Learning Experience on Intelligent Security Face Recognition System Based on Edge Computing

Yangjing Cheng

School of Computer Science and Engineering, Xi'an Technological University, Xi'an, Shaanxi, China

1431807602@qq.com

Abstract. To address the latency, privacy, and performance issues of traditional centralized face recognition systems, edge computing offers an effective solution. This paper details my comprehensive involvement in developing an edge computing-based intelligent security face recognition system, a key project of the National College Students' Innovation and Entrepreneurship Training Program. Throughout the project lifecycle, I independently optimized an ECA-CNN-ViT hybrid model, constructed a three-layer cloud-edge-terminal architecture, and verified system performance. Beyond detailing the technical implementation and the resolution of hardware/software conflicts, this paper summarizes the practical engineering experience gained. It also reflects on the broader impacts of AI on computer science education and career planning, identifying personal areas for improvement to lay a solid foundation for future academic and professional development.

Keywords: Face Recognition; Deep Learning; Edge Computing; Model Lightweight; Learning Experience

1. Introduction

1.1 Learning Motivation and Project Background

As an undergraduate majoring in Internet of Things Engineering, I have learned core courses including Python Programming, Fundamentals of Machine Learning, Computer Networks, Database Principles and Embedded Development, and mastered basic programming syntax and theoretical knowledge. However, my professional learning was limited to memorizing textbook theories, writing simple codes and finishing after-class exercises for a long time. I had never participated in a complete engineering project covering demand analysis, scheme design, algorithm development, system development and deployment.

In class, I only knew that face recognition realized identity authentication through facial feature extraction and matching, but had no idea about the underlying network model principles, training and parameter tuning logic, hierarchical system architecture and edge-end deployment. I could only understand terms such as MobileFaceNet, ECA attention mechanism, CNN-ViT hybrid architecture, model pruning and quantization, and cloud-edge-terminal collaboration literally, without in-depth comprehension of underlying logic or practical operation capabilities. My engineering thinking and practical experience were severely insufficient.

Meanwhile, the intelligent security industry is developing rapidly. Face recognition monitoring and access control devices have been widely used in residential communities, campus gates, stations, airports and office buildings. The integrated solution of edge computing and face recognition has become the mainstream application mode in the industry, and it is also an essential core skill for students majoring in computer science and Internet of Things. For this reason, I selected Development of Intelligent Security Face Recognition System Integrated with Edge Computing as the project topic for the innovation and entrepreneurship training program.

On the one hand, I aimed to systematically master practical technologies such as face recognition algorithms, edge computing architecture, model lightweight, database design and interface development through immersive project practice to make up for technical deficiencies. On the other hand, all computing tasks of traditional centralized face recognition systems are undertaken by cloud servers. Massive video and image data transmission easily leads to network congestion, and the cross-network transmission of facial privacy data brings great security risks. Such systems are highly dependent on networks and cloud servers and will break down once the network or server fails. This

https://doi.org/10.58244/joerai.263894

project explores how edge computing sinks core computing power to terminal devices to fundamentally solve the problems of high latency, high cost, poor security and low reliability, and accumulates experience in AI vision project development to enhance professional competitiveness.

1.2 Initial Foundation and Deficiencies

Before starting the project, I had obvious deficiencies in professional knowledge and practical abilities, which made early learning difficult. The specific problems are summarized as follows.

First, the theoretical foundation of deep learning was weak. I had a vague understanding of core principles including Convolutional Neural Network (CNN), Vision Transformer, attention mechanism, feature extraction and fusion, and model lightweight. I failed to comprehend the network structure of MobileFaceNet, the adaptive logic of feature weights of ECA module, and the underlying mechanism of CNN and ViT integration, and could not quickly extract key information from professional papers and technical documents.

Second, I was unfamiliar with development tools and environment operation. I only mastered basic Python syntax, and had no experience in configuring Anaconda virtual environments, using PyTorch deep learning framework, processing images with OpenCV, developing interfaces with PyQt, and realizing linkage between SQLite and MySQL databases. Environment dependency conflicts and incompatible library versions became the biggest obstacles in the early stage.

Third, I lacked systematic engineering thinking. I used to learn fragmented knowledge points and could not split project modules, arrange development processes or divide tasks reasonably. When encountering program errors, I could only copy codes blindly and was incapable of troubleshooting, step-by-step debugging and independent problem solving, resulting in low learning efficiency.

Fourth, I had no knowledge of hardware and deployment. I knew nothing about hardware parameters of edge gateways, data interaction logic between cloud, edge and terminal, local encrypted storage, RBAC permission management and AES-256 encryption protocol. I also had no idea how to adapt trained models to edge devices with low computing power.

Faced with numerous professional terms, complex network architectures and various development tools, I encountered great difficulties in the early stage. I established the learning principles of progressive theoretical learning, priority of tool training, combination of practice and theory, and decomposition of difficult problems, so as to build a knowledge system from scratch and promote project practice steadily.

1.3 Overall Learning Framework Planning

To avoid fragmented learning, I formulated seven systematic learning stages to form a closed-loop learning system throughout the project.

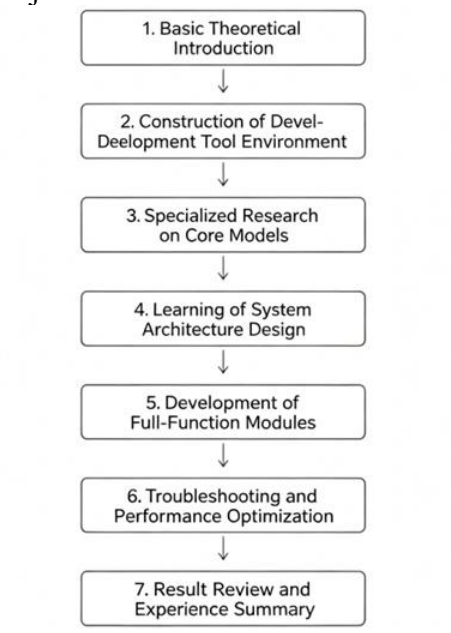

Figure 1. Flowchart of overall learning path

The whole learning process followed the principle of proceeding from the simple to the complex, prioritizing theories before practice and basic knowledge before development. Each stage had clear tasks and objectives. I consolidated theoretical foundations first, then got familiar with development tools, studied core models in depth, built the system architecture, completed function development, debugged and optimized the system, and finally summarized experience, getting rid of disorderly fragmented learning.

2. Review of the Systematic Learning Process

2.1 Basic Theoretical Learning Stage

Theories are the foundation of practical operation. In the initial stage, I spent plenty of time establishing a complete theoretical framework and sorting out the development history of face recognition technology. I learned classic algorithms of traditional machine learning such as Eigenface, Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG), as well as deep learning models including Deep Face and FaceNet. I also studied lightweight models for mobile and edge devices such as MobileFaceNet and ShuffleFaceNet.

By comparing the principles, advantages, disadvantages and applicable scenarios of different algorithms horizontally, I clarified the iteration logic of face recognition technology and realized the irreplaceable value of lightweight models for resource-limited devices. Meanwhile, I focused on three core theoretical modules: the working mechanism of convolution layers, pooling layers and fully connected layers of CNN for extracting local facial texture features; the self-attention mechanism of Vision Transformer for capturing global semantic features; and key technologies including ECA attention mechanism, structured pruning and 8-bit integer quantization to master the implementation logic of model lightweight.

In addition, by referring to literatures, I summarized five major problems of traditional centralized intelligent security systems: excessive network transmission latency, high risk of facial privacy data leakage, high operation and maintenance cost of cloud hardware, heavy reliance on network, and poor recognition robustness under complex illumination and occlusion conditions. I fully understood the research significance and practical application value of this project.

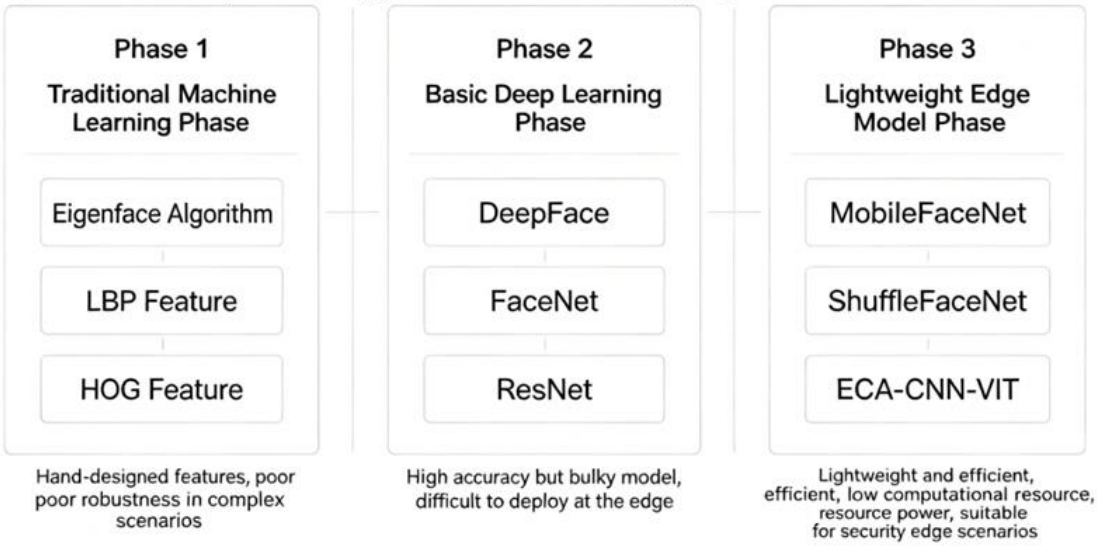

Figure 2. Evolution diagram of face recognition technology

2.2 Tool and Environment Construction Stage

After consolidating theoretical knowledge, I started to learn development tools and configure operating environments, which is a key transition from theories to practice.

I mastered the creation of Anaconda virtual environments, installation of dependent libraries, export and replication of environments to completely solve version conflicts of libraries among different projects. I got started with the PyTorch framework and learned the complete process of model loading,

network construction, training, parameter tuning, model saving and fine-tuning. I studied OpenCV for image preprocessing operations commonly used in security scenarios, including image grayscale processing, Gaussian denoising, face cropping and video stream parsing. I also learned PyQt interface development, including UI layout, control binding and signal-slot communication mechanism, laying a foundation for subsequent visual interface development.

Moreover, I distinguished the differences between lightweight SQLite database on edge terminals and MySQL database on cloud platforms, and mastered data table design as well as data addition, deletion, modification, query and synchronization logic. I learned engineering technologies such as SSH remote connection, edge gateway debugging, AES-256 encryption protocol and RBAC multi-role permission management, and built a complete tool ecosystem for project development.

2.3 Research on Core Models

The core model is the key part of this project. Instead of rote memorization, I deeply studied the ECA-CNN-ViT hybrid feature extraction architecture and lightweight schemes by disassembling network structures, analyzing experimental data and conducting practical verification.

Taking MobileFaceNet as the basic model, I analyzed its inverted residual structure and depthwise separable convolution design. This model can greatly reduce model parameters and computation while ensuring recognition accuracy, which makes it highly applicable as the basic model for edge face recognition.

Then I focused on two core optimization modules.

First, the ECA channel attention mechanism. Compared with the traditional CBAM attention mechanism, ECA removes redundant fully connected layers and realizes channel information interaction through one-dimensional convolution. It reduces computation volume, adaptively strengthens the weight of key facial features such as facial organs and contours, and suppresses background interference, which is more suitable for edge device deployment.

Second, the CNN-ViT hybrid architecture. CNN extracts local texture details of faces, while ViT with self-attention mechanism captures global semantic features. The complementary fusion of local and global features effectively improves recognition accuracy under complex illumination, posture deflection and face occlusion.

In terms of model lightweight, I adopted a combined scheme of structured pruning and mixed-precision quantization. I conducted channel-level redundant pruning within 30%, combined with 8-bit integer quantization and mixed-precision processing for key layers. On the premise of controlling the accuracy loss within 1%, the model parameters were reduced by 32.2%, floating-point operations decreased by 35%, and storage volume was cut by 75%. With the public LFW dataset and self-built security dataset, I completed the whole process of data augmentation, model training, ablation experiments and performance comparison.

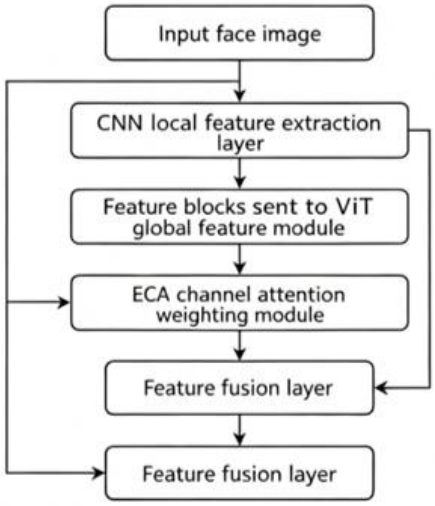

Figure 3. Structure diagram of ECA-CNN-VIT hybrid model

2.4 Learning of System Architecture

After mastering the core model, I learned the three-tier distributed cloud-edge-terminal collaborative architecture from a global engineering perspective, and sorted out the hardware configuration, core functions and data interaction logic of each layer.

The edge terminal layer is responsible for data collection. Equipped with 720P high-definition cameras and infrared fill lights, it only collects face images and video streams, and completes simple preprocessing such as grayscale processing, denoising and face cropping without undertaking complex computing tasks.

The edge node layer acts as the core computing hub of the system. Deployed on ARM-architecture edge gateways with NPU chips, it runs the lightweight ECA-CNN-ViT model to complete core tasks including facial feature extraction, identity matching, abnormal behavior detection and local encrypted storage. It realizes closed-loop local data processing and supports independent operation offline, getting rid of reliance on cloud platforms and networks.

The cloud management platform layer abandons real-time computing functions and focuses on global management services, including online monitoring of all devices, cloud backup of facial data, remote push of model versions, visual analysis of recognition data and multi-role permission management. It is only responsible for management and data backup, and does not participate in real-time face recognition inference.

I also learned the comprehensive security protection system. The AES-256 protocol is adopted for data transmission encryption, and local databases store data in encrypted form. The RBAC model realizes hierarchical management of multi-role permissions, achieving a 100% encryption rate for data transmission and storage to fully protect facial privacy data. I mastered the design fields and correlation of personnel information table, recognition log table, edge device table, user table and model version table, and understood the collaborative design idea of cloud-edge dual databases.

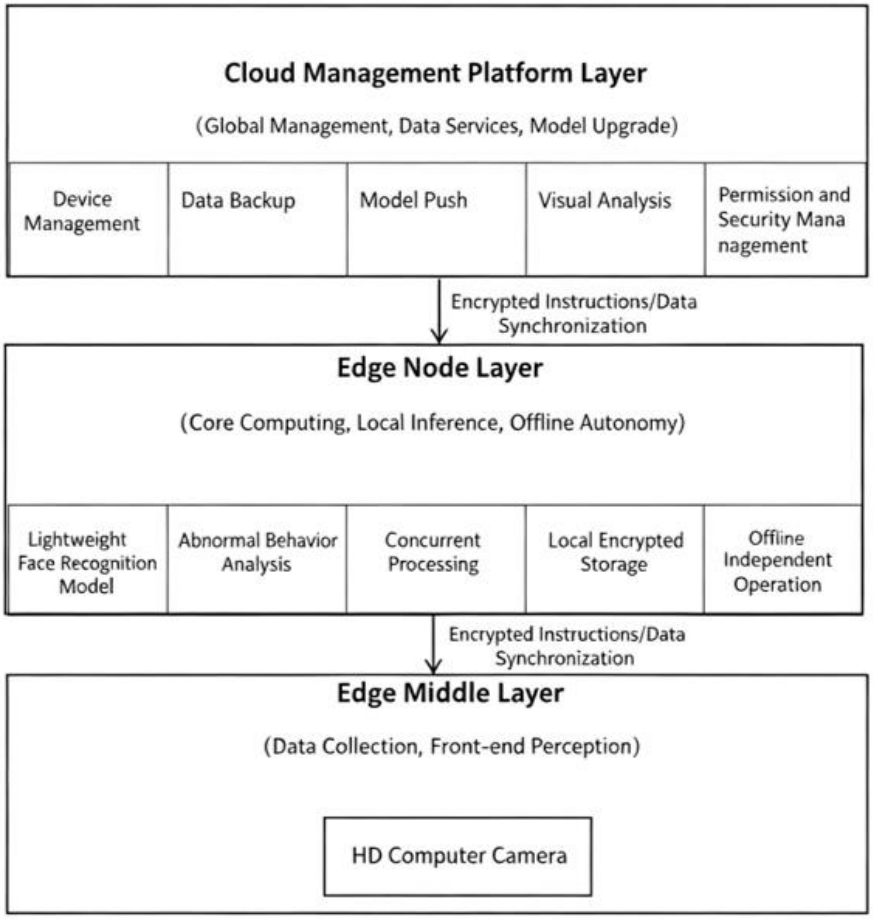

Figure 4. Overall diagram of three-tier distributed cloud-edge-terminal architecture

2.5 Development of Functional Modules

With solid mastery of theories, tools, models and architectures, I started the full-function development of the system. I adopted modular splitting and iterative development to complete the deployment of the entire intelligent security face recognition system, including three core modules.

First, face entry and management module. It supports two entry methods. Local image uploading and real-time camera capture, and realizes addition, editing and deletion of personnel information as well as maintenance of facial feature database.

Second, face recognition module. It provides three recognition modes. Static image recognition, real-time video stream recognition and online camera recognition. After deploying the lightweight model, the recognition latency is controlled within 42 milliseconds.

Third, recognition log query module. The system automatically records all recognition logs, supports multi-condition filtering and retrieval by time, personnel and recognition results, and logs can be exported as CSV reports.

During development, I sorted out business logic and wrote core codes independently, debugged and optimized problems such as interface layout, video frame rate and recognition matching, and accumulated practical engineering experience in the deployment process.

2.6 Problem Solving and Performance Optimization

In the process of system development and model testing, I optimized multiple links including environment compatibility, image processing, interface interaction, recognition accuracy, device deployment and concurrent processing.

Instead of copying codes blindly, I solved problems one by one by referring to official documents and academic papers, consulting supervisors and conducting step-by-step troubleshooting. Through ablation experiments and comparative experiments, I repeatedly adjusted model pruning ratio and quantization accuracy to balance recognition accuracy and inference speed. I also optimized the NMS threshold to reduce false detection rate of faces. Finally, all system indicators met the deployment standards of intelligent security scenarios. My ability of independent troubleshooting and engineering debugging was greatly improved in this process.

2.7 Summary and Review Stage

After completing the main tasks of the project, I reviewed all seven learning stages comprehensively. I sorted out core knowledge points, key technical difficulties, typical problems and corresponding solutions, and compiled reusable environment configuration guides, code snippets and model parameters, forming a standard learning paradigm for AI vision projects. I also reflected on my own deficiencies and accumulated reusable experience and methods for subsequent advanced learning and similar project development.

3. Typical Difficulties and Solutions in the Learning Process

The whole learning and development process was full of challenges ranging from theoretical understanding and tool operation to code programming and project promotion. I overcame various difficulties by consulting materials, conducting literature research, asking supervisors for advice and communicating in technical communities, and improved my abilities rapidly in problem solving.

3.1 Difficulties in Theoretical Understanding and Solutions

In the early stage, it was hard to understand the feature fusion logic of CNN and ViT, the core advantages of ECA attention mechanism compared with traditional CBAM, and how to balance accuracy and speed in model lightweight. I could only memorize concepts mechanically without in-depth comprehension.

To solve this problem, I read original research papers carefully and disassembled network structure diagrams. I visualized feature heat maps to observe the focusing effect of the ECA module on key facial areas intuitively. I designed multiple groups of comparative experiments to test the recognition accuracy and computation of the original model, single optimization module and fully optimized model, and verified the improvement effect of each module with real data. I adjusted pruning and quantization parameters repeatedly to find the optimal balance between accuracy loss and lightweight performance, and finally fully understood the underlying theoretical logic.

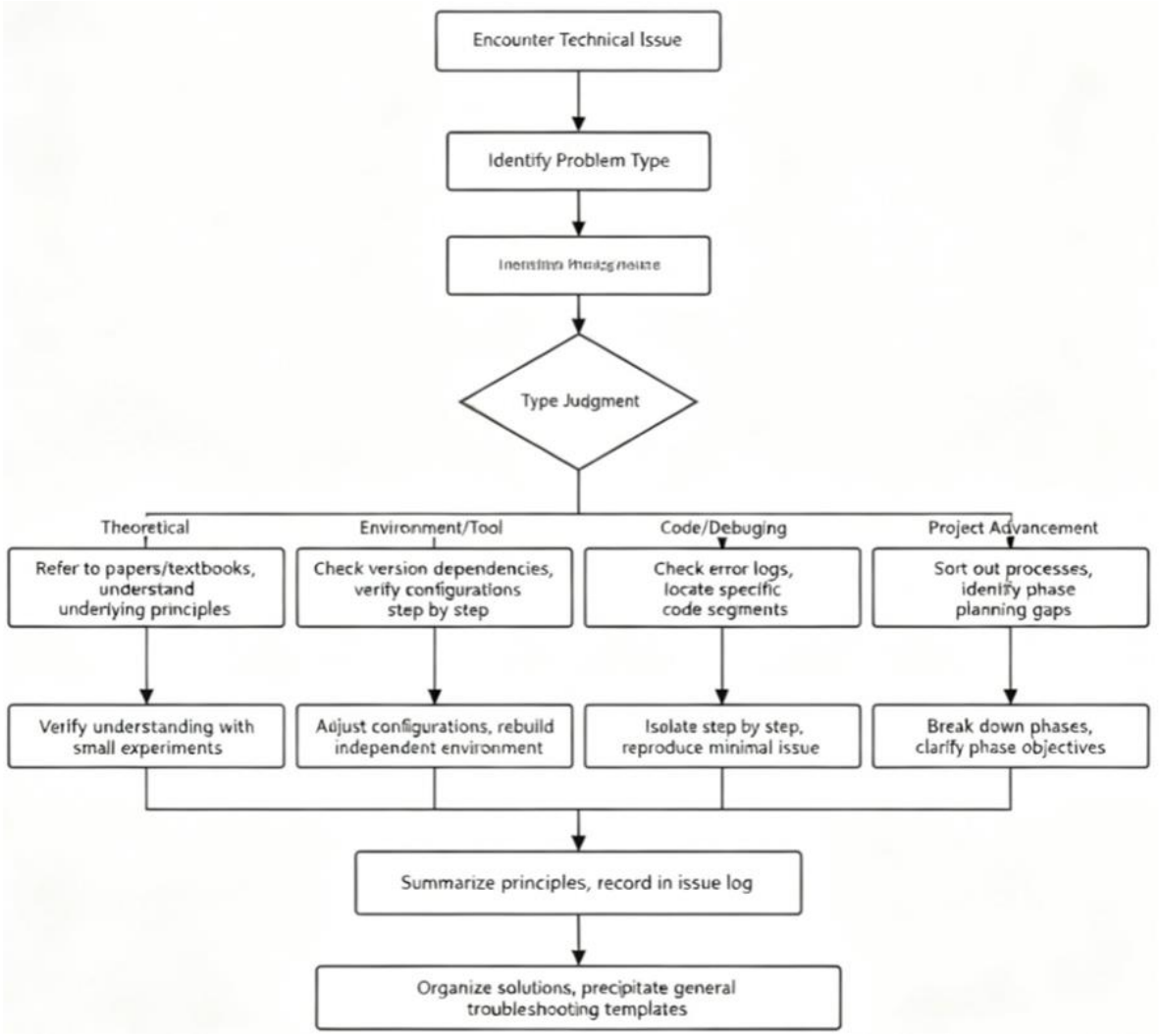

Figure 5. Problem-solving flowchart

3.2 Difficulties in Environment and Tools and Solutions

I frequently encountered problems such as Anaconda environment crash, version incompatibility between PyTorch and CUDA, failure of OpenCV to read files with Chinese paths, PyQt interface garbled characters and dependency library conflicts, which delayed the progress in the early stage.

I gave up copying online installation commands randomly and selected framework versions matching the graphics card computing power. I used the requirements.txt file to uniformly manage all project dependencies for one-click environment replication. I adopted binary reading mode to solve the problem of OpenCV reading files with Chinese paths, and adjusted font parameters to fix garbled characters in PyQt interfaces. I created an independent virtual environment for the project to avoid version conflicts fundamentally.

3.3 Difficulties in Code Development and Solutions

In the coding phase, I faced many problems including non-converged loss during model training, wrong dataset annotation formats, multi-thread conflicts in video recognition, picture stuttering, high false detection rate under occlusion, and failure of data synchronization between edge and cloud terminals. I was confused when facing error reports at first.

I standardized dataset annotation formats, adjusted learning rate and batch size, and added breakpoint resume training codes to solve the non-convergence problem of model training. I adopted QThread sub-threads to isolate video collection and recognition tasks, and realized thread communication via signal slots to avoid multi-thread conflicts. I optimized face quality screening and NMS threshold to reduce false detection rate effectively. I designed an incremental data synchronization strategy to realize regular synchronization between edge logs and cloud databases and ensure data consistency.

3.4 Difficulties in Project Promotion and Solutions

Due to the lack of project planning awareness in the early stage, I learned knowledge in a fragmented way and arranged development work disorderly, leading to repeated rework. I only focused on coding while ignoring architecture design and security protection, which required large-scale modification in the later stage.

I divided the development into fixed stages with clear tasks and deliverables for each stage. I followed the principle of completing architecture design before coding, and realizing core functions before optimization. I established a problem log to record error causes and solutions in a timely manner to avoid repeated mistakes, which greatly improved the project efficiency.

4. Multi-dimensional Learning Experience from Project Practice

4.1 Experience in Topic Selection

I chose the cross-field topic integrating edge computing and face recognition instead of traditional management systems or small program development, and gained more growth than basic projects.

First, we should dare to step out of the comfort zone in topic selection. The core competitiveness of computer majors lies in the ability to learn new knowledge rapidly and solve unknown problems, rather than being proficient in existing technologies. Although cross-field topics have high entry barriers and challenges in the early stage, they can drive us to expand knowledge boundaries and integrate multiple technical stacks.

Second, topic selection should combine open-source resources and practical application values. Face recognition for intelligent security is supported by open-source models and public datasets, which reduces the difficulty of entry. Meanwhile, it solves real pain points in industries such as communities, campuses and stations, so the project has practical deployment value rather than meaningless virtual development. For undergraduates, it is more appropriate to choose directions with open-source support and industrial practicality instead of blindly pursuing algorithm innovation.

Third, communication with teachers and peers is essential for proper topic positioning. I was unconfident about deep learning and edge computing at first. Through in-depth communication with supervisors and senior students, I clarified the technical route, avoided complex risks in algorithm research, and determined the practical logic of realizing functions first and then optimizing performance. It proves that active consultation and rational evaluation of personal foundation and project difficulty can help us avoid detours.

4.2 Experience in Technical Learning

First, theories and practice are inseparable. Learning theories without coding will only lead to armchair strategy, while coding blindly without understanding principles will result in mechanical replication and inability to troubleshoot errors. In this project, I verified every theoretical knowledge point through practical operation immediately after learning. For example, I modified model codes to conduct comparative experiments after learning attention mechanism, and implemented pruning and quantization after learning lightweight principles. The combination of theories and practice helps to grasp the essence of technologies thoroughly.

Second, systematic knowledge construction is superior to fragmented learning. Face recognition involves multiple fields such as deep learning, computer vision, edge computing, interface development, database and network security. Fragmented learning will lead to confused knowledge. I divided the whole knowledge system into basic layer, model layer, architecture layer, development layer and deployment layer according to learning stages, and supplemented knowledge points layer by layer to form systematic cognition, so that I can quickly locate relevant knowledge when encountering associated problems.

Third, engineering practice attaches importance to details and compatibility thinking. The difficulties of practical projects usually lie in details such as version compatibility, path coding, character display, memory occupation and thread scheduling, rather than core algorithms. Problems such as environment conflicts and program stuttering seem trivial, but they test comprehensive engineering literacy. Computer engineering is not only to realize basic functions, but also to ensure stability, compatibility, efficiency and user experience.

Fourth, problem-solving ability is more important than mastering specific technologies. The most valuable growth is to establish a standard problem-solving process: describing phenomena → locating causes → verifying schemes → summarizing experience. We should learn to refer to official documents, study open-source cases and troubleshoot step by step instead of escaping problems or relying on others. This independent problem-solving thinking is more valuable than a single technical skill.

4.3 Experience in Project Literacy and Thinking Development

First, establish the thinking of seeking optimal solutions under constraints. Project development is always restricted by budget, hardware, computing power and time. For instance, we need to balance configuration and cost when renting servers, balance accuracy and speed when optimizing models, and balance functions and complexity when developing systems. This experience helped me transform from pursuing perfect functions blindly to weighing various factors and finding optimal solutions under constraints, realizing the initial transition from student thinking to engineer thinking.

Second, develop the habits of iterative development and regular review. System development cannot be completed at one time. It should be improved module by module and version by version: realize core functions first, then optimize performance, interface and compatibility. I recorded and reviewed difficulties and solutions throughout the project, which consolidated my knowledge and provided reusable experience for similar projects in the future.

Third, deeply understand the value of integration between production and education. Classroom knowledge in universities is mainly theoretical, while technologies such as face recognition and edge computing have been widely applied in industries. This project combined programming, algorithm and database knowledge learned in class with actual industrial demands, making me realize that university learning should be closely connected with industrial development to cultivate engineering talents meeting enterprise requirements.

5. Reflection on Learning and Education under the Development of Artificial Intelligence

5.1 Impact of AI Development on College Students' Professional Learning

The rapid iteration of artificial intelligence, large models and deep learning has completely changed the learning mode, knowledge scope and ability requirements for computer majors, bringing new reflections on my own learning methods.

Firstly, the learning focus has shifted from knowledge memorization to application and innovation capabilities. In the past, professional learning emphasized memorizing grammar, formulas and knowledge points, while AI tools can quickly generate codes, deduce formulas and answer basic questions. Mechanical memorization is no longer a core advantage. Currently, higher-level abilities such as problem definition, scheme design, model optimization and engineering deployment are more valued. We should take AI as an auxiliary tool rather than rely on it lazily. In this project, I used AI to sort out knowledge frameworks, troubleshoot code errors and interpret papers, but independently completed core work including model debugging, architecture design and logical thinking.

Secondly, knowledge updates rapidly, and lifelong learning has become a necessary quality. Technologies of AI and computer vision iterate at a high speed. Knowledge learned from textbooks is easy to fall behind the industry. We must establish the awareness of lifelong learning, track cutting-edge technologies actively, read academic papers and participate in practical projects to break the limitations of textbooks.

Thirdly, interdisciplinary integration has become the core competitiveness. Face recognition integrates computer vision, deep learning, edge computing, network security and software engineering. AI technologies are also applied to security, medical treatment, finance, transportation and other fields. Single professional knowledge can no longer meet industrial demands. College students should break disciplinary barriers, learn related knowledge actively and cultivate interdisciplinary integration capabilities.

Fourthly, independent learning and questioning abilities are increasingly important. In the AI era, knowledge resources such as official documents, open-source communities, academic papers and

intelligent tools are easily accessible. We should learn to retrieve information accurately, read professional literatures, consult experts and use AI for auxiliary learning, which are essential abilities for contemporary college students.

5.2 Reflection on University Education Mode under the AI Trend

The development of artificial intelligence is forcing the transformation of computer major education in universities.

First, transform the teaching mode from theoretical indoctrination to project-driven learning. Traditional classroom teaching focuses on theoretical explanation and lacks practical scenarios, resulting in disconnection between theories and practice. Technologies such as face recognition can only be mastered through complete project practice. Universities should increase the proportion of practical courses, innovation projects and graduation projects, and let students learn and improve in real industrial projects to cultivate engineering capabilities.

Second, update course content timely to keep pace with cutting-edge technologies. Some textbook knowledge lags behind the rapid iteration of computer and AI technologies. Universities should optimize the curriculum system, add cutting-edge knowledge such as deep learning, computer vision and edge intelligence, and introduce open-source projects and real enterprise cases to narrow the gap between classroom learning and industrial development.

Third, focus on thinking cultivation rather than simple skill teaching. AI tools can replace basic coding and simple algorithm implementation, but cannot replace logical thinking, engineering thinking, innovative thinking and problem-solving abilities. University education should reduce mechanical skill training and focus on cultivating students' system design ability, logical reasoning ability, independent problem-solving ability and innovative thinking.

Fourth, strengthen the integration between universities and enterprises. Technologies such as intelligent security and face recognition have been widely deployed in enterprises. Universities should strengthen cooperation with technology companies, introduce enterprise projects, engineer lectures and internship opportunities to help students understand real engineering scenarios and realize seamless connection between campus learning and career demands.

5.3 Learning Orientation and Principles for College Students under the AI Trend

AI is a useful learning assistant, but it also brings risks such as lazy thinking and academic misconduct. Combined with this project experience, I believe college students should adhere to three principles and clarify their learning orientation.

First, use AI as an auxiliary tool instead of a substitute. We can use AI to sort out learning frameworks, explain difficult knowledge, troubleshoot codes and organize notes, but never copy codes, papers or schemes generated by AI and give up independent thinking. During this project, I independently designed core logic, wrote codes and solved problems, and regarded AI as an efficiency-improving tool while maintaining independent thinking ability.

Second, adhere to academic integrity and original learning aspiration. Behaviors such as AI writing, code plagiarism and simple project replication seem time-saving but will hinder personal ability improvement. Real growth comes from facing difficulties and solving problems independently. The experience of completing a complete face recognition system from scratch is irreplaceable.

Third, explore differentiated development and build core advantages. AI is good at data calculation, code generation and information retrieval, but it lacks human abilities such as logical thinking, scenario understanding, innovative design, ethical judgment and overall engineering planning. We do not need to compete with AI in memory or coding speed. Instead, we should focus on capabilities that AI cannot easily replace, such as project design, scheme optimization, solving complex problems and adapting to industrial scenarios, to build irreplaceable core competitiveness.

6. Personal Deficiencies and Future Plans

6.1 Deficiencies in Learning

Although I gained a lot from this project, there are still many deficiencies. First, the depth of model optimization is insufficient. I only completed basic lightweight and optimization for conventional scenarios, while the optimization for extreme illumination, large-area occlusion and low-resolution

faces is inadequate, and the missing detection of small-size targets has not been completely solved. Second, the scope of edge device adaptation is limited. The current model is only applicable to medium and high-end edge gateways, and more aggressive lightweight optimization for low-computing embedded devices such as Raspberry Pi is needed to improve deployment compatibility. Third, the self-built dataset is imperfect with limited scene diversity and sample size, leading to unsatisfactory model generalization ability. Fourth, the system functions need to be expanded. Only basic recognition and log functions are realized, while advanced functions such as multi-type abnormal behavior detection, batch image recognition and cloud visual analysis are not developed. Fifth, academic summarization ability is weak. I lack in-depth analysis and refinement of experimental data and ablation results, and need to improve the logical framework of paper writing and result summarization.

6.2 Follow-up Learning and Improvement Plans

Aiming at the above deficiencies, I formulated the follow-up improvement plans. Firstly, further optimize the model, focusing on small target detection, adaptive enhancement under complex illumination and feature completion for occluded faces to improve model robustness. Secondly, learn embedded deployment technologies, adapt models to low-computing devices such as Raspberry Pi, and explore more extreme lightweight schemes to expand deployment scenarios. Thirdly, expand the self-built face dataset with richer scenes and samples, and adopt knowledge distillation to improve the performance of lightweight models. Fourthly, improve system functions by adding multi-type abnormal behavior detection, batch recognition, data visual analysis and cross-screen adaptation to enhance practicality. Fifthly, strengthen the reading of academic literatures and writing training, sort out experimental data and write complete academic papers to improve scientific research and expression abilities. Sixthly, expand interdisciplinary learning of AI, Internet of Things and big data to accumulate richer technical reserves for competitions, graduation projects and future employment.

7. Summary and Overall Perception

This project practice on face recognition technology has been one of the most valuable independent exploration experiences during my university life. From having no knowledge of deep learning and edge computing at the beginning to building theoretical systems, mastering development tools, researching core models, solving technical problems and completing system development step by step, I have not only acquired professional knowledge and skills, but also comprehensively improved learning ability, engineering thinking, pressure resistance and problem-solving ability.

I realize that professional learning for computer majors is not passive knowledge reception limited to textbooks and classrooms, but active exploration beyond the comfort zone and growth driven by practical projects. Although face recognition is a cutting-edge technology, it follows the universal learning logic of consolidating theories first, preparing tools, decomposing problems step by step, solving difficulties and optimizing iteratively. This learning paradigm can be applied to the study of artificial intelligence, software development, embedded development and many other fields.

The arrival of the AI era brings both learning convenience and higher requirements for college students. We should embrace technological changes actively instead of fear, and avoid losing independent thinking due to over-reliance on AI. We need to clarify learning orientation, adhere to independent thinking and academic integrity, take practical projects as carriers, uphold the concept of lifelong learning, and cultivate irreplaceable engineering capabilities and innovative thinking.

In the future, I will take this project as a new starting point, continue to delve into computer vision and artificial intelligence, combine theories with practice, keep up with cutting-edge technologies, stick to the original aspiration of learning, and strive to grow into a comprehensive engineering talent meeting industrial demands.

Acknowledgements

This article is supported by "Xi'an Technology University's National-level College Students' Innovation and Entrepreneurship Training Program in 2025 (Project Number: 202510702031)".

References

[1] H. Wang, L. Li: Research on Application of Edge Computing in Intelligent Security, Computer Engineering and Applications, Vol. 59 (2023) No.12, p.1-8. (In Chinese)

[2] S. Zhang, L. Si: Research on Lightweight Face Recognition Algorithm Based on MobileFaceNet, Computer Applications, Vol. 42 (2022) No.S1, p.123-127. (In Chinese)

[3] M. Chen, Y. Zhao: Design of Intelligent Video Monitoring System under Cloud-Edge-Terminal Collaborative Architecture, Electronic Technology Application, Vol. 49 (2023) No.5, p.67-71. (In Chinese)

[4] J. Liu, F. Wu: Review on Model Lightweight Technologies for Deep Learning: Pruning, Quantization and Distillation, Computer Science, Vol. 49 (2022) No.8, p.56-65. (In Chinese)

[5] Q. Yang, W. Huang: Face Feature Optimization Method Based on ECA Attention Mechanism, Pattern Recognition and Artificial Intelligence, Vol. 36 (2023) No.3, p.245-252. (In Chinese)

[6] M. Zhou: Deployment Practice of Embedded Edge Gateway in Security Face Recognition, Internet of Things Technologies, Vol. 12 (2022), p.45-48. (In Chinese)

[7] Cyberspace Administration of China: Interpretation of Face Data Security Specifications under Personal Information Protection Law, (2021). (In Chinese)

[8] K. He, X. Zhang, S. Ren: Deep Residual Learning for Image Recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016).

[9] F. Schroff, D. Kalenichenko, J. Philbin: FaceNet: A Unified Embedding for Face Recognition and Clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015).

[10] C. Zhang, J. Li: Data Encryption and Privacy Protection Technology in Intelligent Security System, Information Network Security, (2023) No.4, p.78-85. (In Chinese)