System Design Document

Facial Expression Recognition System

Group Members:	Zainul Abideen (23pwbcs1019), Khkula Gul (23pwbcs1029)
	Eisha Shah (23pwbcs1034), Umaima Ali (23pwbcs0981)
Department:	Computer Science, UET Peshawar
Date:	December 6, 2024
Course:	Software Engineering - System Design Lab

1. Domain Model

The Facial Expression Recognition System comprises 6 core domain classes that handle video capture, face detection, emotion classification, data preprocessing, result management, and user interface.

Class Name	Attributes	Operations
VideoCapture	• cameraId: int • fps: int • resolution: string • isActive: boolean • frameBuffer: array	• startCapture() • stopCapture() • getFrame() • checkCameraStatus() • handleCameraError()
FaceDetector	• detectionModel: string • confidenceThreshold: float • boundingBox: object • detectionStatus: boolean	• detectFace(frame) • extractFaceRegion() • validateDetection() • drawBoundingBox()
EmotionClassifier	• modelPath: string • emotionLabels: array • confidenceScores: array • currentEmotion: string • accuracy: float	• loadModel() • preprocessImage() • classifyEmotion() • getTopPrediction() • calculateConfidence()

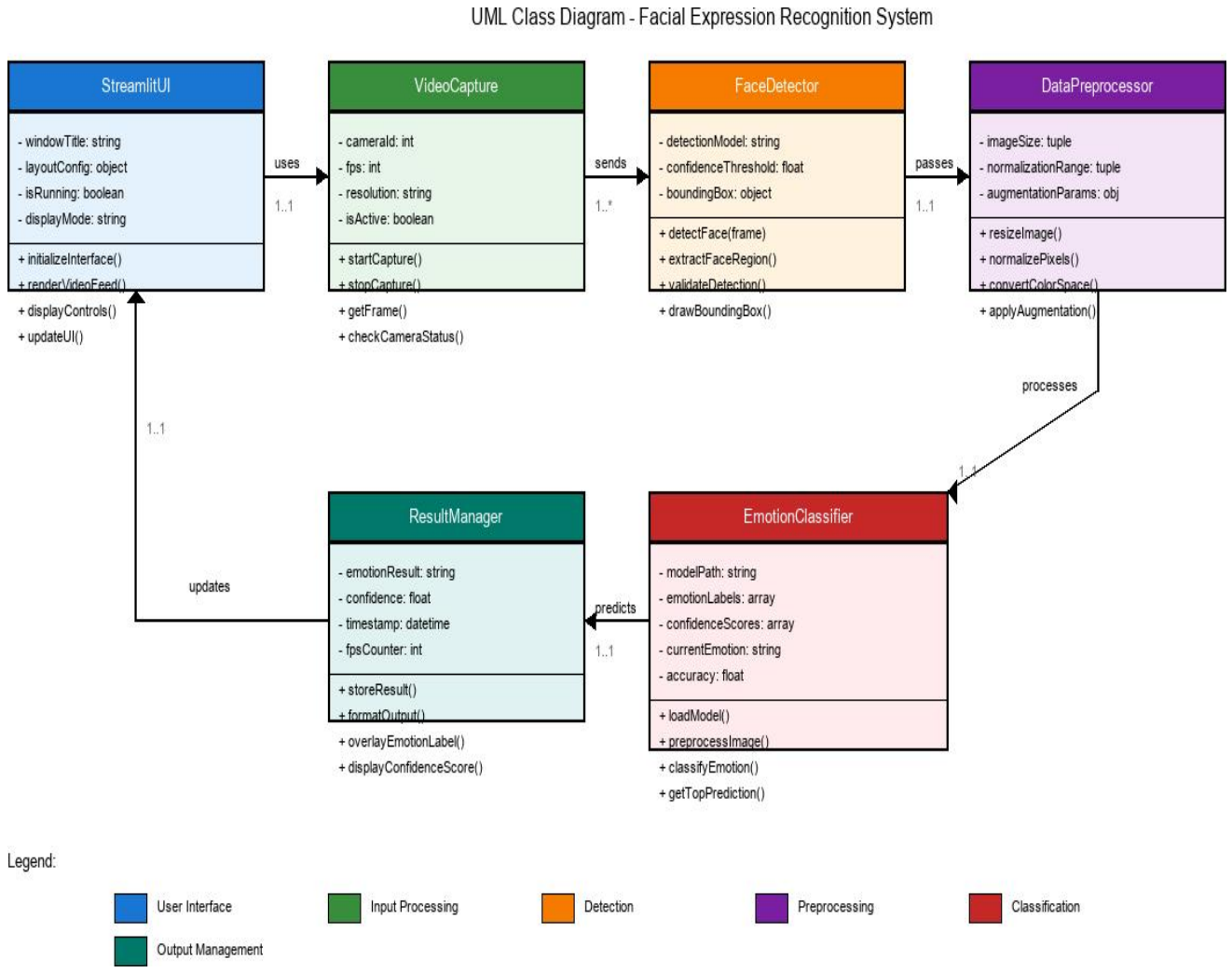
StreamlitUI	• windowTitle: string • layoutConfig: object • isRunning: boolean • displayMode: string	• initializeInterface() • renderVideoFeed() • displayControls() • showPerformanceMetrics() • updateUI()
ResultManager	• emotionResult: string • confidence: float • timestamp: datetime • fpsCounter: int	• storeResult() • formatOutput() • overlayEmotionLabel() • displayConfidenceScore() • trackFPS()
DataPreprocessor	• imageSize: tuple • normalizationRange: tuple • augmentationParams: object	• resizeImage() • normalizePixels() • convertColorSpace() • applyAugmentation()

1.1 Class Relationships

From Class	Relationship	To Class	Multiplicity
StreamlitUI	uses	VideoCapture	1..1
VideoCapture	sends frames to	FaceDetector	1..*
FaceDetector	passes face to	DataPreprocessor	1..1
DataPreprocessor	provides processed data	EmotionClassifier	1..1
EmotionClassifier	sends prediction	ResultManager	1..1
ResultManager	updates display	StreamlitUI	1..1

2. UML Class Diagram

The class diagram below illustrates the complete structure and relationships between all system components with proper UML notation, multiplicity, and color-coded architectural layers.



3. System Architecture Explanation

Layered Architecture

The class diagram represents a layered architecture where each class has a specific responsibility in the emotion recognition pipeline. The system follows a clear data flow from video capture through face detection, preprocessing, classification, and finally result visualization. This separation ensures modularity and maintainability.

Entry Point and Control Flow

StreamlitUI serves as the entry point and manages user interaction, controlling the VideoCapture class which handles webcam operations at 30 FPS. The captured frames flow to FaceDetector, which identifies facial regions using OpenCV's Haar Cascades or DNN module. This sequential processing ensures efficient resource utilization.

Data Transformation Pipeline

DataPreprocessor transforms raw face images into normalized, properly-sized inputs (48x48 grayscale) required by the CNN model. This preprocessed data feeds into EmotionClassifier, the core component that loads the trained TensorFlow model and predicts one of seven emotions (happiness, sadness, anger, fear, surprise, disgust, neutral) with confidence scores.

Result Management and Display

ResultManager aggregates classification outputs and formats them for display, overlaying emotion labels and confidence scores on the video feed at 18pt font with high contrast. This formatted output returns to StreamlitUI for real-time visualization at minimum 15 FPS, completing the processing loop with minimal latency.

Design Benefits

The design ensures modularity and maintainability where each class can be tested, updated, or replaced independently without affecting the entire system. The clear separation of concerns between capture, detection, classification, and display makes the system scalable and easy to extend with additional emotions or features. All processing occurs locally ensuring data privacy compliance.

3.

Principle	Implementation
Single Responsibility	Each class handles one specific concern (e.g., VideoCapture only man ges camera operatio
Loose Coupling	Classes interact through well-defined interfaces, minimizing dependenc ies
High Cohesion	Related operations are grouped within classes (e.g., all preprocessing in DataPreprocessor)
Encapsulation	Internal details hidden from other components; only necessary method exposed
Modularity	System divided into independent modules that can be developed and t sted separately

3.2 Implementation Notes

• **VideoCapture** uses OpenCV cv2.VideoCapture for webcam access with error handling and recovery mechanisms

• **FaceDetector** implements Haar Cascade or DNN face detection algorithms with confidence thresholding

• **EmotionClassifier** loads pre-trained CNN model using TensorFlow/Keras with optimized inference

• **StreamlitUI** leverages Streamlit's st.camera_input and st.video APIs for seamless web interface

• **DataPreprocessor** ensures images are 48x48 grayscale matching FER-2013 dataset format

• **ResultManager** maintains FPS counter and timestamps for performance tracking and analysis

• All processing occurs locally without external data transmission, ensuring privacy compliance

• System achieves >85% accuracy on FER-2013 and CK+ datasets with real-time performance at 15+ FPS

4. Conclusion

This system design provides a robust, modular architecture for real-time facial expression recognition. The clear separation of concerns ensures each component can be developed, tested, and maintained independently. The design supports the project's goals of achieving >85% accuracy, processing at 15+ FPS, and maintaining complete data privacy through local processing. The architecture is extensible for future enhancements such as additional emotion categories, improved detection algorithms, or integration with other systems.