IMAGE FILTERING

Prepared in the partial fulfillment of the Summer Internship Program on Data Analysis

AT

Under the guidance of

Mrs. Meenakshi, APSSDC

Mr. Narmada Mani, APSSDC

Submitted by

Guruvardhini K – 21FE1A0471

Vignan Lara Institute Of Technology And Science, AP

(August, 2023)

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of my summer internship project at **Andhra Pradesh Skill Development Corporation (APSSDC)**. This opportunity has been an enriching and transformative experience for me, and I am truly thankful for the support, guidance, and encouragement I have received along the way.

First and foremost, I extend my sincere regards to Mrs Meenakshi and Mrs Narmada Mani, my supervisors and mentors, for providing me with valuable insights, constant guidance, and unwavering support throughout the duration of the internship. Their expertise and encouragement have been instrumental in shaping the direction of this project.

I would like to thank the entire team at **Andhra Pradesh Skill Development Corporation (APSSDC)** for fostering a collaborative and innovative environment. The camaraderie, knowledge sharing, and feedback I received from my colleagues significantly contributed to the development and success of this project.
In conclusion, I am honoured to have been a part of this internship program, and I look forward to leveraging the skills and knowledge gained to contribute positively to future endeavours.

Thank you.

Sincerely,

Guruvardhini Kolakani-21FE1A0471guruvardhini_kolakani@srmap.edu.in

ABSTRACT

Image processing is an area consisting of different phases for manipulating pixels of an image. Image processing is an enhancement of images which affects an image's pictorial view. Image processing is the process of converting an image into digital form and then performing various operations on it, such as enhancing the image or extracting useful information. Image filtering is a technique for altering the scale, shape, colour, depth, smoothness, and other aspects of images. Essentially, it modifies the pixels of an image to transform it into the desired form using various types of graphical editing methods via graphic design and editing software. In Image Improvement, the removal of noise is very necessary and filters are used for this. This project introduces some filters that will give you a better version of your image. Using different kinds of filters provides you noise free images.

CONTENTS

INTRODUCTION

Image filtering is a step in the image pre-processing process. The recovery of distinct signals from the correspondence channel with no noise is a difficult task. For the removal of noises from a digital signal, a variety of denoising methods have been suggested. Wavelet denoisingwith edge calculation is a useful technique for reducing noise in digital signals.

Images andvideos during the acquisition and transmission phase, impulse noises are often distorted. Thisimpulse noise appears in a picture as a result of bit errors in transmission or as a result of signal acquisition errors. When it comes to detecting edges and contours, noise has a majoreffect on detection accuracy. As a result, noise must be removed and the pixel's intensity must be regulated.

Image filtering removes noise and unwanted features from an image, resulting in a better and more enhanced version. In this project, we are trying to solve a problem with the image filters which are used to increase or decrease noises present in images, also dealing with hue(s).

There are different kinds of image noise, we are providingsome filters that will help to reduce/increase the noise based on user requirements. So, why are we using OpenCV? OpenCV is a library designed to solve a wide number of tasks relating to computer vision. OpenCV supports a wide range of object-oriented languages, including Python, C++, and Java, and it is available on a variety of platforms, including iOS,Android, OS X, Linux, and Windows.

Interfaces based on OpenCL and CUDA are also being dynamically developed for rapid Graphics Processing Unit functions. The OpenCV Python API is referred to as OpenCV-Python. It combines the Python programming language with the OpenCV C++ API. Furthermore, using NumPy to help with the assignment makes it easier to complete. NumPy is an advanced standard library for numerical activities. It is filled with several basic and advanced features, very simple to pickup and available for many languages of programming.

It will be easy for us to provide different kinds of filters. In this we are dealing with some noises like salt and pepper noise, speckle noise. We can also enhance the edgesof an image to detect the feature of that image.

SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements that need to beinstalled on a computer to provide optimal functioning of an application.

- Operating System: Windows7/8/10/11
- Back End: OpenCV, NumPy
- Language: Python-3.6
- Platform: Jupyter Notebook

HARDWARE REQUIREMENTS

- Processor: Intel dual core i3/i5/i7
- RAM: 4 GB

ABOUT THE PROJECT

Image filtering is changing the appearance of an image by altering the colors of the pixels. Increasing the contrast as well as adding a variety of special effects to images are some ofthe results of applying filters. In order to obtain a high success rate of OCR (optical character recognition) performed on text images, the main target of filters is, however, toreduce the noise around characters in the image. Thereby, I created two new nonlinear efficient image filters (for RGB and ARGB images) and elaborated the optimization technique suitable for each of the filters in order to make them perform faster. The first filter, namely, Smart Contrast, increases the contrast of the image in a way depending onthe value of each component (Red, Green, and Blue) of each pixel in the image..

PURPOSE

The main purpose of this project is to reduce the noise of the images. It helps reducing the time to apply multiple filters .It uses various filters at a time and provide the output for input .It excludes human efforts and saves time and resources.

SCOPE

Image Filtering extracts information from images and integrates it for several applications.There are several fields in which image filtering applications are relevant. Medical imaging, industrial applications, remote sensing, space applications, and military applications are a few examples.

DATA STRUCTURE

NumPy, which stands for Numerical Python, is a library consisting of multidimensionalarray objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open-source project.

PROJECT DESCRIPTION

IMAGE FILTER DESCRIPTION:

Image filtering is changing the appearance of an image by altering the colors of the pixels. Increasing the contrast as well as adding a variety of special effects to images are some of the results of applying filters.

Different filters used in this project are:

Blur Image: OpenCV-Python is a library of Python bindings designed to solve computer vision problems. cv2.blur() method is used to blur an image using the normalized box filter.

Grey scaling Image: Gray scaling is the process of converting an image from othercolour spaces e.g. RGB, CMYK, HSV, etc. to shades of ray. It varies between complete black and complete white.

Grey blur Image: Image Blurring refers to making the image less clear or distinct.

The Median Filter often used to remove noise from an image or signal. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise.

Motion Blur Image: When we apply the motion blurring effect, it will look like youcaptured the picture while moving in a particular direction. For example, you can make an image look like it was captured from a moving car.

Cartoon image: Image cartoon frames don't contain so many variants of color and those frames also contain dark edges of each object.

There're lots of ways to do that, but we're going to use color quantization and the adaptive threshold technique. Using this technique, we can achieve a very great result.

Edge Detection image: Edge detection is an image processing technique, which is used to identify the boundaries (edges) of objects, or regions within an image. Edgesare among the most important features associated with images. We come to know ofthe underlying structure of an image through its edges. Computer vision processing pipelines therefore extensively use edge detection in applications.

Edges: Edges are characterized by sudden changes in pixel intensity. To detect edges, we need to go looking for such changes in the neighbouring pixels.

Dilation: In the Dilation, it increases the object area. The Erosion can remove the white noises, but it also shrinks our image, so after Erosion, if Dilation is performed,we can get betternoise removal results. The Dilation can also be used to joins somebroken parts of an object.

Erosion: In the Erosion, it erodes away the boundaries of foreground objects. It is used to remove small white noises from the images. Erosion can also be used to detach two connected images.

CODE:

Image_filter.py

```
# importing the
Librariesimport
numpy as np import
cv2
import imutils

# reading image by using path
args_image ="E:/manpic.jpg"
image =
cv2.imread(args_image)

#original
image=cv2.resize(image,(250,250))
cv2.imshow("Original Image",
image)


# blur
ksize = (10, 10)
blurimage = cv2.blur(image, ksize)
blurimage = cv2.resize(blurimage, (250,
250))cv2.imshow('blur_image',
blurimage)

#gray
grayImage = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)grayImage =
cv2.resize(grayImage, (250, 250)) cv2.imshow("gray",
grayImage)
#grayblur
grayImageBlur = cv2.blur(grayImage,(9,9))
grayImageBlur = cv2.resize(grayImageBlur, (250,
250))cv2.imshow("grayBlur", grayImageBlur)


#Motion Blur
kernel = np.ones((3,3),
np.uint8)size = 15
kernel_motion_blur = np.zeros((size, size))
kernel_motion_blur[int((size-1)/2), :] =
np.ones(size)kernel_motion_blur =
kernel_motion_blur / size
# applying the kernel to the input image
```

```
output = cv2.filter2D(image, -2,
kernel_motion_blur)output = cv2.resize(output,
(250, 250)) cv2.imshow('Motion Blur', output)


#Ca
rtoo
n#
Edg
es
gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)gray = cv2.medianBlur
(gray, 5)
edges = cv2.adaptiveThreshold (gray,
255,cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 9, 9)
# Cartoonization
color=cv2.bilateralFilter(image, 9, 250,
250)
cartoon = cv2.bitwise_and(color, color,
mask=edges)#blur
edges = cv2.resize(edges, (250, 250))
cartoon = cv2.resize(cartoon, (250,
250))cv2.imshow("Cartoon",
cartoon) cv2.imshow("edges",
edges)


#edges
edgedImage = cv2.Canny(grayImage, 100, 300, 3)
edgedImage = cv2.resize(edgedImage, (250,
250))cv2.imshow("Edge Detected Image",
edgedImage)


#dilation
image_dilation = cv2.dilate(image, kernel,
iterations=1) image_dilation =
cv2.resize(image_dilation, (250, 250))
cv2.imshow('Dilation', image_dilation)
#erosion
image_erosion = cv2.erode(image, kernel,
iterations=1) image_erosion =
cv2.resize(image_erosion, (250, 250))
cv2.imshow('Erosion', image_erosion)

# negative image
img_neg =
```

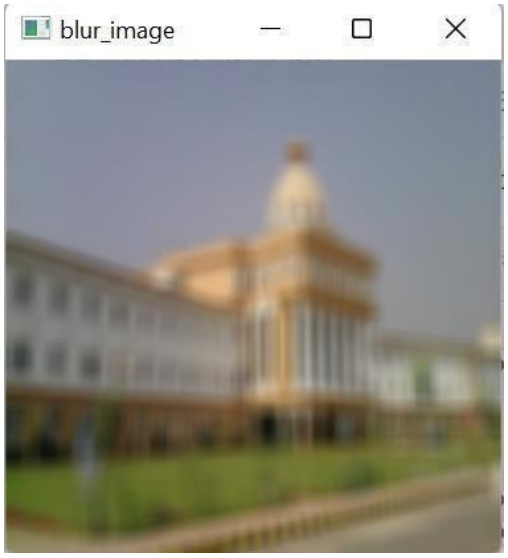
```
cv2.bitwise_not(image)
cv2.imshow('negative',img_n
eg)

cv2.waitKey(0)
cv2.destroyAllWind
ows()
```

OUTPUT SCREENS

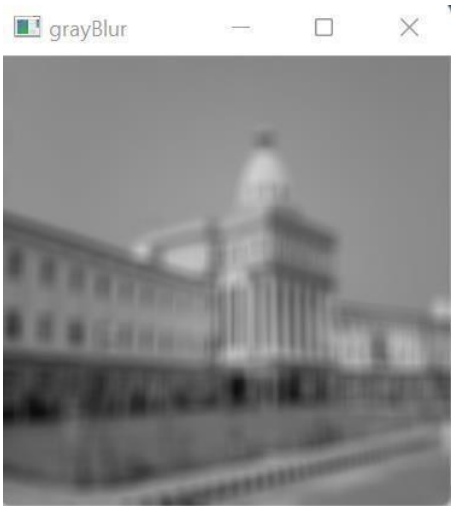
1. ORIGINAL IMAGE

2.BLUR IMAGE

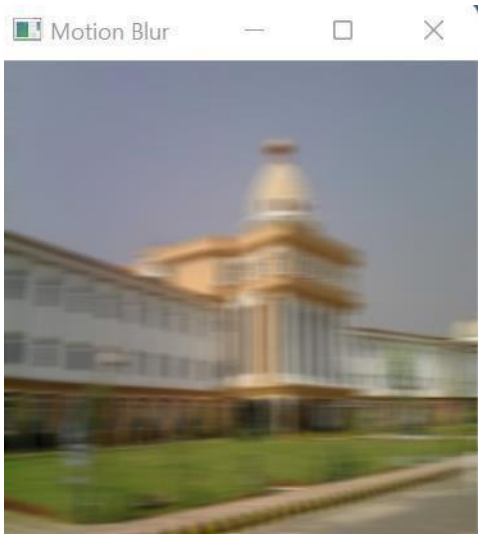


3.GREY SCALING IMAGE

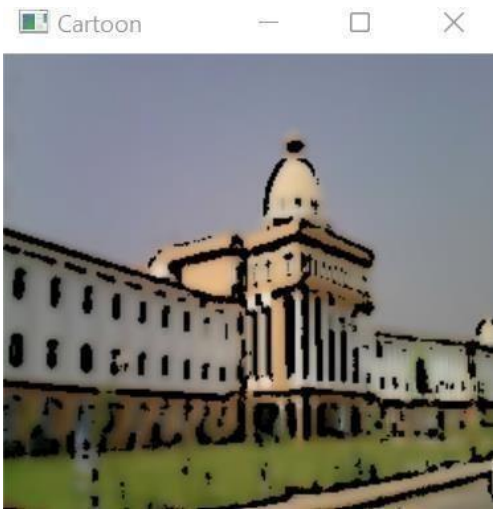
4.GREY BLUR IMAGE



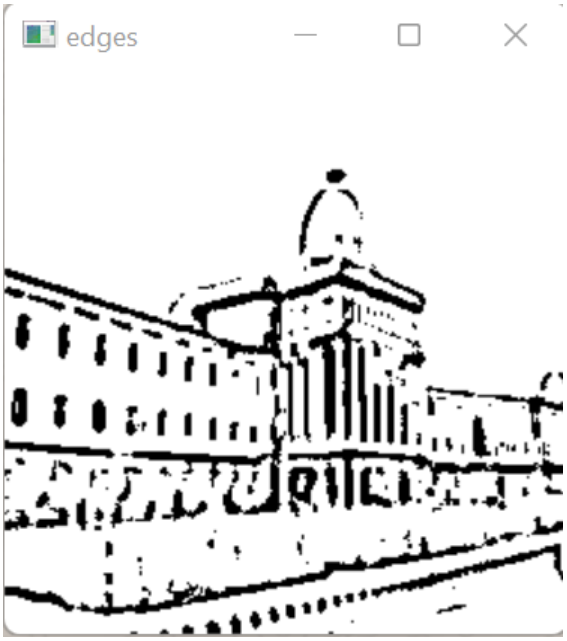
5. MOTION BLUR IMAGE



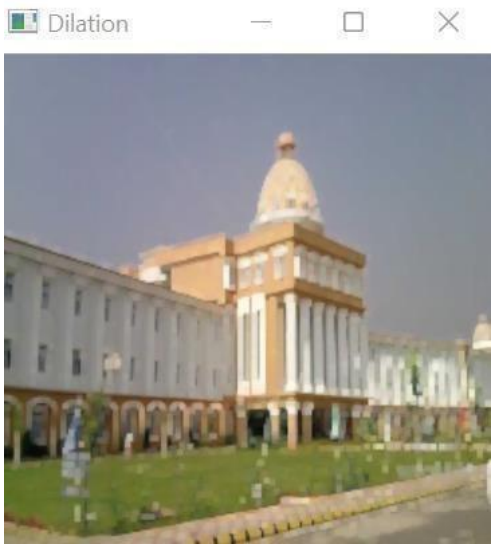
6.CARTOON IMAGE



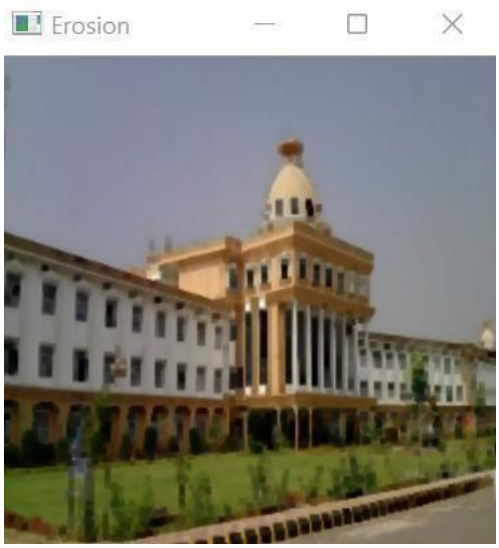
7. EDGES



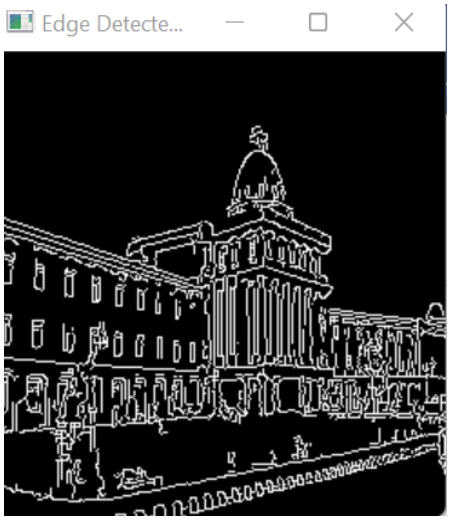
8. DILATION



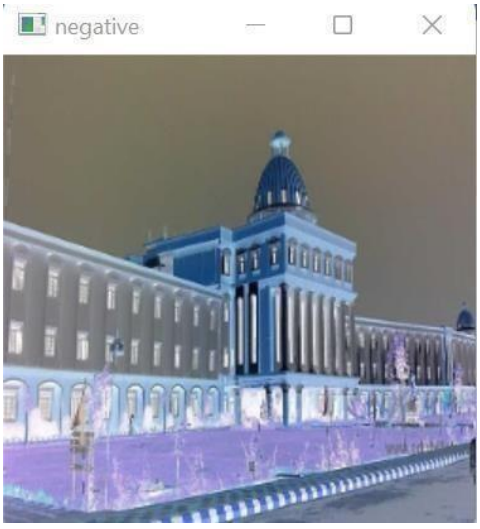
9.EROSION



10.EDGE DETECTION



11.NEGATIVE IMAGE



CONCLUSION

Image filtering has a wide range of applications including edge detection, removing noise, sharpening and smoothing. Image filtering has brought an evolutionary change in the field of image processing.

There are many more complex modifications you can make to the images. In this project, we have used some filters to the image and we have get accurate solutions for the inputs. By using filters we have decreased the noises of the image. we can apply multiple filters at a time and we can decrease the time to apply multiple filters. We can excludes human efforts and saves time and resources.