

### Experiment No: 3

**Aim:** Write a Python program to read an image from a file system and write the negative of the image back with a different name.

#### Algorithm:

1. Read the image and store it into some container to perform operations on it
2. Get rgb value of the pixel
3. Calculate new RGB values as shown :
  1.  $R = 255 - R$
  2.  $G = 255 - G$
  3.  $B = 255 - B$
4. Save the new RGB value in the pixel
5. Repeat step 2 - 4 for each pixel of the image
6. Choose a directory to store the new image
7. Store the negative image into the selected directory

#### Program:

##### Python Code:

```
from PyQt5 import QtCore, QtGui, QtWidgets
import cv2

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(800, 450)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.imageinput = QtWidgets.QLabel(self.centralwidget)
        self.imageinput.setGeometry(QtCore.QRect(0, 30, 400, 225))
        self.imageinput.setText("")
        self.imageinput.setPixmap(QtGui.QPixmap(""))
        self.imageinput.setScaledContents(True)
        self.imageinput.setObjectName("imageinput")
        self.import_image = QtWidgets.QPushButton(self.centralwidget)
        self.import_image.setGeometry(QtCore.QRect(0, 290, 400, 81))
        self.import_image.setObjectName("import_image")
        self.output = QtWidgets.QPushButton(self.centralwidget)
        self.output.setGeometry(QtCore.QRect(400, 290, 400, 81))
        self.output.setObjectName("output")
        self.imageoutput = QtWidgets.QLabel(self.centralwidget)
        self.imageoutput.setGeometry(QtCore.QRect(401, 30, 400, 225))
```

```

self.imageoutput.setText("")
self.imageoutput.setPixmap(QtGui.QPixmap(""))
self.imageoutput.setScaledContents(True)
self.imageoutput.setObjectName("imageoutput")
self.label = QtWidgets.QLabel(self.centralwidget)
self.label.setGeometry(QtCore.QRect(150, 10, 101, 17))
self.label.setObjectName("label")
self.label_2 = QtWidgets.QLabel(self.centralwidget)
self.label_2.setGeometry(QtCore.QRect(550, 10, 101, 17))
self.label_2.setObjectName("label_2")
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 27))
self.menubar.setObjectName("menubar")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

#Importing Image
self.import_image.clicked.connect(self.show_image)

#Checking Output Image
self.output.clicked.connect(self.convert_to_negative)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "Negative of an Image"))
    self.import_image.setText(_translate("MainWindow", "Import"))
    self.output.setText(_translate("MainWindow", "Convert to Negative"))
    self.label.setText(_translate("MainWindow", "Imported Image"))
    self.label_2.setText(_translate("MainWindow", "Converted Image"))

def show_image(self):
    file_filter = 'Image File (*.jpg *.png)'
    fname = QtWidgets.QFileDialog.getOpenFileName(parent=self.centralwidget,
    caption='Select an Image',
    directory="/run/media/deeprajb/HDD/Important Photos/Wallpapers",
    filter=file_filter)
    self.img = cv2.imread(fname[0])
    self.img1 = QtGui.QImage(self.img.data, self.img.shape[1], self.img.shape[0],
    QtGui.QImage.Format_RGB888).rgbSwapped()
    self.imageinput.setPixmap(QtGui.QPixmap.fromImage(self.img1))
def convert_to_negative(self):

```

```

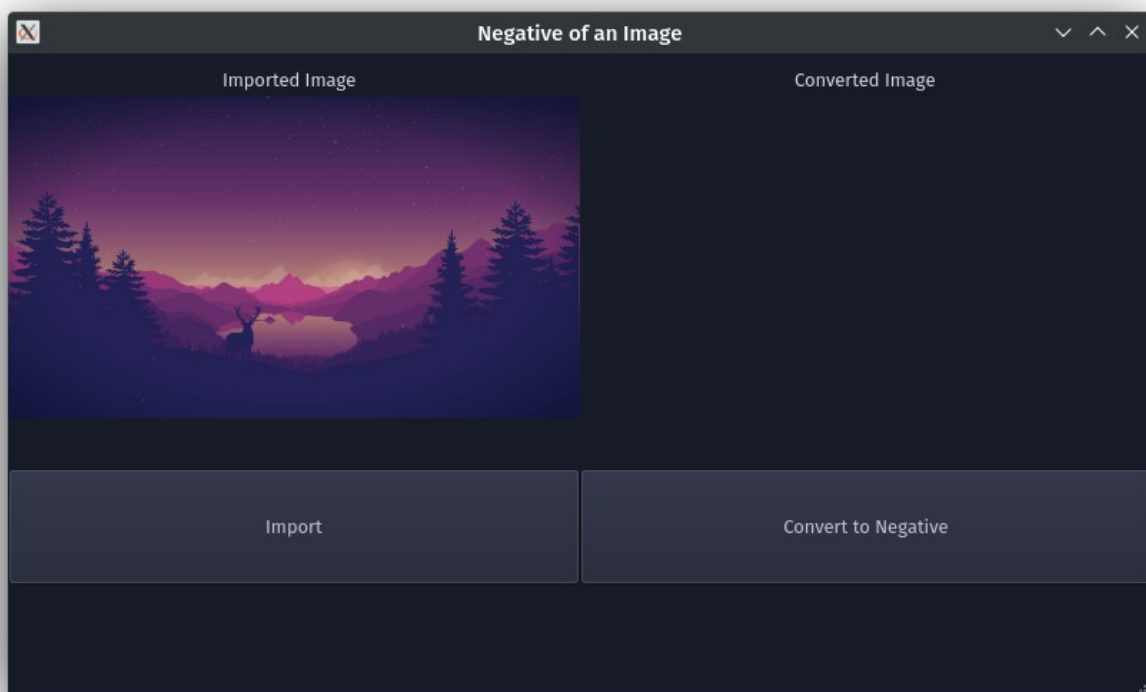
(row, col) = self.img.shape[0:2]
for i in range(0,row-1):
for j in range(0,col-1):
pixel = self.img[i, j]
# get red pixel
pixel[0] = 255 - pixel[0]
# get green pixel
pixel[1] = 255 - pixel[1]
# get blue pixel
pixel[2] = 255 - pixel[2]
self.img[i, j] = pixel
cv2.imwrite('negative_output.jpg',self.img)
self.imageoutput.setPixmap(QtGui.QPixmap("negative_output.jpg"))

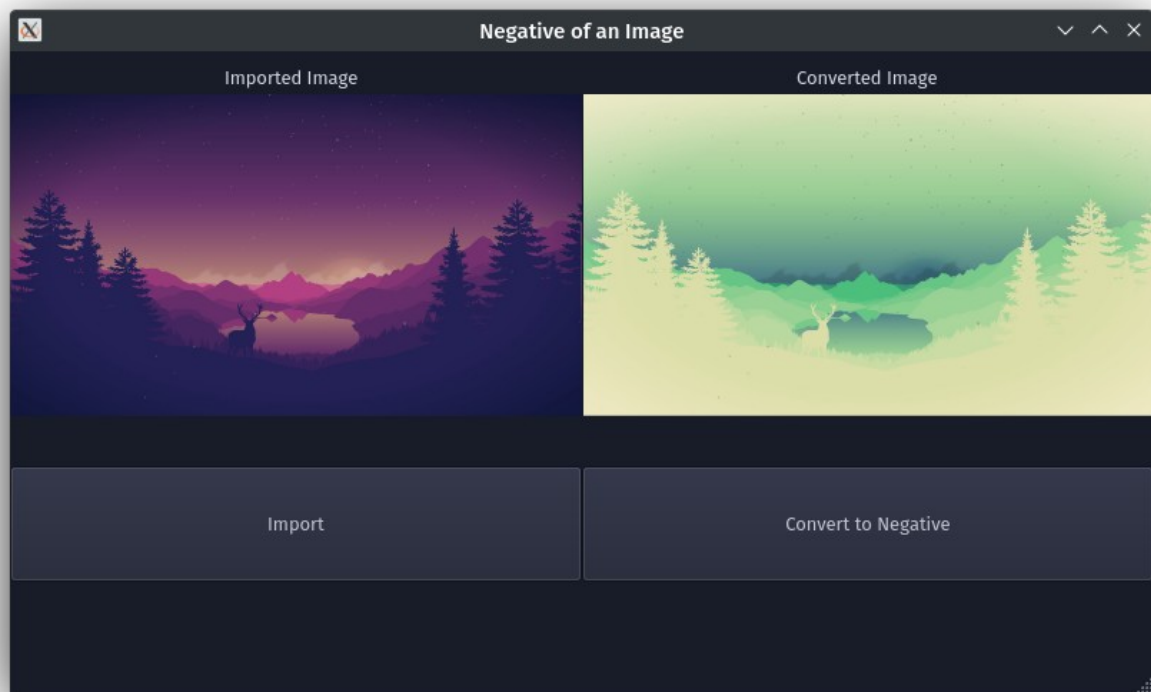
if __name__ == "__main__":
import sys
app = QtWidgets.QApplication(sys.argv)
MainWindow = QtWidgets.QMainWindow()
ui = Ui_MainWindow()
ui.setupUi(MainWindow)
MainWindow.show()
sys.exit(app.exec_())

```

**Output:**

**Python GUI Output:**





**Conclusion:** Program to read an image and convert it into negative was written and executed successfully.

**Deepraj Bhosale**  
**Batch-A**  
**181105016**