2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Genetic Algorithms Based Feature Combination for Salient Object Detection, for Autonomously Identified Image Domain Types

Syed S. Naqvi
Victoria University of Wellington
New Zealand
syed.saud.naqvi@ecs.vuw.ac.nz

Will N. Browne
Victoria University of Wellington
New Zealand
will.browne@ecs.vuw.ac.nz

Christopher Hollitt
Victoria University of Wellington
New Zealand
christopher.hollitt@ecs.vuw.ac.nz

***Abstract*—Combining features from different modalities and domains has been demonstrated to enhance the performance of saliency prediction algorithms. Different feature combinations are often suited to different types of images, but existing techniques attempt to apply a single feature combination across all image types. Furthermore, existing normalization and integration schemes are not utilized in salient object detection as the combination of potential solutions is intractable to test. The aim of this work is to autonomously learn feature combinations for autonomously identified image types. To this end, we learn optimal normalization and integration schemes along with feature weightings using a novel Genetic Algorithm (GA) method. Moreover, we learn multiple image dependent parameters using our novel image-based GA (IGA) approach, to increase the generalization of the system on unseen test images. We present a thorough quantitative and qualitative comparison of our proposed methods with the state-of-the-art benchmark and deterministic methods on two difficult datasets (SED1 and SED2) from the segmentation evaluation database. IGA shows superior performance through learning optimal parameters depending upon the composition of images and using feature combinations appropriately enhances test performance and generalization of the system.**

I. Introduction

Visual attention is a fundamental research problem in psychology, neuroscience and computer vision literature. Researchers have built computational models of visual attention to predict where human are likely to fixate. Recently, it has been expanded to identify regions of interest for object detection and localization. Salient object detection is the task of marking regions of interest in a scene, which constitute a salient object. It finds applications in object recognition [1], salient object segmentation [2], image thumb-nailing [3] and image compression [4].

Most methods specialized for the task of salient object detection concentrate on constructing deterministic tailor-made features [5], [6]. Features, such as color or color gradient, are used to identify important aspects of an image. There has been a class of models that use low, mid and high-level features to learn feature combination for saliency prediction [7], [8], [9], [10].

Many existing models only learn a single set of weighting parameters for combining features, but apply them across multiple types of image, e.g. images with cluttered backgrounds or multiple objects of interest. Therefore inherently losing generalization when operated on test sets with different images having various properties and sets of features. Furthermore, linearly combining features without normalization and integration operations can result in certain features inappropriately dominating other features causing a loss of performance. An alternative approach is to learn model parameters using an assembly of weak learners (hence increasing generalization). However the quality of final solution depends upon the performance of individual learners and can be affected if one of the learners is not optimal.

The aim of this work is to autonomously learn feature combinations for autonomously identified image types, such that Salient Object Detection is improved.

We introduce the novel technique of image dependent GA (IGA) based learning framework for the task of salient object detection based on the following key observation:

- Learning multiple parametric solutions (specialized for a certain domain of images) is likely to increase the generalization of the system on unseen images as compared with learning a single set of parameters for all types of images.
- Combining the identified featured combinations through appropriate integration and normalization operations that have been autonomously learnt for each image domain is likely to overcome the problems associated with linear addition of features.

The academic goal of this paper is to investigate the efficacy of GA and specifically multiple GA based IGA model for learning important parameters for salient object detection. The specific goal with respect to the GA model is to learn various normalization and integration schemes along with feature weighting. With respect to IGA, the main focus is to explore the effect of using multiple (image dependent) parameters for increasing the overall generalization of the system. The specific objectives of this work are:

- Implement and fine-tune a GA system to learn important parameters (i.e., feature weightings, normalization and integration approaches) for effective feature combination in a visual saliency prediction framework and test its effectiveness compared to the benchmark methods, such as Support Vector Machines (SVM).
- Determine an approach to autonomously identify different types of images, which is then to be combined with the GA approach above to form the IGA framework and compare its performance with benchmark

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

learning methods and state-of-the-art deterministic methods.

II. Background

The general form for weighted feature combination for producing the final saliency output can be formulated as

$$S = \circ w_i \mathcal{N} F_i, \tag{1}$$

where w_i is the weight for feature F_i, $\circ$ represents the integration function and $\mathcal{N}$ represents the normalization operator (note: Normalization is used to denote a transform function in this context).

Judd et al. [7] employed SVMs with linear kernels to learn a model of saliency from 33 features (including low, mid and high level features) for predicting human eye fixations. Although their approach achieves good results, their model only learns weighting parameters through linear feature combination and neglects the important step of feature normalization prior to combination.

Zhao et al. [10] employed least square regression for eye fixation prediction using basic saliency features (i.e., color, intensity and orientation). Borji et al. [8] also used linear regression for saliency prediction using eye tracking data. Again both the approaches only learn the weighting parameters neglecting important aspects of feature combination such as normalization and integration operation.

SVM with non-linear kernel (RBF) [8] and AdaBoost [8], [9] have also been adopted in subsequent works to learn saliency models. Non-linear SVMs [8] only learn feature weights to effectively combine features and use addition as the integration function. AdaBoost [8], [9] learn model parameters using an assembly of weak learners (hence increasing generalization). However the quality of final solution depends upon the performance of individual learners and can be affected drastically by one of the learners in the decision tree, degrading the generalization of the overall system.

Integration and normalization functions are required to combine the different features in a meaningful way such that a single feature does not inappropriately dominate another.

A. Feature Weighting

The weight assigned to a feature quantifies its relative importance in predicting saliency. Each feature map is multiplied by its corresponding weight during the combination process to control its relative contribution to the final saliency map. They represent the top-down information, which is used to combine different feature maps having diverse properties. A majority of models try to learn this top-down information to optimize the behavior of the saliency model.

B. Normalization

As various feature maps come from different modalities and dynamic ranges, it is highly important to condition a feature map (before feature combination), such that its strong activation peaks are enhanced and weaker ones are suppressed. This function is termed as normalization and is a crucial step in feature combination. Some normalization functions reported in the literature are global, iterative [11] and identity, exponential and logarithmic [12] and can be expressed as follows:

$$\left\{ x, \exp(x), \frac{-1}{\log(x)}, \frac{1}{1+\exp(-x)}, x(M-\bar{m}), x + x * \mathcal{DoG} \right\}, \tag{2}$$

where $\mathcal{DoG}$ is the difference of Gaussian filter and $*$ denotes convolution. M is the global maximum of the feature map and $\bar{m}$ represents the local minimum of the feature map. For details about the global and iterative normalization please see [11].

C. Integration

The mathematical operation that combines the feature maps to produce the final saliency output is termed as the integration function. Addition of feature maps has always been the norm for most methods [7], [8], [9], [10]. However, Klein et al. used element-wise multiplication [5] and Gazit et al. used harmonic mean [13] to integrate the feature maps. These integration functions can be expressed as follows:

$$\left\{ \sum_{i=1}^{n} x, \prod_{i=1}^{n} x, \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{x} \right)^{-1} \right\}. \tag{3}$$

III. Proposed Methods

A. Feature Extraction

We carefully choose features from literature that have been demonstrated to correlate with visual attention and are well-suited for the task of salient object detection. To thoroughly evaluate the learning performance of models, features having a reasonably high degree of variability in performance are chosen such that there is a considerable difference in performance of the lowest performing feature to the highest one. We extract nine low and mid-level features from the raw image re-sized to 200x200 pixels. We extract the following features $F_{0..8}$ for each pixel of the image:

- F_0: One global feature which assigns low saliency to colors that vary a lot in the spatial domain based on the work of Liu et al. [14].
- F_1: One global feature capturing contrast between clusters obtained from k-means segmentation inspired by the work of Fu et al. [15].
- F_2: One global feature computing the spatial distribution of pixels in a cluster with respect to image center again inspired by the work of Fu et al. [15].
- F_3: One region based feature that computes the global contrast between spatial neighboring regions only [2].
- F_4: One mid-level feature that uses the objectness of image windows to highlight salient objects based on the work of Alexe et al. [16].
- F_5: One feature that groups information based on F_4.
- $F_{6,7}$: Two low-level region-based color features adopted from the work of Naqvi et al. [17].
- F_8: One feature that highlights salient patterns based on the work of Naqvi et al. [17].

B. Genetic Algorithm

As majority of the decision variables in the parameter vector λ_i (please refer to section III-B1 for details) are real, therefore real-coded representation of chromosomes is used. The normalization and integration functions are encoded as integers in the GA. The initial population is chosen from a uniform distribution, while taking into account the bounds on variables. We employ roulette wheel selection. We empirically find the value of elite individuals to be six. Uniform crossover and a custom mutation function is used to create the next generation (for details please refer to section III-B3 and III-B4). The integer variables are truncated according to randomized rounding, using a probability of 0.5. The iterations of the GA solely depend upon the number of generations.

1) Encoding Decision Variables: The chromosomes use real-coded representation with integer constraints. An example phenotype of an individual is depicted in Figure 1.

The weighting parameters for the features are encoded in the GA as in (4). Negative weights are also allowed.

$$w_n \in [-1, 1]\,. \tag{4}$$

Six different normalization schemes are encoded as follows. Each integer represents a different normalization operation.

$$\mathcal{N} \in [1 \cdots 6]\,. \tag{5}$$

The three integration schemes used in this work are encoded in (6). Each integer encodes a single integration function.

$$\circ \in [1 \cdots 3]\,. \tag{6}$$

The i^{th} optimal parameter vector to be searched, denoted by $\lambda_{\mathbf{i}}$ can be obtained by concatenating the weight vector $\mathbf{w_n}$, normalization operation $\mathcal{N}$ and the integration function $\circ$ (note: There are multiple optimal parameter vectors to be searched, please refer to section III-C for details).

2) Objective Function: We model the problem of learning saliency as a binary classification problem similar to previously reported methods [8], [9]. The goal of the objective function is to maximize the classification accuracy of the system. To achieve this goal, we encode our objective function as to minimize the difference between the ideal classification accuracy and the computed classification accuracy. In order to compute the classification accuracy of a particular saliency map, we first compute the saliency map output for the system as follows:

$$S = \circ w_i \mathcal{N} F_i. \tag{7}$$

Afterwards, the saliency map is thresholded and compared with a binary ground truth map to compute the classification accuracy. The objective function can be expressed as

$$O(\lambda_{\mathbf{i}}) = 1 - \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}. \tag{8}$$

3) Crossover: The crossover function used in the GA is the standard uniform crossover operation [18]. To this end, we effectively incorporate constraints arising from the bounds on decision variables.

4) Mutation: To traverse the search space and find optimal solutions, we would like to mutate our individuals, where a function is randomly replaced by another function or a weighting parameter is replaced by a random weight. However, unlike crossover where we explore the search space randomly and combine different weighting parameters, normalization and integration functions, here we adapt when performing mutation based on the last successful or unsuccessful generation. The random search directions and step size take into account the feasible region and the bounds on variables. A combination of step size s, scale sc and direction vector $\mathbf{u}$ are added to the parent chromosome $\mathbf{p}$ to compute the offspring. This procedure is depicted as

$$\mathbf{o} = \mathbf{p} + s \times sc \times \mathbf{u}. \tag{9}$$

C. Image Dependent GA Based Approach

Notation : The training image set is denoted as $\mathrm{G} = \left\{\mathrm{G}_1, \mathrm{G}_2, \ldots, \mathrm{G}_{\frac{N}{k}}\right\}$, where the i^{th} image group is represented by $\mathrm{G}_i \subseteq \mathrm{G}$. The complete feature set for the training images is denoted by $\mathcal{F} = \left\{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{\frac{N}{k}}\right\}$. $\mathcal{F} \in \mathbb{R}^{D \times D \times n \times N}$, where D is the dimension of a single feature (D=200 here), n is the number of features (n=9 here) and N is the number of images in the dataset. The i^{th} feature set for the i^{th} image group G_i is denoted as $\mathcal{F}_i \subseteq \mathcal{F}$ and is given by $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k\}$, where $i \in \left[1 \cdots \frac{N}{k}\right]$. $\mathbf{f}_i \in \mathbb{R}^{D \times n}$ is a feature vector and k is the number of nearest neighbors (k=10 here).

The complete ground truth set for the training set G is denoted by $\mathcal{G} = \left\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{\frac{N}{k}}\right\}$. The i^{th} ground truth set for the i^{th} image group is denoted by $\mathcal{G}_i \subseteq \mathcal{G}$, $i \in \left[1 \cdots \frac{N}{k}\right]$. The optimal parameter set is denoted by $\mathcal{P} = \left\{\lambda_1, \lambda_2, \ldots, \lambda_{\frac{N}{k}}\right\}$, where $\lambda_i \subseteq \mathcal{P}$ is the i^{th} optimal parameter vector for the i^{th} image group. The i^{th} memory set represented as $\mathcal{M}_i \subseteq \mathcal{M}$ is comprised of a feature set $\mathcal{F}_i$ and a parameter set λ_i. Here $\mathcal{M}$ is the set of all memory sets $\mathcal{M}_i$. For a particular image in the test phase, the optimal parameter vector is found by searching for the closest image group in the feature space and is denoted by $\lambda^\star$.

The system model for IGA is shown in Figure 2. The process of feature extraction defined in section III-A is used to extract features for the training image set G. The procedure for autonomous grouping is depicted in Algorithm 1. The outputs of multiple GA models having independent feature and ground truth sets as inputs are connected to multiple independent memory sets.

The procedure for training the IGA is depicted in Algorithm 1. The images are autonomously placed into groups depending upon their feature composition. The process starts by searching for k nearest neighbors for an image based on its distance to other images in feature space. The image features and ground truth along with its nearest neighbors are assigned to the current groups $\mathcal{F}_i$ and $\mathcal{G}_i$ and deleted from the complete feature set $\mathcal{F}$ and ground truth set $\mathcal{G}$, respectively. This process continues until the number of features in the complete feature set $\mathcal{F}$ fall below the nearest neighbors k. After the images are divided into groups, multiple GA algorithms are trained, each with corresponding features and ground truth from different groups. The resulting optimal parameters λ_i obtained from

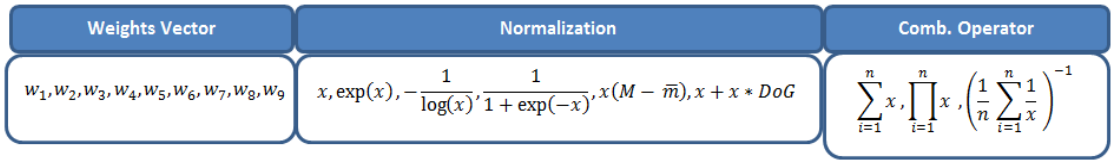

Fig. 1. An example of the phenotype of an individual in our GA population. It shows the possible parameters and functions that can be encoded.

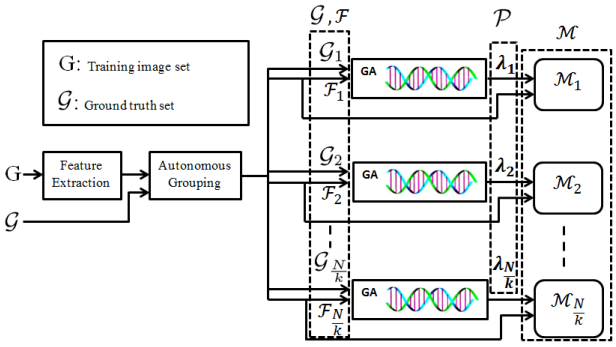

Fig. 2. System model of IGA. Feature extraction process is defined in section III-A. The process of autonomous grouping is defined in Algorithm 1.

different GA algorithms and the corresponding feature set $\mathcal{F}_i$ is stored in the corresponding memory set $\mathcal{M}_i$.

Algorithm 1: Training Process of the IGA

Data: $\mathcal{F}, \mathcal{G}$
Result: $\mathcal{M}$
while $|\mathcal{F}| \geq k$ **do**
  Find k nearest neighbors for the first element of the set $\mathcal{F}$;
  Assign the feature vectors for the current element and nearest neighbors to the set $\mathcal{F}_i$;
  Assign the ground truth for the current element and nearest neighbors to the set $\mathcal{G}_i$;
  $\mathcal{F} - \mathcal{F}_i$, $\mathcal{G} - \mathcal{G}_i$;
  i++;
for $i \leftarrow 1$ *to* $|\mathcal{F}_i|$ **do**
  Train each GA according to the equations (9) and (10) and the settings described in section III-B;
  Find optimal parameter vector λ_i;
  Create a memory set $\mathcal{M}_i = \{\lambda_i, \mathcal{F}_i\}$;

During the testing phase of the IGA, feature vector $\mathbf{f_t}$ for the test image is computed. For the i^{th} feature set $\mathcal{F}_i$, the sum of Euclidean distances $\mathbf{d}_i$ between its feature vectors $\mathbf{f}_i$ and the test feature vector $\mathbf{f}_t$ is computed according to the following equation:

$$\mathbf{d}_i = \sum_{j=1}^{|\mathcal{F}_i|} \|\mathbf{f}_i - \mathbf{f}_t\| . \tag{10}$$

We concatenate $\mathbf{d}_i$ (for feature sets of all image groups) in a vector $\mathbf{d} = \left[\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{\frac{N}{k}}\right]$. The minimum value in $\mathbf{d}$ denoted by D provides the distance between a particular test image and the closest feature set. D is then used to access the corresponding memory set and the optimal parameter vector $\lambda^\star$. Afterwards, the saliency is computed and thresholded to yield a binary saliency map. The binarized saliency is then used to compute precision and recall. This process is depicted in Algorithm 2.

Algorithm 2: Testing Process of the IGA

Data: $\mathcal{M}$
Result: $\hat{\text{Prec}}, \hat{\text{Recall}}$
Compute feature vector $\mathbf{f}_t$;
for $i \leftarrow 1$ *to* $|\mathcal{F}_i|$ **do**
  Compute D according to equation 10;
  Use D to find the corresponding $\lambda_i = \lambda^\star$;
  Compute saliency using the learned parameters;
  Threshold saliency and compute $\hat{\text{Prec}} = \frac{\text{TP}}{\text{TP+FP}}$;
  Compute $\hat{\text{Recall}} = \frac{\text{TP}}{\text{TP+FN}}$;

IV. Experimental Design

A. Data Set

This work uses the two challenging datasets from the segmentation evaluation database (SED) [19]. The database contains 200 images (of varying aspect ratio) in total, divided into two datasets (100 images for each dataset), namely SED1 and SED2. The two datasets include one and two foreground objects respectively. Unlike other databases, SED includes images having multiple objects resulting in increased difficulty level.

B. Ground Truth

The ground truth segmentations were also taken from the SED database [19]. The ground truth segmentations have the same dimensions as the input images. It includes manually annotated segmentations from three different human subjects. The segmentations contain two classes for one object images and three classes for the two object images. We process the human annotations to obtain ground truth segmentations according to the method prescribed by [19]. The binary ground truth segmentations for both two and three class images are acquired by thresholding the votes from each human subject for each foreground pixel. If there exists two or more than two votes for a particular foreground pixel then it is assigned one and all other pixels are assigned zero. In this manner binary ground truth for all images is constructed.

C. Performance Measures

All salient object detection methods evaluate their approach by comparing the thresholded saliency map with the binary

ground truth map. To evaluate the robustness of a saliency map, we use the benchmark method reported in [20] called fixed or naive thresholding. According to this method the saliency map is thresholded at a fixed threshold T_f within $[0, 255]$. To compare saliency maps from different methods, this threshold is varied from 0 to 255 to produce a precision-recall and an F-measure curve.

D. Experimental Setup

Both SED1 and SED2 datasets are randomly divided into 70 training and 30 testing images. The GA is repeatedly run for 30 times (with a different seed for each run) to search for the optimal parameters. The population size is chosen empirically to be 1000 and the GA is run for 200 generations. The crossover fraction is set to 0.8. The solution closest to the mean of the 30 solutions is selected as the representative solution. This is achieved by computing the Euclidean distance (in feature space) between each of the 30 solutions and the mean solution. The solution having the minimum distance from the mean solution is selected as the final representative solution. The same settings are used to train the independent GA models for the proposed IGA model.

V. Results and Discussion

We evaluate our proposed GA and IGA techniques against state-of-the-art benchmark methods namely linear SVM (LSVM) and non-linear SVM (NLSVM), and state-of-the-art methods created for the task of salient object detection, namely AC [21], FTS [20] and MSSS [22] on benchmark datasets.

A. Quantitative Analysis

1) Comparison of GA with Benchmark Methods: Figure 3 presents the average precision-recall and F-measure curves for LSVM, NLSVM and our proposed GA model on the SED1 and SED2 test sets. For SED1 performance, LSVM performs marginally better than NLSVM in terms of precision-recall curve, while the latter performs marginally better than the former with respect to F-measure curve. Our proposed GA model outperforms both versions of SVM by scoring high precision values at all thresholds and recall values. With respect to F-measure curve, the proposed GA model scores higher F-measure values for all thresholds and outperforms other methods.

On the SED2 dataset, LSVM and NLSVM performance is similar in terms of area under the precision-recall and F-measure curve. The proposed GA model outperforms the state-of-the-art methods both in terms of precision-recall and F-measure curves. Our proposed GA model shows robust performance on majority of thresholding levels.

2) Comparison of IGA with Existing Work: Figure 4 and Figure 5, present the test classification accuracies of the solution sets learned by individual GA models trained on different image groups along with the test accuracy of our proposed IGA based approach. The test accuracies of the independent GA models ($G_1 - G_7$) are shown as box plot depicting the dispersion of 30 independent runs of each GA model. The test accuracy of the proposed IGA model evaluates to be a single value due to the fact that IGA selects a single solution (for a particular test image in the test phase) from the optimal solutions produced by independent GA models.

Figure 4 and Figure 5 demonstrate that our proposed IGA method achieves remarkably higher accuracy compared to the individual GA models by effectively utilizing them on image by image basis. As can be seen from Figure 4 that the classification accuracies of individual groups are better than 80% with G_5 reaching a high accuracy of more than 86%. However the mean accuracy of all groups is still 83.8% and the IGA method boosts the accuracy to 94.75% using the learned solutions.

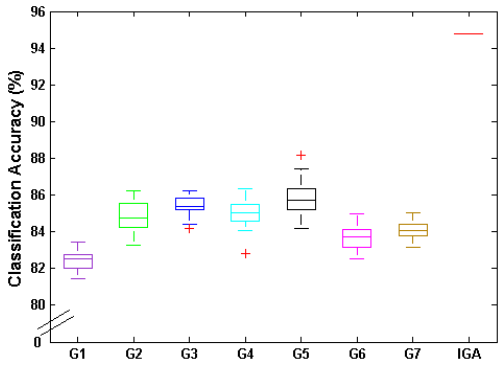

Fig. 4. Comparison of test classification accuracies for the seven groups and IGA on the SED1 dataset.

The best individual test accuracies for SED2 dataset in Figure 5 reach 83%, however the mean accuracy of all groups in this case is also low, i.e. 73.9%. Again IGA exploits the combination of groups and effectively boosts the classification accuracy to 87.9%. It is to be noted that the dispersion in the test accuracies for independent GA models is less as compared with the accuracies for SED2 data set which seem to have more variance. This property can be attributed to the higher difficulty of the SED2 dataset as compared to the SED1 dataset.

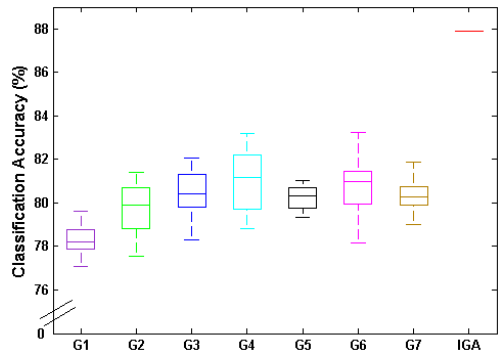

Fig. 5. Comparison of test classification accuracies for the seven groups and IGA on the SED2 dataset.

Figure 6 shows the performance of the proposed IGA model in comparison with the state-of-the-art methods. For the SED1 dataset, the proposed IGA model outperforms all other methods both in terms of precision-recall and F-measure curves. IGA performs notably better than our proposed GA method (the second best performing method) with respect to both precision-recall and F-measure curve. AC [21] performs the worst scoring lowest precision and F-measure values for all

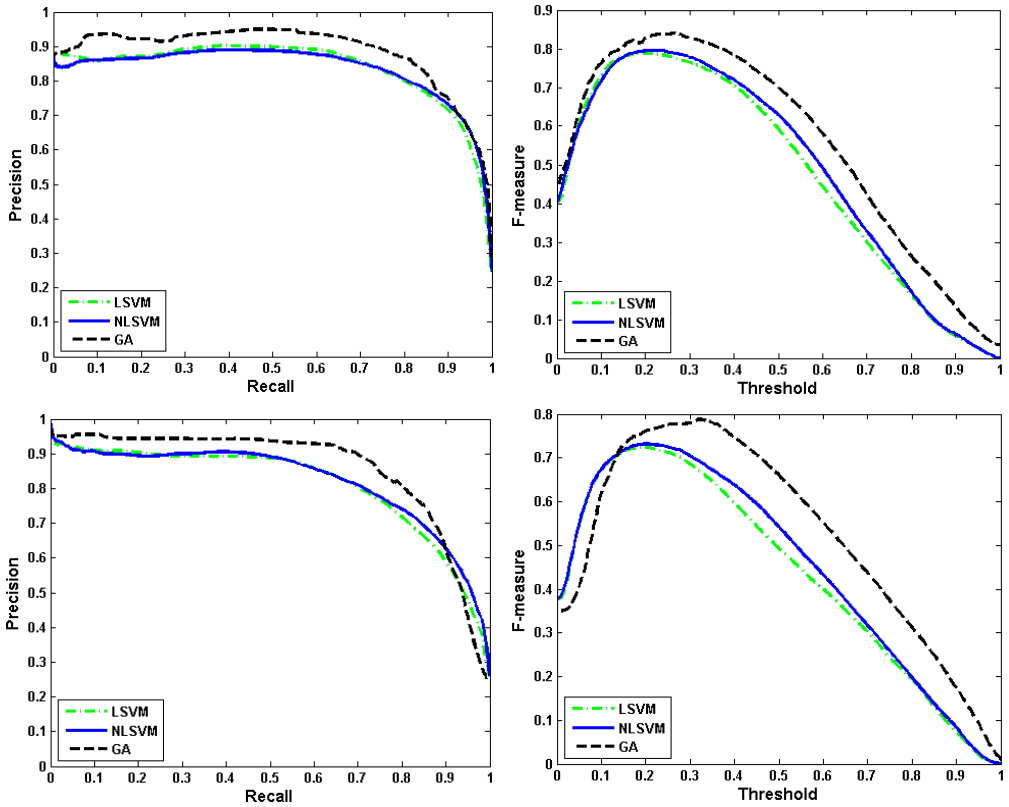

Fig. 3. Top row: Average precision-recall curve and F-measure curve of the GA in comparison with linear combination and SVM benchmarks for SED1 dataset. Bottom row: Average precision-recall curve and F-measure curve for the SED2 dataset. These curves are for the testing phase.

thresholds. FTS [20] and MSSS [22] perform comparably with other methods on lower thresholds in terms of precision-recall curves but do not show robustness to all threshold values.

On the SED2 dataset, IGA again outperforms all other methods in terms of scoring high precision and F-measure values for all threshold levels and in terms of highest area under the curve. According to the F-measure curve, IGA scores above 0.6 for the majority of the thresholds showing the robustness of the system.

B. Analysis of Evolved Solutions

Table I shows the representative optimal parameter set evolved by the GA and IGA models. The first two rows are the optimal parameter sets learned by the GA using complete training image sets for SED1 and SED2 datasets respectively. The last two rows show the optimal parameters learned by the IGA, while it is trained by using one of the image groups for SED1 and SED2 datasets respectively. The indicative results for only one image group are shown here due to the limitation of space.

Weights w_0 and w_2 corresponding to features F_0 and F_2 are found to be consistently low. This can be explained by the fact the feature F_0 (which captures the color spatial distribution) is neither relatively informative for SED1 images due to highly cluttered scenes, nor for SED2 images which include two salient objects usually having different color. Feature F_2 is not highly useful as the objects are not consistently placed in the center of the image for both SED1 and SED2 datasets respectively. Weight w_6 for the feature F_6 was found to be consistently high. This might be attributed to the fact that objects in both SED1 and SED2 datasets have high contrast at their boundaries along with consistent color contrast compared with the background. Weight w_8 for the feature F_8 was generally weighted negatively. This may be due to the reason that objects in both datasets are highly discriminative to the background in term of their color relative to the pattern of the objects and the background. The integration operation was generally found to be either addition or multiplication, which is also the norm. The optimal normalization operation was found to considerably varying for both GA and IGA models, depending upon the nature of conditioning required for features in different scenarios.

TABLE I. THE OPTIMAL PARAMETER SETS LEARNED BY THE GA. NEGATIVE VALUES ARE RESTRICTED TO A SINGLE DECIMAL PLACE.

No.	Parameter Set										
	$w0$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$	$w7$	$w8$	$\mathcal{N}$	$\circ$
GA1	0.00	0.00	0.01	0.95	0.91	0.90	1	0.36	0.00	3.00	1.00
GA2	0.01	0.92	0.07	0.41	0.02	0.23	1	0.10	-0.1	6.00	2.00
IGA1	0.02	0.08	0.00	0.33	0.05	0.46	0.98	0.68	-0.1	4.00	2.00
IGA2	0.34	0.36	0.19	0.68	0.35	0.00	0.99	0.94	-0.5	2.00	1.00

C. Qualitative Comparison

Figure 7 shows the visual comparison of models in terms of their saliency output. The deterministic methods especially FTS [20] and MSSS [22], perform well by assigning low saliency to background but struggle in assigning high values inside salient object contours. As the methods are based on filtering, they might filter out important salient information, when it lies in the same band. LSVM and NLSVM mostly

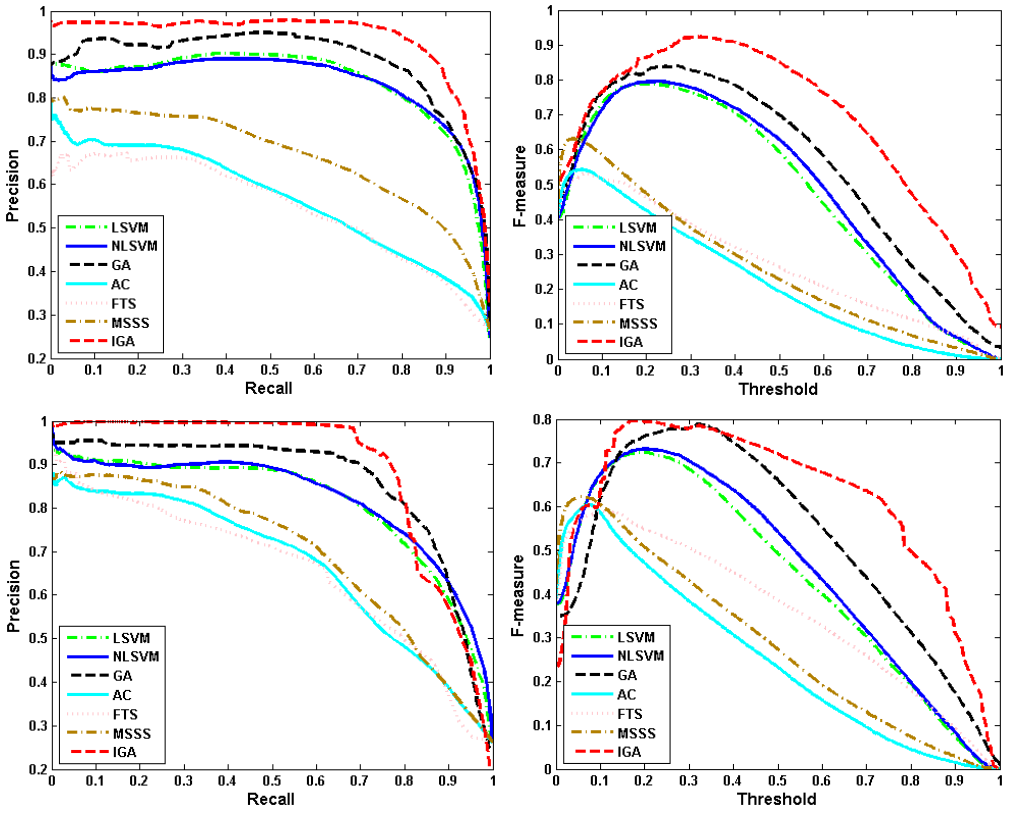

Fig. 6. Top row: Comparison of the proposed GA and IGA models with benchmarks (Linear and Nonlinear SVMs) and state-of-the-art methods (i.e. AC, FTS and MSSS) in terms of average precision-recall curve and F-measure curve Bottom row: Similar set of results for the SED2 dataset. These figures depict the performance of all the models on the test images.

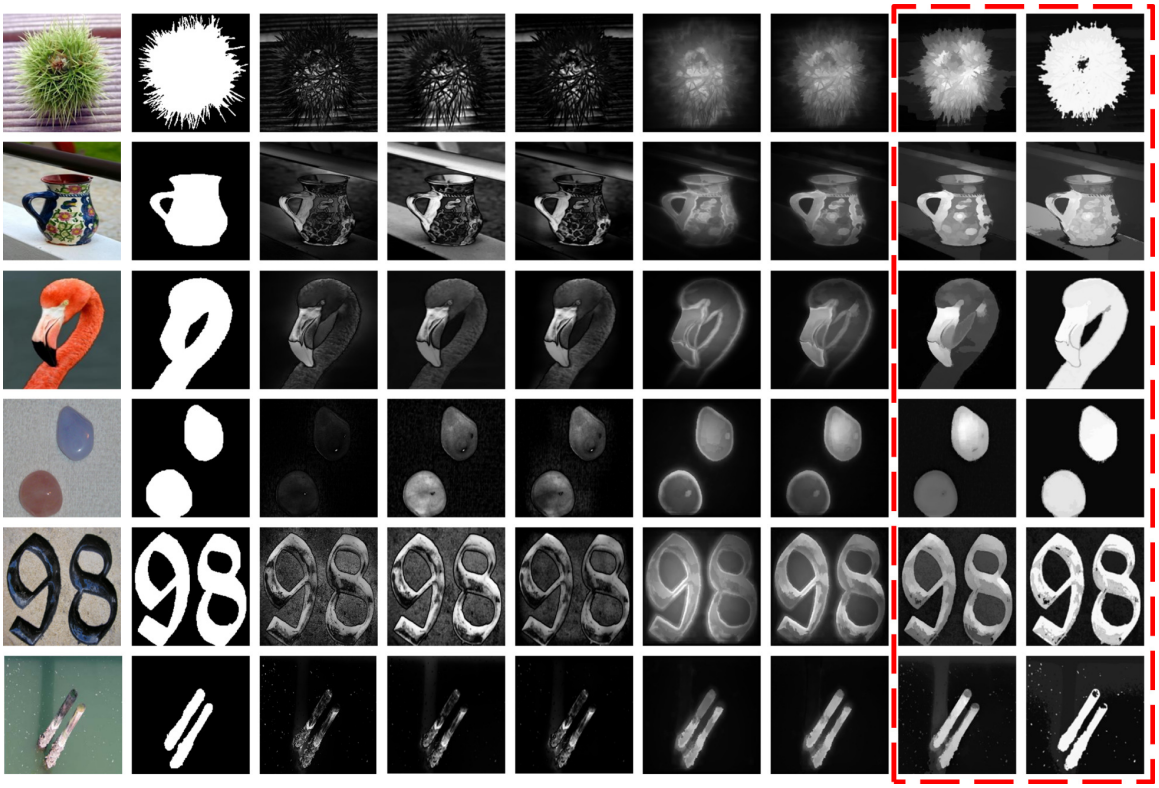

Fig. 7. Visual comparison of models on selected images from both datasets. From left to right: Input, GT, AC, FTS, MSSS, NLSVM, LSVM, GA, IGA. Our GA and IGA models are shown in the red box.

highlight object boundaries and could not appropriately weigh features that score higher inside salient object contours. Our GA method performs better than other state-of-the-art methods, but misses some important salient information in some cases. Our IGA method produces the best saliency maps compared to all other methods, highlighting the salient object and suppressing the background. It effectively captures the neck and head region in third row of the image, which is effectively missed by all other approaches.

D. Run-time Aspects

In most real-time applications, the amount of time required to handle a test scene is crucial, while the training time does not affect the performance of the system. It is to be noted that the training time for our proposed GA and IGA systems is longer than the state-of-the-art methods because of the overhead of complex evolutionary optimization to attain the optimal solution. However, the time required for handling a test image is comparable to all the state-of-the-art methods included in this work for comparison. The total time required to compute the saliency of a test image is highly dominated by the time required to compute the features in case of learning methods (i.e., GA, IGA and SVMs). The average time required to compute the nine features for a single image for SED1 dataset is 5.92(1.40) seconds. The figures in the bracket represent the standard deviation. The timing results were recorded on a desktop computer with a i7-4770 CPU @ 3.40GHz (8 core) processor and 8GB of RAM.

VI. Conclusions

In this work, novel GA and image based GA (IGA) techniques for detecting salient objects were presented. The first objective to implement and fine-tune a GA system to learn important parameters for effective feature combination in a visual saliency prediction framework was achieved successfully showing improvements over state-of-the-art approaches. When complemented with the ability to autonomously identify different types of images, the IGA technique improved over these benchmark approaches in terms of precision-recall, F-measure and qualitative comparison.

Further conclusions were reached: 1) The normalization and integration schemes play a vital role in feature combination and can significantly enhance performance of the system (compared with only feature weighting) as shown by the performance of our GA method. 2) Learning optimal parameters depending upon the feature composition of images and using them accordingly enhances test performance and generalization of the system as depicted by the superior performance of our IGA technique.

References

[1] W. D and K. C, "Modeling Attention to Salient Proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

[2] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global Contrast Based Salient Region Detection," in *Computer Vision and Pattern Recognition, 20011. CVPR'011. IEEE Conf on*, 2011, pp. 409–416.

[3] L. Marchesotti, C. Cifarelli, and G. Csurka, "A Framework for Visual Saliency Detection with Applications to Image Thumbnailing," in *Computer Vision, 2009 IEEE 12th International Conf on*, 2009, pp. 2232–2239.

[4] L. Itti, "Automatic Foveation for Video Compression Using a Neurobiological Model of Visual Attention," *Image Processing, IEEE Transactions on*, vol. 13, no. 10, pp. 1304–1318, 2004.

[5] D. Klein and S. Frintrop, "Center-surround Divergence of Feature Statistics for Salient Object Detection," in *Computer Vision (ICCV), 2011 IEEE International Conf on*, 2011, pp. 2214–2219.

[6] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-Aware Saliency Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1915–1926, 2012.

[7] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to Predict Where Humans Look," in *Computer Vision, 2009 IEEE 12th International Conf on*, 2009, pp. 2106–2113.

[8] A. Borji, "Boosting Bottom-up and Top-down Visual Features for Saliency Estimation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conf on*, 2012, pp. 438–445.

[9] Q. Zhao and C. Koch, "Learning Visual Saliency by Combining Feature Maps in a Nonlinear Manner Using AdaBoost," *Journal of Vision*, vol. 12, no. 6, pp. 1–15, 2012.

[10] ——, "Learning a Saliency Map Using Fixated Locations in Natural Scenes," *Journal of Vision*, vol. 11, no. 3, pp. 1–15, 2011.

[11] L. Itti and C. Koch, "Feature Combination Strategies for Saliency-Based Visual Attention Systems," *Journal of Electronic Imaging*, vol. 10, no. 1, pp. 161–169, 2001.

[12] A. Borji, D. Sihite, and L. Itti, "Salient Object Detection: A Benchmark," in *Computer Vision ECCV 2012*, ser. Lecture Notes in Computer Science, 2012, pp. 414–429.

[13] M. Holtzman-Gazit, L. Zelnik-Manor, and I. Yavneh, "Salient Edges: A Multi Scale Approach," in *ECCV, Workshop on Vision for Cognitive Tasks*, 2010.

[14] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to Detect A Salient Object," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conf on*, 2007, pp. 1–8.

[15] H. Fu, X. Cao, and Z. Tu, "Cluster-Based Co-Saliency Detection," *Image Processing, IEEE Transactions on*, vol. 22, no. 10, pp. 3766–3778, 2013.

[16] B. Alexe, T. Deselaers, and V. Ferrari, "What is an Object?" in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conf on*, 2010, pp. 73–80.

[17] S. S. Naqvi, W. N. Browne, and C. Hollitt, "Combining Object-Based Local and Global Feature Statistics for Salient Object Search ," in *IVCNZ*, 2013, p. 6.

[18] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[19] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image Segmentation by Probabilistic Bottom-up Aggregation and Cue Integration," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conf on*, 2007, pp. 1–8.

[20] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned Salient Region Detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conf on*, 2009, pp. 1597–1604.

[21] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient Region Detection and Segmentation," in *Computer Vision Systems*. Springer, 2008, pp. 66–75.

[22] R. Achanta and S. Susstrunk, "Saliency Detection Using Maximum Symmetric Surround," in *Image Processing (ICIP), 2010 17th IEEE International Conf on*, 2010, pp. 2653–2656.