



Saliency Detection by MICCLLR

Rainu R PadamCSE Department, VJCET, MG University
India**Nimmy George**CSE Department, VJCET, MG University
India

Abstract—Saliency detection means detecting visually attractive regions in images. It is an aspect of exploring visual attention from a computer vision. Each image is segmented to get bags. Features are extracted from each bag. Features, including low-, mid-, and high-level, are incorporated into the learning and testing process. They are position, color, texture, scale, center prior, and boundary. From these features, meta-instance is calculated for each bag. For detecting the salient region, a classifier is learned with meta-instance. In the existing system, saliency value is calculated using EM-DD algorithm and learned using multiple-instance learning. The idea of EMDD is to model the label of each bag with a hidden variable, which is estimated by the expectation-maximization (EM) algorithm. In the proposed system for improving the accuracy of saliency map, algorithm called MICCLLR is used for meta-instance creation. It is a Generalized Multiple-Instance Learning Algorithm Using Class Conditional Log Likelihood Ratio that converts the MI data into a single meta-instance data allowing any propositional classifier to be applied. Each image is tested with the learned model. Experiments show that, MICCLLR algorithm is better than the EM-DD algorithm and will give better saliency map.

Keywords— Saliency, saliency map, meta-instance, machine learning, computer vision.

I. INTRODUCTION

Saliency detection means detecting salient objects in an image. When looking to an image, people concentrate on a particular region in that image. That region is called salient region. Saliency detection has many applications like image resizing, summarization, collage creation etc. So the accuracy of saliency map has a great importance. The ability of withdraw from some things in order to deal effectively with others is called attention. Here, one particular aspect of visual attention is explored from a computer vision viewpoint called saliency detection. The task of saliency detection is to extract the salient objects in an image. First, low-level features, such as color, position, texture, and scale are selected as the basic elements for supporting saliency detection. Second, the saliency value for each pixel in an input image is calculated according to a predefined model. In the end, saliency maps from different sources are integrated and normalized to get a final result. The saliency value is generally denoted by a number scaled to [0, 1] and shown in a gray image. The greater value the pixel has, the higher possibility it is of being salient. Research on visual attention has found use in many applications eg, object recognition, image segmentation, content based retargeting, image retrieval, adaptive image display, and advertising design.



Fig1 (a) Input image. (b) Saliency map

II. RELATED WORKS

There are many algorithms for saliency detection. These algorithms for saliency detection can be generally categorized as model based and computation based.

A. Model based algorithms

In these algorithms, a high-level model is first established empirically. Then, the subsequent calculation is conducted with respect to the defined model. There are works with bionics model to intimate an attention shift of human and primate. A neuron-like network is employed to combine several topographical parallel saliency maps from different image clues. Then, a winner-take-all mechanism keeps the selective attention point drifting from one conspicuous location to another. Another architecture, is to extract multiscale features through a set of linear center surround operations. Then, the focus of attention is determined by combining and normalizing the across-scale maps.

B. Computation based algorithms

Saliency of this type is typically calculated by contrast from low-level features. Some algorithms have two-stage approach to saliency detection. In the first stage, the spectrum residual model is extended by introducing two modules, namely, automatic channel selection and decision reversal. In the second stage, incomplete salient regions are propagated based on the basic Gestalt grouping principles. Algorithm that outputs saliency maps with well-defined boundaries of salient objects, which are obtained by retaining more frequency content from the input image than previous techniques. Saliency is detected with respect to the hypothesis that the scale of an object relates to the image borders. Thus varied the bandwidth of the center surround-filtering near-image borders using symmetric surrounds to detect saliency.

III. SYSTEM OVERVIEW

The system consists of two phases. They are training and testing. In training phase, a model is learned for distinguishing salient and non-salient regions in images. Testing phase is for finding salient regions in new images.

A. Training the model

For training, a data set is chosen which contains images and their ground truth. Each image is segmented using mean shift segmentation to get bags. Features are extracted from each bag. Meta-instance is also created for each bag. The model is trained with meta-instance and its label using SVM. Each bag is assigned label according to the ground truth. Label 1 represents salient bag and Label -1 represents non-salient bag. SVM will train a model to distinguish the salient and non-salient regions in images.

B. Testing the model

The image for testing is fed to the trained model. The image is segmented using mean shift segmentation to get bags. Features are extracted from each bag. Meta-instance is also created for each. The meta-instances are fed to the trained model and it will return label for each bag.

IV. MEAN SHIFT SEGMENTATION

The method for segmentation is taken from the paper, Mean shift: A robust approach toward feature space analysis[3]. Each image is segmented using mean-shift algorithm to get bags. Mean shift considers feature space as a empirical probability density function. If the input is a set of points then Mean shift considers them as sampled from the underlying probability density function. If dense regions are present in the feature space, then they correspond to the mode of the probability density function. Clusters associated with the given mode can also be identified using Mean Shift. For each data point, Mean shift associates it with the nearby peak of the dataset's probability density function. For each data point, Mean shift defines a window around it and computes the mean of the data point. Then it shifts the center of the window to the mean and repeats the algorithm till it converges. After each iteration, consider that the window shifts to a more denser region of the dataset. Steps followed in mean shift algorithm are

1. Fix a window around each data point.
2. Compute the mean of data within the window
3. Shift the window to the mean and repeat till convergence.



A. (b)

Fig 2 (a) is the input image, (b) is the output of mean shift segmentation.

After segmentation, a set of pixels will have a common value. Another set of pixels have another common value. Each of these sets are considered as separate bags. Next, each bag is separated and separate colors are given indicate each bags i.e., each bag will be colored differently.



Fig 3. (a) is the input image. (b) shows different bags.

V. FEATURE EXTRACTION

A. Low-level feature

1. **Position:** The spatially connected pixels are prone to share similar saliency, whereas pixels far away tend to be differently salient. Therefore, the position of each instance is an essential factor for keeping the saliency consistent. Since the sizes of images differ, the absolute horizontal and vertical position is not suitable for an optimal feature. To avoid this problem, the normalized position within the range of [0, 1] is adopted to ensure that the measurement.
2. **Color:** This is the most frequently employed feature. Almost every algorithm for saliency detection will refer to it as the major supporting information for saliency calculation. HSV is selected for this work. With the HSV color space, color contrast is defined for each pixel

$$S_c(i, R_i) = \sum_{j \neq i, j \in R_i} d_c(i, j) \quad (1)$$

where i is the examined pixel, R_i is the supporting region for defining the saliency of pixel i , and $d_c(i, j)$ is the distance of color descriptors between i and j .

3. **Texture:** Different organizations of pixels form different textures, which would provide us with descriptively perceptual information. For saliency detection, this is also a significant feature. The perceived textures can be described by different ways but are generally characterized by the outputs of a set of filters. As an example, the filter bank used in this paper is made of copies of a Gaussian derivative and its Hilbert transform, which model the symmetric receptive fields of simple cells in visual cortex. To be more specific, they are

$$F_1(x, y) = \frac{d^2}{dy^2} \left(\frac{1}{C} \exp\left(\frac{y^2}{\sigma^2}\right) \exp\left(-\frac{x^2}{l^2 \sigma^2}\right) \right) \quad (2)$$

$$F_2(x, y) = \text{Hilbert}(F_1(x, y)) \quad (3)$$

where σ is the scale, l is the ratio of the filter, and C is a normalization constant. Similarly, texture contrast for each pixel is defined as

$$S_t(i, R_i) = \sum_{j \neq i, j \in R_i} d_t(i, j) \quad (4)$$

where R_i and $d_t(i, j)$ have an analogous meaning with color contrast

4. **Scale:** Scale is an effective property for identifying objects of different sizes. It is interpreted as a low-level feature in this paper. The approach taken to incorporate this feature follows. To be more specific, the difference between fine and coarse scales of color images is extracted to simulate the center-surround operations of visual receptive fields. Such an architecture is particularly well suited to detecting the standing-out locations from their surroundings.

B. Middle-level feature

1. **Center-prone prior:** Several eye-tracking experiments have shown that people pay more attention to the center of an image. So pixels at the center of image is given more weight than pixels lie far from the center.

C. High-level feature

1. **Boundary:** Saliency is related to human priors and perception. Low-level features can provide kinds of supporting information to determine the saliency level. However, the involvement of a high-level feature is helpful for the delineation of salient objects. In this work, boundary is taken into consideration as an example for utilizing the high-level feature. Boundary is different from what is traditionally known as edges. Edge is a low-level feature that indicates changes such as brightness, color, or texture. However, boundary implies the ownership from one object to another. Assumption is that, if there is a boundary in a region, it is more possible to indicate a salient object within this region. Boundary is treated as an identifier to infer the salient objects. To get the desired boundary, a learning-based approach is used to train a logistic regression model, which integrates a set of image cues to output a probability boundary.

VI. META-INSTANCE CREATION

Saliency for each bag is calculated according to the learned model. A classifier is trained with the meta-instances by SVM. Meta-instances are created using MICCLR algorithm. It is a Generalized Multiple-Instance Learning Algorithm Using Class Conditional Log Likelihood Ratio that converts the MI data into a single meta-instance data allowing any propositional classifier to be applied. In statistics, a likelihood function is a function of the parameters of a statistical model. The likelihood of a set of parameter values, θ , given outcomes x , is equal to the probability of those observed outcomes given those parameter values.

In statistics, a likelihood ratio test is a statistical test used to compare the fit of two models, one of which (the null model) is a special case of the other (the alternative model). The test is based on the likelihood ratio, which expresses how many times more likely the data are under one model than the other. This likelihood ratio, or equivalently its logarithm, can then be used to compute a p-value, or compared to a critical value to decide whether to reject the null model in favor of the alternative model. When the logarithm of the likelihood ratio is used, the statistic is known as a log-likelihood ratio statistic, and the probability distribution of this test statistic, assuming that the null model is true, can be approximated. Conditional likelihood is conditioning on sufficient statistic for the nuisance parameters, results in a likelihood which does not depend on the nuisance parameters.

Features of each bags are taken and mean values for 2 clusters are calculated using k means clustering. The features in each bags are clustered in to 2 sets using Gaussian-mixture distribution. Then Posterior probabilities of each feature is calculated. ie, probability of each feature to be in one of the two clusters. ie, It will return two probabilities for each feature. Convert each bag to a single meta-instance. Meta-instance is the ratio of each probability multiplied by the reciprocal of the number of pixels in the bag. The model is trained using SVM with meta-instances and their corresponding labels.

$$\text{Meta-instance} = \frac{1}{m_i} \ln \sum_{l=1}^{m_i} \frac{\Pr(x_{ilq} = a_q | c=1)}{\Pr(x_{ilq} = a_q | c=-1)} \quad (5)$$

Where m_i is the number of pixels in the bag. X_{ilq} is the value of feature of instance in bag. The posterior probability of a random event or an uncertain proposition is the conditional probability that is assigned after the relevant evidence is taken into account. Similarly, the posterior probability distribution is the probability distribution of an unknown quantity, treated as a random variable, conditional on the evidence obtained from an experiment or survey. Posterior, in this context, means after taking into account the relevant evidence related to the particular case being examined.

VII. SALIENCY MAP CREATION

Saliency map is created according to the label returned from the trained model. If the label is 1 then the pixels in that bag is assigned value 1. If the label is -1 then the pixels in that bag is assigned value 0.

VIII. EXPERIMENTAL RESULTS

A. Data Set

In order to evaluate the performance of the proposed algorithms, a data set is constructed by Achanta *et al.* and has achieved great popularity in saliency detection. Every image in the data set has a ground-truth label. In this case, the experimental results can be evaluated quantitatively.

B. Output comparison

The results of the proposed algorithms are compared with an existing system-saliency detection by multiple-instance learning. Each output is compared with its ground truth. Fig 4 shows comparison of the proposed system with the existing system.

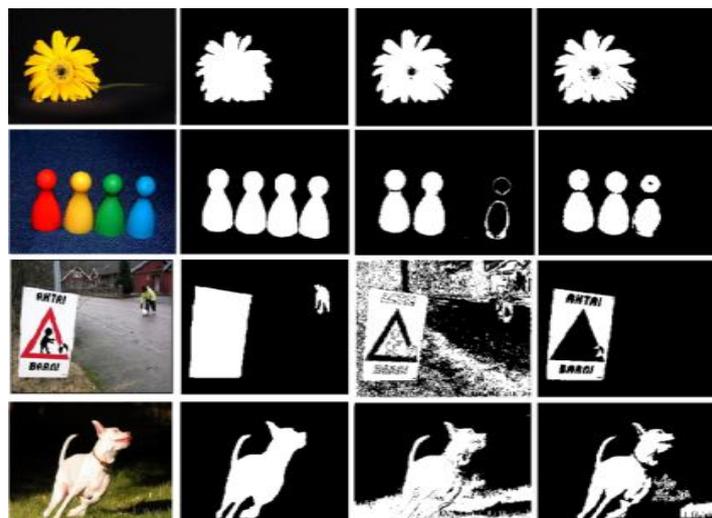


Fig 4. (a) is the input images. (b) shows their corresponding ground-truth. (c) shows output of an existing system. (d) shows the output of the proposed system.

C. Performance

In the experiments, the precision-recall measure is employed to evaluate the performance. It is a parametric curve that captures the trade off between accuracy and noise as the threshold varies. To get a better understanding of these two indexes, *true positives* (TP), *false positives* (FP), and *falsenegatives* (FN) should be first introduced. For an information retrieval problem, suppose there are two classes—*positive* and *negative*. *True positives* are the items that are correctly labelled as the positive class, *false positives* are the ones incorrectly labelled as the positive class, and *false negatives* are the ones which are not labelled as the positive class but should have been. Based on these introductions, *precision* is defined as the rate of true positives divided by the whole labelled positive items, whereas *recall* is defined as the rate of true positives divided by the actual number of items belonging to the positive class. Fig5 shows the performance analysis of the proposed system and the existing system. It shows that proposed system is better than the existing system.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

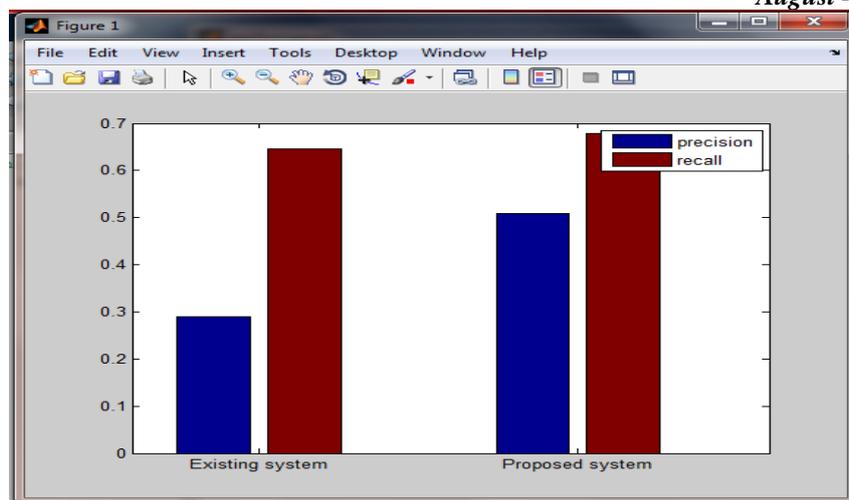


Fig 5.Performance analysis of proposed system with an existing system.

IX. CONCLUSION

The proposed system detects salient regions in images. It uses MICCLLR algorithm for learning the classifier. MICCLLR is a generalized multiple-instance learning algorithm using class conditional log likelihood ratio that converts the mi data into a single meta-instance data allowing any propositional classifier to be applied. The classifier is learned using svm with the meta-instances created by miccllr algorithm. The proposed system gives consistent and accurate saliency maps and outperforms the existing system.

REFERENCES

- [1] Qi Wang, Yuan Yuan, "Saliency Detection by Multiple-Instance Learning," IEEE Transactions On Cybernetics, vol. 43, no. 2, april 2013
- [2] Yasser EL-Manzalawy^{1,2} and Vasant Honavar¹, "MICCLLR: Multiple-Instance Learning using Class Conditional Log Likelihood Ratio" Department of Computer Science,Iowa State University
- [3] D. Comaniciu and P.Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603–619, May 2002.
- [4] L.Itti,C. Koch, and E.Niebur, "A model of saliency based visual attention for rapid scene analysis,"IEEETrans.Pattern Anal. Mach.Intell,vol.20,no.11,pp,Nov 1998
- [5] Q.Zhang and S.A.Goldman , "EM-DD: An improved multiple-instance learning technique," in Proc.Adv.Neural Inf.Process.Syst.2001