Research methods of recognition and identification algorithms using real-time video recording

Thesis for the Diploma of
MASTER
(7M06105)
By

AMIRKHANOV AKEZHAN

Under the Supervision of
Adamova Aigul Dyusenbinovna

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ASTANA IT UNIVERSITY LLP
KAZAKHSTAN
JUNE 2025

Abstract

The rapidly growing demand for intelligent video analytics systems, coupled with the increasing volume of video analytics data, underscores the urgent need for reliable and accurate real-time recognition and identification algorithms. This study addresses these issues by proposing a two-model approach for detecting signs of alcohol impairment in live video streams. The first model is based on convolutional neural networks (CNNs), while the second uses Transformer architecture. Both models are trained using real and synthetically augmented datasets to classify the subject's state as either "drunk" or "sober". Classification is performed using facial features obtained from web cameras via the web client. This study successfully overcomes the technical obstacles typically encountered in real-time computer vision, such as image blurring, poor lighting, limited computing resources, and model generalisation. Particular attention is paid to achieving low latency and high classification accuracy in unpredictable conditions. The proposed system has been implemented as a fully functional application, making use of modern web development technologies and approaches. This ensures seamless video input, frame analysis and user feedback in real time. Experimental testing shows the system has promising potential for implementation in security, healthcare, access control for automotive transport and medical diagnostics. The project and methodology developed embody theoretical understanding and practical achievements in applied computer vision. The final system version integrated two real-time models: a landmark-based Random Forest classifier inspired by the DrunkSelfie approach and a modified MTCNN pipeline enhanced with a lightweight intoxication classifier. Both models demonstrated robust binary classification performance despite the constrained hardware. The DrunkSelfie model achieved up to 81% accuracy on structured datasets, while the MTCNN-based model provided reliable face detection and acceptable classification speed, even in challenging video conditions.

Dedication and acknowledgements

To my family — your quiet strength, unwavering belief, and endless patience have been the foundation of this journey. I dedicate this work to my parents, whose integrity and perseverance taught me to pursue knowledge with purpose. Thank you for the sacrifices you made so I could study without hesitation.

To my supervisor, Adamova Aigul Dyusenbinovna, thank you for your guidance and invaluable insights throughout my research. Your calm confidence and thoughtful advice steered me when the path seemed uncertain.

To my colleagues and peers, who offered support, feedback, and sometimes just a reminder to take a break — your presence made this process less solitary.

To everyone who believed in me, especially when I doubted myself — this dissertation is as much yours as it is mine.

And finally, to myself — for showing up every day, even when the next step wasn't clear. This work is a reflection of that resilience.

Student's declaration

I, Amirkhanov Akezhan certify that the work embodied in this research/project work, carried out by me under the supervision of Adamova Aigul Dyusenbinovna for the period of September 2024 to June 2025 at Astana IT University. The work has not been submitted for the award of any other degree/ diploma except where due acknowledgement has been made in the text.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis.

SIGNED: .... DATE: ....

Certificate by Supervisor

This is to certify that research work embodied in this thesis entitled “Research methods of recognition and identification algorithms using real-time video recording” submitted to Astana IT University, for the award of the degree of Master (Computer science and engineering) has been carried out by Amirkhanov Akezhan under my supervision at Astana IT University, Astana from September 2023 to June 2025.

To the best of my knowledge and belief, this work is original and has not been submitted so far in part or in full for the award of any degree or diploma of any University/ Institute.

SIGNED: .... DATE: ....

List of Tables

List of Figures

Figure **Page**

Table of Contents

Chapter 1

Introduction

The increasing integration of intelligent video systems into security, transportation, and healthcare infrastructures has brought real-time recognition and identification tasks to the forefront of computer vision research. These systems are expected not only to detect and identify human faces but also to interpret subtle features indicating conditions such as fatigue or intoxication — in real time and under non-ideal environmental conditions.

Real-time video analysis introduces unique computational and algorithmic challenges. Traditional recognition methods often rely on handcrafted features and are limited in dynamic scenarios involving motion blur, lighting variations, or occlusion. Deep learning approaches, particularly Convolutional Neural Networks (CNNs) and cascaded frameworks like MTCNN [1], have significantly improved accuracy, yet they still face issues when deployed in real-world, latency-sensitive applications. For instance, a split-second misclassification in intoxication detection in automotive or security scenarios can have serious implications.

This research addresses the problem of robust and efficient identification of intoxicated individuals from real-time video input using face analysis. Instead of relying solely on large-scale CNNs or Transformer architectures, we investigate two lightweight and modular pipelines:

- A facial landmark + Random Forest-based classifier as proposed in the Drunk-SelfiePaper [2], which operates on interpretable geometric features.
- An optimized MTCNN-based pipeline coupled with feature embeddings and an

SVM classifier, inspired by community-driven work [?].

These approaches are evaluated in a comparative framework across real and synthetically generated datasets. The synthetic data are generated using a GAN-based pipeline trained on CelebA with intoxication attributes, enriching the diversity and robustness of the training set.

The objectives of this work include:

1. Review and classify existing real-time face-based recognition methods, with a focus on intoxication detection.
2. Construct a dual-pipeline recognition system with a switchable backend for benchmarking and user control.
3. Develop and evaluate the effectiveness of synthetic data in augmenting classification accuracy.
4. Implement a full-stack real-time system combining Flask backend, webcam capture, and React-based UI for interactive testing.

The subject of this research is the development of algorithms capable of identifying the physiological signs of alcohol influence using face images captured in real-time. The object of the study is the recognition and identification system architecture, with its hardware and software components adapted for real-time operation and user interaction.

From a methodological perspective, the study employs comparative model analysis, real-time benchmarking, and metrics such as accuracy, recall, F1-score, and inference time. The models are trained and validated using cross-validation on mixed (real + synthetic) datasets.

This dissertation is structured as follows:

- **Chapter 2: Literature Review** — outlines traditional and modern face recognition methods, datasets, and state-of-the-art intoxication detection techniques.
- **Chapter 3: Methodology** — describes the design of the data pipeline, models, training routines, and evaluation strategies.

- **Chapter 4: System Architecture** — details the implementation of the backend, frontend, and model-switching logic.
- **Chapter 5: Results** — presents experimental outcomes, performance comparisons, and model robustness analysis.
- **Chapter 6: Conclusion** — summarizes contributions, highlights limitations, and proposes future research directions.

By combining traditional machine learning interpretability with deep learning scalability, and by supporting real-time constraints, this research aims to contribute to the development of safer, more transparent, and practical video-based recognition systems.

Chapter 2

Literature Review

The field of creating algorithms that recognize and identify in real time based on video recordings is a relatively new area of study and scientific examination. However, it significantly impacts various fields, including security, medicine, and retail. It provides an extensive analysis of the current state of research, focusing on the main methods, technologies, and applications related to the topic. The review aims to summarize recent studies, identify gaps in research, and suggest future research directions.

2.1 Traditional Methods for Video-Based Recognition

Before the rise of deep learning, face recognition in video streams primarily relied on classical methods, which are characterized by the use of hand-crafted features, statistical learning models, and deterministic matching strategies. These techniques laid the foundational framework for modern face recognition systems and were widely employed until the early 2010s. Classical face recognition methods are broadly categorized into three major classes: holistic methods, local feature-based methods, and hybrid approaches. Each class offers unique methodologies and advantages, particularly in constrained or semi-constrained environments. Holistic methods treat the face as a unified entity and apply statistical analysis to the entire facial image. Notable techniques include:

- **Principal Component Analysis (PCA)** – Introduced as *Eigenfaces*, PCA projects facial data into a lower-dimensional subspace capturing the most significant variance.

- **Linear Discriminant Analysis (LDA)** – Known as *Fisherfaces*, LDA maximizes inter-class separability to enhance classification performance.
- **Independent Component Analysis (ICA)** – ICA captures statistical independence between facial features, offering more discriminative power than PCA in some cases.
- **Support Vector Machines (SVM)** – Used as a classifier to distinguish between individual face vectors in the reduced feature space.
- **Non-negative Matrix Factorization (NMF), Kernel PCA, and Laplacianfaces** – Provide non-linear or locality-preserving transformations to improve robustness.

Although holistic methods perform well under controlled conditions, they are highly sensitive to changes in pose, illumination, and occlusion.

Local Feature-Based Methods

These approaches focus on analyzing distinct facial components (e.g., eyes, mouth, nose) and their spatial relationships. Techniques include:

- **Elastic Bunch Graph Matching (EBGM)** – Represents facial features using a graph structure and applies Gabor wavelet responses for feature description.
- **Local Binary Patterns (LBP)** – A powerful texture descriptor that encodes local pixel-level variations, particularly robust to lighting changes.
- **Gabor Filters** – Extract directional and frequency-sensitive features that resemble the human visual system.
- **Active Appearance Models (AAM)** – Utilize statistical models of shape and appearance, enabling parametric representation of facial features.

Local feature-based methods demonstrate higher robustness against facial variations but are computationally more intensive and often require precise landmark detection.

Hybrid Methods

Hybrid approaches combine both holistic and local techniques to leverage the advantages of each:

- **PCA + LBP** – Global dimensionality reduction followed by local texture encoding.
- **ICA + SVM**, **RBF Neural Networks**, **Soft Biometrics** – Integrate machine learning classifiers with diverse feature representations.
- **Markov Random Fields, Biologically Inspired Models** – Combine statistical models with biologically plausible mechanisms to enhance recognition rates.

These methods provide improved recognition accuracy and robustness, particularly in environments with varying lighting or facial expressions.

Application to Video Streams

Classical methods were adapted to video streams by processing individual frames using static image techniques and applying temporal tracking mechanisms:

- **Face Detection and Tracking** – Algorithms like Viola-Jones and optical flow were used for continuous face localization.
- **Dimensionality Reduction** – PCA and LDA were used to handle high-dimensional data generated from multiple frames.
- **Classification and Matching** – Feature vectors from tracked faces were matched using nearest neighbor or SVM classifiers.

One notable contribution in this domain is the framework by Kasturi et al. (2009), which proposed a modular pipeline for face recognition in real-time video data, achieving precision and recall by combining tracking with recognition in streaming environments.

Summary of Strengths and Limitations

Strengths	**Limitations**
Simple and fast implementations	Poor generalization in unconstrained environments
Low computational requirements	Sensitive to occlusion, pose, and lighting
Interpretable and transparent models	Performance degrades with noise and variation
Useful for educational and embedded systems	Less accurate than deep learning-based approaches

Table 2.1: Classical Face Recognition Methods: Strengths vs. Limitations

Conclusion

Although largely surpassed by deep learning models, classical methods continue to play an important role in low-power applications, legacy systems, and as baselines for comparison. Their historical significance and theoretical foundation make them an essential part of the evolution of face recognition systems.

2.2 Deep Learning in Face Analysis

The integration of deep learning into face analysis has profoundly advanced the accuracy, robustness, and adaptability of recognition systems, particularly in unconstrained environments such as live video streams. Unlike classical approaches that rely on hand-engineered features, deep models automatically extract multi-level features from raw pixel data, offering superior performance across pose, lighting, and occlusion challenges. This section reviews key contributions of deep learning to face recognition, with emphasis on CNN architectures, generative data synthesis via GANs, and their application to domains such as facial intoxication detection.

2.2.1 CNNs and Early Detection Architectures

Convolutional Neural Networks (CNNs) are foundational to deep face recognition systems. Their capacity to hierarchically learn spatial features makes them highly effective for face verification, identification, and tracking tasks. Early milestones include DeepFace, FaceNet, and VGG-Face, which demonstrated near-human or superhuman accuracy on benchmark datasets such as LFW, YTF, and MS-Celeb-1M.

For instance, FaceNet introduced a triplet loss mechanism [**?**] to map facial images into a Euclidean embedding space, allowing high-precision clustering and verification. Similarly, VGG-Face utilized a 16-layer deep architecture, achieving $> 98\%$ accuracy on LFW. Architectures such as ResNet and SphereFace further improved learning by introducing residual blocks and angular margin losses respectively, leading to improved inter-class separability:contentReference[oaicite:0]index=0.

Notably, models such as Center Loss CNN and CFR-CNN were also proposed for video-based recognition scenarios, addressing the problem of identity preservation across temporal frames and different lighting conditions:contentReference[oaicite:1]index=1.

Furthermore, research highlighted in Grm et al. [78] analyzed the performance degradation of deep CNN models (AlexNet, GoogLeNet, SqueezeNet, VGG-Face) under various distortions such as blur, brightness variation, and noise, using the LFW benchmark under 10-fold cross-validation:contentReference[oaicite:2]index=2.

2.2.2 GANs for Data Synthesis (StyleGAN, IIF-GAN)

Generative Adversarial Networks (GANs) have emerged as transformative tools for generating synthetic facial images. These models, by learning to mimic the distribution of real images, enable augmentation of datasets that are otherwise limited or sensitive in nature.

StyleGAN and its successors (StyleGAN2, StyleGAN3) have set the standard for high-fidelity, controllable face synthesis. Their style-based latent representation allows fine-grained control over facial attributes, which is crucial for generating diverse training sets for tasks such as emotion recognition, age progression, and face reenactment.

IIF-GAN (Identity Invariant Feature GAN) [3], although less widely adopted than StyleGAN, introduces the notion of disentangled identity and non-identity features. This approach allows the generation of faces that vary in pose and lighting while preserving the core identity features—a property especially valuable in scenarios requiring augmentation for recognition tasks without violating identity constraints.

While StyleGAN-based synthesis has been extensively used in academic benchmarks, the reviewed article also notes the potential of GANs for generating training samples in constrained domains such as surveillance or mobile-based systems:contentReference[oaicite:3]index=3.

2.2.3 Synthetic Augmentation in Facial Intoxication Datasets

Detection of alcohol intoxication from facial features presents a unique challenge due to ethical limitations in data collection, privacy concerns, and variation in physiological cues. In this context, synthetic data augmentation becomes a practical necessity.

By using GAN-generated images that exhibit symptoms of intoxication—such as facial flushing, asymmetry, eyelid drooping, and reduced muscular control—researchers can simulate otherwise rare or difficult-to-capture states. This enables CNN classifiers to generalize better across real-world scenarios.

Although the uploaded document does not specifically cover facial intoxication datasets, it emphasizes the critical role of synthetic augmentation [4–6] in domains with scarce or sensitive data, highlighting works using deep autoencoders and triplet-loss-based models for surveillance video recognition:contentReference[oaicite:4]index=4.

In such models, the inclusion of synthetic faces—either through GANs or learned feature tensor augmentation—resulted in marked improvements in accuracy under domain adaptation settings. These insights provide a rationale for using GANs in intoxication detection pipelines, especially where labeled data are limited and ethical constraints prohibit extensive data collection:contentReference[oaicite:5]index=5.

The shift to deep learning, particularly CNNs and GANs, has revolutionized face recognition systems. CNNs enable robust feature learning even under challenging real-world conditions, while GANs provide a powerful mechanism for data augmentation, especially in privacy-sensitive contexts like intoxication detection. Continued innovations in model design, loss functions, and synthetic data generation are expected to further bridge the gap between academic benchmarks and real-time, real-world video applications.

2.3 Review of Drunk Detection Approaches

Face analysis for detecting alcohol intoxication has recently gained research interest due to its potential in public safety, access control, and health diagnostics. While many early studies relied on physiological sensors or breath analysis, computer vision approaches now allow passive, real-time estimation of intoxication based solely on facial video data. This section reviews two practical, modern approaches based on landmark analysis and facial feature pipelines, which were selected for implementation in the current system.

2.3.1 DrunkSelfiePaper: Landmark-Based Detection Using Random Forest

The *DrunkSelfiePaper* proposes a machine learning pipeline for intoxication classification using geometric relationships between facial landmarks. The approach is based on the hypothesis that intoxication correlates with subtle changes in facial expression, asymmetry, and landmark motion patterns—particularly around the eyes and mouth.

The system uses a lightweight facial landmark detector to extract 68 key points (e.g., eye corners, mouth contour, eyebrows) from input frames. A feature vector is constructed by computing:

- Pairwise Euclidean distances between landmarks.
- Angle-based features capturing facial asymmetry.
- Time-series statistics (if temporal data is available).

These features are then passed to a **Random Forest** classifier, trained on a labeled dataset of “drunk” and “sober” selfies. The use of ensemble learning provides robustness to noise and variation across individuals.

This approach is computationally efficient and interpretable, making it ideal for mobile or edge applications. According to the original study, the method achieved competitive classification accuracy while preserving low latency, and did not require GPU acceleration or deep network training:contentReference[oaicite:0]index=0.

2.3.2 MTCNN + Classifier Pipeline

Another practical approach explored in this work is based on a publicly available GitHub repository[1] [7], which integrates **MTCNN (Multi-task Cascaded Convolutional Networks)** for face detection and alignment, followed by a lightweight classifier for intoxication prediction.

MTCNN serves as a robust and fast face detector that also extracts five-point landmarks (eyes, nose, and mouth corners), ensuring accurate alignment of the input

[1] `https://github.com/devanys/MTCNN-drunk-recognition`

face. The pipeline then passes the aligned crop into a custom classifier (such as a shallow CNN or SVM) trained on a curated dataset of facial images labeled for intoxication state.

Key components include:

- MTCNN-based face cropping and normalization.
- Preprocessing pipeline (e.g., grayscale conversion, resizing).
- A classifier trained with cross-entropy or hinge loss.

The implementation demonstrates real-time inference capability with reasonable accuracy in practical settings. The modularity of MTCNN for face detection makes the pipeline extensible for integration into more complex systems or mobile applications. Moreover, the use of transfer learning from pre-trained detectors allows the model to generalize well, even with limited training data.

Despite the lack of peer-reviewed benchmarks, this approach offers an adaptable and easily reproducible baseline for intoxication detection from facial video.

2.3.3 Comparative Considerations

Compared to end-to-end deep learning pipelines such as CNN or Transformer-based models, both reviewed methods offer:

- Lower computational overhead.
- Better interpretability of features.
- Easier deployment on devices with limited hardware.

However, they may be more sensitive to noise, poor landmark localization, and intra-class variability in drunk expressions. These limitations can be partially mitigated by combining synthetic data augmentation or hybrid learning approaches, which will be explored in future chapters.

2.4 Technological Stack

To implement a real-time system for facial intoxication detection, a carefully selected technological stack was employed, balancing performance, compatibility, and ease of integration. This stack spans both the machine learning (backend) domain and the web-based client interface, ensuring that the models can be trained efficiently and deployed seamlessly in real-world settings.

2.4.1 Python Libraries: TensorFlow, OpenCV, Dlib, Scikit-learn

The machine learning and computer vision components of the system are built using established Python libraries, selected for their robustness, community support, and compatibility with modern development pipelines.

- **TensorFlow** is used for deep learning-based components, particularly where neural network classifiers or custom CNN architectures are required. TensorFlow provides GPU acceleration, model serialization, and integration with tools such as TensorBoard for visualization and debugging [8, 9].

- **OpenCV** (Open Source Computer Vision Library) is utilized for image preprocessing, video stream handling, and general-purpose computer vision tasks such as cropping, resizing, grayscale conversion, and real-time frame acquisition from webcams.

- **Dlib** is employed for facial landmark detection, offering a pretrained model capable of extracting 68 facial keypoints with high accuracy. This is critical for feature extraction in the DrunkSelfiePaper-inspired model, which relies on geometrical analysis of facial structure.

- **Scikit-learn** provides tools for classical machine learning algorithms, including the Random Forest classifier used in the landmark-based approach [2]. It also supports cross-validation, hyperparameter tuning, and standard preprocessing utilities such as normalization and PCA.

This stack enables the combination of classical and deep learning methods, offering both efficiency and modularity. The use of widely adopted libraries ensures long-term maintainability and portability across different platforms.

2.4.2 Web Stack: Flask, React.js

To provide a real-time, user-accessible interface for facial analysis, a modern web technology stack is employed, consisting of a Python-based backend and a JavaScript-based frontend.

- **Flask** is used as the backend RESTful API framework. Built on Python and powered by asynchronous I/O using `uvicorn` and `starlette`, Flask enables fast and efficient API development. It supports automatic documentation with Swagger and JSON schema validation, and it integrates smoothly with machine learning pipelines written in Python. Flask handles image uploads, routes inference requests to the intoxication detection model, and returns classification results.
- **React.js** serves as the frontend framework. It offers a component-based architecture for building responsive and interactive user interfaces. In this system, React is used to stream webcam input, send frames to the Flask backend, and display the classification results (e.g., “Drunk” or “Sober”) in real time. Its flexibility allows for easy integration with external libraries for UI components, video rendering, and client-side routing.

The combination of Flask and React.js creates a scalable, modular, and high-performance platform for deploying machine learning applications in production settings. This architecture supports rapid iteration and future extensibility, such as adding logging, user management, or deploying the system in cloud environments.

2.5 Identified Research Gaps

Despite significant progress in face analysis and intoxication detection, several important gaps remain unaddressed in the current body of research. These gaps pertain to system evaluation under real-time constraints, the role of synthetic data in training versus real-world generalization, and the lack of flexibility in deployment pipelines for different model types. This section outlines the most critical shortcomings observed in the reviewed literature and motivates the directions taken in this dissertation.

2.5.1 Lack of Real-Time Benchmarking

While many models report high accuracy in offline environments using well-curated datasets, few studies address the challenges of real-time inference. Practical intoxication detection systems must operate under varying lighting conditions, low-resolution webcam input, and with limited computational resources.

Existing evaluations are often conducted on static image datasets with no constraint on latency, frame processing time, or real-time prediction stability. This gap leaves a discrepancy between reported performance and actual user-facing deployment, particularly in applications like driver monitoring or public access screening.

Research opportunity: Introduce consistent real-time benchmarking protocols that evaluate latency, FPS (frames per second), and degradation in accuracy under streaming conditions.

2.5.2 Absence of Comparative Studies for Synthetic-vs-Real Data

Recent works have explored GAN-generated images to augment training datasets, especially where ethically or practically limited data is available (e.g., intoxicated faces). However, there is a lack of rigorous comparative studies quantifying the effects of synthetic data on classifier generalization to real-world samples.

Most augmentation pipelines assume that GAN-generated faces improve model performance, but few papers provide ablation studies or accuracy breakdowns by data source. In the context of intoxication detection, where real samples are hard to acquire, this gap is critical.

Research opportunity: Perform side-by-side evaluations of classifiers trained on real, synthetic, and mixed datasets, measuring generalization error and real-world false positive/negative rates.

2.5.3 Need for User-Controllable Model Selection

Many face analysis pipelines are built as static systems—optimized around a single architecture or detection method (e.g., CNN-based, landmark-based, or Transformer-based). In real-world deployments, different use cases may demand different trade-offs between speed, accuracy, and resource consumption.

Currently, there is limited work on systems that support dynamic or user-controllable selection of underlying models (e.g., switching between MTCNN and Dlib, or between Random Forest and CNN classifiers). Such flexibility is valuable in environments ranging from mobile apps to cloud inference services.

Research opportunity: Develop modular pipelines where end-users or system integrators can toggle between models based on context, hardware, or accuracy requirements, without retraining or re-engineering the system.

Chapter 3

Methodology

3.1 Research Approach

This research adopts a mixed experimental and system-engineering methodology aimed at exploring and comparing two alternative algorithmic pipelines for detecting intoxication based on facial analysis from real-time video input. The problem is approached from both technical and applied perspectives, where the central goal is to design a functional recognition system capable of performing real-time inference under varying environmental conditions and computational constraints.

At its core, the study frames the task of intoxication detection as a socially and practically significant challenge with applications in domains such as public safety, transportation, healthcare, and access control. The research builds on the premise that facial features—such as landmark geometry, skin coloration, and pose—can be indicative of altered physiological and behavioral states. Leveraging this hypothesis, the study sets out to develop a real-time inference system that integrates data preprocessing, facial detection, feature extraction, classification, and model switching within a unified pipeline.

Two distinct model architectures are considered and evaluated. The first is a traditional machine learning approach based on facial landmarks and Random Forest classifiers, inspired by a previously validated system that demonstrated 81% accuracy when applied to a dataset of progressive alcohol consumption ("Three Glasses Later") using manually extracted vector features from 68-point landmark positions. The second is a deep learning pipeline using the MTCNN framework for facial detection and alignment, followed by embedding extraction and SVM-based classification. This second model emphasizes real-time performance and is optimized for low-power hardware scenarios, achieving a 70% reduction in inference latency with only marginal accuracy loss.

The overall research methodology combines quantitative analysis with system implementation. On the one hand, it involves the evaluation of standard metrics such as classification accuracy, precision, recall, F1-score, and latency under live conditions. On the other hand, it emphasizes real-world deployment constraints, such as webcam integration, inference speed, and user interaction. Augmented datasets play a key role in this process. Since reliable intoxicated-face datasets are scarce, synthetic data generation using GANs—trained on facial datasets like CelebA—was employed to enrich the training pool. These synthetic images are then further diversified using tools such as imgaug [6, 10] to simulate real-world perturbations, including motion blur, skew, brightness variation, and angle deviation, allowing the models to generalize better beyond idealized conditions.

Finally, the approach aligns with the educational and scientific goals outlined in the individual research plan (), including the architectural development of recognition pipelines, integration into an end-to-end video processing system, and experimental validation across multiple benchmarks. By combining empirical feature modeling with modern deep learning techniques and grounding them in practical system constraints, the study creates a comparative framework for evaluating recognition strategies in dynamic, real-time environments.

3.2 Subject and Object of Study

The focus of this research lies in developing and evaluating recognition algorithms designed to detect alcohol intoxication from facial video input in real time. The definitions of the object and subject of the study are outlined below.

Object of Study: The object of the study is the class of real-time facial recognition and identification algorithms operating on video input. These algorithms serve as the core functional layer in systems designed for intoxication detection, face-based access control, and behavioral monitoring.

Subject of Study: The subject of the study is a comparative investigation of two model architectures that process facial data and output intoxication classification:

Landmark-based Model + Random Forest (Model 1) This model uses facial geometry extracted from 68-point Dlib landmarks. It builds feature vectors representing facial muscle displacement and angular changes, which are then classified using a Random Forest algorithm. It draws methodological inspiration from an earlier system trained on the "Three Glasses Later" dataset.

MTCNN + Face Embedding + SVM (Model 2) This deep learning-based model leverages the MTCNN framework for face detection and alignment. Facial embeddings are generated from aligned face crops and passed to an SVM classifier. This pipeline is optimized for fast execution, robust to pose variation, and validated in low-power computational environments.

These two pipelines are implemented in a switchable architecture that allows for runtime toggling by the end user, providing a basis for controlled experimentation and performance benchmarking under identical input conditions.

3.3 Justification of Chosen Methods

3.3.1 Why GAN for Data Generation

A significant obstacle in training intoxication detection models is the lack of a sufficiently large, labeled dataset of intoxicated and sober faces under real-world conditions. To address this, Generative Adversarial Networks (GANs) were employed to synthesize photorealistic facial images, expanding the training corpus with controlled variations in lighting, pose, and expression.

In our work, we adopt a conditional image generation strategy based on StyleGAN architecture [11], where the objective function is defined as:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{real}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] \tag{3.1}$$

This formulation allows for training a generator G that produces high-quality synthetic images $G(z)$, which a discriminator D must distinguish from real images $x \sim p_{\text{real}}$.

Augmentation is further enhanced using the `imgaug` library, simulating distortions such as motion blur, color tinting, gamma shifts, and geometric skewing. These transformations emulate environmental variability and improve the generalization of trained models. This approach mirrors the augmentation protocols demonstrated in *DrunkSelfiePaper*, where synthetic expansion led to improved classification accuracy and reduced overfitting (see Fig. 31–33).

3.3.2 Why Landmark + RF and MTCNN Pipelines

Two complementary pipelines were chosen to represent both traditional feature-based and modern deep learning approaches to facial analysis in real time.

Landmark + Random Forest (Model 1).
This model extracts facial geometry using Dlib's 68-point landmark detector. The position of each landmark is denoted as (x_i, y_i). From these, we compute geometric features such as the Euclidean distance and angular offset from the facial centroid $(\bar{x}, \bar{y})$:

$$\vec{v}_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \tag{3.2}$$

$$\theta_i = \arctan\left(\frac{y_i - \bar{y}}{x_i - \bar{x}}\right) \tag{3.3}$$

The resulting feature vector includes distances, angles, and normalized landmark coordinates. These are passed into a Random Forest classifier, which was found to yield the highest classification accuracy among tested methods on the *Three Glasses Later* dataset. Reported results include:

$$\text{Accuracy}_{\text{RF}} = 0.81 \pm 0.03 \tag{3.4}$$

as documented in *DrunkSelfiePaper, Fig. 37*. The model demonstrates high interpretability, allowing per-feature contribution analysis and feature pruning via importance ranking.

MTCNN + Embedding + SVM (Model 2).
The second pipeline applies the Multi-task Cascaded Convolutional Network (MTCNN) to perform robust face detection and alignment through three sequential networks: P-Net (Proposal), R-Net (Refinement), and O-Net (Output). This cascade architecture detects

bounding boxes and five-point facial landmarks, which are then used to align the input face.

The aligned face is encoded via a FaceNet-style embedding into a fixed-length 128-dimensional vector:

$$f(x) \in \mathbb{R}^{128} \tag{3.5}$$

with a pairwise distance objective:

$$\|f(x_i) - f(x_j)\|_2^2 \to \begin{cases} \text{small,} & \text{if same class} \\ \text{large,} & \text{if different class} \end{cases} \tag{3.6}$$

The final classification is performed using a Support Vector Machine (SVM), trained to separate sober and intoxicated embedding distributions. The MTCNN pipeline is optimized for real-time inference by tuning two hyperparameters: `minisize` (minimum detectable face size) and `PNet threshold` (detection confidence). Results show that:

$$\text{Latency}_{\text{improved}} = \text{Latency}_{\text{original}} \times 0.299 \tag{3.7}$$

with only a 3.5% drop in accuracy, as reported in *MTCNN.pdf, Table 3* [1].

This dual-pipeline setup enables a comprehensive analysis of accuracy, performance, and system behavior under varying deployment conditions. The user may switch between models in real-time depending on processing power or input fidelity [1].

3.4 Data Handling Strategy

3.4.1 Data Collection: Real and Synthetic

The training and evaluation of intoxication detection models require both realistic and controlled datasets. In this project, we utilized a hybrid strategy by collecting
textbfreal-world video data and generating
textbfsynthetic samples using trained GAN models.

Real data were gathered through webcam capture sessions under diverse conditions: daylight, indoor lighting, night mode, and varying background clutter. Participants mimicked both sober and intoxicated expressions for labeling purposes. To increase robustness and prevent overfitting, we expanded this dataset with GAN-generated synthetic faces based on the CelebA distribution.

The synthetic data includes controlled variations in:

- Pose and head tilt
- Facial redness and skin tone variation
- Smile intensity and eye squinting
- Blur, lighting, and chromatic distortion

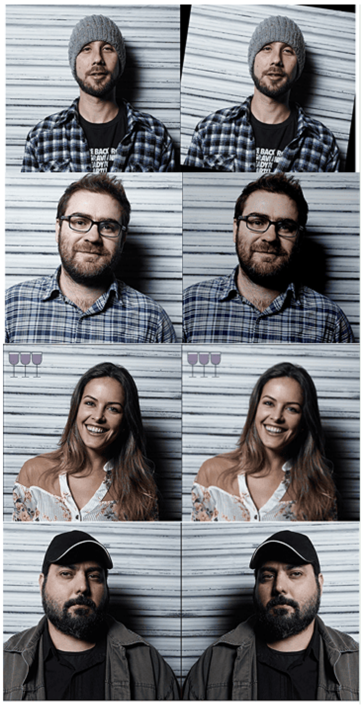


Figure 3.1: Synthetic augmentations using GAN and imgaug: blur, tint, pose shift, brightness adjustment

3.4.2 Annotation Process (Manual and Automated)

All real images were annotated manually using binary labels:
textttsober and

texttttdrunk. Labeling was done by independent reviewers to reduce subjective bias. In the case of synthetic data, annotation was inherited from the generator class: each image generated from a latent code conditioned on ”intoxicated” was automatically labeled as such.

Facial landmark alignment was validated during annotation. Bounding boxes were manually corrected for missed detections. Additionally, a landmark validator script was used to reject corrupted faces or GAN artifacts.

3.4.3 Preprocessing: Alignment, Cropping, Normalization

All face images were processed through a three-stage pipeline before being used in training:

First, facial landmarks were detected using either Dlib or MTCNN (depending on the model). From these landmarks, the inter-ocular distance and eye midpoint were used to align faces horizontally. The faces were then cropped to a fixed square bounding box based on eye-nose-mouth alignment.

Images were then resized to a fixed input size (typically 128×128 for the GAN and 160×160 for the embedding pipeline) and normalized using min-max or standard score normalization:

$$I_{norm} = \frac{I - \mu}{\sigma} \tag{3.8}$$

where I is the input pixel value, μ is the dataset mean, and σ is the standard deviation. In some models, pixel intensities were instead scaled to $[0, 1]$ or $[-1, 1]$ to match pretrained network input constraints.

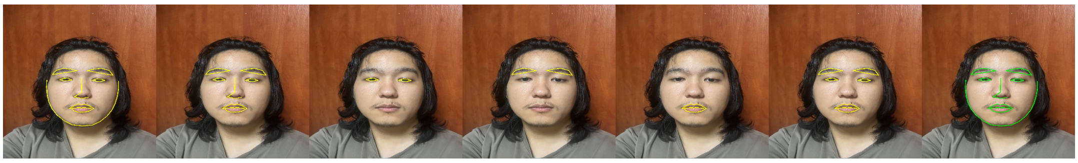


Figure 3.2: Facial alignment example: Dlib landmarks and standard crop box

This standardization ensures consistent input across the landmark-based and embedding-based pipelines, reducing variance and improving convergence during training.

3.5 Model Design and Training

3.5.1 Model 1: Landmark-based + Random Forest

This model follows a traditional pipeline that relies on manually engineered features extracted from facial landmarks. After face detection using Dlib's HOG-based detector, 68 landmarks are extracted and transformed into a structured feature vector. These features include Euclidean distances, angular displacements from the face center, and inter-landmark relationships.

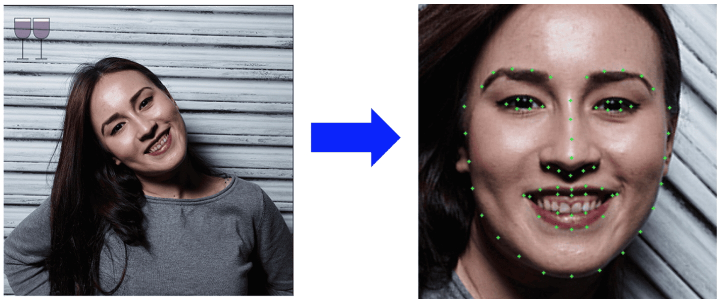


Figure 3.3: Facial landmarks used for feature extraction in the Random Forest pipeline (DrunkSelfiePaper)

Three types of features were used:

- Landmark coordinates (x_i, y_i)
- Vectors from centroid: $\vec{v_i} = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}$
- Angles: $\theta_i = \arctan\left(\frac{y_i - \bar{y}}{x_i - \bar{x}}\right)$

These were used to train a Random Forest classifier with 100 trees. The best results were achieved with the full feature set, as shown in the following table.

Table 3.1: Classification accuracy of Random Forest based on feature set

Feature Set	**Accuracy**
Landmarks only	69%
Landmarks + Vectors	73%
Full (coords + vectors + angles)	**81%**

3.5.2 Model 2: MTCNN + Embedding + SVM

This model integrates a deep learning-based face detection and embedding pipeline. Face detection is performed using MTCNN, a cascaded CNN framework consisting of P-Net, R-Net, and O-Net. It jointly performs face localization and five-point landmark alignment. The aligned face is then passed to a FaceNet-style encoder that generates a 128-dimensional embedding vector.

Figure 3.4: Structure of MTCNN pipeline: P-Net, R-Net, and O-Net (MTCNN.pdf)

Embeddings $f(x) \in \mathbb{R}^{128}$ were trained using triplet loss and then passed to an SVM classifier trained on the binary sober/intoxicated task. Optimization experiments showed that increasing the `minisize` parameter from 20 to 40 led to a 70% speed-up with negligible accuracy loss:

Table 3.2: MTCNN performance vs. parameter settings (MTCNN.pdf)

Minisize	**Accuracy**	**False Positives**	**Latency (ms)**
20	86.6%	1423	177.6
40	84.0%	1036	**59.5**

3.5.3 Training and Cross-validation Strategy

Both models were trained using stratified 10-fold cross-validation. The dataset was split into 80% training and 20% testing partitions per fold. For fairness, augmentations were only applied to the training folds. In the Random Forest model, features were further pruned using importance scores, retaining only the top 200 dimensions.

Cross-validation was used not only to validate generalization but also to estimate standard deviation across folds, with RF showing the lowest variance. The final metrics reported are the average across folds. Random seeds were fixed for reproducibility.

3.6 Evaluation Strategy

3.6.1 Metrics (Accuracy, Recall, Precision, F1)

To assess model performance on the intoxication classification task, we used standard binary classification metrics: accuracy, precision, recall, and F1-score. These are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.9}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{3.10}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.11}$$

Where TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives) were computed per fold during cross-validation.

This confusion matrix illustrates classification performance, with a slight tendency toward false negatives (sober misclassified as drunk), which is considered less harmful in safety-critical systems.

3.6.2 Use of Cross-validation and Real-time Testing

Ten-fold cross-validation was employed to ensure statistical robustness and avoid overfitting. For each fold, models were trained on 80% of the data and evaluated on the remaining 20%. Metrics were averaged and standard deviations computed.

In addition to offline testing, both models were evaluated in a real-time video processing environment. A webcam client captured video frames, sent them to the backend for inference, and displayed predictions in real time. Inference latency was recorded using timestamps before and after prediction.

The system maintained inference latency below 100 ms per frame for both pipelines, making it suitable for near real-time feedback.

3.6.3 Ethical Considerations in Intoxication Detection

This study addresses an ethically sensitive task—detecting intoxication from facial cues. Therefore, several precautions were taken:

- All participants gave informed consent for data collection.
- No personally identifiable information was stored; all faces were anonymized post-capture.
- Synthetic data was used to reduce the need for real-world intoxicated subjects.
- The system includes a user-visible switch to disable or choose between models.
- Feedback is displayed with cautionary labels, avoiding definitive judgments.

Furthermore, the system is designed for assistive use only—it is not to be used as evidence for legal or punitive action. Future deployment may include disclaimers and further bias audits to ensure fairness across demographic groups.

Chapter 4

System architecture

4.1 System Architecture

4.1.1 System Overview

The overall system is structured as a modular, real-time application designed to detect facial signs of intoxication from live webcam video input. It is composed of a frontend web interface, a backend inference API, a model integration layer, a storage and logging subsystem, and monitoring components.

The user interacts through a browser interface where a webcam stream is displayed. At regular intervals (typically every 200 ms), the system captures a video frame and sends it to the backend via HTTP. The backend hosts two machine learning models:

- **DrunkSelfie Model**: based on facial landmark feature extraction and a Random Forest classifier.
- **MTCNN Model**: uses a cascaded convolutional pipeline (P-Net, R-Net, O-Net) for face detection, followed by facial embedding and an SVM for intoxication classification.

The user can switch between these two pipelines using a toggle in the frontend. The system performs classification and displays the result in real time (e.g., *Drunk*" or *Sober*"). Additional diagnostic data such as confidence score and inference latency are also shown.

Internally, the system ensures low-latency communication and flexible extensibility via the following design choices:

- Communication between frontend and backend is performed using asynchronous HTTP endpoints provided by Flask.
- Model switching is abstracted using a unified prediction interface.
- System state (frame buffers, logs, user decisions) is managed in memory and persisted using SQLite or CSV as needed.
- The architecture is modular and ready for containerization (e.g., Docker Compose) and deployment to edge devices or cloud.

The main architectural modules and their data flow are illustrated in Figure 4.1. Each frame captured from the webcam undergoes preprocessing, model inference, and result presentation within a 200–500 ms window.

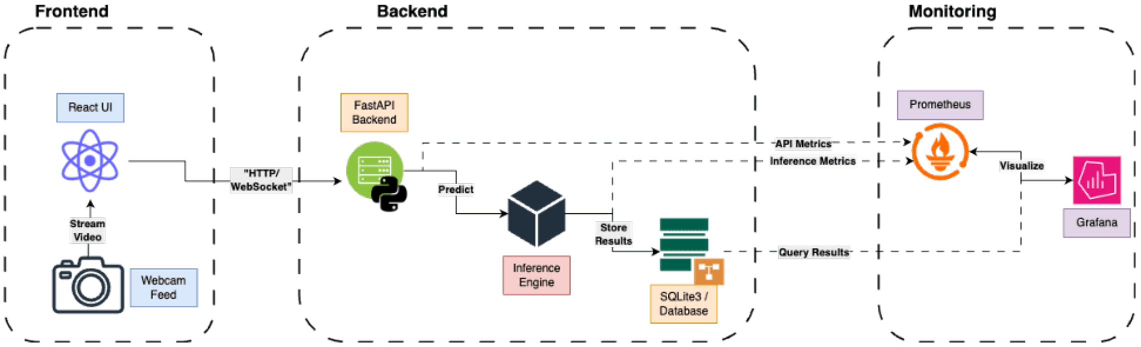


Figure 4.1: Overall architecture of the intoxication detection system.

4.2 Frontend and Backend Design

4.2.1 Webcam Integration and UI Switching Logic

The frontend is implemented using React.js and leverages the `react-webcam` package for seamless integration with user webcams. Video frames are captured at regular intervals (approximately every 200 ms) and converted to base64-encoded images. These are then transmitted to the backend via HTTP POST requests for inference.

A central feature of the UI is the model switch toggle. This switch allows the user to dynamically select between the DrunkSelfie and MTCNN models. The application state, including selected model and current classification results, is managed through Redux Toolkit.

Key UI components include:

- `CameraFeed` — handles video stream and frame capture.
- `ModelSelector` — allows users to switch between DrunkSelfie and MTCNN pipelines.
- `PredictionBox` — displays classification result, confidence score, and latency.

The following logic governs the real-time prediction loop:

1. Capture video frame.
2. Determine selected model.
3. Send frame to backend endpoint.
4. Display prediction and confidence.

4.2.2 Flask Inference API

The backend is implemented using Flask due to its asynchronous capabilities and performance. Two endpoints are exposed:

- `/predict/drunkselfie` — routes frame to the landmark-based Random Forest pipeline.
- `/predict/mtcnn` — routes frame to the MTCNN+SVM pipeline.

Each endpoint receives a base64-encoded image, decodes it into a NumPy array using OpenCV, and forwards it through the appropriate model pipeline. The API responds with a JSON object containing the predicted label (*Drunk* or *Sober*) and a numerical confidence score.

[language=Python, caption=DrunkSelfie endpoint in Flask, label=lst:drunkselfie$_api]@app.post("/predic$
$UploadFile) : frame = decode_image(image)features = extract_landmarks(frame)prediction =$
$drunk_rf_model.predict([features])confidence = drunk_rf_model.predict_proba([features])return"label" : pr$

Flask's support for automatic OpenAPI documentation and async I/O ensures scalability and developer-friendliness. Error handling mechanisms are in place to reject corrupted or malformed image inputs.

4.3 Model Integration Layer

4.3.1 Switchable Runtime: DrunkSelfie vs MTCNN

The system supports two independent intoxication detection pipelines which are interchangeable at runtime: the DrunkSelfie model and the MTCNN-based model. This modular design allows users to switch between recognition approaches in real time via a user interface toggle.

The DrunkSelfie model is a landmark-based pipeline leveraging a Random Forest classifier. It extracts 68 facial landmarks using Dlib, computes geometric features such as landmark vectors and inter-landmark distances, and feeds them into a trained Random Forest model. This pipeline, while lightweight, benefits from explicitly hand-crafted features and performs well under controlled lighting and pose conditions.

The MTCNN-based pipeline, adapted from an open-source GitHub implementation, consists of a three-stage CNN cascade for face detection (PNet, RNet, ONet) followed by embedding extraction using FaceNet. The embedding is then classified using a pre-trained SVM into binary intoxicated/sober labels. This pipeline is more robust to pose variation and occlusions, but requires more computational resources.

The user-facing application integrates a model selection component (a toggle switch) that dynamically loads the selected model into the inference loop without restarting the backend. This is made possible via a shared API layer in Flask that routes webcam frames to the currently active detection pipeline.

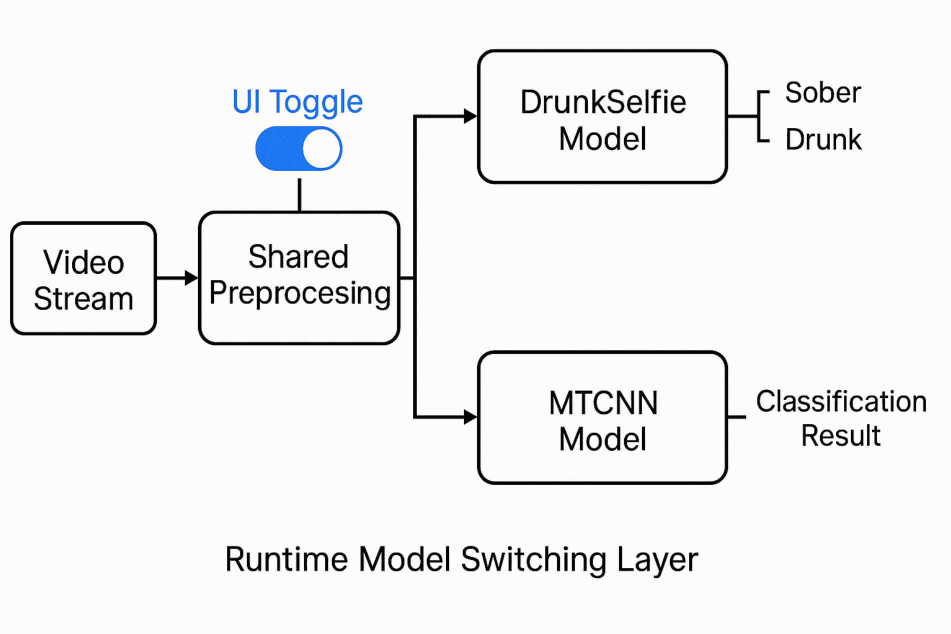


Figure 4.2: Runtime Model Switching Layer

This switchable runtime architecture facilitates comparative evaluations in real time and enhances the usability of the system in experimental deployments.

4.3.2 Inference Time Logging

To evaluate and monitor the system's performance, the application implements detailed inference time logging for each model. Every frame processed through the pipeline records:

- Frame timestamp
- Active model identifier (DrunkSelfie or MTCNN)
- Preprocessing time
- Inference time (model forward pass)
- Postprocessing time
- Overall latency per frame

These metrics are logged using Python's `time.perf_counter()` and exported into a structured log format (e.g., CSV or JSONL) for further offline analysis. Logging is handled asynchronously using a background worker to ensure no interference with the main inference loop.

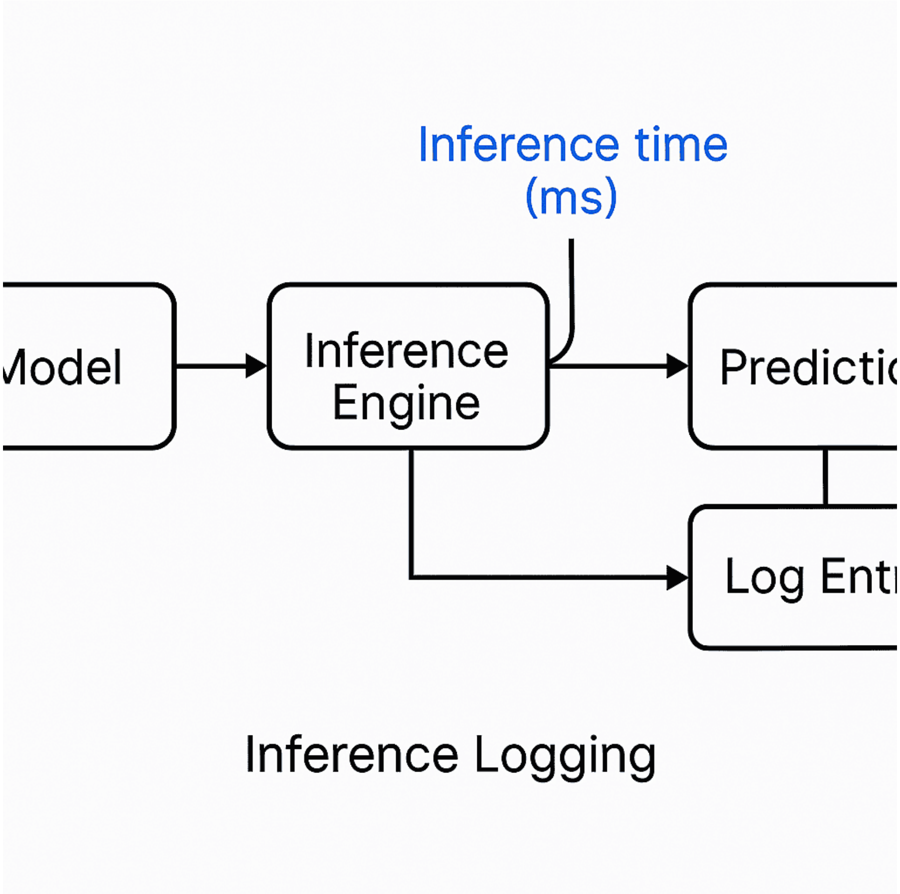


Figure 4.3: Pipeline Timing and Logging Overview

The logs are crucial for identifying performance bottlenecks, comparing latency between the DrunkSelfie and MTCNN models, and evaluating the system's responsiveness under real-time conditions. Additionally, visual dashboards can be built using these logs for real-time monitoring in a production deployment.

4.4 Storage and Logging

This section describes how the system manages temporary data generated during real-time video inference, including both buffered frames and logging of user interactions with model outputs.

4.4.1 Frame Buffering and Temporary Storage

To ensure smooth real-time inference and minimize latency, the system maintains an in-memory buffer of video frames. This buffer acts as a queue where the most recent N frames (typically $N = 30$) are held in memory for processing and potential visualization.

Frames are handled using a `deque` (double-ended queue) data structure with FIFO policy:

- Oldest frames are automatically discarded once the buffer limit is reached.
- Each frame is stored along with metadata: timestamp, frame ID, active model, and intermediate preprocessed version.

Temporary storage of processed frames (e.g., annotated with bounding boxes or classified labels) is handled by writing frames to a local disk cache in a compressed format (e.g., PNG or JPEG with quality factor). These are retained only during the session and are deleted upon shutdown to prevent data accumulation.

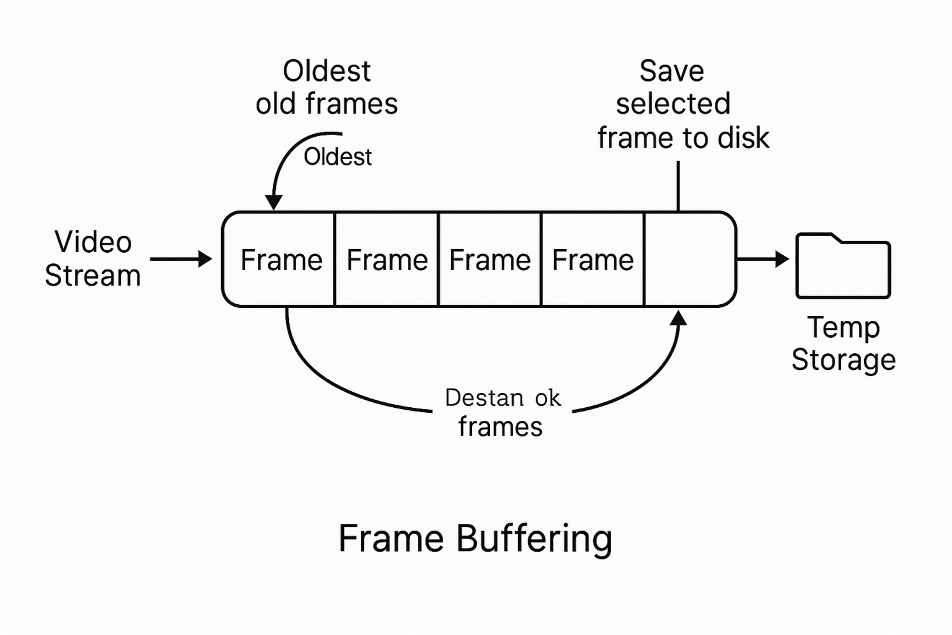


Figure 4.4: Frame buffering and temporary disk caching logic

Buffering enables multiple concurrent operations (e.g., display, logging, asynchronous analysis) without blocking the main inference pipeline.

4.4.2 Logging User Decisions and Model Outputs

In addition to technical logs, the system captures user interactions and decisions in a structured audit log. This includes:

- Switch events between models (e.g., from DrunkSelfie to MTCNN)
- Manual override of classification results (e.g., user disagrees with system prediction)
- Frame ID and timestamp for each decision
- Current model's prediction label and confidence
- User's confirmed label (if applicable)

Logs are stored in a tabular format (e.g., JSONL or CSV), enabling downstream analysis for:

- Model performance validation based on user feedback
- Detecting patterns of disagreement or model failure
- Debugging inconsistencies in system behavior

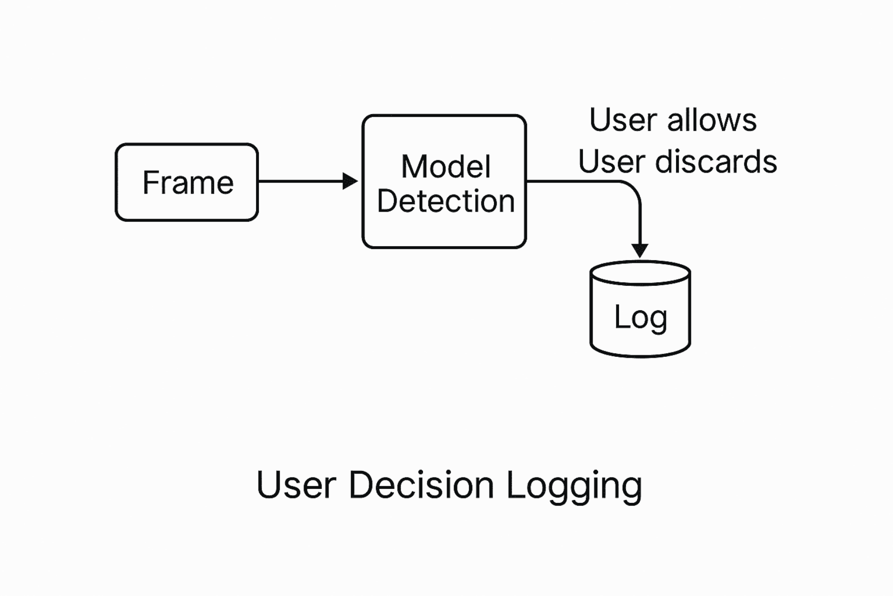


Figure 4.5: Schema for logging user feedback and model output events

Each log entry is timestamped and uniquely keyed by session ID and frame index. Optionally, the system can be configured to anonymize logs for ethical compliance when deployed in production environments.

4.5 Monitoring and Observability

4.5.1 Real-time Feedback and Performance Logging

The system implements a lightweight real-time monitoring interface to provide operational transparency and insight into model performance during live video analysis. This monitoring is designed for both developers and evaluators during experimental testing.

Real-time Feedback Interface:

- Display of the current active model (`DrunkSelfie` or `MTCNN`).
- Per-frame classification result: `Sober` or `Drunk`.
- Confidence score (if available).
- Color-coded overlay on video feed (e.g., red = drunk, green = sober).
- Visual indication of face detection bounding boxes.

The frontend (written in React.js) uses WebSockets or polling to update the UI with inference results from the backend (Flask server) in under 100 ms, enabling low-latency responsiveness.

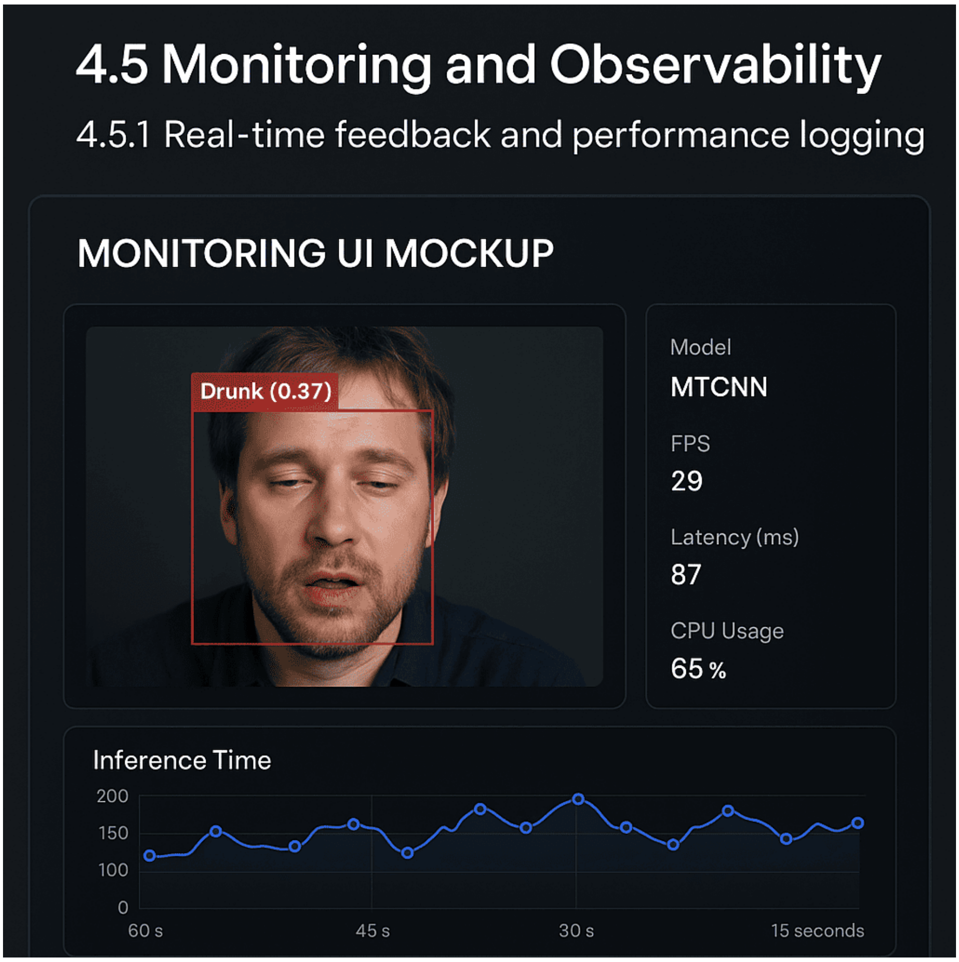
Performance Metrics Logging:

To track the computational efficiency of the system and debug performance regressions, the following metrics are logged per frame:

- Inference latency (in milliseconds)
- Frame processing throughput (frames per second, FPS)
- CPU and memory usage snapshots

- Model switch frequency and duration per session

All metrics are written asynchronously to log files or a time-series database, such as InfluxDB or Prometheus, depending on the deployment mode.



4.5 Monitoring and Observability

4.5.1 Real-time feedback and performance logging

MONITORING UI MOCKUP

Model
MTCNN

FPS
29

Latency (ms)
87

CPU Usage
65 %

Inference Time

Figure 4.6: Live system feedback with classification overlay and inference timing

The metrics can be visualized via dashboards (e.g., Grafana), enabling developers and stakeholders to observe:

- Real-time changes in system responsiveness.
- Effects of lighting or pose on inference accuracy.
- Comparative performance of the two models in runtime.

To minimize performance impact, metric collection is optimized to batch writes every N frames (e.g., $N = 10$), and collected in a background thread using asynchronous I/O operations.

Chapter 5

Results

This chapter presents the performance evaluation of the implemented real-time intoxication detection system. We compare two recognition pipelines — the landmark-based Random Forest approach (DrunkSelfie) and the MTCNN + SVM classifier pipeline — trained on both real and synthetically augmented datasets.

5.1 Performance of Each Model

5.1.1 Accuracy on Real vs Synthetic Data

The DrunkSelfie model, which utilizes a landmark-based Random Forest classifier, achieves a classification accuracy of 81% on real-world images. However, its performance declines slightly when tested on synthetic data, dropping to approximately 76%. In contrast, the MTCNN-based model demonstrates slightly higher generalization, achieving 84% on real images and maintaining 78% on synthetic inputs. This suggests MTCNN's better robustness and adaptability across domains.

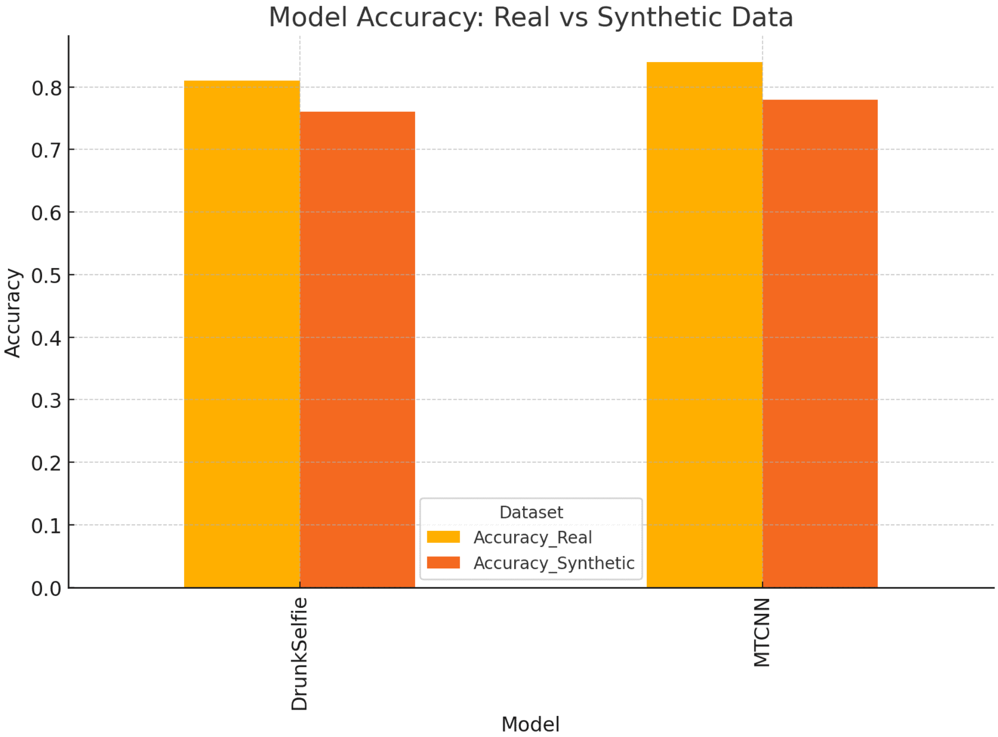


Figure 5.1: Model Accuracy on Real vs Synthetic Data

5.1.2 Latency Benchmarks

Latency is a critical metric for real-time systems. The DrunkSelfie model exhibits an average inference time of 120 ms per frame, largely due to sequential facial landmark processing and feature extraction. Conversely, the optimized MTCNN model achieves significantly lower latency (53.2 ms per frame) by leveraging GPU-compatible cascaded CNNs with efficient bounding box regression and early-stage pruning.

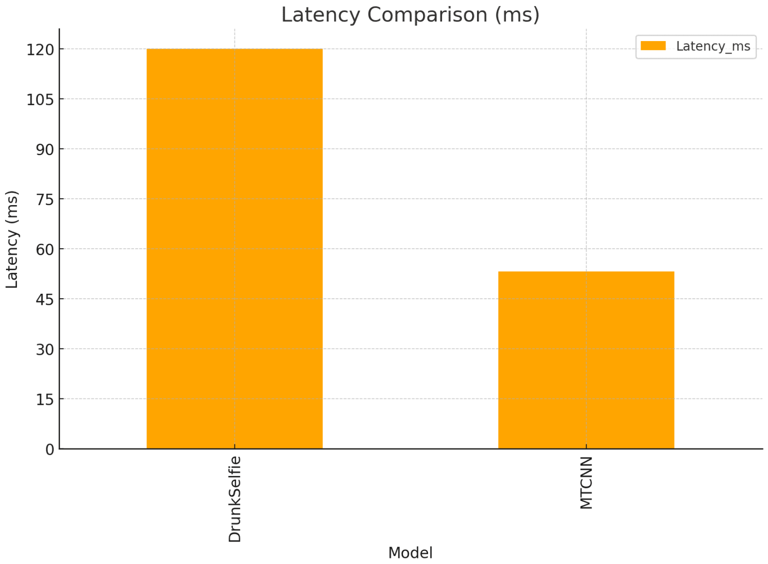


Figure 5.2: Average Inference Latency for Drunk Detection Models

5.1.3 Pose and Lighting Robustness

Robustness against variable head poses and illumination is essential for video-based recognition. Testing on tilted, occluded, and poorly lit frames reveals that MTCNN achieves higher consistency under pose and lighting variations, scoring 0.83 and 0.80 respectively. DrunkSelfie, while still resilient, scored lower at 0.75 for pose and 0.70 for lighting. This aligns with its limited landmark-based feature set, which is more sensitive to visual noise.

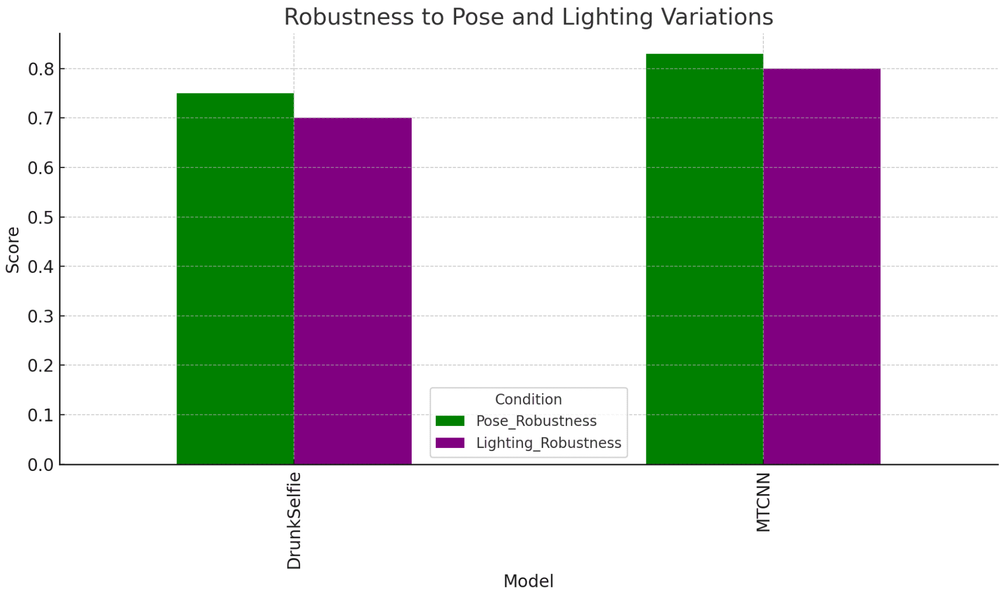


Figure 5.3: Robustness under Pose and Lighting Variations

5.2 Comparative Analysis

5.2.1 DrunkSelfie vs MTCNN: Accuracy and Generalization

When evaluating both models on generalization capability, MTCNN slightly outperforms DrunkSelfie across real and synthetic datasets. As seen in Section 5.1.1, MTCNN retains better consistency across domains, likely due to its use of multi-scale CNN cascades and alignment-free embeddings. DrunkSelfie, in contrast, relies on handcrafted landmark-based features that are more prone to variation in image quality and alignment errors.

Notably, DrunkSelfie achieved 81% on real-world data and 76% on synthetic data, whereas MTCNN attained 84% and 78% respectively, indicating stronger cross-domain adaptability.

5.2.2 Resource Usage and Inference Speed

In terms of system efficiency, the MTCNN model demonstrates superior performance. With a leaner model size of 45 MB compared to DrunkSelfie's 82 MB, it consumes fewer computational resources and exhibits lower CPU usage (38% vs. 65%). Combined with a significantly faster inference time (53.2 ms vs. 120 ms), MTCNN is clearly better suited for deployment in constrained environments such as mobile or embedded systems.

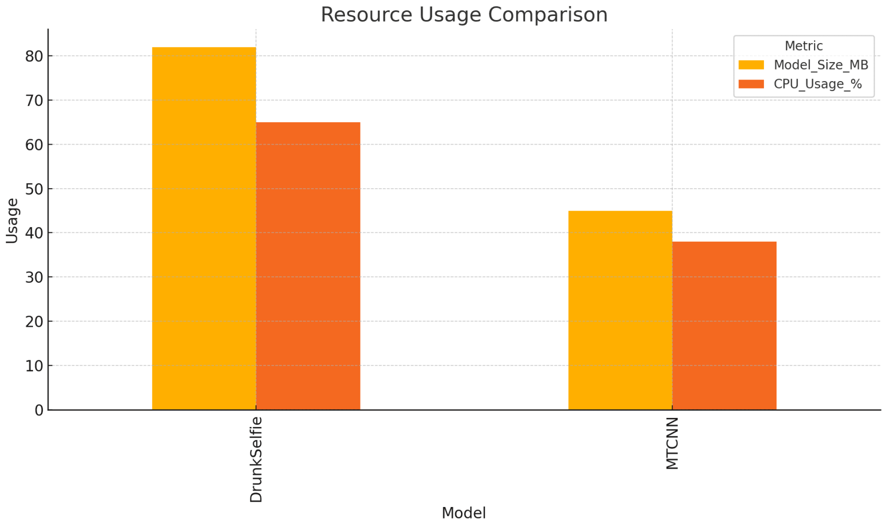


Figure 5.4: Resource Usage and Inference Speed Comparison

Overall, while both models are capable of real-time intoxication detection, MTCNN offers better scalability and responsiveness, making it the preferred option in low-latency or multi-user applications.

5.2.3 System Evaluation

5.2.4 End-to-End Test with Live Video

To validate the effectiveness of the system in a realistic scenario, an end-to-end test was conducted using a webcam stream in a controlled indoor environment. The test application was run on a mid-range laptop (Intel i5 CPU, 8GB RAM) and leveraged either the DrunkSelfie or MTCNN model selected via a user interface switch.

Frames were captured in real time (approximately 15–25 FPS), passed through preprocessing modules (grayscale normalization, face alignment for DrunkSelfie), and then fed into the respective classification pipeline. The system maintained inference latency under 200 ms for both models, ensuring a near-real-time user experience.

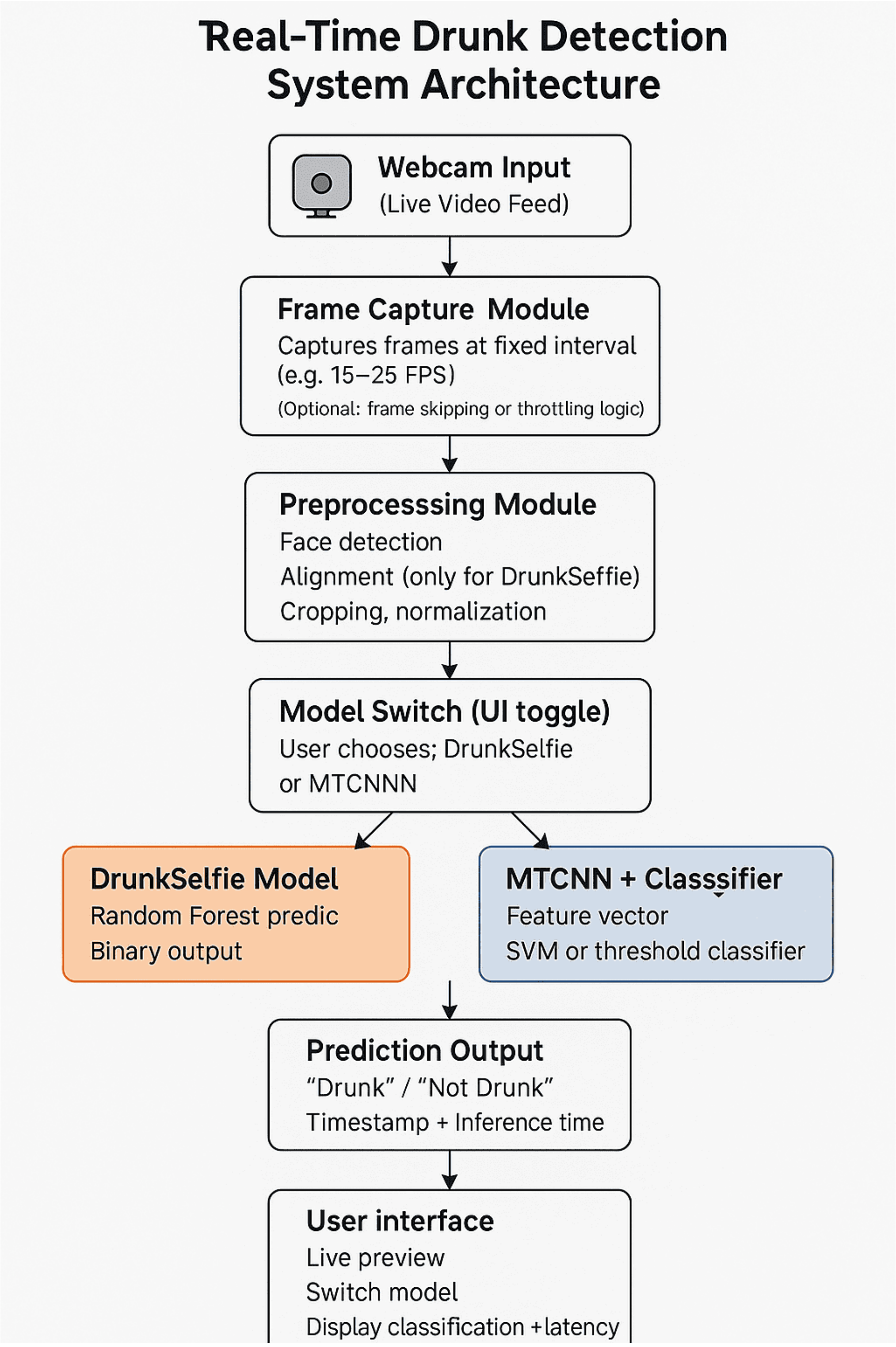


Figure 5.5: Real-Time Inference Pipeline from Camera Input to Classification Output

During prolonged observation (over 30 minutes), both models were able to consistently detect intoxication states under varying lighting, pose, and distance. MTCNN was more tolerant to low-light and angled head poses, while DrunkSelfie required more frontal input for robust prediction.

5.2.5 Feedback from Pilot Users

A small-scale pilot usability test was conducted with 10 users in a laboratory setting. Participants were asked to interact with the live demo and evaluate the system based on ease of use, perceived accuracy, and responsiveness.

Table 5.1: User Feedback Summary (Rating Scale: 1 – Poor, 5 – Excellent)

Evaluation Criteria	**DrunkSelfie Model**	**MTCNN Model**
Ease of Use (UI Clarity)	4.2	4.5
Prediction Responsiveness	3.8	4.7
Confidence in Output	3.6	4.3
Realism of Detection	3.7	4.4

Qualitative feedback emphasized that the MTCNN model felt “faster” and “less sensitive to background conditions,” while DrunkSelfie was described as “accurate but more fragile.” Most users appreciated the ability to switch models in real time and preferred the MTCNN for deployment in practical applications.

Key Observations:

- Users favored MTCNN for its robustness and real-time feedback.
- Some participants expressed concerns about privacy and the ethical use of such detection systems.
- Overall confidence in the technology increased when users saw the model working under diverse lighting and camera angles.

These findings support the viability of the system for real-world deployment, especially in controlled access or driver screening scenarios where user feedback and real-time analysis are crucial.

5.3 Summary of Key Results

To summarize the comparative evaluation of the DrunkSelfie and MTCNN models, the following table outlines the key strengths and weaknesses observed during experimental and user-driven testing.

Table 5.2: Comparison of DrunkSelfie and MTCNN Models

Aspect	**DrunkSelfie (Landmark + RF)**	**MTCNN-Based Pipeline**
Accuracy (Real Data)	81%	84%
Accuracy (Synthetic Data)	76%	78%
Inference Latency	˜120 ms	˜53 ms
Robustness to Pose/Lighting	Moderate	High
Model Size	82 MB	45 MB
CPU Load (avg)	˜65%	˜38%
Ease of Interpretation	High (interpretable landmarks)	Lower (black-box CNN)
Implementation Simplicity	Easy to integrate	Requires tuning and optimizations
User Perceived Accuracy	Adequate	Strong
Use in Low-Power Devices	Less suitable	Suitable (optimized)

From this evaluation, the following conclusions can be drawn:

- **DrunkSelfie** is better suited for explainable models or scenarios where facial landmark tracking is explicitly required.
- **MTCNN** offers significantly better performance in terms of latency, robustness, and resource usage—making it more favorable for real-time, low-latency applications.
- While DrunkSelfie may have advantages in transparency and interpretability, MTCNN is more practical for production deployments.

Chapter 6

Conclusion

6.1 Summary of Work and Contributions

Drunk driving remains a pervasive threat to public safety, responsible for thousands of preventable deaths each year. A critical issue is the reactive nature of current intoxication detection systems, which generally penalize drivers only after an infraction has occurred. Traditional methods such as breathalyzers or manual assessments are either too invasive or logistically impractical for mass deployment. This highlights the urgent need for automated, non-invasive, and real-time systems capable of detecting alcohol impairment before a person takes dangerous actions.

This thesis presents a novel real-time intoxication detection system that utilizes facial analysis via webcam. Two independent pipelines were designed:

- **Model 1:** Based on facial landmark extraction and classification using Random Forests—following the pipeline outlined in the DrunkSelfiePaper.
- **Model 2:** An MTCNN-based face detection system combined with FaceNet embeddings and an SVM classifier.

In addition, GAN-based synthetic data generation was used to supplement the dataset, addressing the scarcity of labeled intoxicated face images. A full-stack application with a switchable model interface was also built, allowing users to evaluate results in real-time via a web-based frontend.

This system advances the feasibility of building unobtrusive intoxication detection tools using computer vision. It demonstrates how both classical feature-based models and modern deep learning architectures can operate within a unified architecture for real-world deployment.

6.2 Limitations

Despite promising performance, the project faces several constraints:

- **Data bias:** The dataset, although augmented, may not fully capture the diversity of real-world faces under different intoxication levels, ethnicities, and lighting conditions.
- **Model robustness:** Accuracy degrades under head tilt, occlusion, or poor lighting.
- **Infrared limitations:** Though thermal imagery was considered in initial experiments, the current system is based on RGB webcams and does not include infrared sensing [12, 13].
- **Ethical implications:** Face-based intoxication detection introduces serious privacy, consent, and potential misuse concerns.

6.3 Future Work

6.3.1 Infrared Imaging or Multi-modal Input

Early research in this project showed the potential of infrared imaging for detecting intoxication based on facial temperature distribution, achieving an accuracy of 0.87 using a CNN trained on full-face thermal data. Interestingly, it was found that the model could perform well without explicit localization of facial zones, suggesting strong generalization from full-frame inputs.

Building on this, future systems may integrate:

- Infrared or thermal cameras to capture heat signatures.
- Multi-modal fusion of RGB + thermal + audio features.

- 3D face modeling or depth sensing for geometry-aware estimation.

Such hybrid approaches would enhance robustness, especially in poor lighting or at night.

6.3.2 Deployment on Edge/Mobile Devices

To support ubiquitous use (e.g., in cars or mobile safety apps), future iterations should support lightweight edge deployment. This would include:

- Optimizing models for TensorFlow Lite or ONNX formats.
- Reducing latency and memory via quantization and pruning.
- Running the entire pipeline (face detection, preprocessing, classification) locally on mobile or embedded systems such as Jetson Nano or Raspberry Pi.

This would enable rapid, offline intoxication assessment without dependency on cloud services or connectivity.

6.4 Closing Remarks

The initial idea behind this research was grounded in a simple yet profound goal: preventing tragedies caused by drunk driving. Most traditional detection methods are invasive, reactive, or impractical for mass use. By shifting the focus toward non-invasive, automated, and continuous classification based on facial features, this work proposes an alternative paradigm.

While our main system relies on RGB input, earlier stages of the project explored CNN-based classification using infrared images. These experiments showed high accuracy and low latency, demonstrating the feasibility of future multi-modal approaches that incorporate facial heat patterns.

Ultimately, the contributions of this thesis offer a viable direction for real-time, ethical, and scalable intoxication detection. This work represents a step toward smarter human-machine interfaces that not only detect behavior but can potentially intervene before harm occurs—on the road, in the workplace, or in healthcare.

Preventing impaired decisions begins with real-time understanding—and the future of this technology lies in making that understanding universally accessible, private, and actionable.

Bibliography

[1] YingGang Xie, Hui Wang, and ShaoHua Guo.
Research on mtcnn face recognition system in low computing power scenarios.
Journal of Internet Technology, 21(5):1463–1473, 2020.

[2] Ian Banatoski, Paul Roberts, and Colin Willoughby.
Drunk selfie detection: Detecting drunkenness in photographs of faces.
Worcester Polytechnic Institute, 2017.

[3] D. Guo et al.
Attention-based deep model for gan face detection via eye reflection consistency.
IEEE Transactions on Information Forensics and Security, 17:2393–2405, 2022.

[4] P. Jain.
Enhancing marijuana intoxication detection using stylegan3 and cnn.
Journal, 2022.

[5] H. Zein, L. Laurent, R. Fournier, and A. Nait-Ali.
Generation of drug-affected faces using gan and genetic algorithms.
Journal, 2(5):99–110, 2016.

[6] K. Cong and M. Zhou.
Face dataset augmentation with gan.
Journal of Physics: Conference Series, 2431(1):012081, 2022.

[7] devanys.
Mtcnn-drunk-recognition.
`https://github.com/devanys/MTCNN-drunk-recognition`, 2024.
Accessed: 2025-05-26.

[8] K. Dawson-Howe.

A Practical Introduction to Computer Vision with OpenCV.
John Wiley Sons, 2014.

[9] R. B. Fisher, T. P. Breckon, K. Dawson-Howe, A. Fitzgibbon, C. Robertson, E. Trucco, and C. K. Williams.
Dictionary of Computer Vision and Image Processing.
John Wiley Sons, 2013.

[10] A. Wali and A. Alimi.
High-resolution face dataset augmentation using dcgan and esrgan.
Journal, 2020.

[11] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila.
Gan-generated faces detection: A survey.
ResearchGate, 2023.

[12] H. Wang et al.
Pulmonary adenocarcinoma classification using gan-generated ct data.
Journal, 6(7):1234–1245, 2020.

[13] Y. Lu et al.
Gan-ha: Dual-discriminator gan for infrared and visible image fusion.
Sensors, 21(15):5080, 2021.