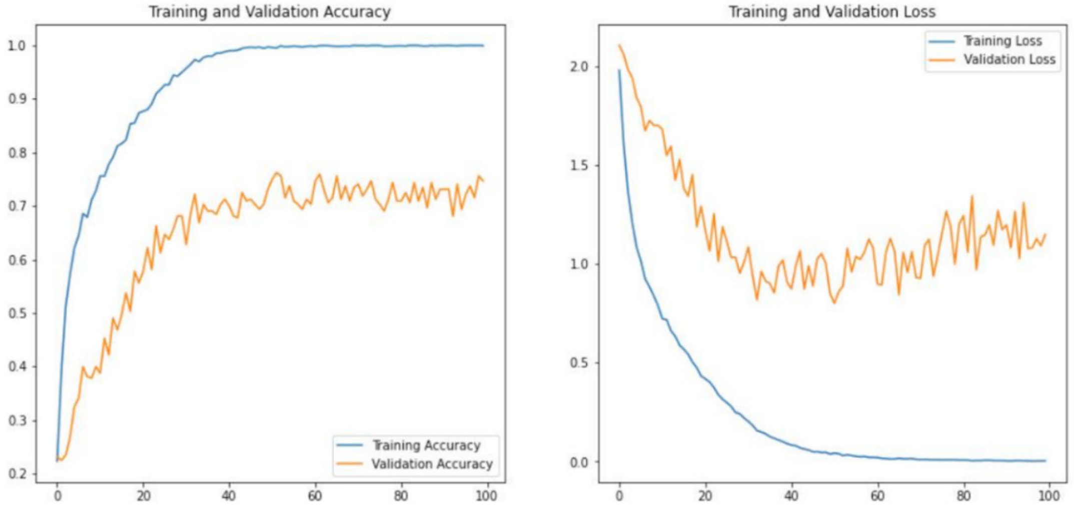


Figure 11. Training result of the CNN after 100 epochs without using transfer learning.

can assume a value between 0.0 and 1.0, with the sum of all entries in the value range again resulting in 1.0. The emotion class with the highest probability value is the classified emotion.

An exemplary metrics measurement is also performed on the notebook mentioned in **Table 3**. Here, the time is also measured between two cycles, whereby one consists of the audio and emotion classification, including the output. The arithmetic mean of the estimated time is 0.856 seconds for 15 observed processes. At this point, a comparison is also made to a cycle without prior audio classification. The time counted for this cycle is 0.119 seconds, also calculated from 15 observed cycles. With a time value of 119 milliseconds for a cycle without audio classification and 856 milliseconds for a cycle with audio classification, respectively, the result is below the set benchmarks and is therefore considered real-time capable. The memory requirement increases from 9.9 gigabytes to 10.6 gigabytes from the start of classification. Relative to the total available memory, this is an increase of 4% points, from 63% utilization for the first time to 67%. The processor utilization also shows an increase of 16% points, from 15% to 31% for the first time.

Based on the implementation of the prototype on the Raspberry Pi, the average time required for a cycle, including audio and emotion classification, is around 4.43 seconds, calculated from 15 observed cycles. In comparison, emotion recognition without prior audio classification requires an arithmetic mean of only 0.393 seconds, again calculated from 15 practical cycles. With a needed time of 393 milliseconds, the latter result is below the set benchmarks, but the previous mark with 4427 milliseconds is not. Thus, implementing the prototype on the Raspberry Pi lacks real-time capability. The memory used increases from 563 megabytes to 675 megabytes during runtime, a rise of 2.9% points for a total availability of 3838 megabytes, from 14.7% utilization for the first time to 17.6% now. On the other hand, processor utilization increases by 22% points during execution, from 2.9% utilization for the first time to 24.9%.

In this study, four core elements are to be noted as findings. First, a tabular comparison of the results of the two methods used is provided in **Table 5**, where the metrics listed here represent the arithmetic mean across all measured metrics.

It can be deduced from the previous table that an SER system can distinguish between speech, non-speech, and silence. To this end, the YAMNet neural network, which is not a primary component of this work and was not developed within this research, is used within the prototypes. Nevertheless, the YAMNet neural network is part of both prototypes, which are thus able to classify audio inputs into different categories, such as music, meowing, barking, silence, or even speech.

Concerning the databases used in this research, it was shown that they contain various emotional audio files, including the six basic emotions mentioned by Ekman (1971), plus further emotional stimuli such as tiredness or boredom. Neutral emotion can also be found in the majority of the databases. The prototypes trained on these databases are thus able to distinguish between the seven emotions. Therefore, an SER system can distinguish between positive, nega-

Table 5. Comparison of the results of the two methods.

Metrics	Machine learning method	Deep learning method
Training duration	96 hours	6 hours
Accuracy	77%	70%
Working memory requirement increase	10.7% points	3.45% points
Processor load increase	21.7% points	19% points
Time consumption emotion classification	225.5 milliseconds	256 milliseconds
Time consumption audio and emotion classification	2565 milliseconds	2642 milliseconds

tive, and neutral emotions but is not limited to this. Instead, such a system can perform a more detailed categorization of speech input into individual emotions with an accuracy of 77% for machine learning and 70% for deep learning.

The third finding relates to the feasibility of an SER system on ambient terminals but is distinguished between the phases and the ambient end devices used in **Table 3**. Due to the intensive computing power and high runtime, the one-time model training step must be executed on a server. Therefore, therefore not feasible on an end device. The subsequent real-time classification phase is based on the trained model and can be performed multiple times on a terminal device. The prototype porting to a notebook is feasible since notebooks generally support corresponding Python runtime environments. Thus, running the emotion classification is possible on a notebook regardless of the method used. Porting the prototypes to a Raspberry Pi, on the other hand, is more complex since, on the one hand, Python runtime environments are supported in principle. Still, on the other hand, the necessary frameworks, openSMILE and TensorFlow, are not available for Raspberry Pi's. Alternatively, for TensorFlow, the porting of the Deep Learning procedure is done with TensorFlow Lite, which runs the trained model on the end device. In the absence of openSMILE compatibility with 32-bit operating systems and a lack of a qualitative alternative, the porting of the machine learning procedure is omitted at this point. In summary, it can be stated that realizing an SER system using edge computing is only possible to a limited extent. While they assist in executing deep learning approaches and neural networks on the end devices, this does not always apply to the machine learning method.

Regarding the real-time capability of the classification system, it is necessary to differentiate which method is used and whether only SER or SER plus prior audio classification is considered. Concerning the machine learning method, the SER system requires an average of 114 milliseconds for pure SER without prior audio classification and is thus below the set index of 1000 milliseconds. Additionally, the SER, including initial audio classification with an average time of 799 milliseconds, is below the set of 1000 milliseconds. Related to Deep Learning, the average time for a cycle without an audio classification is 256 milliseconds. Meanwhile, the average time for a cycle with audio and emotion classification is 2642 milliseconds. According to the results, the fourth finding is that the choice of the method determines whether the real-time capability is given or not. However, since there is no porting and, therefore, no results regarding the machine learning method, there is the possibility that this last finding is falsified.

5. Discussion

Both machine learning and deep learning approaches in this study rely on a shared data corpus, which is obtained and selected based on predefined criteria outlined in the existing literature. These criteria encompass several factors, including a minimum audio duration of one second and a maximum duration of 20 seconds. The choice of one second as the minimum duration is justified because shorter audio files generally lack sufficient information. Conversely, selecting a maximum period of 20 seconds is somewhat arbitrary, as durations of 10, 15, or 30 seconds could have also been considered. However, as depicted in **Figure 2**, most audio files in the chosen databases have durations of less than 10 seconds.

Another criterion is the exclusion of non-spoken sentences since the prototypes focus on speech-emotion recognition (SER) rather than general audio classification. Therefore, the audio files must exclusively contain speech, even though some music may include spoken segments accompanied by instruments.

Furthermore, it is essential to note that this study does not address emotion recognition in music, although it could be a potential avenue for future research. Including audio files with background noise is essential, as real-life communication often occurs in noisy environments. While background noises are prominent in music, they play a secondary role in speech-related scenarios. Therefore, incorporating databases containing audio data with such background noise would be valuable for enhancing the research.

The criteria specify that the files must be in an auditory or audiovisual format, but there are no restrictions regarding file type, sampling rate, or dubbing. Although limiting the selection to purely auditive file formats may influence the choice of databases, it would not impact the subsequent procedures, as all audio files are transformed to a standardized format and file type before model training.

The native language used in the audio files is also not a selection criterion, as indicated in **Table 1**, which demonstrates the inclusion of German, English, and Italian data in the selected databases and audio files in other languages such as Turkish, Danish, or Chinese. Since the six basic emotions described by Darwin (1873) and Ekman (1971) are expressed similarly across cultures, the spoken language does not significantly affect the Mel spectrograms, model training, or results. However, it is crucial to include both male and female voices when selecting databases. Failing to meet this criterion could impede data generalization and lead to overfitting or underfitting of the model.

Open accessibility and availability of labeled data are mandatory for data collection. Without open access to the databases, it would be impossible for third parties to reproduce the procedures and results of this study. Moreover, the absence of labeled data would render supervised machine-learning algorithms infeasible. Investigating the impact of different database quantities or compositions, including language variations, on the outcomes of this research can be pursued in future investigations.

Other methodologies that equally impact both procedures involve dividing the data corpus into training and validation sets using an 80 to 20 ratio. Preprocessing the audio files commonly entails transforming them to a 16,000-hertz format with a mono channel, as frequently reported in the literature.

In the machine learning method, the hyperparameters and their value ranges are defined at the start of the training process. The algorithm then independently determines the optimal values based on these predefined hyperparameters. The selection of these hyperparameters is based on the research conducted by Mao et al. [27]. However, alternative parameters or value ranges described by Wang et al. [44] or Cummins et al. [45] could also be considered. Feature extraction utilizes the openSMILE framework, particularly the eGeMAPS, which aligns with its usage in Cummins et al.'s work.

In contrast, the deep learning method employs explicit hyperparameters. The training process consists of two phases, each comprising 50 epochs, as established by Tan et al.. Alternatively, Zhang et al. utilized a batch size of 30, SGD as the optimization algorithm, and a learning rate of 10-3 as hyperparameters. However, standard hyperparameters employed by Lim et al. include SGD with a learning rate of 10-2, a dropout of 0.25, and a Rectified Linear Unit activation function. Discrepancies also exist in the generation of Mel spectrograms, as mentioned in section 2.3.3 and the relevant literature. For instance, Zhang et al. used 64 Mel filters for audio classification within a frequency range of 20 to 8000 hertz, utilizing a 25-millisecond Hamming window with a ten-millisecond overlap for each window. The variation in Mel spectrogram generation can be justified since speech emotion recognition (SER) and speech recognition are distinct processes, as noted by Zhang et al. Nonetheless, the use of Mel spectrograms aligns with the current state of research. Alternatively, MFCC can also be applied within the deep learning procedure.

The base model utilized in this study is MobileNetV2 when employing transfer learning. However, the literature suggests considering CNN ResNet50 or SqueezeNet [46]. In this study, only the last 54 layers out of 154 are fine-tuned. Optionally, different numbers of trainable layers, such as 32 or 16, can be considered. Exploring the impact of modifications to these hyperparameters on the results can be a subject of future research.

The other cannot be drawn solely from comparing the approaches and their results. **Table 5** does not provide conclusive evidence to support the dominance of either procedure. Examining the training time reveals that the Machine Learning approach requires approximately 16 times the duration compared to the Deep Learning approach. However, the Machine Learning model exhibits higher accuracy, faster classification, and lower increases in processor load and memory requirements.

The speed advantage of the machine learning method in real-time classification arises from the utilization of distinct emotion recognition algorithms. In this case, audio classification is not considered, as it is identical in both approaches and precedes emotion recognition. In the Machine Learning model, speech input undergoes openSMILE processing, normalization, and subsequent classification using SVM. Conversely, in the Deep Learning model, the speech input is initially transformed into a spectrogram, stored, normalized, and processed through all neural network layers. The storage and retrieval of spectrograms involve additional read-and-write transactions that are not required in the machine learning method, thereby impacting the speed of the Deep Learning model. However, the difference in speed is marginal and invisible to humans, as such disparities occur in milliseconds.

Furthermore, **Table 5** highlights that the pure emotion classification alone operates, on average, ten times faster than the combined audio and emotion classification, regardless of the chosen method. This discrepancy significantly influences the declaration of real-time capability, particularly concerning porting to the Raspberry Pi. While pure emotion classification can be deemed real-time capable, the same cannot be said for the combined type due to extended runtime. The disparity may likely stem from the CNN YAMNet employed for audio classification and its external development, which falls outside the scope of this study. Consequently, a comprehensive analysis of the time difference and its origin cannot be provided. Therefore, optimizing speech classification, which is not extensively examined in this paper, could considerably enhance the overall process latency.

When examining the individual process steps in **Figures 4 and 6**, no direct conclusion regarding training duration is apparent. However, the Deep Learning method necessitates more process steps than the Machine Learning method. As mentioned earlier, the models' parameterization is not depicted in these figures, as it is part of the mapped training. Specifically, the choice between fixed hyperparameters and ranges of hyperparameter values significantly impacts training time. In the Deep Learning method, training duration also varies based on parameters such as batch size, number of epochs, and number of steps per epoch. The Machine Learning method determines training duration by the number of hyperparameters and their value ranges. The resulting hyperparameters are determined through the algorithm's processing and optimization of potential combinations. In contrast to explicit parameterization, processing all conceivable combinations is likely responsible for the disparity in training duration. Consequently, this implies that the overall accuracy of the machine learning model surpasses that of the deep learning model due to the processing of all combinations.

Moreover, the results in **Figure 10** reveal that the validation loss curve increases again after epoch 80. A similar phenomenon is observed in **Figure 11** from epoch 50 onwards. However, the validation accuracies in these figures do not exhibit the same increase. The rising course of these curves may indicate overfitting, which warrants further investigation in future research.

The higher memory requirement in the Deep Learning model is attributed to the necessity of storing the generated spectrogram, in addition to the primary audio file, for emotion recognition. Another contributing factor is the higher number of parameters in the CNN than in an SVM, which are also stored in memory.

Contrary to our expectations, the Machine Learning method exhibits higher processor utilization than the Deep Learning method. This phenomenon is likely due to the improper timing of the recording. However, considering the marginal variances, the

21.7% difference in processor utilization between the two ways is negligible.

It should be noted that both models consider the presence of a microphone as an essential prerequisite. Unlike multimodal emotion recognition, a unimodal SER system does not require additional provisions such as cameras. Most ambient terminals are equipped with native microphones but lack native cameras, as seen in smart speakers or smart TVs. Therefore, the prototypes developed in this study are suitable for porting to such devices.

Regarding the theoretical foundations of this research, SER plays an increasingly crucial role in human-computer interaction (HCI). HCI occurs within the context of remote participation, which is a component of the growing computer-supported hybrid communication in everyday life. Consequently, SER holds greater significance in everyday life and is the subject of ongoing research. Similar to this study, there are investigations into real-time SER [33] and edge computing. However, no research on SER applications on edge devices exists, as Shi et al. (2016) defined. Thus, the combination of SER, edge computing, and real-time processing explored in this study represents a novel research extension. To maintain the focus of this work, restrictions are deliberately made. However, other external limitations also limit this work, which will be explained in more detail below. According to Ekman (1971), only the six basic emotions, including a seventh neutral emotion, are considered, which is why emotions such as tiredness or boredom are excluded in this work. Accordingly, the data acquisition is made with the mentioned seven emotions, further limiting the selection of suitable databases.

Furthermore, the dimensions of arousal and valence are also omitted. These dimensions can be considered in continuing work but do not play a role in the mere emotion recognition in this research. Therefore, it is pinpointed that these dimensions exist, but it does not address them in the further course of the study.

A technical limitation, however, is the mapping of processor and memory utilization. Since the system constantly updates these two indicators, it is impossible to identify the exact utilization. Thus, the processor and RAM utilization documentation only represents a snapshot, not a calculated average value. Furthermore, the maximum number of simultaneously recognizable emotions is another technical limitation. This paper assumes that only one emotion is contained in a sentence or voice recording. As the sentences and audio file length increases, the probability that multiple emotions are controlled increases. However, the machine learning method using the SVM can only classify one emotion, which is why the length of the voice recording is limited to three seconds.

On the other hand, the CNN in the Deep Learning method calculates a probability for each of the seven emotions. For this reason, this method can potentially identify multiple emotions within one speech recording. Another technical limitation is the applicability of the prototypes to only one person. The model training is based on emotional content in the audio files of the acquired databases. Individuals can be heard in each audio file so that the prototypes can apply emotion recognition only to individuals. When multiple individuals speak simultaneously, the prototypes cannot distinguish between individuals and their emotions. The extension to multi-person recognition goes beyond the definition of these prototypes and therefore needs to be investigated in further work.

Furthermore, the porting of prototypes is also limited. For example, only two device categories were selected since porting to more devices would exceed the scope of this paper. For this reason, porting to smartphones or tablets, for example, is not included.

6. Conclusions

The outcomes and interpretations presented in this study provide compelling evidence that the developed prototypes are functional and well-suited for practical applications. This Speech Emotion Recognition (SER) systems have the potential for various use cases and can offer extensions to existing products and services. Here are some examples of application areas and their resulting benefits.

(i) Universal Application: SER applications can be utilized wherever speech plays a central role, such as call centers, radio broadcasts, podcasts, and television shows. New business models can be developed that merge physical and virtual presence. For instance, implementing an SER system in a smart speaker can detect vocal activity and emotions in a home environment. These emotions can be visually presented to the user and, with their consent, transmitted to the producer for product optimization, offering the user a premium in return. Similarly, SER can automate the editing of highlights in a broadcast sports game based on detected emotions. Such scenarios can be extended to internet-based broadcasts like Twitch or Netflix.

(ii) Real-time Audience Mood Capture: SER applications can capture the current mood of an audience in real time. Unlike the previous use case, where emotions are summarized over time, this approach focuses on determining emotion levels at precise moments. This can be valuable in political talks or product presentations, where immediate feedback on expressed emotions is crucial. By providing unbiased input to speakers, SER enables them to gauge audience response accurately. These techniques apply to physical, virtual, or hybrid forms of communication, further emphasizing the increasing trend of remote participation.

(iii) Individual-focused Applications: SER applications can cater to individual users, tailoring experiences based on their detected emotions. For example, a smart speaker or automobile with an SER system can adjust music or lighting according to the user's emotional state. In gaming, the algorithm can offer in-game relief when anger is detected. Individualized advertising in social media or e-commerce platforms, varying prices dynamically based on emotional states (e.g., increasing costs for joyful emotions), is also a potential application.

This research investigated the ability of an SER system to distinguish speech, non-speech, and silence, as well as classify different emotions. The study involved a systematic literature review, developing two prototypes using machine learning and deep learning approaches, and training the models using a data corpus comprising five audio databases. Before being used for model training, the audio files underwent preprocessing, including conversion to a sampling rate of 16000 Hz and a mono channel.

In the machine learning approach, the openSMILE framework was employed for feature extraction, generating eGeMAPS features that were normalized and used for classification. Support Vector Machine (SVM) served as the classifier. The model training took approximately 96 hours on a server, and while successful porting to a notebook was achieved, porting to a Raspberry Pi was unsuccessful. The prototype demonstrated the capability to identify different sounds in under 1000 milliseconds and classify seven emotions in the case of speech.

In the Deep Learning Model approach, audio files were transformed into Mel spectrograms, normalized, and used as input for a CNN implemented using TensorFlow. The CNN performed feature extraction and classification. The model training, including transfer learning, took around six hours on the server. The completed model was successfully ported to a notebook and a Raspberry Pi. The notebook achieved classification below 1000 milliseconds, while the Raspberry Pi required approximately 4427 milliseconds. The models' computation time and classification accuracy were evaluated using the provided formula.

SER systems are embedded in Human-Computer Interaction (HCI) systems and can potentially be applied in everyday scenarios. The results of this study demonstrate that the technical feasibility of practical implementation is achievable, and several use cases described in this research can find real-world applications. Moreover, these findings highlight the growing relevance of SER in everyday communication, where remote participation is increasingly combined with a physical presence. Such SER systems have the potential to enhance human-machine interaction, making communication more human-like and intuitive. Based on this research, the acceptance and utilization of SER-enabled remote participation applications can be considered an extension of the fourth criterion of emotion recognition.

The results and discussions presented in this study can be further enriched and expanded through future research. Additional investigations could explore other emotions or broaden the scope of the utilized databases. Furthermore, examining arousal and valence dimensions would be valuable. Investigating the machine's subsequent actions linked to recognized emotions within the SER system is another avenue worth exploring. For example, studying the most suitable lighting settings, color combinations, or music choices to support or counteract specific emotions based on the determined emotions could be interesting. This could involve studying music's genre, volume, and beat rate and its relation to emotion recognition within songs. Combining both approaches, selecting songs based on identified emotions and playing them in response to human emotions, could provide an intriguing direction for further exploration.

Further research could also focus on the Deep Learning method, exploring different hyperparameters for model training and investigating modified transfer learning techniques. Multitask or semi-supervised learning could offer new perspectives in advancing SER research. Additionally, the limitations identified in this study open up opportunities for independent research and raise further questions. For instance, investigating whether an SER system can differentiate between multiple individuals based on speech or identify numerous emotions within a sentence could be explored. Exploring subjective user perception and experience could also be valuable. Lastly, the prototypes developed in this study were ported to two device categories, prompting whether they can be extended to other device categories, such as smartphones or tablets, each with its diverse range of devices and operating systems.

Regarding the real-time capability of the prototype, it would be worthwhile to explore the execution of digital signal processors and their potential for optimizing runtimes. Utilizing digital signal processors optimized for real-time functions like Fast Fourier Transform in mobile devices like the Raspberry Pi could further enhance the prototypes' real-time capability and overall performance.

In conclusion, this study demonstrates the theoretical and practical feasibility of real-time speech-based emotion recognition through edge computing. The implications of this research extend to practical applications and provide a foundation for future investigations.

Author Contributions

D.E.d.A. conceived the idea of researching a real-time processing method that captures and evaluates emotions in speech. R.B. and D.E.d.A. conceived the study. R.B. served as D.E.d.A.'s graduate advisor on his graduate thesis at the FOM University of Applied Sciences. All authors reviewed and approved the final manuscript.

Conflict of Interest

There is no conflict of interest.

Funding

This research received no external funding.

References

[1] El Ayadi, M., Kamel, M.S., Karray, F., 2011. Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognition. 44(3), 572-587. DOI: https://doi.org/10.1016/j.patcog.2010.09.020

[2] Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., et al., 2001. Emotion recognition in human-computer interaction. IEEE Signal Processing Magazine. 18(1), 32-80.

DOI: https://doi.org/10.1109/79.911197

[3] Schuller, B.W., 2018. Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends. Communications of the ACM. 61(5), 90-99.
DOI: https://doi.org/10.1145/3129340

[4] Kraus, M.W., 2017. Voice-only communication enhances empathic accuracy. American Psychologist. 72(7), 644.
DOI: https://doi.org/10.1037/amp0000147

[5] Akçay, M.B., Oğuz, K., 2020. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. Speech Communication. 116, 56-76.
DOI: https://doi.org/10.1016/j.specom.2019.12.001

[6] Dincer, I., 2000. Renewable energy and sustainable development: A crucial review. Renewable and Sustainable Energy Reviews. 4(2), 157-175.
DOI: https://doi.org/10.1016/S1364-0321(99)00011-8

[7] Chao, K.M., Hardison, R.C., Miller, W., 1994. Recent developments in linear-space alignment methods: A survey. Journal of Computational Biology. 1(4), 271-291.
DOI: https://doi.org/10.1089/cmb.1994.1.271

[8] Abbas, N., Zhang, Y., Taherkordi, A., et al., 2017. Mobile edge computing: A survey. IEEE Internet of Things Journal. 5(1), 450-465.
DOI: https://doi.org/10.1109/JIOT.2017.2750180

[9] Cao, K., Liu, Y., Meng, G., et al., 2020. An overview on edge computing research. IEEE Access. 8, 85714-85728.
DOI: https://doi.org/10.1109/ACCESS.2020.2991734

[10] Shi, W., Cao, J., Zhang, Q., et al., 2016. Edge computing: Vision and challenges. IEEE Internet of Things Journal. 3(5), 637-646.
DOI: https://doi.org/10.1109/JIOT.2016.2579198

[11] Lin, Y.L., Wei, G. (editors), 2005. Speech emotion recognition based on HMM and SVM. 2005 International Conference on Machine Learning and Cybernetics; 2005 Aug 18-21; Guangzhou, China. New York: IEEE.
DOI: https://doi.org/10.1109/icmlc.2005.1527805

[12] Nassif, A.B., Shahin, I., Attili, I., et al., 2019. Speech recognition using deep neural networks: A systematic review. IEEE Access. 7, 19143-19165.
DOI: https://doi.org/10.1109/ACCESS.2019.2896880

[13] Schuller, B., Batliner, A., Steidl, S., et al., 2011. Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. Speech Communication. 53(9-10), 1062-1087.
DOI: https://doi.org/10.1016/j.specom.2011.01.011

[14] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation. 9(8), 1735-1780.
DOI: https://doi.org/10.1162/neco.1997.9.8.1735

[15] Khalil, R.A., Jones, E., Babar, M.I., et al., 2019. Speech emotion recognition using deep learning techniques: A review. IEEE Access. 7, 117327-117345.
DOI: https://doi.org/10.1109/ACCESS.2019.2936124

[16] Hinton, G., Deng, L., Yu, D., et al., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine. 29(6), 82-97.
DOI: https://doi.org/10.1109/MSP.2012.2205597

[17] Torrey, L., Shavlik, J., Walker, T., et al., 2010. Transfer learning via advice taking. Advances in machine learning. Springer: Berlin.
DOI: https://doi.org/10.1007/978-3-642-05177-7_7

[18] Eyben, F., Scherer, K.R., Schuller, B.W., et al., 2015. The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. IEEE Transactions on Affective Computing. 7(2), 190-202.
DOI: https://doi.org/10.1109/TAFFC.2015.2457417

[19] Ekman, P., 1971. Universals and cultural differences in facial expressions of emotion. Nebraska Symposium on Motivation. University of Nebraska Press: Nebraska.

[20] Siedlecka, E., Denson, T.F., 2019. Experimental methods for inducing basic emotions: A qualitative review. Emotion Review. 11(1), 87-97.

DOI: https://doi.org/10.1177/1754073917749016

[21] Livingstone, S.R., Russo, F.A., 2018. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PloS One. 13(5), e0196391. DOI: https://doi.org/10.1371/journal.pone.0196391

[22] Burkhardt, F., Paeschke, A., Rolfes, M., et al. (editors), 2005. A database of German emotional speech. 9th European Conference on Speech Communication and Technology; 2005 Sep 4-8; Lisbon, Portugal. DOI: https://doi.org/10.21437/interspeech.2005-446

[23] Choudhury, A.R., Ghosh, A., Pandey, R., et al. (editors), 2018. Emotion recognition from speech signals using excitation source and spectral features. 2018 IEEE Applied Signal Processing Conference (ASPCON); 2018 Dec 7-9; Kolkata, India. New York: IEEE. DOI: https://doi.org/10.1109/ASPCON.2018.8748626

[24] Costantini, G., Iadarola, I., Paoloni, A., et al. (editors), 2014. EMOVO corpus: An Italian emotional speech database. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14); 2014 May; Reykjavik, Iceland.

[25] Martin, O., Kotsia, I., Macq, B., et al. (editors), 2006. The eNTERFACE'05 Audio-Visual emotion database. 22nd International Conference on Data Engineering Workshops (ICDEW'06); 2006 Apr 3-7; Atlanta, GA, USA. New York: IEEE. DOI: https://doi.org/10.1109/ICDEW.2006.145

[26] Lim, W., Jang, D., Lee, T. (editors), 2016. Speech emotion recognition using convolutional and Recurrent Neural Networks. 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA); 2016 Dec 13-16; Jeju, Korea (South). New York: IEEE. DOI: https://doi.org/10.1109/APSIPA.2016.7820699

[27] Mao, Q., Dong, M., Huang, Z., et al., 2014. Learning salient features for speech emotion recognition using convolutional neural networks. IEEE Transactions on Multimedia. 16(8), 2203-2213. DOI: https://doi.org/10.1109/TMM.2014.2360798

[28] Tzirakis, P., Zhang, J., Schuller, B.W. (editors), 2018. End-to-end speech emotion recognition using deep neural networks. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2018 Apr 15-20; Calgary, AB, Canada. New York: IEEE. DOI: https://doi.org/10.1109/ICASSP.2018.8462677

[29] Shinde, P.P., Shah, S. (editors), 2018. A review of machine learning and deep learning applications. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA); 2018 Aug 16-18; Pune, India. New York: IEEE. DOI: https://doi.org/10.1109/ICCUBEA.2018.8697857

[30] Adetiba, E., Adeyemi-Kayode, T.M., Akinrinmade, A.A., et al., 2021. Evolution of artificial intelligence programming languages-a systematic literature review. Journal of Computer Science. 17(11), 1157-1171. DOI: https://doi.org/10.3844/JCSSP.2021.1157.1171

[31] Hershey, S., Chaudhuri, S., Ellis, D.P.W., et al. (editors), 2017. CNN architectures for large-scale audio classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017 Mar 5-9; New Orleans, LA, USA. New York: IEEE. DOI: https://doi.org/10.1109/ICASSP.2017.7952132

[32] Gemmeke, J.F., Ellis, D.P.W., Freedman, D. et al. (editors), 2017. Audio set: An ontology and human-labeled dataset for audio events. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017 Mar 5-9; New Orleans, LA, USA. New York: IEEE. DOI: https://doi.org/10.1109/ICASSP.2017.7952261

[33] Vogt, T., André, E., Wagner, J., 2008. Automatic recognition of emotions from speech: A review of the literature and recommendations for practical realisation. Affect and emotion in human-computer interaction. Springer: Berlin. pp. 75-91.

DOI: https://doi.org/10.1007/978-3-540-85099-1_7

[34] Zhang, S., Zhang, S., Huang, T., et al., 2017. Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching. IEEE Transactions on Multimedia. 20(6), 1576-1590.
DOI: https://doi.org/10.1109/TMM.2017.2766843

[35] Liu, S., Nan, K., Lin, Y., et al. (editors), 2018. On-demand deep model compression for mobile devices: A usage-driven model selection framework. Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services; 2018 Jun 10-15; Munich Germany.
DOI: https://doi.org/10.1145/3210240.3210337

[36] Davis, J., Goadrich, M. (editors), 2006. The relationship between precision-recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning; 2006 Jun 25-29; Pittsburgh Pennsylvania USA.
DOI: https://doi.org/10.1145/1143844.1143874

[37] LeCun, Y., Bottou, L., Bengio, Y., et al., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 86(11), 2278-2324.
DOI: https://doi.org/10.1109/5.726791

[38] Krizhevsky, A., Sutskever, I., Hinton, G.E. (editors), 2012. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems; 2012 Dec 3-6; Lake Tahoe, Nevada, United States.

[39] Simonyan, K., Zisserman, A. (editors), 2015. Very deep convolutional networks for large-scale image recognition. The 3rd International Conference on Learning Representations (ICLR2015); 2015 May 7-9; San Diego, CA, USA.

[40] He, K., Zhang, X., Ren, S. et al. (editors), 2016. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27-30; Las Vegas, NV, USA. New York: IEEE.
DOI: https://doi.org/10.1109/CVPR.2016.90

[41] Sandler, M., Howard, A., Zhu, M., et al. (editors), 2018. MobileNetV2: Inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18-23; Salt Lake City, UT, USA. New York: IEEE.
DOI: https://doi.org/10.1109/CVPR.2018.00474

[42] Abadi, M., Barham, P., Chen, J. et al. (editors), 2016. TensorFlow: A system for large-scale machine learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16); 2016 Nov 2-4; Savannah, GA, USA.

[43] Kingma, D.P., Ba, J.L. (editors), 2015. Adam: A method for stochastic optimization. 3rd International Conference for Learning Representations; 2015 May 7-9; San Diego, CA, USA.

[44] Wang, X., Han, Y., Leung, V.C., et al., 2020. Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials. 22(2), 869-904.
DOI: https://doi.org/10.1109/COMST.2020.2970550

[45] Cummins, N., Amiriparian, S., Hagerer, G., et al. (editors), 2017. An image-based deep spectrum feature representation for the recognition of emotional speech. Proceedings of the 25th ACM international conference on Multimedia; 2017 Oct 23-27; Mountain View California USA.
DOI: https://doi.org/10.1145/3123266.3123371

[46] Ottl, S., Amiriparian, S., Gerczuk, M., et al. (editors), 2020. Group-level speech emotion recognition utilising deep spectrum features. Proceedings of the 2020 International Conference on Multimodal Interaction; 2020 Oct 25-29; Virtual Event Netherlands.
DOI: https://doi.org/10.1145/3382507.3417964