Group 31

Ujjwal Tiwari(2018900100), Harshit Kumar Sankhla(2018900050), Udit Singh Parihar(2018701024)

Computer Vision

Saliency Filters:

Contrast based saliency estimation

Figure 1: From left to right: input images, image abstraction into perceptually homogeneous elements, results of our saliency computation, ground truth labeling.

Project Overview

Saliency estimation is a challenging computer vision task of identifying image elements that are likely to catch the attention of a human observer. These computational methods are challenging due to the complex cross disciplinary nature of the problem. Methods that can find salient image regions find wide application base in image processing tasks such as image resizing, object detection and image segmentation.

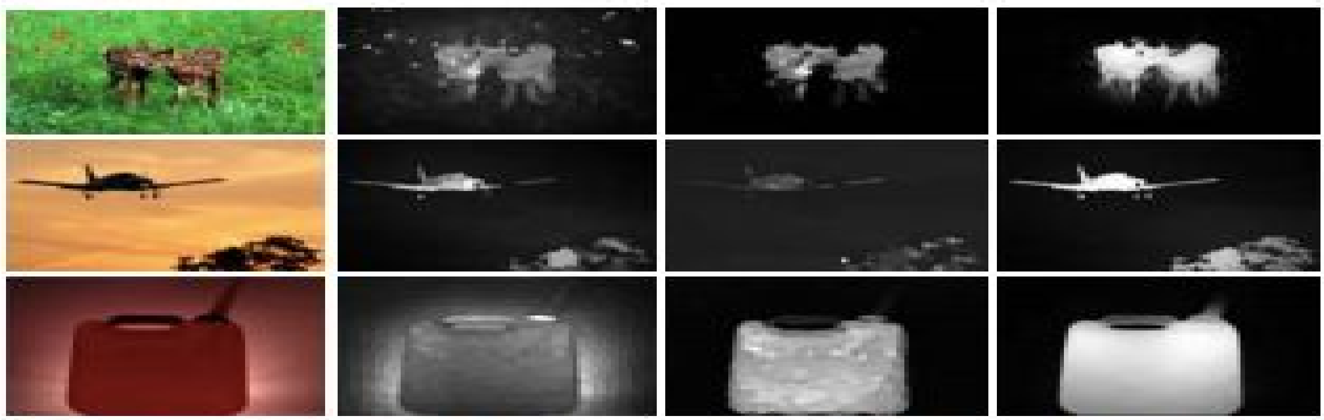
High- level methods take into account insights from a variety of interdisciplinary fields such as neuroscience, computer vision, biology etc. However, low level feature based computational methods based on contrast are quite successful.

The definition of contrast in related works are often related to color variation of individual pixels, edges, gradients, , spatial frequencies, structure and distribution of image patches, histograms, multi-scale descriptors, or combinations thereof. This ambiguous nature of contrast measure fails to attribute improvements in results to design considerations of specific algorithms.

As a part of this project we explore a intuitive yet effective algorithm proposed in CVPR'12 for contrast based saliency estimation. The proposed work improves upon previous work by formulating the problem of saliency estimation in four different steps:

1. **Finding Super Pixels:** Decomposition of Image into perceptually homogeneous clusters using a variant of SLIC superpixels. In this varint the superpixels are computed using K-Means algorithms in CIElab color space rather than in RGBXY space, this accounts for better compactness other than just being local and edge-aware superpixels. Geodesic image distance guarantees connectivity, while retaining the locality, compactness and edge awareness of SLIC superpixels.
2. **Uniqueness (first measure of contrast):** The notion of pixel wise saliency is derived from two different measures of contrast. The first of the these measures is element uniqueness. The uniqueness of every element found in the first step is calculated. We therefore evaluate how different each respective element is from all other elements constituting an image, essentially measuring the "rarity" of each element.
3. **Element Distribution (second measure of contrast):** Element uniqueness scores are conceptually weak directives to pixel saliency estimation as background superpixels can be just as unique as the ones that constitute the actual salient object. Therefore, the intuition behind this second measure of contrast is that salient features in an image will have low spatial variance, which renders unique elements more salient when they are grouped in a particular image region rather than evenly distributed over the whole image.
4. **Saliency Estimation:** In the final step, the two above contrast measures that are defined on a per-element level are clubbed together to assign the actual saliency values to the input image to get a pixel-accurate saliency map

Central to the contrast and saliency computation is our second main contribution; we show that all involved operators can be formulated within a single high-dimensional Gaussian filtering framework. This formulation leads of highly efficient implementation with linear complexity.



(a) Source image. (b) Uniqueness. (c) Distribution. (d) Saliency.

Figure 3: Uniqueness, spatial distribution, and the combined saliency map. The uniqueness prefers rare colors, whereas the distribution favors compact objects. Combined together those measures provide better perfomance.

Implementation:

We have implemented the first two steps namely, abstraction and element uniqueness of this paper till first evaluation. This is a python implementation which will be open sourced upon final submission and evaluation of this project.

For the implementation of this algorithm we have used the following python packages:

- OpenCV2.0
- Scipy
- Scikit-image
- Numpy

Final requirements will be made available in a requirements.txt file.

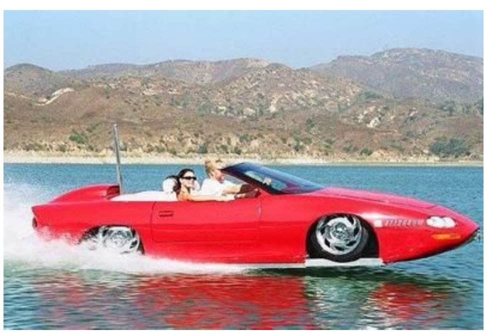
1. **ABSTRACTION:**

   In this step we abstract away the texture in the image by using sklearn based SLIC module implementation of K-means clustering algorithm. Our results are close to that provided by the original paper.

   K-means is a unsupervised learning algorithm that clusters together similar data entries based on a distance measure on the feature space. Number of image segments of this experiment is taken to be 300.

   Refer below to the results section for generated Superpixels on test image.

Results:



2. **ELEMENT UNIQUENESS:**

Element uniqueness is calculated by the mathematical formulation provided in the paper:

$$U_i = \sum_{j=1}^{N} \|\mathbf{c}_i - \mathbf{c}_j\|^2 \cdot \underbrace{w(\mathbf{p}_i, \mathbf{p}_j)}_{w_{ij}^{(p)}} .$$

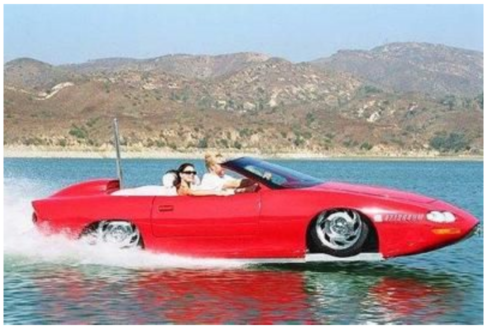
- The weights provide control over the global and local contrast estimation with radius of influence of the uniqueness operator.
- A local function yields a local contrast term, which tends to overemphasize object boundaries in the saliency estimation. whereas w (p) ij ≈ 1 yields a global uniqueness operator, which cannot represent sensitivity to local contrast variation.

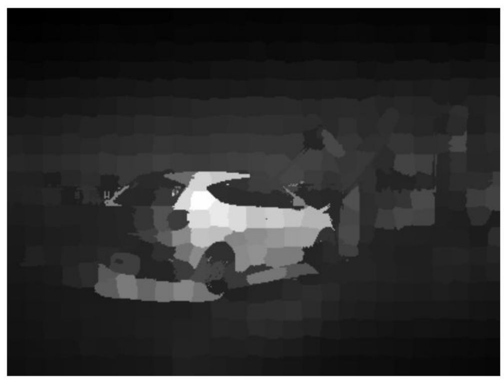
$$\begin{aligned} U_i &= \sum_{j=1}^{N} \|\mathbf{c}_i - \mathbf{c}_j\|^2 w_{ij}^{(p)} \\ &= \mathbf{c}_i^2 \underbrace{\sum_{j=1}^{N} w_{ij}^{(p)}}_{1} - 2\mathbf{c}_i \underbrace{\sum_{j=1}^{N} \mathbf{c}_j w_{ij}^{(p)}}_{\text{blur } c_j} + \underbrace{\sum_{j=1}^{N} \mathbf{c}_j^2 w_{ij}^{(p)}}_{\text{blur } c_j^2} .(\end{aligned}$$

- Using gaussian Kernel reduces time complexity from O(N^2) to O(N). It is advantageous over other methods that subsample the image to a lower resolution which leads to blurring of images.

Refer to the below results section for the Image filters and Uniqueness estimation on test image.

Results:





Computation of element distribution and saliency estimation will be done in phase of the project development.

Find the git repository to all project code.

[git_link]