Combinative hypergraph learning on oil spill training dataset

Binghui Wei[a,b], Ming Cheng[a*], Cheng Wang[a] and Jonathan Li[a,c]
[a]School of Information Science and Engineering, Xiamen University, Xiamen, 361005, China
[b]College of Applied Science, Jiangxi University of Science and Technology, Ganzhou, 341000, China
[c]Faculty of Environment, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Abstract

Detecting oil spill from open sea based on Synthetic Aperture Radar (SAR) image is a very important work. One of key issues is to distinguish oil spill from "look-alike". There are many existing methods to tackle this issue including supervised and semi-supervised learning. Recent years have witnessed a surge of interest in hypergraph-based transductive classification. This paper proposes combinative hypergraph learning (CHL) to distinguish oil spill from "look-alike". CHL captures the similarity between two samples in the same category by adding sparse hypergraph learning to conventional hypergraph learning. Experimental results have demonstrated the effectiveness of CHL in comparison to the state-of-the-art methods and showed that our proposed method is promising.

Keywords: oil spill detection, hypergraph learning, sparse representation

1. INTRODUCTION

Pollution resulting from oil spills in open sea and coastal waters is a major threat to ocean ecosystems. Detection and continuous monitoring of oil spills are important components of law enforcement efforts to minimize the impact that oil polluting events have on the ecosystem [1]. Previous studies have shown that Synthetic Aperture Radar (SAR) is effective in the detection and classification of oil spills. Oil spills appear as dark spots in SAR images [2]. However, similar dark spots ("look-alikes"), resulting in misidentification, may arise from a range of unrelated meteorological and oceanographic phenomena. To distinguish oil spills from "look-alikes" is definitely a key issue in oil spill detection. This process is the final stage of general oil spill detection and it is based on given dark spots feature.

In recent years, there are several classifiers used for distinguishing oil spills from "look-alikes" including: Bayesian classification scheme by combining prior knowledge [3-5], Linear discriminant analysis approach based on Mahalanobis distance [6], Multiple linear regression [7], Artificial neural

* chm99@xmu.edu.cn; phone 15959250541

2nd ISPRS International Conference on Computer Vision in Remote Sensing (CVRS 2015), edited by Cheng Wang, Rongrong Ji, Chenglu Wen, Proc. of SPIE Vol. 9901, 99010Z · © 2016 SPIE · CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2234854

network approach [8,9] and Support vector machine [10]. Especially, Xu et al. [11] compared and analyzed the performance of seven classifiers - bagging, bundling, boosting, Penalized linear discriminant analysis, Generalized additive model, Support vector machine and Artificial neural network.

Currently, there is widespread interest in the development of transductive learning. Because transductive learning explores not only labelled data but also unlabelled data, and it achieves better performance than methods that learn classifiers based only on labelled data. Its success is based on one of the following two assumptions: cluster and manifold assumptions. The cluster assumption supposes that the decision boundary should not cross high-density regions, whereas the manifold assumption means that each
class lies on an independent manifold. Graph-based learning is one of the transductive learning methods. The graph-based learning [12-19] achieves promising performance among existing transductive learning methods. Its development goes through two stages: simple-graph learning and hypergraph learning. This type of learning is built on a graph, in which vertices are samples and edge weights indicate the similarity between two samples. However, the simple-graph learning methods consider only the pairwise relationship between two samples, and it ignores the relationship in a higher order. Hypergraph learning aims to get the relationship among several samples in a higher order. Unlike a simple graph that has an edge between two vertices, a set of vertices is connected by a hyperedge in a hypergraph, and each hyperedge is assigned a weight.

However, these hypergraph-based methods focus only on proximity relation for distance. There is some other high-order relationship, such as linear relationship. Wei et al. proposed combinative hypergraph learning by taking into account the clustering assumption that the similar points in feature space are more likely belong to the same category [19]. This algorithm defines the similarity by two assumptions - distance-similarity and linear-similarity. The distance-similarity is that the points derived from a category are located close to each other. The linear-similarity is that a data point can more likely to be represented linearly by the data points nearby who are belong to the same category with this data point. Inspired by the two assumptions, the following two kinds of hypergraph was constructed on a data set, one is conventional hypergraph which is based on the distance-similarity, the other one is sparse hypergraph which is based on linear-similarity and derived from the thinking in [20-21]. We use this algorithm to learning oil spill from “look-alike” and the experiments results show that it is effective.

The rest of this paper is organized as follows. Section II describes the combinative hypergraph learning. And section III shows experiments on practical image datasets. We draw the conclusions in section IV.

2. HYPERGRAPH LEARNING

This section introduces conventional hypergraph learning and sparse hypergraph learning respectively firstly, and shows combinative hypergraph learning theory in the following.

2.1 Conventional hypergraph learning

Given c categories of images including m training data points ($\boldsymbol{x}_1$, $\boldsymbol{y}_1$), …, ($\boldsymbol{x}_m$, $\boldsymbol{y}_m$), and n testing data points ($\boldsymbol{x}_{m+1}$, $\mathbf{0}$), …, ($\boldsymbol{x}_{m+n}$, $\mathbf{0}$), where $\boldsymbol{x}_i \in \mathrm{R}^d, 1 \le i \le m+n$ is sampled from the input space; $\boldsymbol{y}_i = [0,\cdots,1,\cdots,0]^{\mathrm{T}} \in \mathrm{R}^c, 1 \le i \le m$, is the label vector of $\boldsymbol{x}_i$, where the g-th component is 1 if $\boldsymbol{x}_i$ belongs to the g-th category, otherwise, 0; and $\mathbf{0}$ is a vector with c components of zero.

The hypergraph, $\mathbf{H}=(\mathbf{x}, \boldsymbol{\varepsilon})$, is formed by the vertex set, $\mathbf{x}$, and the hyperedge set, $\boldsymbol{\varepsilon}$, An incidence matrix, $\mathbf{A}$, whose size is $|\mathbf{x}|\times|\boldsymbol{\varepsilon}|$, denotes the hypergraph with the following elements:

$$\mathbf{A}(i,j) = \begin{cases} 1, & \text{if } \boldsymbol{x}_i \in \boldsymbol{\varepsilon}_j \\ 0, & \text{if } \boldsymbol{x}_i \notin \boldsymbol{\varepsilon}_j \end{cases}, \tag{1}$$

where $\boldsymbol{\varepsilon}_j$ denotes the j-th element of the hyperedge set . The distance between two data points is

$$\begin{aligned} &\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2}{\sigma^2}\right) \\ &\sigma = \sqrt{\frac{1}{m+n-1}\sum_{\delta=1}^{m+n}\left\|\boldsymbol{x}_\delta - \bar{\boldsymbol{x}}\right\|^2} \qquad \bar{\boldsymbol{x}} = \frac{1}{m+n}\sum_{\delta=1}^{m+n}\boldsymbol{x}_\delta \end{aligned} . \tag{2}$$

$\boldsymbol{\varphi}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ denote diagonal matrices of vertex degrees, hyperedge degrees and hyperedge weights, respectively. $\boldsymbol{\varphi}_i$ denotes the entry (i, i) of matrix $\boldsymbol{\varphi}$, $\boldsymbol{\omega}_i$ and $\boldsymbol{\gamma}_i$ have similar meanings. Then, the initial weight, $\boldsymbol{\omega}_i$, is

$$\boldsymbol{\omega}_i = \sum_{\boldsymbol{x}_j \in \boldsymbol{\varepsilon}_i} \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{3}$$

Based on $\mathbf{A}$, the i-th vertex degree, $\boldsymbol{\varphi}_i$, is

$$\boldsymbol{\varphi}_i = \sum_{\boldsymbol{\varepsilon}_j \in \boldsymbol{\varepsilon}} \boldsymbol{\omega}_j \mathbf{A}(i,j), \tag{4}$$

and the i-th hyperedge degree, $\boldsymbol{\gamma}_i$, is

$$\boldsymbol{\gamma}_i = \sum_{\boldsymbol{x}_j \in \boldsymbol{x}} \mathbf{A}(j,i). \tag{5}$$

In this paper, we adopt the regularization framework proposed in [17], i.e.

$$\begin{aligned} \arg \min_{\mathbf{F},\,\boldsymbol{\omega}} & \sum_{\eta=1}^{c}(\boldsymbol{f}_\eta^{\mathrm{T}}\mathbf{L}\boldsymbol{f}_\eta) + \lambda\left\|\boldsymbol{f}_\eta - \mathbf{Y}_\eta\right\|^2 + \mu\left\|\mathrm{diag}(\boldsymbol{\omega})\right\|^2 \\ & \text{s.t.} \sum_{j=1}^{l}\boldsymbol{\omega}_j = 1, \quad \boldsymbol{\omega}_j \ge 0, \ j=1, \cdots, l \end{aligned}, \tag{6}$$

where $\mathbf{L} = \mathbf{I} - \boldsymbol{\varphi}^{-(1/2)}\mathbf{A}\,\boldsymbol{\gamma}^{-1}\boldsymbol{\omega}\mathbf{A}^{\mathrm{T}}\boldsymbol{\varphi}^{-(1/2)}$; l is the number of hyperedges; diag($\boldsymbol{\omega}$) is the diagonal vector of $\boldsymbol{\omega}$, i.e., ($\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, …, $\boldsymbol{\omega}_l$); c is the number of classes; $\mathbf{F}$ is a matrix, $\mathbf{F} = (\boldsymbol{f}_1, \cdots, \boldsymbol{f}_c) \in \Re^{(m+n)\times c}$, $\boldsymbol{f}_\eta$ is the confidence of the labelling for the data belonging to the

ηth category; λ>0 and μ>0 are two trade-off parameters to balance the empirical loss, the weight and the regularization; $\mathbf{Y}=(\boldsymbol{y}_1, \boldsymbol{y}_2,\ldots, \boldsymbol{y}_m, \mathbf{0},\ldots, \mathbf{0})^{\mathrm{T}}$, $\mathbf{Y}\in\Re^{(m+n)\times c}$, $\mathbf{Y}_\eta$ is the ηth column of $\mathbf{Y}$. The third term is introduced to avoid a degenerate solution caused by the former two terms existing only in the regularization. Here we add two constraints, one to fix the summation of the weights $\sum_{j=1}^{l}\boldsymbol{\omega}_j=1$ and one to avoid negative weight $\boldsymbol{\omega}_j\geq 0$.

Because the classifier function is not jointly convex with respect to **F** and $\boldsymbol{\omega}$, we solve one variable by fixing another variable.

First we initialize $\boldsymbol{\omega}$ with (3), so the solution of **F** becomes

$$\mathbf{F}=\frac{\lambda}{\lambda+1}\left(\mathbf{I}-\frac{\boldsymbol{\varphi}^{-(1/2)}\mathbf{A}\boldsymbol{\gamma}^{-1}\boldsymbol{\omega}\mathbf{A}^{\mathrm{T}}\boldsymbol{\varphi}^{-(1/2)}}{1+\lambda}\right)^{-1}\mathbf{Y}. \tag{7}$$

Then we update the weights, $\boldsymbol{\omega}$, with an iterative coordinate descent method. Based on the coordinate descent method, an iterative process alternately updates the labels and the weights.

In the next iteration, we calculate the new **F** with new $\boldsymbol{\omega}$. The iteration ceases at a given state. After obtaining **F**, we set the *g*-th class to the *i*-th data point if the *g*-th component is the maximum in the *i*-th row of **F**. A more detailed solution of (6) appears in [19].

2.2 Sparse hypergraph learning

Sparse hypergraph learning is similar to conventional hypergraph learning. Therefore, this section will show only the difference between sparse hypergraph learning and spectral hypergraph learning to avoid repetition. It is the hyperedge construction and the hyperedge weight definition which will be introduced after the introduction of sparse representation in this section.

This paper solves the sparse representation problem based on ℓ_1 minimization.

Given a vector $\boldsymbol{x}$ in R^d, which can be represented in a basis of d vectors $\{\boldsymbol{\varsigma}_i\in\mathrm{R}^n\}_{i=1}^{d}$, where include . Set a matrix $\boldsymbol{\Sigma}=[\boldsymbol{\varsigma}_1,\boldsymbol{\varsigma}_2,\cdots,\boldsymbol{\varsigma}_d]$ and we can rewrite $\boldsymbol{x}$ as

$$\boldsymbol{x}=\sum_{i=1}^{d}w_i\boldsymbol{\varsigma}_i=\boldsymbol{\Sigma}\boldsymbol{w} \tag{8}$$

Where $\boldsymbol{w}=[w_1,w_2,\cdots,w_d]^{\mathrm{T}}$. Both $\boldsymbol{x}$ and $\boldsymbol{w}$ represent the same data point, one in the space domain and the other in the $\sum$ domain. Our object is to find a sparse representation of $\boldsymbol{x}$ in a properly chosen basis $\sum$, namely, $\boldsymbol{w}$ must have as few nonzero components as possible. According to [22], such a sparse representation can be obtained by solving the optimization problem

$$\min \|\boldsymbol{w}\|_0 \quad \text{subject to } \boldsymbol{x} = \boldsymbol{\Sigma}\boldsymbol{w}, \tag{9}$$

where $\|\boldsymbol{w}\|_0$ is the ℓ_0 norm of $\boldsymbol{w}$, *i.e.* the number of nonzero entries. However, such an optimization problem is in general non-convex and NP-hard. According to [20, 21], we can replace the non-convex optimization in (9) by the following convex ℓ_1 optimization formulation,

$$\min \|\boldsymbol{w}\|_1 \quad \text{subject to } \boldsymbol{x} = \boldsymbol{\Sigma}\boldsymbol{w}. \tag{10}$$

Now, consider all data points in a dataset, $\mathbf{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]$ Each data point, $\boldsymbol{x}_i$, has a sparse representation $\boldsymbol{w}_i$. By setting the i-th element of $\boldsymbol{w}_i$, $(\boldsymbol{w}_i)_i$=0, the optimization formulation can be rewrote as

$$\min \sum_{i=1}^{n} \|\boldsymbol{w}_i\|_1 \quad \text{subject to } \boldsymbol{x}_i = \mathbf{x}^{\mathrm{T}} \boldsymbol{w}_i \ . \tag{11}$$

Assume that, the data set $\mathbf{x}$ is drawn from a union of c independent linear subspaces, namely that the data set include c categories of object. According to [21], we can obtain block sparse solutions with the nonzero block corresponding to points in the same subspace if the aforementioned assumption holds. We can recover a block sparse representation of a new data point as a linear combination of the points in the same subspace. This means that a data point $\boldsymbol{x}_i$ and the data points which index is corresponding to nonzero entry of $\boldsymbol{w}_i$ are derived from the same category.

We now show how to define a hyperedge and its weight which are different from spectral hypergraph construction. For discrimination, we use $\mathbf{A}^{\mathrm{sp}}$, $\mathrm{dist}^{\mathrm{sp}}$ and $\mathbf{F}^{\mathrm{sp}}$ to replace incidence matrix $\mathbf{A}$, distance between two data points dist and classifier function $\mathbf{F}$ showed on above section respectively. We define $\mathbf{A}^{\mathrm{sp}}$ as

$$\mathbf{A}^{\mathrm{sp}}(i,j) = \begin{cases} 1, & \text{if } \ i = j \\ 1, & \text{if } \ |(\boldsymbol{w}_i)_j| > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{12}$$

where $|(\boldsymbol{w}_i)_j|$ denotes the absolute of j-th entry of $\boldsymbol{w}_i$, and $\mathrm{dist}^{\mathrm{sp}}$ as

$$\mathrm{dist}^{\mathrm{sp}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = |(\boldsymbol{w}_i)_j|. \tag{13}$$

The following processing is the same as conventional hypergraph learning. Here, we take $\mathbf{F}^{\mathrm{sp}}$ as the solution of the sparse hypergraph learning.

2.3 Combinative hypergraph learning

We obtain two confidences of the labelling, $\mathbf{F}$ and $\mathbf{F}^{sp}$, for conventional hypergraph learning and sparse hypergraph learning aforementioned. We define $\mathbf{F}^*$ as

$$\begin{aligned} &\mathbf{F}^* = \alpha^* \mathbf{F} + (1-\alpha^*)\mathbf{F}^{sp} \\ &\alpha^* = \arg\min_{\alpha} \sum_{i=1}^{m} \left\| \alpha \mathbf{F}_i + (1-\alpha)\mathbf{F}_i^{sp} - \boldsymbol{y}_i \right\|_2^2 \cdot \\ &\text{subject to } \alpha \in [0,1] \end{aligned} \tag{14}$$

It means that, here, we combine linearly only the results of the two hypergraph learning with the weights a^* and 1-a^*. We can get a^* by minimizing the loss of classification contribution value to truth classification value with only the labelled data point. The minimized function is easy to be solved since there is only one variable. After obtaining $\mathbf{F}^*$, we set the g-th class to the i-th data point if the g-th component is the maximum in the i-th row of $\mathbf{F}^*$.

For this strategy, we learn the confidence of the labelling independent for the two hypergraph-based learning. Then, we combine only the two confidence of the labelling linearly with optimal weights.

3 EXPERIMENTS

We performed experiments for oil spill dataset, we compared our method with general classification algorithms including k-nearest neighbor (k-NN), decision tree (C4.5); naïve bayes (NB); neural network (NN); support vector machine (SVM); AdaBoost; transductive support vector machine (TSVM) [12]; hypergraph Learning (HL) [17]; sparse Hypergraph Learning (SHL) [20] and semi-supervised discriminant analysis (SDA) [22] classifications.

3.1 Datasets and configurations

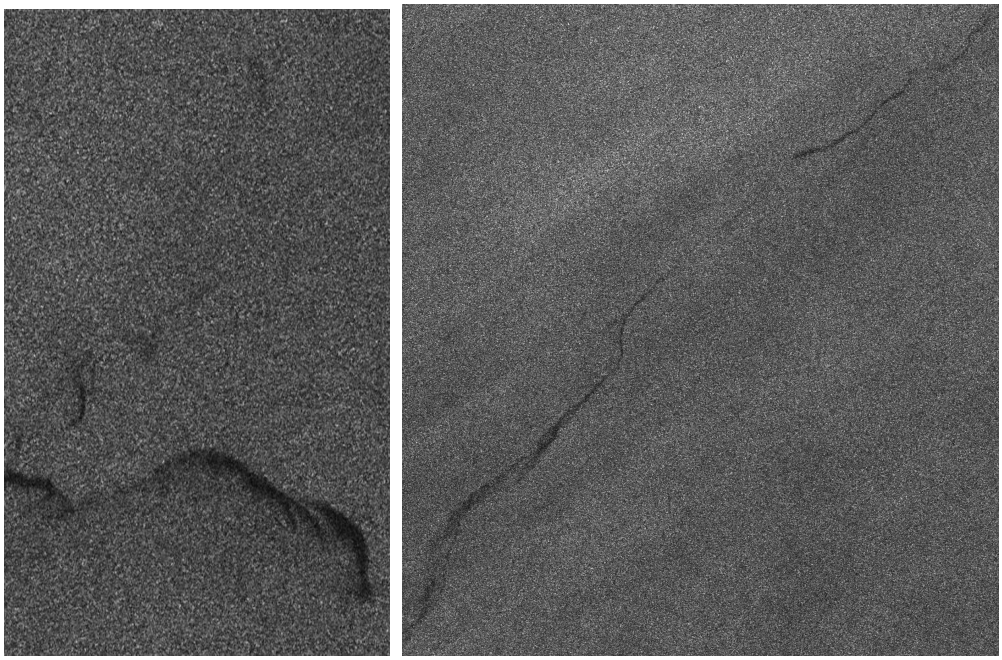
The dataset used in this study is derived from RADARSAT-1 ScanSAR Narrow Beam images with a swath of 300 km and a spatial resolution of 50 m, and covers vast Pacific and Atlantic coastal areas [11]. The dataset used comprises fourteen features of 412 oil spills and "look-alikes". There are thirty oil spills and 382 "look-alikes." Fig. 1 shows examples of oil spill and "look-alike". By visually discerning the gray tone difference between the dark-spots and the background, we delineated dark-spot boundaries; therefore, we need not introduce dark-spot detection in this letter.

Given the dark-spots in pixel-format, features must be extracted as input for the classifiers. For each of the dark spots, we compute three kinds of features: Physical and textural features, Geometric features and Contrast with background. More details about the data and features of dark-spot refer to [11].

To evaluate classifiers on the imbalanced data sets, we use a True Positive Rate (TPR), a True Negative Rate (TNR), and overall accuracy. Here, positive denotes oil spill and negative denotes "look-alike". TPR, TNR and Accuracy metric are defined as follows:

$$\text{TPR}=\frac{\text{TP}}{\text{TP+FP}}\quad \text{TNR}=\frac{\text{TN}}{\text{TN+FN}}\quad \text{Accuracy}=\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}, \tag{15}$$

where P, N, T and F denote Positive, Negative, True and False, respectively.



（a） (b)

Figure 1. The dark spots are the examples of (a) “look-alike” (acquired on Sep. 17, 2013, located at 50°43′N, 53°25′W), and (b) oil spill (acquired on Jul. 23, 2009, located at 42°50′N, 56°38′W) .

Intuitively, TPR and TNR are measure of exactness for actually labeled correctly. The F measure metric combines exactness and completeness as a measure of classification effectiveness. We use the TPR and TNR metrics to illustrate intuitively each method in dealing with an imbalanced training set. We use the accuracy to evaluate the performance of each classification method. For all the classification methods, we independently repeat the experiments twenty times with randomly selected training samples and show the averaged results by mean Accuracy.

We carry out the following classifications: (1) k-NN (k=10); (2) SVM with the library, Libsvm, written by Chih-Chung Chang and Chih-Jen Lin; (3) NB; (4) NN with the code in Matlab7.14 where parameters are set optimally; (5) C4.5; (6) AdaBoost; (7) Semi-supervised Discriminated Analysis (k=5); (8) Transductive Support Vector Machine (RBF kernel, C-SVC, C = 20); (9) Conventional Hypergraph learning (k= 5); (10) Sparse hypergraph learning (k=5); (11) Combinative Hypergraph Learning(initialize a to 0.01, iteration step by 0.01).

3.2 Experimental results

Table 1. Performance of classifiers

Number	Methods	TPR	TNR	Accuracy
1	KNN	0.16	0.72	0.54
2	SVM	0.05	0.92	0.64
3	NB	0.29	0.65	0.53
4	NN	0.18	0.67	0.52
5	C4.5	0.27	0.61	0.50
6	AdaBoosting	0.12	0.76	0.56
7	SDA	0.10	0.71	0.51
8	TSVM	0.28	0.69	0.64
9	HL	0.23	0.66	0.58
10	SHL	0.05	0.67	0.62
11	CHL	0.24	0.70	0.65

Table 1 shows the TPR, TNR and overall accuracy with all methods aforementioned. In term with TPR and TNR, we can see that the oil spill and "look-alike" are so similar to each other that they are very lowly. And the overall accuracy is lowly too. Of course, the experimental results also show that CHL achieves best overall accuracy in all methods. And the results demonstrate the effectiveness of the proposed method. Both SVM and TSVM show a somewhat poorer performance, but, they are very close to that of CHL.

4 CONCLUSIONS

In this paper, we proposed combinative hypergraphlearning to distinguish oil spill from "look-alike". Compared with many existing supervised and semi-supervised learning methods, the proposed method achieved the best performance with respect to overall accuracy. However, due to the limitation of the oil spill training data, we didn't conduct more experiments to further identify the advantages of our proposed framework.

REFERENCES

[1] M. Kubat, R. C. Holte & S. Matwin, (1998). "Machine learning for the detection of oil spills in satellite radar images". *Machine Learning*, 30, 195-215.

[2] C. Brekke & A. H. S. Solberg, (2005). "Oil spill detection by satellite remote sensing". *Remote Sensing of Environment*, 1, 1-13.

[3] Solberg, A. H. S., Storvik, G., Solberg, R., & Volden, E. (1999). Automatic detection of oil spills in ERS SAR images. IEEE Transactions on Geoscience and Remote Sensing, 4, 1916-1924.

[4] Solberg, A. H. S., Dokken, S. T., & Solberg, R. (2003). Automatic detection of oil spills in Envisat, Radarsat and ERS SAR images. Proceedings of the International Geoscience and Remote Sensing Symposium, 4. (pp. 2747-2749).

[5] Solberg, A., Brekke, C., Volden, E., & Husøy, P. (2007). Oil spill detection in RADARSAT and ENVISAT SAR images. IEEE Transactions on Geoscience and Remote Sensing, 45, 746-755.

[6] Fiscella, B., Giancaspro, A., Nirchio, F., Pavese, P., & Trivero, P. (2000). Oil spill detection using marine SAR images. International Journal of Remote Sensing, 18, 3561-3566.

[7] Nirchio, F., Sorgente, M., Giancaspro, A., Biamino, W., Parisato, E., Raverar, et al. (2005). Automatic detection of oil spills from SAR images. International Journal of Remote Sensing, 6, 1157-1174.

[8] Frate, F. D., Petrocchi, A., Lichtenegger, J., & Calabresi, G. (2000). Neural networks for oil spill detection using ERS-SAR data. IEEE Transactions on Geoscience and Remote Sensing, 38(5), 2282-2287.

[9] Topouzelis, K., Karathanassi, V., Pavlakis, P., & Rokos, D. (2007). Detection and discrimination between oil spills and look-alike phenomena through neural networks. ISPRS Journal of Photogrammetry and Remote Sensing, 4, 264-270.

[10] Brekke, C., & Solberg, A. H. S. (2008). Classifiers and confidence estimation for oil spill detection in ENVISAT ASAR images. IEEE Geoscience and Remote Sensing Letters, 1, 65–69.

[11] L. Xu, J. Li, A. Brenning. (2014) A comparative study of different classification techniques for marine oil spill identification using RADARSAT-1 imagery, Remote Sensing of Environment, 141, 14-23.

[12] J. Yu, M. Wang, and D. Tao, "Semi-supervised Multiview Distance Metric Learning for Cartoon Synthesis," *IEEE Transactions on Image Processing*, 21(11): 4636-4648, 2012.

[13] M. Wang, X. Hua, J. Tang, R. Hong. "Beyond Distance Measurement: Constructing Neighborhood Similarity for Video Annotation," *IEEE Transactions on Multimedia*, 11(3) pp. 465-476, 2009.

[14] M. Wang, X. Hua, R. Hong, J. Tang, Qi, G.-J, Y. Song, "Unified Video Annotation via Multi-Graph Learning," *IEEE Transactions on Circuits and Systems for Video Technology* 19(5) pp. 733-746, 2009.

[15] M. Wang, H. Li, D. Tao, K. Lu, X. Wu, "Multimodal Graph-Based Reranking for Web Image Search," *IEEE Transactions on Image Processing*, 21(11) pp.4649-4661, 2012.

[16] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, "3D Object Retrieval and Recognition with Hypergraph Analysis," *IEEE Transactions on Image Processing*, 21(9) pp.4290-4303, 2012.

[17] J. YU, D. TAO and M. WANG, Adaptive Hypergraph Learning and its Application in Image Classification, *IEEE Transactions on Image Processing*, 21, 2012, pp. 3262-3272.

[18] Y. Huang, Q. Liu, F. Lv, Y. Gong, and D. Metaxas, "Unsupervised image categorization by hypergraph partition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1266-1273, Jun. 2011.

[19] B. Wei, M. Cheng, C. Wang and J. Li, Combinative hypergraph learning for semi-supervised image classification, *Neurocomputing*, 2014, In press.

[20] E. Elhamifar, R. Vidal, Sparse subspace clustering, *IEEE-Computer-Society Conference on Computer Vision and Pattern Recognition*, vol.1-4, pp.2782-2789, Miami Beach, FL, Jun.20-25, 2009.

[21] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol.35, no.11, pp. 2765-2781, Nov. 2013.

[22] Emmanuel J. Candès. The restricted isometry property and its implications for compressed sensing. *Theory of Signals/Mathematical Analysis*, Ser. I 346 (2008) 589-592.

[23] B. Wei, J. Yu, C. Wang and J. Li. PolSAR image classification using a semi-supervised classifier based on hypergraph learning. *Remote sensing letters*, vol.5, pp.386-395, Mar.2014.