****

**DEPARTMENT OS COMPUTER SCIENCE AND ENGINEERING**

**CASE STUDY**

**ON**

**YCS302 Digital Image Processing**

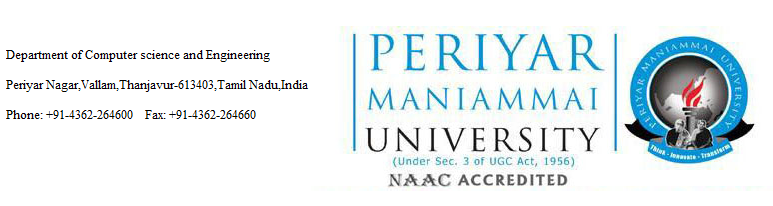
**SUBMITTED BY**

**A.KARTHIKEYAN**

**REG NO: 113012252085**

**SECOND YEAR**

**MASTER OF TECHNOLOGY**



**BONAFIDE CERTIFICATE**

Certified that this case study report on “**YCS-203 DIGITAL IMAGE PROCESSING**”is the bonafide work of “**A.KARTHIKEYAN”** Registration no:113012252085 of III semester M.Tech Degree in “MID SEMESTER CASE STUDY” During the year 2014-2015.

**Staff In-charge**

**Date :**

**Place:**

**TABLE OF CONTENT**

1.Abstract..................................................................................3

2.Basic Concepts.......................................................................3

3.Implementation......................................................................6

4.Screen shots...........................................................................10

5.Conclusion.............................................................................10

6.Reference...............................................................................11

**Abstract:**

Digital image processing is an area which is cover variety of application domains in the real world scenarios. So I am going to implement the concepts of Fourier Transformation technique such One-Dimensional Fourier Transform, Two-Dimensional Fourier Transform and Image Enhancement technique such as Image Inverse, Power Law Transformation and Log Transformation. The output of my Case study is shows how the Fourier Transform and Image Enhancement Techniques are working based on user input image and understand the concepts. In feature we enhance the Applications based on concepts of DIP and their applications.

**Basic Concepts:**

## Digital image processing

The digital image processing deals with developing a digital system that performs operations on an digital image.

**1D fourier transform:**

Fourier transformeis one of the most commonly used techniquesin(linear) signal processingand controltheory.

It provide sone-to-one transform of signals from/to a time-domain representation*f*(*t*) to/from a frequency domain representation *F*(*ξ*).

**2D fourier transform:**

The image function*f*(*x,y*) is decomposed to a linear combination of harmonic (sines and cosines,more generally orthogonal) functions.

## Image enhancement

Enhancing an image provides better contrast and a more detailed image as compare to non enhanced image. Image enhancement has very applications. It is used to enhance medical images , images captured in remote sensing , images from satellite e.t.c

The transformation function has been given below

s = T ( r )

where r is the pixels of the input image and s is the pixels of the output image. T is a transformation function that maps each value of r to each value of s.

## Gray level transformation

There are three basic gray level transformation.

* Linear
* Logarithmic
* Power – law

## Linear transformation

First we will look at the linear transformation. Linear transformation includes simple identity and negative transformation. Identity transformation has been discussed in our tutorial of image transformation, but a brief description of this transformation has been given here.

Identity transition is shown by a straight line. In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. And hence is called identity transformation. It has been shown below

## Negative transformation

The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image.

since the input image of Einstein is an 8 bpp image , so the number of levels in this image are 256. Putting 256 in the equation, we get this

s = 255 – r

So each value is subtracted by 255 and the result image has been shown above. So what happens is that , the lighter pixels become dark and the darker picture becomes light. And it results in image negative.

## Logarithmic transformations:

Logarithmic transformation further contains two type of transformation. Log transformation and inverse log transformation.

## Log transformation

The log transformations can be defined by this formula

s = c log(r + 1).

Where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added , to make the minimum value at least 1.

During log transformation , the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation.

## Power – Law transformations

There are further two transformation is power law transformations, that include nth power and nth root transformation. These transformations can be given by the expression:

s=cr^γ

This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.

Variation in the value of γ varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that’s why they display their image at different intensity.

This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5 , that means the image displayed on CRT is dark.

### CORRECTING GAMMA.

s=cr^γ

s=cr^(1/2.5)

Implementation

**Program:**

import Tkinter as tk

import Image

import ImageTk

import numpy as np

import tkFileDialog

import scipy.misc

import numpy, math

from scipy.misc.pilutil import Image

import math, numpy

import scipy.fftpack as fftim

from scipy.misc.pilutil import Image

import numpy as np

from scipy.fftpack import rfft, irfft, fftfreq

import pylab as plt

class DIP(tk.Frame):

def \_\_init\_\_(self, parent):

tk.Frame.\_\_init\_\_(self, parent)

self.parent = parent

self.initUI()

def initUI(self):

self.parent.title("Digital Image Processing Application")

self.pack(fill = tk.BOTH, expand = 1)

menubar = tk.Menu(self.parent)

self.parent.config(menu = menubar)

self.label1 = tk.Label(self, border = 25)

self.label2 = tk.Label(self, border = 25)

self.label1.grid(row = 1, column = 1)

self.label2.grid(row = 1, column = 2)

fileMenu = tk.Menu(menubar)

fileMenu.add\_command(label = "Open", command = self.onOpen)

menubar.add\_cascade(label = "File", menu = fileMenu)

basicMenu = tk.Menu(menubar)

basicMenu.add\_command(label="1DDFT",command = self.onOdft)

basicMenu.add\_command(label="2DDFT",command = self.onTdft)

basicMenu.add\_command(label="Comparision of 1DDFT,2DDFT")

basicMenu.add\_separator()

basicMenu.add\_command(label = "Negative", command = self.onNeg)

basicMenu.add\_command(label="Log Transformation",command = self.onLog)

basicMenu.add\_command(label="Power Log transformation",command = self.onPow)

basicMenu.add\_command(label="Comparision of IN,LT,PLT")

menubar.add\_cascade(label = "Basic", menu = basicMenu)

def onNeg(self):

I2 = 255-self.I;

im = Image.fromarray(np.uint8(I2))

photo2 = ImageTk.PhotoImage(im)

self.label2.configure(image = photo2)

self.label2.image = photo2 # keep a reference!

#inverted\_image = PIL.ImageOps.invert(image)

def setImage(self):

self.img = Image.open(self.fn)

self.I = np.asarray(self.img)

l, h = self.img.size

text = str(2\*l+100)+"x"+str(h+50)+"+0+0"

self.parent.geometry(text)

photo = ImageTk.PhotoImage(self.img)

self.label1.configure(image = photo)

self.label1.image = photo # keep a reference!

def onOpen(self):

ftypes = [('Image Files', '\*.tif \*.jpg \*.png')]

dlg = tkFileDialog.Open(self, filetypes = ftypes)

filename = dlg.show()

self.fn = filename

self.setImage()

def main():

root = tk.Tk()

DIP(root)

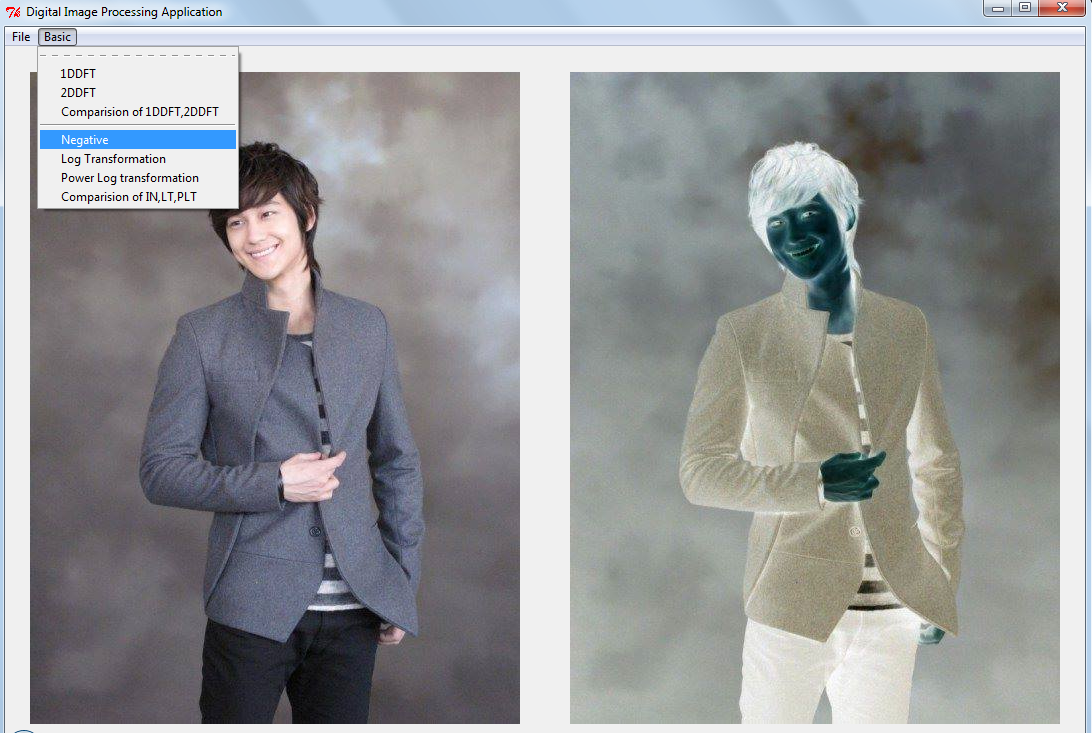
root.geometry("320x240")

root.mainloop()

if \_\_name\_\_ == '\_\_main\_\_':

main()

SCREEN SHOT:



**Conclusion:**

The output of my Case study is shows how the Fourier Transform and Image Enhancement Techniques are working based on user input image and understand the concepts.In feature we Enhace the Applications based on concepts of DIP and their applications.

References:

**Websites:**

[www.docs.python.org](http://www.docs.python.org)

[www.tutorialpoint.com](http://www.tutorialpoint.com)

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.google.com](http://www.google.com)

**Books:**

1)Image Processing and Acquisition using Python

2)Python and Tkinter Programming

*3)* Python Imaging Library (PIL)