PROBLEM STATEMENT - 1

IMAGE SEGMENTATION

Definition and Importance

Image segmentation is the process of separating and identifying objects in an image, allowing extraction of features and data from the image.

Image segmentation has become a crucial tool across various industries:

1. **Medical Imaging**: Image segmentation is widely used in radiography, MRI and CT scans. It supports tasks such as brain segmentation and surgical planning.

2. **Autonomous Driving**: Self-driving cars rely on image segmentation to navigate safely by identifying lanes, recognizing traffic signs, and avoiding obstacles.

3. **Manufacturing:** In manufacturing, image segmentation is used for product sorting, defect detection, and enhancing robotic automation tasks.

- **Thresholding -** Thresholding in image processing is a technique used to create binary images from grayscale images. The process involves setting a threshold value and converting all pixels in the grayscale image to either black or white based on whether their intensity values are below or above the threshold. This technique is widely used in various applications such as image segmentation, object detection, and feature extraction.
- **Clustering -** Clustering segmentation is a method of segmenting images by grouping pixels based on their similarity or proximity. It relies on clustering algorithms, such as K-means or Mean Shift clustering, to partition the image into distinct regions with similar attributes.
- **Deep learning techniques -** Deep learning techniques particularly **Convolutional Neural Networks (CNNs)**, have significantly transformed image segmentation by offering highly accurate and efficient solutions. These methods utilize a hierarchical approach to image analysis, where multiple layers of filters are applied to the input image to extract progressively complex features. For a deeper understanding, explore the fundamentals of Convolutional Neural Networks.

Basic Thresholding using OpenCV

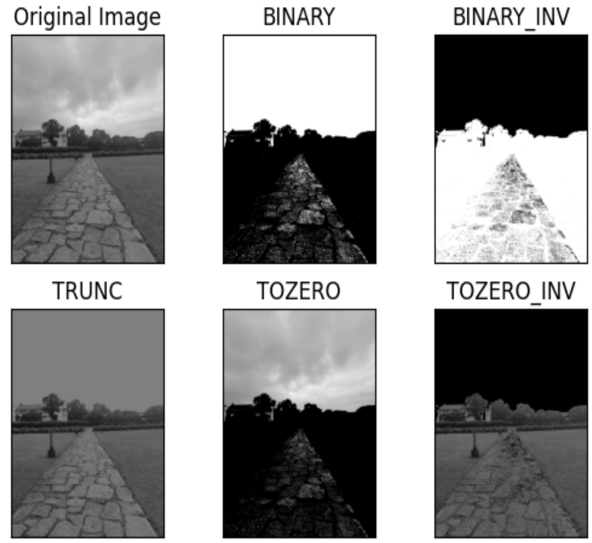
```python
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

img = cv.imread('bitsatoclane.jpg', cv.IMREAD_GRAYSCALE)
ret,thresh1 = cv.threshold(img,127,255,cv.THRESH_BINARY)
ret,thresh2 = cv.threshold(img,127,255,cv.THRESH_BINARY_INV)
ret,thresh3 = cv.threshold(img,127,255,cv.THRESH_TRUNC)
ret,thresh4 = cv.threshold(img,127,255,cv.THRESH_TOZERO)
ret,thresh5 = cv.threshold(img,127,255,cv.THRESH_TOZERO_INV)

titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in range(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray',vmin=0,vmax=255)
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])

plt.show()
```



K-Clustering using OpenCV

Color Quantization - Color quantization is critical for displaying images with many colors on devices that can only display a limited number of colors, usually due to memory limitations, and enables efficient compression of certain types of images. The k-means clustering algorithm takes unlabelled data points. It seeks to assign them to k clusters, where each data point belongs to the cluster with the nearest cluster center, and the center of each cluster is taken as the mean of the data points that belong to it. The algorithm requires that the user provide the value of k as an input; hence, this value needs to be known a priori or tuned according to the data.

```python
import numpy as np
import cv2 as cv
from google.colab.patches import cv2_imshow
img = cv.imread('photo_9.jpg')
Z = img.reshape((-1,3))
Z = np.float32(Z)
criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 10, 1.0)
K = 4
ret,label,center=cv.kmeans(Z,K,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)
center = np.uint8(center)
res = center[label.flatten()]
res2 = res.reshape((img.shape))
cv2_imshow(res2)
cv.waitKey(0)
cv.destroyAllWindows()
```




k=2 **k=4** **k=8** **k=16**

Applications in Autonomous Driving

Image segmentation plays a vital role in autonomous driving. Self-driving cars need appropriate pixel knowledge about the surroundings to drive without accidents. Therefore, image segmentation can assist in recognizing lanes, traffic lights, highways, street signs, cross marks, other cars, pedestrians, and other essential information. The ability of autonomous cars to perceive, understand, and react to their environment in real time is greatly improved by image segmentation. This is especially important while negotiating intricate and changing traffic situations.

Use cases -

1. **Lane Detection -** Lane detection is one of the most important aspects of an autonomous car. A camera, usually installed in the center of the dashboard behind the windshield, provides a constant stream of images of the road ahead. Lane detection is necessary, as it provides the car with information about whether it is maintaining lane discipline, and the angle of the lanes can tell it about upcoming turns. It is responsible for keeping the vehicle in the middle of the lane at all times.

   The process implemented goes through a few steps:

   - Gray scale conversion

- Yellow and white filter
- Gaussian blur
- Canny edge detection
- Region of interest
- Hough transform

2. **Object Detection -** Object detection using image segmentation is a critical technology in autonomous driving, enabling vehicles to perceive their environment accurately. The preprocessed images are fed into an object detection model, often based on deep learning architectures like Faster R-CNN, YOLO, or SSD. These models detect and classify objects such as vehicles, pedestrians, cyclists, and traffic signs in the image. The output is a set of bounding boxes with associated class labels.

3. **Traffic Sign Detection -** Traffic sign detection and recognition systems play a vital role in enhancing road safety and the efficiency of transportation systems. These systems are instrumental in aiding drivers and autonomous vehicles in understanding and obeying traffic regulations.The conventional methods of traffic sign recognition often faced challenges related to variations in illumination, sign occlusion, and complex backgrounds. However, the integration of deep learning techniques like CNNs has revolutionized the field, allowing for highly accurate and robust recognition.

Challenges

- Hazardous driving scenarios often involve variations and anomalies in the environment that may go beyond the predefined categories. These variations can include unexpected objects, unusual conditions, or dynamic elements that pose risks to the safety of autonomous vehicles.
- Collecting training datasets that encompass every possible variation in hazardous scenarios is impractical, as it would require extensive effort, time, and resources. Therefore, semantic segmentation models need to address the challenge of detecting and adapting to variations and anomalies that may arise in real-world driving environments.

- Objects like pedestrians and vegetation are very difficult to segment accurately because of their complex shapes, complex motion, different sizes and the different types of clothing pedestrians can be wearing.
- Challenges with adverse weather, changing lighting conditions, and the computing difficulties involved in real-time segmentation still need to be solved. The goal of overcoming these obstacles and advancing technology is to increase the reliability and durability of autonomous driving systems through continued research and development.
- Time Constraints: Real-time analysis requires timely and efficient processing to ensure immediate responses and decision-making, leaving minimal room for processing delays.
- Processing Power: Real-time analysis requires significant computational power to perform pixel-level classification quickly, necessitating efficient hardware and optimized algorithms.

Solutions

- Real-time task scheduling and system optimization to reduce processing delays.
- Data fusion techniques, advanced sensor technologies, and efficient data preprocessing methods.
- Future directions of segmentation in autonomous driving include exploring advanced attention mechanisms, optimizing the use of synthetic data, and expanding the applicability of the method to other perception tasks in autonomous driving systems. Semantic segmentation advancements contribute to safer and more efficient autonomous driving in various scenarios.