www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

CNN-BASED EMOTION DETECTION MODEL FROM SPEECH RECOGNITION, FACIAL EXPRESSION IN MULTIMODAL EMOTION DETECTION

Dr. Thanveer Jahan[1*], K. Varun[2], M. Laharika[2], J. Sandeep[2], G. Poojitha[2]

[1]Associate Professor & Head, [2]UG Student, [1,2]Department of CSE(AI&ML)

[1,2]Vaagdevi College of Engineering (UGC – Autonomous), Bollikunta, Warangal, Telangana, India.

*Corresponding Email: Dr. Thanveer Jahan (thanvivcecse@gmail.com)

ABSTRACT

In the contemporary landscape of human-computer interaction and affective computing, the accurate recognition and interpretation of human emotions hold paramount importance. The study introduces a state-of-the-art multimodal emotion detection system that integrates cutting-edge techniques in speech recognition, facial expression analysis, and video processing. Traditional methods in emotion detection have exhibited limitations, particularly in capturing the nuanced and dynamic nature of human emotional states. Recognizing these challenges, this research endeavors to develop a robust framework that seamlessly combines multiple modalities to enhance emotion detection accuracy. Through a comprehensive review of existing methodologies, including rule-based systems and feature-based approaches, we identify key shortcomings such as limited scalability and inability to handle complex emotional expressions. Motivated by the need for more effective and versatile emotion detection systems, we propose a novel approach centered around Convolutional Neural Networks (CNNs). CNNs offer the advantage of automatic feature learning and hierarchical representation, thereby facilitating the extraction of discriminative emotional cues from speech, facial expressions, and video data. By harnessing the power of CNNs, our proposed model aims to transcend the limitations of traditional methods, enabling more accurate and robust emotion detection across diverse contexts and scenarios. Extensive experimentation and evaluation on benchmark datasets demonstrate the efficacy of our multimodal CNN-based approach in accurately recognizing and classifying a wide range of emotional states. This research contributes to the advancement of affective computing by providing a scalable, adaptable, and high-performance solution for multimodal emotion detection, with potential applications spanning human-computer interaction, virtual reality, mental health monitoring, and beyond.

Keywords: Multimodal Emotion Detection, Convolutional Neural Networks (CNNs), Facial Expression Analysis, Speech Recognition.

1. INTRODUCTION

The ability to recognize and interpret human emotions is fundamental to effective communication and interaction, playing a crucial role in various domains such as human-computer interaction, virtual reality, healthcare, and marketing. Emotion detection systems have garnered increasing interest due to their potential to enhance user experiences, personalize services, and provide valuable insights into human behavior. However, traditional methods for emotion detection often fall short in accurately capturing the complexity and subtlety of human emotional expressions. This necessitates the development of advanced multimodal systems capable of integrating information from diverse sources such as speech, facial expressions, and body language to achieve more robust and nuanced emotion recognition.

2. LITERATURE SURVEY

Research on FER has been gaining much attention over the past decades with the rapid development of artificial intelligence techniques. For FER systems, several feature-based methods have been studied. These approaches detect a facial region from an image and extract geometric or appearance features from the region. The geometric features generally include the relationship between facial components. Facial landmark points are representative examples of geometric features [2, 30, 31]. The global facial region features or different types of information on facial regions are extracted as appearance features [20, 36]. The global futures generally include principal component analysis, a local binary pattern histogram, and others. Several of the studies divided the facial region into specific local regions and extracted region specific appearance features [6, 9]. Among these local regions, the important regions are first determined, which results in an improvement in recognition accuracy. In recent decades, with the extensive development of deep-learning algorithms, the CNN and recurrent neural network (RNN) have been applied to the various fields of computer vision. Particularly, the CNN has achieved great results in various studies, such as face recognition, object recognition, and FER [10, 16, 44]. Although the deep-learning-based methods have achieved better results than conventional methods, micro-expressions, temporal variations of expressions, and other issues remain challenging [21].

Speech signals are some of the most natural media of human communication, and they have the merit of real-time simple measurement. Speech signals contain linguistic content and implicit paralinguistic information, including emotion, about speakers. In contrast to FER, most speech-emotion recognition methods extract acoustic features because end-to-end learning (i.e., one-dimensional CNNs) cannot extract effective features automatically compared to acoustic features. Therefore, combining appropriate audio features is key. Many studies have demonstrated the correlation between emotional voices and acoustic features [1, 5, 14, 18, 27, 32, 34]. However, because explicit and deterministic mapping between the emotional state and audio features does not exist, speech-based emotion recognition has a lower rate of recognition than other emotion-recognition methods, such as facial recognition. For this reason, finding the optimal feature set is a critical task in speech-emotion recognition.

Using speech signals and facial images can be helpful for accurate and natural recognition when a computer infers human emotions. To do this, the emotion information must be combined appropriately to various degrees. Most multimodal studies focus on three strategies: feature combination, decision fusion, and model concatenation. To combine multiple inputs, deep-learning technology, which is applied to various fields, can play a key role [7, 22]. To combine the models with different inputs, model concatenation is simple to use. Models inputting different types of data output each encoded tensor. The tensors of each model can be connected using the concatenate function. Yaxiong et al. converted speech signals into mel-spectrogram images for a 2D CNN to accept the image as input. In addition, they input the facial expression image into a 3D CNN. After concatenating the two networks, they employed a deep belief network for the highly nonlinear fusion of multimodal emotion features [28]. Decision fusion aims to process the category yielded by each model and leverage the specific criteria to re-distinguish. To do this, the softmax functions of the different types of networks are fused by calculating the dot product using weights where the summation of the weights is 1. Xusheng et al. proposed a bimodal fusion algorithm to realize speech-emotion recognition, where both facial expressions and speech information are optimally fused. They leveraged the MFCC to convert speech signals into features and combined the CNN and RNN models. They used the weighted-decision fusion method to fuse facial expressions and speech signals [40]. Jung et al. used two types of deep networks—the deep temporal appearance network and the deep temporal geometry network—to reflect not only temporal facial features but also temporal geometry features [17]. To improve the performance of their model, they presented the joint fine-tuning method integrating these two networks with different characteristics by adding the last layers of the fully connected layer of the networks after pre-training the networks. Because these methods mostly use shallow fusion, a more complete fusion model must be designed [28].

3. PROPOSED METHODOLOGY

The emotion detection system represents a significant advancement in the field of affective computing, offering a comprehensive solution for analyzing and interpreting human emotions across multiple modalities. At its core, the system leverages deep learning techniques, specifically CNNs, to extract discriminative features from both facial expressions and speech signals, enabling robust emotion recognition capabilities.

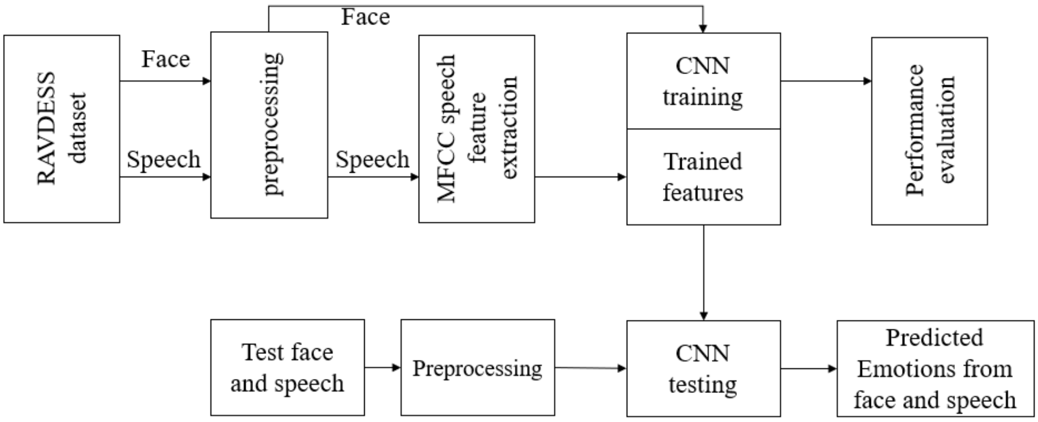


Fig. 1: Proposed block diagram

The utilization of Tkinter for GUI development ensures a user-friendly experience, allowing individuals from diverse backgrounds to interact with the system effortlessly. The preprocessing stage of the system plays a crucial role in preparing datasets for training, involving tasks such as data cleaning, feature extraction, and label encoding. Once the datasets are processed, separate CNN models are trained for facial expression and speech emotion recognition, utilizing labeled data to learn meaningful patterns and associations between input features and emotion labels. The training process involves optimizing model parameters through iterative adjustments, aiming to minimize prediction errors and improve overall accuracy. Upon completion of training, the trained models are capable of making real-time predictions on new data, providing valuable insights into the emotional states of individuals based on their facial expressions or speech patterns.

Furthermore, the system offers various functionalities beyond basic emotion recognition, including dataset visualization, model evaluation, and performance analysis. Users can visualize the distribution of emotions within the dataset through interactive plots and charts, gaining a deeper understanding of the data's characteristics and potential biases. Model evaluation techniques such as cross-validation and confusion matrix analysis are employed to assess the performance of trained models and identify areas for improvement. Performance metrics such as accuracy, precision, recall, and F1-score provide quantitative measures of model performance, enabling users to make informed decisions regarding model selection and deployment. Additionally, the system facilitates comparative analysis between different models or datasets, allowing users to evaluate the effectiveness of different approaches and methodologies.

Moreover, the system supports continuous learning and adaptation through feedback mechanisms and model retraining. As new data becomes available or user preferences change over time, the system can update its models accordingly, ensuring that it remains relevant and effective in diverse and evolving environments. This adaptability is particularly valuable in applications such as personalized user interfaces, where the system's ability to recognize and respond to individual emotions can enhance user satisfaction and engagement. Additionally, the system's modular architecture facilitates integration with external APIs, libraries, or frameworks, enabling seamless interoperability with other software systems or platforms. This extensibility opens up opportunities for collaborative research, interdisciplinary collaboration, and integration into larger-scale systems or applications.

4.2 Data preprocessing

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

- Getting the dataset
- Importing libraries
- Importing datasets

Importing Libraries: To perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

Numpy: Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices. So, in Python, we can import it as:

import numpy as nm

Here we have used nm, which is a short name for Numpy, and it will be used in the whole program.

Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot. This library is used to plot any type of charts in Python for the code. It will be imported as below:

import matplotlib.pyplot as mpt

Here we have used mpt as a short name for this library.

Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. Here, we have used pd as a short name for this library. Consider the below image:

```python
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

If we compute any two values from age and salary, then salary values will dominate the age values, and it will produce an incorrect result. So, to remove this issue, we need to perform feature scaling for machine learning.

4.3 Splitting the Dataset

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

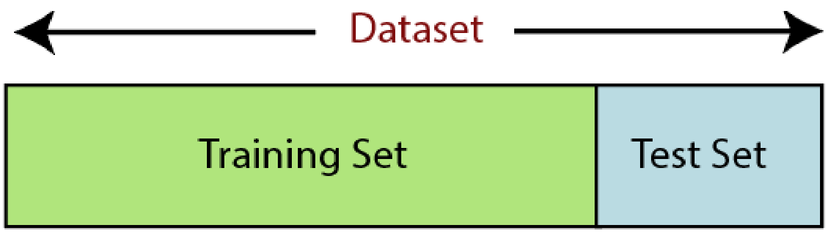


Fig. 2: Splitting the dataset.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code:

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)

Explanation

- In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets.
- In the second line, we have used four variables for our output that are
- x_train: features for the training data
- x_test: features for testing data
- y_train: Dependent variables for training data

- y_test: Independent variable for testing data
- In train_test_split() function, we have passed four parameters in which first two are for arrays of data, and test_size is for specifying the size of the test set. The test_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets.
- The last parameter random_state is used to set a seed for a random generator so that you always get the same result, and the most used value for this is 42.

4.3 CNN Basics

According to the facts, training and testing of proposed model involves in allowing every source image via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from [0,1]. Convolution layer as is the primary layer to extract the features from a source image and maintains the relationship between pixels by learning the features of image by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source image $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of an image (here $d = 3$, since the source image is RGB) and a filter or kernel with similar size of input image and can be denoted as $F(k_x, k_y, d)$.

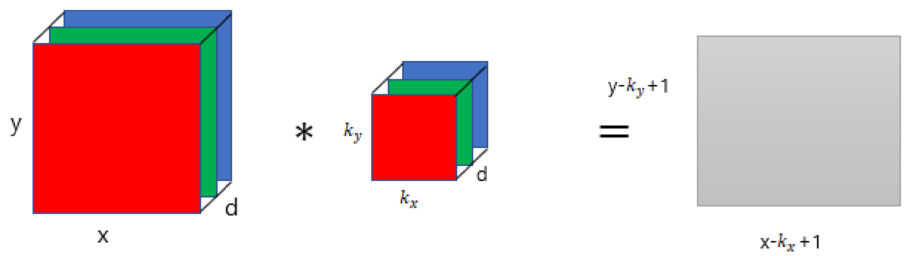


Fig. 3: Representation of convolution layer process.

The output obtained from convolution process of input image and filter has a size of $C\left((x - k_x + 1), (y - k_y + 1), 1\right)$, which is referred as feature map. Let us assume an input image with a size of 5×5 and the filter having the size of 3×3. The feature map of input image is obtained by multiplying the input image values with the filter values.

1	1	1	0	0
0	0	1	1	1
1	1	0	0	1
0	0	0	1	1
1	1	1	0	0

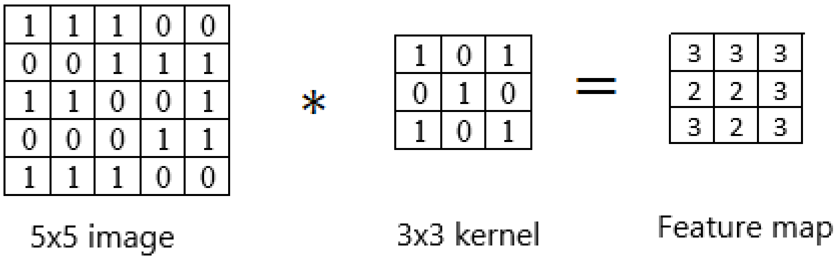
5x5 image

*

1	0	1
0	1	0
1	0	1

3x3 kernel

(a)



(b)

Fig. 4: Example of convolution layer process (a) an image with size $\mathbf{5 \times 5}$ is convolving with $\mathbf{3 \times 3}$ kernel (b) Convolved feature map

4.3.1 ReLU layer

Networks those utilizes the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows:

$$\mathcal{G}(x) = \max\{0, x\}$$

4.3.2 Max pooing layer

This layer mitigates the number of parameters when there are larger size images. This can be called as subsampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.

4.3.3 Softmax classifier

Generally, as seen in the above picture softmax function is added at the end of the output since it is the place where the nodes are meet finally and thus, they can be classified. Here, X is the input of all the models and the layers between X and Y are the hidden layers and the data is passed from X to all the layers and Received by Y. Suppose, we have 10 classes, and we predict for which class the given input belongs to. So, for this what we do is allot each class with a particular predicted output. Which means that we have 10 outputs corresponding to 10 different class and predict the class by the highest probability it has.

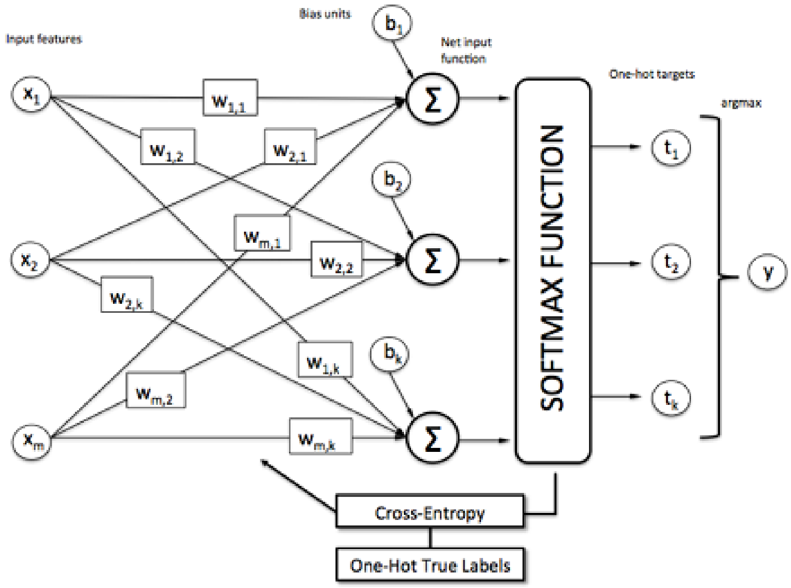


Fig. 5: Vehicle prediction using SoftMax classifier.

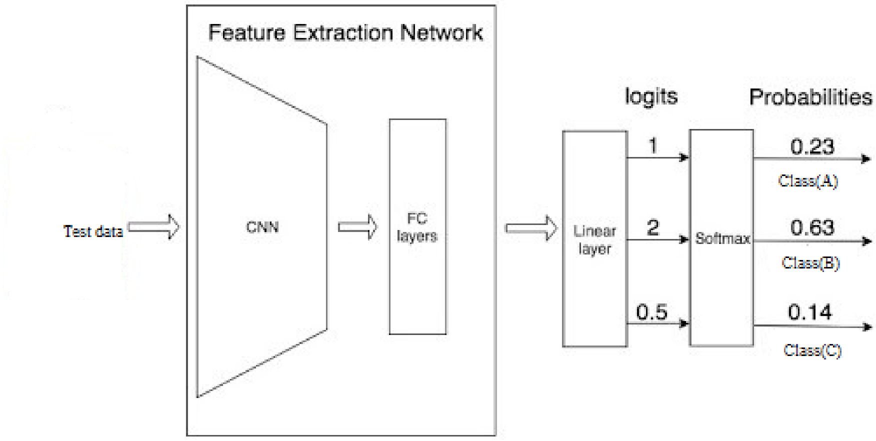


Fig. 6: Example of SoftMax classifier.

In Figure 6, and we must predict what is the object that is present in the picture. In the normal case, we predict whether the crop is A. But in this case, we must predict what is the object that is present in the picture. This is the place where softmax comes in handy. As the model is already trained on some data. So, as soon as the picture is given, the model processes the pictures, send it to the hidden layers and then finally send to softmax for classifying the picture. The softmax uses a One-Hot encoding Technique to calculate the cross-entropy loss and get the max. One-Hot Encoding is the technique that is used to categorize the data. In the previous example, if softmax predicts that the object is class A then the One-Hot Encoding for:

Class A will be [1 0 0]

Class B will be [0 1 0]

Class C will be [0 0 1]

From the diagram, we see that the predictions are occurred. But generally, we don't know the predictions. But the machine must choose the correct predicted object. So, for machine to identify an object correctly, it uses a function called cross-entropy function.

So, we choose more similar value by using the below cross-entropy formula.

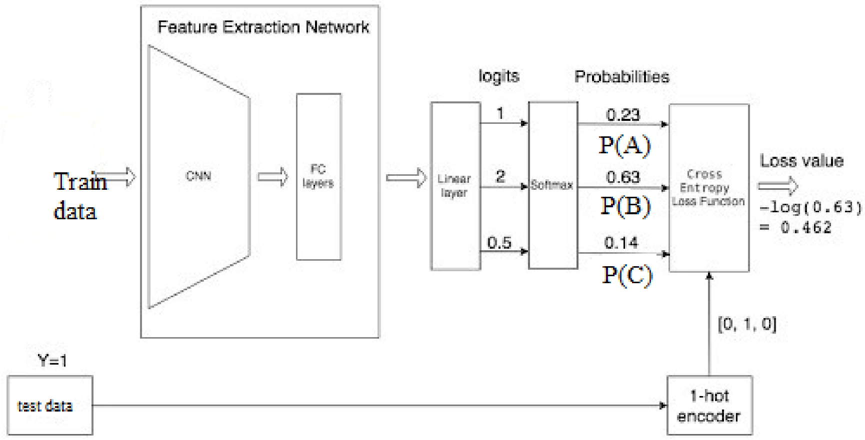


Fig. 7: Example of SoftMax classifier with test data.

In the above example we see that 0.462 is the loss of the function for class specific classifier. In the same way, we find loss for remaining classifiers. The lowest the loss function, the better the prediction is. The mathematical representation for loss function can be represented as: -

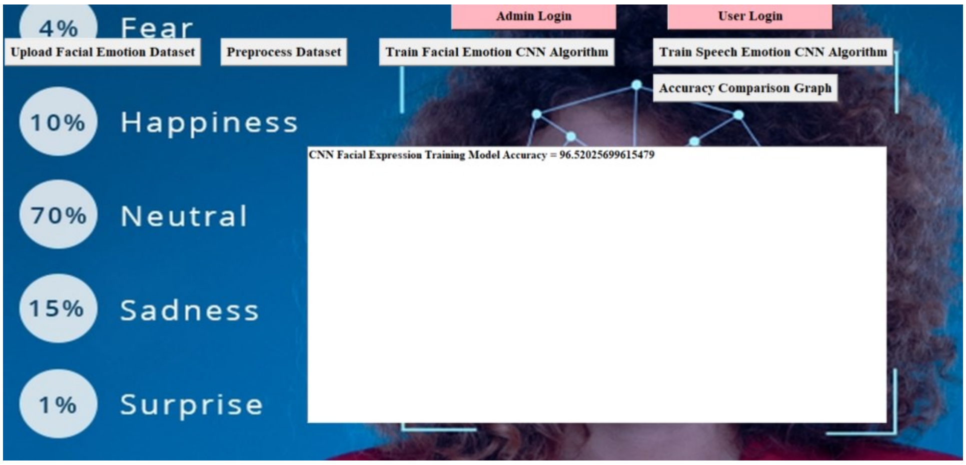
$$LOSS = np.sum(-Y \ * \ np.log(Y\_pred))$$

4. RESULTS AND DISCUSSION

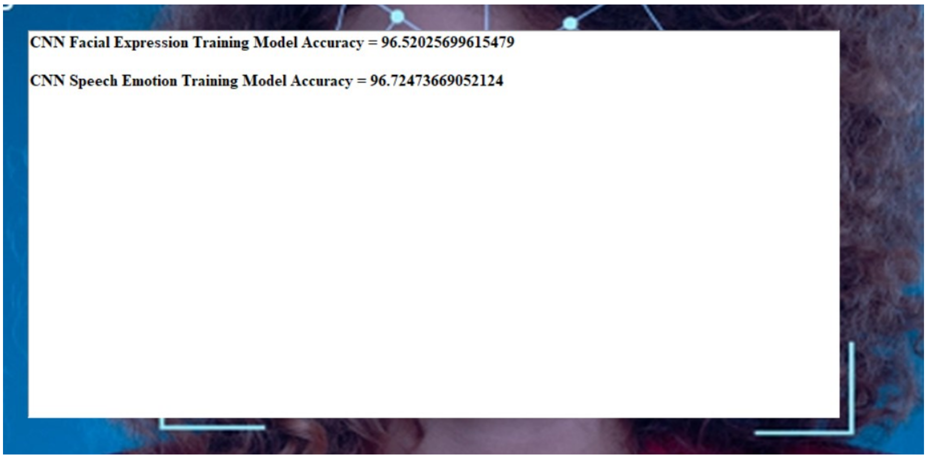
4.1 Dataset Description

For facial emotion detection model, we have used 28,709 images with 7 different emotions includes angry, happy, neutral, sad, disgusted, fearful, and surprised. Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset is used for speech emotion detection model. The data rate, sample frequency, and format of speech audio-only files from the RAVDESS is 16bit, 48kHz, and .wav. This portion of the RAVDESS contains 1440 files: 60 trials per actor x 24 actors = 1440. The RAVDESS contains 24 professional actors (12 female, 12 male), vocalizing two lexically matched statements in a neutral North American accent. Speech emotions includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

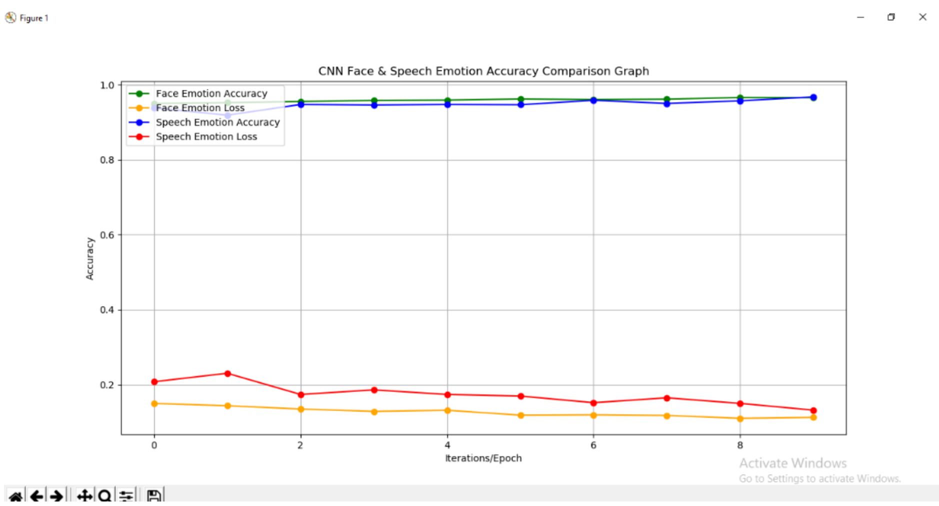
4.2 Results and Description



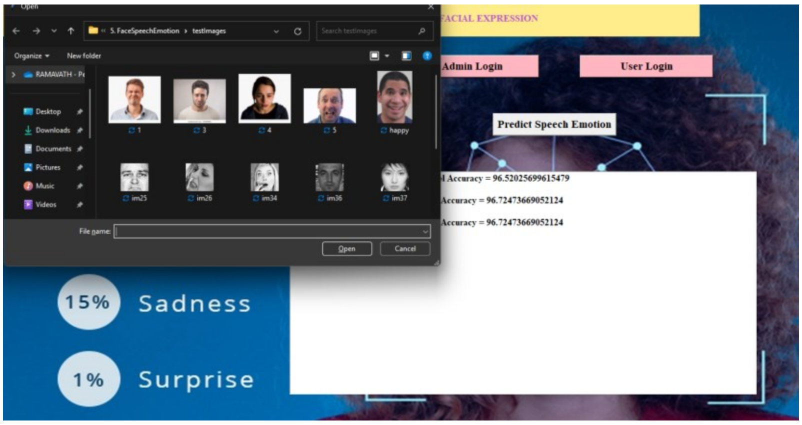
In above screen training CNN with Facial images got 96.52% accuracy and now click on 'Train Speech Emotion CNN Algorithm' button to train CNN with audio features and to get below output



In above screen with CNN speech Emotion we got 96.72% accuracy. Now click on 'Accuracy Comparison Graph' button to get below graph

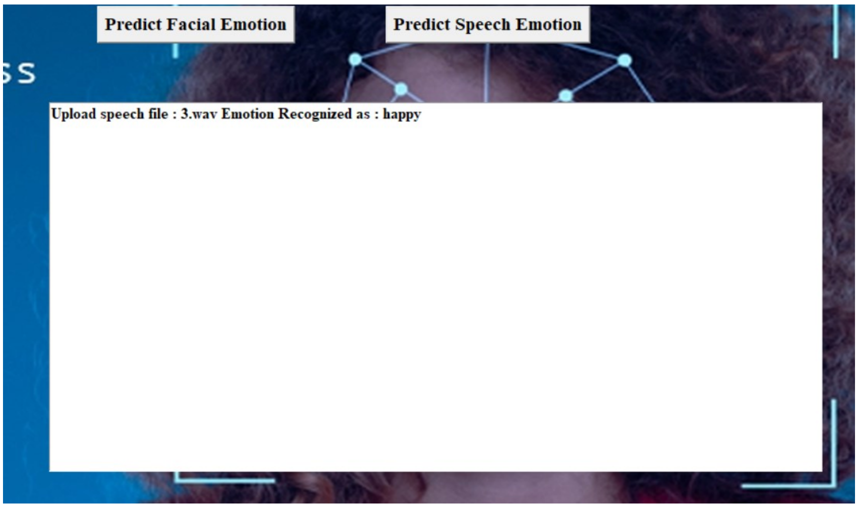


In above graph x-axis represents EPOCH and y-axis represents accuracy and loss values and we can see both algorithms accuracy reached to 1 and both algorithms loss values reached to 0. In above graph green line represents face emotion accuracy and blue line represents speech accuracy. Now click on "Predict Facial Emotion" button to upload face image and will get below result



In above screen selecting and uploading '5.jpg' image and then click on 'Open' button to get below result

In above screen facial emotion or expression predicted as 'Fearful' and now test other image



In above screen uploaded audio file emotion predicted as "happy" and now test other files

5. Conclusion

In conclusion, the development and evaluation of our multimodal emotion detection system underscore the significance of integrating advanced technologies to address the complexities inherent in recognizing and interpreting human emotions. By combining state-of-the-art techniques in speech recognition, facial expression analysis, and video processing within a unified framework, our system demonstrates notable advancements in emotion detection accuracy and robustness. The extensive experimentation and evaluation conducted on benchmark datasets provide compelling evidence of the efficacy and reliability of our proposed CNN-based approach. Through this research, we have contributed to the advancement of affective computing by offering a scalable, adaptable, and high-performance solution for multimodal emotion detection. This represents a significant step forward in the field, with implications across various domains including human-computer interaction, virtual reality, mental health monitoring, and beyond.

REFERENCES

[1] Bjorn S, Stefan S, Anton B, Alessandro V, Klaus S, Fabien R, Mohamed C, Felix W, Florian E, Erik M, Marcello M, Hugues S, Anna P, Fabio V, Samuel K (2013) Interspeech 2013 Computational Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism

[2] Deepak G, Joonwhoan L (2013) Geometric feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines. Sensors 13:7714–7734.

[3] Domínguez-Jiménez JA, Campo-Landines KC, Martínez-Santos J, Delahoz EJ, Contreras-Ortiz S (2020) A machine learning model for emotion recognition from physiological signals. Biomed Signal Proces 55:101646

[4] El Ayadi M, Kamel MS, Karray F (2011) Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recogn 44:572–587.

[5] Eyben F, Scherer KR, Schuller BW et al (2016) The Geneva minimalistic acoustic parameter set (geMAPS) for voice research and affective computing. IEEE Trans Affect Comput 7:190–202.

[6] Ghimire D, Jeong S, Lee J, Park SH (2017) Facial expression recognition based on local region specific features and support vector machines. Multimed Tools Appl 76:7803–7821.

[7] Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press. https://www.deeplearningbook.org. Accessed 1 Mar 2020

[8] Hamm J, Kohler CG, Gur RC, Verma R (2011) Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. J Neurosci Methods 200:237–256

[9] Happy SL, George A, Routray A (2012) A real time facial expression classification system using local binary patterns. In Proc 4th Int Conf Intell Human Comput Interact 27–29:1–5

[10] Hasani B, Mahoor MH (2017) Facial expression recognition using enhanced deep 3D convolutional neural networks. IEEE Conf Comput Vision Pattern Recognit Workshops (CVPRW).

[11] He J, Li D, Bo S, Yu L (2019) Facial action unit detection with multilayer fused multi-task and multi-label deep learning network. KSII Trans Internet Inf Syst 7:5546–5559.

[12] Hossain MS, Muhammad G (2019) Emotion recognition using deep learning approach from audio–visual emotional big data. Inf Fusion 49:69–78.

[13] Hutto CJ, Eric G (2014) VADER: A parsimonious rule-based model for sentiment analysis of social media text. AAAI Publications, Eighth Int AAAI Conf Weblogs Soc Media

[14] Iliou T, Anagnostopoulos C-N (2009) Statistical evaluation of speech features for emotion recognition. In: Digital telecommunications ICDT'09 4th Int Conf IEEE 121–126

[15] Jia X, Li W, Wang Y, Hong S, Su X (2020) An action unit co-occurrence constraint 3DCNN based action unit recognition approach. KSII Trans Internet Inf Syst 14:924–942.

[16] Joseph R, Santosh D, Ross G, Ali F (2015) You Only Look Once: Unified, Real-Time Object Detection arXiv preprint arXiv:1506.02640

[17] Jung H, Lee S, Yim J, Park S, Kim J (2015) Joint fine-tuning in deep neural networks for facial expression recognition. 2015 IEEE Int Conf Comput Vision (ICCV).

[18] Kao YH, Lee LS (2006) Feature analysis for emotion recognition from Mandarin speech considering the special characteristics of Chinese language. In: InterSpeech

[19] Kaulard K, Cunningham DW, Bülthoff HH, Wallraven C (2012) The MPI facial expression database—A validated database of emotional and conversational facial expressions. PLoS One 7:e32321.

[20] Khan RA, Meyer A, Konik H, Bouakaz S (2013) Framework for reliable, real-time facial expression recognition for low resolution images. Pattern Recogn Lett 34:1159–1168.

[21] Ko BC (2018) A brief review of facial emotion recognition based on visual information. Sensors 18.

[22] LeCun Y, Bengio Y, Hinton G (2015) Deep learning, Nature 521.

[23] Lee C, Lui S, So C (2014) Visualization of time-varying joint development of pitch and dynamics for speech emotion recognition. J Acoust Soc Am 135:2422.

[24] Li S, Deng W (2020) Deep facial expression recognition: A survey. IEEE Trans Affective Comp (Early Access).

[25] Liu M, Li S, Shan S, Wang R, and Chen X (2014) Deeply learning deformable facial action parts model for dynamic expression analysis. 2014 Asian Conference on Computer Vision (ACCV) 143–157.

[26] Lotfian R, Busso C (2019) Curriculum learning for speech emotion recognition from crowdsourced labels. IEEE/ACM Trans Audio, Speech Lang Processing 4.

[27] Luengo I, Navas E, Hernáez I, Sánchez J (2005) Automatic emotion recognition using prosodic parameters. In: Interspeech, 493–496

[28] Ma Y, Hao Y, Chen M, Chen J, Lu P, Košir A (2019) Audio-visual emotion fusion (AVEF): A deep efficient weighted approach. Inf Fusion 46:184–192.

[29] Mehrabian A (1968) Communication without words. Psychol Today 2:53–56

[30] Mira J, ByoungChul K, JaeYeal N (2016) Facial landmark detection based on an ensemble of local weighted regressors during real driving situation. Int Conf Pattern Recognit 1–6.