Image Processing Term Project Report

Group-3

Sourav Pal 14CS10062
Vedant Pate 14CS30021
Shashwat Gupta 14IE10028

Problem Statement:

Segmentation of Objects in an image: Segment the salient boundaries in any 10 images given in the dataset provided by UC Berkeley namely **The Berkeley Segmentation Dataset and Benchmark**. This was to be compared with the human annotations.

Motivation:

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristics.

Some of the practical applications of image segmentation are:

- Industrial inspection
- Content-based image retrieval
  - Image search applications
  - Object Recognition
  - Optical character recognition (OCR)
- Medical imaging, including volume rendered images from computed tomography and magnetic resonance imaging
  - Locate tumors and other pathologies
  - Measure tissue volumes

    - Diagnosis, study of anatomical structure
    - Surgery planning
    - Virtual surgery simulation
    - Intra-surgery navigation
- Object detection
    - Pedestrian detection
    - Face detection
    - Brake light detection
    - Locate objects in satellite images (roads, forests, crops, etc.)
- Recognition Tasks
    - Face recognition
    - Fingerprint recognition
    - Iris recognition
    - Detection and measurement of bone, tissue, etc., in medical image
    - Classification of terrains visible in satellite images.
- Traffic control systems
    - Traffic level computation
- Video surveillance
    - Object Detection
    - Object Tracking

Several general-purpose algorithms and techniques have been developed for image segmentation. To be useful, these techniques must typically be combined with a domains specific knowledge in order to effectively solve the domain's segmentation problem

The image segmentation approaches can be categorized based on properties of image:

1. **Discontinuity detection based approach:** Based on discontinuity region for segmentation of an image.In this technique, partition of an image is based on change in the intensity. It includes an edge detection algorithm for image segmentation.
2. **Similarity detection based approach:** Based on finding the similar region of an image. The following are the techniques that come under this are: region growing, thresholding techniques and region merging and splitting. All these techniques divide an image with the help of similar pixels. This approach is also used for clustering the data. In this approach, clusters of pixels are formed that have similar features.

Some Image Segmentation techniques have been discussed below:

1. **Thresholding Method:** Image pixels are divided with the help of intensity level of an image. This method is mainly used to distinguish the foreground objects from

background images. In this objects are lighter than its background. Selection of this method depends upon our prior knowledge.

2. **Edge Based Segmentation method:** A connected pixel that is found on the boundary of the region is called an edge. So these pixels on an edge are known as edge points. Edge can be calculated by finding the derivative of an image function. Some edges are very easy to find like Ramp edge, Step edge, Roof edge, Spike edge.
3. **Region Based Segmentation Method:** Segment an image on the basis of similar characteristics of the pixels. Region Based segmentation method is further divided into two categories:
    a. Region growing methods: Segments the image into various regions based on the growing of seeds (initial pixels). These seeds can be selected manually (based on prior knowledge) or automatically.
    b. Split and Merge: Initially the whole image which is taken as a single region is repeatedly split until no more splits are possible, then two regions are merged if they are adjacent and similar, merging is repeated until no more merging is possible.
4. **Clustering Based Segmentation Method:** Segment the image into clusters having pixels with similar characteristics. Data clustering is the method that divides the data elements into clusters such that elements in same cluster are more similar to each other than others.
5. **Watershed Based Method:** Uses the concept of topological interpretation. In this the intensity represents the basins having hole in its minima from where the water spills. When water reaches the border of basin the adjacent basins are merged together. To maintain separation between basins dams are required and are the borders of region of segmentation. These dams are constructed using dilation. The watershed methods consider the gradient of image as topographic surface. The pixels having more gradient are represented as boundaries which are continuous.
6. **Artificial Neural Network Based Segmentation Method:** Simulate the learning strategies of human brain for the purpose of decision making. Now days this method is mostly used for the segmentation of medical images. It is used to separate the required image from background. A neural network is made of large number of connected nodes and each connection has a particular weight. The problem is converted to issues which are solved using neural network. This method has basic two steps: extracting features from the given data of images and segmentation by neural network.

Objective:

The main objective as mentioned in the problem statement was to segment natural images of the dataset provided by Berkley. In order to do so we need to find out the salient boundaries, meaning that the boundaries that are the most important in the image should be identified among all the line segments present. This will help us in detecting objects as humans perceive them. On comparing the output of the algorithm used with the human annotated images which are present in the dataset, one can gauge the effectiveness of the segmentation process used in place.

Design:

In order to solve the problem at hand we restored to designing our solutions in two different approaches.

The first one was an approach based on image statistics. The input image is an RGB image with 3 channels in it. As a first step we apply a Gaussian Blur on the input image to reduce the noise and smoothen the image. Then the blurred image is converted to a grayscale image in which the three channeled image is converted image into a single channel image. Thereafter we use image statistics to compute the median grayscale value from the blurred and the converted image. The median is computed using the numpy library of python. We use a parameter, sigma for our computation. After several rounds of experimentation we found out that the value of sigma=0.3 gives the best segmentation results when compared with human annotated images. The lower bound is computed as the maximum of 0 and (1.0 - sigma)*image_median. The upper bound is computed as the minimum of 255.0 and (1.0 + sigma)*image_median. Then Canny Edge detector is applied on the input image with the lower and upper bounds as computed above. We use the OpenCv library to perform the Canny Edge detection algorithm. As a standard procedure the canny edge detector is a multistage edge detection algorithm. The steps are: Preprocessing, Calculating gradients, Non Maximum suppression, Thresholding with hysteresis. The two key parameters of the algorithm are - an upper threshold and a lower threshold. The upper threshold is used to mark edges that are definitely edges. The lower threshold is to find faint pixels that are actually a part of an edge. As a comparison measure we also perform Canny Edge detection with fixed thresholds of two types, one is in which the range of values is wide and the other with a tighter range.

In the second approach we take advantage of the structure present in local image patches to learn both an accurate and computationally efficient edge detector. The problem is formulated as that of predicting local edge masks in a structured learning framework applied to random decision forests.

Implementation:

The first approach is implemented in **Python and using library functions from OpenCv and Numpy libraries**.

Some crucial snippets.

```python
#a function auto canny wich is a helper function
#it computes the median of the single channel pixels
#and then computes the upper and lower bounds depending
#image statistics computed using the median function of
#the numpy library.
def find_stats(image, sigma=0.3):
    # compute the median of the single channel pixel intensities
    v = np.median(image)

    # apply automatic Canny edge detection using the computed median
    lower_bound = int(max(0, (1.0 - sigma) * v)) #compute lower threshold
    upper_bound = int(min(255, (1.0 + sigma) * v)) #compute upper threshold
    edged = cv2.Canny(image, lower_bound, upper_bound)

    # return the edged image
    return edged
```

The second approach is implemented using **C++ and using OpenCv library.**

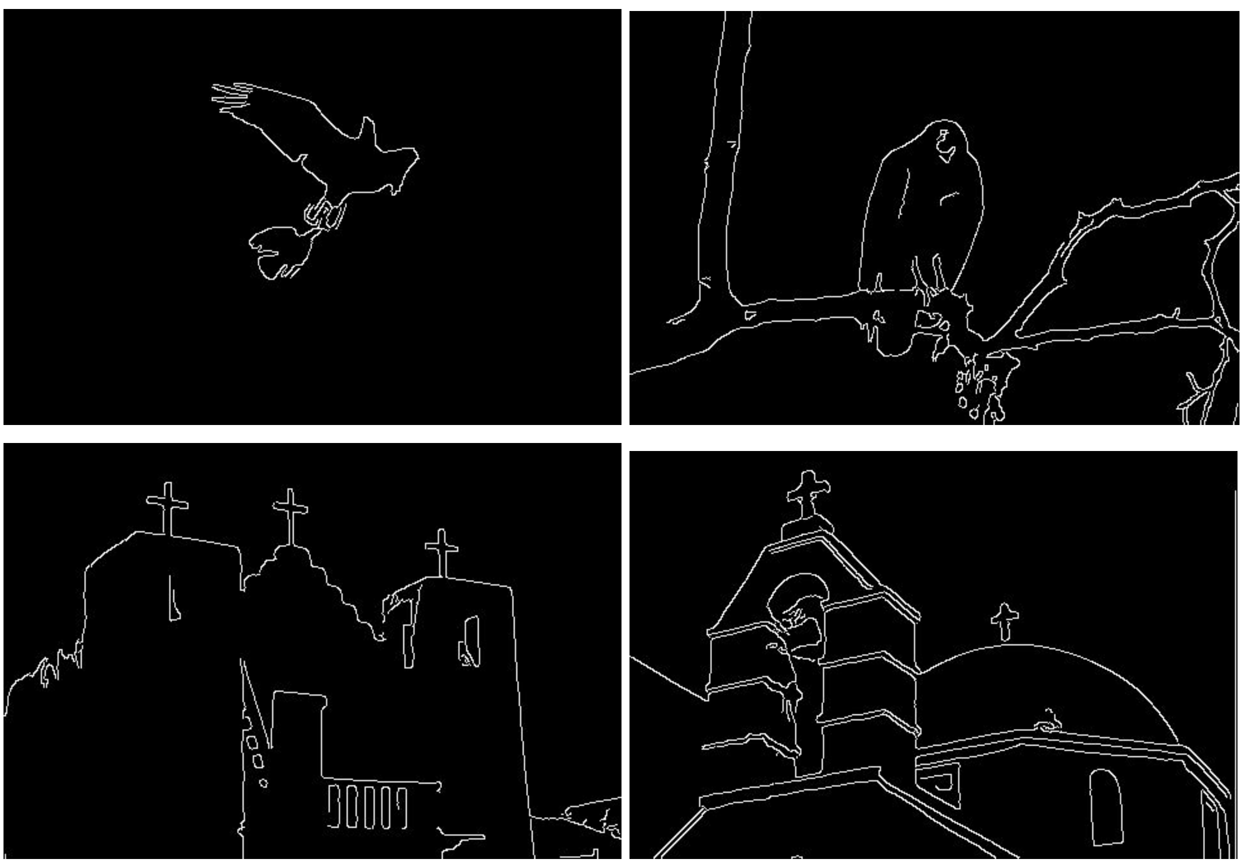
Test Data:

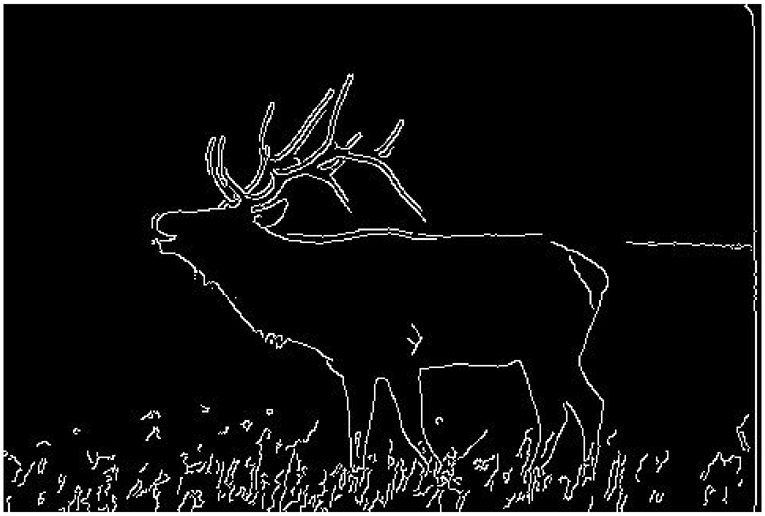
Below we present some of our test images, more results are uploaded along with the repository.



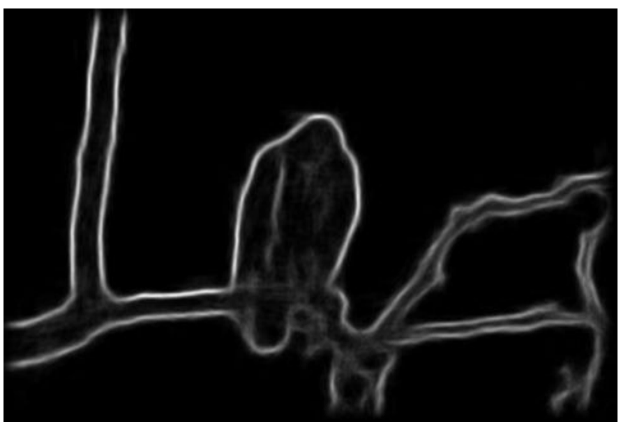
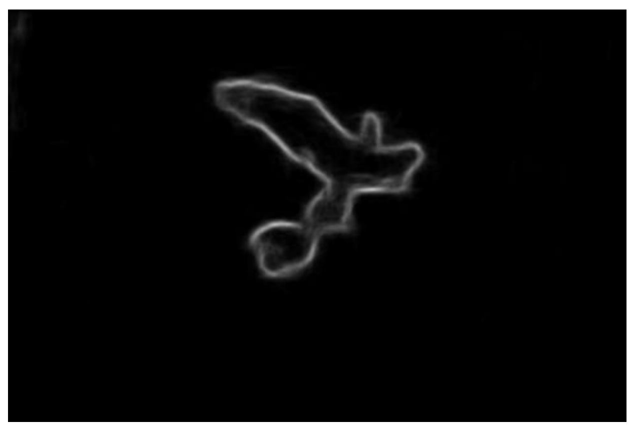
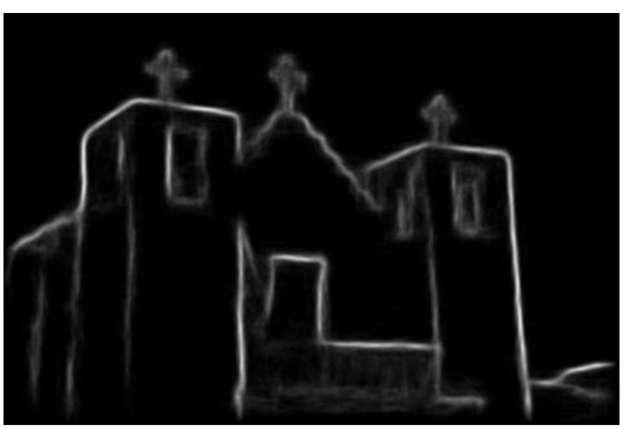

Results:

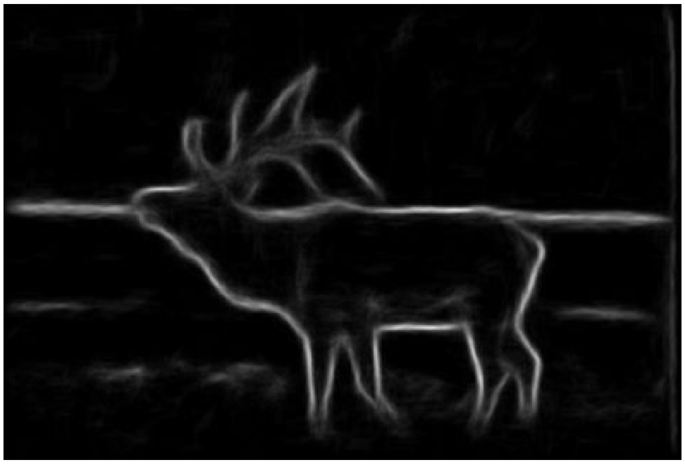
Below are the results for our statistical approach



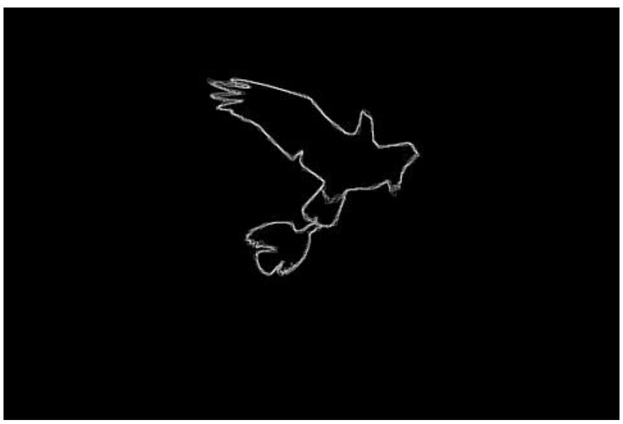
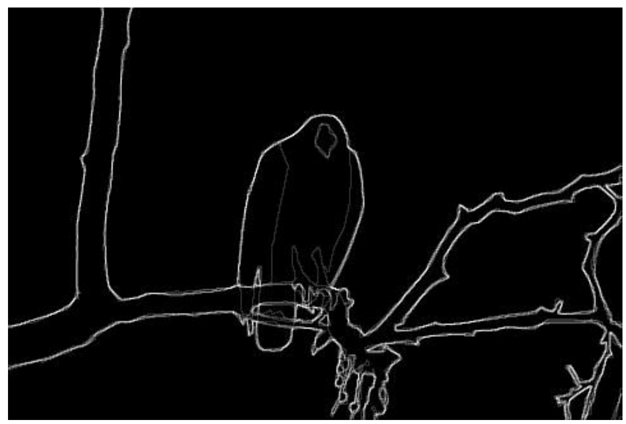


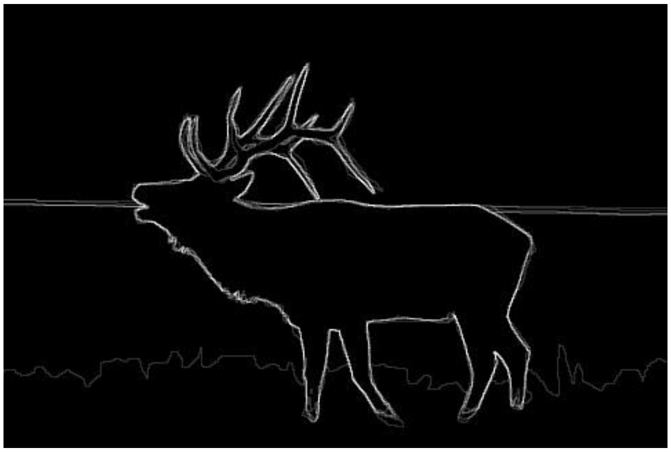
Below are the results for our structured forest approach

Below are the ground truth annotations of the example images.

Discussion:

We can see that with the statistical approach we have results which are very much similar to the ground truth observations. This is because the thresholds are dependent on the individual image and not hard coded, hence the segments are comparable to the human groundtruth. On hard coding the results were also computed and they were not as good as the statistical based approach. The outputs of the structured forest approach are also comparable enough to the ground truth.

Reference:

- Dataset used with the images and human annotations : The Berkeley Segmentation Dataset and Benchmark

- OpenCv

- Dollár, Piotr, and C. Lawrence Zitnick. "Structured forests for fast edge detection." *Proceedings of the IEEE International Conference on Computer Vision*. 2013.

- Lim, Joseph J., C. Lawrence Zitnick, and Piotr Dollár. "Sketch tokens: A learned mid-level representation for contour and object detection." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013.