iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72052

AI based Two-Stream CNN-based Hand Gesture Recognition System for PowerPoint Control

Karthikeyan S[1], Sowmiya S[2]
[1]Assistant Professor, II MCA, [2]Department of Master of Computer Applications, Er.Perumal Manimekalai College of Engineering, Hosur,

Abstract: In today's digital world, using a slideshow for presentations is a great way to share information & impress people. Speakers can control their slides using devices like a mouse, keyboard, or even a laser pointer. However, traditional devices, like keyboards or remotes, don't always make things easy. Sometimes, presenters need to be close to the screen or interact directly, which can identify the hand and its landmarks using a hand detection algorithm.

That's where hand gesture recognition technology comes into the picture. This cool tech helps with smoother & more interactive presentations. This project introduces a system that uses hand gestures to control PowerPoint. It works with a special model called a Two-Stream Convolutional Neural Network (CNN). The goal is to create an easy and quick way to manage Microsoft PowerPoint presentations just by moving hands. The Two-Stream CNN looks at both the still images and the movement of hands. The first stream focuses on fixed hand positions, spotting important points & how they relate in each gesture.

Keywords: Hand gesture recognition, hand detection algorithm, presentation control, keyboards or remotes, Two-Stream Convolutional Neural Network (CNN), landmarks.

I. INTRODUCTION

A gesture is a form of non-verbal communication or non-vocal communication in which visible bodily actions communicate particular messages, either in place of, or in conjunction with, speech. Gestures include movement of the hands, face, or other parts of the body. Gestures differ from physical non-verbal communication that does not communicate specific messages, such as purely expressive displays, proxemics, or displays of joint attention. Gestures allow individuals to communicate a variety of feelings and thoughts, from contempt and hostility to approval and affection, often together with body language in addition to words when they speak. Gesticulation and speech work independently of each other, but join to provide emphasis and meaning. Hand gestures, are gestures performed by one or two hands. For movements involving the rest of the body, see gesture. Some hand gestures are closely tight to speech; some are like word themselves. The gestures listed below of such kind, they are so-called emblems, or emblematic gestures (Ekman & Friesen, 1972) or quotable gestures (Kendon 2004). These gestures are conventionalized and culture specific. This means that this type of gestures has a fixed meaning that can be verbalized in a couple of words within one culture. The same hand shape may mean something different in another culture. There are not only culture specific emblems, even within a culture there are gestures that are specific to a sub-community of the population. Hence, the gestures that can be found below may mean something specific within a small group of people. Gestures are hard to categorize because they're often culture-specific and ever-changing. The previous "call me" gesture, holding a pinkie and thumb to your ear and mouth like a handheld phone, has been replaced with a flat palm to indicate a smartphone.

A. Proposed Work

The proposed system is a real-time hand gesture recognition interface designed to control Microsoft PowerPoint presentations using a Two-Stream Convolutional Neural Network (CNN) that analyzes both spatial and temporal features of hand movements to accurately interpret and execute slide navigation commands without the need for physical input devices.

B. Methods

1) PowerPoint Controller Web App

The PowerPoint Controller Web App is meticulously crafted using Python and Flask, leveraging MySQL for database management and Wampserver for local hosting. Integration of TensorFlow enables efficient gesture recognition, while Pandas and Scikit Learn handle data processing and analysis.

Matplotlib, NumPy, Seaborn, and Pillow enhance visualization and image processing, ensuring a seamless user experience. Bootstrap ensures a responsive and visually appealing interface, culminating in a robust and user-friendly presentation control solution. The PowerPoint Controller Web App streamlines presentation control with intuitive hand gestures. Through robust authentication, users securely access the platform for managing presentations, navigating slides, and controlling presentation flow. Real-time feedback ensures seamless interaction, while accessibility features cater to diverse user needs. With stringent security measures and insightful analytics, users can trust their data's safety and gain valuable insights. Customization options enhance user experience, and the integration of the GestureNet Model enables precise gesture recognition, making presentation control effortless and engaging.

II. END USER DASHBOARD

A. Admin Functionality

Login: Admins authenticate securely to access the dashboard, ensuring data protection and system integrity.

Upload Gesture Dataset: Admins have the capability to upload datasets essential for training the GestureNet Model, ensuring the availability of relevant data for model development.

Train the GestureNet Model: This module initiates the training process of the GestureNet Model using the uploaded dataset. Admins oversee and manage the model training process to ensure optimal performance.

B. User Functionality

Register
Login
Upload PPT
PowerPoint Management
Control the PPT

III. GESTURENET MODEL: BUILD AND TRAIN

The GestureNet Model Build and Train submodule involves constructing and training a convolutional neural network (CNN) to recognize hand gestures effectively. It includes importing datasets, preprocessing data for consistency, extracting features through convolutional layers, classifying gestures, and deploying the trained model into the PowerPoint Controller Web App for real-time recognition and control.

A. Dataset Description

HaGRID (HAnd Gesture Recognition Image Dataset) for hand gesture recognition (HGR) systems. HaGRID size is 716GB and dataset contains 552,992 FullHD (1920 × 1080) RGB images divided into 18 classes of gestures. Also, some images have no_gesture class if there is a second free hand in the frame. This extra class contains 123,589 samples. The data were split into training 92%, and testing 8% sets by subject user-id, with 509,323 images for train and 43,669 images for test. The dataset contains 34,730 unique persons and at least this number of unique scenes. The subjects are people from 18 to 65 years old. The dataset was collected mainly indoors with considerable variation in lighting, including artificial and natural light. Besides, the dataset includes images taken in extreme conditions such as facing and backing to a window. Also, the subjects had to show gestures at a distance of 0.5 to 4 meters from the camera.

B. Import Dataset and Visualization

The Import Dataset module is integrating the HaGRID dataset into the hand gesture recognition system. It includes functionalities for accessing dataset files, parsing image data and labels, and splitting the dataset into training and testing sets. Optionally, it may incorporate data augmentation techniques to enhance dataset diversity.

C. Preprocessing

The preprocessing stage is for preparing the HaGRID dataset for effective utilization in the hand gesture recognition system. Here's a detailed description of each preprocessing step:

- Resize

All images in the dataset are resized to a standard size of 640x640 pixels to ensure uniformity and consistency across the dataset. Resizing helps to standardize the input dimensions, facilitating efficient processing and model training.

Resized Image (I_resized) = resize (I_gray, (new_height, new_width))

Formula: $I_{resized}$ =resize (I_{gray},(new_height,new_width))

Standardizes image sizes across the dataset, ensuring uniformity and consistency in subsequent processing steps and model training.

- RGB to Grey Conversion

RGB images are converted to grayscale, reducing the color channels from three (red, green, blue) to one. This simplification aids in reducing computational complexity while preserving essential image features for gesture recognition.

Grayscale Image (I_gray) = (0.2989 * R) + (0.5870 * G) + (0.1140 * B)

Formula: I_{gray}=0.2989×R+0.5870×G+0.1140×B

This conversion simplifies processing and reduces computational complexity while preserving relevant information for Gesture detection.

- Noise Filter using Median Filter

A median filter is applied to the grayscale images to reduce noise and smooth out pixel intensity variations. The median filter is effective in removing impulse noise and preserving edge details, resulting in cleaner and more robust images for subsequent processing.

Filtered Image (I_filtered) = filter2D(I_resized, -1, kernel)

Formula: $I_{filtered}$=filter2D($I_{resized}$,−1,kernel)

Applies Median filters to reduce noise and enhance image quality, improving the clarity and reliability of gesture features extracted during subsequent analysis.

- Binarization

Binarization converts the grayscale images into binary images by thresholding pixel intensity values. This step helps to enhance image contrast and segment hand gestures from the background, making them easier to detect and analyze.

Binary Image (I_binary) = threshold (I_filtered, threshold_value)

Formula: I_{binary}=threshold($I_{filtered,}$threshold_value)

Enhances contrast and facilitates segmentation by distinguishing between gesture and non-gesture regions, laying the groundwork for accurate gesture localization.

- RPN Segmentation

Region Proposal Network (RPN) segmentation is utilized to segment hand regions from the binary images. RPN identifies potential regions of interest (ROIs) within the image, focusing on areas likely to contain hand gestures. This segmentation step helps to isolate and extract relevant features for gesture recognition.

Identified Regions (regions) = RPN(I_binary)

Formula: regions= regions=RPN(I_{binary})

Utilizes a Region Proposal Network (RPN) to identify and localize potential gesture regions within the preprocessed images, enabling precise delineation and localization of suspicious areas for subsequent analysis and classification.

By performing these preprocessing steps, the HaGRID dataset is transformed into a standardized and enhanced form, optimized for subsequent analysis and model training in the hand gesture recognition system.

D. Feature Extraction

Feature extraction plays a pivotal role in extracting relevant information from input data, enabling effective pattern recognition and classification in hand gesture recognition systems. Here's a detailed description of each component involved in feature extraction:

- Convolutional Layer
- Activation Layer
- Pooling Layer

E. Classification

Classification using a fully connected layer is a crucial step in the hand gesture recognition system, enabling the model to predict the corresponding gesture classes, such as Start, Stop, Next Slide, and Previous Slide. Here's a detailed description of the classification process:

- Input Features

The input to the fully connected layer consists of high-level features extracted from the preceding layers of the neural network. These features encode the relevant spatial and temporal information extracted from hand gesture images, captured through convolutional and pooling layers.

- Fully Connected Layer

The fully connected layer, also known as a dense layer, connects every neuron in the preceding layer to every neuron in the subsequent layer. Each neuron in the fully connected layer receives input from all the neurons in the preceding layer, making it capable of capturing complex feature interactions. In the context of hand gesture recognition, the fully connected layer takes the flattened feature vector as input and performs a series of matrix multiplications followed by bias addition and activation functions to transform the input into class scores. The number of neurons in the fully connected layer corresponds to the number of output classes, with each neuron representing a class label. For example, in a multi-class classification task with four gesture classes (Start, Stop, Next Slide, Previous Slide), the fully connected layer would typically have four output neurons.

- Activation Function

Each neuron in the fully connected layer is associated with an activation function, typically a softmax function in multi-class classification tasks. The softmax function computes the probability distribution over the output classes, ensuring that the predicted probabilities sum up to one.The softmax activation function maps the raw class scores obtained from the fully connected layer to probabilities, indicating the likelihood of each class given the input features. This enables the model to make confident predictions about the gesture classes.

- Loss Function and Optimization

During training, the model compares the predicted class probabilities with the ground truth labels using a suitable loss function, such as categorical cross-entropy.

F. Train and Build

The Gesture Net Model plays a central role in the hand gesture recognition system, utilizing Convolutional Neural Networks (CNNs) for effective feature extraction and classification of hand gestures. Here's a detailed description of the build and train process

- Model Architecture Design
- Training Process

IV. GESTURE RECOGNITION

Hand Gesture Recognition in real-time involves capturing live video input of hand movements and processing it to identify and interpret gestures accurately. Here's a detailed description of the process, including Hang Gesture Recognition using Two Stream Networks with the trained GestureNet Model:

A. Live Hand Video Capture

The system utilizes a webcam or other video input device to capture live video of hand movements in real-time. This live video feed serves as the input for hand gesture recognition. Video frames containing hand movements are continuously captured and processed to detect and recognize gestures as they occur.

Preprocessing: Before performing gesture recognition, the live video frames undergo preprocessing steps to enhance their quality and suitability for analysis. Preprocessing may include resizing the frames, converting them to grayscale, and applying noise reduction techniques to improve clarity.

B. Hand Gesture Recognition

The hand gesture recognition process employs Two Stream Networks, a deep learning architecture designed to analyze spatial and temporal features concurrently.

Spatial Stream: One stream of the network focuses on analyzing still frames of the live video feed. It identifies key points and spatial relationships within hand gestures using convolutional layers, activation layers, and pooling layers.

Temporal Stream: The other stream processes optical flow information between consecutive video frames, capturing the dynamic motion patterns associated with different gestures. This stream is essential for understanding the temporal aspects of hand movements.

Integration: By integrating the spatial and temporal streams, the system comprehensively analyzes hand gestures, capturing both static and dynamic features for accurate recognition.

Gesture Recognition and Response: As the live video frames are processed by the Two Stream Networks with the GestureNet Model, hand gestures are recognized and classified in real-time. Based on the recognized gestures, appropriate actions are triggered within the PowerPoint Controller Web App, such as advancing to the next slide, going back to the previous slide, starting or stopping the presentation, etc.

By employing Two Stream Networks with the trained GestureNet Model, the system achieves robust and accurate hand gesture recognition in real-time, providing users with an intuitive and interactive means of controlling PowerPoint presentations.

V. GESTURE INTERPRETATION

The Gesture Interpretation Module is responsible for analyzing the recognized hand gestures and interpreting them into meaningful actions within the PowerPoint presentation. Here's a detailed description of its functionality:

- Gesture Recognition Output Processing

The module receives the output from the hand gesture recognition system, which includes the recognized gesture classes or labels along with any associated confidence scores or probabilities. This information serves as the input for the gesture interpretation process, providing the module with the recognized gestures detected in the live video feed.

- Mapping Gestures to Presentation Actions

Each recognized hand gesture is mapped to a specific action within the PowerPoint presentation, such as advancing to the next slide, going back to the previous slide, starting or pausing the presentation, etc. This mapping is predefined based on the desired functionalities of the PowerPoint Controller Web App and the corresponding gestures identified during the training of the GestureNet Model.

- Action Triggering

Once the hand gestures are mapped to presentation actions, the module triggers the corresponding actions within the PowerPoint presentation software. For example, if the recognized gesture corresponds to the "Next Slide" action, the module sends a command to the PowerPoint software to advance to the next slide in the presentation.

- Real-time Interaction

The module operates in real-time, continuously monitoring the live video feed for recognized gestures and interpreting them into presentation actions on the fly. This real-time interaction enables seamless and intuitive control of PowerPoint presentations using hand gestures, enhancing the user experience and engagement during presentations.

By effectively interpreting recognized hand gestures into meaningful presentation actions, the Gesture Interpretation Module facilitates seamless and intuitive control of PowerPoint presentations, empowering users to navigate through slides and interact with presentation content effortlessly.

VI. PERFORMANCE EVALUATION

Performance evaluation metrics such as confusion matrix, accuracy, precision, recall, and F1-score are essential for assessing the effectiveness of the hand gesture recognition system. Here's a detailed description of each metric along with the corresponding formulas:

1) *Confusion Matrix*

A confusion matrix is a table that summarizes the performance of a classification model by comparing predicted classes with actual classes. It provides insights into the model's ability to correctly classify instances into different categories. The confusion matrix consists of four quadrants: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

2) *Accuracy*

Accuracy measures the overall correctness of the model's predictions and is calculated as the ratio of correctly classified instances to the total number of instances:

Accuracy = (TP + TN) / (TP + FP + TN + FN)

3) *Precision*

Precision quantifies the proportion of correctly predicted positive instances among all instances predicted as positive, indicating the model's ability to avoid false positives:

Precision = TP / (TP + FP)

Recall (Sensitivity)

Recall, also known as Sensitivity or True Positive Rate (TPR), measures the proportion of correctly predicted positive instances among all actual positive instances, assessing the model's ability to capture positive instances:

Recall = TP / (TP + FN)

4) *F1-Score*

F1-Score is the harmonic mean of Precision and Recall, providing a balanced measure of a model's performance. It combines Precision and Recall into a single metric, considering both false positives and false negatives:

F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

These performance evaluation metrics provide comprehensive insights into the accuracy, precision, recall, and overall effectiveness of the hand gesture recognition system, enabling quantitative assessment and optimization of its performance.

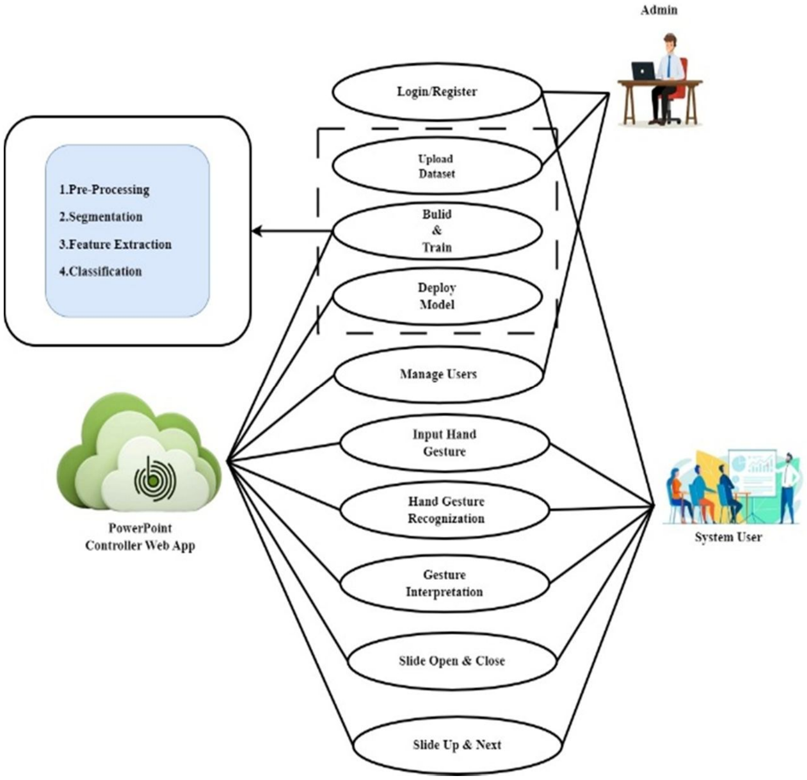


Figure 1: UML Diagram

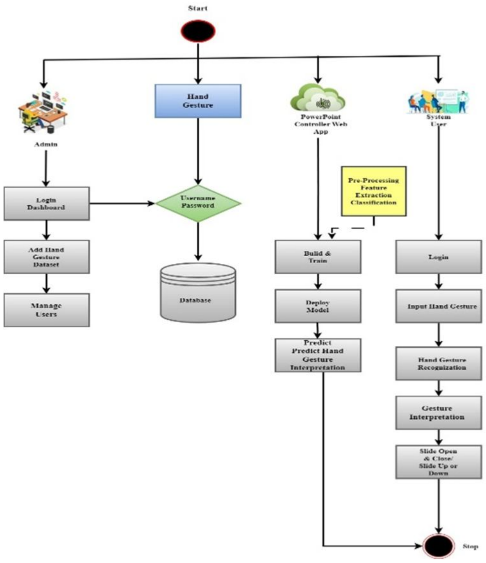


Figure 2:Activity Diagram

VII. RESULT

1. Test Case ID: TC001

- Input: User logs in with valid credentials.
- Expected Result: User is successfully logged in and redirected to the dashboard.
- Actual Result: User is logged in and directed to the dashboard.
- Status: Pass

2. Test Case ID: TC002

- Input: Admin uploads a gesture dataset.
- Expected Result: Dataset is successfully uploaded and stored in the database.
- Actual Result: Dataset is uploaded and accessible in the database.
- Status: Pass

3. Test Case ID: TC003

- Input: User registers with valid details.
- Expected Result: User account is created successfully.
- Actual Result: User account is created with the provided details.
- Status: Pass

4. Test Case ID: TC004

- Input: User uploads a PowerPoint presentation.
- Expected Result: Presentation is uploaded and added to the user's presentation list.
- Actual Result: Presentation is successfully uploaded and visible in the user's presentation list.
- Status: Pass

5. Test Case ID: TC005

- Input: Admin trains the GestureNet model with the uploaded dataset.
- Expected Result: GestureNet model is trained and deployed successfully.
- Actual Result: Training process completes without errors, and the model is deployed.
- Status: Pass

6. Test Case ID: TC006

- Input: User performs a start gesture during a presentation.
- Expected Result: Presentation starts playing from the first slide.
- Actual Result: Presentation begins playing from the first slide.
- Status: Pass

7. Test Case ID: TC007

- Input: User performs a next slide gesture during a presentation.
- Expected Result: Presentation advances to the next slide.
- Actual Result: Presentation successfully moves to the next slide.
- Status: Pass

8. Test Case ID: TC008

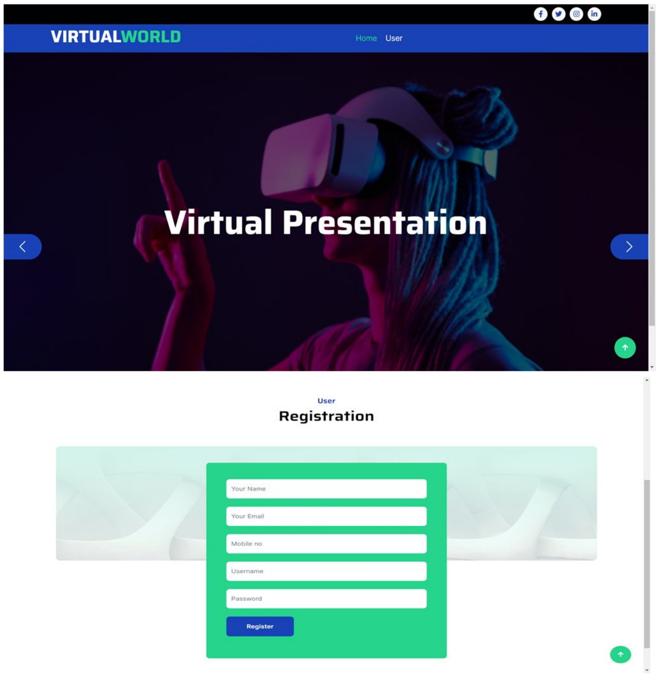
- Input: User performs a stop gesture during a presentation.
- Expected Result: Presentation pauses or stops playing.
- Actual Result: Presentation pauses or stops as expected.
- Status: Pass

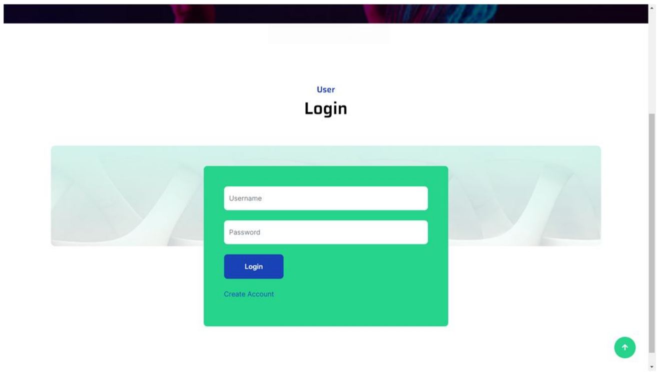
9. Test Case ID: TC009

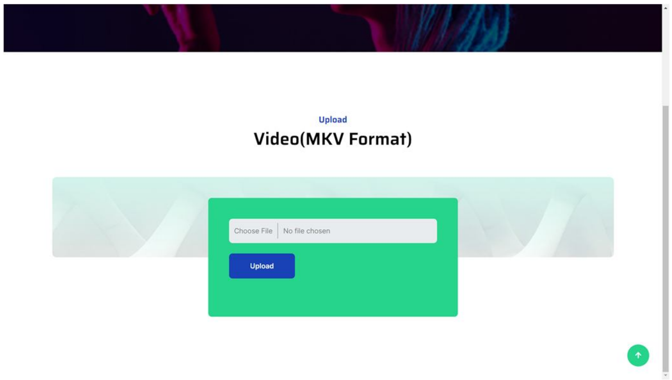
- Input: User logs in with invalid credentials.
- Expected Result: Login attempt fails, and appropriate error message is displayed.
- Actual Result: Login fails, and error message prompts user to enter valid credentials.
- Status: Pass

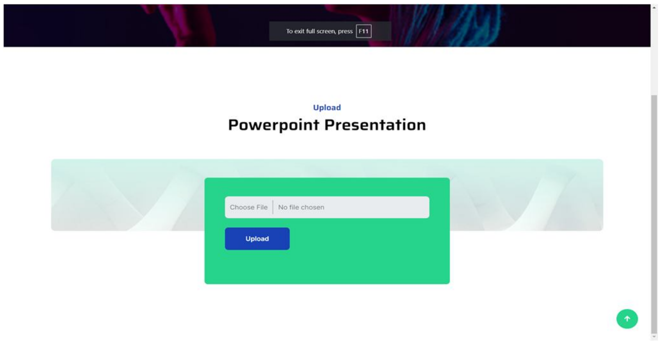
10. Test Case ID: TC010

- Input: User uploads a corrupted PowerPoint file.
- Expected Result: System detects the corrupted file and prompts the user to upload a valid file.
- Actual Result: Corrupted file is detected, and user is notified to upload a valid file.
- Status: Pass









VIII. CONCLUSION

The project concludes with the successful development and implementation of a comprehensive solution for controlling PowerPoint presentations using hand gestures. By integrating various technologies and methodologies, including web development frameworks, deep learning techniques, and real-time video processing, the system offers a sophisticated yet user-friendly platform for enhancing presentation control. Throughout the project, feasibility analysis played a crucial role in assessing the practicality and viability of the proposed solution. Technical feasibility was confirmed through the availability of suitable tools and technologies, such as Python, Flask, TensorFlow, and OpenCV, for implementing the required functionalities. Economic feasibility was established by evaluating the cost-effectiveness of developing and deploying the system compared to potential benefits and returns on investment

IX. ACKNOWLEDGMENT

The authors declare that they have no reports of acknowledgments for this.

REFERENCES

[1] K. S. Yadav, A. Monsley, K. Monsley and R. H. Laskar, "Gesture objects detection and tracking for virtual text entry keyboard interface", Multimedia Tools Appl., vol. 82, no. 4, pp. 5317-5342, Feb. 2023.

[2] J. Gangrade and J. Bharti, "Vision-based hand gesture recognition for Indian sign language using convolution neural network", IETE J. Res., vol. 69, no. 2, pp. 723-732, Feb. 2023.

[3] L. Liu, W. Xu, Y. Ni, Z. Xu, B. Cui, J. Liu, et al., "Stretchable neuromorphic transistor that combines multisensing and information processing for epidermal gesture recognition", ACS Nano, vol. 16, no. 2, pp. 2282-2291, Jan. 2022.

[4] J. P. Sahoo, A. J. Prakash, P. Pławiak and S. Samantray, "Real-time hand gesture recognition using fine-tuned convolutional neural network", Sensors, vol. 22, no. 3, pp. 706, Jan. 2022.