Software Requirements Specification (SRS)

Standard: IEEE 830 Format

Project Title: Facial Expression Recognition System

Team Members / Roll Numbers:

- Zainul Abideen (23pwbcs1019)
- Khkula Gul (23pwbcs1029)
- Eisha Shah (23pwbcs1034)
- Umaima Ali (23pwbcs0981)

Instructor: Ma'am Kanwal Aneeq

Department: Computer Science & IT, University of Engineering and Technology Peshawar

Date: October 23, 2025

---

1. Introduction

1.1 Purpose

This SRS document provides complete requirements for the Facial Expression Recognition System. It serves as the primary reference for developers, testers, and stakeholders throughout the project lifecycle.

Intended Audience:

- Development team for implementation guidance
- Testers for validation criteria
- Supervisor for project evaluation
- Future maintainers for system understanding

1.2 Scope

The Facial Expression Recognition System uses deep learning and computer vision to detect and classify human emotions from facial expressions in real-time.

Main Objectives:

- Real-time facial emotion recognition through webcam
- Classify seven basic emotions: happiness, sadness, anger, fear, surprise, disgust, neutral
- Achieve >85% accuracy on standard datasets (FER-2013, CK+)
- Process at minimum 15 FPS for smooth performance
- Provide intuitive web-based interface with emotion visualization

System Boundaries:

- **Included:** Single-face emotion detection, real-time processing, emotion visualization
- **Excluded:** Multi-face detection, mobile app, voice analysis, cloud deployment, identity recognition

Benefits:

- Eliminates subjective emotion interpretation
- Automated, consistent assessment for healthcare and education
- Real-time human-computer interaction based on emotions
- Foundation for affective computing research

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
CNN	Convolutional Neural Network - Deep learning architecture for image processing
FER	Facial Expression Recognition
FPS	Frames Per Second - Video processing speed
ROI	Region of Interest - Detected facial area
FER-2013	Public emotion recognition dataset with 35,887 images
CK+	Extended Cohn-Kanade emotion dataset
OpenCV	Open Source Computer Vision Library
Haar Cascade	Machine learning object detection algorithm
Transfer Learning	Using pre-trained models for new tasks

1.4 References

1. Rahman et al., "Improved facial emotion recognition model," Scientific Reports, 2024
2. Goodfellow et al., "Challenges in representation learning," Neural Information Processing, 2013
3. Li & Deng, "Deep facial expression recognition: A survey," IEEE Trans., 2022
4. IEEE Std 830-1998, "Software Requirements Specifications"
5. TensorFlow Documentation: https://www.tensorflow.org/api_docs
6. OpenCV Documentation: https://docs.opencv.org/
7. Streamlit Documentation: https://docs.streamlit.io/
8. Project Proposal & Feasibility Report (Internal Documents)

1.5 Overview

Section 2 - Overall Description: System context, main functions, user characteristics, constraints, and assumptions.

Section 3 - Specific Requirements: Functional and non-functional requirements, external interfaces.

Section 4 - System Models: Use Case and Data Flow Diagrams.

Section 5 - Other Requirements: Data management, legal compliance, backup considerations.

Section 6 - Appendices: Glossary and technical specifications.

---

2. Overall Description

2.1 Product Perspective

The system is a **standalone desktop application** operating independently without external system integration.

System Context:

- Independent local application with no cloud dependencies
- Uses publicly available datasets and pre-trained models
- Leverages open-source libraries (TensorFlow, OpenCV, Streamlit)
- Designed as educational/research tool

Key Characteristics:

- Self-contained with local processing
- No internet required for operation (only initial setup)
- Privacy-focused with no external data transmission
- Platform-independent (Windows, macOS, Linux)

2.2 Product Functions

1. **Real-Time Video Capture & Display**
    - Capture live webcam stream at 30 FPS
    - Display video feed with face detection overlay
2. **Face Detection & Extraction**
    - Detect faces using OpenCV algorithms
    - Extract facial ROI for emotion analysis
3. **Image Preprocessing**
    - Normalize and resize images (48x48 pixels)
    - Apply noise reduction and enhancement
4. **Emotion Classification**
    - Process images through CNN model
    - Classify into seven emotion categories
    - Generate confidence scores
5. **Real-Time Visualization**
    - Display emotion labels and confidence scores
    - Update predictions in real-time (15+ FPS)
    - Show system status and performance metrics
6. **User Interface Management**
    - Streamlit-based web interface
    - Start/stop controls
    - Clear visual feedback

2.3 User Characteristics

Primary Users:

1. **Healthcare Professionals**
    - Intermediate computer skills
    - Need: Accurate detection, simple operation
2. **Educators/Teachers**
    - Basic to intermediate skills
    - Need: Real-time feedback, minimal complexity
3. **Researchers/Students**
    - Advanced technical skills
    - Need: Access to confidence scores, system flexibility
4. **General Users**
    - Basic computer skills
    - Need: Easy interface, clear feedback

2.4 Constraints

Hardware Constraints:

- Intel Core i5+ processor (or equivalent)
- 8GB RAM minimum
- 50GB storage space

- USB webcam (640x480+ resolution)

Software Constraints:

- Python 3.8 or higher
- TensorFlow 2.x framework
- OpenCV 4.x library
- Streamlit framework
- OS: Windows 10+, macOS 10.14+, or Linux

Technical Constraints:

- Processing limited by CPU (target 15+ FPS)
- Single-face detection only
- Lighting conditions affect accuracy
- Camera positioning impacts reliability

Legal Constraints:

- Open-source license compliance (Apache 2.0, MIT)
- Local processing for data privacy
- Educational/research use only
- Public dataset academic usage restrictions

Development Constraints:

- 14-week timeline
- Zero budget (free tools only)
- Limited to team's current expertise
- Personal computer development

2.5 Assumptions and Dependencies

Assumptions:

- Team computers meet minimum specifications
- USB webcams available for testing
- Stable internet for initial setup
- Public datasets remain accessible
- Team can learn required technologies within timeline
- Adequate lighting during testing
- Users position themselves appropriately

Dependencies:

- TensorFlow/Keras for CNN operations
- OpenCV for video capture and face detection
- Streamlit for web interface
- NumPy for numerical operations
- FER-2013 and CK+ datasets
- USB webcam functionality
- Python package manager (pip)

---

3. Specific Requirements

3.1 Functional Requirements

ID	Requirement Description	Priority
FR1	System shall capture live video from webcam at 30 FPS	High
FR2	System shall detect human faces using OpenCV algorithms	High
FR3	System shall extract facial ROI from detected faces	High
FR4	System shall preprocess images: normalize, resize to 48x48, grayscale conversion	High
FR5	System shall classify facial images into seven emotions using CNN model	High
FR6	System shall generate confidence scores for predictions	High
FR7	System shall display emotion labels and scores in real-time	High
FR8	System shall update predictions at minimum 15 FPS	High
FR9	System shall provide Streamlit web interface accessible via browser	High
FR10	System shall allow start/stop controls for emotion recognition	High
FR11	System shall display live feed with face bounding box and emotion overlay	Medium
FR12	System shall show "No face detected" message when appropriate	Medium
FR13	System shall load pre-trained CNN model on startup	High
FR14	System shall display system status (camera, model, processing state)	Medium
FR15	System shall handle errors gracefully with user notifications	Medium

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

PE1: System shall achieve >85% accuracy on FER-2013 and CK+ test datasets

PE2: System shall process at minimum 15 FPS on standard hardware (i5, 8GB RAM)

PE3: System shall initialize within 10 seconds of startup

PE4: System shall detect faces with maximum 100ms latency per frame

PE5: System shall display predictions within 200ms of frame capture

PE6: System shall maintain stable performance during 30-minute continuous operation

PE7: System shall consume maximum 2GB RAM during operation

3.2.2 Reliability Requirements

RE1: System shall maintain >80% accuracy across varying lighting conditions

RE2: System shall detect faces between 30cm to 100cm distance

RE3: System shall handle minor occlusions (glasses, light facial hair)

RE4: System shall auto-recover from camera disconnection within 5 seconds

RE5: System shall operate reliably for 2-hour continuous sessions

3.2.3 Security and Privacy Requirements

SE1: All processing shall occur locally with no external data transmission

SE2: System shall not store video frames or facial images permanently

SE3: System shall not perform identity recognition - only emotion classification

SE4: System shall not require internet connectivity for operation

SE5: System shall not collect or transmit personally identifiable information

3.2.4 Maintainability Requirements

MA1: Code shall be modular with separate components for video, detection, classification, UI

MA2: System shall use well-documented libraries (TensorFlow, OpenCV, Streamlit)

MA3: Code shall include comprehensive inline comments

MA4: Code shall follow Python PEP 8 style guidelines

MA5: Architecture shall support future emotion category additions

3.2.5 Usability Requirements

US1: Interface shall be learnable within 5 minutes for new users

US2: System shall provide clear visual feedback for face detection and emotion status

US3: Emotion labels shall be readable (minimum 18pt, high contrast)

US4: Interface shall include brief on-screen instructions

US5: Error messages shall be clear and non-technical

US6: Interface shall update smoothly without flickering

3.3 External Interface Requirements

3.3.1 User Interface Requirements

Main Components:

1. **Video Display:**
   - Live webcam feed with face bounding box (green/blue)
   - Emotion label and confidence score overlay (e.g., "Happy 87%")
2. **Control Panel:**
   - Start/Stop button with clear state indication
   - Camera selection (if multiple available)
3. **Status Display:**
   - System status: Ready, Processing, No Face Detected, Error
   - FPS counter showing processing speed

Design Specifications:

- Browser-based Streamlit interface
- Minimum 1024x768 resolution support

- Clean, minimalist design
- High contrast text (dark on light)
- Sans-serif fonts (14pt body, 18pt labels, 24pt headings)

3.3.2 Hardware Interface Requirements

Camera:

- Standard USB webcam or laptop camera
- USB 2.0+ connection
- Minimum 640x480 resolution (recommended 1280x720)
- 30 FPS capture capability
- UVC driver support

Computer:

- Intel Core i5 (5th gen+) or AMD Ryzen 5
- 8GB RAM
- 50GB storage
- 1024x768 display
- Windows 10+, macOS 10.14+, or Linux

3.3.3 Software Interface Requirements

Python Environment:

- Python 3.8, 3.9, or 3.10
- pip package manager

TensorFlow/Keras:

- Version 2.x (2.8 to 2.14)
- Model loading and inference
- NumPy array interface

OpenCV:

- Version 4.x (opencv-python)
- Video capture, face detection, preprocessing
- cv2 module interface

Streamlit:

- Version 1.20+
- Web interface generation
- Local server (port 8501)

NumPy:

- Version 1.21+
- Array operations and data exchange

3.3.4 Communication Interface Requirements

Local Network:

- HTTP protocol for Streamlit interface
- Port 8501 (localhost only)
- No external network access required

Inter-Process:

- In-memory communication between components
- NumPy arrays for data exchange

3.4 System Features

Feature 1: Real-Time Face Detection

Description: Detect and track faces in live video stream

Inputs: Live video stream (640x480 to 1280x720, 30 FPS)

Processing:

- Apply OpenCV face detection (Haar Cascade/DNN)
- Identify facial coordinates (x, y, width, height)
- Extract ROI containing face
- Handle "no face detected" cases

Outputs: Cropped facial image, bounding box coordinates, detection status

---

Feature 2: Image Preprocessing

Description: Prepare facial images for CNN input

Inputs: Raw facial ROI from detection

Processing:

- Convert to grayscale if required
- Resize to 48x48 pixels
- Normalize pixel values [0, 1]
- Apply noise reduction
- Format for model input tensor

Outputs: Preprocessed image array ready for classification

Feature 3: CNN Emotion Classification

Description: Classify emotions using trained CNN model

Inputs: Preprocessed 48x48 facial image tensor

Processing:

- Forward propagation through CNN layers
- Apply activation functions (ReLU, softmax)
- Generate probability scores for seven emotions

- Identify highest probability emotion
- Calculate confidence percentage

Outputs: Emotion label, confidence score (0-100%)

Feature 4: Real-Time Visualization Description:

Display video with emotion results

Inputs: Video frames, bounding boxes, emotion predictions, system status

Processing:

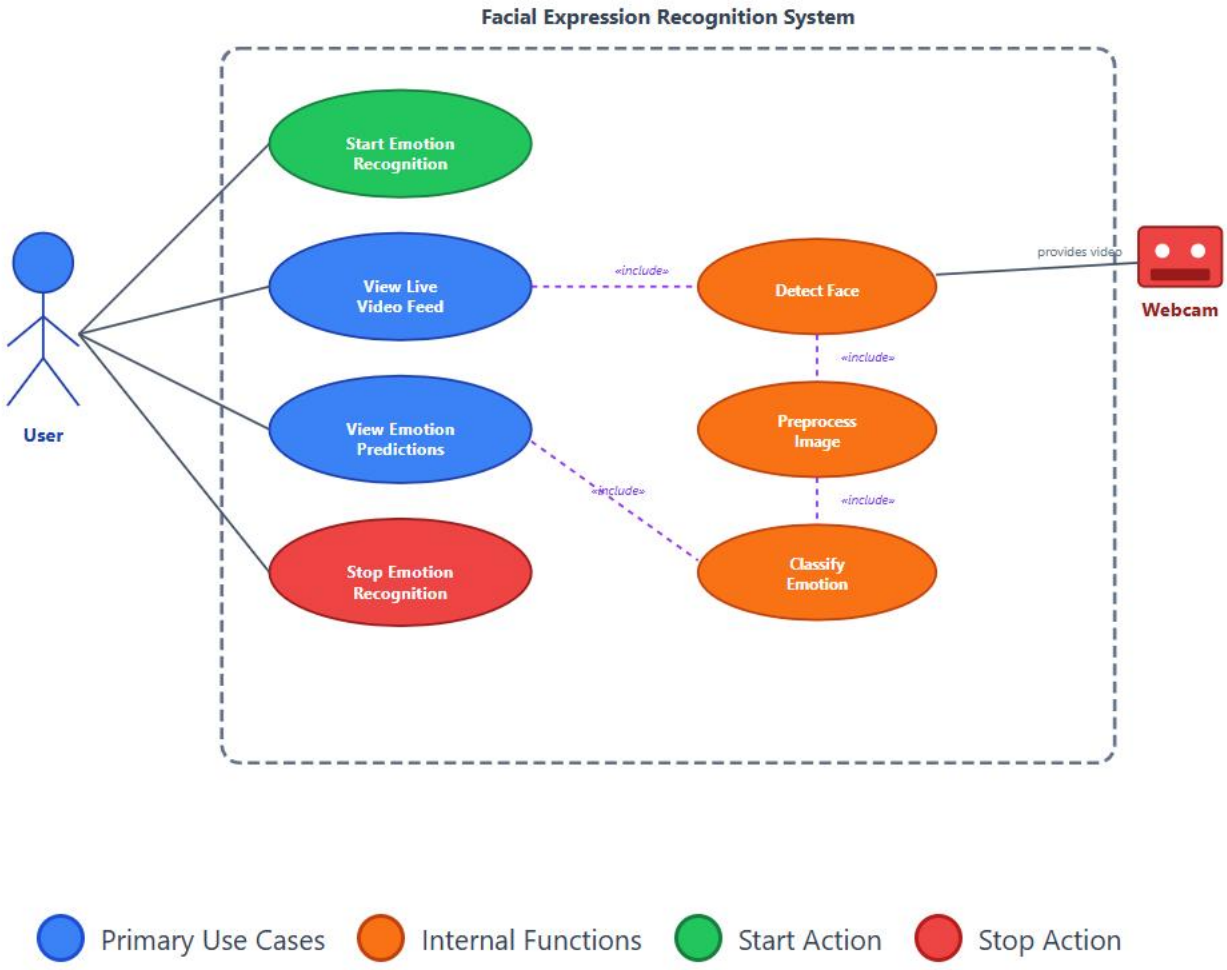
- Draw bounding box on frame
- Overlay emotion label and confidence
- Update display continuously
- Show FPS and status indicators

Outputs: Streamlit interface with annotated live video, emotion labels, status indicators

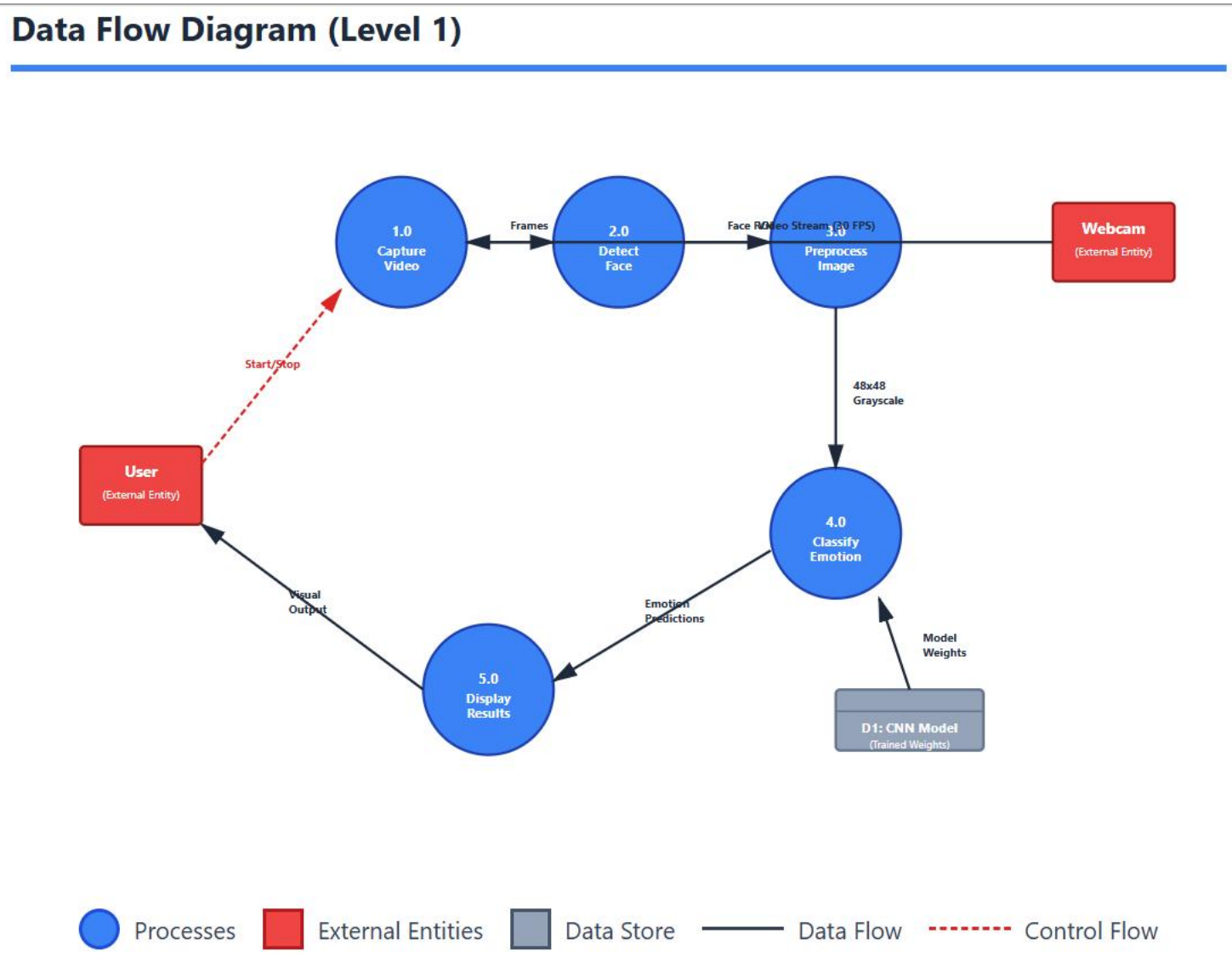
---

4. System Models

4.1 Use Case Diagram



4.2 Data Flow Diagram



5. Other Requirements

5.1 Data Management

Training Data:

- FER-2013 dataset (35,887 grayscale images, 48x48 pixels)
- CK+ dataset for validation
- Data stored locally during development
- No personal data collection

Model Storage:

- Trained CNN model saved as .h5 or .keras file
- Stored locally with application
- Model weights loaded on startup

5.2 Legal and Compliance

Open-Source Licenses:

- TensorFlow: Apache License 2.0
- OpenCV: Apache License 2.0
- Streamlit: Apache License 2.0
- Compliance through proper attribution

Data Privacy:

- GDPR compliant through local processing
- No data collection or transmission
- No personally identifiable information
- Educational/research exemptions apply

Dataset Usage:

- FER-2013: Kaggle competition terms (academic use)
- CK+: Academic research license
- No commercial use of trained models

5.3 Backup and Recovery

Source Code:

- Version control using Git
- Regular commits during development
- Backup on team members' machines

Model Checkpoints:

- Save best model during training
- Keep multiple checkpoint versions
- Store training configuration files

Documentation:

- Regular document backups
- Cloud storage for proposals and reports

---

6. Appendices

6.1 Glossary

Affective Computing: Field studying systems that recognize and respond to human emotions

Bounding Box: Rectangle drawn around detected face in image

Confidence Score: Percentage indicating model's certainty in prediction

Convolutional Layer: CNN layer that extracts features from images **Epoch:**

One complete pass through training dataset

Grayscale: Image with single color channel (black to white)

Hyperparameter: Configuration value set before training (learning rate, batch size)

Inference: Using trained model to make predictions

Normalization: Scaling pixel values to standard range **Overfitting:**

Model memorizes training data, poor generalization **ROI (Region**

of Interest): Extracted facial area from full frame **Softmax:**

Activation function converting scores to probabilities **Transfer**

Learning: Fine-tuning pre-trained model for new task

6.2 Technical Specifications

Model Architecture:

- Input layer: 48x48x1 (grayscale image)
- Multiple convolutional layers with ReLU activation
- Max pooling layers for dimensionality reduction
- Fully connected dense layers
- Output layer: 7 neurons with softmax (one per emotion)

Training Configuration:

- Optimizer: Adam or SGD
- Loss function: Categorical cross-entropy
- Metrics: Accuracy, precision, recall
- Batch size: 32-64
- Epochs: 50-100 with early stopping
- Data augmentation: rotation, zoom, horizontal flip

Face Detection Options:

- Haar Cascade: Fast, less accurate
- DNN (Deep Neural Network): Slower, more accurate
- Default: Haar Cascade for speed, DNN as fallback

6.3 Acronyms Summary

- **API:** Application Programming Interface **CNN:**
- Convolutional Neural Network **CK+:**
- Extended Cohn-Kanade Dataset **DNN:** Deep
- Neural Network
- **FER:** Facial Expression Recognition
- **FPS:** Frames Per Second
- **GPU:** Graphics Processing Unit **HTTP:**
- Hypertext Transfer Protocol **ML:** Machine
- Learning
- **PEP:** Python Enhancement Proposal
- **RGB:** Red Green Blue (color model) **ROI:**
- Region of Interest
- **SRS:** Software Requirements Specification
- **UI:** User Interface
- **USB:** Universal Serial Bus
- **UVC:** USB Video Class

End of Document

This SRS document will be reviewed and updated as project progresses. Version control maintained through Git repository