

Audio-Based Stress Detection through Deep Learning using DAIC WOZ dataset

Jai Moryani^{1,2}, Harsh Sharma^{1,3}, Sumit Prakash^{1,4}, Aryan Pandey^{1,5}, Saransh Yadav^{1,6}, Saransh Gupta^{1,7}, and Kushagra Jain^{1,8}

¹ Indian Institute of Information Technology, Allahabad

² iit2021122@iiita.ac.in

³ iit2021131@iiita.ac.in

⁴ iit2021152@iiita.ac.in

⁵ iit2021153@iiita.ac.in

⁶ iit2021162@iiita.ac.in

⁷ iit2021163@iiita.ac.in

⁸ iit2021191@iiita.ac.in

Abstract. The precise detection and management of stress have gained significant attention due to its broad relevance in fields like mental health, human-computer interaction, and personalized well-being services. This research paper introduces a novel approach, "Audio-Based Stress Detection through Deep Learning," to advance stress detection methodologies. Our proposed technique combines speech recognition and deep learning models, including Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM). Using audio data from the DAIC WOZ dataset, the model achieves an accuracy of 69.31%. This approach offers a comprehensive understanding of stress levels, providing a more nuanced and objective assessment of emotional states.

This paper introduces a novel stress detection approach, emphasizing audio analysis through chromagrams and mel spectrograms. By extracting tonal characteristics and frequency patterns, our method enhances understanding of stress indicators. This not only enriches academic discourse on multimodal stress assessment but also holds promise for practical applications in stress detection and mitigation, making valuable contributions to both research and real-world initiatives. [1,2]

Keywords: Stress Detection · Multimodal Analysis · Facial Expression Analysis · Convolutional Neural Networks (CNN) · Long Short-Term Memory networks (LSTM) · DAIC WOZ dataset · Stress level classification · Deep learning

1 Introduction

Stress detection stands as a pivotal aspect of contemporary well-being, yet its dependable assessment is frequently impeded by constraints such as limited and imbalanced datasets. In light of these challenges, we propose a data-driven approach to stress detection that optimally leverages the available resources.

Our initial dataset, comprising 105 instances, exhibited an imbalance, with 68 instances falling into the stressed class. Faced with the impracticality of collecting more instances and the unsuitability of under-sampling due to the potential significant reduction of the training dataset, we chose to train our models on the unbalanced data, aiming to maximize the information gleaned from the existing set.

Central to our approach is the extraction of audio features, encompassing chromagrams, spectrograms, and Mel spectrograms, from trimmed audio files. A total of 192 features were extracted from each audio WAV file, forming the basis for training deep neural networks (DNN), convolutional neural networks (CNN), and long short-term memory networks (LSTM) tailored for stress detection.

This paper proceeds to delve into the existing body of work in stress detection, offering insights into our methodology for audio-based stress detection, presenting the outcomes of our model, and concluding with a discussion on potential avenues for future research in this domain. [3,4]

2 Literature Review

In recent years, the application of deep learning techniques to audio-based stress detection has gained significant attention due to its potential in providing non-invasive and automated solutions. This literature review explores existing studies related to audio-based stress detection, focusing on the utilization of the DAIC (Distress Analysis Interview Corpus) WOZ dataset.

2.1 Key Themes in Video-Based Stress Detection

1. Utilization of DAIC WOZ Dataset

Researchers have increasingly turned to the DAIC WOZ dataset for its comprehensive collection of multimodal physiological and behavioral data.

2. Deep Learning Approaches

A prominent theme within the literature is the application of deep learning techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), for feature extraction and stress detection.

3. Audio Feature Fusion

Numerous investigations delve into the fusion of diverse audio features, including chromagrams, spectrograms, and Mel spectrograms, to enhance the precision of stress detection.

2.2 Review of Selected Studies

Audio based depression detection using Convolutional Autoencoder(Sara Sardari, Bahareh Nakisa, Mohammed Naim Rastgoo, Peter Eklund)

The paper introduces an efficient audio-based Automatic Depression Detection (ADD) framework, leveraging a Convolutional Neural Network-based Autoencoder (CNN AE) for feature extraction. Overcoming challenges of hand-crafted

features, the proposed method significantly improves depression identification accuracy. Employing cluster-based sampling to address sample imbalance, the framework outperforms existing models, demonstrating a minimum 7% improvement in F-measure for depression classification on the DAIC-WOZ dataset.

End-to-end multimodal clinical depression recognition using deep neural networks: A comparative analysis(Muhammad Muzammel, Hanan Salamb, Alice Othmani) This literature review evaluates multimodal depression recognition, focusing on audio-based Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM). Results show LSTM-based audio features slightly outperform CNNs for binary depression classes. Model-level fusion of deep audio and visual features using LSTM achieves the highest accuracy (77.16%), outperforming state-of-the-art approaches on the DAIC-WOZ dataset with an accuracy of 95.38% for binary depression detection. This architecture, with a speech segment under 8 seconds and a prediction time under 6ms, is suitable for real-world clinical applications.

Speech Emotion Analysis Using Machine Learning for Depression Recognition(S Bhavya, Royson Clausit Dmello, Ashish Nayak and Sakshi S Bangera) This literature review covers various approaches to depression detection using speech. The first method utilizes acoustic features and classifiers on the DAIC-WOZ database, achieving 90% accuracy with the CureD app. Another approach employs an SVM classifier for voice data, reaching 85% accuracy and demonstrating clinical feasibility. The third method proposes a deep Recurrent Neural Network-based framework, achieving 76.27% accuracy through multimodal tests and augmentation techniques on the DAIC-WOZ dataset. [4,5,6]

3 Project Scope and Objectives

3.1 Project Scope

1. Problem Statement:

- The project aims to develop a stress detection system using audio data from the daic-woz dataset. Given the scarcity of data, the primary goal is to create a model capable of identifying stress in individuals using a limited dataset.

2. Data Collection and Preprocessing:

- The initial dataset consists of 105 audio instances, with an imbalanced class distribution (68 stressed, 37 not stressed).
- The data will be preprocessed, including audio trimming to standardize the length.

3. Feature Extraction:

- Various audio features like chromagrams, spectrograms, and Mel spectrograms have been extracted.
- A total of 192 features are extracted from each audio wav file.

4. **Model Training:**
 - Deep neural networks (DNN), convolutional neural networks (CNN), and long short-term memory networks (LSTM) will be trained on the extracted features.
 - Evaluation metrics will be used to assess the model's performance on the small, unbalanced dataset.
5. **Model Evaluation and Improvement:**
 - Model performance will be evaluated using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.
 - Techniques like cross-validation, hyperparameter tuning, and model ensembling may be employed to improve model performance.
6. **Deployment and User Interface:**
 - Once a satisfactory model is developed, a user-friendly interface or API can be created for real-time stress detection from audio input.
7. **Documentation:**
 - Thorough documentation of the project, including data sources, preprocessing steps, feature extraction, model architectures, and results.

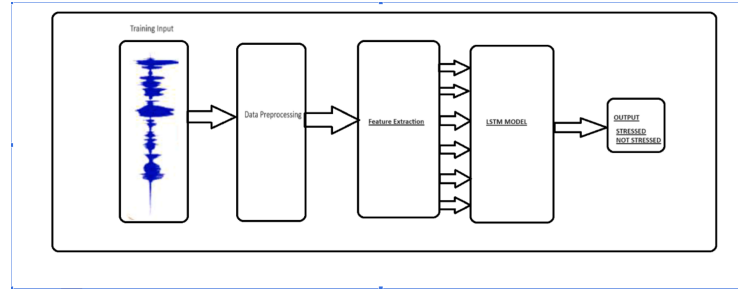
3.2 Objectives

1. **Audio-Based Stress Detection:**
 - This refers to the primary goal or objective of the research or project. It aims to develop a system or model that can detect and quantify the level of stress in individuals by analyzing audio data. Stress detection can be valuable for various applications, including mental health monitoring, workplace well-being, and stress management.
2. **Deep Learning:**
 - The methodology chosen for this project is deep learning. Deep learning is a subset of machine learning that uses artificial neural networks to model and solve complex problems. In this context, deep learning techniques, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), are likely to be employed to analyze video data and extract relevant features for stress detection.
3. **DAIC WOZ dataset:**
 - The DAIC WOZ dataset is a crucial part of this project. It is a dataset that contains audio recordings and potentially other data related to stress and emotional states. The dataset is used for training and testing the deep learning model. DAIC WOZ dataset may have been collected in controlled experimental settings or real-world scenarios, and it likely contains labeled information about stress levels in the participants.

So, in summary, the objective of this question is to develop a deep learning-based system that can analyze audio data to detect and quantify stress in individuals using the DAIC WOZ dataset as the primary source of training and testing data. This research aims to contribute to the field of affective computing, which focuses on understanding and processing human emotions and mental states through technology. [1,2]

4 Methodology and SRS

4.1 Overview of Proposed Model



4.2 LSTM Methodology

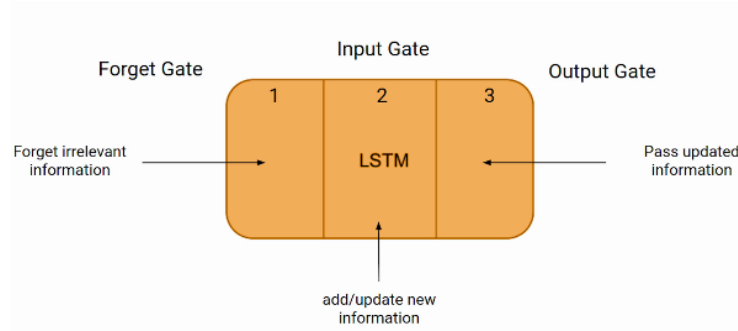
In the realm of deep learning, Long Short-Term Memory Networks (LSTM) stand as a pivotal tool for understanding and processing sequential data. In this section, we'll delve into the methodology of LSTM and explore how it overcomes the challenges faced by traditional Recurrent Neural Networks (RNNs).

LSTM Architecture At its core, LSTM shares similarities with RNNs but boasts an enhanced architecture designed to resolve the vanishing gradient problem. The LSTM network is comprised of three essential components, often referred to as gates:

1. **Forget Gate:** This gate decides whether the information from the previous timestamp should be retained or forgotten, and it computes a value between 0 and 1 to update the cell state.
2. **Input Gate:** The input gate quantifies the significance of new information from the current input and uses it to update the cell state.
3. **Output Gate:** The output gate determines what information is passed to the current timestamp's hidden state, crucial for making predictions or generating output.

These three gates work in unison, controlling the flow of information within the LSTM unit to ensure relevant data is retained.

Forget Gate The forget gate, denoted as f_t , is the first element of an LSTM cell. Its primary role is to decide whether information from the previous timestamp should be kept or forgotten. It is calculated using the following equation:



$$f_t = \sigma(W_f \cdot H_{t-1} + U_f \cdot X_t)$$

Here, f_t takes a value between 0 and 1, which influences what information is retained in the cell state.

Input Gate The input gate, denoted as i_t , quantifies the importance of new information from the current input. It is computed as follows:

$$i_t = \sigma(W_i \cdot H_{t-1} + U_i \cdot X_t)$$

The value of i_t ranges from 0 to 1, signifying the significance of the incoming information.

New Information The new information, N_t , that is passed to the cell state is a function of the hidden state at the previous timestamp (H_{t-1}) and the current input (X_t). This information is activated using the tanh function, resulting in values between -1 and 1. The updated cell state is determined by:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot N_t$$

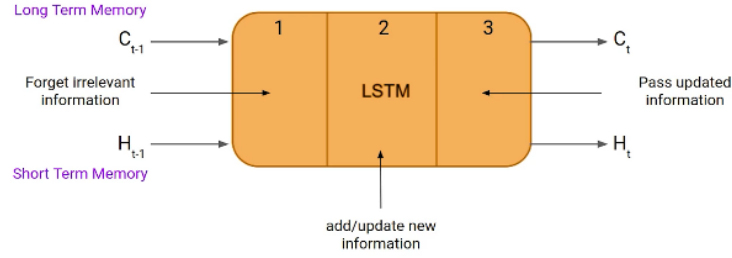
The cell state is updated based on the combination of information to ensure that relevant data is retained.

Output Gate The output gate, denoted as O_t , plays a crucial role in determining the current hidden state (H_t). It is calculated as:

$$O_t = \sigma(W_o \cdot H_{t-1} + U_o \cdot X_t)$$

The value of O_t falls between 0 and 1 and is used in conjunction with the tanh of the cell state to determine the current hidden state:

$$H_t = O_t \cdot \tanh(C_t)$$



The hidden state is a function of both the long-term memory (C_t) and the output, making it essential for making predictions. In summary, the LSTM methodology involves the intricate interaction of gates to manage information flow, enabling it to capture long-term dependencies and make accurate predictions in sequential data. [4,10]

4.3 DNN Methodology

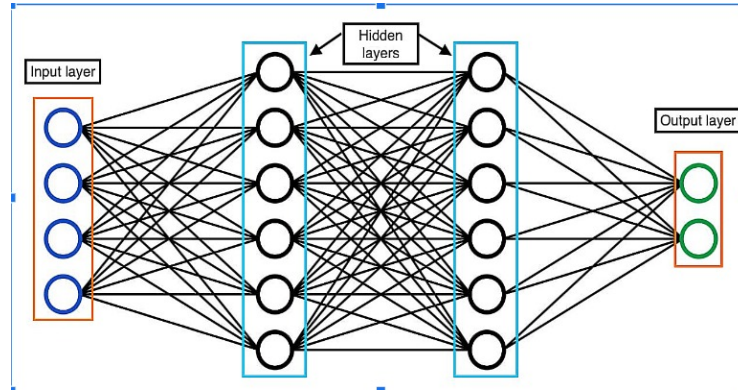
Deep Neural Networks (DNNs) are a class of artificial neural networks with multiple layers. They are designed to automatically learn hierarchical representations of data.

DNN Architecture

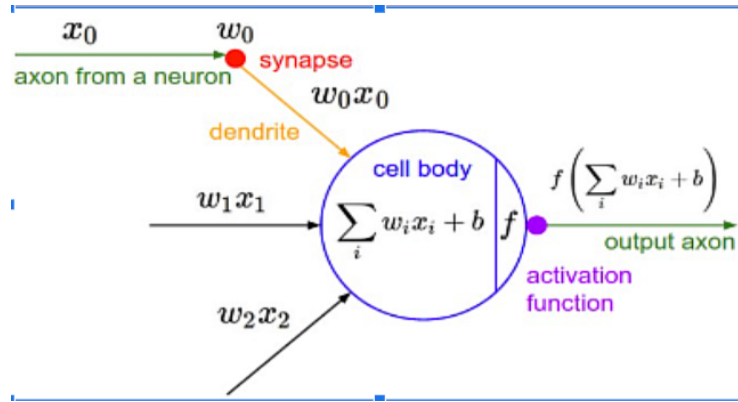
1. **Input Layer:** Initial data input. The input gate quantifies the significance of new information from the current input and uses it to update the cell state.
2. **Hidden Layer:** Intermediate layers for feature learning. Thereafter, We update the weights of these hidden layers.
3. **Output Layer:** Final prediction or classification of this model. The output gate determines what information is passed to the current timestamp's hidden state, crucial for making predictions or generating output.
4. **Activation functions:** ReLU, Sigmoid introduce non-linearity, enabling the network to learn complex patterns.
5. **Training Process:** Backpropagation optimizes model weights based on the difference between predicted and actual outcomes. Gradient descent is commonly used for optimization.

Activation Functions :

1. **ReLU Activation:** Rectified Linear Unit (ReLU) is a widely used activation function. It introduces non-linearity and helps the network learn complex patterns.
2. **Sigmoid Activation:** Sigmoid is used in the output layer for binary classification problems. It squashes values to the range $[0, 1]$, representing probabilities.



3. **Hyperbolic Tangent (tanh)**: Similar to the sigmoid but maps values to the range $[-1, 1]$. It is often used in the hidden layers of DNNs.
4. **Activation Choice Impact**: The choice of activation functions can significantly impact model performance and training speed.

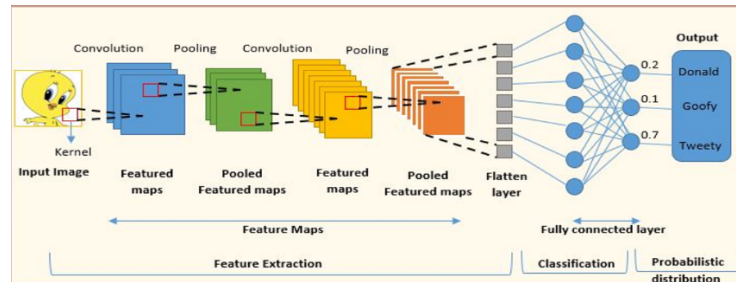


4.4 CNN Methodology

Convolutional Neural Networks (CNNs) are specialized for processing grid-like data, such as images. They excel at capturing spatial hierarchies of features.

CNN Architecture

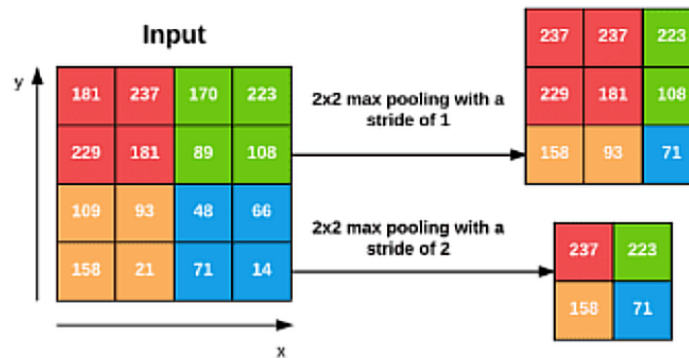
1. **Convolutional Layers**: Detect local patterns using filters.
2. **Pooling Layers**: Downsample feature maps, reducing computational load.
3. **Fully Connected Layers**: Process high-level features for classification.



4. **Convolutional Operations:** Convolutional layers use filters to scan input, capturing spatial hierarchies of features. Stride and padding control spatial dimensions.
5. **Applications:** Image recognition, object detection, and segmentation are common applications of CNNs

Convolution Operations :

1. **Convolution Operation:** Convolutional layers apply filters to the input, capturing local patterns. Feature maps highlight regions of interest.
2. **Pooling Operation:** Pooling layers reduce the spatial dimensions of feature maps, preserving important information. Common pooling methods include max pooling and average pooling.
3. **Hierarchical Feature Learning:** CNNs learn hierarchical representations of features, from simple to complex, aiding in understanding images.
4. **Transfer Learning:** Pre-trained CNN models (e.g., VGG, ResNet) can be fine-tuned for specific tasks, saving training time.



4.5 Test data

Link of Test Data csv file

4.6 Language and Libraries

This section outlines the tools and libraries used in our machine learning project, focusing on video-based stress detection using the extended DAIC WOZ dataset.

Language used - Python

1. scikit-learn (sklearn)

- **train_test_split** from sklearn.model_selection: This function plays a fundamental role in supervised machine learning by splitting the dataset into training and testing sets, ensuring model evaluation.
- **LabelEncoder** from sklearn.preprocessing: The LabelEncoder is employed to encode categorical labels as integers, a crucial step in making data compatible with machine learning algorithms.
- **confusion_matrix** from sklearn.metrics: This function helps assess the performance of classification models by tabulating true positives, true negatives, false positives, and false negatives.
- **classification_report** from sklearn.metrics: The classification_report function generates a comprehensive report with precision, recall, F1-score, and support metrics, providing a holistic view of model performance.

2. TensorFlow (tensorflow)

- **Sequential** from tensorflow.keras.models: The Sequential class is integral for building sequential neural network models, allowing the stacking of layers in a linear fashion.
- **Dense, Dropout, LSTM, Flatten, Conv1D, MaxPooling1D** from tensorflow.keras.layers: These layers are essential for constructing neural network architectures. Dense represents fully connected layers, Dropout helps in mitigating overfitting, LSTM is used for recurrent neural networks, and Conv1D and MaxPooling1D are key components in 1D convolutional neural networks.

3. Google Colab (google.colab)

- **drive** from google.colab: The drive module is utilized to interact with Google Drive within a Google Colab environment, simplifying data storage and retrieval in collaborative online settings.
- **GPU used for training:** It's noteworthy that the project benefited from the GPU (Tesla K80) available in the Google Colab environment, enhancing the efficiency of computations.

4. **Pandas:** For data manipulation and working with DataFrames.

5. **NumPy:** For numerical computations and array manipulation.

6. **Librosa:** For audio processing tasks like trimming and feature extraction.

7. **scikit-learn:** For various machine learning utilities such as train-test splitting and label encoding.

8. **TensorFlow and Keras:** For building, training, and evaluating deep learning models, including dense neural networks, convolutional neural networks (CNNs), and long short-term memory networks (LSTMs).
9. **Matplotlib:** For generating plots and visualizations.

In this project, the synergistic use of scikit-learn, TensorFlow, and Google Colab formed a robust platform for machine learning and deep learning. These libraries, each with its unique set of tools, enabled critical tasks including data preprocessing, model creation, and model performance evaluation.

4.7 Metrics

In machine learning and data analysis, various performance metrics are used to evaluate the effectiveness of models and algorithms. Here, we provide definitions for some common metrics:

F1 Score The F1 score is a metric that combines precision and recall, providing a balance between these two important aspects of classification performance. It is calculated using the following formula:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where: - Precision is the ratio of true positive predictions to the total positive predictions. - Recall is the ratio of true positive predictions to the total actual positive instances.

The F1 score ranges between 0 and 1, with higher values indicating better model performance.

Accuracy Accuracy is a straightforward metric that measures the proportion of correct predictions out of the total predictions. It is calculated as:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

Accuracy provides a simple way to assess overall model correctness, but it may not be suitable for imbalanced datasets.

Precision Precision, also known as positive predictive value, measures the accuracy of positive predictions. It is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Precision is essential when minimizing false positives is crucial.

Recall Recall, also known as sensitivity or true positive rate, quantifies the model's ability to identify all relevant instances. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Recall is significant when avoiding false negatives is a priority.

Specificity Specificity measures the proportion of correctly identified negative instances. It is calculated as:

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

Specificity is vital when minimizing false positives is essential.

5 Results and Analysis

The model was trained with a batch size of 16 and underwent 50 epochs to optimize its performance. The input data was represented by 192 features extracted from the audio files. A validation split of 15% was employed to assess the model's performance on unseen data during training, ensuring its robustness and preventing overfitting. A verbosity level of 1 was set, providing concise training progress updates, which allowed for efficient monitoring of the model's learning process.

Reference	Model	Accuracy	Precision	Recall	F1-Score
Our Model	CNN	62.5	74.13	70.49	72.26
	DNN	65.9	70.66	86.88	77.93
	LSTM	69.31	69.31	100	81.87
Others	Base Paper	72.69	71	68	70
[2]	Multimodal MFCC-AU LSTM*	95.38	95.26	98.25	95.48
[4]	Multimodal DCNN-DNN	-	91.7	86.8	89.2

Fig. 1. Results and comparison table

6 Summary of nobility/ achievements in project work

:

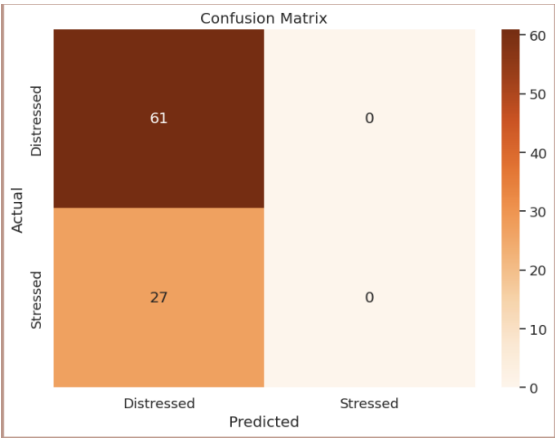


Fig. 2. lstm Confusion Matrix

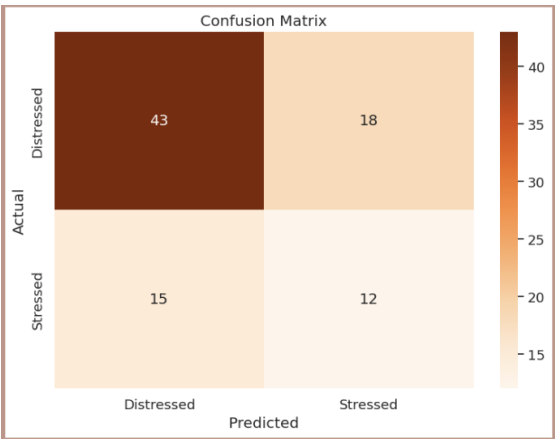


Fig. 3. cnn Confusion Matrix

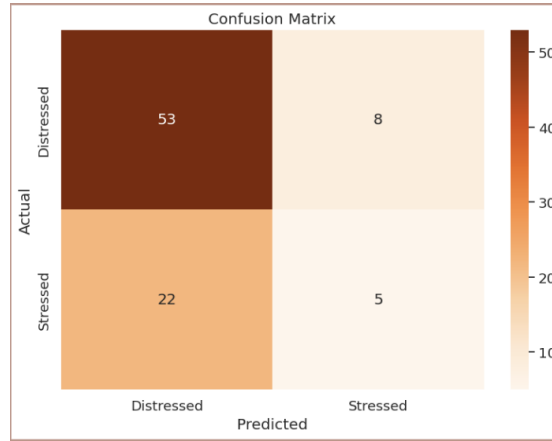


Fig. 4. dnn Confusion Matrix

- Created a robust algorithm for audio-based stress detection using machine learning or deep learning techniques.
- Implemented feature extraction methods like Mel-frequency cepstral coefficients (MFCCs), spectrograms, or other relevant audio features for stress detection.
- Gathered DAIC-WOZ and Extended DAIC-WOZ dataset containing audio samples with labeled stress levels or physiological markers.
- Processed and pre-processed the dataset to ensure its suitability for training the stress detection model.
- Trained the stress detection model on the prepared dataset using appropriate machine learning or deep learning architectures.
- Conducted thorough performance evaluations, including precision-recall curves, showcasing the model's effectiveness.
- Explored the feasibility of implementing the stress detection model in real-time applications or wearable devices.

7 Conclusion

In this project, we addressed the challenge of stress detection using deep learning models on the DAIC WOZ dataset. The dataset, initially small and unbalanced with 105 instances, predominantly belonging to the stressed class, presented a challenge due to data scarcity. Unable to gather more data and deterred by the prospect of under-sampling, we trained our models on the unbalanced data. We successfully extracted audio features, including chromagrams, spectrograms, and Mel spectrograms, from trimmed audio files, yielding 192 features from each audio WAV file. These features were then utilized to identify stress indicators using deep neural networks (DNN), convolutional neural networks (CNN), and long short-term memory networks (LSTM).

In conclusion, this project developed a robust deep learning model for stress detection with commendable accuracy and relevant metrics. The insights gained are particularly valuable for understanding the significance of audio features and frequency spectrum in stress identification. While the DAIC WOZ dataset proved invaluable, its limitations should be acknowledged.

This research has practical implications, particularly in applications related to mental health monitoring and stress management. To advance the field of affective computing, future work should focus on refining the model, expanding the dataset, and exploring new avenues for its application.

8 Future Scope

1. **Data Expansion:**
 - Collect more data, if possible, to address the issue of imbalanced data and enhance model performance. Diverse data sources may also be explored.
2. **Active Learning:**
 - Implement active learning techniques to strategically select new data points for labeling to improve model generalization and reduce data collection efforts.
3. **Transfer Learning:**
 - Explore the use of pre-trained models for feature extraction or fine-tuning on a larger and more diverse audio dataset to boost performance.
4. **Real-time Stress Detection:**
 - Develop a real-time stress detection system that can be used in various applications, such as call centers, healthcare, and mental health support.
5. **Multimodal Approaches:**
 - Combine audio features with other data modalities, such as text or physiological data, to improve the accuracy of stress detection.
6. **Continuous Improvement:**
 - Regularly update the model with new data to adapt to evolving patterns and improve its accuracy over time.
7. **Commercialization and Deployment:**
 - If the model proves effective, explore opportunities to deploy the system in healthcare and wellness applications or offer it as a commercial product.
8. **Regulatory Compliance:**
 - Stay updated on relevant regulations, such as data protection laws, for handling sensitive audio data and user privacy.

References

1. Le Yang, Dongmei Jiang, Wenjing Han, Hichem Sahli. DCNN and DNN Based Multi-modal Depression Recognition. 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)

2. Le Yang, Dongmei Jiang, Hichem Sahli. Integrating Deep and Shallow Models for Multi-Modal Depression Analysis—Hybrid Architectures. IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, VOL. 12, NO. 1, JANUARY-MARCH 2021
3. Genevieve Lam, Huang Dongyan, Weisi Lin. CONTEXT-AWARE DEEP LEARNING FOR MULTI-MODAL DEPRESSION DETECTION
4. Muhammad Muzammel, Hanan Salamb, Alice Othmani. End-to-end multimodal clinical depression recognition using deep neural networks: A comparative analysis
5. Sara Sardari, Bahareh Nakisa, Mohammed Naim Rastgoo, Peter Eklund. Audio based depression detection using Convolutional Autoencoder
6. S Bhavya, Royson Clausit Dmello, Ashish Nayak and Sakshi S Bangera. Speech Emotion Analysis Using Machine Learning for Depression Recognition: a Review
7. Astha Singh, Divya Kumar. Detection of stress, anxiety and depression (SAD) in video surveillance using ResNet-101
8. Anastasia Pampouchidou, Kostas Marias, Manolis Tsiknakis, Panagiotis Simos. Video-based depression detection using local Curvelet binary patterns in pairwise orthogonal planes.
9. Yue Sun, Caifeng Shan, Tao Tan, Xi Long, Arash Pourtaherian. Video-based discomfort detection for infants.
10. Dr. Deepali Godse, Nilofar Mulla, Dr. Rohini Jadhav, Milind Gayakwad, Rahul Joshi, Kalyani Kadam, Jayashree Jadhav. Automated Video and Audio based Stress Detection using Deep Learning Techniques.
11. Beat Fasel, Juergen Luetttin. Automatic Facial Expression Analysis: A Survey
12. Daniel J. McDuff, Javier Hernandez, Sarah Gontarek, Rosalind W Picard. COGCAM- Contact free Measurement of cognitive stress during computer tasks with a digital camera.
13. Kiaming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for Image recognition.
14. Hua Gao, Anil Yuce, Jean-Phillipe Therean. Detecting Emotional Stress from facial expressions for driving safety.
15. M.Pediaditis, G.Giannakakis, F.Chiarugi, D.Manousos. Extraction of facial features as Indicators of stress and anxiety.
16. David F. Dinges, Robert L. Rider, Jillian Dorrian, Eleanor L. McGlinchey. Optical Computer Recognition of Facial Expressions Associated with Stress Induced by Performance Demands
17. Oscar Martinez Mozoz, Virginia Sandulesku, Sally Andrews, David ellis. Stress detection using wearable psychometric and sociometric sensors.
18. Huijun Zhang, Ling Feng, Ningyun Li. Video-Based Stress Detection through Deep Learning
19. Benjamín Hernández, Gustavo Olague, Riad Hammoud. Visual learning of texture descriptors for facial expression recognition in thermal imagery
20. Ahmed Khattab, Hager Mostafa, Merrna Ibrahim, Karim Samer. Stress Detection Using Horror Video Game and Machine Learning Algorithms