iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 Issue: II Month of publication: February 2015

DOI:

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Fingerprint Compression Based On Online Dictionary Learning and Orthogonal Matching Pursuit

[1]Dinesh.E, [2]Ramyasridevi.R, [3]Revathi.K, [4]Selvam.V, [5]Vicithra.S
[1]*Assistant Professor, Department of CSE, Narasu's Sarathy Institute of Technology, Salem.*
[2,3,4,5] *UG Students, Department of CSE, Narasu's Sarathy Institute of Technology, Salem.*

***Abstract*: A new fingerprint compression technique based on the concept of sparse representation is introduced. Given a set of fingerprint patches, an over complete dictionary is constructed and the patches are represented as a sparse linear combination of dictionary atoms. The methods prior to this used K-SVD algorithm for constructing dictionary and MP (Matching Pursuit) algorithm to solve the l^0 minimization problem. The common step in all of the algorithms is that a dictionary is first constructed for predefined fingerprint image patches and for a new given fingerprint image its patches are represented according to the dictionary by computing l^0 minimization. The next step is to quantize and encode the representation. This technique uses an method called ODL (Online Dictionary Learning) for dictionary construction and OMP (Orthogonal Matching Pursuit) algorithm to solve the l^0 minimization problem. The results of this method are provided in this paper and compared with the results of batch and stochastic gradient method to determine the efficient compression method.**
Keywords: Sparse Representation, K-SVD, Matching Pursuit, Online Dictionary Learning, Orthogonal Matching Pursuit.

I. INTRODUCTION

Biometrics is an important technology of analyzing and measuring biological data. It is used for authentication purposes and analyzing human body characteristics such as facial patterns, voice patterns, and fingerprints. Fingerprint recognition is most widely deployed technology. It is used for personal identification and forensic science [1]. Large number and size of fingerprint images consumes large memory space, fingerprint compression technique is used to make limited storage space and it requires less time to transmit data. Compression technology is of two types lossless and lossy. In lossless compression reconstruct the original image from compressed data without data loss. Original images are same as the reconstructed image from the compressed data. During compression redundant data is removed and that data is added in decompression. When we need all data after decompression also lossless methods are used. In lossy compression reconstruct the original image from compressed data with some loss of data. It allows reconstruction only of an approximation of the original data. In this original data are not equal to the decompressed data. It reduces the size of the data by eliminating redundant data [2].

In existing three dictionary designing algorithms are used, K-SVD [3], orientation and random methods. K-SVD is an iterative method, based on the current dictionary it alternates the input data; It is used for better data by updating the data in the dictionary. K-SVD can work with any pursuit method. When the size of dictionary is large K-SVD is not effective. Matching pursuit (MP) [4] optimization algorithm is used; it is a greedy algorithm that sequentially select the dictionary data. It is approximation of sparse. In this paper we proposed ODL [5] to overcome the problems of K-SVD and OMP [6] optimization algorithm.

II. RELATED WORK

Sparse signal representation [7] has emerged to be an extremely powerful tool for obtaining and compressing high-dimensional signals. This success is mainly due to the reason is that important classes of signals such as audio and images have naturally sparse representations with respect to fixed bases like Fourier, Wavelet transformations or combination of such bases. Variations and extensions of $l1$- minimization have been applied to many computer vision tasks, like face recognition, image super-resolution, motion and data segmentation, supervised denoising and background modeling and image classification[8]. Therefore to successfully apply sparse representation to computer vision tasks, we need to address the problem of how to correctly choose the basis for representing the data. There are many domains where sparse representation plays a vital role.

A. Sparse representation in image inpainting

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

There are some patch wise image inpainting [9] algorithms available to deal with large holes and preserve image details while taking less risk. Here every image patch admits a sparse representation over a redundant dictionary. Now, we construct a redundant signal dictionary by directly sampling from the intact source region of current image. Then we sequentially compute the sparse representation for each incomplete patch at the boundary of the hole and recover it until the whole hole is filled.

B. Sparse representation in image denoising

In the field of image denoising, the K-SVD [10] algorithm is extended mainly for gray-scale image denoising. This puts forward the ways for handling non homogeneous noise and missing information, giving way to best results in applications such as color image denoising and inpainting.

C. Sparse representation in image restoration

We need to consider an image patch of certain size including the overlaps. After, learning the dictionaries of the corrupted image, the sparse coding retrieves a sparse approximation of noisy patches[11]. Finally, averaging the approximation of each patch gives the path to reconstruct the whole image. This is done with help of K-SVD algorithm.

D. Sparse representation in image resolution

Sparse representation is also used in the field of image super-resolution [12]. Here also we adapt the same process of representing the image as patches and then classifying these patches into low and high resolution patches. In the first step, we obtain the coefficients of low resolution input and then generate the high resolution output from them. . By training two dictionaries for both low and high resolution image patches simultaneously, we can obtain the similarity of sparse representations between the low resolution and high resolution image patch pair with respect to their own dictionaries. This reduces the computational cost and in addition, the local sparse modeling of this approach is naturally robust to noise, and therefore it can handle super-resolution with noisy inputs in a more unified way. It is used in the field of machine learning and other important applications are in sensing, communications [13] and control like medical imaging, Radar Imaging, sparsity-based event detection and classification in sensor networks. In recent times new techniques have been proposed to find sparse representation of signals mainly for analyzing musical signals. While analyzing this, we will be able to recover the sparse objects that produced the musical audio signal and this kind of representation could be used for identifying musical notes from the audio, efficient coding of musical audio.

III. FINGERPRINT COMPRESSION BASED ON SPARSE CONCEPT

In this section, the details about compressing a fingerprint image based on sparse representation [14] are provided. This section includes the process of slicing an image into patches, dictionary construction, compression of the given fingerprint image and finally quantization and coding of the coefficients. Nowadays lots of information is stored and the size of the dictionary also increases in a constant pace. Therefore, a dictionary of a modest size is obtained by doing the necessary preprocessing steps. Due to factors like transformation, rotation and noise the fingerprint of the same finger may have various looks. The common pre-alignment technique is to translate and rotate the fingerprint according to the position of the core point. However the detection of the core point is a difficult task in images that are poor in quality. Even though the core is correctly detected, the dictionary size may be huge because the size of a whole fingerprint image is too large. Compared to natural images, the fingerprint images are simpler in structure. They comprise of only ridges and valleys. But in the local regions these ridges and valleys look similar. Therefore, the solution to these two problems is that the whole fingerprint image is sliced into square and non-overlapping small patches. These patches are independent of transformation and rotation. The size of the dictionary is small because the small blocks are relatively smaller.

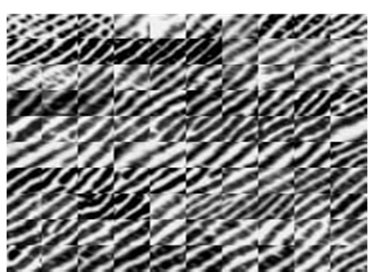


Fig.3.1. A patched Fingerprint image

A. Dictionary construction

In this paper, the dictionary is constructed using ODL [15] method. The training set is constructed first and from this set the dictionary is obtained. The predefined fingerprint images are cut into fixed size square patches. Then these patches are used to construct the training samples. The first patch is added to the empty dictionary and then the next patch is tested for similarity by solving the optimization problem. If similar the patch is avoided or else added to the dictionary. The mean value is calculated for each patch and subtracted from the particular patch before the dictionary is constructed.

1) *Online Dictionary Learning Method (ODL):* Assume that the training set is composed of samples of a distribution p(x) [16]. The inner loop draws one element x_t at a time and differs with the classical sparse coding steps for computing the sparse vector over α_t of x_t over the dictionary D_{t-1} that was obtained from the previous iteration, with dictionary update steps where the new dictionary D_t is computed by minimizing over the function.
The vector α_i is computed in the previous steps. The advantages of this approach is that it can :

a) handle infinitely large datasets
b) faster than batch algorithms
c) easily adaptable to dynamic training sets

Algorithm 1: Require: $x \in \mathbb{R}^m \sim p(x)$ (random variable and an algorithm to draw i.i.d samples of p), $\lambda \in \mathbb{R}$(regularization parameter), $D_0 \in \mathbb{R}^{m\times\kappa}$ (initial dictionary), T(number of iterations).
1:$A_0 \leftarrow 0, B_0 \leftarrow 0$ (reset the "past" information)
2: for t=1 to T do
3: Draw x_t from p(x)
4: Sparse coding: compute using LARS

$$\alpha_t \triangleq \arg\min_{\alpha\in\mathbb{R}^\kappa} \frac{1}{2}\|X_t - D_{t-1}\alpha\|_2^2 + \lambda \parallel \alpha \parallel_1$$

5: $A_t \leftarrow A_{t-1} + \alpha_t\alpha_t^T$
6: $B_t \leftarrow B_{t-1} + X_t\alpha_t^T$
7: Compute D_t using Dictionary Update algorithm with D_{t-1} as warm restart, so that

$$D_t \triangleq \arg\min_{D\in\mathcal{C}} \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\|X_i - D\alpha_i\|_2^2 + \lambda \parallel \alpha_i \parallel_1,$$

$$= \arg\min_{D\in\mathcal{C}} \frac{1}{t}\left(\frac{1}{2}Tr(D^T D A_t) - Tr(D^T B_t)\right)$$

8: end for
9: Return D_T (learned dictionary)

B. Dictionary update

The algorithm for dictionary update is provided and it uses block coordinate descent with warm restarts, and it does not use any parameter so there is no need for learning rate tuning. This algorithm sequentially updates D. The j-th column is updated while keeping the other constraints constant. As the vectors α_i are sparse, the coefficients in the matrix A are majorly in the diagonal, that makes the block-coordinate descent more efficient. A single iteration is enough because D_t is found from D_{t-1}.There are also other methods to update the dictionary but this is found to be efficient.

Algorithm 2: Require: $D = [d_1, \ldots., d_k] \in \mathbb{R}^{m\times k}$(input dictionary)

$$A = [a_1, \ldots., a_k] \in \mathbb{R}^{k\times k} = \sum_{i=1}^{t} \alpha_i\, \alpha_i^T$$

$$B = [b_{1,\ldots\ldots\ldots,} b_k] \in \mathbb{R}^{m\times k} = \sum_{i=1}^{t} X_i\, \alpha_i^T$$

1: repeat
2: for j=1 to k do
3: Update the j-th column to optimize for (9):

$$u_j \leftarrow \frac{1}{A_{jj}}(b_j - Da_j) + d_j$$

4: end for

$$d_j \leftarrow \frac{1}{\max(\| u_j \|_2, 1)} u_j$$

5: until convergence
6: Return D (update dictionary).

IV. ORTHOGONAL MATCHING PURSUIT

A pursuit method for sparse approximation is a greedy approach that iteratively refines the current estimate for the coefficient vector x by modifying one or several coefficients chosen to yield a substantial improvement in approximating the signal. We begin by describing the simplest effective greedy algorithm, orthogonal matching pursuit (OMP) [17], and summarizing its theoretical guarantees. Afterward, we outline a more sophisticated class of modern pursuit techniques that has shown promise for compressive sampling problems. We briefly discuss iterative thresholding methods, and conclude with some general comments about the role of greedy algorithms in sparse approximation.

A. Orthogonal matching pursuit

It is a greedy method for solving the sparse approximation problem. This method is very straight forward as the approximation is generated by going through an iteration process. Through an iteration process it constructs an approximation. It is used to calculate locally optimum solution. This is done by finding the column vector in A which most closely resembles a residual vector r. The residual vector equals to the vector that is required to be approximated (i.e) r = b and is adjusted at each iteration to take into account the vector previously chosen. It is the hope that this sequence of locally optimum solutions will lead to the global optimum solution. As usual this is not the case in general although there are conditions under which the result will be the optimum solution. OMP is based on a modification of an earlier algorithm called Matching Pursuit (MP). MP simply removes the selected column vector from the residual vector at each iteration.

$$r_t = r_{t-1} < a_{op}, r_{t-1} > r_{t-1}$$

Where a_{op} is the column vector in A which most closely resembles r_{r-1} [18]. OMP uses a least-squares step at each iteration to update the residual vector in order to improve the approximation.

Input:

1) Signal b and matrix A.
2) Stopping criterion e.g. until a level of accuracy is reached.

Algorithm 3:

1. Start by setting the residual $r_0 = b$, the time t = 0 and index set $V_0 = \Phi$;
2. Let $v_t = i$; where ai gives the solution of max $< r_t, a_k >$ where a_k are the row vectors of A.
3. Update the set Vt with
v_t: $V_t = V_{t-1} \cup \{ v_t \}$
4. Solve the least-squares problem
$\min_{c \in C^{v_t}} ||b - \sum_{j=1}^{t} c(v_j)\, a_{V_j}||_2$
5. Calculate the new residual using c

$$r_t = r_{t-1} \sum_{j=1}^{t} c(v_j)\, a_{vj}$$

6. Set $t \leftarrow t + 1$
7. Check stopping criterion if the criterion has not been satisfied then return to step2.

Output:

Approximation vector c

We examine the performance limits of the Orthogonal Matching Pursuit (OMP) algorithm, which has proven to be good in solving for sparse solutions to inverse problem arising in over complete representations. We exploit the connection between sparse solution problem and multiple access channel (MAC) in wireless communication domain to identify these limits. Some of the performance criteria of OMP are:

OMP as a successive cancellation scheme of interference [19], Criterion for OMP to success, Limit of OMP is insights into the performance.

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

V. EXPERIMENTAL VALIDATION

In this section, we experimented on few fingerprint images to demonstrate the efficiency of our method.

A. Performance Evaluation

To construct a dictionary of fingerprint patches, we use a set of training fingerprints which includes the major pattern types. The distribution of different pattern in this set is not necessarily similar to the distribution in the other database. There are 5 groups of fingerprint images (referred to as DATABASE 1, DATABASE 2, DATABASE 3, DATABASE 4 and DATABASE 5) in the experiments.

Data Signal size m N_{bk} of atoms Type
DATABASE 1: 8 × 8 = 64 256 b&w
DATABASE 2: 12 × 12 × 3 = 432 512 color
DATABASE 3: 16 × 16 = 256 1024 b&w

We have normalized the patches to have unit ℓ2-norm and used the regularization parameter $\lambda = 1.2/\sqrt{m}$ in all of our experiments. The $1/\sqrt{m}$ term is a classical normalization factor and the constant 1.2 has been experimentally shown to yield reasonable sparsities (about 10 nonzero coefficients) in these experiments.

1) *Online vs Batch:* Comparing the online and batch setting, the online version of our algorithm draws samples from the entire set, and we have run its batch version on the full dataset as well as subsets. The online setting systematically outperforms its batch counterpart for every training set size and desired precision. We use a logarithmic scale for the computation time, which shows that in many situations, the difference in performance can be dramatic. Similar experiments have given similar results on smaller datasets.
2) *Comparison with Stochastic Gradient Descent:* Our experiments have shown that obtaining good performance with stochastic gradient descent requires using both the mini-batch heuristic and carefully choosing the learning rate ρ. To give the fairest comparison possible, we have thus optimized these parameters, sampling η values among. Powers of 2 and ρ values among powers of 10.The combination of values ρ = 104, η = 512 gives the best results on the training and test data for stochastic gradient descent. We observe that the larger the value of ρ is, the better the eventual value of the objective function is after much iteration, but the longer it will take to achieve a good precision. Although our method performs better at such high-precision settings for dataset C, it appears that, in general, for a desired precision and a particular dataset, it is possible to tune the stochastic gradient descent algorithm to achieve a performance similar to that of our algorithm. Note that both stochastic gradient descent and our method only start decreasing the objective function value after a few iterations. Slightly better results could be obtained by using smaller gradient steps during the first iterations, using a learning rate of the form $\rho/(t+t_0)$ for the stochastic gradient descent, and initializing $A_0 = t_0$ I and $B_0 = t_0 D_0$ for the matrices A_t and Bt,where t_0 is a new parameter.
3) *Comparison between MP and OMP:* The OMP we have described is a modification of the Matching Pursuit (MP) algorithm of that improves convergence using an additional orthogonalization step. The advantage of OMP over MP is the fact that it is guaranteed to converge in a finite number of steps for a finite dictionary. We also showed that all additional computation that is required for OMP may be performed repeatedly. Both algorithms were compared on two criteria's of decomposition with respect to some particular dictionary. It was seen that although OMP converges in fewer iterations than MP, the calculating effort required for each algorithm depends on both the class of signals and choice of dictionary. Although we do not provide a meticulous argument here, it seems reasonable to speculation that OMP will be computationally cheaper than MP for very redundant dictionaries, as insight of the redundancy is exploited in OMP to reduce the error as much as possible.

VI. CONCLUSION

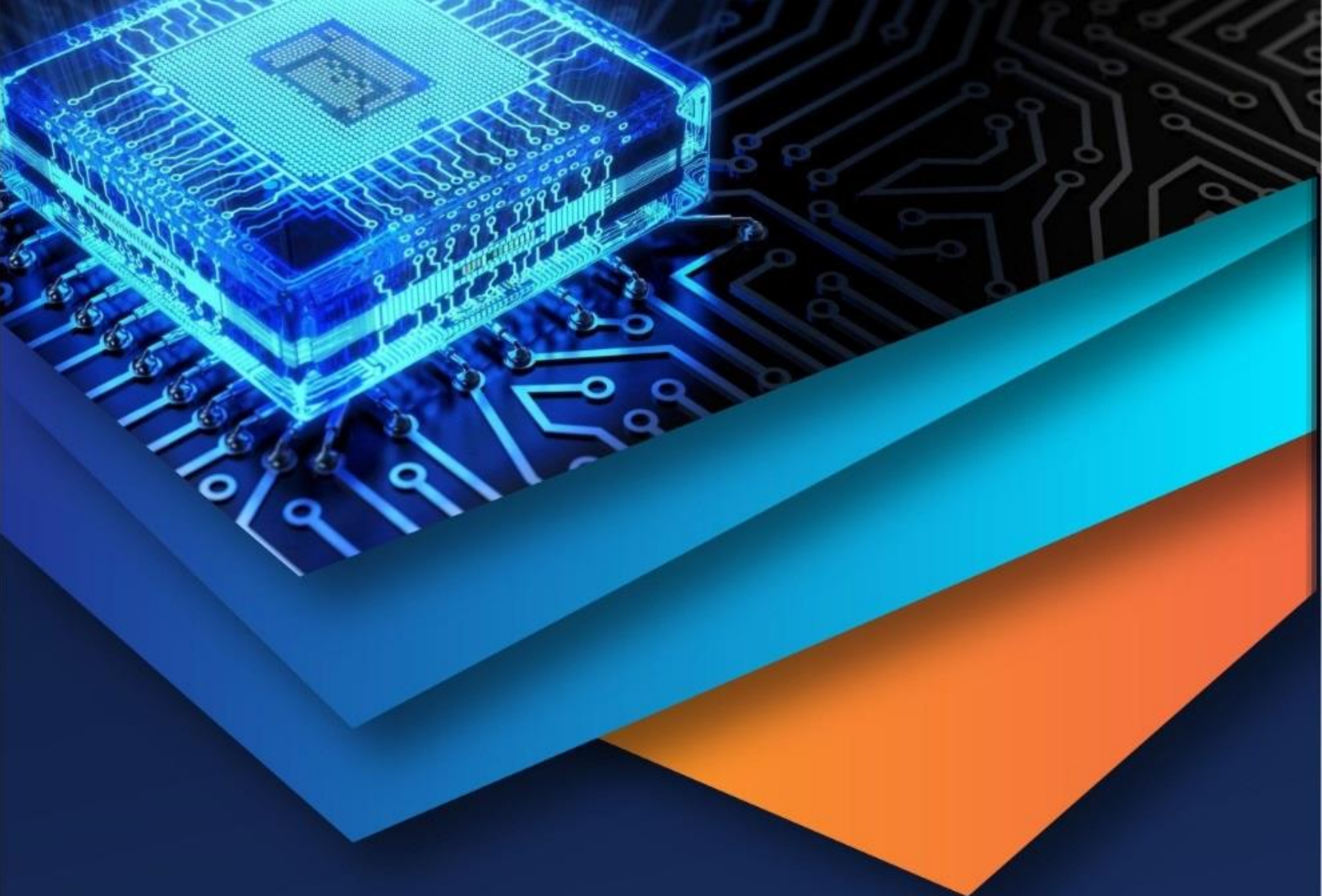
A new fingerprint compression technique based on sparse representation is provided. The proposed system is found to be more efficient than the existing compression techniques that use K-SVD and MP algorithm. The results show that the patching of the image and then processing it has less serious computations than JPEG. The experiment results show that the larger the training set better is the compression results. This algorithm is capable of processing minute details of the fingerprint. The future work can be based on the investigation of different methods for constructing the dictionary other than ODL and optimization algorithms that can further enhance the efficiency of the compression technique to find the sparse vector. Finally various other

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

applications based on sparse representation should be explored.

REFERENCES

[1] D. Maltoni, D. Miao, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. London, U.K.: Springer-Verlag, 2009.
[2] Ms.MansiKambli and Ms.Shalini Bhatia, *Comparison of different Fingerprint Compression TechniquesSIPIJ Vol.1, No.1, September 2010*
[3] O. Bryt and M. Elad, *Compression of facial images using the K-SVD algorithm, J. Vis. Commun. Image Represent., vol. 19, no. 4,pp. 270–282, 2008.*
[4] S.G.Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process., vol. 41, no. 12,pp. 3397–3415, Dec. 1993.*
[5] Julien Mairal, Francis Bach, Jean Ponce and Guillermo Sapiro, *Online Dictionary Learning for Sparse Coding.*
[6] T. Tony Cai and Lie Wang, *Orthogonal Matching Pursuit for Sparse Signal Recovery.*
[7] *Sparse approximation* from Wikipedia,the free encyclopedia.
[8] John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas Huang, Life Fellow, Shuicheng Yan *Sparse Representation For Computer Vision and Pattern Recognition* proceedings of ieee, march 2009.
[9] Bin Shen,WeiHu, Yimin Zhang,Yu-Jin Zhang *Image inpainting via sparse representation.*
[10] Francis Bach, Julien Mairal, Jean Ponce, Guillermo Sapiro *ICCV Tutorial on Sparse Coding and Dictionary Learning for Image Analysis.*
[11] Elad, M, Aharon.M *Image Denoising Via Sparse a1nd Redundant Representations Over Learned Dictionaries* Image processing, IEEE Transactions on Vol:15, Issue: 12 pp.3736 – 3745.
[12] Francis Bach, Julien Mairal, Jean Ponce, Guillermo Sapiro *ICCV Tutorial on Sparse Coding and Dictionary Learning for Image Analysis.*
[13] John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas Huang, Life Fellow, Shuicheng Yan *Sparse Representation For Computer Vision and Pattern Recognition* proceedings of ieee, march 2009.
[14] Guangqi Shao, Yanping Wu, Yong A, Xiao Liu, and Tiande Guo, *Fingerprint Compression based on Sparse Representation*, Ieee Transactions On Image Processing, Vol. 23, No. 2, February 2014
[15] Karl Skretting, *Dictionary Learning Tools for Matlab*, University of Stavanger
[16] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, *Online Dictionary Learning for Sparse Coding.*
[17] Donoho, D.L, Tsaig, Y, Drori. I,Starck, J.-L *Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit* Information Theory, IEEE Transactions on Vol:58 , Issue: 2 PP: 1094 – 1121
[18] YuzheJin, La Jolla,Rao, *Performance limits of matching pursuit algorithms* Information Theory, 2008. ISIT 2008. IEEE International Symposium on6-11 July 2008 PP: 2444 – 2448.
[19] Philip Breen ,*Algorithms for Sparse Approximation* Year 4 Project School of Mathematics University of Edinburgh 2009



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY