

Chapter 3: Proposed Approach

1.1. Introduction

The proposed approach adopted for the development of the proposed Facial Emotion Recognition (FER) system which aims to help visually challenged people to understand the facial expression by giving an auditory feedback is presented in this chapter. The chapter outlines the overall methodology adopted for designing and implementing the system starting from the identification of the subsequent requirements of the user and the system, followed by converting the requirements into engineering specifications that provide direction to the development process.

Moreover, this chapter explains the system architecture and the reasons behind the selection of the used hardware and software components in the system implementation. The selection of the embedded platform, camera module, and deep learning approaches used for facial emotion recognition is highlighted. Furthermore, a mapping relation between user requirements and engineering requirements is presented to make sure that the developed system meets the functional and non-functional objectives.

The key points of the proposed approach are very high recognition accuracy, fast response time, cost effectiveness and user-friendliness along with portability and reliability for assistive technologies. Thus, this chapter is the basis for the implementation and experimental evaluation of the subsequent chapters.

1.2. User Needs

User needs in projects are the needs, preferences and expectations of end users that are to be satisfied by the project deliverables or project outputs.

- An off-the-shelf general purpose embedded platform should be used.
- Facial expression recognition and classification with high accuracy should be possible in the system.
- The system should be easy to use: The user must take a snapshot of the person's face and then analyze the facial expression.
- The system should play back the identified emotion (s) into a speaker.
- The system needs to be flexible so that it can accurately identify a large variety of facial expressions.
- The system should be robust and accurate in detecting emotions from various people and under various illumination.
- The system should be cost-effective, both in terms of hardware and software.

- The system should possess high scanning speed that can be used to analyze facial expression quickly.
- The system should be designed to save power consumption.

1.3. Requirement Specification

- The system should be based on an off-the-shelf embedded platform as it performs better with deep learning models than other embedded platforms.
- The system needs to be able to identify and classify the facial expressions with a precision of at least 85%.
- The training data of the system should contain variety of facial expressions for the system to recognize with high accuracy.
- The design targets real-time operation with low latency
- The system shall recognize and output emotion in less than 3 seconds.
- The system should have (ON/OFF) to facilitate optimum use of energy.
- The system needs to provide the result of the recognition within 5 seconds in audible form.
- The project budget is restricted to 1000 SAR, and cannot exceed this amount.

1.4. Mapping User Requirements to Engineering Requiremen

Table 1 matching Marketing requirements to Engineering requirements

Marketing Requirements	Engineering Requirements	Justification
2,6	The system should use an off-the-shelf embedded platform, due to its superior performance with deep learning models compared to other options.	Using an off-the-shelf embedded platform simplifies the system's development and integration process while delivering the performance needed for deep learning models.
1	The system's training data should include a diverse range of facial emotion datasets (CK+, FER2013) to ensure high accuracy in recognition.	Diverse facial expression training data reduces bias and ensures high recognition accuracy across different demographics.
8	The system aims to achieve high recognition accuracy exceeding 80%, depending on training outcomes.	High accuracy ensures reliable recognition, providing visually impaired users with dependable emotional cues, which improves their social interactions.
2,5	The system should perform a quick scan of facial images within 3 seconds or less.	A quick scan minimizes latency, allowing users to receive results promptly and interact with others without delays.
2,3,8	The system shall classify and give an output of recognized emotion within 3 seconds	Fast classification and output ensure that users receive timely results, enabling them to understand emotions in real time.
4	The system should audibly output recognition results within 5 seconds.	Delivering the recognized emotion audibly within seconds ensures users can respond promptly to the emotional cues.
7,9	The system should include (ON/OFF) to optimize energy consumption.	Energy-saving modes optimize power consumption and extend battery life, ensuring reliable performance for mobile or wearable systems.

7	The project budget is limited to 1000 SAR and must not exceed this limit.	Limiting the budget ensures cost-effectiveness and accessibility, while still providing high-quality performance.
---	---	---

Marketing Requirements:

1. It is very important that the system is based on a general off-the-shelf embedded platform.
2. The system should be able to detect and categorize Facial emotions with high accuracy.
3. Easy to operate system, user takes an image of a facial emotion.
4. The system should play back the facial expression recognized verbally.
5. The system must be able to detect a variety of facial expressions.
6. The system should be reliable.
7. The system should be cost-effective.
8. The system must have a high speed in scanning.
9. The system should not use excessive electricity..

1.5. Overview of The System and Concept Design

The general off-the-shelf platforms that are available in the embedded system market with an appropriate programming language should be used in our FER design. It must be able to identify facial expressions accurately, and provide the voice feedback of the recognized facial emotion, so that the visually impaired can get timely, accurate and accessible feedback.

Table 2 Concept overview

Machine Learning	Hardware Platform	Camera	Speaker
Deep Learning (TensorFlow / CNN)	Google Colab	Laptop/PC Webcam	Speaker (Gradio Audio / gTTS)
Image Processing (OpenCV Haar Cascade)	Cloud & Local RAM	From Browser	

1.6. Machine/Deep learning:

Machine/Deep Learning is a subcategory of Artificial Intelligence, where learning and improvement take place through experience and without explicit programming.

Table 3 Technique comparison.

Method	Strengths	Weaknesses
Deep Learning	- Excellent accuracy, especially with large datasets. - Can handle complex data and recognize patterns with high precision. - Capable of improving its accuracy with additional data.	- Requires significant training time. - Requires large datasets. - May require specialized hardware or software.
Image Processing	- Excellent accuracy, especially with large datasets. - Can handle complex data and recognize patterns with high precision. - Capable of improving its accuracy with additional data.	- May not be as accurate as deep learning-based models. - May not handle complex recognition tasks.

There are two ways to go about our project, image processing and deep learning, as it pertains to the visually impaired. In image processing, the features are extracted manually and classifiers like Support Vector Machines or k-nearest Neighbors are used to detect emotions, for example, using Histograms of Oriented Gradients or Local Binary Patterns. These methods are less complex and more readily implemented, but are less accurate due to changes in lighting, facial expression and occlusion. However, deep learning, especially CNN automatically learns features from the data, thus deep learning has greater tolerance to these differences and greatly enhances the recognition accuracy. Such complex patterns in facial expressions are hard to detect from traditional machine learning models, but can be captured by deep learning models, which are able to process large and diverse datasets. Considering the requirements of high accuracy and real-time processing in assistive technologies for visually impaired users, deep learning is the most powerful and flexible approach that can be used for our project.

So, we chose: **Deep Learning.**

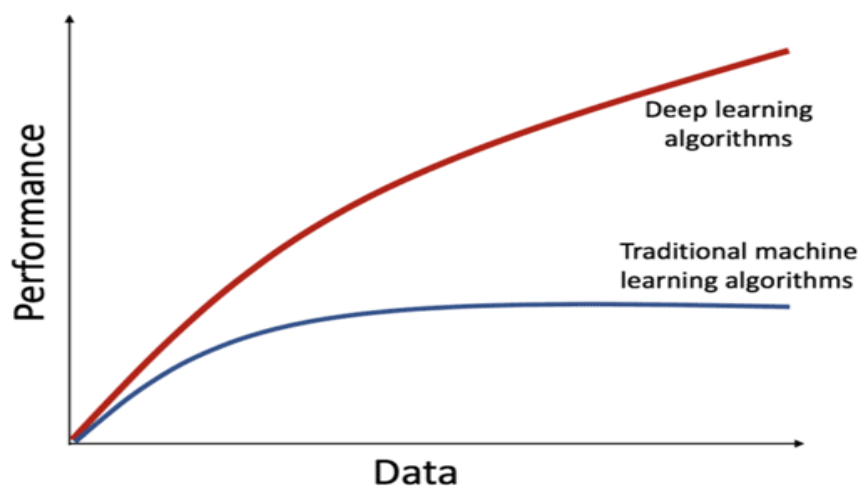


Figure 1 Deep Learning: Comparative Study

1.7. Hardware Platform

For our project, we utilized a hybrid computing platform approach combining Cloud Computing (Google Colab) for model training and a Personal Computer (PC/Laptop) for deployment and testing. Deep Learning models require significant computational power, which traditional microcontrollers cannot provide. Therefore, we evaluated the environment setup suitable for our software application.

Table 4 Hardware comparison

Method	Strengths	Weaknesses
Google Colab (Cloud)	- High-performance GPU acceleration - Eliminates local hardware strain. - Pre-installed Python libraries. - Seamless Google Drive integration for checkpoints.	- Requires a continuous high-speed internet connection. - Session timeout and usage limits on free tiers.
Personal Computer (PC/Laptop)	- Supports real-time local webcam testing. - No dependency on internet connection for prediction. - Easily hosts local web interfaces (Gradio).	- Limited training speed due to lack of high-end dedicated GPUs. - High power consumption during local model compilation.

We selected this hybrid approach because training our **Facial Expression Recognition (FER) CNN model** on the large FER2013 dataset requires heavy parallel processing, which is efficiently provided by Google Colab's cloud GPUs.

For the final deployment, the system runs smoothly on a standard PC/Laptop, utilizing the built-in webcam for real-time face capturing, the local CPU to run the Gradio interface, and the system speakers to deliver the emotional voice output generated by the gTTS module. This approach provides a high-accuracy, cost-effective, and fully functional software system without the need for expensive embedded hardware.

Table 5 Hardware comparison.

Local PC / Laptop (Local Environment)		Google Colab (Cloud Environment)
- A personal hardware system running a local OS (Windows/macOS), used to deploy the final user application.	Definition	- A cloud-based development environment designed for executing Python code and heavy machine learning tasks via web browsers.
- Utilizes built-in system hardware (Webcam for face capture, Speakers for audio output).	Number of USB ports	- Virtual cloud allocation (Depends on Google servers, no physical ports needed).
- Python (To execute the local Gradio interface and run inference using the .h5 model).	Languages Supported	- Python (Fully supported with pre-installed Deep Learning libraries like TensorFlow and Keras).
- 64-bit Standard Consumer Architecture (Intel Core / AMD Ryzen / Apple Silicon).	Architecture	- 64-bit Cloud Architecture (Intel Xeon / AMD EPYC specs).
- Standard CPU/GPU speeds optimized for running pre-trained real-time prediction (Inference).	Processor speed.	- High-performance cloud processing accelerated by dedicated Nvidia GPUs (T4 / A100).

1.8. Camera

The camera is an essential component that captures real-time facial images to be processed by the facial expression recognition pipeline. Since our application relies on real-time prediction via a web interface, we evaluated the camera capture methodology suited for our environment.

Table 6 Camera (Strengths & Weaknesses).

Capture Method / Type	Strengths	Weaknesses
Dedicated Embedded Camera (e.g., Raspberry Pi Camera)	- Compact for physical hardware integration.	- Requires hardware setup, driver configuration, and dedicated microcontrollers.
Integrated PC / Laptop Webcam (Web Browser Capture)	- Native integration with JavaScript and OpenCV. - No extra setup or hardware cost. - High resolution (HD/FHD) for accurate face detection. - Seamless real-time video stream in browser (Gradio).	- Requires hardware setup, driver configuration, and dedicated microcontrollers. - Dependent on the user's computer hardware setup.

We selected the Integrated Laptop/PC Webcam utilizing browser-based capturing (JavaScript media stream) combined with OpenCV (Haar Cascade) for automatic face detection and cropping.

This decision was made because our Deep Learning model requires clear, high-resolution facial images to accurately detect subtle expression variations. Using the integrated webcam allows the system to process incoming image frames at standard HD quality without requiring additional driver configurations or hardware extensions. Once captured, the frame is resized to 48×48 pixels and fed directly into our CNN model for real-time classification.

So, we chose: **Integrated Laptop/PC Webcam.**

1.9. Preliminary Work

The Datasets are a key component of any computer vision system. A benchmark dataset enables researchers to easily and fairly evaluate and validate various machine learning methods and techniques. In this project, we selected the widely recognized benchmark dataset ****FER2013 (Facial Expression Recognition 2013)****.

1.9.1. FER2013Dataset:

FER2013 is a popular open-source dataset designed for facial expression recognition tasks, introduced during the ICML 2013 Challenges in Representation Learning. The dataset consists of 35,887 grayscale, 48x48 pixel images categorized into 7 distinct facial expression classes: Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral. The images were gathered from various online sources to reflect real-world diversity in age, gender, pose, and lighting conditions, making it an ideal standard for training, testing, and fine-tuning our Convolutional Neural Network (CNN) model.



Figure 2 sample from FER2013

The reasons for using a Convolutional Neural Network (CNNs)

We utilized a customized Convolutional Neural Network (CNN) trained from scratch as the primary deep learning architecture for our Facial Expression Recognition (FER) system. CNNs are exceptionally well-suited for computer vision tasks as they automatically and adaptively learn a spatial hierarchy of visual features—ranging from low-level edges to complex facial expressions.

In this project, three convolutional feature extraction blocks followed by one fully connected classification block. Instead of using heavy pre-trained models (like VGGNet or DenseNet), which often require significant computational resources, our custom CNN was designed to be lightweight, fast, and highly effective for real-time inference.

To overcome dataset imbalance and improve accuracy on challenging expressions, our training pipeline was divided into two distinct phases:

1. Phase 1 (Initial Training): Training the CNN with Data Augmentation (ImageDataGenerator) and balanced class weights.
2. Phase 2 (Hard-Images Fine-Tuning): Extracting misclassified "hard" images from the validation set and retraining the model to refine decision boundaries.

Table 7 FER2013 Emotion Classes and System Mapping

Class	Emotions	Output Character Voice (gTTS)
0	Angry	"I am feeling angry right now."
1	Disgust	"This is disgusting."
2	Fear	"I am so scared!"
3	Happy	"I am feeling very happy today!"

4	Sad	"I feel a bit sad."
5	Surprise	"Wow! What a big surprise!"
6	Neutral	"I am feeling calm and neutral."

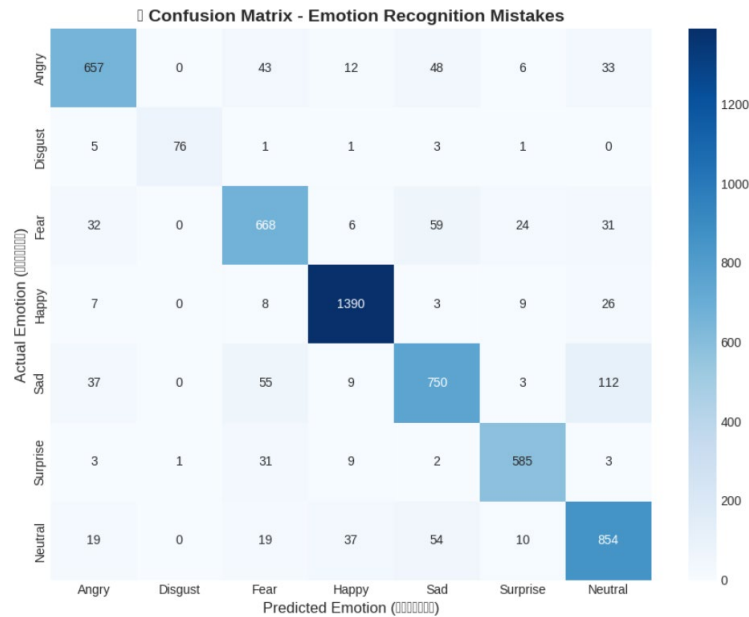


Figure 3 Confusion matrix

The Final Model (Custom CNN):

After evaluating our architecture and implementing the two-phase training strategy, our Custom Convolutional Neural Network (CNN) demonstrated optimal performance and accuracy for the facial expression recognition task. By building a custom lightweight architecture rather than relying on heavy pre-trained models like VGGNet or ResNet, we achieved an efficient balance between prediction accuracy and fast inference speed.

Using the computational resources of Google Colab, the final trained weights were saved as `FER2013\_Final.h5`. This trained model was then successfully integrated with our OpenCV face detection pipeline and the Gradio web interface, enabling real-time emotion analysis and emotional voice generation via `gTTS`.

Model Performance Evaluation and Confusion Matrix

To rigorously evaluate the classification efficacy of the proposed Custom Convolutional Neural Network (CNN) model, a comprehensive quantitative assessment was conducted on the unseen test subset of the FER2013 dataset. The evaluation encompasses all seven target facial emotion categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

The overall confusion parameters comprising True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) were extracted across all seven emotion classes to analyze the model's classification performance, as summarized in Table 9.

Class	Emotions	Output Character Voice (gTTS)
0	Angry	"I am feeling angry right now."
1	Disgust	"This is disgusting."
2	Fear	"I am so scared!"
3	Happy	"I am feeling very happy today!"
4	Sad	"I feel a bit sad."
5	Surprise	"Wow! What a big surprise!"
6	Neutral	"I am feeling calm and neutral."

Figure 8 Confusion Matrix of the Trained Custom CNN Model

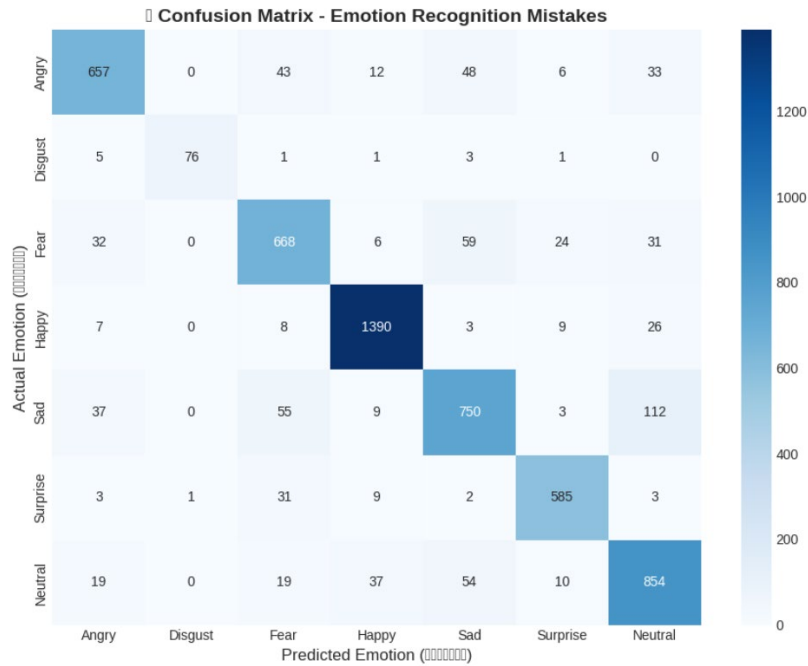


Figure 9 Confusion Matrix of the Trained Custom CNN Model

Furthermore, the customized design of our lightweight CNN architecture facilitates robust stability in feature extraction and classification across complex facial data. Overall, the model's performance highlights its strength, efficiency, and real-time adaptability, establishing a solid foundation for assistive technologies that support seamless social interaction for visually impaired individuals.

1.10. System Design Overview Design Level 0

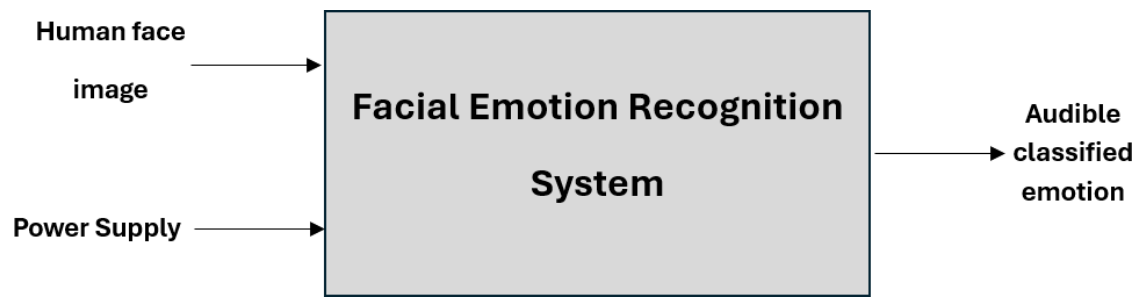


Figure 4 System design level 0.

Table 8 Detailed System Design

Module	FER & Voice Output System
Input	Human Face Image via Laptop/PC Webcam
Output	Audio Voice Output Signal (.mp3 via System Speakers)
Functionality	Captures a live frame via the web interface, detects and crops the face using OpenCV Haar Cascade, classifies the emotion using the trained Custom CNN model, and converts the predicted emotion to a spoken voice message using gTTS.

1.11. System Design Overview Design Level 1:

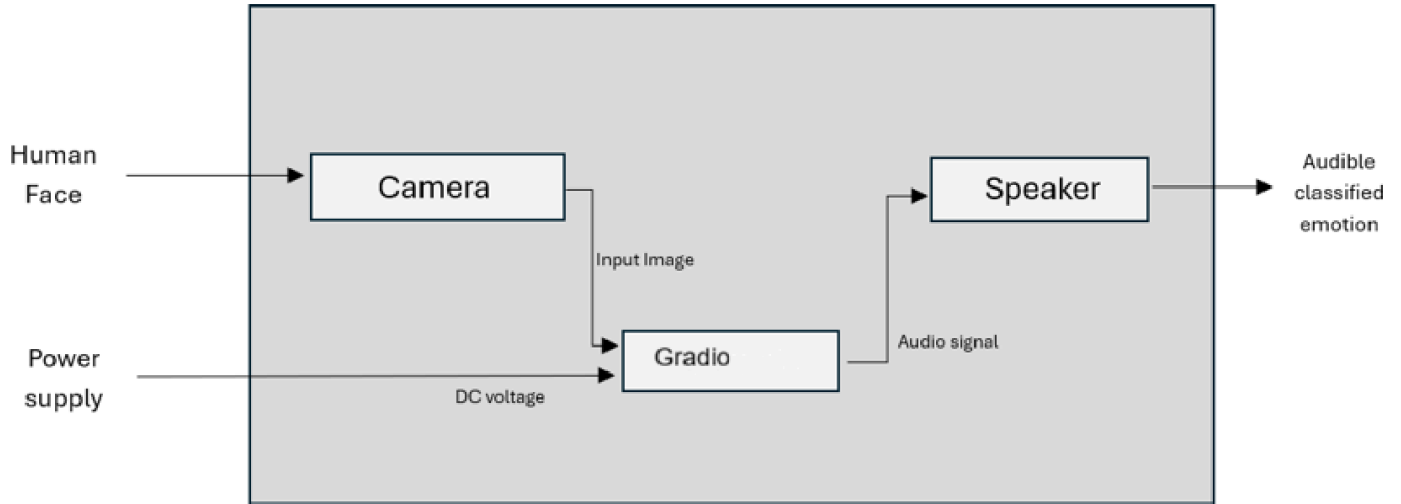


Figure 5 Detailed system design

Table 9 Functionality of Camera

Module	Integrated PC / Laptop Webcam
Input	Live human face video stream or Adding manual photo
Output	Captured Image Frame
Functionality	Captures real-time frames via browser JavaScript media stream and passes them to OpenCV for cropping.

Table 10 Functionality of Software & Computing Platform

Module	Local PC / Google Colab Environment
Input	Cropped Face Image (48 × 48 Grayscale)
Output	Predicted Emotion Class & Generated .mp3 Audio File
Functionality	Runs the saved FER2013_Final.h5 CNN model to calculate probability scores, maps the emotion to a predefined text phrase, and synthesizes the audio via gTTS.

Table 11 Functionality of Audio Output

Module	PC / Laptop Speakers & Gradio Interface
Input	Generated Audio File (emotion_voice.mp3)
Output	Audible Emotional Voice Phrase
Functionality	Plays the audio output directly to the user through the Gradio web UI audio component and system speakers.