Visual Routine for Eye Detection Using Hybrid Genetic Architectures

J. Bala*, K. DeJong, J. Huang, H. Vafaie, and H. Wechsler

Computer Science Department
George Mason University
Fairfax, VA 22030

*Datamat Systems Research, Inc.
8260 Greensboro Dr.
Mclean, VA 22102

Abstract

We address the problem of crafting visual routines for detection tasks. Emphasis is placed on both competition and learning to help with specific visual tasks involved in localization and identification. Crafting of visual routines presents difficult optimization problems and leads to evolutionary computation using a hybrid genetic architecture consisting of natural selection, learning, and their beneficial interactions. Base features representations and visual routines for detection represented as decision trees are evolved. The visual routine considered is that of eye detection. The experimental results reported herein prove the feasibility of our approach in terms of feature selection ('data compression') and the corresponding eye detection ('pattern recognition').

1. Introduction

Active vision (AV) has advanced the widely held belief that intelligent data collection rather than image recovery and reconstruction is the goal of perception. AV, also referred to as Active, Purposive and Selective Perception (APSP), involves a large degree of adaptation, and provides a mobile and intelligent observer with the capability to decide *where* to seek information, *what* information to pick up, and *how* to process it. APSP is a process of intelligent control applied to both data acquisition and processing, and its specific ('reactive') activation depends on the current state of data interpretation.

One possible explanation for how one could implement active processes as suggested above comes from the concept of visual routines suggested by Ullman [1] and most recently referred to as a visual routine processor (VRP) by Horswill [2]. The VRP assumes the existence of a set of visual routines that can be applied to base image representations subject to specific functionalities, which are driven by the task at hand and are encoded through centralized control. The relevant question for VRP is how to actually craft such visual routines. Early attempts, developed using manual design, involved simulations lacking low-level ('base') representations entirely and operating on bitmaps only [3]. Ramachandran [4], an advocate of the utilitarian theory of perception, has suggested as an alternative that one could craft such visual routines by evolving 'bag of <perceptual> tricks' whose survival is dependent on functionality and fitness. This approach, which can be directly traced to the earlier 'neural Darwinism' theory of neuronal group selection as a basis for higher brain function, and it is reviewed by Edelman [5], suggests natural selection as the major force behind the design of visual routines. Most recently, Johnson et al [6] have suggested using Genetic Programming [7] for evolving visual routines whose perceptual task is finding the left and right hand of a person in a black and white silhouette of a person. Restrictions are placed on this problem by presuming that some other universal routines has already detected the largest connected component and returned its bounding box and centroid.

This paper expands on the approach taken by Johnson, Maes and Darell [6] for evolving visual routines by (*i*) using base features representations and visual routines for (eye) detection represented as decision trees, and (*ii*) by providing proof of concept using real gray level facial images rather than bitmaps. Rather than using genetic programming (GP) as Johnson et al do, we use genetic algorithms (GAs). The reason for using GAs rather than GP stems from our belief that what is needed is a compact way to represent a successful problem-solving strategy rather than evolving the elaborate program for implementing the same strategy. The experimental results reported prove the feasibility of our approach in terms of feature selection ('data compression') and the corresponding eye detection ('pattern recognition').

Face recognition starts with the detection of face patterns, proceeds by normalizing the face images to account for geometrical and illumination changes, possibly using information about the location and appearance of facial landmarks such as the eyes, identifies the faces using appropriate classification algorithms, and post processes the results using model-based schemes and logistic feedback. Detecting the eyes serves an important role in face normalization and facilitates further localization of other facial landmarks.

Any object or pattern that has to be detected should possess a number of discriminatory properties or features. The first step in any detection process, performed either by a machine or by a human being, is to choose discriminatory features and to measure them. Feature selection concerns the extraction of characteristic features or attributes from raw input data and their eventual data compression leading to reduced dimensionality pattern vectors. The number of features needed to successfully perform a visual routine task, such as eye detection, depends on the discriminatory qualities of the selected features.

2. Hybrid Genetic Architectures

The first problem one has to address in the context of developing visual routines is what form the base representations ('features') the visual routines will be call to operate on should assume. One needs not only to derive optimal features but also to find the means to integrate them for building optimal classifiers. This problem requires searching over a large space of non-linear and higher-order combinations and it is usually prohibitively expensive. As the search space defining both the features themselves and their possible ('data fusion') integration is exponential in its complexity, one comes to employ alternative methods, such as those underlying natural selection and evolution. Such methods are often driven by statistical considerations using information theory concepts such as the Kullback divergence measure and maximum information preservation principle (infomax). As an example, Linsker [8], evolved receptive fields, characteristic of center surround cells and/or orientation columns, that maximize their output activity variance. One of the goals of this paper is to illustrate the feasibility of deriving appropriate subsets of features and then integrating them for detecting facial landmarks such as the eye region using decision trees. Note that the derivation of decision trees, as discussed later in this paper, is driven by the infomax principle mentioned above.

Genetic algorithms (GAs) mimic what nature has done all along and it does that using similar principles. GAs are further defined when one provides a specific strategy for choosing the offsprings and/or the next generation. Simulated breeding is one of the possible strategies where offsprings are selected according to their fitness. Note also that simulated breeding is conceptually similar to stochastic search in general, and to simulated annealing in particular, for the case when the size of the offspring population is limited to one individual only.

Our approach expands on the one proposed by Johnson et al (1994) for crafting visual routines by choosing a hybrid genetic architecture, which consists of both evolution and learning (decision trees). Hinton and Nowlan [9] were among the first to consider computational ways in which evolution of species and the learning of individuals might interact. Our decision for implementing the learning component of the hybrid genetic approach using decision trees has been motivated by them incorporating the infomax principle characteristic of developmental processes as discussed earlier.

3. System Architecture

The architecture for the derivation of base representations ('feature selection') and the visual routines operating on them ('detection of facial landmarks') integrates genetic algorithms (GAs) and inductive generalization, characteristic of learning from examples. Fig.1 illustrates our general approach. An initial but large list of low-level image features is provided together with the images from which the feature vectors ('examples') for both positive (eye) and negative (non-eye) classes are extracted. The GA is used to explore the space of all subsets of the given feature list where preference is given to those features subsets which achieve best detection performance using small dimensionality feature sets ('Occam's razor'). Each of the selected feature subsets is evaluated - its fitness measured - using the induction of decision trees. The above process is iterated along evolutionary lines and the best feature subset found is then recommended to be used in the actual design of the corresponding facial landmark detection system. Note that the examples referred in Fig. 1 are represented by all the features from the original list, while the training and tuning data consists of only those features passed through the feature subset controller as a result of natural selection.

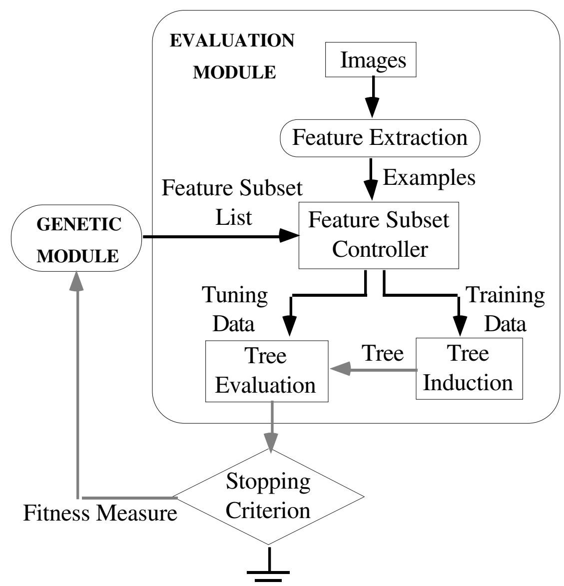

Figure 1. Hybrid Architecture for Detection of Facial Landmarks

3.1. Genetic Algorithms

Genetic algorithms (GAs) are adaptive search techniques initially introduced by Holland [10]. Genetic algorithms typically maintain a constant-sized population of individuals ('chromosomes') which represent samples from the space to be searched. Each individual is evaluated on the basis of its overall fitness with respect to some prespecified functionality and across some application domain. New individuals (samples from the search space) are produced by selecting high performing individuals to produce "offspring" which retain many of the features of their 'parents'. The result is an evolving population that has improved fitness with respect to the given functionality ('goal').

New individuals (offsprings) for the next generation are formed by using two main genetic operators, crossover and mutation. Crossover operates by randomly selecting a point in the two selected parents gene structures and exchanging the remaining segments of the parents to create new offspring. Mutation operates by randomly changing one or more components of a selected individual and it acts as a perturbation operator to allow for inserting new information ('genetic material') into the population. Mutation prevents any stagnation ('premature convergence') that might occur during the search process. The main issues in applying GAs consist of choosing an appropriate base representation and an adequate evaluation function.

The first step in applying GAs to the problem of feature selection is to map the search space into a representation suitable for genetic search. Since the main interest is in representing the space of all possible subsets of the original feature list, the simplest form of base representation considers each feature as a binary gene. Each individual is then represented as a fixed-length binary string standing for some subset of the original feature list. An individual corresponds to an N dimensional binary feature vector X, where each bit represents the elimination or inclusion of the associated feature. The advantage of this representation is that the classical GAs operators described before (binary mutation and crossover) can be easily applied to this representation without any modification.

The choice of an appropriate evaluation procedure is an essential step for the successful application of GAs to any problem domain. Evaluation procedures provide GAs with feedback about the fitness of each individual in the population. GAs use this feedback to bias the search process and improve the population’s average fitness. Fitness evaluation becomes available once performance of trees being induced is assessed. For the experiments reported later on, both parents and offspring compete to be included in the next generation.

3.2. Decision Trees and Fitness Evaluation

The ID3 algorithm is the most commonly used algorithm for the induction of decision trees [11]. The ID3 algorithm, in an iterative fashion, determines the feature which is most discriminatory and then it dichotomizes (splits) the data into classes categorized by this feature. The next significant feature of each of the subsets is then used to further partition them and the process is repeated recursively until each of the subsets contain only one kind of labeled data. The resulting structure is called a decision tree, where nodes stand for feature discrimination tests while their exit branches stand for those subclasses of labeled examples satisfying the test. An unknown example is classified by starting at the root of the tree, performing the sequential tests and following the corresponding branches until a node is reached indicating that some class has been decided on. Decision trees are disjunctive, since each branch leaving a decision node corresponds to a separate disjunctive case.

The learning strategy, conceptually akin to cross-validation (CV) and bootstrapping, draws randomly from the original list of examples, both positive and negative, and generates two labeled sets consisting of training and tuning data. The training data is used to induce trees, while the tuning data is used to evaluate the trees and generate appropriate fitness measures. Once the evolution stops the best tree is frozen and it is used for testing on unseen facial imagery to assess on how well the visual routine performs on eye detection.

The decision tree induction process (ID3) is applied to *training data* and it generates decision trees for each of the given classes. The evaluation procedure assesses the fitness of each feature subset, as given by its corresponding induced tree, using now *tuning data* sets and employing an integrated measure consisting of both the number of features used and the error rate. This measure of fitness is then passed to the Genetic Module stage to further evolve optimal candidate feature subsets. The whole process is initialized by randomly selecting subsets of feature and making them available to ID3. Note that detection of the eyes, respectively, is successful if the corresponding region is reached. For the case one is interested to detect only the center of the landmark region, penalty factors (*i*) multiple detections, and (*ii*) the distance between the center of the landmark and the actual detection, should be also included.

A description generated by a decision tree induction process such as ID3, is both descriptive and procedural. The descriptive representation is used to evaluate a given subset of features by measuring the error rate on the tuning data. The procedural representation describes the way in which a given subset of feature should be applied (i.e. not only it describes the feature subset itself but also an order in which the features should be applied). Since the ID3 learning process determines the feature which is most discriminatory in the recursive process of dichotomizing the training data, the procedural representation is inherently embedded in the decision tree. This procedural descriptiveness of the evaluation process is the main motivation to use decision trees. In order to implement the Genetic Module and ID3 we use GENESIS (GENEtic Search Implementation System [12], and C4.5, respectively.

4. Eye Detection - Experimental Results

This section describes the data used, the original list of possible features, and the detection architectures used to train and test the eye detection routine. The facial images are of 64x64 resolution - Fig. 2 shows examples of facial images used in our experiments. The data base (DB) consists of 120(+) eye examples. As training involves negative examples as well, DB also includes 480(-) non-eye examples drawn randomly across the facial landscape. The resolution for both the positive and negative eye examples is 24x16. Once training is completed exhaustive testing ('search') of the detection routine learned is performed across 58 unseen facial images. Testing takes place as 24x16 search windows are shifted four pixels to cover the entire facial image. Each test image generates 143 examples the eye detection routine is asked to decide on. The detection strategy includes cross validation (CV) and winner-take-all (WTA).

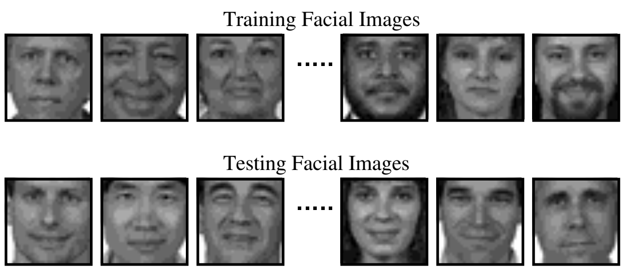

Figure 2. Examples of Facial Images

4.1. Facial Features

The feature list, consists of one hundred and forty seven features $\{x_1\ x_2,..., x_{147}\}$. The set of features, measured over 6x4 windows and using two pixels overlap, are:

- x_1 to x_{49} - means for each window
- x_{50} to x_{98} - entropies for each window
- x_{99} and x_{147} - means for each window after applying the Laplacian over 24x16 frames

Fig. 3 illustrates the feature extraction process.

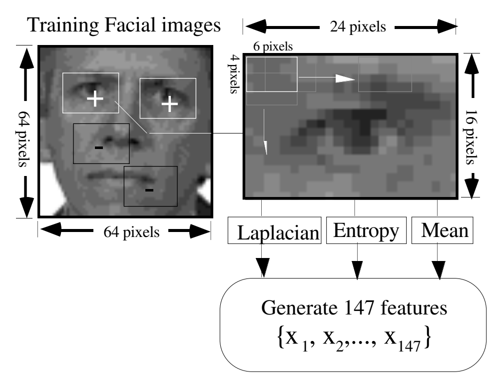

Figure 3. Feature Extraction

4.2. Eyes Detection

The experiment carried out to assess the feasibility of our hybrid genetic approach for eye detection using the general architecture described in Sect. 3. Each generation keeps a constant population size of fifty individuals, the crossover rate and mutation rate are 0.6 and 0.001, respectively.

The set of six hundred examples, 120(+) eye examples. and 480(-) non-eye examples, is divided into three equal subsets for cross validation training and tuning, and a tournament consisting of three sets of CV rounds takes place. The corresponding error rates, including both false positive - false detection of eyes - and false negative - missing the eyes, on tuning data, are shown as fitness measures in Fig. 4. The feature subset corresponding to the tree derived from the third CV round, which achieved the smallest - 4.87% - error rate, consists of only 60 of the original 147 features. This feature subset is used to evaluate the performance for eye detection on unseen test examples.

Table 1 gives the average confusion matrix for the eye detection task. These statistics were obtained by visually inspecting for the detection of eye regions. Fig. 5 shows examples for eye detection.

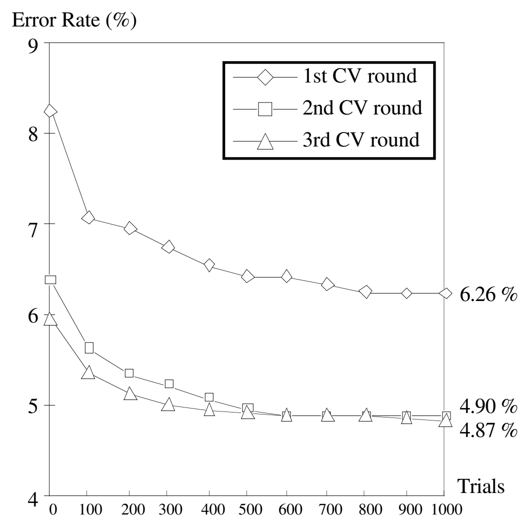

Figure 4. Fitness Measures for Learning on DB.

Ground Truth / Test Decision	Eye Regions	Non-Eye Regions
Eye	95.69 %	4.31 %
Non-Eye	6.03 %	93.97 %

Table 1. Confusion Matrix for Eye Detection on DB

The experiment performed so far detects clusters of candidates. As a consequence post processing is warranted and in our case WTA (Winner Take All) is used. The WTA procedure clusters adjacent candidates, find their corresponding centers, and filters out those who fail pairwise distance constraints, like those holding between the two eyes. Correct performance results for the case when the center of the cluster is within four pixels from the true eye location. The motivation for using the four pixel distance comes from testing taking place over 24x16 search windows shifted each time four pixels in order to cover the entire facial image. The results obtained should be assessed with the understanding that: (*i*) only two candidates, corresponding to the left and right eye, are allowed and misses could occur - for our testing we missed one eye in two out of 58 test images; and (*ii*) the false positive rate is evaluated now in terms of missing the eyes rather than for frames all across the facial landscape. The performance - 89.66% correct detection - is satisfactory when taking into account that the visual routine can now actually lock on the eye. Post processed eye detection results using WTA are shown in Fig. 6.

The results presented above for eye detection compare favorably with those reported by Johnson et al (1994) but on a different task related to human actions. Note that what is most important when one makes such comparisons is to use the results obtained during testing rather than those derived during learning. It is testing after all that provides an indication about the intrinsic generalization ability of the visual routine. Johnson et al (1994), using bitmap rather than gray level images and presegmented images, report their best cross validation performance, on one subtask only, that of left hand detection, in the range of 77% $\pm$21 %. Neither the false positive rate experienced during learning nor testing results are reported.

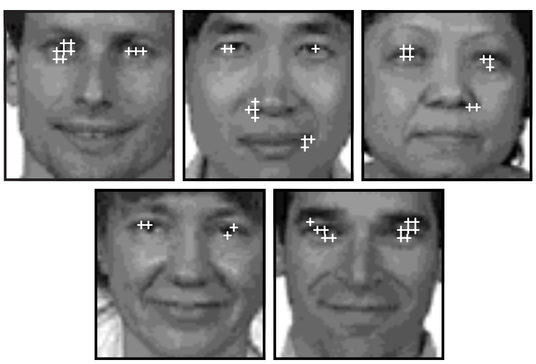

Figure 5. Eye Detection Results

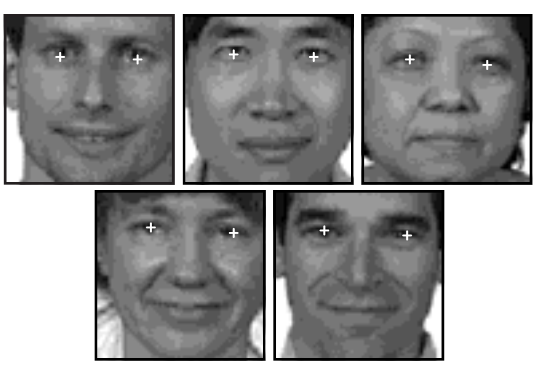

Figure 6. Eye Detection Results on DB Using WTA

7. Conclusions

We have addressed the problem of crafting visual routines for detection tasks using the framework of hybrid genetic architectures. The novelty of our approach and the associated benefits come from using natural selection (Genetic Algs.), learning (Decision Trees - ID3), and their beneficial interactions. The visual routine task for consideration is that of eye detection from real gray-level facial imagery. The experimental results reported herein prove the feasibility of our approach in terms of feature selection ('data compression') and the corresponding eye detection ('pattern recognition').

One possible area for improvement could come from us implementing also the control stage for a visual routine processor (VRP). Specifically, the regularities holding between different facial landmarks, in terms of both topology and geometry, should help in providing a sense of direction for the visual routines and discarding wrong hypotheses. Once the control stage is implemented we would test the VRP on several detection tasks whose overall goal would be to detect all the major facial landmarks such as the two eyes, and the nose and mouth regions.

8. References

[1] Ullman, S. (1984), Visual routines, *Cognition*, Vol. **18**, pp.97-159.

[2] Horswill, I. (1995), Visual routines and visual search, *Int. Joint Conf. on Artificial Intelligence*, Montreal, Canada.

[3] Chapman, D. (1993), *Vision, Instruction, and Action*, MIT Press.

[4] Ramachandran, V.S. (1985), Apparent motion of subjective surfaces, *Perception*, **14**, pp.127-134.

[5] Edelman, G.M. (1987), *Neural Darwinism*, Basic Books.

[6] Johnson, M. P., P. Maes, and T. Darell (1994), Evolving visual routines, in *Artificial Life IV*, edited by R.A. Brooks and P. Maes, MIT Press.

[7] Koza, J.R. (1992), *Genetic Programming*, MIT Press.

[8] Linsker, R. (1988), Self-Organization in a perceptual network, *IEEE Computer* Vol. **21**(3), pp.105-117.

[9] Hinton, G. E. and S. J. Nolan (1987), How learning can guide evolution, *Complex Systems*, Vol. **1**, pp.495-502.

[10] Holland, J.H. (1975), *Adaptation in Neural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

[11] Quinlan, J.R. (1986), The effect of noise on concept learning, in *Machine Learning: an Artificial Intelligence Approach*, edited by R.S. Michalski, J.G. Carbonell and T.M. Mitchell, Morgan Kaufmann, San Mateo, CA, pp.149-166.

[12] Grefenstette, J.J. (1984), Genesis: A system for using genetic search procedures, *Proc. Conf. Intelligent Systems and Machines*, pp.161-165.