Evolved Feature Ensemble Learning

Adam Gaier & Alexander Hagg
Bonn-Rhein-Sieg University of Applied Sciences
Master Course on Autonomous Systems
Grantham-Allee 20, 53757 Sankt Augustin
Germany

Sankt Augustin, June 16, 2014

1 Introduction

In this project we replicate the work of Lillywhite et al. ([3]) in his work to automatically construct image features for image classification using evolutionary techniques. We will present the system as was described by the author, along with criticisms and proposed improvements.

2 State of the Art

Lillywhite et al. ([3]) attempt to automatically construct image features for object recognition by using an evolutionary algorithm. This has been done for feature selection and extraction before (e.g. by [4]).

They create individuals, consisting of both a varying-length number of image transformations (chromosome), applied to both a labelled training set (sensory input, knowledge about the world) and a test set. The individual also contains a simple perceptron, connecting to all pixels in the input space, which enables weak classification. A weak classifier is defined as a classifier that only needs to be better than a random guess.

An evolutionary algorithm is then constructed, that uses the perceptron's degree of misclassification as a fitness function to a process of recombination and mutation of all individuals.

After n generations, the process is stopped and the weak classifiers are filtered using another fitness function, one that is more stable against badly weighed training/testing sets (see figure 3). The classifiers are then combined using Adaboost (short for Adaptive Boosting), as defined in [1] and later in [5].

The paper uses regions of interest as part of the chromosome chain, but were only able to show effects on computations by a factor of 1.75. They did not provide any insight which pixels of the regions of interests were actually used by the perceptrons for learning.

The image databases used in the paper were a bit out of date and might have been too simple for comparison against other state of the art approaches.

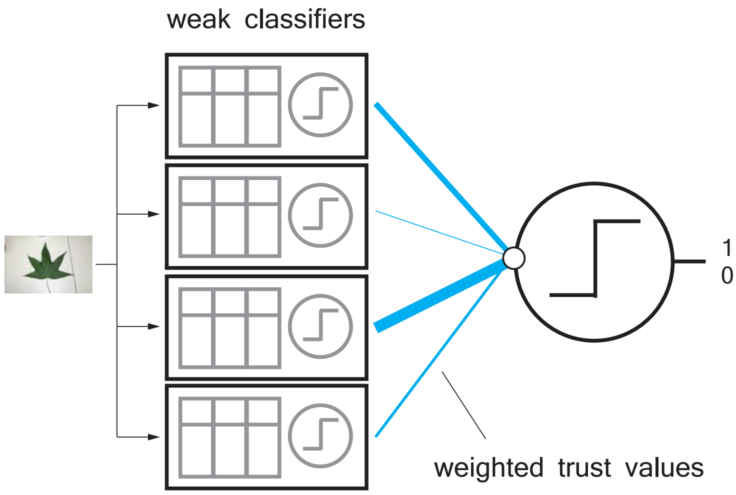
3 Approach

- *System Overview*

  The system which we produce uses evolutionary techniques to optimize sets of image transforms. These transformations form a preprocessing step for individual weak classifiers which are then combined and weighted via Adaboost.

  When a set of images is read in, they are altered according to a set of transforms (ie. gaussian blur, canny edge detector) defined by an individual's genome. Included in the set of possible transforms was a *NoOp* transform, allowing individuals using less than the maximum number of transforms to be evolved. An associated single perceptron is then trained to classify the altered image. Once trained these weak classifiers are combined through the Adaboost algorithm to form a strong classifier.

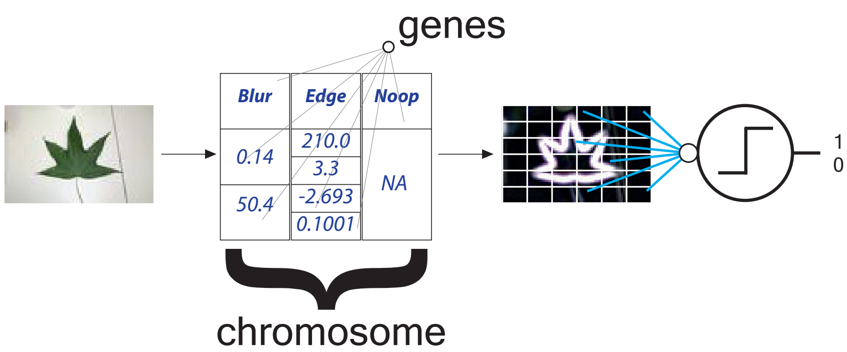
Figure 1: Evolved Feature Ensemble Learning



- *Individual Representation*

  An individual is composed of two parts: a sequence of transforms, and a single perceptron (see Figure 2). For every image the set of transforms, and their associated parameters, are applied to the image in series, i.e. with the example sequence shown below (Blur, Edge, NoOp) the image is first blurred and then edges are found of the blurred image. The attached perceptron is trained on the set of transformed images, with each pixel as one input to the perceptron. Fitness (s) is defined by the performance of the perceptron in classifying a hold out set 3.

Figure 2: Representation of Individual



As we chiefly require only a general notion of quality by which to direct evolution we use the normalized perceptron training rule as it converges in a minimal number of epochs, even if it is not always possible to reach the same levels of error. As we are performing

Figure 3: Fitness Function

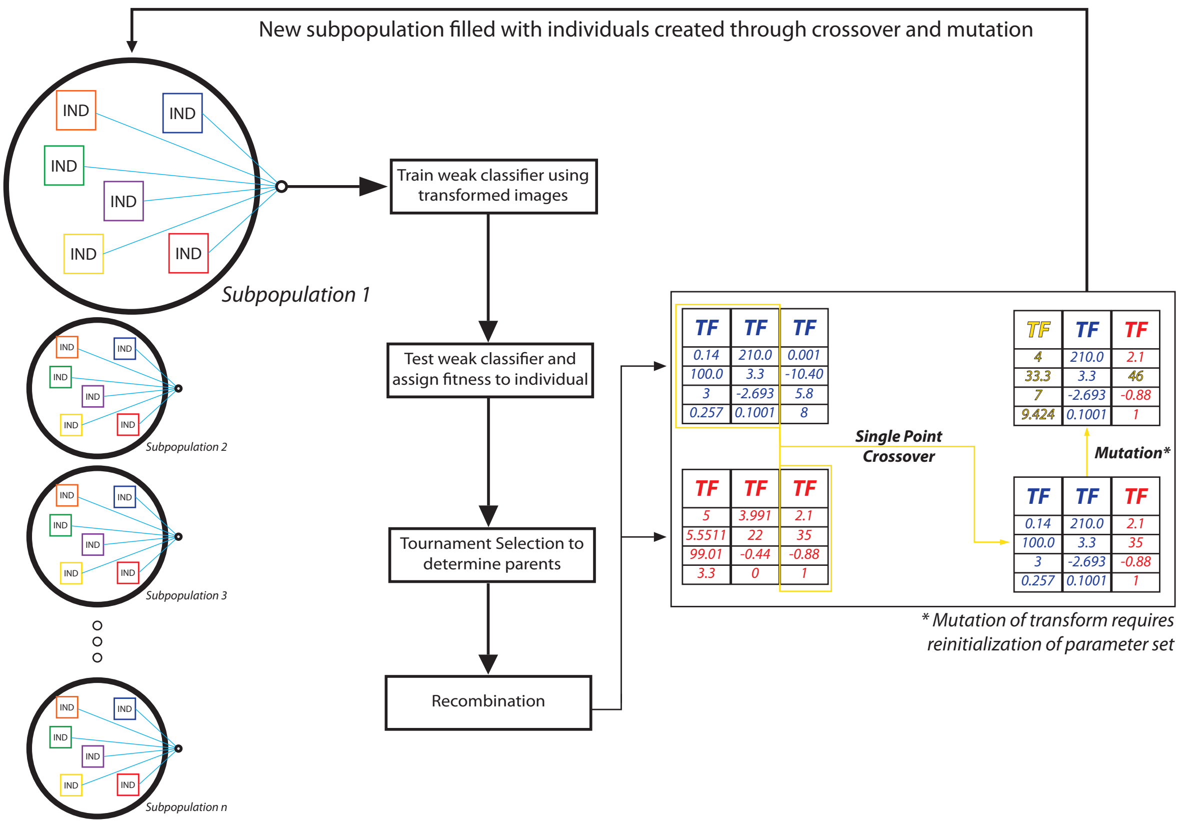
$$s = \frac{t_p}{f_n + t_p} + \frac{t_n}{f_p + t_n}$$

training over a large number of individuals through a large number of generations, this faster, less precise training procedure is appropriate.

- *Evolutionary Algorithm*

  Figure 4 illustrates the evolutionary process. Individuals are evolved in parallel subpopulations to encourage the diversity of solutions required by Adaboost. Selection is performed via tournament selection with elitism, with a selection pressure set to 10% of the total subpopulation size. We employ single point crossover when producing new individuals, with mutation applied to either the parameters of the transforms or the type of transform itself. When the transform is mutated, as the number of associated parameters often differs, and almost always hold different meanings, all of the parameters are reinitialized randomly.

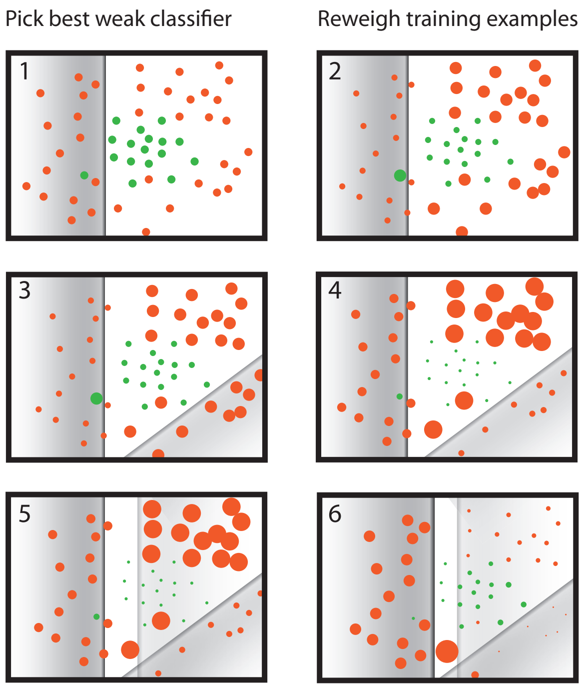
Figure 4: Evolution of Individuals



- *Adaboost discrete ensemble learning*

  As perceptrons are useable feasible for linearly separable data, they only serve as weak classifiers. Fusing their knowledge together into one strong classifier is done by ensemble learning using the Adaboost algorithm ([1]) for binary classification problems. Figure 5 shows this process. The training samples are weighted equally in the beginning. Now the best separating weak classifier is taken and given a credibility weight, based on its fitness over the training set. Now the training samples are reweighed. Samples that were classified incorrectly by the first weak classifier are given a larger weight and vice versa. This way the algorithm forces a focus on wrongfully classified training samples. Adaboost tries to maximize the weak classifier's margin, which represents the weighted difference of correctly and incorrectly classified training samples. In the next round, the perceptron that maximizes the weighted classification margin is taken into the ensemble, correcting some of the earlier mistakes, but also adding new ones. Adaboost was shown to converge to an upper error bound ([6]).

Figure 5: Adaboost



4 Evaluation

4.1 Experiments and Results

- *Data Set*

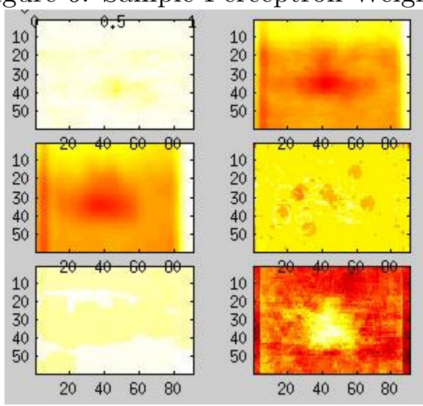
  We evaluated our system initially on an admittedly simple data set. 256 90X59 pixel images, half of leaves and half of airplanes, both on noisy backgrounds, were classified as belonging to either class. Other free parameters were as follows:

  - $generations = 200$
  - $size\ of\ each\ subpopulation = 20$
  - $number\ of\ subpopulations = 10$
  - $transforms\ per\ individual = 5$
  - $mutation\ rate = 0.1$
  - $crossover\ rate = 0.85$

In figure 7 we see the results of four separate runs (below) and the mean results (above). In every run a strong classifier was produced at some point during the run which performed with a 100% recognition rate on the test set.

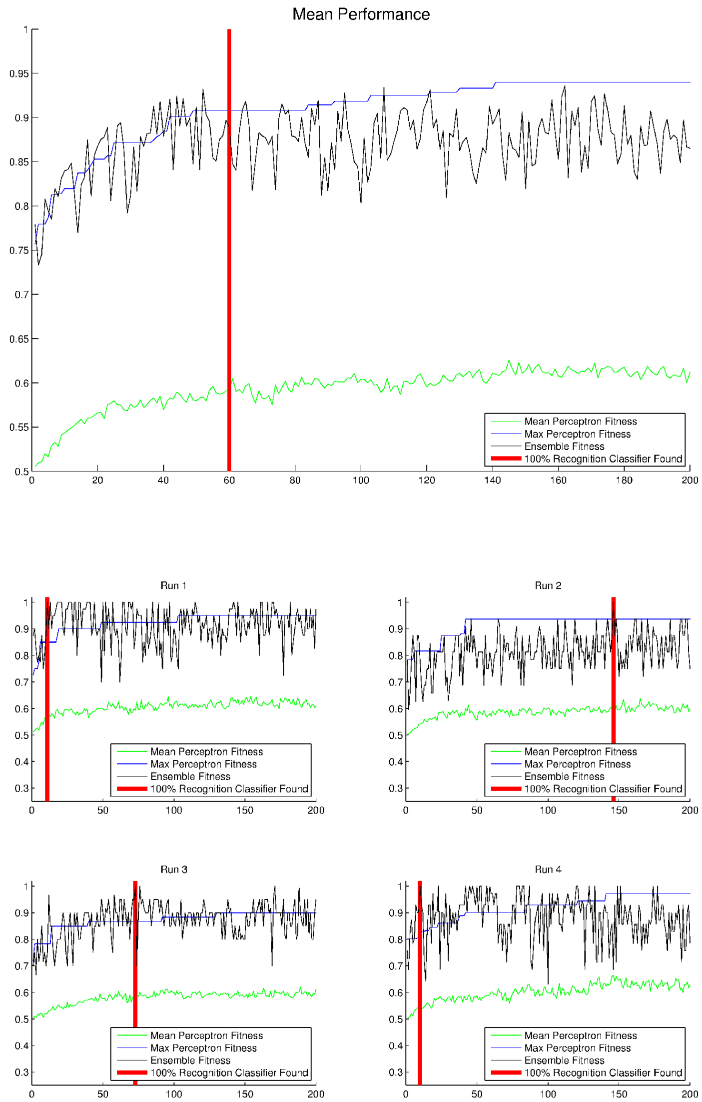
In previous work, perceptrons are trained over a number of generations and then at the end of the evolutionary optimization the perceptrons are combined with Adaboost to form a strong classifier. The performance of the resulting strong classifier varies on the particulars of the individual weak classifiers of which it is composed. In Figure 6 we see the resulting weights of some high performing perceptrons, with high weight values associated to areas in which leaf shapes commonly appear.

Figure 6: Sample Perceptron Weights



That the performance of the strong classifier varies so wildly belies a fundamental disconnect in the overall optimization process. The single perceptrons are optimized through evolution to perform object recognition independently. This goal, though certainly desirable, does not match the goal of the algorithm as a whole: to construct an effective strong classifier. We will discuss ways which we plan to address this problem at the core of the algorithms operation below.

Figure 7: Algorithm Performance



5 Conclusion

5.1 Future Work

- *Optimizing for effective ensembles, rather than individual weak classifiers*
  As mentioned above, there is a disparity between the fitness function used by the EA, rewarding the performance of individual weak classifiers, and the algorithms goal of producing a strong classifier. One of the first improvements we suggest would be to include the performance of the strong classifier into the evaluation loop. One approach would involve sampling the population, taking different individuals at random and composing a strong classifier from them using the same Adaboost techniques. Individual fitness would then be awarded based on the average of performance of all strong classifiers which the individual was a part of. In this way those perceptrons with high classification rates by themselves would still recieve high fitness, as the ensembles which they were part of would perform well, and those individuals which were able to fill in the 'blind spots' of these high performing individuals would also be highly rewarded. Such techniques, which reward individuals which cooperate effectively, has proven successful in evolutionary optimization of recurrent neural networks, structures even more highly connected then Adaboost ensembles ([2]).

- *Speciation*
  As diversity in weak classifiers is important to produce an effective ensemble with Adaboost, efforts must be made to prevent a population-wide convergence on a single high performing solution. This was done by isolating subpopulations, and hoping that they converged on different solutions. While the above ensemble-wise fitness calculations will do something to discourage convergance on a single solution (via game theory), a more elegant method of speciation is required. As the number of parameter combinations is quite large, a simple euclidean distance between them is unrealistic. A system which tracks the heredity of individuals may be more realistic. At the least, the size of subpopulations should not be kept constant, but vary in size to encourage diversity.

- *Generalized filter kernels*
  The use of standard OpenCV image transforms seems to confine the search space of the evolutionary algorithm. Many image operators can be represented as a linear system of convolved square kernel matrices. This representation would allow a search in a bigger space of filters, as a kernel can be represented by a set of parameters, representing the kernel matrix and its anchor point. On top of that, a single representation of transform operations would make it easier to specify the distances between transforms, making speciation a lot easier and dynamic. The current approach uses an "unnatural" kind of speciation, where a subpopulation is not allowed to interact with others. A distance measure could form the basis of a more natural way of judging, whether two individuals belong to the same species.

  Furthermore, the use of a single kernel representation would allow the system to work independent of the dimensions of the domain, as a 1D filter kernel could form the basis, and depending on the dimensionality of the domain, it could be convolved with n 1D filter kernels. An exploration of this approach, all be it an addition to or replacement of the current set of transforms, will be performed.

- *Other domains*
  The application of automated feature construction in other areas, such as voice recognition or face recognition, will be investigated.

- *Multimodal approaches*
  In robotics, data from multiple sensors (RGB, depth) used for a certain application such as object recognition are usually fused using hand crafted sensor fusion schemes. Using multiple sensor spaces as an input, it would be interesting to see if single-domain features as well as crossmodal features could be found using our approach.

References

[1] Freund, Y., and Schapire, R. A desicion-theoretic generalization of on-line learning and an application to boosting. *Computational learning theory* (1995), 1–34.

[2] Gomez, F., and Miikkulainen, R. 2-D pole balancing with recurrent evolutionary networks. *ICANN 98* (1998), 2–7.

[3] Lillywhite, K., Lee, D., Tippetts, B., and Archibald, J. A feature construction method for general object recognition. *Pattern Recognition 46*, 12 (2013), 3300–3314.

[4] Neshatian, K., Zhang, M., and Johnston, M. Feature construction and dimension reduction using genetic programming. *AI 2007: Advances in Artificial ...* (2007), 160–170.

[5] Rudin, C., Daubechies, I., and Schapire, R. The dynamics of adaboost: Cyclic behavior and convergence of margins. *The Journal of Machine Learning ... 5* (2004), 1557–1595.

[6] Schapire, R. The Convergence Rate of AdaBoost. *COLT* (2010), 2004–2005.