Enhancing Quality of Virtual Meetings through Facial and Vocal Emotion Recognition

Page, Philipp[a]1, Karaus, Kilian[a]2, and Donner, Maximilian[a]3

[a] University of Cologne, Albertus-Magnus-Platz, 50923 Cologne, Germany

Abstract Due to the COVID-19 pandemic, presentations via video conferencing software are part of the daily routine for many people. However, the quality of the presentations, which are used in universities to transfer knowledge but also in widespread areas of the business world, differs significantly. In order to improve meeting quality we have developed a web application allowing presenters to gather real-time metrics about the participants' moods during the meeting. In particular, the proposed system tracks the emotional state of the meeting participants and combines this data with subjective ratings. Emotional data of the audience is collected with the help of face recognition. At the same time, the systems analyzes the speaker's voice to deduce transmitted emotions. The collected data can help researchers to form hypotheses about what makes a "good" meeting. Additionally, presenters are able to change their presentation style during a meeting or in future meetings after performing a post hoc analysis.

1 Introduction

The global pandemic caused by the COVID-19 virus had a strong impact on the professional environment in 2020 and the first half of 2021. Employers sent their staff into the home office as far as possible. A survey found that before the Corona crisis, around 40 percent of employees in companies worked from home in the home office. During the pandemic, this share increased by about 20 percentage points to around 60 percent. This has sometimes meant that normal business meetings, whether with clients or colleagues, have had to take place via video conferencing platforms such as Zoom. Nevertheless, in the business world, as well as in university life, presentations are an essential part of sharing knowledge. It is important to present in the right way to have the best possible impact on the audience. For a good presentation, the way in which the presenters express their information using emotions is therefore of essential importance (D'Errico & Poggi, 2019).

Emotions in presentations can be transmitted in various ways, such as vocabulary and emotional words like "grateful", "unique", and "honored" when expressing

[1] Corresponding author: Philipp Page, e-mail: mail@philipp-page.de

[2] Corresponding author: Kilian Karaus, e-mail: kkaraus@smail.uni-koeln.de

[3] Corresponding author: Maximilian Donner, e-mail: mdonner@smail.uni-koeln.de

inspiration, whether the presenter communicates enthusiastically vs. indifferently or the provided gestures and facial expressions while presenting, such as smiling, being surprised or sad (Rößler et al., 2021; Zeng et al., 2019). Using deep neural networks, in particular Convolutional Neural Networks (CNNs), it is possible to determine emotions from facial expressions with a high degree of accuracy. Many researchers have shown that their deep learning-based methods perform very well on various datasets for recognizing emotions in the face (Ko, 2018). Voice emotions can also be determined using such neural networks. Using facial emotion recognition, researchers analyzed the relationship between facial expressions and the learning process to monitor and measure student engagement (De Carolis et al., 2019), for example, to provide personalized feedback to improve the learning experience.

In our seminar paper, we look more closely at the relationship between emotion through facial expressions and voice in presentation. The goal of this seminar is to build a web app (www.moody.digital) which measures and stores both facial and vocal emotions in real-time during live presentations with the help of CNNs and trained emotion recognition models. In addition, feedback from the participants is requested after each presentation to put the collected data into context. Such an evaluation could help researchers and practitioners to better understand the inherent relationship between emotions and presentations, and furthermore to improve the quality of presentations in the long term, thereby promoting better knowledge transfer. Based on this goal we formulate our research question as follows:

(RQ) How can we design a system to collect real-time feedback with the goal to reduce video conference fatigue?

2 Related Work

In the following sections, a literature review is conducted. The first two parts describe the current scientific approaches to measure emotions for both faces and voices using deep neural networks. The third part explains the relationship between the quality of presentations and emotions.

2.1 Facial Emotion Recognition

Facial Emotion Recognition (FER) is a subarea of computer vision and deals with the prediction of human emotional states using facial mimics and expressions in images, moving pictures or videos (Jain et al., 2019). According to Rößler et al. (2021) FER literature and research suggest two separate ways of feature generation: manually or automatically through a deep neural network. Additionally, they outline the differentiation of their underlying emotional model, which is either based on discrete emotional states or continuous dimensions (Rößler et al., 2021). Five fundamental discrete emotional states have been identified by Ekman and Keltner

(1997): Happiness, sadness, fear/surprise, disgust, and anger. Several different pieces of literature are recognizing other fundamental emotional states, and thus a single definition is not available. The continuous dimensions are defined differently. For instance, Mollahosseini et al. (2017) use two or three dimensions to explain emotions, with valence or pleasantness as one dimension and arousal or activation as the other.

In this work, we will focus on leveraging discrete emotional states since it is more maturely researched and more suitable to generate features using deep learning-based approaches. Considering the number of developments of recent years in deep learning, we concentrate on using the available variety of deep neural networks as the most appropriate technique in FER (Jain et al., 2019). We could also focus on FER approaches that use handcrafted features and which are according to Ko (2018) often deployed in the following three steps. First, face and facial component detection from the input image. Second, spatial and temporal feature extraction. And third, expression classification by e.g. Support Vector Machine or Random Forests to recognize emotional states. Although this manual way often leads to accurate results and requires less computational resources, many types of research, such as Jung et al. (2015), showed the superiority of deep learning algorithms with neural network architectures like CNNs over handcrafted approaches.

One of the main advantages of CNNs and other neural networks is the possibility to automatically learn features from the input images, which is called "end-to-end" learning (Ko, 2018). This Moody Prototype also uses a CNN implementation via the `face-api.js` library, which is described further in Section 4.2. These in the context of facial emotion recognition most used CNNs operate and process images, as the name discloses, "convolutionally" and can consider spatial information (Ko, 2018; Rößler et al., 2021). Thus, we utilize a CNN to recognize the emotional states of Ekman and Keltner (1997): happy, sad, fearful, angry, surprised, and disgusted. Additionally, since audiences listen and their faces appear often simply neutrally, we added another emotional state to our recognition model, called "neutral".

2.2 Vocal Emotion Recognition

The use of emotional impulses that portray emotion in a single modality, typically through facial expressions, has become popular in emotion research. Emotional communication in the natural world, in contrast, is temporal and multimodal. Multisensory integration is important while processing effective cues, according to research (Livingstone & Russo, 2018). Researchers have created their own multimodal stimuli in the absence of proven multimodal sets (Livingstone and Russo 2018). Researchers have also combined two different unimodal sets (Delle-Vigne et al., 2014) or joined self-created stimuli with an existing unimodal set (Zvyagintsev et al., 2013) to create multimodal stimuli. As each set differs in features, technical quality, and expressive intensity, comparing findings across studies may be difficult. As a consequence, differences in results could be attributed in part to differences in stimulus sets (Livingstone & Russo, 2018).

In their research paper, Livingstone and Russo (2018) describe the creation and validation of the RAVDESS, a dataset of dynamic and multimodal emotional emotions. The RAVDESS dataset has several key features that make it ideal for scientists, engineers, and physicians to use: It is provided freely available under a Creative Commons non-commercial license and is spoken by professional actors from North America. It has a variety of emotional reactions at two levels of emotional intensity (Livingstone & Russo, 2018).

The RAVDESS was validated with 247 speakers from across North America. The accuracy with which participants properly identified the actors' intended emotions was referred to as validity. As is customary in the literature, Livingstone and Russo (2018) looked at proportion correct scores. Overall, the results were excellent, with an average of 80% for audio-video, 75% for video-only, and 60% for audio-only (Livingstone & Russo, 2018).

Next to the RAVDESS dataset, there exist other voice datasets to train CNNs for voice emotion Recognition. The JL-Corpus is a set of four New Zealand English speakers who each say 15 sentences in each of five basic emotions with two repetitions, and another 10 sentences in each of five secondary emotions (An Open Source Emotional Speech Corpus for Human Robot Interaction Applications).

Another example is the Toronto Emotion Speech Set (TESS). It includes sentences of 200 words, which always have the same structure (Pichora-Fuller & Dupuis, 2020). The carrier phrase "Say the word ..." was spoken and recorded by two actors (aged 26 and 64), and these were each divided into seven different emotions (anger, disgust, fear, happiness, pleasant surprise, sadness and neutral) (Pichora-Fuller & Dupuis, 2020). Together, this resulted in 2800 different stimuli. The two actresses are from the Toronto area and their mother language is English (Pichora-Fuller & Dupuis, 2020). The EMO-DB database is a free emotional database. The Institute of Communication Science, Technical University of Berlin, Germany, established the database. The datasets consist of the voices of ten German professional speakers (five males and five females), which display seven different emotions (anger, boredom, anxiety, happiness, sadness, disgust, and neutral). There are 800 sentences recorded and seven emotions in the EMO-DB database (Burkhardt et al., 2005).

We use the four named datasets to train our own voice emotion CNN. The total combined amount of audio snippets from all datasets is 6216.

2.3 Presentations and Emotions

As stated by D'Errico and Poggi (2019) great speeches or presentations depend significantly on if the presenter transmits enough emotions while transferring their information. When compared to emotionless presentations, presentations that leverage emotional experiences and elicit an emotional response from the audience are more likely to capture the audience's attention (Gallo, 2014). Further facts prove that emotions are an important driver for engaging presentations and are a crucial enabler for successful knowledge transfer. Brain researchers have discovered that

presented knowledge will be remembered more likely if the presenter communicates emotionally (Tyng et al., 2017). Moreover, politicians frequently postulate wrath and despair in order to convey empathy and concern about the matter at hand (D'Errico & Poggi, 2019). Also, most people do not judge brands with facts and data but rely heavily on feelings and emotions they have about the brand (Damasio, 2006).

An exemplary study by De Carolis et al. (2019) researched a tool for emotion recognition from facial expressions during e-learnings and comes close to our developed tool concerning the facial emotion recognition part. By evaluating facial expressions, head movements, and gaze behavior from 5.5-hour video recordings, the authors were able to automatically quantify students' engagement. The information gathered was linked to a subjective evaluation of engagement based on a four-dimensional questionnaire: challenge, skill, engagement, and perceived learning. According to De Carolis et al. (2019), the less anxious and the more relaxed students are, the more they evaluated a more engaged questionnaire. They also showed that the more excited and engaged students felt during a presentation, the more the emotional analysis perceived them like this.

Another study by Chen et al. (2014) analyzes besides nonverbal behaviors also the speech delivery, such as fluency, pronunciation, and prosody. Here the authors collected several presentations from different speakers, which then were evaluated by human raters in different dimensions concerning the felt engagement during the presentation. They found out that indeed machine learning algorithms on vocal analysis can be used to assess the performance of presentations. Nevertheless, this study did not explicitly take into account emotion recognition but shows that the way the presenter delivers his speech correlates with the effectiveness of presentations.

Our paper project differs from previous studies since there has not yet been a tool developed, which can recognize emotions from two input sources: facial and vocal expressions. With this tool at hand, and previous findings in literature that emotions are an important driver for successful presentations, future presenters could enhance their presentation style by exactly knowing the audiences' current emotional state.

3 Hypothesis

Our hypothesis is based on the findings of Rößler et al. (2021), who used CNNs and face emotion recognition to find that the presenter must always maintain a positive attitude to convey enthusiasm and positive energy to the audience. This was studied using only face emotion recognition technology. Our hypothesis goes further in this regard, and we are therefore also investigating the emotional tone of voice of the presenter and its influence on the feedback from the participants of the presentation. With the support of both FER and vocal emotion recognition, and the subsequent feedback from the participants, we want to build an open source application which is showing the presenter facial and voice emotions of the audience in real-time. Therefore, the presenter has a tool to improve his presentation live or also later looking at the feedback and the collected emotions. The application can also operate

as a data collection tool, to get further information about online virtual meetings, and about what makes it for the audience a good or a bad presentation. Thereby, users of our application are given the possibility to improve the experience even during lengthy video presentations.

4 Method

In this section, we describe the way of how we realized the Moody tool. We present the system architecture and explain how we implement the facial and vocal emotion recognition model in more detail.

4.1 System Architecture

The Moody app is a browser-based application written in TypeScript[4] using the React framework[5]. We make use of the AWS platform for data storage and hosting. In particular, we leverage AWS Amplify which covers the whole software lifecycle of mobile- and web applications from development to continuous deployment (CD) and maintenance[6]. Figure 1 depicts the high-level system architecture.

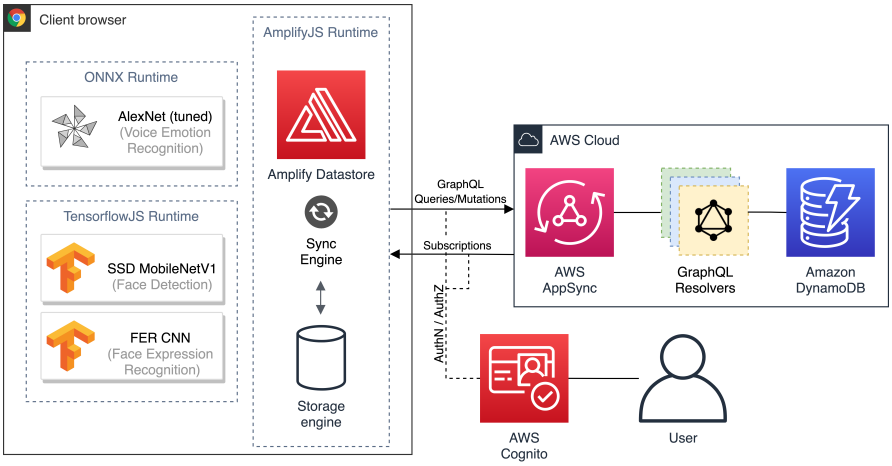

Fig. 1 System Architecture Diagram. Moody features a client-server architecture built on top of the AWS platform.

[4] https://www.typescriptlang.org/ (last accessed 07/28/2021)

[5] https://reactjs.org/ (last accessed: 07/23/2021)

[6] https://aws.amazon.com/amplify/ (last accessed: 07/23/2021)

The client-side of the application consists of three major subsystems. The first subsystem is AmplifyJS, which is AWS Amplify's software development kit (SDK) for JavaScript-based applications. For us, it provides the means for data modeling, data persistence across devices as well as user authentication and authorization. Our data is modeled according to the third normal form (Codd, 1972). The full relational schema can be found in the Appendix, Figure 7. Amplify Datastore provides an API to a native storage engine that synchronizes asynchronously with the Cloud in the background[7]. The main advantage of this local first approach is a reactive UI because user interaction is decoupled from the network. We keep the UI state consistent across all components using Redux[8] as a state management tool.

The two other subsystems form the machine learning part of Moody. We decided to run all machine learning algorithms locally in the browser of the user. This is beneficial because the internet connection is not impacted by transferring large image and sound data to a remote server every second. In addition, it supports user privacy as only aggregated emotion scores are saved in the Cloud. One drawback is that slower computers might not handle the inference load very well.

The two face models use the TensorflowJS runtime[9] and run in a WebGL context[10] allowing for fast computations of the video stream provided by the user. This video stream is captured using the Screen Capture API[11] implemented by all major browsers. As a presenter, you typically want to select the window where the faces of your audience are located. Moody will then detect bounding boxes for these faces and perform expression prediction every second giving the presenter feedback. Please note that at the time of writing it is not possible to select a specific window to share in Safari.

Our voice emotion model is originally written in Python with the PyTorch framework[12]. As it is not possible to run Python code efficiently in the browser we leverage ONNX. ONNX is a standardized open-source format to represent neural networks as computation graphs and is currently maintained by the community and Microsoft. It comes along with highly optimized runtimes for different platforms with even better inference performance as compared to the original framework (in our case PyTorch) (ONNX Runtime developers, 2021). `onnxruntime-web`[13] is the browser runtime which we use to run the voice emotion model. Since the WebGL backend is not yet feature-complete we run our model on the client CPU with WebAssembly[14]. WebAssembly is a compilation target for native languages like C(++) or Rust and allows code execution at close to native speed in the browser. It is currently supported by

[7] https://docs.amplify.aws/lib/datastore/getting-started/q/platform/js (last accessed: 07/23/2021)

[8] https://redux.js.org/ (last accessed: 07/23/2021)

[9] https://www.tensorflow.org/js/ (last accessed: 07/23/2021)

[10] https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API (last accessed: 07/23/2021)

[11] https://developer.mozilla.org/en-US/docs/Web/API/Screen_Capture_API (last accessed: 07/23/2021)

[12] https://pytorch.org/ (last access: 07/23/2021)

[13] https://github.com/microsoft/onnxruntime/tree/master/js/web (last accessed: 07/23/2021)

[14] https://webassembly.org/ (last accessed: 07/23/2021)

more than 90% of all browsers[15]. To capture the sound of the speaker's microphone we utilize the Web Audio API[16]. As a speaker, you are also given the possibility to select the input device. When the voice tracking is active Moody will predict voice emotions every 2.1 seconds and give the speaker live feedback.

All data which is collected from user interaction and the machine learning model predictions is persisted in the AWS Cloud to be available across devices and for further research analysis. Therefore, we have deployed a GraphQL API endpoint with AWS AppSync[17]. This endpoint is called by Amplify Datastore in the background to synchronize new data with the backend. AppSync authorizes each request with AWS Cognito which we use for user authentication. As AppSync is data source agnostic it needs resolvers telling it how to interact with a backend. We chose DynamoDB as the database backend[18]. Luckily, Amplify creates these resolvers automatically given a GraphQL compliant data schema[19].

4.2 Facial Emotion Recognition

We implement our facial emotion recognition with the `faceapi.js`[20] JavaScript API for face detection, which is implemented itself on top of the TensorFlowJS core API and can perform face recognition in the browser. To execute in this project both, the face detection itself and the face expression/emotion recognition, we use two neural networks. The input data are single images from the live video conference, which are picked up from the meeting tool (e.g. Zoom) window where the participants' cameras appear. This desktop window has to be selected by the Moody user before he starts the emotion tracking. The two models, which are briefly presented below, were already developed and provided by the authors of `faceapi.js`, which is why no more data preprocessing was necessary and these only had to be implemented in our system architecture.

First, the algorithm has to detect the faces in the live video and create bounding boxes around the corner points of the faces. For this purpose `face-api.js` uses the SSD (Single Shot Multibox Detector) MobileNetV1 face detection model[21] (Howard et al., 2017). The neural net will compute the positions of each face in a picture and return the bounding boxes and their occurrence probabilities for each face. Instead of then focusing on short inference time, this face detector aims for high

[15] https://www.tensorflow.org/js/guide/platform_environment#why_wasm (last accessed: 07/23/2021)

[16] https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API (last accessed: 07/23/2021)

[17] https://aws.amazon.com/appsync/ (last accessed: 07/23/2021)

[18] https://aws.amazon.com/dynamodb/ (last accessed: 07/23/2021)

[19] https://spec.graphql.org/June2018/#sec-Type-System (last accessed: 07/23/2021)

[20] https://justadudewhohacks.github.io/face-api.js/docs/index.html (last accessed: 07/23/2021)

[21] https://justadudewhohacks.github.io/face-api.js/docs/classes/ssdmobilenetv1.html (last accessed: 07/30/2021)

accuracy in recognizing face bounding boxes. Nevertheless, the size of the quantized model is rather small at about 5.4 MB and thus it works at an acceptable speed. This face detection model has been trained on a dataset called *WIDERFACE* which was developed by the authors Yang et al. (2016) and is a face detection benchmark dataset from publicly available images. The authors chose 32,303 images and labeled 393,703 faces on these with a high degree of variability in scale, pose, and occlusion. The weights for the SSD MobileNetV1 model and the final face detector, powered by the TensorFlow object detection API[22] and trained by Yang et al. (2016), were provided by yeephycho[23] in a GitHub-Repository[24].

The second neural net uses the detected faces and their bounding boxes as an input for the facial expression recognition model. When all faces were detected on an image, the model receives this information and returns an array consisting of all detected faces and their belonging face expressions. The input images in our tool are single frozen frames of the videoconference camera screen. The authors of `faceapi.js` claim that it is fast and provides reasonable accuracy. The model is about 310 kB in size and uses depthwise separable convolutions as well as densely connected blocks. It was trained on a variety of photos, including photographs scraped from the web and images from publicly available sources. Wearing glasses or low light conditions may reduce the accuracy of the prediction results.

4.3 Vocal Emotion Recognition

In the following sections, we will explain how we preprocess our data regarding our voice emotion model, as well as how we set the model up and train it. We use the datasets introduced in Section 2 as well as a self-designed CNN based on an AlexNet architecture.

4.3.1 Data Preprocessing

The input we will need later for our voice model is the waveform of the sound which needs to be preprocessed. To avoid fitting the dataset specific volume of the audio samples we apply RMS audio normalization. It is a form of loudness normalization which brings heterogeneous audio sequences to a consistent sound level. We use the formula described by Equation 1 for this purpose where y_n is the n^{th} normalized audio sample and x_n the original audio sample in an audio sequence of N samples

[22] https://github.com/tensorflow/models/tree/master/research/object_detection (last accessed: 07/30/2021)

[23] https://github.com/yeephycho (last accessed: 07/23/2021)

[24] https://github.com/yeephycho/tensorflow-face-detection (last accessed: 07/23/2021)

in total. r is a hyperparameter describing the target RMS level[25]:

$$y_n = \sqrt{\frac{N - 10\left(\frac{r}{20}\right)}{\sum_{i=0}^{N-1} x_i^2}} \cdot x_n \tag{1}$$

Then the normalized waveform is converted into Mel spectrograms. This step is part of a feature extractor block in the final neural network. We use 128 Mel coefficients. We do this with the `torchlibrosa` library (Kong et al., 2020). In this library we have re-implemented the Short-Time-Fourier-Transform (STFT) as a CNN with Conv2d layers so that it is compatible with `onnxruntime-web` (ONNX Runtime developers, 2021). The STFT is a Fourier Transform version that uses a sliding time window to break up the audio signal into smaller chunks. We use a window size of 2,048 and a sample rate of 22,050. It takes each section of the Fast Fourier Transform (FFT) and then combines them. As a result, it can catch changes in frequency over time (Griffin & Lim, 1984). To get in the end more data and therefore better training results, we augmented our existing data sets (RAVDESS, TESS, JL-Corpus, EMO-DB) with common data augmentation methods. We used stretching and squeezing (randomly slow down or speed up the sound), background noise (add some random values to the sound), random shifting (shift audio to the left or the right by a random amount), left- and right-trimming (cut off any silence in the beginning or end), and pitch tuning (randomly modify the frequency of parts of the sound)(McFee et al., 2015). The voice emotions are recorded in time windows of 2.1 seconds. This time is the average duration of the samples in the datasets we use.

4.3.2 AlexNet Architecture

The input for the voice emotion model is an RMS-normalized sound sequence of length 463,050. In a first feature extractor block, the neural network automatically extracts Mel spectrograms from the sound waveform as described in Section 4.3.1. A major advantage of having the feature extraction as part of the model is that this step does not have to be re-implemented for each programming language – in our case JavaScript – thus leading to consistent preprocessing and prediction results independent of the environment.

These Mel spectrograms serve as input for the convolutional neural network part. Based on the datasets introduced in Section 2.2 we train the model. We have tried out two well-known architectures: ResNet18 and AlexNet (He et al., 2015; Krizhevsky et al., 2012). The performance of both models is very similar, only the size of the models differs. Since the AlexNet with 32.3 MB is almost half the size of the ResNet18 with 59.7 MB and has fewer parameters, we decided to use the AlexNet for our web app (cf. Section 5.3). This is then exported to ONNX to be executed in the browser with the help of `onnxruntime-web` for inference.

[25] https://pydiogment.readthedocs.io/en/latest/_modules/pydiogment/auga.html#normalize (last accessed: 07/23/2021)

Additionally, we have modified the default AlexNet architecture with the goal to reduce overfitting by adding batch normalization, layer normalization, and dropout layers. Figure 2 illustrates the detailed architecture. Since we use cross-entropy loss[26] as loss function the model output are the raw logits. Applying the softmax function[27] on this vector will yield the corresponding class probabilities for each emotion.

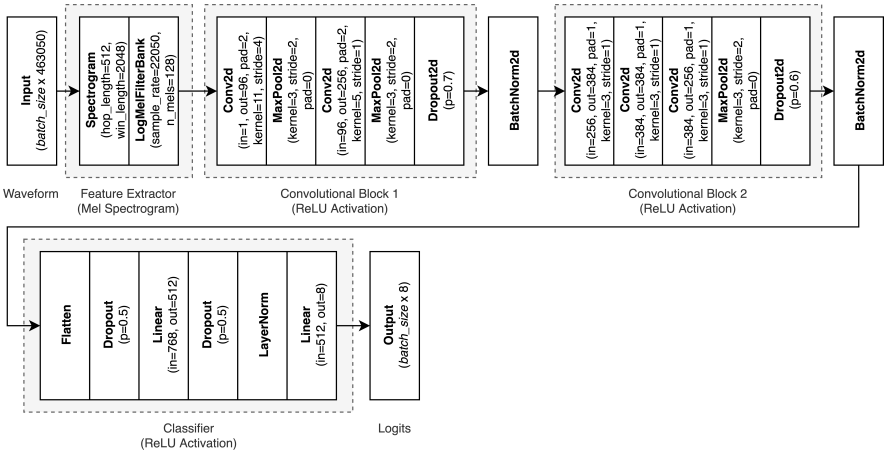

Fig. 2 The modified AlexNet architecture of Moody's voice emotion model. All *Conv2d* and *Linear* layers are connected via the ReLU activation function.

5 Results

In this section, we show the outcomes of the implemented system architecture as well as the facial and vocal emotion recognition as described in Section 4.

5.1 Experiment Environment & Testing Setup

We conducted our experiments during a virtual seminar "Collaborative Innovation Networks" (COINs) in the summer semester of 2021 held by Prof. Peter Gloor (MIT). The seminar consisted of 23 students from the Universities of Cologne, Bamberg and Lucerne and three instructors. All students formed in total six teams with three

[26] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html (last accessed: 08/02/2021)

[27] https://pytorch.org/docs/stable/generated/torch.nn.functional.softmax.html (last accessed: 02/08/2021)

to five participants working on complex subject-related and practical topics. Due to the global COVID-19 pandemic at that time and agile project management, all teams presented their results in weekly and bi-weekly sprints during a virtual Zoom meeting. In these virtual status meetings, every group presented their project goals, progress, results or plans of the last and next iteration and a retrospective in a short presentation of approximately ten minutes.

Since our prototype Moody was not ready for use right at the first meeting and still needed to be developed to track emotions we were only able to ask the other groups to use our Moody Tracker from the third meeting onwards. For this purpose, we wrote a short usage guide within the invitation before the meeting, so that each group could activate the tracker during their presentation. Therefore, in addition to the usage instructions, we asked the other teams to generate a feedback link via our website (www.moody.digital) after their presentation and share it with the other meeting participants.

The underlying intention here was primarily to receive instant feedback about our application in real-world usage. This made continuous testing possible, which enabled us to constantly work on errors and bugs reported by the users in each status meeting. Also, we had the chance to quickly collect subjective ratings about the perceived quality of each presentation and compare it to the tracked emotion data. The focus of this paper lies neither in statistically analyzing the correlation nor the causality between the experienced emotionality of the audience and the presentation quality, but we had the chance to cursorily analyze the first insights.

Exemplarily, Figure 3 shows a screenshot of the recorded statistics of our status presentation on the 6th July 2021. With this recording, we found that our last presenter on that day seemed to speak rather sad than happy. He was able to use this information for the next presentation to increase his positive emotion transmissions via voice. Additionally, we have seen that the faces and voices were interpreted mainly neutrally, because of the formal and informative character of the presentations. This aligns with our expectations and indicates that the models are predicting correctly.

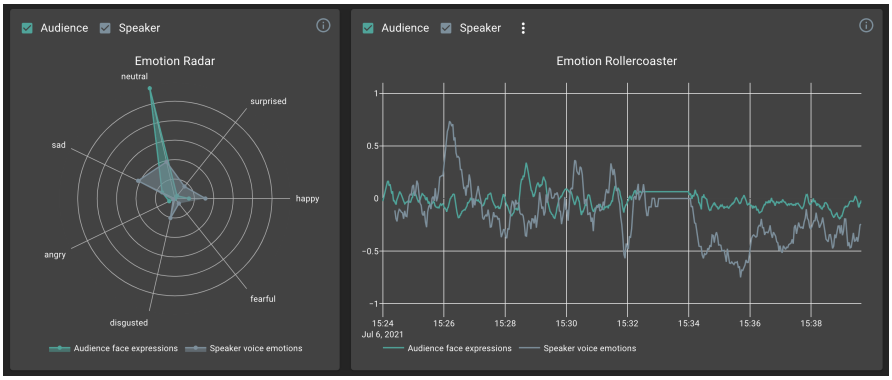

Fig. 3 Recorded statistics of the status presentation on the 6th July 2021.

5.2 Face Emotion Recognition

Since we adopt the pre-developed `faceapi.js` and its two underlying models for face detection and expression recognition, we do not want to focus in this chapter on presenting the resulting metrics and accuracies. Instead, we briefly show that we implemented the API correctly in our tool and how it works. Figure 4 illustrates the presenter's view of the detected faces and their current emotions when he clicks on the "Faces" tab while the emotion tracking is running. It seems that the model identifies the emotions expressed by the faces correctly.

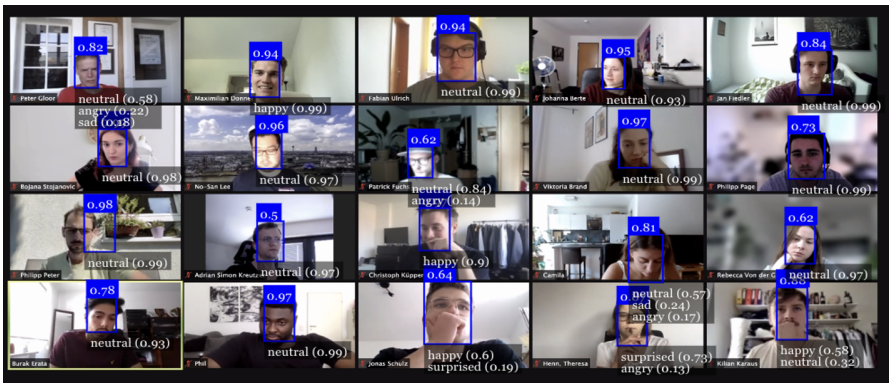

Fig. 4 Presenter's view of the detected faces in the according "Faces" tab.

The models receive every second a frozen frame from the video conference window with all participants' cameras and calculate every second the probabilities for the current Ekman and Keltner (1997) emotion per person respectively per detected face. Thus, every emotion out of neutral, happy, sad, angry, fearful, disgusted, and surprised gets a probability between 0 and 1 assigned. Exemplarily, "happy (0.8)" and "neutral (0.2)" mean that the particular face looks 80% happy and smiling, and is by 20% neutrally predicted. In order to receive the overall audience emotion score to visualize in the Emotion Rollercoaster (cf. Figure 3) we calculate the mean face emotion value from all detected faces. The emotion score lies between -1 and 1 since we assign the emotions from Ekman and Keltner (1997) to negative, neutral, and positive values. The probabilities for happy and surprised are assigned to 1, neutral to 0, and sad, disgusted, angry, and fearful to -1.

5.3 Vocal Emotion Recognition

Since the AlexNet was trained on the four different data sets (RAVEDESS, EMO-DB, TESS, JL-Corpus) to get as much variety as possible, the accuracy of the model is 87.17% (cf. Appendix, Figure 6).

To get the best possible result, early stopping is included. This is best at the eighth epoch, as the loss does not change significantly afterwards (cf. Appendix, Figure 7).

The difference to the ResNet model, apart from minor differences in accuracy, is mainly the size of the model (cf. Appendix, Figures 8 and 9). The AlexNet has a size of 32.3 MB whereas the ResNet is almost twice as big with a size of 59.6 MB. Since the data model is reloaded in the browser every time the voice emotion model is used, which takes up time as well as memory, we finally decided to use the AlexNet. It has solid accuracy and smaller memory size which is ideal for usage on end-user devices. Table 1 depicts the model metrics in comparison.

	AlexNet	ResNet18
Training accuracy	**95.66%**	91.43%
Validation accuracy	87.17%	**87.85%**
Training loss	**0.1297**	0.2521
Validation loss	0.3827	**0.3520**
Best epoch	**30**	35
Epoch (early stopping)	8	**7**
Model size	**32.3 MB**	59.6 MB
Training duration	**11min 17s**	21min 51s

Table 1 Model metrics in comparison. The AlexNet model is used because it is more parsimonious in model size with a comparable accuracy and cross-entropy loss. Bold values are preferred.

Finally, the confusion matrix shows that our model has fairly high true prediction ratios and low deviations for almost all emotions (cf. Figure 5). The highest ratio of wrong predictions is for the emotion “sad”. While the model predicts 71.19% correctly, it predicts “neutral” wrongly in 20.90% of all cases in the validation set.

The voice emotion model can be activated optionally at any time during a meeting. It can be activated during the whole meeting or just at the beginning or end of the meeting and will then be downloaded by the browser in case it is not yet cached. During the meeting, in addition to the Emotion Rollercoaster, the voice emotions can also be determined in real-time. A time span of 2.1 seconds is always taken to track the current emotion, which is delivering a good prediction of the emotion at that moment.

The app calculates a moving average to make the prediction curves smoother and better readable. This way, the presenter can identify emotional trends more clearly even during longer meetings. Proper values for the moving average span are guessed

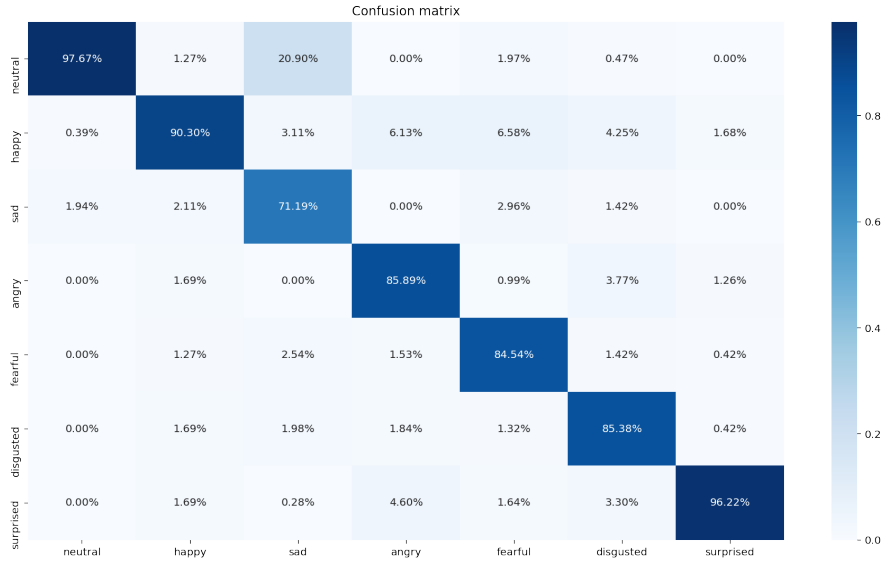

Fig. 5 AlexNet Confusion Matrix

in real-time by calculating $10 \cdot std(E) + log_{10}(N)$ for the N tracked emotions E and can be overwritten by the user after a meeting.

6 Discussion

In this seminar paper, we describe how we built our moody.digital web application, and how it recognizes face emotions with `face-api.js` and voice emotions with our self-built AlexNet. The web app is an extension of the existing web app from the previous semester of the COINs seminar at the University of Cologne in collaboration with the University of Bamberg and the University of Applied Science and Arts in Lucerne. The previous project already built an application that was able to track face emotions during virtual meetings in real-time. With our additions, the app has gained further features. On the one hand, it is now able to recognize live voice emotions in addition to face emotions. On the other hand, emotions are displayed live during the meeting in an Emotion Rollercoaster and can be further analyzed in a facial and vocal emotion radar chart. In addition to the live components, the data is also stored persistently in a database for a post hoc analysis. This way makes the emotional investigation of the presentation, lecture, or meeting temporally flexible in comparison to the tool from the previous COINs team or to the work of Rößler et al. Also, when our tool is compared to the work of Rößler et al. it has to be pointed out, that our focus lies in the system development in order to put their findings into practice. Our web-based tool can be understood as an answer to the

work of Rößler et al. to extend it with a useful real-world application and adding voice emotion recognition and data persistence. The feedback by the audience can also be visualized and analyzed in real-time and afterward. In our Zoom meetings during the COINs seminars, we were able to determine a first correlation between the face and voice emotions (cf. Figure 3). To draw a meaningful conclusion, further video meetings must be analyzed with the tool. However, based on the accuracy of the emotion recognition models, we have designed an app that recognizes these with high accuracy and is therefore capable of such analyses. With this application, we have built a tool with which future presenters can improve their style of presentation and get more audience with it. It can also be used for further research purposes to gain more insights into presentations.

7 Limitations and Future Work

While the Moody web app is able to capture and persist emotions with reasonable accuracy in real-time during virtual meetings, there is still potential for improvement and future work. In addition, at the time of writing, certain limitations are imposed by the early stage of development of `onnxruntime-web` and the implementations of web standards in major browsers.

The current implementation of the voice emotion model predicts the speaker's vocal emotions in windows of 2.1 seconds. This way of predicting is unfortunately prone to the start and end of a sound sequence even though we already counteract this behavior by using random shifting during data augmentation. One way to make the model more robust against shifted sentences might be the usage of a rolling estimator. Instead of predicting in 2.1-second windows, one could add three estimators each shifted by 0.3 seconds and perform three additional predictions on the shifted sound sequences. In the end, one would reconcile the results to one predicted value, for example by averaging. This way, the impact on the prediction result of sentences starting in one window and ending in another window can eventually be reduced.

A limitation going in line with the rolling estimator is the computational power required to perform multiple predictions simultaneously. As the machine learning model runs on the end-user device it is important not to impact the user experience. `onnxruntime-web` does not yet support all operators needed by the voice model in WebGL[28]. Therefore, it might be necessary to wait until the GPU-accelerated WebGL context is supported by ONNX before implementing this feature.

Another hypothesis for improving the voice emotion recognition accuracy is to add more data (e.g. by creating an own dataset) and to try out language and gender-specific models.

A limitation of the current web app is the feature completeness of Web Media APIs in different browsers. For example, Safari does not allow selecting a specific

[28] Current status can be checked here: https://github.com/microsoft/onnxruntime/tree/master/js/web#webgl-backend (last accessed: 07/29/2021)

window when sharing the user screen. This way it is not possible to use the Moody app with only one screen. Mozilla Firefox does not support an audio sample rate different from the default sample rate of the user device[29]. This leads to some users not being able to use the voice emotion recognition if their device does not use the sample rate 22,050 which the model expects.

[29] Current status can be checked here: https://developer.mozilla.org/en-US/docs/Web/API/MediaTrackConstraints/sampleRate (last accessed: 07/29/2021)

Appendix

AlexNet Model

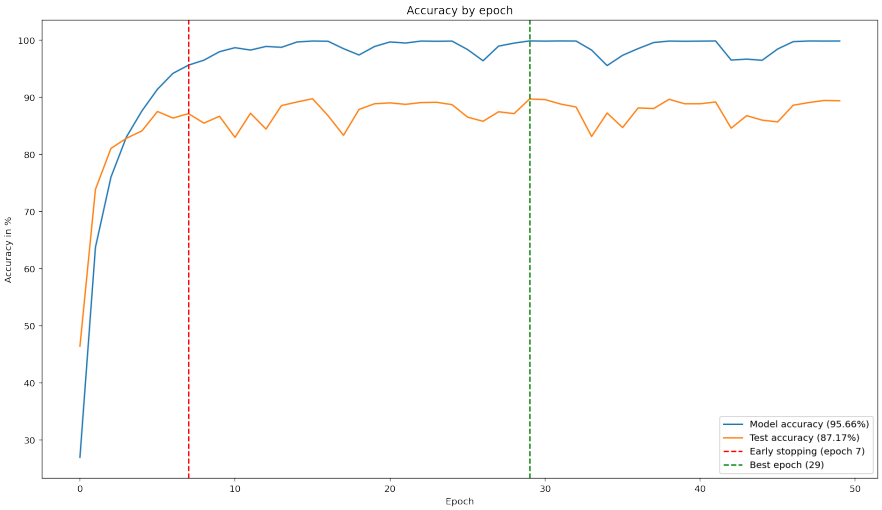

Fig. 6 AlexNet accuracy by epoch.

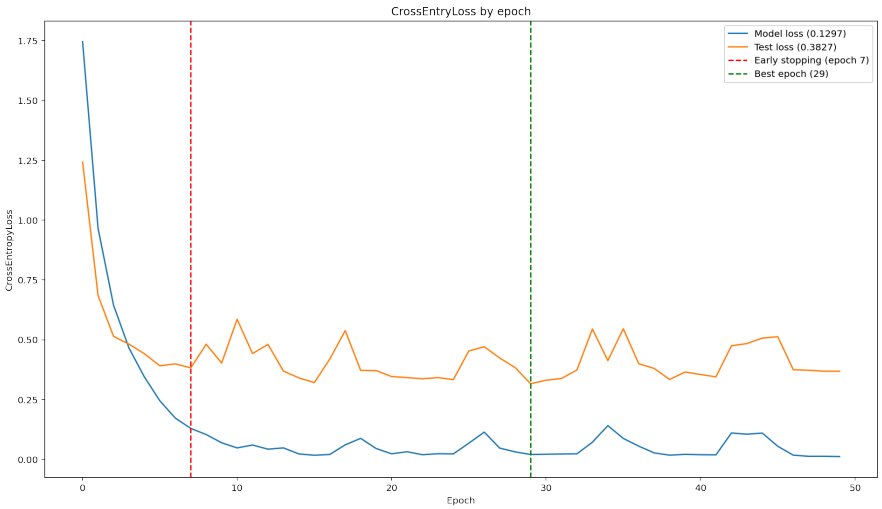

Fig. 7 AlexNet cross-entropy loss by epoch.

ResNet18 Model

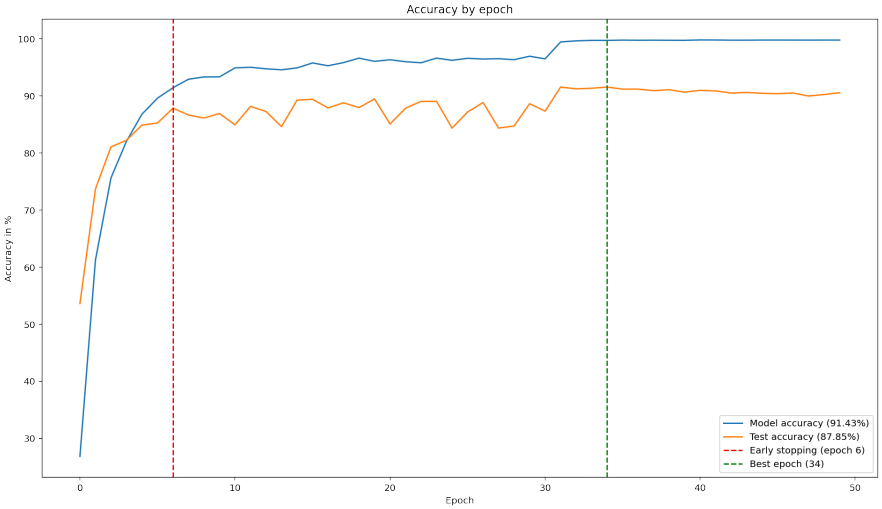

Fig. 8 ResNet18 accuracy by epoch.

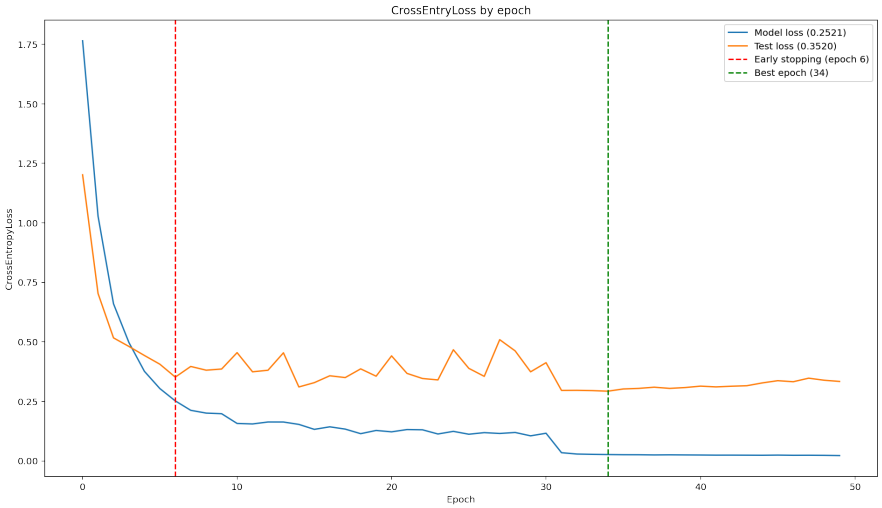

Fig. 9 ResNet18 cross-entropy loss by epoch.

Relational Data Model

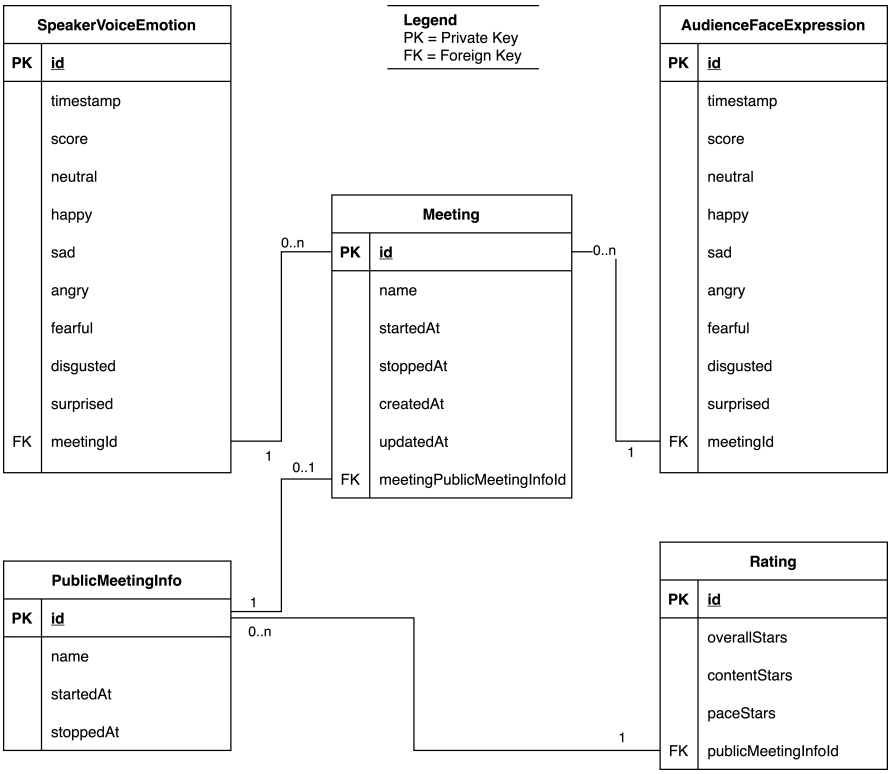

Fig. 10 The entity relationship diagram (ERD) for the database structure of Moody. Relationships are modeled in UML style. Each table has an additional owner field storing the user id from AWS Cognito which is omitted in the illustration for brevity.

Source Code

All source code related to the Moody application can be found at the GitHub organization `COINS-SS21`: https://github.com/COINS-SS21.

- Web application: https://github.com/COINS-SS21/moody
- Speech emotion recognition model: https://github.com/COINS-SS21/moody-ser
- LaTeX source of this paper: https://github.com/COINS-SS21/moody-paper

References

Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W., & Weiss, B. (2005). A Database of German Emotional Speech, 4.

Chen, L., Feng, G., Joe, J., Leong, C. W., Kitchen, C., & Lee, C. M. (2014). Towards Automated Assessment of Public Speaking Skills Using Multimodal Cues [event-place: Istanbul, Turkey]. *Proceedings of the 16th International Conference on Multimodal Interaction*, 200–203. https://doi.org/10.1145/2663204.2663265

Codd, E. F. (1972). Further normalization of the data base relational model [Publisher: Prentice-Hall Englewood Cliffs, NJ]. *Data base systems*, *6*, 33–64.

Damasio, A. (2006, July 6). *Descartes' Error: Emotion, Reason and the Human Brain*. Vintage.

De Carolis, B., D'Errico, F., Macchiarulo, N., & Palestra, G. (2019). "Engaged Faces": Measuring and Monitoring Student Engagement from Face and Gaze Behavior, 80–85.

Delle-Vigne, D., Kornreich, C., Verbanck, P., & Campanella, S. (2014). Subclinical alexithymia modulates early audio-visual perceptive and attentional event-related potentials [Publisher: Frontiers]. *Frontiers in Human Neuroscience*, *8*, 106.

D'Errico, F., & Poggi, I. (2019). Tracking a leader's humility and its emotions from body, face and voice. *Web Intelligence and Agent Systems*, *17*, 63–74. https://doi.org/10.3233/WEB-190401

Ekman, P., & Keltner, D. (1997). Universal facial expressions of emotion. *Segerstrale U, P. Molnar P, eds. Nonverbal communication: Where nature meets culture*, *8*, 27–46.

Gallo, C. (2014). *Talk like TED: the 9 public-speaking secrets of the world's top minds*. St. Martin's Press.

Griffin, D., & Lim, J. (1984). Signal estimation from modified short-time Fourier transform [Publisher: IEEE]. *IEEE Transactions on acoustics, speech, and signal processing*, *32*(2), 236–243.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition [_eprint: 1512.03385]. *CoRR*, *abs/1512.03385*. http://arxiv.org/abs/1512.03385

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications [_eprint: 1704.04861]. *CoRR*, *abs/1704.04861*. http://arxiv.org/abs/1704.04861

Jain, D. K., Shamsolmoali, P., & Sehdev, P. (2019). Extended deep neural network for facial emotion recognition [Publisher: Elsevier]. *Pattern Recognition Letters*, *120*, 69–74.

Jung, H., Lee, S., Yim, J., Park, S., & Kim, J. (2015). Joint fine-tuning in deep neural networks for facial expression recognition, 2983–2991.

Ko, B. C. (2018). A brief review of facial emotion recognition based on visual information [Publisher: Multidisciplinary Digital Publishing Institute]. *sensors*, *18*(2), 401.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., & Plumbley, M. D. (2020). Panns: Large-scale pretrained audio neural networks for audio pattern recognition [Publisher: IEEE]. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 2880–2894.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks [event-place: Lake Tahoe, Nevada]. *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 1097–1105.

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 [Publisher: Public Library of Science San Francisco, CA USA]. *PloS one*, *13*(5), e0196391.

McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and Music Signal Analysis in Python. *14*, 18–24. https://doi.org/10.25080/Majora-7b98e3ed-003

Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). Affectnet: A database for facial expression, valence, and arousal computing in the wild [Publisher: IEEE]. *IEEE Transactions on Affective Computing*, *10*(1), 18–31.

ONNX Runtime developers. (2021). ONNX Runtime. https://onnxruntime.ai/

Pichora-Fuller, M. K., & Dupuis, K. (2020). Toronto emotional speech set (TESS) [Publisher: Scholars Portal Dataverse Type: dataset]. https://doi.org/10.5683/SP2/E8H2MF

Rößler, J., Sun, J., & Gloor, P. (2021). Reducing Videoconferencing Fatigue through Facial Emotion Recognition [Publisher: Multidisciplinary Digital Publishing Institute]. *Future Internet*, *13*(5), 126.

Tyng, C. M., Amin, H. U., Saad, M. N. M., & Malik, A. S. (2017). The Influences of Emotion on Learning and Memory [Publisher: Frontiers]. *Frontiers in Psychology*, *0*. https://doi.org/10.3389/fpsyg.2017.01454

Yang, S., Luo, P., Loy, C.-C., & Tang, X. (2016). Wider face: A face detection benchmark. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5525–5533.

Zeng, H., Wang, X., Wu, A., Wang, Y., Li, Q., Endert, A., & Qu, H. (2019). EmoCo: Visual analysis of emotion coherence in presentation videos [Publisher: IEEE]. *IEEE transactions on visualization and computer graphics*, *26*(1), 927–937.

Zvyagintsev, M., Parisi, C., Chechko, N., Nikolaev, A. R., & Mathiak, K. (2013). Attention and multisensory integration of emotions in schizophrenia [Publisher: Frontiers]. *Frontiers in human neuroscience*, *7*, 674.