Improvement of Face Recognition Algorithm for Residential Area Surveillance System Based on Graph Convolution Network

Tan Heyi[1], Byung-Won Min[2*]
[1]Ph.D. Student, Division of Information and Communication Convergence Engineering, Mokwon University
[2]Professor, Division of Information and Communication Convergence Engineering, Mokwon University

그래프 컨벌루션 네트워크 기반 주거지역 감시시스템의 얼굴인식 알고리즘 개선

담하의[1], 민병원[2*]
[1]목원대학교 정보통신융합공학부 박사과정, [2]목원대학교 정보통신융합공학부 교수

Abstract The construction of smart communities is a new method and important measure to ensure the security of residential areas. In order to solve the problem of low accuracy in face recognition caused by distorting facial features due to monitoring camera angles and other external factors, this paper proposes the following optimization strategies in designing a face recognition network: firstly, a global graph convolution module is designed to encode facial features as graph nodes, and a multi-scale feature enhancement residual module is designed to extract facial keypoint features in conjunction with the global graph convolution module. Secondly, after obtaining facial keypoints, they are constructed as a directed graph structure, and graph attention mechanisms are used to enhance the representation power of graph features. Finally, tensor computations are performed on the graph features of two faces, and the aggregated features are extracted and discriminated by a fully connected layer to determine whether the individuals' identities are the same. Through various experimental tests, the network designed in this paper achieves an AUC index of 85.65% for facial keypoint localization on the 300W public dataset and 88.92% on a self-built dataset. In terms of face recognition accuracy, the proposed network achieves an accuracy of 83.41% on the IBUG public dataset and 96.74% on a self-built dataset. Experimental results demonstrate that the network designed in this paper exhibits high detection and recognition accuracy for faces in surveillance videos.

Key Words : Face Recognition, Vision Transformer, Graph Convolution, Residential Area, Surveillance System

요 약 스마트 지역사회의 구축은 지역사회의 안전을 보장하는 새로운 방법이자 중요한 조치이다. 촬영 각도로 인한 얼굴 기형 및 기타 외부 요인의 영향으로 인한 신원 인식 정확도 문제를 해결하기 위해 이 논문에서는 네트워크 모델을 구축할 때 전체 그래프 컨벌루션 모델을 설계하고, 그래프 컨벌루션 모델에 협력하여 얼굴의 핵심을 추출한다. 또한 얼굴의 핵심을 특정 규칙에 따라 핵심 포인트를 구축하며 이미지 컨벌루션 구조를 구축한 후 이미지 컨벌루션 모델을 추가하여 이미지 특징의 핵심을 개선한다. 마지막으로 두 사람의 얼굴의 이미지 특징 텐서를 계산하고 전체 연결 레이어를 사용하여 집계된 특징을 추출하고 판별하여 인원의 신원이 동일한지 여부를 결정한다. 최종적으로 다양한 실험과 테스트를 거쳐 이 글에서 설계한 네트워크의 얼굴 핵심 포인트에 대한 위치 정확도 AUC 지표는 300W 오픈 소스 데이터 세트에서 85.65%에 도달했다. 자체 구축 데이터 세트에서 88.92% 증가했다. 얼굴 인식 정확도 측면에서 이 글에서 제안한 IBUG 오픈 소스 데이터 세트에서 네트워크의 인식 정확도는 83.41% 증가했으며 자체 구축 데이터 세트의 인식 정확도는 96.74% 증가했다. 실험 결과는 이 글에서 설계된 네트워크가 얼굴을 모니터링하는 데 더 높은 탐지 및 인식 정확도를 가지고 있음을 보여준다.

주제어 : 얼굴인식, 비전 트랜스포머, 그래프 컨벌루션, 주거지역, 감시시스템

*교신저자 : 민병원(minfam@mokwon.ac.kr)
접수일 2024년 02월 18일 수정일 2024년 03월 04일 심사완료일 2024년 03월 15일

1. Introduction

1.1 Background

The rise of the AI wave and the rapid development of the Internet of Things have promoted the ecological construction of smart communities. Due to the unique characteristics of the community's personnel composition and its functions, communities have a large population, and the accompanying social attributes make the identification of individuals entering and exiting the community more complex. Therefore, community security has been widely concerned as an important project. Currently, many community security measures still heavily rely on manual checks and patrols, supplemented by surveillance equipment for investigation. However, the overall efficiency of these methods is severely low and unstable. Therefore, this paper proposes a solution that combines deep learning technology with surveillance video systems to automatically identify personnel information in surveillance videos, thereby eliminating suspicious individuals in the community and improving the efficiency of community security personnel.

There have been related research cases both domestically and internationally regarding the application of deep learning technology in face recognition. For example, Zhang Jinjing, Liu Shuangfeng, and other researchers have designed a FaceNet face recognition algorithm that integrates the attention mechanism [1]. In terms of network design, they use U-Net as the feature extraction network and incorporate attention mechanism and feature pyramid structure to address the issue of low face recognition accuracy when the face is occluded. However, this network has high complexity due to the addition of multiple structures, which requires more computational resources and inference time during model inference. Therefore, it is not suitable for real-time detection tasks such as monitoring videos. Fanyihua, Wang Yongzhen, and other researchers have designed a contrastive learning-based non-paired low-light image enhancement network [2]. This network adopts U-Net as the underlying network structure and includes three sub-functional networks: feature preservation, semantic segmentation, and face recognition. During network training, contrastive learning paradigm is used to learn from low-light and normal-light images, enhancing the model's generalization ability. This method effectively improves the accuracy of face recognition under low-light conditions. However, this algorithm has limitations as it relies on a large amount of data for contrastive learning and requires facial images to be captured in a frontal view, which differs from the top-down view perspective of surveillance videos, limiting its applicability in certain scenarios. Feng Jing, Gu Meihua, and other researchers have designed an improved MobileNetV3-based face recognition algorithm [3]. This algorithm adopts global depthwise convolutional layers to capture more comprehensive facial information. It also utilizes a coordinated attention mechanism to enhance the spatial and channel information of the face, thereby improving the model's representation capability. The algorithm has been applied to face identity recognition tasks in classroom scenarios. Due to the adoption of the lightweight MobileNet network, this network has a significant advantage in recognition speed. However, the simplified network structure also weakens its detection accuracy, limiting its application in the field of security where high accuracy is required. On the other hand, academic teams abroad have also made relevant contributions in this field. F Boutros, N Damer, and other researchers have proposed an Embedded Unmasking Model (EUM) to address the issue of facial occlusion [4]. They also introduced a self-constrained triplet loss function, called Self-constrained Triplet (SRT) loss, which allows the network to learn the relationship between facial features of

the same identity under occluded and unoccluded conditions during model training. Although this method achieves high detection accuracy for occluded faces, it requires pre-matching of occluded and unoccluded faces in the dataset creation stage to learn the relationship between them. Additionally, it also requires capturing facial images in a frontal view, making it unsuitable for the scenario described in this paper. In terms of fast detection and recognition, Z Chen, J Chen, and other researchers proposed a lightweight face feature extraction network [5]. They employed a designed inverse residual recombination module to perform channel redundancy reduction and recombination operations on the feature maps, thereby reducing the parameter and computational complexity of the overall network. Although the reduction of feature map redundancy enables the network to achieve faster inference speed, it also results in the loss of effective features, leading to a significant impact on recognition accuracy.

In summary, although there are many research efforts in the field of facial recognition, there are currently no well-established solutions for applying it to community surveillance and security. There are two main issues that need to be addressed. Firstly, the recognition accuracy of the network model needs to be improved. The captured face images from surveillance devices may have some angle distortion, which greatly affects the recognition accuracy. Secondly, the parameters and computational complexity of the network model need to be controlled. Due to the large number of surveillance cameras in the community, all video data is processed on a single terminal, requiring the parallel operation of multiple models for comprehensive recognition. Therefore, the network needs to be designed to be lightweight. This paper will focus on these two aspects and propose a high-accuracy, overhead view facial recognition network with certain lightweight characteristics.

1.2 Technical Approach

The implementation of facial recognition technology for community surveillance and security can be divided into the following three parts:

[Fig. 1] Facial Recognition Workflow Diagram

Firstly, it is necessary to retrieve faces from the surveillance video frames and extract the pixel blocks corresponding to the face regions. Secondly, the detected faces need to be aligned by locating the facial landmarks such as contours and key points. Finally, based on the key point coordinates, the corresponding facial identities can be determined [6]. In the facial retrieval stage, this paper will construct a convolutional neural network to detect faces in the images. In the facial alignment stage, there are currently two types of methods: "key point regression" and "heatmap regression". The key point regression method involves extracting facial features using a convolutional neural network and mapping the feature maps to the key point coordinates through fully connected layers [7]. The heatmap regression method also starts with feature extraction using a convolutional neural network, and the detector outputs heatmaps with the same number of key points as the detected features, where each heatmap corresponds to a facial key point [8]. Among these two methods, key point regression has the advantage of faster detection speed [9]. Additionally, the influence of prior knowledge during model training results in higher detection accuracy even in occlusion scenarios [10]. On the other hand, the heatmap regression method detects individual key points separately [11], which leads to higher accuracy but also requires more computational resources and inference time [12]. Therefore, considering the task described in this paper, the key point

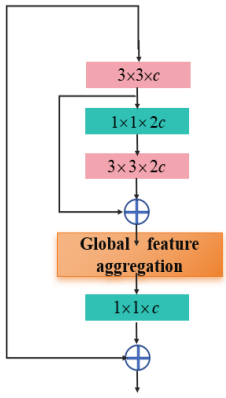
regression method is more suitable. In the facial identity recognition stage, this paper will construct a graph convolutional network. The facial key points obtained from the previous stage will be input into the graph convolutional network to extract the relationships between various key points. By constructing a graph structure for the key points using graph convolution, the distortion caused by the overhead view can be reduced, thereby improving the recognition accuracy.

In conclusion, this network model effectively addresses the issue of low recognition accuracy caused by facial distortion and can be integrated with community security monitoring to build a smart IoT system. By detecting and identifying faces in surveillance footage and integrating with relevant IoT devices, it enables automatic management of community entrances and exits, alerts for non-smoking areas within community buildings, 24/7 security management of community parking lots, localization and alarms for unauthorized entry into the community, thereby reducing the workload of security personnel and enhancing the protective capabilities of the community security system.

2. Building the Facial Recognition Network

2.1 Designing a Lightweight CNN for Face Detection

For the design of the face detection network , this paper adopts a fully convolutional network architecture. An inverse bottleneck residual module is constructed to stack the network, aiming to achieve a lightweight face detection network with a simplified network structure and low parameter count. The structure of the inverse bottleneck residual module and the corresponding parameters of its network layers are illustrated in the figure below:



[Fig. 2] Inverse Bottleneck Residual Module

As shown in the figure above, the inverse bottleneck residual module consists of four convolutional layers. In the parameter design of the convolutional layers, the first and fourth layers have c channels, while the second and third layers have $2c$ channels. This means that the module employs a reverse convolution calculation operation, with dimensionality expansion followed by dimensionality reduction, while ensuring that the input and output channels of the module are the same. In terms of the configuration of the convolutional kernel size, a 3x3 kernel is initially used for feature extraction. Then, a 1x1 kernel is used for channel adjustment. In the third layer, the kernel size is increased to 3x3. After the feature output of the third layer, a concatenation layer is used to aggregate the output of the first layer. The aggregated features are globally normalized and input into a 1x1 kernel in the fourth layer to restore the channel count to the same as the input. Finally, the input and output of the module are mapped and connected to construct the residual structure. In this module, global normalization response is used to enhance the channels that contain relevant features and suppress redundant channel features. In the calculation process, a global function is first used to compress the spatial dimensions of the feature map. Let the input feature map be x and the

compressed feature channel vector be gx. The parameter changes during this process are:

$$G(X) := X \in R^{H \times W \times C} \rightarrow gx \in R^{C} \quad (1)$$

For this compression operation, this paper employs the L2-norm normalization to extract each channel of the feature map.

$$G(X) = gx = (\| X_1 \|, \| X_2 \|, ..., \| X_c \|) \in R^{c} \quad (2)$$

Afterwards, division normalization is used to calculate the activation of the extracted channel features:

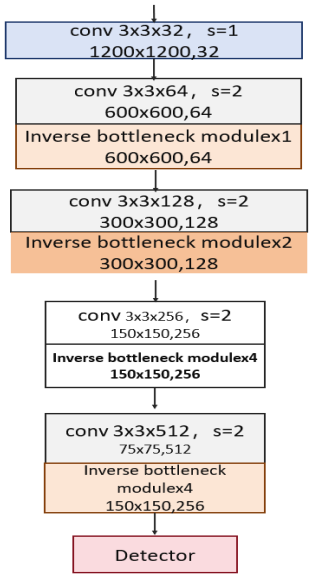
$$N(\| X_i \|) := \| X_i \| \in R \rightarrow \frac{\| X_i \|}{\sum_{j=1,2,...,c} \| X_j \|} \in R \quad (3)$$

In the above equation, $\| X_i \|$ represents the L2 norm calculation of the i-th channel. Finally, the calculated channel normalization coefficient is multiplied by the input to complete the normalization activation operation, expressed as:

$$X_i = X_i \times N(G(X)_i) \in R^{H \times W} \quad (4)$$

The inverse bottleneck residual module optimizes the computational complexity by optimizing the convolutional parameters and the configuration of the convolutional layers. It also enhances the importance of facial-related features by incorporating global normalization response to weight the activation of feature channels. Based on this module, the facial detection network structure and its corresponding network parameters constructed in this paper are illustrated in the figure below:

As shown in the above figure, this article resizes the input video image to a size of 1200x1200 pixels. In the input side, the deep depth map conversion is performed by the convolutional layer, followed by feature extraction by the inverse bottleneck residual module. For



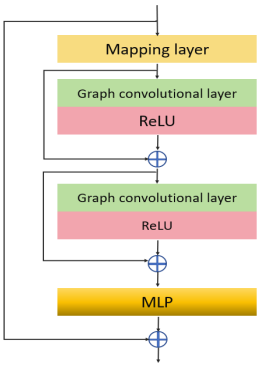
[Fig. 3] Facial Detection Network

downsampling the feature map and controlling the channel number, this article uses convolution with a sliding stride of 2 for downsampling. Since the pixel area of the face is small, the network performs only 4 downsampling operations overall to avoid loss of facial features. Additionally, due to consistent shooting height of the surveillance video and similar sizes of captured faces, only a single detector is used for face detection.

2.2 Design of Face Alignment Network based on CNN-ViT Fusion Network

After the face detection network detects the face panel and crops out the facial region, the face alignment network is used to locate the facial keypoints. Since the input images to the alignment network contain only the complete face and few background elements, we employ the graph convolution module to build the full-face feature relation [13]. Additionally, we use the convolutional module to extract fine-grained features in the image. The features extracted by these two types of modules are then combined to enhance the overall feature richness. Finally, the coordinates of each keypoint are

obtained using fully connected layers [14]. The construction and workflow of the graph convolution module are described as follows:



[Fig. 4] Global Graph Convolution Module

First, the input feature map $X \in R^{H \times W \times C}$ is divided into N feature blocks using a graph construction layer, and each feature block is transformed into a feature matrix $x_i \in R^D$. In this encoding layer, the feature map can be transformed into a feature matrix $X = [x_1, x_2, ..., x_N]$. At the same time, each feature matrix x_i can be regarded as a node v_i, so the feature matrix can be represented as an unordered set of points $V = \{v_1, v_2, ..., v_N\}$. For each node, this paper sets it to connect with the nearest four nodes and constructs edges e_{ij} between two nodes, resulting in the edge set E involving all nodes. Based on the nodes and edges, the graph structure $G = (V, E)$ of the feature matrix can be obtained. After the construction of the graph structure is completed, the graph convolution layer is used to extract global features. The extraction operation involves aggregating the features of neighboring nodes and updating them with the features of the current node. This process can be represented as:

$$g' = F(g, W) = Update(Aggregate(g, W_{agg}), W_{update}) \quad (5)$$

In the above equation, W_{agg} represents the weights for aggregating the features of two nodes, and W_{update} represents the weights for updating the parameters of the aggregated nodes. The update of each node's features involves aggregating the features of its adjacent nodes and updating the corresponding weights. The formula can be represented as:

$$x_i' = h(x_i, g(x_i, N(x_i), W_{agg}), W_{update}) \quad (6)$$

In the above equation, $N(x_i)$ represents the adjacent nodes of the current node, and the aggregation operation is performed using the Max Correlated Graph Convolution (MCGC) calculation, which can be represented as:

$$g(\bullet) = x_i'' = [x_i, \max(\{x_j - x_i \mid j \in (x_i)\})] \quad (7)$$

$$h(\bullet) = x_i' = x_i'' W_{update} \quad (8)$$

In the above equation, x_i'' represents the aggregated features. In the feature update calculation, a multi-branch head parallel computation is used, enabling the network layer to obtain more spatial information. Finally, all the branch computation results are aggregated and concatenated to enhance the overall feature richness. The calculation formula of $h(*)$ can be further optimized as:

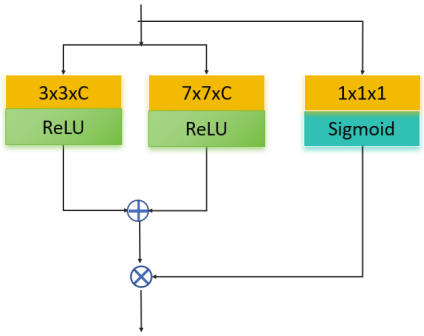
$$h(\bullet) = x_i' \quad (9)$$

$$x_i' = [head^1 W_{update}^1, head^2 W_{update}^2, ..., head^n W_{update}^n] \quad (10)$$

After each graph convolutional layer completes the operation of global feature extraction, the ReLU activation function is applied to add non-linearity to the features, preventing feature loss during training. Finally, a multi-layer perceptron is used to further generalize and enhance the extracted features, mapping the input to the output.

In the design of the convolutional module, this paper proposes a compact multi-scale boost

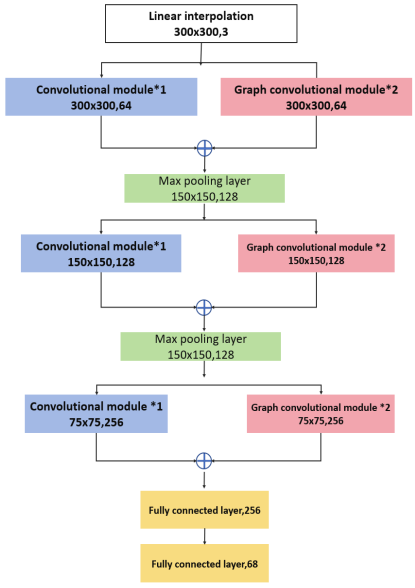
residual structure, responsible for extracting fine-grained and medium-scale features from the image. Its features serve as a complement to the features of the graph convolutional module. The overall module structure is shown in the following figure:



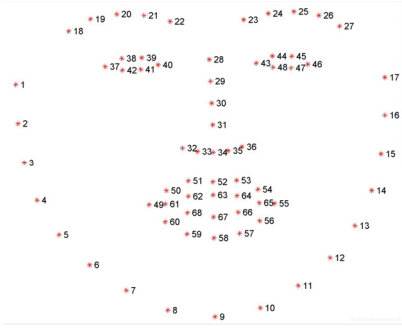
[Fig. 5] Multi-scale Boost Residual Structure

In this module, parallel convolutions with 3x3 and 7x7 kernel sizes are used to extract features, and the extracted features are merged at the output to combine the two parts of the features. In the residual mapping structure, spatial attention operation is employed. Firstly, a 1x1 convolutional kernel is used to compress the fine-grained feature channels of the feature map, resulting in a feature map with a channel size of 1. Then, a sigmoid activation function is applied to calculate spatial weights, and these weights are assigned to the outputs of the parallel convolutions to perform spatial feature boost operation. The face alignment network, constructed based on the graph convolution module and the multi-scale boost residual structure, is shown in the following figure:

In the overall structure, this article also adopts a parallel form to simultaneously obtain fine-grained features and global features. The down-sampling operation for the image uses max pooling. Finally, two fully connected layers are used at the end of the network to map key points and coordinate values, resulting in 68 key points for the face. The numbers and corresponding facial feature information for each key point are as follows: 1-17 represent the outline of the face, 18-22 represent the left eyebrow, 23-27 represent the right eyebrow, 28-36 represent the nose, 37-42 represent the left eye, 43-48 represent the right eye, and 49-68 represent the mouth[15]. The distribution diagram of each key point on the face is shown below.



[Fig. 6] Face Alignment Network Structure

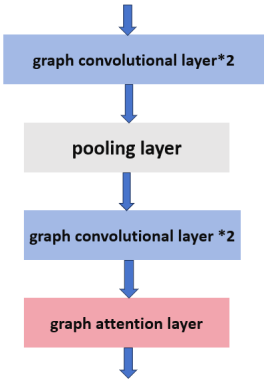


[Fig. 7] Diagram illustrating facial keypoint markers

2.3 Design of Face Recognition Network Based on GCN Module

After obtaining the coordinates of facial key points, a directed graph is constructed based on the relationships between key point numbers. The key points 1-17 representing the facial outline are connected to the remaining 51 key

points, establishing the relationships between the facial features and the outline. This process generates 17 groups of subgraphs, each representing a specific facial region. These 17 subgraphs are then connected in the order of the outline's numbers from 1 to 17, forming a directed graph structure. Subsequently, the graph convolutional layers described in section 2.2 are utilized to extract features through aggregation. The resulting network structure diagram is shown below:



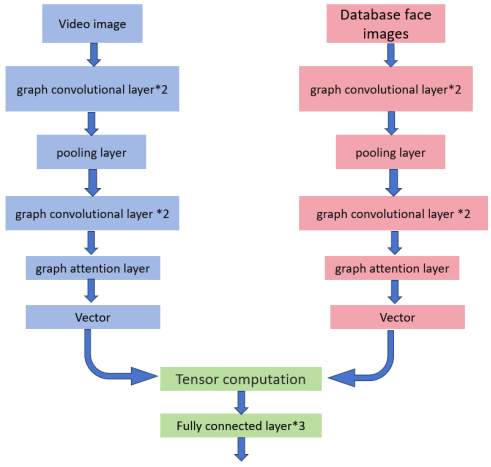
[Fig. 8] Illustration of the Graph Convolutional Network (GCN) structure

As shown in the above diagram, the constructed graph structure is input into the network. Firstly, it goes through 4 graph convolutional layers to perform feature aggregation operations on each key point. Pooling layers are also added to increase the receptive field of graph convolution. At the end of the network, this paper adopts graph attention mechanism. It calculates the relationship between each key point and the entire graph and assigns weight coefficients to each key point in the graph. The formula for computing and assigning graph attention is as follows:

$$h = \sum_{n=1}^{N} \sigma\left(u_n^T c\right) u_n \tag{11}$$

$$h = \sum_{n=1}^{N} \sigma\left(u_n^T \tanh\left(\frac{1}{N} W \sum_{m=1}^{N} u_m\right)\right) u_n \tag{12}$$

In the above equation, $\sigma(\cdot)$ represents the sigmoid activation function, u_n represents the nodes in the graph, c represents the calculation of the average value of the nodes and the non-linear activation function, and W represents the learnable parameters. After the graph attention calculation, the vector h representing the global coordinates of facial key points can be obtained. Subsequently, face identity recognition can be performed. The schematic diagram of this recognition process is shown below:



[Fig. 9] Illustration of Face Identity Recognition

As shown in the diagram, assuming the face in the surveillance video and the face in the database are processed by the graph convolutional network to obtain the keypoint feature vectors h_i and h_j, the similarity between the two vectors can be calculated. Firstly, this paper uses a neural tensor network to concatenate and combine the two vectors, which can be represented by the following formula:

$$g(h_i, h_j) = f\left(h_i^T W^{[1:K]} h_j + V\left[h_i \mid h_j\right] + b\right) \tag{13}$$

In the above equation, both W and V represent weights, where $W^{[1:K]} \in R^{D \times D \times K}$ and D are the dimensions of h_i and h_j, respectively. K

represents the correlation factor between the two vectors of interest. Finally, the concatenated features are processed through a fully connected network to determine whether the input graph structures are from the same source. The loss function of the discriminator can be represented as follows:

$$L = \frac{1}{D} \sum_{(i,j) \in d} \left(\hat{s}_{ij} - s\left(G_i, G_j\right) \right)^2 \tag{14}$$

In the above equation, D represents the set of face data, $S(\bullet)$ represents the calculated similarity of the two graph structures, and $\hat{S}$ represents the predicted result of the network.

3. Experiment and Data Analysis

3.1 Experimental Setup

In this chapter, the detection and recognition accuracy and speed of the designed network model will be compared and tested with other similar algorithms to verify the performance of the constructed network in face identity recognition in surveillance videos. Firstly, in terms of dataset, as there is currently no open-source face dataset from surveillance angles, this paper collected information from residents in a certain community to create a dataset of 100 users' face identity. (We have informed the property management, homeowners' committee, and relevant homeowners in advance that facial data information will only be used for identity recognition experiments and future practical applications, and we have obtained consent from the relevant individuals. Furthermore, the images captured during our testing were strictly saved in accordance with the requirements, prohibiting unauthorized use by any other individuals, in order to respect personal privacy protection.) Each user's face was captured from different angles and directions, resulting in 10 images per user, with 68 key points marked for 3000 facial images. Secondly, the 300W keypoint dataset and the IBUG face identity dataset were selected for testing to evaluate the overall performance of the model in face keypoint detection and identity recognition. Additionally, to evaluate the keypoint detection accuracy, identity recognition accuracy, and lightweight performance of the network model, different evaluation metrics and formulas were used. Firstly, for the evaluation of keypoint detection accuracy, the following metrics were chosen in this paper:

(1) Normalized Mean Error (NME): Used to evaluate the detection accuracy of the model, a lower value indicates a higher detection accuracy of the model.

$$NME = \frac{1}{N \times L} \sum_{i=1}^{N} \frac{\| s_i - \hat{s}_i \|_2}{d_i} \tag{15}$$

In the equation, N represents the total number of detected faces, and L represents the number of key points (which is 68 in this paper). s_i represents the actual coordinates of the key points, $\hat{s}_i$ represents the predicted coordinates of the key points, and d_i represents the normalization method (in this paper, the inter-pupillary distance is chosen for calculating the normalized mean error).

(2) False Rate (FR): Used to evaluate the detection accuracy of the model, a lower value indicates a higher detection accuracy of the model.

$FR = \frac{f_N}{N} \times 100\%$ When the NME of the detection result is larger than a threshold value, the detection result can be considered as incorrect. f_N represents the number of times the detection result is greater than the NME threshold, and N represents the total number of samples.

(3) Cumulative Error Distribution Curve Area (AUC): The horizontal axis of this curve represents the NME threshold, and the vertical axis represents the proportion of samples that satisfy this threshold out of the total number of samples. Therefore, the larger the area under this curve (AUC), the higher the detection accuracy of the model. Next, the evaluation of face identity recognition accuracy is calculated. As the tested face identities are all within the dataset and belong to a closed-set scenario, accuracy is used as the evaluation metric.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{16}$$

In the above formulas, TP, TN, FP, and FN represent the number of correct identification results in the confusion matrix, which is the ratio of the correct detection frequency in all identification results. A higher value indicates a higher recognition accuracy of the model. Finally, the evaluation of the model's lightweight performance is calculated using the following metrics:

Parameter Count: Represents the total number of parameters in the network. A smaller parameter count indicates better lightweight performance of the model.

$$Parameter = \left(C_{\in} \times K^2+1\right) C_{out} \tag{17}$$

Recognition Frame Rate (FPS): Represents the number of images that the model can recognize per second. (The graphics processor used in the experiment is NVIDIA RTX3080Ti)

3.2 Hyperparameter Settings Experiment

Firstly, in the face alignment network, a global graph convolution module is used to aggregate node features. The number of connections each node has with other nodes can impact the performance of the network. To investigate the optimal number of connections, different numbers of connection points were set and tested on the 300W dataset and the self-built dataset in this paper. The test results are shown in the table below.

〈Table 1〉 Global Graph Convolution Node Aggregation Test (300W Dataset)

Number of Nodes	NME	FR	AUC	Parameter	FPS
2	6.43%	9.24%	66.31%	13.01M	37.2
3	3.95%	6.26%	78.67%	13.88M	36.4
4	2.05%	3.89%	85.65%	14.25M	35.8
5	2.14%	3.84%	85.33%	14.73M	35.0
6	2.09%	4.02%	84.25%	15.24M	34.3

〈Table 2〉 Global Graph Convolution Node Aggregation Test (Self-built Dataset)

Number of Nodes	NME	FR	AUC	Parameter	FPS
2	5.67%	5.34%	80.76%	13.01M	37.2
3	2.88%	3.08%	83.64%	13.88M	36.4
4	1.47%	2.11%	88.92%	14.25M	35.8
5	1.55%	2.20%	88.45%	14.73M	35.0
6	1.54%	2.22%	88.07%	15.24M	34.3

As shown in the table above, the testing results on both the open source dataset and the self-built dataset indicate that the network performance reaches its optimal when the number of aggregation points is 4. Furthermore, increasing the number of aggregation points to 5 and 6 does not improve the model's alignment accuracy, but rather shows fluctuations, with an inverse relationship between computational complexity and speed. On the other hand, setting the number of connections to 2 and 3 may have certain advantages in terms of lightweightness, but it significantly deviates from other connection settings in terms of precision in locating key points, making it unsuitable for use. Therefore, based on the above analysis, setting the number of node connections to 4 achieves the best overall performance for the model.

Next, in the face recognition network, when

performing tensor computations on two facial feature vectors, it is necessary to set the correlation factor K between the two vectors (with the number of aggregated nodes in the network set to 4). Similarly, by evaluating the model's recognition accuracy metrics with different K values, the optimal K value hyperparameter can be selected. This experiment was tested on the IBUG dataset and the self-built dataset, and the results are shown in the table below:

〈Table 3〉 Vector Correlation Value Settings (IBUG Dataset)

K value	Accuracy	Parameter	FPS
16	76.23%	21.38M	26.8
32	83.41%	25.62M	25.6
64	79.10%	30.81M	22.0
128	69.77%	37.35M	19.4
256	65.35%	49.07M	15.5

〈Table 4〉 Vector Correlation Value Settings (Self-built Dataset)

K value	Accuracy	Parameter	FPS
16	92.31%	21.38M	26.8
32	96.74%	25.62M	25.6
64	93.42%	30.81M	22.0
128	91.86%	37.35M	19.4
256	88.92%	49.07M	15.5

From the test results in Table 3 and Table 4, it can be seen that the network achieves the highest recognition accuracy when the number of correlation factors is set to 32. However, increasing the number of factors leads to a decrease in recognition accuracy, and the computational complexity of the network also increases due to the increase in correlation calculations, impacting the network's lightweight performance. Therefore, it is recommended to set the number of correlation factors for the recognition network to 32.

3.3 Cross-Comparison Experiment

To further validate the comprehensive performance of the proposed network for face identity verification, we conducted a cross-comparison test with other networks of the same type. In terms of the alignment performance of facial keypoints, we selected the following networks for comparative analysis:

(1) PIPNet[16]: This network is based on heatmap regression and can simultaneously predict scores and offset values on low-resolution feature maps. The network as a whole demonstrates good lightweight performance.

(2) HIH[17]: This network is based on heatmap regression and utilizes the Hourglass module as its backbone. It also employs the argmax function to reduce quantization error in the heatmaps.

(3) SPIGA[18]: This algorithm combines a multi-level heatmap backbone with a graph attention network regressor in a cascaded network form. The cascaded network incorporates position encoding and attention mechanisms, enabling it to learn the geometric relationships between landmarks.

(4) AnchorFace[19]: This network is based on coordinate regression and addresses the landmark prediction problem in images with large poses by configuring anchor targets.

The four aforementioned networks and the proposed graph convolutional network were tested for face alignment on the 300W dataset and a self-built dataset. This evaluation aimed to assess the models' accuracy in locating key points and their lightweight performance.

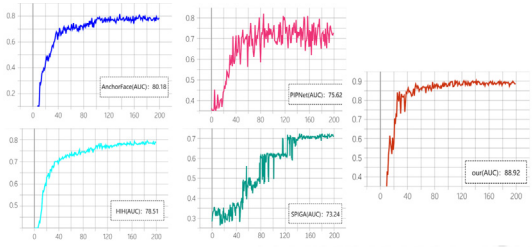
〈Table 5〉 Key Point Alignment Performance Test (300W Dataset)

Network	NME	FR	AUC	Parameter	FPS
PIPNet	4.34%	5.82%	82.33%	12.0M	39.5
HIH	3.89%	4.87%	82.74%	16.23M	31.2
SPIGA	5.01%	5.58%	82.98%	32.59M	19.4
AnchorFace	3.35%	4.01%	83.67%	20.67M	27.3
Our	2.05%	3.89%	85.65%	14.25M	35.8

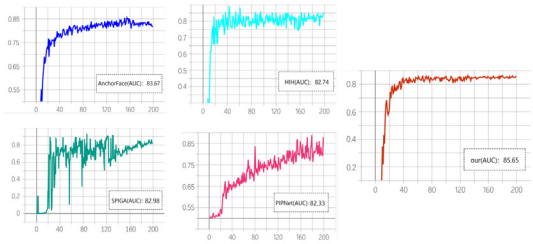
〈Table 6〉 Key Point Alignment Performance Test (Self-built Dataset)

Network	NME	FR	AUC	Parameter	FPS
PIPNet	7.12%	8.21%	75.62%	12.0M	39.5
HIH	5.35%	6.55%	78.51%	16.23M	31.2
SPIGA	6.87%	7.32%	73.24%	32.59M	19.4
AnchorFace	5.28%	7.24%	80.18%	20.67M	27.3
Our	1.47%	2.11%	88.92%	14.25M	35.8

From the comparative analysis in the table above, the proposed graph convolutional network demonstrates superior accuracy performance in key point recognition compared to other types of networks. Furthermore, when comparing the results obtained from the two datasets, it can be observed that the performance of other networks on the self-built dataset is inferior to that on the 300W dataset. In contrast, the proposed algorithm shows further improvement in testing accuracy, indicating its robustness to facial distortions caused by variations in camera angles. The graph convolution's invariant property enhances the network's detection performance. Moreover, from the perspective of model lightweightness, the proposed algorithm is only slightly less lightweight than the PIPNet network. However, there is a significant difference in the detection accuracy of key points between the two, with the proposed network outperforming PIPNet. Therefore, considering both detection accuracy and speed, the proposed network in this study exhibits greater advantages. The iteration curves of the positioning accuracy for each network for the key points are shown in the following composite graph:



[Fig. 10] Table 5: Iteration curves of the positioning accuracy for each network



[Fig. 11] Table 6: Iteration curves of the positioning accuracy for each network

By comparing the positioning accuracy curves from both datasets, it is evident that the proposed network in this paper outperforms in terms of performance.

Next, we evaluated the accuracy of face identity recognition and the model's lightweight performance by conducting comparative tests with other types of networks:

(1) FaceNet[20]: This network uses convolutional neural networks to map facial image features into a 128-dimensional Euclidean space. It calculates the distance in the Euclidean space to obtain similarity scores between images.

(2) SphereFace[21]: This network utilizes the MTCNN network for face alignment and subsequently employs the Angular Softmax Loss for classification. It optimizes the measurement of features.

(3) CosFace[22]: Similarly, this network uses convolutional neural networks for feature extraction. It introduces the Large Margin Cosine Loss, which maximizes the inter-class variance and minimizes the intra-class variance.

(4) ARCFace[23]: It also enhances the discriminative power of the extracted features by designing a loss function.

Similarly, the four aforementioned networks and the network designed in this paper were evaluated for face recognition accuracy on the IBUG open source dataset and a self-constructed dataset. The accuracy of the models in recognizing faces under normal and special viewing angles was analyzed. The test results are shown in the following table:

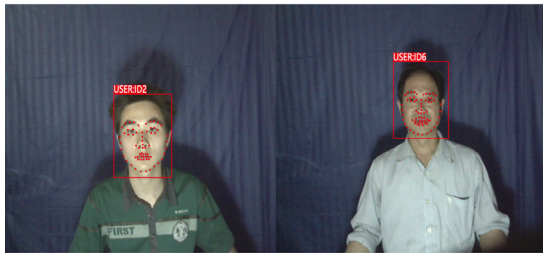
〈Table 7〉 Face recognition performance test (IBUG dataset)

Network	Accuracy	Parameter	FPS
FaceNet	73.56%	43.8M	17.5
SphereFace	74.69%	46.6M	16.8
CosFace	78.74%	45.2M	16.2
ARCFace	80.38%	53.5M	13.8
our	83.41%	25.62M	25.6

〈Table 8〉 Face recognition performance test (Self-constructed dataset)

Network	Accuracy	Parameter	FPS
FaceNet	73.56%	43.8M	17.5
SphereFace	74.69%	46.6M	16.8
CosFace	78.74%	45.2M	16.2
ARCFace	80.38%	53.5M	13.8
our	96.74%	25.62M	25.6

The above test results intuitively demonstrate the proposed network's accuracy in face recognition under surveillance angles. Compared to other networks that extract facial features or compare features based on facial keypoints, the proposed method in this paper, which is based on the graph convolution feature extraction method using 68 facial keypoints, effectively suppresses facial variations in the non-Euclidean feature space, resulting in higher accuracy even under surveillance angles. Moreover, the graph convolution-based recognition network model proposed in this paper also shows certain advantages when analyzed on the IBUG dataset with normal viewing angles. The use of 68 keypoints ensures coverage of the main facial regions, and the construction of a graph structure stabilizes the facial features, leading to superior recognition accuracy compared to other networks. On the other hand, in terms of lightweight performance, the proposed model has an advantage due to the one-dimensional output of the graph convolution, resulting in fewer parameters compared to other convolutional networks. This also translates to faster detection speed compared to other networks. Finally, due to privacy concerns of individuals involved, this paper showcases the detection and recognition results using facial samples from an open-source dataset.



[Fig. 12] Facial detection and recognition result images

As shown in the frontal view facial detection and recognition result images above, the proposed network model in this paper accurately predicts the bounding boxes for facial regions, which closely align with the actual faces. Furthermore, it successfully locates the facial keypoints with precision, enabling the correct identification of individuals based on the final results.

4. Conclusion

This paper addresses the challenge of extracting facial information from surveillance videos in residential areas. We propose a network based on graph convolutional networks for facial keypoint localization and face recognition. In this network, we first use a convolutional neural network to quickly retrieve the faces from video frames. Then, a global graph convolution module combined with a multi-scale enhanced convolution module is employed to localize the facial keypoints. Finally, a graph convolution layer and graph attention calculation are used to determine whether the face in the video frame belongs to the same person as the faces in the user dataset, enabling thorough investigation of individuals in residential areas.In the proposed

network, the global graph convolution module decomposes the face image into node features and utilizes graph convolution to extract facial features. Unlike traditional convolutional calculations, graph convolution breaks spatial limitations and aggregates features based on the special relationships between nodes, enhancing the network's generalization capability. The specific type of graph structure is well-suited for facial keypoint localization, thereby improving the model's expressive power. The graph attention calculation computes the relationship between each node and the graph structure to obtain the node weights, which are then used to assign values to the graph features. This enhances the feature representation of keypoint nodes, suppresses redundant nodes, and strengthens the facial keypoint features. Through extensive experimentation, the proposed network model has demonstrated high recognition accuracy for faces captured from a top-down surveillance perspective, while maintaining a reasonable recognition speed. These findings provide a solid foundation for practical applications. Overall, this research contributes to the development of a network model that effectively addresses the challenges associated with recognizing faces from surveillance videos in residential areas, thus providing valuable insights for real-world deployments.

References

[1] J.Zhang, S.F.Liu, , L.FENG and Y. Zhang. "Research on Face Recognition Algorithm with Fusion Attention Mechanism". Foreign Electronic Measurement Technology, vol.42,No.02, pp.107-113,2023.

[2] Y.H.Fan, Y.Z.Wang, X.F.Yan, L.L.Gong, Y.W.Guo and M.Q.Wei,. "Low-light Image Enhancement Algorithm Driven by Face Recognition Task". Journal of Graphics, Vol.43, No.06, pp.1170-1181, 2022

[3] J.Feng, M.H.Gu, X.M.Liu and L.Cui, "Improving MobileNetv3 Face Recognition Algorithm in Classroom Scenarios." Foreign Electronic Measurement Technology, Vol.41, No.10, pp.47-55, 2022

[4] Boutros F, Damer N and Kirchbuchner F ,et al.Self-restrained triplet loss for accurate masked face recognition[J]. Pattern Recognition, 2022, 124:108473-. DOI:10.1016/j.patcog.2021.108473.

[5] Chen Z, Chen J and Ding G ,et al.A lightweight CNN-based algorithm and implementation on embedded system for real-time face recognition. Multimedia systems, 2023.

[6] T.C.Zhu and X.B.Zhou, "A comprehensive review of deep learning-based methods for face recognition", Modern Computer, Vol.29, No.17, pp.36-40, 2023

[7] L.Di, J.h.Zhangi and J.z.Liang. "Face alignment combined with shape constraints and Gaussian heatmap". International Journal of Machine Learning and Cybernetics, Vol.14, NO.12, PP. 4311-4324, 2023

[8] S.Y.Guan, L.Di,J and Z.Liang ,"Face feature point localization algorithm with adaptive weighting of occlusion", Journal of Miniaturization of Microsystem, Vol.44, No.12, pp.2773-2783, 2023

[9] Y.Q.li, J.Y.Cai, X.Z.main and L.H.Lu, "Face alignment and reconstruction algorithm based on encoder-decoder network". Journal of Computer System Applications, Vol.30,No.07,pp.184-189,2021

[10] L.M.Han, C.Yang, Q.Li, B.Yao, Z.X.Jiao and Q.Y.Xie. "Dynamic deformable transformer for end-to-end face alignment". IET Computer Vision, Vol.17, No.8, pp.948-961, 2023

[11] X.Q.Yang and J.W.Mo, "End-to-end face alignment algorithm based on deep convolutional neural network", Computer Engineering and Design, Vol.40, No.09, pp.2666-2671+2717, 2019

[12] Y.H.Li, X.D.Liu, J.L.Chen and X.P.Su, "Accurate face alignment algorithm based on K-means", Sensors and Microsystems, Vol.40, No.03, pp.120-122+126, 2021

[13] J.Q.Liu, W.X.Tu and E.Zhu, "A comprehensive review of graph convolutional neural networks", Journal of Computer Engineering and Science, Vol.45, No.08, pp.1472-1481, 2023

[14] W.J.Li, J.Bai, B.Peng and Z.Y.Yang,. "A comprehensive review of graph convolutional neural networks and their applications in image recognition",Journal of Computer Engineering and Applications, Vol.59, No.22, pp.15-35, 2023.

[15] X.X.Zhang, Q.C.Tian, L.Lian and R.Tan, "A comprehensive review of facial keypoint detection", Journal of Computer Engineering and Applications, pp.1-13. 2023

[16] Jin H, Liao S and Shao L ."Pixel-in-Pixel Net: Towards Efficient Facial Landmark Detection in the Wild". International Journal of Computer Vision, NO.12, pp.129,2021. DOI:10.1007/s11263-021-01521-4.

[17] Lan X , Hu Q and Cheng J , "HIH: Towards More Accurate Face Alignment via Heatmap in Heatmap". 2021. DOI:10.48550/arXiv.2104.03100.

[18] Andrés Prados-Torreblanca, José Miguel Buenaposada

and Luis Baumela."Shape Preserving Facial Landmarks with Graph Attention Networks." 2022. DOI:10.48550/arXiv.2210.07233

[19] Xu Z, Li B and Yuan Y "AnchorFace: An Anchor-based Facial Landmark Detector Across Large Poses", National Conference on Artificial Intelligence.2021.

[20] Schroff F , Kalenichenko D and Philbin J . "FaceNet: A Unified Embedding for Face Recognition and Clustering". IEEE, 2015. DOI:10.1109/CVPR.2015.7298682.

[21] Wen Y , Liu W and Weller A , "SphereFace2: Binary Classification is All You Need for Deep Face Recognition", 2021. DOI:10.48550/arXiv.2108.01513.

[22] Wang H, Wang Y and Zhou Z, "CosFace: Large Margin Cosine Loss for Deep Face Recognition". computer vision and pattern recognition, pp.5265-5274. 2018

[23] Deng J, Guo J and Zafeiriou S , "ArcFace: Additive Angular Margin Loss for Deep Face Recognition", 2018. DOI:10.48550/arXiv.1801.07698.

담 하 의(He-Yi Tan) [정회원]

- September 1999 ~ July 2003, Hubei Minzu University, Bachelor's Degree in Computer Science and Technology.
- September 2006 ~ December 2008, Graduated with a Master's degree in Software Engineering from Sichuan University.
- March 2020 ~ Present, has been pursuing Ph.D. in Intelligent Fusion in IT at Mokwon University, Daejeon, Korea.

〈관심분야〉

Computer software development, application of artificial intelligence technology, robot control technology

민 병 원(Byung-Won Min) [정회원]

- He received M.S. degree in computer software from Chungang University, Seoul, Korea in 2005.
- He received Ph.D. degree in the dept. of Information and Communication Engineering, Mokwon University, Daejeon, Korea, in 2010.
- He is currently a professor of Mokwon University since 2010.

〈관심분야〉

digital communication systems, Big Data