Feature Selection Using Genetic Algorithms for Hand Posture Recognition

Uriel H. Hernandez-Belmonte and Victor Ayala-Ramirez(✉)

División de Ingenierías, Campus Irapuato- Salamanca DICIS,
Universidad de Guanajuato DICIS, Carr. Salamanca-Valle Km. 3.5+1.8,
Palo Blanco, 36700 Salamanca, Mexico
hailehb@laviria.org, ayalav@ugto.mx

Abstract. In this work, we propose a feature selection algorithm to perform hand posture recognition. The hand posture recognition is an important task to perform the human-computer interaction. The hand is a complex object to detect and recognize. That is because the hand morphology varies from human to human. The object recognition community has developed several approaches to recognize hand gestures, but still, there are not a perfect system to recognize hand gestures under diverse conditions and scenarios. We propose a method to perform the hand recognition based on feature selection. The feature selection is performed by a genetic algorithm that combines several features to build a descriptor. The evolved descriptor is used to train a perceptron, which is used as a weak classifier. Each weak learner is used in the AdaBoost algorithm to build a strong classifier. To test our approach, we use a standard image dataset and the full image evaluation methodology. The results were compared with a state of the art algorithm. Our approach demonstrated to be comparable with this algorithm and improve its performance in the some of the cases.

1 Introduction

Hand gesture recognition is an important task to perform interactions with computers and robots. This is because the hand is the body part most used in the interaction between humans. The hand has proven to be the most challenging body part to be recognized.

Several approaches have been developed to recognize the hand gestures. There are approaches which are based on wearable sensors, where the user needs to use gloves or markers to perform the hand recognition. Other non-intrusive methods use 3D sensors and computer vision to perform this task. In this work, we focus on a computer vision based method to recognize the hand in video sequences.

There are many ways to recognize the hand using computer vision. The most used approaches in the literature to classify hand gestures is based on hand segmentation. That is because the hand can be segmented by using color information or 3D information.

J.F. Martínez-Trinidad et al. (Eds.): MCPR 2016, LNCS 9703, pp. 208–218, 2016.
DOI: 10.1007/978-3-319-39393-3_21

Some of the problems that arise in the methods based on hand segmentation are their sensitivity to illumination changes, a need for initialization step, the problems of the sensors (like Kinect) to work well in outdoor locations and a need for controlling the scenario to exhibit good performance. Conversely, object detection methods have proven to be an excellent alternative to detect objects in several types of scenarios (both indoor and outdoor environments).

The feature selection step is a core element in the object detection frameworks. Viola and Jones object detection framework [14] performs an exhaustive search in each round to find iteratively good features to classify the objects. To select a feature, they test a set of M predefined features in several sizes and positions in the image. The ranking of a feature is based on its discriminative power.

This feature selection approach has proven to be a powerful strategy to build a good classifier [7]. Because of the good results of this type of feature selection strategy, several works have been focused in proposing new methodologies to improve the selection.

To avoid the exhaustive evaluation of features, a common strategy is to define a fixed number of features to be tested [3]. Each feature is created randomly. The number of features is defined in most cases experimentally. Hidaka and Kurita [9] proposed the use of the Particle Swarm Optimization to perform the search of the features and reduce the learning time.

In contrast, Dalal and Triggs [5] proposed the use of the Histogram of Oriented Gradients (HOG) and a Support Vector Machine (SVM) to perform the feature selection in a pedestrian detection task.

They designed a descriptor manually by dividing the pattern into cells. Each cell is a rectangle of equal dimensions. The cells can also have an overlapping with others cells. The HOG is computed for each cell. The descriptor is the result of the concatenation of histograms. The resulting descriptor has a high dimensionality. To handle this, the authors propose the use of an SVM to perform the classification. This approach to building descriptors has been used in several works. The combination of HOG features and SVM has been widely used to detect and classify objects. Malisiewicz *et al.* [11] proposed the use of a predefined pattern to extract the HOG, taking account of the object size. Their work is focused on learning and classifying several object views. The design of an HOG descriptor is a complex task and it has a strong relationship with the structure of the object that we want to recognize.

In this work, we propose a learning method to deal with the hand posture detection problem. The method is based on performing an efficient search for a set of features. We use a genetic algorithm (GA) to perform this search, instead of manually designing a pattern. We use the standard AdaBoost algorithm as a learning method. We use a perceptron as a weak classifier. To obtain a better performance and reduce the number of false positives, we propose the use of hard negative sampling. We use the Full Image Evaluation framework for the performance evaluation of our approach. We also compare our results with the obtained by the real-time deformable detector [1].

The rest of the paper is organized as follows. In Sect. 2, we describe the proposed methodology to perform the hand detection. The performance of our system is presented in Sect. 3. Finally, some concluding remarks are given in Sect. 4.

2 Methodology

In this section, we describe the proposed methodology to recognize hand postures. We introduce the key elements of the proposed algorithm, and we explain its importance. Our proposed methodology is based on two central concepts: the feature selection step and the design of discriminative descriptors using HOG and variance features.

The feature selection step is the process to find a subset of d useful features from a finite set of features D [12]. The cardinality of the D set is too huge to use all the features at the same time. For this purpose, the GA offers an alternative to search for a subset of features in an efficient way [15].

The manual design of good descriptors involves the analysis of the pattern to classify to select the number of cells, the size, and their positions. The number of cells, the dimension of a cell and its position can be encoded as a candidate solution in the GA. The use of a GA instead of the manual design of a descriptor allows the reduction in the size of the descriptor used to classify the pattern.

We use a perceptron to classify the descriptor. The election of this classifier is based on the learning algorithm. We use AdaBoost as a learning algorithm.

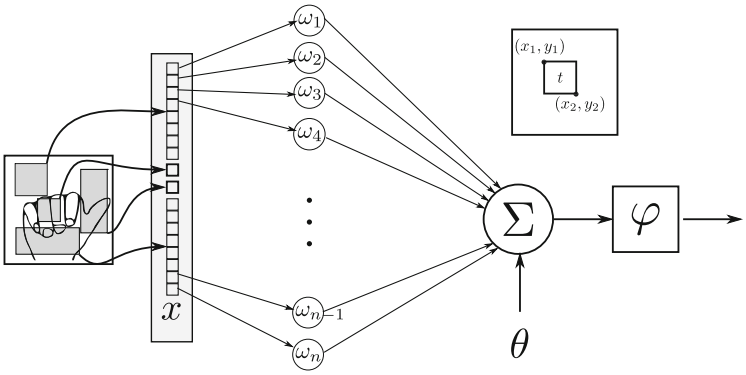


Fig. 1. Description of the proposed weak classifier. The descriptor x is composed of several features. A feature is defined by its type and the support area. The number of features and the types are determined by a genetic algorithm. The descriptor x is the input of the perceptron.

The proposed system is shown in Fig. 1. The result of the feature selection is a descriptor x. Each type of feature is composed of a different number of elements,

eight values for the HOG and one value for the variance. The descriptor x is the input of a perceptron. We combine the classification results of several weak classifiers to perform the object detection using AdaBoost algorithm.

2.1 Feature Selection

According to Lillywhite *et al.* [10], feature selection is the process of choosing a subset of features from the original feature space. This selection is based on an optimality criterion. The feature selection step is widely used in learning algorithms. In our proposal, we use two features based on the edge of a region and based on the variance. To select the features, we use a genetic algorithm. The genetic algorithm is useful to perform an efficient search to select the best set of features.

There are several proposed features in the literature: e.g. the Haar-like features proposed by Viola-Jones, the Histogram of oriented gradients (HOG), interest points, etc. All these features have proven their effectiveness as features in methods to detect and classify objects.

The real-time detection restriction imposed to perform a natural and fluent interaction with the robot requires, as a consequence, the need for fast computation of the selected features. To overcome this problem, Viola and Jones proposed the use of features based on integral images. An integral image is a representation that allows us to compute the sum of all elements in a rectangular area of the image in a fast and efficient way. We use two type of features, based on the integral image: the variance [4] and the HOG computed in eight orientations [1].

2.2 Genetic Algorithm

The genetic algorithm (GA) is a useful method to solve optimization problems. The GA performs a heuristic search, inspired by the biological evolution of species. The heuristic search uses a population, where each individual is a possible solution for the problem. A fitness function is used to evaluate the goodness of the individuals and there are three operators to simulate the evolutionary process. The GA operators are the selection, crossover, and mutation. Each of them is similar to the natural processes that appear during the evolution of the species.

We use the GA to find the best combination of features to build a descriptor. A feature is defined by a rectangle inside of the support area. The support area is a square that covers all the object to learn (the hand for our case).

In the GA, each candidate solution (individual) is represented by a string of bits, called a chromosome. The chromosome is divided into sets of bits. Each set is used to represent a variable of the solution. Our chromosome c is composed by quintuples of variables $c = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ where $\mathbf{x} = [t, x_1, y_1, x_2, y_2]$. Each tuple is used to describe a feature, t is used for the type of feature (edge map or variance) and $\{x_1, y_1, x_2, y_2\}$ for the rectangular area of the feature. The rectangular area is represented by two points, upper-left(x_1, y_1) and bottom-right(x_2, y_2).

The fitness function used in our approach is the weighted training error. The weighted training error is used for the AdaBoost algorithm to select the best classifier during the training procedure.

The GA crossover operation is used to obtain a new individual from the combination of two individuals. The point where the individuals are divided and recombined is randomly selected. For our approach, the new individual is the result of the combination of several 5-tuples $\mathbf{x}$. Using this crossover approach, the feature information is preserved across the generations. In the Fig. 2 the crossover operation is represented graphically.

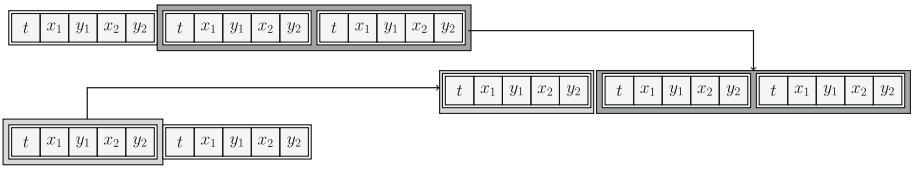


Fig. 2. The crossover operation combines two individuals in a new one. The points where the crossover is performing, are selected randomly. The crossover only combines tuples of values, to keep the feature information during the evolutionary process.

The GA mutation operation is used to modify the individual in a random way. These modifications introduce a new diversity in the individuals. This diversity is useful to explore the solution space. In our proposal, we only mutate the elements $\{x_1, y_1, x_2, y_2\}$. This mutation allows searching for the features in all the pattern.

The GA selection operation is used to retain the best individuals across the generations. There are several methods to perform the selection. We use the roulette method in our approach.

2.3 Learning Method

We use a standard AdaBoost learning algorithm to train our classifier. The AdaBoost is a method that combines the decision of several weak classifiers in a strong classifier. Each weak classifier is weighted according to his discriminative capability. The final decision combines all the results of the weighted classifiers. This weighted sum allows each weak classifiers to focus in different parts of the model to be learned.

The weak classifier used in our approach is a combination of features and the perceptron as a classifier. The perceptron is an Artificial Neural Network with only one layer. In each round, a GA algorithm performs a search for the best combination of features that minimizes the weighted error. The learning procedure is presented in the Algorithm 1. The value of the variable J is the number of individuals in the population.

Algorithm 1. Learning process

Given training data (x_i, y_i) $x_i \in \mathcal{X}$ and $y_i \in \{-1, 1\}$ Initialize weights $w_{1,i} = \frac{1}{2a}, \frac{1}{2b}$ where a and b are the total number of positive(hand posture images) and negative examples respectively (images that not contain hand postures). T is the number of weak classifiers. J is the number of individuals in the population

for $t = 1$ **to** T **do**

  for $j = 1$ **to** J **do**

    Perform the feature selection process and train the weak classifier h_j. Evaluate weighted classification error:

$$\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$$

  end

  Choose the classifier h_j, with the lowest error ϵ_j.

  Calculate:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_j}{\epsilon_j} \right)$$

  Update data weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

  where $e_i = 1$ if the image x_i is classified correctly and $e_i = 0$ otherwise, $\beta_t = \exp(\alpha_t)$.

  Normalize the weights

$$w_{t+1,i} \leftarrow \frac{w_{t,i}}{\sum_n w_{n,i}}$$

end

The final detector is given by:

$$f(x) = \text{sign} \left(\sum_{k=0}^{N} \omega_k h_k(x) \right)$$

2.4 Hard Negative Mining

The quality of the samples used in the learning setup is crucial to obtain a good classifier. An intuitive approach to get a good classifier is to increase the number of samples used during the training step. The increment of samples used during the training step implies an increase in the computer resources needed and also in the time spent to compute train the classifier.

To avoid these problems, it is preferable to use a small training set composed of useful samples. The process to obtain these samples from a larger dataset is called bootstrapping[1].

[1] In the statistical field, the bootstrapping process refers to a re-sampling method.

The bootstrapping methods are an essential component in different object recognition frameworks. To perform the bootstrapping, we need a sampling methodology and a classifier to evaluate the quality of the samples. The training set is constructed actively, during the training process or before the training process, using the hard negative mining (HNM) approach [2]. The selection of the bootstrapping methodology is based on the experimentation.

In our work we use HNM to improve the quality of the final classifier. The HNM allows constructing a small training set with relevant samples from a pool of images. This training set is built in two steps. First, a classifier f_1 is trained with a small set of images from the pool. The images are randomly sampled from the pool. The resulting classifier is then used to obtain higher quality negative samples.

The images in the negative training set, are sequentially evaluated using the classifier f_1. During this procedure, if a sample is misclassified then it is added to the training set. Then, using the new training set a new classifier f_2 is trained. The classifier f_2 is the final classifier.

3 Tests and Results

The results obtained from our proposal are presented in this section. This section is divided into two parts. The first part are the results obtained from the training process. We describe quantitatively the classifier resulting from the training process. In the second part the detection system is evaluated. The protocol used to evaluate our approach is the Full Image Evaluation, used by Dollár *et al.* [6]. This protocol is useful to obtain a fair comparison among the several approaches. We compare our approach results with those obtained by Hernandez and Ayala [8].

All test were performed in the GNU/Linux operating system using a general-purpose computer using 8 GB RAM and a processor running at 2.7 GHz. No parallel strategy of specific optimization was used in the implementation.

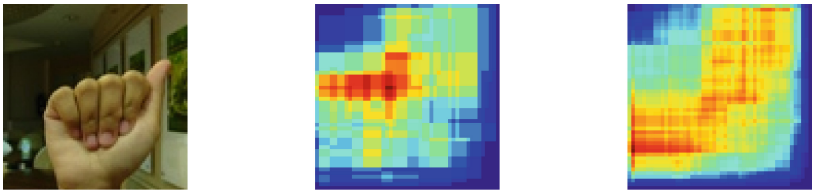
3.1 Training

The training procedure was performed using the AdaBoost learning algorithm and the Hard Negative Mining process. The number of weak learners used for the classifier f_1 was five, and the maximum number of weak classifiers for the second classifier f_2 was twenty. We used the National University of Singapore (NUS) hand posture dataset-II propose by Pisharady *et al.* [13] in the training and testing process. The NUS-II is divided in images where the hand posture appear alone (NUS dataset A), and where the hand posture appear with people in the scene with human noise(NUS dataset B). For this test we only use one posture.

The NUS dataset A was used for training purposes. The training samples were obtained by cropping the samples from the dataset and resizing them to be rectangles of 50×50 pixel size. We use this size in all our tests.

We use 1000 positive samples and 20000 negative samples in the training step. The parameters used for the GA were: crossover probability 0.8, mutation probability 0.01, two elite members, one hundred individuals in the population and fifty generations. The minimum number of features per individual was one, and the maximum number per individual was 10. These parameters have shown to produce the best results in the training step. Using these parameters, we train 45 classifiers. In each round of the AdaBoost training, the time consumed to find the best descriptor was 11.4175 min., with $\sigma = 0.0677$ min. The elapsed time to obtain f_1 y f_2 was around five hours (25 weak classifiers and HNM procedure). The performance obtained from the best classifier was 0.8900 for true positive classification, 0.1100 for false negative detection and 0.9452 for true negative classification. The mean time to process an image of 340×240 pixels was 96 ms (around 10 frames per second).

Figure 3 depicts the qualitative results for the HNM process using a classifier response heat map. This map is built using the weights of the weak classifiers and the rectangular area of the features. We use the best classifier obtained from the 45 trained classifiers. The object to learn is shown in the Fig. 3a. The first heat map obtained from the classifier f_1 is shown Fig. 3b. In this heat map, the weights of the weak classifiers are concentrated in the center of the pattern. In contrast, the heat map obtained from the classifier f_2 the weights are distributed along the pattern.



(a) (b) (c)

Fig. 3. The number and the type of features vary for each classifier. The heat map is a representation used to determine the areas where the classifier is focused. (a) hand posture (b) is the classifier f_1 before the HNM and (c) is the classifier obtained after applying HNM.

3.2 Full Image Evaluation

The full image evaluation is a methodology to measure the performance of the whole detection system. Using this methodology, the results of several methods can be compared fairly. To use this methodology, we need an annotated dataset. This annotated dataset was used as ground truth to perform the test. We use the ground truth used in [8]. The False Positive Per Image metric is computed $FPPI = FP/NI$, where NI is the number of image in the ground truth. The Miss Rate metric is computed $MR = 1 - TP/NO$, where NO is the number of

object in the ground truth. The results obtained from the full image evaluation are presented in the Fig. 4. We use the sliding window approach using a scanning step of 5 pixels in x and y axis and a scale factor of 1.4. Using the NUS Dataset II A (4(a)), our proposed approach exhibits a better performance. This is because, our approach has a better detection in all the range and the number of false positive is less than the RTDD approach. Using the NUS Dataset II A (4(b)) our results are similar to the RTDD approach. The number of false positive images detected by our approach is moderately greater that the RTDD approach. Nevertheless, the miss rate for our approach is greater that the RTDD. The full image evaluation results are promising. We conclude this because, in this test, we improve the performance of the results obtained by a state of the art method [8] for one hand posture.

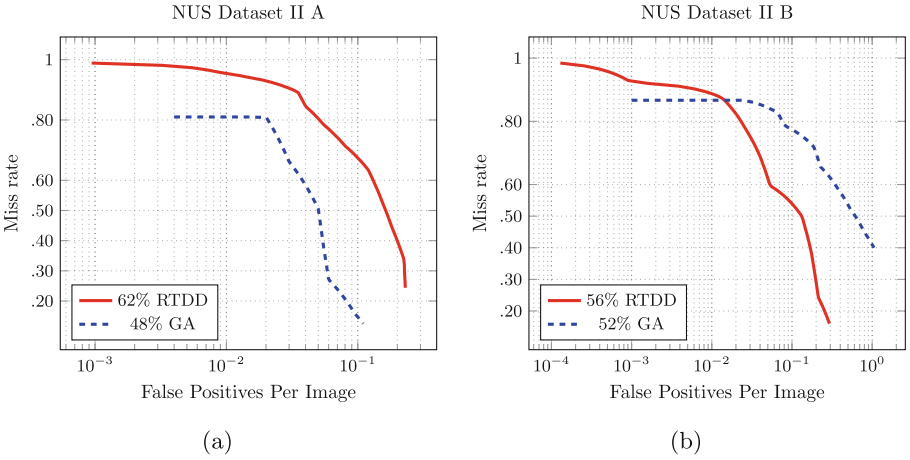


Fig. 4. Results obtained using the full image evaluation. (a) results using the NUS II dataset A, in this test our approach is better that the RTDD approach. (b) results using the NUS II dataset B, the performance of our approach is similar to RTDD. The percentage displayed in the plots is the area under the curve. The lower this value, the better the performance.

4 Conclusions and Perspectives

In this work, we proposed the use of a feature selection method for hand posture detection. The feature selection process was performed by using a genetic algorithm and two types of features. The proposed features were implemented using integral image computations, that are useful for fast computation. The genetic algorithm combines a different number of features and it varies their configuration. This combination and its variations allow to the genetic algorithm to find a good solution in a limited time. The results of the genetic algorithm have proven that this strategy is useful to build image descriptors. We used AdaBoost

algorithm to combine the response of several weak classifiers. A weak classifier is composed by the descriptor and a perceptron. The use of this type of weak classifier proved to be useful in the detection of hand postures. Our proposal was compared with a state of the art algorithm. The results of this comparison were favorables to our approach. From the results, we can say that the generalization ability of the classifier is good enough to detect hand postures. Future work will be to implement a CPU or GPU parallelization approach of the learning algorithm. This parallelization will reduce the time needed to train the classifiers. With a reduced training time, it is possible to perform experimentation about the influence of the training parameters in the resulting classifier.

Acknowledgments. The authors gratefully acknowledge to the Mexico's CONACYT (229784/329356) for the financial support through the scholarship is given by the programs "Convocatoria de Becas Nacionales 2013 Primer Periodo".

References

1. Ali, K., Fleuret, F., Hasler, D., Fua, P.: A real-time deformable detector. IEEE Trans. Pattern Anal. Mach. Intell. **34**(2), 225–239 (2012)
2. Canévet, O., Fleuret, F.: Efficient sample mining for object detection. In: Proceedings of the Asian Conference on Machine Learning (ACML), pp. 48–63 (2014)
3. Chen, Q., Georganas, N.D., Petriu, E.: Hand gesture recognition using Haar-like features and a stochastic context-free grammar. IEEE Trans. Instrum. Meas. **57**(8), 1562–1571 (2008)
4. Correa-Tome, F.E., Sanchez-Yanez, R.E.: Integral split-and-merge methodology for real-time image segmentation. J. Electron. Imaging **24**(1), 013007 (2015). http://dx.org/10.1117/1.JEI.24.1.013007
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005, vol. 1, pp. 886–893 (2005)
6. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**, 743–761 (2012)
7. Fürst, L., Fidler, S., Leonardis, A.: Selecting features for object detection using an adaboost-compatible evaluation function. Pattern Recogn. Lett. **29**(11), 1603–1612 (2008)
8. Hernandez-Belmonte, U.H., Ayala-Ramirez, V.: Real-time hand posture recognition for human-robot interaction tasks. Sensors **16**(1), 36 (2016). http://www.mdpi.com/1424-8220/16/1/36
9. Hidaka, A., Kurita, T.: Fast training algorithm by Particle Swarm Optimization and random candidate selection for rectangular feature based boosted detector. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IJCNN 2008, pp. 1163–1169, June 2008
10. Lillywhite, K., Lee, D.J., Tippetts, B., Archibald, J.: A feature construction method for general object recognition. Pattern Recogn. **46**(12), 3300–3314 (2013)
11. Malisiewicz, T., Gupta, A., Efros, A.: Ensemble of exemplar-SVMs for object detection and beyond. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 89–96, November 2011

12. Oh, I.S., Lee, J.S., Moon, B.R.: Hybrid genetic algorithms for feature selection. IEEE Trans. Pattern Anal. Mach. Intell. **26**(11), 1424–1437 (2004)
13. Pisharady, P., Vadakkepat, P., Loh, A.: Attention based detection and recognition of hand postures against complex backgrounds. Int. J. Comput. Vision **101**(3), 403–419 (2013)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: 2001 Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. 511–518 (2001)
15. Qian, C., Yu, Y., Zhou, Z.H.: Subset selection by Pareto optimization. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28, pp. 1774–1782. Curran Associates, Inc. (2015)