

Term Project Report



과목명: 파이썬과학프로그래밍기초

담당교수: 송성호 교수님

학과: 소프트웨어융합대학 콘텐츠 IT 전공

학번: 20205220

이름: 이바다

목차

1. 프로그램 설명
2. 동작 설명
3. 결과

1. 프로그램 설명

◆ 시작화면

개인정보

Image Processing

RC circuit

성명

나이

학번

학년

전공

email

학점

1학년

2학년

3학년

4학년

입력완료

저장

출력

RC circuit

Image processing

◆ 입력완료

이름, 나이, 학번, 전공, email, 학년별 학점을 입력한 후 입력완료 버튼을 누르면 콘솔에 출력된다.

◆ 저장

텍스트 박스에 입력한 내용을 20205220_이바다.txt 파일에 저장한다.

◆ 출력

20205220_이바다.txt 에 저장된 이름, 나이, 학번, 학년, 전공, email, 학년별 학점을 읽어와서 텍스트박스에 표시한다.

◆ RC circuit

R=1000000, C=0.000001 일 때 시뮬레이션을 수행하여 step response 그래프를 표시한다.

◆ Image Processing

두 개의 이미지 Pic1.png 와 Pic2.png 의 이미지 파일을 불러와서 표시한다.

2. 동작 설명

◆ 시작화면 만들기

```
7 import cv2
8 import numpy as np
9 import PIL
10 import matplotlib.pyplot as plt
11 from tkinter import *
12
13
14 class MyInformation:
15
16
17     def __init__(self):
18         self.window = Tk()
19         self.window.title("개인정보")
20         self.window.geometry("750x320+700+10")
21         self.window.resizable(True, True)
22
23         self.finishButton = Button(self.window, text = "입력완료", command = self.print).place(x = 0, y = 265, width = 100)
24         self.saveButton = Button(self.window, text = "저장", command = self.save).place(x = 150, y = 265, width = 100)
25         self.printButton = Button(self.window, text = "출력", command = self.scan).place(x = 300, y = 265, width = 100)
26         self.RCButton = Button(self.window, text = "RC circuit", command = self.RC_Circuit).place(x = 450, y = 265, width = 100)
27         self.IPButton = Button(self.window, text = "Image processing", command = self.Image_Processing).place(x = 600, y = 265, width = 100)
28
29         self.label1 = Label(self.window, text = "개인정보").grid(row = 1, column = 1)
30         self.label2 = Label(self.window, text = "학점").grid(row = 9, column = 1)
31         self.label3 = Label(self.window, text = "Image Processing").place(x = 250, y = 0)
32         self.label4 = Label(self.window, text = "RC circuit").place(x = 550, y = 0)
33
34         self.label_name = Label(self.window, text = "성명").grid(row = 2, column = 0)
35         self.name = StringVar()
36         self.entry_name = Entry(self.window, width = 17, textvariable = self.name)
37         self.entry_name.grid(row = 2, column = 1)
38
39
40         self.label_age = Label(self.window, text = "나이").grid(row = 3, column = 0)
41         self.age = StringVar()
42         self.entry_age = Entry(self.window, width = 17, textvariable = self.age)
43         self.entry_age.grid(row = 3, column = 1)
44
45
46         self.label_studentNumber = Label(self.window, text = "학번").grid(row = 4, column = 0)
47         self.studentNumber = StringVar()
48         self.entry_studentNumber = Entry(self.window, width = 17, textvariable = self.studentNumber)
49         self.entry_studentNumber.grid(row = 4, column = 1)
50
51
52         self.label_grade = Label(self.window, text = "학년").grid(row = 5, column = 0)
53         self.grade = StringVar()
54         self.entry_grade = Entry(self.window, width = 17, textvariable = self.grade)
55         self.entry_grade.grid(row = 5, column = 1)
56
57
58         self.label_major = Label(self.window, text = "전공").grid(row = 6, column = 0)
59         self.major = StringVar()
60         self.entry_major = Entry(self.window, width = 17, textvariable = self.major)
61         self.entry_major.grid(row = 6, column = 1)
```

```

62     self.label_email = Label(self.window, text = "email").grid(row = 7, column = 0)
63     self.email = StringVar()
64     self.entry_email = Entry(self.window, width = 17, textvariable = self.email)
65     self.entry_email.grid(row = 7, column = 1)
66
67
68     self.label_score1 = Label(self.window, text = "1학년").grid(row = 10, column = 0)
69     self.score1 = StringVar()
70     self.entry_score1 = Entry(self.window, width = 3, textvariable = self.score1)
71     self.entry_score1.grid(row = 10, column = 1)
72
73     self.label_score2 = Label(self.window, text = "2학년").grid(row = 11, column = 0)
74     self.score2 = StringVar()
75     self.entry_score2 = Entry(self.window, width = 3, textvariable = self.score2)
76     self.entry_score2.grid(row = 11, column = 1)
77
78     self.label_score3 = Label(self.window, text = "3학년").grid(row = 12, column = 0)
79     self.score3 = StringVar()
80     self.entry_score3 = Entry(self.window, width = 3, textvariable = self.score3)
81     self.entry_score3.grid(row = 12, column = 1)
82
83     self.label_score4 = Label(self.window, text = "4학년").grid(row = 13, column = 0)
84     self.score4 = StringVar()
85     self.entry_score4 = Entry(self.window, width = 3, textvariable = self.score4)
86     self.entry_score4.grid(row = 13, column = 1)
87
88     self.window.mainloop()
89

```

- ➔ tkinter 를 사용하기 위해 import 해준다.
- ➔ ImageProcessing 과 그래프를 그리기 위해 opencv 인 cv2 와 numpy, 이미지와 관련된 모듈 pillow 인 PIL, matplotlib 을 import 해준다.
- ➔ MyInformation class 를 정의한다.
- ➔ __init__ 함수를 정의하여 객체가 생성될 때 초기화한다.
- ➔ tkinter 함수 Tk()를 이용하여 윈도우창을 생성한다. 윈도우 창의 제목은 "개인정보"이고, 크기는 너비 750, 높이 320, x=700, y=10 으로 설정한다.
- ➔ tkinter 함수 Button()을 이용하여 입력완료, 저장, 출력, RC circuit, Image processing 버튼을 생성한다. 각 버튼 클릭 시 print 함수, save 함수, scan 함수, RC_Circuit 함수, Image_Processing 함수가 각각 호출된다. command 를 이용하여 Button 과 함수를 연결한다. 절대 배치 관리자인 place 를 이용하여 버튼들의 위치를 절대 좌표로 정한다.

- ➔ tkinter 함수 Label()을 이용하여 "개인정보", "학점", "Image Processing", "RC circuit"를 윈도우에 입력한다. "개인정보", "학점"은 격자 배치 관리자인 grid 를 이용하여 row 와 column 위치에 위젯을 배치하고, "Image Processing"과 "RC circuit"은 절대 배치 관리자 place 를 이용하여 라벨 위치를 절대 좌표로 정한다.
- ➔ tkinter 함수 Entry()를 사용하여 이름, 나이, 학번, 학년, 전공, email, 학년별 학점을 입력 받을 텍스트 박스를 만든다. 텍스트 박스는 격자 배치 관리자 grid 를 이용하여 row 와 column 위치에 위젯을 배치하고 StringVar()와 textvariable 를 이용하여 엔트리 박스에 입력한 내용을 변수에 저장한다. 변수는 클래스 객체 변수로 정의한다.
- ➔ self.window.mainloop()로 이벤트 처리를 위한 메시지 루프를 실행한다.

◆ print 함수

```

91     def print(self) :
92         print("이름 :", self.name.get())
93         print("나이 :", self.age.get())
94         print("학번 :", self.studentNumber.get())
95         print("학년 :", self.grade.get())
96         print("전공 :", self.major.get())
97         print("email :", self.email.get())
98         print("1학년 학점 :", self.score1.get())
99         print("2학년 학점 :", self.score2.get())
100        print("3학년 학점 :", self.score3.get())
101        print("4학년 학점 :", self.score4.get())

```

- ➔ 텍스트 박스에 입력한 내용을 저장한 변수를 get()으로 불러와서 str 로 print 한다.

◆ save 함수

```

103     def save(self) :
104         f = open("20205220_이바다.txt", 'w')
105         f.write("{}\n".format(self.name.get()))
106         f.write("{}\n".format(self.age.get()))
107         f.write("{}\n".format(self.studentNumber.get()))
108         f.write("{}\n".format(self.grade.get()))
109         f.write("{}\n".format(self.major.get()))
110         f.write("{}\n".format(self.email.get()))
111         f.write("{}\n".format(self.score1.get()))
112         f.write("{}\n".format(self.score2.get()))
113         f.write("{}\n".format(self.score3.get()))
114         f.write(self.score4.get())
115
116         f.close()
117         print("파일에 저장되었습니다.")

```

- ➔ open()을 이용하여 "20205220_이바다.txt" 파일을 쓰기모드로 연다.
- ➔ write()를 이용하여 텍스트 박스에 입력된 내용을 "20205220_이바다.txt"파일에 쓴다.
- ➔ close()를 이용하여 파일을 닫은 후 "파일에 저장되었습니다."를 출력하여 정상적으로 저장되었음을 알려준다.

◆ scan 함수

```

119     def scan(self) :
120         f = open("20205220_이바다.txt", 'r')
121         self.entry_name.delete(0,END)
122         self.entry_age.delete(0,END)
123         self.entry_studentNumber.delete(0,END)
124         self.entry_grade.delete(0,END)
125         self.entry_major.delete(0,END)
126         self.entry_email.delete(0,END)
127         self.entry_score1.delete(0,END)
128         self.entry_score2.delete(0,END)
129         self.entry_score3.delete(0,END)
130         self.entry_score4.delete(0,END)
131
132         self.entry_name.insert(0, f.readline())
133         self.entry_age.insert(0, f.readline())
134         self.entry_studentNumber.insert(0, f.readline())
135         self.entry_grade.insert(0, f.readline())
136         self.entry_major.insert(0, f.readline())
137         self.entry_email.insert(0, f.readline())
138         self.entry_score1.insert(0, f.readline())
139         self.entry_score2.insert(0, f.readline())
140         self.entry_score3.insert(0, f.readline())
141         self.entry_score4.insert(0, f.readline())
142
143         f.close()
144

```

- ➔ open()을 이용하여 "20205220_이바다.txt"파일을 읽기모드로 연다.
- ➔ delete()를 이용하여 텍스트 박스에 있는 내용을 지운다.
- ➔ insert()와 readline()을 이용하여 파일에 있는 내용을 한 줄씩 읽은 후 텍스트 박스에 출력한다.
- ➔ close()를 이용하여 파일을 닫는다.

◆ RC_Circuit 함수

```

235     def RC_Circuit(self) :
236         Ydat = []
237         Xdat = []
238         Ndat = []
239         n_max=6000
240
241         R = 10000
242         C = 0.000001
243         dt = 0.00001
244
245         Ytemp =0
246         Xdat.append(Ytemp)
247         Ydat.append(Ytemp)
248         num = dt/R/C
249         for n in range(0, n_max) :
250             Xtemp=1
251             Ytemp1=Ytemp
252             Ytemp=(1-num)*Ytemp1+num*Xtemp
253             Xdat.append(n*dt)
254             Ydat.append(Ytemp)
255             Ndat.append(n)
256         print(Ydat)
257
258
259
260         plt.figure(1)
261
262         plt.plot(Xdat, Ydat)
263         plt.xlabel('Time(sec)')
264         plt.ylabel('VC', fontsize=20)
265         plt.axis([0,0.1,0,2])
266         plt.title("Step Response")
267         plt.grid(True)
268         plt.savefig('graph.png', dpi=45)
269
270         self.gImage = PhotoImage(file = 'graph.png')
271         self.label_g = Label(self.window, image = self.gImage).place(x = 450, y = 25)
272         self.window.mainloop()
273
274

```

- ➔ Ydat 은 출력과 관련된 데이터를 담고, Xdat 는 시간 데이터를 담기 위해 array 로 설정한다. list 형태이다.
- ➔ sample array 의 max 는 6000 으로 설정한다.

- ➔ $R = 1000000$, $C = 0.000001$, $dt = 0.00001$ 로 설정한다. R 은 resistance, C 는 capacitance, dt 는 sampling period 이다.
- ➔ 초기 출력값을 $Y_{temp} = 0$ 으로 저장하고, X_{dat} 와 Y_{dat} 를 초기값으로 정리를 한다.
- ➔ num 은 RC 분의 dt 를 나타낸 것이다.
- ➔ for 루프를 돌면서 Y_{temp} 를 현재 직전의 출력값과 직전의 입력값을 가지고 업데이트하고, 이를 X_{dat} , Y_{dat} , N_{dat} 에 append 한다.
- ➔ `plt.figure(1)`는 첫번째 그래프를 나타낸다.
- ➔ x 축은 X_{dat} , y 축은 Y_{dat} 로 그래프를 그린다.
- ➔ x 축 라벨을 'Time(sec)', y 축 라벨을 'VC'로 설정하고, 제목은 "Step Response"로 설정한다. `grid(True)`를 이용하여 그리드가 보이게 한다.
- ➔ `plt.axis()`를 이용하여 0, 0.1 은 x 축의 시작점과 끝점, 0, 2 는 y 축의 시작점과 끝점으로 설정한다.
- ➔ `plt.savefig()`를 이용하여 그래프를 이미지 파일로 저장한다.
- ➔ 저장된 그래프 이미지 파일을 `PhotoImage` 로 읽어와서 라벨을 이용해 윈도우창에 표시되게 한다. 절대 배치 관리자 `place` 를 사용하여 절대 좌표에 위치한다.

◆ Image_Processing 함수

```

46 def Image_Processing(self) :
47     image_shape = (384, 384, 3)
48     working_dir = "/"
49     def lr_images(images_real, downscale) :
50         images = []
51         images.append(np.array(PIL.Image.fromarray(images_real).resize([images_real.shape[0]//downscale, images_real.shape[1]//downscale],
52                                     resample=PIL.Image.BICUBIC)))
53         images_lr = np.array(images)
54         return images_lr
55
56     img2 = cv2.imread(working_dir + 'Pic1.png', cv2.IMREAD_UNCHANGED)
57     print("img2", img2)
58     print("img2 size :", img2.size)
59     print("img2 shape :", img2.shape)
60
61     img3 = np.reshape(img2[:, :, :3], (384, 384, 3))
62     print("img3", img3)
63     print("img3 size :", img3.size)
64     print("img3 shape :", img3.shape)
65
66     img4 = lr_images(img3, 4)
67     print("img4", img4)
68     print("img4 size :", img4.size)
69     print("img4 shape :", img4.shape)
70
71     figsize = (24, 10)
72     dim = (1, 2)
73     plt.figure(figsize=figsize)
74     plt.subplot(dim[0], dim[1], 1)
75     plt.imshow(img2, interpolation = 'nearest')
76     plt.title('Original : 384x384 ')
77     plt.subplot(dim[0], dim[1], 2)
78     plt.imshow(img4[0], interpolation = 'nearest')
79     plt.title('Original Low Resolution : 96x96 ')
80     plt.savefig("o1.png")
81     plt.show()
82
83     cv2.imwrite(working_dir + 'Pic1.png', img4[0])
84
85     img2 = cv2.imread(working_dir + 'Pic2.png', cv2.IMREAD_UNCHANGED)
86     print("img2", img2)
87     print("img2 size :", img2.size)
88     print("img2 shape :", img2.shape)
89
90     img3 = np.reshape(img2[:, :, :3], (384, 384, 3))
91     print("img3", img3)
92     print("img3 size :", img3.size)
93     print("img3 shape :", img3.shape)
94
95     img4 = lr_images(img3, 4)
96     print("img4", img4)
97     print("img4 size :", img4.size)
98     print("img4 shape :", img4.shape)
99

```

```

199
200     figsize = (24, 10)
201     dim = (1, 2)
202     plt.figure(figsize=figsize)
203     plt.subplot(dim[0], dim[1], 1)
204     plt.imshow(img2, interpolation = 'nearest')
205     plt.title('Original : 384x384 ')
206     plt.subplot(dim[0], dim[1], 2)
207     plt.imshow(img4[0], interpolation = 'nearest')
208     plt.title('Original Low Resolution : 96x96 ')
209     plt.savefig("o2.png")
210     plt.show()
211
212     cv2.imwrite(working_dir + 'Pic2.png', img4[0])
213
214     photo = cv2.imread("o1.png", cv2.IMREAD_UNCHANGED)
215     photo_resize = cv2.resize(photo, dsize=(200, 100), interpolation=cv2.INTER_AREA)
216     cv2.imwrite(working_dir + 'o1.png', photo_resize)
217     self.photo1 = PhotoImage(file = 'o1.png')
218     self.photo1_label = Label(self.window, image = self.photo1).place(x = 200, y = 25)
219
220
221     photo = cv2.imread("o2.png", cv2.IMREAD_UNCHANGED)
222     photo_resize = cv2.resize(photo, dsize=(200, 100), interpolation=cv2.INTER_AREA)
223     cv2.imwrite(working_dir + 'o2.png', photo_resize)
224     self.photo = PhotoImage(file = 'o2.png')
225     self.photo_label = Label(self.window, image = self.photo).place(x = 200, y = 125)
226
227     img_destroy = 1
228     while img_destroy == 1 :
229         img_destroy = int(input("input 0 to Clear images."))
230     else :
231         cv2.destroyAllWindows()
232
233     self.window.mainloop()
234

```

- ➔ (384,384)는 이미지 가로, 세로 사이즈이고, 3 은 RGB 이다.
- ➔ `working_dir="./"`는 현재 폴더이다.
- ➔ `lr_images` 함수는 low image 를 얻기 위해 이미지 크기를 변경하는 함수이다. PIL 을 이용하여 이미지 크기를 변경하였다. `downscale` 은 어느 정도로 크기를 변경할 것인지 정해줘야 한다. 위 코드에서는 4 를 줘서 4 분의 1 크기로 변경하였다.
- ➔ `opencv` 에 있는 `imread` 함수를 이용하여 이미지를 읽어온다.
`cv2.IMREAD_UNCHANGED` 는 변경없이 이미지를 읽어오는 것이다.
- ➔ 이미지를 프린트해보고, 이미지의 `size` 와 `shape` 를 출력해본다.
- ➔ `reshape` 는 `numpy` 에 있는 함수로 이미지를 바꾸는 것인데, 위 코드에서는 똑같이 (384,384,3)으로 설정하였기 때문에 바뀌는 것이 없다.
- ➔ `img4` 는 low image 를 구하기 위해 `lr_images` 함수를 이용한다.
(384,384,3) -> (96,96,3)으로 변경되고, 이는 `np.array` 를 하기 때문에 [1,[96,96,3]]이 된다.
- ➔ "Pic2.png"도 "Pic1.png"와 똑같이 위 과정을 한다.
- ➔ 그림으로 나타나기 위해 `figsize` 를 (25, 10)로 설정하고, `dim` 을 (1,2)로 설정한다.
- ➔ `dim` 은 튜플이기 때문에 `dim[0]`은 첫번째, `dim[1]`은 두번째이다. (1, 2, 1)로 써도 된다.
- ➔ `plt` 함수인 `imshow` 를 이용하여 그림을 그린다. `img2` 의 제목은 "Original : 384x384"로 설정하고, `img4` 의 제목은 "Original Low

Resolution : 96x96"으로 설정한다. img4 는 (1,96,96,3)으로 되어 있기 때문에 img4[0]로 인덱싱을 해야한다.

- ➔ plt.savefig()를 사용하여 그려진 그림을 저장한다.
- ➔ plt.show()를 해야지만 이미지가 그래프에 그려진다.
- ➔ opencv 에 있는 함수를 이용하여 그릴 수도 있다.
- ➔ opencv 에 있는 imwrite()함수를 이용하여 이미지를 파일로 저장한다.
- ➔ resize()함수를 이용하여 이미지의 크기를 변경하고, imshow()함수를 이용하여 resize 된 이미지를 그린다.
- ➔ waitKey(5000)은 opencv 에서 영상 이미지를 그려놓고 5 초 정도 기다리고 key 를 누르면 끝내라는 것이다.
- ➔ img_destroy 를 이용하여 0 을 쓰면 끝낸다.
- ➔ imwrite()함수를 이용하여 저장된 'o1.png'파일과 'o2.png'파일을 터킨터를 이용하여 윈도우창에 넣기 위해 imread()로 다시 읽어들이고, resize()를 이용하여 크기를 조절하고 다시 imwrite()를 이용하여 파일로 저장한다.
- ➔ resize 된 이미지 파일을 PhotoImage 로 읽은 후 라벨을 이용하여 윈도우 창에 보이도록 한다. 절대 배치 관리자 place 를 이용하여 절대 좌표에 위치하게 한다.

```
287 myinformation = MyInformation()
```

- ➔ MyInformation 클래스의 객체 myinformation 을 생성한다.

➔ `__init__` 함수에서 각 함수를 호출하였기 때문에 객체만 생성하여도 각 함수들이 호출된다.

3. 결과

◆ 입력완료

개인정보

이름 : 이바다
나이 : 21
학번 : 20205220
학년 : 2
전공 : 콘텐츠 it
email : dlqkek8@naver.com

학점
1학년 : B
2학년 : C
3학년 : A
4학년 : A+

입력완료 저장 출력 RC circuit Image processing

◆ 저장

개인정보

이름 : 이바다
나이 : 21
학번 : 20205220
학년 : 2
전공 : 콘텐츠 it
email : dlqkek8@naver.com

학점
1학년 : B
2학년 : C
3학년 : A
4학년 : A+

입력완료 저장 출력 RC circuit Image processing

Console 1/A

Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/82102/Desktop/2021/파이썬과학프로그래밍기초/파이썬과학프로그래밍기초/20205220_이바다_기말_프로젝트.py', wdir='C:/Users/82102/Desktop/2021/파이썬과학프로그래밍기초/파이썬과학프로그래밍기초')
이름 : 이바다
나이 : 21
학번 : 20205220
학년 : 2
전공 : 콘텐츠 it
email : dlqkek8@naver.com
1학년 학점 : B
2학년 학점 : C
3학년 학점 : A
4학년 학점 : A+
파일에 저장되었습니다.

◆ 출력

개인정보

성명

이바다2

나이

21

학번

20205220

학년

2

전공

콘텐츠 it

email

dlqkek8@naver.com

학점

1학년

B

2학년

C

3학년

A

4학년

A++

입력완료

저장

출력

RC circuit

Image processing

20205220_이바다 - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

이바다2

21

20205220

2

콘텐츠 it

dlqkek8@naver.com

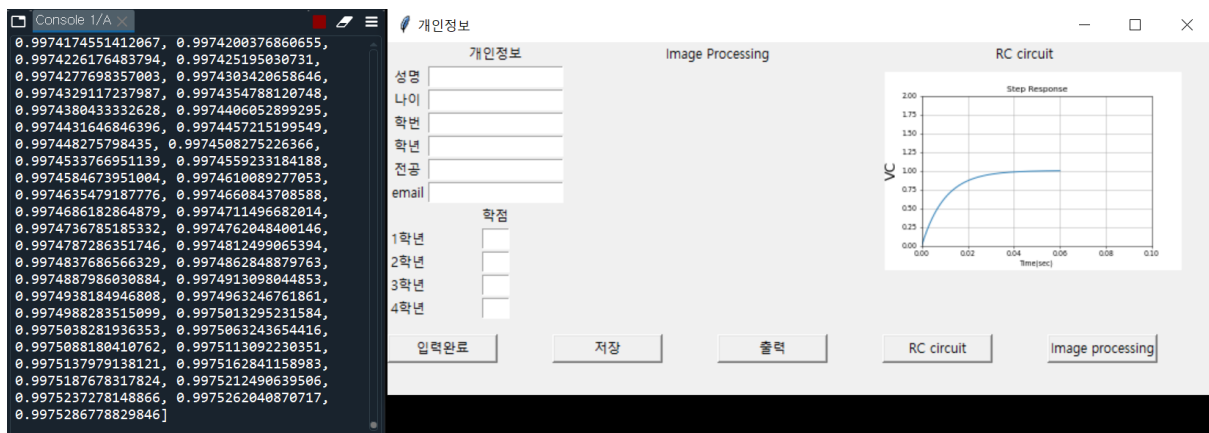
B

C

A

A++

◆ RC circuit



◆ Image Processing

The screenshot displays a software interface with three main sections:

- Console (Left):** Shows a list of coordinates and image data.


```

[101 101 101]
[101 101 101]
[101 101 101]]

[[137 137 137]
 [141 141 141]
 [143 143 143]
 ...
 [102 102 102]
 [102 102 102]
 [102 102 102]]

[[138 138 138]
 [142 142 142]
 [146 146 146]
 ...
 [105 105 105]
 [106 106 106]
 [108 108 108]]]]
img4 size : 27648
img4 shape : (1, 96, 96, 3)
input 0 to clear images. 0
      
```
- Image Processing (Center):** Displays four images of a hand. The top two images show the hand with a blue and red color map, while the bottom two show the original grayscale image.
- RC circuit (Right):** Shows a plot titled "Step Response" with the y-axis labeled "VC" (ranging from 0.00 to 2.00) and the x-axis labeled "Time(sec)" (ranging from 0.00 to 0.10). The plot shows a curve that rises from 0 and levels off at approximately 1.0.

At the bottom of the interface, there are buttons for "저장" (Save), "출력" (Output), "RC circuit", and "Image processing".