Article

Color Occlusion Face Recognition Method Based on Quaternion Non-Convex Sparse Constraint Mechanism

Chenglin Wen 1,∗ and Yiting Qiu 2

1 School of Automation, Guangdong University of Petrochemical Technology, Maoming 525000, China
2 School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China; qiu_yt@hdu.edu.cn
∗ Correspondence: wencl@hdu.edu.cn

Abstract: As the acquisition and application of color images become more and more extensive, color face recognition technology has also been vigorously developed, especially the recognition methods based on convolutional neural network, which have excellent performance. However, with the increasing depth and complexity of network models, the number of calculated parameters also increases, which means the training of most high-performance models depends on large-scale samples and expensive equipment. Therefore, the key to the current research is to realize a lightweight model while ensuring the recognition accuracy. At present, PCANet, a typical lightweight framework for deep learning, has achieved good results in most of the image recognition tasks, but its recognition accuracy for color face images, especially under occlusion, still needs to be improved. Therefore, a color occlusion face recognition method based on quaternion non-convex sparse constraint mechanism is proposed in this paper. Firstly, a quaternion non-convex sparse principal component analysis network model was constructed based on $L_p$ regularization of strong sparsity. Secondly, the fixed point iteration method and coordinate descent method were established to solve the non-convex optimization problem. Finally, the occlusion recognition performance of the proposed method was verified on Georgia Tech, Color FERET, AR, and LFW-A Color face datasets.

Keywords: occluded color face; PCANet; $L_p$ non-convex sparse; coordinate descent; fixed point iterative

1. Introduction

As one of the important biometric recognition technologies, face recognition technology has been more and more widely used in social security, video surveillance, identity verification, mobile payment, and other fields. In recent years, research on face recognition technology have made a series of important achievements [1], especially the deep learning method based on large label samples, which has achieved very high accuracy.

The main drawback of deep learning methods is that they rely on large datasets to train models, and those datasets need to contain enough variation to generalize to previously unseen samples. However, in many real-world use cases, most datasets are not on the scale that deep learning uses, and even small-scale data collection can be very expensive or sometimes nearly impossible [2]. If the deep learning model is trained with small samples, when the model starts to use irrelevant features for prediction, it leads to the over-fitting problem, which greatly limits its classification recognition performance. As the number of network layers increases, the model structure becomes more complex, which greatly increases the amount of computation. This makes most of the advanced deep learning algorithms need to be trained and deployed on expensive high-performance graphics cards, which largely hinders the practical application and development of deep learning.

Therefore, in order to solve the problems of the above deep learning methods, the principal component analysis network (PCANet) was proposed [3]. PCANet is a lightweight deep learning algorithm combining PCA and CNN. It has a simple structure and can achieve good results in most image classification tasks with two-layer convolutional networks. Compared with CNN, it does not need a large number of training samples and
the support of highly configured hardware, and the parameters and calculations are also very small while the recognition effect is guaranteed. Moreover, compared with traditional machine learning methods, PCANet has a stronger feature extraction ability.

At the same time, with the increasing maturity of multimedia equipment and technology, daily visual information has more abundant forms of expression, especially the acquisition and application of color image is more and more extensive. A color face image can obtain more sufficient color information, such as skin color, hair color, etc. Color image has a natural recognition advantage compared with gray image, and color face recognition technology has also attracted the attention of scholars [4]. The famous feature face method proposed by Turk is mainly to extract feature space after converting a color face image to a gray image [5]. After grayscale processing, the dimension of the image matrix decreases and the operation speed increases, but this causes the loss of image color information, and the recognition effect is not good because of not making full use of the characteristics of color. Torres pointed out that color information in color images is composed of RGB in different proportions, which is of great significance for face recognition, and proposed to extend the RGB three-color channels to the traditional PCA method for color face recognition [6]. At the same time, most of the existing convolutional neural networks usually regard RGB color channels as three independent feature channels, and use three convolution kernels to convolve and add the three channels respectively in the convolution layer. The above two methods based on color image recognition can improve the target recognition rate more effectively than gray image recognition, but these methods ignore the cross-channel correlation of RGB. To overcome this shortcoming, a generalized discriminant model was proposed by Yang, converting three-color channels into one channel $P$, namely $P = x_1R + x_2G + x_3B$ [7]. In this method, the three-color channel is separated into three images, which are processed separately, and then weighted fusion. Although this method improves the level of face recognition to a certain extent, it is difficult to directly apply this method in practice because it is often difficult to obtain the optimized fusion basis $x_1, x_2, x_3$. In order to avoid solving the problem of fusion basis, Li proposed an integration method based on quaternion principal component analysis (QPCA) to solve the above problems by importing quaternion to process three channels of color image at the same time, that is to say, a quaternion is used to represent the color image. Moreover, the quaternion complex representation is used to transform it from quaternion domain to complex domain; the quaternion matrix of color image in complex domain can be established, which greatly improves the recognition accuracy. However, dimensionality expansion increases the computational complexity, which affects the efficiency of recognition in complex scenarios and reduces the interpretability of principal components to recognition results.

Aiming at a sparse principal component of a color image, Lin proposed a quaternion sparse principal component analysis method based on $L_1$ norm for sparse optimization of quaternion principal components [9]. However, the principal components obtained are not sparse enough. When a face image has a large area occlusion, it is difficult to achieve ideal recognition accuracy and computational complexity. Moreover, in the practical application of face image recognition, occlusion is a relatively common type which is especially affected by COVID-19; mask occlusion is inevitable in various face recognition scenes. It is necessary to improve the algorithm recognition accuracy under occlusion.

Therefore, a color occlusion face recognition method based on the quaternion non-convex sparse constraint mechanism is proposed in this paper, and the quaternion non-convex sparse principal component analysis network model (QNSPCANet) is established. On the basis of the PCANet model structure, the quaternion representation method was used to construct the quaternion sample matrix of a color image. Then, $L_p$ non-convex regularization was used as the constraint term in the convolution kernel sparse optimization problem. $L_p$ regularization has a good feature extraction effect for face images with large area occlusion due to its strong sparsity [10]. Secondly, as the non-convex and non-smooth properties of the objective function built based on $L_p$ regularization, the coordinate descent method was used to solve the sparse principal components, and the fixed point
iterative method was used to obtain the optimal numerical solution of the variables. Finally, experiments were performed on Georgia Tech, Color FERET, AR, and LFW-A face datasets.

2. Main Research Work

In order to overcome the problem that the convolution kernel of quaternion sparse principal component analysis is not sparse enough, the quaternion non-convex sparse principal component analysis network (QNSPCANet) based on $L_p$ regularization is proposed in this section: the establishment of the quaternion non-convex sparse PCANet model with $L_p$ regularization constraints, the quaternion sparse vector solution based on coordinate descent method, and the variable solution based on fixed point iteration method are proposed in order to obtain a more sparse convolution kernel and further improve the recognition performance. The basic framework of a two-order QNSPCANet is shown in Figure 1.

![Figure 1. Basic framework of two-order QNSPCANet.](image)

Since $L_{1/2}$ regularization has the ability of being the most sparse in nature among $L_p$ ($0 < p < 1$) regularization [10], $L_{1/2}$ regularization represents $L_p$ regularization to verify the performance of network model in actual simulation experiments.

2.1. The Establishment of QNSPCANet Model

2.1.1. Quaternion Representation of Color Images

$\{S_t\}_{t=1}^{N}$ denotes a set of training samples, and the size of each image is $m \times n$. The red, green, and blue channel matrices of each color image $S_t$ are extracted and denoted as $R_t, G_t, B_t \in R^{m \times n}$, and the mean matrices $(R), (G), (B) \in R^{m \times n}$ of all color images are respectively calculated, and $\tilde{R}_t, \tilde{G}_t, \tilde{B}_t \in R^{m \times n}$ are obtained after average processing of each image. $C_t$ can be regarded as the pixel matrix of the background color of the image, and it is the zero matrix when the background of the image is white. $Q_t \in H^{m \times n}$ is the quaternion matrix of each image, i.e.,

$$Q_t = C_t + \tilde{R}_t \cdot i + \tilde{G}_t \cdot j + \tilde{B}_t \cdot k$$  \hspace{1cm} (1)

where $i, j, k$ represent the three axes of the imaginary part of quaternion.

Then, the complex representation of quaternion is introduced, and the quaternion matrix is reconstructed into a general complex matrix. Let:

$$Q_{ta} = C_t + \tilde{R}_t \cdot i$$
$$Q_{tb} = \tilde{G}_t + \tilde{B}_t \cdot i$$  \hspace{1cm} (2)

And the Equation (1) can be equivalently converted to:

$$Q_t = Q_{ta} + Q_{tb} \cdot j$$  \hspace{1cm} (3)
The complex representation of quaternion matrix is obtained by reconstruction.

\[
\chi_t = \begin{bmatrix} Q_{t\alpha} & Q_{t\beta} \\ -Q_{t\beta} & Q_{t\alpha} \end{bmatrix}
\]

where \(\chi_t \in C^{2m \times 2n}\), \(-Q_{t\alpha} = -C_t + \bar{R}_t \cdot i\), and \(Q_{t\beta} = \bar{G}_t - \bar{B}_t \cdot i\).

Annotation 2.1: By reconstructing the color image input samples into the complex representation of quaternion, it can not only associate the color information of RGB three channels simultaneously, but also transform it from quaternion field to complex number field, which is convenient for the subsequent convolution layer operation. However, at the same time, the dimension of the input matrix is doubled and the computational complexity is increased.

\[2.1.2. \text{Quaternion Non-Convex Sparse Principal Component Analysis Convolution Kernel}\]

It can be concluded from the previous section that the original input of the convolutional network is \(\{\chi_t\}_{t=1}^{N} \in C^{2m \times 2n}\). Then, the sliding window of size \(k_1 \times k_2\) is used for block sampling of the \(t\)-th color image, and \(2m \times 2n\) sample blocks can be obtained, where each sample block is a quaternion complex representation matrix. The \(t\)-th color image matrix \(q_t = [q_{t,1}, q_{t,2}, \ldots, q_{t,4mn}] \in C^{k_1k_2 \times 4nm}\) is obtained by means of all sampling blocks and series processing. The same operation can be performed for each sample in \(\{\chi_t\}_{t=1}^{N}\) to obtain the processed input sample matrix:

\[
X = [q_1, q_2, \ldots, q_N] \in C^{k_1k_2 \times 4Nm}
\]

Next, the quaternion sparse convolution kernel is calculated. Firstly, the covariance matrix of sample matrix \(X\) needs to be calculated. The calculation formula is as follows:

\[
\Phi = XX^H
\]

where \(X^H\) is the corresponding conjugate transpose of \(X\). Then, \(\Phi\) undergoes Eigendecomposition, and the Eigenvectors corresponding to the first \(M_1\) largest Eigenvalues are retained as initial values, that is, \(A = B = [\beta_1, \beta_2, \ldots, \beta_{M_1}] \in C^{k_1k_2 \times M_1}\), where \(M_1 \leq k_1 \times k_2\). The quaternion non-convex sparse optimization problem is established as follows:

\[
(\hat{A}, \hat{B}) = \arg \min_{A, B} \left( \|X - AB^HX\|_F^2 + \lambda_2 \sum_{j=1}^{M_1} \|\beta_j\|_2^2 + \sum_{j=1}^{M_1} \lambda_{p,j} ||\beta_j||_p^p \right) \quad \text{s.t.} A^HA = I_{M_1}
\]

where \(\sum_{j=1}^{M_1} \lambda_{p,j} ||\beta_j||_p^p\) represents the \(L_p\) regularization constraint term, the sparsity of load \(\beta_j\) is controlled by \(\lambda_{p,j}\), \(L_p\) norm is defined as \(\|x\|_p = (\sum_i ||x_i||^p)^{\frac{1}{p}}\), and \(\sum_{j=1}^{M_1} \|\beta_j\|_2^2\) is to avoid the over-fitting problem. The quaternion sparse vector basis obtained after optimization solution is \(V_S = [v^1_{s1}, v^1_{s2}, \ldots, v^1_{sM_1}] \in C^{k_1k_2 \times M_1}\), where \(v^1_{sj} = \beta_j/||\beta_j|| \in C^{k_1k_2}\).

Therefore, the corresponding single QNSPCA convolution kernel is represented as follows:

\[
W_l^1 = \text{matrices}_{k_1, k_2}(v^1_{sl_1}) \in C^{k_1 \times k_2}
\]

where \(l_1 = 1, 2, \ldots, M_1\), \(\text{matrices}_{k_1, k_2}(\bullet)\) maps the vector to a matrix, and \(v^1_{sl_1}\) represents the \(l_1\)-th principal component vector in the first layer quaternion sparse vector matrix.
2.1.3. Two-Order Convolution Layer

The quaternion sparse convolution kernel calculated in the previous section is used to perform convolution operation with the sample image, then the output after the first convolution is:

$$f^{l_1}_t = \chi_t \ast W^{l_1}_t$$  \hspace{1cm} (9)

where $l_1 = 1, 2, \cdots, M_1$, $t = 1, 2, \cdots, N$. After a convolution, a color image $\chi_t$ can obtain $M_1$ corresponding Eigenmatrices, which serve as the input of the next layer.

The second convolution operation has the same principle as the first convolution, but the image size is reduced after convolution. Therefore, the edge of the matrix output of the previous layer needs to be zeroed before the second convolution, and the final output of the first convolution is $\{f^{l_1}_t\}_{t=1}^{M_1} \in \mathbb{C}^{2m \times 2n}$. Then each Eigenmatrix $f^{l_1}_t$ is sampled by $k_1 \times k_2$ sliding window, and $2m \times 2n$ sampling blocks can be obtained. The $l_1$-th sampling Eigenmatrix $z^{l_1}_t = [z^{l_1,1}_t, z^{l_1,2}_t, \cdots, z^{l_1,4mn}_t] \in \mathbb{C}^{k_1k_2 \times 4mn}$ is obtained by means removal and series connection of all sampling blocks. Then the same operation can be performed for each matrix in $\{f^{l_1}_t\}_{t=1}^{M_1}$, and the second layer input matrix $Z^t = [z^{1}_t, z^{2}_t, \cdots, z^{M_1}_t] \in \mathbb{C}^{k_1k_2 \times 4M_1mn}$ corresponding to the $t$-th color image sample can be obtained. Finally, the sample matrix of the second convolution input of $N$ image samples can be obtained.

$$Z = [Z^1, Z^2, \cdots, Z^{M_1}] \in \mathbb{C}^{k_1k_2 \times 4M_1mn}$$  \hspace{1cm} (10)

Similarly, the quaternion sparse convolution kernel of the second convolution can also be calculated through Equations (6) and (7). The second QSPCA convolution kernel is expressed as follows:

$$W^2_{l_2} = \text{matrix}_{k_1,k_2} \left( v^2_{sl_2} \right) \in \mathbb{C}^{k_1 \times k_2}$$  \hspace{1cm} (11)

where $l_2 = 1, 2, \cdots, M_2$, $v^2_{sl_2}$ represents the $l_2$-th principal component vector in the second quaternion sparse vector matrix.

For $M_1$ feature matrices output after the first convolution, each feature matrix $F^{l_1}_t$ corresponds to $M_2$ feature matrices after the second convolution

$$R^{l_1}_t = \{F^{l_1}_t \ast W^2_{l_2}\}_{l_2=1}^{M_2}$$  \hspace{1cm} (12)

Therefore, each input sample $\chi_t$ can obtain $M_1 \times M_2$ feature matrices after feature extraction of QSPCANet two-layer convolution.

2.1.4. Pooling and Feature Output

The quaternion sparse feature matrix obtained from the second-order convolution layer can be used as the feature output of the sample only after the pooling operation. Therefore, each feature matrix of the convolution output is input to the binarization function first, and then the binarization feature matrix is encoded. Each output feature matrix is obtained by convolving the input sample with different quaternion sparse convolution kernels. The larger the Eigenvalue of the convolution kernel is, the greater its contribution is. Therefore, the corresponding output feature matrix should also be given greater weight, and the weighted feature matrix can be obtained as follows:

$$\Gamma^{l_1}_t = \sum_{l_2=1}^{M_2} 2^{l_2-1} H\left(F^{l_1}_t \ast W^2_{l_2}\right)$$  \hspace{1cm} (13)

where $H(\cdot)$ represents the given binarization function, that is, it is set to 0 when the modulus of the element is less than the given threshold, and 1 otherwise. After the above binarization hash coding, the $M_1 \times M_2$ Eigenmatrices obtained by the second convolution
become $M_1$ Eigenmatrices, and the pixel values in each Eigengraph are integers with a range of $[0, 2^{M_2} - 1]$.

Finally, we form the area histogram statistics for $M_1$ matrices corresponding to color image $\chi_t$. Each matrix is divided into $C$ blocks, and then the statistical interval is set as $2^{M_2}$. The information in each histogram block is counted and connected in series to obtain the histogram feature $T_l^i \in R^{2^{M_2}C}$ of each feature matrix $T_l^i$. Finally, the histogram feature vector of each input sample is output as:

$$f_t = [T^1_t, T^2_t, \cdots, T^{M_1}_t]^T \in R^{2^{M_2}CM_1} \quad (14)$$

The sample can output the final quaternion sparse feature matrix after feature extraction by QSPCANet. Then, in order to be consistent with other PCANet methods, the SVM classifier is also used to realize color face recognition. The advantage of SVM is that it can effectively solve the problems of a small sample, nonlinear and high-dimensional regression, and classification. Compared with the complexity of the problem, SVM requires a relatively small number of samples; a case where sample data are linearly indivisible can be solved by kernel function and relaxation variable. High dimension means that the sample dimension is very high, because the classifier generated by SVM is very simple, and the sample information only uses a support vector. At the same time, because SVM is only determined by support vector and has its own $L_2$ regularization, it can effectively prevent the over-fitting problem.

Annotation 2.2: the QNSPCANet method described in this section is a new network model proposed in this chapter, especially the $L_p$ regularization of strong sparsity. Moreover, the computational complexity of QNSPCANet feature extraction is $4mnM_1(M_2 + 1)k_1k_2$, and the convolution storage space is $8k_1k_2(M_1 + M_2)$ bytes. Compared with other methods, QNSPCANet has the following advantages:

1. QNSPCANet uses $L_p$ regularization to compute sparse convolution kernels, which has higher sparse efficiency and can reduce computational complexity compared with general $L_1$ regularization;
2. Sparse regularization is beneficial to identify important variables related to outliers, while the principal component convolution check outliers calculated by non-convex regularization of strong sparsity have better robustness and improve model recognition performance;
3. For the image with occlusion, the sparse principal component convolution kernel can reduce the influence of outliers in the occlusion area and further improve the recognition accuracy.

However, the QNSPCANet model established in this section has non-convex and non-smooth problems in sparse optimization, which makes it difficult to solve. Although the establishment of an alternate solution model can effectively overcome the difficulty of solving two variables simultaneously and reduce the complexity to a certain extent, it still cannot overcome the essential difficulty brought by the introduction of $L_p$ norm. Therefore, we discuss the solution method for $L_p$ non-convex optimization problems in the next section.

2.2. $L_p$ Non-Convex Sparse Optimization Method for Model Parameters

In order to overcome the difficulty in solving the parameters of QNSPCA convolution kernel, the variables of Equation (7) are first divided into two coordinate blocks, $A$ and $B$, and one coordinate block is fixed to solve the sub-problems of the other coordinate block, and the sub-problems of the two variables are solved in turn until the termination condition is met [11]. In the algorithm, the initial value of $A$ and $B$ are the first $k$ principal components obtained from QPCA, that is, $A = B = [v_1, v_2, \cdots, v_k]$. 
Fixing $A$ to solve problem (7) is equivalent to solving:

$$
\hat{B} = \arg\min_B \left( \|XA - XB\|_F^2 + \lambda_2 \sum_{j=1}^k \|\beta_j\|_2^2 + \sum_{j=1}^k \lambda_{1,j} \|\beta_j\|_p^p \right)
$$

(15)

$Y = XA$, and $y_j = X_{ij}, j = 1, \ldots, k$ is initialized, then solving Equation (15) is equivalent to solving $k$ independent optimization problems:

$$
\hat{\beta}_j = \arg\min_{\beta_j} \left( \|y_j - X\beta_j\|_F^2 + \lambda_2 \|\beta_j\|_2^2 + \lambda_{1,j} \|\beta_j\|_p^p \right)
$$

(16)

Based on the obtained $B$, the singular value decomposition of $XX^H B$ is calculated:

$$
XX^H B = UDV^T
$$

(17)

And the $A$ is updated:

$$
\hat{A} = UV^T
$$

(18)

$B$ and $A$ can continue to be solved alternately until the convergence.

Annotation 2.3: the establishment of the alternate solution model can effectively overcome the difficulty of solving two variables at the same time and reduce the complexity to a certain extent, but it still cannot overcome the essential difficulty caused by the introduction of the $L_p$ norm. Therefore, we introduce the coordinate descent method and fixed point iterative method in the next section to overcome this difficulty.

2.2.1. Coordinate Descent

Coordinate descent (CD) [12] is a simple but efficient optimization algorithm. It does not calculate gradient, but minimizes the objective function along the direction of each coordinate axis; that is, only one coordinate direction is found at this point and the remaining coordinate directions are kept unchanged. Then we can iterate over each coordinate axis; that is, only one coordinate direction is found at this point and does not calculate gradient, but minimizes the objective function along the direction of

$$
X^{H}B = UV^T
$$

(17)

and the $A$ is updated:

$$
\hat{A} = UV^T
$$

(18)

$B$ and $A$ can continue to be solved alternately until the convergence.

If there is an objective function $f(x_1, x_2, \ldots, x_n)$, and it needs to solve its minimum point, $x = [x_1, x_2, \ldots, x_n]$, $x$ is initialized, it is called $x^0$, and then the cycle is started, the iteration process of the $i$-th dimension in the $t$-th cycle is as follows:

$$
x_i^{(t)} = \arg\min_{x_i} f \left( x_1^{(t)}, x_2^{(t)}, \cdots, x_{i-1}^{(t)}, x_i, x_{i+1}^{(t)}, \cdots, x_n^{(t-1)} \right)
$$

(19)

It is equivalent to solving $x_i$ only as a variable in each iteration, while the remaining $n - 1$ dimensions are regarded as constants and keep the current value unchanged. Then, $f(x_1, x_2, \ldots, x_n)$ is minimized to obtain the new value of $x_i$, and it is substituted into the next iteration as a constant.

If the relative changes in $x^{(t)}$ and $x^{(t-1)}$ in each dimension are less than the specified threshold, the $x^{(t)}$ is the final result. Otherwise, the cycle continues for the $t + 1$-th time until it is less than the change threshold or reaches the maximum number of cycles, and finally reaches the local minimum point.

For $k$ principal components, these are $k$ independent problems, so we only need to provide the coordinate descent solution algorithm for the first optimization problem, and so on for the other $k - 1$ problems. Without loss of generality, the subscripts of $\beta_j$ and $y_j$ are omitted, i.e., $\beta \in C^{k \times 1}$, $y \in C^{N \times 1}$. Equation (16) can be simplified as

$$
\hat{\beta} = \arg\min_\beta \left( \|X^H \beta - y\|_2^2 + \lambda_2 \|\beta\|_2^2 + \lambda_p \|\beta\|_p^p \right)
$$

(20)

The coordinate descent method is applied to Equation (20). In the process of each cycle iteration, only one target variable is minimized and the values of other variables are fixed,
which is equivalent to solving the unitary optimization problem. Therefore, for the $i$-th component $\beta_i, i = 1, 2, \cdots, k_1k_2$ of $\beta$, problem (20) is solved, the current value of $\beta_i(s \neq i)$ is unchanged, and only $\beta_i$ is optimized in each iteration, which is equivalent to solving:

$$
\hat{\beta}_i = \arg\min_{\beta_i} \left( \left( \lambda_2 + \sum_{j=1}^{4Nmn} x_{ij}^2 \right) \beta_i^2 - \left( \sum_{j=1}^{4Nmn} x_{ij} r_j(i) \right) \beta_i + \lambda_p |\beta_i|^p \right)
$$

(21)

where $r_j(i) = y_j - \sum_{s \neq i} x_{sj} \beta_s, j = 1, 2, \cdots, 4Nmn$, that is, the residual of $y_j$ is fitted only with other fixed variables.

Annotation 2.4: In reference [11], this iterative updating method is called trivial updating, and the computational complexity of this method is $O(mn)$. The coordinate descent method adopted in this paper has low complexity, and the number of cycles selected by the algorithm termination condition is less than the maximum number of cycles, or the update rate of the objective function value is less than the given threshold. However, in the process of solving a single component, the introduction of $L_p$ regularization is still unavoidable.

2.2.2. Fixed Point Iterative Method

In order to overcome the univariate solving problem caused by $L_p$ regularization, the fixed point iterative method is used to optimize the numerical solution in this section.

Theorem 1. Given a function $f(x_1, \cdots, x_i, \cdots, x_n)$, suppose $x_i$ exists such that $x_i = g(x_i)$, then the point $x_i$ is a fixed point of $f(x_1, \cdots, x_i, \cdots, x_n)$.

For Equation (21), $f(\beta_1, \cdots, \beta_i, \cdots, \beta_n) = b \beta_i^2 - a \beta_i + \lambda_p |\beta_i|^p$, the first derivative equation of it can be expressed as follows:

$$
\begin{align*}
(b \beta_i^2 - a \beta_i + \lambda_p |\beta_i|^p)' &= 0 \\
\Rightarrow & \begin{cases} 
2b \beta_i - a + p \lambda_p \beta_i^{p-1} = 0 & \beta_i > 0 \\
2b \beta_i - a - p \lambda_p (-\beta_i)^{p-1} = 0 & \beta_i < 0 
\end{cases}
\end{align*}
$$

(22)

Since the function is non-convex after $L_p$ regularization is added, the variable optimization solution cannot be obtained directly from the first derivative equation. Therefore, the fixed point iteration method is used to transform the first-order derivation problem into the fixed point iteration problem $\beta_i = g(\beta_i)$, and the equivalent deformation of Equation (22) can be obtained:

$$
\beta_i = \begin{cases} 
\frac{a}{2b} - \frac{p \lambda_p \beta_i^{p-1}}{2b} & \beta_i > 0 \\
\frac{a}{2b} + \frac{p \lambda_p \beta_i^{p-1}}{2b} & \beta_i < 0 
\end{cases}
$$

(23)

Therefore:

$$
\beta_i = \begin{cases} 
\frac{a}{2b} - \frac{p \lambda_p \beta_i^{p-1}}{2b} & \beta_i > 0 \\
\frac{a}{2b} + \frac{p \lambda_p \beta_i^{p-1}}{2b} & \beta_i < 0 
\end{cases}
$$

(24)

where $b = \lambda_2 + \sum_{j=1}^{4Nmn} x_{ij}^2$, $a = \sum_{j=1}^{4Nmn} x_{ij} r_j(i)$. The initial value of the input variable is taken as the initial iteration value, and then the numerical solution is iterated. Therefore, the numerical solution obtained iteratively by using Equation (23) is the optimal solution $\hat{\beta}_i$ of $\beta_i$ under the current cycle.

Annotation 2.5: The fixed point iterative method is an important method for solving nonlinear equations [13]. After transformation into fixed point equation, approximate solution of the equation can be obtained through iteration, which is not limited by nonlinear and non-convex conditions. Therefore, this section adopts fixed point iteration method to solve variables.
3. Algorithm Simulation Experiments

In this section, four color face datasets including Georgia Tech [14], Color FERET [15], AR [16], and LFW-A [17] were selected to conduct algorithm simulation experiments. Color face image samples of four datasets are shown in Figure 2.

![Sample for color face image](image)

Figure 2. Sample for color face image (a) Georgia Tech; (b) Color FERET; (c) AR; (d) and LFW-A.

The Georgia Tech dataset consists of 750 color images, 15 images for each of 50 photographers, and is widely used in face recognition. The face images were taken in the Georgia Tech Lab, and most of the images included changes in light, expression, and posture.

Color FERET is a Color version of the gray face dataset FERET, with 11,338 color face images from 994 people. Since the number of images varies from person to person, 200 photographers were selected with seven images each, and the 1400 images were recorded as the Color FERET subset. Seven images of each person were selected with changes in lighting, expression, and posture.

AR is a face dataset composed of 3120 color face images, 26 images per 120 photographers. Face images were taken from the front, so the pose changes are relatively few. In addition to the illumination and expression changes, the influence of occlusion factors is also considered.

LFW-A is a version of LFW face dataset after face alignment processing, which includes 13,233 color images from 5749 photographers. As the images come from the network, it is suitable for face recognition research in natural scenes. The face images in the dataset include a variety of factors such as illumination, age, posture, expression, and occlusion, so the dataset is very challenging. In order to evenly distribute samples, photographers with more than nine images were selected and A subset of LFW-A was constructed.

The images of each person in each database were randomly divided into the training set and test set in accordance with 2:1. Since the distribution of training samples and test samples in the experiment was random, the experiment was repeated for ten times and the average value was taken. In order to test more effectively in the actual experiment, the size of all images was uniformly set as 32 × 32.
Meanwhile, based on the experimental situation in this paper, the SVM classifier chose the LIBLINEAR library based on linear kernel function. The linear kernel is suitable for cases where the number of samples is much smaller than the number of features (no need to map to higher dimensions), or where both the number of samples and the number of features are large (mainly considering the training speed).

In addition, the hardware of this experiment is Intel(R) Core (TM) i5-8265U CPU@1.80 GHz, NVIDIA GTX 1060, and the software is Matlab 2016b and Anaconda3.

3.1. Comparison of Algorithm Performance under Different Occlusion Conditions

Firstly, the QNSPSCANet proposed in this paper and the three algorithms, PCANet, QPCANet, and QSPCANet, were compared under different occlusion conditions. Different occlusion conditions mainly include occlusion contained in the dataset, self-added pure color occlusion, and salt-and-pepper noise occlusion. At the same time, Color FERET, AR, and LFW-A datasets also compared the recognition accuracy of other latest occlusion algorithms under the same dataset. The Georgia Tech dataset was only compared with similar structure algorithms due to its few applications.

Since there are no large-area occlusion elements in the face images of Georgia Tech, Color FERET, and LFW-A datasets, the experiments of these three datasets can be divided into three groups: (1) when the first group is without occlusion, the images of the training set and test set are randomly selected in 2:1; (2) in the second group, a blue block with 20% pixel area was added to the randomly selected face images in the test set; (3) and the third group is the condition of salt-and-pepper noise occlusion. Salt-and-pepper noise blocks are added to randomly selected face images in the test set.

For AR dataset, each person should have 26 face images, among which, eight have no occlusion factor, six have illumination change, six have sunglasses occlusion, and six have scarf occlusion. The experiment can be divided into five groups: (1) in the first group, under the condition of no occlusion, nine images were randomly selected from 14 images with no occlusion containing light changes to form the training set, and the remaining five images were formed into the test set; (2) in the second group, 14 face images without occlusion were included in the training set, and six face images with sunglasses occlusion were included in the test set; (3) the third group was scarf occlusion, 14 images without occlusion were used as the training set, and six images with scarf were used as the test set; (4) the fourth group was in the condition of self-added pure color occlusion. The training set consisted of nine randomly selected face images without occlusion, and then added blue occlusion blocks (about 20% occlusion area) to the remaining five images, which were used as test samples; (5) and the fifth group was the condition of salt-and-pepper noise occlusion. Nine images were randomly selected from 14 images without occlusion as the training set, and then the remaining five images were added with salt-and-pepper noise blocks as the test set. Color face samples of the AR dataset, self-added solid color occlusion of images, and different areas of salt-and-pepper noise occlusion are shown in Figure 3.

![Color face sample of AR dataset and self-added occlusion processing example.](image)

The convolution order of each principal component analysis network is set as 2, the number of convolution kernels at each layer is $M_1 = M_2 = 8$, the size of sampling matrix is set as the optimal size of each dataset, the size of histogram window is set as $7 \times 7$, and the corresponding overlap rate is set as 0.5. For the quaternion sparse optimization problem in the convolution layer, $\lambda_2 = 0.001$ is set, the appropriate sparse parameter $\lambda_p, \lambda_s$ for each feature vector is selected, the update rate threshold is set as $1e-4$, and the maximum cycle time is set as 1000. Finally, a trained SVM classifier is used for color face recognition based on the extracted features.
Table 1 shows the correct recognition rate of each algorithm in Georgia Tech dataset under different occlusion conditions.

Table 1. The correct recognition rate of each algorithm in Georgia Tech dataset under different occlusion conditions (%).

| Algorithm      | Normal  | 20% Block Occlusion | 20% Noise Occlusion |
|----------------|---------|---------------------|---------------------|
| PCANet         | 93.20   | 83.60               | 83.00               |
| QPCANet        | 95.50   | 87.30               | 87.70               |
| QSPCANet       | 97.50   | 92.50               | 91.40               |
| QNSPCANet      | 97.70   | 95.40               | 94.40               |

It can be seen from Table 1 that the algorithm introducing $L_p$ non-convex regularization achieves a relatively high recognition rate under different occlusion conditions. When there is no occlusion, the recognition rate of the two algorithms under sparse constraint is close. In the case of 20% pure color occlusion area and salt-and-pepper noise occlusion, the recognition rate of QNSPCANet is the highest. Due to the few occlusion applications in Georgia Tech dataset, this part is only compared with the relevant PCANet method.

The correct recognition rate of each algorithm in Color FERET dataset under different occlusion conditions is shown in Table 2.

Table 2. The correct recognition rate of each algorithm in Color FERET dataset under different occlusion conditions (%).

| Algorithm     | Normal  | 20% Block Occlusion | 20% Noise Occlusion |
|---------------|---------|---------------------|---------------------|
| FRAD [18]     | 96.59   | 94.80               | 94.33               |
| GMSRC [19]    | 97.07   | 95.21               | 95.42               |
| DDRC [20]     | 98.60   | 94.53               | -                   |
| PCANet        | 93.75   | 84.13               | 84.42               |
| QPCANet       | 96.44   | 88.52               | 88.11               |
| QSPCANet      | 98.04   | 92.98               | 92.38               |
| QNSPCANet     | 98.72   | 96.02               | 95.39               |

As can be seen from Table 2, in the Color FERET dataset, QNSPCANet achieves the highest recognition rate in the case of non-noise occlusion. In the case of 20% salt-and-pepper noise, the recognition rate is slightly lower than GMSRC, but the overall recognition performance is still superior.

The correct recognition rate of each algorithm in AR dataset under different shielding conditions is shown in Table 3. The recognition rate of all algorithms in AR dataset is generally higher. QNSPCANet has the best recognition effect under all occlusion conditions, and the difference of recognition rate is larger when there is occlusion, indicating that $L_p$ regularization has a good suppression effect on both outliers and occlusion.

Table 3. The correct recognition rate of each algorithm in AR dataset under different occlusion conditions (%).

| Algorithm     | Normal  | Sunglasses | Scarf  | 20% Block Occlusion | 20% Noise Occlusion |
|---------------|---------|------------|--------|---------------------|---------------------|
| Gabor-SRC [21]| 96.72   | 95.83      | 95.26  | 92.41               | 92.89               |
| VGG-Face [22] | 98.05   | 97.21      | 93.40  | 89.47               | 91.55               |
| Lightened-CNN [22]| 98.79     | 98.14      | 98.56  | 96.01               | -                   |
| PCANet        | 96.71   | 95.83      | 96.12  | 87.14               | 88.32               |
| QPCANet       | 97.83   | 97.66      | 96.23  | 90.69               | 90.94               |
| QSPCANet      | 99.41   | 98.50      | 98.74  | 93.56               | 92.71               |
| QNSPCANet     | 99.62   | 99.25      | 99.31  | 96.77               | 95.80               |
The correct recognition rate of each algorithm in LFW-A dataset under different shielding conditions is shown in Table 4. It can be seen from the table that QNSPCANet, which introduced non-convex regularization, has a higher recognition rate than PCANet and QPCANet under different occlusion conditions. Compared with QSPCANet based on $L_1$ regularization, the recognition rate is close to that of QSPCANet without occlusion, while the recognition rate is greatly improved with occlusion. At the same time, compared with other existing methods in the table, the algorithm proposed in this paper can also achieve better recognition effect. Although there is still a certain gap between QNSPCANet and MobileFaceNet when there is no occlusion, QNSPCANet has better performance when there is occlusion and noise.

Table 4. The correct recognition rate of each algorithm in LFW-A dataset under different occlusion conditions (%).

| Algorithm       | Normal  | 20% Block Occlusion | 20% Noise Occlusion |
|-----------------|---------|---------------------|---------------------|
| ProCRC [23]     | 94.82   | 86.77               | 88.51               |
| CRDDL [23]      | 95.20   | 89.56               | 90.13               |
| MobileFaceNet [24]| 98.20  | 90.53               | 95.08               |
| PCANet          | 91.56   | 80.25               | 80.64               |
| QPCANet         | 94.15   | 86.33               | 87.59               |
| QSPCANet        | 97.10   | 92.17               | 93.36               |
| QNSPCANet       | 97.35   | 95.08               | 95.21               |

Figure 4 shows the recognition rate curves of each algorithm under different solid color occlusion areas. As can be seen from the figure, with the increase in the occlusion area, the gap between the correct recognition ability of the four recognition algorithms gradually increases and QNSPCANet performs better under different occlusion areas.

Figure 4. The recognition rate curves of each algorithm under different solid color occlusion areas.
Since no specific training time is provided in the literature of other occlusion methods, this paper only compares the training time of the four PCANet algorithms, as shown in Table 5.

Table 5. Algorithm training time comparison.

| Algorithm       | Georgia Tech | Color FERET | AR        | LFW-A     |
|-----------------|--------------|-------------|-----------|-----------|
| PCANet          | 49.06 s      | 81.82 s     | 210.27 s  | 63.15 s   |
| QPCANet         | 156.24 s     | 277.52 s    | 342.86 s  | 180.44 s  |
| QSPCANet        | 194.36 s     | 325.58 s    | 481.39 s  | 237.89 s  |
| QNSPCANet       | 226.57 s     | 369.14 s    | 510.44 s  | 268.51 s  |

As can be seen from Table 5, the overall training time of the four PCANet related algorithms is short. The training time of the QNSPCANet method proposed in this paper is increased, mainly because non-convex sparse optimization produces certain calculation consumption, but the recognition accuracy is improved, and the overall recognition performance is still superior.

Based on the above comparative experimental results, it can be shown that in the case of no occlusion, the recognition rate of non-convex sparse convolution check is improved, but the effect is not obvious because the image contains fewer outliers at this time, which has a slight impact on model recognition results. In the case of occlusion, the strong sparse performance of non-convex regularization can effectively reduce the influence of occlusion outliers to improve the identification accuracy of the model.

3.2. Algorithm Sparsity Verification

Secondly, the sparsity of QNSPCANet proposed in this paper is verified. The validation experiment is mainly carried out on AR datasets and compared with $L_1$ regularization and SCAD regularization [25]. In order to test the sparsity of $L_{1/2}$ non-convex regularizer ($L_p$ is the most representative of sparsity), this section adopts two sparsity measures: the proportion of non-zero elements in the sparse matrix and Hoyer’s sparsity measure. Hoyer’s sparsity measures the components of small values and is a more refined sparse measurement method than non-zero values, which is defined as:

$$
Hoyer(V_i^j) = \frac{\sqrt{M_i k_1 k_2} - \|V_i^j\|_1 / \|V_i^j\|_F}{\sqrt{M_i k_1 k_2} - 1} \in [0, 1]
$$

(25)

where $V_i^j$ represents the quaternion sparse vector matrix in the $i$-th convolution layer, and $M_i k_1 k_2$ represents the number of elements of the matrix.

The proportional convergence curve of non-zero elements of the sparse matrix and the convergence curve of Hoyer’s sparsity are shown in Figure 5.

![Convergence Curve of Non-zero Element Ratio](image1)

![Convergence Curve of Hoyer’s Sparsity](image2)

Figure 5. Convergence curve of sparsity of the first convolution layer sparse vector matrix.
As can be seen from Figure 5, the proportion of $L_1$ regularization non-zero elements is 0.56, while the Hoyer’s sparsity is about 0.70. The proportion of non-zero elements convergent in $L_{1/2}$ regularization is 0.41, and the Hoyer’s sparsity is about 0.77. The proportion of non-zero elements regularized by SCAD is 0.46, and the Hoyer’s sparsity is about 0.75. Compared with $L_1$ regularization, $L_{1/2}$ regularization decreased by 26.8% and SCAD regularization decreased by 17.9%. Compared with $L_1$ regularization, $L_{1/2}$ regularization improved Hoyer’s sparsity by 10.0%, while SCAD regularization improved by 7.1%. Therefore, $L_{1/2}$ regularization has stronger sparsity than SCAD and $L_1$ regularization.

3.3. Algorithm Robustness Verification

In order to verify the robustness of the model under $L_{1/2}$ non-convex regularization, different samples were randomly selected to form the training set and test set under different salt-and-pepper noise occlusion areas. The input model was repeated ten times and the corresponding root mean square error (RMSE) was calculated. Root mean square error can effectively reflect the stability of the model for random sample changes, and its calculation results are shown in Table 6.

| Algorithm   | 10%  | 20%  | 30%  | 40%  | 50%  |
|-------------|------|------|------|------|------|
| PCANet      | 4.20 | 5.73 | 7.82 | 9.02 | 9.85 |
| QPCANet     | 2.51 | 4.26 | 5.33 | 7.84 | 9.26 |
| QSPCANet    | 0.94 | 1.12 | 1.58 | 1.93 | 2.29 |
| QNSPCANet   | 0.79 | 0.98 | 1.24 | 1.57 | 2.01 |

Then, a certain degree of rotation transformation and translation transformation were carried out on the samples of the test set to construct similar samples to test the stability of the model for transformation. For the image rotation transformation processing, the overall rotation angle increases within the range of $-20^\circ \sim +20^\circ$; for translation transformation processing, $-10 \sim +10$ pixels of image can be translated based on the horizontal direction. Figure 6 shows the correct recognition rate curves of the four algorithms under different rotation and translational transformations.

![Recognition Rate of Rotation Transformation](image1)

**Figure 6.** Four algorithms correctly identify rate curves under rotation transformation and translational transformation.

Combined with the results in Table 5 and Figure 6, it can be shown that QNSPCANet still has a low root mean square error in the case of noise. Meanwhile, QNSPCANet has good robustness to translation and rotation changes in samples. It can be seen that the proposed algorithm has higher stability, and the model is more robust.
4. Conclusions

In this paper, we propose a QNSPCANet model based on quaternion non-convex sparse constraint mechanism for color face image recognition with large area occlusion. $L_p$ regularization is introduced into PCANet as a constraint term in the convolution layer convolution kernel sparse optimization problem, and the non-convex optimization problem is innovatively converted into fixed point equation, avoiding the problem that the coordinate descent method cannot be directly derived in the single variable solution. Sparse regularization recognition of outliers can effectively solve the problems of face image occlusion and noise, and the strong sparsity of non-convex regularization can further improve the recognition performance of the model.

In order to verify the recognition performance of QNSPCANet model proposed in this paper, especially in the case of occlusion and noise, self-added occlusion and noise processing are performed on Georgia Tech, Color FERET, AR, and LFW-A Color face datasets and compared with PCANet, QPCANet, QSPCANet, and other latest occlusion algorithms in the same dataset. The experimental results show that the quaternion non-convex sparse principal component analysis network proposed in this chapter has a high recognition rate under different occlusion conditions. This paper also verifies the sparsity and robustness of $L_p$ regularization, and further proves the sparsity and robustness of $L_p$ regularization through experiments of sparsity, root mean square error, rotation, and translation transformation, etc.

The PCANet framework based on this paper adopts a relatively simple number of network structure layers and lacks deeper feature extraction. If the deep features are acquired only by increasing the layers of the network model, the number of parameters and computational complexity will also be greatly increased. Therefore, it is necessary to build a new framework that is more advanced, simple and can extract deeper features.

Author Contributions: Conceptualization, Y.Q.; data curation, Y.Q.; formal analysis, C.W.; funding acquisition, C.W.; methodology, Y.Q. and C.W.; project administration, Y.Q.; resources, C.W.; supervision, C.W.; validation, C.W.; writing—original draft, Y.Q.; writing—review and editing, Y.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grants 61933013, 61733015 and 61733009.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Li, Z. Face recognition technology research status review. Electron. Technol. Softw. Eng. 2020, 13, 106–107.
2. Zhao, K.; Jin, X.; Wang, Y. A review of research on small sample learning. J. Softw. 2021, 32, 349–369.
3. Chan, T.-H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A Simple Deep Learning Baseline for Image Classification? IEEE Trans. Image Process. 2015, 24, 5017–5032. [CrossRef] [PubMed]
4. Bao, S.; Song, X.; Hu, G.; Yang, X.; Wang, C. Colour face recognition using fuzzy quaternion-based discriminant analysis. Int. J. Mach. Learn. Cybern. 2019, 10, 385–395. [CrossRef]
5. Turk, M.; Pentland, A. Eigenfaces for recognition. J. Cogn. Neurosci. 1991, 3, 71–86. [CrossRef] [PubMed]
6. Torres, L.; Reutter, J.; Lorente, L. The importance of the color information in face recognition. In Proceedings of the 6th IEEE International Conference on Image Processing, Kobe, Japan, 24–28 October 1999; pp. 627–631.
7. Yang, J.; Liu, C. Color Image Discriminant Models and Algorithms for Face Recognition. IEEE Trans. Neural Netw. 2008, 19, 2088–2098. [CrossRef] [PubMed]
8. Li, Y.; Zhu, S.; Zhu, L. Face recognition algorithm based on quaternion principal component analysis. Signal Processing 2007, 2, 214–216.
9. Xiao, X.; Zhou, Y. Two-Dimensional Quaternion PCA and Sparse PCA. IEEE Trans. Neural Netw. Learn. Syst. 2019, 30, 2028–2042. [CrossRef] [PubMed]
10. Xu, Z.; Zhang, H.; Wang, Y.; Change, X.; Liang, Y. L_{(1/2)} regularization. Sci. China Inf. Sci. 2010, 53, 1159–1169. [CrossRef]
11. Yang, X. Two-stage method for sparse principal component analysis. Prog. Appl. Math. 2017, 6, 1174–1181.
12. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw. APubl. IEEE Neural Netw. Counc.* 2008, 20, 61–80. [CrossRef] [PubMed]
13. Huang, X.; Ren, M.; Hu, Z. An Improvement of K-Medoids Clustering Algorithm Based on Fixed Point Iteration. *Int. J. Data Warehous. Min.* 2020, 16, 84–94. [CrossRef]
14. Georgia Tech Face Database. Available online: http://www.anefian.com/research/face_reco.html (accessed on 20 May 2022).
15. Color FERET Database. Available online: https://www.nist.gov/itl/products-and-services/color-feret-database (accessed on 20 May 2022).
16. Martinez, A.; Benavente, R. *The AR Face Database*; Technical Report; The Computer Vision Center (CVC): Barcelona, Spain, 1998.
17. Gary, B.; Manu, R.; Tamara, B.; Erik, L. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*; University of Massachusetts: Amherst, MA, USA, 2007; pp. 7–49.
18. Asem, M. A 3D-based Pose Invariant Face Recognition at a Distance Framework. *IEEE Trans. Inf. Forensics Secur.* 2014, 9, 2158–2169.
19. Cao, L.; Li, H.; Guo, H.; Wang, B. Robust PCA for Face Recognition with Occlusion Using Symmetry Information. In Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 9–11 May 2019; pp. 323–328. [CrossRef]
20. Cen, F.; Wang, G. Dictionary Representation of Deep Features for Occlusion-Robust Face Recognition. *IEEE Access* 2019, 7, 26595–26605. [CrossRef]
21. Yang, M.; Zhang, L. Gabor Feature Based Sparse Representation for Face Recognition with Gabor Occlusion Dictionary. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 448–461. [CrossRef]
22. Wang, Y. Detection and Recognition of Occluded Face Based on Deep Neural Network; Northeast Petroleum University: Heilongjiang, China, 2021; pp. 30–32.
23. Lv, S.; Liang, J.; Di, L.; Xia, Y.; Hou, Z. A probabilistic collaborative dictionary learning-based approach for face recognition. *IET Image Process.* 2021, 15, 868–884. [CrossRef]
24. Chen, S.; Liu, Y.; Gao, X.; Han, Z. MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices. In Proceedings of the Chinese Conference on Biometric Recognition, Urumqi, China, 11–12 August 2018; pp. 428–438.
25. Fan, J.; Li, R. Statistical Challenges with High Dimensionality: Feature Selection in Knowledge Discovery. In Proceedings of the International Congress of Mathematicians, Madrid, Spain, 22–30 August 2006; pp. 595–622.