<FaceAdmin>

<Nuraly Baktygaliyev>

<22066772>

UXCFXK-30-3

Digital Systems Project



# Abstract

**Your Abstract**

# Acknowledgements

# Table of Contents

[Abstract 1](#_Toc188486645)

[Acknowledgements 2](#_Toc188486646)

[Table of Contents 3](#_Toc188486647)

[Table of Figures 4](#_Toc188486648)

[Introduction 5](#_Toc188486649)

[Literature Review 6](#_Toc188486650)

[Overview of Facial Recognition Systems 6](#_Toc188486651)

[Machine Learning and Deep Learning for Face Recognition 7](#_Toc188486652)

[Overview of Web Development Frameworks 8](#_Toc188486653)

[Related Work and Existing Solutions 9](#_Toc188486654)

[Requirements 10](#_Toc188486655)

[Methodology 10](#_Toc188486656)

[Design 10](#_Toc188486657)

[Implementation 10](#_Toc188486658)

[Project Evaluation 10](#_Toc188486659)

[Further Work and Conclusions 11](#_Toc188486660)

[Glossary 11](#_Toc188486661)

[Table of Abbreviations 12](#_Toc188486662)

[References / Bibliography 13](#_Toc188486663)

[Appendix A: First Appendix 15](#_Toc188486664)

# Table of Figures

# Introduction

Traditional entry methods using keycards and PIN codes are very prone to problems like unauthorized sharing, loss of tokens, and human error in log maintenance. These signal an urgent need for a more robust and reliable solution that reduces administrative overhead while enhancing workplace security. Recently, it has emerged as a viable alternative to address such issues: facial recognition technology, which identifies individuals by their unique biometric features. A well-designed face recognition system will speed up the verification processes and eliminate the use of physical tokens, hence smoothing operations and reducing unauthorized access attempts significantly.

The project in the report covers a face recognition integrated system for use in offices. Its focus is to automate and simplify the management of employee attendance and visitors' access, powered by the capabilities of deep learning models and web technologies. The goal is to develop not just a functional prototype that would be able to detect and verify people's identities reliably but also an easy-to-use administrative dashboard that will enable the office managers and other authorized personnel to stay in control of the system. This includes real-time monitoring, data analytics, and role-based access to the platform, ensuring sensitive information is protected from unauthorized manipulation.

A review of related literature on the face recognition system shows the immense accuracy the deep learning methodologies, especially using Convolutional Neural Networks, have gotten while identifying minute facial features. Similarly, training and deploying such models by using TensorFlow and Keras, and efficient preprocessing and capture using libraries like OpenCV, remain paramount. This is where powerful tools are implemented within a system that has architectural consistency and scalability. Attention must also be given to the security aspect and user experience. Besides, research in the area shows potential issues of bias, privacy, and regulatory compliance; any such successful system would have to balance strong technological underpinnings with steadfast ethical boundaries.

A hybrid approach to development was followed for handling the multidimensional nature of the problem. This puts a lot of emphasis on iterative and incremental improvements so that adjustments in model performance or user interface design can be rapidly integrated. The system backend is built using Django and Django REST Framework, providing a secure set of API endpoints for face recognition requests, user administration, and session management. The admin dashboard and office employee portal are based on React, with Vite employed to work on build performance. Containerization with Docker is used to ensure that the whole solution-from the database layer into the face recognition engine-can be deployed consistently in different environments, such as local servers or the cloud. Additionally, PostgreSQL was chosen as the database layer because to its dependability and smooth integration with Django's ORM, which provides comprehensive data handling for user profiles, attendance logs, and facial embeddings. By partitioning the system into modular containers, scalability and maintenance become much easier, allowing updates and monitoring to be easily achieved.

It is expected that this project will result in a working prototype capable of recognising authorised employees in real time, recording attendance data and providing the administrator with a user-friendly web-based dashboard to control this data. This development is focused solely on the office environment, although given its modular design, it can easily be adapted to other environments should this be required in the future. Ultimately, the project is intended to illustrate how new biometric technologies can be utilised to improve security and administrative efficiency. It is equally important to demonstrate a design philosophy based on ethical responsibility and compliance with data protection regulations.

This report is structured according to a sequence starting with the problem statement and ending with a detailed description of the implementation steps, testing procedures and evaluation of the results. Chapter two provides a critical literature review, analyses various existing face recognition systems, discusses the basic principles and identifies the gaps that this project seeks to fill. Chapter three describes the functional and non-functional system requirements, assumptions and limitations. Chapter four details the methodology, project management strategies, and specific tools used for data collection and processing. Chapter five discusses the architecture of the overall system, the design decision made, which shows how the various components interact to create a holistic solution. Chapter six describes the technical aspect of the implementation, including the organisation of the code base, model integration and deployment methods. Chapter seven describes testing and evaluation procedures to validate the functionality, performance, and security of the system. Chapter eight critically assesses the extent to which the project achieved its objectives, highlighting the problems found and comparing the proposed solution with existing options. Finally, chapter nine of the report summarises the results of the work on the system and identifies areas for further work in this area.

# Literature Review

### Overview of Facial Recognition Systems

Face recognition has undergone many changes over the past few decades, ranging from manual feature extraction methods to sophisticated models based on deep learning. Early research in this area often used approaches that attempted to encode facial features through geometric distances between key points such as eyes, nose, and mouth. These methods often proved vulnerable to variations in illumination, pose, and facial expression (Zhao et al., 2003). As computational power increased and more robust algorithms emerged, researchers turned their attention to statistical models like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). These techniques enabled systems to reduce high-dimensional facial data into feature-rich subspaces, thereby improving recognition accuracy under controlled conditions.

Building on these foundations, the 1990s and early 2000s saw face recognition mature into a field of considerable practical interest for security applications in airports, government facilities, and corporate offices (Jain and Li, 2005). During this period, better imaging hardware and the rapid rise of internet connectivity allowed the discipline to extend beyond constrained laboratory settings. Researchers explored more sophisticated modeling techniques to cope with real-world challenges, including partial occlusion and low-quality images. Despite the achievements of feature-based and holistic approaches, face recognition models of this era still encountered difficulties with large-scale databases and complex lighting environments, prompting deeper investigation into methods capable of learning invariant face descriptors.

By the mid-2010s, the advent of deep learning profoundly reshaped the landscape of face recognition (Zafeiriou et al., 2015). CNNs have demonstrated an outstanding ability to capture the smallest details in faces, raising the performance bar to new heights in standard benchmark tests. These data-driven models learn hierarchical feature representations from large training sets that are invariant to rotation, occlusion, and changes in lighting conditions. More importantly, the computational gain due to GPUs accelerates training and inference, enabling real-time or near real-time use in security, law enforcement, and commercial applications.

Another ethical aspect is the possibility of biassed performance among different demographic groups. According to research, face recognition algorithms trained on datasets with low diversity can produce disproportionately high error rates for people of under-represented ethnicities or ages. As a result, many developers are currently working to create balanced training corpora and frequently assess model findings, breaking them down by demographic categories to discover and fix potential biases. Furthermore, compliance with regulatory frameworks such as the General Data Protection Regulation (GDPR) is becoming increasingly important for organisations looking to adopt real-world face recognition (Issaoui et al., 2023). Modern design methods typically contain rigorous data governance standards, emphasising transparent data collecting and the protection of user consent throughout the system's lifecycle.

Taken together, the evolution from manual feature extraction to deep learning-based solutions highlights the dynamic nature of this field and its resilience in solving complex visual problems. Data privacy, model training and potential bias, and regulatory compliance are now as much a part of system design as algorithm accuracy and computational efficiency. Therefore, many of today's facial recognition works must make a trade-off between exploiting deep learning improvements and meeting standards that protect human rights and maintain public trust.

### Machine Learning and Deep Learning for Face Recognition

The field of face recognition has undergone a profound transformation as machine learning (ML) methods have become increasingly sophisticated. Early ML-based systems commonly employed traditional classifiers such as Support Vector Machines (SVMs) or Bayesian models, often in conjunction with dimension reduction techniques like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA). Although these approaches laid the groundwork for computational face recognition, they typically relied on handcrafted features, which could be sensitive to variations in pose, illumination, and facial expression. As a result, these methods exhibited limited adaptability when applied to complex, real-world scenarios.

A major leap forward occurred with the rise of deep learning, a subfield of ML that focuses on learning hierarchical feature representations directly from raw data (LeCun et al., 2015). By training deep networks—particularly Convolutional Neural Networks (CNNs)—on large-scale image datasets, researchers discovered that the layers of these networks naturally learned robust, discriminatory features relevant to face recognition. This paradigm shift allowed systems to capture subtle variations in facial attributes far more effectively than was possible with conventional ML techniques (Goodfellow et al., 2016). Not only do deep architectures obviate the need for handcrafted features, but they also enable end-to-end training, in which the model learns how to perform feature extraction and classification simultaneously.

On the software side, deep learning frameworks like TensorFlow and Keras have greatly facilitated the development and deployment of CNN-based face recognition solutions. TensorFlow, created by Google, provides a highly scalable platform for distributed computing and can handle large datasets efficiently (Abadi et al., 2016). Keras, a high-level neural network API, streamlines model building by offering intuitive interfaces for layer configuration and training loops (Chollet, 2015). This ecosystem of tools has not only shortened development cycles but also lowered the barrier to entry for researchers and practitioners aiming to implement deep learning techniques in face recognition tasks.

Collectively, the convergence of powerful deep learning architectures, accessible software frameworks, and affordable high-performance hardware (e.g., GPUs) has revolutionized face recognition. Systems can now operate effectively in diverse and challenging settings, detecting and matching faces with remarkable speed and accuracy. Nevertheless, deep learning’s reliance on substantial training data raises concerns related to privacy, bias, and regulatory compliance. Consequently, modern ML-based face recognition projects often adopt stringent data governance policies and explore methods to mitigate algorithmic bias, showcasing the continual interplay between technological innovation and ethical responsibility.

Beyond the works cited here, recent initiatives in transfer learning, few-shot learning, and self-supervised learning also warrant attention for future facial recognition research. These approaches enable more efficient use of limited data, potentially benefiting organizations that operate under strict data sharing regulations or possess highly specialized datasets. As the field progresses, striking a balance between model complexity, training resources, and ethical considerations remains an ongoing challenge for both researchers and industry practitioners.

### Overview of Web Development Frameworks

Web development frameworks play a significant role in today's software projects by providing already-implemented components, standardized structures, and accompanying tools for fast and maintainable application development. On the server side, Django stands out as a popular Python-based framework that emphasizes a clean and pragmatic design philosophy (Holovaty et al., 2009). The appeal of Django is a set of features including an object-relational mapper, built-in authentication, and a templating engine that allows the developer to focus on the business logic rather than low-level tasks. Django REST Framework further simplifies the creation of APIs by providing out-of-the-box functionalities such as serialization, view sets, and a browsable API interface for projects that require RESTful architectures (Christie, 2011). This makes it particularly well-suited for applications that involve biometric or image-processing logic, since data structures and endpoints can be standardized quickly.

On the client-side, React has been widely used for making interactive user interfaces. Developed by Facebook, it encourages developers to break down interfaces into reusable components, making the codebase more maintainable and scalable (Krill, 2014). Its immediate advantage, which makes React widely used in many leading web applications, is its use of a virtual DOM to efficiently track changes to the user interface. This approach offers significant performance benefits, especially in scenarios where client-side updates must occur in near real-time, such as systems that track facial recognition events in an office environment. Beyond React, additional libraries have been created - React Router for client-side routing - that make it easy to create complex single-page applications with little overhead. (Elrom, 2021).

Combining these features with Django's powerful backend functionality and React's componentised frontend approach allows developers to achieve significant separation of concerns. This is particularly advantageous in facial recognition projects where huge amounts of image data need to be processed securely and asynchronously. Django, for example, has built-in authentication and user permissions mechanisms that provide another layer of security needed to process biometric data. At the same time, React allows for faster prototyping of new dashboard features such as real-time attendance widgets, image verification, and others.

PostgreSQL also plays an important part in such an ecosystem by providing a strong relational database solution that works well with Django's Object-Relational Mapping (ORM). PostgreSQL, known for its extensive features, enables complex queries, custom indexing options, and strong ACID (Atomicity, Consistency, Isolation, Durability) compliance (Worsley et al., 2002). In a database performance comparison research, PostgreSQL was found to maintain excellent levels of stability and efficiency under virtualisation and heavy data loads, consistently beating competitors like as MySQL and Microsoft SQL Server (Truskowski et al., 2020). These benefits make PostgreSQL ideal for applications involving biometric data storage and real-time attendance logs, where dependability, scalability, and data integrity are critical.

In practice, these frameworks are combined with containerisation technologies such as Docker to further simplify deployment. When each service, say Django API and React frontend, is packaged in its own container, updates can be applied independently of each other without disrupting the entire system. In this way, the combination of Django and React provides a flexible, efficient and secure ecosystem for implementing a facial recognition platform that can be adapted to changing business requirements and compliance standards.

### Related Work and Existing Solutions

Advances in machine learning, deep learning, and computer vision have resulted in broad adoption of facial recognition systems across a wide range of applications. The following sections address two important works that inform the creation of automated systems for controlled environments, such as offices and educational institutions.

Ejaz et al. (2023) created a facial recognition-based access system to improve security in student residential halls. Their approach detects faces using a Multi-Task Cascaded Convolutional Neural Network (MTCNN) and recognises them using the FaceNet model. This system displays a strong design by training on a dataset of 3000 photos and obtaining good accuracy in real-time facial recognition. This system demonstrates a robust design by training on a dataset of 3000 images and achieving high accuracy in real-time face recognition. MTCNN handles variations in pose and lighting, while FaceNet’s 128-dimensional embeddings ensure efficient and accurate identification. The study emphasises the value of incorporating deep learning techniques for improved security and automated monitoring, such as logging access and exit timings in a CSV file. However, scalability remains a difficulty, especially in environments with thousands of people.

Another relevant work by Kar et al. (2012) describes an automated attendance system utilizing Principal Component Analysis (PCA) for facial recognition. This system uses cameras to record images, which are then processed into grayscale before being used to generate eigenfaces using PCA. While this technique works well for small-scale datasets, it has difficulties when dealing with variations in stance and lighting. The system functions in a classroom environment, automatically collecting attendance and keeping logs of clock-in and clock-out times. However, the use of PCA limits its resilience in real-world applications when compared to newer deep learning-based systems. Regardless of these restrictions, the study shows how biometric technologies can eliminate manual errors and streamline operations.

Both systems underscore the evolution of facial recognition technology. Ejaz et al. demonstrate the potential of contemporary deep learning models like FaceNet to overcome traditional challenges, such as scalability and environmental variability. In contrast, Kar et al.’s PCA-based solution highlights the earlier stages of automated facial recognition development and the foundational concepts upon which modern systems are built. Together, these works emphasize the importance of selecting appropriate algorithms and system architectures based on the application domain and operational requirements.

Commercial platforms, such as Amazon Rekognition and Microsoft Azure Face API, demonstrate how large-scale face recognition solutions can be used in a variety of applications. Although they tend to attain high accuracy via vast proprietary training datasets, their performance and security characteristics might differ greatly based on network reliability, dataset diversity, and recurring subscription fees. Open-source frameworks, such as InsightFace, provide for greater customisation and transparency, but require more in-house experience to maintain, especially when scaling to thousands of users (Deng et al. 2019). These trade-offs highlight the significance of weighing both technological performance and business restrictions before committing to a single approach, particularly in situations where data privacy and regulatory compliance are critical.

By addressing research gaps such as scalability, real-time performance, and adaptability to changing conditions, the present project aims to combine the characteristics of previous systems into a scalable solution for office contexts. Using deep learning technology and modular web development frameworks, the suggested system aims to deliver smooth, secure, and efficient facial recognition capabilities.

# Requirements

# Methodology

# Design

# Implementation

# Project Evaluation

# Further Work and Conclusions

# Glossary

# Table of Abbreviations

# References / Bibliography

1. Zhao, W., Chellappa, R., Phillips, P.J. and Rosenfeld, A. (2003) Face recognition. ACM Computing Surveys [online]. 35 (4), pp. 399–458. [Accessed 23 December 2024].
2. Li, S.Z. and Jain, A.K. (2011) Handbook of Face Recognition [online]. London, Springer London. Available from: https://link.springer.com/book/10.1007%2F978-0-85729-932-1 [Accessed 22 December 2024].
3. Zafeiriou, S., Zhang, C. and Zhang, Z. (2015) A survey on face detection in the wild: Past, present and future. Computer Vision and Image Understanding [online]. 138, pp. 1–24. [Accessed 24 December 2024].
4. Issaoui, A., Örtensjö, J. and Islam, M.S. (2023) Exploring the General Data Protection Regulation (GDPR) compliance in cloud services: insights from Swedish public organizations on privacy compliance. ProQuest [online]. p. 107. Available from: https://www.proquest.com/docview/2902172913/D88FB90847A44045PQ [Accessed 26 December January 2025].
5. LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. Nature [online]. 521 (7553), pp. 436–444. Available from: https://www.nature.com/articles/nature14539 [Accessed 27 December 2024].
6. Goodfellow, I. (2016) Convolutional Networks-12 Adapted by m.n. for CMPS 392 [online]. Available from: https://mnassar.github.io/deeplearninghandbook/slides/09\_conv.pdf [Accessed 27 December 2024].
7. Abadi, M. et al. (2016) TensorFlow: A System for Large-Scale Machine [online]. Available from: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf [Accessed 30 December 2024].
8. Chollet, F. (2021) Deep Learning with Python, Second Edition Google Books [online]. Manning Publications Co., pp. 70–84. Available from: https://books.google.kz/books?hl=ru&lr=&id=XHpKEAAAQBAJ&oi=fnd&pg=PA1&dq=Chollet [Accessed 2 January 2025].
9. Holovaty, A. and Kaplan-Moss, J. (2009) The definitive guide to Django web development done right; [Django is a framework that saves you time and makes Web development a joy ; updated for Django 1.1]. Berkeley, Calif. Apress. [Accessed 3 January 2025]
10. Christie, T. (2011) Django REST framework documentation Django-rest-framework.org. 2011 [online]. Available from: https://www.django-rest-framework.org/ [Accessed 3 December 2025].
11. Krill, P. (2014) React: Making faster, smoother UIs for data-driven Web apps InfoWorld. 15 May 2014 [online]. Available from: https://www.infoworld.com/article/2179365/react-making-faster-smoother-uis-for-data-driven-web-apps.html [Accessed 3 January 2025].
12. Elad Elrom (2021) React Router and Material-UI. React and Libraries, Your Complete Guide to the React Ecosystem [online]. pp. 79–113. Available from: https://www.researchgate.net/publication/349815388\_React\_Router\_and\_Material-UI [Accessed 8 June 2023].
13. Md. Sabbir Ejaz, Debnath, S., Mohammad Kamrul Hasan and Md. Mahbubul Alam (2023) Facial Recognition-Based Entry System for Student Residence Halls: Enhancing Security and Accessibility. Asian Journal of Research in Computer Science [online]. 16 (4), pp. 344–353. Available from: http://archive.pcbmb.org/id/eprint/1766/1/Ejaz1642023AJRCOS110323.pdf [Accessed 8 January 2025].
14. Kar, N., Debbarma, M.K., Saha, A. and Pal, D.R. (2012) Study of Implementing Automated Attendance System Using Face Recognition Technique. International Journal of Computer and Communication Engineering [online]. 1 (2), pp. 100–103. Available from: http://www.ijcce.org/papers/28-N010.pdf [Accessed 8 January 2025].
15. Deng, J., Guo, J., Xue, N. and Zafeiriou, S. (2019) ArcFace: Additive Angular Margin Loss for Deep Face Recognition openaccess.thecvf.com. 2019 [online]. pp. 4690–4699. Available from: https://openaccess.thecvf.com/content\_CVPR\_2019/html/Deng\_ArcFace\_Additive\_Angular\_Margin\_Loss\_for\_Deep\_Face\_Recognition\_CVPR\_2019\_paper.html [Accessed 12 January 2025].
16. Douglas, K. and Douglas, S. (2003) PostgreSQL: a comprehensive guide to building, programming, and administering PostgreSQL databases [online]. Indianapolis, Ind., Sams Publ, pp. 1–207. Available from: https://books.google.kz/books?hl=ru&lr=&id=gkQVL9pyFVYC&oi=fnd&pg=PA1&dq=postgresql+advantages&ots=EaueVYHwlV&sig=sfczDpCJPGjSXveou4lfrob7QT0&redir\_esc=y#v=onepage&q=postgresql%20advantages&f=false [Accessed 12 January 2025].
17. Truskowski, W., Klewek, R. and Skublewska-Paszkowska, M. (2020) Comparison of MySQL, MSSQL, PostgreSQL, Oracle databases performance, including virtualization. Journal of Computer Sciences Institute [online]. 16, pp. 279–284. Available from: https://ph.pollub.pl/index.php/jcsi/article/view/2026/1981 [Accessed 12 January 2025].

# Appendix A: First Appendix