img\_proc\_qt.ui

| <?xml version="1.0" encoding="UTF-8"?>  <ui version="4.0">  <class>Dialog</class>  <widget class="QDialog" name="Dialog">  <property name="geometry">  <rect>  <x>0</x>  <y>0</y>  <width>1280</width>  <height>720</height>  </rect>  </property>  <property name="windowTitle">  <string>Dialog</string>  </property>  <widget class="QLineEdit" name="line\_edit">  <property name="geometry">  <rect>  <x>20</x>  <y>10</y>  <width>981</width>  <height>40</height>  </rect>  </property>  </widget>  <widget class="QPushButton" name="btn\_load">  <property name="geometry">  <rect>  <x>1010</x>  <y>5</y>  <width>120</width>  <height>50</height>  </rect>  </property>  <property name="text">  <string>이미지로드</string>  </property>  </widget>  <widget class="QPushButton" name="btn\_run">  <property name="geometry">  <rect>  <x>1140</x>  <y>5</y>  <width>120</width>  <height>50</height>  </rect>  </property>  <property name="text">  <string>RUN</string>  </property>  </widget>  <widget class="QLabel" name="lbl\_src">  <property name="geometry">  <rect>  <x>20</x>  <y>120</y>  <width>600</width>  <height>450</height>  </rect>  </property>  <property name="text">  <string/>  </property>  </widget>  <widget class="QLabel" name="lbl\_dst">  <property name="geometry">  <rect>  <x>650</x>  <y>120</y>  <width>600</width>  <height>450</height>  </rect>  </property>  <property name="text">  <string/>  </property>  </widget>  </widget>  <resources/>  <connections/>  </ui> |
| --- |

img\_proc\_qt.py

| from PyQt5 import QtCore, QtGui  from PyQt5 import QtWidgets, uic  from PyQt5.QtGui import QPixmap, QImage  import sys  import cv2  import numpy as np  class Ui(QtWidgets.QDialog):  def \_\_init\_\_(self):  super(Ui, self).\_\_init\_\_()  uic.loadUi('img\_proc\_qt.ui', self)  self.btn\_load = self.findChild(QtWidgets.QPushButton, 'btn\_load')  self.btn\_run = self.findChild(QtWidgets.QPushButton, 'btn\_run')  self.lbl\_src = self.findChild(QtWidgets.QLabel, 'lbl\_src')  self.lbl\_dst = self.findChild(QtWidgets.QLabel, 'lbl\_dst')  self.line\_edit = self.findChild(QtWidgets.QLineEdit, 'line\_edit')  self.show()  app = QtWidgets.QApplication(sys.argv)  window = Ui()  app.exec\_() |
| --- |

함수연결

| from PyQt5 import QtCore, QtGui  from PyQt5 import QtWidgets, uic  from PyQt5.QtGui import QPixmap, QImage  import sys  import cv2  import numpy as np  class Ui(QtWidgets.QDialog):  def \_\_init\_\_(self):  super(Ui, self).\_\_init\_\_()  uic.loadUi('img\_proc\_qt.ui', self)  self.btn\_load = self.findChild(QtWidgets.QPushButton, 'btn\_load')  self.btn\_load.clicked.connect(self.btn\_load\_clicked)  self.btn\_run = self.findChild(QtWidgets.QPushButton, 'btn\_run')  self.btn\_run.clicked.connect(self.btn\_run\_clicked)  self.lbl\_src = self.findChild(QtWidgets.QLabel, 'lbl\_src')  self.lbl\_dst = self.findChild(QtWidgets.QLabel, 'lbl\_dst')  self.line\_edit = self.findChild(QtWidgets.QLineEdit, 'line\_edit')  self.line\_edit.clear()  self.show()  def btn\_load\_clicked(self):  self.line\_edit.setText("로드버튼 클릭")  def btn\_run\_clicked(self):  self.line\_edit.setText("런버튼 클릭")  app = QtWidgets.QApplication(sys.argv)  window = Ui()  app.exec\_() |
| --- |

def\_btn\_load\_clicked() 업데이트

| from PyQt5 import QtCore, QtGui  from PyQt5 import QtWidgets, uic  from PyQt5.QtGui import QPixmap, QImage  import sys  import cv2  import numpy as np  class Ui(QtWidgets.QDialog):  def \_\_init\_\_(self):  super(Ui, self).\_\_init\_\_()  uic.loadUi('img\_proc\_qt.ui', self)  self.btn\_load = self.findChild(QtWidgets.QPushButton, 'btn\_load')  self.btn\_load.clicked.connect(self.btn\_load\_clicked)  self.btn\_run = self.findChild(QtWidgets.QPushButton, 'btn\_run')  self.btn\_run.clicked.connect(self.btn\_run\_clicked)  self.lbl\_src = self.findChild(QtWidgets.QLabel, 'lbl\_src')  self.lbl\_dst = self.findChild(QtWidgets.QLabel, 'lbl\_dst')  self.line\_edit = self.findChild(QtWidgets.QLineEdit, 'line\_edit')  self.line\_edit.clear()  self.show()  def btn\_load\_clicked(self):  path = 'images'  filter = "All Images(\*.jpg; \*.png; \*.bmp);;JPG (\*.jpg);;PNG(\*.png);;BMP(\*.bmp)"  fname = QtWidgets.QFileDialog.getOpenFileName(self, "파일로드", path, filter)  filename = str(fname[0])  self.line\_edit.setText(filename)  def btn\_run\_clicked(self):  self.line\_edit.setText("런버튼 클릭")  app = QtWidgets.QApplication(sys.argv)  window = Ui()  app.exec\_() |
| --- |

이미지를 선택하면 이미지가 화면 왼쪽에 출력됨

| from PyQt5 import QtCore, QtGui  from PyQt5 import QtWidgets, uic  from PyQt5.QtGui import QPixmap, QImage  import sys  import cv2  import numpy as np  class Ui(QtWidgets.QDialog):  def \_\_init\_\_(self):  super(Ui, self).\_\_init\_\_()  uic.loadUi('img\_proc\_qt.ui', self)  self.btn\_load = self.findChild(QtWidgets.QPushButton, 'btn\_load')  self.btn\_load.clicked.connect(self.btn\_load\_clicked)  self.btn\_run = self.findChild(QtWidgets.QPushButton, 'btn\_run')  self.btn\_run.clicked.connect(self.btn\_run\_clicked)  self.lbl\_src = self.findChild(QtWidgets.QLabel, 'lbl\_src')  self.lbl\_dst = self.findChild(QtWidgets.QLabel, 'lbl\_dst')  self.line\_edit = self.findChild(QtWidgets.QLineEdit, 'line\_edit')  self.line\_edit.clear()  self.show()  #cv2.imread가 한글 지원하지 않으므로 새로운 방식으로 파일 조합  def imread(self, filename, flags=cv2.IMREAD\_COLOR, dtype=np.uint8):  try:  n = np.fromfile(filename, dtype)  img = cv2.imdecode(n, flags)  return img  except Exception as e:  print(e)  return None  def btn\_load\_clicked(self):  path = 'images'  filter = "All Images(\*.jpg; \*.png; \*.bmp);;JPG (\*.jpg);;PNG(\*.png);;BMP(\*.bmp)"  fname = QtWidgets.QFileDialog.getOpenFileName(self, "파일로드", path, filter)  filename = str(fname[0])  self.line\_edit.setText(filename)  self.img\_src = self.imread(filename)  self.display\_output\_image(self.img\_src, 0)  def btn\_run\_clicked(self):  self.line\_edit.setText("런버튼 클릭")  def display\_output\_image(self, img, mode):  h, w = img.shape[:2] # 그레이영상은 ndim이 2이므로 h,w,ch 형태로 값을 얻어올수 없다  if img.ndim == 2:  qImg = QImage(img, w, h, w \* 1, QImage.Format\_Grayscale8)  else:  bytes\_per\_line = img.shape[2] \* w  qImg = QImage(img, w, h, bytes\_per\_line, QImage.Format\_BGR888)  pixmap = QtGui.QPixmap(qImg)  pixmap = pixmap.scaled(600, 450, QtCore.Qt.KeepAspectRatio) # 이미지 비율유지  #pixmap = pixmap.scaled(600, 450, QtCore.Qt.IgnoreAspectRatio) # 프레임에 맞춤  if mode == 0:  self.lbl\_src.setPixmap(pixmap)  self.lbl\_src.update() # 프레임 띄우기  else:  self.lbl\_dst.setPixmap(pixmap)  self.lbl\_dst.update() # 프레임 띄우기  app = QtWidgets.QApplication(sys.argv)  window = Ui()  app.exec\_() |
| --- |

## 이진화

| import cv2  img\_src = cv2.imread('images/animal-05.jpg', cv2.IMREAD\_COLOR)  img\_src = cv2.resize(img\_src, None, fx=0.8, fy=0.8)  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  max\_value = 255  thr = 160  # 이진화  # THRESH\_BINARY : img\_dst = (img\_src > thr) ? max\_value : 0  ret, img\_dst1 = cv2.threshold(img\_gray, thr, max\_value, cv2.THRESH\_BINARY)  # THRESH\_BINARY\_INV : img\_dst = (img\_src > thr) ? 0 : max\_value  ret, img\_dst2 = cv2.threshold(img\_gray, thr, max\_value, cv2.THRESH\_BINARY\_INV)  # THRESH\_TRUNC : img\_dst = (img\_src > thr) ? thr : img\_src  ret, img\_dst3 = cv2.threshold(img\_gray, thr, max\_value, cv2.THRESH\_TRUNC)  # THRESH\_TOZERO : img\_dst = (img\_src > thr) ? img\_src : 0  ret, img\_dst4 = cv2.threshold(img\_gray, thr, max\_value, cv2.THRESH\_TOZERO)  # THRESH\_TOZERO\_INV : img\_dst = (img\_src > thr) ? 0 : img\_src  ret, img\_dst5 = cv2.threshold(img\_gray, thr, max\_value, cv2.THRESH\_TOZERO\_INV)  ret, img\_dst6 = cv2.threshold(img\_gray, thr, max\_value, cv2.THRESH\_OTSU)  img\_dst\_up = cv2.hconcat([img\_dst1,img\_dst2,img\_dst3])  img\_dst\_dn = cv2.hconcat([img\_dst4,img\_dst5,img\_dst6])  img\_dst = cv2.vconcat([img\_dst\_up,img\_dst\_dn])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

에지검출 : sobel

ref : <https://docs.opencv.org/4.5.1/d2/d2c/tutorial_sobel_derivatives.html>

| import cv2  img\_src = cv2.imread('images/animal-08.jpg', cv2.IMREAD\_COLOR)  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  img\_sobel\_x = cv2.Sobel(img\_gray, cv2.CV\_64F, 1,0,ksize=3)  img\_sobel\_x = cv2.convertScaleAbs(img\_sobel\_x)  cv2.imshow('sobel x', img\_sobel\_x)  img\_sobel\_y = cv2.Sobel(img\_gray, cv2.CV\_64F, 0,1,ksize=3)  img\_sobel\_y = cv2.convertScaleAbs(img\_sobel\_y)  cv2.imshow('sobel y', img\_sobel\_y)  img\_sobel = cv2.addWeighted(img\_sobel\_x,1.0, img\_sobel\_y, 1.0, 0)  img\_dst = cv2.hconcat([img\_gray,img\_sobel])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

라플라시안 에지

ref : <https://docs.opencv.org/master/d5/db5/tutorial_laplace_operator.html>

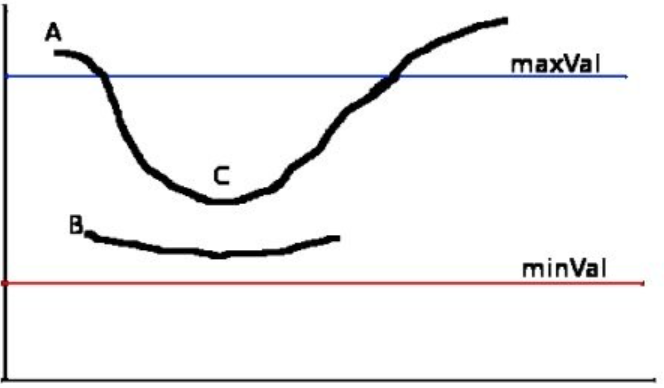
| import cv2  img\_src = cv2.imread('images/animal-08.jpg', cv2.IMREAD\_COLOR)  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  img\_laplacian = cv2.Laplacian(img\_gray, cv2.CV\_16S, ksize=3)  img\_laplacian = cv2.convertScaleAbs(img\_laplacian)  img\_dst = cv2.hconcat([img\_gray,img\_laplacian])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

소벨 라플라시안 비교

| import cv2  img\_src = cv2.imread('images/animal-08.jpg', cv2.IMREAD\_COLOR)  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  img\_sobel\_x = cv2.Sobel(img\_gray, cv2.CV\_64F, 1,0,ksize=3)  img\_sobel\_x = cv2.convertScaleAbs(img\_sobel\_x)  img\_sobel\_y = cv2.Sobel(img\_gray, cv2.CV\_64F, 0,1,ksize=3)  img\_sobel\_y = cv2.convertScaleAbs(img\_sobel\_y)  img\_sobel = cv2.addWeighted(img\_sobel\_x,1.0, img\_sobel\_y, 1.0, 0)  img\_laplacian = cv2.Laplacian(img\_gray, cv2.CV\_16S, ksize=3)  img\_laplacian = cv2.convertScaleAbs(img\_laplacian)  img\_dst = cv2.hconcat([img\_gray,img\_sobel,img\_laplacian])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

케니(Canny) 에지

ref : <https://docs.opencv.org/3.4/da/d5c/tutorial_canny_detector.html>



| import cv2  img\_src = cv2.imread('images/animal-08.jpg', cv2.IMREAD\_COLOR)  img\_gray = cv2.cvtColor(img\_src,cv2.COLOR\_BGR2GRAY)  img\_gray = cv2.blur(img\_gray, (3, 3), anchor=(-1,-1),borderType=cv2.BORDER\_DEFAULT)  img\_canny = cv2.Canny(img\_gray, 50, 150)  img\_dst = cv2.hconcat([img\_gray,img\_canny])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

opencv documents 코드

| from \_\_future\_\_ import print\_function  import cv2 as cv  import argparse  max\_lowThreshold = 100  window\_name = 'Edge Map'  title\_trackbar = 'Min Threshold:'  ratio = 3  kernel\_size = 3  def CannyThreshold(val):  low\_threshold = val  img\_blur = cv.blur(src\_gray, (3,3))  detected\_edges = cv.Canny(img\_blur, low\_threshold, low\_threshold\*ratio, kernel\_size)  mask = detected\_edges != 0  dst = src \* (mask[:,:,None].astype(src.dtype))  cv.imshow(window\_name, dst)  parser = argparse.ArgumentParser(description='Code for Canny Edge Detector tutorial.')  parser.add\_argument('--input', help='Path to input image.', default='images/animal-01.jpg')  args = parser.parse\_args()  src = cv.imread(cv.samples.findFile(args.input))  if src is None:  print('Could not open or find the image: ', args.input)  exit(0)  src\_gray = cv.cvtColor(src, cv.COLOR\_BGR2GRAY)  cv.namedWindow(window\_name)  cv.createTrackbar(title\_trackbar, window\_name, 0, max\_lowThreshold, CannyThreshold)  CannyThreshold(0)  cv.waitKey(0) |
| --- |