A Semi-supervised Approach for Industrial Workflow Recognition

Eftychios E. Protopapadakis, Anastasios D. Doulamis,
Konstantinos Makantasis
Computer Vision and Decision Support Lab.
Technical University of Crete
Chania, Greece
eft.protopapadakis@gmail.com
adoulam@ergasya.tuc.gr
konst.makantasis@gmail.com

Athanasios S. Voulodimos
Distributed Knowledge and Media Systems Group
National Technical University of Athens
Athens, Greece
thanosv@mail.ntua.gr

***Abstract*—In this paper, we propose a neural network based scheme for performing semi-supervised job classification, based on video data taken from Nissan factory. The procedure is based on (a) a nonlinear classifier, formed using an island genetic algorithm, (b) a similarity-based classifier, and (c) a decision mechanism that utilizes the classifiers' outputs in a semi-supervised way, minimizing the expert's interventions. Such methodology will support the visual supervision of industrial environments by providing essential information to the supervisors and supporting their job.**

Keywords-semi-supervised learning; activity recognition; pattern classification; industrial environments.

I. INTRODUCTION

Visual supervision is an important task within complex industrial environments; it has to provide a quick and precise detection of the production and assembly processes. When it comes to smart monitoring of large-scale enterprises or factories, the importance of behavior recognition relates to the safety and security of the staff, to the reduction of bad quality products cost, to production scheduling, as well as, to the quality of the production process.

In most current approaches, the goal is either to detect activities, which may deviate from the norm, or to classify some isolated activities [1],[2]. Modern techniques are based on supervised training using large data sets. The need of a significant amount of labeled data during the training phase makes classifiers data expensive. In addition, that data demands an expert's knowledge that increases further the cost.

Modern industry is based on the flexibility of the production lines. Therefore, changes occur constantly. These changes call for appropriate modifications to the supervising systems. A considerable amount of new training paradigms is required in order to adjust the system [3] at the new environment. In order to provide all the training data an expert, whose services will not be at a low-cost, is needed.

A variety of methods has been used for event detection and especially human action recognition, including semi-latent topic models [4], spatial-temporal context [5], optical flow and kinematic features [6], and random trees and Hough transform voting [7]. Comprehensive literature reviews regarding isolated human action recognition can be found in [8],[9].

The idea of this paper is the creation of a decision support mechanism for the workflow surveillance in an assembly line that would use few training data, initially; as time passes could be self-trained or, if it is necessary, ask for an expert assistance. That way, the human knowledge is incorporated at the minimum possible cost.

The innovation can be summarized to the following sentence: *We propose a cognitive system which is able to survey complex, non-stationary industrial processes by utilizing only a small number of training data and using a self-improvement technique through time.*

This paper is organized as follows: Section 2 provides a brief description of the proposed methodology. Section 3 refers to the data extraction methodology. Section 4 describes the genetic algorithm application. Section 5 presents the main classifier for the system. Section 6 presents the semi-supervised approach. Section 7 explains the decision mechanism of the system, and Section 8 provides the experimental results.

II. THE PROPOSED SELF COGNITIVE VISUAL SURVEILLANCE SYSTEM

The proposed system was tested using the NISSAN video dataset [10], which refers to a real-life industrial process videos regarding car parts assembly. Seven different, time-repetitive, workflows have been identified, exploiting knowledge from industrial engineers. Challenging visual effects are encountered, such as background clutter/motion, severe occlusions, and illumination fluctuations.

The presented approach employs an innovative self-improvable cognitive system, which is based on a semi-supervised learning strategy as follows: Initially, appropriate visual features are extracted using various techniques (Section 3). Then, visual histograms are formed, from these features, to address temporal variations in executing different instances of the same industrial workflow. The created histograms are fed as inputs to a non-linear classifier.

The heart of the system is the automatic self-improvable methodology of the classifier. In particular, we start feeding the classifier with a small but sufficient number of training samples (labeled data). Then, the classifier is tested on new incoming unlabeled data. If specific criteria are met, the classifier automatically selects suitable data from the set of the unlabeled data for further training. The criteria are set so

 ISBN: 978-1-61208-226-4

that only the most confident unlabeled data will be used on the new training set.

If a vague output occurs, for any of the new incoming unlabeled data, a second classifier, which exploits similarity measure among the in-sampled and the unlabeled data, is used. If classifiers disagree, an expert is called to interweave at the system to improve the classifier accuracy. The intervention is performed, in our case with a totally transparent and hidden way without imposing the user to acquire specific knowledge of the system and the classifier.

III. VISUAL REPRESENTATION OF INDUSTRIAL CONTENT

From all videos, holistic features such as Pixel Change History (PCH) are used. These features remedy the drawbacks of local features, while also requiring a much less tedious computational procedure for their extraction [11]. A very positive attribute of such representations is that they can easily capture the history of a task that is being executed. These images can then transformed to a vector-based representation using the Zernike moments (up to sixth order, in our case) as it was applied at [12].

The video features, once exported, had a 2 dimensional matrix representation of the form $m \times l$, where m denotes the size of the $1 \times m$ vectors created using Zernike moments, and l the number of such vectors. Although m was constant, l varies according to the video duration. In order to create constant size histogram vectors, which would be the system's inputs, the following steps took place:

1. The hyperbolic tangent sigmoid transformation was applied to every video feature. As a result the prices of the 2-d matrices range from -1 to 1.
2. Histogram vectors of 33 classes were created. The number of classes was defined after various simulations. Higher number of classes leads to poor performance due to the small training sample (in our case 48 vectors). Fewer classes also caused poor performance probably due to loss of important information from the original features. Each class counts the frequency of the appearance of a value (within a specific range) for a particular video feature.
3. Finally, each histogram vector value is normalized. Thus, the input vectors were created.

It is clear that each histogram vector describes a specific job among seven different. These histograms, one at a time, are the inputs for a feed forward neural network (FFNN). The target vectors are seven-element arrays. The value at each array will be either one or zero. The number one denotes in which category is categorized the video (e.g., 0 0 0 1 0 0 0 correspond to assembly procedure number four).

IV. THE ISLAND GENETIC ALGORITHM

The usefulness of the genetic algorithms (GAs) is generally accepted [13]. The island GA uses a population of alternative individuals in each of the islands. Every individual is a FFNN. While eras pass networks' parameters are combined in various ways in order to achieve a suitable topology.

A pair of FFNNs (parents) is combined in order to create two new FFNNs (children). Children inherit randomly their topology characteristics from both their parents. Under specific circumstances, every one of these characteristics may change (mutation). The quartet, parents and children, are then evaluated and the two best will remain, updating that way the island's population. An era has passed when all the population members participate in the above procedure. In order to bate the genetic drift, population exchange among the islands, every four eras. The algorithm terminates when all eras have passed. Initially, the parameters' range is described in Table 1 and the main steps of the genetic algorithm are shown in Figure 1. The algorithm is used to parameterize the topology of the non-linear classifier (Section 5).

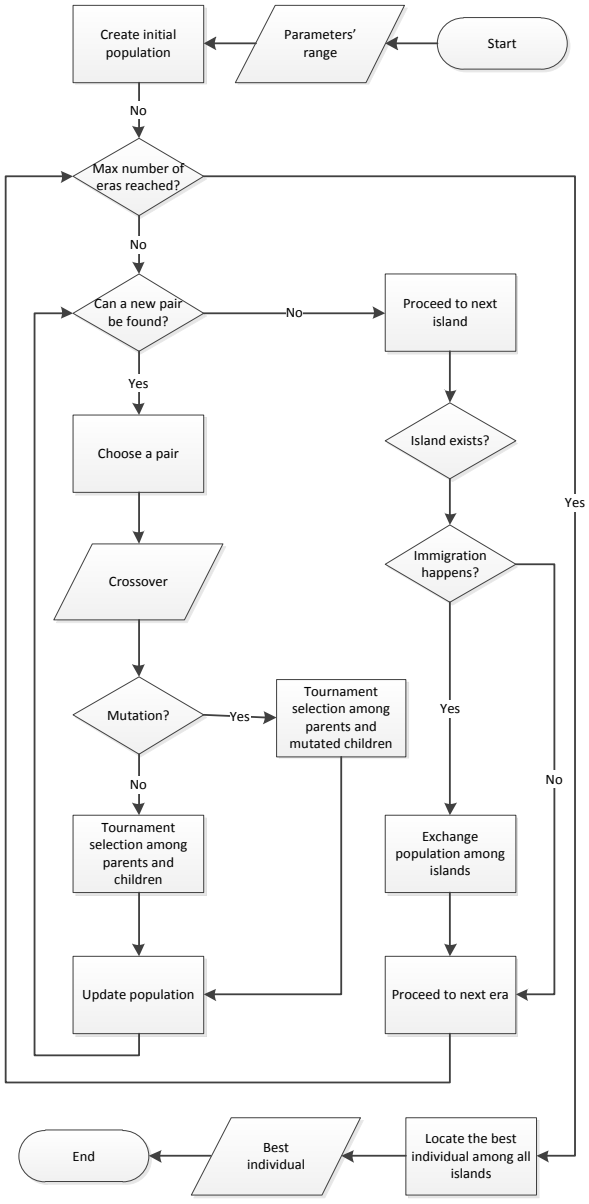

Figure 1. The island genetic algorithm flowchart.

Regarding the activation functions, the alternatives were five: tansig, logsig, satlin, hardlim, and hardlims. Individuals may mutate at any era. Mutation can change any

of the, previously stated, topology parameters therefore individuals' parameters outside the initially defined range may occur. The fitness of a network is evaluated using the following equation:

$$f_i = \lambda p_i + (1-\lambda)a \quad (1),$$

where f_i denotes the network's fitness score, p_i is the percentage of the correct in-sample classification and a is the average percentage difference, between the two greatest prices, among all the individual's outputs.

TABLE 1 ISLAND GENETIC ALGORITHM PARAMETERS' RANGE.

Parameter	Min value	Max value
Training epochs	100	400
Number of layers	1	3
Number of neurons (per layer)	4	10
Number of islands	3	3
Number of eras	10	10
Population (per island)	16	16

V. THE NONLINEAR CLASSIFIER

In this paper, the nonlinear classifier is a genetically optimized (topologically) feed forward neural network, according to the training sample. The neural network's topology is defined by the number of hidden layers, the neurons at each layer, the activation functions. All of the above as well as the number of training epochs were optimized using an island genetic algorithm.

Synaptic weights and bias values are, also, major factors of a network's performance. Nevertheless, since the initial training sample is small and noise exist at the data a good weight adaptation, for the in sample data, would not lead, necessarily, at an acceptable for the out of sample, performance.

Once the training phase is concluded, we start feeding the optimal network unlabeled data. Since the output vector of the classifier contains various values (its actual size is 1×7 as the number of the possible tasks), the output element with the greatest value will be turned into 1 while all the other ones will be set to 0. This is performed only if the greatest value is reliable. The conditions for the reliability are explained at the following section.

VI. THE SEMI SUPERVISED APPROACH

The main issue, in order to improve network's performance, is the reliability of labeling the new data, deriving from the pool of the unlabeled ones, exploiting network's performance in the already labeled data. In this approach output reliability is performed by comparing the absolute value of the greatest output element with the second greatest according to some criteria. If these criteria are not met, the output is considered vague, otherwise the classifier output is considered as reliable.

An unsupervised algorithm, like the k-means [14], is used in case of ambiguous results to support the decision. In particular, the unlabeled input vector that yields the vague output, say $\mathbf{u}$, is compared with all the labeled data, say $\mathbf{l}_i$, based on a similarity distance and then the distance values are normalized in the range of [0 1] so that all comparisons lie within a pre-defined reference frame, say $d(\mathbf{u},\mathbf{l}_i)$. Then, the k-means algorithm is activated to cluster, in an unsupervised way, all the normalized distances $d(\mathbf{u},\mathbf{l}_i)$ into a number of classes, equal to the number of available industrial tasks (7 in our case). In the sequel, the cluster that provides the maximum similarity (highest normalized distance) score, of the unlabeled data that yield the vague output and the labeled ones, is located. Let us denote as K the cardinality of this cluster (e.g., the number of its elements). In the following, the neural network output for the given unlabeled datum is linearly transformed according to the following formula,

$$\mathbf{n}_f = \mathbf{n}_p + \sum_{i=1}^{K} d(\mathbf{u},\mathbf{l}_i)\cdot \mathbf{v}_i \quad (2),$$

where $\mathbf{n}$ is the modified output vector, $\mathbf{n}_p$ the previous network output before the modification, while $d(\mathbf{u},\mathbf{l}_i)$ is the similarity score (distance) for the i-th labeled datum $\mathbf{l}_i$ and the unlabeled datum $\mathbf{u}$ within the cluster of the highest normalized distance, while $\mathbf{v}_i$ is the neural network output when input is the i-th labeled vector $\mathbf{l}_i$ and K is the cardinality of the cluster of the maximum highest similarity.

The modified output vector $\mathbf{n}$ which is the base for the decision is created using both manifold (FF neural network) and cluster assumption (similarity mechanism) [15].

VII. THE DECISION MECHANISM

According to the nonlinear classifier output, there are three possible cases:

1. The network made a robust decision that should not be defied. Therefore, the unlabeled data is used for further training but it is not incorporated at the initial training set.
2. The output is fuzzy, in other words, the difference among the two greatest prices does not exceed the threshold values. The similarity-based classifier is activated. If both systems indicate the same then the unlabeled data is used for further training but it is not incorporated at the initial training set.
3. The two classifiers do not agree. Therefore, an expert is called and specifies where the video should be classified. That video is added to the initial training data set.

The combination of these cases leads to a semi-supervised decision mechanism. Threshold values define which from the above scenarios will occur. The threshold value is defined as the percentage of the difference between the two greatest prices at the output vector. The overall process for the decision making is shown in Figure 2.

Initially, the first threshold value is set to 0.6. That value means that if the percentage difference of the two greatest values is above or equal to 60% we will be at scenario No 1.

The second threshold value is set to 20%. If the percentage difference of the two greatest values is less than that, the system is unable to make a decision and an expert is needed to interfere. Therefore, scenario No 3 will occur. Any value between these two thresholds activates scenario case No 2.

Since the model is self-trained, the first threshold value does not need to be so strict. The model learns through time, thus a reduction at that value would be acceptable. Nevertheless, at the beginning small threshold value could lead the model to wrong learning. Using simulated

 ISBN: 978-1-61208-226-4

annealing method, the threshold descents to a 40% through time.

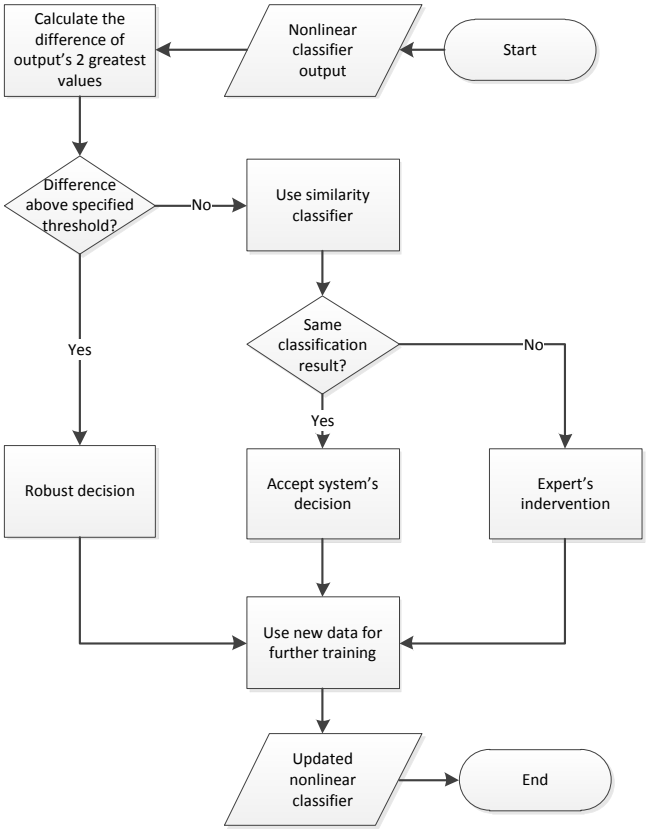

Figure 2. The decision mechanism flowchart.

VIII. EXPERIMENTAL VALIDATION

The production cycle on the industrial line included tasks of picking several parts from racks and placing them on a designated cell some meters away, where welding took place. Each of the above tasks was regarded as a class of behavioral patterns that had to be recognized. The behaviors (tasks) we were aiming to model in the examined application are briefly the following:

1. One worker picks part #1 from rack #1 and places it on the welding cell.
2. Two workers pick part #2a from rack #2 and place it on the welding cell.
3. Two workers pick part #2b from rack #3 and place it on the welding cell.
4. One worker picks up parts #3a and #3b from rack #4 and places them on the welding cell.
5. One worker picks up part #4 from rack #1 and places it on the welding cell.
6. Two workers pick up part #5 from rack #5 and place it on the welding cell.
7. Workers were idle or absent (null task).

For each of the above scenarios, 20 videos were available. An illustration of the working facility is shown in Figure 3.

A. *Experimental setup*

Initially, the best possible network is produced using the island genetic algorithm and 40% of the available data. The remaining data are fed to the network, one video at a time, and the overall out of sample performance is calculated.

In every case, all the data that activated scenario No 3 is excluded. Then, we reefed the network, one by one, with the rest data. If the network's suggestions were correct it will perform better since more training data (excluding these from scenario No 3) were used for further training.

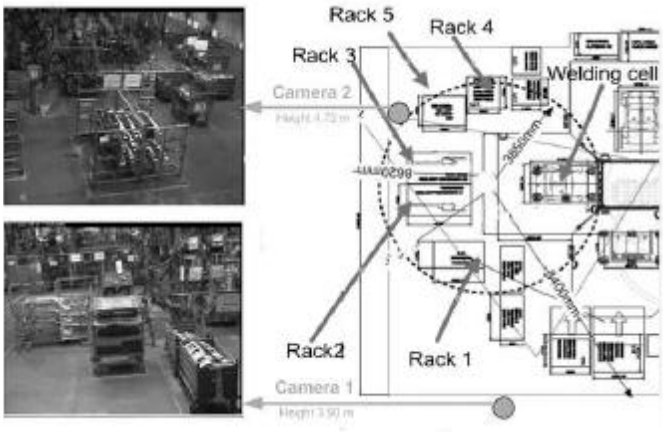

Figure 3. Depiction of a work cell along with the position of camera 1 and the racks #1-5.

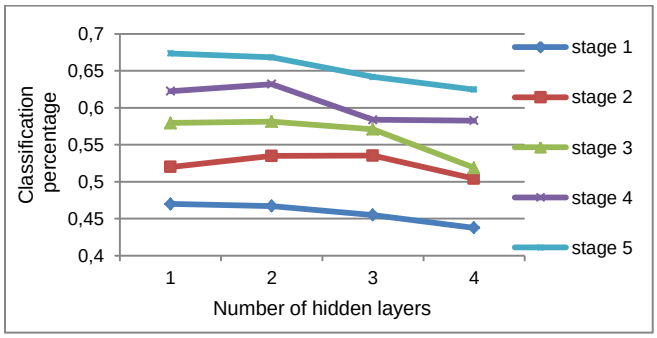

Figure 4. Classification percentages for each of the 5 evaluation stages – out of sample data.

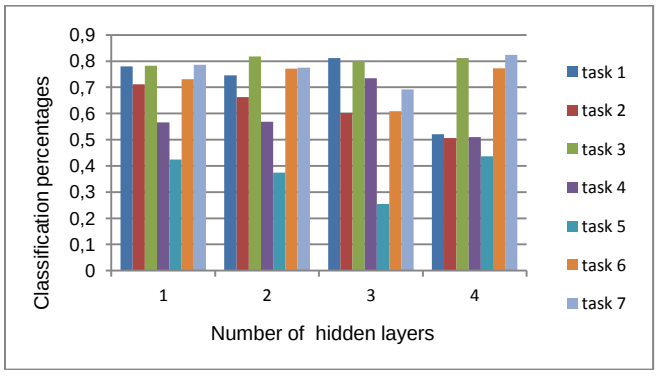

Figure 5. Stage 5 results for each one of the 7 tasks – out of sample data.

 ISBN: 978-1-61208-226-4

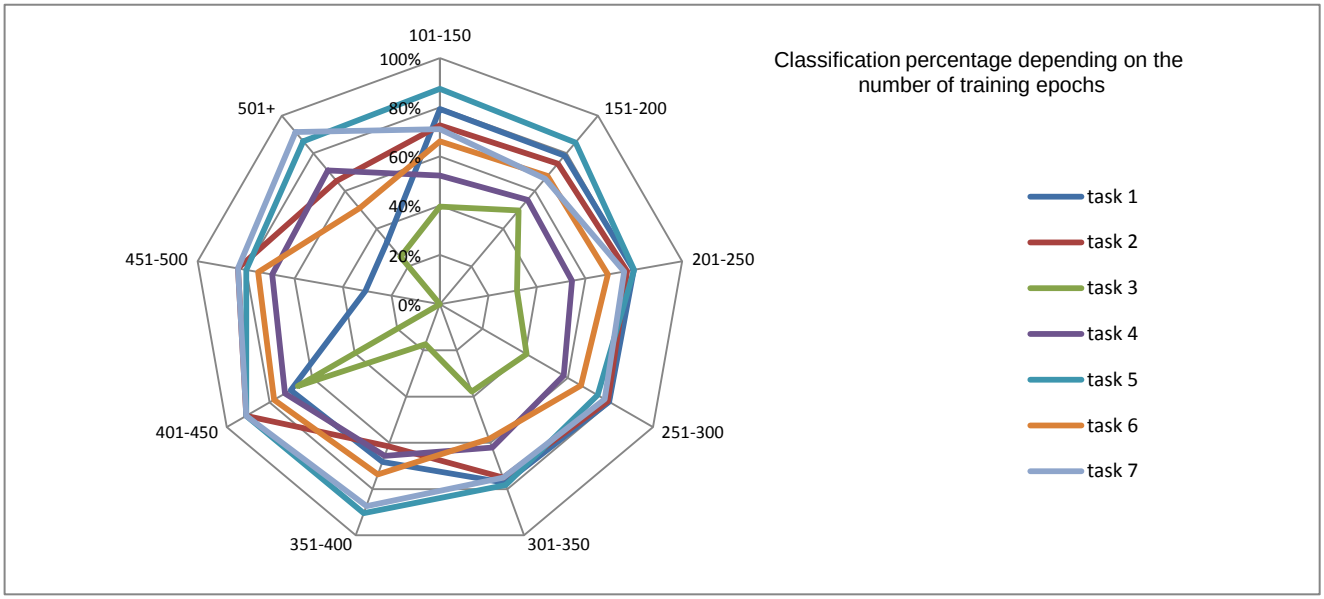

Figure 6. Classification percentage of the system depending on the number of training epochs of the nonlinear classifier.

By doing so, the unlabeled data fall below 60% and training data increases further. The above procedure concludes after five iterations. At that time the ratio between in sample data and out of sample data does not exceed 50%.

B. Results

The results displayed below are the average numbers after a total of 150 simulations of the proposed methodology. It appears that a two hidden layers neural network using tansig or logsig activation functions with an average of 9 neurons in each layer is the most suitable solution.

The proposed system is able to use the new knowledge to its benefit. The overall performance increases through iterations, using a small amount of data, as it is shown in Figure 4. Actually, by using additionally 10% of the videos, the system reached a 75% correct classification. This is important because the system saves time and resources during the initialization and provides good classification percentages using less than 50% of the available data.

The impact of the training epochs at the overall performance is shown at Figure 6. There appear to be a tradeoff between overall and individual task classification. Although 200 up to 300 training epochs provide significant classification accuracy further training increases partially the accuracy only on specific tasks in expense on others.

IX. CONCLUSION AND FUTURE WORK

In this work, we have proposed a novel framework for behavior recognition in workflows. The above methodology handles with an important problem in visual recognition: it requires a small training sample in order to efficiently categorize various assembly workflows. Such methodology will support the visual supervision of industrial environments by providing essential information to the supervisors and supporting their job.

Improvements at any stage of the system can be made in order to further refine the system's performance. Future work will be based on the usage of different classifiers (e.g. neuro-fuzzy, linear Support Vector Machines) and decision mechanism (e.g. voting-based). In addition, instead of using all frames of a specific task to create classifiers' input, only a subset of them may be used providing equivalent results.

ACKNOWLEDGMENT

The research leading to these results has been supported by European Union funds and national funds from Greece and Cyprus under the project "POSEIDON: Development of an Intelligent System for Coast Monitoring using Camera Arrays and Sensor Networks" in the context of the inter-regional programme INTERREG (Greece-Cyprus cooperation) - contract agreement K1 3 10–17/6/2011.

REFERENCES

[1] Y. Kim and H. Ling, "Human Activity Classification Based on Micro-Doppler Signatures Using a Support Vector Machine," IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 5, pp. 1328 –1337, May 2009.

[2] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities: A Survey," IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 11, pp. 1473 –1488, Nov. 2008.

[3] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models," IEEE Transactions on Image Processing, vol. 16, no. 7, pp. 1912 –1919, Jul. 2007.

[4] Y. Wang and G. Mori, "Human Action Recognition by Semilatent Topic Models," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 10, pp. 1762 –1774, Oct. 2009.

[5] Q. Hu, L. Qin, Q. Huang, S. Jiang, and Q. Tian, "Action Recognition Using Spatial-Temporal Context," in Pattern Recognition (ICPR), 2010 20th International Conference on, 2010, pp. 1521 –1524.

 ISBN: 978-1-61208-226-4

[6] S. Ali and M. Shah, “Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 2, pp. 288 –303, Feb. 2010.

[7] Yao, J. Gall, and L. Van Gool, “A Hough transform-based voting framework for action recognition,” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 2061 –2068.

[8] R. Poppe, “A survey on vision-based human action recognition,” Image and Vision Computing, vol. 28, no. 6, pp. 976–990, Jun. 2010.

[9] W. Hu, T. Tan, L. Wang, and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 34, no. 3, pp. 334 –352, Aug. 2004.

[10] Voulodimos, D. Kosmopoulos, G. Vasileiou, E. Sardis, A. Doulamis, V. Anagnostopoulos, C. Lalos, and T. Varvarigou, “A dataset for workflow recognition in industrial scenes,” in 2011 18th IEEE International Conference on Image Processing (ICIP), 2011, pp. 3249 –3252.

[11] M. Ahad, J. Tan, H. Kim, and S. Ishikawa, “Motion history image: its variants and applications,” Machine Vision and Applications, vol. 23, no. 2, pp. 255–281, 2012.

[12] D. I. Kosmopoulos, N. D. Doulamis, and A. S. Voulodimos, “Bayesian filter based behavior recognition in workflows allowing for user feedback,” Computer Vision and Image Understanding, vol. 116, no. 3, pp. 422–434, 2012.

[13] D. Whitley, S. Rana, and R. B. Heckendorn, “The island Model Genetic algorithm : On separability, population size and convergence,” CIT. Journal of computing and information technology, vol. 7, no. 1, pp. 33–47.

[14] J. Wu, “Cluster Analysis and K-means Clustering: An Introduction,” in Advances in K-means Clustering, Springer Berlin Heidelberg, 2012, pp. 1–16.

[15] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, “SemiBoost: Boosting for Semi-Supervised Learning,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 11, pp. 2000 –2014, Nov. 2009.

 ISBN: 978-1-61208-226-4