19CSE413
DIGITAL IMAGE PROCESSING

CB.EN.U4CSE20309
AVANTIKA BALAJI

EVALUATION - 1

Problem Statement:
The goal of our project is to develop a Pothole Detection System using various image processing techniques that will serve as a framework to assist drivers avoid potholes on poorly constructed roads. We chose a dataset in which we assume that an image sensor on the front of the vehicle will point at a low angle on the road to detect potholes.

Abstract:
Roads serve as a platform for transportation and contribute significantly to the economy. One of the main issues with transportation infrastructure is potholes on the roadways. Numerous studies have suggested employing computer vision techniques, such as various image processing and object detection algorithms, to automate pothole detection. Pothole detection must be automated with sufficient speed and accuracy, and the system must be simple to use and inexpensive to set up. Here, we are investigating and applying several spatial processing techniques and drawing an analysis based on their outcomes. Image noise is a random variation of brightness or color information in images, and it is commonly associated with electronic noise. It can be generated by the digital camera's image sensor and circuitry. Image noise can also be caused by film grain and the inherent shot noise of an ideal photon detector.

Spatial Processing Methods explored :

1. **Image acquisition** *(storage and retrieval - position of camera)*

- Image acquisition in image processing and machine vision is the process of obtaining an image from a source, typically hardware such as cameras, sensors, etc.
- Since the system cannot perform any meaningful processing without an image, it is both the first and most crucial stage in the workflow sequence. Typically, the image acquired by a system is a fully unprocessed image.

Why do we need image acquisition?
One of the ultimate objectives of this procedure is to have a source of input that operates within such controlled and measured parameters that the same image can, if necessary, be almost flawlessly recreated under the same conditions, making it easier to find and remove anomalous variables.

Image Pre-Processing:

2. **Image Interpolation - Image Resizing**

- The process of resizing or distorting an image from one pixel grid to another is known as image interpolation. When the total number of pixels needs to be increased or decreased, image resizing is required.
- Image interpolation works in two directions, attempts to achieve the best approximation of a pixel's color and intensity based on the values of neighbouring pixels.
- When you resize an image without resampling it, you change the size of the image without changing the amount of data in the image. There is no data added or removed from the image.

Why are we resizing our image?

- Since neural networks receive inputs of the same size, all images need to be resized to a fixed size before inputting them to the CNN. The larger the fixed size, the less shrinking required. Less shrinking means less deformation of features and patterns inside the image.
- An input image that is twice as large requires our network to learn from four times as many pixels — and that time adds up.
- Using the original pothole image may result in a longer computational time. As a result, the image must be resized to a lower resolution for faster processing and improved segmentation.

3. **Gray Scale Conversion**

The average pixel values (ranging from 0-255) of the primary colours red, green, and blue (commonly known as RGB) are combined. Each colour band's luminous intensity (24 bits) is combined into a reasonable approximated grayscale value (8 bits)

Why do we convert color image to gray scale image?

- For many applications of image processing, color information doesn't help us identify important edges or other features. So if we don't need color, then we can consider it noise which in turn helps reduce color complexity
- Simplifies algorithms and eliminates complexities related to computational requirements
- Also helps enhance visualisation as it can differentiate between shadow details and highlights of an image as it is majorly in 2D

Best application of gray scale in real-life:

Image segmentation and object detection in the medical field

- It is pivotal because ultrasound, X-ray, and computer tomography (CT) scan images rely heavily on them to provide accurate advice and treatment.

- Image segmentation is crucial here because the various organs and tissues of the human body have different grayscale values. Each organ or tissue's distinct characteristics and defects are thus easily identified.

CODE

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from PIL import Image, ImageFilter

from google.colab import drive
drive.mount('/content/drive')
```

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import glob
import os
from matplotlib import pyplot as plt
from os import listdir
from scipy.ndimage.filters import median_filter
img_dir = "/content/drive/MyDrive/Pothole_dataset" # Enter Directory of all images
data_path = os.path.join(img_dir,'*jpg')
files = glob.glob(data_path)
for f1 in files:
   img = cv2.imread(f1)
   # resize image
   img = cv2.resize(img, (300, 300), interpolation = cv2.INTER_LINEAR)
   print("Original image")
   cv2_imshow(img)
   #gray scaling
   gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
   print("Gray-scale image")
   cv2_imshow(gray_image)
```

OUTPUT SCREENSHOTS: