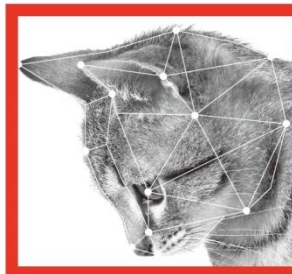


# 컴퓨터 비전과 딥러닝

## [강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 처벌을 받을 수 있습니다.

COMPUTER VISION



DEEP  
LEARNING

# 컴퓨터 비전과 딥러닝

## Chapter 06 비전 에이전트

# 차례

6.1 지능 에이전트로서 비전 에이전트

6.2 PyQt를 이용한 사용자 인터페이스

6.3 [비전 에이전트 1] 오림

6.4 [비전 에이전트 2] 교통약자 보호구역 알림

6.5 [비전 에이전트 3] 파노라마 영상 제작

6.6 [비전 에이전트 4] 특수 효과

# Preview

## ■ 컴퓨터 비전이 쓸모 있으려면 환경과 상호작용 필요

- 환경에서 영상을 획득하는 기능과 환경에 영향을 미치는 기능 추가

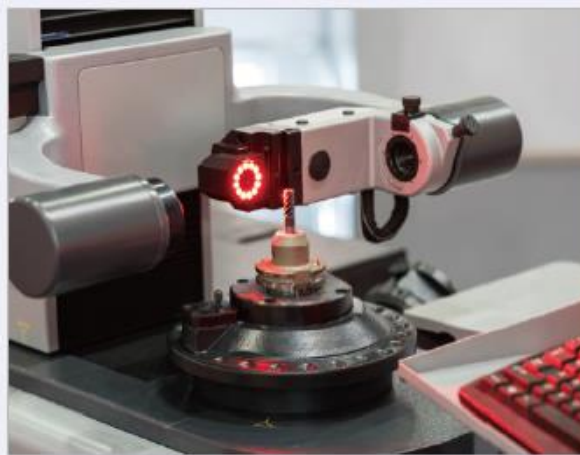


그림 6-1 환경과 상호작용하는 컴퓨터 비전 시스템

## ■ 이 장은 PyQt 라이브러리를 사용한 사용자 인터페이스 실현

- 컴퓨터 비전 분야 프로그래밍 개발 능력 향상에 효과적

## 6.1 지능 에이전트로서 비전 에이전트

### ■ 지능 에이전트

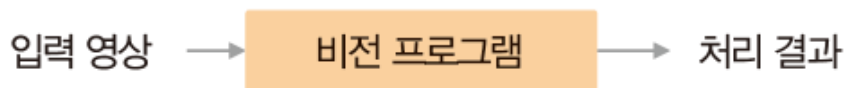
- 『Artificial Intelligence: A Modern Approach(4<sup>th</sup> Edition)』의 정의

anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators

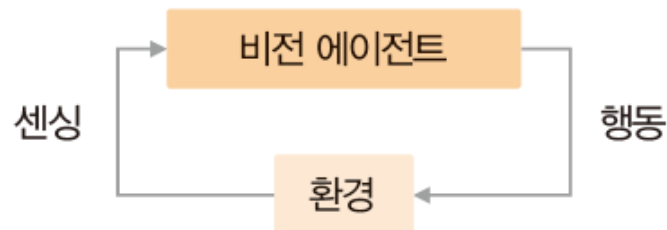
센서를 통해 환경을 지각하고 액츄에이터를 통해 환경에 행동을 가한다고 볼 수 있는 모든 것

- 위 정의에서 sensors를 vision sensors로 바꾸면 비전 에이전트

### ■ 비전 프로그램을 비전 에이전트로 확장



(a) 비전 프로그램



(b) 비전 에이전트

그림 6-2 환경과 상호작용하는 비전 에이전트

## 6.2 PyQt를 이용한 사용자 인터페이스

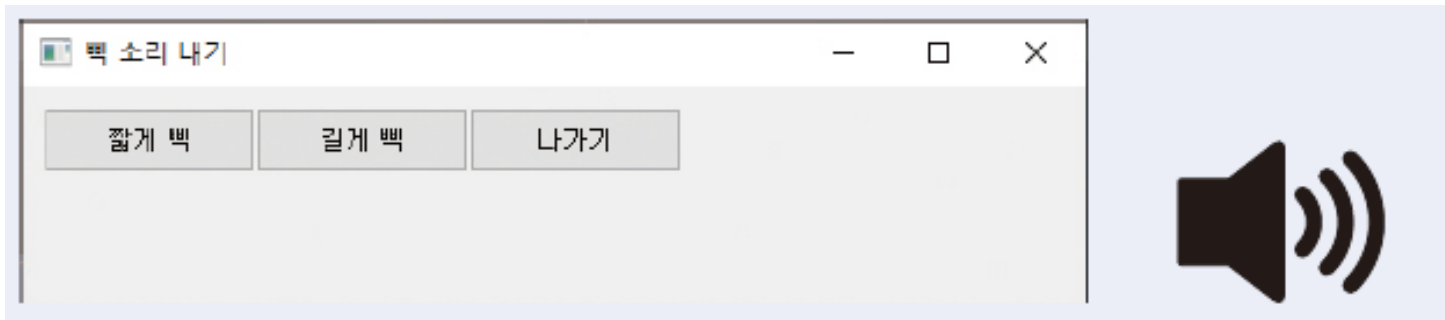
---

- 비전 에이전트는 그래픽 사용자 인터페이스(GUI) 필요
  - 이 책은 PyQt 라이브러리로 GUI 구현

# PyQt 기초 프로그래밍

## ■ 아래 사용자 인터페이스 구현

- <짧게 뽕> 버튼은 0.5초 동안 뽕 소리, <길게 뽕> 버튼은 3초 동안 뽕
- <나가기> 버튼은 윈도우를 닫고 프로그램 종료



# PyQt 기초 프로그래밍

프로그램 6-1

PyQt로 간단한 GUI 만들기(버튼을 클릭하면 뽁 소리 들려주기)

```
01 from PyQt5.QtWidgets import *
02 import sys
03 import winsound
04
05 class BeepSound(QMainWindow):
06     def __init__(self):
07         super().__init__()
08         self.setWindowTitle('뽁 소리 내기')
09         self.setGeometry(200,200,500,100)
10
11         shortBeepButton=QPushButton('짧게 뽁',self)
12         longBeepButton=QPushButton('길게 뽁',self)
13         quitButton=QPushButton('나가기',self)
14         self.label=QLabel('환영합니다!',self)
15
16         shortBeepButton.setGeometry(10,10,100,30)
17         longBeepButton.setGeometry(110,10,100,30)
18         quitButton.setGeometry(210,10,100,30)
19         self.label.setGeometry(10,40,500,70)
20
21         shortBeepButton.clicked.connect(self.shortBeepFunction)
22         longBeepButton.clicked.connect(self.longBeepFunction)
23         quitButton.clicked.connect(self.quitFunction)
```

BeepSound 클래스 선언(클래스에 대한 설명은 부록 A.2 참조)

생성자 함수

# 윈도우 이름과 위치 지정

# 버튼 생성

# 버튼 위치와 크기 지정

# 콜백 함수 지정

ShortBeepButton  
버튼이 눌리면  
ShortBeepFunction 실행



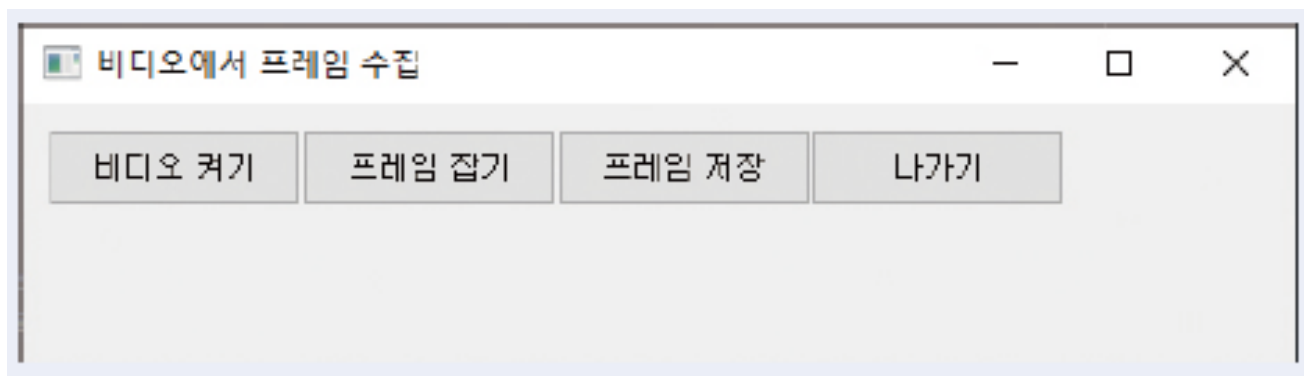
# PyQt 기초 프로그래밍

```
24
25     def shortBeepFunction(self):
26         self.label.setText('주파수 1000으로 0.5초 동안 뽕 소리를 냅니다.')
27         winsound.Beep(1000,500)
28
29     def longBeepFunction(self):
30         self.label.setText('주파수 1000으로 3초 동안 뽕 소리를 냅니다.')
31         winsound.Beep(1000,3000)
32
33     def quitFunction(self):
34         self.close()
35
36 app=QApplication(sys.argv)
37 win=BeepSound() ← BeepSound 클래스의 객체 생성
38 win.show()
39 app.exec_()
```

# OpenCV에 PyQt를 붙여 프로그램 확장하기

## ■ [프로그램 6-2]

- 비디오에서 프레임을 획득하고 저장하는 간단한 기능 제공
- [프로그램 6-1]과 비슷한 구조 → 프로그램 구조에 대한 눈썰미 필요



# OpenCV에 PyQt를 붙여 프로그램 확장하기

프로그램 6-2

OpenCV에 PyQt의 GUI 붙이기(비디오에서 프레임을 잡아 저장하기)

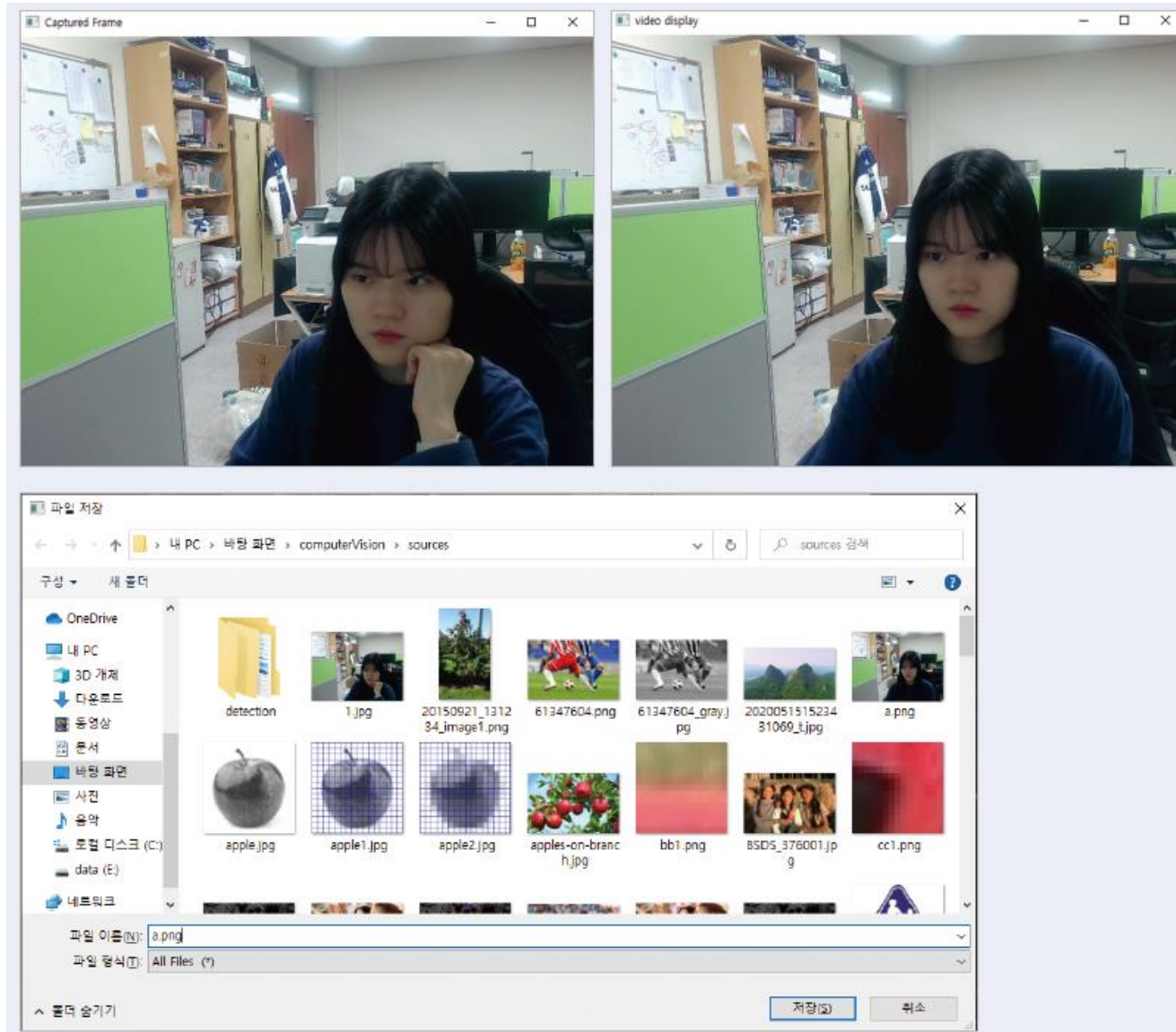
```
01 from PyQt5.QtWidgets import *
02 import sys
03 import cv2 as cv
04
05 class Video(QMainWindow):
06     def __init__(self):
07         super().__init__()
08         self.setWindowTitle('비디오에서 프레임 수집') # 윈도우 이름과 위치 지정
09         self.setGeometry(200,200,500,100)
10
11         videoButton=QPushButton('비디오 켜기',self) # 버튼 생성
12         captureButton=QPushButton('프레임 잡기',self)
13         saveButton=QPushButton('프레임 저장',self)
14         quitButton=QPushButton('나가기',self)
15
16         videoButton.setGeometry(10,10,100,30) # 버튼 위치와 크기 지정
17         captureButton.setGeometry(110,10,100,30)
18         saveButton.setGeometry(210,10,100,30)
19         quitButton.setGeometry(310,10,100,30)
20
21         videoButton.clicked.connect(self.videoFunction) # 콜백 함수 지정
22         captureButton.clicked.connect(self.captureFunction)
23         saveButton.clicked.connect(self.saveFunction)
24         quitButton.clicked.connect(self.quitFunction)
```

Video 클래스 선언

# OpenCV에 PyQt를 붙여 프로그램 확장하기

```
25
26     def videoFunction(self):
27         self.cap=cv.VideoCapture(0,cv.CAP_DSHOW)           # 카메라와 연결 시도
28         if not self.cap.isOpened(): self.close()
29
30         while True:
31             ret,self.frame=self.cap.read()
32             if not ret: break
33             cv.imshow('video display',self.frame)
34             cv.waitKey(1)
35
36     def captureFunction(self):
37         self.capturedFrame=self.frame
38         cv.imshow('Captured Frame',self.capturedFrame)
39
40     def saveFunction(self):                                   # 파일 저장
41         fname=QFileDialog.getSaveFileName(self,'파일 저장','./')
42         cv.imwrite(fname[0],self.capturedFrame)
43
44     def quitFunction(self):
45         self.cap.release()                                   # 카메라와 연결을 끊음
46         cv.destroyAllWindows()
47         self.close()
48
49 app=QApplication(sys.argv)
50 win=Video()
51 win.show()
52 app.exec_()
```

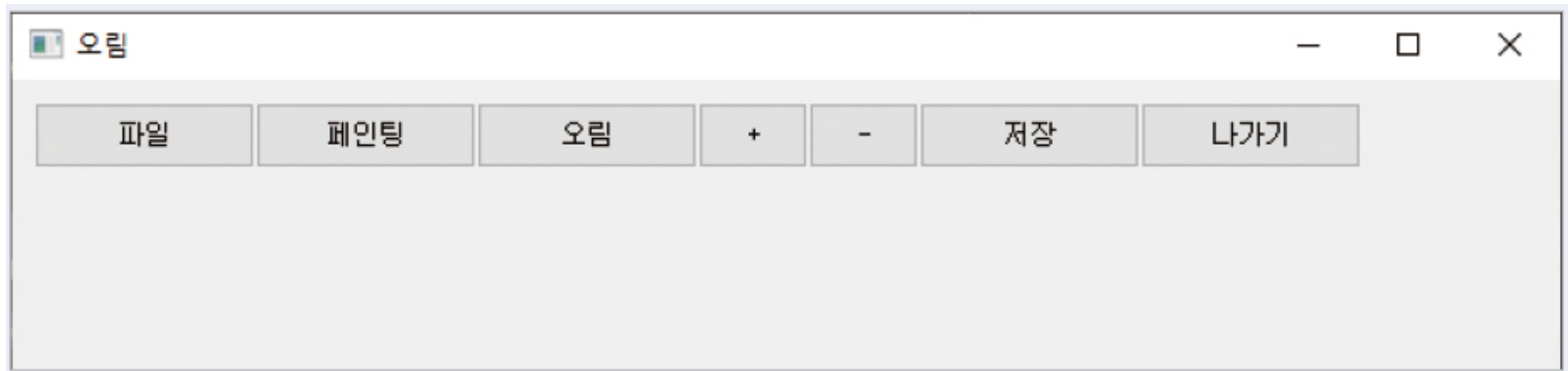
# OpenCV에 PyQt를 붙여 프로그램 확장하기



## 6.3 [비전 에이전트 1] 오림

### ■ GrabCut(4.5절 대화식 영상 분할)을 이용한 관심 물체 오림

- 붓칠과 오림을 반복하여 만족할 때까지 분할하는 기능 제공
- 마우스 왼쪽 버튼으로 물체에 파란 칠, 오른쪽 버튼으로 배경에 빨간 칠



## 6.3 [비전 에이전트 1] 오림

프로그램 6-3

GrabCut을 이용해 관심 물체 오리기

```
01 import cv2 as cv
02 import numpy as np
03 import sys
04 from PyQt5.QtWidgets import *
05
06 class Orim(QMainWindow):
07     def __init__(self) :
08         super().__init__()
09         self.setWindowTitle('오림')
10         self.setGeometry(200,200,700,200)
11
12         fileButton=QPushButton('파일',self)
13         paintButton=QPushButton('페인팅',self)
14         cutButton=QPushButton('오림',self)
15         incButton=QPushButton('+',self)
16         decButton=QPushButton('-',self)
17         saveButton=QPushButton('저장',self)
18         quitButton=QPushButton('나가기',self)
19
```

## 6.3 [비전 에이전트 1] 오림

```
20     fileButton.setGeometry(10,10,100,30)
21     paintButton.setGeometry(110,10,100,30)
22     cutButton.setGeometry(210,10,100,30)
23     incButton.setGeometry(310,10,50,30)
24     decButton.setGeometry(360,10,50,30)
25     saveButton.setGeometry(410,10,100,30)
26     quitButton.setGeometry(510,10,100,30)
27
28     fileButton.clicked.connect(self.fileOpenFunction)
29     paintButton.clicked.connect(self.paintFunction)
30     cutButton.clicked.connect(self.cutFunction)
31     incButton.clicked.connect(self.incFunction)
32     decButton.clicked.connect(self.decFunction)
33     saveButton.clicked.connect(self.saveFunction)
34     quitButton.clicked.connect(self.quitFunction)
35
36     self.BrushSiz=5                                # 페인팅 붓의 크기
37     self.LColor,self.RColor=(255,0,0),(0,0,255)    # 파란색 물체, 빨간색 배경
38
```





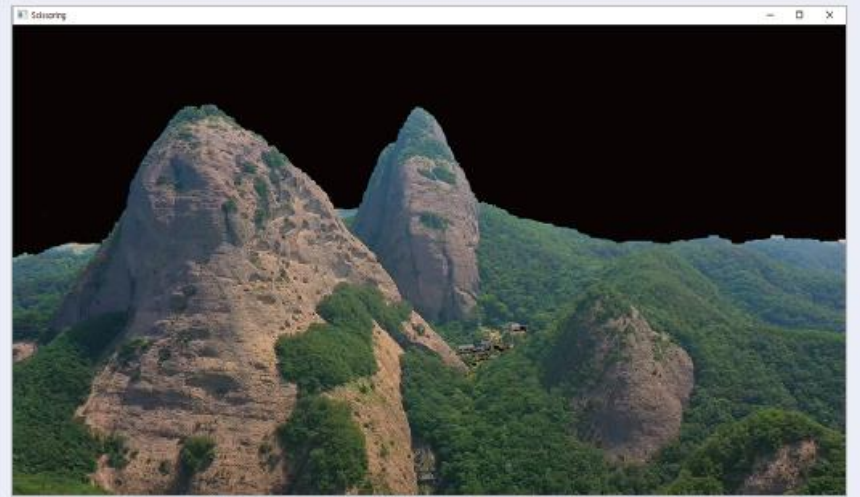
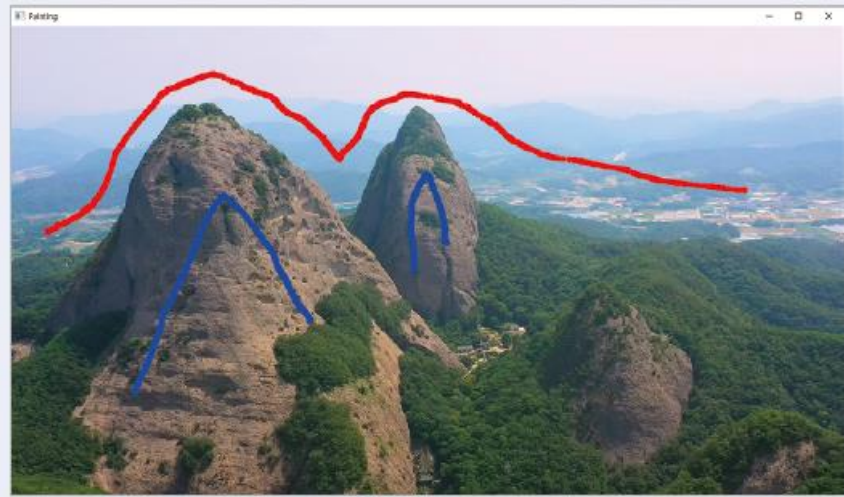
## 6.3 [비전 에이전트 1] 오림

```
56         cv.circle(self.mask,(x,y),self.BrushSiz,cv.GC_FGD,-1)
57     elif event==cv.EVENT_RBUTTONDOWN:
58         cv.circle(self.img_show,(x,y),self.BrushSiz,self.RColor,-1)
59                                     # 오른쪽 버튼을 클릭하면 빨간색
60         cv.circle(self.mask,(x,y),self.BrushSiz,cv.GC_BGD,-1)
61     elif event==cv.EVENT_MOUSEMOVE and flags==cv.EVENT_FLAG_LBUTTON:
62         cv.circle(self.img_show,(x,y),self.BrushSiz,self.LColor,-1)
63                                     # 왼쪽 버튼을 클릭하고 이동하면 파란색
64         cv.circle(self.mask,(x,y),self.BrushSiz,cv.GC_FGD,-1)
65     elif event==cv.EVENT_MOUSEMOVE and flags==cv.EVENT_FLAG_RBUTTON:
66         cv.circle(self.img_show,(x,y),self.BrushSiz,self.RColor,-1)
67                                     # 오른쪽 버튼을 클릭하고 이동하면 빨간색
68         cv.circle(self.mask,(x,y),self.BrushSiz,cv.GC_BGD,-1)
69
70     cv.imshow('Painting',self.img_show)
71
72     def cutFunction(self):
73         background=np.zeros((1,65),np.float64)
74         foreground=np.zeros((1,65),np.float64)
75         cv.grabCut(self.img,self.mask,None,background,foreground,5,cv.GC_INIT_
76             WITH_MASK)
77         mask2=np.where((self.mask==2)|(self.mask==0),0,1).astype('uint8')
78         self.grabImg=self.img*mask2[:, :, np.newaxis]
79         cv.imshow('Scissoring',self.grabImg)
```

## 6.3 [비전 에이전트 1] 오림

```
76
77     def incFunction(self):
78         self.BrushSiz=min(20,self.BrushSiz+1)
79
80     def decFunction(self):
81         self.BrushSiz=max(1,self.BrushSiz-1)
82
83     def saveFunction(self):
84         fname=QFileDialog.getSaveFileName(self,'파일 저장','./')
85         cv.imwrite(fname[0],self.grabImg)
86
87     def quitFunction(self):
88         cv.destroyAllWindows()
89         self.close()
90
91 app=QApplication(sys.argv)
92 win=Orim()
93 win.show()
94 app.exec_()
```

## 6.3 [비전 에이전트 1] 오림



## 6.4 [비전 에이전트 2] 교통약자 보호구역 알림

### ■ 교통약자에 대한 보호정책이 점점 강화

- 컴퓨터 비전 기술로 교통표지판 인식하여 운전자에게 알려 사고 위험 줄임
- 여기서는 SIFT를 활용



## 6.4 [비전 에이전트 2] 교통약자 보호구역 알림

프로그램 6-4

교통약자 보호구역 알림 구현하기

```
01  import cv2 as cv
02  import numpy as np
03  from PyQt5.QtWidgets import *
04  import sys
05  import winsound
06
07  class TrafficWeak(QMainWindow):
08      def __init__(self):
09          super().__init__()
10          self.setWindowTitle('교통약자 보호')
11          self.setGeometry(200,200,700,200)
12
13          signButton=QPushButton('표지판 등록',self)
14          roadButton=QPushButton('도로 영상 불러옴',self)
15          recognitionButton=QPushButton('인식',self)
16          quitButton=QPushButton('나가기',self)
17          self.label=QLabel('환영합니다!',self)
```



## 6.4 [비전 에이전트 2] 교통약자 보호구역 알림

```
18
19     signButton.setGeometry(10,10,100,30)
20     roadButton.setGeometry(110,10,100,30)
21     recognitionButton.setGeometry(210,10,100,30)
22     quitButton.setGeometry(510,10,100,30)
23     self.label.setGeometry(10,40,600,170)
24
25     signButton.clicked.connect(self.signFunction)
26     roadButton.clicked.connect(self.roadFunction)
27     recognitionButton.clicked.connect(self.recognitionFunction)
28     quitButton.clicked.connect(self.quitFunction)
29
30     self.signFiles=[['child.png','어린이'],['elder.png','노인'],['disabled.
31     png','장애인']]                # 표지판 모델 영상
32     self.signImgs=[]                # 표지판 모델 영상 저장
33
34     def signFunction(self):
35         self.label.clear()
36         self.label.setText('교통약자 표지판을 등록합니다.')
37
38         for fname,_ in self.signFiles:
39             self.signImgs.append(cv.imread(fname))
40             cv.imshow(fname,self.signImgs[-1])
```

## 6.4 [비전 에이전트 2] 교통약자 보호구역 알림

```
40
41     def roadFunction(self):
42         if self.signImgs==[]:
43             self.label.setText('먼저 표지판을 등록하세요.')
44         else:
45             fname=QFileDialog.getOpenFileName(self,'파일 읽기','./')
46             self.roadImg=cv.imread(fname[0])
47             if self.roadImg is None: sys.exit('파일을 찾을 수 없습니다.')
48
49             cv.imshow('Road scene',self.roadImg)
50
51     def recognitionFunction(self):
52         if self.roadImg is None:
53             self.label.setText('먼저 도로 영상을 입력하세요.')
54         else:
55             sift=cv.SIFT_create()
56
57             KD=[]                                # 여러 표지판 영상의 키폰트와 기술자 저장
58             for img in self.signImgs:
59                 gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)
```



## 6.4 [비전 에이전트 2] 교통약자 보호구역 알림

```
60         KD.append(sift.detectAndCompute(gray,None))
61
62     grayRoad=cv.cvtColor(self.roadImg,cv.COLOR_BGR2GRAY) # 명암으로 변환
63     road_kp,road_des=sift.detectAndCompute(grayRoad,None)
64                                     # 키포인트와 기술자 추출
65
66     matcher=cv.DescriptorMatcher_create(cv.DescriptorMatcher_FLANNBASED)
67     GM=[] # 여러 표지판 영상의 good match를 저장
68     for sign_kp,sign_des in KD:
69         knn_match=matcher.knnMatch(sign_des,road_des,2)
70         T=0.7
71         good_match=[]
72         for nearest1,nearest2 in knn_match:
73             if (nearest1.distance/nearest2.distance)<T:
74                 good_match.append(nearest1)
75         GM.append(good_match)
76
77     best=GM.index(max(GM,key=len)) # 매칭 쌍 개수가 최대인 표지판 찾기
```

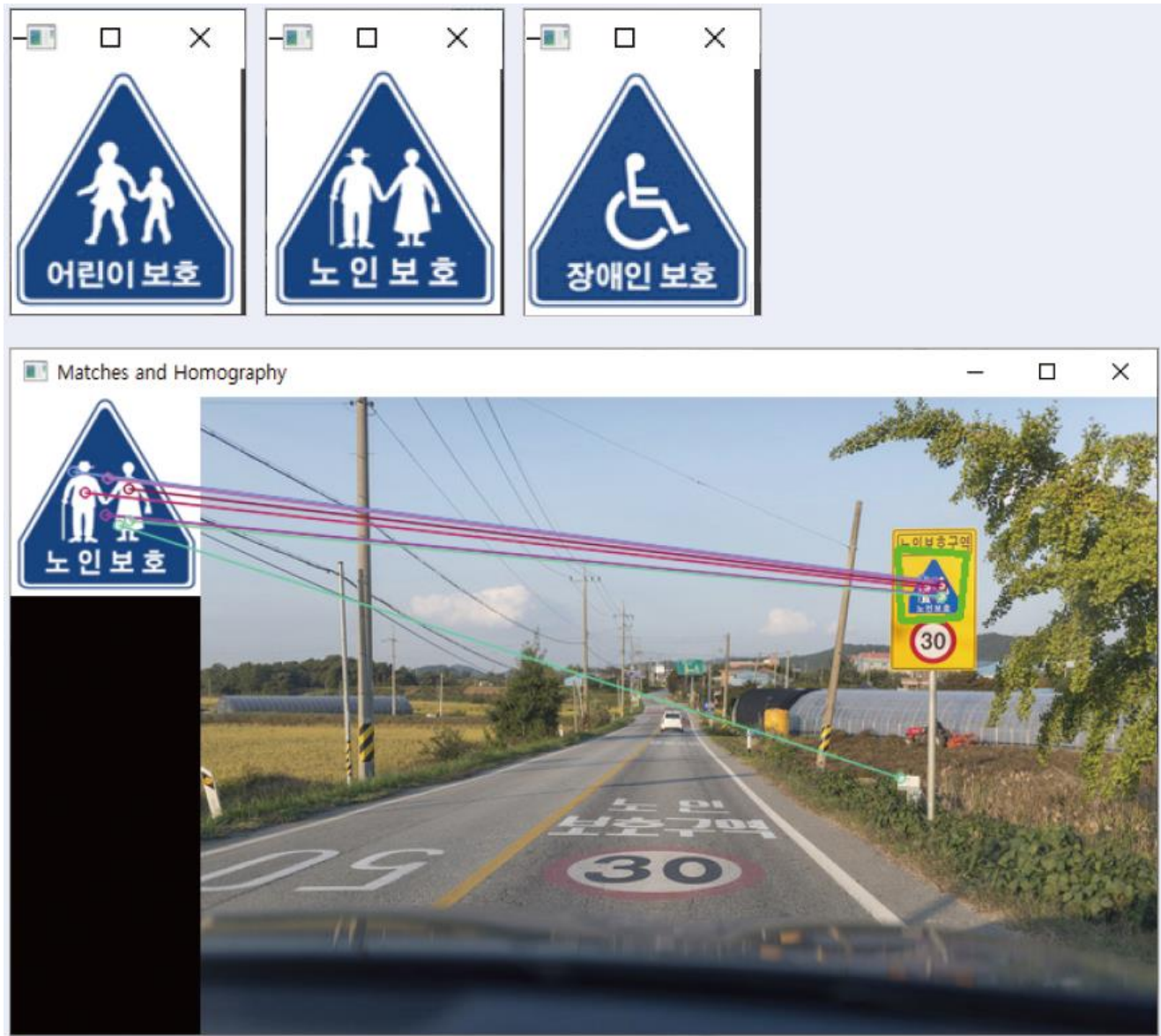
## 6.4 [비전 에이전트 2] 교통약자 보호구역 알림

```
78         if len(GM[best])<4:                # 최선의 표지판이 매칭 쌍 4개 미만이면 실패
79             self.label.setText('표지판이 없습니다.')
80         else:                               # 성공(호모그래피 찾아 영상에 표시)
81             sign_kp=KD[best][0]
82             good_match=GM[best]
83
84             points1=np.float32([sign_kp[gm.queryIdx].pt for gm in good_match])
85             points2=np.float32([road_kp[gm.trainIdx].pt for gm in good_match])
86
87             H_=cv.findHomography(points1,points2,cv.RANSAC)
88
89             h1,w1=self.signImgs[best].shape[0],self.signImgs[best].shape[1]
90                                     # 표지판 영상의 크기
91             h2,w2=self.roadImg.shape[0],self.roadImg.shape[1] # 도로 영상의 크기
92
93             box1=np.float32([[0,0],[0,h1-1],[w1-1,h1-1],[w1-1,0]]).reshape(4,1,2)
94             box2=cv.perspectiveTransform(box1,H)
95
96             self.roadImg=cv.polylines(self.roadImg,[np.int32(box2)],True,
97                                     (0,255,0),4)
98
99             img_match=np.empty((max(h1,h2),w1+w2,3),dtype=np.uint8)
100             cv.drawMatches(self.signImgs[best],sign_kp,self.roadImg,road_
101                             kp,good_match,img_match,flags=cv.DrawMatchesFlags_NOT_DRAW_
102                             SINGLE_POINTS)
```

## 6.4 [비전 에이전트 2] 교통약자 보호구역 알림

```
99         cv.imshow('Matches and Homography',img_match)
100
101         self.label.setText(self.signFiles[best][1]+ '보호구역입니다. 30km
102         로 서행하세요.')
103         winsound.Beep(3000,500)
104
105     def quitFunction(self):
106         cv.destroyAllWindows()
107         self.close()
108
109 app=QApplication(sys.argv)
110 win=TrafficWeak()
111 win.show()
112 app.exec_()
```

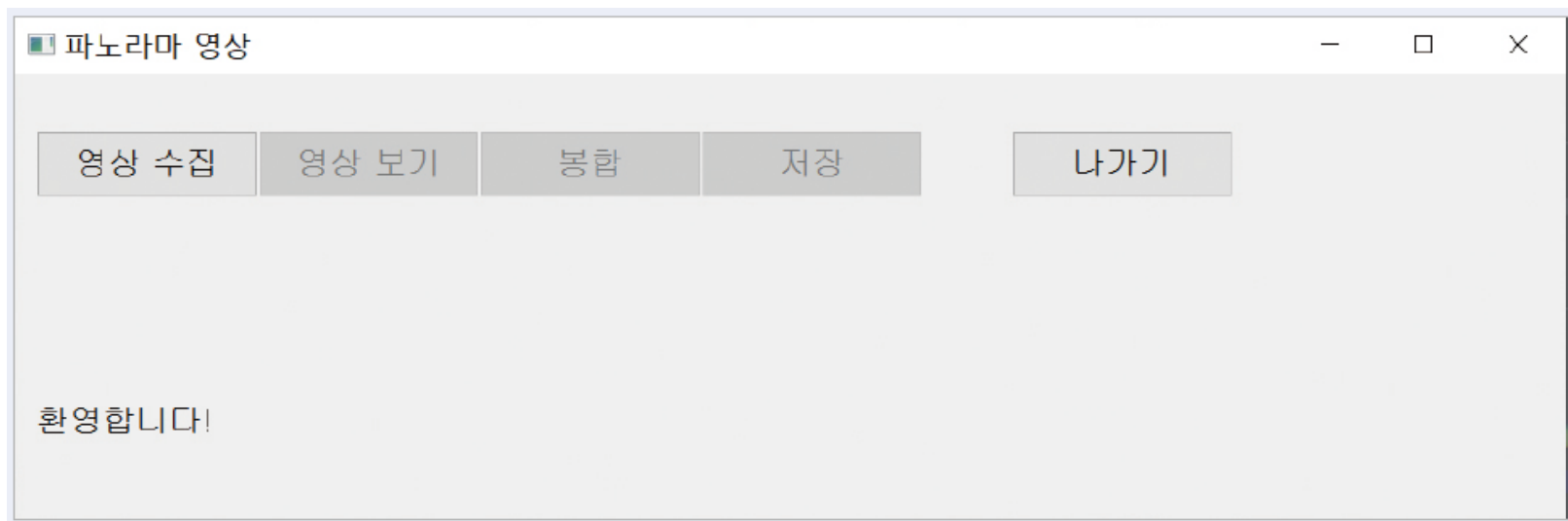
## 6.4 [비전 에이전트 2] 교통약자 보호구역 알림



## 6.5 [비전 에이전트 3] 파노라마 영상 제작

### ■ 사진 여러 장을 봉합하여 파노라마 영상 제작

- OpenCV의 sticth 함수 활용



## 6.5 [비전 에이전트 3] 파노라마 영상 제작

프로그램 6-5

비디오에서 수집한 영상을 봉합하여 파노라마 영상 제작하기

```
01  from PyQt5.QtWidgets import *
02  import cv2 as cv
03  import numpy as np
04  import winsound
05  import sys
06
07  class Panorama(QMainWindow) :
08      def __init__(self) :
09          super().__init__()
10          self.setWindowTitle('파노라마 영상')
11          self.setGeometry(200,200,700,200)
12
13          collectButton=QPushButton('영상 수집',self)
14          self.showButton=QPushButton('영상 보기',self)
15          self.stitchButton=QPushButton('봉합',self)
16          self.saveButton=QPushButton('저장',self)
17          quitButton=QPushButton('나가기',self)
18          self.label=QLabel('환영합니다!',self)
```



## 6.5 [비전 에이전트 3] 파노라마 영상 제작

```
19
20     collectButton.setGeometry(10,25,100,30)
21     self.showButton.setGeometry(110,25,100,30)
22     self.stitchButton.setGeometry(210,25,100,30)
23     self.saveButton.setGeometry(310,25,100,30)
24     quitButton.setGeometry(450,25,100,30)
25     self.label.setGeometry(10,70,600,170)
26
27     self.showButton.setEnabled(False)
28     self.stitchButton.setEnabled(False)
29     self.saveButton.setEnabled(False)
30
31     collectButton.clicked.connect(self.collectFunction)
32     self.showButton.clicked.connect(self.showFunction)
33     self.stitchButton.clicked.connect(self.stitchFunction)
34     self.saveButton.clicked.connect(self.saveFunction)
35     quitButton.clicked.connect(self.quitFunction)
36
```

## 6.5 [비전 에이전트 3] 파노라마 영상 제작

```
37     def collectFunction(self):
38         self.showButton.setEnabled(False)
39         self.stitchButton.setEnabled(False)
40         self.saveButton.setEnabled(False)
41         self.label.setText('c를 여러 번 눌러 수집하고 끝나면 q를 눌러 비디오를 끕니다.')
42
43         self.cap=cv.VideoCapture(0,cv.CAP_DSHOW)
44         if not self.cap.isOpened(): sys.exit('카메라 연결 실패')
45
46         self.imgs=[]
47         while True:
48             ret,frame=self.cap.read()
49             if not ret: break
50
51             cv.imshow('video display', frame)
52
53             key=cv.waitKey(1)
54             if key==ord('c'):
55                 self.imgs.append(frame) # 영상 저장
56             elif key==ord('q'):
57                 self.cap.release()
58                 cv.destroyAllWindows('video display')
59                 break
60
61         if len(self.imgs)>=2: # 수집한 영상이 두 장 이상이면
62             self.showButton.setEnabled(True)
63             self.stitchButton.setEnabled(True)
64             self.saveButton.setEnabled(True)
```



## 6.5 [비전 에이전트 3] 파노라마 영상 제작

```
65
66     def showFunction(self):
67         self.label.setText('수집된 영상은 '+str(len(self.imgs))+ '장입니다.')
68         stack=cv.resize(self.imgs[0],dsize=(0,0),fx=0.25,fy=0.25)
69         for i in range(1,len(self.imgs)):
70             stack=np.hstack((stack,cv.resize(self.imgs[i],dsize=(0,0),fx=0.25,
71             fy=0.25)))
72
73         cv.imshow('Image collection',stack)
74
75     def stitchFunction(self):
76         stitcher=cv.Stitcher_create()
77         status,self.img_stitched=stitcher.stitch(self.imgs)
78         if status==cv.STITCHER_OK:
79             cv.imshow('Image stitched panorama',self.img_stitched)
80         else:
81             winsound.Beep(3000,500)
82             self.label.setText('파노라마 제작에 실패했습니다. 다시 시도하세요.')
```

## 6.5 [비전 에이전트 3] 파노라마 영상 제작

```
82     def saveFunction(self):
83         fname=QFileDialog.getSaveFileName(self, '파일 저장', './')
84         cv.imwrite(fname[0],self.img_stitched)
85
86     def quitFunction(self):
87         self.cap.release()
88         cv.destroyAllWindows()
89         self.close()
90
91 app=QApplication(sys.argv)
92 win=Panorama()
93 win.show()
94 app.exec_()
```

## 6.5 [비전 에이전트 3] 파노라마 영상 제작



## 6.6 [비전 에이전트 4] 특수 효과

### ■ OpenCV는 특수 효과를 위한 흥미로운 함수 제공

```
cv.stylization(src, sigma_s=60, sigma_r=0.45) → dst
```

매개변수:

src: 입력 영상(8-비트 3-채널 입력 영상)

sigma\_s: 스무딩을 위한 가우시안의 표준편차  $\sigma$ (0~200 범위)

sigma\_r: 양방향 필터가 사용하는 두 번째 가우시안의 표준편차  $\sigma$ (0~1 범위)

반환값:

dst: 특수 효과 처리된 영상(8-비트 3-채널 영상)

**TIP** 특수 효과를 위한 함수 참고 자료

- stylization, pencilSketch: [https://docs.opencv.org/3.4/df/dac/group\\_\\_photo\\_\\_render.html](https://docs.opencv.org/3.4/df/dac/group__photo__render.html), <https://learnopencv.com/non-photorealistic-rendering-using-opencv-python-c>
- oilPainting: [https://docs.opencv.org/4.x/de/daa/group\\_\\_xphoto.html](https://docs.opencv.org/4.x/de/daa/group__xphoto.html)

## 6.6 [비전 에이전트 4] 특수 효과

### ■ OpenCV는 특수 효과를 위한 흥미로운 함수 제공

```
cv.pencilSketch(src, sigma_s=60, sigma_r=0.07, shade_factor=0.02) → dst1, dst2
```

매개변수:

src: 입력 영상(8-비트 3-채널 입력 영상)

sigma\_s: 스무딩을 위한 가우시안의 표준편차 (0~200 범위)

sigma\_r: 양방향 필터가 사용하는 두 번째 가우시안의 표준편차  $\sigma$  (0~1 범위)

shade\_factor: 출력 영상의 밝은 정도(0~0.1 범위)

반환값:

dst1: 특수 효과 처리된 명암 영상(8-비트 1-채널 영상)

dst2: 특수 효과 처리된 컬러 영상(8-비트 3-채널 영상)

```
cv.xphoto.oilPainting(src, size, dynRatio,, code) → dst
```

매개변수:

src: 입력 영상(8-비트 3-채널 또는 1-채널 입력 영상)

size: 2\*size+1 패치에서 히스토그램을 구함

dynRatio: 명암값을 dynRatio로 나누고 히스토그램을 구함

code: 컬러 공간 지정

반환값:

dst: 특수 효과 처리된 영상(입력 영상과 같은 모양)

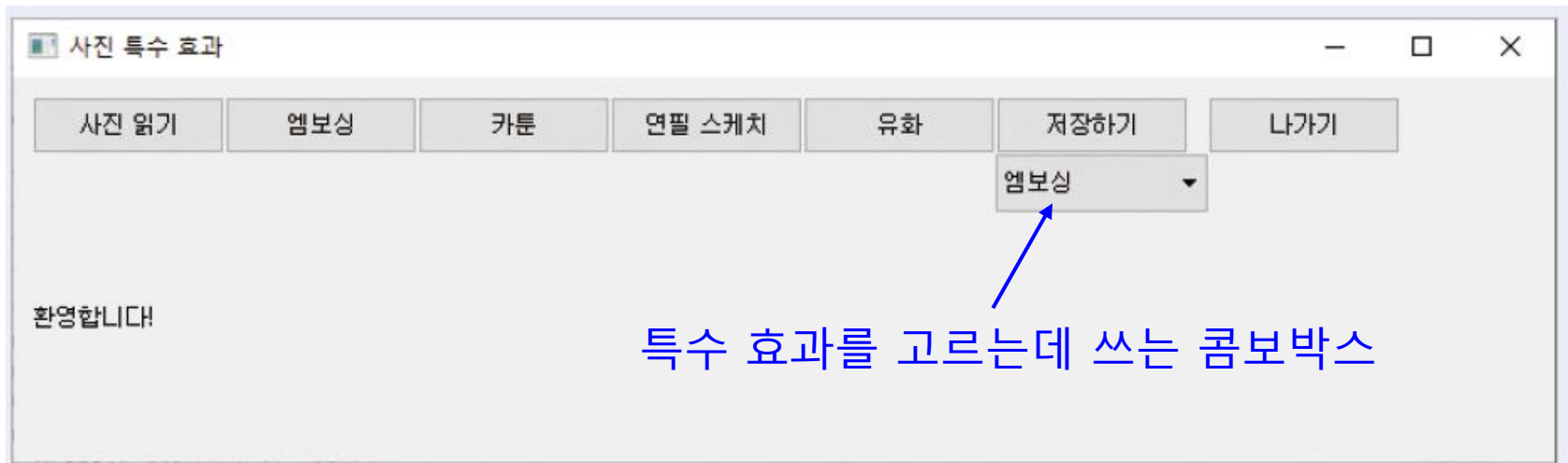
## 6.6 [비전 에이전트 4] 특수 효과

### ■ 에지 보존 필터

- 이들 함수는 에지 보존 필터를 사용하여 특수 효과 발휘

$$f'(x) = \sum_{i=-(w-1)/2}^{(w-1)/2} g_s(i) g_r(f(x) - f(x+i)) f(x+i) \quad (6.1)$$

### ■ 사용자 인터페이스



## 6.6 [비전 에이전트 4] 특수 효과

프로그램 6-6

사진 영상에 특수 효과 처리하기

```
01 import cv2 as cv
02 import numpy as np
03 from PyQt5.QtWidgets import *
04 import sys
05
06 class SpecialEffect(QMainWindow):
07     def __init__(self):
08         super().__init__()
09         self.setWindowTitle('사진 특수 효과')
10         self.setGeometry(200,200,800,200)
11
12         pictureButton=QPushButton('사진 읽기',self)
13         embossButton=QPushButton('엠보싱',self)
14         cartoonButton=QPushButton('카툰',self)
15         sketchButton=QPushButton('연필 스케치',self)
16         oilButton=QPushButton('유화',self)
17         saveButton=QPushButton('저장하기',self)
18         self.pickCombo=QComboBox(self)
19         self.pickCombo.addItem(['엠보싱', '카툰', '연필 스케치(명암)', '연필 스케치(컬러)',
20                                 '유화'])
21
22         quitButton=QPushButton('나가기',self)
23         self.label=QLabel('환영합니다!',self)
```

## 6.6 [비전 에이전트 4] 특수 효과

```
23     pictureButton.setGeometry(10,10,100,30)
24     embossButton.setGeometry(110,10,100,30)
25     cartoonButton.setGeometry(210,10,100,30)
26     sketchButton.setGeometry(310,10,100,30)
27     oilButton.setGeometry(410,10,100,30)
28     saveButton.setGeometry(510,10,100,30)
29     self.pickCombo.setGeometry(510,40,110,30)
30     quitButton.setGeometry(620,10,100,30)
31     self.label.setGeometry(10,40,500,170)
32
33     pictureButton.clicked.connect(self.pictureOpenFunction)
34     embossButton.clicked.connect(self.embossFunction)
35     cartoonButton.clicked.connect(self.cartoonFunction)
36     sketchButton.clicked.connect(self.sketchFunction)
37     oilButton.clicked.connect(self.oilFunction)
38     saveButton.clicked.connect(self.saveFunction)
39     quitButton.clicked.connect(self.quitFunction)
40
```



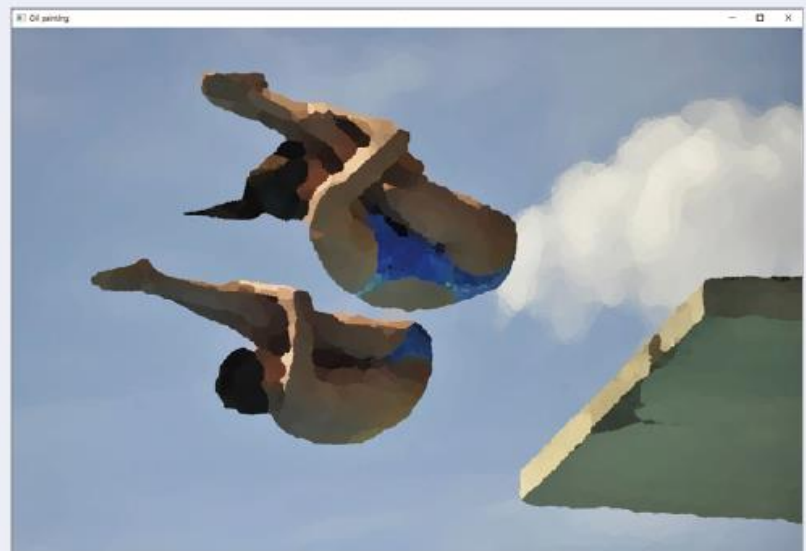
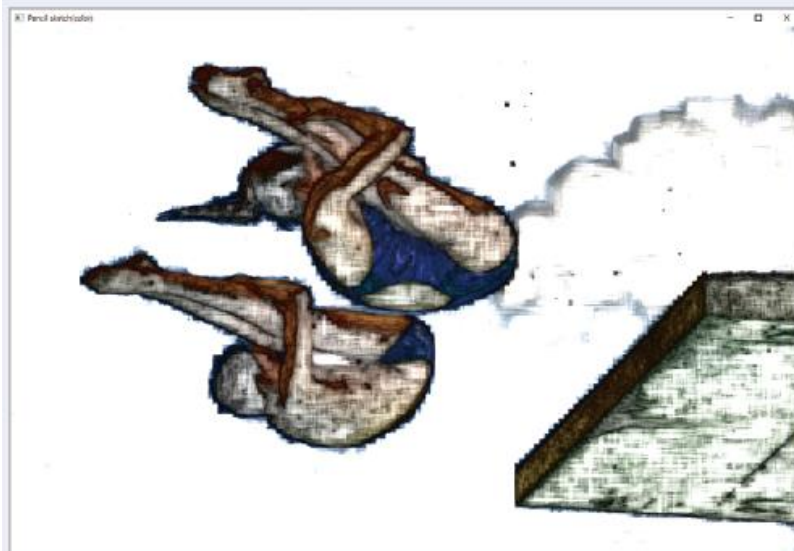
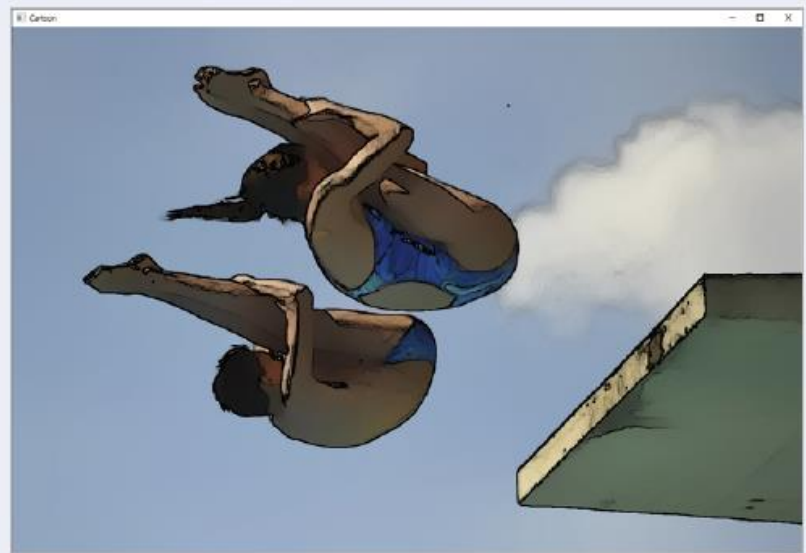
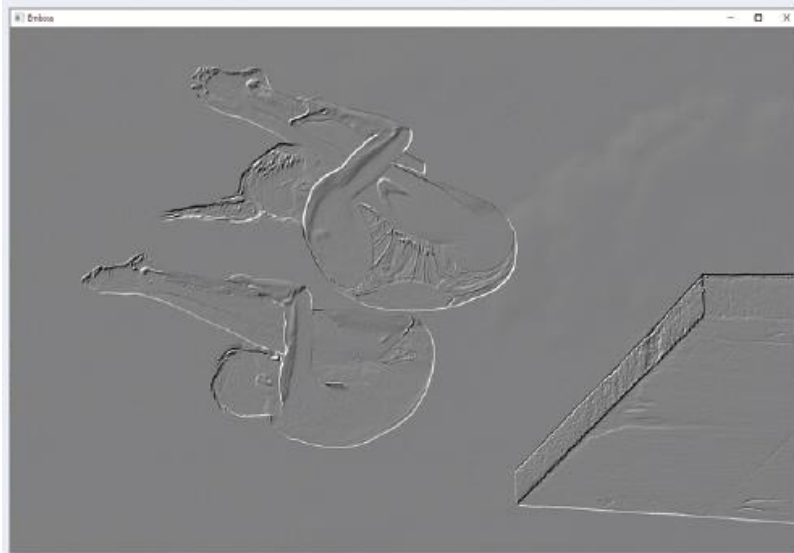
## 6.6 [비전 에이전트 4] 특수 효과

```
41 def pictureOpenFunction(self):
42     fname=QFileDialog.getOpenFileName(self, '사진 읽기', './')
43     self.img=cv.imread(fname[0])
44     if self.img is None: sys.exit('파일을 찾을 수 없습니다.')
45
46     cv.imshow('Painting',self.img)
47
48 def embossFunction(self):
49     femboss=np.array([[ -1.0,  0.0,  0.0],[ 0.0,  0.0,  0.0],[ 0.0,  0.0,  1.0]])
50
51     gray=cv.cvtColor(self.img,cv.COLOR_BGR2GRAY)
52     gray16=np.int16(gray)
53     self.emboss=np.uint8(np.clip(cv.filter2D(gray16,-1,femboss)+128,0,255))
54
55     cv.imshow('Emboss',self.emboss)
56
57 def cartoonFunction(self):
58     self.cartoon=cv.stylization(self.img,sigma_s=60,sigma_r=0.45)
59     cv.imshow('Cartoon',self.cartoon)
60
61 def sketchFunction(self):
62     self.sketch_gray,self.sketch_color=cv.pencilSketch(self.img,sigma_
        s=60,sigma_r=0.07,shade_factor=0.02)
63     cv.imshow('Pencil sketch(gray)',self.sketch_gray)
64     cv.imshow('Pencil sketch(color)',self.sketch_color)
```

## 6.6 [비전 에이전트 4] 특수 효과

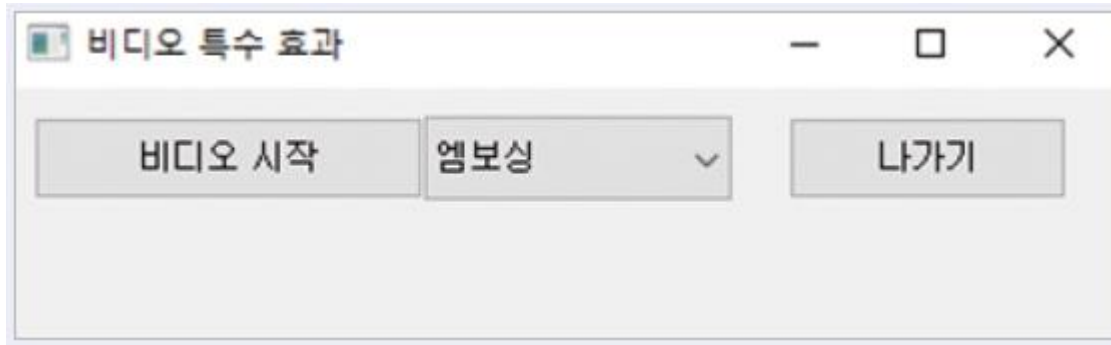
```
65
66     def oilFunction(self):
67         self.oil=cv.xphoto.oilPainting(self.img,10,1,cv.COLOR_BGR2Lab)
68         cv.imshow('Oil painting',self.oil)
69
70     def saveFunction(self):
71         fname=QFileDialog.getSaveFileName(self,'파일 저장','./')
72
73         i=self.pickCombo.currentIndex()
74         if i==0: cv.imwrite(fname[0],self.emboss)
75         elif i==1: cv.imwrite(fname[0],self.cartoon)
76         elif i==2: cv.imwrite(fname[0],self.sketch_gray)
77         elif i==3: cv.imwrite(fname[0],self.sketch_color)
78         elif i==4: cv.imwrite(fname[0],self.oil)
79
80     def quitFunction(self):
81         cv.destroyAllWindows()
82         self.close()
83
84 app=QApplication(sys.argv)
85 win=SpecialEffect()
86 win.show()
87 app.exec_()
```

## 6.6 [비전 에이전트 4] 특수 효과



## 6.6 [비전 에이전트 4] 특수 효과

### ■ 비디오에 특수 효과 적용



## 6.6 [비전 에이전트 4] 특수 효과

프로그램 6-7

비디오 영상에 특수 효과 처리하기

```
01 import cv2 as cv
02 import numpy as np
03 from PyQt5.QtWidgets import *
04 import sys
05
06 class VideoSpecialEffect(QMainWindow):
07     def __init__(self):
08         super().__init__()
09         self.setWindowTitle('비디오 특수 효과')
10         self.setGeometry(200,200,400,100)
11
12         videoButton=QPushButton('비디오 시작',self)
13         self.pickCombo=QComboBox(self)
14         self.pickCombo.addItem('엠보싱','카툰','연필 스케치(명암)','연필 스케치(컬러)',
15                                 '유화'])
16
17         quitButton=QPushButton('나가기',self)
18
19         videoButton.setGeometry(10,10,140,30)
20         self.pickCombo.setGeometry(150,10,110,30)
21         quitButton.setGeometry(280,10,100,30)
22
23         videoButton.clicked.connect(self.videoSpecialEffectFunction)
24         quitButton.clicked.connect(self.quitFunction)
```

## 6.6 [비전 에이전트 4] 특수 효과

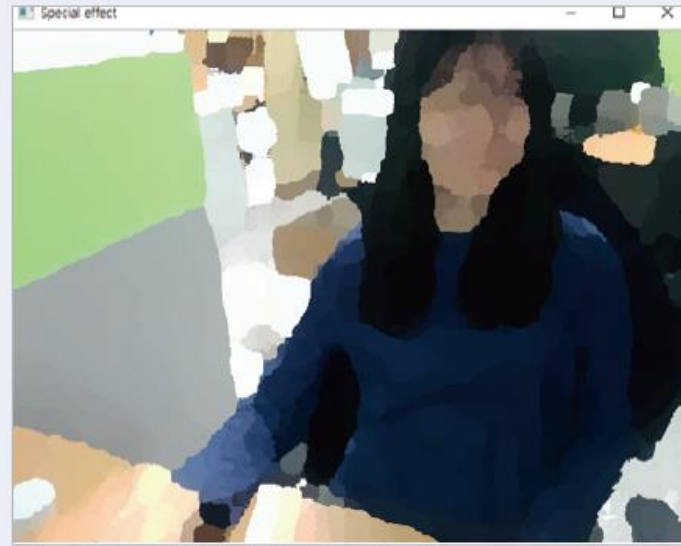
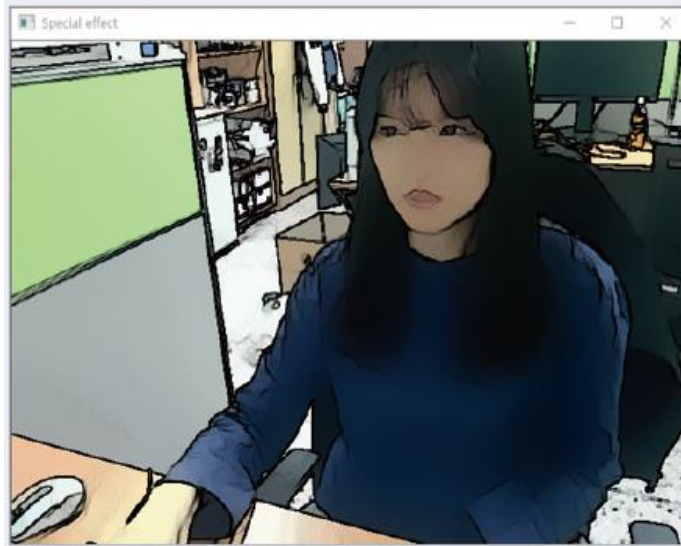
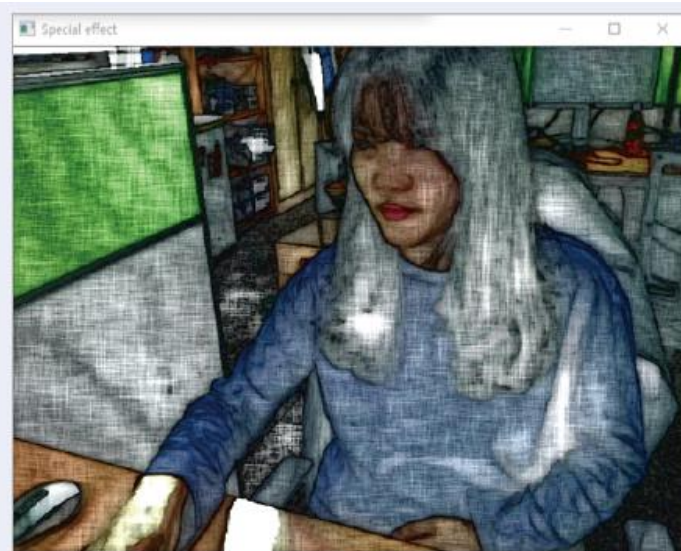
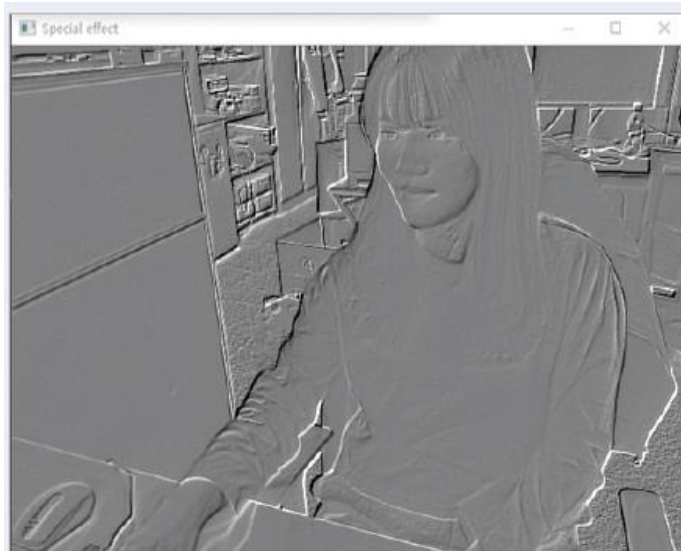
```
24     def videoSpecialEffectFunction(self):
25         self.cap=cv.VideoCapture(0,cv.CAP_DSHOW)
26         if not self.cap.isOpened(): sys.exit('카메라 연결 실패')
27
28         while True:
29             ret,frame=self.cap.read()
30             if not ret: break
31
32             pick_effect=self.pickCombo.currentIndex()
33             if pick_effect==0:
34                 femboss=np.array([[ -1.0,  0.0,  0.0],[ 0.0,  0.0,  0.0],[ 0.0,  0.0,  1.0]])
35                 gray=cv.cvtColor(frame,cv.COLOR_BGR2GRAY)
36                 gray16=np.int16(gray)
37                 special_img=np.uint8(np.clip(cv.filter2D(gray16,-1,femboss)+
38                                     128,0,255))
39             elif pick_effect==1:
40                 special_img=cv.stylization(frame,sigma_s=60,sigma_r=0.45)
41             elif pick_effect==2:
42                 special_img,_=cv.pencilSketch(frame,sigma_s=60,sigma_r=0.07,
43                 shade_factor=0.02)
44             elif pick_effect==3:
45                 _,special_img=cv.pencilSketch(frame,sigma_s=60,sigma_r=0.07,
46                 shade_factor=0.02)
47             elif pick_effect==4:
48                 special_img=cv.xphoto.oilPainting(frame,10,1,cv.COLOR_BGR2Lab)
49
50             cv.imshow('Special effect',special_img)
51             cv.waitKey(1)
```

## 6.6 [비전 에이전트 4] 특수 효과

```
49
50     def quitFunction(self):
51         self.cap.release()
52         cv.destroyAllWindows()
53         self.close()
54
55 app=QApplication(sys.argv)
56 win=VideoSpecialEffect()
57 win.show()
58 app.exec_()
```

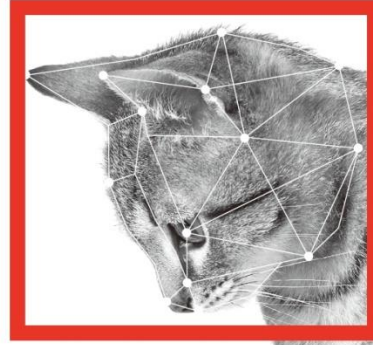


## 6.6 [비전 에이전트 4] 특수 효과





COMPUTER VISION



DEEP  
LEARNING



# 컴퓨터 비전과 딥러닝

감사합니다.