ML-based Video Analytics Tool for Multimodal Emotion Recognition

Akhras Emad
Department of Statistics and Actuarial Science
The University of Hong Kong

APAI3799

Abstract

The automation of the interview process through machine learning presents a significant challenge. Despite remarkable advancements in software and hardware, the creation of an agent that entirely eliminates the need for a human interviewer remains elusive. However, extensive research has been conducted to develop methods and models that can evaluate individual aspects of human behavior, such as facial expression recognition (FER), eye gaze tracking, eye emotion recognition (EER), and speech emotion recognition (SER). This study aims to demonstrate the potential of a machine learning-based video analytics tool for conducting an in-depth analysis of a candidate's behavior during an interview setting. This is achieved by combining the results from running concurrent models, where each model is designed for a specific purpose. Each recognition task has its neural network architecture and is trained on respective datasets. The tool provides two modes of analysis, namely real-time video analysis and static or pre-saved video analysis. The results of each model are displayed on a front-end interface and aggregated to provide numerical scores of nonverbal metrics. The models are evaluated on publicly available databases, and the achieved results are close to state-of-the-art. This paper proposes a new approach to address the complexity of human behavior analysis and contributes to the development of video analytics tools.

1. Introduction

The proliferation of software applications facilitating online communication has been further accelerated with the advent of the COVID-19 pandemic. Tudor [51] highlighted the global shift towards remote work, brought about by the pandemic, has resulted in an exponential increase in the adoption of web and videoconferencing Software as a Service (SaaS). In particular, there was a 90% increase in mobile SaaS downloads compared with the pre-COVID-19 weekly download average [52]. The research has concluded that interest in platforms such as Zoom and Microsoft Teams will not revert to pre-pandemic levels. As some institutions have adapted to continue operating remotely, others, particularly in the academic job market, have gone a step further to utilize SaaS platforms for seeking candidates through virtual interviews [48]. It is from this context that our interest arises in determining whether an analytical tool can aid the interviewer and assess the interviewee during a virtual interview process.

There is a growing demand for machine-based solutions to assess the nonverbal cues of candidates during video-based interviews. This is because ML techniques are believed to offer more reliable predictions of the psychological state of the candidate through the automatic analysis of their facial expression, eye gaze, and speech, as compared to human raters [23, 44]. Ekman *et al.* [12] identified six basic expressions, besides neutral, that are recognised among all humans, including anger, disgust, fear, happiness, sadness and surprise. These expressions form a foundation upon which modern researchers build datasets and models that aim to perceive and interpret discrete human emotion categories. Sebe demonstrated that using both visual and acoustic information in emotion recognition significantly improves accuracy, as opposed to relying on a single modality [41]. This finding highlights the potential benefits of combining multi-modal data to infer emotions.

While Facial Expression Recognition (FER) has been studied for many years and has experienced advancements with the advent of neural networks, most existing research and databases focus on the analysis of static images [42]. The majority of the traditional approaches for FER are predicated upon a specific task and are often designed based on a particular database, thereby displaying inadequate performance when evaluated against external datasets [54]. Additionally, the corresponding databases exhibit a homogeneous distribution of facial images with respect to affective states and lack frames that capture spontaneous human behaviour. Variations in head-pose, illumination conditions, occlusion, and registration errors have a significant impact on the performance of these algorithms [38]. Furthermore, many of these databases suffer from class imbalance. For instance, the FER2013 database contains 35,887 images, yet a mere 2% of the images are labelled as disgust [16]. The task of FER in a video setting, however, poses greater challenges as it requires additional analysis of the spatio-temporal relationships between frames to mitigate information loss in the transition of emotions across frames [19].

The existing literature on speech emotion recognition primarily focuses on analyzing isolated sentences within a lengthy and continuous audio recording. However, in practical applications, it is imperative to consider the entire recording as a single piece to prevent the loss of contextual information that may result from segmenting the speech at critical arousal points [7]. In other words, the temporal structure of speech is crucial to preserving the linguistic and emotional properties of the speech. The phonetic delivery and structure of the utterance are just as important as the emotional cues it contains [14]. Moreover, segmenting the audio may limit the ability to build real-time models, especially when emotions fluctuate throughout the utterance. This is because the processing of segmented audio may occur before the whole utterance has been pronounced, leading to inaccuracies in emotion recognition [35].

Eye movement signals can be a useful tool for understanding a candidate's behavior during an interview. By registering the candidate's eye movements, we can gain insights into their level of attention, focus, and awareness of their surroundings, providing supplementary data that can help guide emotion recognition models [26]. In particular, the duration that the candidate spends focusing on a particular spot in their environment can be indicative of what triggers psychological or emotional state changes, especially when combined with facial and acoustic analyses. However, it is important to consider the temporal relations between consecutive frames of the candidate's pupil size, fixation, and gaze direction in order to accurately determine their emotional state [20]. This is where CNNs can prove useful, as they are able to capture indentations, bulges, and other eye-region related physical distortions that are important for eye emotion classification [5].

The remainder of this paper is as follows: Section 2 provides a concise overview of the relevant literature pertaining to all four topics. Section 3 details the architectures of the neural networks employed in the research, and the overall design of the application. Section 4 presents results and demonstrate the functionality of the application. Section 5 concludes the report.

2. Related Work

Prior to the introduction of Convolutional Neural Networks (CNNs), the traditional pipeline of a facial expression recognition method consisted of three main procedures, namely face registration or acquisition, feature extraction, and classification or expression recognition [38]. During the registration stage, faces within the image are detected by means of a set of distinctive landmark points. The registration step proved crucial for improving the classification accuracy in facial detection and identification applications [37]. In the feature extraction phase, a distinction is made between two types of features; namely, geometric features and appearance features. The former represents permanent features like eyes, nose , mouth, etc. that may be distorted due to a facial expression. The latter represents features that appear temporarily such as wrinkles and bulges [45]. In essence, a feature vector is computed from the registered image using an engineered technique such as Histogram of Oriented Gradients (HOG) [8], Local Binary Patterns (LBP) [42], Scale Invariant Feature Transform (SIFT) [15], Local Phase Quantization (LPQ) [53], Principal Component Analysis (PCA) based methods [32], and Gabor Filters [27]. The feature vector is subsequently used as input to classifiers such Support Vector Machines (SVMs), Nearest Neighbours (NNs), and Multiple Kernel Learning, etc. The classifier in question attempts to identify expressions on the given face based on standard facial substructures called Action Units (AUs) [13, 45].

Recently, the advent of computational resources and the availability of diverse facial expression databases (such as Multi-PIE [17], MMI [36], extended CK+ [29], DISFA [30], GEMEP-FERA [6], SFEW [10], FER2013 [16] , AFF-Wild [31] , and AFEW [9] among others) has enabled researcher to develop CNN and DNN architectures that have the capability to learn salient features that enhance the network's classification performance. AlexNet employs a traditional architecture that consists of five convolutional layers to extract features from the input image followed by max-pooling layers and ReLUs, and three fully-connected layers. The output of the latter is then passed through a softmax function to produce a probability distribution over the 1000 class labels [22]. This network introduced a novelty; that is a "dropout" method for solving the over fitting problem [43].

FaceNet [40] is a deep CNN designed for the task of face recognition. It employs a novel triplet mining method that utilizes triplets of roughly aligned face patches. The network calculates the Euclidean distances between the embeddings of face images as a proxy for face similarity. This calculation is used to generate a dictionary-like data structure, which maps face images to their respective distances. This approach reduces the time required for lookup operations, however, it also increases the required storage space. The efficacy of FaceNet has been demonstrated through its state-of-the-art performance on the benchmark datasets, LFW and YTB, with accuracy scores of 99.63% and 95.12%, respectively. The DeepFace [47] architecture partitions the feature extraction process into two sub-tasks: face alignment and face representation. To derive the face representation, a deep neural network with nine layers and approximately 120 million parameters is utilized. The network was trained on a dataset of four million facial images belonging to roughly 4,000 individuals. The performance of the DeepFace system was evaluated on the benchmark dataset, LFW, and achieved an impressive accuracy score of 97.35%.

Szegedy *et al.* [46] introduced a novel architecture with GoogLeNet that revolves around "Inception" modules. The latest version, Inception-v3, is composed of a total of 42 convolutional and pooling layers, each of which constitutes a micro-network that contributes to the larger scheme. The network is robust in that it is designed to extract local and global features and managed to achieve state-of-the-art performance on the ILSVRC-2014 image classification task. Mollahosseini *et al.* [33, 34] made modification to the Inception layers of the GoogLeNet in order to optimise its performance on the FER task. The resulting network consisted of two convolutional layers, max-pooling, and 4 Inception layers. The network obtained state-of-the-art results on the CK+ dataset achieving a test accuracy of 93%.

In an effort to create a real-time video-based facial expression model that is lightweight and suitable for deployment on mobile devices, Savchenko *et al.* [39] proposed the use of the EfficientNet model. The model is pre-trained on the AffectNet dataset and designed to perform facial expression recognition, valence and arousal prediction, and action unit detection through the use of a frame-level emotion recognition algorithm. This algorithm allows the model to process individual frames of the video stream, rather than relying on batch processing of the entire video which makes it well-suited for applications on mobile devices. In contrast, Hasani *et al.* [19] proposed a CNN architecture that integrates 3D Inception-ResNet layers and a long short-term memory unit, with the latter being crucial in considering the spatio-temporal relationship between frames of a video. The authors also emphasized the significance of facial landmarks in their model by incorporating them as important components for learning. As a result, their model achieved state-of-the-art performance in evaluations conducted in both subject-independent and cross-database settings.

Zhao *et al.* [55] introduced a hybrid neural network architecture comprising of one-dimensional and two-dimensional convolutional neural networks, and long short-term memory (CNN LSTM) networks for speech emotion recognition. The incorporation of both CNN and LSTM allowed the researchers to overcome the limitations of each individual component. The 1D and 2D CNN LSTM networks share a similar architectural design consisting of four local feature learning blocks (LFLBs) and one LSTM layer to learn both local and global features. Each LFLB corresponds to a simple CNN of one convolutional layer followed by one max-pooling layer which effectively extract local correlations both vertically and horizontally. The LSTM layer builds on the learned local features by identifying and extrapolating long-term correlations. In public database tests, the 2D CNN LSTM network produced state-of-the-art results and outperformed traditional CNNs. Specifically, it achieved recognition accuracies of 95.33% and 95.89% on speaker-dependent and speaker-independent experiments, respectively, on the Berlin EmoDB database, an increase of approximately 3% in both experiments over the best traditional CNN. Moreover, on the IEMOCAP database, the 2D CNN LSTM improved existing CNNs by at least 12%, producing recognition accuracies of 89.16% and 52.14% on speaker-dependent and speaker-independent experiments, respectively.

Cen *et al.* [7] presented a speech emotion recognition system that bears resemblance to the approach we plan to adopt in this research. The system is capable of processing both offline and real-time continuous speech inputs, and segmenting speech intervals by utilizing a voice activity detection algorithm that discards inactive or silent intervals. The system is designed to extract features and classify emotions using a trained Support Vector Machine (SVM) model that recognizes the basic four emotional states - neutral, happy, angry, and sad. The authors conducted experimental evaluations that demonstrated classification accuracies averaging 90% and 78.78% on pre-recorded and real-time databases, respectively.

Tripathi and colleagues [50] aimed to perform multimodal emotion recognition using data from speech, text, and facial expressions, rotational motions, and hand gestures. They introduced four speech models, inspired by the works of Tashive *et al.* [18, 24]. The first model, influenced by [18], adopted an MLP-based architecture that processes a flattened 750-dimensional input feature vector through three fully connected layers with 1024, 512, and 256 hidden neurons, respectively. A ReLU activation layer followed by four output neurons and a Softmax layer were added to the end of the network. The second model, influenced by [24], divided each voice activity region into small segments, extracting 32 features with 12-dimensional Mel-frequency cepstral coefficients (MFCC). Two LSTM layers with 512 and 256 neurons followed by a Dense layer with 512 neurons and a ReLU activation function were trained. Models 3 and 4 improved upon the second model by incorporating an LSTM with attention-focused properties and a bidirectional LSTM, respectively. Adadelta was used as the optimizer for all four models. The performance of all models was evaluated on the IEMOCAP dataset. The inclusion of bidirectional attention-focused LSTM significantly improved the accuracy, with models 3 and 4 achieving classification accuracies of 54.15% and 55.65%, respectively.

3. Proposed Method

Designing complex neural architectures, whether by increasing the number of neurons or adding additional layers, is computationally intensive and can result in over-fitting of the training data, leading to a decline in the model's predictive capabilities [25]. In my previous attempts to build such networks, higher accuracy rates were accompanied by a worse run-time performance of the application, posing a serious challenge to improving the user experience. While accuracy rates of the models are crucial to the interpretive capabilities of the application, the bottleneck of the application's run-time performance lies in the architectural design of the networks as well as the code-base.

To tackle the code-base issue, I employed a structure that prioritized generalizability and abstraction. I constructed a unique and independent client class for each of the four models with appropriate methods. Models with neural networks were trained and validated on their respective datasets, and a copy of the trained model was saved for future use. For real-time or static analysis, the entry point of the application used the CV2 library to sequentially consume image frames from either live webcam or a pre-saved video. Although a 15-second video generates 350 frames on average, attempting to discard redundant data in real-time to select the most expressive image or audio snippet from neighboring clusters has proven computationally costly. Moreover, using a simple correlation matrix to detect the most important piece of data resulted in significant information loss. Each image frame was then processed and resized to fit the input requirements of the receiving model before being passed to all four models. The main working pipeline of the application should not be interrupted while waiting for the returned values of the models. In addition, the application would significantly slow down if each model was run sequentially before aggregating the returned values and passing them to be displayed on the front-end. Thus, I opted for a fail-safe parallel threading design choice, where a call to classify a piece of data spawns a new separate thread that completes its task and returns the result to the main thread without interrupting the flow of the main application. Both real-time and static modes of analysis had their independent client classes that inherit all four models.

Regarding the architectural design of the neural networks, a trade-off between accuracy and performance was necessary, particularly in the real-time mode. After experimenting with several models, the final architecture of each of the four models described in their respective subsections resulted in a significant increase in performance speed while maintaining near state-of-the-art accuracy. Each model was trained for 100 epochs.

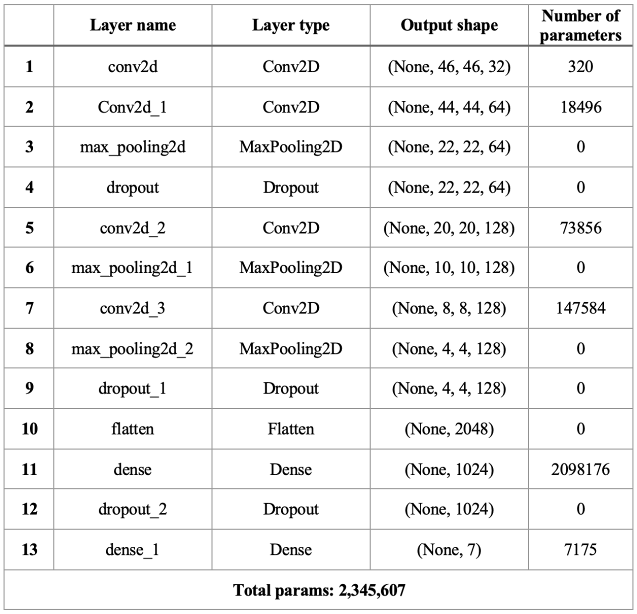
3.1. Facial Expression Recognition

This subsection outlines the network architecture of the FER task, which is heavily influenced by the GoogLeNet and AlexNet architectures discussed in Section 2. The input to the network is an image with dimensions of $(48 \times 48 \times 1)$. The network starts with two modules of two-dimensional convolutional neural networks. Each module consists of a convolution layer followed by a max pooling layer. The rectified linear unit (ReLU) activation function is chosen due to its ability to prevent the vanishing gradient problem caused by other activation functions [21]. The ReLU activation function is defined as

$$f(n) = max(0, n)$$

A Dropout layer with a rate of 0.25 is utilized to prevent over-fitting. Figure 1 provides an overview of the entire network architecture.

Figure 1. A summary of the network architecture used for the FER task

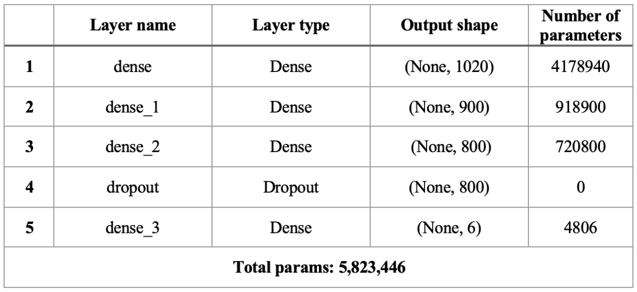


	Layer name	Layer type	Output shape	Number of parameters
1	conv2d	Conv2D	(None, 46, 46, 32)	320
2	Conv2d_1	Conv2D	(None, 44, 44, 64)	18496
3	max_pooling2d	MaxPooling2D	(None, 22, 22, 64)	0
4	dropout	Dropout	(None, 22, 22, 64)	0
5	conv2d_2	Conv2D	(None, 20, 20, 128)	73856
6	max_pooling2d_1	MaxPooling2D	(None, 10, 10, 128)	0
7	conv2d_3	Conv2D	(None, 8, 8, 128)	147584
8	max_pooling2d_2	MaxPooling2D	(None, 4, 4, 128)	0
9	dropout_1	Dropout	(None, 4, 4, 128)	0
10	flatten	Flatten	(None, 2048)	0
11	dense	Dense	(None, 1024)	2098176
12	dropout_2	Dropout	(None, 1024)	0
13	dense_1	Dense	(None, 7)	7175
Total params: 2,345,607				

3.2. Eye Emotion Recognition

The EER task of this research involves the challenging process of extracting features from the region around the eye that are indicative of human emotions. Therefore, to overcome this challenge, a pre-processing step is required to prepare the input. The input is a $(350 \times 350 \times 3)$ image which is processed using two advanced neural networks, namely InceptionV3 and Xception. Both these models are initialised with weights from pre-training on ImageNet, and global average pooling is employed to down-sample the input along its spatial dimensions. This process is designed to extract intricate features from the images of the dataset that will be used to train the model. The model is compiled using an Adam optimiser with a 0.01% learning rate. Notably, the total number of parameters is highest in this model, which is not surprising given the complexity of the task. Figure 2 shows a summary of the entire network architecture.

Figure 2. A summary of the network architecture used for the Eye Emotion task



	Layer name	Layer type	Output shape	Number of parameters
1	dense	Dense	(None, 1020)	4178940
2	dense_1	Dense	(None, 900)	918900
3	dense_2	Dense	(None, 800)	720800
4	dropout	Dropout	(None, 800)	0
5	dense_3	Dense	(None, 6)	4806
Total params: 5,823,446				

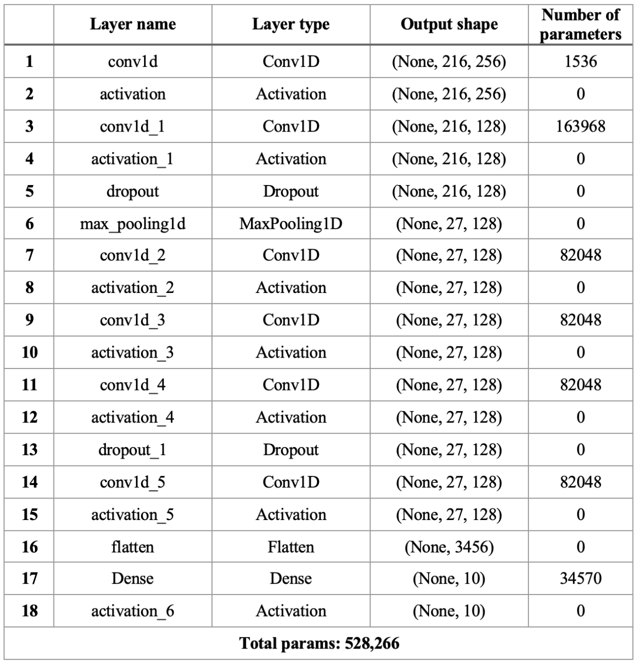
3.3. Speech Emotion Recognition

In the proposed architecture of the SER task, the network employs a Rectified Linear Unit (ReLU) activation function concatenated between almost every other layer. Due to the complexity of the network, we chose to use the RMSprop optimiser instead of Adam in compiling the model. The RMSprop optimiser is well-known for its faster convergence speed in solving optimisation problems [11]. With plain momentum, it restricts oscillations in the vertical direction, which allows us to increase the learning rate and take larger steps in the horizontal direction, thereby achieving faster convergence [49]. Given that the application needs to process a continuous audio stream, the model needs to perform quickly, making RMSprop an optimal choice of optimiser. The final layer of the network employs a Softmax activation function, given the multi-class nature of audio classification. By computing class probabilities, we can gain insights into the importance of the second most significant emotion, which can inform certain generic metrics. Figure 3 summarises the entire network architecture.

3.4. Eye Gaze Tracking

The eye gaze tracking task does not attempt to solve a classification nor regression problem. As such, the client does not require the construction of a convolutional neural network (CNN). Instead, we have leveraged a Python library that provides a webcam-based eye tracking system to address this challenge. The library provides precise information on the position of the pupils and gaze direction in real-time, by using CV2 to stream input from the webcam. It processes each image frame from the video stream, calibrates, locates and isolates both eyes, pupils, and iris, and tracks their movements in a two-dimensional Cartesian plane. This allows us to collect information about the candidate's gaze, as the client provides live coordinates of each pupil. Additionally, the client includes specific methods to help determine whether the client is looking left, right, up, down, or center, providing additional information on the candidate's eye movements.

Figure 3. A summary of the network architecture used for the Speech Emotion task



	Layer name	Layer type	Output shape	Number of parameters
1	conv1d	Conv1D	(None, 216, 256)	1536
2	activation	Activation	(None, 216, 256)	0
3	conv1d_1	Conv1D	(None, 216, 128)	163968
4	activation_1	Activation	(None, 216, 128)	0
5	dropout	Dropout	(None, 216, 128)	0
6	max_pooling1d	MaxPooling1D	(None, 27, 128)	0
7	conv1d_2	Conv1D	(None, 27, 128)	82048
8	activation_2	Activation	(None, 27, 128)	0
9	conv1d_3	Conv1D	(None, 27, 128)	82048
10	activation_3	Activation	(None, 27, 128)	0
11	conv1d_4	Conv1D	(None, 27, 128)	82048
12	activation_4	Activation	(None, 27, 128)	0
13	dropout_1	Dropout	(None, 27, 128)	0
14	conv1d_5	Conv1D	(None, 27, 128)	82048
15	activation_5	Activation	(None, 27, 128)	0
16	flatten	Flatten	(None, 3456)	0
17	Dense	Dense	(None, 10)	34570
18	activation_6	Activation	(None, 10)	0
Total params: 528,266				

4. Experimental Results

In this section, we provide a concise review of the databases employed in evaluating the effectiveness of our proposed method. Subsequently, we present the results of our experiments, obtained from using these databases, and compare them with the state of the art. It is pertinent to note that all models used in the experiments were trained for 100 epochs with a batch size of 64 on a standard MacBook Pro with a 1.4 GHz Quad-Core Intel Core i5 processor.

4.1. Databses

We have chosen to employ The Facial Expression Recognition 2013 database to train our facial expression recognition model. This particular database was initially introduced in the ICML 2013 Challenges in Representation Learning conference [1]. The FER-2013 database consists of images of faces that were programmatically pulled and recorded using the Google image search API. These images were then standardised in two steps. Firstly, the images are converted into $(48 \times 48 \times 1)$ pixel gray-scale images. Secondly, the faces in the images were positioned in the center such that the face to image ratio is somewhat constant across the database. A sample of four images with different labels is shown in figure 4. The model is trained to categorize each face based on the emotion depicted in the facial expression into one of seven categories Angry (0), Disgust (1), Fear (2), Happy (3), Sad (4), Surprise (5), Neutral (6). The resulting database comprises 35,878 images, with the majority of them being captured in a natural and uncontrolled environment. The distribution of emotions in both the training and testing data is shown in *Figures 6 and 7* respectively.

To train the eye emotion detection model, we have opted to employ the Eye Emotion Dataset DIU as described in [2]. The dataset composition primarily involves images from the six fundamental, universal emotions, namely happiness, disgust, sadness, fear, anger, and surprise. The dataset in question comprises 408 distinct double eye images that possess varying resolutions and sizes. A sample of four images with different labels is shown in figure 5. The distribution of emotions, concerning both the training and testing data, is graphically represented in *Figures 8 and 9* respectively. All the images feature the eyes and their surrounding areas, specifically the upper and lower eyelids, glabella, and brow, which are believed to be integral in detecting emotional states. The common features prevalent throughout the entire image under different emotional states, in conjunction with data regarding the gaze direction of the iris and pupils, significantly facilitate a Convolutional Neural Network (CNN) classification task.

For the speech emotion task, we have decided to use the RAVDESS database to train our speech emotion detection model [3, 28]. This database contains a total of 7356 files comprising a multitude of audio, video and visual data files. Particularly, there are 1440 speech audio files that are solely comprised of audio data (16 bit, 48kHz .wav). The naming configuration is standardised such that each filename comprises a 7-sections encoder (*e.g., 03-01-03-01-02-02-09.wav*) where each part is indicative of a particular characteristic. Although all parts are important during the pre-processing step, the 7^{th} character is the most important for the model because it points to the corresponding emotion according to the mapping defined below. The dataset involves 24 professional actors equally distributed between males and females, uttering two short, matched statements in a North American accent. Each speech sample is classified into one of six fundamental emotional categories, namely calm (0), happy (1), sad (2), angry (3), fearful (4), surprise (5), and disgust (6). Furthermore, each sentence is vocalised at two distinct tones of emotional intensity, namely normal and intense, along with an additional neutral expression.

Figure 4. A sample of four images with different labels from the FER Database

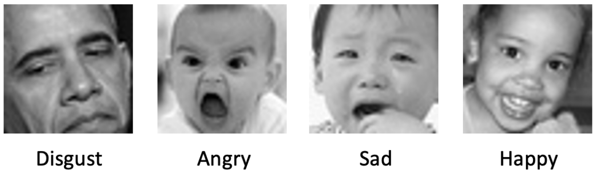
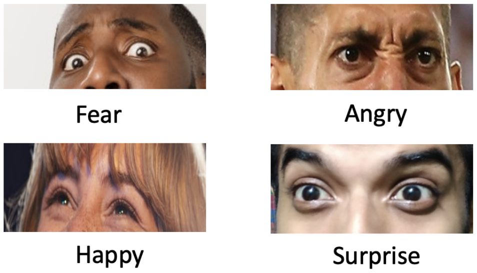


Figure 5. A sample of four images with different labels from the DIU Database



4.2. Procedures and Results

This subsection discusses the technical details of the application. The main script, which is the entry point for both real-time and static video processing, operates through a while loop, whereby each frame of the live or pre-recorded video is processed until the end. To store the classification results for each model globally while the application is running, a unique hash map is initiated for each model. For instance, the FER hash map is initiated with all corresponding emotions as keys and 0's as values. These hash maps are updated after each classification, and their values are made accessible to the front-end for display through a dedicated endpoint in the Flask app.

For the FER task, the image frame is resized to $(1280 \times 720 \times 1)$ and converted to grayscale. This modified image frame is then used as input for the Haar feature-based cascade classifiers to detect the candidate's face. The gray image frame is cropped around the Region of Interest (ROI), resized to $(48 \times 48 \times 1)$, and classified using the trained FER CNN. The emotion with the highest probability is stored, and its corresponding counter in the FER hash map is incremented. On the other hand, for the eye gaze task, the raw image frame is passed directly to the eye gaze client, which records the 2D coordinates of the gaze. Due to the infinite number of combinations of coordinates in a continuous 2D plane, the plane is thus divided into nine discrete areas, and the corresponding counter value in the eye gaze hash map is incremented based on the coordinates' location.

Figure 6. Distribution of the six emotions in the training set of the Facial Expression Emotion Dataset FER2013

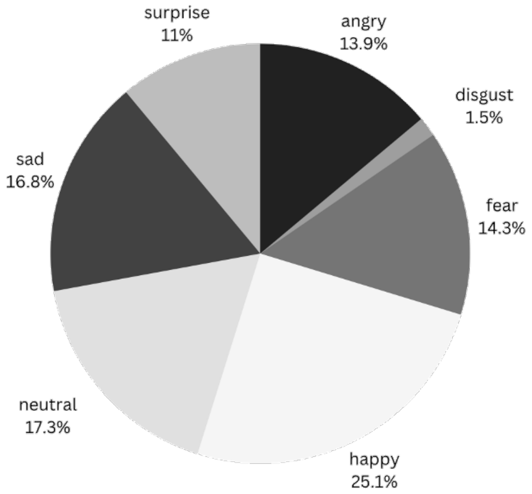
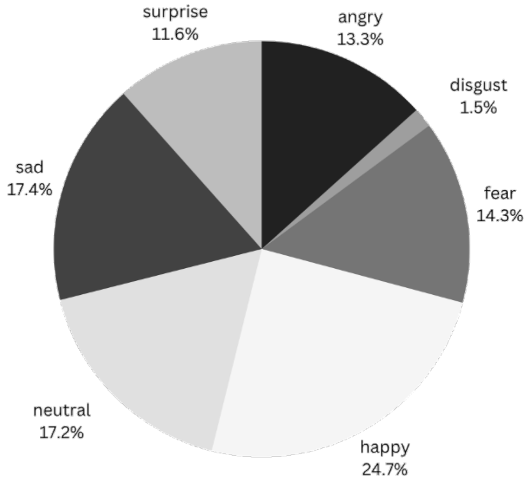


Figure 7. Distribution of the six emotions in the testing set of the Facial Expression Emotion Dataset FER2013



In accordance with the database's architecture employed in the task of detecting speech emotion, the initial phase involves the extraction of features from the audio files, which will facilitate the learning process of our model. To carry out this feature extraction, we utilize the LibROSA library. During this process, all audio files have been temporally aligned to a duration of 3 seconds, thereby ensuring an equivalent number of features across all files. Moreover, the sampling rate of each file has been doubled while maintaining a constant sampling frequency, enabling the generation of a greater number of features for classifying audio files when the size of the input file is small.

Figure 8. Distribution of the six emotions in the training set of the Eye Emotion Dataset DIU

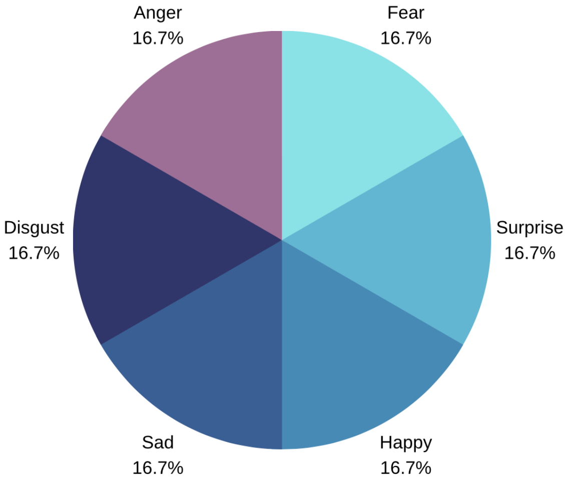
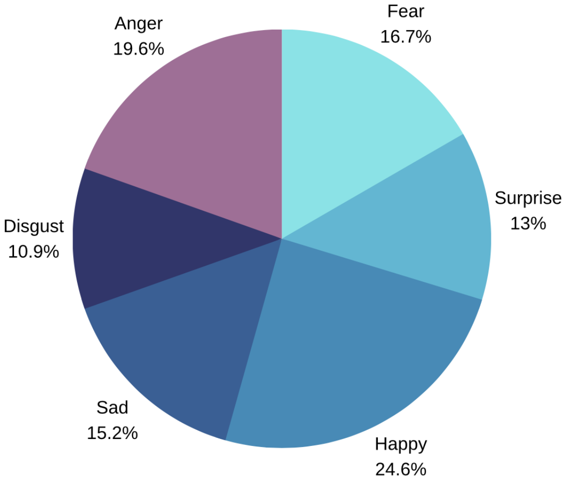


Figure 9. Distribution of the six emotions in the testing set of the Eye Emotion Dataset DIU



4.2.1 Real Time

During the implementation phase, it was challenging to conduct speech and eye emotion analyses in parallel with the other models, as the machine used had limited processing capacity. To address this, we decided to record audio and capture image frames of the face while the interview is ongoing, and defer the speech and eye emotion analyses until after the interview. However, this presented a challenge when generating results in real-time mode. In the context of a live interview, audio must be recorded separately since CV2 does not offer an API to access video and audio data simultaneously. To address this, we created a script using the PyAudio library to record the candidate's speech every three seconds in a separate thread, thus avoiding interrupting the main application.

To extract features from the recorded audio files, we used the Librosa library, which automatically resampled the audio files to a given rate of $sr = 22050 \times 2$ and extracted Mel-frequency cepstral coefficients (MFCCs). The extracted features were then flattened and passed to the pre-trained speech emotion model for classification. The emotion with the highest probability was stored and decoded, and its corresponding counter in the audio hash map was incremented.

In parallel, the eye emotion analysis was conducted using a separate thread. Every saved image frame was resized to (350×350), as the sizes of the images across the database were not uniformly distributed. All the images were stored in a list, which was then converted to a numpy array and passed to the pre-trained InceptionV3 and Xception models for feature extraction. The extracted features were then passed onto the eye emotion model for classification. The emotion with the highest probability was stored and decoded, and its corresponding counter in the eye emotion hash map was incremented.

4.2.2 Static

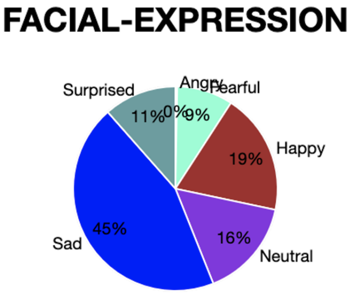
The static client pipeline is relatively straightforward. Users are provided with a dedicated button on the front-end to upload a pre-recorded video from their local machine into the server through a POST request. After checking the extension of the video file for eligibility, it is saved on the server for further analysis. The pre-processing procedure for the static client is executed in two steps. In the first step, the video is processed frame-by-frame using the CV2 library, and the region of interest, i.e., the face, is detected and saved onto a designated directory on the server. Next, the video is converted into a WAV audio file, since the CV2 library doesn't provide an API to access video and audio data simultaneously. In the second step, the audio file is split into smaller, non-overlapping audio snippets of length 3 seconds, which can then be fed into the speech emotion model for classification. The subsequent procedures follow those in the real-time case, which have been explained earlier. Ultimately, the corresponding emotion labels are stored in the hash maps and the results are presented to the user.

4.2.3 Results

Each one of the four hash maps introduced earlier is then converted to a pie chart. As such, each hash map operates as a store of the count of the number of occurrence of each emotion in their respective task. Figure 10 shows an example of the facial expression recognition pie chart from running the application on a test video interview. In this example, it can be interpreted that the candidate was sad 45% of the time.

In the preceding sections, we introduced four hash maps, each of which serves as a repository for the count of the number of occurrences of the emotions in their respective tasks. These hash maps are subsequently transformed into pie charts, such that each chart represents the distribution of emotions across the task. An example of a facial expression recognition pie chart is depicted in Figure 10, which shows that the frequency of sadness displayed by a candidate during a test video interview is 45%.

Figure 10. A pie chart display of the FER analysis of the candidate in a test interview



In addition, the application computes several nonverbal metrics that are pertinent to a candidate's performance during an interview. These metrics, namely attentiveness, deep thinking, confidence, and potential lie, are graphically represented on a bar chart displayed on the front-end. See Figure 11 for reference. The computation of each metric is based on the frequency of occurrence of the relevant emotions in the global hash maps. For instance, the attentiveness score is calculated based on the total duration of direct eye contact with the webcam, while discounting any prolonged periods where the candidate looks away from the center. The potential lie score is determined by detecting a prolonged disturbance in the candidate's voice that appears after being calm or happy, accompanied by any facial expression or eye emotion other than neutral or happy. In the case of deep thinking score, it is recorded only when a surprise signal is received following a question, and the duration that the candidate looked away from the center after that. For the confidence score, consistency in the candidate's voice while speaking (i.e. maintaining a calm, happy voice while answering) is detected, while discounting any undesirable emotions (e.g. disgust, anger) from facial expression or eye emotion.

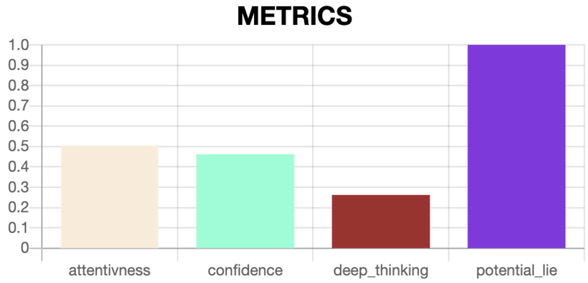
To evaluate the effectiveness of our system, we compared the accuracy rates of the individual simplified models against their corresponding state-of-the-art models in Table 1. For testing, we randomly sampled test data and evaluated

	FER2013	DIU	RAVDESS
Our Model	74.78	72.76	89.45
State-of-the-art	76.82 [2]	77.54 [1]	92.08 [4]

Table 1. Recognition Rates (%)

the accuracy of the models. The process is repeated several times and an average is taken.

Figure 11. A bar chart display of the nonverbal metrics of the candidate in a test interview



5. Conclusion

In this project, we presented an application that allows us to gain valuable insights into a candidate's behaviour and emotional state during an interview setting. The application is based on three concurrent neural network clients, supported by a gaze detector clients. These clients enable the recognition of facial expressions, eye emotions, and speech emotions. Additionally, the application aggregates information to provide numerical scores of nonverbal cues that are instrumental in evaluating the candidate's suitability for the position in question. These metrics, such as attentiveness, deep thinking, confidence, and potential lie, provide a comprehensive picture of the candidate's nonverbal behavior during the interview process.

The results indicate that our models have achieved a relatively high level of accuracy compared to state-of-the-art models while maintaining an adequate user experience. Overall, the application devised in this study represents a significant step forward in the field of interview evaluations and has the potential to improve the overall efficacy of the interview process.

References

[1] Challenges in representation learning: Facial expression recognition challenge. http://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge. Accessed: 2023-02-22. 5, 9

[2] Eye emotion dataset diu. https://www.kaggle.com/datasets/mdnymurrahmanshuvo/eye-emotion-dataset-diu. Accessed: 2023-03-08. 6, 9

[3] Ravdess emotional speech audio. https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio. Accessed: 2023-02-28. 6

[4] Temporal modeling matters: A novel temporal emotional modeling approach for speech emotion recognition. https://paperswithcode.com/paper/temporal-modeling-matters-a-novel-temporal. Accessed: 2023-04-28. 9

[5] Claudio Aracena, Sebastián Basterrech, Vaclav Snasel, and Juan Velasquez. Neural networks for emotion recognition based on eye tracking data. pages 2632–2637, 10 2015. 2

[6] Tanja Bänziger, Marcello Mortillaro, and Klaus Scherer. Introducing the geneva multimodal expression corpus for experimental research on emotion perception. *Emotion (Washington, D.C.)*, 12:1161–79, 11 2011. 2

[7] Ling Cen, Fei Wu, Zhu Yu, and Fengye Hu. *A Real-Time Speech Emotion Recognition System and its Application in Online Learning*, pages 27–46. 12 2016. 2, 3

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. 2

[9] Abhinav Dhall, Roland Goecke, Jyoti Joshi, Michael Wagner, and Tom Gedeon. Emotion recognition in the wild challenge 2013. pages 509–516, 12 2013. 2

[10] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2106–2112, 2011. 2

[11] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, pages 92–99, 2018. 5

[12] Paul Ekman and W V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17 2:124–9, 1971. 1

[13] Paul Ekman and Wallace V Friesen. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*, 1978. 2

[14] Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011. 2

[15] Cong Geng and Xudong Jiang. Face recognition using sift features. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3313–3316, 2009. 2

[16] Ian Goodfellow, Dumitru Erhan, Pierre Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov,

John Park, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64, 07 2013. 2

[17] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and S. Baker. Multi-pie. 12 2013. 2

[18] Kun Han, Dong Yu, and Ivan Tashev. Speech emotion recognition using deep neural network and extreme learning machine. 09 2014. 3

[19] Behzad Hasani and Mohammad Mahoor. Facial expression recognition using enhanced deep 3d convolutional neural networks. pages 2278–2288, 07 2017. 2, 3

[20] Eckhard H. Hess and James M. Polt. Pupil size as related to interest value of visual stimuli. *Science*, 132(3423):349–350, 1960. 2

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012. 4

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 2

[23] Markus Langer, Cornelius König, and Kevin Krause. Examining digital interviews for personnel selection: Applicant reactions and interviewer ratings. *International Journal of Selection and Assessment*, 25, 12 2017. 1

[24] Jinkyu Lee and Ivan Tashev. High-level feature representation using recurrent neural network for speech emotion recognition. 09 2015. 3

[25] Haidong Li, Jiongcheng Li, Xiaoming Guan, Binghao Liang, Yuting Lai, and Xinglong Luo. Research on overfitting of deep learning. In *2019 15th International Conference on Computational Intelligence and Security (CIS)*, pages 78–81, 2019. 4

[26] Jia Zheng Lim, James Mountstephens, and Jason Teo. Emotion recognition using eye-tracking: Taxonomy, review and current challenges. *Sensors*, 20(8), 2020. 2

[27] Chengjun Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–476, 2002. 2

[28] Steven R Livingstone and Frank A. Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. volume PLoS ONE 13(5): e0196391, 2018. 6

[29] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101, 2010. 2

[30] S. Mohammad Mavadati, Mohammad H. Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F. Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013. 2

[31] Dimitrios Kollias Mengyao Liu. Aff-wild database and affwildnet. *CoRR*, abs/1910.05318, 2019. 2

[32] Mohammad Reza Mohammadi, Emad Fatemizadeh, and Mohammad Mahoor. Pca-based dictionary building for accurate facial expression recognition via sparse representation. *Journal of Visual Communication and Image Representation*, 25:1082–1092, 07 2014. 2

[33] Ali Mollahosseini, David Chan, and Mohammad H. Mahoor. Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10, 2016. 3

[34] Ali Mollahosseini, Behzad Hassani, Michelle J. Salvador, Hojjat Abdollahi, David Chan, and Mohammad H. Mahoor. Facial expression recognition from world wild web. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1509–1516, 2016. 3

[35] Albino Nogueiras, Asunción Moreno, Antonio Bonafonte, and José Mariño. Speech emotion recognition using hidden markov models. pages 2679–2682, 09 2001. 2

[36] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *2005 IEEE International Conference on Multimedia and Expo*, pages 5 pp.–, 2005. 2

[37] Elias Rentzeperis, Andreas Stergiou, Aristodemos Pnevmatikakis, and Lazaros Polymenakos. Impact of face registration errors on recognition. volume 204, pages 187–194, 06 2006. 2

[38] Evangelos Sariyanidi, Hatice Gunes, and Andrea Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1113–1133, 2015. 2

[39] Andrey V. Savchenko. Video-based frame-level facial analysis of affective behavior on mobile devices using efficientnets. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2358–2365, 2022. 3

[40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 2

[41] N. Sebe, I. Cohen, T. Gevers, and T.S. Huang. Emotion recognition based on joint visual and audio cues. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 1136–1139, 2006. 1

[42] Caifeng Shan, Shaogang Gong, and Peter Mcowan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27:803–816, 05 2009. 1, 2

[43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 2

[44] Hung-Yue Suen, Kuo-En Hung, and Chien-Liang Lin. Tensorflow-based automatic personality recognition used in asynchronous video interviews. *IEEE Access*, PP:1–1, 03 2019. 1

[45] CP Sumathi, T Santhanam, and M Mahadevi. Automatic facial expression analysis a survey. *International Journal of Computer Science and Engineering Survey*, 3(6):47, 2012. 2

[46] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 3

[47] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 2

[48] Christina M. Termini, Florentine U.N. Rutaganira, Caroline B. Palavicino-Maggio, Chelsey C. Spriggs, Chantell S. Evans, and Melanie R. McReynolds. Using virtual interviewing to create a more accessible hybrid academic job market. *Cell*, 184(26):6217–6221, 2021. 1

[49] Milan Tripathi. Facial emotion recognition using convolutional neural network. *ICTACT Journal on Image and Video Processing*, 12:2531–2536, 08 2021. 5

[50] Samarth Tripathi and Homayoon S. M. Beigi. Multi-modal emotion recognition on iemocap with neural networks. 2018. 3

[51] Cristiana Tudor. The impact of the covid-19 pandemic on the global web and video conferencing saas market. *Electronics*, 11:2633, 08 2022. 1

[52] Sherry Wang and Marilyn Roubidoux. Covid-19, videoconferencing, and gender. *Journal of the American College of Radiology*, 17, 05 2020. 1

[53] Zhen Wang and Zilu Ying. Facial expression recognition based on local phase quantization and sparse representation. In *2012 8th International Conference on Natural Computation*, pages 222–225, 2012. 2

[54] Marcus Zavarez, Rodrigo Berriel, and Thiago Oliveira-Santos. Cross-database facial expression recognition based on fine-tuned deep convolutional network. pages 405–412, 10 2017. 1

[55] Jianfeng Zhao, Xia Mao, and Lijiang Chen. Speech emotion recognition using deep 1d 2d cnn lstm networks. *Biomedical Signal Processing and Control*, 47:312–323, 2019. 3