IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

DISTRESS CALL DETECTION SYSTEM FOR EMERGENCY SCENARIOS USING CONVOLUTIONAL NEURAL NETWORKS

A.Durga Praveen Kumar,Assistant Professor , IT Anil Neerukonda Institute of Technology and sciences , G. S.Pavan, D. Harsha Vardhan Raju, D. Mahesh Reddy, P. Sukesh Chandu

Information Technology, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India- 530026

Abstract: This paper discusses the development of a robust system for the automated detection of distress calls in emergency situations, leveraging advanced technologies and signal processing techniques. In contemporary emergency response scenarios, accurate identification of distress signals is crucial for efficient and swift assistance. The proposed system will employ state-of-the-art algorithms and artificial intelligence to analyze various communication channels, including audio and environmental files, to recognize patterns associated with distress signals. The paper will explore the integration of sophisticated convolutional networks to process real-time data, differentiating distress signals from background noise and non-emergency communications. This model will be trained on a diverse dataset of diverse audio speech datasets to enhance the system's adaptability and accuracy in recognizing varying communication patterns. Key objectives include the development of a user-friendly interface for emergency response teams, facilitating seamless integration of distress call data into their decision-making processes. The project also aims to address challenges such as signal variability, multiple communication formats, and evolving technologies by implementing adaptive algorithms.

Keywords: *Distress Call Detection, AI-based solution, Disaster Scenarios, Audio analysis, Pre-Processing, Convolutional Neural Network,*

Performance Metrics, Emotion Classification, Web Application Interface

1. INTRODUCTION

In the contemporary landscape of emergency response systems, the prompt and accurate detection of distress calls plays a pivotal role in ensuring the safety and well-being of individuals in critical situations. Recognizing the significance of this challenge, the proposed project, titled "Distress Call Detection in Emergency Situations," endeavors to develop an innovative system that leverages advanced technologies to automate the identification of distress signals. Emergencies, ranging from natural disasters to accidents, often necessitate swift and coordinated responses from emergency services. The traditional reliance on manual distress call identification poses limitations in terms of speed and accuracy. The advent of sophisticated technologies, coupled with the power of artificial intelligence, provides an opportune avenue for the creation of an automated system capable of efficiently parsing through diverse communication channels to pinpoint distress signals.

One of the key challenges the project aims to address is the variability in distress signal formats, spanning traditional distress beacons to modern communication devices. The system will be engineered to adapt and learn from a diverse dataset of distress signals, enhancing its ability to recognize evolving communication patterns. In addition to the technical aspects, the project places a strong emphasis on user-friendliness, envisioning a user interface that empowers emergency response teams with seamless integration of distress call data into their decision-making processes. The goal is to streamline the workflow of response teams, enabling them to act swiftly and decisively in emergency scenarios.

By the trained output of the model, it is anticipated that the Distress Call Detection System will emerge as a critical component in the emergency response tools. Its successful implementation promises to significantly improve the efficiency and effectiveness of emergency response efforts, contributing to enhanced public safety and potentially saving lives in critical situations.

2. RELATED WORKS

The requirement and lower availability of similar systems that work on disaster situations has become a dire need. There are few models that are focused on automatic emotion recognition and its application to speech-based assistance systems are crucial for enhancing human-computer interaction and improving user experience. By incorporating emotion recognition technology into these systems, machines can provide more empathetic and personalized responses, leading to a more natural and engaging interaction. This is particularly important for individuals with disabilities or special needs, as emotion recognition technology can improve accessibility and inclusivity

Automatic Emotion Recognition and Its Application to Speech-Based Assistance Systems

Automatic Emotion Recognition and Its Application to Speech-Based Assistance Systems is a field dedicated to enhancing human-computer interaction by integrating emotion recognition technology into speech-based systems. This advancement aims to refine user experience and accessibility, particularly for individuals with disabilities or special needs. Through the analysis of speech patterns and the identification of emotional cues, these systems can deliver more personalized and empathetic responses, fostering a more natural and engaging.

Furthermore, this technology holds promise in crisis intervention scenarios, where it can swiftly identify signs of distress in emergency calls, enabling rapid and efficient responses. Its implementation also drives forward the fields of artificial intelligence and machine learning, catalyzing innovation and benefiting diverse industries and applications

Automated Emotion Analysis of Emergency Calls for Mental Health Support

In the realm of mental health support, the Automated Emotion Analysis of Emergency Calls project aims to revolutionize the field by developing a system capable of analyzing emotions in emergency calls related to mental health issues. This technology utilizes advanced machine learning algorithms to discern emotional cues embedded in callers' voices, enabling mental health professionals to provide timely and targeted support. By automating the emotion analysis process, this project seeks to enhance the efficiency and effectiveness of emergency mental health services, ultimately improving outcomes for individuals in crisis.

This innovative model not only aims to improve the quality of support provided to individuals in crisis but also to streamline the workflow of mental health professionals. By automating the emotion analysis process, the system can quickly and accurately identify callers' emotional states, allowing professionals to focus more on providing empathetic and effective support rather than spending time on initial assessment.

3. PROPOSED SYSTEM

The proposed system for distress call detection in emergency situations aims to leverage cutting-edge technologies to enhance the speed, accuracy, and efficiency of identifying signals indicative of urgent assistance needs.

Primarily utilizing Convolutional Neural Networks (CNN) to enhance the analysis of emergency distress calls. By utilizing audio files containing emotional speech patterns from various subjects, the system trains the CNN model to recognize and classify emotions. The primary objective is to determine whether the emotional state conveyed in a distress call indicates a distress scenario or not.

Through this approach, the system seeks to improve the efficiency and accuracy of identifying distress situations, enabling faster and more effective responses from emergency services. This research-oriented approach involves the creation of a specialized dataset comprising diverse emotional states, enabling the CNN model to learn complex patterns associated with distress calls. Through rigorous experimentation and validation, the system aims to achieve high accuracy in identifying distress scenarios, contributing to the body of knowledge in emotion recognition and its application in emergency response systems. Furthermore, the system's potential for real-time analysis opens up avenues for future research in developing efficient and responsive distress call detection systems.

The system's output includes the predicted emotion, such as anger, fear, joy, sadness, or neutral, and a binary distress prediction indicating whether the emotion suggests a distress scenario or not. The system also provides a visual representation of the audio file's spectrogram, allowing users to observe the underlying patterns in the audio data. This comprehensive approach aims to provide a reliable and efficient method for analyzing emergency distress calls, ultimately improving outcomes for individuals in crisis.

4. DATA FLOW DIAGRAM

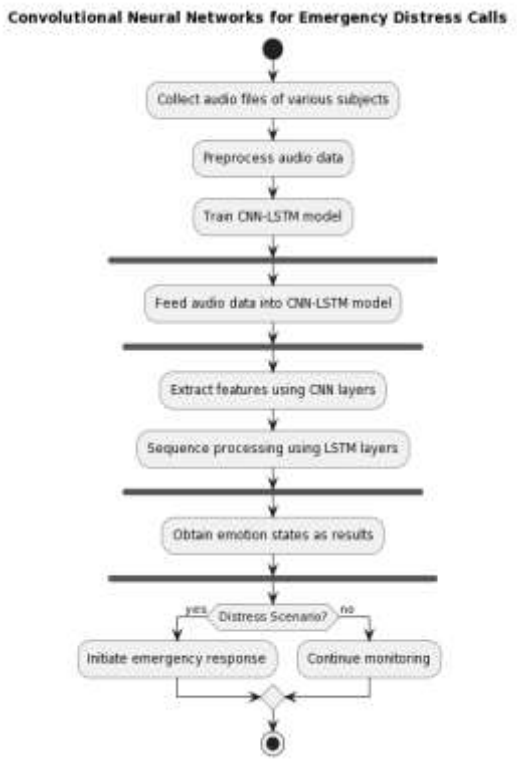


Fig.1 – Data Flow Diagram

5. METHODOLOGY

The proposed model works upon various procedural steps which are referenced in (Fig-4.1). The Dataflow of the model shows that steps included are contextual and has a vast depiction of the work flow in a production medium.

5.1 Data Collection

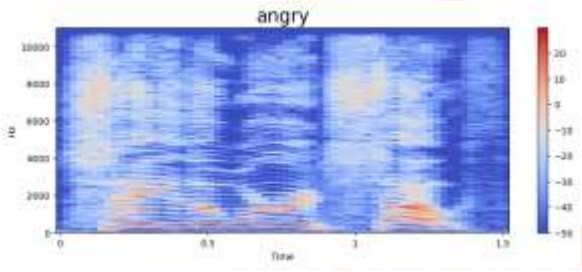
Data is gathered from various Royality-free sound production and Kaggle web space which featured numerous audio datasets on speeches of various actors, machinery, indoor and environmental sounds. We tried to integrate all the collected datasets into a single unit of source file. This process gave more computational issues. Hence, the data is made to upload each file iteratively considering every dataset to the processing and training of the model we require.

The dataset is being processed on Google's Colaboratory which provides a ground to perform data analysis and machine learning tasks with more performance mechanisms.

The decision to choose colab over other workbook is that Google's space provides additional computational RAM.

5.2 Data Processing

```
#Data Loading

paths = []
labels =[]
for dirname, _, filenames in os.walk('/datasetsource/input'):
for filename in filenames:
paths.append(os.path.join(dirname, filename))
label = filename.split('_')[-1]
label = label.split('.')[0]
labels.append(label.lower())
if len(paths) == 2800:
break

#Creating Dataframes

df = pd.DataFrame()
df['speech'] = paths
df['label'] = labels
df.head()

#Data processing
# Example case : anger

emotion = 'angry'
path = np.array(df['speech'][df['label']==emotion])[1]
data, sampling_rate = librosa.load(path)
librosa.display.waveshow(data)
spectogram(data, sampling_rate, emotion)
Audio(path)

#result image
```



The results of these audio signals are represented in a spectrogram system. A spectrogram visualizes how the intensity of different frequencies in an audio signal changes over time. It provides a detailed representation of the audio content, allowing users to see the distribution of frequencies and their variations. In the context of emotion recognition and distress call analysis, spectrograms are valuable for identifying patterns and features that correspond to different emotional states or distress indicators. For example, certain emotions may be associated with specific patterns of intensity in certain frequency ranges. By analysing spectrograms, researchers and practitioners can gain insights into the underlying characteristics of the audio signal, which can be used to develop more accurate and effective algorithms for emotion recognition and distress call detection.

6. Machine Learning

The dataset's anomalies are cleaned and processed. On the further procedure we incorporate machine learning methodologies to build a model from the processed data units and make it reliable for prediction scenarios.

6.1 Feature Extraction

To extract required audio features to analyse and attain a training method to the system, Mel-frequency cepstral coefficients (MFCC) is used. This is a feature extraction method where the audio signal is divided into short frames, typically 20-40 milliseconds long. Each frame is then passed through a filterbank that mimics the human ear's frequency response, emphasizing frequencies that are more perceptually relevant. The log of the filterbank energies is computed, followed by a Discrete Cosine Transform (DCT) to decorrelate the features and reduce dimensionality. The resulting coefficients represent the spectral envelope of the audio signal in a compact form, capturing important characteristics such as timbre and pitch.

6.2 Feature Labelling

Post feature extraction phase, which resulted in converting 2800 audio files and calculating 40 features from these files. The units of calculated features were mapped with corresponding emotion quotient as per the requirement for the model. Updating the current code base with python's package, Scikit-learn. This is employed to scale input features, thus potentially improving model performance and convergence during training. Additionally, scikit-learn's train_test_split function can be utilized to split the dataset into training and testing sets, simplifying the evaluation of the CNN model's performance.

The model, which after few trials gave suitable responses by fitting it with Scikit-learn's OneHotEncoder. It is a preprocessing tool used to convert categorical variables into a numerical format suitable for machine learning algorithms. It operates by creating a binary matrix representation of categorical variables, where each column corresponds to a unique category and each row corresponds to an instance in the dataset. Initially, the encoder is fitted to the data, identifying the unique categories in the variable and assigning a unique integer value to each category. Subsequently, the transform method is applied to convert the categorical variable into a one-hot encoded representation. For each instance, this method generates a binary array, setting the value to 1 for the category the instance belongs to and 0 for all others. By default, the encoder returns a sparse matrix, which is memory-efficient for large datasets with numerous unique categories. It also offers the ability to handle unknown categories by ignoring them during transformation.

Extracted features were labelled respectively with the emotion labels that we required to train. The labels were mapped numerically to the features for corresponding procedure methodology.

This resulted in obtaining required features as numerical inputs for further build of Convolutional Neural Network model.

6.3 CNN model design

A Standard Sequential Convoultional Neural Network is preferred in training the model from the pre processed data that was obtained. The
selection criteria for sequential model is made because the model architecture is a straightforward stack of layers, where the output of one layer serves as the input to the next layer.

Sequential models are ideal for constructing simple architectures where the data flows sequentially through the layers, such as in typical CNNs for image or sequential data processing. It is easy to understand and implement, making it suitable for many common deep learning tasks.

The requirement of keras framework seems feasible as with the training a model, importing sequential method CNN model is determined for the audio processing and training.

The model further uses a Conv1D layer which is suitable for sequential data like audio. The first Conv1D layer has 64 filters with a kernel size of 3 and uses the 'relu' activation function. It takes an input shape of (40, 1), indicating 40 time steps and 1 feature (assuming a single-channel input). Further importing MaxPooling layer to reduce the dimensionality of feature maps.

By reducing the spatial dimensions of the feature maps, max pooling helps to control the number of parameters and computations in the network, which can help prevent overfitting and reduce the computational cost of the model. Additionally, max pooling can help make the learned features more invariant to small translations in the input, which can improve the model's ability to generalize to new data.

Construction

```
from keras.models import Sequential
from keras.layers import Dense, Conv1D, MaxPooling1D, Flatten,
Dropout

model = Sequential([
    Conv1D(filters=64, kernel_size=3, activation='relu',
input_shape=(40, 1)),
    MaxPooling1D(pool_size=2),
    Conv1D(filters=32, kernel_size=3, activation='relu'),
    MaxPooling1D(pool_size=2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(7, activation='softmax')
])

model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
model.summary()
```

Metrics obtained from the above CNN model are stated to be definite for the training of the model as the neural network seems to be performing at a higher accuracy. The metrics are results of how the model may perform for final tests and different use case events.

6.4 Model Training

The designed model is now subjected to train with 70 percent of the input data that is provided using random train test splitting method. The model is pushed to iterate of 50 times for obtaining the metrics regarding the results.
Epochs = 50*

After training the model, the resultant metrics are compared with the standard units and are displayed in a graphical representation

Trained accuracy and validation accuracy comparison

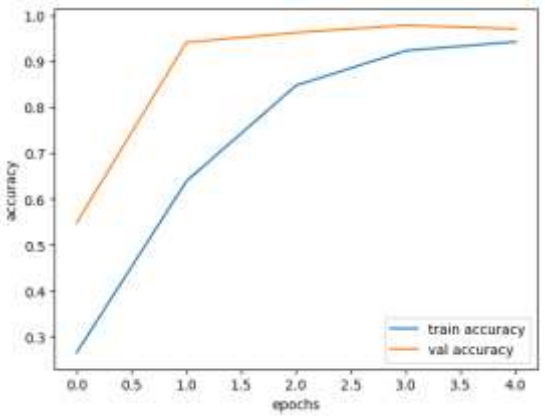


Fig. 2- Training and accuracy graph

Loss and Validation loss graph

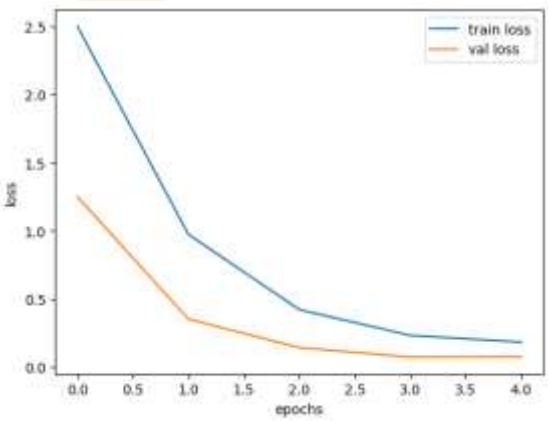


Fig. 3 – Loss and validation graph

6.5 Model Testing and Performance metrics

The model produced a seemingly satisfied results and the metrics involved with it are stated with the help of Sklearn's accuracy metrics component. The Convoluted model shows an accuracy of around 96 percent after iterative fine tuning of the system. This shows us that the performance entirely depends on the feature extraction that states the emotion as a different set of attributes in the training model.

These metrics, indicating the percentage of correctly classified instances, demonstrates the effectiveness of our model in accurately recognizing emotions from speech signals. In addition to accuracy, we evaluated our model using other performance metrics such as precision, recall, and F1-score. Precision measures the proportion of correctly predicted positive instances among all instances predicted as positive, while recall measures the proportion of correctly predicted positive instances among all actual positive instances. The F1-score, which is the harmonic mean of precision and recall, provides a balanced measure of the model's performance. These metrics help provide a comprehensive evaluation of the model's ability to detect different emotions in speech signals accurately.

7. MODEL EVALUATION

Since the model is now ready for new test case scenarios, there was a shortcoming while presenting it in a concise and suitable manner.
A thought of presenting the distress signal as a textual data is achieved but turning those distress signals into another audio signal for indication is what the initial model was supposed to perform. After many trials on various frameworks, that kind of output involving an audio indicator seemed to give inaccurate results as the audio files are not just a plain voice-only modules but also contained background noise to it. To eliminate the and other anomalies has become a complicated state of events that pushed us to perform the utilisation of clean audio files with different modulations of voices in various environmental settings as input training data.

Different frameworks were tested but the most feasible one was determined to be a web interface system. This web interface solves some of the major complications involving an audio distress detection model. Some of them are the elimination of complex mobile application interface. Utilising the APIs on constant basis to receive information to the mobile has been tested and has become not so ideal way of presenting the output signals as the need of distress call has to be taken in a timely basis, and the APIs request may not be the accurate basis of information retrieval in those scenarios.

The use of a full stack web application takes more framework integration, scaling the data needs a database. This results in operational costs which cannot be bore by the other members of the Project. This gives us an ultimate solution of using a simple GUI system which can carry all the tasks of a web application without actually using the entirety of the application.

The system needed a better working model as the result to show it as the face of the model developed. Many improvements were made but the integration with online service like colaboratory has been difficult to achieve. This brought us to look around a simple interface system that are both functional and light weight which can be easily incorporated with the existing model.

There are many Python induced web interface systems and the most popular ones are streamlit and gradio.

The model integration on streamlit was projected in a smooth manner but the mishap occurred during uploading a new audio through record option. This put the Streamlit interface to a hold. Lots of attempts to integrate streamlit to colaboratory also resulted negative.

This brought us to another web interface system which is, Gradio.
Gradio is a Python library that allows you to quickly create and share web-based interfaces for machine learning models. It provides a simple and intuitive way to deploy models, making them accessible to a wider audience without the need for expertise in web development or deployment. With Gradio, you can create interfaces for your models with just a few lines of code, allowing users to interact with the model through a web browser.

One of the key functionalities of Gradio is its ability to create input and output interfaces for your models. You can define input components such as text boxes, sliders, and image uploaders, allowing users to provide input data in a variety of formats. Gradio also supports a wide range of output components, including text boxes, images, and plots, making it easy to visualize the model's predictions.

Gradio interfaces are highly customizable, allowing you to control the layout, style, and behaviour of the interface components. You can also add custom preprocessing and postprocessing functions to handle input and output data in the desired format.

This web-based interface made procedure a lot more linearly upwards. The input is well right received by the Gradio interface providing the message of distress detection as output system.

Another persistent issue was with the enormous anomalies present in audio files. The noise and different durations of the audio files made it difficult to process and load into the system. The duration of audio files varied from seconds to minutes. To trim down those files to a determined length. Even with all the accuracy improvements, the model showed miserable results while recording and uploading the audio file, this on iterative testing resulted in a conclusion that the audio contains many issues which includes high noise and low volume of the user. Background sounds dominating the emotional impact of the speech. Modulations and accents of different people were processed different. This resulted in a less accurate outcomes, repeatedly giving 'disgust' as the result. The issue is then sorted out by providing recorded data only when the environmental noise is low or in minute amounts.

The CNN model thus gave favourable outcomes with the testing data which is derived from the entire input dataset (30 percent). Considering the pre recorded and cleaned wav files gave better and accurate results.

8. USER INTERFACE

Working of Gradio web interface has made the CNN model's output results more feasible to view and communicate. This resulted in no cost integration into the colaboratory itself without involving complex database architecture or operational charges. The gradio interface has been beneficial in terms of the model requirement.

#Gradio code snippet

```
import gradio as gr
import librosa
import numpy as np
from scipy.io import wavfile
```

Then a function to predict the model is incorporated into the code space. With the ongoing iterations from the CNN model, inputs from various audio files are taken and are run across the given trained model.

```
# Create an interface

audio_input = gr.Audio()
output_text = gr.Textbox()
iface = gr.Interface(fn=model_prediction, inputs=audio_input,
outputs=output_text, title="Distress Detection")
```

we are using the Gradio library to create a web-based interface for a machine learning model that predicts distress in audio signals.

First, we define an audio_input variable using gr.Audio(), which represents the input component of the interface. This component allows users to upload audio files or record audio directly through their device's microphone.

Next, we define an output_text variable using gr.Textbox(), which represents the output component of the interface. This component will display the textual output of the model's prediction.

We then create the interface using gr.Interface(), where we specify the function model_prediction as the function that will be called when the user interacts with the interface. We set audio_input as the input component and output_text as the output component. Additionally, we provide a title for the interface, which will be displayed at the top of the interface.

When the interface is launched, users can upload an audio file or record audio, which will be passed to the model_prediction function. This function will use the machine learning model to predict distress in the audio signal and return the prediction as text, which will be displayed in the output_text component of the interface

```
# Launch the interface
iface.launch(debug=True)
```

The iface.launch(debug=True) function launches the Gradio interface in debug mode. When debug mode is enabled, additional information and error messages are displayed, which can be helpful for troubleshooting and development purposes.

By launching the interface in debug mode, you can monitor the behaviour of the interface, check for any issues or errors, and ensure that everything is functioning as expected. However, it's important to note that debug mode should be disabled in a production environment to prevent sensitive information from being exposed and to optimize the performance of the interface.

9. DISCUSSION AND RESULTS

The speech emotion distress detection system described in this study represents a significant advancement in the field of affective computing, with potential applications in various domains such as mental health, human-computer interaction, and customer service. By leveraging Convolutional Neural Networks (CNNs) and advanced audio processing techniques, the system is able to accurately classify emotions from speech signals, providing valuable insights into the emotional states of individuals.

The system's use of Mel-frequency cepstral coefficients (MFCCs) and spectrograms for feature extraction is another strength, as these features are known to capture important characteristics of speech signals related to emotion. Additionally, the use of data augmentation techniques helps to enhance the diversity of the training data, improving the model's ability to generalize to new, unseen data.

Furthermore, the system's user-friendly interface, implemented using the Gradio library, allows users to easily interact with the model and obtain real-time emotion predictions from their speech. This feature enhances the system's usability and accessibility, making it suitable for a wide range of applications and user groups.

The system's performance in real-world scenarios could be further enhanced by considering the nuances of emotional expression in speech. Emotions are complex and can vary widely between individuals and cultures, making it challenging to accurately classify them from speech signals alone. Future research could focus on incorporating contextual information, such as the speaker's tone of voice, facial expressions, and body language, to improve the accuracy of emotion detection.

Additionally, the system could benefit from continuous learning and adaptation mechanisms to improve its performance over time. By collecting feedback from users and incorporating this feedback into the model, the system could become more accurate and personalized, leading to better outcomes in real-world applications.

Furthermore, the system's ethical implications should be carefully considered, particularly regarding privacy and data security. Emotion detection from speech involves the processing of sensitive personal data, and measures should be in place to ensure that this data is handled responsibly and in accordance with relevant regulations and guidelines.

10. CONCLUSION AND FUTURE SCOPE

The project on speech emotion detection using Convolutional Neural Networks (CNNs) has demonstrated promising results and has significant implications for various fields, including mental health, human-computer interaction, and customer service. The high accuracy achieved by the model, reaching 96 percent, highlights its potential for accurately classifying emotions from speech signals. This level of accuracy is crucial for applications where understanding the emotional states of individuals is essential, such as in mental health diagnosis, customer sentiment analysis, and human-robot interaction.

The project's use of advanced audio processing techniques, such as Mel-frequency cepstral coefficients (MFCCs) and spectrograms, for feature extraction has been effective in capturing important characteristics of speech signals related to emotion. Additionally, the use of data augmentation techniques has enhanced the diversity of the training data, improving the model's ability to generalize to new, unseen data.

The user-friendly interface developed using the Gradio library has made the model accessible to a wide range of users, allowing them to easily interact with the model and obtain real-time emotion predictions from their speech. This feature enhances the usability and accessibility of the model, making it suitable for deployment in various applications and user groups.

Furthermore, the ethical implications of the project should be carefully considered, particularly regarding privacy and data security. Measures should be in place to ensure that sensitive personal data is handled responsibly and in accordance with relevant regulations and guidelines.

Looking ahead, there are several future prospects for the project. One direction for future research could be to explore the integration of contextual information, such as the speaker's tone of voice, facial expressions, and body language, to improve the accuracy of emotion detection. Additionally, continuous learning and adaptation mechanisms could be incorporated into the model to improve its performance over time.

There are various issues and other future prospects that include the integration with edge computing offers several potential benefits for the distress call detection system. By deploying components of the system on edge devices, such as IoT devices or edge servers, latency can be reduced, and real-time processing capabilities can be improved. This is especially important in disaster scenarios where timely response is critical. On the basis of technical advancements, edge computing can enhance the system's performance in remote or resource-constrained areas where access to centralized computing resources may be limited. By deploying components of the system on edge devices, the system can continue to operate even when connectivity to centralized servers is limited or unavailable.

Another Future scope of distress detection includes incorporating geospatial analysis into the distress call detection system that can significantly enhance its capabilities, particularly in terms of more accurate location tracking of distress signals. By leveraging geospatial data, such as GPS coordinates or location information embedded in distress calls, the system can provide responders with precise information about the location of individuals in distress.

On more example systems, use of blockchain technology can greatly enhance the security and integrity of distress signal data in the distress call detection system. Blockchain, as a decentralized and tamper-proof ledger, can provide a secure and transparent way to record distress signal events, ensuring that the data is immutable and cannot be altered or tampered with.

By implementing blockchain technology, distress signal events can be recorded in a decentralized manner, eliminating the need for a central authority to verify the authenticity of the data. This can help improve the trustworthiness of the data and ensure that it remains secure and unaltered.

One of many future use cases of this model is the integration of drone technology into the distress call detection system. This can significantly enhance its capabilities, particularly in covering large or remote areas where traditional methods may be impractical or insufficient. Drones equipped with distress signal sensors can be deployed to quickly and efficiently survey areas that are difficult to access, providing comprehensive coverage during emergencies.

One key advantage of using drones is their ability to cover large areas quickly and efficiently. By equipping drones with distress signal sensors, responders can quickly identify and locate individuals in distress, enabling faster response times and potentially saving lives.

11. REFERENCES

1. Kejriwal M, Zhou P. On detecting urgency in short crisis messages using minimal supervision and transfer learning. Soc Netw Anal Min. 2020;10(1):58. doi: 10.1007/s13278-020-00670-7. Epub 2020 Jul 8. PMID:32834866; PMCID: PMC7341028.

2. Shiqing Zhang , Shiliang Zhang , Tiejun Huang , Wen Gao
Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching
IEEE Transactions on Multimedia , volume 20 , issue 6 Posted: 2017

3. Tripathi, Anjali and Singh, Upasana and Bansal, Garima and Gupta, Rishabh and Singh, Ashutosh Kumar, A Review on Emotion Detection and Classification using Speech (May 15, 2020). Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020, Available at SSRN: https://ssrn.com/abstract=3601803 or http://dx.doi.org/10.2139/ssrn.3601803

4. D. Bertero and P. Fung, "A first look into a Convolutional Neural Network for speech emotion detection," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 2017, pp. 5115-5119, doi: 10.1109/ICASSP.2017.7953131.

5. Yu, F., Chang, E., Xu, YQ., Shum, HY. (2001). Emotion Detection from Speech to Enrich Multimedia Content. In: Shum, HY., Liao, M., Chang, SF. (eds) Advances in Multimedia Information Processing — PCM 2001. PCM 2001. Lecture Notes in Computer Science, vol 2195. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45453-5_71

6. Z. Tariq, S. K. Shah and Y. Lee, "Speech Emotion Detection using IoT based Deep Learning for Health Care," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 4191-4196, doi: 10.1109/BigData47090.2019.9005638.

7. S. K. Shah, Z. Tariq and Y. Lee, "Audio IoT Analytics for Home Automation Safety," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 5181-5186, doi: 10.1109/BigData.2018.8622587.

8. M. Y. Alva, M. Nachamai and J. Paulose, "A comprehensive survey on features and methods for speech emotion detection," 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2015, pp. 1-6, doi: 10.1109/ICECCT.2015.7226047.

9. H. J. M. Steeneken and J. H. L. Hansen, "Speech under stress conditions: overview of the effect on speech production and on system performance," 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258), Phoenix, AZ, USA, 1999, pp. 2079-2082 vol.4, doi: 10.1109/ICASSP.1999.758342.

10. I. Shahin and N. Botros, "Modeling and analyzing the vocal tract under normal and stressful talking conditions," Proceedings. IEEE SoutheastCon 2001 (Cat. No.01CH37208), Clemson, SC, USA, 2001, pp. 213-220, doi: 10.1109/SECON.2001.923118.

11. K. V. Krishna Kishore and P. Krishna Satish, "Emotion recognition in speech using MFCC and wavelet features," 2013 3rd IEEE International Advance Computing Conference (IACC), Ghaziabad, India, 2013, pp. 842-847, doi: 10.1109/IAdCC.2013.6514336.

12. D. Muttaqin and S. Suyanto, "Speech Emotion Detection Using Mel-Frequency Cepstral Coefficient and Hidden Markov Model," 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2020, pp. 463-466, doi: 10.1109/ISRITI51436.2020.9315433.

13. D. S. Widyowaty and A. Sunyoto, "Accent Recognition by Native Language Using Mel-Frequency Cepstral Coefficient and K-Nearest Neighbor," 2020 3rd International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2020, pp. 314-318, doi: 10.1109/ICOIACT50329.2020.9332026.