



A Hybrid Boosted-SVM Classifier for Recognizing Parts of 3D Objects

Omar Herouane^{1*} Lahcen Moumoun¹ Taoufiq Gadi¹ Mohamed Chahhou²

¹ Faculty of Science and Technology, Hassan Ist University Settat, Laboratory Informatics, Imaging and Modelling of Complex Systems, Morocco

² Faculty of Science Dhar El Mahraz, Sidi Mohamed Ben Abdellah University Fes, Laboratory Informatics, Modelling and Systems, Morocco

* Corresponding author's Email: omarherouane@gmail.com

Abstract: The main objective of classifying parts of a 3D object is to group them into similar and meaningful categories. This purpose is a challenging task in pattern recognition, especially when the risk of assigning simultaneously the same label of a class to many homogenous categories is high. Hence, this paper aims to develop a robust algorithm for 3D objects-parts recognition based on a hybrid approach combining a boosting algorithm and a SVM classifier. This approach called Boosted-SVM consists of two steps, the first one is training the strong SVM classifier to be a weak base learner, and the second one is combing these trained classifiers into a strong one using the Adaboost algorithm. The training of the classification model is based on a significant and robust feature vector representing parts of the 3D object, the Shape Spectrum Descriptor (SSD) is used for this purpose. The performed experimental results have shown excellent recognition rate and very short training time of the built system.

Keywords: Boosted-SVM, SVM, Machine learning, 3D object-parts Recognition, Shape spectrum descriptor.

1. Introduction

The fast development of science and technology related to 3D scanning allows us to obtain a large number of 3D objects from scanners, networks, and databases. As a result, the request for 3D objects processing has become more important. Recognizing 3D objects in a database is a challenging and valuable research field in pattern recognition in order to analyze and understand the information processing. It has been widely applied in the graphical analysis for various areas including video gaming, design engineering and virtual reality.

Recently, a number of studies have been made for recognizing components of an image, these studies can be divided into two major categories: the traditional boundary based image analysis and the object-oriented image analysis. The first approach aims to predict whether each element from an image belongs to the boundary of an object or not [1], for this purpose, a classification model is trained using human-labeled images. The output of this classifier

provides the posterior probability of a boundary at each image location and orientation. The second approach has to predict whether each element from the image belongs to the interior of an object or not [2]. Furthermore, many machine learning methods are used to classify components of an image or a scene into similar parts. Support Vector Machine algorithms (SVM) are one of these methods, many recent research works were conceived for this goal. The authors of [3] developed a robust SVM classifier to classify images in a scene into different semantic categories, the optimization of parameters of SVM is done using a particle swarm optimization (PSO) based algorithm. The one-versus-one strategy is used to construct the Multiclass-SVM to classify the multiple categories scene. In [4], the authors developed a new relevant SVM-RBF based approach using the probabilistic feature and weighted kernel function for image retrieval. The authors of [5] presented an automatic pattern recognition system for the classification of the cell images into a set of predefined classes, in which the

Radial Basis Kernel is utilized. The one-versus-one strategy is used to construct the SVM classifier.

Boosting algorithms are also machine learning methods used in cognitive systems. Therefore, several studies used these techniques in recognizing objects in a database, classifying a text and classifying components of an image. In [6], the authors built an intelligent vehicle system for detecting and recognizing multi-type road making, they used Adaboost and Haar-like feature to fast eliminate non-significant candidates regions, their experimental results showed a much better recognition performance. Augmented reality where virtual objects are overlaid into a real world is another domain where boosting technique is used; in [7], Mahmood et al. developed a novel approach to display related statistics to a player in an image. They used Adaboost classifier with Linear Discriminant Analysis as a weak learner and Haar-like features to recognize faces in an image. The authors of [8] developed a new human action recognition system based on Adaboost algorithm. The proposed system was able to select the most significant samples from a large training data, to reduce the computational complexity and to achieve high recognition accuracy. Lee et al. have also developed a new system for text detection in a natural scene using Adaboost algorithm [9].

For 3D objects recognition, most of the built SVM systems for detecting objects used a 2D-3D approach [10,11-12]. Due to the complexity of 3D objects shape and the diversity of their format that make their study difficult, the authors have chosen to recognize 3D models in 2D image instead of using 3D scanned models. Moreover, many researches works using boosting algorithms for recognizing components of scanned 3D objects were developed. In one of our early works [13], we conceived a framework for 3D object-parts learning in which we used Adaboost.M1 with decision-stump as a weak learner. The obtained results are very encouraging however, the system still find some difficulties concerning recognizing some heterogeneous classes. Therefore, we aim to strengthen the classifier used for recognition. There exist also other systems for recognizing components of 3D objects; the majority of them used Adaboost algorithm variants to build functions used for recognition [14, 15-16]. Most of these proposed approaches using machine learning used a multitude of geometric and topologic characteristics and need a post-processing to refine the obtained results. Minimizing the number of these characteristics will have a direct impact on the learning approach computational complexity. Also, these works need a

consistent labeling of the training data, which requires some manual interactions.

For the same goal, we exploit the relevance of the two supervised machine learning algorithms, SVM and Adaboost, to develop a new relevant hybrid algorithm named Boosted-SVM in order to recognize parts of 3D objects. These objects' parts to be recognized are those extracted from the Ground-truth (a set of manually segmentations generated by human experts) of the "3D mesh segmentation benchmark" database introduced by Chen et al. [17]. Furthermore, our proposed algorithm can be used later to recognize parts generated by automatic segmentation's algorithms like those accompanying the cited benchmark or some new relevant algorithms like those of Arhid et al.[18] and Chahhou et al. [19].

Boosted-SVM is a machine learning technique for improving the performance of weak learners. It invokes iteratively a base learner and then, a single strong one is built by combining these weak learners generated by re-sampling the training set. The learned strong function is then used to recognize parts of 3D objects. For this purpose, we use a base learner that is already strong (SVM) instead of a weak one. We choose SVM since it converges very fast and the execution time is not consuming when compared with other strong and stable classifiers like Artificial Neural Network. Also, little modification in extracted feature vectors does not affect its performance. SVM can as well perform in n-Dimensional space with a few training data [20]. The choice of the kernel function is also crucial for the success of the SVM algorithm, thus, for an ideal kernel, input patterns in the same class should have high kernel value while input patterns in different classes should have low kernel value. The RBF (Radial Basis Function) kernel [21] is chosen in this work, considering it can handle the case of the nonlinearity between the attributes and the class labels. After building the Boosted-SVM classifier, we extract the feature vectors representing parts of a 3D object. The Shape Spectrum Descriptor (SSD) was used since it can characterize efficiently the shape of different parts of the 3D objects [22]. Then, we feed these extracted feature vectors to the Boosted-SVM algorithm to learn the classification function.

Contributions of this paper are organized as follows. In the next section, we introduce the Support Vector Machine algorithm and the strategy used for the multiclass-SVM. In section 3, we describe the proposed algorithm combining SVM with Adaboost algorithm to obtain a powerful one

called Boosted-SVM and the strategy used to recognize parts of a 3D object. To our knowledge, this is the first framework combining SVM with Adaboost algorithm to recognize parts of scanned 3D objects in a database. Section 4 will contain the experiments performed to access the relevance of our approach. Finally, we conclude this paper in the last section.

2. Support vector machines algorithm

2.1 Support vector machines

Support Vector Machines are powerful methodologies for solving linear or nonlinear classification or regression problems [23]. Their aim is to find a computationally efficient way for learning good separating hyperplanes in a hyperspace. These hyperplanes are built based on the maximization of the distance between them and samples in different categories. SVMs are also a set of supervised learning machines used for training a classification model that will assign to the appropriate class the new input given patterns.

Considering the classification of two classes, given a training dataset $\{x_i, y_i\}_{i=1}^N$, with $x_i \in R^d$ being the input vector and $y_i \in \{-1, +1\}$ the class labels. SVMs map the d -dimensional input vector x from the input space to the d_h -dimensional feature space using a nonlinear function $\varphi(i): R^d \rightarrow R^{d_h}$.

The separating hyperplane in the feature space is then defined as:

$$w^T \varphi(x) + b = 0 \quad (1)$$

with $b \in R$ the bias and w represents the weight vector with the same dimension as $\varphi(x)$.

The SVM classifier satisfies the following conditions:

$$\begin{cases} w^T \varphi(x) + b \geq +1, & \text{if } y = +1, \\ w^T \varphi(x) + b \leq -1, & \text{if } y = -1, \end{cases} \quad (2)$$

The classifier can be then given by:

$$f(x) = \text{sign}(w^T \varphi(x) + b). \quad (3)$$

The optimal hyperplane is the one that maximizes the margin between the samples and the separating hyperplane which is equal to:

$$2/\|W\| = 2/\sqrt{W^T W} \quad (4)$$

Therefore, maximizing Eq. (4) is the same as minimizing the standard formula of SVM Eq. (1):

$$\begin{aligned} \min_{w, \xi} & \frac{1}{2} W^T W + C \sum_{i=1}^N \xi_i \\ \text{Subject to} & y_i (w^T \varphi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \dots, N. \end{aligned} \quad (5)$$

The C parameter trades off the misclassification of training patterns against the simplicity of the decision surface and ξ_i is the slack variable that indicates the distance of x_i with respect of the distance boundary.

This problem can be solved and written in the dual space using the Lagrangian with Lagrange multipliers, thus the solution for the Lagrange multipliers is obtained by solving a quadratic programming problem [24].

Finally, the Decision function takes the form:

$$f(x) = \text{sign}(\sum_{i=1}^{\#Ns} \alpha_i y_i K(x, x_i) + b) \quad (6)$$

where $\#Ns$ represents the number of support vectors, α_i Lagrange multipliers and $K(x, x_i)$ the kernel function.

Several forms of kernel functions are employed in SVM classifiers, the most frequently used are the polynomial kernel and the RBF kernel.

2.2 Multiclass-SVM algorithm

Intrinsically, the SVM has been developed for binary classification. However, different multiclass strategies can be adopted, the one-versus-one (*OVO*) and one-versus-all (*OVA*) strategies [25, 26]. The difference between them is the number of classifiers we have to train, which basically correlates with the decision boundary they build. The *OVO* strategy consists on constructing one SVM for each pair of classes. To classify a new input data, the maximum voting is performed, where one SVM classifier votes for each class. Hence, if we consider n classes, $n(n-1)/2$ classifiers are built where each classifier is trained using data from two classes and the final decision of classification is done using the majority vote strategy. However, the *OVA* strategy consists on constructing one SVM classifier for each class fitted against all the others. Therefore, if we consider n classes, only n classifiers for the *OVA* approach are needed.

The authors of this paper developed and implemented also a single Multiclass-SVM algorithm based on the *OVO* strategy. The *OVO* strategy is used instead of *OVA* for handling multiclass categorization for the reason that, *OVO* is significantly less sensitive to the imbalanced datasets problems and the time we need for training classifiers may decrease. This decrease is due to the fact that the training dataset for each classifier is much smaller. The algorithm also uses the Radial Basis Function kernel for SVM since, it works well in practice and relatively easy to calibrate as opposed to other kernels. RBF can be formulated as Eq. (7):

$$K(x, y) = \exp(-\gamma \|x - y\|^2) \quad (7)$$

where $K(x, y)$ the function for all $\gamma > 0$ that controls the shape of the decision boundary of the RFB kernel.

2.3 Parameters optimization

The optimal values of the parameters C and Γ belonging to the RBF-SVM kernel should be determined in advance, since they affect the generalization performance of SVM. The C parameter trades off misclassification of training patterns against the simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly by giving the model freedom to select more samples as support vectors. However, the Γ parameter defines how far the influence of a single training example reaches, it can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

The process of parameters optimization used in this work is *10-fold* cross validation, the classifier is trained on *9/10* folds and tested on the remaining *1/10* fold. This process is repeated ten times to let the classifier go through each fold for training and testing. In practice, cross-validation based techniques are preferred over generalization error bounds. Even though some of these bounds hold strong theoretical properties closely linked to the SVM foundations, experimental results on a large number of problems still favor cross-validation strategies.

3. Proposed approach for 3D object-parts recognition

3.1 Boosted-SVM algorithm

Boosting algorithms are commonly used in Computer Vision [27]. They are machine learning techniques for improving the performance of a weak learner [28, 29]. The method invokes iteratively a weak learner and the training data used is taken from various subsets of the whole database. Then, a single strong classifier is built by combining these weak learners generated by re-sampling the training set. This ensemble method in general is a very successful technique for solving the binary classification problem. The majority of algorithms reduce the multiclass classification problem to multiple binary ones [30].

Boosted-SVM is a generalization of AdaBoost for multiclass tasks. It is the same as AdaBoost.M1 [31], except that the used base learner is already strong (SVM) instead of a weak one (one-dimensional decision stumps).

SVM is a strong classifier, hence using it as the base learner in AdaBoost does not give significant results in terms of generalization error and not always appropriate for the boosting constraint. However, to be able to use definitely SVM as a base learner we should use different configurations to make it weak by giving large values to Γ in the RBF kernel and by using smaller training data set. To adjust values of SVM parameters, we use cross-validation procedure. This procedure will be described in a following section.

The following pseudo-code describes different instructions of the proposed Boosted-SVM algorithm.

- The classifier takes as **input**:
 - Sequence of m examples:
 - $\ll (x_1, y_1), \dots, (x_m, y_m) \gg$, x_i observed feature vectors with their labels $y_i \in Y = \{1, \dots, k\}$.
 - SVM with RBF kernel as a base learner, with fixed values of Γ and C .
 - T Number of predefined iterations.
- Initialize $D_1(i) = \frac{1}{m}$, weights of all instances.
- For $t = 1$ to T do
 - Construct the distribution D_t and Call Multiclass Weak Learner.

- Get back a hypothesis $h_t : X \rightarrow Y$ with a small error $\varepsilon = Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$.
- Set $\beta_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$, weight assigned to every individual classifier.
- Update Distribution D_t , increases weights of all misclassified instances and decreases those of correctly classified instances.
- **Output:** The final classifier combining the T hypothesis weighted h_t by a voting approach.

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad (8)$$

3.2 3D object-parts recognition process

The first problem of the 3D Object recognition is to express the form of its parts into a representative term as a feature vector. The more detailed a shape is described the more a classifier will be able to distinguish different parts of the object. In this work, Shape Spectrum Descriptor (*SSD*) will represent the shape of 3D object-parts, it has been proposed by Zaharia et al. [22]. *SSD* is based on the notion of shape index [32] that characterizes locally the shape of a surface. Each part of a 3D object is represented by the distribution of its faces areas with respect to their shape index values. The choice of the shape index is done as it allows the shape of a 3D object surface to be defined [13].

Processes of the proposed 3D object-parts recognition approach, based on Boosted-SVM algorithm, consist of the two following steps: the feature learning process in which a classification function is learned and the recognition process where the learned function assigns a label to the appropriate class.

Let SSD_{vi} be the observed feature vector associated with a segment of a 3D object. The objective is to assign the appropriate class C_i to SSD_{vi} , where $C = \{C_1, \dots, C_n\}$ is a predefined set of all possible classes for a part of a 3D object. The first step is to select the training samples and compute the feature information. We then use these feature vectors to train the recognition function using the Boosted-SVM algorithm. The second step is to select the testing samples and test them using the recognition function. This function will recognize and assign to each testing sample the appropriate class label.

4. Experiments

To access the effectiveness of the proposed system for 3D object-parts recognition, we use the “3D mesh segmentation benchmark” database

Table 1. Description of the “3D mesh segmentation benchmark” database used for tests.

Class Name	Segments/ Object	Training sample	Testing sample
Human	6	76	38
Cup	2	27	13
Glasses	5	47	23
Airplane	7	89	44
Ant	11	73	36
Chair	8	69	34
Octopus	9	114	57
Table	5	37	18
Teddy	8	107	53
Hand	6	80	40
Plier	5	60	30
Fish	6	52	26
Bird	5	63	31
Armadillo	7	61	30
Bust	8	53	26
Mech	2	27	13
Bearing	5	23	11
Vase	5	47	23
Fourleg	9	78	39

proposed by Chen et al. [17]. The benchmark contains 380 3D-models of triangular meshes grouped into 19 categories. We divided these 19 categories into homogeneous and heterogeneous classes, this distinction depends on the visual aspect of the shape of the 3D object.

A set of manually segmentations generated by experts is provided for each 3D-model. In our experiments, we choose four different segmentations per object to train our recognition function since, each object of the ground-truth has a different segmentation according to the expert who performed it. Also, each object of the same category may have different shapes and poses. Besides, for each category we fix the number of segments per object in order to be sure that experiments are performed in the same conditions. In order to train and test the proposed system, the size of each training subset was chosen to be 2/3 and the remaining 1/3 for testing. Table 1 gives more details about the distribution of the number of training and testing samples.

The parameters optimization of our system is a crucial step, we use *10-cross-validation* process (sec.2.3) to fix values of C , γ and T for Boosted-SVM. We also adjust the parameters C and γ of a single Multiclass-SMV to do a comparison with the proposed approach based on Boosted-SVM algorithm. For the Multiclass-SVM experiments, the suitable values of the optimal parameters C and Gamma of the RBF kernel for each classier are:

Table 2. Recognition accuracy (%)

Class Name	Multiclass-SVM (%)	AdaBoost.M1 (%)	Boosted-SVM (%)	JointBoost [14] (%)
Human	86.00	97.00	97.50	89.4
Cup	90.00	96.00	98.40	99.1
Glasses	87.20	94.40	96.50	96.1
Airplane	76.30	89.50	92.10	95.5
Ant	82.30	90.50	94.50	98.7
Chair	77.50	95.30	95.80	97.8
Octopus	81.50	95.20	98.75	98.6
Table	79.50	91.10	92.80	99.1
Teddy	85.60	97.10	97.85	93.3
Hand	78.50	93.60	95.15	82.4
Plier	75.50	94.70	94.90	94.3
Fish	75.12	95.30	96.20	95.6
Bird	71.23	88.50	90.70	84.2
Armadillo	74.50	84.60	89.90	84.0
Bust	69.00	67.90	70.00	53.9
Mech	91.20	94.50	94.80	88.9
Bearing	74.60	94.50	95.60	84.8
Vase	69.20	80.50	85.00	77.0
Fourleg	77.00	95.20	96.00	85.0
Average	79.00	91.34	93.3	89.4

Table 3. Training time (s)

Classifier	Multiclass-SVM	Adaboost.M1	Boosted-SVM
Training time (s)	1.63	5.00	6.00

$C = 20$ and $\gamma = 0.0001$. While, for the Boosted-SVM, the suitable value of the optimal parameters C and γ for each classifier are: $C = 5$ and $\gamma = 0.05$ (these values make the SVM weak), and we fix the number of iterations T to 10.

The classification performances can be determined by computing the accuracy parameter which is defined as shown in Eq. (9).

$$Accuracy = \frac{T_N + T_P}{T_N + T_P + F_N + F_P} \quad (9)$$

where T_N , T_P , F_N , and F_P are respectively *True Negative*, *True Positive*, *False Negative*, and *False Positive* testing data computed using the confusion matrix.

All of our experiments are conducted on an Intel CORE i3.4 GHz dual-core processor with 4 GB RAM. Table 2 illustrates the recognition statistical results using Boosted-SVM, Multiclass-SVM, AdaBoost.M1 [13] classifiers under different training samples and this of Kalogerakis et al. [14] using the JointBoost classifier.

From Table 2, the recognition average accuracy shows that the relevance of Boosted-SVM outperforms the ones obtained by Multiclass-SVM AdaBoost.M1 and the JointBoost. The average

accuracy is 93.3% for Boosted-SVM, 91.34% for Adaboost.M1, 89.4% for JointBoost and 79% for Multiclass-SVM. From these results, we can remark that SVM classifiers are strong, which mean that the adjustment of the parameters we have done is necessary to make it weak and combine it with a stronger one. Also, the increasing recognition accuracy of Boosted-SVM compared to Adaboost.M1 means its ability to solve problems encountered by Adaboost.M1 for recognizing specific classes from the database. From Table 3, we can see that Multiclass-SVM classifier is faster than the others; the training time is very short and can be ignored in practice (1.63s). While, the training time for AdaBoost.M1 and Boosted-SVM classifiers is respectively (5.0s), (6.0s). Consequently, it cannot be ignored in practice.

The Area under the curve (*AUC*) is another parameter that we used to access the effectiveness of a classifier. In order to compute this parameter, we have to plot the *ROC* curve that measures the accuracy of a classifier. To do that, two specific parameters are computed:

$$Sensitivity = \frac{T_P}{T_P + F_N} \quad (10)$$

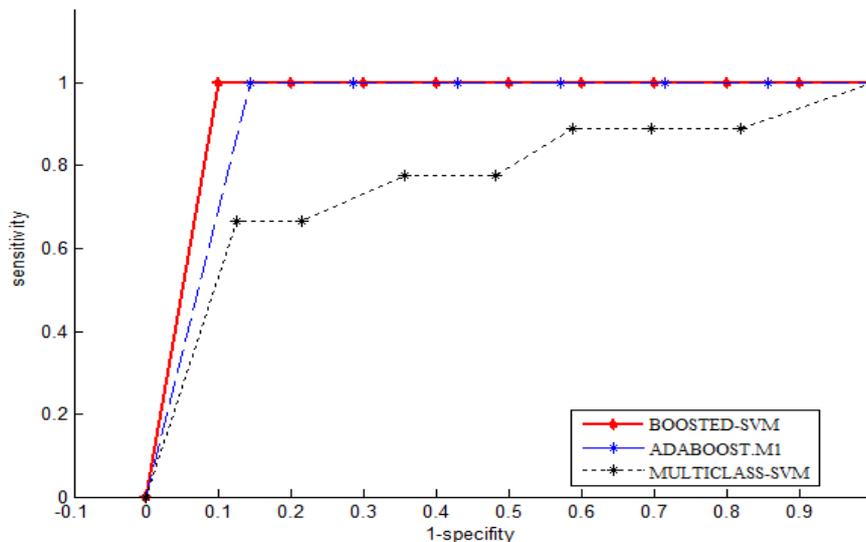


Figure.1 Mean Roc Curve of homogenous classes

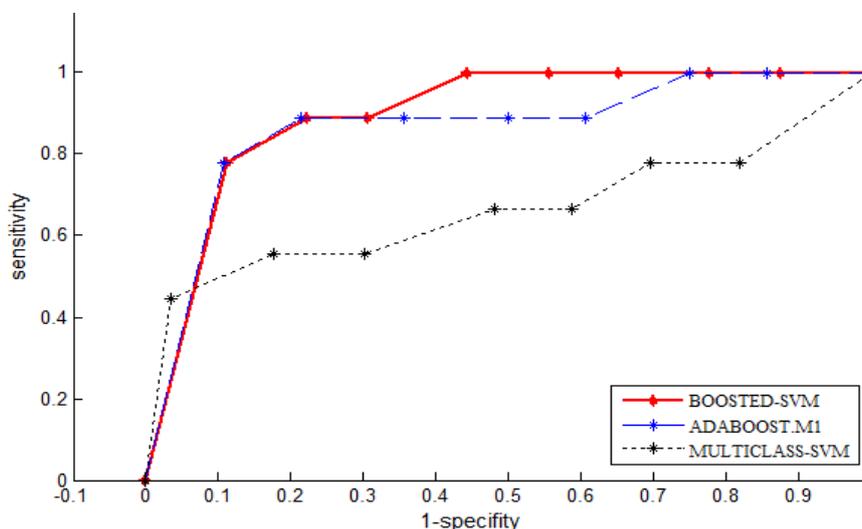


Figure.2 Mean Roc Curve of heterogeneous classes

$$1 - Specificity = \frac{FP}{TN+FP} \tag{11}$$

After computing the two parameters, the *1-Specificity* is taken along the x-axis whereas the *Sensitivity* parameter takes place along the y-axis at various threshold settings. The curve is then generated by plotting the *Sensitivity* against the *1-Specificity*. The performances of the three classifiers for different homogeneous and heterogeneous categories have been evaluated by plotting their *ROC* curves.

Fig. 1 illustrates the mean *ROC* Curve accuracy of the homogenous classes for the three approaches. From this *ROC* Curve in term of the *AUC*, we can see that the Boosted-SVM based system gives high *AUC* with 0.96. While, for AdaBoost.M1 and Multiclass-SVM based systems it gives respectively 0.94, 0.85. These results indicate that the Boosted-

SVM provides better recognition accuracy than any other classifier for the homogenous classes with a low improvement compared to AdaBoost.M1.

From the *Roc* curve plotted for heterogeneous categories (Fig. 2), the mean accuracy in term of the *AUC* is 0.90 for Boosted-SVM, 0.88 for Adaboost-M1 and 0.70 for Multiclass-SVM. Therefore, it's crystal clear that the results of the two first mentioned classifiers are encouraging and show good accuracy within experimental errors. We can remark also an improvement of Boosted-SVM compared to Adaboost.M1, the combination of SVM as a weak learner with Adaboost seems to be the best solution to recognize 3D objects-parts of the same category with different shapes. Finally, we conclude that the Boosted-SVM classifier outperforms both Adaboost.M1 and SVM in this benchmark in term of recognition rate.

5. Conclusion

The purpose of this study is to combine the SVM classifier with Radial Basis Function kernel and Adaboost algorithm for automatic 3D object-parts recognition in a database. We evaluated the performance of the proposed system in terms of recognition rate and training time using “3D mesh segmentation benchmark”. The experimental results obtained demonstrate that the proposed Boosted-SVM approach and the utilized shape descriptor are discriminative and robust in terms of recognition accuracy rate. The obtained 93.3% accuracy rate for the Boosted-SVM classifier shows undoubtedly its powerful recognition ability with an effortless way and not time consuming.

In our Future works, we attempt to improve the performance of our proposed Boosted-SVM algorithm by using other shape descriptors and minimizing the number of training samples. We are also implementing the algorithm for recognizing components of 3D objects generated by automatic segmentation’s algorithm.

References

- [1] D. R. Martin, C. C. Fowlkes, and J. Malik, “Learning to detect natural image boundaries using local brightness, color, and texture cues”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, No. 5, pp. 530–549, 2004.
- [2] Y. Chen and J. Z. Wang, “Image categorization by learning and reasoning with regions”, *Journal of Machine Learning Research*, Vol. 5, No. Aug, pp. 913-939, 2004.
- [3] H. Yin, X. Jiao, Y. Chai, and B. Fang, “Scene classification based on single-layer SAE and SVM”, *Expert Systems with Applications*, Vol. 42, No. 7, pp. 3368–3380, 2015.
- [4] X.-Y. Wang, L.-L. Liang, W.-Y. Li, D.-M. Li, and H.-Y. Yang, “A new SVM-based relevance feedback image retrieval using probabilistic feature and weighted kernel function”, *Journal of Visual Communication and Image Representation*, Vol. 38, pp. 256–275, 2016.
- [5] D. Cascio, V. Taormina, M. Cipolla, S. Bruno, F. Fauci, and G. Raso, “A multi-process system for HEp-2 cells classification based on SVM”, *Pattern Recognition Letters*, Vol. 82, pp. 56–63, 2016.
- [6] W. Liu, J. Lv, B. Yu, W. Shang, and H. Yuan, “Multi-type road marking recognition using adaboost detection and extreme learning machine classification”, In: *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, Seoul, Korea (South), pp. 41-46, 2015.
- [7] Z. Mahmood, T. Ali, and S. Khattak, “Automatic player detection and recognition in images using AdaBoost”, In: *Proc. of IEEE 9th International Bhurban Conf. on Applied Sciences and Technology (IBCAST)*, Islamabad, Pakistan, pp. 64-69, 2012.
- [8] X. Ji, L. Zhou, and Y. Li, “Human Action Recognition Based on AdaBoost Algorithm for Feature Extraction”, In: *Proc. of IEEE International Conf. on Computer and Information Technology*, Xi'an, China, pp. 801-805, 2014.
- [9] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, “Adaboost for text detection in natural scene”, In: *Proc. of IEEE International Conf. on Document Analysis and Recognition (ICDAR)*, Beijing, China, pp. 429-434, 2011.
- [10] B. Pepik, M. Stark, P. Gehler, T. Ritschel, and B. Schiele, “3d object class detection in the wild”, In : *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Boston, USA, pp. 1-10, 2015.
- [11] M. Pontil and A. Verri, “Support vector machines for 3D object recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 6, pp. 637–646, 1998.
- [12] X. Li and I. Guskov, “3D object recognition from range images using pyramid matching”, In: *Proc. of IEEE 11th International Conf. on Computer Vision (ICCV)*, Rio de Janeiro, Brazil , pp. 1-6, 2007.
- [13] O. Herouane, L. Moumoun, and T. Gadi, “Using bagging and boosting algorithms for 3D object labelling”, In: *Proc. of 7th IEEE International Conf. on Information and Communication Systems (ICICS)*, Irbid, Jordan, pp. 310-315, 2016.
- [14] E. Kalogerakis, A. Hertzmann, and K. Singh, “Learning 3D mesh segmentation and labelling”, *ACM Transactions on Graphics (TOG)*. ACM., pp. 102, 2010.
- [15] H. Benhabiles, G. Lavoué, J.-P. Vandeborre, and M. Daoudi, “Learning Boundary Edges for 3D-Mesh Segmentation”, *Computer Graphics Forum*, Vol. 30, No. 8, pp. 2170–2182, 2011.
- [16] J. Lv, X. Chen, J. Huang, and H. Bao, “Semi-supervised Mesh Segmentation and Labeling”, *Computer Graphics Forum*, Vol. 31, No. 7, pp. 2241–2248, 2012.
- [17] X. Chen, A. Golovinskiy, and T. Funkhouser, Thomas, “A benchmark for 3D mesh segmentation”, *ACM Transactions on Graphics (tog)*, ACM., pp. 73, 2009.

- [18] K. Arhid, F. Zakani, M. Bouksim, M. Aboulfatah, and T. Gadi, "An Efficient Hierarchical 3D Mesh Segmentation Using Negative Curvature and Dihedral Angle", *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 5, pp. 143–152, 2017.
- [19] M. Chahhou, L. Moumoun, M. E. Far, and T. Gadi, "Segmentation of 3D Meshes Using Spectral Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 8, pp. 1687–1693, 2014.
- [20] R. S. Lodhi and S. K. Shrivastava, "Evaluation of Support Vector Machines Using Kernels for object detection in images", *International Journal of Engineering Research and Applications (IJERA)*, Vol. 2, pp. 269-273, 2012.
- [21] B. Scholkopf, K.K. Sung, C.J.C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers", *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp. 2758–2765, 1997.
- [22] T. Zaharia and F. Preteux, "Shape-based retrieval of 3D mesh models", In: *Proc. of IEEE International Conf. on Multimedia and Expo (ICME'02)*, Lausanne, Switzerland, pp. 437-440, 2002.
- [23] V. N. Vapnik and V. Vapnik, *Statistical learning theory*, New York : Wiley, 1998.
- [24] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms", *IEEE Transactions on Neural Networks*, Vol. 12, No. 2, pp. 181-201, 2001.
- [25] E. Mayoraz and E. Alpaydin, *Support vector machines for multi-class classification*, Engineering Applications of Bio-Inspired Artificial Neural Networks, pp. 833-842, 1999.
- [26] J. Weston, C. Watkins, *Multi-class support vector machines*, Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, May, 1998.
- [27] M. Korytkowski, L. Rutkowski, and R. Scherer, "Fast image classification by boosting fuzzy classifiers", *Information Sciences*, Vol. 327, pp. 175–182, 2016.
- [28] R.E. Schapire, "The strength of weak learnability", *Machine learning*, Vol. 5, No. 2, pp. 197-227, 1990.
- [29] M. Kearns and L. Valiant, "Cryptographic limitations on learning Boolean formulae and finite automata", *Journal of the ACM*, Vol. 41, No. 1, pp. 67–95, 1994.
- [30] R.E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions", *Machine learning*, Vol. 37, No 3, pp. 297-336, 1999.
- [31] Y. Freund, and R.E. Schapire, "Experiments with a new boosting algorithm", In: *Proc. of International Conf. on Machine Learning (Icml)*, Bari, Italy, pp. 148-156, 1996.
- [32] J. J. Koenderink and A. J. van Doorn, "Surface shape and curvature scales", *Image and Vision Computing*, vol. 10, No. 8, pp. 557–564, 1992.