International Journal of Advanced Technology in Engineering and Science
Vol. No. 13, Issue No. 03, March 2025
www.ijates.com
ijates
ISSN 2348 - 7550

Smart Feedback System for Real-Time Presentation Assessment

Prof. Prajwala. R. Patil[1], Dr. K. B. Manawade[2], Miss. Sanika Parit[3], Miss. Haviya. M. Dhannur[4]

[1]*Department of Computer Science Engineering, Hirasugar Institute of Technology, Nidasoshi, Karnataka, India*

ABSTRACT

The "Smart Feedback System for Real-Time Presentation Assessment" is a system designed to provide automated, real-time feedback to presenters by leveraging machine learning and computer vision technologies. It captures and analyzes various aspects of a presentation, such as speech clarity, pace, volume, body language, gestures, and facial expressions, offering immediate feedback on these performance metrics. The system aims to help presenters improve their delivery by providing actionable insights and scores during the presentation, making it useful in educational, professional, and training environments where presentation skills are critical.

1. INTRODUCTION

Delivering an effective presentation is a critical skill in education, business, and beyond. However, evaluating the quality of a presentation in real-time is often subjective, inconsistent, and time-consuming. To address these challenges, a Smart Feedback System for Real-Time Presentation Assessment offers a technological solution that leverages artificial intelligence and advanced analytics. The primary purpose of the Real-Time Feedback System for Student Presentations is to enhance the educational experience by providing students with immediate, actionable feedback during their presentations. By leveraging mobile phones for high-quality video and audio capture and utilizing Python for real-time data processing, this system aims to improve students' public speaking and presentation skills through in-depth analysis and constructive feedback.

Key objectives include:

1. **Enhancing Presentation Skills**: By delivering instant feedback, the system helps students identify their strengths and areas for improvement, promoting continuous skill development in public speaking and presentations.
2. **Promoting Engagement and Learning:** Real-time feedback fosters an interactive and supportive learning environment, encouraging students to engage more actively in their educational journey.
3. **Utilizing Advanced Technologies:** The project incorporates advanced Python libraries like OpenCV, MediaPipe, and Speech Recognition to analyse various aspects of presentations, such as body language, gestures, and speech clarity.
4. **Providing Accessible Feedback:** A user-friendly web interface developed with Django ensures that students can easily access and understand the feedback,making it an effective tool for personal development.

International Journal of Advanced Technology in Engineering and Science
Vol. No. 13, Issue No. 03, March 2025
www.ijates.com
ijates
ISSN 2348 - 7550

By addressing these objectives, the Real-Time Feedback System aims to create a dynamic and responsive educational tool that supports students in honing their presentation skills, ultimately contributing to their academic and professional success.

2. METHODOLOGY

2.1 Hardware-

- Processor: Intel Core i5
- RAM: 8 GB to 16 GB
- Storage: 256 GB SSD to 512 GB SSD
- GPU: Integrated GPU
- Camera: 720p (min), 1080p
- Microphone: Good-quality for voice capture
- Internet: 10 Mbps or higher
- Display: Full HD (1080p) monitor

2.2 Software-

- OS: Windows 10 or higher
- Languages: Python 3.x, HTML, CSS, JavaScript
- Frameworks: TensorFlow/Keras, OpenCV, Mediapipe, Django
- Database: MySQL
- IDE: VSCode
- Web Server: Apache
- Browser: Chrome, Firefox

Data Collection

Data will be sourced from platforms like Kaggle, UCI Machine Learning Repository, and other open data platforms. These platforms offer a wealth of resources in terms of images, audio files, speech datasets, and text annotations **Feature Extraction**

Volume and Pace, Gesture features, Body posture, Facial expressions, NLP Features.

Algorithm Development and Model Training

- Volume and Pace Analysis Models- Regression model using supervised learning
- Gesture, Posture and Emotion Recognition Models- Convolutional Neural Network (CNN)
- Natural Language Processing- BERT based model

UI Design

- **Dashboard Overview:** Real-time metrics for volume, pace, gestures, posture, emotion.Timer to track presentation duration
- **Feedback Panel:**

1. **Volume Feedback:** Gauge for loudness levels.
2. **Pace Feedback:** Visual indication for speed.
3. **Gesture/Posture Feedback:** Suggestions for posture and hand.

Validation and Testing

By comparing system generated feedback with expert evaluations to assess accuracy.

2.3. System Architecture

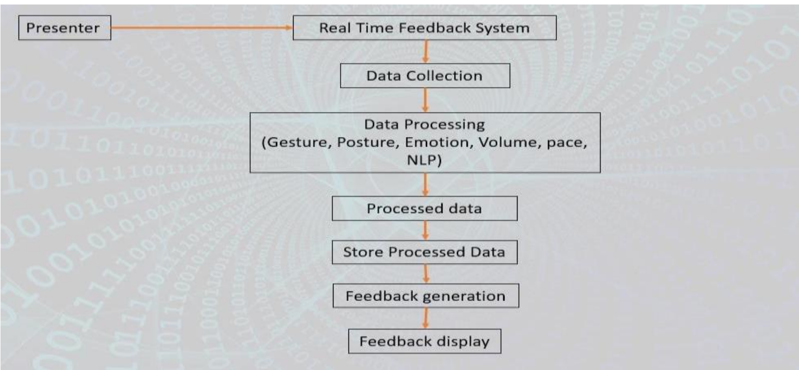

Fig: Block diagram of the proposed systems

3. DESIGN AND IMPLEMENTATION

System Overview

1. **Video and Audio Capture**:

- **Mobile Devices**: The system utilizes mobile phones to capture high-quality video and clear audio of student presentations.
- **Stability**: A stable connection between the mobile devices and the central server ensures uninterrupted video and audio streaming for real-time processing.

2. **Real-Time Processing**:

- **OpenCV for Video Processing**: The system employs OpenCV to process video feeds in real- time, enabling the detection and tracking of presenter movements and gestures.
- **MediaPipe for Gesture Recognition**: MediaPipe is used to recognize and analyse gestures, identifying common gestures such as hand movements, pointing, and changes in body posture.
- **SpeechRecognition for Audio Analysis**: The SpeechRecognition library processes and analyses audio feeds in real-time, detecting aspects such as speech clarity, volume, pace, and the use of filler words.
- **Low Latency**: The system ensures minimal latency (preferably under 1 second) in processing video and audio data to provide timely feedback.

3. **Feedback Generation**:

- **Immediate Feedback**: Based on real-time analysis of video and audio data, the system generates immediate feedback highlighting areas of strength and suggesting improvements.
- **Customizable Parameters**: Instructors can customize feedback parameters based on specific presentation criteria, allowing tailored feedback for different presentation contexts.

4. **User Interface**:

- **Web Interface**: A simple and intuitive web interface, developed using Django, presents feedback to students. The interface is designed to be accessible from any web browser and is responsive to different screen sizes, including desktops, laptops, tablets, and smartphones.

4. PSEUDO CODE

Model training

```
from keras.utils import to_categorical
from keras_preprocessing.image import load_img from
keras.models import Sequential
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D import
os import pandas as pd import
numpy as np
TRAIN_DIR = 'images/train'
TEST_DIR = 'images/test' def
createdataframe(dir):
image_paths = [] labels =
[]
for label in os.listdir(dir):
for imagename in os.listdir(os.path.join(dir,label)):
image_paths.append(os.path.join(dir,label,imagename)) labels.append(label)
    print(label, "completed")
return image_paths,labels
train = pd.DataFrame()
train['image'], train['label'] = createdataframe(TRAIN_DIR) print(train)
test = pd.DataFrame()
test['image'], test['label'] = createdataframe(TEST_DIR)
print(test) print(test['image'])
from tqdm.notebook import tqdm def
extract_features(images):
features = []
for image in tqdm(images):
img = load_img(image,grayscale =  True ) img = np.array(img)
    features.append(img) features =
np.array(features)
features = features.reshape(len(features),48,48,1) return
features
test_features = extract_features(test['image']) x_train =
train_features/255.0
x_test = test_features/255.0
from sklearn.preprocessing import LabelEncoder le =
LabelEncoder()
le.fit(train['label'])
y_train = le.transform(train['label'])  y_test
```

```
= le.transform(test['label'])
y_train = to_categorical(y_train,num_classes = 7) y_test =
to_categorical(y_test,num_classes = 7) model =
Sequential() # convolutional layers
model.add(Conv2D(128, kernel_size=(3,3), activation='relu',
input_shape=(48,48,1))) model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))
model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2))) model.add(Dropout(0.4))
model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2))) model.add(Dropout(0.4))
model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2))) model.add(Dropout(0.4))
model.add(Flatten())
# fully connected layers model.add(Dense(512,
activation='relu')) model.add(Dropout(0.4))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
# output layer
model.add(Dense(7, activation='softmax'))
model.compile(optimizer='adam',
loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(x= x_train,y = y_train, batch_size = 128, epochs = 100, validation_data = (x_test,y_test))
model_json = model.to_json()
with open("emotiondetector.json",'w') as json_file:
    json_file.write(model_json)
    model.save("emotiondetector.h5")
```

5. RESULTS:

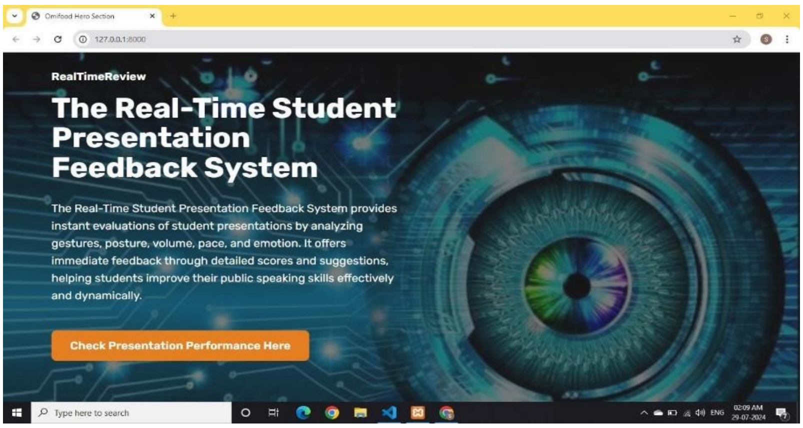

Fig: Introduction Page of system

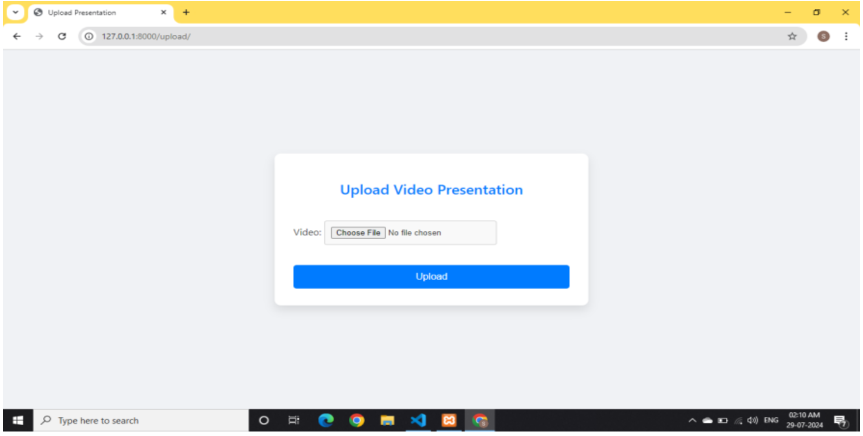

Fig: Presentation video uploading page

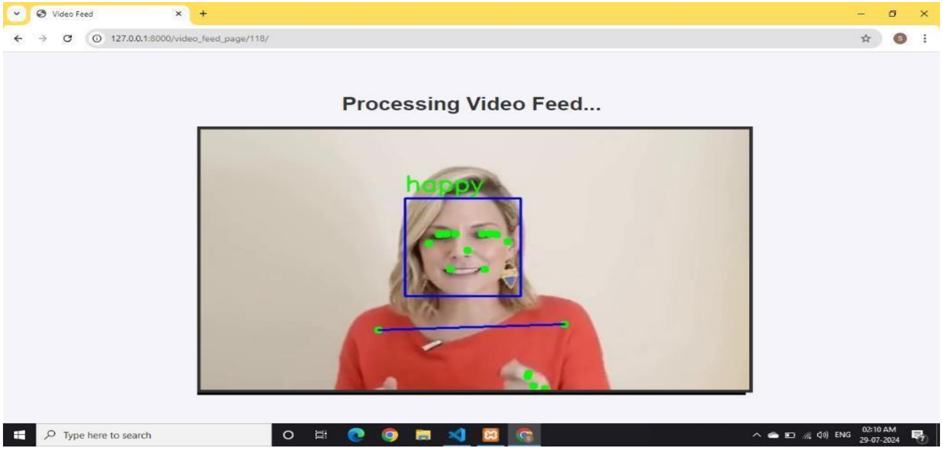

Fig : Video under processing

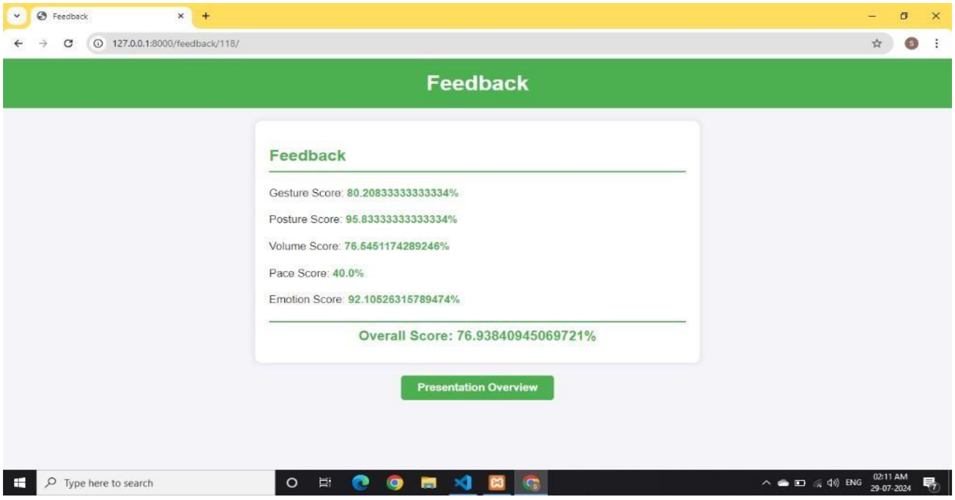

Fig: Generated feedback score

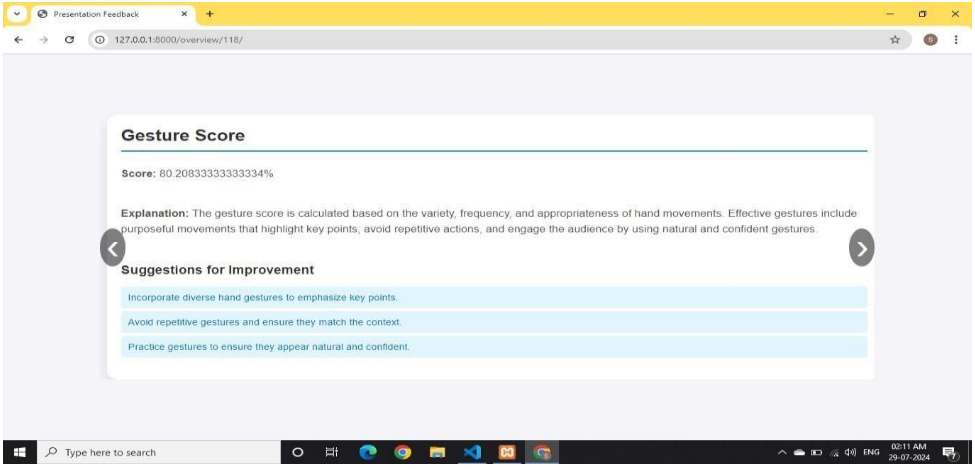

Fig: Feedback overview

6. CONCLUSION

The Real-Time Feedback System for Student Presentations is a comprehensive solution aimed at enhancing the educational experience by providing students with immediate and actionable feedback on their presentation skills. By utilizing mobile phones for high-quality video and audio capture and leveraging powerful Python libraries such as OpenCV, MediaPipe, and SpeechRecognition for real-time data processing, the system offers a robust platform for improving students' public speaking abilities.

The system's design ensures scalability, performance, and ease of use, making it a valuable tool for both students and instructors. By providing instant feedback, the system helps students identify and improve their presentation skills, fostering continuous improvement and confidence in public speaking.

REFERENCES

[1] H. Abdi, "Presentations: A real-time feedback tool using machine learning," *IEEE Access*, vol. 8, pp. 120- 130, Feb. 2020.

[2] M. Chaturvedi, P. Kumar, and S. Gupta, "Real-Time Gesture Recognition Using Machine Learning for Presentation Feedback," *IEEE Access*, vol. 9, pp. 25410-25421, Feb. 2021.

[3] Y. Li, X. Yu, J. Yang, and Y. Yang, "Automatic Assessment of Presentation Skills Using Computer Vision Techniques," in *Proc. IEEE Conf. on Multimedia and Expo*, Hong Kong, July 2017, pp. 894-899.

[4] A. K. Verma, A. Jha, and P. Kumar, "Emotion Detection and Feedback System Using Deep Learning," *IEEE Transactions on Affective Computing*, vol. 12, no. 2, pp. 459-470, April 2021.

[5] F. Ahmed and M. U. Akram, "Gesture and Posture Recognition in Presentation Feedback Using Mediapipe and OpenCV," in *Proc. IEEE Int. Conf. on Signal Processing and Communications*, Dec. 2020, pp. 1332-1338.