

Late Fusion of Weak Information for Incompleted Face Recognition using Convolutional Neural Networks: A Novel Approach

Vo Hoang Trong¹ and Pham The Bao²

¹Department of ICT Convergence System Engineering, Chonnam National University, Republic of Korea ²Department of Computer Science, Sai Gon, University, Vietnam
hoangtrong2305@gmail.com ptbao@sgu.edu.vn

Abstract: This work presents a novel method for facial recognition that aggregates weak facial parts to overcome the problem of recognizing faces with incomplete information. By applying a landmark detection method that approximates 70 facial landmarks, we divide the face into 12 weak information parts. These parts are further refined using superpixel segmentation, which increases the precision of face feature analysis. A weak classifier is training a weak part in order to recognize the face. We then aggregate these weak classifiers with a late-fusion voting technique to build a robust classification system. Our technique has been examined on subsets of the FaceScrub and VGG datasets, with a particular focus on frontal face pictures. Experiments show the effectiveness of our method, with an accuracy range of 54.5593% to 91.6346% on the tiny FaceScrub dataset and 45.8654% to 76.4231% on the tiny VGG dataset, depending on how many weak information parts are considered. This result demonstrates the potential for combining available weak information parts to improve the accuracy of face recognition systems, especially in situations where conventional approaches may be hampered by missing or occluded facial data.

Keywords: face recognition, incomplete face information, multi-CNN, late fusion method.

1. Introduction

The field of computer vision has been extensively involved in addressing the complex issue of face recognition, a crucial element of biometrics that seeks to identify or authenticate people. This technique is essential in security systems, leveraging the distinctiveness of facial characteristics. Despite the fact that there have been numerous advances in face recognition technology, a number of factors can significantly affect how precise these methods are. Factors such as noise in facial images, oblique angles, profile faces, low image quality, or hidden essential facial characteristics might significantly diminish the effectiveness of these methods. Historically, facial recognition methods have been developed from frontal face images, acquiring information about facial characteristics globally. Nevertheless, their effectiveness dramatically diminishes when performed on incomplete face images, whereby only a limited number of characteristic parts, known as 'weak information parts', are available.

This research presents a novel framework for incomplete face recognition by combining various weak information parts. The framework utilizes a method of local feature learning, which involves dividing the face into 12 weak information parts such as the eyes, nose, mouth, cheeks, and forehead. The division is accomplished by employing Kazemi et al.'s method [1] of localizing 68 landmarks, with an extra estimation of 2 landmarks in the forehead region. Nevertheless, these parts may not exhibit consistent distinctiveness. For example, the nose region may overlap with some portions of the eye region. To tackle this issue, we apply superpixel clustering to combine adjacent pixels that have similar features, thereby effectively separating weak information parts. After separations, a specialized convolutional neural network (CNN) model is used to analyze each part. This model serves as a weak classifier that has been trained to identify people using only one weak information part. These classifiers are then combined through a late-fusion voting method to create a strong facial recognition system. This method aggregates the results of multiple weak classifiers, selecting the identity label with the highest score after voting.

Received: January 26th, 2023. Accepted: June 27th, 2024

DOI: 10.15676/ijeel.2024.16.2.2

The effectiveness of our novel framework is evaluated using subsets of the FaceScrub and VGG datasets, which exclusively consist of frontal face images. The results of this study are encouraging, indicating an average precision of 54.5593% with a single weak information part and 91.6346% with all 12 parts combined on the tiny FaceScrub dataset. Similarly, the accuracy of the tiny VGG dataset is 45.8654% when considering only one weak information part and 76.4231% when considering all 12 parts. The findings emphasize the potential of our suggested framework for improving the precision of face recognition systems, especially in difficult circumstances where complete facial data is unavailable.

2. Related works

Many approaches have been investigated in the field of face recognition to increase precision and flexibility in various scenarios. One approach worth mentioning is the aggregation of sub-regions for face recognition. Li *et al.* [2] split the face into many rectangular blocks and obtained SIFT feature vectors from each using the Bag-of-Words (BoW) approach. These vectors were then quantized into codewords for each block, and histograms of codeword distribution were combined to represent the face. Although this approach demonstrated remarkable accuracy, particularly in dealing with facial emotions or occlusion, its effectiveness was mostly limited to images with minimal occlusion. The use of histogram representation made the BoW technique sensitive to face variances, hence making it less suitable for incomplete facial information. Furthermore, the division process using uniform rectangular blocks could include non-facial areas, thus potentially disrupting the histogram representation. In contrast, our work utilizes superpixels for division, a technique that reduces processing complexity, simplifies feature calculation, and better accommodates the natural contours of the face [3].

Conventional approaches that use global features like PCA and LDA or local features such as Gabor and LBP typically require face alignment and yield uniformly sized concatenated feature vectors. However, these approaches were not effective in incomplete face recognition scenarios, where the face might lack essential information, causing difficulties with alignment and capturing critical features, especially in profile views. In response, Liao *et al.* [4] introduced the Multi-Keypoint Descriptor for facial representation and the Gabor Ternary Pattern for recognition without requiring facial alignment. This method surpassed PCA, LDA, and LBP in accuracy, especially in challenging scenarios such as face occlusion, face accessories, restricted field of view, intense light, and sensor saturation. Despite achieving higher accuracy, it required a substantial amount of keypoints and dictionary usage, resulting in significant computational load and a disregard for the geometric details of feature sets [5]. To address the challenges of partially captured facial images, Cao *et al.* [6] explored the mapping process between frontal and profile faces, developing a novel technique called the DREAM block. Furthermore, Schroff *et al.* [7] introduced Facenet, a system that learned to map face images to a condensed Euclidean space, generating a 128-dimensional feature vector for tasks like recognition, verification, and clustering. Despite producing state-of-the-art results, Facenet required training on a large dataset of over 200 million images across 8 million classes, necessitating significant computational resources.

The development of specific loss functions further distinguished the advancement in facial recognition. Wen *et al.* [8] presented the center loss technique combined with a summation operator and Softmax loss to refine facial recognition features. Similarly, Liu *et al.* [9] developed the A-Softmax loss function, which aimed to derive angularly discriminative features from the CNN's fully connected layer. Meanwhile, Deng *et al.* [10] created the Additive Angular Margin Loss, normalizing features and weights to enhance CNN-based learning. A common limitation of these methods was the potential misidentification of non-facial areas as facial components, depending on the face dataset used. Danh *et al.* [11] proposed a method for recognizing faces through the late fusion of multiple facial components, such as the eyes, nose, mouth, and the entire face. This method combined several independently trained weak classifiers for each facial component to form a strong classifier, achieving high accuracy on frontal face datasets with varying expressions. However, their research was limited to frontal faces with a small number

of facial components, segmented into rectangular blocks. This segmentation risked disrupting the geometric structure of certain non-frontal faces, thereby reducing accuracy. Our work expands on this approach by examining a larger number of weak information parts on the face, such as the cheeks, chin, and forehead. These parts are segmented using superpixels and landmark localization techniques, ensuring their uniformity in non-frontal faces and under varying emotional expressions.

3. Methodology

The foundation of our methodology is the idea that combining the outcomes of weak classifications from various facial regions can yield robust face recognition. Figure 1 provides a simple visual representation of our approach, outlining the full procedures from image acquisition to final classification. Initially, the Division of Facial Parts module processes the input image by segmenting the face into various weak information parts. This module consists of three key stages: facial segmentation, which detects the face within the image; facial preprocessing, which standardizes the image for consistent analysis; and the division of the face into weak information parts using superpixel techniques, where each part represents a distinct facial feature, such as the left eye, right eye, nose, mouth, forehead, etc. We design a weak classifier for each segmented weak information part that can recognize faces based on the information in that part alone. These classifiers are considered 'weak' since each part has insufficient characteristics for effective facial recognition on its own. The performance metrics from these classifiers are then used as coefficients in our late fusion voting technique, which strengthens the decision-making process in the final classification stage.

Experiments are conducted with a specific subset of weak information regions on faces in order to validate our methodology. With a voting method and a late fusion strategy, we build a strong classifier for a face that has only a few weak information parts, such as 2, 3, and 4. Creating a scoreboard for each potential identity is required for this procedure; these scores are calculated using a combination of the probability vectors p_i and the coefficients α_i of each part i . Subsequently, the test image is categorized according to the identity label that has accumulated the highest cumulative score. This methodology leverages the capabilities of a combination of multiple weak classifiers with a late-fusion voting method. This not only enhances accuracy in challenging scenarios such as partial or occluded faces but also ensures robustness against variations in facial expression and orientation.

A. Division of Facial Parts

During this stage, our aim is to divide the face into weak information parts by employing a series of three modules, as illustrated in Figure 2.

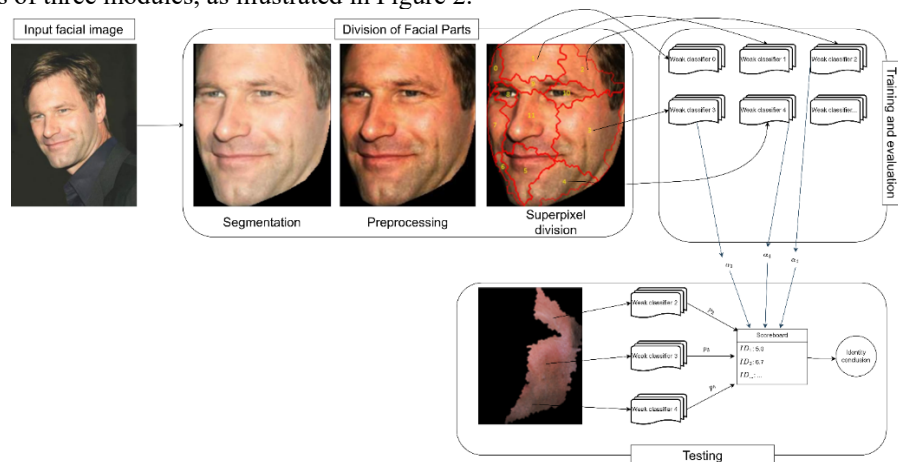


Figure 1. Overview of the proposed facial recognition methodology. The method begins with the segmentation of an input facial image into many weak information parts, followed by

extensive preprocessing and the use of superpixel segmentation to improve feature delineation.

Next, a specialized weak classifier is assigned to each part, which is trained to identify facial features alone from that part. During the testing step, assuming that only parts 2, 3, and 4 of the face are available, we use a late fusion technique to combine the matching weak classifiers. This involves starting a voting process and making a scoreboard for identity labels. Eventually, the identity of the face is decided by selecting which identity has the best score.

A.1. Facial Segmentation

We use a technique introduced by Kazemi and Sullivan [1] in conjunction with the Histogram of Oriented Gradient (HOG) technique, as proposed by Dalal and Triggs [12], to detect the face in an image and locate 68 landmark coordinates $coord_i, i = \overline{0..67}$. Using the description of the 68 landmark points on the face I , we calculate the centroids of the left eye I_{Leye} and right eye I_{Reye} using formula (1), with $n = 36, k = 41$ for I_{Leye} and $n = 42, k = 47$ for I_{Reye} . Since the identified face could be skewed, we normalize it by positioning the line between centroids I_{Leye} and I_{Reye} of two eyes, parallel to the lower border of the image, as shown in Figure 3.

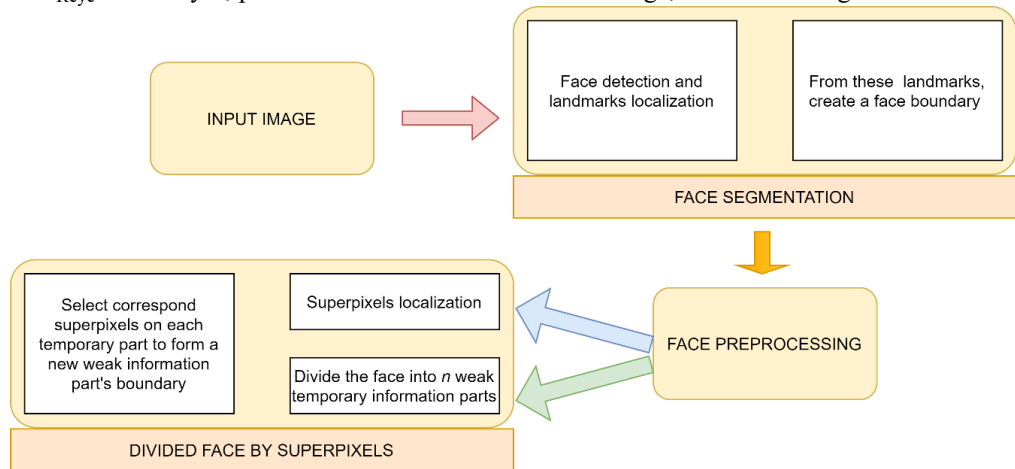


Figure 2. Modules for the division of facial parts. Given an input image, the face is segmented by detecting and localizing landmarks to define the face border, setting pixels beyond this boundary to zero. Then, facial image preprocessing begins. Finally, we divide the facial image into n weak temporary information parts, then apply superpixels to build fresh boundary lines.



Figure 3. Face normalization. Detecting a face in the left image and localizing features, especially eye landmarks, calculates their centroid. The image on the right shows the face after rotation.

$$I_{\text{centroid}} = \frac{1}{n-k+1} \sum_{i=k}^n coord_i \quad (1)$$

Kazemi's method for locating the 68 markers does not include those on the forehead. Two additional landmarks, one located above the left eye and the other above the right eye, are necessary to acquire the forehead part. We define the vertical face axis of the normalized face as the line that goes through the middle point of the two eye centres and the point on top of the chin. To identify a left (or right) forehead landmark, as illustrated in Figure 4, we choose a point above the image's edge such that a line passing through it and the left (or right) eye centroid is parallel to the vertical face axis. This results in a total of 70 landmarks, with landmarks 68 and 69 representing the left and right foreheads, respectively. We design the smallest rectangle capable of encompassing all of the 70 landmarks by identifying the two largest and smallest values on the horizontal and vertical axes. We crop and normalize the face image within this rectangle to produce the corrected image shown in Figure 4, setting all pixel values outside the face region to zero.

With landmarks 0 through 16 located on the face boundary, Kazemi's description 68 landmarks aid in the formation of a closed face boundary. We create a set Q_1 of pixel coordinates by identifying those for every two consecutive landmarks that are on the same line segment. In addition, two lines parallel to the vertical face axis are drawn, one going through landmark 0 and the other through landmark 16, crossing at locations A_1 and A_2 on the image's top edge, from left to right, respectively. The pixels on the line segments connecting landmark 0 to A_1 , landmark 16 to A_2 , and from A_1 to A_2 create sets Q_2 , Q_3 , and Q_4 . The closed face border is defined by pixels in the set $Q = \bigcap_{i=1}^4 Q_i$. All pixels inside this border are considered part of the face region, while all pixels outside of it are considered part of the background. Figure 4 illustrates the results of the segmentation process, which begins with the original image on the left and continues via face detection and landmark localization. The normalized face is in the center image, and the segmented face is in the right image, which is then utilized in the preprocessing stage.



Figure 4. Left: Input image. Middle: Face crop from the input image and localize 68 landmarks plus two more landmarks on top of the forehead, and from landmarks that land on the face edge, we create a face boundary. Right: Pixels outside the face boundary have a value of 0.

A.2. Facial Preprocessing

Different lighting or illumination conditions frequently have an impact on facial images in practical situations. This variability may appear as shadows, overexposure, or underexposure, impacting the visibility and distinguishability of facial features. The challenges become more noticeable in settings with varying light conditions, causing some areas of the face to be illuminated while others appear in shadow. In our proposed methodology, weak classifiers are trained on the weak information parts of the face, which are particularly affected by a lack of illumination. These parts contain low-level features that are particularly vulnerable to degradation due to lighting conditions. Shadows or bright light may obscure features within these parts, making accurate facial recognition more challenging. Technically, the lighting source L that strikes the surface of the face R can impact the intensity of an image. In order to restore the initial facial information and alleviate the impact of L , gamma correction is applied subsequent

to face segmentation. Because gamma correction modifies intensity levels to correct shadows and highlights resulting from uneven lighting, it can restore the original appearance of face features, which is useful for preserving low-level features on weak information parts.

The gamma correction technique, which is a nonlinear grayscale transformation, modifies the pixel intensity $I(x, y)$ in image I to $(I(x, y))^\gamma$, where $\gamma > 0$ and the intensity is contained within the interval $[0, 1]$. This technique enhances the visibility of darker pixels when the value of γ is less than 1, decreases the brightness of pixels when the value of γ is more than 1, and maintains the intensity of the pixels when the value of γ is equal to 1. For images with intensity values ranging $[a, b]$, we suggest a heuristic for calculating γ by dividing $[a, b]$ into three equal segments:

- Dark region: The intensity is in the range $[a, (b - a)/3]$.
- Medium region: The intensity is in the range $[(b - a)/3, 2 \times (b - a)/3]$.
- Bright region: The intensity is in the range $[2 \times (b - a)/3, b]$.

The application of gamma correction varies between channels. After segmentation, we calculate the mode and median by analyzing the density of pixel intensities in the facial region. Experimentally, we choose $\gamma = 0.75$ if the mode and median are in the dark region, suggesting that facial enhancement is required. Conversely, $\gamma = 2$ is selected if they are in the light region, which requires a reduction in intensity. Otherwise, γ is set to 1. The intensity of pixels (x, y) in image I is modified as follows, following the selection of an appropriate γ value:

$$(x, y) := (b - a) \times \left(\frac{I(x, y)}{b - a} \right)^\gamma \quad (2)$$

With $\gamma = 2$, Figure 5 shows how gamma correction reduces lighting noise by comparing the image before (left) and after (right) the correction. By normalizing the lighting conditions across images, gamma correction indirectly contributes to the enhancement of the overall performance of the facial recognition system. It enables the deep neural network models to learn and generalize better from the data, resulting in improved accuracy and robustness against variations in lighting conditions encountered in real-world scenarios.



Figure 5. Left: Image before gamma correction. Right: Image after gamma correction ($\gamma = 2$).

A.3. Division of Face by Superpixel

Certain face features, such as the lips, nose, eyes, wrinkles on the forehead, or a scar on the left cheek, can act as weak parts for recognition when a face is not entirely captured in the image. In this module, we divide the face into 12 weak information parts that correspond to specific features: the left forehead, middle forehead, right forehead, right cheek, right chin, mouth, left chin, left cheek, left eye, the skin between the eyes, right eye, and nose. These locations are shown in the right image of Figure 6. Starting with 70 landmarks, we select 18 main points, as

indicated in the left image of Figure 6, and divide the face into 12 weak temporary information parts. Practically, these divisions may not often encompass entire parts in a uniform manner, which results in characteristics being represented in a variety of ways. In order to ensure feature consistency over these parts, we utilize superpixel segmentation to cluster neighboring pixels with similar features.

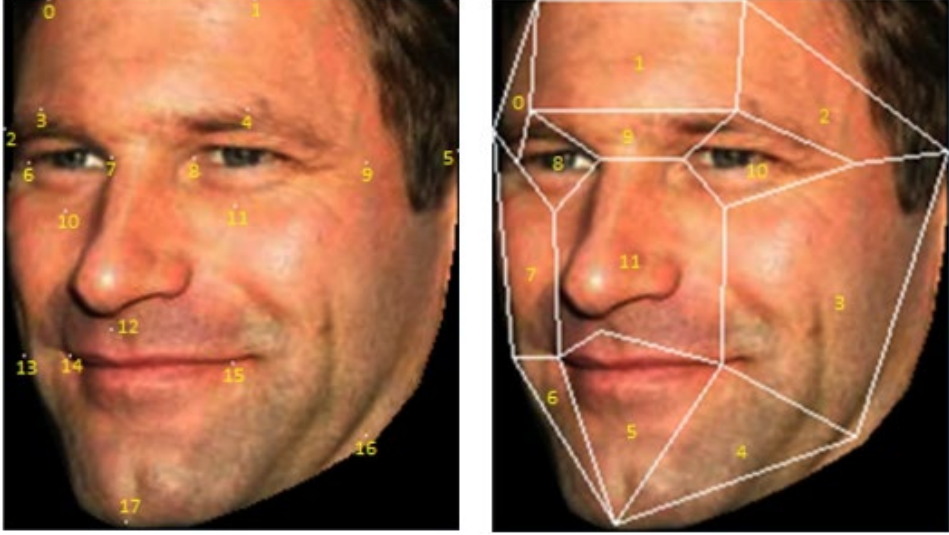


Figure 6 Left: 18 main points on the face. Right: Delineation of 12 weak temporary information parts.

Given the variety in image sizes across the dataset, the number of superpixel segments required is dependent on the image size. In order to estimate the proper number of superpixel segments, we propose the following linear function $f(x)$:

$$f(x) = ax + b \quad (3)$$

where:

- x represents the area of the input image,
- $f(x)$ indicates the number of superpixel segments,
- a and b are coefficient.

This function suggests that if $x \leq -b/a$, then $f(x) \leq 0$, which is not possible because it would result in a negative number of superpixel segments. Hence, in order to guarantee at least 50 superpixel segments, we apply this function to images with an area of $x \geq (50 - b)/a$. After superpixel segments have been distributed across the face, one segment might cover numerous weak temporary information parts. Hence, we say that a superpixel segment A is assigned to a weak temporary part i if the following conditions are met:

$$i = \arg \max(\{P_j \cap A\}) \quad (4)$$

where P is the set of all temporary information parts intersected by A .

After assigning superpixel segments to each weak temporary information part, we redraw the borders of these parts depending on the superpixel segments they include, as seen in the right image of Figure 7. These parts are subsequently used in the specific DNNs architecture for classification.

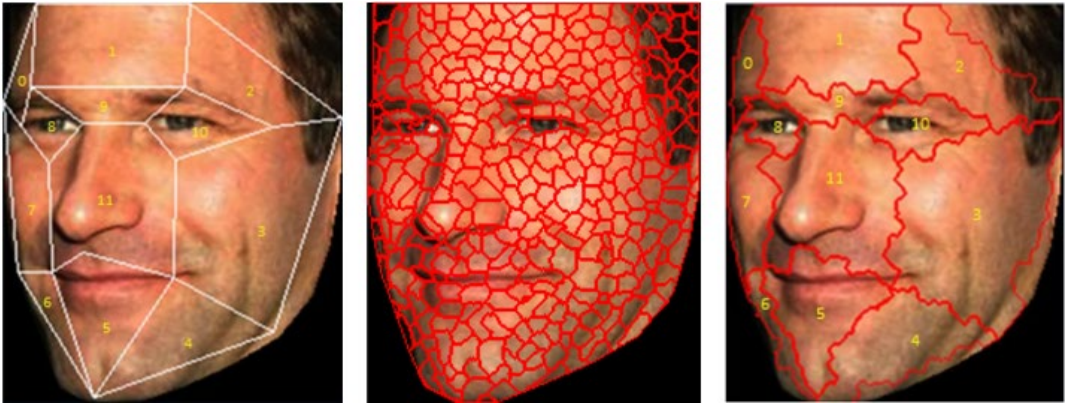


Figure 7. A general overview of facial division. The left image shows 12 weak temporary parts. The middle image shows the application of superpixels on the face. The right image shows the selection of superpixels for each part.

B. Deep Neural Networks

B.1. Rescale Weak Information Parts

Because image sizes vary within the facial dataset, the dimensions of weak information parts can range dramatically. Before applying deep neural networks (DNNs), the dimensions of each part are standardized to mitigate this issue. As shown in Table 1, we establish constant dimensions for each weak information part in consideration of the face's vertical symmetry. The specified dimensions are defined in natural numbers: $(a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, e_1, e_2, f_1, f_2, g_1, g_2, h_1, h_2)$. For any given weak information part i across the dataset, we construct a generic size of $x_i \times y_i \times 3$ by averaging the dimensions of the image height and width (assigned to x_i and y_i).

Table 1. Dimensions $x_i \times y_i \times 3$ of information parts, with x_i and y_i representing height and width, respectively

Size	Information part
$a_1 \times a_2 \times 3$	0, 2
$b_1 \times b_2 \times 3$	1
$c_1 \times c_2 \times 3$	3, 7
$d_1 \times d_2 \times 3$	4, 6
$e_1 \times e_2 \times 3$	5
$f_1 \times f_2 \times 3$	8, 10
$g_1 \times g_2 \times 3$	9
$h_1 \times h_2 \times 3$	11

B.2. Image Augmentation

We enhance the dataset for each weak information part using image augmentation, producing an additional 30 images from each original image in the training set, to limit the risk of overfitting. This augmentation consists of adding blur filtering, bright and dark gamma correction, and sharpening filtering to the source image. Subsequent manipulations of these five images include left and right rotations, vertical flipping, and flips paired with rotations, yielding a total of 30 augmented images, as shown in Table 2.

Table 2. 30-image augmentation strategy

	Original	Gaussian blur filtering	Bright Gamma correction	Dark Gamma correction	Sharpen filtering
Left rotation					
Right rotation					
Vertical flip					
Left- rotation vertical flip					
Right- rotation vertical flip					

B.3. Data Normalization

The application of normalization helps to align the distribution of the data, which in turn promotes faster convergence during training. We use equations (2) and (3) to find the c -dimensional mean (\bar{M}) and standard deviation (σ) for a training set of n images. Every image in the training, validation, and test sets is then normalized to zero mean and unit variance by removing \bar{M} and dividing by σ , according to equation (7).

$$\bar{M}(x, y, c) = \frac{\sum_{i=1}^n I_i(x, y, c)}{n} \quad (2)$$

$$\sigma(x, y, c) = \sqrt{\frac{\sum_{i=1}^n (\bar{M}(x, y, c) - I_i(x, y, c))^2}{n}} \quad (3)$$

$$I(x, y, c) := \frac{I(x, y, c) - \bar{M}(x, y, c)}{\sigma(x, y, c) + 1e^{-7}} \quad (4)$$

B.4. CNN Architectures

In our study, the weak information parts of a face are primarily characterized by low-level features. These characteristics, though not as prevalent as those used in recognizing complete frontal faces, play a vital role in identifying faces with incomplete information. Our aim requires the use of weak classifiers that capture these small, low-level features. It is important to realize that not all low-level features in weak information parts play an equal role in facial recognition. Certain parts of the face, such as the mouth or nose, may contain more unique characteristics that are closely tied to a person's identity, while parts like the cheek or forehead may provide fewer characteristics. This variability requires a CNN model capable of capturing a wide range of low-

level features with different levels of recognizability. Thus, a multi-CNN approach is utilized, with a CNN model assigned to each weak information part as a weak classifier. The VGG network employs a series of convolutional layers (CONV) followed by max-pooling (POOL) layers, repeating this pattern to progressively deepen the network. The research employs a simplified version of the VGG architecture [13] that iterates the CONV-->CONV-->POOL (CCP) sequence across the network. This structure allows for a hierarchical processing of features, with initial layers capturing basic visual features such as edges and textures, which are then combined in subsequent layers to form complex patterns. This design is adept at extracting and refining low-level features across its layers, making it particularly suitable for identifying and enhancing low-level features within each weak information part.

The architectural layers used for training are illustrated in Figure 8. Parameter optimization is carried out using Adadelta, and the loss function is estimated using Softmax and cross-entropy [14]. In the final block, average pooling is used to match the size of the previous convolution layer, followed by batch normalization [15] to prevent overfitting and accelerate training. The data within each identity is partitioned in the following proportions: 6:2:2 for training, validation, and testing. On the training set, the CNN model is trained, and the model that achieves the largest accuracy β_i in the validation set is retained during the training process.

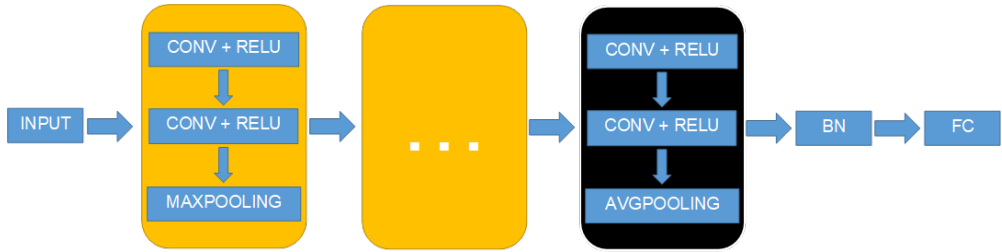


Figure 8. General CNN architecture with repeated CCP blocks and average pooling at the end. The last layers are batch normalization (BN) and a fully connected (FC) layer.

C. Late Fusion of Weak Information Parts Using a Voting Method

In this paper, we present a late fusion strategy that uses a voting method to combine the weak information parts discovered on the face, with the goal of developing a robust classification mechanism. For k weak classifiers, we consider the probability vector p_i for each weak information part i , with $1 \leq i \leq k$. This vector, which has a length that is equal to the number of identities that are contained in the dataset, is derived from the Softmax layer of each classifier i . We then select the top m_i identities inside p_i with the highest probability. The voting procedure begins by creating a score table in which each identity starts with a score of zero. For each weak information part i , the product of its likelihood and the part coefficient α_i raises the score of identities inside the top m_i . The identity with the largest cumulative score in this table is then identified as the classification result for the image.

In mathematical terms, let n represent the entire amount of weak information parts present on the face. We make use of $\beta_i \in [0,1]$, which represents the largest probability that was attained on the validation set when training, in order to calculate the coefficient α_i for each weak information part i . The equation used to determine α_i is:

$$\alpha_i = \frac{\beta_i}{\sum_{j=1}^n \beta_j} \quad (5)$$

The Softmax layer of the CNN architecture, which is responsible for weak information part i , generates a probability vector p_i of length c , which corresponds to the number of identities of the dataset. Each element $p_i[t]$, where $1 \leq t \leq c$, represents the probability of part i belonging to an identity t . Assuming the test face has k weak information parts ($k \leq n$), for each part i , we extract the top m_i identities with the largest probabilities, forming the set O_i as indicated in equation (6). The scoring for each identity is conducted as per equation (7).

$$O_i = \{c^{m_i} | p_i[c^{m_i}] \text{ is one of the top } m_i \text{ largest value on } p_i\} \quad (6)$$

$$S(u) = \sum_{i=1}^k (\alpha_i \times p_i[u]) \times P(u, O_i) \quad (7)$$

with

$$P(u, O_i) = \begin{cases} 1, & u \in O_i \\ 0, & u \notin O_i \end{cases} \quad (8)$$

Finally, $\arg \max_u S(u)$ determines the predicted identity of the face image, thereby combining contributions of each weak classifier into a complete decision-making framework.

4. Experiments

We evaluated our method on a system with 24GB of RAM, an Intel (R) CoreTM i7-6700K CPU running at 4.00GHz, and a 4GB NVIDIA GeForce GTX 980 GPU. We implemented software using Python in conjunction with Keras and TensorFlow.

A. Datasets

To evaluate our novel approach, we exclusively utilized frontal face images from the dataset. We examined both scenarios in which all 12 weak information parts were present and those in which only a subset of these parts were visible, as outlined in Table 3. The original FaceScrub dataset [16], [17] has 106,863 images spread across 530 identities, with an average of 200 images per identity. Images from the FaceScrub dataset are shown in Figure 9. We created a smaller



Figure 9. Sample images from the FaceScrub dataset.



Figure 10. A selection of VGG dataset images.

subset of this dataset called 'tiny FaceScrub', which contains 65 identities, each represented by 80 randomly picked images. Similarly, the original VGG [18] [19] dataset, which is famous for its collection of celebrity images on the internet, contains 2,622 identities. A selection of samples taken from the VGG dataset is presented in Figure 10. Our studies were conducted on a 'tiny VGG' dataset, which had 100 identities and 130 images for each. The tiny VGG dataset, in contrast to the tiny FaceScrub dataset, contains cases in which images from one identity may show individuals from different classes. This is demonstrated in Figure 11, which contains images from the Adam Beach class that mistakenly include the faces of Tom Jackson, Robin Williams, and Adam Walsh.



Figure 11. 26 preprocessed test images from the Adam Beach class, including Tom Jackson, Adam Walsh, and Robin Williams.

Table 3 Image illustration on information parts used for testing

Weak informa tion parts on the face	All 12	1, 5, 9, 11	0, 1, 2, 8, 9, 10	4, 5, 6	3, 7, 11	0, 6, 7	2, 3, 4
Illustrat ion							

B. Data Preprocessing

For both the tiny FaceScrub and tiny VGG datasets, we divided the detected faces into 12 weak information parts. To determine the necessary amount of superpixel segments for every face, we utilized formula (3) with $a = 0.07$ and $b = -1125.46$. We made sure to use only face images that had a minimum of 16,793 pixels, so each image would have at least 50 superpixel segments. As shown in Table 4, each part of the tiny FaceScrub dataset was scaled to conform to a standard dimension. We estimated the mean \bar{M}_{FS}^i and standard deviation σ_{FS}^i for all images in each weak information part i in the training set. Equation (4) used these statistics to standardize the images across parts.

Table 4 Dimensions of weak information parts of the FaceScrub dataset

Part	0	1	2	3	4	5
Size	$76 \times 59 \times 3$	$80 \times 148 \times 3$	$76 \times 59 \times 3$	$94 \times 50 \times 3$	$63 \times 68 \times 3$	$94 \times 94 \times 3$
Part	6	7	8	9	10	11
Size	$63 \times 68 \times 3$	$94 \times 50 \times 3$	$98 \times 95 \times 3$	$40 \times 104 \times 3$	$55 \times 62 \times 3$	$108 \times 107 \times 3$

After resizing each weak information part in the tiny VGG dataset according to Table 5, we computed the mean \bar{M}_{VGG}^i and standard deviation σ_{VGG}^i for each part i . Normalization of images within these parts was conducted using equation (4), based on the obtained \bar{M}_{VGG}^i and σ_{VGG}^i .

Table 5 Dimensions of weak information parts of the VGG dataset

Part	0	1	2	3	4	5
Size	$68 \times 48 \times 3$	$50 \times 100 \times 3$	$68 \times 48 \times 3$	$106 \times 58 \times 3$	$60 \times 66 \times 3$	$80 \times 76 \times 3$
Part	6	7	8	9	10	11
Size	$60 \times 66 \times 3$	$106 \times 58 \times 3$	$46 \times 50 \times 3$	$36 \times 82 \times 3$	$46 \times 50 \times 3$	$88 \times 88 \times 3$

C. CNNs

We analyzed images of weak information parts, each of which had dimensions of $x \times y \times 3$, in accordance with our experiments. We utilized two different forms of the small VGG design, one with three CCP (CONV->CONV->POOL) blocks and another with four. The size of the part determined the number of CCP blocks used in each variant. For the tiny FaceScrub dataset, which consists of 65 identities with 80 images each, we allocated 3120 images (60%) for training, 1040 images (20%) for validation, and the remaining 1040 for testing. The augmentation of the training set images produced a total of 93,600 images for training. The 12 weak information parts were processed using four CCP blocks for parts 1, 5, and 11, and three CCP blocks for the other parts. The model parameters were optimized according to the highest probability on the validation set, after 300 epochs of training with a minibatch size of 150 were completed. The accuracy found on the validation set is shown in Figure 12. These were then used in equation (5) to get the coefficients for each weak information part, which are listed in Table 6.

Table 6. Validation set accuracy and coefficients for weak information parts on the FaceScrub dataset

Part	0	1	2	3	4	5
Accuracy on validate set	41.923%	57.019%	43.077%	51.058%	34.904%	67.212%
Coefficient	$\frac{41923}{654712}$	$\frac{57019}{654712}$	$\frac{43077}{654712}$	$\frac{51058}{654712}$	$\frac{34904}{654712}$	$\frac{67212}{654712}$
Part	6	7	8	9	10	11
Accuracy on validate set	34.327%	49.615%	65.673%	61.442%	64.712%	83.750%
Coefficient	$\frac{34327}{654712}$	$\frac{49615}{654712}$	$\frac{65673}{654712}$	$\frac{61442}{654712}$	$\frac{64712}{654712}$	$\frac{83750}{654712}$

The tiny VGG dataset, which had 100 identities and 130 images each, was divided into 72 images for training (after producing an extra 30 images per original image for augmentation, for a total of 216,000 training images), 2600 images for validation, and another 2600 for testing. Because of the smaller image sizes in this dataset, parts 5 and 11 received four CCP blocks, while the remaining areas received three CCP blocks. The training had a minibatch size of 100 and consisted of 300 epochs, which was similar to the approach used for the FaceScrub dataset. The parameters were chosen based on validation set accuracy, and the results were verified on the test set. With coefficients determined using equation (5) and shown in Table 7, Figure 13 shows the training and validation accuracy rates throughout the 12 weak information parts.

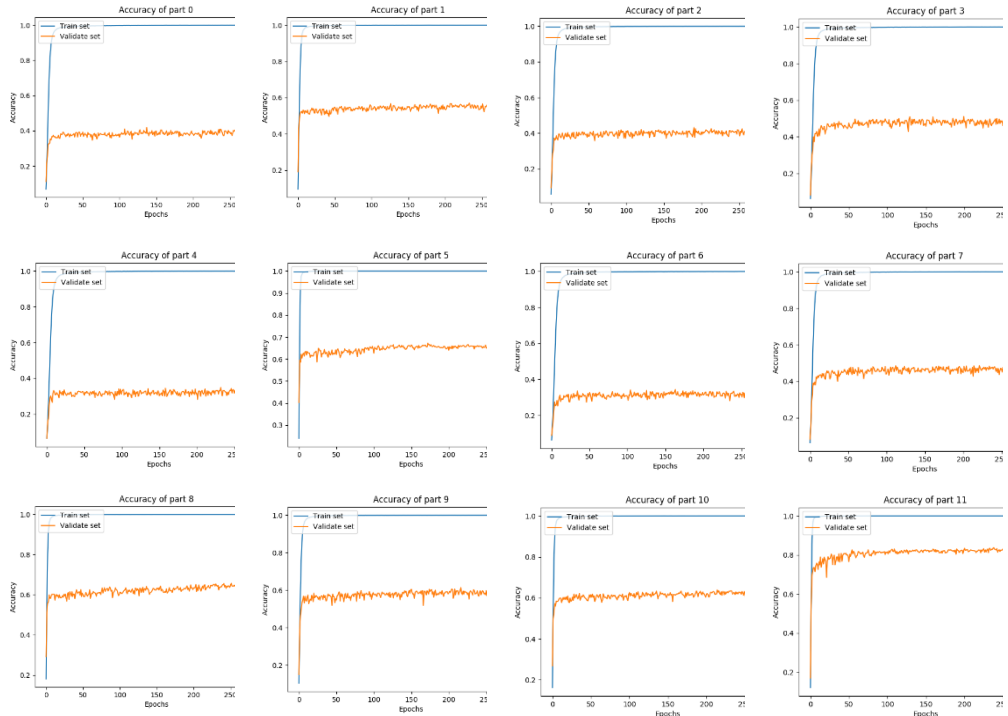


Figure 12. Sequentially organized accuracy rates on the training and validation sets throughout the 12 FaceScrub dataset weak information parts.

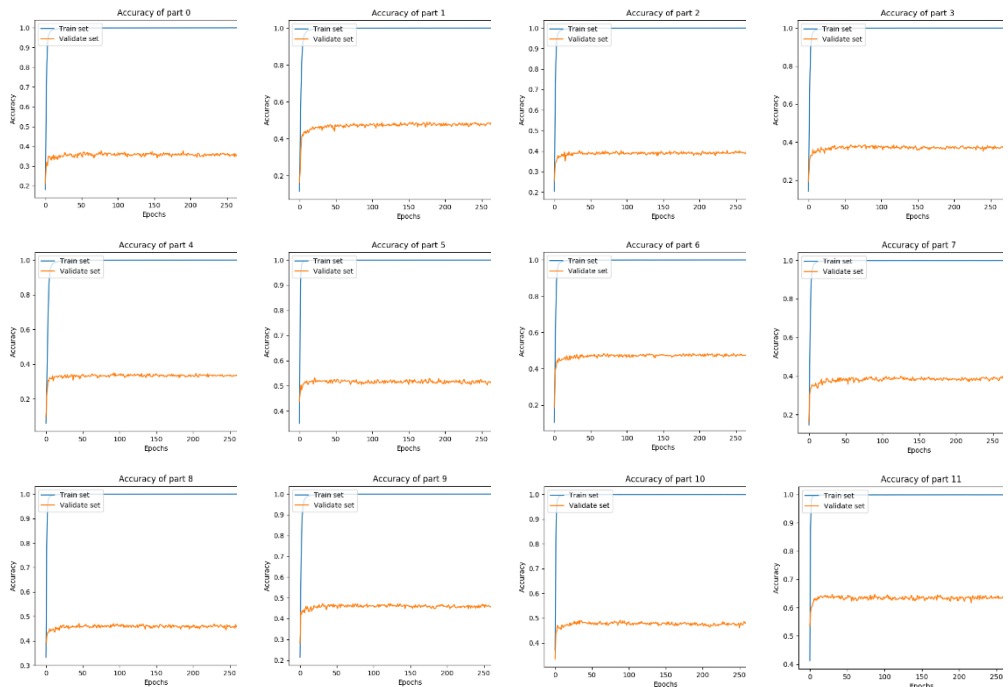


Figure 13. Training and validation accuracy for each weak information part in the VGG dataset, displayed sequentially.

Table 7. set accuracy and coefficients for weak information parts on the VGG dataset.

Part	0	1	2	3	4	5
Accuracy on validate set	37.731%	49.038%	40.269%	38.615%	34.962%	53.192%
Coefficient	<u>37731</u>	<u>49038</u>	<u>40269</u>	<u>38615</u>	<u>34962</u>	<u>53192</u>
	<u>550384</u>	<u>550384</u>	<u>550384</u>	<u>550384</u>	<u>550384</u>	<u>550384</u>
Part	6	7	8	9	10	11
Accuracy on validate set	48.462%	39.885%	47.115%	47.308%	49.115%	64.692%
Coefficient	<u>48462</u>	<u>39885</u>	<u>47115</u>	<u>47308</u>	<u>49115</u>	<u>64692</u>
	<u>550384</u>	<u>550384</u>	<u>550384</u>	<u>550384</u>	<u>550384</u>	<u>550384</u>

D. Result

To build a strong classification base on weak classifications, we applied late fusion of weak information parts by a voting method. Suppose we had k weak information parts on a test face, then for each weak information part k_i , we selected $m_i = 5$ and applied to equation (6) to get the set O_i . This set contained five identities with weak information part i that most similar to weak information part i of test face, using probability vector returned from the Softmax layer in the network. Finally, we scored each identity by using equation (7) and concluded that the test face belongs to the identity with the highest score. We experimented with our method on the face with enough 12 weak information parts and just a few parts. The result for cases in Table 3 shown in Table 8. Generally, we experimented with any k weak information parts and archived the average accuracy given a test face contain k weak information part in Table 9. Compared to the tiny FaceScrub dataset, the tiny VGG dataset's accuracy was lower because each class on the tiny VGG may contained face images from another identity, which could cause noise due to the training process.

Table 8. Experimental results for scenarios detailed in Table 3, across the FaceScrub and VGG datasets

Weak information parts on the face	FaceScrub	VGG
All 12	91.6346%	76.423%
1, 5, 9, 11	88.0769%	71.231%
0, 1, 2, 8, 9, 10	81.8269%	67.500%
4, 5, 6	69.0385%	58.192%
3, 7, 11	83.9423%	66.346%
0, 6, 7	55.4808%	54.346%
2, 3, 4	57.5962%	49.692%

We used a late fusion approach, which involved combining weak information parts through a voting procedure, in order to generate a robust classifier from weak classifications. To build the set O_i , we used equation (6) to choose the top five identities for each part k_i , assuming that the test face contained k weak information parts. This allowed us to form the set O_i . The probability vectors in this set came from the Softmax layer of the network that was trained on the weak information part i . These identities were very close to the real label of the test face. Then, using equation (7), each identity was assigned a score, and the test face was ultimately recognized as the identity with the highest score. Our technique was evaluated on faces that contained all twelve weak information parts as well as those that contained only a subset of those parts. The results for individual situations are shown in Table 8. Table 9 shows the average accuracies obtained across different setups of k weak information parts. The accuracy of the tiny VGG dataset was significantly lower than that of the tiny FaceScrub dataset, which was due to

the presence of images from other identities within each identity, introducing noise and potentially affecting training efficacy.

Table 9. Average accuracy given a face image has k weak information parts.

Number of weak information parts (k)	FaceScrub	VGG
1	54.5593%	45.8654%
2	65.4560%	53.6964%
3	73.0039%	59.8802%
4	78.0027%	64.1960%
5	81.4535%	67.2827%
6	83.9780%	69.5785%
7	85.9229%	71.3441%
8	87.4660%	72.7512%
9	88.7426%	73.9035%
10	89.8412%	74.8601%
11	90.8013%	75.6667%
12	91.6346%	76.4231%

5. Conclusion

In conclusion, this work presented a cutting-edge facial recognition technique developed specifically to address the challenges posed by incomplete facial features. Our method, which improves the accuracy of face recognition even when there is little facial information available or when the images are subject to variations, lighting irregularities, or angular distortions, is based on the strategic aggregation of weak information parts within frontal facial images. Using a late fusion voting mechanism, we systematically combined the classification results of numerous weak information parts, considerably improving the overall accuracy of the facial recognition system. The effectiveness of our approach was thoroughly tested on the tiny FaceScrub and tiny VGG datasets, which are subsets of the larger FaceScrub and VGG collections, respectively. The results of our research highlight the potential of utilizing weak information parts in order to improve the accuracy of facial recognition. By taking into account all 12 weak information locations, our algorithm was able to attain an accuracy of up to 91.6346% on the tiny FaceScrub dataset and 76.4231% on the tiny VGG dataset. Even with a single weak information part, the system was able to achieve respectable accuracies of 54.5593% on the little FaceScrub dataset and 45.8654% on the tiny VGG dataset. The results show that our suggested framework can withstand typical obstacles in the recognition of faces such as occlusion, illumination changes, and facial accessories. In the future, the methodology described here provides new opportunities to improve face recognition systems, particularly in situations where conventional methods can fail because of insufficient facial features.

6. References

- [1]. V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1867-1874, 2014.
- [2]. Z. Li, J. I. Imai and M. Kaneko, "Robust face recognition using block-based bag of words," in *2010 20th International Conference on Pattern Recognition*, 2010, August.
- [3]. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, pp. 2274-2282, 2012.
- [4]. S. Liao, A. K. Jain and S. Z. Li, "Partial face recognition: Alignment-free approach," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 1193-1205, 2012.

- [5]. R. Weng, J. Lu, J. Hu, G. Yang and Y. P. Tan, "Robust feature set matching for partial face recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.
- [6]. K. Cao, Y. Rong, C. Li, X. Tang and C. Change Loy, "Pose-robust face recognition via deep residual equivariant mapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [7]. F. Schroff, D. Kalenichenko and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.
- [8]. Y. Wen, K. Zhang, Z. Li and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*, 2016, October.
- [9]. W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj and L. Song, "SphereFace: Deep Hypersphere Embedding for Face Recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [10]. J. Deng, J. Guo, N. Xue and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," 2019
- [11]. N. V. Danh, V. H. Trong and P. T. Bao, "Developing a late fusion of multi facial components for facial recognition with a voting method and global weights," *International Journal of Computational Vision and Robotics*, vol. 13, no. 6, pp. 619-640, 2023.
- [12]. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.
- [13]. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [14]. M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.
- [15]. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International conference on machine learning*, 2015.
- [16]. H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," *Image Processing (ICIP), 2014 IEEE International Conference on*, 2014.
- [17]. University of Washington, "MegaFace and MF2: Million-Scale Face Recognition," [Online]. Available: <http://megaface.cs.washington.edu/>.
- [18]. Visual Geometry Group, "VGG Face Dataset," 4 September 2015. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/data/vgg_face/.
- [19]. O. M. Parkhi, A. Vedaldi and A. Zisserman, "Deep Face Recognition," *British Machine Vision Conference*, 2015.
- [20]. Stanford University, "CS231n: Convolutional Neural Networks for Visual Recognition," Spring 2017. [Online]. Available: <http://cs231n.github.io/convolutional-networks/#layers>.
- [21]. "Keras: The Python Deep Learning library," [Online]. Available: <https://keras.io/>.
- [22]. "TensorFlow: An open-source machine learning framework for everyone," [Online]. Available: <https://www.tensorflow.org/>.
- [23]. N. Abudarham, L. Shkiller and G. Yovel, "Critical features for face recognition," *Cognition*, pp. 73-83, 2019.



Vo Hoang Trong received the Bachelor's in Computer Science from Faculty of Mathematics and Computer Science, University of Science - VNUHCM, in 2017, and the M.Eng. and Doctoral degree from Chonnam National University, Korea, in 2020 and 2023. Since 2023, he has been a member of DFocus company in data solution, Korea. His research interests are Deep Neural Networks, Pattern Recognition, and Machine Learning.



Pham The Bao is Associate Professor and the Dean of Information Science Faculty, Sai Gon University. Dr. Pham teaches and conducts research in the areas of image processing, pattern recognition, and intelligent computing. His Lab is Intelligent Computing – Image Processing (IC-IP Lab). Dr. Pham has BS in Pure Mathematics (The field is Algebra – Topological Geometry) from University of HCM City and MS in Computer Science from University of Natural Science – National University of HCM City, Vietnam in 1995 and 2000. After getting BS, he became lecturer of Mathematics and Computer Science Faculty to 2018. He completed doctoral studies and earned a PhD in Computer Science at University of Science – National University of HCM City, Vietnam in 2009. From 2007 to 2016, he is Vice Dean of Mathematics & Computer Science Faculty. And from 2004 to 2018, he is head of Computer Science Department, University of Science, Vietnam. He moved to Sai Gon University from 2019.