Artificial Intelligence System for Emotion Recognition and Text Analytics

Namrata Chaudhari, Reshu Agarwal, Meghna Narwade, Xinrui Yu, and Jafar Saniie
Embedded Computing and Signal Processing Research Laboratory (http://ecasp.ece.iit.edu/)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, IL, U.S.A.

***Abstract*— Harnessing the power of emotional intelligence by analyzing a person's behavioral and linguistic skills can help humans improve their approach to social interactions. In this paper, we propose an artificial intelligence-based stand-alone system that will allow us to classify and analyze facial expressions in real-time and perform sentiment analysis by examining the body of the text (extracted from audio) to understand the opinion expressed by it. This helps us provide a deeper understanding of how humans really feel at a given time. The proposed system uses a deep neural network (DNN) for classifying eight basic emotions based on features extracted from facial expressions and uses pre-trained sentiment analysis tools to quantify text (extracted from audio) based on polarity. The system is implemented by extracting the audio and visual cues from real-time scenarios and using these extracted cues to perform facial expression recognition and text sentiment analysis. The integrated system extracts video and audio simultaneously with a frame rate of 4-5 fps. The facial emotion detection system successfully detects facial expressions of faces detected in real-time video with an accuracy of about 86.75%. The speech extracted is converted to text, cleaned, and processed to determine if the attitude of the speaker in a given situation is positive, negative, or neutral.**

Keywords—Facial expression detection, Text sentiment analysis, Deep Neural Network, Feature Classification

I. INTRODUCTION

Effective communication involves two components: Verbal cues and Nonverbal cues. The most prominent and effective way of communicating non-verbal cues is facial expressions. Humans share a universal set of fundamental emotions and over the past decade, numerous systems have been proposed for recognizing these emotions [1]. For a detection approach, it is important to have a taxonomic reference for classifying the eight basic emotions which consist of anger, contempt, disgust, fear, happiness, sadness, surprise as well as neutral. The analysis of sentiments behind the verbal cues allows us to get an insight into the emotion and tone of the conveyed message. Sentiment analysis (opinion mining) is a text mining technique that uses machine learning and natural language processing (NLP) to automatically analyze text for the sentiment of the speaker (positive, negative, neutral). Text sentiment can be classified using machine learning models like Support Vector Machine (SVM), Naive Bayes, and Decision Tree.[2]. Using unsupervised lexicon-based approaches. Determining polarity of text using pre-trained sentiment analysis tools from various Python NLP libraries (TextBlob, Vader) [10,11]. In this paper, we research unsupervised lexicon-based approaches to implement text sentiment analysis. In the proposed system, text sentiment analysis is performed on the extracted real-time audio which is converted to text. Section II presented the related works; details about our methodology and procedure, including facial landmark vectorization, deep neural network for real-time emotion recognition, and the text sentiment analytics system are introduced in Sections III to VII; results and conclusions are covered in Sections VIII and IX.

II. RELATED WORKS

A. Facial Emotion Detection using Landmark Extraction

In recent years, advances in facial expression detection have accelerated, and more and more experts have been involved in the development of emotion recognition. The research of expression recognition in computer vision focuses on feature extraction and feature classification [3,4,14,15]. Feature extraction refers to extracting landmarks from faces that can be used for classification from input pictures or video streams. There are multiple methods for feature extraction from detected faces. Facial expression classification refers to the use of specific algorithms to identify the categories of facial expressions according to the extracted features. Commonly used methods of facial expression classification are Hidden Markov Model (HMM), Support Vector Machine (SVM), AdaBoost, and Artificial Neural Networks (ANN).[16]

B. Text Sentiment Analysis

The development of a text sentiment analysis system has many problems. The first step is to determine the text's content. Due to the nature of language, which contains a considerable degree of semantic complexities not found in other sorts of data, this is not an easy process. Second, emotions must be categorized in some way to establish their orientation. There are several approaches to dealing with this issue.[17] Sentiment classification can traditionally be done in two ways: supervised and unsupervised based on semantics. Support vector machines and the naive Bayes classifier are the most commonly used supervised techniques. [18] Machine Learning involves building classification models from a document corpus, where each document can be represented as a bag of words. [18] It is typical to employ stemming and stop word elimination techniques. In general, classifiers that perform well in the domain where they were trained, do not perform well in other domains since they are largely dependent on the training data.

Unsupervised semantics-based methods employ dictionaries to classify different types of words based on their sentiment polarity. Unlike traditional machine learning methods, semantics-based unsupervised methods are more domain-specific, with performance varying from domain to domain. There are two major sub-categories to consider: Lexicon and corpus-based. The dictionary-based technique uses a set of initial terms usually manually collected. An example of this type of dictionary is WordNet, which was used for developing SentiWordNet [18].

III. Methodology and Procedure

A. Methodology and Procedure of Emotion Recognition and Text analysis System Interface Design

In this paper, we propose a system that uses a camera to extract visual cues which are used to perform facial expression recognition and uses a microphone to extract audio cues which are converted to text and used to perform text sentiment analysis. The interface can be implemented using a laptop PC. The workflow of the proposed model can be found in Fig. 1.

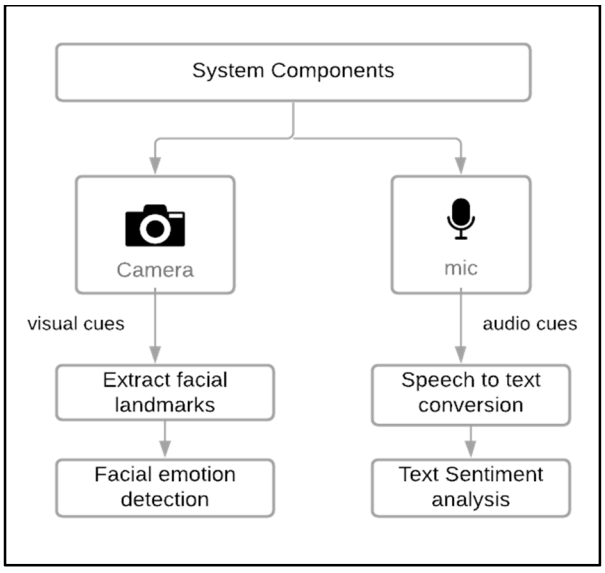

Fig. 1. Workflow for emotion detection and sentiment analysis

We need to extract the generated audio and visual cues simultaneously from a real-time scenario. The proposed system implements this using multi-threading which helps us to run multiple functions calls simultaneously i.e., one thread records the video using OpenCV and the other thread records the audio using Pyaudio and the output of each of these threads will then be served as an input to the two modules implemented which will then predict emotions and analyze the polarity of the content obtained from the audio. The frame rate for the multithreading process is calculated by: dividing the total number of frames by the elapsed time of the program & the fps recorded was about 4-5fps.

B. Implementation of the Facial Emotion Recognition System.

The facial emotion detection module is built from scratch to detect one of eight emotions: happiness, sadness, anger, surprise, fear, disgust, and contempt [1], The visual cues are used to detect faces and extract 68 landmarks (features) which are then fed to the deep neural network (DNN) to classify emotion from the given frame [19]. Facial emotion recognition consists of two parts (i) Image processing that extracts facial landmarks. (ii) Neural network for emotion recognition.

IV. Vectorization of Facial Landmark For Feature Extraction

Convolutional neural networks can be used to classify raw input images but performing feature landmark extraction allows us to achieve comparable results with a simpler neural network [20].

Facial landmark extraction is performed using the Dlib library in python. The extracted features are then fed as an input to the neural network. The Dlib library detects faces from the input image and uses the predictor function to place 68 landmarks on the detected faces. It uses Histogram of Oriented Gradients (HOG) for Object Detection with a linear classifier, an image pyramid, and a sliding window detection scheme to detect faces in an image. Once the region of the face is determined, facial landmarks will be detected using One Millisecond Face Alignment with an Ensemble of Regression Trees [4]. The Dlib library accurately detects landmarks from the detected faces at an angle of -30 to +30 degrees in any direction.

The coordinates of the 68 landmarks have a fixed orientation (shown in Fig. 2). The resultant landmarks are given in the form of an array.

Resultant array: = $[(x0, y0), (x1, y1), \ldots\ldots, (x67, y67)]$

Fig. 2. Facial Landmarks

Extracting features from faces allows us to construct a simple neural network with less training data which will converge faster as compared to traditional CNNs. Neural Networks perform best when the feature vector is scaled to a small range of values [-1, 1]. To optimize the gradient descent process, normalize the facial landmarks and align them at the tip of the nose $(x33, y33)$. Vectorization of facial landmarks is achieved by putting tensors of 2-dimensional coordinates into a vector which is fed into the neural network.

Shifting the origin to the tip of the nose $(x33, y33)$:
For (x, y) in the resultant array:

$$x = x - x33$$
$$y = y - y33$$

Normalizing the coordinates in range [-1, 1]:
scale height = y8 // coordinate $(x8, y8)$:= (*, -1)
scale width = max $(|x0|, |x16|)$
For (x, y) in resultant array:

$$x = x/\text{ scale width}$$
$$y = y/\text{ scale height}$$

The normalized coordinates are stored in the form of a feature vector.

$$[(x0, y0), (x1, y1), \dots, (x67, y67)]$$
$$\rightarrow [x0, y0, x1, y1, \dots, x67, y67]$$

The result data can be stored in a CSV file with an integer indicating the emotion in the last column (label L) which can be used to train and test the neural network.

V. Deep Nuearl Network (DNN) For Emotion Recognition

The dataset was created using images from CK+ (Extended Cohn-Kanade dataset), JAFFE dataset, TFEID (Taiwanese Facial Expression Image Database), and RaFD (Radboud Faces Database) [21,22,23]. The created dataset is composed of eight classes with a total of 3000 images divided into training and test sets. The vectorized facial landmarks of images from the dataset are stored in a CSV file along with an integer indicating the emotion. The test and train CSV files are then used to train and evaluate the DNN. The flowchart of facial emotion detection can be found in Fig. 3.

The model used in building the deep neural network is a sequential model with three hidden layers. The type of layer used is dense which implies that every neuron in the dense layer receives input from all neurons of the previous layer (refer to Fig. 4). The activation function used was a sigmoid [24]. Adam optimizer allows the framework to adjust the step size depending on the loss [25]. The accuracy obtained after testing the model is 86.75%.

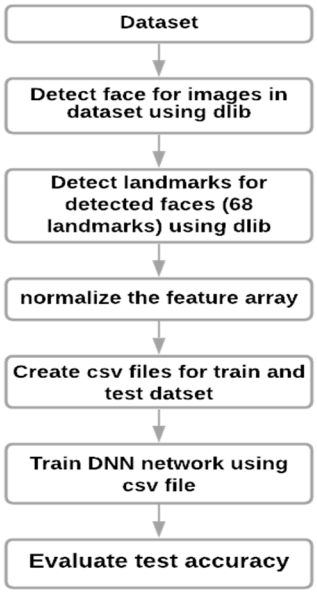

Fig. 3. Facial Emotion Detection Flowchart

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 272)               37264
_________________________________________________________________
dense_1 (Dense)              (None, 544)               148512
_________________________________________________________________
dense_2 (Dense)              (None, 272)               148240
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 2184
=================================================================
Total params: 336,200
Trainable params: 336,200
Non-trainable params: 0
_________________________________________________________________
```

Fig. 4. Facial Emotion Detection DNN Model Summary

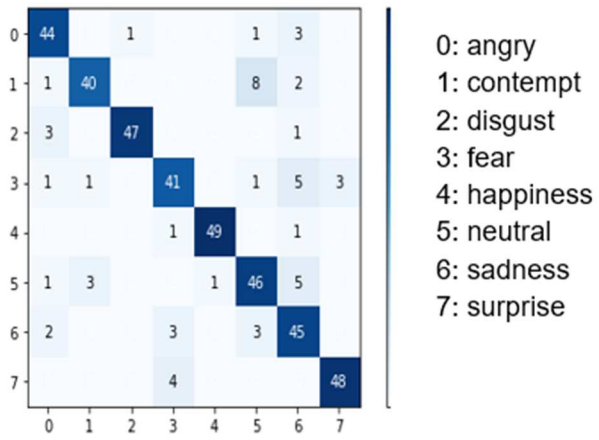

Fig. 5. Confusion Matrix for the test set classification

VI. Real-Time Analysis For Facial Emotion Detection

The system uses OpenCV, to read video frames either by using the feed from a camera connected to a computer or by reading a video file. We then perform face detection and facial landmark extraction on the frame and feed the normalized landmark coordinates into the DNN which classifies the emotion of the faces in the frame.

Since we use the sigmoid activation function in the neural network, the output of the DNN is an array in which each element represents the probability of (indexed) emotion occurring independently of other emotions. The sum of the array elements may not necessarily be 1 as the sigmoid function doesn't treat emotions to be mutually exclusive. This allows us to improve the accuracy of our system while performing real-time processing by setting a threshold for the level of confidence for each of the eight emotions. We only display the emotion if the confidence level of that emotion is greater than its threshold value. If the emotion detected does not cross the threshold value, we display the emotion rendered in the previous frame. The facial emotion detection of a video performed with and without threshold can be found in Fig. 6 and Fig. 7 respectively.

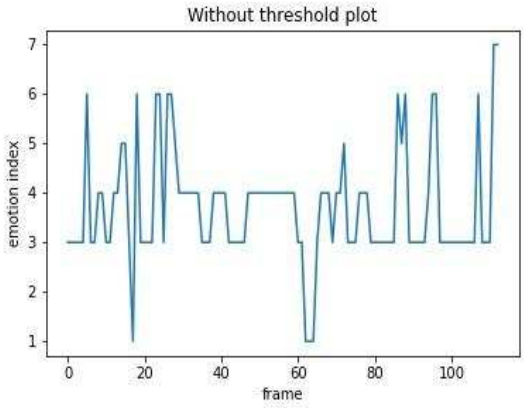

Fig. 6. Realtime face emotion detection using DNN model without threshold

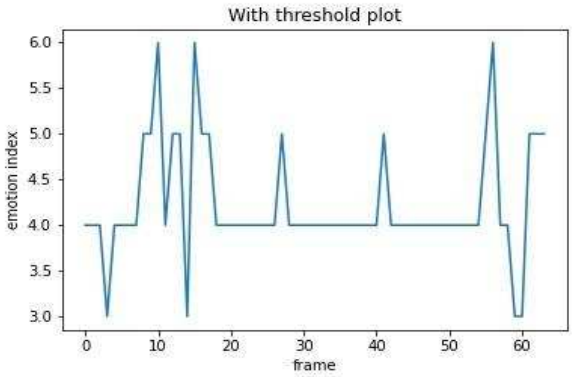

Fig. 7. Realtime face emotion detection using DNN model with threshold

The frame rate achieved for real-time face emotion detection is about 8.9 fps for laptop PC (refer to Fig. 8) and 4.1 on Jetson Nano (refer to Fig.9).

```
fps start
fps stop

[INFO] elapsed time: 12.74
[INFO] approx. FPS: 8.95
```

Fig. 8. FPS obtained On Laptop

```
fps stop

fps recorded on Jetson nano
[INFO] elapsed time: 27.20
[INFO] approx. FPS: 4.19
```

Fig. 9. FPS obtained On Jetson Nano

VII. Implementation Of The Text Sentiment Analytics System

The system converts real-time audio to text using the Speech Recognition library in python [26]. We use the Pyaudio library to record audio from a mic [27]. The recorded audio is broken down into chunks and processed bit by bit using the Recognizer function in the Speech recognition library which transcribes the audio. The transcribed audio is split. The workflow of the text sentiment analysis can be found in Fig. 12.

The accuracy of TextBlob vs Vader was compared by testing these models on the IMDB dataset and the product review dataset (refer to Fig.10) [28,29]. It can be seen that TextBlob has higher precision and F1 score for these datasets. [30] The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. [31] The formula for the standard F1-score is the harmonic mean of the precision and recall (refer to eq. 1). A perfect model has an F-score of 1.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (1)$$

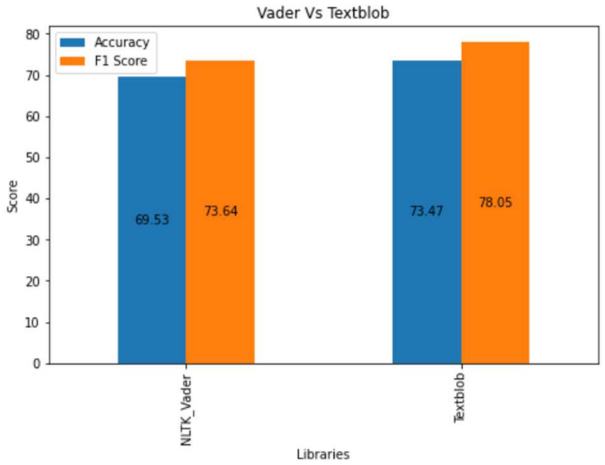

Fig. 10. Comparison between Vader and TextBlob

The proposed system determines the polarity of text using pre-trained sentiment analysis tools from various Python NLP libraries (Text Blob, Vader). The most widely used pre-trained libraries for estimating the polarity of text are Text Blob and Vader.

Fig. 11 shows some negative and positive interviewee responses to check how well these libraries can classify their polarity and overall, we find TextBlob with Naive Bayes yields more satisfying results. The numbers shown in the table are the polarity of each sentence where -100 means negative and +100 means positive.

Content	textblob	textblob_ bayes	bltk_ vader
I've enjoyed and grown in my current role	25	65	51
I am an ambitious man and driven individual. I thrive in a goal-oriented environment.	12	92	48
What makes me unique is my ability to meet and exceed deadlines.	38	59	32
While I highly valued my time at my previous company, there are no longer opportunities for growth that align with my career goals.	0	3	73
I hated the job and the company. They were awful to work for	-95	-60	-80
I do good work	70	4	44
I tend to lose my patience with incompetent people.	-35	-33	-70
I missed too much work.	20	10	-30

Fig. 11. Comparison among various libraries

The proposed system uses the TextBlob library with Naive Bayes Classifier to estimate the polarity of the text. TextBlob is a python library of Natural Language Processing (NLP) that uses the Natural Language Toolkit (NLTK) to perform its functions [28]. NLTK is a library that provides easy access to many lexical resources and allows users to work with categorization, classification, and many other tasks. It calculates average polarity and subjectivity over each word in a given text using a dictionary of adjectives and their hand-tagged scores. It uses a pattern library for that, which takes the individual word scores from sentiwordnet. The TextBlob with Naive Bayes calculates the sentiment score by NaiveBayesAnalyzer trained on a dataset of movie reviews. We use the polarity calculated by TextBlob to classify text as either positive, negative, or neutral by setting a threshold value. The polarity value lies in the range of [-1, 1], where -1 indicates negativity and +1 indicates positivity.

Threshold values are set to classify text into three classes:

- Polarity above 60% is classified as **Positive**
- The polarity between 40% and 60% is classified as **Neutral**
- Polarity below 40% is classified as **Negative**

Analysis of a transcribed text passage is done as follows:

- Number of positive sentences in the passage: x
- Number of negative sentences in the passage: y
- Number of neutral sentences in the passage: z
- Total number of sentences in a passage: $x + y + z$
- Overall positivity of the passage: Sum of polarities above 60% / Total number of sentences in a passage.
- Overall neutrality of the passage: Sum of polarities between 40% - 60% / Total number of sentences in a passage.
- Overall negativity of the passage: Sum of polarities below 40% / Total number of sentences in a passage.

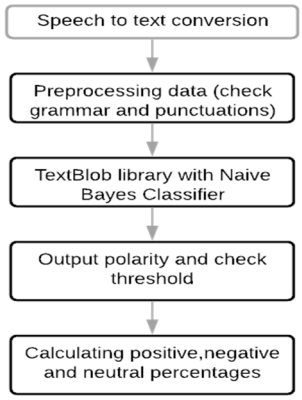

Fig. 12. Text Sentiment Analysis Workflow

VIII. Results and Discussion

The proposed integrated system extracts video and audio simultaneously with a frame rate of 4-5fps. The facial emotion detection system successfully detects facial expressions detected in real-time video with an accuracy of about 86.75%. The audio from the video is successfully extracted by the system, converted to text, cleaned, and processed to determine if the attitude of the speaker in a given situation is positive, negative, or neutral.

IX. Conclusion

This paper presents a system that uses technologies such as Machine Learning and Artificial Intelligence that operate on a principle of Deep Learning to classify facial emotions and analyze text sentiments. The proposed system allows us to know a way of sensing emotions that can be considered as mostly used AI and pattern analysis applications which enables neural networks with fewer layers (with the help of four popular datasets) which compete with, or rather outperforms much complex and deeper networks in FER. Companies with much more data and resources can build this type of algorithm and gain an even more accurate model.

References

[1] Cherry, K. (n.d.). *The 6 types of basic emotions and their effect on human behavior*. Verywell Mind. from https://www.verywellmind.com/an-overview-of-the-types-of-emotions-4163976

[2] *Sentiment Analysis & Machine Learning*. MonkeyLearn Blog. (2020, April 20). Retrieved from https://monkeylearn.com/blog/sentiment-analysis-machine-learning

[3] Suk, M., & Prabhakaran, B. (1970, January 1). *Real-time mobile facial expression recognition system - a case study*. Page Redirection. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2014/W03/html/Suk_Real-time_Mobile_Facial_2014_CVPR_paper.html

[4] Bahreini, K., van der Vegt, W., & Westera, W. (2019, February 6). *A fuzzy logic approach to reliable real-time recognition of facial emotions - multimedia tools and applications*. SpringerLink. from https://link.springer.com/article/10.1007/s11042-019-7250-z

[5] Mallick, S., Singh, A., Allard, M., Sahu, N., White, H., Steven, S., Pipi, Mueen, S., Tzatter, Kavianathar, H., Moreira, D., Ricardo, Lima, N., DavidBlancarte, Vareto, R., Ivatury, P., Zhao, A., Zhang, B., Cefoot, … A Crash Course with Dlib Library. (2021, May 5). *Facial Landmark Detection: LEARNOPENCV #*. LearnOpenCV. Retrieved March 5, 2022, from https://learnopencv.com/facial-landmark-detection/

[6] Google. (n.d.). *Toward better phone call and video transcription with new cloud speech-to-text | google cloud blog*. Google. from https://cloud.google.com/blog/products/gcp/toward-better-phone-call-and-video-transcription-with-new-cloud-speech-to-text

[7] Filippidou, F., & Moussiades, L. (2020, May 6). *A benchmarking of IBM, Google and wit automatic speech recognition systems*. Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part I. from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7256403/

[8] DeLancey, J. (2020, May 29). *Improving NLTK sentiment analysis with data annotation*. CodeProject. Retrieved from https://www.codeproject.com/Articles/5269453/Improving-NLTK-Sentiment-Analysis-with-Data-Annota

[9] BOLDEnthusiast. (2021, March 2). *Sentiment analysis - the lexicon-based approach*. Top Microsoft Dynamics and NetSuite Partner & Dynamics CRM Consultant in San Diego. Retrieved from https://www.alphabold.com/sentiment-analysis-the-lexicon-based-approach/

[10] *Overview*. Using opencv and dlib for face pose estimation (python) - Programmer Sought. (n.d.). Retrieved from https://www.programmersought.com/article/27703847966/

[11] Hassouneh, A., Mutawa, A. M., & Murugappan, M. (2020, June 12). Development of a real-time emotion recognition system using facial expressions and EEG based on machine learning and deep neural network methods. Informatics in Medicine Unlocked. from https://www.sciencedirect.com/science/article/pii/S235291482030201X#bib33

[12] Cambridge, P. M. U. of, Michel, P., Cambridge, U. of, Rana El Kaliouby University of Cambridge, Kaliouby, R. E., University, O. H. & S., Technology, M. I. of, Mitre, Dfki, & Metrics, O. M. V. A. (2003, November 1). *Real time facial expression recognition in video using support vector machines: Proceedings of the 5th International Conference on Multimodal Interfaces*. ACM Conferences. Retrieved March 5, 2022, from https://dl.acm.org/doi/abs/10.1145/958432.958479

[13] Rajan, S., Chenniappan, P., Devaraj, S., & Madian, N. (2019, May 7). *Facial expression recognition techniques: A comprehensive ...* https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ipr.2018.6647. Retrieved 2021, from https://ietresearch.onlinelibrary.wiley.com/doi/epdf/10.1049/iet-ipr.2018.6647

[14] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1–135. https://doi.org/10.1561/1500000011

[15] Ravi, K., Siddeshwar, V., Ravi, V., & Mohan, L. (2015). Sentiment analysis applied to educational sector. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. https://doi.org/10.1109/iccic.2015.7435667

[16] P.D. Turney, P. Pantel et al., From frequency to meaning: Vector space models of semantics, Journal of Artificial Intelligence Research 37(1) (2010), 141–188

[17] Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, *37*, 141–188. https://doi.org/10.1613/jair.2934

[18] Baccianella, S., Esuli, A., & Sebastiani, F. (n.d.). sentiNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. ACL Anthology. Retrieved 2021, from https://aclanthology.org/L10-1531/

[19] Deep Neural Networks (DNN module). OpenCV. (n.d.). Retrieved from https://docs.opencv.org/4.x/d2/d58/tutorial_table_of_content_dnn.html

[20] Mehendale, N. (2020, February 18). Facial emotion recognition using convolutional neural networks (FERC) - SN applied sciences. SpringerLink. Retrieved from https://link.springer.com/article/10.1007/s42452-020-2234-1

[21] The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. IEEE Xplore. (n.d.). Retrieved from https://ieeexplore.ieee.org/document/5543262

[22] Lyons, M., Kamachi, M., & Gyoba, J. (1998, April 14). The Japanese Female Facial Expression (Jaffe) dataset. Zenodo. Retrieved from https://zenodo.org/record/3451524#.YiNWJ3pBzrc

[23] Radboud Faces Database - Rafd.socsci.ru.nl. (n.d.). Retrieved from https://rafd.socsci.ru.nl/RaFD2/RaFD?p=main

[24] Tf.keras.activations.sigmoid :   Tensorflow core v2.8.0. TensorFlow. (n.d.). Retrieved from https://www.tensorflow.org/api_docs/python/tf/keras/activations/sigmoid

[25] Tf.keras.optimizers.Adam :   Tensorflow core v2.8.0. TensorFlow. (n.d.). Retrieved from https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam

[26] Speechrecognition. PyPI. (n.d.). Retrieved from https://pypi.org/project/SpeechRecognition/

[27] Pyaudio. PyPI. (n.d.). Retrieved from https://pypi.org/project/PyAudio/

[28] Simplified text processing. TextBlob. (n.d.). Retrieved from https://textblob.readthedocs.io/en/dev/

[29] vader. NLTK. (n.d.). Retrieved from https://www.nltk.org/_modules/nltk/sentiment/vader.html

[30] *F-score*. DeepAI. (2019, May 17). Retrieved from https://deepai.org/machine-learning-glossary-and-terms/f-score

[31] *Sklearn.metrics.f1_score*. scikit. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html