Face Recognition System

### Project Proposal

## 

## Supervisor

Ms. Hafsa Batool

## Submitted by

Muhammad Ubaidullah [BSCS 22087]

Mudassir Hussain [BSCS 22131]

**Department of Computer Science**

Information Technology University

[10-08-2025]

# **Introduction and Background**

In recent years, face recognition technology has become a core component of modern security and authentication systems, with widespread use in smartphones, surveillance, and access control. However, most existing solutions rely on powerful hardware, making them unsuitable for low-resource environments like older computers, embedded systems, or budget smartphones.

This project aims to develop a lightweight and efficient face recognition system optimized for such devices, which are common in rural areas, educational institutions, and low-budget setups. The system will prioritize low memory usage, reduced latency, and compatibility with CPUs, ensuring reliable performance even under hardware limitations.

We plan to explore optimization and fine-tuning techniques such as pruning and the use of lightweight algorithms suitable for low-resource environments. This year-long project focuses on applying machine learning, particularly pre trained convolutional neural networks (CNNs), for real-time face detection and recognition. The goal is to develop an accurate and efficient system while enhancing our understanding of machine learning techniques. Ultimately, this project seeks to make face recognition more accessible and practical for broader, resource-constrained deployments.

# **Problem Description and Project Objective**

Face recognition has become a widely adopted technology in modern systems for authentication, surveillance, and access control. However, most current solutions are designed for high-performance environments with access to GPUs and advanced computing resources. These requirements limit the deployment of such systems in areas where only low-cost or outdated hardware is available, such as rural institutions, small businesses, or edge devices.

This gap between advanced technology and limited infrastructure creates a need for lightweight, efficient, and reliable face recognition systems that can work effectively under hardware constraints even on laptop’s low-resolution cameras. Key challenges include maintaining accuracy under low-light conditions, handling partially visible faces, and ensuring low memory and processing overhead. There is also a growing need to balance speed and performance without compromising recognition reliability.

**Project Objective:**  
To design a lightweight and efficient machine learning-based face recognition system optimized for real-time performance on low-resource devices without relying on GPU support.

# **Application Areas and Related Work**

Our project has **real-world value and strong practical relevance**, especially in environments where high-end computational resources are not available. Face recognition systems are increasingly used in sectors such as security surveillance, attendance tracking, and access control. However, most existing systems are designed for powerful hardware and are not feasible for widespread deployment in **low-resource settings** like public schools, rural institutions, or budget devices.

By focusing on lightweight, CPU-compatible models, our system offers an **affordable, efficient, and scalable** solution. It can be deployed on older computers or entry-level smartphones, making it ideal for:

* **Educational Institutions** – Automated attendance systems using webcams or low-end cameras.
* **Public Sector Offices** – Identity verification in budget-constrained setups.
* **Small Businesses & Rural Areas** – Basic surveillance and authentication without expensive biometric systems.

This ensures that **facial recognition becomes accessible** beyond urban or enterprise-level deployments, addressing a **gap in inclusivity and affordability**.

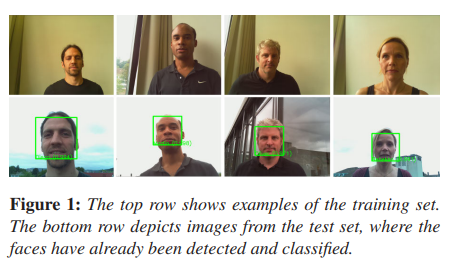
**Related Work and Literature Survey:**

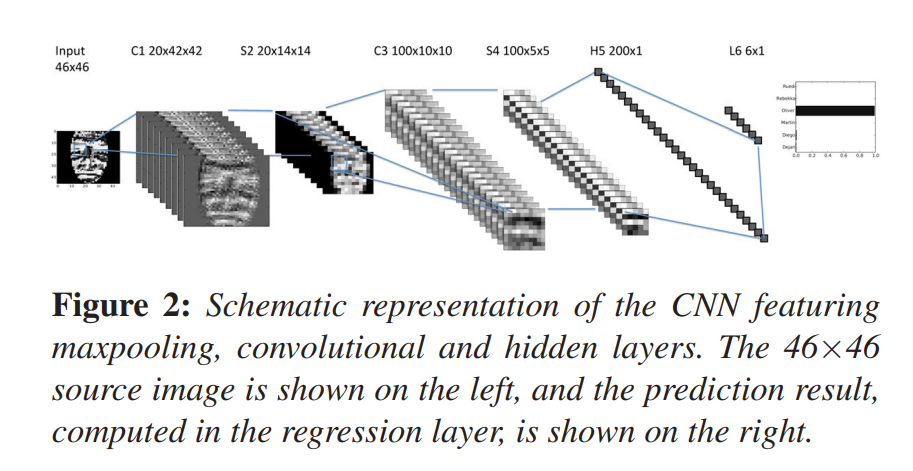
Many face recognition models like **FaceNet** and **DeepFace etc** have achieved high accuracy but rely heavily on GPUs and deep architectures. Lightweight models such as **MobileFaceNet** and **TinyFace** attempt to solve this by reducing parameters, but still present challenges for true low-resource hardware environments.

Numerous studies have focused on the challenges and potential solutions for running face recognition on constrained hardware. George et al. [1] introduced **EdgeFace**, a model tailored specifically for **edge devices**, by reducing computation and memory requirements without a significant drop in accuracy. Their work validates the feasibility of deploying face recognition models in non-GPU environments.

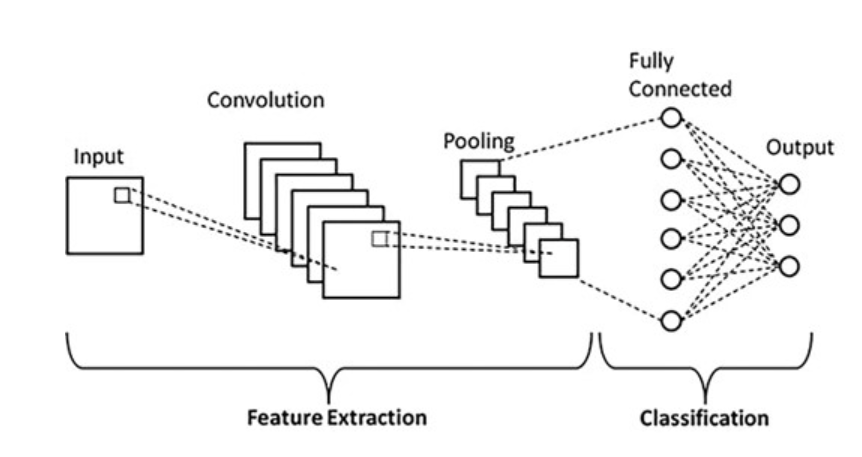
In Zhang et al. [2], a mobile-friendly CNN architecture was introduced, balancing accuracy and efficiency, but it required fine-tuning for specific environments. Similarly, the Face Recognition using OpenCV and Haar Cascades technique is fast and CPU-compatible but lacks robustness under variations in lighting, angles, and partial occlusions.

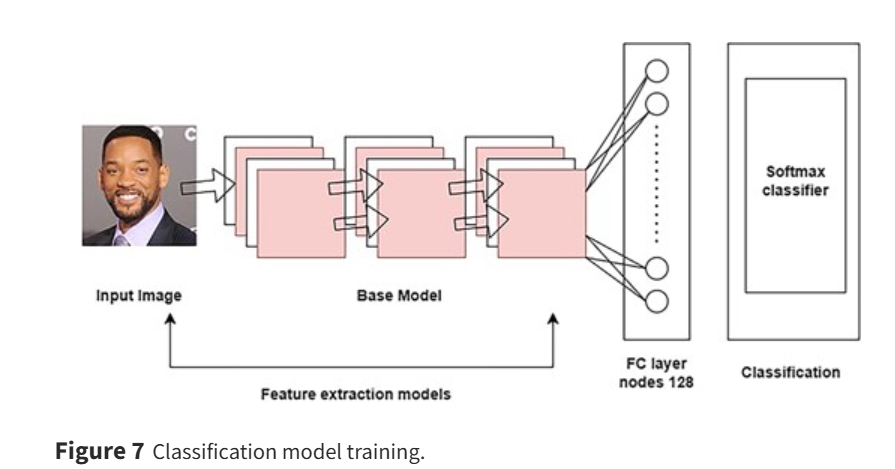
Dürr et al. [3] explored deploying deep learning-based recognition on a **Raspberry Pi**, showcasing how low-power boards can be used for practical face recognition tasks. This supports our idea of targeting **CPUs and embedded systems** as deployment environments.





Aboluhom and Kandilli [4] took a similar approach and implemented a **real-time face recognition system using transfer learning on Raspberry Pi**, demonstrating how pre-trained models can be fine-tuned and compressed for low-resource use cases.





Furthermore, Gkrispanis et al. [5] utilized **pruning techniques** to optimize lightweight face detectors, significantly reducing their size while maintaining performance. Techniques like **filter pruning and geometric median-based selection** discussed in their work are also under consideration in our project for improving model speed and responsiveness.

These studies collectively confirm the **importance and viability of lightweight face recognition systems**. Our solution builds on this foundation, aiming to create a **real-time, low-latency recognizer** using publicly available datasets and **self-captured 3D facial images** from university peers to test and refine model performance.

# **Methodology and Scope**

**Methodology:** Our approach will be divided into three main stages:

* **Data Collection & Preprocessing**: We will use publicly available datasets such as LFW and VGGFace2 for initial training and detection tasks, followed by creating a custom dataset of 3D facial images of our friends and and other students under low light and low resolution conditions. Preprocessing will involve normalization, resizing, and augmentation to improve robustness in challenging environments [6].
* **Model Development & Optimization**: We will fine-tune **pre trained machine learning models** for face recognition to adapt them for low-resource environments. Optimization methods such as model quantization and pruning will be applied to reduce computational overhead while preserving accuracy [2].
* **Testing & Deployment**: The system will be tested on low-resource devices (e.g., older laptops, cameras) to evaluate CPU-only performance, memory usage, and latency. Evaluation metrics will include accuracy, precision, recall, and inference speed, ensuring practical applicability in real-world low-resource environments [7].

**Scope:** This project focuses on the development of a **lightweight and resource-efficient face recognition system** optimized for constrained devices. Our work will not cover other biometric methods such as iris or fingerprint recognition, nor will we develop large-scale server-side infrastructures. The emphasis will be on **on-device processing** using fine-tuned pre trained models to ensure privacy, offline functionality, and adaptability to challenging conditions like low light and partial occlusion. Extreme cases, such as full face obstruction or ultra longdistance recognition, will remain outside the project scope.

# **Tools and Technology**

* **Programming Languages**: Python 3.8 or above.
* **Machine Learning Frameworks**: PyTorch or MXNet (for model training and fine-tuning)
* **Lightweight Deployment Frameworks:** TensorFlow Lite, PyTorch Mobile, ONNX Runtime (for running models on low-resource devices)
* **Libraries:** NumPy(data manipulation), OpenCV(image processing, face alignment, detection)
* **Dataset Sources**: Public datasets (LFW, VGGFace2) from Kaggle for detection and custom datasets stored locally for training and testing.

**Web Framework:**

* Django / FastAPI (for creating a simple interface to test face recognition in real-time).

**Development Tools:**

* Google Colab (for model training experiments)
* Jupyter Notebook (for local experiments and debugging)
* Git & GitHub (for version control and collaboration)

**Operating System:**

* Windows.

**Deployment:**

 The system will first be tested as a standalone application on laptops and low-processor computers using the built-in camera.

 Once validated, the solution can be integrated with low-cost security cameras or embedded devices for real-world surveillance and access control applications.

Both team members possess prior experience in Python, image processing, and neural networks, making this project well-aligned with our skills and interests.

# **6. Milestones**

|  |  |
| --- | --- |
| **Activity** | **Tentative Date** |
| Literature review and explore public datasets for detection purposes | 15 Aug 2025 - 1 sep 2025 |
| Understand, run and test existing Face Recognition implementations | 2 sep 2025 - 18 sep 2025 |
| Select datasets (public for detection + custom captured 3D images for recognition) and perform embedding extraction etc | 19 sep 2025 - 5 oct 2025 |
| Apply data augmentation techniques for robustness in low-light and partial occlusion scenarios | 6 oct 2025 - 31 oct 2025 |
| Fine-tune pre trained **machine learning/CNN models** for face recognition in constrained environments | 1 nov 2025 - 20 nov 2025 |
| Evaluate models and compare performance in terms of accuracy, speed, and resource usage | 21 nov - 10 dec |
| **Semester Break** | 15 dec 2025 - 15 jan 2026 |
| Design frontend layout and backend API structure for deployment | 15 jan 2026 - 4 feb 2026 |
| Integrate trained model with application backend for real-time camera input | 5 feb 2026 - 20 feb 2026 |
| Build and test face detection/recognition interface using laptop or low-resource device cameras | 21 feb 2026 - 17 mar 2026 |
| Perform testing on unseen datasets and real-world hardware setups (older laptops etc) | 18 mar 2026 - 7 apr 2026 |
| Write documentation and prepare FYP report | 8 apr 2026 - 25 apr 2026 |
| Create final presentation | 26 apr 2026 - 10 may |

# **References**

# [1]. A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel, “EdgeFace: Efficient Face Recognition Model for Edge Devices,” arXiv preprint, Jul. 2023. https://www.researchgate.net/publication/377324904\_EdgeFace\_Efficient\_Face\_Recognition\_Model\_for\_Edge\_Devices

**[2].** K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.  
https://ieeexplore.ieee.org/document/7553523  
[3]. O. Dürr, Y. Pauchard, D. Browarnik, R. Axthelm, and M. Loeser, “Deep Learning on a Raspberry Pi for Real Time Face Recognition,” in Proc. EUROGRAPHICS: pp. 1–2, 2015.

https://diglib.eg.org/server/api/core/bitstreams/f2ea04cd-9ad7-4410-8405-b20147533265/content

[4]. A. A. A. Aboluhom and I. Kandilli, “Face Recognition Using Deep Learning on Raspberry Pi,” The Computer Journal, vol. 67, no. 10, pp. 3020–3030, Oct. 2024.  
https://academic.oup.com/comjnl/article/67/10/3020/7754455

[5]. K. Gkrispanis, N. Gkalelis, and V. Mezaris, “Filter‑Pruning of Lightweight Face Detectors Using a Geometric Median Criterion,” arXiv preprint, Nov. 2023.  
<https://arxiv.org/abs/2311.16613>

[6]. G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments,” University of Massachusetts, Amherst, 2008.  
<https://vis-www.cs.umass.edu/lfw/>

[7]. A. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv preprint arXiv:1704.04861, 2017.  
<https://arxiv.org/abs/1704.04861>

**Image Links from Papers**

1. <https://diglib.eg.org/server/api/core/bitstreams/f2ea04cd-9ad7-4410-8405-b20147533265/content>
2. <https://diglib.eg.org/server/api/core/bitstreams/f2ea04cd-9ad7-4410-8405-b20147533265/content>
3. <https://academic.oup.com/comjnl/article/67/10/3020/7754455>
4. <https://academic.oup.com/comjnl/article/67/10/3020/7754455>

**Public Dataset Links for detection**

1. <https://www.kaggle.com/datasets/jessicali9530/lfw-dataset>
2. <https://www.kaggle.com/datasets/hearfool/vggface2>

"For recognition, we will use 3D pictures of our friends to test the system with a real-time interface."