Indian Statistical Institute, Kolkata

M. Tech. (Computer Science) Dissertation

Detecting Salient regions in Image using visual attention

A dissertation submitted in partial fulfillment of the requirements
for the award of Master of Technology
in
Computer Science

Author:
Pritish Ranjan Pradhan
Roll No: CS-1426

Supervisor:
Dr. Bhabatosh Chanda
ECSU Unit, ISI

M.Tech(CS) DISSERTATION THESIS COMPLETION CERTIFICATE
Student: Pritish Ranjan Pradhan (CS1426)
Topic: Detecting Salient regions in Image using visual attention
Supervisor: Dr. Bhabatosh Chanda

This is to certify that the thesis titled "***Detecting Salient regions in Image using visual attention***" submitted by **Pritish Ranjan Pradhan** in partial fulfillment for the award of the degree of Master of Technology is a bonafide record of work carried out by him under my supervision. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in my opinion, has reached the standard needed for submission.

Date: Dr.Bhabatosh Chanda

Declaration

I, **Pritish Ranjan Pradhan (CS1426)**, registered as a student of M. Tech program in Computer Science, Indian Statistical Institute, Kolkata do hereby submit my Dissertation Report titled **" Detecting Salient Regions in image using Visual Attention"**. I certify

1. The material contained in this Dissertation Report has not been submitted to any University or Institute for the award of any degree.
2. I followed the guidelines provided by the Institute in preparing the report.
3. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of report and giving their details in the bibliography.

.....

Place : ISI, Kolkata **Pritish Ranjan Pradhan**
Date : July, 2016 **(CS1426)**

Acknowledgements

I would like to thank my dissertation supervisor Dr. Bhabatosh Chanda for agreeing to guide me and for helping me to undertake work in the topic.

Last but not the least I am grateful to all my lab mates, batchmates and seniors of Image processing Lab of Indian Statistical Institute, Kolkata for helping me though out the project with their valuable suggestions.

Abstract

We see a wide range of images everyday. In fact around 1 Giga bits of data enters our eyes every second. However we do not focus on all of this data. Our brain reduces the amount of visual data through high-level cognitive and complex processes. This process is called visual attention and a selection mechanism lies at the core of the process.

In the method of salient region detection here, bottom up approach has been used. So no prior information or assumption on the object of interest is used. The algorithm tries to detect objects in images that are different from the rest of the image and consequently capture our attention.

In the method here we first pre-process the image by doing albedo correction. We then do a bilateral filtering of the image in *Lab space*. We find the mean of each of the individual *L, a and b* channels and then subtract the mean from each of the *L, a and b* channels. This acts as a band pass filter. We then create the saliency map by taking the L2 norm of the mean subtracted *L, a and b* channels. Finally we divide the saliency map into superpixels of appropriate size and select the most salient superpixels that are not lying on the boundary and get information regarding the object of attention. The resulting segmented image gives the salient region that is more likely to grab our attention in case if such an image is presented to us.

Contents

List of Algorithms

List of Figures

Chapter 1

Introduction

Twenty-first century is filled with a range of gadgets that can capture photos within a fraction of seconds. Even cameras are integrated to mobile phones and hand held devices. We have several inexpensive gadgets that can capture a wide range of images. So the number images available is numerous. Humans have an inherent capacity to look at images find the desired object and use the information available for further uses. As the number and volume of images increase, we would want that these images could be even analysed on a machine and important regions could be found out. In these cases, salient region detection would be desirable. So this can help in other image processing applications like image storage and retrieval and also in image data mining.

1.1 Motivation

A particular approach to solve this problem is to build a system which can take RGB images and gives a segmented image as output which contains the most salient region that is likely to catch attention. In this project we developed such a system with some technique. Also some results have been given in subsequent sections.

There are large number of area in which salient region detection can be applied.One of the areas can be automated image annotation. We can have several training images which are labelled based on the salient object present in the image. We can find the salient object and extract some feature descriptors from it. Now given another input image, we can extract salient object and corresponding feature descriptors. We can use the corresponding feature descriptors and match the available images in the database and can annotate the image automatically.

1.2 Our work

In our method we have tried to segment the given input image so that the most salient object which is likely to be visually attentive gets more importance. This can be done by creating a saliency map. The saliency map would assign more weight to the salient region

and corresponding areas so that the object can be segmented out. The segmented image would be the visually attentive part of the image and would be distinct.

Chapter 2

The Gestalt Principles

Gestalt is a psychology term which means "unified whole". It refers to theories of **visual perception** developed by German psychologists in the 1920s. These theories attempt to describe how people tend to organize visual elements into **groups** or **unified wholes** when certain principles are applied. The principles are :

- **Similarity**
  Similarity occurs when **objects look similar** to one another. People often perceive them as a group or pattern.

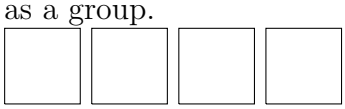
  The example above (containing 4 distinct objects) appears as as **single unit** because all of the shapes have **similarity**.
  Unity occurs because the rectangular shapes **look similar**.
  When similarity occurs, an object can be emphasised if it is dissimilar to the others. This is called **anomaly**.

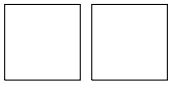
- **Closure**
  Closure occurs when an object is **incomplete** or a space is not completely enclosed. If enough of the shape is indicated, people perceive the whole by filling in the missing information.

Although the moon shown above is not complete, enough is present for the eye to complete the shape. When the viewer's perception completes a shape, closure occurs.

- **Proximity**

  Proximity occurs when elements are placed close together. They tend to be perceived as a group.

  The four squares are put together are perceived as a group. The remaining two squares form a group among them.

- **Figure and Ground**

  The eye differentiates an object form its surrounding area. a form, silhouette, or shape is naturally perceived *as figure* (object), while the surrounding area is perceived **as ground** (background).

  Balancing figure and ground can make the perceived image more clear. Using unusual figure/ground relationships can add interest and subtlety to an image.

  CLEAR

  We can read the letters C, L, E, A and R and distinguish it from the background. Foreground and background are separate.

- **Law of symmetry**

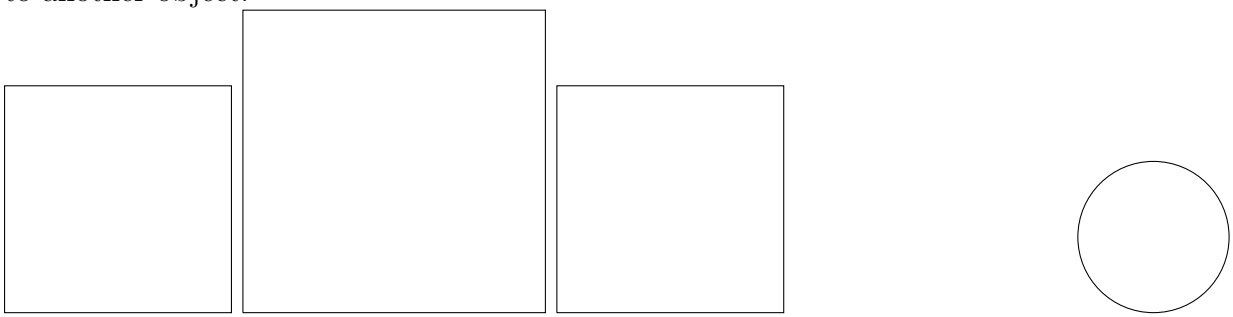
  The law of symmetry captures the idea that when we perceive objects we tend to perceive them assymmetrical shapes that form around their centre. Most objects can be divided in two more or less symmetrical halves and when for example we see two unconnected elements that are symmetrical, we unconsciously integrate them into one coherent object (or percept). The more alike objects are, they more they tend to be grouped.

  [ ] [ ] [ ]

  We perceive the above as three pair of brackets instead of six brackets.

- **Continuation**

  Continuation occurs when the eye is compelled to **move through** one object and **continue** to another object.



  Continuation occurs in the example above, because the viewer's eye will naturally follow the square pictures to find out what is in the end.

Chapter 3

Visual attention and saliency

According to **William James**, *Principles of Psychology* **attention** is the process of taking possession by the mind in clear and vivid form,out of several simultaneously possible objects or trains of thought. It implies withdrawal from some things in order to deal effectively with others.

3.1 Visual attention

Visual attention is said to operate in two stages:

1. 1. In the first stage, attention is distributed uniformly over the entire perceived visual scene and processing of information is performed in parallel.

2. 2. In the second stage, attention is concentrated to a specific area of the visual scene (i.e., it is focused), and processing is performed in a serial fashion.

There are several approaches to visual attention modelling. Some of the well known models are:

- ***Bottom-Up* versus Top-Down Models**
  Bottom-up cues are mainly based on the characteristics of a visual scene (it is stimulus-driven), whereas top-down cues are determined by cognitive phenomena like knowledge, expectations, reward, and current goals(it is goal-driven).
  Bottom-up attention is fast, involuntary, and feed-forward. An example of bottom-up attention is looking at a scene with only one horizontal bar among several vertical bars where attention is immediately drawn to the horizontal bar.
  Top-down attention is slow, task driven, voluntary, and closed-loop . One of the most famous examples of top-down attention guidance is from *Yarbus* in 1967 , who showed that eye movements depend on the current task with the following experiment: Subjects were asked to watch the same scene (a room with a family and an unexpected visitor entering the room) under different conditions (questions) such as “estimate the

material circumstances of the family," "what are the ages of the people?", or simply to freely examine the scene. Eye movements differed considerably for each of these cases.

- ***Spatial* versus Spatio-Temporal Models**
  In the real world, we are face visual information that constantly changes due to dynamics of the world. Visual selection is then dependent on both current scene saliency as well as the accumulated knowledge from previous time points. Therefore, an attention model should be able to capture scene regions that are important in a spatio-temporal manner.
  Almost all attention models include a *spatial* component. On the other hand, some models aim to capture the *spatio-temporal* aspects of a task, for example, by learning sequences of attended objects or actions as the task progresses.

- **Overt versus Covert Attention**
  Attention can be differentiated based on its attribute as *overt* or *covert*.
  Overt attention is the process of directing the fovea towards a stimulus, while **covert attention** is mentally focusing onto one of several possible sensory stimuli. An example of covert attention is staring at a person who is talking but being aware of visual space outside the central foveal vision. Another example is driving, where a driver keeps his eyes on the road while simultaneously covertly monitoring the status of signs and lights.
  The current belief is that covert attention is a mechanism for quickly scanning the field of view for an interesting location.

The visual attention mechanism may have at least the following basic components *[Tsotsos, et. al. 1995]*:

1. **The selection of a region of interest in the visual field**
   In visual search,in cases where explicit targets are given in advance, has time complexity which is linear in the size of the image. On the other hand, if no such explicit target is provided, the task is NP-complete. Thus, it may be concluded that the brain is not solving this general problem .
   There are an exponential number of such image subsets. Attentional selection may determine which mapping to attempt to verify first. If the first such mapping selected is a good one, a great deal of search can be avoided, otherwise there is the potential for a very inefficient search process. For sufficiently small images and/or sufficiently massive computational power, the brute-force search strategy will work perfectly well without attention. For most of the tasks, this brute-force approach fails.

2. **The selection of feature dimensions and values of interest**
   Search within feature space seems to also have an exponential nature. Though the number of feature types seems much smaller than the size of an image, the number of feature values is very large.

The mere presence of a feature type gives little discriminating power for a vision system unless there is an associated restriction on the set of objects or events in the task. The number of feature value subsets is an exponential function of the set size, and brute-force search in natural images for feature value subsets which may be the best candidates for matching will not suffice.

3. **The control of information flow through the network of neurons that constitutes the visual system**
   The computational complexity of vision suggests pyramidal processing. Although pyramids solve part of the complexity problem by reducing the size of the representations to be processed, they introduce others. They corrupt the signals flowing through them unless some additional mechanisms are included.
   Assume an architecture with a hierarchical arrangement of computing units. Values represented at each unit are coded by their response strength similar in spirit to other pyramid. Connectivity from layer to layer need not be fixed and each layer (indeed, each unit) may have different connectivity patterns including overlap. There may be more than one output representation.

4. **The need to shift selection in time**
   Once a region is selected, it then follows that the remainder of the visual field cannot be processed unless a sequence of different regions are selected which together cover the visual field. There is the possibility that many regions might be selected simultaneously and matched in parallel. There are many choices for how to select next regions to process. An algorithm might simply tile visual space, selecting regions in some arbitrary order that will eventually cover the entire visual field. Alternatively, an ordering might be imposed on image subregions such that after one is processed it is not processed ever again unless some new image event occurs in that region.

3.2 Saliency

Attention is a general concept covering all factors that influence selection mechanisms, whether they are scene driven bottom-up (BU) or expectation-driven top-down (TD). **Saliency** intuitively characterizes some parts of a scene— which could be objects or regions—that appear to an observer to stand out relative to their neighboring parts. The term "salient" is often considered in the context of bottom-up computations.

The term **saliency** was first used by *Tsotsos et al* and *Olshausen et al* in their work on visual attention, and by *Itti et al* in their work on rapid scene analysis.

3.3 Attentional Models

The model types are here are models of saliency instead of those approaches that detect and segment the most salient region or object in a scene. The models here can be used to create saliency map which gives a measure of saliency and can help in further segmenting out the most salient object from the image or scene.

1. **Cognitive Models**
   Itti et al.'s basic model uses three feature channels color, intensity, and orientation. This model has been the basis of later models and the standard benchmark for comparison. It has been shown to correlate with human eye movements in free-viewing tasks. An input image is subsampled into a Gaussian pyramid and each pyramid level is decomposed into channels for Red (R), Green (G), Blue (B), Yellow (Y), Intensity (I), and local orientations (O_θ).
   From these channels, center-surround *feature maps* for different features are constructed and normalized.

2. **Bayesian Models**
   Bayesian modeling is used for combining sensory evidence with prior constraints. In these models, prior knowledge (e.g., scene context or gist) and sensory information (e.g., target features) are probabilistically combined according to Bayes' rule (e.g., to detect an object of interest).
   Torralba and Oliva et al. proposed a Bayesian framework for visual search tasks. Bottom-up saliency is derived from their formulation as $\frac{1}{p(f|f_G)}, where \mathbf{f}_G$ represents a global feature that summarizes the probability density of presence of the target object in the scene, based on analysis of the scene gist.

3. **Decision Theoretic Models**
   The decision-theoretic interpretation states that perceptual systems evolve to produce decisions about the states of the surrounding environment that are optimal in a decision theoretic sense (e.g., minimum probability of error). The overarching point is that visual attention should be driven by optimality with respect to the end task.

   Both Decision theoretic and Bayesian approaches have a biologically plausible implementation.

4. **Information Theoretic Models**
   These models are based on the fact that localized saliency computation helps to maximize information sampled from one's environment. They deal with selecting the most informative parts of a scene and discarding the rest.
   Bruce and Tsotsos proposed the **AIM model** (Attention based on Information Maximization) which uses Shannon's self-information measure for calculating saliency of image regions. Saliency of a local image region is the information that region conveys relative to its surroundings.

5. **Graphical Models**
   A graphical model is a probabilistic framework in which a graph denotes the conditional independent structure between random variables. Attention models in this category treat eye movements as a time series. Since there are hidden variables influencing the generation of eye movements, approaches like Hidden Markov Models (HMM), Dynamic Bayesian Networks (DBN), and Conditional Random Fields (CRF) have been incorporated.

   Liu et al. proposed a set of novel features and adopted a Conditional Random Field to combine these features for salient object detection on their regional saliency dataset. Later, they extended this approach to detect salient object sequences in videos.

6. ***Spectral Analysis Models***
   Instead of processing an image in the spatial domain, models in this category derive saliency in the frequency domain.
   Achanta et al. implemented a frequency-tuned approach to salient region detection using low-level features of color and luminance. First, the input RGB image I is transformed to CIE Lab color space. Then, the scalar saliency map S for image I is computed as $S(x,\ y)=\|\ I_{\mu}\ -\ I_{hc}\ \|$, where I_{μ} is the arithmetic mean image feature vector, I_{hc} is a Gaussian blurred version of the original image using a 5×5 separable binomial kernel, $\|\ .\ \|$ is the L2 norm (euclidean distance), and x, y are the pixel coordinates.

3.4 Salient object detection

Salient object detection in computer vision is a problem that has two stages:

1. Detect the most salient object using any of the attention models
2. Segment out the most salient object

There are several models that adopt saliency concept to extract the two steps together. The first stage need not be restricted to one object alone. The prediction map used may detect more than one salient object in the image. The second stage is similar to segmentation problem but here we have to segment out the most salient object and not just any object.

For good saliency detection, a model has to meet three criteria:

1. **Good detection**
   Probability of missing the salient object and marking the background as salient object should be very low.
2. **High resolution**
   Saliency maps should have high or full resolution to accurately locate salient objects and retain original image information.

3. **Computational efficiency**
   The model needs to be computationally efficient and detect the salient regions quickly.

3.5 Salient object detection models

There are two major categories:

1. **Block-based vs. Region-based analysis**
   There are mainly two kinds of visual subsets, including *blocks* and *regions*, that are used to detect salient objects. Blocks are usually adopted by many early approaches, while regions are increasingly popular with the development of superpixel algorithms.

2. **Intrinsic cues vs. Extrinsic cues**
   Intrinsic cues approaches propose to extract various cues only from the input image itself to pop-out targets and suppress distracts. However distracts may share some common visual attributes and the intrinsic cues are often insufficient to distinguish them.
   Extrinsic cues use external information such as user annotations, depth map, or statistical information of similar images to facilitate detecting salient objects in the image.

Most of existing salient object detection approaches into three major subgroups according to two attributes given above:-

1. **Block-based models with intrinsic cues**
   The method aims to detect salient objects based on pixels or patches where only intrinsic cues are utilized.
   These approaches usually suffer from two shortcomings:

   (a) High-contrast edges usually stand out instead of the salient object.

   (b) The boundary of the salient object is not preserved well (especially when using large blocks).

   To overcome these issues, more and more methods propose to compute the saliency map based on regions. This offers two main advantages.

   (a) The number of regions is far less than the number of blocks, which can be used to develop highly efficient algorithms.

   (b) More sophisticated features can be extracted from regions, leading to better performance.

2. ***Region-based model with intrinsic cues***

   Saliency models in this subgroup adopt intrinsic cues extracted from image regions to estimate their saliency scores. Different from the block-based models, region-based models often segment an input image into regions aligned with intensity edges first and then compute a regional saliency map. The regional saliency score is defined as the average saliency scores of its contained pixels.

3. **Models with Extrinsic Cues**

   Models in this subgroup adopt the *extrinsic cues* to assist the detection of salient objects in images and videos. In addition to the visual cues observed from the single input image, the extrinsic cues can be derived from the ground truth annotations of the training images, similar images, the video sequence, a set of input images containing the common salient objects, depth maps, or light field images.

   This can be further classified as:

   (a) **Supervised salient object detection**

   Each element (e.g., a pixel or a region) in the input image will be represented by a feature vector $f \in \mathrm{R}^D$, where D is the feature dimension. Such a feature vector is then mapped to a saliency score $s \in \mathrm{R}^+$ based on the learned linear or non-linear mapping function $f : \mathrm{R}^D \rightarrow \mathrm{R}^+$.

   (b) **Salient object detection with similar images**

   With the availability of increasingly larger amount of visual content on the web, salient object detection by leveraging the visually similar images to the input image has been studied in recent years. Generally, given the input image I, K similar images $C_I = \{\mathrm{I}_k\}_{k=1}{}^K$ are first retrieved from a large collection of images C. The salient object detection on the input I can be assisted by examining these similar images.

   (c) **Co-salient object detection**

   Instead of concentrating on computing saliency on a single image, co-salient object detection algorithms focus on discovering the common salient objects shared by multiple input images. That is, such objects can be the same object with different view points or the objects of the same category sharing similar visual appearances. The key characteristic of co-salient object detection algorithms is that their input is a set of images, while classical salient object detection models only need a single input image.

Chapter 4

Our Work

In our method of saliency detection, a **region bases model with intrinsic cues** has been used. This is a **computational approach** that takes less time to detect a salient region in an image. The method is a **bottom up approach**. The image is initially preprocessed for albedo correction and smoothing operation is done using bilateral filter. Thereafter **spectral analysis model** by *Achanta et al.* has been used to compute the feature map. Superpixels of desired size are formed on the feature map and the most salient superpixels are selected out using a proper thresholding to segment out the most salient object in the image. All the key terminologies and concepts are discussed here.

4.1 Albedo correction

Usually most of the image segmentation techniques (thresholding, edge detection, region growing etc) assume that the image is piecewise constant i.e. composed of regions each having approximately constant intensity. If an image consists of diffusely reflecting planar surfaces, then the assumption is valid. One the other hand, if a surface has substantial curvature, there will be significant changes in intensity across its image, so that it will give rise to smeared-out collection of intensities rather than a histogram peak. Histogram- based segmentation of such an image will yield regions that do not correspond to surfaces on the scene. So an albedo correction of the image is necessary before proceeding.
A method for albedo correction has been given by *Lee et al.* in their paper *Albedo estimation for scene segmentation.*

Algorithm 1 Albedo correction for RGB images

procedure ALBEDO_CORRECTION($Image$) ▷ Albedo corrected image
  $Gdir$= gradient map of input image $Image$
  count←0
  rows← rows in Image
  cols← columns in Image
  while *count* $\neq 4$ **do** ▷ Do the below for 4 iterations for each color channel
    for $i < rows$ **do**
      for $j < cols$ **do**
        Let point (i,j) be P. Find neighbouring points R and Q in the direction of P i.e. $Gdir(P)$=$Gdir(Q)$=$Gdir(R)$.
        $\Delta = \sqrt{\frac{2I_P - I_Q - I_R}{I_P}}$ ▷ I is the intensity of image Image
        $\omega = \cos(tan^{-1}(\frac{I_R - I_P(1 - \frac{\Delta^2}{2})}{\Delta}))$
        $I_P = \frac{I_P}{\omega}$
        $i \leftarrow i + 1$
      $j \leftarrow j + 1$
    Smooth image Image with a 5×5 median filter.
    $count \leftarrow count + 1$
  return $Image$ ▷ Image is the albedo corrected image

4.2 Bilateral filters

In *filtering*, the value of the filtered image at a given location is a function of the values of the input image in a small neighborhood of the same location. For example, Gaussian low-pass filtering computes a weighted average of pixel values in the neighborhood, in which the weights decrease with distance from the neighborhood center.
The bilateral filter is technique to smooth images while *preserving edges*.
The **bilateral filter** takes a weighted sum of the pixels in a local neighborhood. The weights depend on both the *spatial distance* and the *intensity distance*. In this way, edges are preserved well while noise is averaged out. Mathematically, at a pixel location x, the output of a bilateral filter is calculated as follows,

$$I(x) = \frac{1}{C}\sum_{y \in N(x)} e^{\frac{-\|y-x\|^2}{2{\sigma_d}^2}} e^{\frac{-\|I(y)-I(x)\|^2}{2{\sigma_r}^2}} I(y)$$

σ_d is the parameter controlling fall off of weight in spatial domain
σ_r is the parameter controlling fall off of weight in intensity domain
$N(x)$ is a spatial neighbourhood of pixel x
$I(x)$ is intensity of pixel x
$I(y)$ is intensity of pixel y

C is the normalization constant

So, a bilateral filter takes two value of sigma where the spatial-domain standard deviation is given by σ_d and the intensity-domain standard deviation is given by σ_r.

4.3 Frequency tuned saliency detection

4.3.1 Requirements

A method of saliency detection has been suggested by *Achanta et al.* in his paper *Frequency tuned saliency detection.*
A saliency detector should have the following properties:

1. It should emphasize the most salient object present in the image.
2. It should uniformly highlight the whole salient region.
3. It should establish well defined boundaries of the image.
4. It should disregard high frequencies arising out of texture and noise.

4.3.2 Using DoG band pass filters

Let ω_{lc} be lower cut off frequency and ω_{hc} be higher cut off frequency. To get most salient object and highlight the whole salient region(*condition 1 and 2*) we need ω_{lc} to be low. To get well defined boundaries(*condition 3*), we need to retain high frequencies from the original image. To discard noise (*condition 4*) we need to ignore the highest frequencies. To satisfy all the above properties we need the output of a range of frequencies. So we need to combine the output of several contiguous band pass filters [ω_{lc},ω_{hc}].

The *DoG filter* can be used in band pass filtering. The DoG filter is widely used in edge detection. Besides it is also used to detect interest points.The DoG filter is given by:

$$DoG(x,y) = \frac{1}{2\pi}\left[\frac{1}{{\sigma_1}^2}e^{\frac{-(x^2+y^2)}{2{\sigma_1}^2}} - \frac{1}{{\sigma_2}^2}e^{\frac{-(x^2+y^2)}{2{\sigma_2}^2}}\right]$$

$$=G\Big(x,y,\sigma_1\Big) - G\Big(x,y,\sigma_2\Big)$$

σ_1 and σ_2 are standard deviations of Gaussian where $\sigma_1 > \sigma_2$.
The pass band width of a DoG filter is controlled by the ratio $\frac{\sigma_1}{\sigma_2}$. Let $\sigma_1 = \rho\sigma$ and $\sigma_2 = \rho\sigma$.
A summation over DoG where standard deviations are in ratio ρ*resultsin*

$$\sum_{n=0}^{N-1}\Big(G(x,y,\rho^{n+1}\sigma) - G(x,y,\rho^n\sigma)\Big) = G(x,y,\rho^n\sigma) - G(x,y,\sigma)$$

for $N > 0$.
As $\sigma_1 > \sigma_2$, ω_{lc} is determined by σ_1 and ω_{hc} s determined by σ_2. To implement a large ratio in standard deviations, we drive σ_1 to infinity. This results in a notch in frequency at DC while retaining all other frequencies. To remove high frequency noise and textures, we use

a small Gaussian kernel keeping in mind the need for computational simplicity. So ω_{hc} can be set to $\frac{\pi}{2.75}$.

4.3.3 Saliency estimation

The finding the saliency map **S** for an image **I** of width W and height H pixels can thus be formulated as
$S(x, y) = \|I_\mu - I_{\omega_{hc}}\|$.
I_μ is the mean saliency value of the image.
$I_{\omega_{hc}}$ Gaussian blurred version of the original image to eliminate fine texture details as well as noise and coding artifacts.
The image is converted to Lab space from RGB space before carrying out computations and the overall saliency would be the *L2* norm of the saliency value of the three channels. This is computationally efficient and is faster.

4.4 Superpixels

A superpixel map has many desired properties:

1. It is **computationally efficient**. It reduces the complexity of images from hundreds of thousands of pixels to only a few hundred superpixels.

2. The superpixels are **percetually meaningful**. Each superpixel is a perceptually consistent unit, i.e. all pixels in a superpixel are most likely uniform in, say, color and texture.

3. It is **near-complete**. This is because superpixels are results of an oversegmentation, most structures in the image are conserved. There is very little loss in moving from the pixel-grid to the superpixel map.

Some of the low-level image segmentation methods are as follows.

1. **SLIC (Simple Linear Iterative Clustering)**
   SLIC clusters pixels in the combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform superpixels.

2. **Mean shift**
   Mean shift is a mode-seeking algorithm that generates image segments by recursively moving to the kernel smoothed centroid for every data point in the pixel feature space, effectively performing a gradient ascent. The generated segments/superpixels can be large or small based on the input kernel parameters, but there is no direct control over the number, size, or compactness of the resulting superpixels.

3. ***Quick shift***
   Quick-shift is also a mode-seeking segmentation scheme like mean shift, but is faster in practice. It moves each point in the feature space to the nearest neighbor that increases the *Parzen density estimate*. The algorithm is non iterative, and like mean shift, does not allow to explicitly control the size or number of superpixels. Superpixels from quick-shift have been used in applications like object localization and motion segmentation.

4. **Graph-based image segmentation**
   Felzenszwalb and Huttenlocher proposed a technique that segments an image by agglomerate clustering of pixels on such an image based graph. The method is very similar to the Kruskal's shortest spanning tree algorithm where the tree is formed by choosing edges in increasing order of their weight. The main difference is that goal is not to create a single spanning tree cluster but several subtree clusters by introducing stopping criteria during tree formation.

Our method has used quick shift method for forming superpixels because the method is faster and does not makes any assumption on the number of pixels. So a required segmentation can generate some pixels which can be used to aggregated to form bigger superpixels by merging with similar neighbours nearby based on the average R,G and B values of the individual superpixels. This can be used to segment out the salient object.
The quick shift implementation of *VLFeat open source library* has been used in this work. The method takes the following arguments:

- **Sigma**
  Sigma is the standard deviation of the Parzen window density estimator(width of the Gaussian Kernel used to smooth sample density).
  A higher value of σ would create bigger clusters.

- **Distance**
  Distance denotes the maximum distance in feature space between the nodes in a quick-shift tree. This is used to cut links in a tree to form segments.
  A higher value of *Distance* would create larger clusters.

- **Ratio**
  Ratio is the trade-off between distance in color-space and distance in image space.
  A small value of *Ratio* would give more importance to space.
  A greater value of *Ratio* would give more importance to colors.

The following steps are involved in quick-shift:

1. For each pixel (x, y) , quick-shift regards $(x, y, I(x, y))$ as a $(d+2)$ dimensional feature space. $I(x, y)$ is the intensity matrix that is of d-dimensions.

2. Parzen density estimate for each point is then calculated with a Gaussian Kernel of standard deviation σ.

   $$E(x,y) = P(x,y,I(x,y)) = \sum_{x'y'} \frac{1}{2\pi\sigma^{d+2}} e^{\frac{-1}{2\sigma^2}\|x-x'\|+\|y-y'\|+\|I(x,y)-I(x',y')\|}$$

3. Quick shift then constructs a tree connecting each image pixel to its nearest neighbour that has a greater density value.

4. Then depending on the segment value, edges of the trees are cut off and superpixels are created.

Figure 4.1: Input images

Figure 4.2: After superpixel division. Superpixel has been created in LAB space using quick shift. Green lines represent the boundary of individual superpixels.

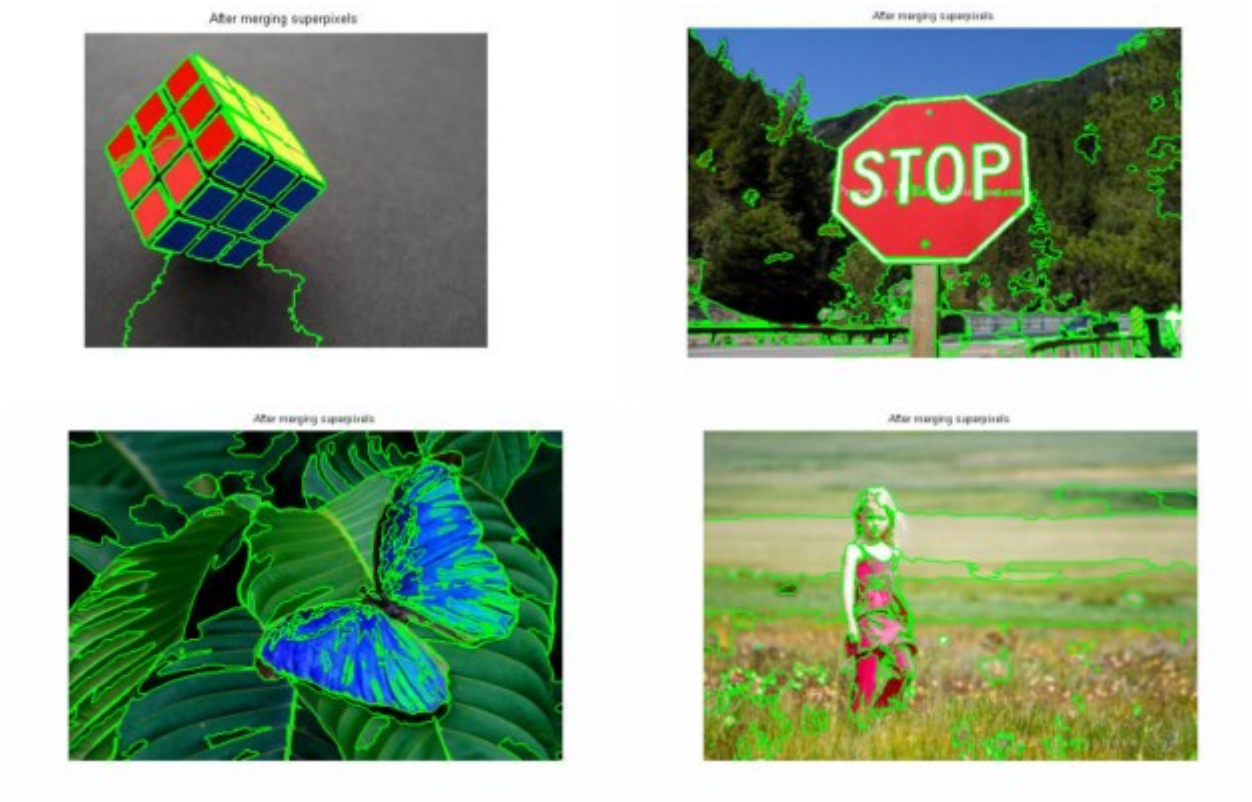


Figure 4.3: After merging superpixels using mean of R,G and B value of each individual superpixel.

4.5 Developed algorithm

Algorithm 2 Mean of an image with single channel

procedure MEAN($Image$) ▷ Mean of an image with single channel
    rows← rows in Image
    cols← columns in Image
    $imageSum \leftarrow 0$
    $imageAvg \leftarrow 0$
    $i \leftarrow 0$
    $j \leftarrow 0$
    while $i \neq rows$ **do** ▷ Iterate for total number of rows
        while $j \neq cols$ **do** ▷ Iterate for total number of cols
            $imageSum \leftarrow imageSum + Image(i, j)$
            $j \leftarrow j + 1$
        $i \leftarrow i + 1$
    $imageAvg \leftarrow \dfrac{imageSum}{rows \times cols}$
    return $imageAvg$ ▷ $imageAvg$ is the average value of $Image$

Algorithm 3 Merge the smaller superpixels to create bigger superpixels

procedure $MERGE\_SUPPIXELS(segments, Image)$ ▷ Merge smaller superpixels
    rows← rows in Image
    cols← columns in Image
    $noOfSupPix = MAXIMUM(segments)$ ▷ Get total number of superpixels
    $avgR \leftarrow zeros(noOfSupPix)$ ▷ Get average R value of superpixels
    $avgG \leftarrow zeros(noOfSupPix)$ ▷ Get average G value of superpixels
    $avgB \leftarrow zeros(noOfSupPix)$ ▷ Get average B value of superpixels
    $adjGraph \leftarrow \phi$ ▷ Adjacency matrix to represent superpixels as an undirected graph
    $count \leftarrow 0$
    while $count \neq$noOfSupPix **do** ▷ Iterate for total number of superpixels
        $adjGraph[count]$ should contain pixels adjacent to it
        $count \leftarrow count + 1$
    $count \leftarrow 0$
    while $count \neq$noOfSupPix **do** ▷ Iterate for total number of superpixels
        Calculate $avgR[count], avgG[count], avgB[count]$ for all superpixels
        $count \leftarrow count + 1$
    $count \leftarrow 0$
    while $count \neq$noOfSupPix **do** ▷ Iterate for total number of superpixels
        for all superpixels $neighbour$ in $adjGraph[count]$ **do**
            if $\|avgR[neighbour] - avgR[count]\| \leq \epsilon \wedge \|avgG[neighbour] - avgG[count]\| \leq \epsilon \wedge \|avgB[neighbour]$-$avgB[count]\| \leq \epsilon$ **then** Merge superpixel $neighbour$ and $count$
        $count \leftarrow count + 1$
    return $segments$ ▷ $segments$ is the new superpixel map after merging

Algorithm 4 Detect salient region in RGB images

procedure SALIENT_SEGMENT($Image$) ▷ Get salient region in image
  $Image$= ALBEDO_CORRECTION($Image$) ▷ Albedo correction of $Image$
  $Image$= BILATERAL_SMOOTH($Image$,σ_1, σ_2) ▷ Bilateral smoothing of $Image$
  $LabImage$=RGB_TO_LAB($Image$) ▷ Convert RGB Image to its Lab form
  $mean_L \leftarrow MEAN(LabImage_L)$ ▷ mean of L component
  $mean_a \leftarrow MEAN(LabImage_a)$ ▷ mean of a component
  $mean_b \leftarrow MEAN(LabImage_b)$ ▷ mean of b component
  $sal_L \leftarrow mean_L - LabImage_L$ ▷ band pass filtering of L component
  $sal_a \leftarrow mean_a - LabImage_a$ ▷ band pass filtering of a component
  $sal_b \leftarrow mean_b - LabImage_b$ ▷ band pass filtering of b component
  $SalMap = \sqrt{{sal_L}^2 + {sal_a}^2 + {sal_b}^2}$ ▷ Saliency map
  $segments = QUICK\_SHIFT(SalMap)$▷ Create superpixels based on Saliency map
  $segments = MERGE\_SUPPIXELS(segments, Image)$ ▷ Merge the superpixels based on average RGB value to reduce number of superpixels
  $noOfSupPix = MAXIMUM(segments)$ ▷ Get total number of superpixels
  $avgSupVal = AVERAGE\_SUP\_PIXEL(segments, SalMap)$ ▷ Find average of each superpixel from Saliency map
  $count \leftarrow 0$
  while $count \neq noOfSupPix$ **do** ▷ Do the below to cover all superpixels
    if If superpixel $count$ touches boundary **then** ▷ Cover all superpixels
      $avgSupVal[count] = 0$
    $count \leftarrow count + 1$
  $avgImgSal = MEAN(SalMap)$ ▷ Mean of image saliency value
  $supPixSelect \leftarrow zeros(noOfSupPix)$ ▷ Initialize superpixel select array
  $\nu \leftarrow 1.6$
  $count \leftarrow 0$
  while $count \neq noOfSupPix$ **do** ▷ Do the below to cover all superpixels
    if $avgSupVal[count] > \nu \times avgImgSal$ **then**
      $supPixSelect[count] = 1$
    $count \leftarrow count + 1$
  $supPixSelect = POST\_PROCESS(supPixSelect)$ ▷ To remove small patches
  $FinalImage = SELECT\_SUP\_PIXELS(Image, supPixSelect, segments)$ ▷ Segment the salient region
  return $FinalImage$ ▷ $FinalImage$ is the salient segment of image

Chapter 5

Results of Saliency Detection

In this section we analyze the saliency result detected by the algorithm on several images. The algorithm is run on some images of several categories given in the ImgSal data set and the Microsoft Saliency dataset and the results are then compared. The percentage of salient object covered in each image is varying. The salient objects do not touch boundary in any of the images. The images in the ImgSal dataset are of the size 480×640 pixels and the images in the Microsoft saliency dataset are of the size size 300×400 pixels. To display the results, they have been scaled to 200×300 pixels.
Initially the images are bilaterally smoothed taking value of σ_1 and σ_2 as 2 and 3 respectively. When the implementation of quick shift given in the vlfeat library is applied on the saliency map, the *ratio* is kept at 0.9 to give more importance to contrast. The maximum distance is kept at 20. While merging the smaller superpixels into larger superpixels, ϵ is kept at 15 for each of channel of RGB. The value of ν in the thresholding step can be kept somewhere between 1.5 to 2 depending on the resolution and size of the image dataset being worked on. In the post processing phase, the region with the largest area in the salient region after thresholding can be selected as the most salient region to remove smaller regions.

Some of the results have been put up here.

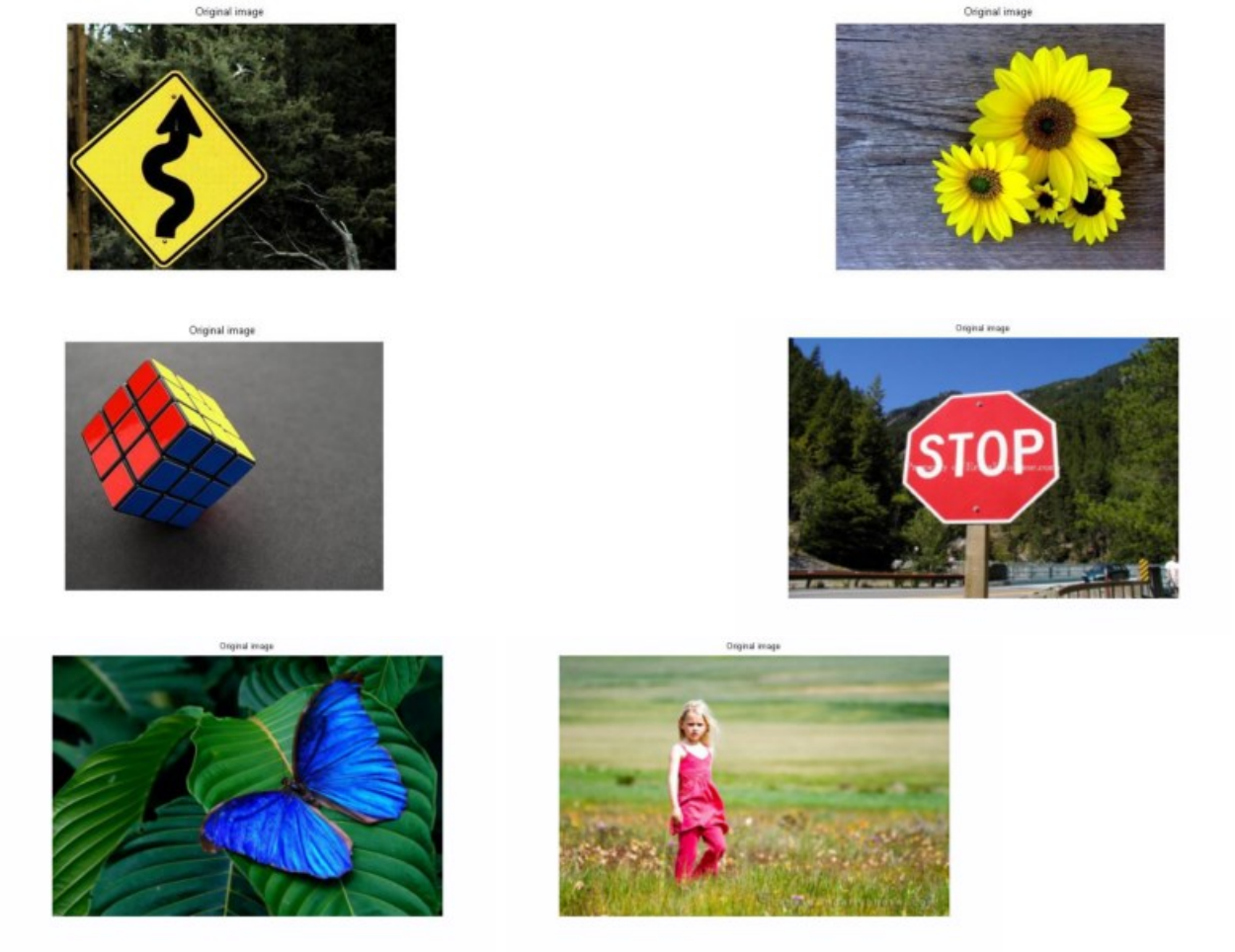


Figure 5.1: Input images

Figure 5.2: After Albedo correction.

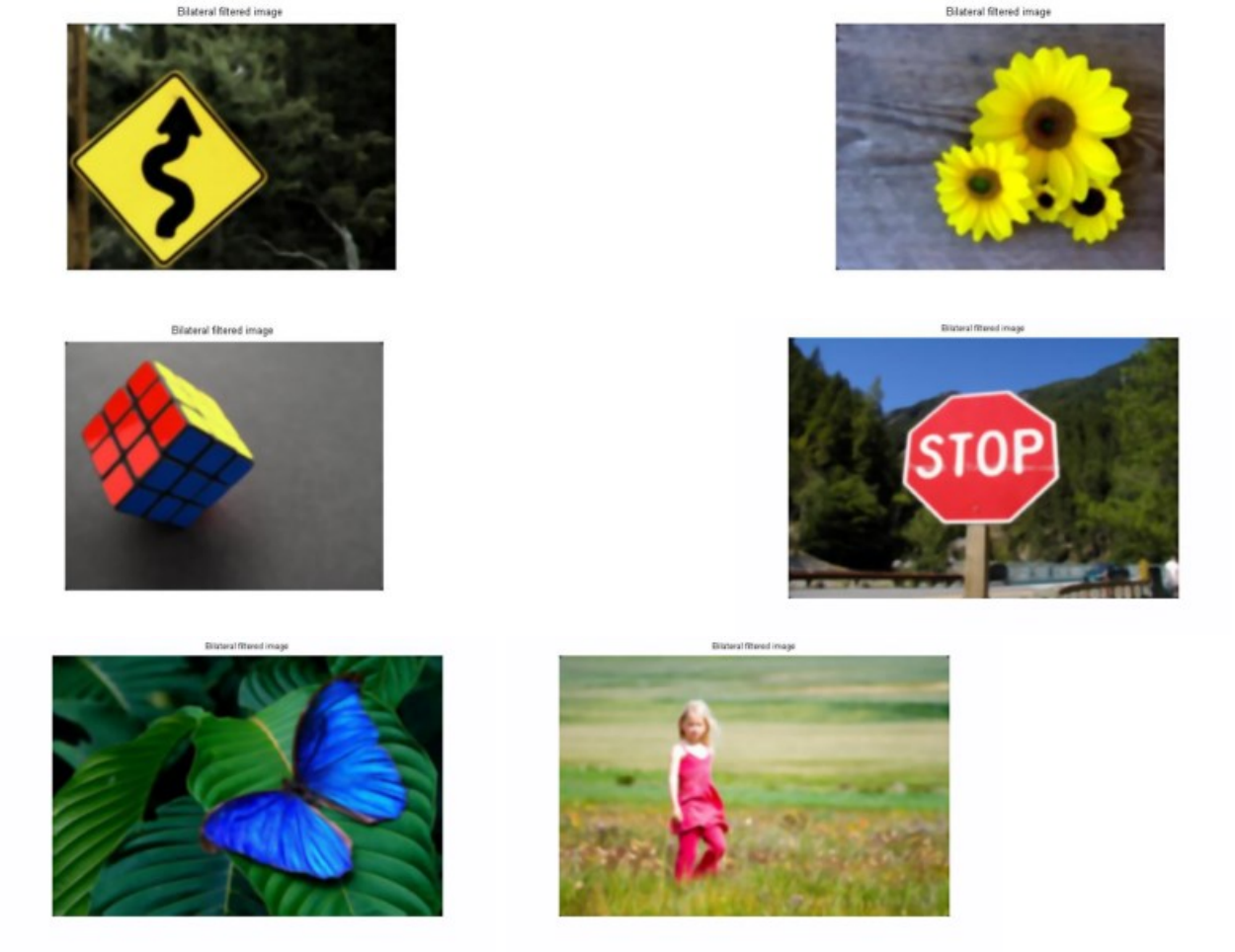


Figure 5.3: After bilateral filtering.

Figure 5.4: Difference between mean of L component and L component after the above images after bilateral filtering are converted to LAB space.

Figure 5.5: Difference between mean of A component and A component.

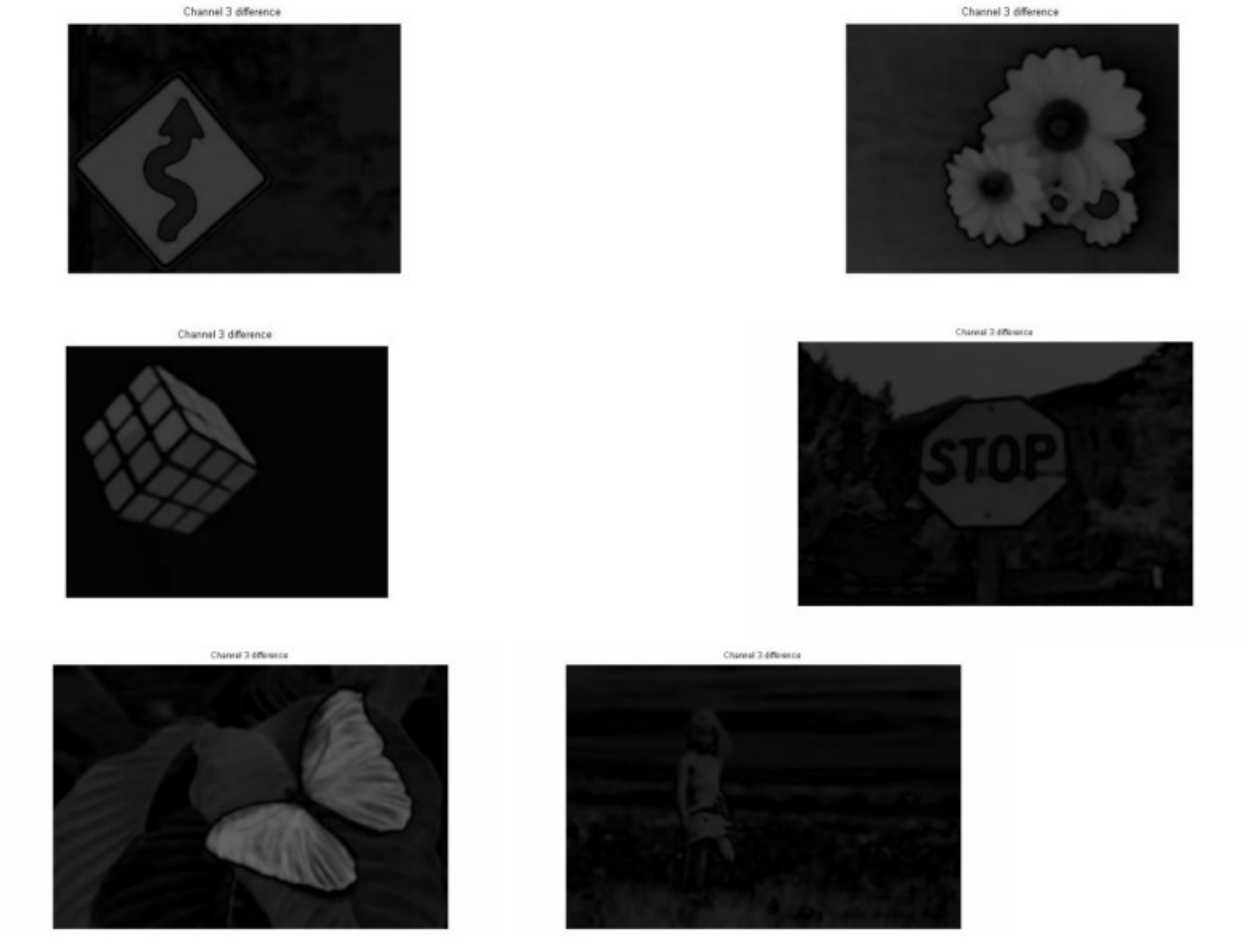


Figure 5.6: Difference between mean of B component and B component.

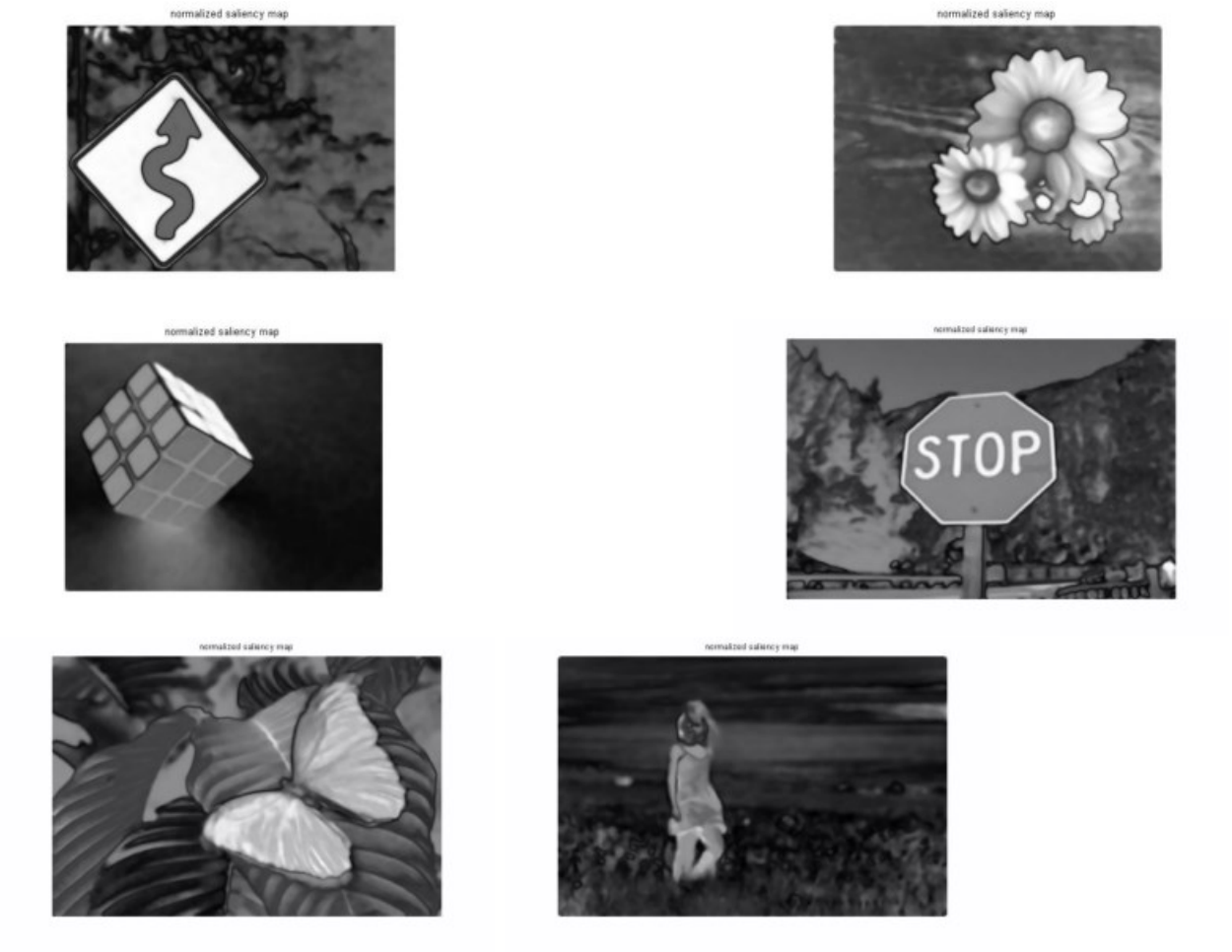


Figure 5.7: Saliency map created after L2 norm of the three channels.

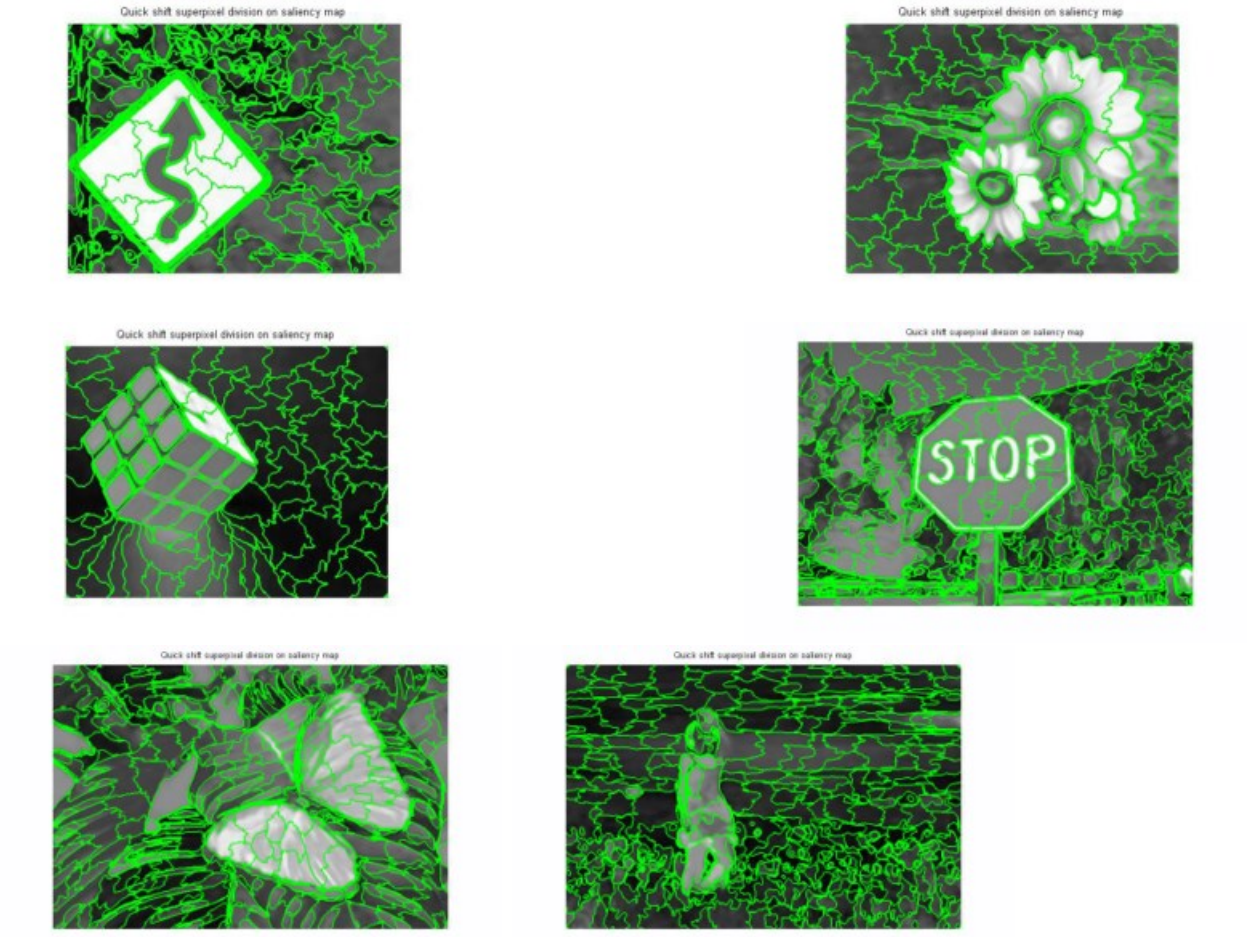


Figure 5.8: Applying quick shift on saliency map.

Figure 5.9: Merging the smaller superpixels based on the differences between the average of R, G and B values between them.

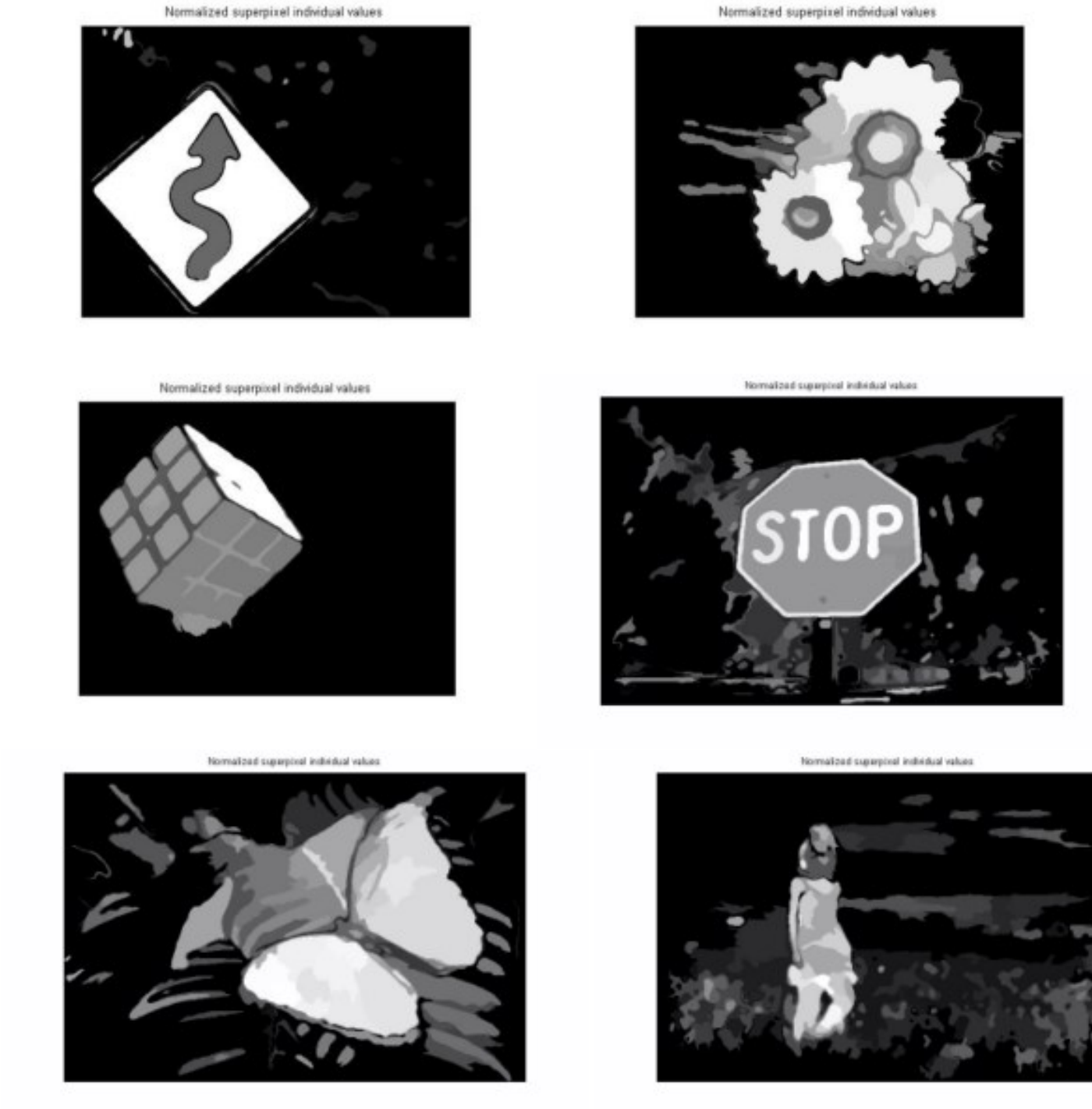


Figure 5.10: Normalized values of individual superpixels. Superpixels touching boundary has been assigned zero.

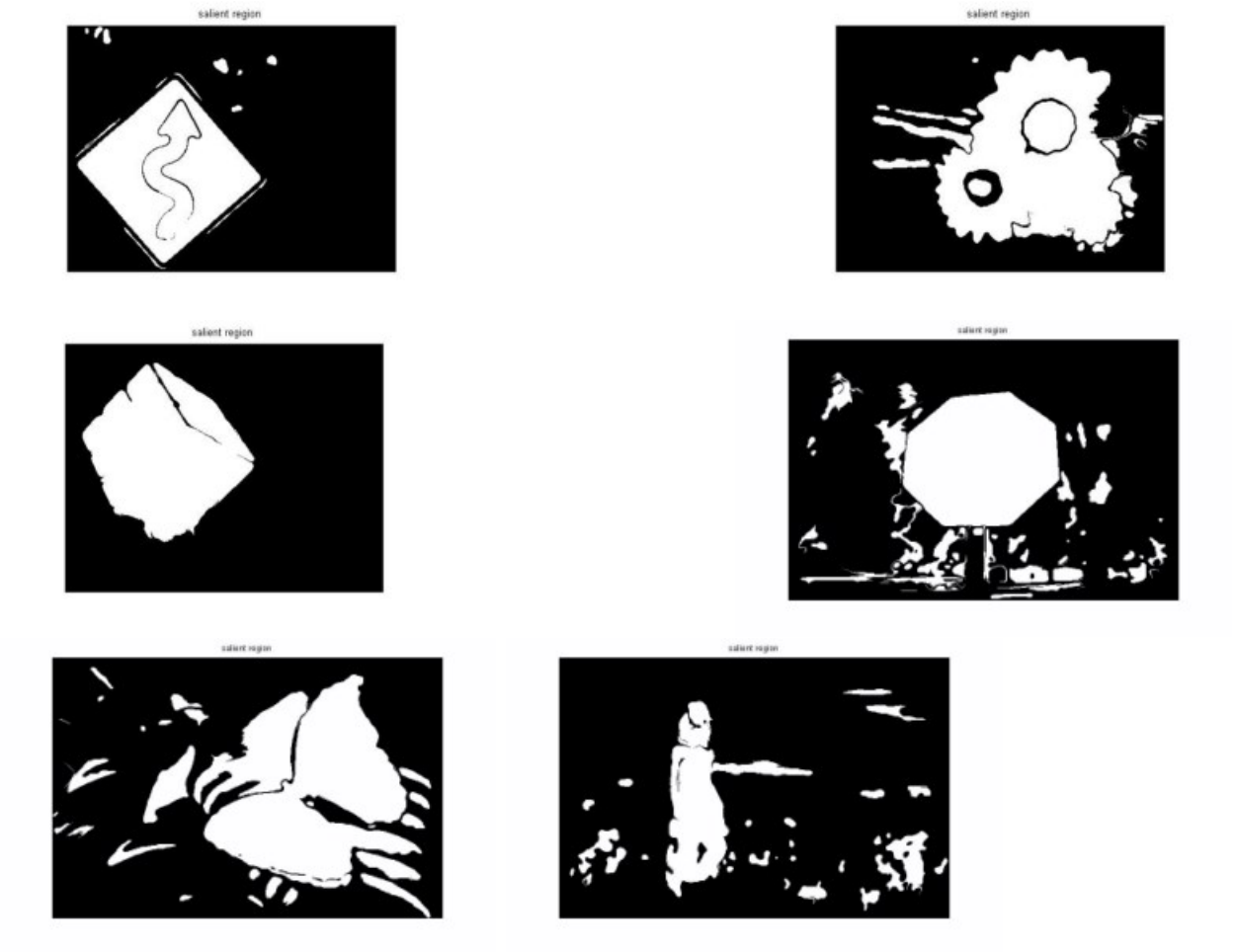


Figure 5.11: Saliency segment after thresholding.

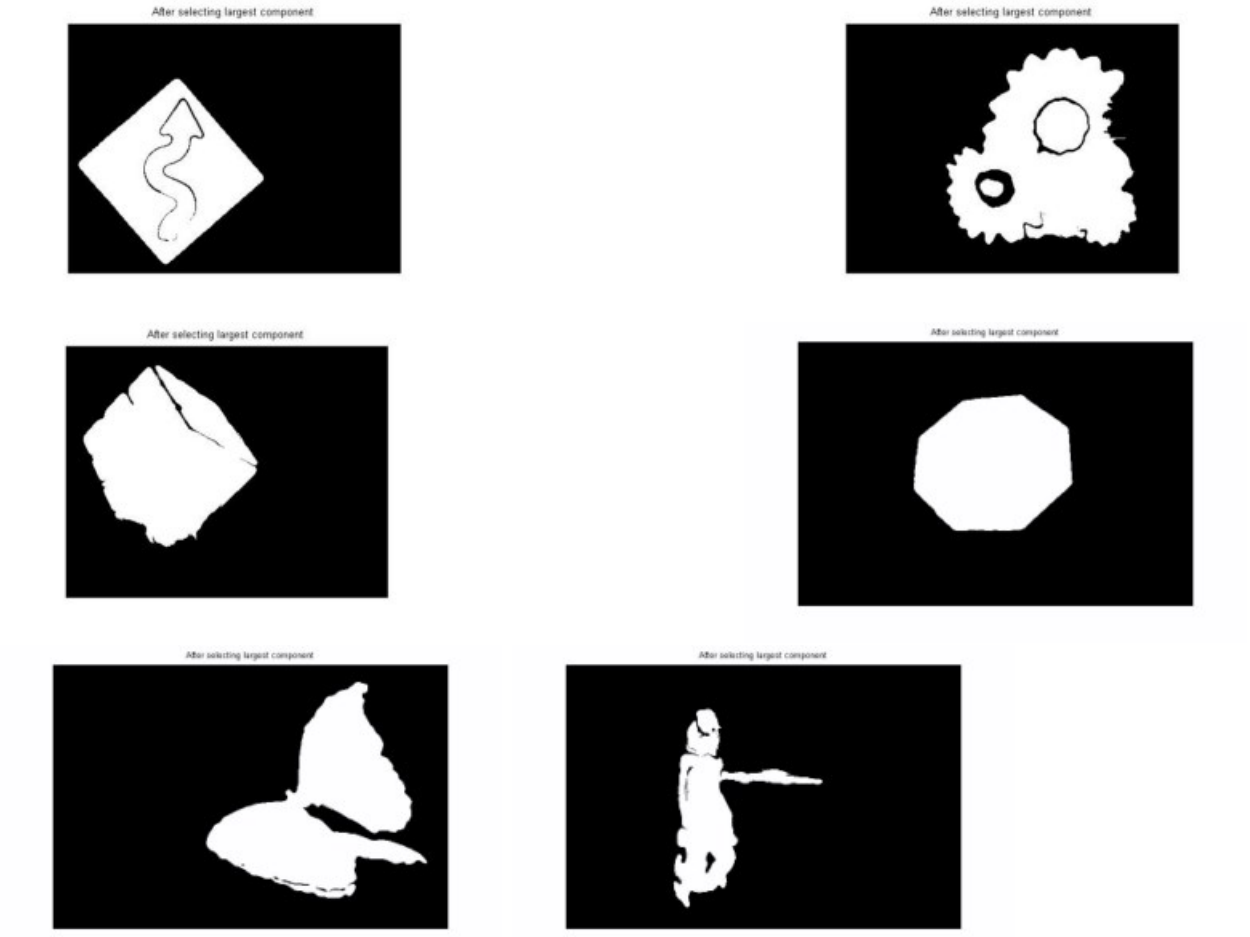


Figure 5.12: Saliency segment after postprocessing the thresholded image.

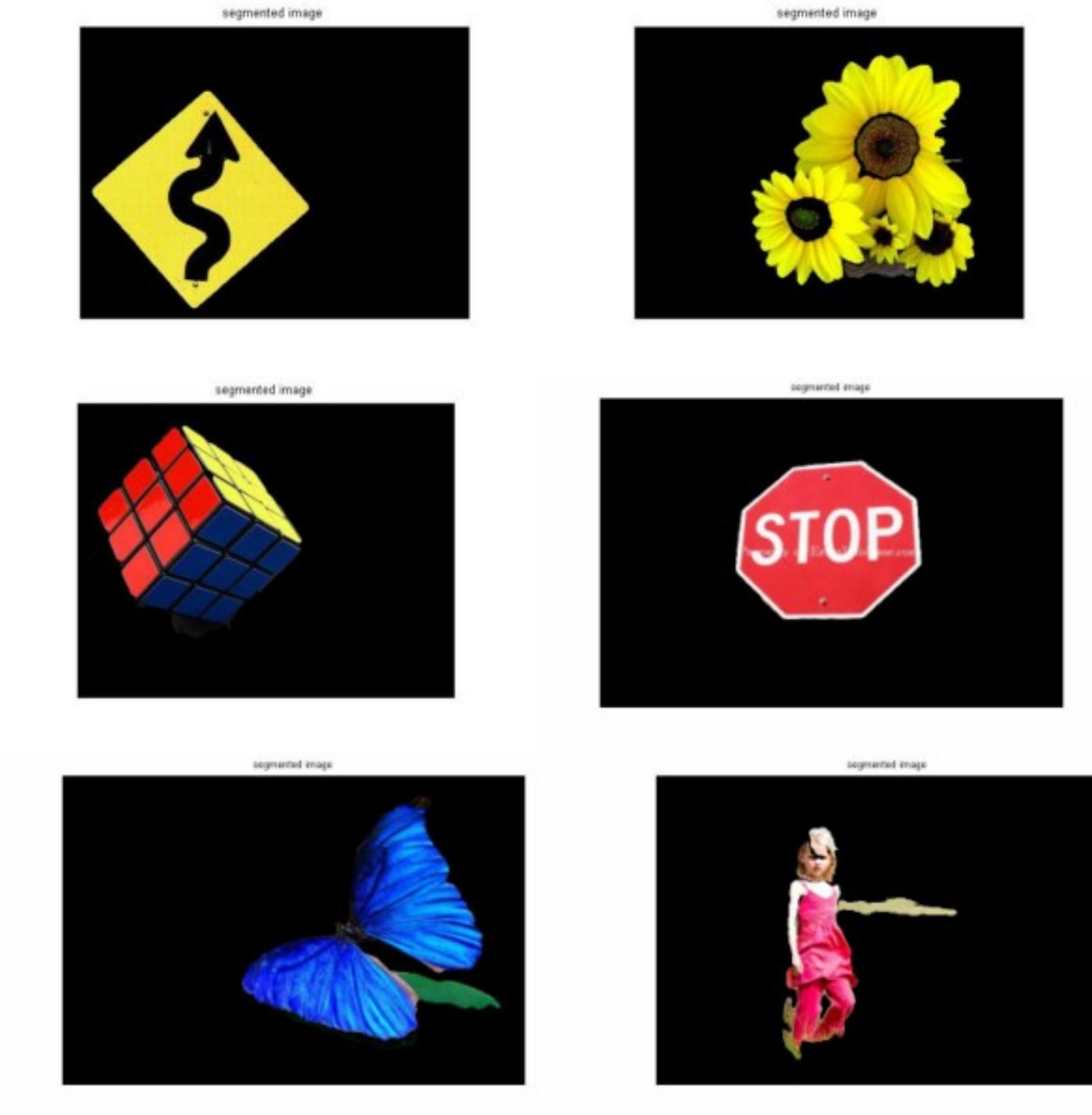


Figure 5.13: Segmenting the most salient content of the image.

Chapter 6

Conclusion

In this thesis an approach to extract the most visually attentive part of an image has been presented. The above method can be applied on a variety of images with several types of object. By able to extract the most salient region, the results can be used further in other tasks which can help in reducing the storage and computation time of several other applications. A fixed threshold has not been used here. The threshold is varying and depends on image size and content. So we need not worry about the threshold for each image. The method is simple and takes less time to run.

The above method can be further extended for image retrieval applications. One such can be automatic annotations on images. Here we can extract salient regions from a set of images called as the training images. We can have certain feature descriptors for the training images. Then for a given unknown image, we can extract feature descriptors and find close matching images and can annotate the unknown image. The method helps in reducing search space in the images. So the method can be extended further to several application areas.

Bibliography

[1] A. Borji, M. Cheng, H. Jiang and J. Li *Salient Object Detection: A Survey*. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2010.

[2] R. Achanta, S. Hemami, F. Estrada and S. Susstrunk. *Frequency-tuned Salient Region Detection*. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 1597 - 1604, 2009.

[3] C. Lee and A. Rosenfled. *Albedo estimation for scene segmentation*. Pattern Recognition Letters volume 1, page 155-160, 1983.

[4] J. Jhang and S. Scarloff *Saliency Detection: A Boolean Map Approach*. IEEE International Conference on Computer Vision (ICCV), 2013.

[5] J. K. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, F. Nuflo *Modeling visual attention via selective tuning*. Artificial Intelligence 78 (1995) 507-545.

[6] Open source library VLFeat for Quick shift ,
`http://www.vlfeat.org/overview/quickshift.html`

Introduction to Gestalt principles ,
`http://graphicdesign.spokanefalls.edu/tutorials/process/gestaltprinciples/gestaltprinc`