Face Recognition Using Dictionary Learning Algorithms

by

Mohammad Mehdi Khalili
B.Sc., Iran University of Science and Culture, 2007
M.Sc., Tehran Polytechnic, 2011

A Report Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

Supervisory committee

Face Recognition Using Dictionary Learning Algorithms

by

Mohammad Mehdi Khalili

B.Sc., Iran University of Science and Culture, 2007

M.Sc., Tehran Polytechnic, 2011

Supervisory committee

Dr. T. Aaron Gulliver, Department of Electrical and Computer Engineering, University of Victoria (Supervisor)

Dr. Amirali Baniasadi, Department of Electrical and Computer Engineering, University of Victoria (Departmental Member)

Supervisory committee

Dr. T. Aaron Gulliver, Department of Electrical and Computer Engineering, University of Victoria (Supervisor)

Dr. Amirali Baniasadi, Department of Electrical and Computer Engineering, University of Victoria (Departmental Member)

ABSTRACT

Face recognition is one of the most challenging and important topics in computer vision, pattern recognition and image processing. It has experienced a recent advance by using dictionary learning algorithms. These algorithms benefit from sparse coding techniques to achieve more accurate and faster classifications. Three dictionary learning algorithms for face recognition, Label Consistent K-SVD (LC-KSVD), Fisher Discriminative Dictionary Learning (FDDL), and Support Vector Guided Dictionary Learning (SVGDL), are investigated in this project. The reason for choosing these algorithms is their high accuracy in dictionary learning based image recognition. Accuracy, speed, and variability are used as measures to test these algorithms. The number of training images, atoms, and iterations are considered as parameters in order to evaluate the algorithms. The extended Yale B image database is used for testing. Simulations are performed using MATLAB. The results obtained indicate that SVGDL is the best algorithm followed by LC-KSVD and then FDDL.

Contents

List of Figures

Glossary

D	Dictionary
DL	Dictionary Learning
DDL	Discriminative Dictionary Learning
FDDL	Fisher Discriminative Dictionary Learning
GC	Global Classifier
K-SVD	K-means SVD
LC	Local Classifier
LC-KSVD	Label Consistent K-SVD
SV	Support Vector
SVD	Singular Value Decomposition
SVGDL	Support Vector Guided Dictionary Learning
SVM	Support Vector Machine

ACKNOWLEDGMENTS

I would like to express my deepest thanks to my supervisor Dr. T. Aaron Gulliver for his patience, kindness, guidance, support, highly valuable advice and helpful comments on my project. He has always been open and honest in communicating with me and other students and I would have never completed my degree without his supervision. I would also like to express my gratitude to Dr. Amirali Baniasadi for being on my supervisory committee and for providing useful knowledge in my field of study. He has always encouraged me to continue my studies. His advice and viewpoints have been a guide for me to interact with others and live happily in Victoria. Finally, I would like to thank my parents and my friends for their support, patience and motivation through my studies away from my homeland.

Chapter 1

Introduction

Humans typically use faces to recognize people so it is not surprising that face recognition has become very important in the modern digital world. In recent years, biometric based techniques have emerged as the most important option for recognizing individuals. These techniques examine an individual's physical characteristics in order to determine identity instead of using passwords, PINs, smart cards, tokens or keys. Passwords and PINs are hard to remember and can be stolen or guessed easily. Cards, tokens, and keys can be misplaced, forgotten and duplicated, and magnetic cards can become corrupted and unreadable. However, biological traits cannot be forgotten, misplaced or stolen.

Face recognition is used to identify or verify a person by comparing and analyzing patterns that are based on facial features. These features include the eyes, ears, nose, lips, chin, teeth and cheeks. Some of these features are used to recognize individuals. Face recognition is mainly used for security purposes, but there has been increasing interest in other areas. Compared to other biometric recognition techniques, face recognition has many advantages [1]. Facial images can be obtained easily with an inexpensive camera as opposed to other biometrics like the retina and iris that require the use of more expensive equipment. The working range is larger than other methods such as fingerprints, iris scanning and signatures. Facial recognition is used for entry and exit to secure places such as borders, military bases and nuclear power plants. It is also used to access restricted resources such as computers, networks, personal devices, banking transactions, trading terminals and medical records. Face recognition is also be used in the automobile industry. For instance, companies such as Toyota are developing sleep detectors based on face recognition to increase safety. It is a non-contact technique as images are captured and then analyzed without requiring any interaction with the person. Compared with other biometric techniques, face recognition is an inexpensive technology as less processing is required [2].

1.1 Applications

Face recognition is an excellent technique for tracking time and attendance. It can be used in military and medical applications, mobile phones and automobiles, airports and other places [3]. Face recognition is used to unlock the iPhone X and XS phones. In military applications, data confidentiality is very important, so face recognition is used to verify users in order to access information. In medical centers, face recognition is used to access patient information. This allows doctors to easily check patient health records. Marketers and advertisers often consider factors such as gender, age, and ethnicity when targeting groups for a product or area, and face recognition can be used to define these audiences. At universities and colleges, face recognition can be used during exams and classes to identify students. Today, face recognition is used to detect passport fraud, support law enforcement, identify missing children, and minimize business and identity fraud. Systems based on face recognition can be used in airports, multiplexes, and other public places to detect criminals among the crowds.

1.2 Limitations

There are several limitations for face recognition [1-3].

Face aging: Over time changes happen to the human body and thus also to the face because of hormonal and biological changes.

Accidents: The face of a person can change due to an accident.

Cosmetic surgery: Many people undergo plastic or cosmetic surgery to change their faces.

Pose: Rotation can change the appearance of a face.

Lighting conditions: Background light, brightness, contrast or shadows can change the appearance of a face.

Accessories: Accessories such as glasses, nose rings and beards can affect face recognition.

Permission: The permission of a person is often needed to take an image.

Policies: Policies on access to a database must be determined and clearly stated.

1.3 Dictionary Learning

Face recognition is done by comparing selected features within an image with other images in a database. The facial features are extracted from each image and stored. Linear combinations of these features as well as the features themselves are stored as atoms which are used to build a dictionary (D) and has an important impact on classification performance. A dictionary is able to effectively model the pose, illumination and facial expression information including the corresponding variations so an image can be represented by atoms of the dictionary [4]. Training images are used to build a dictionary and are optimized and classified by an objective function of each algorithm which leads to have several classes. Each class has specific or main characteristics of the face images. The dictionary is used to find a sparse representation of the input images. This process is called sparse coding and will be presented in Section 1.4.

Dictionary Learning (DL) algorithms have been used for image processing and classification as well as face recognition [4, 5]. Discriminative Dictionary Learning (DDL) algorithms learn a dictionary through training images of all classes to improve the classification performance. Thus DL should have discriminative ability for all classes. The dictionary is constructed by minimizing the error such as the discriminative sparse code error which is explained in Section 1.4. In DDL algorithms, the discrimination of the dictionary is enforced by either imposing structural constraints on the dictionary or imposing a discrimination term on the coding vectors [6, 7].

In this project, three face recognition algorithms are used, LC-KSVD which is a shared dictionary learning algorithm, and FDDL and SVGDL which are class-specific dictionary learning algorithms. A shared dictionary learning algorithm can capture the common characteristics of face images, but cannot usually capture specific characteristics of the images in each class [8]. When the inter-class variations of the images are large, a dictionary can adequately capture the main characteristics of the images. Then a shared dictionary learning algorithm can learn a dictionary for all classes while the number of atoms is small. However, class-specific dictionary learning algorithms learn a sub-dictionary for the face images in each class and so capture particular characteristics of the images in a class [9]. Because the images of a person vary due to poses and expressions as well as illumination, the intra-class variation of face images is usually large and can be even greater than the inter-class variance.

1.4 Sparse Coding for Classification

Sparse coding has been successfully applied to a variety of problems in computer vision and image analysis, including image de-noising, image restoration, and image classification [10]. Sparse coding approximates a training image (y) by a linear combination of a few atoms sparsely selected from a dictionary. The performance of sparse coding relies on the quality of D. Employing a dictionary of training images for discriminative sparse coding has achieved good face recognition performance [11]. The dictionary is constructed by minimizing the reconstruction error and satisfying the sparsity conditions. Let Y be a set of N n-dimensional training images, $Y = [y_1, y_2, \dots, y_N] \in R^{n\times N}$. Learning a dictionary with K atoms for sparse representation of X based on Y can be achieved as [12]

$$X \ = \arg\min_{X} ||Y - DX||_2^2 \qquad s.t. \forall i, \|x_i\|_0 \leq T \tag{1}$$

where $D = [d_1, d_2, \dots, d_K] \in R^{n\times K}$ $(K > n)$ is the dictionary, $X = [x_1, x_2, \dots, x_N] \in R^{K\times N}$ is the sparse code of the training images Y, and T is the sparsity constraint. The term $||Y - DX||_2^2$ is the reconstruction error.

1.5 Report Outline

Chapter 1 provided a brief introduction to face recognition and its applications and limitations, as well as dictionary learning and sparse coding for classification. Chapter 2 introduces three face recognition algorithms, namely Label Consistent K-SVD (LC-KSVD), Fisher Discriminative Dictionary Learning (FDDL), and Support Vector Guided Dictionary Learning (SVGDL). Chapter 3 provides simulation results for these algorithms regarding the accuracy, speed and variability. Finally, some conclusions and suggestions for future work are presented in Chapter 4.

Chapter 2

Methodology

In this chapter, three face recognition algorithms, LC-KSVD, FDDL and SVGDL, are described in detail. The reason for choosing these algorithms is their accuracy in dictionary learning based image recognition [4].

2.1 Label Consistent K- SVD (LC-KSVD)

The K-SVD algorithm is one of the most well-known shared dictionary learning algorithms. Many variants of the original K-SVD algorithm have been used and applied in image de-noising and image reconstruction [13]. The K-SVD algorithm constructs the best sparse representation of the dictionary obtained from training images. This property makes K-SVD a good dictionary learning algorithm for face recognition [14]. The Label Consistent K-SVD (LC-KSVD) algorithm assigns a label to each atom using the K-SVD algorithm and then minimizes the discriminative sparse coding error by exploiting the labels of the atoms. Thus, it can improve the discriminative ability of the dictionary.

The objective function for learning a dictionary is

$$\arg\min_{D,W,A,X} \|Y - DX\|_2^2 + \alpha\|Q - AX\|_2^2 + \beta\|H - WX\|_2^2 \tag{2}$$

$$s.t. \forall i, \quad \|x_i\|_0 \leq T_0$$

where $Y = [y_1, y_2, \dots, y_N] \in R^{n\times N}$ are the training images, and n and N are the dimension and number of images, respectively. $D = [d_1, \dots, d_K] \in R^{n\times K}$ is the dictionary where K is the number of atoms. α and β are the regularization parameters, T_0 is the sparsity constraint that limits the number of non-zero elements, $X = [x_1, \dots, x_N] \in R^{K\times N}$ is the coding coefficient matrix, W is the classifier parameter, and $\|H - WX\|_2^2$ is the classification error. $H = [h_1, \dots, h_N]$ is the label matrix of Y, and Q is the discriminative sparse code of Y which can be defined as $Q = [q_1, \dots, q_N] \in$

$R^{K\times N}$. A is the linear transformation matrix and $\|Q - AX\|_2^2$ is the discriminative sparse code error [13-15].

2.1.1 Optimization

The algorithm used to find the optimal solution for LC-KSVD [13, 15] is

$$\arg\min_{D,W,A,X} \left\| \begin{bmatrix} Y \\ \sqrt{\alpha}Q \\ \sqrt{\beta}H \end{bmatrix} - \begin{bmatrix} D \\ \sqrt{\alpha}A \\ \sqrt{\beta}W \end{bmatrix} X \right\|_2^2 \tag{3}$$

$$s.t. \forall i, \|x_i\|_0 \leq T_0$$

LC-KSVD learns D, A, and W simultaneously. This is scalable to a large number of classes. In addition, it combines the discriminative sparse code error into the objective function, and produces a discriminative sparse representation regardless of the size of the dictionary.

2.1.2 Classification

After obtaining $D = \{d_1, d_2, \dots, d_k\}$, $A = \{a_1, a_2, \dots, a_K\}$ and $W = \{\omega_1, \omega_2, \dots, \omega_K\}$, the desired dictionary $\widehat{D}$, transform parameters $\hat{A}$, and classifier parameters $\widehat{W}$ are computed [10, 13, 16] as

$$\widehat{D} = \left\{\frac{d_1}{\|d_1\|_2}, \frac{d_2}{\|d_2\|_2}, \dots, \frac{d_K}{\|d_K\|_2}\right\}$$

$$\hat{A} = \left\{\frac{a_1}{\|d_1\|_2}, \frac{a_2}{\|d_2\|_2}, \dots, \frac{a_K}{\|d_K\|_2}\right\}$$

$$\widehat{W} = \left\{\frac{\omega_1}{\|d_1\|_2}, \frac{\omega_2}{\|d_2\|_2}, \dots, \frac{\omega_K}{\|d_K\|_2}\right\} \tag{4}$$

For a training image y_i, the sparse representation x_i is first computed by

$$x_i = \arg\min_{x_i} \left\|y_i - \widehat{D}x_i\right\|_2^2 \quad s.t. \|x_i\|_0 \leq T_0 \tag{5}$$

Then the label j of y_j is obtained as

$$j = \arg\max \ (\widehat{W}x_i) \tag{6}$$

W can be calculated using the coding coefficient matrix X and label matrix H of the training images, where I is the identity matrix, as

$$W = HX^T(XX^T + I)^{-1} \tag{7}$$

2.2 Fisher Discriminative Dictionary Learning (FDDL)

Fisher Discrimination Dictionary Learning (FDDL) produces a dictionary $D = [D_1,\ D_2, \dots, D_c]$, where D_i is the sub-dictionary related to class i and c is the number of classes. The classification criteria is the residual associated with each class. These residuals are obtained by representing the training images in the dictionary [4, 7]. The representation coefficients are also made discriminative under the Fisher criterion which further enhances the discrimination ability of the dictionary [17].

If the training images are $Y = [Y_1,\ Y_2, \dots, Y_c]$ and X is the sparse representation matrix of Y over D, then X can be written as $X = [X_1,\ X_2, \dots, X_c]$ where X_i is the representation matrix of Y_i over D. The FDDL objective function [4, 18, 19] is

$$J_{(D,X)} = \text{argmin}_{(D,X)}\{r(Y,D,X) + \lambda_1\|X\|_1 + \lambda_2 f(X)\} \qquad s.t. \|d_n\|_2 = 1,\ \ \forall n \tag{8}$$

where $r(Y,D,X)$ is the discriminative fidelity, $\|X\|_1$ is the sparsity penalty, $f(X)$ is a discrimination term imposed on the coefficient matrix X, and λ_1 and λ_2 are scalar parameters.

Discriminative Fidelity Term $\boldsymbol{r(Y,D,X)}$

X_i can be written as $X_i = [X_i^1 + \cdots + X_i^j + \cdots + X_i^c]$, where X_i^j is the representation of Y_i over D_j. First, the dictionary D should represent Y_i well, so $Y_j \approx DX_i = D_1X_i^1 + \cdots + D_iX_i^i + \cdots + D_cX_i^c = R_1 + \cdots + R_i + \cdots + R_c$, where $R_i = D_iX_i^i$. Second, since D_i is related to the i-th class, Y_i can be represented better by D_i than by D_j, $j \neq i$, which implies that X_i^i has large coefficients that make $\left\|Y_i - D_iX_i^i\right\|_F^2$ relatively small. Further, X_i^j should have small coefficients making $\left\|D_iX_i^j\right\|_F^2$ small. Therefore, the discriminative fidelity term [4, 19] is

$$r(Y_i, D, X_i) = \|Y_i - DX_i\|_F^2 + \left\|Y_i - D_i X_i^i\right\|_F^2 + \sum_{j=1}^{c}\left\|D_j X_i^j\right\|_F^2 \qquad j \neq i \tag{9}$$

Discriminative Coefficient Term $f(X)$

To further increase the discrimination capability of dictionary D, we can enforce the representation matrix of Y over D, i.e. X, to be discriminative. Considering the Fisher discrimination criterion, this can be achieved by minimizing $S_W(X)$ and maximizing $S_B(X)$ [4] which are the within-class and between-class scatter of X, respectively, formulated as

$$S_W(X) = \sum_{i=1}^{c}\sum_{x_k \in X_i}(x_k - m_i)(x_k - m_i)^T \tag{10}$$

$$S_B(X) = \sum_{i=1}^{c} n_i(m_i - m)\,(m_i - m)^T \tag{11}$$

where m_i and m are the mean vectors of X_i and X, respectively, and n_i is the number of samples in Y_i. The discriminative coefficient term is

$$f(X) = tr(S_W(X)) - tr\big(S_B(X)\big) + \eta\|X\|_F^2 \tag{12}$$

where $tr(\bullet)$ denotes the trace of a matrix, η is a regularization parameter, and the term $\eta\|X\|_F^2$ makes $f(X)$ smoother and convex [20]. Incorporating (9) and (12) into (8), the FDDL is

$$\min_{(D,X)}\left\{\sum_{i=1}^{c}\left(\|Y_i - DX_i\|_F^2 + \left\|Y_i - D_i X_i^i\right\|_F^2 + \sum_{j=1}^{c}\left\|D_j X_i^j\right\|_F^2\right) + \lambda_1\|X\|_1 + \lambda_2(tr\big(S_W(X) - S_B(X)\big) + \eta\|X\|_F^2\right\} \qquad s.t.\ \|d_n\|_2 = 1, \forall n;\ \left\|D_j X_i^j\right\|_F^2 \leq \varepsilon_f\ , \forall i \neq j \tag{13}$$

where ε_f is a small positive scalar. Because $\left\|D_j X_i^j\right\|_F^2$ is very small for $j \neq i$, FDDL can be simplified by assuming $X_i^j = 0$ so then $\left\|D_j X_i^j\right\|_F^2 = 0$. Thus, the simplified FDDL [19, 20] can be written as

$$\min_{(D,X)}\left\{\sum_{i=1}^{c}\left(\|Y_i - DX_i\|_F^2 + \left\|Y_i - D_i X_i^i\right\|_F^2\right) + \lambda_1\|X\|_1 + \lambda_2(tr\big(S_W(X) - S_B(X)\big) + \eta\|X\|_F^2\right\} \qquad s.t.\ \|d_n\|_2 = 1, \forall n;\ X_i^j = 0\ , \forall i \neq j \tag{14}$$

2.2.1 Optimization

Optimizing the FDDL objective function can be divided into the sub-problems of optimizing D and X alternatively, i.e. updating X with D fixed, and updating D with X fixed. This is iteratively implemented to find the desired dictionary D and coefficient matrix X [4, 19].

Update of *X*

If the dictionary D is fixed, then the FDDL objective function can be reduced to a sparse representation problem to obtain $X = [X_1, X_2, \dots, X_K]$. The objective function [4] is then

$$\min_{X_i}\{r(Y_i, D, X_i) + \lambda_1\|X_i\|_1 + \lambda_2 f_i(X_i)\} \tag{15}$$

with

$$f_i(X_i) = \|X_i - M_i\|_F^2 - \sum_{k=1}^{c}\|M_k - M\|_F^2 + \eta\|X_i\|_F^2 \tag{16}$$

where M_k and M are the mean vector matrices (by taking the mean vectors m_k and m as the column vectors) of class k and all classes, respectively. In order to make $f_i(X_i)$ not only convex but also have enough discrimination, η is set to 1. Then all terms in (15) except $\|X_i\|_1$ are differentiable, and the objective function is strictly convex.

Update of *D*

To update $D = [D_1, D_2, \dots, D_c]$ when $X = [X_1, X_2, \dots, X_c]$ is fixed, the D_i are updated separately [19]. For the update of D_i, D_j, $j \neq i$, are fixed, so the objective function is simplified to

$$\min_{D_i}\left\{\left\|Y - D_i X^i - \sum_{j=1, j\neq i}^{c} D_j X^j\right\|_F^2 + \left\|Y_i - D_i X_i^i\right\|_F^2 + \sum_{j=1, j\neq i}^{c}\left\|D_i X_j^i\right\|_F^2\right\} \tag{17}$$

where X^i is the coding coefficients of Y over D_i.

2.2.2 Classification

Once the dictionary D is learned, a training image can be classified via coding it over D. A training image y is sparsely represented by sub-dictionary D_i as

$$\hat{\alpha} = \min \|y - D_i x_i\|_2^2 \tag{18}$$

and then y is classified using

$$j = \min \|y - D_i x_i\|_2^2 \tag{19}$$

where j is the label for y. According to the number of training images, two classification schemes are described as follows [4].

Global Classifier (GC): When the number of training images in a class is small, the dictionary D_i cannot represent the training images of the class, and hence y is coded over D. In this case, the sparse coding coefficients are obtained as

$$\hat{\alpha} = \arg\min_{\alpha}\{\|y - D\alpha\|_2^2 + \Upsilon\|\alpha\|_1\} \tag{20}$$

where Υ is a constant. Let $\hat{\alpha} = [\hat{\alpha}_1, \hat{\alpha}_2, \dots, \hat{\alpha}_K]$, where $\widehat{\alpha_\iota}$ is the coefficient vector associated with sub-dictionary D_i. The classification is

$$e_i = \|y - D_i\widehat{\alpha_\iota}\|_2^2 + w\|\hat{\alpha} - m_i\|_2^2 \tag{21}$$

where the first term is the reconstruction error for class i, the second term is the distance between the coefficient vector $\hat{\alpha}$ and the mean vector m_i of class i, and w is a weight to balance the contribution of the two terms.

Local Classifier (LC): When the number of training images in a class is large, y is coded directly by D_i instead of the whole dictionary D to reduce the computational cost. If $m_i = [m_i^1, \dots, m_i^k, \dots, m_i^c]$, where m_i^k is the sub-vector associated with sub-dictionary D_k, the coding coefficients associated with D_i are

$$\hat{\alpha} = \arg\min_{\alpha}\{\|y - D_i\alpha\|_2^2 + \Upsilon_1\|\alpha\|_1 + \Upsilon_2\left\|\alpha - m_i^i\right\|_2^2\} \tag{22}$$

where Υ_1 and Υ_2 are constants. y is coded by D_i with sparse coefficients and the coding vector α is close to m_i^i. The classification is

$$e_i = \|y - D_i\hat{\alpha}\|_2^2 + \Upsilon_1\|\hat{\alpha}\|_1 + \Upsilon_2\left\|\hat{\alpha} - m_i^i\right\|_2^2 \tag{23}$$

2.3 Support Vector Guided Dictionary Learning (SVGDL)

In DDL, the discrimination of the dictionary is enforced by either imposing structural constraints on the dictionary or by imposing a discrimination term on the coding vectors. Support Vector Guided Dictionary Learning (SVGDL) is a new approach in class-specific dictionary learning algorithms in which the discrimination term is formulated as the weighted sum of the squared distances between all pairs of coding vectors [21]. Unlike other sparse coding techniques that employ the similarity between sample pairs to calculate the corresponding weights [22], SVGDL incorporates the sample label information into determining the weights. Therefore, the FDDL method can be viewed as a special case of SVGDL. The difference is that in the SVGDL approach, the weights are determined by the number of images in each class [23].

SVGDL makes the task of weight assignment more adaptive and flexible. It incorporates a parameterizing method with symmetry that simplifies the problem of weight assignment optimization to the dual form of a linear Support Vector Machine (SVM). This allows SVGDL to use a multi-class linear SVM for efficient DDL. In the weight assignment, most vectors will be zero except for the weights of pairs of support vectors in learning a discriminative dictionary. This property makes SVGDL superior to FDDL in terms of classification performance [24].

Assuming that the weight ω_{ij} can be parameterized as a function of variable β instead of directly assigning weight ω_{ij} for each pair [4], SVGDL defines the parameterized formulation of the discrimination term as

$$f\left(Z, \omega_{ij}(\beta)\right) = \sum_{i,j} \left\| z_i - z_j \right\|_2^2 \omega_{ij}(\beta) \tag{24}$$

where z_i and z_j are the coding vectors of samples i and j, $Z = [z_1, z_2, \dots, z_n]$ are the coding vectors of Y over D, and $Y = [y_1, y_2, \dots, y_N]$ and $D = [d_1, d_2, \dots, d_K]$ are the training images and the dictionary, and N and n are the number of images and dimension, respectively.

Parameterization should have the following constraints in order to function properly:

a) symmetry: $\omega_{ij}(\beta) = \omega_{ji}(\beta)$;
b) consistency: $\omega_{ij}(\beta) \geq 0$ if $y_i = y_j$, and $\omega_{ij}(\beta) \leq 0$ if $y_i \neq y_j$;
c) balance: $\sum_{j=1}^{n} \omega_{ij}(\beta) = 0$, $\forall i$.

Consistency means that the weight ω_{ij} should be non-negative when z_i and z_j are from the same class. In addition, ω_{ij} should be non-positive when z_i and z_j are from different classes. Balance is introduced to balance the contributions of positive and negative weights [21, 23].

A special instance of the parameterization for $\omega_{ij}(\beta)$ is introduced, $\omega_{ij}(\beta) = y_i y_j \beta_i \beta_j$ and $\sum_{j=1}^{n} y_j \beta_j = 0$ are defined where $\beta = [\beta_1, \beta_2, \dots, \beta_n]$ is a nonnegative vector. The discrimination term $f(Z, \omega_{ij}(\beta))$ is

$$f\left(Z, \omega_{ij}(\beta)\right) = -2 \sum_{i,j} y_i y_j \beta_i \beta_j z_i^T z_j = \beta^T K \beta \tag{25}$$

where K is a negative semidefinite matrix.

The objective function of $f(Z, \omega_{ij}(\beta))$ is maximized as

$$\begin{aligned} &\text{argmax}\ \beta^T K \beta + r(\beta) \\ &s.t. \quad \beta_i >= 0, \forall i, \sum_{j=1}^{n} y_j \beta_j = 0 \end{aligned} \tag{26}$$

where $r(\beta)$ is a regularization term to avoid the trivial solution $\beta = 0$ [4]. The parameterized DDL formulation is then

$$\arg\min_{D,Z} \left(\|Y - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + \lambda_2 \max_{\beta \in dom(\beta)} (\sum_{i,j} \left\|z_i - z_j\right\|_2^2 \omega_{ij}(\beta) + r(\beta)) \right) \tag{27}$$

where the domain of variable β is $\text{dom}(\beta)$: $\beta \geq 0$, $\sum_{j=1}^{n} y_j \beta_j = 0$. The weight assignment in coding space falls into the appropriate selection of $\text{dom}(\beta)$, $\omega_{ij}(\beta)$ and $r(\beta)$. Considering $r(\beta) = 4 \sum_{i=1}^{n} \beta_i$ and the appropriate selection of $\text{dom}(\beta)$ and $\omega_{ij}(\beta)$, (27) can be simplified as

$$\begin{aligned} &\arg\min_{D,Z} \left(\|Y - DZ\|_F^2 + \lambda_1 \|Z\|_p^p + \lambda_2 \max_{\beta} (4 \sum_{i=1}^{n} \beta_i - 2 \sum_{i,j} y_i y_j \beta_i \beta_j z_i^T z_j) \right) \\ &s.t. \quad \beta_i \geq 0, \forall i \text{ and } \sum_{j=1}^{n} y_j \beta_j = 0 \end{aligned} \tag{28}$$

In order to simplify the solution, it is assumed that $\beta_i \leq \frac{1}{2}\theta$ for all i, where θ is a fixed constant. An SVM performs classification by finding the hyperplane which maximizes the margin between the two classes [24]. The vectors that define the hyperplane are the support vectors. The SVGDL formulation is then

$$\arg\min_{D,Z,u,b}\left(\|Y-DZ\|_F^2+\lambda_1\|Z\|_p^p+2\lambda_2 f(Z,y,u,b)\right) \tag{29}$$

where u is the normal to the hyperplane of SVM, b is the corresponding bias, $y=[y_1,y_2,\dots,y_n]$ is the label vector, and

$$f(Z,y,u,b)=\|u\|_2^2+\theta\sum_{i=1}^{n} l(z_i,y_i,u,b) \tag{30}$$

where $l(z_i,y_i,u,b)$ is the loss function used for training the classifiers.

Representing the solution as a linear combination of coding vectors combined with the sparsity of β, the general DDL formulation can be written as

$$\arg\min_{D,Z}\left(\|Y-DZ\|_F^2+\lambda_1\|Z\|_p^p+\lambda_2\sum_{i,j\in SV}\left\|z_i-z_j\right\|_2^2\omega_{ij}(\beta)\right) \tag{31}$$

where SV is the set of support vectors.

It should be noted that SVGDL has two characteristics that support coding vectors. These characteristics are the most important factors in DDL and are as follows.

1. SVGDL adopts an adaptive weight assignment (unlike FDDL which incorporates a deterministic method).
2. Only pairwise support coding vectors are assigned non-zero weights (instead of all pairwise coding vectors).

In machine learning, multi-class is the problem of classifying samples into one of three or more classes. A one-vs-all strategy is used for multi-class classification that trains a single classifier for each class, with the samples of that class as positive and all other samples as negative [24, 25]. This is done by merging C hyperplanes $U=[u_1,u_2,\dots,u_C]$ and corresponding biases $b=[b_1,b_2,\dots,b_C]$, which reformulates SVGDL as

$$\arg\min_{D,Z,U,b}\left(\|Y-DZ\|_F^2+\lambda_1\|Z\|_p^p+2\lambda_2\sum_{c=1}^{C} f(Z,y^c,u_C,b_C)\right) \tag{32}$$

where $y^c=[y_1^c,y_2^c,\dots,y_n^c]$, $y_i^c=1$ if $y_i=c$, and otherwise $y_i^c=-1$.

2.3.1 Optimization

The general multi-class SVGDL in (32) is not jointly convex for D, Z, U, and b, but is convex with respect to each variable. Therefore, an updating scheme is presented as follows [25].

With D and Z fixed, minimization of U and b becomes a multi-class linear SVM problem which can be further simplified as C linear one-vs-all SVM sub-problems [21, 24]

$$l(z_i, y_i^c, u_c, b_c) = [\min(0, y_i^c[u_c;\ b_c]^T[z_i; 1] - 1)]^2 \tag{33}$$

With D, U and b fixed, the columns z_i of the coefficient matrix Z are optimized as

$$\arg\min_{z_i}(\|y_i - Dz_i\|_2^2 + \lambda_1\|z_i\|_2^2 + 2\lambda_2.\theta\sum_{c=1}^{C} f(z_i, y_i^c, u_c, b_c)) \tag{34}$$

With Z, U and b fixed, the optimization problem with respect to D is

$$\arg\min_{D}\|Y - DZ\|_F^2 \quad s.t.\ \|d_k\|^2 \leq 1, \forall k \in \{1, 2, \dots, K\} \tag{35}$$

2.3.2 Classification

After D and classifier U based on b are obtained, classification is performed by projecting x with a fixed matrix P [4, 21] so that $z = Px$, where $P = (D^T D + \lambda_1 I)^{-1} D^T$. Then the label of the sample is predicted by applying the C linear classifiers on the coding vector z, where $c \in [1, 2, \dots, C]$ which gives

$$y = \arg\max_{c\,\in\,1,2,\dots,C} u_c^T z + b_c \tag{36}$$

Chapter 3

Results and Discussion

In this chapter, face recognition results for the Label-Consistent K-SVD (LC-KSVD), Fisher Discriminative Dictionary Learning (FDDL), and Support Vector Guided Dictionary Learning (SVGDL) are presented. These algorithms were implemented using MATLAB.

3.1 Image Database

The extended Yale B image database was used for training and testing the face recognition algorithms. This database contains more than 2000 front face images of 38 people which were taken with various illumination conditions and expressions. Each person has 64 images (32×32 pixels), and 20 images for each person were randomly selected as the test set for this project.

3.2 Measures

Three measures were considered to test the algorithms. The first is accuracy which is the percentage of training images correctly assigned. The second is speed which is the time for the algorithm to converge and is defined as the MATLAB run-time of the algorithm. The third is variability which measures the dependency of the accuracy of each algorithm on a specific set of training images. A new set of training images is used for multiple experiments and the corresponding accuracy error is the variability.

3.3 Input Parameters

In order to evaluate the relationship between the output measures and the initial parameters of each algorithm, each experiment used a combination of three input parameters, the number of training images which affects the accuracy of the results, the number of atoms in the dictionary which affects the accuracy and speed, and the number of iterations which also affects the accuracy and speed. Since the purpose of evaluating different measures is a fair comparison of the algorithms and FDDL did not converge in some cases, the corresponding curves were ignored for SVGDL and LC-KSVD.

3.4 Accuracy of the face recognition algorithms

In this section the accuracy of the LC-KSVD, FDDL and SVGDL algorithms is evaluated. As there are three different input parameters (number of training images, atoms, and iterations), the results obtained for each individual parameter are presented with the other two fixed.

3.4.1 Effect of the number of training images

In this section, the accuracy of the three algorithms versus the number of training images is compared. Figures 1 to 3 present the accuracy versus the number of training images for the three algorithms with 150 atoms and 4, 6, and 10 iterations, respectively. Figures 4 to 6 present the results for 300 atoms and 4, 6, and 10 iterations, respectively. These results indicate that SVGDL has a higher face recognition accuracy, increasing from 83% with 300 training images to 95% with 900 training images. An increase in the number of training images results in better accuracy as expected. In the case of FDDL, the accuracy decreased with an increase in the number of training images. When the number of atoms is 300 and the number of training images less than 600, no results were obtained. Increasing the number of atoms from 150 to 300 did not change the accuracy of LC-KSVD with 4 to 10 iterations. In summary, the results indicate that SVGDL is more accurate than the other algorithms.

3.4.2 Effect of the number of atoms

In this section, the accuracy of the three algorithms versus the number of atoms is compared. Figures 7 to 9 present the accuracy versus the number of atoms for the three algorithms with 650 training images and 4, 6, and 10 iterations, respectively. Figures 10 to 12 present the results for 950 training images and 4, 6, and 10 iterations, respectively. The results indicate that SVGDL has an accuracy greater than 90% in all cases whereas the other two algorithms have accuracy less than 90%. When the number of training images is 650 and the number of atoms is 150, FDDL performs similar to LC-KSVD with 83% accuracy. With 300 atoms, the accuracy of FDDL is similar to SVGDL at up to 90%. Thus, the number of atoms affects the performance of FDDL, while the number of iterations does not. With 600 atoms and 950 training images, LC-KSVD accuracy is similar to that of SVGDL at up to 90% as shown in Figures 10 to 12.

3.4.3 Effect of the number of iterations

In this section, the accuracy of the three algorithms versus the number of iterations is compared. Figures 13 and 14 present the accuracy versus the number of iterations for the three algorithms with 150 atoms and 650 and 950 training images, respectively. Figures 15 and 16 present the results for 300 atoms and 650 and 950 training images, respectively. It is expected that increasing the number of iterations will improve the accuracy. However, it has a reverse effect in the case of FDDL when the number of atoms is 300. The accuracy decreases from 95% with 650 training images to 85% with 950 training images. The accuracy of SVGDL is between 90% and 95%, whereas the accuracy of the other two algorithms is less than 90%. Thus, SVGDL provides better performance than the other algorithms.

3.5 Speed of the face recognition algorithms

In this section the speed of the LC-KSVD, FDDL and SVGDL algorithms is evaluated. As there are three different input parameters (number of training images, atoms, and iterations), the results obtained for each individual parameter are presented with the other two fixed.

3.5.1 Effect of the number of training images

In this section, the speed of the three algorithms versus the number of training images is compared. Figures 17 to 19 present the speed versus the number of training images for the three algorithms with 150 atoms and 4, 6, and 10 iterations, respectively. Figures 20 to 22 present the results for 300 atoms and 4, 6, and 10 iterations, respectively. For 300 to 900 training images with 150 atoms, the speed of SVGDL and LC-KSVD is less than 100 seconds as shown in Figures 17 to 19 whereas FDDL requires more than 400 seconds. Moreover, the results in Figures 20 to 22 show that with 300 atoms, the speed of SVGDL and LC-KSVD is less than 200 seconds whereas with FDDL it is more than 1500 seconds. In addition, the number of training images does not affect the speed of SVGDL and LC-KSVD. In summary, the slowest algorithm is FDDL followed by SVGDL, and the fastest is LC-KSVD.

3.5.2 Effect of the number of atoms

In this section, the speed of the three algorithms versus the number of atoms is compared. Figures 23 to 25 present the speed versus the number of atoms for the three algorithms with 650 training images and 4, 6, and 10 iterations, respectively. Figures 26 to 28 present the results for 950 training images and 4, 6, and 10 iterations, respectively. With 150 to 300 atoms and 650 training images, the speed of SVGDL and LC-KSVD is less than 200 seconds as shown in Figures 23 to 25, whereas the speed of FDDL jumps from 400 seconds to 2000 seconds. Further, the results in Figures 26 to 28 show that FDDL has the highest dependency on the number of atoms used to construct the dictionary. In addition, the speed of the LC-KSVD algorithm is not dependent on the number of atoms. In summary, LC-KSVD has the fastest speed, followed by SVGDL and FDDL.

3.5.3 Effect of the number of iterations

In this section, the speed of the three algorithms versus the number of iterations is compared. Figures 29 and 30 present the speed versus the number of iterations for the three algorithms with 150 atoms and 650 and 950 training images, respectively. Figures 31 and 32 present the results for 300 atoms and 650 and 950 training images, respectively. With 2 to 10 iterations and 150 atoms, the speed of SVGDL and LC-KSVD is less than 100 seconds, whereas the speed of FDDL increases significantly from 400 seconds to 800 seconds as shown in Figures 29 and 30. Moreover, the results in Figures 31 and 32 show that the speed of LC-KSVD does not change when the number of atoms increases from 150 to 300. Meanwhile, the speed of SVGDL increases from 100 seconds to 200 seconds, whereas the speed of FDDL has a dramatic increase to 3500 seconds. In summary, the results indicate that the number of iterations affects the speed of the algorithms as expected. Further, Figures 29 to 32 indicate that the speed of LC-KSVD is the best while the speed of FDDL is the worst.

3.6 Variability of the face recognition algorithms

In this section the variability of the LC-KSVD, FDDL and SVGDL algorithms is evaluated. As there are three different input parameters (number of training images, atoms, and iterations), the results obtained for each individual parameter are presented with the other two fixed.

3.6.1 Effect of the number of training images

In this section, the variability of the three algorithms versus the number of training images is compared. Figures 33 to 35 present the variability versus the number of training images for the three algorithms with 150 atoms and 4, 6, and 10 iterations, respectively. Figures 36 to 38 present the results for 300 atoms and 4, 6, and 10 iterations, respectively. The results in Figures 33 to 38 indicate that with 150 atoms, the number of training images has an inverse relationship to the variability of the algorithm which is between 0.002 and 0.02. Increasing the number of training images from 600 to 900 with 300 atoms results in a higher variability between 0.002 and 0.04. In general, SVGDL is the algorithm most affected by increasing the number of training images, which results in the highest variability.

3.6.2 Effect of the number of atoms

In this section, the variability of the three algorithms versus the number of atoms is compared when the number of atoms is increased from 150 to 750. Figures 39 to 41 present the variability versus the number of atoms for the three algorithms with 650 training images and 4, 6, and 10 iterations, respectively. Figures 42 to 44 present the results for 950 training images and 4, 6, and 10 iterations, respectively. The results in Figures 39 to 44 indicate that increasing the number of atoms from 150 to 300 with 650 training images results in a higher variability between 0.0025 and 0.016. With 950 training images, the number of atoms has an inverse relationship to the variability of the algorithm which is between 0.0025 and 0.04. In these cases, the variability of SVGDL and LC-KSVD is only affected by the number of atoms with 950 training images.

3.6.3 Effect of the number of iterations

In this section, the variability of the three algorithms versus the number of iterations is compared. Figures 45 and 46 present the variability versus the number of iterations for the three algorithms with 150 atoms and 650 and 950 training images, respectively. Figures 47 and 48 present the results for 300 atoms and 650 and 950 training images, respectively. The results in Figures 45 to 48 indicate that increasing the number of iterations from 2 to 10 with 650 and 950 training images results in a higher variability which is between 0.003 and 0.08. In general, with 950 training images and 300 atoms, FDDL is the algorithm most affected by increasing the number of iterations, which results in the highest variability.

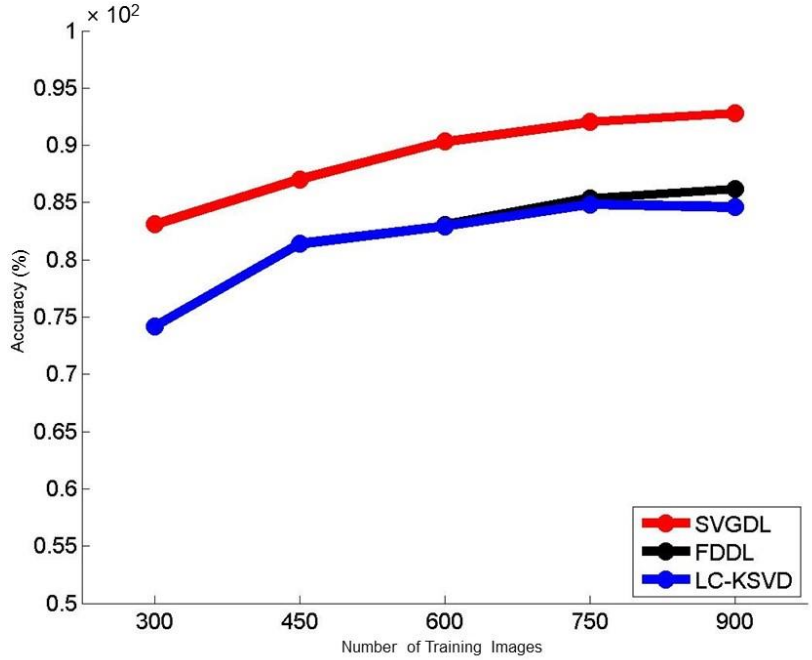

Figure 1. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 4, respectively.

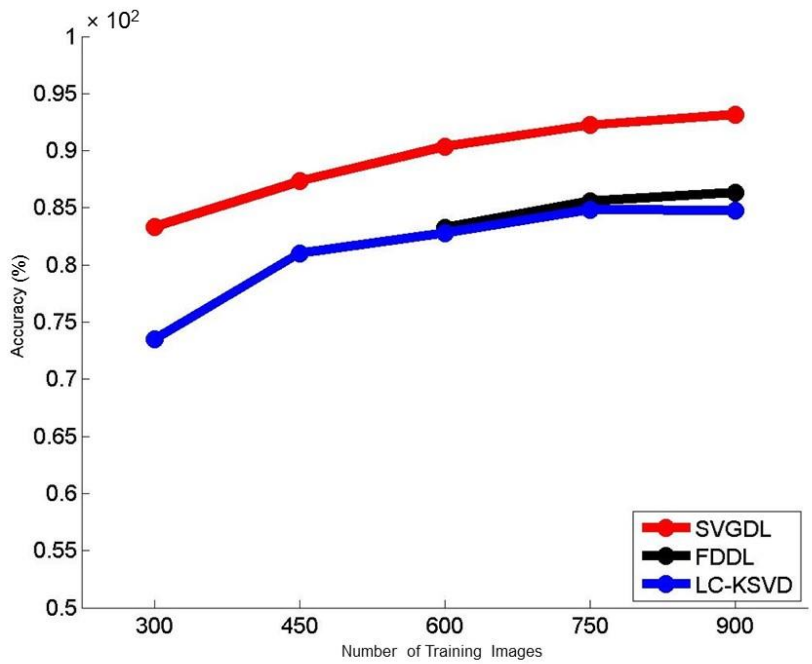

Figure 2. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 6, respectively.

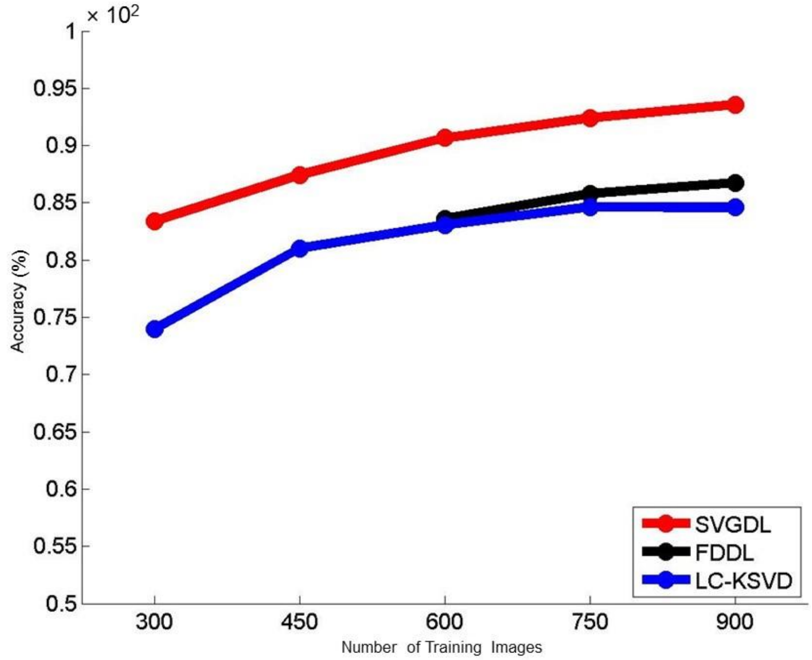

Figure 3. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 10, respectively.

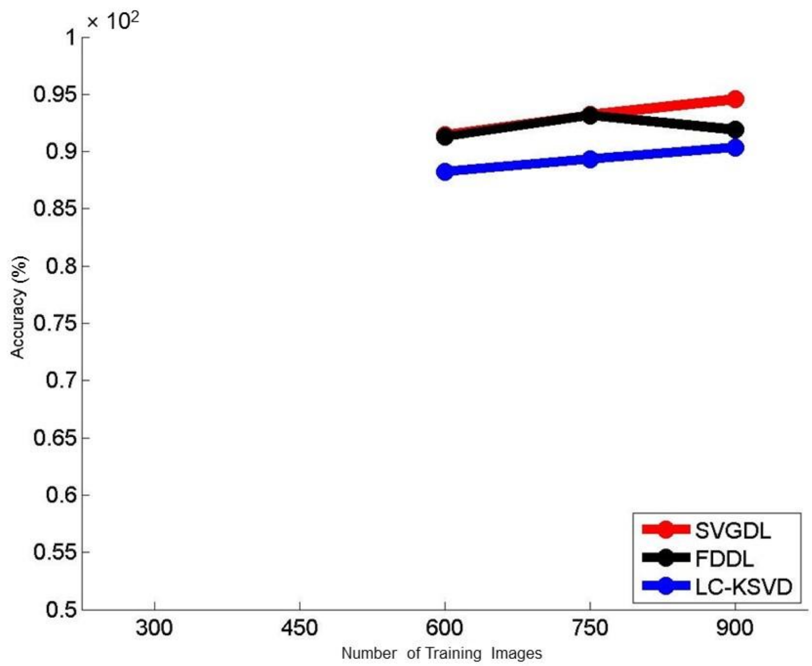

Figure 4. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 4, respectively.

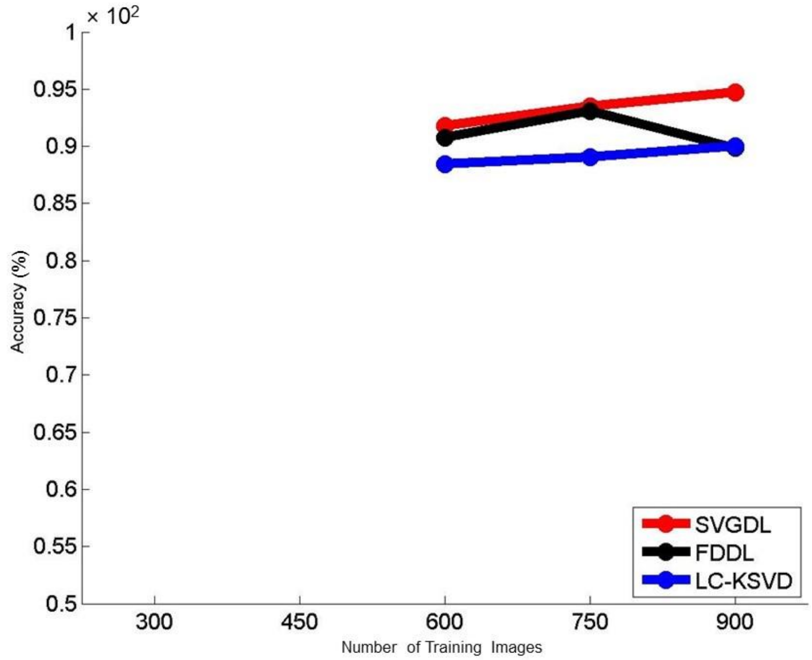

Figure 5. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 6, respectively.

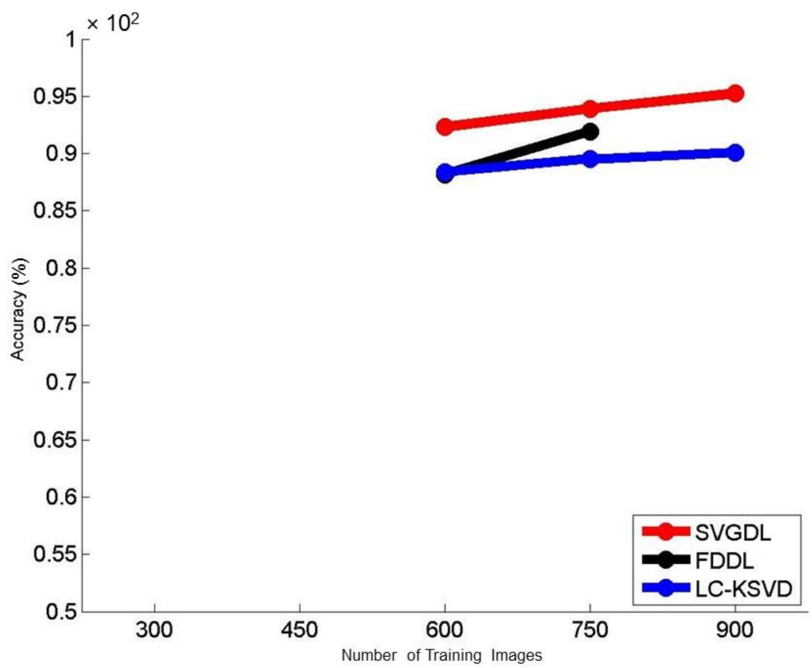

Figure 6. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 10, respectively.

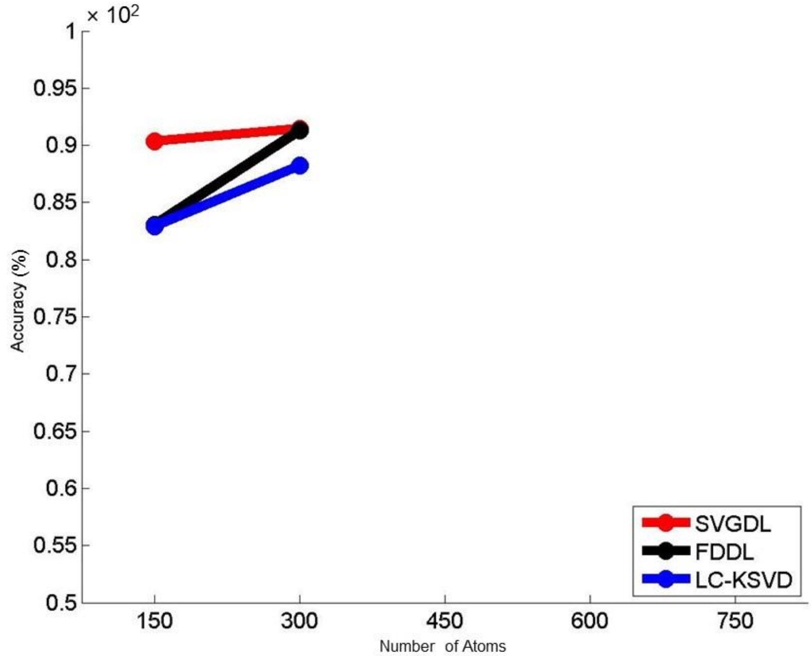

Figure 7. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 4, respectively.

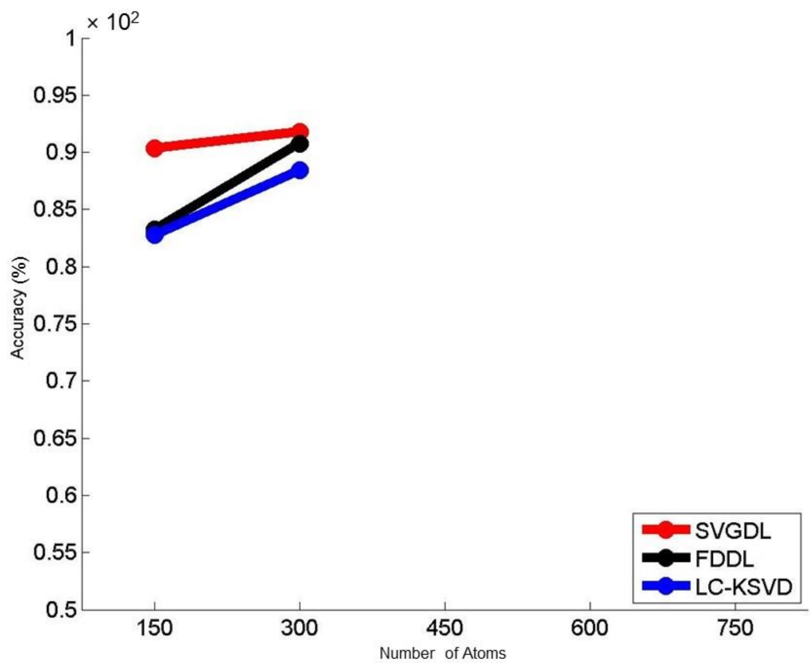

Figure 8. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 6, respectively.

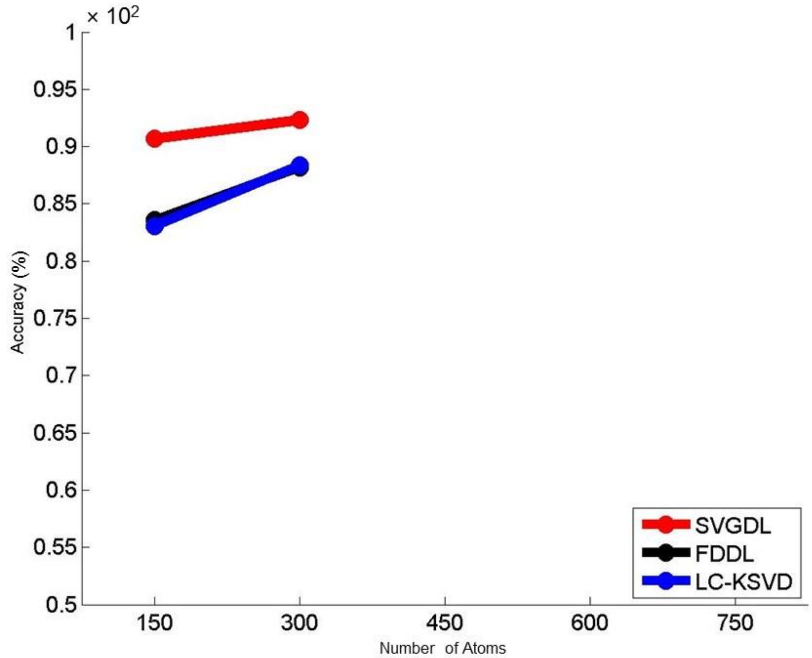

Figure 9. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 10, respectively.

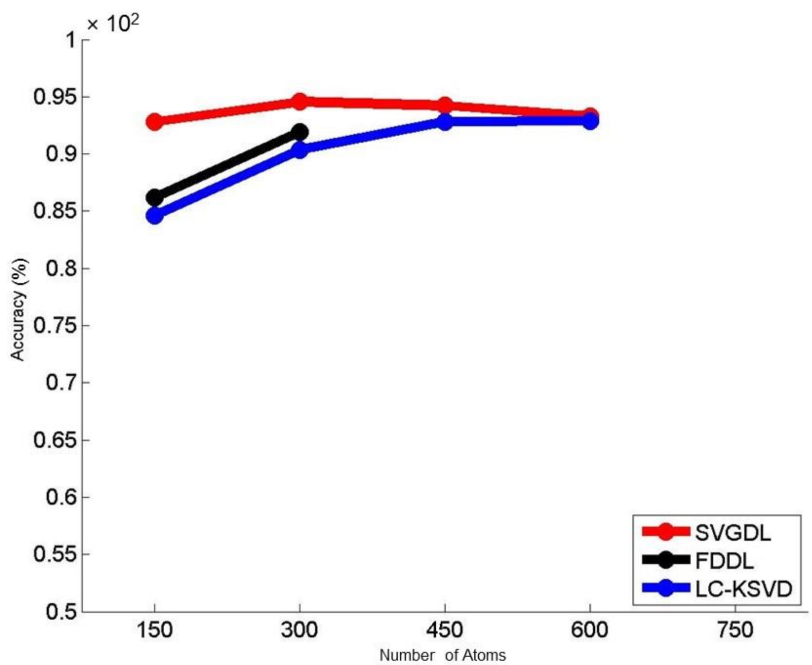

Figure 10. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 4, respectively.

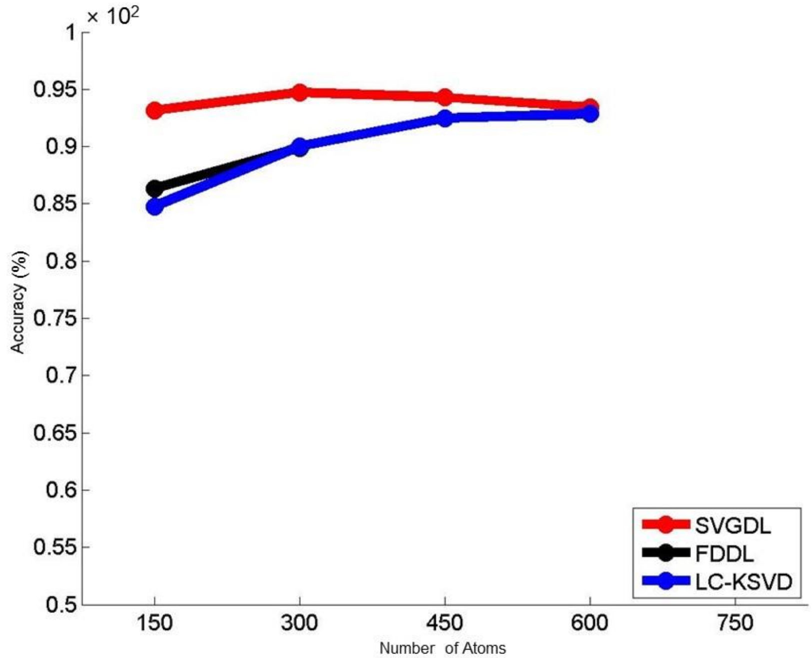

Figure 11. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 6, respectively.

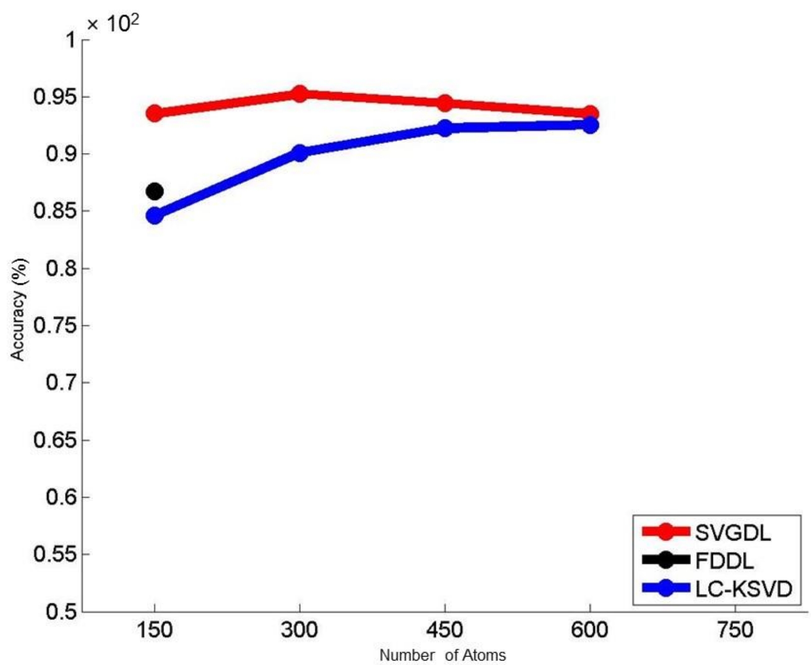

Figure 12. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 10, respectively.

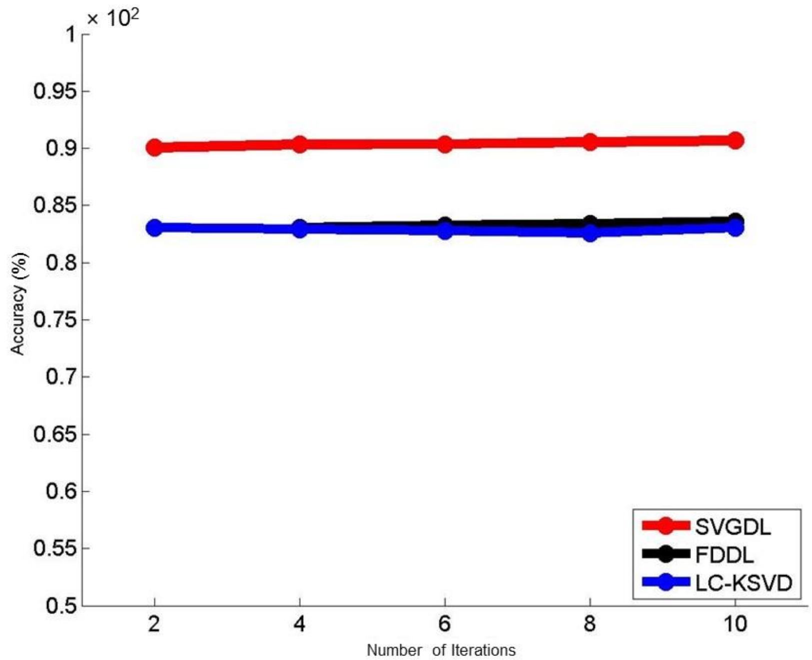

Figure 13. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 650 and 150, respectively.

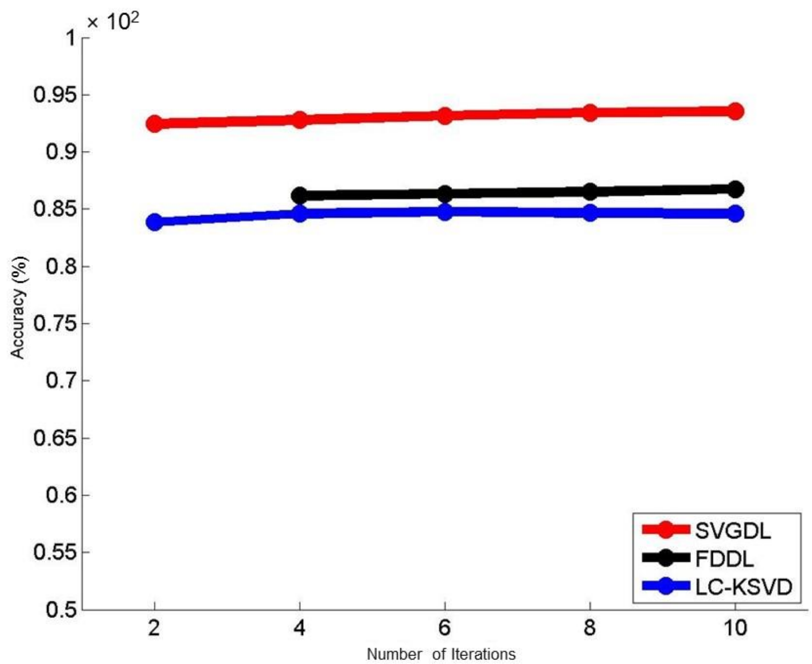

Figure 14. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 950 and 150, respectively.

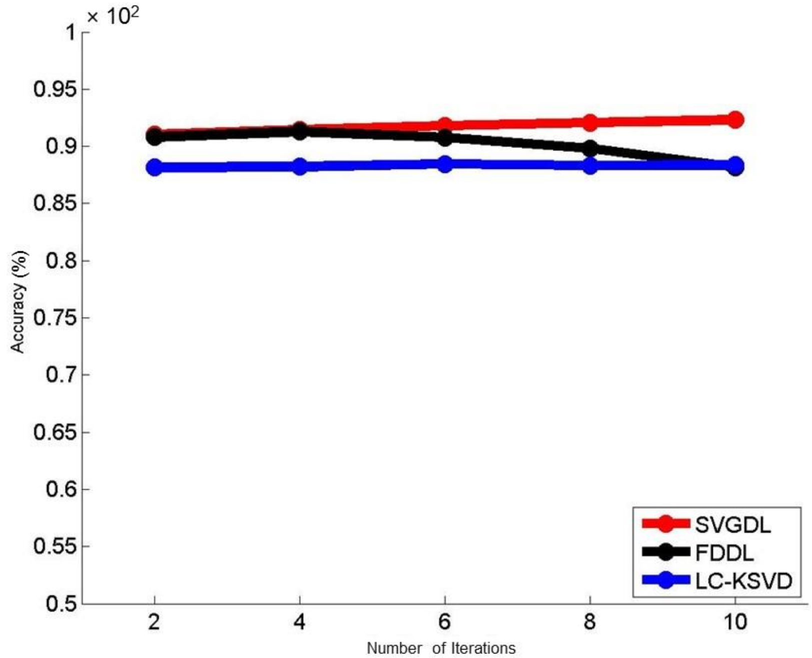

Figure 15. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 650 and 300, respectively.

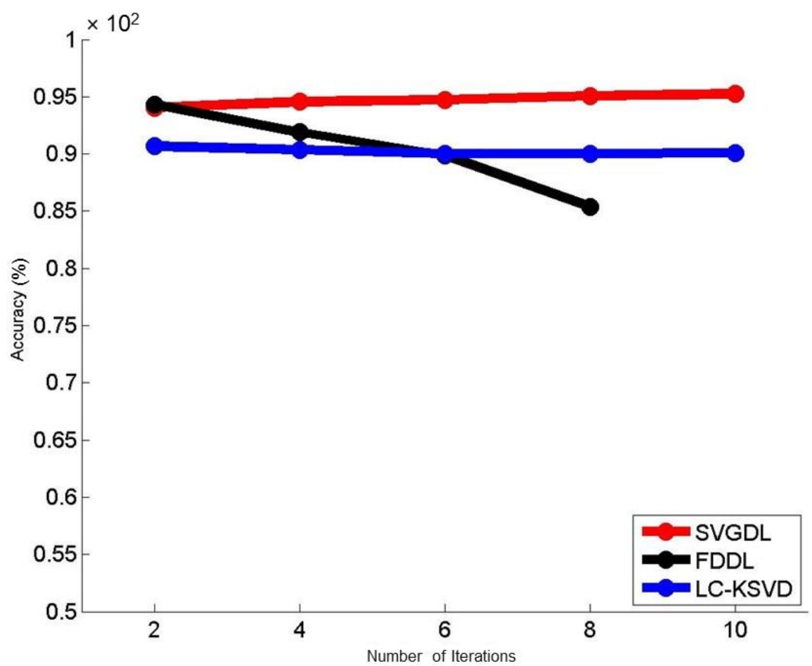

Figure 16. Face detection accuracy for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 950 and 300, respectively.

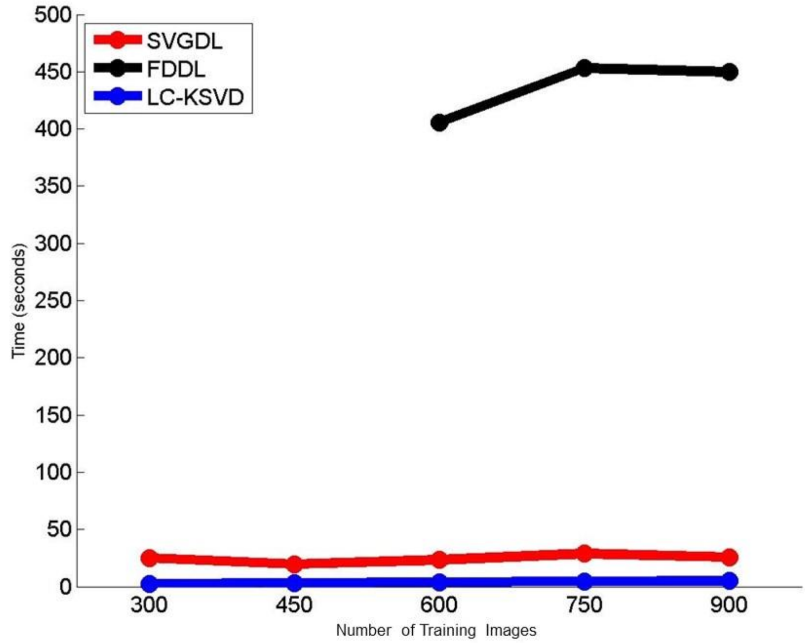

Figure 17. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 4, respectively.

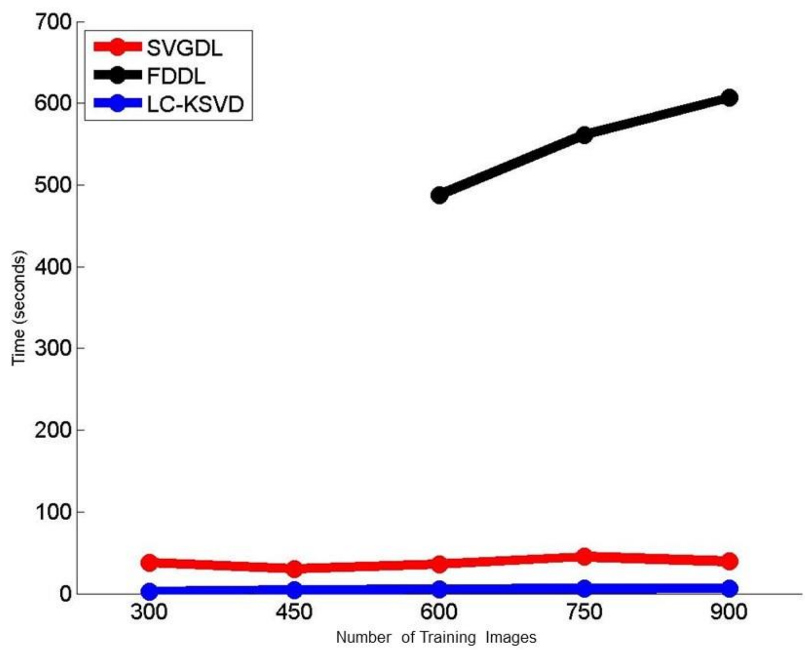

Figure 18. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 6, respectively.

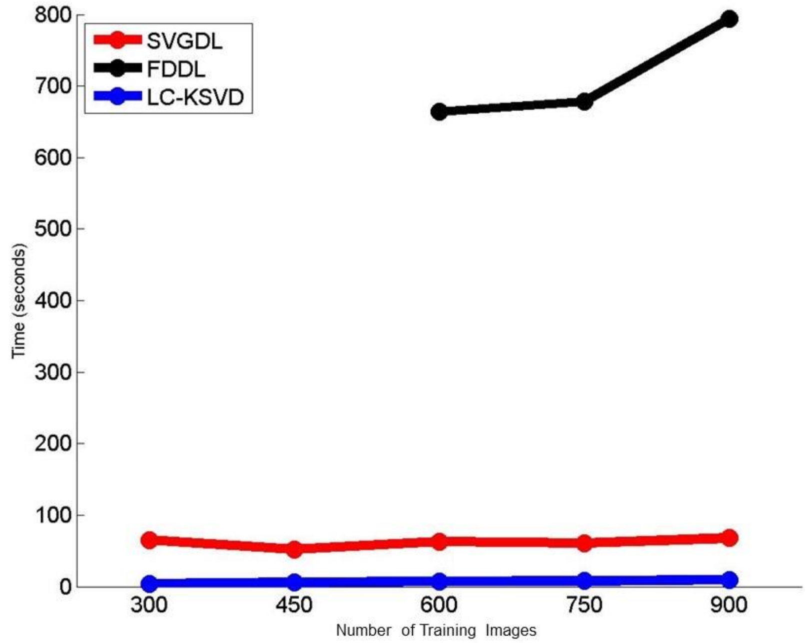

Figure 19. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 10, respectively.

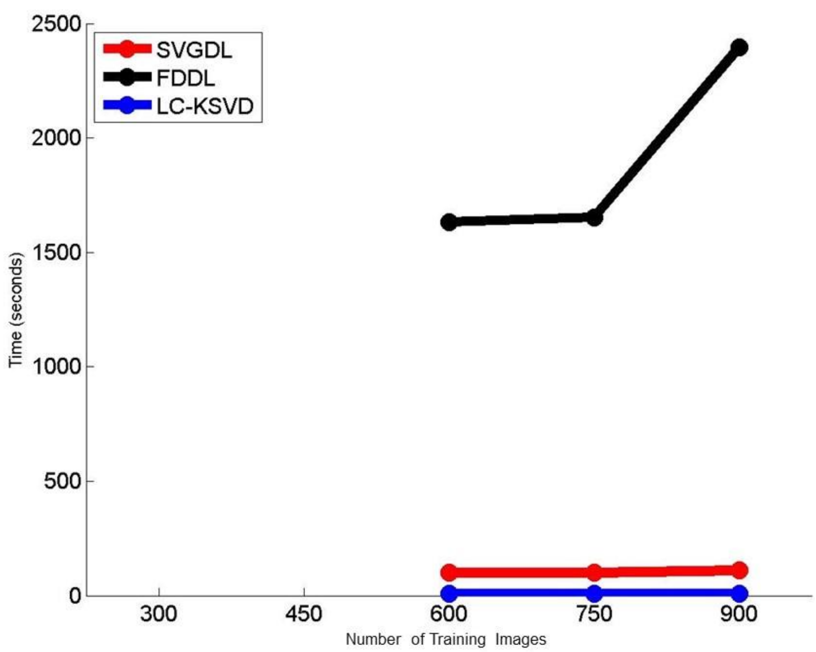

Figure 20. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 4, respectively.

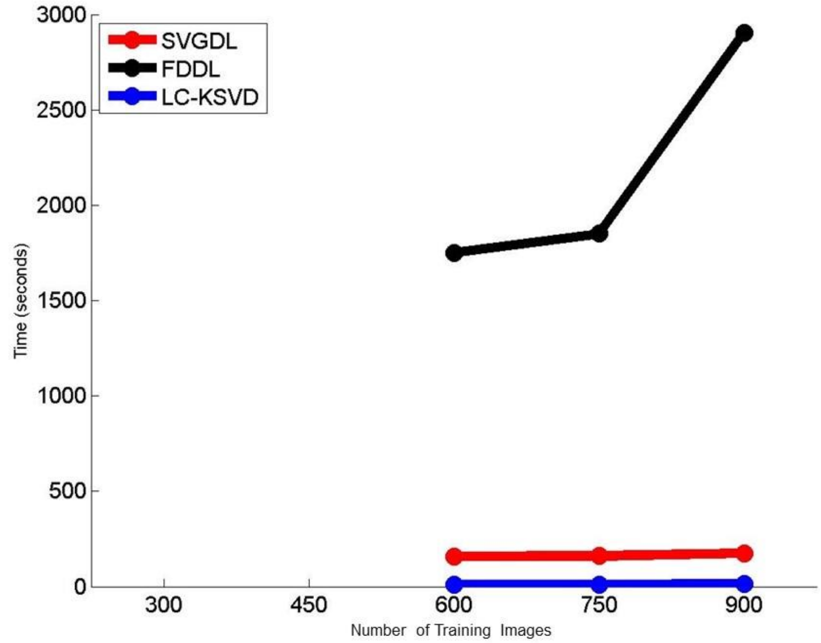

Figure 21. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 6, respectively.

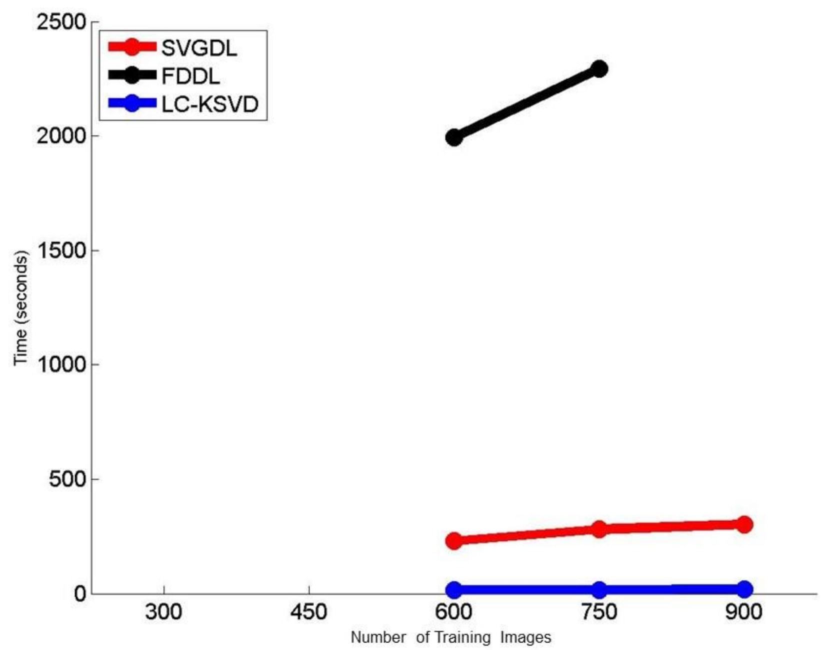

Figure 22. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 10, respectively.

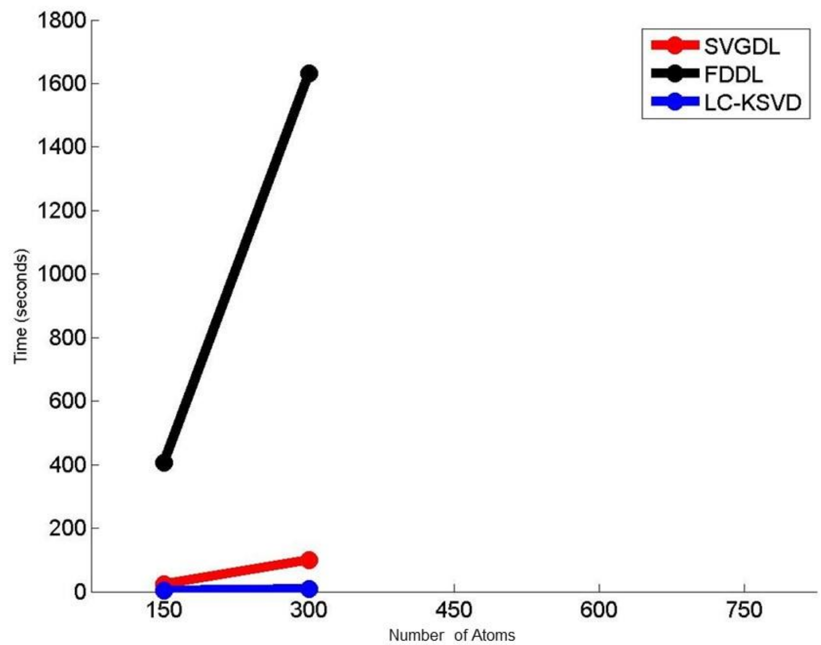

Figure 23. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 4, respectively.

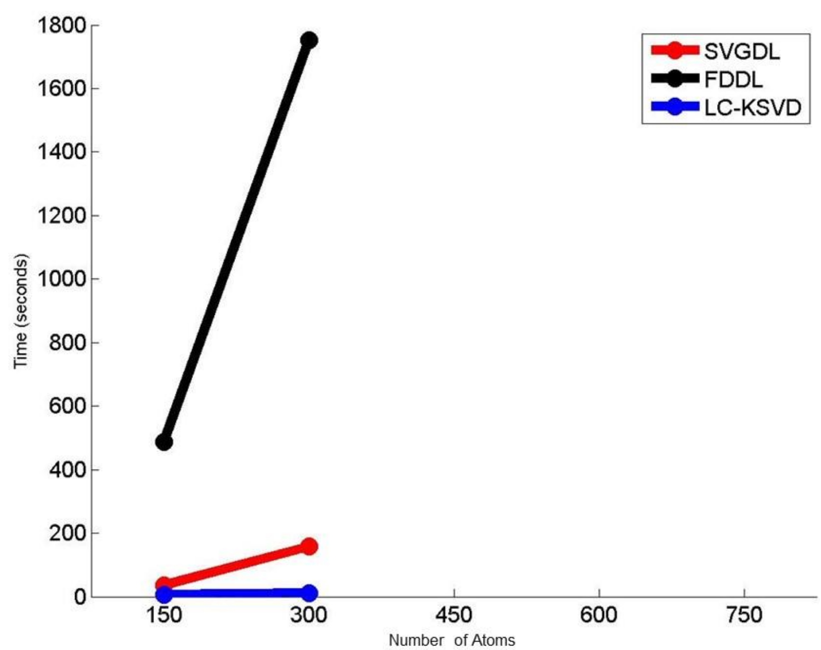

Figure 24. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 6, respectively.

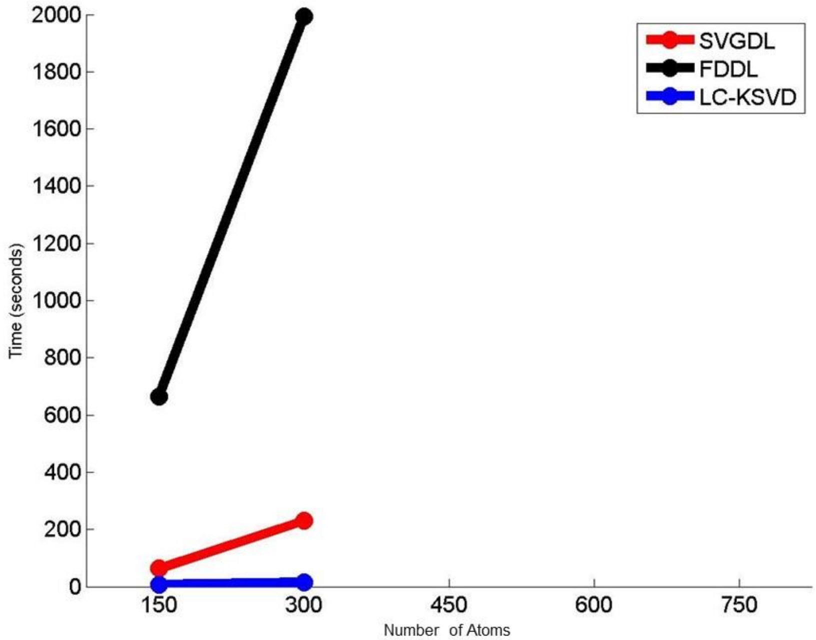

Figure 25. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 10, respectively.

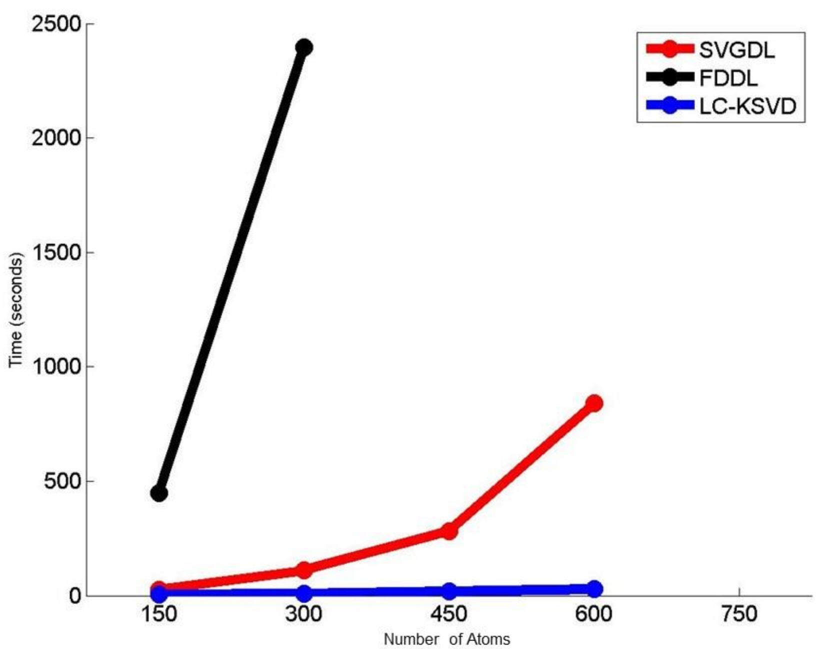

Figure 26. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 4, respectively.

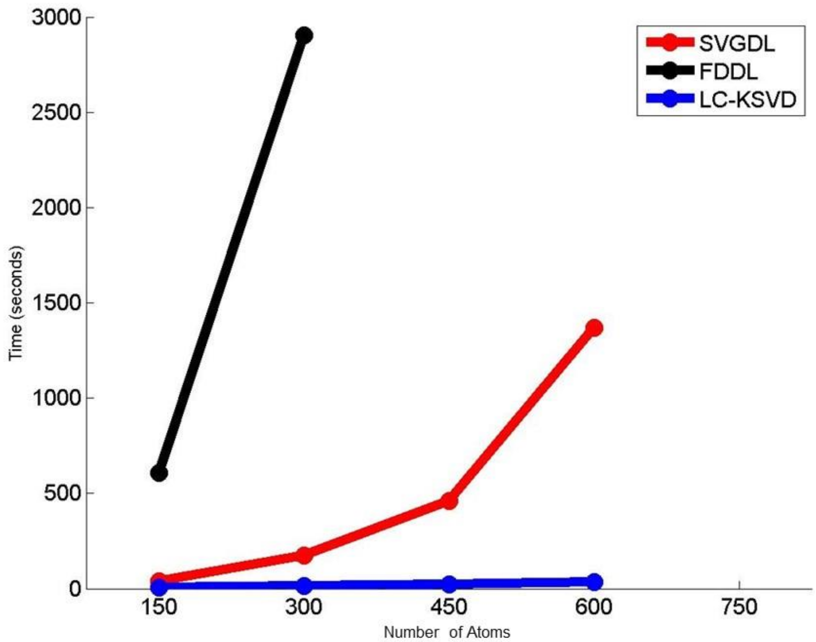

Figure 27. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 6, respectively.

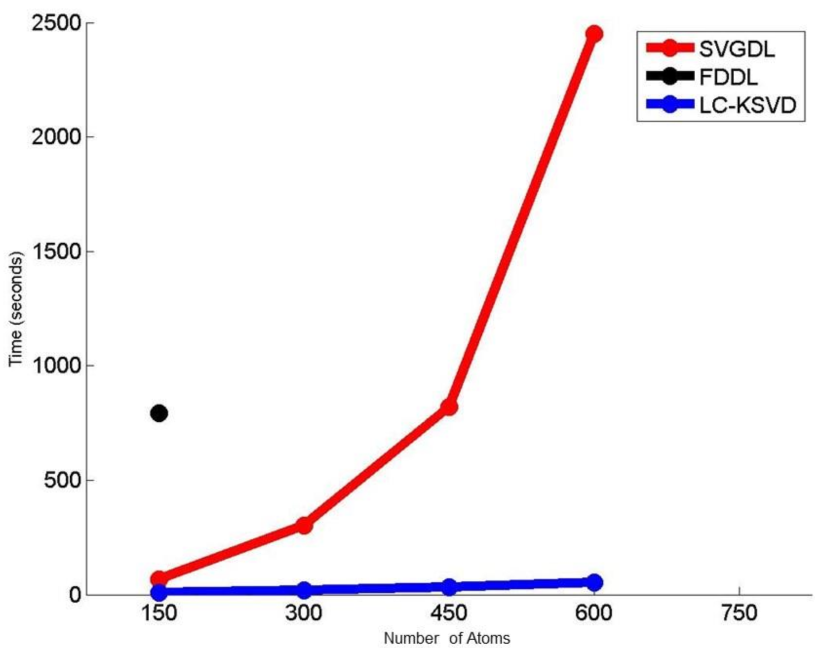

Figure 28. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 10, respectively.

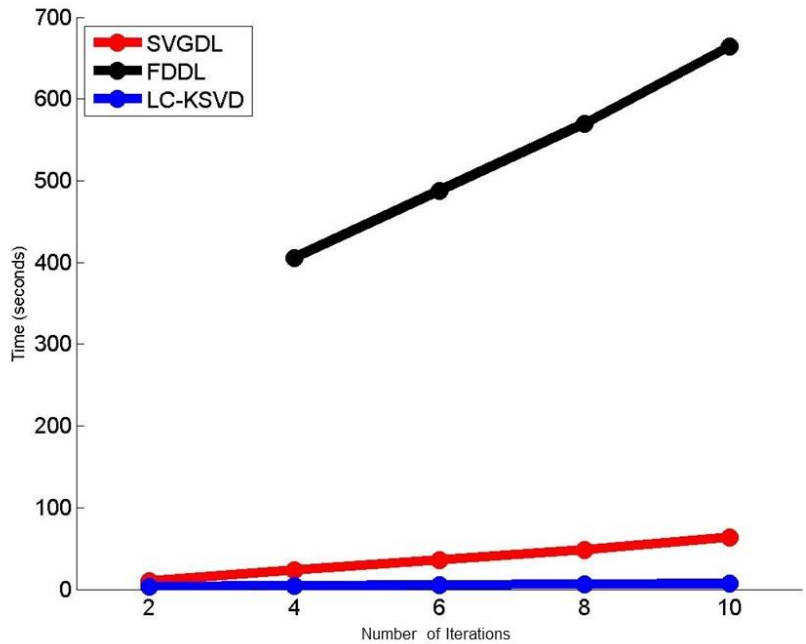

Figure 29. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 650 and 150, respectively.

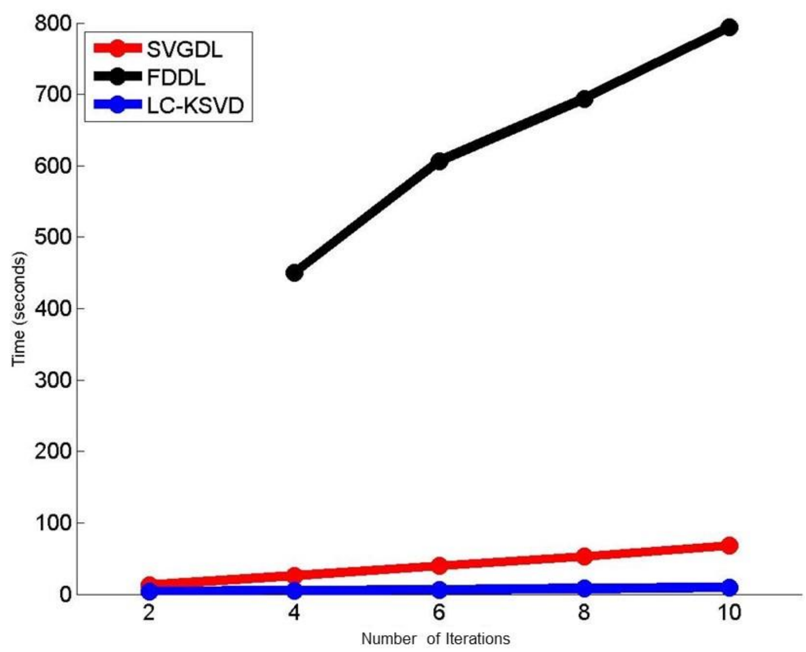

Figure 30. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 950 and 150, respectively.

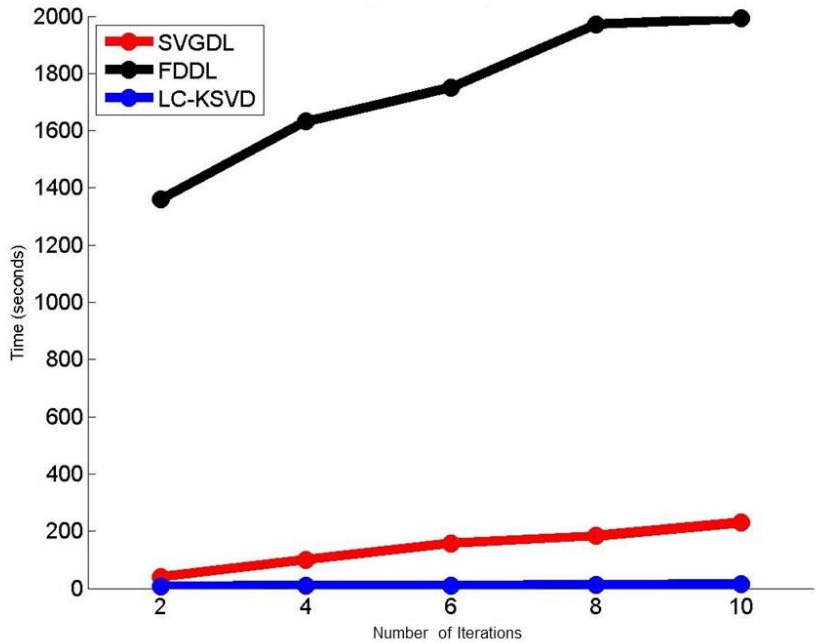

Figure 31. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 650 and 300, respectively.

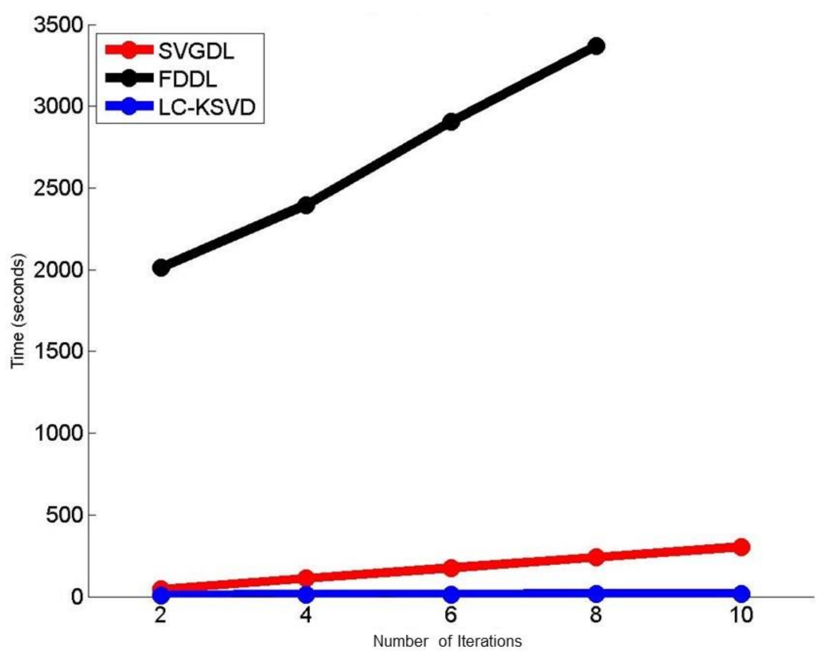

Figure 32. Face detection time for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 950 and 300, respectively.

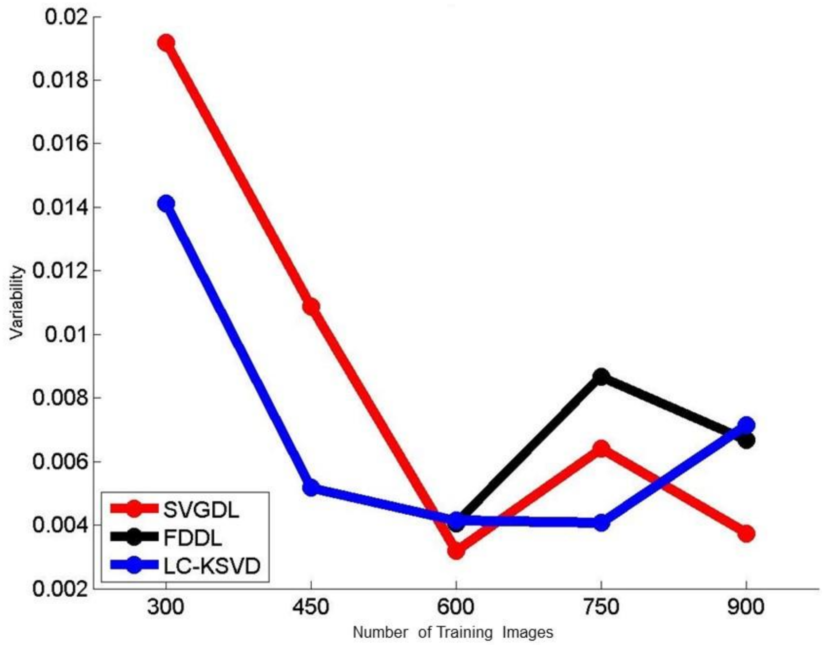

Figure 33. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 4, respectively.

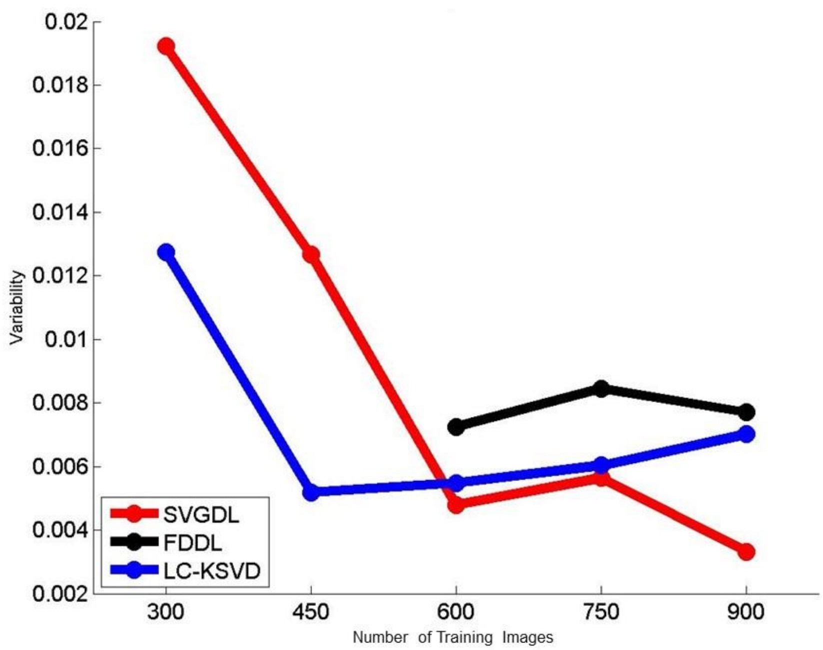

Figure 34. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 6, respectively.

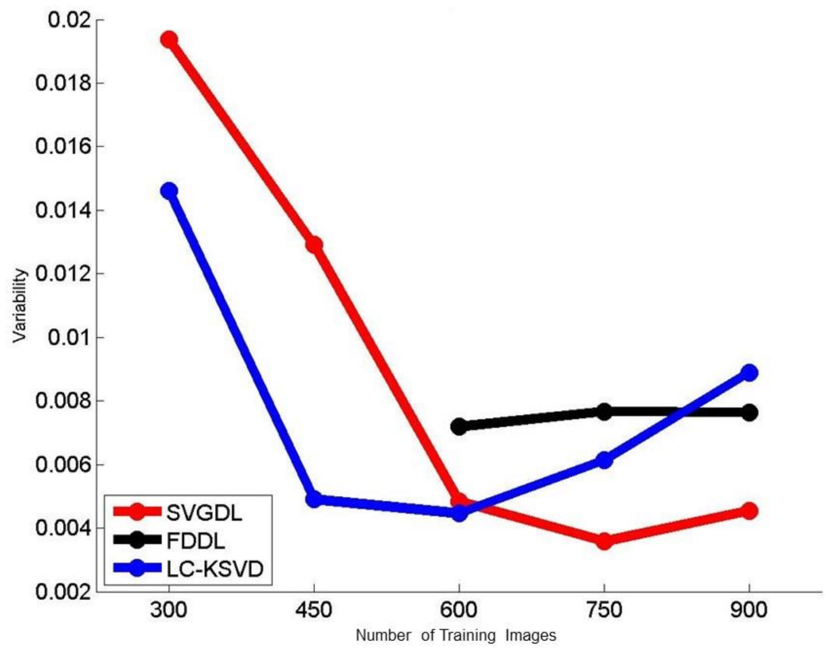

Figure 35. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 150 and 10, respectively.

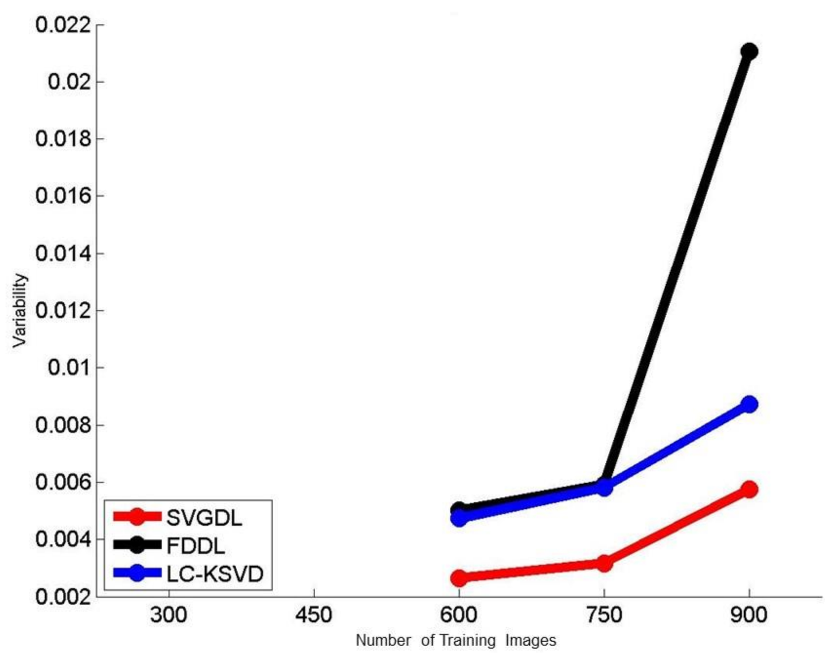

Figure 36. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 4, respectively.

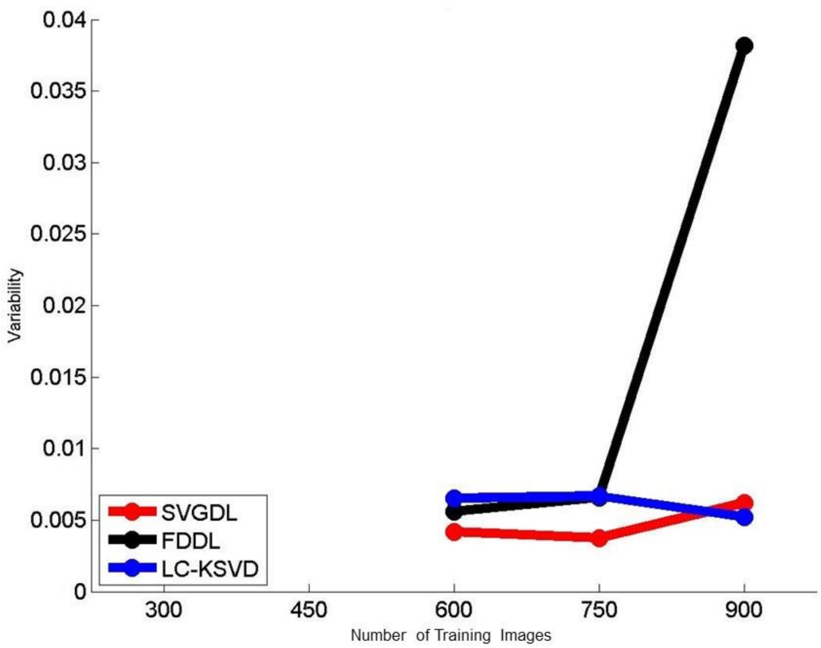

Figure 37. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 6, respectively.

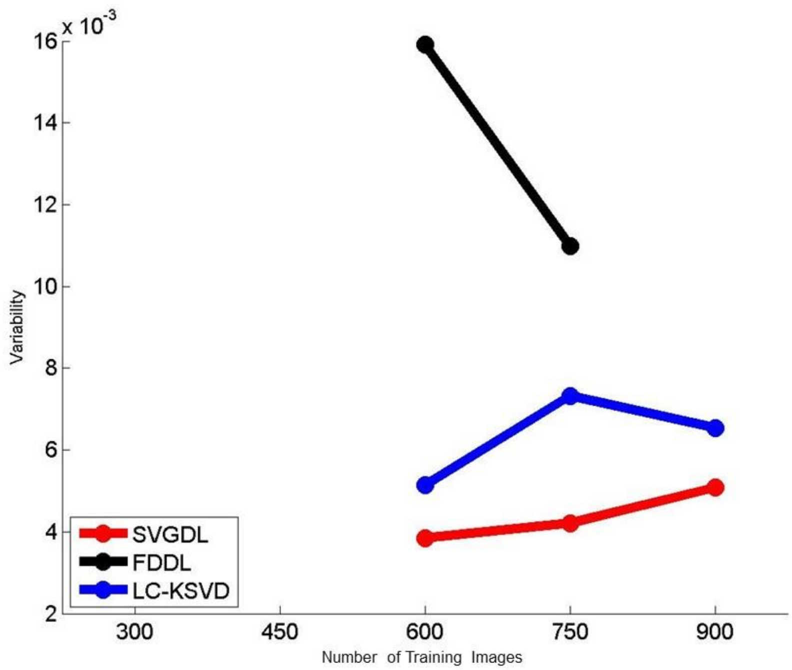

Figure 38. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of training images increases. The number of atoms and iterations are 300 and 10, respectively.

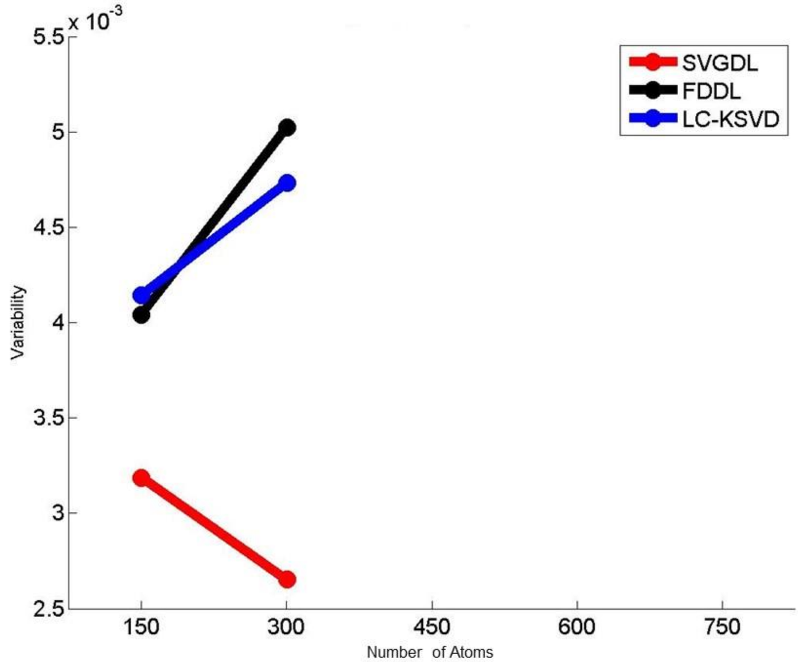

Figure 39. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 4, respectively.

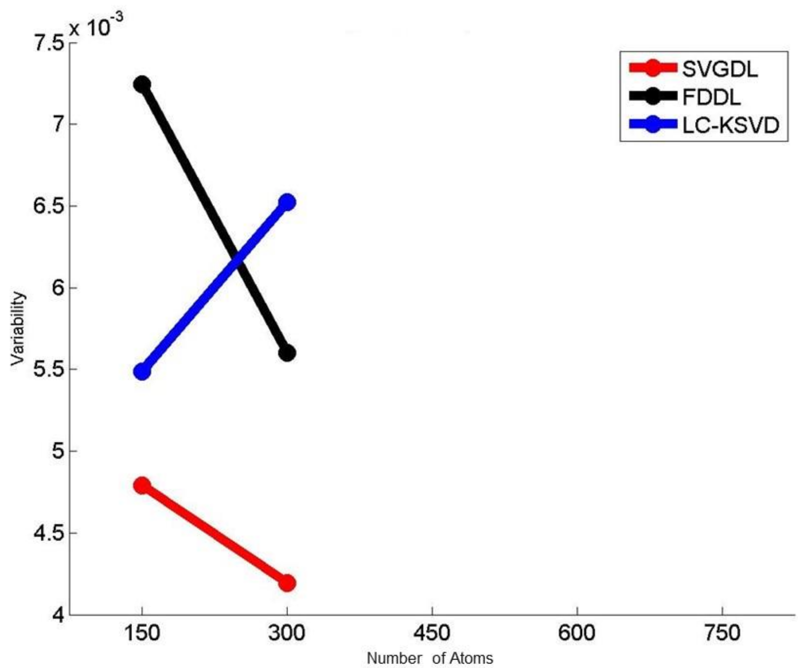

Figure 40. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 6, respectively.

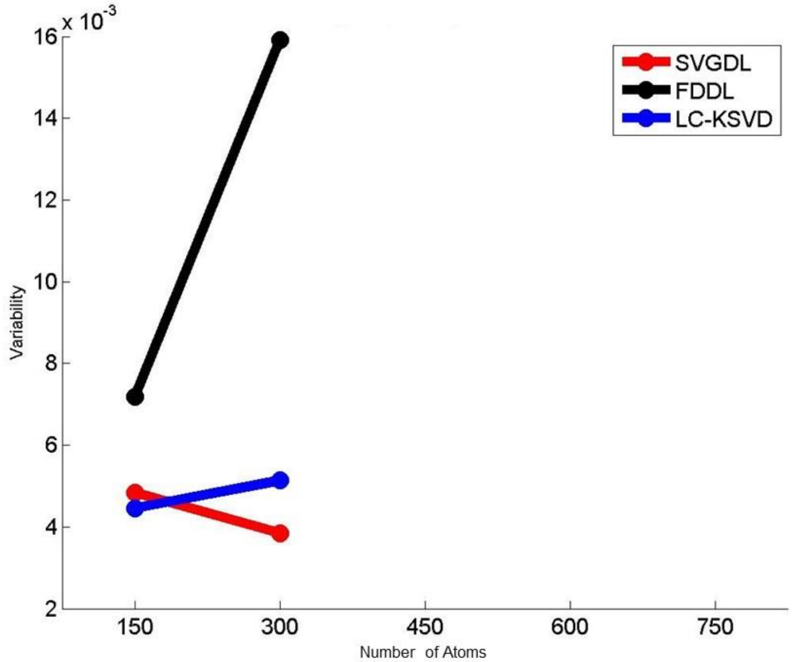

Figure 41. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 650 and 10, respectively.

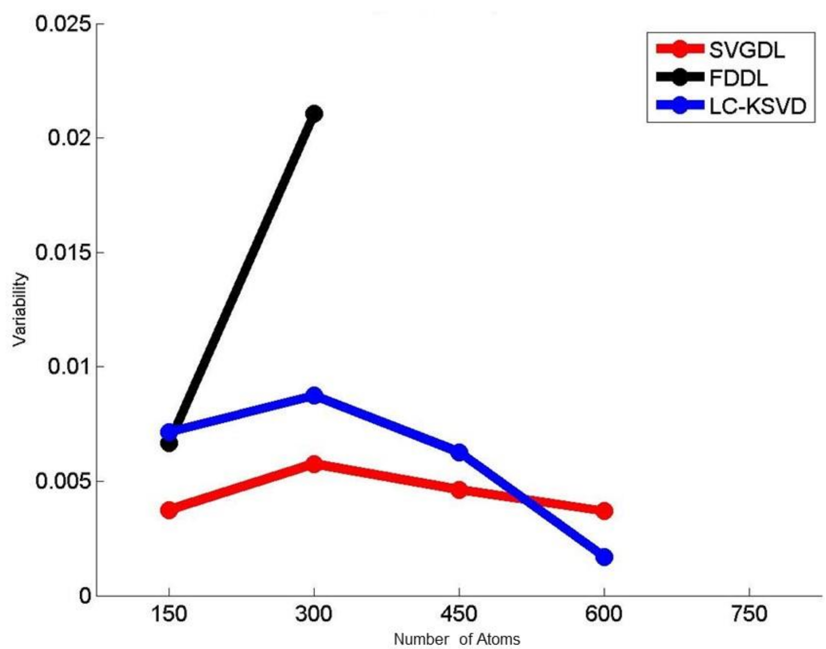

Figure 42. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 4, respectively.

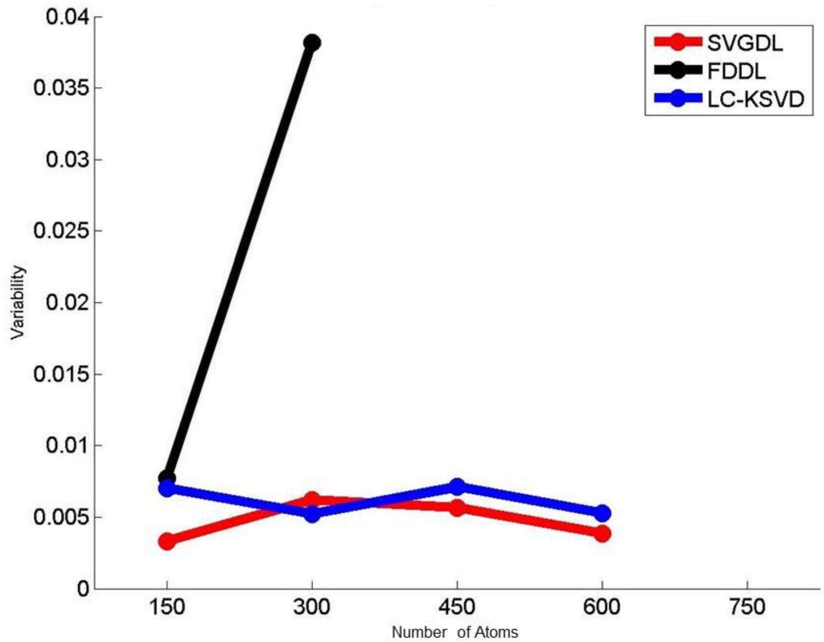

Figure 43. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 6, respectively.

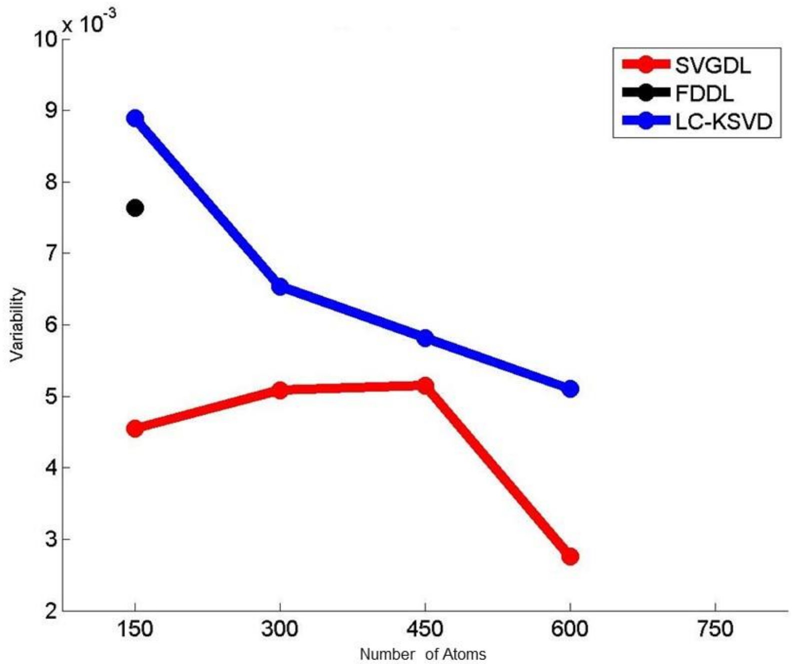

Figure 44. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of atoms increases. The number of training images and iterations are 950 and 10, respectively.

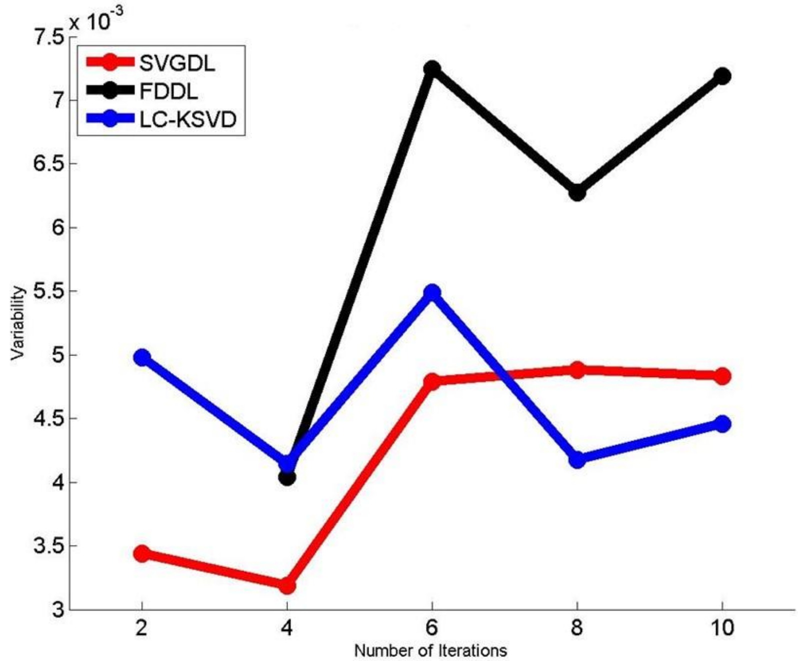

Figure 45. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 650 and 150, respectively.

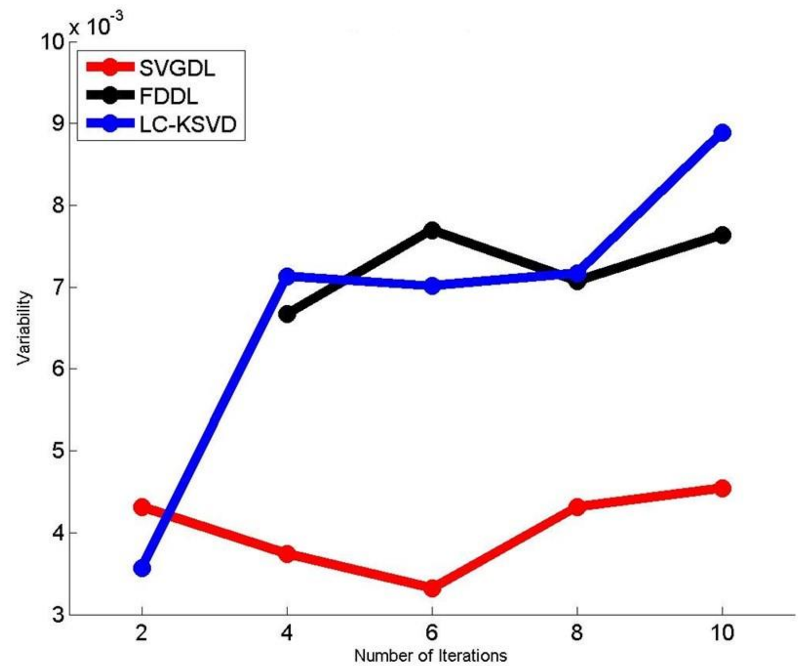

Figure 46. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 950 and 150, respectively.

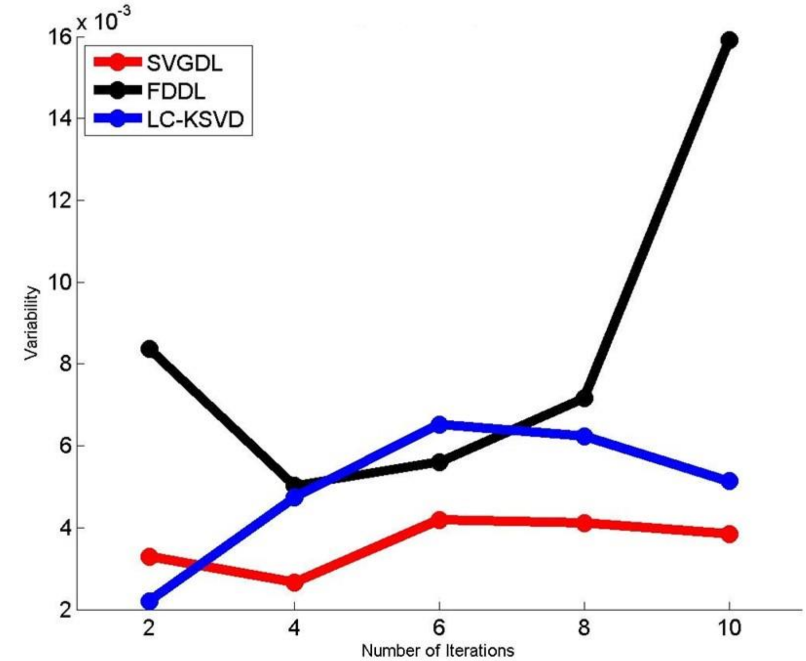

Figure 47. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 650 and 300, respectively.

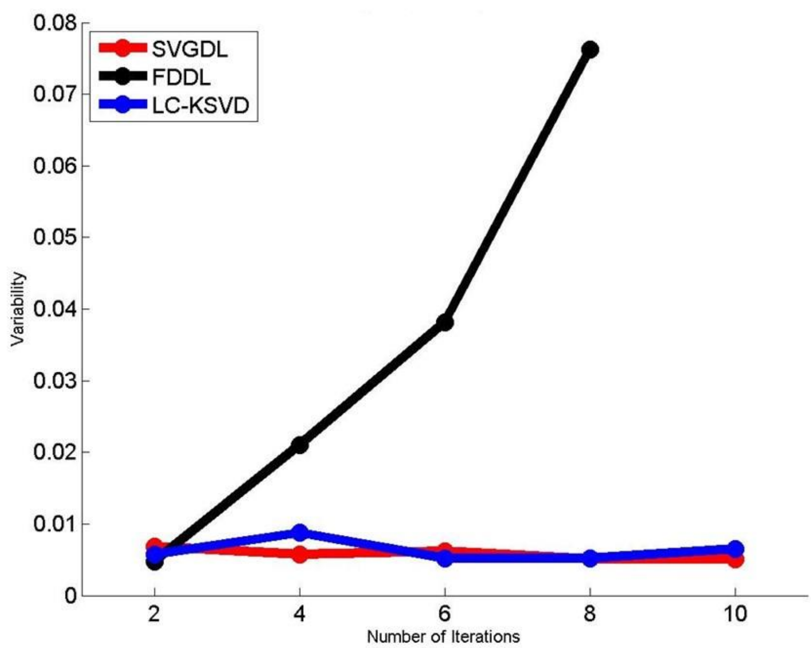

Figure 48. Face detection variability for the LC-KSVD, FDDL and SVGDL algorithms versus the number of iterations increases. The number of training images and atoms are 950 and 300, respectively.

Chapter 4

Conclusion and Future Work

In this project, three dictionary learning algorithms for face recognition were implemented in MATLAB and compared using the Extended Yale B database. These algorithms were Label-Consistent K-SVD (LC-KSVD), Fisher Discriminative Dictionary Learning (FDDL), and Support Vector Guided Dictionary Learning (SVGDL). Accuracy, speed, and variability were considered as measures to test these algorithms. The number of training images, atoms, and iterations were considered as input parameters in order to evaluate the relationship between the measures and parameters. The results obtained for each parameter were presented with the other two fixed.

The FDDL and SVGDL algorithms are both specific class dictionary learning algorithms, and SVGDL is a shared dictionary algorithm as discussed in Chapter 1. In FDDL and SVGDL, the intra-class variation of face images is large and can be greater than the inter-class variance of the face images. These algorithms build a dictionary for each class and so different dictionaries are constructed. This is why the speed of these algorithms is slow. However, the inter-class variations of the face images with the LC-KSVD algorithm are large so a dictionary can adequately capture the main characteristics of the images. Therefore, the speed of LC-KSVD algorithm is fast because only a shared dictionary is constructed using training images from all classes.

Increasing the number of training images results in a dictionary with more images and so the percentage of training images correctly assigned is increased. Hence, an increase in the number of training images results in better accuracy. SVGDL preserves the main characteristics of the face images better than the other two algorithms so it achieves a higher accuracy and provides better performance than LC-KSVD and FDDL. There were some variations in the results because each experiment was performed using randomly selected test images.

SVGDL and FDDL are less sensitive to variations in the number of atoms than LC-KSVD. Since the purpose of evaluating different measures is a fair comparison of the algorithms and FDDL did not converge in some cases, the corresponding curves were ignored for SVGDL and LC-KSVD.

To evaluate the variability, the set of images was changed for a fixed number of training images and the corresponding accuracy error was calculated. LC-KSVD has similar performance to SVGDL in terms of speed, variability and accuracy with a high number of atoms. FDDL has the worst performance. The reason is its low speed, high variability and low accuracy in the majority of conditions. In summary, the accuracy and variability results showed that SVGDL is better than the other two algorithms. Further, LC-KSVD is the fastest algorithm followed by SVGDL and then FDDL.

Future work can compare additional face recognition algorithms as well as other image databases or parameters that have not yet been examined and may affect the recognition efficiency. In addition, a multi-parametric analysis can be useful to understand the complex relations between several input and output parameters at the same time.

References

[1] T. Nagaria, D. Chourishi, "A comprehensive survey and detailed study on various face recognition methods", Int. Journ. Eng. Tech., vol. 5, no. 12, pp. 655-661, Dec. 2018.

[2] B. Khade, H. Gaikwad, A. Aher, K. Patil, "Face recognition techniques: A survey", Int. Journ. Comput. Sci. Mob. Comput., vol. 5, no. 11, pp. 65-72, Nov. 2016.

[3] G. Kaur, M. Sandhu, "Facial recognition: Issues, techniques and applications", Int. Journ. Adv. Comput. Sci., vol. 6, no. 2, pp. 508-512, Feb. 2016.

[4] Y. Xu, Z. Li, J. Yang, D. Zhang, "A survey of dictionary learning algorithms for face recognition", IEEE Access, vol. 5, pp. 8502-8514, Apr. 2017.

[5] Y. Suo, M. Dao, U. Srinivas, V. Monga, T. Tran, "Structured dictionary learning for classification", IEEE Trans. Signal Process., vol. 14, pp. 150-154, Jun. 2014.

[6] K. Huang, D. Dai, C. Ren, Z. Lai, "Discriminative kernel collaborative representation with locality constrained dictionary for face recognition", IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 5, pp. 1082-1094, May. 2017.

[7] Y, Sun, Q. Liu, J. Tang, D. Tao, "Learning discriminative dictionary for group sparse representation", IEEE Trans. Image Process., vol. 23, no. 9, pp. 3816-3828, Sep. 2014.

[8] S. Gao, I. Tsang, Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization", IEEE Trans. Image Process., vol. 23, no. 2, pp. 623-634, Feb. 2014.

[9] W. Deng, J. Hu, J. Guo, "Undersampled face recognition via intraclass variant dictionary", IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 9, pp. 1864-1870, Sep. 2012.

[10] Z. Wang, J. Yang, N. Nasrabadi, T. Huang, "A max-margin perspective on sparse representation-based classification", Proc. IEEE Int. Conf. Comput. Vis., pp. 1217-1224, Dec. 2013.

[11] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, "Robust face recognition via sparse representation", IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 2, pp. 210-227, Feb. 2009.

[12] N. A. Mehta, A. G. Gray, "Sparsity-based generalization bounds for predictive sparse coding", Proc. Int. Conf. Mach. Learn., pp. 36-44, Jun. 2013.

[13] Z. Jiang, Z. Lin, L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1697-1704. Jun. 2011.

[14] Q. Zhang, B. Li, "Discriminative K-SVD for dictionary learning in face recognition", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2691-2698, Jun. 2010.

[15] M. Aharon, M. Elad, A. M. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation", IEEE Trans. Signal Process., vol. 54, no. 11, pp. 4311-4322, Nov. 2016.

[16] Z. Jiang, Z. Lin, L. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition", IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 11, pp. 2651-2664, Nov. 2013.

[17] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, J. Ponce, "Discriminative sparse image models for class-specific edage detection and image interpretation", Proc. Eur. Conf. Comput. Vis., pp. 43-56, Oct. 2008.

[18] H. Zheng, D. Tao, "Discriminative dictionary learning via Fisher discrimination K-SVD algorithm", Neurocomputing, vol. 62, pp. 9-15, Aug. 2015.

[19] R. Jiang, B. Zhang, H. Qiao, "Efficient Fisher discrimination dictionary learning", IEEE Trans. Signal Process., vol. 128, pp. 28-39, Mar. 2016.

[20] M. Yang, L. Zhang, X. Feng, D. Zhang, "Fisher discrimination dictionary learning for sparse representation", Proc. IEEE Int. Conf. Comput. Vis., pp. 543-550, Nov. 2014.

[21] S. Cai, W. Zuo, L. Zhang, X. Feng, P. Wang, "Support vector guided dictionary learning", Proc. Eur. Conf. Comput. Vis., pp. 624-639. Sep. 2014.

[22] S. Gao, I. W.-H. Tsang, L.-T. Chia, "Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications", IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 92-104, Jan. 2013.

[23] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, "Supervised dictionary learning", Proc. Adv. Neural Inf. Process. Syst. Conf., vol. 21, pp. 1033-1040, Jun. 2009.

[24] C. Chang, C. Lin, "LIBSVM: A library for support vector machines", IEEE Trans. Intell. Syst. Techno., vol. 2, pp. 1889-1918, Mar. 2011.

[25] L. Shen, S. Wang, G. Sun, S. Jiang, Q. Huang, "Multi-level discriminative dictionary learning towards hierarchical visual categorization", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 383-390, Jun. 2013.