

Chapter 4: Experimental Results and Discussion

1.1. Introduction:

Implementation phase is the period where all components of the project are integrated; in the implementation phase we present our explanations about the project idea. The purpose of this section is to provide a clear concept of the feasibility and applicability of the project towards the desired goals. We will also be talking about our challenges that we had when implementing it, so that we can get a whole picture of our journey.

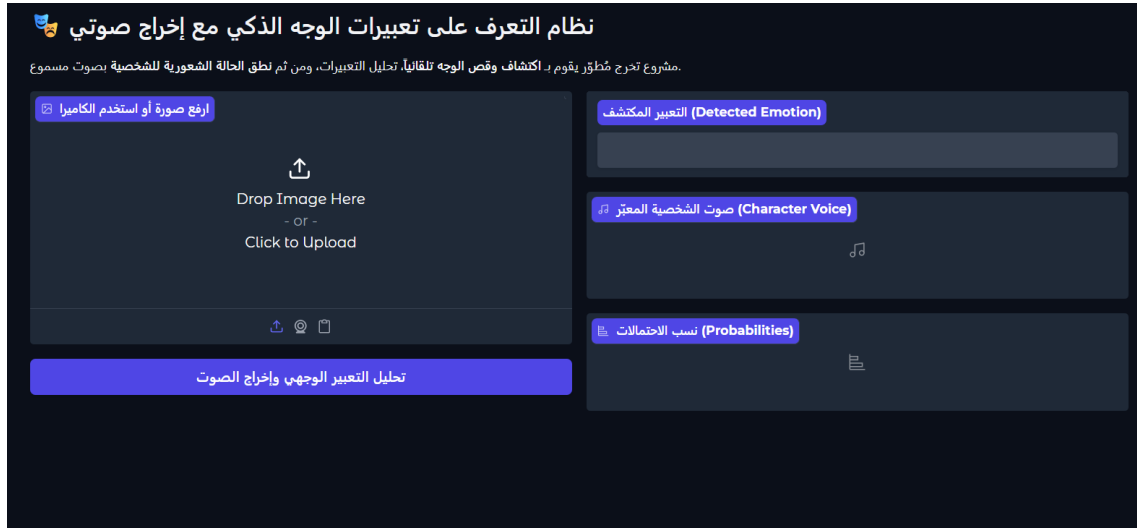


Figure 1 Component of the system

1.2. Prototype Implementation

The implementation process began by configuring the required deep learning environment and installing essential software dependencies, including TensorFlow/Keras, gTTS, OpenCV and Gradio

Rather than relying on heavy pre-trained architectures via transfer learning (such as VGGNet), we implemented a custom, lightweight, 4-block deep Sequential Convolutional Neural Network (Custom CNN) built with TensorFlow. This choice ensured low computational latency while maintaining high feature extraction

capability for facial expressions.

The preprocessing pipeline automatically detects faces and extracts a Region of Interest (ROI) around the face using OpenCV's Haar Cascade classifier. The cropped 48x48 grayscale face image is then passed to the Custom CNN model. Once the model outputs the predicted emotion category, real-time speech feedback is dynamically synthesized using the gTTS (Google Text-to-Speech) library.

The training and fine-tuning were executed within the Google Colab environment using GPUs. To handle real world image variations and class imbalances in the FER2013 dataset, advanced techniques such as Data Augmentation and a hard sample fine tuning phase were applied.



Figure 2 sample from dataset and the predicted output

User Interface Implementation:

To provide an intuitive and accessible interface especially designed for ease of use by visually impaired individuals we developed a web-based user interface using the Gradio framework.

The interface captures a live frame from the webcam, automatically performs face detection and cropping via OpenCV, passes the cropped image to our trained Custom CNN model ('FER2013_Final.h5'), displays the prediction probabilities across all 7

emotions, and immediately plays the generated `.mp3` emotional voice feedback via `gTTS` (e.g., "I am feeling very happy today!").



Figure 3 System Interface and Real-Time Emotion Classification View

1.3. Prototype Verification

The results of prototype testing are included in this section for the purpose of verifying the design requirements. Testing process consisted of acceptance tests and integration tests.

Table 1 Acceptance Test

Test Case Name:		Facial emotion recognition System – Camera module test	Test ID #:		FERS-01	
Description:		Capture the emotion image via webcam and return the corresponding vocalized audio response via speakers.	Type:		■ Black box	
Setup:		Webcam connected with system speakers, clear ambient lighting, and configured gTTS output stream.				
Step	Action	Expected Results	Pass	Fail	N/A	Comments
1	Connect a speaker or headphone	Sound is clear	✓			
2	Capture face by Gradio interface	Face is captured, detected, and cropped via OpenCV Haar Cascade.	✓			
3	Listen to spoken audio output	System vocalizes the predicted emotional state accurately via gTTS	✓			
Overall test result:			✓			

Table 2 Integration Test

Test Case Name:		Facial Emotion Recognition system – Camera with deep learning models	Test ID #:			FERS-02
Description:		Verify that the deep learning model accurately recognizes facial expressions across all 7 target classes.	Type:			■ black box
Setup:		Load the trained Custom CNN model (`FER2013_Final.h5`) into the Gradio/OpenCV prediction pipeline under standard lighting conditions.				
Step	Action	Expected Results	Pass	Fail	N/A	Comments
1	Load the deep learning model using python code	The Custom CNN model loads seamlessly using TensorFlow/Keras libraries.	✓			
2	Present "Happy" face expression	Model predicts "Happy" with high probability and speaks the happy response phrase.	✓			
3	Present "Sad" face expression	Model predicts "Sad" with high probability and speaks the sad response phrase.	✓			

4	Present "Angry" face expression	Model predicts "Angry" with high probability and speaks the angry response phrase.	✓			
5	Present "Neutral" face expression	Model predicts "Neutral" with high probability and speaks the neutral response phrase.	✓			
6	Present "Fear" face expression	Model predicts "Fear" with high probability and speaks the fear response phrase.	✓			
7	Present "Surprise" face expression	Model predicts "Surprise" with high probability and speaks the surprise response phrase.	✓			
Overall test result:			✓			