Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 4, April -2017

Harmonizing real time data using maximum weighted graph matching

[1]R.Pavithra, [2]Dr.R.Prabhakar

PG Scholar, Department of Computer Science and Engineering, Coimbatore Institute Of Technology, Coimbatore.
Professor, Department of Computer Science and Engineering, Coimbatore Institute of Technology, Coimbatore.

Abstract-*Graph matching is a fundamental problem that arises frequently in the areas of distributed control, computer vision, and facility allocation. In this paper, we consider the optimal graph matching problem for real time dataset like fish, house, and bird. Weighted graph matching algorithm is efficient for large number of dataset. The WGMP is the problem of finding the optimum matching between two weighted graphs, which are graphs with weights at each arc. The proposed method employs an analytic, instead of a combinatorial or iterative, approach to the optimum matching problem of such graphs. By using the eigendecompositions of the adjacency matrices (in the case of the undirected graph matching problem) or some Hermitian matrices derived from the adjacency matrices (in the case of the directed graph matching problem), a matching close to the optimum one can be found efficiently when the graphs are sufficiently close to each other. Simulation experiments are also given to evaluate the performance of the proposed method.*

Keywords---*Graph Matching, Weighted graph matching algorithm, modified frank-Wolfe, frank Wolfe algorithm.*

I. INTRODUCTION

Given two graphs with weights on edges, the weighted graph matching problem searches for an optimal permutation of nodes of one graph so that the difference between the edge weights is minimized. Graph matching problems arise frequently in computer vision, facility allocation problems, as well as distributed control.

In computer vision, matching structural descriptions of an object to those of a model is formulated as a graph matching problem [6], [7], [8].They treat weighted graphs with the same number of nodes and employ an analytic approach by using the Eigen-structure of adjacency matrices (undirected graph matching) or some Hermitian matrices derived from the adjacency matrices (directed graph matching). An almost optimal matching can be found when the graphs are sufficiently close to each other. In [7], [15], [19] the authors propose a Lagrangian Relaxation Network for the same problem. They formulate the permutation matrix constraints in the framework of deterministic annealing and achieve exact constraint satisfaction at each temperature within deterministic annealing. More recently, semi-definite programming relaxations for the quadratic assignment problem have been proposed in [9] and [10]. In particular, in [9] the authors propose a cutting planes algorithm that provides good solutions.

Since permutation matrices live in the intersection of the non-convex space of orthogonal matrices and the space of non-negative (element-wise) matrices, the above optimization-based approaches relax the non-convex orthogonality constraint. In this paper, we take the opposite approach, and relax the non-negativity constraint by defining dynamical systems that are, by construction, guaranteed to evolve on the manifold of orthogonal matrices. In particular, we construct two gradient flows, one that minimizes the cost of weighted graph matching over orthogonal matrices, and a second that minimizes the distance of an orthogonal matrix from the set of permutations[16],[11],[21].

The combination of the two dynamical systems converges to a permutation matrix, which provides a suboptimal solution to the weighted graph matching problem. In the spirit of analogsolutions to combinatorial problems, our approach is inspired by the so-called iso-spectral double-bracket dynamical system that sorts lists and solves various combinatorial problems [1], [2] (see also [3], [4], [5]). We illustrate our approach in examples involving more than 50 nodes, which are considered practically intractable, and also challenging for semi-definite relaxations using standardized optimization packages. This shows that our method is very promising. We also argue that, for applications where mobility is critical, such as distributed robotics, our approach is also more natural.

II. EXISTING METHOD

A. *Path following algorithm for graph matching*

Our approach is similar to graduated non-convexity. This approach is often used to approximate the global minimum of a non-convex objective function. This function consists of two parts, the convex component and non-convex component, and the graduated non-convexity framework proposes to track the linear combination of the convex and nonconvex parts (from the convex relaxation to the true objective function) to approximate the minimum of the non-convex function. The PATH algorithm may indeed be considered as an example of such an approach. However, the main difference is the construction of the objective function. Unlike, we construct two relaxations of the initial optimization problem, which lead to the same value on the set of interest (P), the goal being to choose convex/ concave relaxations that approximate in the best way the objective function on the set of permutation matrices.

B. *Pseudo Code For Path Following Algorithm*

Input: Kp,Kq,G1,H1,G2,H2,Θ,α0
Output: X

1. Initialize X to be a doubly stochastic matrix
2. Factorize Kq: UV^T
3. For α=0: Θ :1 do path-following
4. If α=0.5 & Jgm(X)<Jgm(X0) then
5. Update X-X0
6. Optimize the data using Modified Frank-wolfe algorithm to obtain X*
7. If Jgm(X*)< Jgm(X) then
8. Optimize data using Frank-wolf algorithm to obtain X*
9. Update X-X0
10. End

Given a pair of graphs, G1 = {P1, Q1, G1, H1} and G2 = {P2, Q2, G2, H2}.we compute two affinity matrices, Kp ∈ R n1×n2 and Kq ∈ R m1×m2 , to measure the similarity of each node and edge pair respectively.Given two graphs and the associated affinity matrices, the problem of GM consists in finding the optimal correspondence X between nodes, such that the sum of the node and edge compatibility is maximized:

$$J_{gm}(\mathbf{X}) = \mathrm{tr}\left(\mathbf{K}_p^T\mathbf{X}\right) + \mathrm{tr}\left(\mathbf{K}_q^T(\underbrace{\mathbf{G}_1^T\mathbf{X}\mathbf{G}_2 \circ \mathbf{H}_1^T\mathbf{X}\mathbf{H}_2}_{\mathbf{Y}})\right),$$

Where,
X=input data, Kp,Kq-Edge pair, G1,G2-Verticies
By using this equation and modified frank-Wolfe, frank Wolfe algorithm we find out optimal path between two graphs.

III. PROPOSED METHOD

A. Block Diagram For Proposed Method

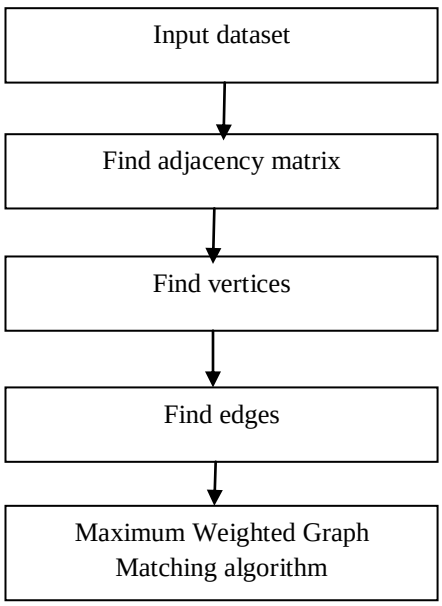

Fig.1 Process flow for proposed method

In our project, the dataset here we are taken as Fish, house, pot, jag and etc.., we will see the dataset description in below section.

B. Adjacency Matrix

An adjacency matrix is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph.In the special case of a finite simple graph, the adjacency matrix is a (0, 1)-matrix with zeros on its diagonal. If the graph is undirected, the adjacency matrix is symmetric [13],[17],[20]. The relationship between a graph and the eigenvalues and eigenvectors of its adjacency matrix is studied in spectral graph theory.

C. Vertices And Edges

A vertex (plural: vertices) is a point where two or more lines meet. An edge is a line segment that joins two vertices.

D. Maximum Weighted Graph Matching

A weighted graph G is an ordered pair(V,W), where V is a set of nodes, w is the weighting function which gives a real non-negative value. Let G and H be weighted undirected graphs and AG and A, be their adjacency matrices, respectively. The optimum matching between G and H is a permutation matrix P which minimizes J(P). It is, in general, difficult to find this matrix P directly. However, if we extend the domain of J to the set of orthogonal matrices, the orthogonal matrices (Q) which minimize J(Q) can be obtained in closed forms by using the Eigen decomposition of the adjacency matrices AG and AH. This extension of the domain is very natural because a permutation matrix is a kind of orthogonal matrix. A nearly optimum permutation matrix is determined by using these orthogonal matrices as clues. Here, we assume that both adjacency matrices AG and A, have n distinct eigenvalues, respectively. This is not a strong assumption for real data. Besides that, even if they have multiple roots, this can be overcome by perturbing them. Small perturbations will not effect the result of matching.

E. Pseudo Code For Weighted Graph Matching

Before looking into the pseudo code and the formal analysis of the algorithm, let us first gain some intuition on how the algorithm works. . Each process sitting on its corresponding node can see who the neighboring nodes are, and what are the costs (weights) of edges that lead to those nodes.

It can also communicate with all the neighboring nodes by sending and receiving messages.How can a node tell if some edge is locally heaviest? Well, suppose that for an edge e = (u, v) both u and v see this edge as the heaviest one of all the edges they can see. Then, e is heavier than all the edges incident to it, that is, locally heaviest edge.Now, if u and v choose each other as a matching pair, and notify all their remaining neighbors about it, their neighbors can stop looking at the edges incident to (u, v), which corresponds to removing these edges from the graph. Intuitively, by running this procedure on all the nodes over and over again, until all of them find their matching pair (or become lonely–i.e., see no neighbours).

F. Pseudo Code For Maximum Weighted Graph Matching

```
Greedy-Distributed (G, V)
        R=Ø
        N=T(U)
        C=candidate(V,N)
If C`=null
        Send <req> to C
While(N=Ø)
        Receive message from neighbor u
If m=<req>
        R=RÙ{u}
If m=<drop>
        N=N\{u},R=R\{u}
If u=c
        C=candidate (V, N)
If c~=null
        Send <req> to C
If c~=null^C€R
        Forall w€N\{C}
Send <drop> to w;N=Ø
End
```


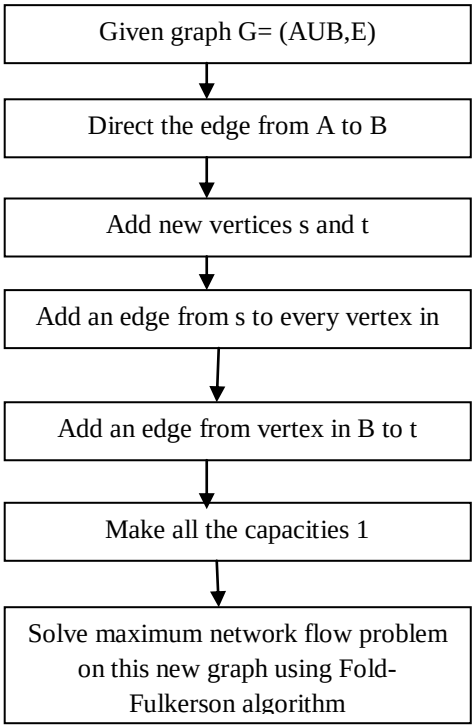

Fig.2 Flow diagram for maximum weighted graph matching

We should eventually get the same matching as we would for some run of the Greedy-Centralized algorithm.To formalize the statement of the algorithm described above, let us introduce some notation. We will denote by Γ(v) the set of neighbors of the node v in the original graph G. As every time a v's neighbor u gets matched it cannot be considered as a matching pair for v anymore, we want to drop an edge (u, v) from the graph when this happens, and observe only those neighbors that v can still match to. We will assume that when u gets matched to another node it sends a drop message to v (and all other non-matching neighbors), and denote the set of neighbors who are possible matches for v by N (v). Looking at N (v), v will choose the heaviest edge it can see and send a request to the neighbor adjacent to that edge. We will call this neighbor a candidate and denote it by c(v). If v receives a request message from its candidate neighbor, it knows that the edge (v, c(v)) is locally heaviest and chooses c(v) as its matching pair. We will allow every node to store the list R of the unmatched nodes it received a request message from, and update it every time a request or drop message arrives.

G. Dataset Description

This experiment performed a comparative evaluation of GM algorithms on randomly synthesized graphs. For each trial, we constructed two identical graphs, G1 and G2, each of which consists of 20 inliers nodes and later we added nout outlier nodes in both graphs. edge density parameter $\rho \in [0, 1]$. For each pair of nodes, the edge was randomly generated according to the edge density parameter $\rho \in [0, 1]$. . Each edge in the first graph was assigned a random edge score distributed uniformly as q 1 c ∼ U(0, 1) and the corresponding edge q 2 c = q 1 c + in the second graph was perturbed by adding a random Gaussian noise ∼ N (0, σ2). Notice that the edge feature was asymmetrical. The edge affinity Kq was computed as k q c1c2 = exp(−(q 1 c1−q 2 c2) 2/0.15) and the node affinity Kp was set to zero. The CMU house image sequence4 was commonly used to test the performance of graph matching algorithms. This dataset consists of 111 frames of a house, eachof which has been manually labeled with 30 landmarks. We used Delaunay triangulation to connect the landmarks. The edge weight qc was computed as the pair wise distance between the connected nodes. Due to the symmetry of distance-based feature, the edge was undirected. Given an image pair, the edge-affinity matrix Kq was computed by k q c1c2 = exp(−(q 1 c1 −q 2 c2) 2/2500) and the node-affinity Kp was set to zero. We tested the performance of all methods as a function of the separation between frames. We matched all possible image pairs, spaced by 0 : 10 : 90 frames and computed the average matching accuracy and objective ratio per sequence gap. We tested the performance of GM methods under two scenarios. In the first case we used all 30 nodes (i.e., landmarks) and in the second one we matched sub-graphs by randomly picking 25 landmarks. First of all, DEN achieved the worst performance, because it only relies on the graph topology to find the correspondence. Secondly, IPFP-S, RRWM, CAV, FGMU and FGM-D almost obtained perfect

matching of the original graphs in the first case. As some nodes became invisible and the graph got corrupted, the performance of all the methods degraded. The results demonstrate the advantages of the maximum weighted graph matching algorithm over other state of-the-art methods in solving general GM problems.

IV. RESULTS AND DISCUSSION

Project, the dataset here we are taken as Fish, house, pot, jag and described below the diagram

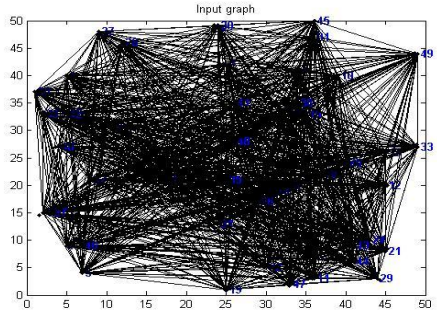

Fig.3 Input dataset

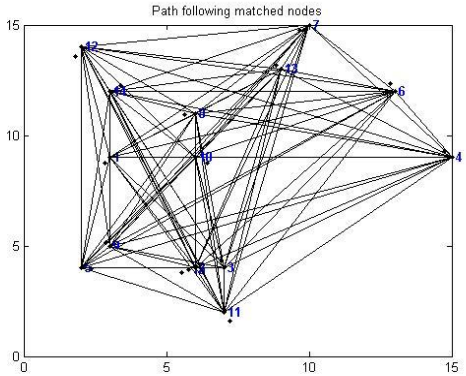

Fig.4 Path following algorithm

AMatrix is stored as a general graph, where rows and columns of the matrix represent vertices, and the nonzero elements represent edges.

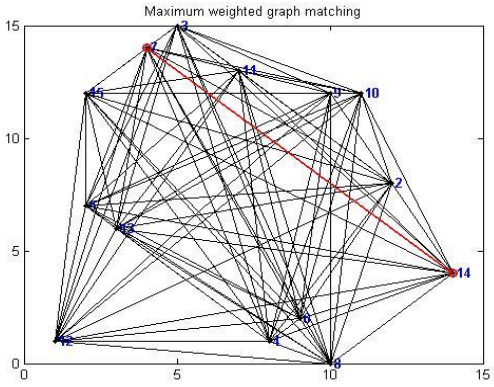

Fig.5 Maximum weighted graph matching

A d-dimensional random geometric graph (RGG), represented as G(n, r(n)), is a graph generated by randomly placing n vertices in a d-dimensional space and connecting pairs of vertices whose Euclidean distance is less than or equal to r(n).

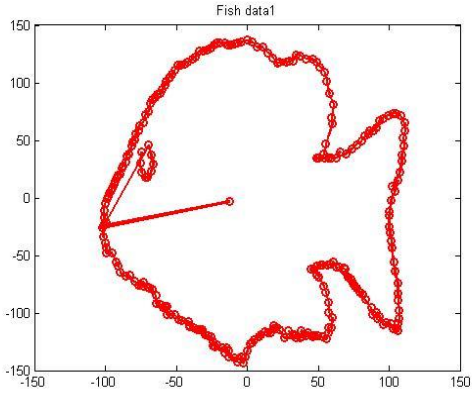

Fig.6 Fish data1

Fig6 and fig 7 represents the various fish data set snap on the proposed system algorithm

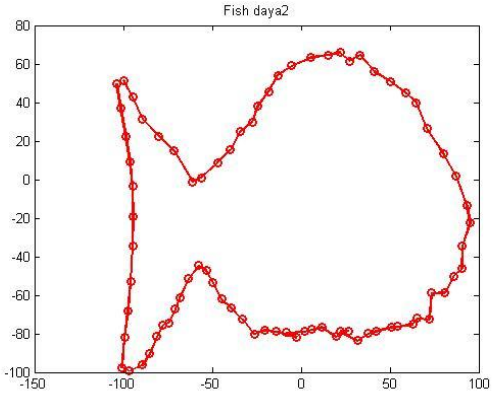

Fig.7 Fish data2

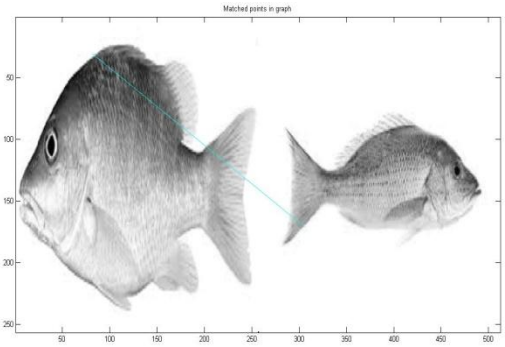

Fig.8 Matching using maximum weighted graph matching

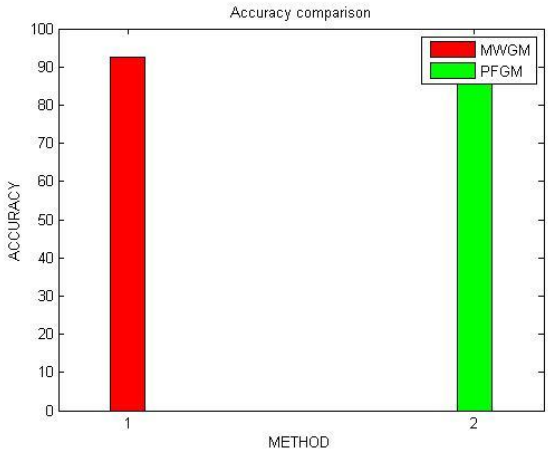

Fig.9 Accuracy comparison

Fig 9 presents the relative performance of different half approximation algorithms. The two main categories of approximation algorithms are the sorting-based algorithms.From the experimental results it can be observed that the proposed algorithm computes matching's of high quality at high speed.

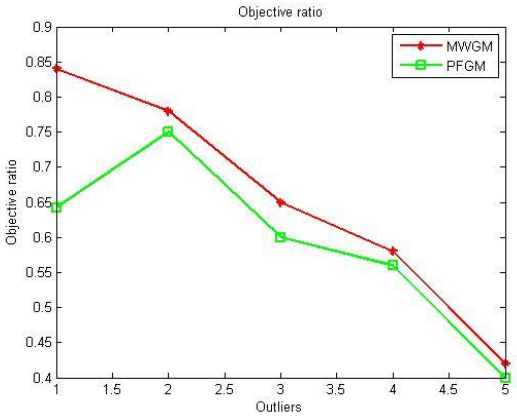

Fig.10 Objective ratio

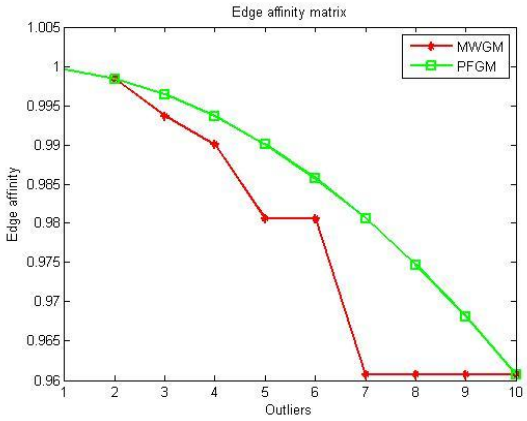

Fig.11 Edge affinity matrices

Fig 10 and 11 Compute time for different matrices with different number of processors. Compute time in seconds (log2 scale) is plotted on the Y-axis, and the number of processors is plotted on the X-axis. Max is the maximum time on any given processor in the set, and Avg is the average time for a given set of processors.

V. CONCLUSION

The approximate solution for the WGMP has been given in both the undirected and directed cases. The proposed method employs an analytic approach based on the Eigen decomposition of the adjacency matrix of a graph and al- most always gives the optimum matching when a pair of graphs is nearly isomorphic. If graphs are not sufficiently close to

each other, the proposed method sometimes fails to give the optimum matching. However, the hill-climbing method can improve the obtained matching even in this case since the obtained matching reflects the global correspondence between graphs. This method is resistant to the combinatorial explosion in execution time compared to the purely combinatorial approach. It also has the virtue of global optimization, unlike the local optimization method.

REFERENCES

1. H. Jiang, S. X. Yu, and D. R. Martin, "Linear scale and rotation invariant matching," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 7, pp. 1339–1355, 2011
2. K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in ICCV, 2005.
3. P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pp. 239–256, 1992.
4. D. Mateus, R. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer, "Articulated shape matching using Laplacianeigenfunctions and unsupervised point registration," in CVPR, 2008.
5. M. Leordeanu, M. Hebert, and R. Sukthankar, "Beyond local appearance: Category recognition from pairwise interactions of simple features," in CVPR, 2007
6. O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," in ICCV, 2011.
7. J. Hays, M. Leordeanu, A. A. Efros, and Y. Liu, "Discovering texture regularity as a higher-order correspondence problem," in ECCV, 2006.
8. W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in ICCV, 2011.
9. U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, "A "string of feature graphs model" for recognition of complex activities in natural videos," in ICCV, 2011.
10. N. Quadrianto, A. J. Smola, L. Song, and T. Tuytelaars, "Kernelized sorting," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 10, pp. 1809–1821, 2010.
11. F. Zhou and F. De la Torre, "Factorized graph matching," in CVPR, 2012.
12. D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," Int. J. Pattern Recognit. and Artificial Intelligence, vol. 18, no. 3, pp. 265–298, 2004
13. T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," Econometrica, vol. 25, no. 1, pp. 53–76, 1957.
14. Lu, S., Li, L., Lam, K.Y. and Jia, L., 2009, December. SAODV: A MANET routing protocol that can withstand black hole attack. In Computational Intelligence and Security, 2009. CIS'09. International Conference on (Vol. 2, pp. 421-425). IEEE.
15. Wan MohdFadzil W.M.N, Shah Rizam M.S.B, R. Jailani, Nooritawati M.T, "Orchid Leaf Disease Detection using BorderSegmentation Techniques" IEEE Conference on Systems, Process and Control,2014
16. S. S. Sannakki, V. S. Rajpurohit, V. B. Nargundand P. Kulkarni, "Diagnosis and Classification of Grape Leaf Diseases using Neural Networks", IEEE 4th ICCCNT, 2013
17. Vijay Satti, Anshul Satya and Shanu Sharma, "An Automatic Leaves Recognition System for Plant Identification Using Vision Technology", International Journal of Engineering Science and Technology (IJEST) ISSN:0975-5462, Vol 5, Issue 4, pp. 874-879, 2013.
18. H. G. Wang, G. L. Li, Z. H. Ma, and X. L. Li. "Application of (FNN) neural networks to image recognition of plant diseases", International Systems and Informatics, 2012
19. Yinmao Song, ZhihuaDiao, Yunpeng Wang,Huan Wang, "Image Feature Extraction of Crop Disease" IEEE Symposium on Electrical & Electronics Engineering,2012
20. P.Revathi, M.Hemalatha, "Classification of Cotton Leaf Spot Diseases UsingImage Processing Edge Detection Techniques" IEEE International Conference on Emerging Trends in Science, Engineering and Technology,2012
21. P.Revathi, M.Hemalatha, "Advance Computing Enrichment Evaluation ofCotton Leaf Spot Disease Detection using Image Edge detection" IEEE conference,2012

AUTHORS

R.Pavithra[1],PG Scholar, Department of Computer Science and Engineering,Coimbatore Institute Of Technology, Coimbatore has received B.E(Computer Science and Engineering),SreeSakthi Engineering College and pursuing M.E(Computer Science and Engineering) in Coimbatore Institute Of Technology.

Dr.R.Prabhakar[2],Professor, Department of Computer Science and Engineering, Coimbatore Institute Of Technology, Coimbatore has received B.Tech (Mechanical Engineering), IIT Madras,1969,the M.S (Mechanical Engineering),Okalahoma State University, USA,1970 and pursued Ph.D in Purdue University,USA,1975.

