Hybridizing Teaching Learning Based Optimization with Genetic Algorithm for Colour Image Segmentation

R.V.V.Krishna, Member, IAENG and S. Srinivas Kumar

Abstract — Color Image segmentation is a challenging low level image analysis task, which has varied engineering and industrial applications. Clustering based image segmentation algorithms group the color and texture features of the image into different clusters. The performance of the clustering algorithms depends on the cardinality and choice of initial cluster centroids, and end up in a different solution each time the clustering algorithm is executed. Finding the best solution from these set of solutions needs an efficient optimization algorithm. In this paper, a new hybrid algorithm which combines the merits of two popular evolutionary algorithms, Teaching Learning Based Optimization (TLBO) and Genetic Algorithm (GA), are combined for solving color image segmentation problem. The texture feature of the image, obtained by using Weber Power Local Binary Pattern (WPLBP), is binary and the color feature obtained by homogeneity model is real variable. GA is more suitable for variable binary optimization problems and TLBO is suitable for optimizing real variables. Further TLBO is computationally efficient and does not need parameter tuning.

Index Terms—Clustering, Teaching Learning Based Optimization, Genetic Algorithm, Segmentation, Hybrid Algorithms, Rough Sets, Fuzzy Sets, Soft Sets

I. INTRODUCTION

Color image segmentation is to divide a chromatic image into different homogeneous and connected regions based on color, texture and their combination [10]. It is an essential part of image analysis and decides the final output of any image analysis task. In this paper, color image segmentation is based on the feature clustering technique. The steadiness of clustering based segmentation methods such as k-means, Rough-k-means etc is limited by the initially chosen cluster centers, and also on the cardinality of cluster centers chosen. The problem is addressed by evolutionary computing techniques. A population of initial cluster centroids is formed by repeated application of Soft rough fuzzy c-means clustering (SRFCM) algorithm. The optimal cluster centers, are evolved by hybridizing TLBO and Genetic algorithm.

Usage of Evolutionary methods viz., Genetic Algorithm, Differential Evolution, and Simulated Annealing for optimizing the performance of classical clustering methods, such as Fuzzy-C-Means and K-means is observed in the literature. Maulik et al., [14] proposed an improved differential evolution method to optimize multi-objective parameters in fuzzy clustering (XB and Jm), where Jm stands for the global cluster variance, while XB is a combination of global and local situations. In [4] Genetic Algorithm was used for multi objective parameter (XB and Jm) optimization.

Hybridization of different evolutionary algorithms are traced in the literature. Juang et al.,[11] proposed a recurrent network design by hybridizing GA and PSO where in one half of the best contributing chromosomes are grouped as elitist and the remainder are left over. The next generation consists of enhanced elites after PSO application, and GA offspring of enhanced elites. Hybridization of Differential Evolution (DE) and Quantum PSO (QPSO), named DEQPSO, is proposed for planning routes of unmanned aerial vehicle in [7]. In DEQPSO, sequential hybridization of QPSO and DE is performed where in, at each iteration, the parent generation undergoes evolution using QPSO and DE in sequential order. Lei Wang et al.,[28] proposed a hybridization of TLBO and DE for chaotic time series prediction. DE is incorporated into update the previous best positions of individuals to force TLBO jump out of stagnation, because of its strong searching ability.

In general it is observed that GA very ably handles binary variables and TLBO is more capable in handling continuous variables. Motivated by this fact, a composite feature of both colour and texture is formed to solve color image segmentation problem. Texture feature constitutes the binary part of the solution and color the real part. GA operates on the texture part and TLBO operates on the color part of solution, so that the hybrid optimizer effectively explores both the binary and real search domain.

The main contributions in this paper are as follows
1) A novel hybridization of TLBO and GA, where in the individual performances of TLBO and GA are effectively enhanced and tested on the color image segmentation problem.
2) A new hybrid texture feature named “Weber Power Local Binary Pattern (WPLBP)” which is a hybrid of LBP and Power Law Descriptor is proposed in this paper.

The rest of the paper is organized as follows. In Section II the extraction of color and texture features required for clustering is discussed. In Section III the Soft Rough Fuzzy C Means Clustering, which is used for generating the initial population of cluster centers is presented. In Section IV the
II. COLOR AND TEXTURE FEATURE EXTRACTION

A. Color Feature Extraction:

The color image consists of multiple bands, with each band containing a range of intensity values. Here, the Lab color model is used for color feature extraction because it is very convenient to measure small color difference. Let \(P_i = (P_i^L, P_i^a, P_i^b)\) represent the color components of a pixel at the location \((i,j)\) in an \(M \times N\) image, pixel level color feature \(C_{ijk}^k\) of color component \(P_i = (k=L, a, b)\) can be computed as follows:

1. Prepare a window of size \(3 \times 3\) for construction of pixel level color feature.
2. Pixel variance in terms of standard deviation and discontinuity in terms of edge detection, of color component \(C_{ijk}^k\) are calculated.

\[
v_{ij}^k = \sqrt{\frac{1}{d^2} \sum_{m=0}^{d-1} \sum_{n=0}^{d-1} (C_{m,n}^k - \mu_{ij}^k)^2} \tag{1}
\]

and

\[
\mu_{ij}^k = \frac{1}{d^2} \sum_{m=0}^{d-1} \sum_{n=0}^{d-1} (C_{m,n}^k)^2 \tag{2}
\]

where \(\mu_{ij}^k\) is mean of color component \(C_{ij}^k\) (\(k = L, a, b\)).

The edge variations are calculated in terms of the absolute difference between the neighbors and the current pixel, whereas

\[
\Delta I = I - I' \tag{6}
\]

where \(\Delta I\) denotes the incremental change in intensity (or) just noticeable difference for discrimination. \(I\) denotes the original stimulus intensity and \(k\), the proportionality constant suggests that the ratio does not change, even when there is a variation in the original stimulus value \(I\). The fraction \(\Delta I/I\) is known as the Weber fraction. Weber’s law says that the size of a just noticeable difference is a constant proportion of the original stimulus value.

Chen et al., [5] proposed Weber Local Descriptor, as a texture descriptor, by considering the concepts of weber’s law. But Guilford observed that empirical data such as an image does not always fit well into weber’s law. He suggested a modification to weber’s law as follows and hence called as Guilford power law [1].

\[
\frac{\Delta I}{I} = k \tag{7}
\]

where \(\alpha\) is an exponent slightly less than 1. The perceived brightness of the human eye is proportional to the logarithm of actual pixel value, rather than the pixel value itself. The power law is also scale invariant. Hence the proposed power law descriptor models the perception of human beings better than weber local descriptor. The Power law descriptor consists of two components differential excitation \(\xi\) and orientation \(\theta\).

Differential excitation finds the salient variations within an image to simulate the pattern perception of human beings. It is defined as the ratio between two terms \(V_i^{m_0}\) and \([V_i^{m}]^{'0}\).

\[
\xi(x) = \arctan \left[ \frac{V_i^{m_0}}{[V_i^{m}]^{'0}} \right] \tag{8}
\]

where \(V_i^{m_0}\) at any pixel is the sum of the differences between the neighbors and the current pixel, whereas \(V_i^{m}\) is the value of the current pixel to a power of \(\alpha\).

\[
V_i^{m_0} = \sum_{i,j} \Delta x_i = \sum_{i,j} (x_i - x_j) \tag{9}
\]

These values are obtained by convolving the image with the following filters

\[
\begin{array}{c|c|c|c}
1 & 1 & 1 & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
\end{array}
\]

(a) Filter used to realize \(V_i^{m_0}\) (b) Filter used to realize \(V_i^{m}\)

\[
\begin{array}{c|c|c|c|c|c|c}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

(c) Template

The orientation component is the gradient orientation which is computed as

\[
\theta(x_i) = \frac{\pi}{2} - \arctan \left[ \frac{v_i^{m_1}}{v_i^{m_0}} \right] \tag{10}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c}
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

(a) Filter used to realize \(V_i^{m_0}\) (b) Filter used to realize \(V_i^{m}\)
The two dimensional histogram of the differential excitation and orientation component is the power law descriptor.

\[ TFP LD_{ij}^k = 2D \text{Histogram} \left[ \phi(x_c), \theta(x_c) \right]_{ij} \quad k = L, a, b \]  

(11)

**Local Binary Pattern:**

LBP, [2] originally proposed by Ojala et al. for texture description, belong to the class of non-parametric texture analysis, describing, the local texture of any image by thresholding each pixel in the image against its neighbors. LBP is spatial based texture descriptor, and is robust to illumination changes and computationally feasible.

The pixels in the image are encoded into a decimal number which is referred to as LBP code. The center pixel (pc) is compared with its neighbors (pn) which lie at a distance of \( R \) from pc. If the neighbor is greater than or equal to pc, it is coded as 1, else it is coded as 0. The resulting binary number is then converted into a decimal number, by binary to decimal conversion. The LBP codes of all the pixels are found is then converted into a decimal number, by binary to decimal conversion. The LBP codes of all the pixels are found by a similar procedure. Henceforth, the texture is represented by the histogram found in the defined local neighborhood.

\[ \text{LBP}_{p,n} = \sum_{i=1}^{p} 2^{(i-1)} g(p(g_{ij}) - p(g_{ic})) \]  

(12)

\[ g(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{else} \end{cases} \]

where \( p(g_{ij}) \) denotes the gray value of the center pixel, \( p(g_{ic}) \) denotes the gray value of its neighbors. \( P \) indicates the number of neighbors and \( R \) indicates the radius of the neighborhood.

(a) Gray levels (b) Threshold values (c) Local Binary Pattern

At any location \((i,j)\) in the image, the texture feature is defined as

\[ T_{ij}^k = \text{CAT}(TFPLD_{ij}^k, \text{LBP}_{ij}^k) \quad k = L, a, b \]  

(13)

**III. SOFT ROUGH FUZZY C-MEANS ALGORITHM (SRFCM)**

SRFCM has its roots in the k-means algorithm. This basic algorithm was evolved to the Rough k-means (RKM) that was proposed in [12] borrowing some of the concepts of rough set theory [19]. Rough fuzzy c-means (RFCM) algorithm which was applied to medical image segmentation problem [15].

The fundamental steps of SRFCM are as follows:

1. Assume m random initial cluster prototypes
2. Find membership uk between m cluster centers and k data points
3. Allocate each data point to the lower or upper approximation
4. Make the final assignment based on the difference between the highest and next highest membership of a data point in all clusters
5. Compute the similarity of sample points soft set to the cluster center soft set, calculate the maximum similarity and assign a pixel to a cluster to which it has maximum similarity after fuzzification.
6. Compute the updated cluster prototype for each cluster
7. Iterate and run steps 2–6 until there are no further changes in cluster centroids.

**IV. PROPOSED ALGORITHM (TLBOGAH FOR COLOR IMAGE SEMENTATION)**

In this section a vivid presentation of the proposed hybrid algorithm “TLBOGAH” in the context of its application to color image segmentation problem is shown in Fig 1

**Fig.1.Block Diagram of Proposed Algorithm**

\[ \text{Start} \]

\[ \text{Initialization of Population} \]

\[ \text{Fitness Assessment of all Chromosomes} \]

\[ \text{Note the best chromosome} \]

\[ \text{Termination Criteria met} \]

\[ \text{Output the Optimal Solution} \]

\[ \text{Apply roulette wheel selection on} \]

\[ \text{Apply TLBO Teacher Learning on color (real) part of chromosome} \]

\[ \text{Apply GA crossover operation on texture (binary) part of chromosome} \]

\[ \text{Apply TLBO Student Learning on color (real) part of chromosome} \]

\[ \text{Fitness evaluation of next (children) generation} \]

\[ \text{Form next generation population by considering elite parent and meritorious child chromosomes} \]

A **Brief Review of Genetic Algorithm Extraction:**

Evolutionary Genetic Algorithms are heuristic global search methods that mimic the process of natural selection and uses fixed-length strings to represent possible solutions. GA is driven by a fitness function defined to evaluate a solution’s ability to deal with a given task, ending up in bringing out an optimum solution. The GA is an ensemble application of three major operations selection, crossover and mutation, which contribute to the task of chromosome variation. A pseudo code of simple ge-
A Genetic algorithm is shown in Fig 2. An interested reader is referred to the review on Genetic Algorithm.[17]

\textbf{function genetic algorithm ( )}
\begin{align*}
\text{Initialize population;}
\text{Calculate fitness function;}
\text{while (fitness value != termination criteria) }
\begin{cases}
\text{Selection;}
\text{Crossover;}
\text{Mutation;}
\text{Calculate fitness function;}
\end{cases}
\end{align*}
\text{) }

\textbf{Pseudo code of Genetic Algorithm}

\textbf{A. Brief Review of TLBO Extraction:}

TLBO is a simple yet powerful EA for real parameter optimization proposed by Rao et al [24]. TLBO simulates the teaching learning process and is based on the effect of the influence of a teacher on the output of learners in a class. TLBO consists of two phases. 1) Teacher Phase 2) Learner Phase. In the teacher phase all the students learn from the teacher and in the learning phase the learners learn from each other. In recent years TLBO has been applied to a number of real world problems due to its simple and robust nature. Interested readers are referred to [22] for comprehensive review on TLBO.

\textbf{function TLBO ( )}
\begin{align*}
\text{Initialize } P = (x_1, x_2, \ldots, x_N), \text{ (N points in D)}

\text{While ( fitness vale != termination criteria ) }
\begin{cases}
\text{for } i := 1 \text{ to } N \text{ do}
\text{for } i := 1 \text{ to } N \text{ do}
\text{compute new population;}
\text{Teacher Phase}
\text{New } X_i = X_i + \text{Difference } \text{Mean}
\text{Difference } \text{Mean} = r \times (\text{Teacher } - \text{TF x Mean})
\text{TF = round } [1 + \text{rand } (0, 1)]
\text{Learner Phase}
\text{New } X_i = X_i + r \times (X_j - X_i) \text{ if } f (X_i) > f (X_j)
\text{xi + r x (Xj - Xi) otherwise}
\end{cases}
\end{align*}

\textbf{Pseudo code of TLBO}

\textbf{V. PERFORMANCE MEASURES}

There exists many segmentation evaluation measures in the literature viz [6] sensitivity, specificity, Precision, Recall, ROC, F-measure, Local consistency Error, Global consistency Error etc. The Performance measures proposed by Unmi Krishnan et al., [16] which are Rand Index (RI), Variation of Information (VOI), Global Consistency Error (GCE), and Boundary Displacement Error (BDE) are used in evaluating and comparing our segmentation results with benchmark algorithms.

\textbf{A. Rand Index}

The Rand index indicates the proportion of pixels which are in agreement between the Computed Segmentation (CS) and the Ground Truth (GT). [16]. The rand index is given by the formula

\[ R = \frac{A + B}{A + B + C + D} \]

- \( A \) - The pairs of pixels that are in similar sets of CS and GT.
- \( B \) - The pairs of pixels that are in dissimilar sets of CS and GT.
- \( C \) - The pairs of pixels that are in similar sets of CS and disimilar sets of GT.
- \( D \) - The pairs of pixels that are in dissimilar sets of CS and similar sets of GT.

\( A+B \) is the number of coincidences between CS and GT and \( C+D \) is the number of non-coincidences between CS and GT. The rand index ranges between 0 and 1, where 0 confirms that CS and GT do not have common attributes and 1 confirms that CS and GT are indistinguishable.

\textbf{B. Variation of Information}

The variation of information (VOI) is a measure that specifies the variation between computed segmentation and ground Truth. The difference between average conditional entropy of computed segmentation (CS) and Ground Truth (GT) is used to measure the vagueness in CS which cannot be expressed by GT.

Let X and Y be the computed segmentation and Ground Truth which are defined as \( X = \{x_1, x_2, x_3, \ldots, x_i\} \) and \( Y = \{y_1, y_2, y_3, \ldots, y_j\} \)

\[ n = \sum |X_i| = \sum |Y_j| \]

\[ r_{ij} = \frac{\sum |X_i \cap Y_j|}{n} \]

The variation of information between CS and GT is given by

\[ \text{voi}(X, Y) = - \sum_{i,j} r_{ij} \left[ \log \left( \frac{r_{ij}}{p_i} \right) + \log \left( \frac{r_{ij}}{q_j} \right) \right] \] \hspace{1cm} (14)

The lower is the value of VOI, the better is the segmentation result.

\textbf{C. Global Consistency Error}

Global consistency error is a measure of the limits to which the computed segmentation can be seen as transformation of Ground Truth towards Computed Segmentation. Similar segmentations match, as both have genesis in the same image, but undergo segmentation at different scales. If one segment is proper subset of the other, then the pixel lies in an area of refinement, and the error should be zero. If there is no subset relationship, then the two regions overlap. The formula for GCE is as follows

\[ \text{GCE} = \frac{1}{n} \min \left[ \sum_{i} E(s_i, s_2, p_i) \sum_{i} E(s_2, s_1, p_i) \right] \] \hspace{1cm} (15)

GCE ranges between 0 and 1 where 0 signifies no error. Lower the value of GCE better is the segmentation result.

\textbf{D. Boundary Displacement Error}

The Boundary Displacement Error is a measure of the displacement error averaged between boundary pixels in computed segmentation and the nearest boundary pixels in the ground truth. BDE should be low for good segmentation.

\textbf{VI. RESULTS AND DISCUSSION}

Deng et al., [9] proposed the well-known J-SEGmentation (JSEG) algorithm, which combines both quantization pro-
cess and clustering techniques for extraction of color-texture cues in images. Mean Shift clustering in sync with edge information was employed by Christoudias et al. [6] in their work using edge detection and image segmentation (EDISION) system. The proposed algorithm is applied on natural color images obtained from Berkeley Segmentation Database. The results in the proposed algorithm are compared with the authors work in [27], GA without hybridization, and also the works in [6] and [9] which are known to be bench mark algorithms in the field of color segmentation.

![Images of original and segmented images](image-url)
TABLE I
RAND INDEX & VARIATION OF INFORMATION

<table>
<thead>
<tr>
<th>Image</th>
<th>RI</th>
<th>VOI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEGa</td>
<td>GA</td>
</tr>
<tr>
<td>Bear</td>
<td>0.69</td>
<td>0.39</td>
</tr>
<tr>
<td>Boat</td>
<td>0.62</td>
<td>0.57</td>
</tr>
<tr>
<td>Church</td>
<td>0.73</td>
<td>0.72</td>
</tr>
<tr>
<td>Horse</td>
<td>0.65</td>
<td>0.62</td>
</tr>
<tr>
<td>Tiger</td>
<td>0.82</td>
<td>0.71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DEGa</th>
<th>GA</th>
<th>JSEG</th>
<th>EDISON</th>
<th>TLBOGAH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE II
GLOBAL CONSISTENCY ERROR & BOUNDARY DISPLACEMENT ERROR

<table>
<thead>
<tr>
<th>Image</th>
<th>GCE</th>
<th>BDE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEGa</td>
<td>GA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES


[28] Lei Wang, Feng Zou, Xinhong Hei, Lei Wang, Feng Zou, Xinhong Hei, A hybridization of teaching-learning-based optimization and differential evolution for chaotic time series prediction, 2014, Volume 25, Number 6, Page 1407


ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)