CS 512: (Computer Vision)

Final Project Report

Title	*Image Segmentation Based on Global Extraction and Local Repair of Boundaries*	
Team	*Veera Raghava Reddy Katamreddy* ***(A20339534)***	*Noor Afshan Fathima* ***(A20385838)***

Introduction:

Image segmentation is a process of partitioning a digital image into multiple segments. The goal is to represent the image into features, easier to analyze. There are two main techniques for image segmentation: Region based, and Edge based techniques. In this project we specifically focus on the Edge based image segmentation. The edges computed in this project are based on intensity gradients in an image only. It doesn't specifically concentrate on the texture gradients as gradients are highly sensitive to noise and textures acts like noise in a gradient function. This project addresses the methods to compute the boundaries in the image and repair the boundaries using edge repair techniques and finally compute a mask to isolate the most significant edges in the image. [9]

Approach:

The implementation is staged into 3 phases: [8]

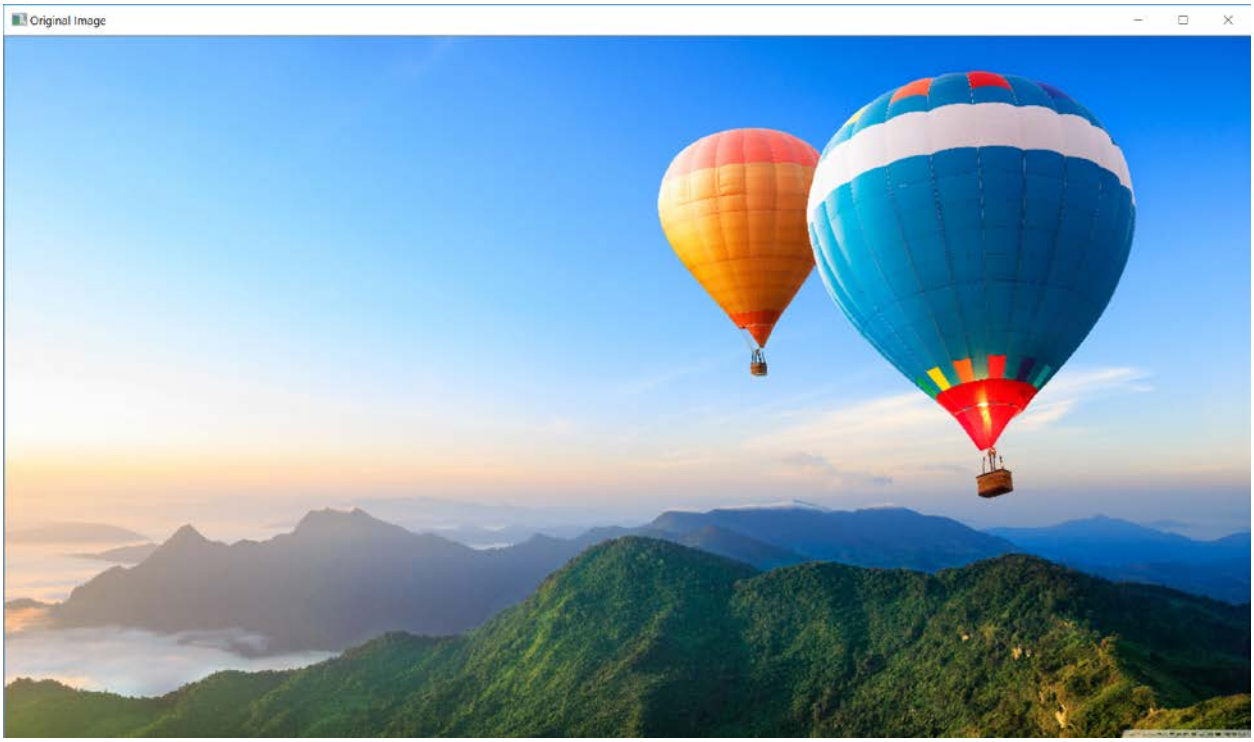
- Extracting Global boundaries by computing edges with canny operator.
- Repairing Local boundaries to close the boundaries as much as possible.
- Morphological processing to obtain a mask to isolate the most significant boundaries based on the computed boundaries from above.

Methodology:

The Image signal is converted to Grayscale to help perform faster, estimating boundaries require estimating image gradients which require computing image derivatives, But Image derivatives are very sensitive to noise, so smoothing is required before derivatives are computed.

- **Smoothing:**

  Smoothing is a process of diffusing the intensity of the pixels to the surrounding pixels. Here Gaussian smoothing is applied (Low-pass filter) to attenuate the high frequency components in the image. Because gaussian low-pass filter has excellent performance in practice. Here we use a two-dimensional gaussian filter. The gaussian filter is convolved with the image signal, giving a smoothed signal. The kernel size used is standard to 5, variance of (1,1) along x and y axis. [8]



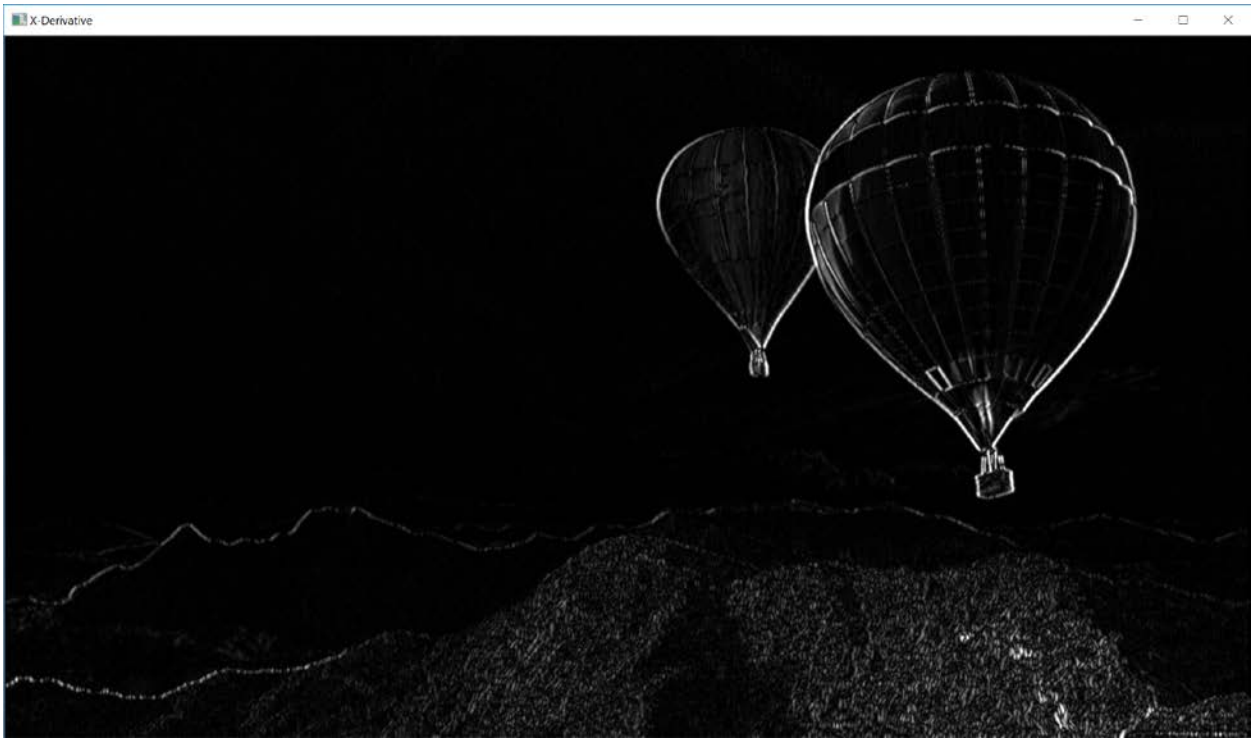
Original Image

Grayscale Image

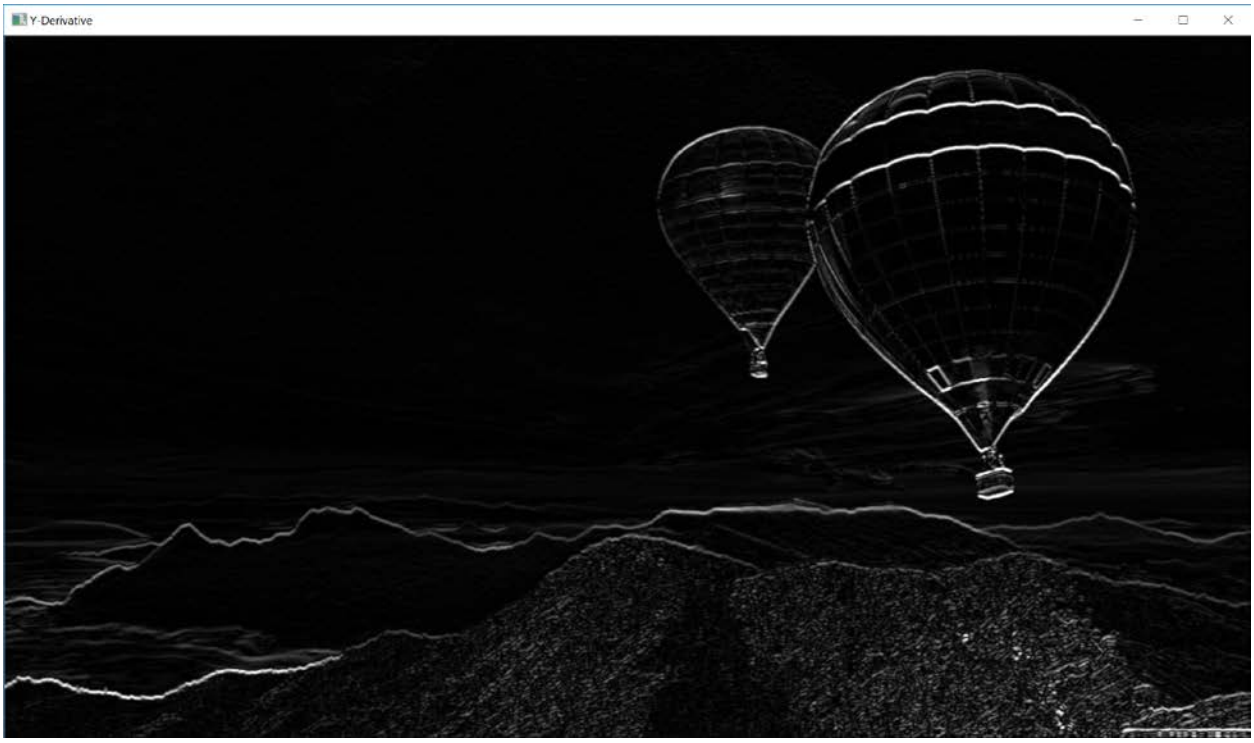
Gaussian Smoothed Image

- **Computing Image Derivatives:**

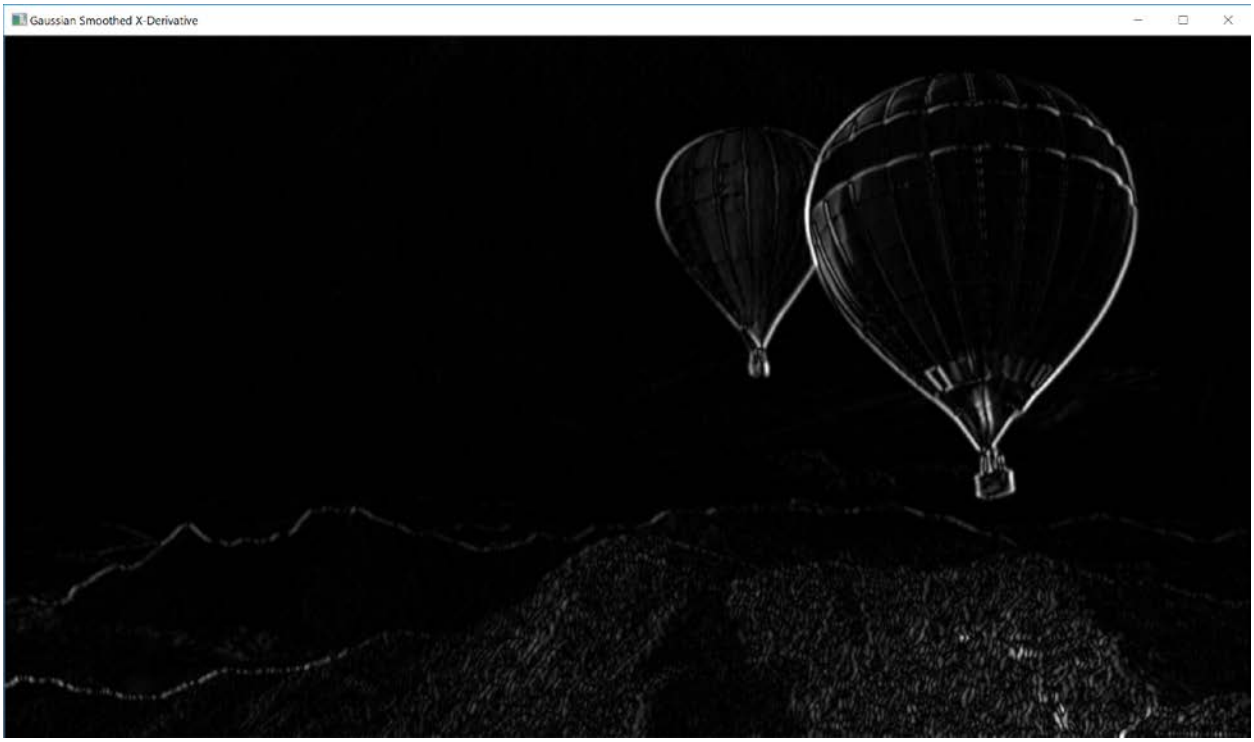
  Computing image derivatives is a vital in boundary estimation. A Sobel operator of Kernel size 3 is standardly used to estimate the image derivatives. Image derivatives along both X and Y Axis are computed using the Sobel operator. To help prevent the effects of noise the image derivatives are re-smoothed with a Gaussian low pass filter with similar Kernel size 5, but with a variance of (1.5, 1.5) along x and y axis. This smooths out the noise to a manageable level. [8]



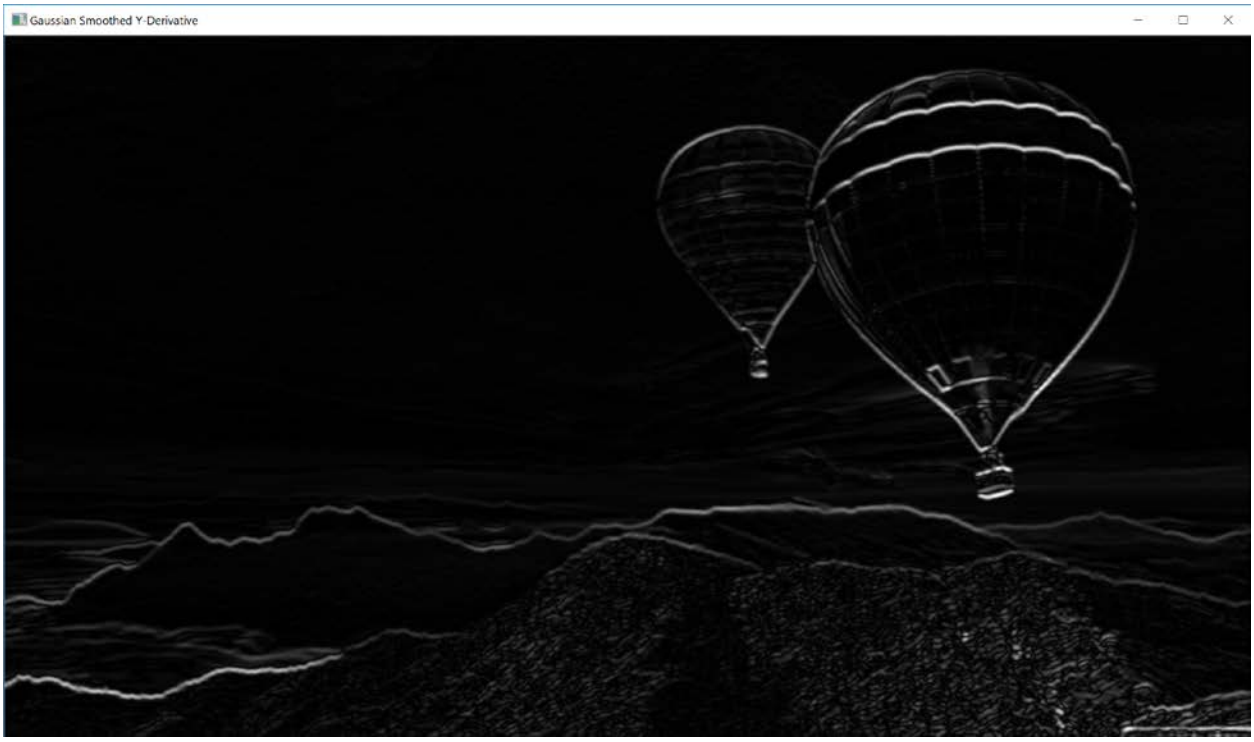
X-Derivative (Sobel Operator)



Y-Derivative (Sobel Operator)



Gaussian Smoothed X-Derivative

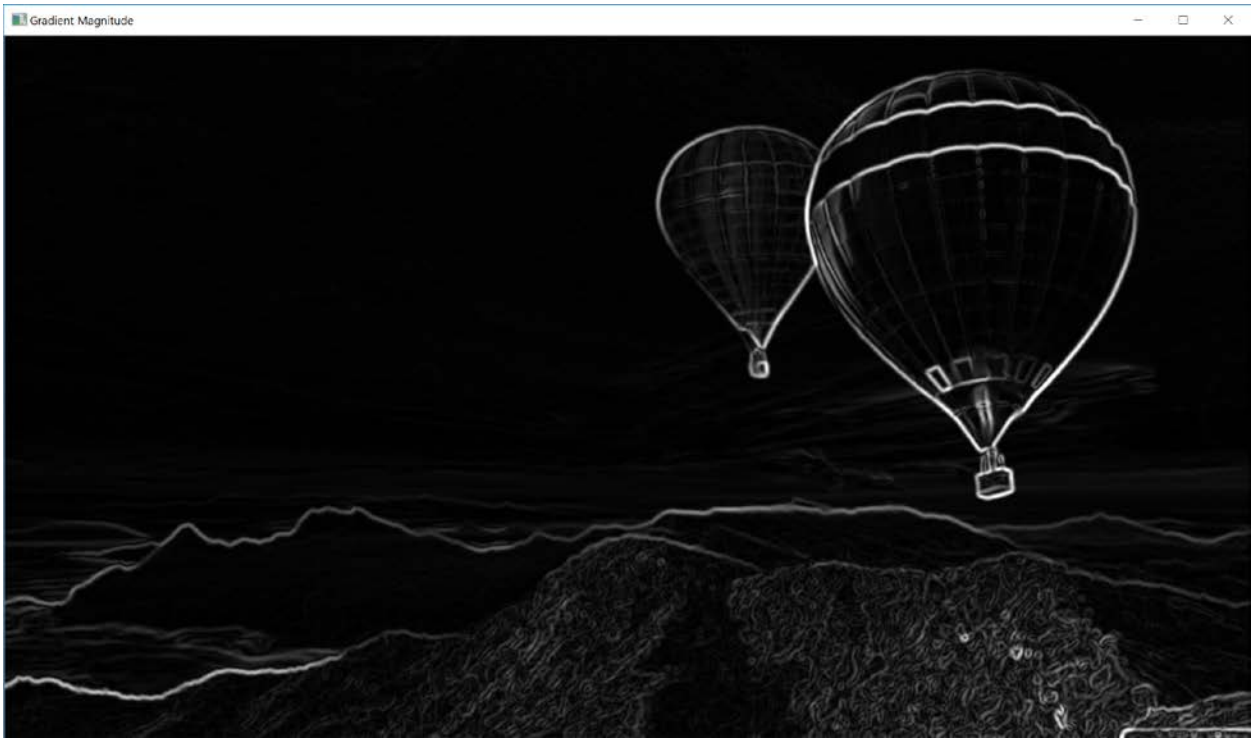


Gaussian Smoothed Y-Derivative

- **Computing Gradient Magnitude:**

  A resultant gradient magnitude is computed from the X and Y Derivatives of the image from the previous step. Given as:

$$\mathbf{G} = \sqrt{\boldsymbol{X^2 + Y^2}}$$



Gradient Magnitude

- **Computing Angles:**

  Angle of gradients are computed from the X and Y derivatives of the image from the previous step. Given as:

$$\mathbf{A = arctan2\ (Y / X)}$$

- **Computing Absolute Angles:**

  Absolute angles are required for the next steps in Boundary creation. Absolute angles are given in degrees. Given as:

$$\mathbf{A = A * 180 / pi}$$

- **Non-Maximum Suppression:**

  This is a crucial stage in boundary detection, using the gradient magnitude we can estimate the rough boundary of the significant, but for estimating the precise boundaries we need to suppress the gradients that are not significant or redundant. Here Non-Maximum suppression is done by interpolation of gradient vectors and suppressing the gradients that are redundant and insignificant gradients with low intensity. [8]

Non-Maximum suppressed image

- **Threshold Based Edge Isolation:**

  In this step a user defined High and Low threshold ratio is used to isolate the Strong and Weak Edges in the image. All the pixels with values less than the threshold value becomes zero. Eliminating the insignificant pixels altogether. This step gives Strong Edges, Weak Edges and Combined Edges, Strong Edges act as baseline, while weak edges act as support points during the repair phase. [8]

Strong Edges after Thresholding

Weak Edges after Thresholding

Combined Edges after Thresholding

- **Hysteresis Tracing and Edge Repairs:**

  In this step we trace the weak edges attached to the strong edges based on the Threshold images computed above. In this phase if the weak edges are connected to the Strong edges we set the value of the pixel to 1 else we set the value of pixel to 0, thus eliminating a lot of noise and insignificant weak edges. For this to work properly the threshold value selected by the user should be as minimal as possible. So, there won't be a need for multiple edge detection steps. After this step all the edges that are insignificant are destroyed leaving only the Strong edges and their connected weak edges to complete a boundary. However, some boundaries cannot be connected due to insignificant edge pixels etc. which will be addressed in the next phase (Morphological processing). The Hysteresis traced image is given below with cleaning of weak edges and boundary completion to a maximum extent. [8]

Hysteresis traced Image

In these steps are part of boundary estimation, so far, the boundaries are estimated and repaired based on the threshold values given by the user.

- **Morphological Processing for Image Segmentation:**

  When the image is obtained after boundary repair, we apply morphological transformation. The following morphological transformations are used to isolate the boundary of interest.

  [1]. Dilation:

  In this stage if an edge pixel has a value "1", a kernel with all ones runs as a sliding window increasing the pixel area. This stage increases the white region in the image. This is a useful method to join broken parts. Paired with bridge operation, which connects the disjoined pixels and creates a closed boundary. [8]

  [2]. Flood Fill:

  In this stage a binary mask is formed, which separates the foreground from the background. The pixel (0.0) is connected to the background, therefore we can extract the background by applying a flood fill operation from pixel (0,0). We next invert the flood filled image (black is swapped with white and vice-versa). We then combine the dilated image with the inverted image using bitwise OR operation to obtain the final foreground mask with holes filled in. [8]

[3]. Erosion:

In this stage it erodes away the boundaries of foreground object. The kernel slides through the image, a pixel in the obtained image will be considered 1 only if all pixels under the kernel is 1, otherwise it is made to zero. All the pixels near boundary will be discarded depending upon the size of the kernel. [8]

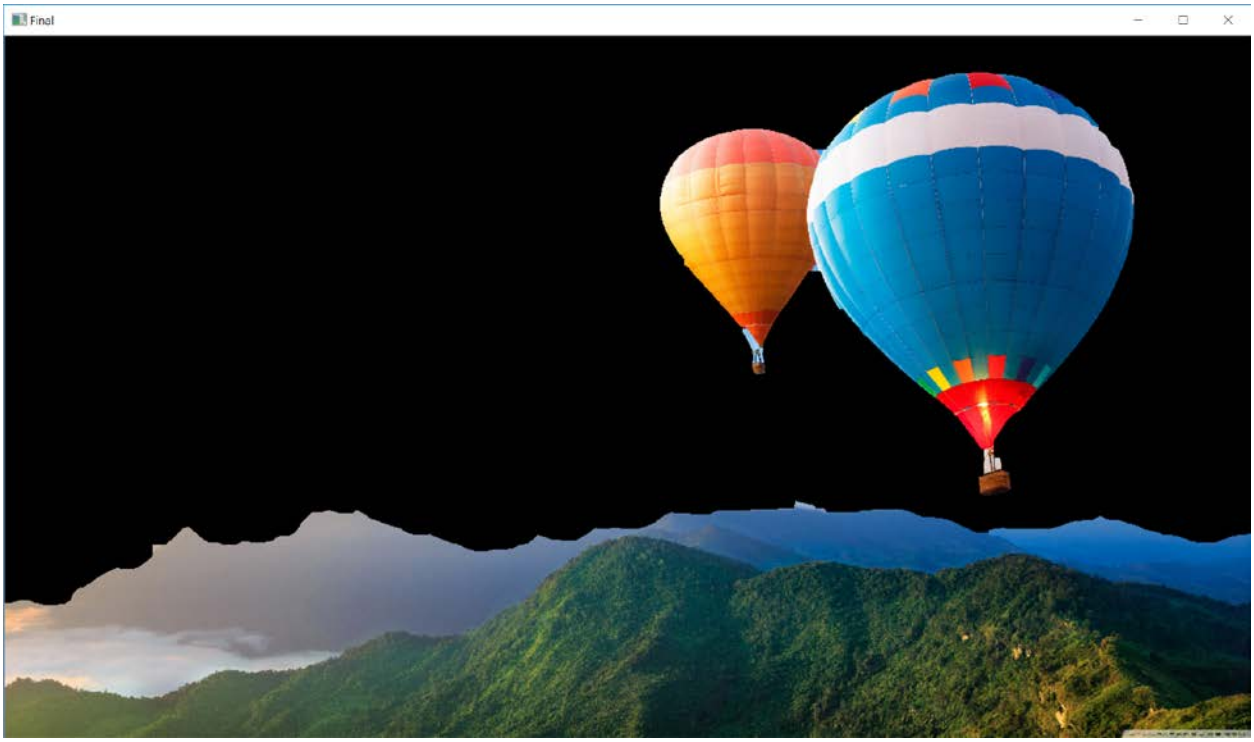
From this above procedure large gaps if remain disconnected will be eliminated by erosion. When we convolve the original image with the mask obtained after processing we can extract the foreground from the background

Morphological Transformed Mask (Dilate + Bridge + Flood Fill + Erode)

The Final mask obtained here resembles the most significant boundary in the image, based on the Intensity gradient computed in the previous steps.

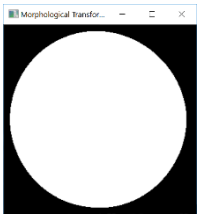
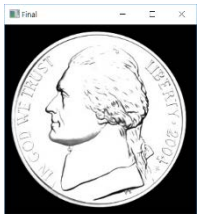
- **Applying Mask to extract final image:**

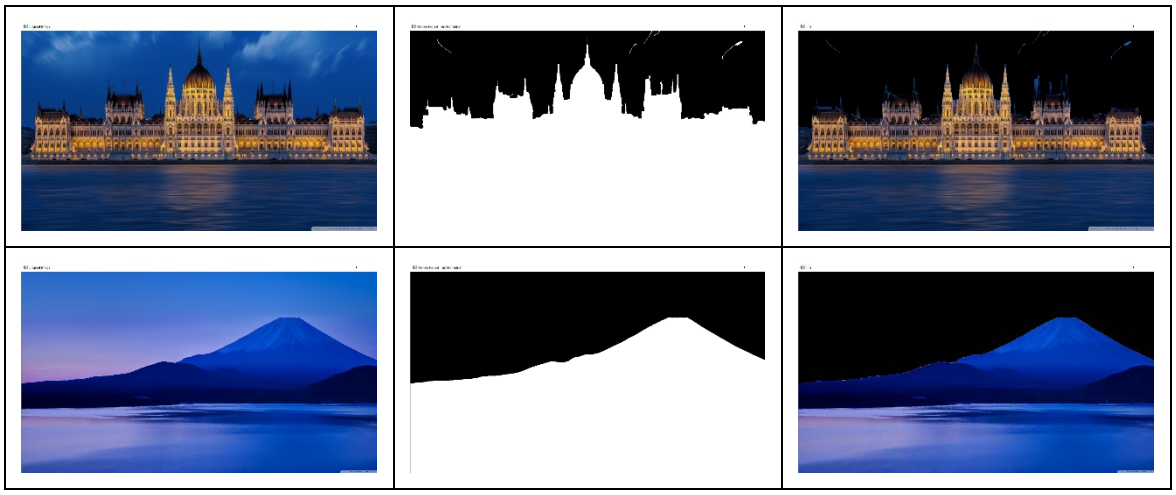
  Using the above obtained binary image as mask, original image is extracted using a bit wise or function. [8]



Original image with mask extracted from it

More Instances:

Original Image	Mask	Final Extract
		
		
		



Assumption:

The Algorithm is very effective in computing boundaries if the derivatives have only intensity gradients, however if the images have texture gradients the derivatives will have lot of noise to deal with which requires contextual smoothing based on the object. Which becomes a machine learning problem. So, for that reason texture based gradients will have very bad effects on the segmentation method proposed.

Performance:

The Algorithm implemented is image segmentation based on edge detection. So, there is an inherent drawback with edge detection under noisy image data, which is the reason why the algorithm cannot perform better on textured images. However, we cannot assume the performance of the algorithm based on Ground truth values of the segments as we try to estimate the boundaries based on the most significant edge pixels computed via Non-maximum suppression, Thresholding and Hysteresis tracing. Which means the performance of the algorithm depends on the image signal, type of gradients in the image and the efficiency of the repairs based on the gradients of the image. The Algorithm implemented can isolate edge pixels well enough up to a point which is why we have the issue of Morphological transformations. Since all the boundaries cannot be estimated under all circumstances, Dilation is one good way to improve the boundary extraction. Flood fill gives the final mask with most edge pixels as boundary.

Therefore, the quality of the segment is the performance in this case, If the problem is to isolate a mask in an image with subtle intensity gradients, the algorithm works at its best. If there are texture gradients in the image, then the derivatives create lot of weak pixels connecting to the strong pixels leading to in appropriate boundaries in the image.

Bibliography:

Software Libraries used:

[1]. Open CV 3.3
[2]. Numpy
[3]. Python 3.6

Web resources used:

[4]. Open CV Documentation
[5]. http://www.codepasta.com/site/vision/segmentation/Python 3.6
[6]. https://dufferdev.wordpress.com/2014/12/21/image-segmentation-using-k-means/
[7]. https://pythonprogramming.net/loading-images-python-opencv-tutorial/

Research Articles used:

[8]. Image Segmentation Based on Global Extraction and Local Repair of Boundaries; *http://ieeexplore.ieee.org/document/5522821/*
[9]. https://en.wikipedia.org/wiki/Image_segmentation