Improving Haar Cascade Classifiers Through the Synthesis of New Training Examples

João Correia
CISUC, Department of Informatics Engineering, University of Coimbra
Coimbra, Portugal
jncor@dei.uc.pt

Penousal Machado
CISUC, Department of Informatics Engineering, University of Coimbra
Coimbra, Portugal
machado@dei.uc.pt

Juan Romero
Faculty of Computer Science
University of A Coruña
Coruña, Spain
jj@udc.pt

ABSTRACT

A Genetic Programming approach for the improvement of the performance of classifier systems through the synthesis of new training instances is explored. Genetic Programming is used to exploit shortcomings of classifiers systems and generate misclassified instances. The proposed approach performs multiple parallel evolutionary runs to generate a large number of potentially misclassified samples. A supervisor module determines which of the generated images have been misclassified and which should be added to the training set. New classifiers are trained based on the original training set augmented by the selected evolved instances. The results attained while using face detection classifiers are presented and discussed. Overall they indicate that significant improvements are attained when using multiple evolutionary runs.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming; I.7.2 [**Pattern Recognition**]: Design Methodology

General Terms

Algorithms

Keywords

Genetic Programming, Machine Learning

1. INTRODUCTION

In a previous study [3] we presented a framework that combines a Genetic Programming (GP) image generation system with a state of the art face detector (FD). The GP engine evolved images that were incorrectly identified as faces by the FD. Later, these images were added to the negative dataset and the classifier was retrained. The experimental results obtained by the classifiers trained with a negative dataset augmented through the addition of images evolved in a single evolutionary run indicate that statistically significant performance improvements can be attained.

In the present paper, we focus on: (i) gathering individuals of several parallel evolutionary runs and augmenting the negative dataset with these new instances; (ii) exploring alternatives for the supervisor module responsible selecting which of the evolved images should be added to the negative dataset.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

2. THE FRAMEWORK

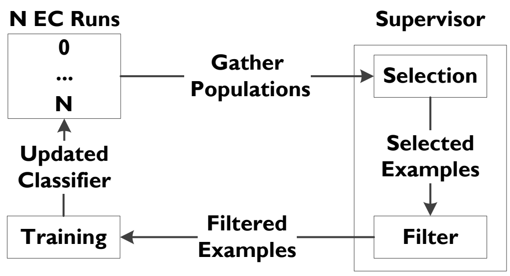

Figure 1: Overview of the framework.

A conceptual model of the framework is presented in figure 1. It is composed of three main modules: EC engine, Classifier and Supervisor. The process involves the following steps: (i) Selection of a positive and a negative training data set; (ii) Training a Classifier System (CS) using the positive and negative instances; (iii) N EC runs are started, with different random seeds; The CS classifies the generated individuals; (iv) Each EC run stops when a termination criterion is met; (v) The individuals generated throughout all EC runs go through the Supervisor module that selects and filters instances; (vi) The resulting sub-set of instances updates the negative image set; (vii) The process is repeated from step (ii) until the boosting criterion is met. A general purpose, expression-based, GP image generation engine that allows the evolution of populations of images is used as the EC engine. A thorough description of the GP engine can be found in [2]. The Classifier is a Haar Cascade classifier (see Viola et al. [5]) trained to detect frontal faces. This algorithm uses a set of small features in combination with a variant of Adaboost, and is able to attain efficient classifiers. The Supervisor module has the responsability of choosing the set of images to be added to the training set. This module comprises two parts, *select* and *filter*.

Select chooses a subset of the images evolved during the EC runs. In the experiments performed we considered two selection modes: *negative selection* and *FDLib selection*. In *negative selection* mode, all the images generated throughout the EC runs and identified as containing a frontal face by the CS used to guide the EC runs are selected. Implicitly this selection mode assumes that all evolved images classified as faces are classification errors, false positives. However, this is not necessarily true, some of the evolved images may actually look like faces and, as such, adding them to the training set may hinder performance. To cope with this potential problem we tested the *FDLib selection* mode. In this case the Face Detection Library (FDLib [1]) is used as an external classifier. The evolved images will only be selected if they are classified as containing faces by the CS guiding the EC runs and as **not** containing faces by FDLib, which gives us some assurance that they are indeed false positives.

The resulting sub-set of images is submitted to a filter operation to remove similar images. We considered two filtering modes: *equal* and *RMSE*. The first performs a pixel based comparison of the images subset, discarding images that are duplicated, discarding images that are duplicated. RMSE mode, also performs a pixel based comparison, calculating the root mean square error between all pairs of images of the sub-set, and discarding images that are bellow a given RMSE threshold.

3. EXPERIMENTAL RESULTS

The CS used to guide the runs was trained with a dataset of 1905 positive and 1905 negative images. We performed 30 independent EC runs, with a length of 100 generations each and a population size of 50. The remaining experimental parameters can be consulted in [3].

After the EC runs were completed, the images evolved through the 30 runs were gathered and submitted to selection and filtering. The resulting set of images was added to the negative dataset and the CS was retrained.

To assess the performance of the classifiers in test data we considered three independent datasets: Flickr – 2166 negative images; Feret – 902 positive images from Feret Database (http://face.nist.gov/colorferet/colorferet.html); CMU-MIT – 130 positive and negative images [4]. The performance is measured in terms of total hits (H), misses (M), false alarms (FA), and percentage of correctly classified instances (%C).

Table 1 presents a synthesis of the results obtained in the validation datasets by the considered classifiers: Initial – the classifier used to guide the evolutionary runs; FDLib – the external classifier used in supervision; IEC avg. – the average performance attained by retraining the classifiers using a negative dataset expanded by adding images of a single evolutionary run (see [3]); Manual – the classifier resulting from augmenting the negative dataset by hand-picking the evolved images to add; The classifiers resulting from the four possible combinations of selection and filtering modes.

There is an improvement of performance over the *Initial*, *FDLib*, and *IEC* classifiers in terms of %C and FA. On the downside, generally, there is small decrease on the number of hits and a corresponding increase in the number of misses, which is an expected result. Nevertheless, *Negative Equal* outperforms the *Initial* classifier in all metrics. The classifiers resulting from *FDLib selection* attain the lowest FA rates and obtain %C's that are only surpassed by *Negative Equal*. *Manual* supervision was outperformed by 3 out of 4 of the automatic supervision modes in terms of %C. Overall, the results suggest that is possible to achieve performance improvements through the evolution of new training instances without human supervision.

Table 1: Results obtained by the classifiers in validation datasets. Performance improvements over the initial model and IEC models are presented in bold typeface. Improvements over the initial model in italic. Decreases of performance over the initial model are underlined.

Classifier	H	M	FA	avg(%C)
Initial	1045	168	1018	69.2
FDLib	821	392	565	63.5
IEC avg.	1040	173	884	71.2
Manual	<u>1043</u>	<u>170</u>	**543**	**74.7**
Negative sel. Equal	**1062**	**151**	**615**	**76.4**
Negative sel. RMSE	<u>1037</u>	<u>176</u>	*930*	*71.1*
FDLib sel. Equal	<u>1022</u>	<u>191</u>	**488**	**75.7**
FDLib sel. RMSE	<u>1018</u>	<u>195</u>	**409**	**75.6**

4. CONCLUSIONS

An evolutionary framework for the improvement of classifier's performance is revisited. We focused on gathering individuals from several parallel evolutionary runs, and on the development of a supervisor module, responsible for selecting and filtering the evolved instances that should added to the training set. The approach was tested in three validation datasets and brought significant improvements of performance. In terms of future work, we plan to further refine the supervision module and perform several boosting iterations.

5. ACKNOWLEDGMENTS

This research is partially funded by projects: PTDC/EIA–EIA/115667/2009, Portuguese Foundation for Science and Technology; TIN2008–06562/TIN, Spanish Ministry for Science and Technology; XUGA–PGIDIT10TIC105008PR, Xunta de Galicia.

6. REFERENCES

[1] W. Kienzle, G. H. Bakır, M. O. Franz, and B. Schölkopf. Face detection — efficient and rank deficient. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 673–680. MIT Press, Cambridge, MA, 2005.

[2] P. Machado and A. Cardoso. All the truth about NEvAr. *Applied Intelligence, Special Issue on Creative Systems*, 16(2):101–119, 2002.

[3] P. Machado, J. Correia, and J. Romero. Improving face detection. In A. Moraglio, S. Silva, K. Krawiec, P. Machado, and C. Cotta, editors, *Genetic Programming*, volume 7244 of *Lecture Notes in Computer Science*, pages 73–84. Springer Berlin / Heidelberg, 2012.

[4] H. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1998.

[5] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I–511–I–518 vol.1, Hawaii, 2001.