electronics

MDPI

Article

Few-Shot Learning for Multi-POSE Face Recognition via Hypergraph De-Deflection and Multi-Task Collaborative Optimization

Xiaojin Fan [1], Mengmeng Liao [2,*], Lei Chen [3] and Jingjing Hu [1,*]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2] School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China
[3] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
* Correspondence: mmliao16@fudan.edu.cn (M.L.); hujingjing@bit.edu.cn (J.H.)

Abstract: Few-shot, multi-pose face recognition has always been an interesting yet difficult subject in the field of pattern recognition. Researchers have come up with a variety of workarounds; however, these methods make it either difficult to extract effective features that are robust to poses or difficult to obtain globally optimal solutions. In this paper, we propose a few-shot, multi-pose face recognition method based on hypergraph de-deflection and multi-task collaborative optimization (HDMCO). In HDMCO, the hypergraph is embedded in a non-negative image decomposition to obtain images without pose deflection. Furthermore, a feature encoding method is proposed by considering the importance of samples and combining support vector data description, triangle coding, etc. This feature encoding method is used to extract features from pose-free images. Last but not the least, multi-tasks such as feature extraction and feature recognition are jointly optimized to obtain a solution closer to the global optimal solution. Comprehensive experimental results show that the proposed HDMCO achieves better recognition performance.

Keywords: few-shot learning; face recognition; pose variations; hypergraph

Citation: Fan, X.; Liao, M.; Chen, L.; Hu, J. Few-Shot Learning for Multi-POSE Face Recognition via Hypergraph De-Deflection and Multi-Task Collaborative Optimization. *Electronics* **2023**, *12*, 2248. https://doi.org/10.3390/electronics12102248

Academic Editor: George A. Papakostas

Received: 28 March 2023
Revised: 6 May 2023
Accepted: 10 May 2023
Published: 15 May 2023

1. Introduction

Face recognition is a very important technology with a wide range of applications, such as video surveillance, forensics, and security [1–3]. Pose change is one of the difficulties in face recognition. Posture changes involved in images can cause images of one person to look like images of other people. That is to say, the change in pose will lead to an increase in intra-class difference and a decrease in inter-class difference, which will hinder the classifier from correctly recognizing the face images. One study shows that the performance of most algorithms decreases by more than 10% from frontal-frontal to frontal-profile face verification; however, there is only a small drop in the recognition performance of the human eye [4]. Therefore, it is of great significance to study face recognition involving pose change.

Many methods have been proposed to solve the multi-pose face recognition problem [5–11]. These methods can be divided into the following categories: face normalization, feature representation, spatial mapping, and pose estimation.

The method based on face normalization can better identify the image by normalizing the image with attitude deflection to the front image or the image close to the front image. For example, Ding et al. [12] transform images with pose deflection into frontal images by pose normalization. Luan et al. [13] take geometry preservation into account in GAN networks and exploit perceptual loss constraints along with norm loss to obtain the frontal images that preserve global and local information. Liu et al. [14] use pixel-level loss, feature space perception loss, and identity-preserving loss to generate real class-invariant frontal

images. Yin et al. [15] embed the contextual dependency and the local consistency into GAN networks to extract the frontal images. Lin et al. [16] use the deep representation alignment network to extract the pose-invariant face feature. Yang et al. [17] use the multi-bit binary descriptor to extract the pose-invariant feature. Tu et al. [18] jointly optimize image inpainting and image frontalization to deal with the recognition of low-resolution face images involving pose variations.

Learning the effective feature representations of images can be beneficial for tackling the task of classification. For example, Zhou et al. [19] use the divide-and-strategy to deal with the representation and classification of samples, which can reduce the challenge of posture. Zhang et al. [20] use locality-constrained and label information to enhance the representational power of regression-based methods. Gao et al. [21] use the multi-modal hashing and discriminative correlation maximization analysis for feature representation learning to allow them to obtain the easily distinguishable feature representation of each pose image. Yang et al. [22] learn the more discriminative feature representations by imposing penalties on weighted vectors. Huang et al. [23] use the samples and feature centers to enhance the similarity of features between samples of the same class.

The method based on spatial mapping can reduce the intra-class differences and increase the inter-class differences by mapping samples into a new space, which is beneficial for classification. For example, He et al. [24] use the identity consistency loss and the pose-triplet loss to minimize the intra-class and maximize the inter-class. Wang et al. [25] use the divergence loss to increase the diversity among multiple attention maps. Furthermore, the attention sparsity loss is used to highlight the regions with strong discriminative power. He et al. [26] reduce the difference between images with different modes by applying adversarial learning to both image-level and feature level. Liu et al. [27] use the source domain data to improve the performance of target domain data so that the poses of two images with the same category from the source and target domains are markedly different. Sun et al. [28] use the equalized margin loss to reduce the impact of unevenly distributed data (uneven distribution of attitude deflection).

It is also a good way to estimate the attitude deflection angle of the image and use the information of the deflection angle to recognize the image. For example, Zhang et al. [29] use the pose-guided margin loss to estimate the head poses, then the recognition process can be completed in the same pose. Badave et al. [30] use multiple cameras for pose estimation and then use the estimated pose to recognize the face images. Wang et al. [31] combine the learned sub-classifiers into a classifier with a strong performance by learning the dictionaries and the sub-classifiers at the same time.

The above methods have good effects on face recognition involving small posture deflection or face recognition with a large number of samples. However, most face recognition is either few-shot face recognition or face recognition involving large pose deflection. It is difficult for these methods to learn the intrinsic relationship between multiple samples of the same category in the process of model learning, and the essential attributes of the samples of the same category summarized by the learned model are incomplete or inaccurate.

Hypergraphs can represent complex relationships between objects. Unlike ordinary graphs (where each edge of an ordinary graph can only connect two nodes), each edge of a hypergraph can connect multiple nodes. That is, a hypergraph can reveal complex relationships between multiple nodes. Furthermore, non-negative matrix factorization is widely used in the field of computer vision, such as feature extraction. A matrix can be decomposed into two matrices with different properties by non-negative matrix factorization. Inspired by non-negative matrix factorization, we can decompose each image involving attitude changes through non-negative matrix decomposition, and one matrix obtained by the decomposition is used as the image without attitude deflection, and the other matrix is used as the attitude change matrix. The image without pose deflection is finally obtained through multiple iterative decompositions. Inspired by the hypergraph, we treat each image as a node in the hypergraph and embed the hypergraph formed by

multiple images into a non-negative matrix decomposition to extract images with better performance and no attitude deflection. A few-shot multi-pose face recognition method based on hypergraph de-deflection and multi-task collaborative optimization (HDMCO) is proposed in this paper. First, HDMCO uses the hypergraph and non-negative matrix factorization to obtain the images that are approximately frontal. Then, a novel feature encoding method based on the improved support vector data description is proposed, and it is jointly optimized with a dictionary learning-based classifier for feature extraction and feature classification. Figure 1 shows the flowchart of the proposed HDMCO.

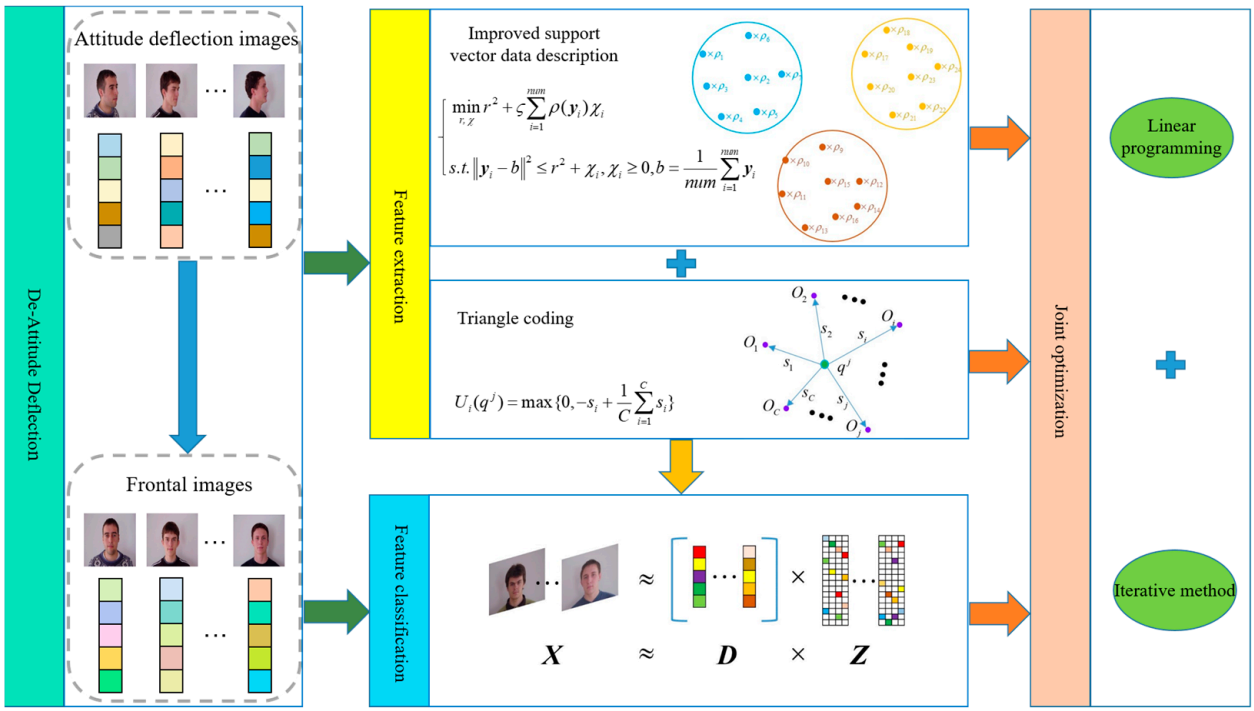


Figure 1. The flowchart of the proposed HDMCO. s_i represents the distance between the patch q^j and O_i, O_i is the center of the i -th SVDD sphere, $i = 1, 2, \cdots, C$. $\boldsymbol{X}$ is the training. Sample set, $\boldsymbol{D}$ is the dictionary, $\boldsymbol{Z}$ is the representation coefficient matrix.

In the de-attitude deflection phase of Figure 1, the image without attitude deflection is separated from the image with attitude deflection by non-negative matrix decomposition. In this process, the hypergraph is embedded in the non-negative decomposition to protect the structural information of the image. In the feature extraction phase, the improved support vector data description is used to obtain the clustering center and radius of each cluster, and triangle coding is used to encode features for each patch. Then, image coding can be obtained. In the feature classification phase, the dictionary learning-based classifier recognizes the features of the image and then determines the category of the image.

The main innovations of this paper are as follows.

(1) A novel multi-pose face recognition framework based on hypergraph de-deflection is proposed. The framework first isolates the pose-free deflection images, then utilizes the proposed feature coding method based on improved support vector data description to extract the features of the pose-free deflection images, and recognizes the extracted features.

(2) A new feature encoding method based on improved support vector data description is proposed. The feature encoding method utilizes the improved support vector data description and triangle encoding to make the extracted features more discriminative.

(3) An effective feature extraction and feature classification optimization model is constructed, which makes it easy to obtain a solution closer to the global optimum and helps to improve the recognition performance of the algorithm.

The subsequent sections of this article are arranged as follows: Section 2 introduces related studies. Section 3 describes the proposed method. Section 4 outlines the details of the experiments, and Section 5 presents the conclusion.

2. Relate Studies

This section will introduce some theories related to the proposed method. Specifically, few-shot face recognition, non-negative matrix factorization, and hypergraph theory will be introduced in turn.

2.1. Few-Shot Face Recognition

Few-shot face recognition has always been an interesting yet difficult research topic. Few-shot learning provides an effective solution to the very relevant and unavoidable problem of data scarcity in many applications. Prior knowledge is applied to small datasets so that few-shot learning can be generalized to new tasks and samples [32].

Researchers have proposed many methods to solve the problem of few-shot face recognition by using few-shot learning [33–35]. Masi et al. [36] propose the pose-aware model (PAM). PAM uses multiple networks to synthesize various pose images and uses the synthesized pose images to train the model to improve its recognition ability. However, this method needs a large amount of memory to store a large number of training images when using a variety of networks to generate a large number of images of various poses, so it is difficult to carry out during the actual process. Elharrouss et al. [37] propose the cascade networks (abbreviated as MCN) corresponding to multiple tasks to enhance the recognition ability of the recognition network for images involving pose variations. However, the diversity of attitude changes considered by this method is limited during model training, so the learned model is invalid when processing images involving other pose changes. Liu et al. [38] use multiple profile images to generate frontal images and use the Siamese network to learn the depth representation of the generated frontal images. The depth representation of the images is more easily recognized by the classifier, which helps to improve the recognition rate of the algorithm. Tao et al. [39] use the identity information of the images and the latent relationship between the frontal and profile images to model the distribution of the profile images and reduce the difference between the profile images and the frontal images. However, it is difficult to judge whether the underlying relationship between the frontal and profile images used is correct and comprehensive. Gao et al. [40] propose a multilayer locality-constrained structural orthogonal Procrustes regression (MLCSOPR) and use MLCSOPR to extract pose-robust features. This method only considers the horizontal change in the posture, but in practice, the image involves both the horizontal and vertical changes of the posture, so the application scope of this method is very narrow.

2.2. Non-Negative Matrix Factorization

Given any non-negative matrix $\boldsymbol{X}_0$, it can be decomposed into two non-negative matrices $\overset{\leftrightarrow}{\boldsymbol{Y}}$ and $\boldsymbol{P}^T$.

$$\begin{cases} \min\limits_{\overset{\leftrightarrow}{\boldsymbol{Y}},\boldsymbol{P}^T} \|\boldsymbol{X}_0 - \overset{\leftrightarrow}{\boldsymbol{Y}}\boldsymbol{P}^T\|_F^2 \\ s.t. \overset{\leftrightarrow}{\boldsymbol{Y}} \geq 0, \boldsymbol{P} \geq 0 \end{cases} \tag{1}$$

where $\boldsymbol{X}_0 \in \Re^{m \times n}$ is the non-negative matrix, $\overset{\leftrightarrow}{\boldsymbol{Y}} \in \Re^{m \times r}$ is the basis matrix, $\boldsymbol{P}^T \in \Re^{r \times n}$ is the submatrix.

Then, $\overleftrightarrow{\boldsymbol{Y}}$ and $\boldsymbol{P}^T$ can be updated by

$$\begin{cases} \overleftrightarrow{\boldsymbol{Y}}_{ij} \leftarrow \overleftrightarrow{\boldsymbol{Y}}_{ij} \frac{(\boldsymbol{XP})_{ij}}{(\overleftrightarrow{\boldsymbol{Y}}\boldsymbol{P}^T\boldsymbol{P})_{ij}} \\ (\boldsymbol{P}^T)_{jk} \leftarrow (\boldsymbol{P}^T)_{jk} \frac{(\overleftrightarrow{\boldsymbol{Y}}^T\boldsymbol{X})_{jk}}{(\overleftrightarrow{\boldsymbol{Y}}^T\boldsymbol{Y}\boldsymbol{P}^T)_{jk}} \end{cases} \tag{2}$$

2.3. Hypergraph Theory

A hypergraph is very helpful for maintaining the internal structure of the data. Next, we will briefly introduce the hypergraph theory.

Hypergraph is defined as follows: Hypergraph G is an ordered binary group $G = (V, e)$, where V is a non-empty set with nodes or vertices as elements, which is called vertex set; e is a cluster of non-empty subsets whose elements are called hyperedges. Unlike ordinary graphs, each edge of the hypergraph can connect not only two vertices but also more vertices. Here, the hypergraph is undirected.

Given a hypergraph $G = (V, e)$, $V = \{v_1, v_2, \cdots, v_k\}$ is a set of finite data points, $v_i (i = 1, 2, \cdots, k)$ is a vertex. $e = \{e_1, e_2, \cdots, e_t\}$ is the set of hyperedges, e_j is a hyperedge.

The hyperedge set e satisfies the following two conditions:

(a) $e_j \notin \phi$, $j = 1, 2, \cdots, t$;

(b) $e_1 \cup e_2 \cup e_3 \cdots \cup e_t = V$;

Each hyperedge e_j has a corresponding weight w_j. Vertices hyperedges will form an association matrix $\boldsymbol{H} \in \Re^{|V| \times |e|}$, any element in $\boldsymbol{H}$ can be calculated by Equation (3):

$$\boldsymbol{H}_{ij} = \begin{cases} 1, v_i \in e_j \\ 0, v_i \notin e_j \end{cases} \tag{3}$$

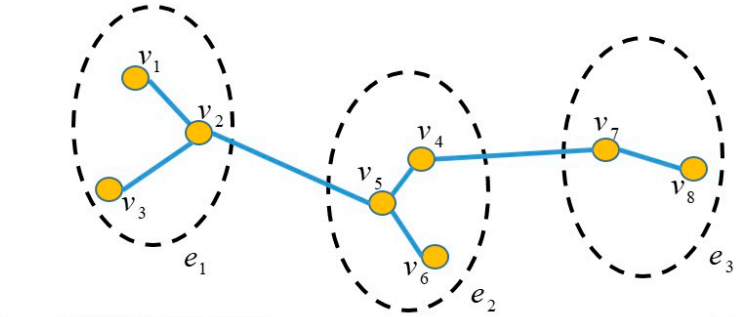
To better understand the hypergraph theory, we take the hypergraph in Figure 2 as an example to illustrate the knowledge of the hypergraph. In Figure 2, the set of all vertices is denoted as $V = \{v_1, v_2, \cdots, v_8\}$, $e_1 = \{v_1, v_2, v_3\}$, $e_2 = \{v_4, v_5, v_6\}$ and $e_3 = \{v_7, v_8\}$ denote the three hyperedges of G. The set of all hyperedges is denoted as $e = \{e_1, e_2, e_3\}$. The value of each element in $\boldsymbol{H}$ can be obtained according to Equation (3) and shown in Figure 2. Each image serves as a data point and becomes a vertex in the hypergraph. Hyperedges are composed of several similar data points. Similar data points indicate images in which the contents of the images appear to be relatively close, such as two images of the same person with small differences in attitude.

The degree d_i of each vertex in the hypergraph is defined as the sum of the weights of the hyperedges to which it belongs, and the degree ρ_i of the hyperedges is defined as the number of nodes to which the hyperedge belongs. d_i and ρ_i are calculated as follows:

$$\begin{cases} d_i = \sum_{j=1}^{t} w_j \boldsymbol{H}_{ij} \\ \rho_i = \sum_{i=1}^{k} \boldsymbol{H}_{ij} \end{cases} \tag{4}$$

Let $\boldsymbol{D}_v$ denotes a diagonal matrix, whose main diagonal elements are $\boldsymbol{D}_{v_{ii}} = d_i$, where $i = 1, 2, \cdots, k$. Similarly, let $\boldsymbol{D}_e$ and W be the diagonal matrices generated by ρ_j and w_j, respectively, where $j = 1, 2, \cdots, t$. Then, the non-regularized hypergraph Laplacian matrix can be calculated by Equation (5).

$$\boldsymbol{L}^{\boldsymbol{H}} = \boldsymbol{D}_v - \boldsymbol{HWD}_e^{-1}\boldsymbol{H} \tag{5}$$



	e_1	e_2	e_3
v_1	1	0	0
v_2	1	0	0
v_3	1	0	0
v_4	0	1	0
v_5	0	1	0
v_6	0	1	0
v_7	0	0	1
v_8	0	0	1

$$\boldsymbol{H}_{ij} = \begin{cases} 1, & v_i \in e_j \\ 0, & v_i \notin e_j \end{cases}$$

$\boldsymbol{H}$

$\boldsymbol{H}_{11}$=1	$\boldsymbol{H}_{12}$=0	$\boldsymbol{H}_{13}$=0
$\boldsymbol{H}_{21}$=1	$\boldsymbol{H}_{22}$=0	$\boldsymbol{H}_{23}$=0
$\boldsymbol{H}_{31}$=1	$\boldsymbol{H}_{32}$=0	$\boldsymbol{H}_{33}$=0
$\boldsymbol{H}_{41}$=0	$\boldsymbol{H}_{42}$=1	$\boldsymbol{H}_{43}$=0
$\boldsymbol{H}_{51}$=0	$\boldsymbol{H}_{52}$=1	$\boldsymbol{H}_{53}$=0
$\boldsymbol{H}_{61}$=0	$\boldsymbol{H}_{62}$=1	$\boldsymbol{H}_{63}$=0
$\boldsymbol{H}_{71}$=0	$\boldsymbol{H}_{72}$=0	$\boldsymbol{H}_{73}$=1
$\boldsymbol{H}_{81}$=0	$\boldsymbol{H}_{82}$=0	$\boldsymbol{H}_{83}$=1

Figure 2. Examples of the hypergraph G. v_i is the vertex, $i = 1, 2, \cdots, 8$. e_j is the hyperedge, $j = 1, 2, 3$. $\boldsymbol{H}$ is the association matrix, $\boldsymbol{H}_{ij}$ is the element in row i and column j in $\boldsymbol{H}$.

3. Proposed Method

In this section, we introduce the proposed method (the few-shot, multi-pose face recognition method based on hypergraph de-deflection and multi-task collaborative optimization). The main idea of the proposed method is as follows. First, we propose a feature discrimination enhancement method based on non-negative matrix factorization and hypergraph embedding and use it to extract near-frontal images from pose-deflected images. After that, we propose a feature encoding method based on improved support vector data description and use it to extract the distinguishing features. Meanwhile, those distinguishing features are classified by the dictionary learning-based classifier. When performing feature extraction and feature classification, these two processes are jointly optimized. Hence, we mainly introduce the feature discrimination enhancement method based on non-negative matrix factorization and hypergraph embedding, feature encoding method, dictionary learning-based classifier, joint optimization of the feature extraction, and feature classification.

3.1. Feature Discrimination Enhancement Method Based on Non-Negative Matrix Factorization and Hypergraph Embedding

Suppose a given dataset is denoted as $\boldsymbol{Y} \in \Re^{m \times n}$, and each column in $\boldsymbol{Y}$ represents an image sample. First, we apply a Gaussian filter to each image in $\boldsymbol{Y}$ to remove the noise in the image. Next, we check whether the pixel value of each image is negative, change the negative value to 0 for the negative values, and keep the original value for the positive values, then obtain $\boldsymbol{Y}^W$. After that, we construct the deregularized hypergraph Laplacian matrix $\boldsymbol{L}^H$ of $\boldsymbol{Y}^W$. Assume that the number of hyperedges is t, the number of data in the hypergraph is N and t is equal to N. The number of vertices contained in each hyperedge is s. The vertices contained in each hyperedge are generated by $\boldsymbol{Y}_n^W$ itself and its nearest $s-1$

neighbors, where $\boldsymbol{Y}_n^W$ is the n^{th} column of $\boldsymbol{Y}^W$. $\boldsymbol{L}^H$ is calculated according to Equation (5), where w_j can be calculated by Equation (6).

$$w_j = \sum_{\boldsymbol{Y}_{n_1}^W, \boldsymbol{Y}_{n_2}^W \in e_j} \exp\left(-\frac{\|\boldsymbol{Y}_{n_1}^W - \boldsymbol{Y}_{n_2}^W\|}{\delta^2}\right) \tag{6}$$

where $\delta = \frac{1}{s \times t} \sum_{j=1}^{t} \sum_{\boldsymbol{Y}_{n_1}^W, \boldsymbol{Y}_{n_2}^W \in e_j} \|\boldsymbol{Y}_{n_1}^W - \boldsymbol{Y}_{n_2}^W\|$.

After $\boldsymbol{Y}^W$ and $\boldsymbol{L}^H$ are obtained, the objective function is as follows.

$$\begin{cases} \min \|\boldsymbol{Y}^W - \overset{\leftrightarrow}{\boldsymbol{Y}} \boldsymbol{P}^T\|_F^2 + \lambda Tr(\boldsymbol{P}^T \boldsymbol{L}^H \boldsymbol{P}) \\ s.t. \overset{\leftrightarrow}{\boldsymbol{Y}} \geq 0, \boldsymbol{P} \geq 0 \end{cases} \tag{7}$$

where $\boldsymbol{Y}^W \in \Re^{m \times n}$, $\overset{\leftrightarrow}{\boldsymbol{Y}} \in \Re^{m \times n}$, $\boldsymbol{P} \in \Re^{n \times n}$, $\boldsymbol{L}^H \in \Re^{n \times n}$, $\|\boldsymbol{Y}^W - \overset{\leftrightarrow}{\boldsymbol{Y}} \boldsymbol{P}^T\|_F^2$ represents the error resulting from the non-negative decomposition of $\boldsymbol{Y}^W$. $Tr(\boldsymbol{P}^T \boldsymbol{L}^H \boldsymbol{P})$ is the hypergraph regular term, which can protect the local geometric structure of the data and improve the performance of the algorithm. The value of λ is set to 0.3.

It is difficult to solve Equation (7) directly, so an iterative solution method is adopted to solve this problem. The Lagrangian function corresponding to Equation (8) is:

$$\Delta = \|\boldsymbol{Y}^W - \overset{\leftrightarrow}{\boldsymbol{Y}} \boldsymbol{P}^T\|_F^2 + \lambda Tr(\boldsymbol{P}^T \boldsymbol{L}^H \boldsymbol{P}) + Tr(\boldsymbol{\Psi} \overset{\leftrightarrow}{\boldsymbol{Y}}^T) + Tr(\boldsymbol{\Phi} \boldsymbol{P}^T) \tag{8}$$

where $\boldsymbol{\Psi}$ is the matrix formed by the Lagrange multipliers of $\boldsymbol{\Psi}_{mk}$ for $\overset{\leftrightarrow}{\boldsymbol{Y}}_{mk} \geq 0$, $\boldsymbol{\Phi}$ is the matrix formed by the Lagrange multipliers of $\boldsymbol{\Phi}_{nk}$ for $\boldsymbol{P}_{mk} \geq 0$.

Δ in Equation (8) can be rewritten as

$$\begin{aligned} &\Delta = Tr(\boldsymbol{Y}^{W^T} \boldsymbol{Y}^W) - Tr(\boldsymbol{Y}^{W^T} \overset{\leftrightarrow}{\boldsymbol{Y}} \boldsymbol{P}^T) - Tr(\boldsymbol{P} \overset{\leftrightarrow}{\boldsymbol{Y}}^T \boldsymbol{Y}^W) \\ &+ Tr(\boldsymbol{P} \overset{\leftrightarrow}{\boldsymbol{Y}}^T \overset{\leftrightarrow}{\boldsymbol{Y}} \boldsymbol{P}^T) + \lambda Tr(\boldsymbol{P}^T \boldsymbol{L}^H \boldsymbol{P}) + Tr(\boldsymbol{\Psi} \overset{\leftrightarrow}{\boldsymbol{Y}}^T) + Tr(\boldsymbol{\Phi} \boldsymbol{P}^T) \\ &= Tr(\boldsymbol{Y}^{W^T} \boldsymbol{Y}^W) - 2Tr(\boldsymbol{P} \overset{\leftrightarrow}{\boldsymbol{Y}}^T \boldsymbol{Y}^W) + Tr(\boldsymbol{P} \overset{\leftrightarrow}{\boldsymbol{Y}}^T \overset{\leftrightarrow}{\boldsymbol{Y}} \boldsymbol{P}^T) \\ &+ \lambda Tr(\boldsymbol{P}^T \boldsymbol{L}^H \boldsymbol{P}) + Tr(\boldsymbol{\Psi} \overset{\leftrightarrow}{\boldsymbol{Y}}^T) + Tr(\boldsymbol{\Phi} \boldsymbol{P}^T) \end{aligned} \tag{9}$$

By taking the partial derivatives of Δ with respect to $\overset{\leftrightarrow}{\boldsymbol{Y}}$ and $\boldsymbol{P}$, respectively, we obtain

$$\begin{cases} \frac{\partial \Delta}{\partial \overset{\leftrightarrow}{\boldsymbol{Y}}} = -2\boldsymbol{Y}^W \boldsymbol{P} + 2\overset{\leftrightarrow}{\boldsymbol{Y}} \boldsymbol{P}^T \boldsymbol{P} + \boldsymbol{\Psi} \\ \frac{\partial \Delta}{\partial \boldsymbol{P}} = -2\boldsymbol{Y}^{W^T} \overset{\leftrightarrow}{\boldsymbol{Y}} + 2\boldsymbol{P} \overset{\leftrightarrow}{\boldsymbol{Y}}^T \overset{\leftrightarrow}{\boldsymbol{Y}} + 2\lambda \boldsymbol{L}^H \boldsymbol{P} + \boldsymbol{\Phi} \end{cases} \tag{10}$$

According to the KKT conditions $\boldsymbol{\Psi}_{mk} \overset{\leftrightarrow}{\boldsymbol{Y}}_{mk} = 0$ and $\boldsymbol{\Phi}_{nk} \boldsymbol{P}_{nk} = 0$, we obtain

$$-(\boldsymbol{Y}^W \boldsymbol{P})_{mk} \overset{\leftrightarrow}{\boldsymbol{Y}}_{mk} + (\overset{\leftrightarrow}{\boldsymbol{Y}} \boldsymbol{P}^T \boldsymbol{P})_{mk} \overset{\leftrightarrow}{\boldsymbol{Y}}_{mk} = 0 \tag{11}$$

$$-(\boldsymbol{Y}^{W^T} \overset{\leftrightarrow}{\boldsymbol{Y}})_{nk} \boldsymbol{P}_{nk} + (\boldsymbol{P} \overset{\leftrightarrow}{\boldsymbol{Y}}^T \overset{\leftrightarrow}{\boldsymbol{Y}})_{nk} \boldsymbol{P}_{nk} + \lambda (\boldsymbol{L}^H \boldsymbol{P})_{nk} \boldsymbol{P}_{nk} = 0 \tag{12}$$

In Equations (11) and (12), the subscript of each variable indicates the number of iterations of the variable.

Then, $\overleftrightarrow{\boldsymbol{Y}}_{mk}$ and $\boldsymbol{P}_{nk}$ can be updated by the following two equations.

$$\overleftrightarrow{\boldsymbol{Y}}_{mk} \leftarrow \overleftrightarrow{\boldsymbol{Y}}_{mk} \otimes \frac{(\boldsymbol{Y}^{W}\boldsymbol{P})_{mk}}{(\overleftrightarrow{\boldsymbol{Y}}\boldsymbol{P}^{T}\boldsymbol{P})_{mk}} \tag{13}$$

$$\boldsymbol{P}_{nk} \leftarrow \boldsymbol{P}_{nk} \otimes \frac{(\boldsymbol{Y}^{W^{T}}\overleftrightarrow{\boldsymbol{Y}})_{nk} + (\lambda \boldsymbol{H}\boldsymbol{W}\boldsymbol{D}_{e}^{-1}\boldsymbol{H}\boldsymbol{P})_{nk}}{(\boldsymbol{P}\overleftrightarrow{\boldsymbol{Y}}^{T}\overleftrightarrow{\boldsymbol{Y}})_{nk} + (\lambda \boldsymbol{D}_{v}\boldsymbol{P})_{nk}} \tag{14}$$

The variables in Equations (13) and (14) have appeared before; please see the previous section for their definitions. The subscript of each variable indicates the number of iterations of the variable. $\otimes$ represents the element-wise multiplication of two matrices. The output $\overleftrightarrow{\boldsymbol{Y}}$ is the image set with almost no attitude deflection. The features of each image with almost no attitude deflection can be obtained by using the proposed feature coding method, which has high-class discrimination.

Figure 3 shows the process of extracting near-frontal images from images involving pose variations. $\boldsymbol{Y}$ represents the original image set involving pose deflection, $\boldsymbol{Y}^{W}$ represents the image set after preprocessing $\boldsymbol{Y}$, $\overleftrightarrow{\boldsymbol{Y}}$ represents the image set of the approximate frontal image obtained by decomposition and iteration, $\boldsymbol{P}$ represents the pose change matrix. In Figure 3, we first preprocess each image in the original image set to obtain a non-negative image set without noise pollution. Then, the hypergraph is embedded into the non-negative matrix factorization to preserve the structure of the decomposed images. Finally, the image set with almost no deflection is obtained through matrix factorization and multiple iterative updates.

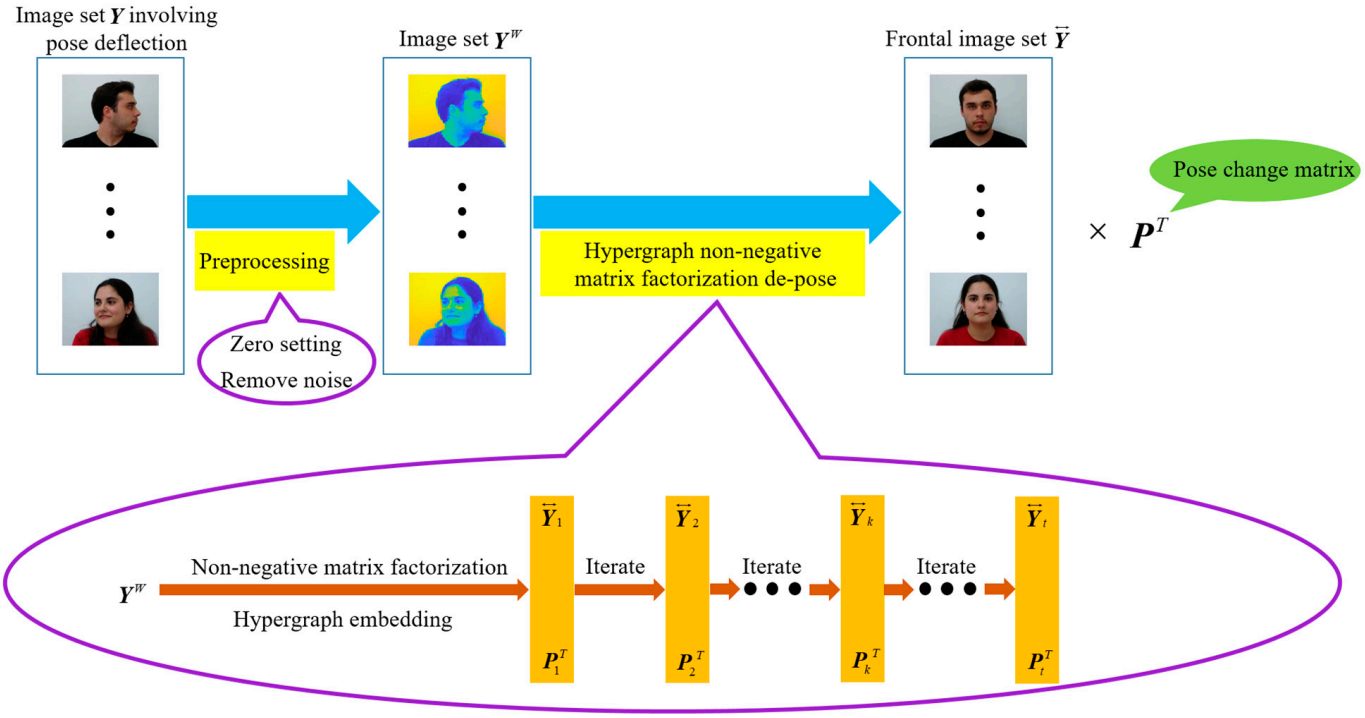


Figure 3. The process of extracting near-frontal images from images involving pose variations. $\boldsymbol{Y}$ is the original image set involving pose deflection, $\boldsymbol{Y}^{W}$ is the image set after preprocessing $\boldsymbol{Y}$, $\overleftrightarrow{\boldsymbol{Y}}$ is the image set of the approximate frontal image obtained by decomposition and iteration, $\boldsymbol{P}$ is pose change matrix, $\boldsymbol{P}^{T}$ is the transpose of $\boldsymbol{P}$, $\overleftrightarrow{\boldsymbol{Y}}_i$ is the value of $\overleftrightarrow{\boldsymbol{Y}}$ at the i -th iteration, $\boldsymbol{P}_i{}^{T}$ is the value of $\boldsymbol{P}^{T}$ at the i -th iteration.

3.2. Feature Coding Method Based on Improved Support Vector Data Description

The main idea of the proposed feature coding method based on improved support vector data description is as follows. First, we propose an improved support vector data description and use it to obtain the sphere center and radius of each cluster. After that, the radius and center of the ball corresponding to each cluster are used for feature encoding. The existing support vector data description considers that each data point plays the same role when calculating the radius of each cluster, which is not in line with reality. Hence, we assign a learned weight to each data in the model learning and propose an improved support vector data description; its model is as follows.

$$\begin{cases} \min\limits_{r,\chi} r^2 + \varsigma \sum\limits_{i=1}^{num} \rho(\boldsymbol{y}_i)\chi_i \\ s.t. \|\boldsymbol{y}_i - b\|^2 \le r^2 + \chi_i, \chi_i \ge 0, b = \frac{1}{num}\sum\limits_{i=1}^{num} \boldsymbol{y}_i \end{cases} \tag{15}$$

where r is the radius of the ball, $\boldsymbol{y}_i$ is the i^{th} sample, $\rho(\boldsymbol{y}_i)$ is the weight of $\boldsymbol{y}_i$, b is the center of the ball, num is the number of the samples, χ_i is the slack variable. ς is a parameter whose value is set to 0.4.

The weight of any sample is calculated as follows.

First, we divide the whole data set into C clusters, and assume that the sample set of the kth cluster is denoted as $\left\{y_1^k, y_2^k, \cdots, y_{P_k}^k\right\}$, where y_i^k is the ith data point in $\left\{y_1^k, y_2^k, \cdots, y_{P_k}^k\right\}$, $i = 1, 2, \cdots, P_k$. P_k is the number of data points in $\left\{y_1^k, y_2^k, \cdots, y_{P_k}^k\right\}$, and $y_i^k = [v_1^{ki}, v_2^{ki}, \cdots, v_d^{ki}]^T \in \Re^{d\times 1}$.

Denote the average distance between two data points in $\left\{y_1^k, y_2^k, \cdots, y_{P_k}^k\right\}$ as m_k.

If the number of data points contained in $\left\{y_1^k, y_2^k, \cdots, y_{P_k}^k\right\}$ is greater than one, then

$$m_k = \frac{2}{p_k(p_k-1)} \sum_{i=1}^{p_k} \sum_{j=i+1}^{p_k} d(y_i^k, y_j^k) \tag{16}$$

$$d(y_i^k, y_j^k) = \sqrt{\left(v_1^{ki} - v_1^{kj}\right)^2 + \left(v_2^{ki} - v_2^{kj}\right)^2 + \cdots + \left(v_d^{ki} - v_d^{kj}\right)^2} \tag{17}$$

If the number of data points contained in $\left\{y_1^k, y_2^k, \cdots, y_{P_k}^k\right\}$ is equal to one, then

$$m_k = \frac{1}{\sum\limits_{i=1, i\neq k}^{C} p_k} \sum_{t=1, t\neq k}^{C} \sum_{i=1}^{P_t} d(y_1^k, y_i^t) \tag{18}$$

Generally speaking, the distances between data points in the same cluster are far less than the distances between data points in different clusters. Thus, we assume that data points in the same cluster have the same weight.

$$\rho_k = 1 - \frac{m_k}{\sum\limits_{i=1}^{C} m_i} \tag{19}$$

The Lagrange function of Equation (15) can be written as

$$\begin{aligned} \widetilde{L}(r, \chi, \alpha, \beta) = r^2 + \varsigma \sum_{i=1}^{num} \rho(\boldsymbol{y}_i)\chi_i + \sum_{i=1}^{num} \alpha_i \\ \left\{ \|\boldsymbol{y}_i - \frac{1}{num}\sum_{j=1}^{num} \boldsymbol{y}_j\|^2 - r^2 - \chi_i \right\} - \sum_{i=1}^{num} \beta_i \chi_i \end{aligned} \tag{20}$$

Let $\frac{\partial \widetilde{L}}{\partial r} = 0$ and $\frac{\partial \widetilde{L}}{\partial \chi_i} = 0$, we can obtain

$$\begin{cases} \min\limits_{\boldsymbol{\alpha}} \frac{2}{num} \boldsymbol{\alpha} \boldsymbol{Q} \boldsymbol{e} - \boldsymbol{\alpha}^T \boldsymbol{\Omega} \\ s.t.\ \boldsymbol{\alpha}^T \boldsymbol{e} = 1 \end{cases} \tag{21}$$

where $\boldsymbol{Q} = (< \boldsymbol{y}_i, \boldsymbol{y}_j >)_{num \times num}$, $\boldsymbol{\Omega} = (< \boldsymbol{y}_i, \boldsymbol{y}_j >)_{num \times 1}$, $e = (1, 1, 1, \cdots, 1)^T$, $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are the i^{th} sample and j^{th} sample in the dataset with attitude deflection removed, respectively, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_{num}]$. $\boldsymbol{\alpha}$ can be obtained by using the linear programming algorithm.

r can be obtained by using Equation (22).

$$r^2 = \boldsymbol{y}_i \cdot \boldsymbol{y}_j - 2 \sum_{i,j=1}^{\Upsilon} \alpha_i (\boldsymbol{y}_i \cdot \boldsymbol{y}_j) + \sum_{i,j=1}^{\Upsilon} \alpha_i \alpha_j (\boldsymbol{y}_i \cdot \boldsymbol{y}_j) \tag{22}$$

where Υ is the set of support vectors, the sample points used in Equation (22) are the support vectors. Whether the data point is a support vector, the following condition needs to be met: if the data point $\boldsymbol{y}_i$ is a support vector, its corresponding α_i is non-zero. $r = [r_1, r_2, \cdots, r_C]$, C is the number of clusters in the dataset.

Then, for each image with pose deflection removed, it is decomposed into $\widetilde{N}$ patches (each patch has the same size), and each patch is encoded. The schematic diagram of the image being divided into small pieces is shown in the Figure 4. For example, for an image q with attitude deflection removed, it is divided into $\widetilde{N}$ small patches. $\widetilde{N}$ is determined by our experience. For any small patch q^j, $j = 1, 2, \cdots, \widetilde{N}$, it can be encoded as $U(q^j)$.

$$U(q^j) = \begin{bmatrix} U_1(q^j) & U_2(q^j) & \cdots & U_C(q^j) \end{bmatrix}^T \tag{23}$$

where $U_i(q^j) = \begin{bmatrix} U_{i,1}(q^j) & U_{i,2}(q^j) \end{bmatrix}$, $i = 1, 2, \cdots, C$, $j = 1, 2, \cdots, \widetilde{N}$, $U_{i,1}(q^j)$ and $U_{i,2}(q^j)$ are obtained by the triangle coding. $U_{i,1}(q^j) = \max\{0, d(s) - s_i(q^j)\}$, $s_i(q^j) = \|q^j - o_i\|_2$ represents the distance from q^j to o_i, $d(s)$ is the mean of all $s_i(q^j)$ values. $U_{i,2}(q^j) = \max\{0, A(m) - m_i(q^j)\}$, $m_i(q^j) = \frac{r_i}{\sum\limits_{k=1}^{C} r_k}$, $A(m)$ is the mean of all m_i values.

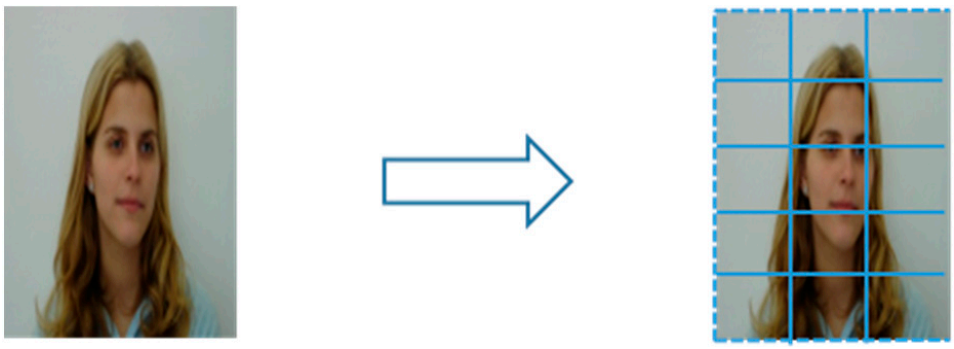


Figure 4. In the picture, each small grid represents a patch. The number of patches is chosen based on our experience.

Figure 5 shows the schematic diagram of the encoding. q^j represents the j^{th} patch of the image q (The image q is divided into $\widetilde{N}$ patches). o_i denotes the center of the SVDD sphere formed by the j^{th} cluster (multiple sample points are clustered into a cluster.), r_i.

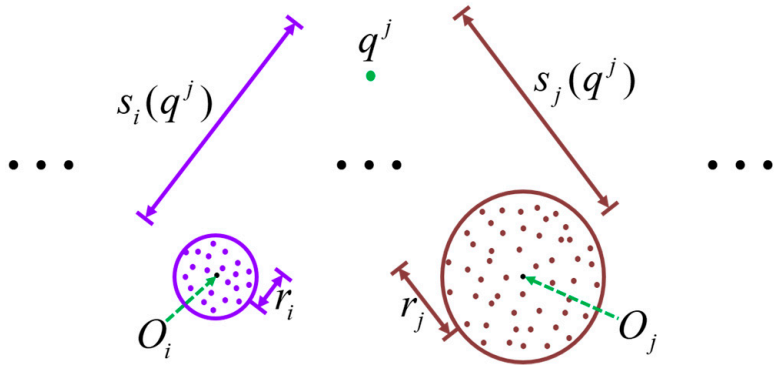


Figure 5. The schematic diagram of the encoding. o_i and o_j are the centers of the i -th and j -th SVDD balls, respectively. r_i and r_j are the radius of the i th and j th SVDD balls, respectively. $s_i(q^j)$ and $s_j(q^j)$ represent the distance from patch q^j to o_i and o_j, respectively. denotes the radius of the SVDD sphere formed by the i^{th} cluster. $s_i(q^j)$ represents the distance between q^j and o_i. o_j denotes the center of the SVDD sphere formed by the j^{th} cluster, r_j denotes the radius of the SVDD sphere formed by the j^{th} cluster. $s_j(q^j)$ represents the distance between q^j and o_j. For the specific encoding of each patch, please refer to Equation (19).

Hence, the image q can be encoded as F_q, and the expression of F_q is as follows.

$$F_q = \left[\begin{array}{cccc} (U(q^1))^T & (U(q^2))^T & \cdots & (U(q^{\tilde{N}}))^T \end{array} \right]^T \tag{24}$$

3.3. Dictionary Learning-Based Classifier

The de-deflection operations and feature encoding operations described above greatly reduce the influence of posture changes on face recognition. To further improve the recognition rate of the whole algorithm on this basis, we decided to learn the classifier, questioning which classifier can not only realize the learning function but also learn the characteristics related to the classified samples in the process of learning. Recent studies have shown that sparse representations have been successfully applied in many fields, such as image restoration and image classification. Dictionaries play an important role in sparse representation, and the quality of dictionaries greatly affects the performance of sparse representation. The latest research on dictionary learning shows that learning a desirable dictionary from the training data itself can usually yield good results for tasks on images or video [41]. Inspired by this, we are ready to learn the dictionary and use the learned dictionary to represent the test samples, and then determine the category of the test samples according to the representation residual.

The basic model of the dictionary learning-based classifier is as follows.

$$\begin{cases} \min\limits_{\boldsymbol{D},\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}\|_F^2 + \eta \|\boldsymbol{Z}\|_1 \\ s.t. \|d_i\|_2^2 \leq 1 \end{cases} \tag{25}$$

where $\boldsymbol{X}$ is the training samples, $\boldsymbol{D}$ represents the dictionary to be learned, $\boldsymbol{Z}$ is the representation coefficient, $\boldsymbol{d}_i$ represents the i^{th} atom in $\boldsymbol{D}$. η is set to 0.3.

3.4. Joint Optimization of the Feature Extraction and Feature Classification

To obtain the globally optimal solution of HDMCO, we jointly optimized the feature extraction and feature classification.

The model for jointly optimizing the feature extraction and feature classification is as follows.

$$\begin{cases} \min\limits_{\boldsymbol{\alpha},\boldsymbol{D},\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}\|_F^2 + \eta \left(\frac{2}{num} \boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{e} - \boldsymbol{\alpha}^T \boldsymbol{\Omega} \right) \|\boldsymbol{Z}\|_1 \\ s.t. \boldsymbol{\alpha}^T \boldsymbol{e} = 1, \|\boldsymbol{d}_i\|_2^2 \leq 1 \end{cases} \tag{26}$$

According to Equation (26), we can obtain $\boldsymbol{\alpha}$, $\boldsymbol{D}$ and $\boldsymbol{Z}$.

$\boldsymbol{\alpha}$ can be obtained by using Equation (27).

$$\begin{cases} \min\limits_{\boldsymbol{\alpha}} \eta(\frac{2}{num}\boldsymbol{\alpha}^T\boldsymbol{Q}\boldsymbol{e} - \boldsymbol{\alpha}^T\boldsymbol{\Omega})\|\boldsymbol{Z}\|_1 \\ s.t.\boldsymbol{\alpha}^T\boldsymbol{e} = 1 \end{cases} \tag{27}$$

Then, the value of $\boldsymbol{\alpha}$ can be obtained by using the linear programming algorithm. $\boldsymbol{D}$ can be obtained by solving Equation (28).

$$\begin{cases} \min\limits_{\boldsymbol{D}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}\|_F^2 \\ s.t.\|\boldsymbol{d}_i\|_2^2 \le 1 \end{cases} \tag{28}$$

Solving Equation (28) can be converted to solving Equation (29).

$$\begin{cases} \boldsymbol{D} = \underset{D}{\operatorname{argmin}} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}\|_F^2 + \vartheta\|\boldsymbol{D} - \boldsymbol{V} + \boldsymbol{J}\|_F^2 \\ \boldsymbol{V} = \underset{V}{\operatorname{argmin}} \vartheta\|\boldsymbol{D} - \boldsymbol{V} + \boldsymbol{J}\|_F^2, s.t.\|v^i\|_2^2 \le 1 \\ \boldsymbol{J} = \boldsymbol{J} + \boldsymbol{D} - \boldsymbol{V} \end{cases} \tag{29}$$

where ϑ is set to 0.2.

Then, $\boldsymbol{D}$ can be obtained by iteratively solving the variables in Equation (29). $\boldsymbol{Z}$ can be obtained by solving Equation (30).

$$\min_{Z} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}\|_F^2 + \eta(\frac{2}{num}\boldsymbol{\alpha}^T\boldsymbol{Q}\boldsymbol{e} - \boldsymbol{\alpha}^T\boldsymbol{\Omega})\|\boldsymbol{Z}\|_1 \tag{30}$$

The solution to Equation (30) is as follows.

$$\boldsymbol{Z} = \text{shrink}(\boldsymbol{D}^{-1}\boldsymbol{X}, \frac{\eta(\frac{2}{num}\boldsymbol{\alpha}^T\boldsymbol{Q}\boldsymbol{e} - \boldsymbol{\alpha}^T\boldsymbol{\Omega})}{2}) \tag{31}$$

where $\text{shrink}(x, a) = \text{signmax}(|x| - a, 0)$.

Figure 6 shows the schematic diagram of seeking a globally optimal solution.

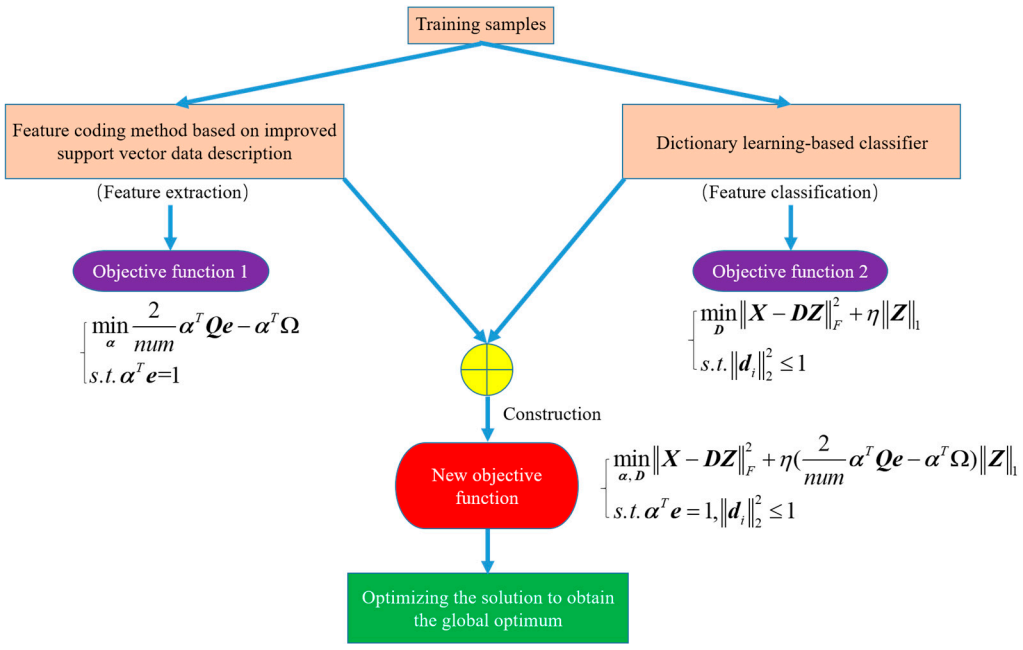


Figure 6. The schematic diagram of seeking a globally optimal solution.

4. Experiments

4.1. Dataset

Here, Multi-PIE [42], MegaFace [43], CAS-PEAL [44], YTF [45], CPLFW [46], and CVL [47] are used in experiments to verify the performance of HDMCO.

Multi-PIE mainly involves pose variations and illumination variations, and includes a total of more than 750,000 images of 337 different people. Figure 7a shows some samples of multi-PIE.

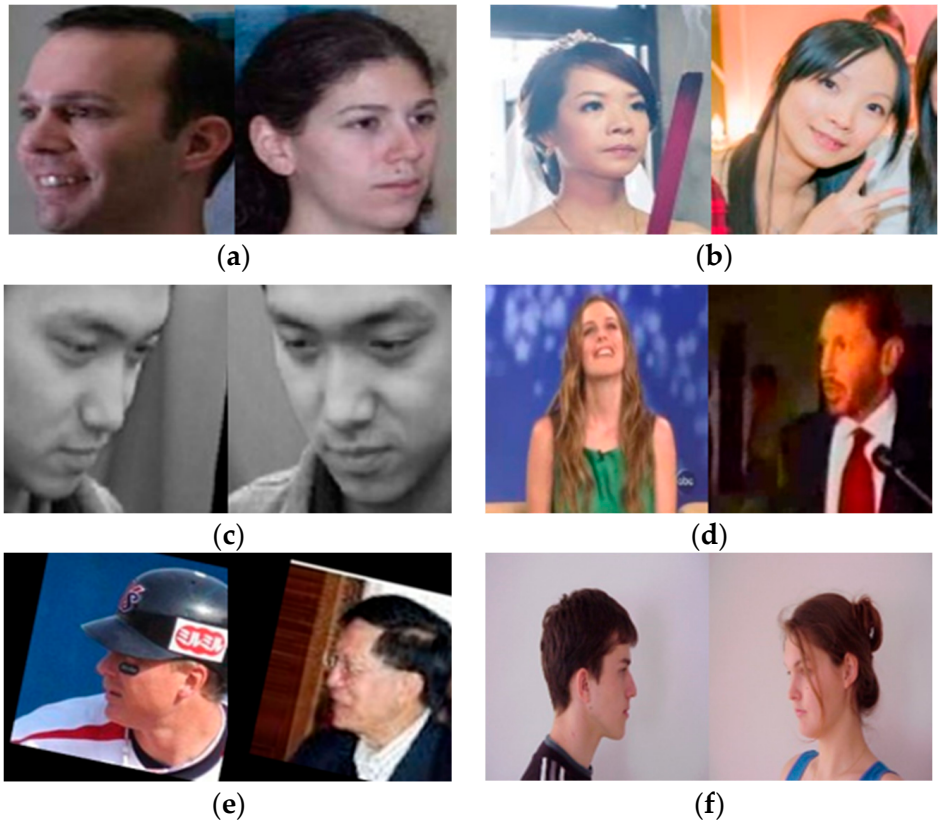


Figure 7. Example images from different datasets. (**a**) Multi-PIE (**b**) MegaFace (**c**) CAS-PEAL (**d**) YTF (**e**) CPLFW (**f**) CVL.

MegaFace is a challenging, large-scale face dataset. It contains the gallery set and the probe set. The gallery set contains more than 1 million face images, while the probe set contains 106,863 face images of 530 celebrities. Figure 7b shows some samples of MegaFace.

CAS-PEAL includes 99,450 images of 1040 different people, which mainly involve pose variations, expression variations, and lighting variations. Figure 7c shows some samples of CAS-PEAL.

YTF contains 3425 videos of 1595 subjects with diverse ethnicities. Figure 6d shows some samples of YTF.

CPLFW includes 11,652 images of 5749 different people, which mainly involves pose variations. Figure 7e shows some samples of CPLFW.

CVL contains 798 images of 114 different people, which mainly involves pose variations. Figure 7f shows some samples of CVL.

4.2. Experimental Results and Analysis

4.2.1. Comparison with State-of-the-Art Methods

Experimental Setup

Resnet [48], Duan's method [49], PGM-Face [29], PCCycleGAN [14], LDMR [20], MH-DNCCM [21], DRA-Net [16], TGLBP [35], MCN [37], 3D-PIM [50] WFH [17], mCNN [51],

HADL [31], RVFace [52], DTDD [53], ArcFace [54], VGG [55], and DeepID [56] are used as the comparison methods.

For multi-PIE, we choose images with pose deflection angles of -45°, -30°, -15°, 0°, 15°, 30°, 45° for experiments. In other words, a total of 2359 images of 337 subjects were used for the experiments. For images of each subject, we randomly selected three images for training and the remaining images for testing. It means that the number of training images accounted for 42.85% of the total number of images, and the number of testing images accounted for 57.15% of the total number of images.

For MegaFace, we selected the samples of categories with the number of images greater than or equal to two for experiments. For each class of samples used for experiments, we randomly selected one image for training and one image for testing. Namely, the number of training images accounted for 50% of the total number of images, and the number of testing images accounted for 50% of the total number of images.

For CAS-PEAL, we choose those images involving 800 subjects in three different poses (0°, -45° and 45°) for experiments. That is to say, each subject contains three images deflected at different angles. The image with a deflection angle of 0 degrees in each subject is used for training and the rest are used for testing. Specifically, the number of training images accounted for 33.33% of the total number of images and the number of testing images accounted for 66.67% of the total number of images.

For YTF, we selected 226 subjects with four or more videos available. Then, we selected 225 subjects from 226 subjects for experiments and divided the 225 subjects into five groups, each group involving 45 subjects. For each group, the first three videos of each subject as gallery sets and the remaining videos for testing. The results obtained from the five groups of experiments are averaged as the final experimental result. The number of training samples accounted for 43.29% of the total number of images, and the number of testing samples accounted for 56.71% of the total number of images.

For CPLFW, we selected the samples of 2000 classes to form a subset. For samples belonging to a certain class (each class) in this subset, we randomly selected one image as the training sample and one image as the test sample. Precisely, the number of training images accounted for 50% of the total number of images, and the number of testing images accounted for 50% of the total number of images.

For CVL, we choose three images in each class for training and the rest for testing. That is to say, the number of training images accounted for 42.85% of the total number of images, and the number of testing images accounted for 57.15% of the total number of images.

The images used in the experiments are cropped to 60×80.

Experimental Result

The Accuracies of different methods on different datasets are shown in Table 1. It can be seen from the experimental results on multi-PIE that ArcFace has the highest recognition rate, reaching 95.89%. This may be because the proportion of images involving large pose changes is relatively small, resulting in the difference between most training images and test images not being too large, and ArcFace can achieve good recognition results. Furthermore, almost all methods have achieved good results. The reason for this result is as follows. The Multi-PIE dataset involves relatively few images with large pose deflection. For example, the number of images with an attitude deflection of 45 degrees only accounts for two-sevenths of the total dataset, which means most of the images used for training have little difference from the test images. Then, the trained model can better identify the test samples.

Table 1. Accuracies (%) of Different Methods on Different Datasets.

Methods\Datasets	Multi-PIE [40]	MegaFace [41]	CAS-PEAL [42]	YTF [43]	CPLFW [44]	CVL [45]
Reset [48]	91.06	87.77	90.77	76.05	82.36	89.56
Duan [49]	87.68	82.55	89.37	73.88	81.06	85.17
PGM-Face [29]	90.23	85.33	90.01	73.20	78.58	88.06
PCCycleGAN [14]	88.99	85.01	88.85	75.16	80.66	87.38
LDMR [20]	91.33	86.23	92.29	77.22	85.06	90.23
MH-DNCCM [21]	91.78	85.89	90.46	76.11	82.50	87.98
DRA-Net [16]	93.06	83.99	93.16	76.47	82.97	90.01
TGLBP [35]	89.06	86.13	90.67	75.60	83.71	86.97
MCN [37]	92.01	86.24	91.37	77.05	85.20	88.31
3D-PIM [50]	93.02	86.31	91.79	78.05	85.07	89.22
WFH [17]	91.55	86.01	92.70	74.88	84.01	87.34
mCNN [51]	88.68	85.17	87.58	72.89	80.59	81.38
HADL [31]	90.82	85.35	90.47	75.95	84.35	86.40
RVFace [52]	92.10	88.03	93.17	78.05	85.97	90.03
DTDD [53]	90.39	88.37	93.55	77.35	86.23	88.80
ArcFace [54]	95.89	91.37	92.13	83.40	84.88	87.23
VGG [55]	95.14	89.29	90.92	81.05	83.06	85.71
DeepID [56]	93.88	87.58	88.15	78.45	83.46	85.12
HDMCO [ours]	95.19	90.67	95.88	80.34	88.41	92.19

For the experimental results on MegaFace, almost all methods based on deep learning achieved good results. The possible reasons are as follows: although the number of samples used for training in each category is not large, the difference between the large number of samples used for pre-training and the test samples is not too large. Thus, the final learned model has better classification ability for the test samples. Among all the methods, Duan's method has the worst performance, which may be because the performance of the method depends on finding the parts related to the pose. However, it cannot completely and correctly determine which parts of the image are related to the pose. Furthermore, this method mainly solves face recognition involving pose changes, while the MegaFace dataset involves not only pose changes but also other changes, so the recognition rate of this method on MegaFace is not very high.

The experimental results on YTF show that the recognition rate of all algorithms does not exceed 85%. This is because YTF datasets involve large changes (e.g., large pose changes, large expression changes), so their performance is not very good. Specifically, for methods based on deep learning, the pre-trained model is not suitable for the classification of test images. This is because a large number of samples used for pre-training are quite different from the images in the used dataset. For HADL, because the samples used for training may be quite different from the samples used for testing, the learned dictionary cannot accurately represent the test samples, which means that the algorithm cannot obtain a higher recognition rate. For Duan's method, because the samples used for training may be quite different from the samples used for testing, the learned characteristics of a certain category are quite different from those of the same category of images in the test set. Then, the recognition rate of the algorithm on the YTF dataset is not very high.

The experimental results on CPLFW show that the recognition rate of our method is higher than that of other algorithms. This may be because our proposed non-negative matrix factorization based on hypergraph embedding extracts the frontal images with better quality. In other words, we use hypergraph and non-negative matrix factorization to

separate the frontal image from the profile image. The extracted pose-free features are then used to learn the dictionaries with strong performance, and the learned dictionaries are used to accurately represent the test samples, thereby greatly improving the recognition rate of the algorithm. The reasons why the recognition rate of the deep learning-based

Method is not as high as that of our method are as follows. A large number of samples used for pre-training are too different from the samples in the test set. For example, many samples used for pre-training are images with small attitude deflection, while many test samples are images with large attitude deflection. Then, the rules summarized for each category through training are not suitable for the rules of the same category of images in the test set.

The experimental results on CVL show that the recognition rate of our method is 92%, which is higher than that of other methods. The reasons for this result are as follows: the hypergraph is embedded in the non-negative matrix factorization so that the resulting images retain the intrinsic properties of the original images. Furthermore, triangular encoding is used to encode the obtained pose-free images, which makes the extracted features highly unique. Furthermore, we use the encoded features to train the dictionary, so that the learned dictionary has a stronger representation ability. Then, the test samples can be accurately represented by the dictionaries, thereby achieving the purpose of improving the recognition rate. The performance of deep learning-based methods is not the best among all methods, and the reasons for this result are as follows. The rules summarized for each category through pre-training are quite different from the rules of the same category of images in the test set. Therefore, the model obtained by training is not suitable for the classification of the test images, or the model obtained by training cannot correctly classify many test images. For HADL and LDMR, it is difficult for them to extract the pose-invariant features of the images when dealing with images with large pose changes, which makes it difficult for subsequent classifiers to correctly identify samples.

Tables 2 and 3 show the recall and precision of different methods on different datasets. The experimental results obtained are generally consistent with those in Table 1; HDMCO has the best effect.

Table 2. Recall (%) of Different Methods on Different Datasets.

Methods\Datasets	Multi-PIE [40]	MegaFace [41]	CAS-PEAL [42]	YTF [43]	CPLFW [44]	CVL [45]
Reset [48]	78.35	70.68	76.18	82.67	76.83	81.33
Duan [49]	76.02	65.43	72.67	77.61	73.25	78.69
PGM-Face [29]	79.66	73.14	75.68	75.60	77.25	80.39
PCCycleGAN [14]	81.64	72.95	77.62	73.08	76.89	76.28
LDMR [20]	83.58	76.89	72.99	75.03	75.88	79.01
MH-DNCCM [21]	82.05	73.91	70.03	73.26	74.19	80.06
DRA-Net [16]	85.11	80.34	77.68	79.32	80.64	81.39
TGLBP [35]	80.24	78.92	78.33	75.17	78.38	79.68
MCN [37]	83.67	80.20	81.08	77.68	80.34	78.18
3D-PIM [50]	85.02	81.35	81.69	79.67	77.58	76.64
WFH [17]	82.67	78.54	76.44	77.39	79.14	75.89
mCNN [51]	80.33	78.67	79.58	78.99	80.01	78.66
HADL [31]	78.89	80.59	79.44	79.88	78.46	80.62

Table 2. *Cont.*

Methods\Datasets	Multi-PIE [40]	MegaFace [41]	CAS-PEAL [42]	YTF [43]	CPLFW [44]	CVL [45]
RVFace [52]	82.24	80.30	77.89	79.01	81.33	80.23
DTDD [53]	80.95	80.68	79.25	79.31	80.64	82.07
ArcFace [54]	82.53	85.01	82.34	80.09	81.69	80.60
VGG [55]	79.28	81.02	78.08	75.89	79.88	79.47
DeepID [56]	81.08	82.16	79.66	81.06	81.06	78.30
HDMCO [ours]	88.37	85.67	86.17	85.60	88.32	88.97

Table 3. Precision (%) of Different Methods on Different Datasets.

Methods\Datasets	Multi-PIE [40]	MegaFace [41]	CAS-PEAL [42]	YTF [43]	CPLFW [44]	CVL [45]
Reset [48]	89.26	86.08	86.92	78.34	80.16	86.57
Duan [49]	85.06	80.38	88.15	75.06	79.68	86.23
PGM-Face [29]	89.32	86.42	88.95	75.01	77.19	86.27
PCCycleGAN [14]	86.27	85.39	86.19	77.12	79.18	85.61
LDMR [20]	88.97	85.09	90.87	75.80	83.97	87.18
MH-DNCCM [21]	88.39	82.17	88.69	78.02	80.05	85.10
DRA-Net [16]	90.86	82.34	92.05	75.24	80.34	87.19
TGLBP [35]	86.95	85.06	91.21	75.32	81.99	85.43
MCN [37]	90.67	83.97	89.68	76.38	83.97	87.32
3D-PIM [50]	90.98	85.11	90.08	75.86	83.46	87.68
WFH [17]	89.30	83.67	90.79	74.02	83.97	86.22
mCNN [51]	86.91	85.06	86.40	70.66	79.30	79.66
HADL [31]	88.69	84.39	88.67	76.08	83.97	85.88
RVFace [52]	90.68	86.92	91.86	78.68	85.02	88.60
DTDD [53]	88.67	87.08	92.43	76.18	85.15	87.67
ArcFace [54]	93.91	90.28	90.88	82.91	82.69	86.41
VGG [55]	95.86	88.06	88.67	79.38	81.67	85.02
DeepID [56]	93.05	86.14	85.97	77.68	81.97	84.67
HDMCO [ours]	96.08	91.35	94.86	81.57	88.05	92.30

4.2.2. Cross-Validation Experiment

In order to further verify the performance of HDMCO, cross-validation experiments are carried out in this section. For each data set, we selected the face image with an attitude deflection angle greater than 45°, and 5-fold cross-validation was performed. Specifically, the data set was divided into five parts, four of which were taken as training data and one as test data in turn, and the experiment was carried out. Each trial obtained the corresponding recognition rate. The average recognition rate of the results of five times was used as the estimation of the algorithm accuracy.

As can be seen from Table 4, the average recognition rate of many algorithms on the multi-PIE data set and CAS-PEAL data set is more than 80%. At the same time, it can also be seen that the recognition rate of the proposed HDMCO is higher than that of other algorithms.

Table 4. The Results (%) of Cross-validation.

Methods\Datasets	Multi-PIE [40]	MegaFace [41]	CAS-PEAL [42]	YTF [43]	CPLFW [44]	CVL [45]
Reset [48]	81.32	78.92	81.24	68.05	73.68	80.92
Duan [49]	80.68	75.60	80.38	63.58	71.99	78.96
PGM-Face [29]	77.68	79.31	77.59	72.38	68.56	77.90
PCCycleGAN [14]	76.82	75.66	74.97	68.33	69.98	78.62
LDMR [20]	80.38	83.29	80.64	73.20	76.82	80.93
MH-DNCCM [21]	81.60	81.32	83.67	64.51	71.93	79.71
DRA-Net [16]	83.64	83.16	81.46	66.49	74.69	80.97
TGLBP [35]	80.32	80.97	76.91	64.98	72.64	77.62
MCN [37]	80.06	79.86	80.46	71.61	71.62	78.59
3D-PIM [50]	81.30	80.61	78.67	70.38	73.92	78.61
WFH [17]	81.69	80.67	81.33	63.89	73.68	79.68
mCNN [51]	79.37	75.31	76.82	63.99	71.68	70.29
HADL [31]	80.34	73.97	80.34	73.61	73.61	77.85
RVFace [52]	77.31	76.89	83.89	75.06	75.38	79.33
DTDD [53]	78.39	80.67	82.58	73.68	78.99	78.99
ArcFace [54]	82.67	78.59	81.37	77.31	74.63	77.97
VGG [55]	80.69	77.98	80.59	73.68	73.91	76.89
DeepID [56]	83.99	77.86	78.61	71.68	77.35	77.95
HDMCO [ours]	89.30	85.07	85.99	78.95	83.93	83.97

4.2.3. The Effect of Feature Dimension on the Recognition Performance of the Algorithms

To illustrate the effect of feature dimension on the recognition rate of our method, we conducted experiments. DDTD, HADL, and PCCycleGAN are used as comparison methods. The experimental conditions are the same as the experimental conditions in Section 4.2.1. The only difference is that the dimension of the features ranges from 100 to 600. Figure 8 shows the effect of feature dimension on the recognition rate of different methods. It can be seen from Figure 8 that the recognition rates of all methods first gradually increase with the feature dimension and then remain unchanged. Furthermore, the recognition performance of our method is better than other methods.

4.2.4. The Display of the Extracted Frontal Images

To illustrate that our method can effectively separate pose-free images from pose-deflected images, we show the obtained separated images. In Figure 9, the left half of each subfigure shows the original image, and the right half shows the pose-free deflection image separated from the original image. As can be seen from Figure 9, the separated images are close to the frontal image. This shows that the proposed feature discrimination enhancement method based on non-negative matrix factorization and hypergraph embedding can indeed achieve the de-pose function.

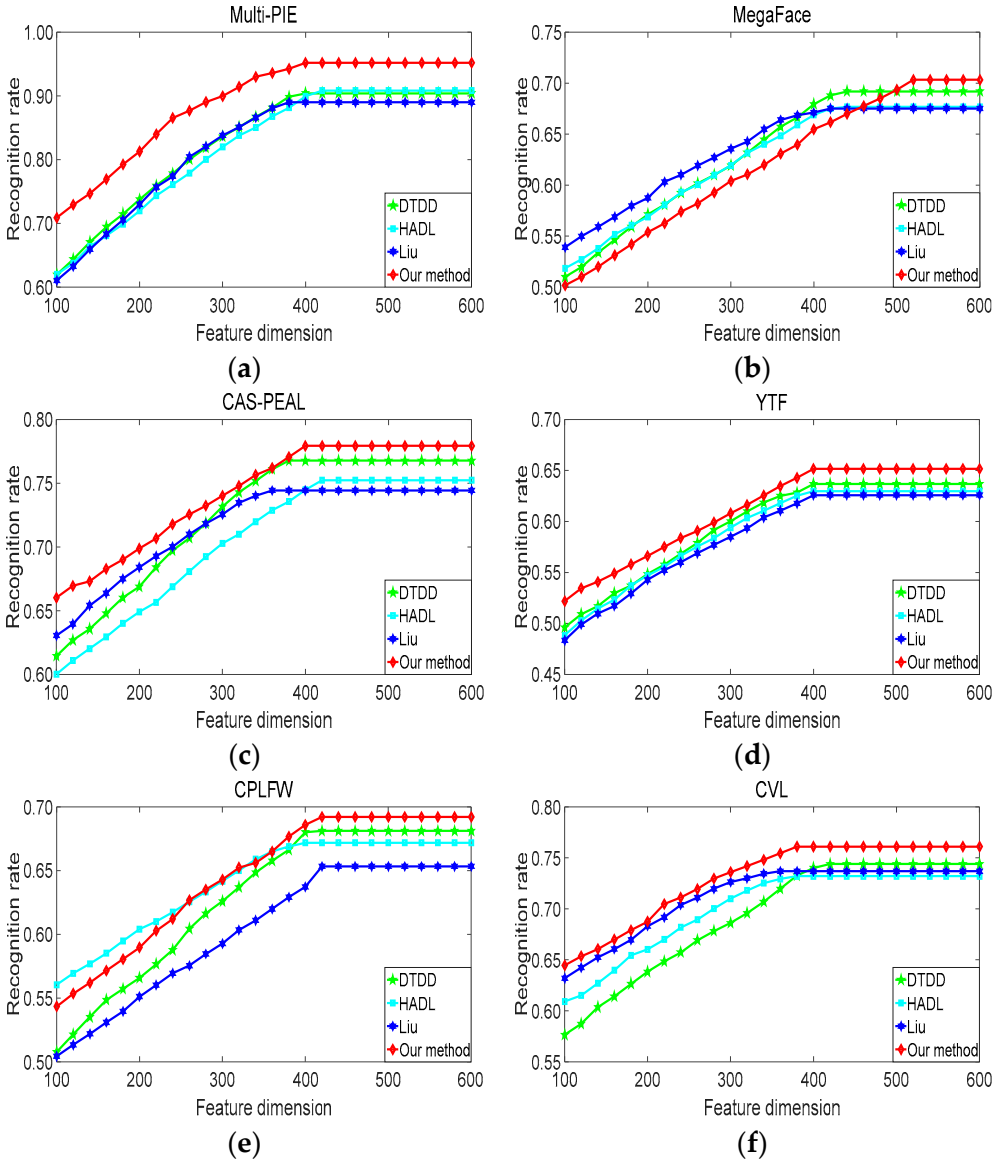


Figure 8. The effect of feature dimension on the recognition rate of different methods. (**a**) Multi-PIE (**b**) MegaFace (**c**) CAS-PEAL (**d**) YTF (**e**) CPLFW (**f**) CVL.

4.2.5. Ablation Experiment

To verify the role of each component in the proposed method, we performed ablation experiments. The main components of the method proposed in this paper are the "feature discrimination enhancement method based on non-negative matrix factorization and hypergraph embedding", the "feature coding method based on improved support vector data description", "dictionary learning-based classifier", and "joint optimization of the feature extraction and feature classification", which are abbreviated as de-deflection, feature coding, dictionary learning, and joint optimization.

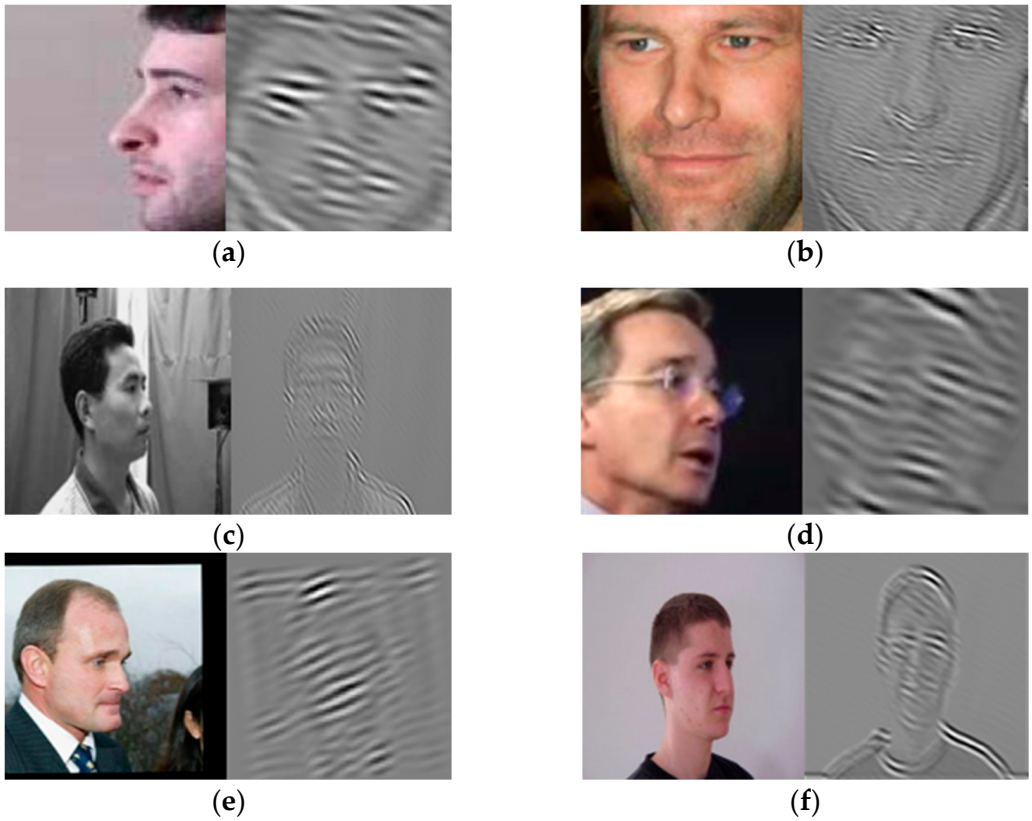


Figure 9. Images are separated by non-negative matrix factorization based on the hypergraph (**a**) Multi-PIE (**b**) MegaFace (**c**) CAS-PEAL (**d**) YTF (**e**) CPLFW (**f**) CVL.

Experimental Setup

The experimental conditions are the same as in Section 4.2.1.

Experimental Results

Figure 10 shows the results of ablation experiments. It can be seen from Figure 10a that using the de-deflection component can improve the recognition rate of the algorithm by about 2% on some datasets, and more on some datasets, such as 5% and 7%. As can be seen from Figure 10b, the use of feature encoding component improves the recognition rate of the algorithm by about 2% on almost all datasets. It can be seen from Figure 10c that the use of the dictionary learning component improves the recognition rate of the algorithm by about 1% on some datasets and by about 2% on others. It can be seen from Figure 10d that using the joint optimization component improves the recognition rate of the algorithm by about 3% on almost all datasets.

4.2.6. The Effect of Parameters on the Recognition Performance of HDMCO

In HDMCO, η and λ are the main parameters. To explore their impact on the recognition rate of HDMCO, we conducted experiments. The experimental conditions are the same as the experimental conditions in Section 4.2.1. The only difference is that η ranges from 0.1 to 0.6, and λ ranges from 0 to 1. Figure 11 shows the effect of the main parameters on the recognition rate of HDMCO. It can be seen from Figure 11 that the recognition rate of HDMCO is the highest when the value of η is about 0.3, and the recognition rate of HDMCO is the highest when the value of λ is about 0.5.

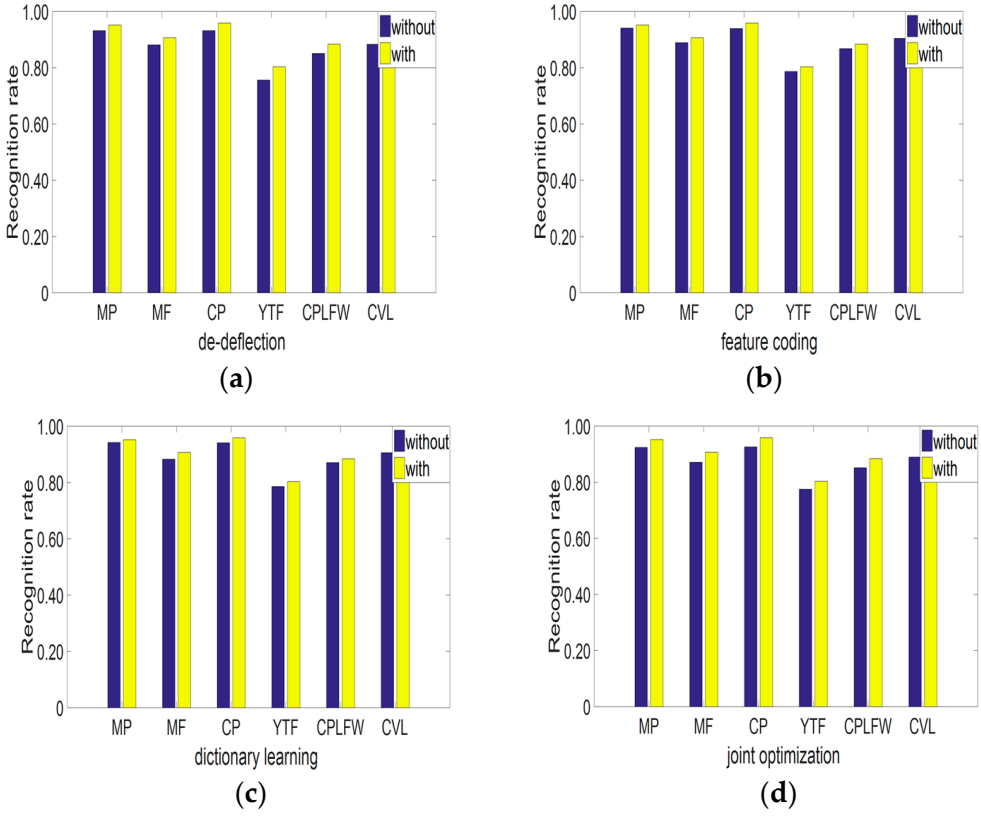


Figure 10. Results of ablation experiments. MP represents Multi-PIE, MF represents MegaFace, CP represents CAS-PEAL (**a**) de-deflection (**b**) feature learning (**c**) dictionary learning (**d**) joint optimization.

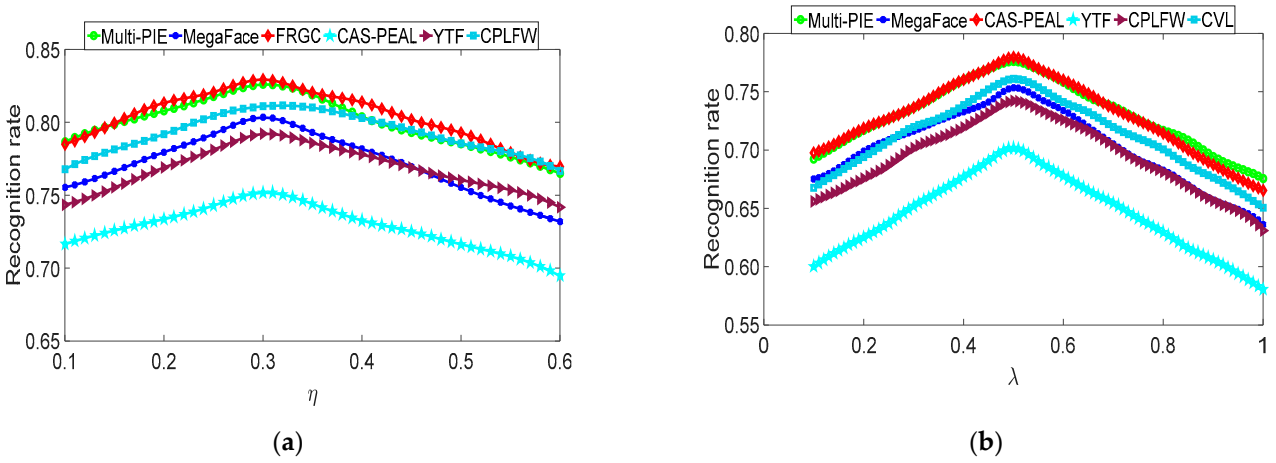


Figure 11. The effect of main parameters on the recognition rate of HDMCO. (**a**) η (**b**) λ.

4.2.7. Comparison of Computational Complexity

In this section, we analyze the computational complexity of the proposed algorithm and compare it with the computational complexity of several existing methods. The computational complexity of HDMCO is mainly dervived from solving $\boldsymbol{\alpha}$ using linear programming; meanwhile, the computational complexity of calculating $\boldsymbol{\alpha}$ is $o(n_0^2)$, and n_0 is the number of training samples. Thus, the computational complexity of HDMCO is $o(n_0^2)$. HADL [27] and LDMR [19] are used as comparative methods. The computational complexity of HADL is $O(M\tau(Kn_0^3 + K\max(L, K)))$, where τ is the iteration number, and L is the dimension of each sample, K is the number of atoms in the dictionary. M is the

maximum number of the iteration number. The computational complexity of LDMR is $O(u_0v_0n_0^2 + n_0^3 + \tau(u_0v_0^2 + u_0v_0n_0))$, and u_0 and v_0 are the width and height of the image, respectively. It is easy to see from the computational complexity expressions of the three algorithms that the computational complexity of HDMCO is n_0^2, while the computational complexity of the other two algorithms is n_0^3. Hence, HDMCO has low computational complexity. Meanwhile, for example, the running time of HDMCO on the multi-PIE database is 713.45 s, while the running time of HADL and LDMR are 4397.45 s and 5813.24 s. The configuration of our computer is as follows: Intel Core i7-9700 K, 3.6 GHz, Nvidia GeForce RTX 2080 Ti.

5. Conclusions

In this paper, we propose a novel few-shot, multi-pose face recognition method based on hypergraph de-deflection and multi-task collaborative optimization (HDMCO). HDMCO uses the hypergraph theory and non-negative matrix decomposition to separate the frontal images from the attitude deflection images, and then uses the improved support vector data description and triangle coding to extract the features of the separated images without attitude deflection. Dictionary learning-based classifier is then also used to classify those features. The feature extraction process and feature classification process are jointly optimized. The large number of experimental results show that the proposed HDMCO does achieve good results. Although we have jointly optimized feature extraction and feature classification and achieved better results, since the separation of frontal images is separate from the subsequent feature extraction, the obtained recognition result is still not the ultimate optimal result of HDMCO. In future work, we will continue to explore the joint optimization of the separation of frontal images and feature extraction to obtain the ultimate optimal recognition effect of HDMCO.

Author Contributions: Conceptualization, X.F.; Methodology, X.F.; Validation, M.L.; Formal analysis, M.L.; Data curation, L.C.; Writing—review & editing, X.F.; Supervision, L.C.; Funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by the Post-doctoral Innovative Talent Support Program (Grant no. BX20200048), and in part by the General Program of China Postdoctoral Science Foundation (Grant no. 2021M700405).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jeevan, G.; Zacharias, G.C.; Nair, M.S.; Rajan, J. An empirical study of the impact of masks on face recognition. *Pattern Recognit.* **2022**, *122*, 108308. [CrossRef]
2. Solovyev, R.; Wang, W.; Gabruseva, T. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image Vis. Comput.* **2021**, *107*, 104117. [CrossRef]
3. Wu, C.; Ju, B.; Wu, Y.; Xiong, N.N.; Zhang, S. WGAN-E: A generative adversarial networks for facial feature security. *Electronics* **2020**, *9*, 486. [CrossRef]
4. Sengupta, S.; Chen, J.C.; Castillo, C.; Patel, V.M.; Chellappa, R.; Jacobs, D.W. Frontal to profile face verification in the wild. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–10 March 2016; pp. 1–9.
5. Khrissi, L.; El Akkad, N.; Satori, H.; Satori, K. Clustering method and sine cosine algorithm for image segmentation. *Evol. Intell.* **2022**, *15*, 669–682. [CrossRef]
6. Zhao, J.; Xiong, L.; Cheng, Y.; Cheng, Y.; Li, J.; Zhou, L.; Xu, Y.; Karlekar, J.; Pranata, S.; Shen, S.; et al. 3D-aided deep pose-invariant face recognition. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018; Volume 2, p. 11.
7. Zhao, J.; Xiong, L.; Li, J.; Xing, J.; Yan, S.; Feng, J. 3D-aided dual-agent gans for unconstrained face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2380–2394. [CrossRef]
8. Zhao, J.; Cheng, Y.; Xu, Y.; Xiong, L.; Li, J.; Zhao, F.; Jayashree, K.; Pranata, S.; Shen, S.; Xing, J.; et al. Towards pose invariant face recognition in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2207–2216.

9. Zhao, J.; Xiong, L.; Karlekar Jayashree, P.; Li, J.; Zhao, F.; Wang, Z.; Sugiri Pranata, P.; Shengmei Shen, P.; Yan, S.; Feng, J. Dual-agent gans for photorealistic and identity preserving profile face synthesis. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
10. Zhao, J. Deep Learning for Human-Centric Image Analysis. Ph.D. Thesis, National University of Singapore, Singapore, 2018.
11. Khrissi, L.; EL Akkad, N.; Satori, H.; Satori, K. An Efficient Image Clustering Technique based on Fuzzy C-means and Cuckoo Search Algorithm. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 423–432. [CrossRef]
12. Ding, C.; Tao, D. Pose-invariant face recognition with homography-based normalization. *Pattern Recognit.* **2017**, *66*, 144–152. [CrossRef]
13. Luan, X.; Geng, H.; Liu, L.; Li, W.; Zhao, Y.; Ren, M. Geometry structure preserving based gan for multi-pose face frontalization and recognition. *IEEE Access* **2020**, *8*, 104676–104687. [CrossRef]
14. Liu, Y.; Chen, J. Unsupervised face frontalization for pose-invariant face recognition. *Image Vis. Comput.* **2021**, *106*, 104093. [CrossRef]
15. Yin, Y.; Jiang, S.; Robinson, J.P.; Fu, Y. Dual-attention gan for large-pose face frontalization. In Proceedings of the 2020 15th IEEE international conference on automatic face and gesture recognition (FG 2020), Buenos Aires, Argentina, 16–20 November 2020; pp. 249–256.
16. Lin, C.-H.; Huang, W.-J.; Wu, B.-F. Deep representation alignment network for pose-invariant face recognition. *Neurocomputing* **2021**, *464*, 485–496. [CrossRef]
17. Yang, H.; Gong, C.; Huang, K.; Song, K.; Yin, Z. Weighted feature histogram of multi-scale local patch using multi-bit binary descriptor for face recognition. *IEEE Trans. Image Process.* **2021**, *30*, 3858–3871. [CrossRef]
18. Tu, X.; Zhao, J.; Liu, Q.; Ai, W.; Guo, G.; Li, Z.; Liu, W.; Feng, J. Joint face image restoration and frontalization for recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 1285–1298. [CrossRef]
19. Zhou, L.-F.; Du, Y.-W.; Li, W.-S.; Mi, J.-X.; Luan, X. Pose-robust face recognition with huffman-lbp enhanced by divide-and-rule strategy. *Pattern Recognit.* **2018**, *78*, 43–55. [CrossRef]
20. Zhang, C.; Li, H.; Qian, Y.; Chen, C.; Zhou, X. Locality-constrained discriminative matrix regression for robust face identification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 1254–1268. [CrossRef]
21. Gao, L.; Guan, L. A discriminative vectorial framework for multi-modal feature representation. *IEEE Trans. Multimed.* **2021**, *24*, 1503–1514.
22. Yang, S.; Deng, W.; Wang, M.; Du, J.; Hu, J. Orthogonality loss: Learning discriminative representations for face recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 2301–2314. [CrossRef]
23. Huang, F.; Yang, M.; Lv, X.; Wu, F. Cosmos-loss: A face representation approach with independent supervision. *IEEE Access* **2021**, *9*, 36819–36826. [CrossRef]
24. He, M.; Zhang, J.; Shan, S.; Kan, M.; Chen, X. Deformable face net for pose invariant face recognition. *Pattern Recognit.* **2020**, *100*, 107113. [CrossRef]
25. Wang, Q.; Guo, G. Dsa-face: Diverse and sparse attentions for face recognition robust to pose variation and occlusion. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4534–4543. [CrossRef]
26. He, R.; Li, Y.; Wu, X.; Song, L.; Chai, Z.; Wei, X. Coupled adversarial learning for semi-supervised heterogeneous face recognition. *Pattern Recognit.* **2021**, *110*, 107618. [CrossRef]
27. Liu, H.; Zhu, X.; Lei, Z.; Cao, D.; Li, S.Z. Fast adapting without forgetting for face recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 3093–3104. [CrossRef]
28. Sun, J.; Yang, W.; Xue, J.H.; Liao, Q. An equalized margin loss for face recognition. *IEEE Trans. Multimed.* **2020**, *22*, 2833–2843. [CrossRef]
29. Zhang, Y.; Fu, K.; Han, C.; Cheng, P.; Yang, S.; Yang, X. PGM-face: Pose-guided margin loss for cross-pose face recognition. *Neurocomputing* **2021**, *460*, 154–165. [CrossRef]
30. Badave, H.; Kuber, M. Head pose estimation based robust multicamera face recognition. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 492–495.
31. Wang, L.; Li, S.; Wang, S.; Kong, D.; Yin, B. Hardness-aware dictionary learning: Boosting dictionary for recognition. *IEEE Trans. Multimed.* **2020**, *23*, 2857–2867. [CrossRef]
32. Holkar, A.; Walambe, R.; Kotecha, K. Few-shot learning for face recognition in the presence of image discrepancies for limited multi-class datasets. *Image Vis. Comput.* **2022**, *120*, 104420. [CrossRef]
33. Guan, Y.; Fang, J.; Wu, X. Multi-pose face recognition using cascade alignment network and incremental clustering. *Signal, Image Video Process.* **2021**, *15*, 63–71. [CrossRef]
34. Zhang, Y.; Fu, K.; Han, C.; Cheng, P. Identity-and-pose-guided generative adversarial network for face rotation. *Neurocomputing* **2021**, *450*, 33–47. [CrossRef]
35. Qu, H.; Wang, Y. Application of optimized local binary pattern algorithm in small pose face recognition under machine vision. *Multimed. Tools Appl.* **2022**, *81*, 29367–29381. [CrossRef]
36. Masi, I.; Chang, F.J.; Choi, J.; Harel, S.; Kim, J.; Kim, K.; Leksut, J.; Rawls, S.; Wu, Y.; Hassner, T.; et al. Learning pose-aware models for pose-invariant face recognition in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 379–393. [CrossRef]
37. Elharrouss, O.; Almaadeed, N.; Al-Maadeed, S.; Khelifi, F. Pose-invariant face recognition with multitask cascade networks. *Neural Comput. Appl.* **2022**, *34*, 6039–6052. [CrossRef]

38. Liu, J.; Li, Q.; Liu, M.; Wei, T. CP-GAN: A cross-pose profile face frontalization boosting pose-invariant face recognition. *IEEE Access* **2020**, *8*, 198659–198667. [CrossRef]
39. Tao, Y.; Zheng, W.; Yang, W.; Wang, G.; Liao, Q. Frontal-centers guided face: Boosting face recognition by learning pose-invariant features. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2272–2283. [CrossRef]
40. Gao, G.; Yu, Y.; Yang, M.; Chang, H.; Huang, P.; Yue, D. Cross-resolution face recognition with pose variations via multilayer locality-constrained structural orthogonal procrustes regression. *Inf. Sci.* **2020**, *506*, 19–36. [CrossRef]
41. Wang, H.; Kawahara, Y.; Weng, C.; Yuan, J. Representative selection with structured sparsity. *Pattern Recognit.* **2017**, *63*, 268–278. [CrossRef]
42. Gross, R.; Matthews, I.; Cohn, J.; Kanade, T.; Baker, S. Multi-pie. *Image Vis. Comput.* **2010**, *28*, 807–813. [CrossRef]
43. Kemelmacher-Shlizerman, I.; Seitz, S.M.; Miller, D.; Brossard, E. The megaface benchmark: 1 million faces for recognition at scale. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4873–4882.
44. Gao, W.; Cao, B.; Shan, S.; Chen, X.; Zhou, D.; Zhang, X.; Zhao, D. The CAS-PEAL large-scale chinese face database and baseline evaluations. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2007**, *38*, 149–161.
45. Wolf, L.; Hassner, T.; Maoz, I. Face recognition in unconstrained videos with matched background similarity. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 529–534.
46. Zheng, T.; Deng, W. *Cross-Pose LFW: A Database for Studying Cross-Pose Face Recognition in Unconstrained Environments*; Technical Report; Beijing University of Posts and Telecommunications: Beijing, China, 2018; Volume 5.
47. Peer, P. CVL Face Database, Computer Vision Lab., Faculty of Computer and Information Science, University of Ljubljana, Slovenia. 2005. Available online: http://www.lrv.fri.uni-lj.si/facedb.html (accessed on 27 March 2023).
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
49. Duan, X.; Tan, Z.-H. A spatial self-similarity based feature learning method for face recognition under varying poses. *Pattern Recognit. Lett.* **2018**, *111*, 109–116. [CrossRef]
50. Wu, H.; Gu, J.; Fan, X.; Li, H.; Xie, L.; Zhao, J. 3D-guided frontal face generation for pose-invariant recognition. *ACM Trans. Intell. Syst. Technol.* **2023**, *14*, 1–21. [CrossRef]
51. Zhao, J.; Li, J.; Zhao, F.; Nie, X.; Chen, Y.; Yan, S.; Feng, J. Marginalized CNN: Learning deep invariant representations. In Proceedings of the British Machine Vision Conference (BMVC), London, UK, 4–7 September 2017. [CrossRef]
52. Wang, X.; Wang, S.; Liang, Y.; Gu, L.; Lei, Z. RVFace: Reliable vector guided softmax loss for face recognition. *IEEE Trans. Image Process.* **2022**, *31*, 2337–2351. [CrossRef]
53. Zhong, Y.; Deng, W.; Fang, H.; Hu, J.; Zhao, D.; Li, X.; Wen, D. Dynamic training data dropout for robust deep face recognition. *IEEE Trans. Multimed.* **2021**, *24*, 1186–1197. [CrossRef]
54. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4690–4699.
55. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
56. Sun, Y.; Wang, X.; Tang, X. Deep learning face representation from predicting 10,000 classes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1891–1898.