

컴퓨팅사고와 파이썬 프로그래밍

Ch 13. 파이썬 프로그램의 응용



교수 김 영 탁

영남대학교 정보통신공학과

(Tel : +82-53-810-2497; E-mail : ytkim@yu.ac.kr)

Outline

- ◆ 터틀그래픽 응용 - 아날로그 시계 구현
- ◆ 애니메이션 - Hanoi Tower
- ◆ pyaudio 모듈기반 전자피아노 구현
- ◆ bottle과 paste 기반의 web server 구현
- ◆ pygame 기반 게임 만들기
- ◆ 파이썬 기반 기계학습 (machine learning)
- ◆ TensorFlow와 Keras 기반 Deep learning, 필기체 숫자 인식



터틀그래픽 기반 아날로그 시계 구현

register_shape()과 tracer()를 사용한 아날로그 시계 구현 (1)

```
# Python Clock Animation Demo (1)
import turtle
from turtle import *
from datetime import datetime

def jump(distance):
    penup();    forward(distance);    pendown()

def rectangle(width, height):
    fd(width/2); lt(90); fd(height); lt(90);
    fd(width); lt(90); fd(height); lt(90);
    fd(width/2)

def make_hand_shape(name, width, height):
    reset()
    left(90); jump(-height*0.15); right(90)
    begin_poly()
    rectangle(width, height*1.15)
    end_poly()
    clock_hand = get_poly()
    register_shape(name, clock_hand)
```

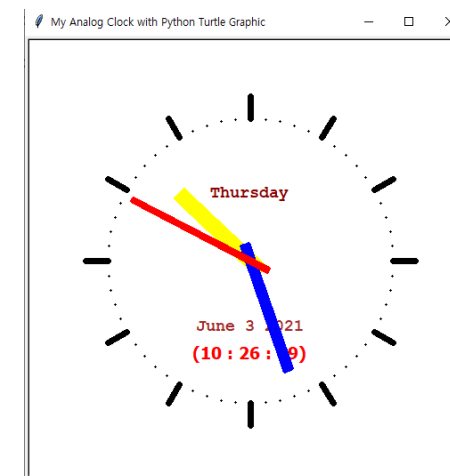
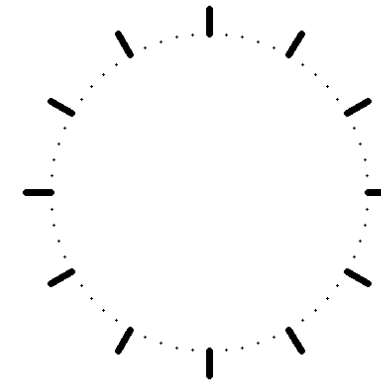


Python Clock Animation Demo (2)

```
def clockface(radius):
    reset()
    pensize(7)
    for i in range(60):
        jump(radius)
        if i % 5 == 0:
            fd(25)
            jump(-radius-25)
        else:
            dot(3)
            jump(-radius)
        rt(6)

def setup():
    global sec_hand, min_hand, hour_hand, writer
    mode("logo")
    make_hand_shape("sec_hand", 5, 150)
    make_hand_shape("min_hand", 10, 130)
    make_hand_shape("hour_hand", 15, 110)
    clockface(160)
    hour_hand = Turtle()
    hour_hand.shape("hour_hand")
    hour_hand.color("black", "black")

    min_hand = Turtle()
    min_hand.shape("min_hand")
    min_hand.color("blue1", "blue1")
```



```
# Python Clock Animation Demo (3)
```

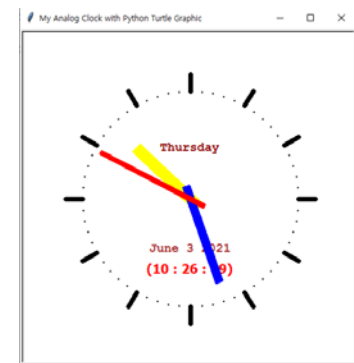
```
sec_hand = Turtle()  
sec_hand.shape("sec_hand")  
sec_hand.color("red", "red")  
  
for hand in sec_hand, min_hand, hour_hand:  
    hand.resizemode("user")  
    hand.shapesize(1, 1, 3)  
    hand.speed(0)  
ht()  
writer = Turtle()  
#writer.mode("logo")  
writer.ht()  
writer.pu()  
writer.bk(85)
```

```
def getWeekDayString(t):  
    weekday_name = ["Monday", "Tuesday", "Wednesday",  
                    "Thursday", "Friday", "Saturday", "Sunday"]  
    return weekday_name[t.weekday()]
```

Python Clock Animation Demo (4)

```
def getDateString(date):
    month_name = ["Jan.", "Feb.", "Mar.", "Apr.", "May", "June",
                  "July", "Aug.", "Sep.", "Oct.", "Nov.", "Dec."]
    yr = date.year
    mn = month_name[date.month - 1]
    dy = date.day
    return "%s %d %d" % (mn, dy, yr)

def tick():
    t = datetime.today()
    sec = t.second + t.microsecond*0.000001
    minute = t.minute + sec/60.0
    hour = t.hour + minute/60.0
    try:
        tracer(False) # Terminator can occur here
        writer.clear()
        writer.home()
        writer.pencolor("darkred")
        writer.forward(65)
        writer.write(getWeekDayString(t),
                    align="center", font=("Courier", 14, "bold"))
        writer.back(150)
        writer.pencolor("brown")
        writer.write(getDateString(t),
                    align="center", font=("Courier", 14, "bold"))
```

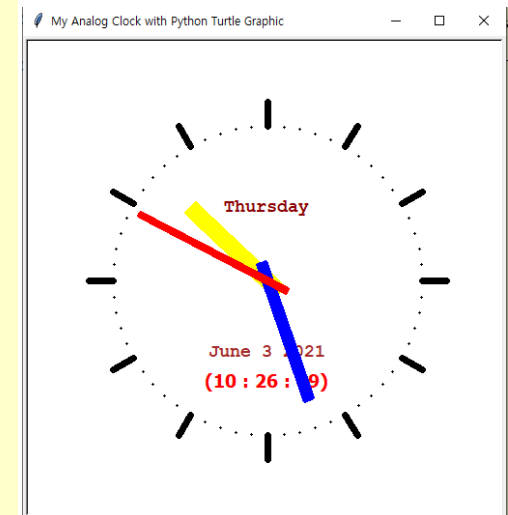


Python Clock Animation Demo (3)

```
writer.back(30)
hhmmss = "(%2d : %2d : %2d)"%(hour, minute, sec)
writer.pencolor("red")
writer.write(hhmmss, align="center", font=("Tahoma", 14, "bold"))
writer.forward(115)
tracer(True)
sec_hand.setheading(6*sec + 90) # or here
min_hand.setheading(6*minute + 90)
hour_hand.setheading(30*hour + 90)
tracer(True)
ontimer(tick, 100)
except Terminator:
    pass # turtle demo user pressed STOP
```

```
def main():
    tracer(False)
    setup()
    tracer(True)
    tick()
    return "Analog clock demo"
```

```
if __name__ == "__main__":
    mode("logo")
    turtle.setup(500, 500)
    turtle.title("My Analog Clock with Python")
    msg = main()
    #print(msg)
    mainloop()
```



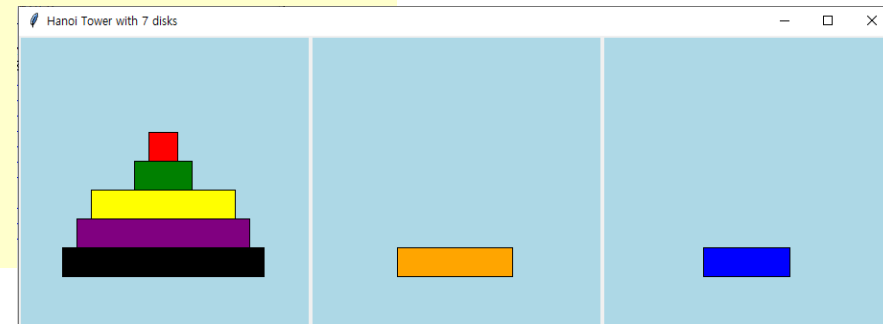
tkinter 기반 애니메이션 - 하노이 탑

하노이 탑 (1)

```
# tkinter animation of HanoiTower with canvas.update() and time.sleep() (1)

import turtle
from tkinter import *
import time

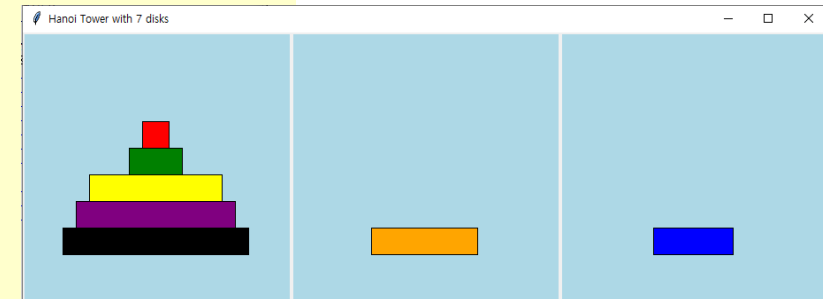
Disk_thickness = 30
Canvas_Width = 300
Canvas_Height = 300
Colors = ["black", "red", "green", "blue", "orange", "yellow", "purple"]
class Disk(object):
    def __init__(self, size, length, color):
        self.diameter= size
        self.length = length
        self.color = color
    def setTower(self, tower):
        self.tower = tower
    def __str__(self):
        return "Disk_{:>3}".format(self.diameter)
```



하노이 탑 (2)

tkinter animation of HanoiTower with canvas.update() and time.sleep() (2)

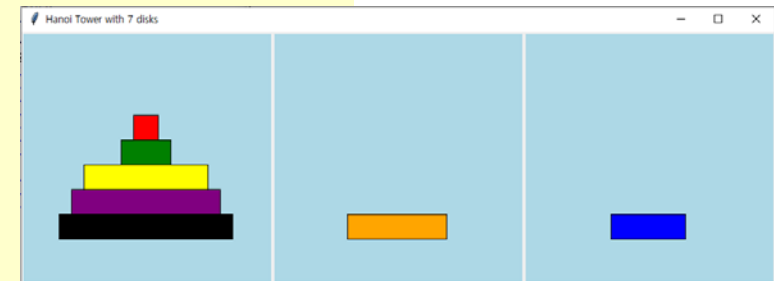
```
class Tower(object):
    def __init__(self, name, canvas, cv_id):
        self.name = name
        self.disks = []
        self.canvas = canvas
        self.cv_id = cv_id
    def addDisk(self, disk):
        self.disks.append(disk)
    def getLength(self):
        return len(self.disks)
    def top(self):
        if len(self.disks) <= 0:
            return None
        disk = self.disks[-1]
        if disk == None:
            return None
        else:
            return disk.diameter
    def pop(self):
        if len(self.disks) > 0:
            disk = self.disks[-1]
            self.disks.pop(-1)
            return disk
        else:
            return None
```



하노이 탑 (3)

tkinter animation of HanoiTower with canvas.update() and time.sleep() (3)

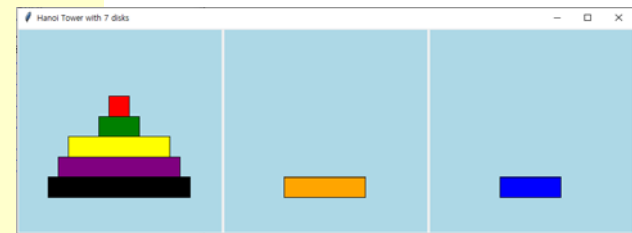
```
def drawTower(self):
    cv = self.canvas
    x0 = Canvas_Width // 2
    y0 = Canvas_Height - 50
    #print("{} is drawing disks with x0({}), y0({})"\
    #.format(self.name, x0, y0), end=' ')
    for i in range(len(self.disks)):
        disk = self.disks[i]
        if disk == None:
            continue
        #print("{}".format(disk.diameter), end=' ')
        cv.create_rectangle(x0 - disk.diameter//2, y0 - i*disk.length, x0\
            + disk.diameter//2, y0 - (i+1)*disk.length, fill=disk.color)
    #print()
def __str__(self):
    r_str = ""
    r_str += self.name + " : "
    if len(self.disks) <= 0:
        r_str += "Empty"
        return r_str
    for i in range(len(self.disks)):
        disk = self.disks[i]
        if disk == None:
            continue
        r_str += str(self.disks[i].diameter) + " "
    return r_str
```



하노이 탑 (4)

tkinter animation of HanoiTower with canvas.update() and time.sleep() (4)

```
def HanoiTower(n, tower_from, tower_tmp, tower_to):
    global Tower_1, Tower_2, Tower_3, window
    if (n == 1):
        if tower_from.getLength() > 0:
            disk = tower_from.pop()
            tower_to.addDisk(disk); disk.setTower(tower_to)
            print("{} is moved from({}) to({})"\
                  .format(disk, tower_from.name, tower_to.name))
            #print(Tower_1); print(Tower_2); print(Tower_3)
            tower_from.canvas.delete("all"); tower_from.drawTower()
            tower_to.canvas.delete("all"); tower_to.drawTower()
            tower_from.canvas.update(); tower_to.canvas.update(); time.sleep(1)
        else:
            HanoiTower(n-1, tower_from, tower_to, tower_tmp)
            if tower_from.getLength() > 0:
                disk = tower_from.pop()
                tower_to.addDisk(disk); disk.setTower(tower_to)
                print("{} is moved from({}) to({})"\
                      .format(disk, tower_from.name, tower_to.name))
                #print(Tower_1); print(Tower_2); print(Tower_3)
                tower_from.canvas.delete("all"); tower_from.drawTower()
                tower_to.canvas.delete("all"); tower_to.drawTower()
                tower_from.canvas.update(); tower_to.canvas.update(); time.sleep(1)
            HanoiTower(n-1, tower_tmp, tower_from, tower_to)
```



하노이 탑 (5)

tkinter animation of HanoiTower with canvas.update() and time.sleep() (5)

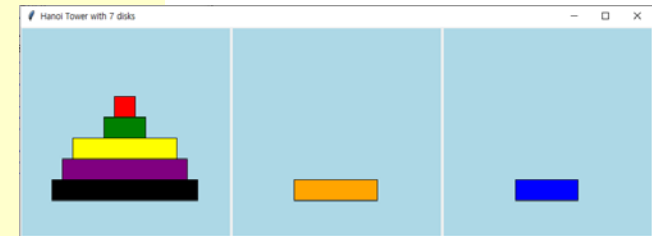
```
def initHanoiTower(n):
    global Tower_1, Tower_2, Tower_3, window
    window = Tk()
    window.title("Hanoi Tower with {} disks".format(n))
    canvas_1 = Canvas(window, bg="light blue",\
                      width=Canvas_Width, height= Canvas_Height)
    canvas_1.pack(side="left")

    canvas_2 = Canvas(window, bg="light blue",\
                      width=Canvas_Width, height= Canvas_Height)
    canvas_2.pack(side="left")

    canvas_3 = Canvas(window, bg="light blue",\
                      width=Canvas_Width, height= Canvas_Height)
    canvas_3.pack(side="left")

    Tower_1 = Tower("Tower_1", canvas_1, 1)
    Tower_2 = Tower("Tower_2", canvas_2, 2)
    Tower_3 = Tower("Tower_3", canvas_3, 3)

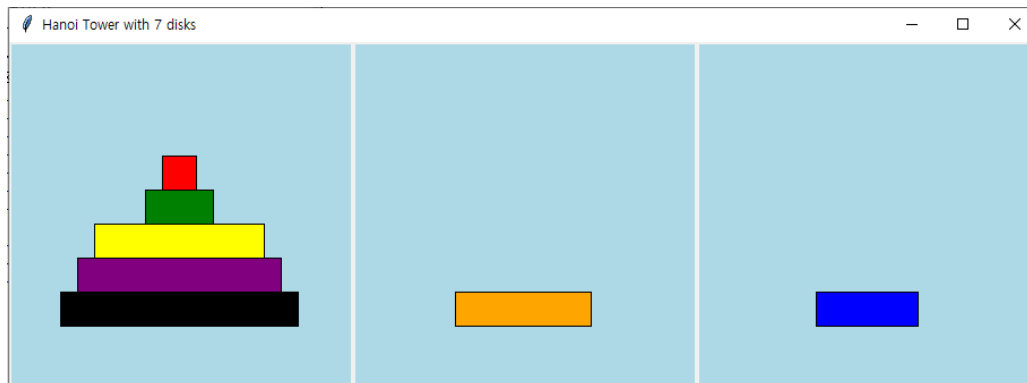
    for i in range(n, 0, -1):
        disk = Disk(i*Canvas_Width//10, Disk_thickness, Colors[i%len(Colors)])
        Tower_1.addDisk(disk)
```



하노이 탑 (6)

tkinter animation of HanoiTower with canvas.update() and time.sleep() (6)

```
#-----  
num_disks = 7  
if __name__ == "__main__":  
    #time.sleep(5)  
    initHanoiTower(num_disks)  
    HanoiTower(num_disks, Tower_1, Tower_2, Tower_3)
```



```
Disk_30 is moved from(Tower_1) to(Tower_3)  
Disk_60 is moved from(Tower_1) to(Tower_2)  
Disk_30 is moved from(Tower_3) to(Tower_2)  
Disk_90 is moved from(Tower_1) to(Tower_3)  
Disk_30 is moved from(Tower_2) to(Tower_1)  
Disk_60 is moved from(Tower_2) to(Tower_3)  
Disk_30 is moved from(Tower_1) to(Tower_3)  
Disk_120 is moved from(Tower_1) to(Tower_2)  
Disk_30 is moved from(Tower_3) to(Tower_2)  
Disk_60 is moved from(Tower_3) to(Tower_1)  
Disk_30 is moved from(Tower_2) to(Tower_1)  
Disk_90 is moved from(Tower_3) to(Tower_2)  
Disk_30 is moved from(Tower_1) to(Tower_3)  
Disk_60 is moved from(Tower_1) to(Tower_2)  
Disk_30 is moved from(Tower_3) to(Tower_2)  
Disk_150 is moved from(Tower_1) to(Tower_3)  
Disk_30 is moved from(Tower_2) to(Tower_1)  
Disk_60 is moved from(Tower_2) to(Tower_3)  
Disk_30 is moved from(Tower_1) to(Tower_3)  
Disk_90 is moved from(Tower_2) to(Tower_1)  
Disk_30 is moved from(Tower_3) to(Tower_2)  
Disk_60 is moved from(Tower_3) to(Tower_1)  
Disk_30 is moved from(Tower_2) to(Tower_1)  
Disk_120 is moved from(Tower_2) to(Tower_3)  
Disk_30 is moved from(Tower_1) to(Tower_3)  
Disk_60 is moved from(Tower_1) to(Tower_2)  
Disk_30 is moved from(Tower_3) to(Tower_2)  
Disk_90 is moved from(Tower_1) to(Tower_3)  
Disk_30 is moved from(Tower_2) to(Tower_1)  
Disk_60 is moved from(Tower_2) to(Tower_3)  
Disk_30 is moved from(Tower_1) to(Tower_3)
```

pyaudio 모듈 응용 - 전자피아노 구현

Keyboard

◆ Standard Keyboard



Frequency of Music Code

◆ Frequency of Music Codes

	1 Octave	2 Octave	3 Octave	4 Octave	5 Octave	6 Octave	7 Octave	8 Octave
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133



pyaudio 모듈

◆ pyaudio 모듈의 설치

- pyaudio 모듈은 pipwin을 사용하여 설치
- python -m pip install --upgrade pipwin
- pipwin install pyaudio

◆ pipwin을 사용한 pyaudio 설치시 발생 가능 문제

- pipwin 사용시 설치 경로에 ASCII 코드가 아닌 문자 (예: 한글문자)가 포함 될 경우 pipwin 패키지가 경로를 정상적으로 인식하지 못하여 pyaudio 설치가 불가능한 경우가 있었음

```
C:\Users\관리자> pipwin install pyaudio
SyntaxError: Non-UTF-8 code starting with '\xb1' in file C:\Users\관리자\AppData\Local\Programs\Python\Python39\Scripts\pipwin-script.py on line 1, but no encoding declared: see http://python.org/dev/peps/pep-0263/ for details
C:\Users\관리자>
```

- 해결 방법:
 - 파이썬이 설치되는 경로상에 한글 폴더/디렉토리 이름이 포함되지 않도록 설정하여 파이썬을 재설치
 - MS-Windows의 사용자 이름을 영문으로 설정

pyaudio 모듈 기반 전자피아노

```
# Simple Electric Piano with pyaudio, tkinter and numpy (1)
import pyaudio
import numpy as np
from tkinter import *
```

```
lower_cases = \
[262, 196, 165, 330, 659, 349, 392, 440, 1046, 494,
 # a(C4), b, c, d, e, f, g, h, i, j,
 523, -1, 247, 220, -1, -1, 523, 698, 294, 784,
 # k, l, m, n, o, p, q(C5), r, s, t,
 988, 175, 587, 147, 880, 131]
# u, v, w, x, y, z (C3)

upper_cases = \
[2093, 1568, 1318, 2637, 5274, 2794, 3136, 3520, -1, 3951,
 # A(C7), B, C, D, E, F, G, H, I, J,
 4186, -1, 1975, 1760, -1, -1, 4186, 5587, 2349, 6272,
 # K, L, M, N, O, P, Q(C8), R, S, T,
 7902, 1397, 4969, 1175, 7040, 1046]
# U, V, W, X, Y, Z(C6)
```

	1 Octave	2 Octave	3 Octave	4 Octave	5 Octave	6 Octave	7 Octave	8 Octave
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133



pyaudio 모듈 기반 전자피아노

```
# Simple Electric Piano with pyaudio, tkinter and numpy (2)

def pyaudio_init():
    global pa, stream
    volume=.5
    pa = pyaudio.PyAudio()
    stream = pa.open(format=pyaudio.paFloat32, channels=1, rate=48000, output=True)

def pyaudio_play(freq, duration, volume):
    global pa, stream
    rate=48000
    sample = (np.sin(2*np.pi * \
        np.arange(rate * duration)*freq/rate)).astype(np.float32)
    stream.write(volume * sample)

def pyaudio_close():
    global pa, stream
    stream.stop_stream()
    stream.close()
    pa.terminate()
```



pyaudio 모듈 기반 전자피아노

Simple Electric Piano with pyaudio, tkinter and numpy (2)

```
def keyEvent(event):
    #key = event.keycode
    if len(event.char) == 0:
        return
    key = ord(event.char)
    if (ord('a') <= key <= ord('z')):
        freq = lower_cases[key - ord('a')]
    elif(ord('A') <= key <= ord('Z')):
        freq = upper_cases[key - ord('A')]
    else: # undefined key
        return
    #print("Keyboard_Event, Pressed Key : {0}, freq {1}".format(chr(key), freq))
    if freq > 40:
        print("keyChar({}), freq({})".format(chr(key), freq), end=" ")
        pyaudio_play(freq, 1, 0.5)

# main loop
def main():
    pyaudio_init()
    window = Tk()
    window.bind("<Key>", keyEvent)
    window.mainloop()
    pyaudio_close()

if __name__ == "__main__":
    main()
```



bottle과 paste 기반의 Web Server 구현

Web Server의 실행, Web Client 실행

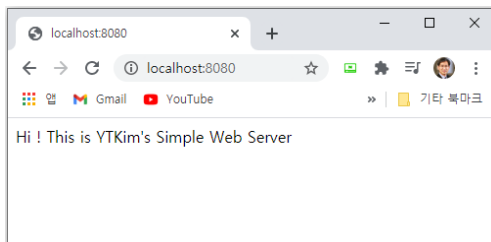
◆ index.html

```
<html>
<body>
Hi !
This is YTKim's Simple Web Server
</body>
</html>
```

◆ index.html 준비 후, Command 창에서 다음 명령어 실행

```
C:\MyWeb>python -m http.server 8080
Serving HTTP on :: port 8080 (http://[::]:8080/) ...
::1 - - [02/Dec/2020 13:44:00] "GET / HTTP/1.1" 304 -
```

◆ Client 실행: Web browser에서 **http://localhost:8080**



Python 프로그램으로 Web Server 접속

◆ web_app.py

```
# Simple Web Application with Python
import http.server
import socketserver

handler = http.server.SimpleHTTPRequestHandler

with socketserver.TCPServer(('', 8080), handler) as httpd:
    print("Server listenig on port 8080 ...")
    httpd.serve_forever()
```

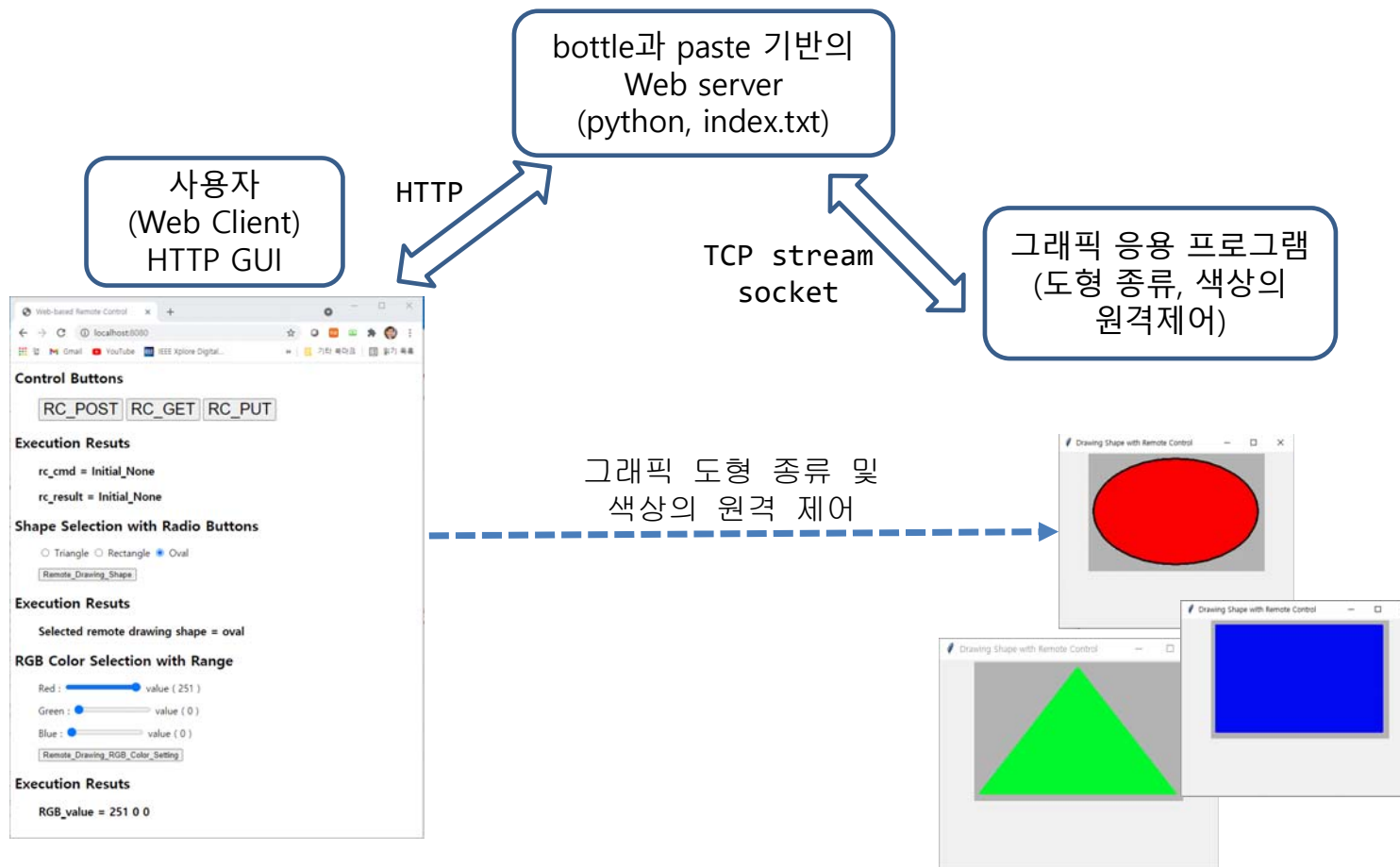
◆ 실행 결과

```
>>>
===== RESTART: C:\MyWeb\web_app.py =====
Server listenig on port 8080 ...
127.0.0.1 - - [02/Dec/2020 13:31:18] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [02/Dec/2020 13:31:18] code 404, message File not found
127.0.0.1 - - [02/Dec/2020 13:31:18] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [02/Dec/2020 13:31:46] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [02/Dec/2020 13:31:49] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [02/Dec/2020 13:35:54] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [02/Dec/2020 13:36:02] "GET / HTTP/1.1" 304 -
```



웹서버를 통한 그래픽 원격 제어

◆ 구성도



Simple Python-based Web Server with bottle and paste

```
# web_remote_control_server.py - Simple Web Application with Python (1)
# for web server, bottle module is used
# for multi-threading, paste module is used
# So, bottle and paste modules must be installed before execution !!

import turtle
import threading
import socket
from queue import Queue
from _thread import *
from bottle import route, run, get, post, response, static_file, request
from Remote_controlled_drawing import *

Port_Num = 18080
hostname = socket.gethostname()
hostAddr = socket.gethostbyname(hostname)
servSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
servSock.bind((hostAddr, Port_Num))
print("Web server ({} ) is waiting a client to connect ....".format(hostAddr))
servSock.listen(1)
sock_conn, cliAddr = servSock.accept()
print("Web Server is connected to the RC_Drawing client ({} )...".format(cliAddr))
```

web_remote_control_server.py - Simple Web Application with Python (2)

@route('/') # invoked

def do_root_index():

```
    print("do_root_index('/') is invoked ==> ./index.html will be executed ...")
    return static_file("index.html", root=".")
```

@route('/demo') # invoked by localhost:8080/demo

def do_demo():

```
    print("do_demo('/demo') was invoked ...")
    return "<H2>do_demo('/demo') was invoked ...</H2>"
```

@route('/login', method='GET')

def login():

```
    return '''
        <form action="/login" method="post">
        Username: <input name="username" type="text"/>
        Password: <input name="password" type="password" />
        <input value="Login" type="submit" />
        </form>
    '''
```

@route('/login', method='POST')

def do_login():

```
    username = request.forms.get('username')
    passwd = request.forms.get('password')
    return "login (user_name = {}, passwd = {})".format(username, passwd)
```

Username:

Password:



web_remote_control_server.py - Simple Web Application with Python (3)

```
@route('/remote_control', method='POST')
```

```
def rc_POST():
```

```
    recv_cmd=request.forms.get('command')
    print("rc_POST({}) was invoked ...".format(recv_cmd))
    return_msg = "result of {}".format(recv_cmd)
    print("return_msg = {}".format(return_msg))
    return return_msg
```

```
@route('/remote_control', method='GET')
```

```
def rc_GET():
```

```
    print("rc_GET() was invoked ...")
    return_value = '7'
    return_msg = "result of RC_GET = {}".format(return_value)
    print("return_msg = {}".format(return_msg))
    return return_msg
```

```
@route('/remote_control', method='PUT')
```

```
def rc_PUT():
```

```
    recv_cmd=request.forms.get('put_value')
    print("rc_PUT({}) was invoked ...".format(recv_cmd))
    return_msg = "result of {}".format(recv_cmd)
    print("return_msg = {}".format(return_msg))
    return return_msg
```



```
# web_remote_control_server.py - Simple Web Application with Python (3)
```

```
@route('/remote_drawing_shape', method='POST')
```

```
def remote_drawing_shape_POST():
```

```
    shape_name = request.forms.get('remote_drawing_shape')
    print("Web Server::remote_drawing shape({}) was invoked ...".format(shape_name))
    msg_to_rc_drawing = "change_shape " + shape_name
    sock_conn.send(bytes(msg_to_rc_drawing.encode()))
    return_msg = "Web server::remote_drawn_shape({})".format(shape_name)
    print("return_msg = {}".format(return_msg))
    return return_msg
```

```
@route('/remote_drawing_color', method='POST')
```

```
def rgb_color_set_POST():
```

```
    rgb_value=request.forms.get('rgb_value')
    print("/remote_drawing - rgb_color_set_POST({}) was invoked ...".format(rgb_value))
    msg_to_rc_drawing = "change_color " + rgb_value
    sock_conn.send(bytes(msg_to_rc_drawing.encode()))
    return_msg = "Web server::rgb_color_set ({} )".format(rgb_value)
    print("return_msg = {}".format(return_msg))
    return return_msg
```

```
#-----
```

```
run(host='', port=8080, server='paste') # using Paste multi-thread web-server module
```



index.html

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Web-based Remote Control</title>
  <meta name="viewport" content="width=200, initial-scale=1, maximum-
scale=1">
</head>

  <script type="text/javascript">
    function rc_post(value) {
      var req_post = new XMLHttpRequest();
      var cmd_msg = "command=" + value;
      req_post.open('POST', '/remote_control', false);
      // 'false' makes the request synchronous
      req_post.setRequestHeader('Content-Type',
        'application/x-www-form-urlencoded');
      req_post.setRequestHeader('Content-Length', cmd_msg.length);
      req_post.setRequestHeader('Connection', 'close');
      req_post.send(cmd_msg);
      document.getElementById('rc_cmd').innerHTML = value;
      var res_msg = req_post.responseText;
      //alert("rc_post (" + value + ")_result = " + res_msg);
      document.getElementById('rc_result').innerHTML = res_msg;
    }
  </script>
```



```

<script type="text/javascript">
function rc_put(value) {
    var req_put = new XMLHttpRequest();
    var rc_cmd = "put value=" + value;
    //alert("rc_put (" + value + ") invoked => rc_cmd = (" + rc_cmd + ")");
    req_put.open('PUT', '/remote_control', false);
    // 'false' makes the request synchronous
    req_put.setRequestHeader('Content-Type',
        'application/x-www-form-urlencoded');
    req_put.setRequestHeader('Content-Length', rc_cmd.length);
    req_put.setRequestHeader('Connection', 'close');
    req_put.send(rc_cmd);
    document.getElementById('rc_cmd').innerHTML = value;
    var res_msg = req_put.responseText;
    //alert("rc_put (" + value + ") result = " + res_msg);
    document.getElementById('rc_result').innerHTML = res_msg;
}
</script>

```




```

<script type="text/javascript">
    function Remote_Drawing_Shape() {
        //alert("Remote_Drawing_Shape")
        var radio_btn = document.getElementsByName("remote_drawing_shape")
        var radio_btn_check = 0;
        var selected_shape;
        for (var i=0; i<radio_btn.length; i++){
            if (radio_btn[i].checked == true) {
                selected_shape = radio_btn[i].value;
            }
        }
        document.getElementById('remote_drawing_shape').innerHTML = selected_shape;
        //alert("remote_drawing_shape (" + selected_shape + ")")
        var req_post = new XMLHttpRequest();
        var remote_drawing_cmd = "remote_drawing_shape=" + selected_shape;
        req_post.open('POST', '/remote_drawing_shape', false);
        // 'false' makes the request synchronous
        req_post.setRequestHeader('Content-Type',
            'application/x-www-form-urlencoded');
        req_post.setRequestHeader('Content-Length', remote_drawing_cmd.length);
        req_post.setRequestHeader('Connection', 'close');
        req_post.send(remote_drawing_cmd);

        var res_msg = req_post.responseText;
        //alert("remote_drawing_shape_(" + selected_shape + ")_result => "
        //    + res_msg);
    }
</script>

```



```

<script type="text/javascript">
    function Remote_RGB_Color_Set() {
        var req_post = new XMLHttpRequest();
        var rgb_value = document.getElementById('RGB_value').innerHTML
        var rc_rgb_cmd = "rgb_value=" + rgb_value;
        //alert("RGB_btn () invoked => rc_rgb_cmd = (" + rc_rgb_cmd + ")");
        req_post.open('POST', '/remote_drawing_color', false);
        // 'false' makes the request synchronous
        req_post.setRequestHeader('Content-Type',
            'application/x-www-form-urlencoded');
        req_post.setRequestHeader('Content-Length', rc_rgb_cmd.length);
        req_post.setRequestHeader('Connection', 'close');
        req_post.send(rc_rgb_cmd);
        document.getElementById('rc_rgb_cmd').innerHTML = rgb_value;
        var res_msg = req_post.responseText;
        //alert("RGB_btn_(" + rgb_value + ")_result => " + res_msg);
    }
</script>

<script type="text/javascript">
    function Update_RGB_value()
    {
        //alert("Update_RGB_valued ()")
        var rd = document.getElementById('RGB_red_value').innerHTML;
        var gr = document.getElementById('RGB_green_value').innerHTML;
        var bl = document.getElementById('RGB_blue_value').innerHTML;
        var rgb = rd + " " + gr + " " + bl
        //alert("Update_RGB_valued (" + rgb + ")")
        document.getElementById('RGB_value').innerHTML = rgb
    }
</script>

```



```

<script type="text/javascript">
    function Set_RGB_Red(value)
    {
        //alert("Set_RGB_Red (" + value + ")")
        document.getElementById('RGB_red_value').innerHTML = value
        Update_RGB_value()
    }
</script>
<script type="text/javascript">
    function Set_RGB_Green(value)
    {
        //alert("Set_RGB_Green (" + value + ")")
        document.getElementById('RGB_green_value').innerHTML = value
        Update_RGB_value()
    }
</script>
<script type="text/javascript">
    function Set_RGB_Blue(value)
    {
        //alert("Set_RGB_Blue (" + value + ")")
        document.getElementById('RGB_blue_value').innerHTML = value
        Update_RGB_value()
    }
</script>

```



```

<body>
  <H2>Control Buttons</H2>
  <div align="left" style="margin:0 0 10px 10px">
    <ul>
      <input type="button" style="font-size:20pt; width:70; height:60
        value="RC_POST" onClick="rc_post('RC_POST');">
      <input type="button" style="font-size:20pt; width:70; height:60
        value="RC_GET" onClick="rc_get('RC_GET');">
      <input type="button" style="font-size:20pt; width:70; height:60
        value="RC_PUT" onClick="rc_put('RC_PUT');">
    </ul>
  </div>
  <H2>Execution Results</H2>
  <div>
    <ul>
      <H3> rc_cmd  = <span id="rc_cmd"> Initial_None </span> </H3>
      <H3> rc_result = <span id="rc_result"> Initial_None </span> </H3>
    </ul>
  </div>
  <H2>Shape Selection with Radio Buttons</H2>
  <div>
    <ul>
      <input type="radio" id="triangle" name="remote_drawing_shape"
        value="triangle" checked>
      <label for="triangle">Triangle</label>
      <input type="radio" id="rectangle" name="remote_drawing_shape"
        value="rectangle" >
      <label for="rectangle">Rectangle</label>
      <input type="radio" id="oval" name="remote_drawing_shape"
        value="oval">
      <label for="oval">Oval</label>
    </ul>
  </div>

```

Control Buttons

RC_POST RC_GET RC_PUT

Execution Results

rc_cmd = Initial_None

rc_result = Initial_None

Shape Selection with Radio Buttons

☐ Triangle ☒ Rectangle ☐ Oval

Remote_Drawing_Shape

Execution Results

Selected remote drawing shape = rectangle



```

<div>
  <ul>
    <input type="button" style=font-size:10pt; width:70; height:60
      value="Remote_Drawing_Shape" onClick="Remote_Drawing_Shape();">
    </ul>
  </div>
<H2>Execution Results</H2>
<div>
  <ul>
    <H3> Selected remote drawing shape = <span id="remote_drawing_shape">
      Initial_None </span> </H3>
    </ul>
  </div>

<H2>RGB Color Selection with Range</H2>
<div>
  <ul>
    <label for="Red"> Red : </label>
    <input type="range" id="red" name="RGB_red_value" min="0" max="255"
      value="0" onChange="Set_RGB_Red(this.value);">
    <label for="value"> value ( <span id="RGB_red_value"> 0 </span> ) </label>
  </ul>
  <ul>
    <label for="green">Green : </label>
    <input type="range" id="green" name="RGB_green_value" min="0" max="255"
      value="0" onChange="Set_RGB_Green(this.value);">
    <label for="value"> value ( <span id="RGB_green_value"> 0 </span> )
      </label>
  </ul>

```



```

<ul>
  <lable for="blue">Blue : </lable>
  <input type="range" id="blue" name="RGB_blue_value" min="0" max="255"
    value="0" onchange="Set_RGB_Blue(this.value);">
  <label for="value"> value ( <span id="RGB_blue_value"> 0 </span> ) </label>
</ul>
<ul>
  <input type="button" style=font-size:10pt; width:70;height:60
    value="Remote_Drawing_RGB_Color_Setting" onClick="Remote_RGB_Color_Set();">
</ul>
</div>
<H2>Execution Resuts</H2>
<div>
  <ul>
    <H3> RGB_value  = <span id="RGB_value"> 0 0 0 </span> </H3>
  </ul>
</div>
</body>
</html>

```

Control Buttons

RC_POST RC_GET RC_PUT

Execution Resuts

rc_cmd = Initial_None

rc_result = Initial_None



RemoteControlled_Drawing

```
# RemoteControlledDrawing (1)
import time
from threading import Thread
from tkinter import *
import socket

class RemoteControlledDrawing:
    def __init__(self):

        self.win = win = Tk()
        self.win.geometry("400x300")
        self.win.wm_title('Drawing Shape with Remote Control')

        frame = Frame(self.win)
        frame.pack()
        self.canvas = Canvas(self.win, bg="grey70", width=300, height=200)
        self.canvas.pack()

        self.red = self.green = self.blue = 0
        self.color_sequence = 0
        self.colors = [(255,255,255), (255,0,0), (0,255,0), (0,0,255), (0,0,0)]
        self.shapes = ["oval", "triangle", "rectangle"]
        self.shape_name = "oval" # initial/default shape_name
        self.shape = self.canvas.create_oval(10, 10, 290, 190, fill="white", width=3)

        rc_drawing_agent_thread = Thread(target=self.rc_drawing_agent, daemon=True)
        rc_drawing_agent_thread.start()
```



```

# RemoteControlledDrawing (2)
def rc_drawing_agent(self):
    Port_Num = 18080
    hostname = socket.gethostname()
    hostAddr = socket.gethostbyname(hostname)
    #servAddr_str = input("Server IP addr = ")
    servAddr_str = "165.229.185.251"
    cliSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    cliSock.connect((servAddr_str, Port_Num))
    servAddr = cliSock.getpeername()
    print("RC_drawing client is connected to server({})".format(servAddr))

    while True:
        recvMsg = cliSock.recv(100).decode()
        L_cmd = recvMsg.split()
        cmd = L_cmd[0]
        print("L_cmd = {}".format(L_cmd))
        if cmd == "change_shape":
            self.changeShape(L_cmd[1])
        if cmd == "change_color":
            red_str, green_str, blue_str = L_cmd[1], L_cmd[2], L_cmd[3]
            red, green, blue = int(red_str), int(green_str), int(blue_str)
            self.changeColor_RGB((red, green, blue))

    cliSock.close()

def changeColor(self):
    color_seq = (self.color_sequence) % len(self.colors)
    (rd, gr, bl) = self.colors[color_seq]
    self.red = rd
    self.green = gr
    self.blue = bl
    color = "#%02x%02x%02x"%(self.red, self.green, self.blue)
    self.canvas.itemconfig(self.shape, fill=color)
    self.color_sequence += 1

```




```
# RemoteControlledDrawing (3)
```

```
def changeColor_RGB(self, color_code):
```

```
    (rd, gr, bl) = color_code
    self.red = rd
    self.green = gr
    self.blue = bl
    color = "#%02x%02x%02x"%(self.red, self.green, self.blue)
    self.canvas.itemconfig(self.shape, fill=color)
    self.color_sequence += 1
```

```
def draw_oval(self, color_code):
```

```
    self.canvas.delete("all")
    (rd, gr, bl) = color_code
    color = "#%02x%02x%02x"%(rd, gr, bl)
    self.shape = self.canvas.create_oval(10, 10, 290, 190, outline=color, fill="white", width=3)
    self.canvas.itemconfig(self.shape, fill=color)
```

```
def draw_triangle(self, color_code):
```

```
    self.canvas.delete("all")
    (rd, gr, bl) = color_code
    color = "#%02x%02x%02x"%(rd, gr, bl)
    points = [10, 190, 290, 190, 150, 10]
    self.shape = self.canvas.create_polygon(points, outline=color, fill="white", width=3)
    self.canvas.itemconfig(self.shape, fill=color)
```

```
def draw_rectangle(self, color_code):
```

```
    self.canvas.delete("all")
    (rd, gr, bl) = color_code
    color = "#%02x%02x%02x"%(rd, gr, bl)
    points = [10, 190, 290, 190, 290, 10, 10, 10]
    self.shape = self.canvas.create_polygon(points, outline=color, fill="white", width=3)
```



```
# RemoteControlledDrawing (4)

def changeShape(self, shape_name):
    if shape_name in self.shapes:
        self.shape_name = shape_name
    else:
        self.shape_name = "oval" # default shape
    color = (self.red, self.green, self.blue)
    if self.shape_name == "triangle":
        self.draw_triangle(color)
    elif self.shape_name == "rectangle":
        self.draw_rectangle(color)
    else:
        self.draw_oval(color)
```



```

# RemoteControlledDrawing (5)

def keyEvent(event):
    global app
    input_char = event.keysym
    print("keyEvent({})".format(input_char))
    if input_char == '3':
        app.changeShape("triangle")
    elif input_char == '4':
        app.changeShape("rectangle")
    elif input_char == '0':
        app.changeShape("oval")
    elif input_char == 'r':
        app.changeColor_RGB((255, 0, 0))
    elif input_char == 'g':
        app.changeColor_RGB((0, 255, 0))
    elif input_char == 'b':
        app.changeColor_RGB((0, 0, 255))
    elif input_char == 'w':
        app.changeColor_RGB((255, 255, 255))
    elif input_char == 'k':
        app.changeColor_RGB((0, 0, 0))
    else:
        app.changeColor_RGB((0, 0, 0))

if __name__ == "__main__":
    global app

    app = RemoteControlledDrawing()
    app.win.bind("<Key>", keyEvent)

    app.win.mainloop()

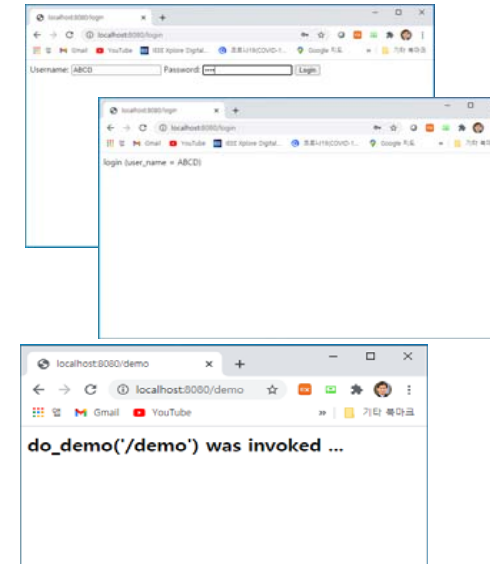
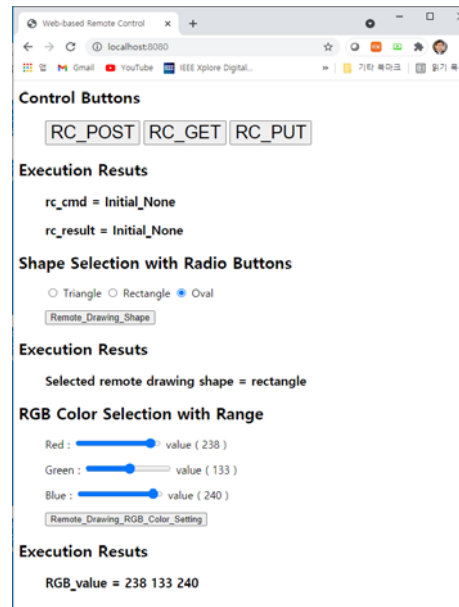
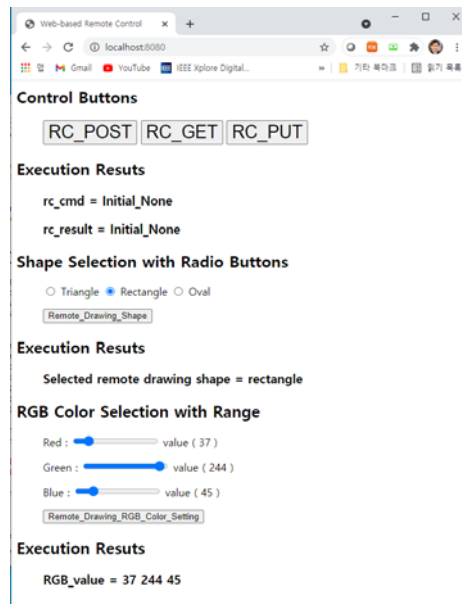
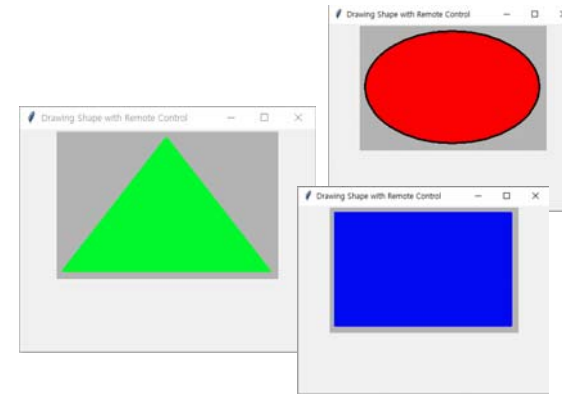
```



Web Browser에서의 실행결과

◆ Web Browser에서 실행

- <http://localhost:8080/login>
- <http://localhost:8080/demo>
- <http://localhost:8080>

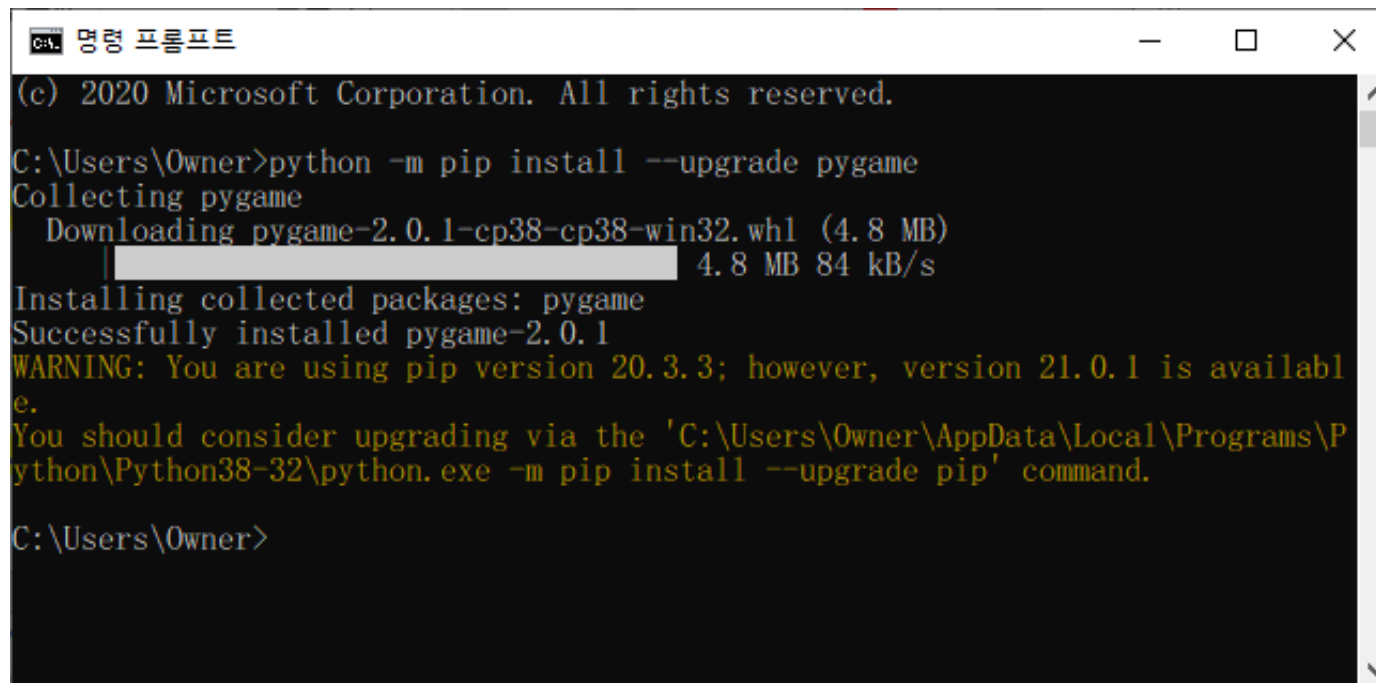


pygame 모듈

pygame 모듈 설치

◆ pygame 모듈 설치

> python -m pip install --upgrade pygame



```
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Owner>python -m pip install --upgrade pygame
Collecting pygame
  Downloading pygame-2.0.1-cp38-cp38-win32.whl (4.8 MB)
    | 4.8 MB 84 kB/s
Installing collected packages: pygame
Successfully installed pygame-2.0.1
WARNING: You are using pip version 20.3.3; however, version 21.0.1 is available.
You should consider upgrading via the 'C:\Users\Owner\AppData\Local\Programs\Python\Python38-32\python.exe -m pip install --upgrade pip' command.

C:\Users\Owner>
```

pygame module

◆ Methods of pygame module

- (<https://devdocs.io/pygame/>)

pygame method	설명
init()	pygame 초기화
screen_pad = display.set_mode(win_size)	pygame display 영역 (screen_pad)의 크기 설정
display.set_caption(title)	pygame display 영역의 제목 설정
image.load(file_name)	pygame에서 사용할 image를 지정된 파일로 부터 설치
transform.scale(img, (size))	pygame에서 사용할 image를 지정된 크기 (size_x, size_y)로 조정
screen_pad.blit(img, pos)	pygame display 영역 (screen_pad)에서 지정된 image를 지정된 위치에 표시
font = pygame.font.SysFont()	font 설정 font = pygame.font.SysFont("arial", 30, True, False)
clock = pygame.time.Clock()	pygame에서 사용되는 clock 생성
clock.tick(ticks_per_sec)	pygame에서 사용되는 clock의 초당 ticks 수를 설정
display.flip()	pygame display 영역 전체를 update
mixer.Sound()	pygame에서 사용되는 sound 객체 생성



Game용 이미지 준비

◆ Game용 무료 이미지 download

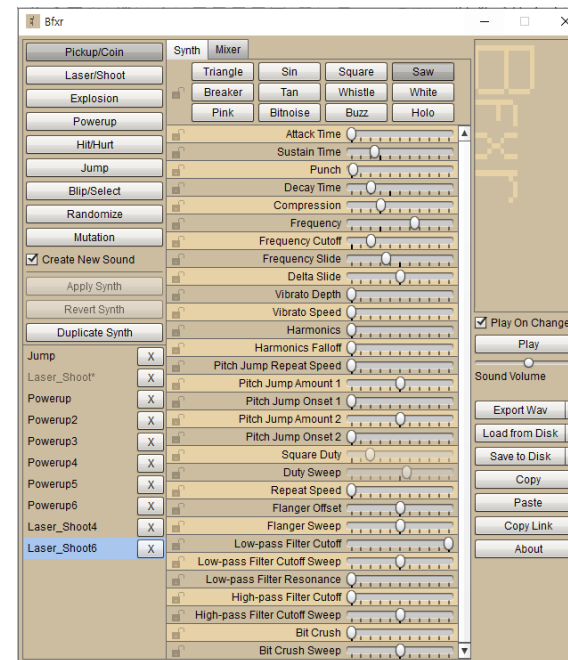
- <https://opengameart.org/>
- PNG 파일: Space ship, bullet, 외계인



Game용 효과음 생성

◆ Bfxr을 사용한 효과음 생성

- 설치: <https://www.bfxr.net/>
- 사용 방법: <https://steemit.com/gamedev/@adamak2k/tutorial-how-to-make-sound-effects-for-my-game>
- 다양한 효과음의 생성 및 저장



pygame 응용 - Space Battle

```
# PyGame - Space Battle (1)
import pygame
import random
import time
import os
import os.path

# stage 1. Definition of class Game_character and function crash
class Game_Character:
    def __init__(self):
        self.pos_x = 0
        self.pos_y = 0
        self.move = 0
    def set_img(self, file_name):
        if file_name[-3:] == ".png":
            self.img = pygame.image.load(file_name).convert_alpha()
        else:
            self.img = pygame.image.load(file_name)
        self.size_x, self.size_y = self.img.get_size()
    def set_size(self, sx, sy):
        self.img = pygame.transform.scale(self.img, (sx, sy))
        self.size_x, self.size_y = self.img.get_size()
    def show(self):
        game_pad.blit(self.img, (self.pos_x, self.pos_y))

def crash(a, b):
    if (a.pos_x - b.size_x <= b.pos_x) and (b.pos_x <= a.pos_x + a.size_x) and \
        (a.pos_y - b.size_y <= b.pos_y) and (b.pos_y <= a.pos_y + a.size_y):
        return True
    else:
        return False
```



```

# PyGame - Space Battle (2)

# stage 2. Initialization of Game
def init_game():
    global win_size, game_pad, font, sound_dir, img_dir
    pygame.init()
    pygame.font.init()
    pygame.mixer.init()
    sound_dir = "sound"
    img_dir = "img"

    win_size = [600, 900] # [width, height]
    game_pad = pygame.display.set_mode(win_size)
    game_title = "Space Battle"
    pygame.display.set_caption(game_title)
    font = pygame.font.SysFont("arial", 30, True, False)

# stage 3. Handling main events, loop
def run_game():
    Max_num_bullets = 200
    Limit_Missed = 20
    Bullet_Interval = 6

    clock = pygame.time.Clock()
    ss = Game_Character() # space ship
    ss.set_img(os.path.join(img_dir, "SpaceShip.png"))
    ss.set_size(80, 80)
    ss.pos_x = round(win_size[0]/2 - ss.size_x/2)
    ss.pos_y = win_size[1] - ss.size_y - 15
    ss.move = 5

```



PyGame - Space Battle (3)

```
move_left = False
move_right = False
space_key_on = False
L_bullet = [] # list of bullets
L_alien = [] # list of aliens

black = (0, 0, 0) # RGB code
white = (255, 255, 255)
bullet_interval = 0
killed = missed = bullet_used = 0
FPS = 60 # frames per second
stop_game = False
while stop_game == False:
    clock.tick(FPS)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            stop_game = True
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                move_left = True
            elif event.key == pygame.K_RIGHT:
                move_right = True
            elif event.key == pygame.K_SPACE:
                space_key_on = True
                bullet_interval = 0
        elif event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT:
                move_left = False
            elif event.key == pygame.K_RIGHT:
                move_right = False
            elif event.key == pygame.K_SPACE:
                space_key_on = False
```



PyGame - Space Battle (4)

```
if move_left == True:
    ss.pos_x -= ss.move
    if ss.pos_x <= 0:
        ss.pos_x = 0
elif move_right == True:
    ss.pos_x += ss.move
    if ss.pos_x >= win_size[0] - ss.size_x:
        ss.pos_x = win_size[0] - ss.size_x

if space_key_on == True and bullet_used < Max_num_bullets and\
bullet_interval % Bullet_Interval == 0:
    blt = Game_Character()
    blt.set_img(os.path.join(img_dir, "bullet.png"))
    blt.set_size(10, 15)
    blt.pos_x = round(ss.pos_x + ss.size_x/2 - blt.size_x/2)
    blt.pos_y = ss.pos_y - blt.size_y - 10
    blt.move = 15
    L_bullet.append(blt)
    sound_bullet = pygame.mixer.Sound(os.path.join(sound_dir, "Laser_Shoot"))
    pygame.mixer.Sound.play(sound_bullet)
    bullet_used += 1

if bullet_used >= Max_num_bullets:
    text = font.render("Out of bullets !!", True, (255, 0, 0))
    game_pad.blit(text, (10, 40))
    pygame.display.flip()

bullet_interval += 1
L_delete_blt = [] # list of characters/objects to be deleted
```



```
# PyGame - Space Battle (5)
```

```
    for i in range(len(L_bullet)):
        b = L_bullet[i]
        b.pos_y -= b.move
        if b.pos_y <= -b.size_y:
            L_delete_blt.append(i)
    L_delete_blt.reverse()
    for d in L_delete_blt:
        del L_bullet[d]

    if random.random() > 0.98:
        aa = Game_Character()
        aa.set_img(os.path.join(img_dir, "alien.png"))
        aa.set_size(40, 40)
        aa.pos_x = random.randrange(0, win_size[0] - aa.size_x - round(ss.size_x / 2))
        aa.pos_y = 10
        aa.move = 1
        L_alien.append(aa)

    L_delete_alien = []
    for i in range(len(L_alien)):
        a = L_alien[i]
        a.pos_y += a.move
        if a.pos_y >= win_size[1]:
            L_delete_alien.append(i)
            missed += 1
    L_delete_alien.reverse()
    for d in L_delete_alien:
        del L_alien[d]
```



PyGame - Space Battle (6)

```
L_delete_blt = []
L_delete_alien = []
for i in range(len(L_bullet)):
    for j in range(len(L_alien)):
        b = L_bullet[i]
        a = L_alien[j]
        if crash(b, a) == True:
            L_delete_blt.append(i)
            L_delete_alien.append(j)
            killed += 1
L_delete_blt = list(set(L_delete_blt))
L_delete_alien = list(set(L_delete_alien))
L_delete_blt.reverse()
L_delete_alien.reverse()

try:
    for db in L_delete_blt:
        del L_bullet[db]
    for da in L_delete_alien:
        del L_alien[da]
except:
    pass

for i in range(len(L_alien)):
    a = L_alien[i]
    if crash(a, ss) == True:
        stop_game = True
        text = font.render("Space ship is crashed !! Gameover", True, (255, 0, 0))
        game_pad.blit(text, (10, 60))
        pygame.display.flip()
        print("Game over with final score: killed ({}), missed ({}),\
            bullet_used ({})".format(killed, missed, bullet_used))
        time.sleep(5))
```



```
# PyGame - Space Battle (7)
```

```
    if missed >= Limit_Missed:
        stop_game = True
        text = font.render("Missed is over the limit !! Gameover", True, (255, 0, 0))
        game_pad.blit(text, (10, 60))
        pygame.display.flip()
        print("Game over with final score: killed ({}), missed ({}), \
              bullet_used ({}).format(killed, missed, bullet_used))
        time.sleep(5)
```

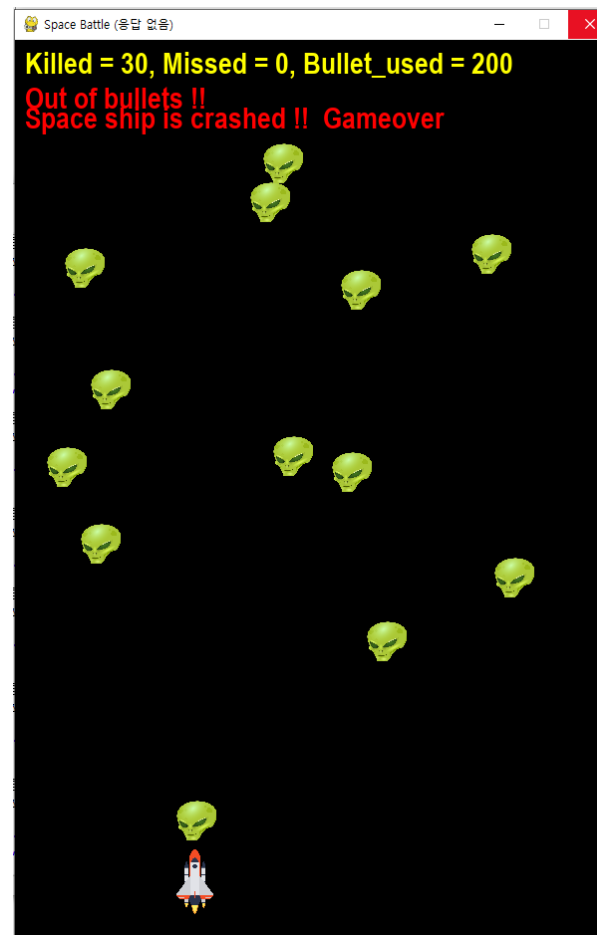
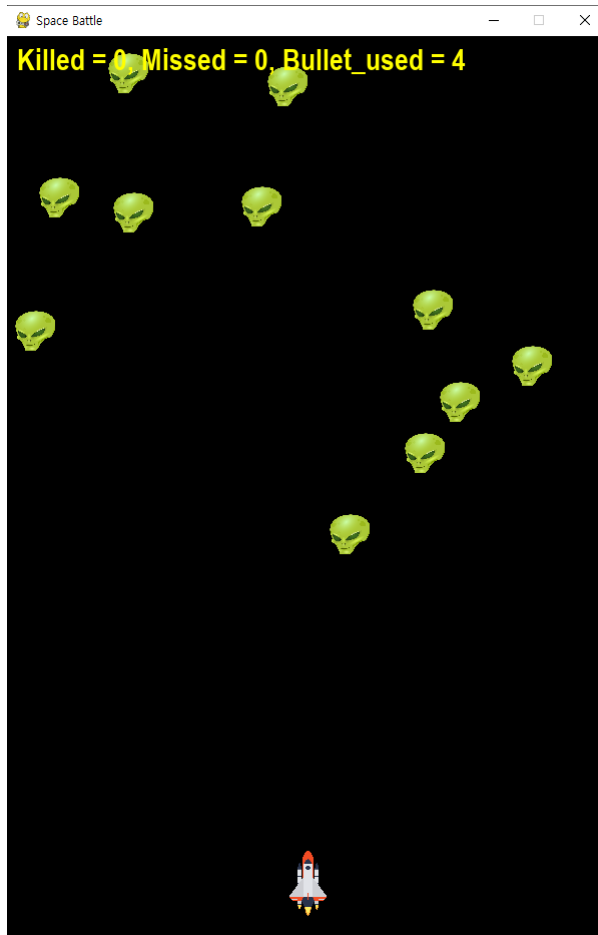
```
    game_pad.fill(black)
    ss.show()
    for blt in L_bullet:
        blt.show()
    for a in L_alien:
        a.show()
```

```
    text = font.render("Killed = {}, Missed = {}, Bullet_used = \
{}".format(killed, missed, bullet_used), True, (255, 255, 0))
    game_pad.blit(text, (10, 5))
    pygame.display.flip()
pygame.quit()
```

```
# main()
if __name__ == "__main__":
    init_game()
    run_game()
```



실행 결과

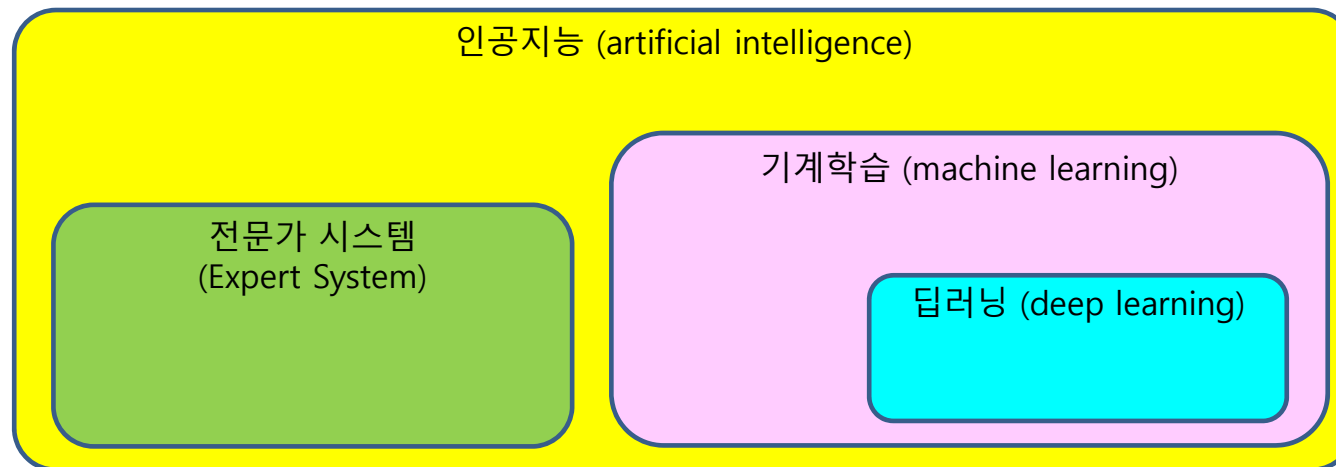


파이썬 기반 기계학습

인공지능, 기계학습, 딥러닝 (심화학습)

◆ 인공지능, 기계학습, 딥러닝 (심화학습)

- 인공지능 (artificial intelligence) : 인간처럼 학습하고 추론할 수 있는 소프트웨어 시스템 연구
- 기계학습 (machine learning) : 인공지능의 한 분야이며, 별도의 프로그래밍 없이 스스로 학습할 수 있는 소프트웨어 시스템 연구
- 딥러닝 (deep learning) : 인공 신경망 등을 사용하여 빅데이터로부터 스스로 학습할 수 있는 소프트웨어 시스템 연구



기계 학습 (machine learning)

◆ 기계 학습 (machine learning)

- 인공지능의 한 분야로 컴퓨터에 학습 기능을 부여하기 위한 연구
- 1959년 Arthur Samuel이 처음 사용
- 패턴 인식 및 계산 학습 이론에서 진화하여 컴퓨터가 주어진 데이터를 학습하는 알고리즘을 연구
- 학습할 수 있는 데이터가 많아지면 알고리즘 성능이 향상됨
- 기계학습 알고리즘은 항상 고정적인 의사결정을 하는 프로그램과 달리, 데이터 중심의 예측 또는 결정을 내릴 수 있음
- 기계 학습은 어떤 문제에 대하여 명시적 알고리즘을 설계하고, 프로그래밍하는 것이 어렵거나 불가능한 경우에 주로 사용됨
- 사용분야: spam E-mail filtering, 네트워크 침입자 검출, 광학문자 인식 (OCR), 필기체 인식, 컴퓨터 비전



기계학습의 종류

◆ 기계학습의 종류

기계학습 (Machine Learning) 알고리즘

지도 학습	비지도 학습	강화 학습
회귀 (regression) - Linear - Polynomial 의사 결정 트리 랜덤 포레스트 분류 - kNN - Trees - Logistic Regression - Naïve-Bayes - SVM Convolutional Neural Network (CNN)	클러스터링 - SVD - PCA - K-means	AutoEncoder Q-learning



지도학습, 비지도 학습, 강화학습

◆ 기계학습 유형별 특징

기계학습 유형	특징
지도학습 (supervised learning)	<ul style="list-style-type: none"> 교사에 의하여 주어진 예제 (샘플)과 정답 (레이블)을 제공받음 지도 학습의 목표는 입력을 출력에 매칭하는 일반적인 규칙(함수)를 학습 예를 들어 강아지와 고양이를 구분하는 문제인 경우, 강아지와 고양이에 대한 영상을 제공한 후, 교사가 어떤 영상이 강아지인지, 어떤 영상이 고양이인지 구분하여 정답을 알려줌
비지도학습 (unsupervised learning)	<ul style="list-style-type: none"> 외부에서 정답(레이블)이 주어지지 않고, 학습 알고리즘이 스스로 입력에서 어떤 구조를 발견하는 학습 비지도 학습을 사용하면 데이터에 숨겨져 있는 패턴을 발견할 수 있음
강화학습 (reinforcement learning)	<ul style="list-style-type: none"> 보상 및 처벌 형태로 학습 데이터가 주어짐 주로 차량 운전이나 상대방과의 경기와 같은 동적인 환경에서 프로그램의 행동에 대한 피드백만 제공되는 경우 예를 들어 바둑에서 어떤 수를 두어서 승리하였다면 보상이 주어지며, 실패하였다면 처벌이 주어짐 강화학습에서는 보상과 처벌을 통하여 학습이 이루어 짐

지도학습

◆ 지도학습 (supervised learning)

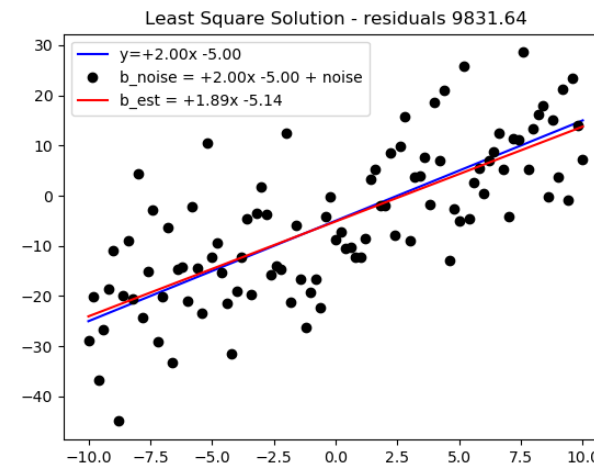
- 지도학습은 학습을 시키는 교사가 존재하는 학습방법
- 학습데이터에서 정답 (회귀: 출력값, 분류: 레이블)이 제공되므로 지도라는 용어를 사용함
- 지도학습에서는 입력을 결합하여 모델을 만들고, 이전에 보지 못한 데이터도 적절히 예측하는 방법을 학습 시킴
- 지도학습은 크게 회귀 (regression)과 분류 (classification)으로 구분

지도학습 유형	특 징
회귀 (regression)	<ul style="list-style-type: none"> ▪ 회귀는 주어진 입력-출력 값쌍을 학습한 후, 새로운 입력값이 들어왔을 때 합리적인 출력 값을 예측 ▪ 회귀에서는 학습시키는 데이터가 이산적인 아니고 연속적이며, 입력과 출력이 모두 실수 (real number)로 표현되며, 연속적인 값을 예측 ▪ 예: 입력값 (x)에 대한 출력값을 $y = f(x)$의 방정식으로 예측
분류 (classification)	<ul style="list-style-type: none"> ▪ 입력을 두 개 이상의 레이블 (유형)으로 분류하는 것 ▪ 해당 모델을 학습시킬 때 레이블을 제공하며, 올바른 레이블을 알려 줌 ▪ 학습이 끝나면 학습자가 한 번도 보지 못한 입력을 이들 레이블 중의 하나로 분류하는 시스템 ▪ 예: 스팸 필터링, 필기체 숫자 인식

선형 회귀 (linear regression)

◆ 선형 회귀

- 회귀는 입력 (x)와 출력 (y)값이 주어질 때, 입력에서 출력으로의 매핑 함수 $y = f(x)$ 를 학습하는 것
- 입력 (x)는 다차원일 수 있음
- 선형 모델 ($f(x) = mx + b$)을 사용하여 회귀문제를 풀 때 선형회귀 (linear regression)이라 함
- 선형회귀의 사용 예
 - 부모의 키와 자녀의 키의 상관관계
 - 면적에 따른 주택의 가격
 - 나이 (연령)에 따른 실업률 예측
 - 공부시간과 학업 성적의 관계
 - CPU속도와 프로그램 실행 시간관계



선형회귀에서 손실함수 최소화 방법

◆ 손실함수

- 선형 회귀에서 학습데이터 (x_1, x_2, x_3, \dots)를 사용한 예측에서 선형 모델 $f(x) = Wx + b$ 을 사용할 때,
- 손실함수 (loss function, cost function)는 실제 데이터 x_i 에 대한 $f(x_i)$ 와 예측된 선형 함수의 직선 방정식과의 차이로 표현
$$\text{Loss}(W, b) = \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2, n \text{은 학습 데이터의 개수}$$
- 학습에서는 손실함수 값이 최소가 되는 W 와 b 를 찾는 것



경사하강법 (Gradient Descent Method)

◆ 경사하강법 (gradient descent method)

- 손실함수 (loss function)

$$\begin{aligned}\text{Loss}(W, b) &= \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2 \\ &= \frac{1}{n} \sum_{i=1}^n ((Wx_i + b) - y_i)^2\end{aligned}$$

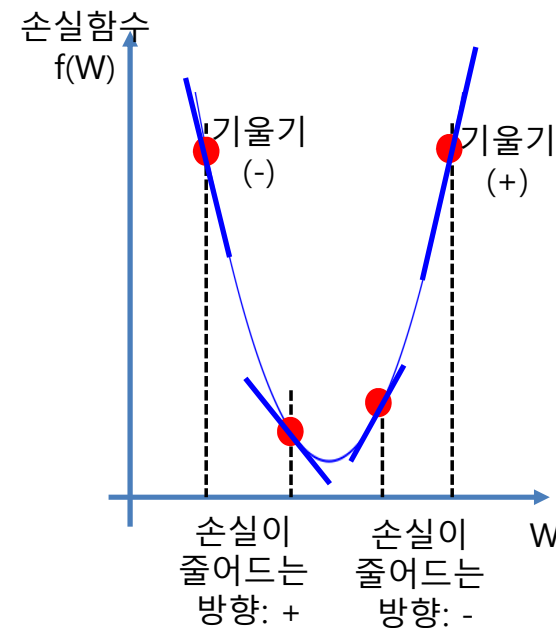
- 손실함수를 W 에 대하여 미분

$$\frac{\partial \text{Loss}(W, b)}{\partial W} = \frac{2}{n} \sum_{i=1}^n ((Wx_i + b) - y_i)$$

- 손실함수 결과값이 줄어드는 방향으로 W 와 b 를 update (ρ : learning rate, 학습률)

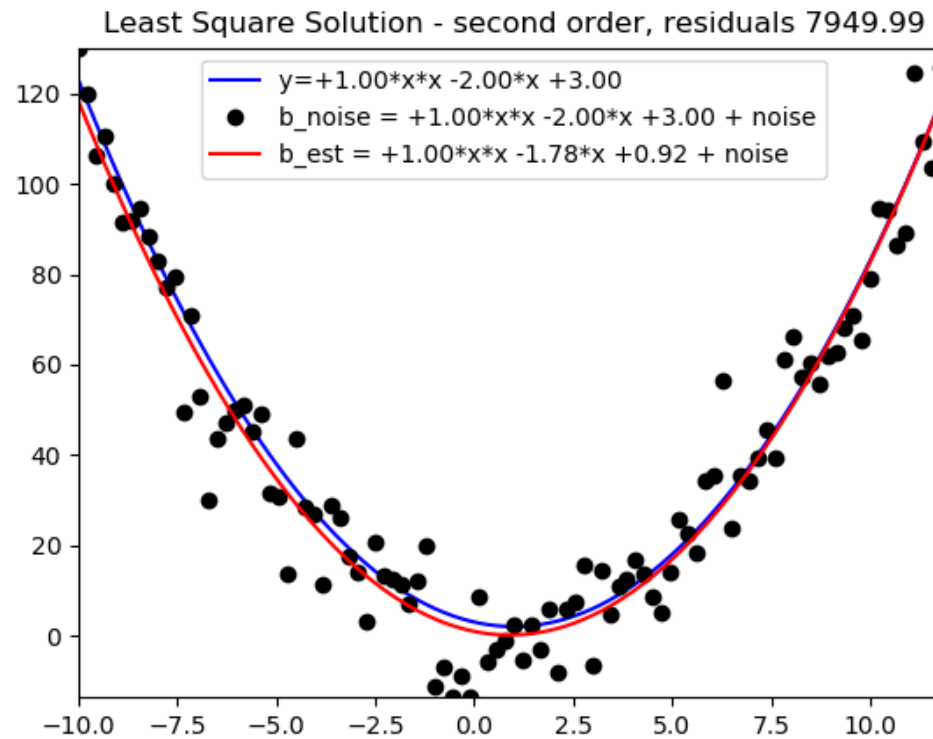
$$W = W - \rho * \frac{\partial \text{Loss}(W, b)}{\partial W}$$

$$b = b - \rho * \frac{\partial \text{Loss}(W, b)}{\partial b}$$



손실함수 최소화 방법 예 – 최소 자승 해

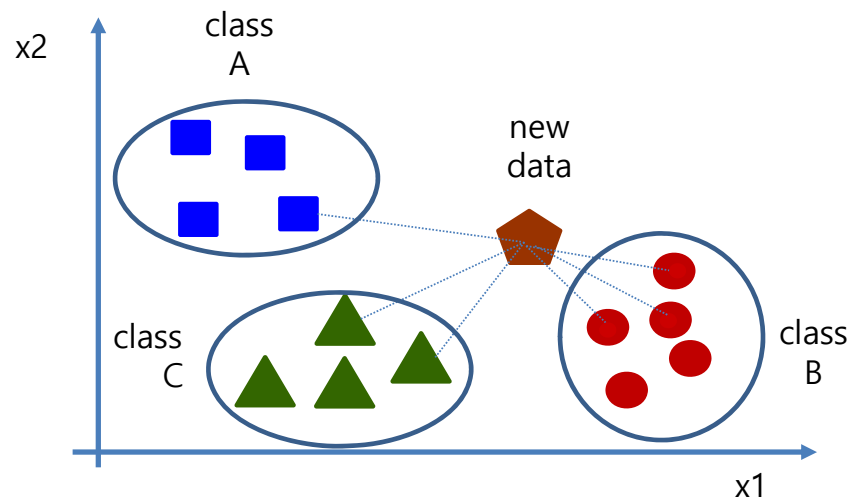
◆ Least Square Solution – Second Order



분류 (Classification)

◆ k-Nearest Neighbor (kNN) 알고리즘

- 학습데이터는 특징 공간 (feature space)에 클래스 (class)들로 분류
- 새로운 입력 데이터를 k개의 최근접 이웃 (nearest neighbor)들과 비교하여 어떤 클래스에 속하게 되는가를 결정
- 가장 가까운 k개의 이웃 중에서 가장 많은 표를 얻은 클래스로 분류



K-means clustering

◆ K-means clustering

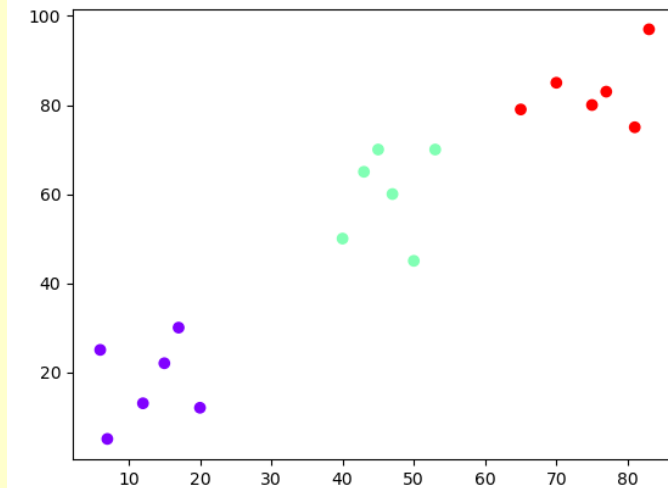
- 비지도 분할 학습
- 주어진 데이터를 k개의 그룹으로 클러스터링
- Sklearn 모듈의 Kmeans() 사용하여 구현

```
# K-means clustering
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

sample_data = np.array([[7, 5], [12, 13], [15, 22], [20, 12], [6, 25], [17, 30],
                        [53, 50], [43, 65], [40, 50], [45, 70], [50, 45], [47, 60],
                        [75, 80], [81, 75], [65, 79], [70, 85], [83, 97], [77, 83]])

#plt.scatter(sample_data[:, 0], sample_data[:, 1])
#plt.show()

kmeans = KMeans(n_clusters = 3)
kmeans.fit(sample_data)
print(kmeans.cluster_centers_)
print(kmeans.labels_)
plt.scatter(sample_data[:, 0], sample_data[:, 1], c=kmeans.labels_, cmap='rainbow')
plt.show()
```

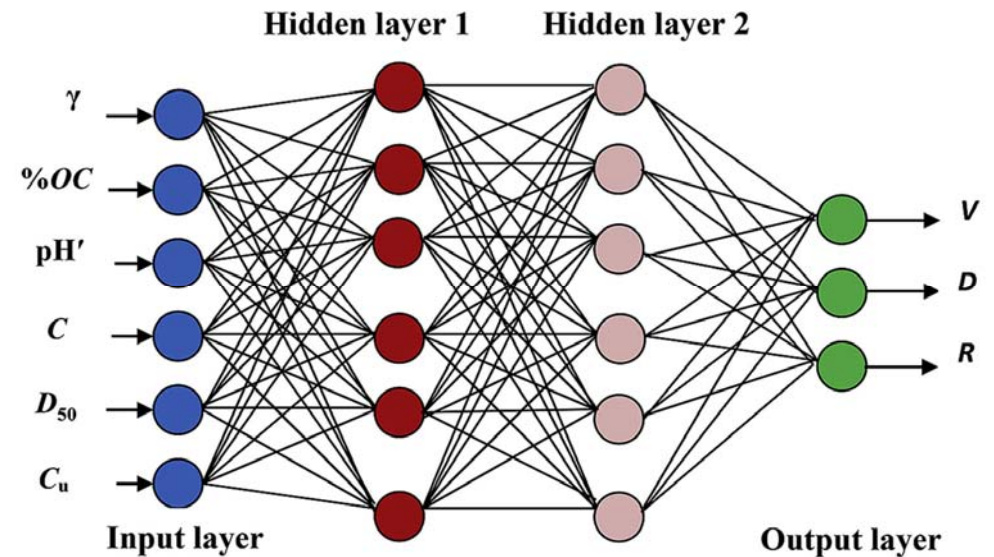
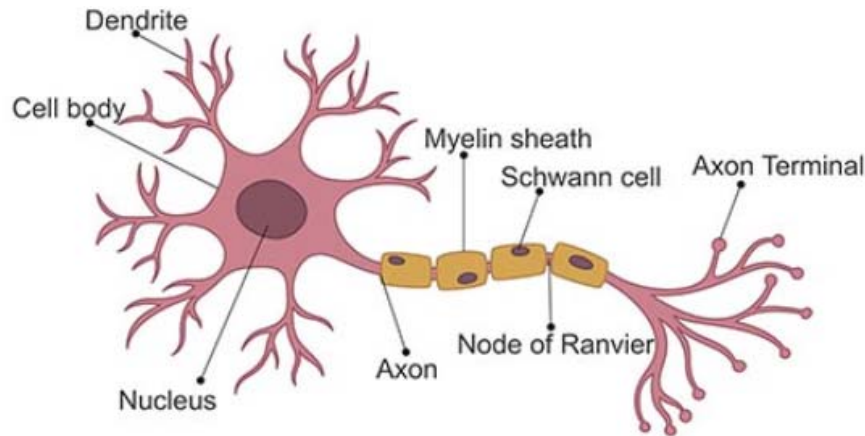


파이썬 기반 심층학습 (deep learning)

신경망(neural network)과 딥러닝 (deep learning)

◆ 딥러닝 (deep learning)

- 신경망 (neural network)을 사용하는 기계학습의 한 분야
(참고: https://ko.wikipedia.org/wiki/인공_신경망)
- 인간의 신경세포 (neuron)와 인공 지능망 (artificial neural network)
- Multi-Layer Perceptron (MLP)



Deep Learning과 응용 – 영상 인식, 자동 번역

◆ TensorFlow

- 구글내 연구와 제품개발을 위한 목적으로 구글 브레인팀이 개발
- 2015년 11월 9일 [아파치 2.0 오픈소스 라이선스](#)로 공개
- <https://www.tensorflow.org/>
- tensor: 물리학에서 다차원 배열을 의미 (tensor에 차원을 지정하면 스칼라, 벡터, 행렬, 텐서를 모두 지원할 수 있음)
- flow: data flow

◆ Keras

- 파이썬으로 구현된 딥러닝 라이브러리이며, TensorFlow 2.0 이후 버전에서 딥러닝 모델을 쉽게 구성할 수 있게 함

◆ 영상 인식

- 얼굴인식, 도로 표지판 인식
- 다양한 개체 인식: 사람, 동물, 개, 고양이
- 주로 OpenCV 패키지를 함께 사용함

◆ 자동 번역 및 통역

- 인터넷 웹 문서의 자동 번역, 문맥 (context) 분석
- 자동 통역



Tesorflow와 Keras 설치

```
C:\Users\Owner>python -m pip install --user --upgrade tensorflow
Collecting tensorflow
  Downloading tensorflow-2.5.0rc1-cp39-cp39-win_amd64.whl (422.5 MB)
    |██████████████████████████████| 422.5 MB 6.8 MB/s
Collecting termcolor~=1.1.0
  Downloading termcolor-1.1.0.tar.gz (3.9 kB)
Collecting google-pasta~=0.2
  Downloading google_pasta-0.2.0-py3-none-any.whl (57 kB)
    |██████████████████████████████| 57 kB 2.4 MB/s
Requirement already satisfied: numpy~=1.19.2 in c:\users\owner\AppData\Local\Programs\Python\Python39\lib\site-packages
(from tensorflow) (1.19.4)
Collecting keras-preprocessing~=1.1.2
  Downloading Keras Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
    |██████████████████████████████| 42 kB 3.4 MB/s
Collecting keras-nightly~=2.5.0.dev
  Downloading keras_nightly-2.5.0.dev2021032900-py2.py3-none-any.whl (1.2 MB)
    |██████████████████████████████| 1.2 MB ...
```

```
Successfully installed absl-py-0.12.0 astunparse-1.6.3 cachetools-4.2.1 flatbuffers-1.12 gast-0.4.0 google-auth-1.30.0 google-auth-oauthlib-0.4.4 google-pasta-0.2.0 grpcio-1.34.1 h5py-3.1.0 keras-nightly-2.5.0.dev2021032900 keras-preprocessing-1.1.2 markdown-3.3.4 oauthlib-3.1.0 opt-einsum-3.3.0 protobuf-3.15.8 pyasn1-0.4.8 pyasn1-modules-0.2.8 requests-oauthlib-1.3.0 rsa-4.7.2 tensorboard-2.5.0 tensorboard-data-server-0.6.0 tensorboard-plugin-wit-1.8.0 tensorflow-2.5.0rc1 tensorflow-estimator-2.5.0rc0 termcolor-1.1.0 typing-extensions-3.7.4.3 werkzeug-1.0.1 wheel-0.36.2
```

```
C:\Users\Owner>
```

Keras import에서 발생 가능 문제

◆ Keras-nightly 문제

- 텐서플로우가 Keras를 2.0버전 부터 내장하여 배포하기 시작
- 최근 2.5버전을 설치하면 Keras-nightly 패키지가 함께 설치되지만 사용되지는 않음
- 2.5버전에서 해당 패키지와 충돌로 인해 Keras를 import를 하면 오류가 발생하는 것을 확인

◆ 해결 방법

- cmd 창에서 `python -m pip uninstall keras-nightly` 입력
- 제거하여도 동작하지 않는다면 아래 명령어 입력하여 텐서플로우를 재설치
 - `python -m pip install tensorflow --upgrade --force-reinstall`



Keras Sequential Model 관련 함수

◆ Keras Sequential Model

- https://www.tensorflow.org/guide/keras/sequential_model?hl=ko
- Sequential model: 각 레이어에 정확히 하나의 입력 텐서와 하나의 출력 텐서가 있는 일반 레이어 스택에 적합



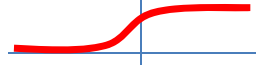
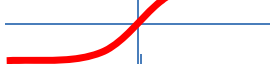




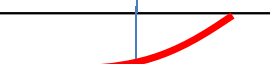
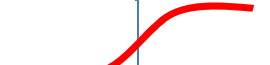
◆ Model에 추가될 수 있는 layer

- <https://zereight.tistory.com/227>

Layer 종류	설명
Flatten	2차원의 특징 맵을 전결합층으로 전달하기 위하여 1차원 형식으로 변환
Dense	모든 입력 뉴런과 출력 뉴런을 연결하는 전 결합층
Conv2D	필터를 사용하여 영상 특징을 추출하는 Convolution2D
MaxPooling2D	입력벡터에서 특정 구간마다 값을 골라 벡터를 구성한 후 반환
Dropout	과적합을 방지하기 위하여 학습시에 지정된 비율만큼 임의의 노드(뉴런)들을 제외시킴



Activation Functions in Neural Networks

activation	equation	plot
identity	$f(x) = x$	
binary step	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	
logistic (soft step)	$f(x) = \frac{1}{1 + e^{-x}}$	
tanh	$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	
arctan	$f(x) = \tan^{-1}(x)$	
rectified linear unit (relu)	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	
parametric rectified linear unit (prelu)	$f(x) = \begin{cases} ax & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	
exponential linear unit (elu)	$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	
softplus	$f(x) = \log_e(1 + e^x)$	
logistic sigmoid	$\phi(x) = \frac{1}{(1 + e^{-x})}$	



Model related functions compile(), fit(), evaluate()

◆ CNN Model related methods

- Convolutional Neural Network (CNN) - 합성곱 신경망
- <https://keras.io/api/metrics/>

method	description
compile()	model의 optimizer, loss, metrics를 선택 <ul style="list-style-type: none">- optimizer: sgd, rmsprop, adam, adadelata, adagrad, adamax, nadam, ftrl- loss: probabilistic, regression, hinge- metrics: accuracy, probabilistic, regression, classification, image segmentation, hinge
fit()	모델의 weight와 bias 값을 학습을 통해 결정
evaluate()	test data set에 대하여 성능을 평가
summary()	model의 구조와 관련 파라미터를 출력
save()	model을 파일로 출력
load_model()	지정된 파일로 부터 model을 읽고 설치
predict()	model을 사용하여 주어진 패턴에 대하여 예측



Keras 기반 선형회귀예제

```
# Keras application - linear regression

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

# generate data set
X = data = np.linspace(1, 2, 50)
y = X*4 + np.random.randn(50) * 0.3 # add noise

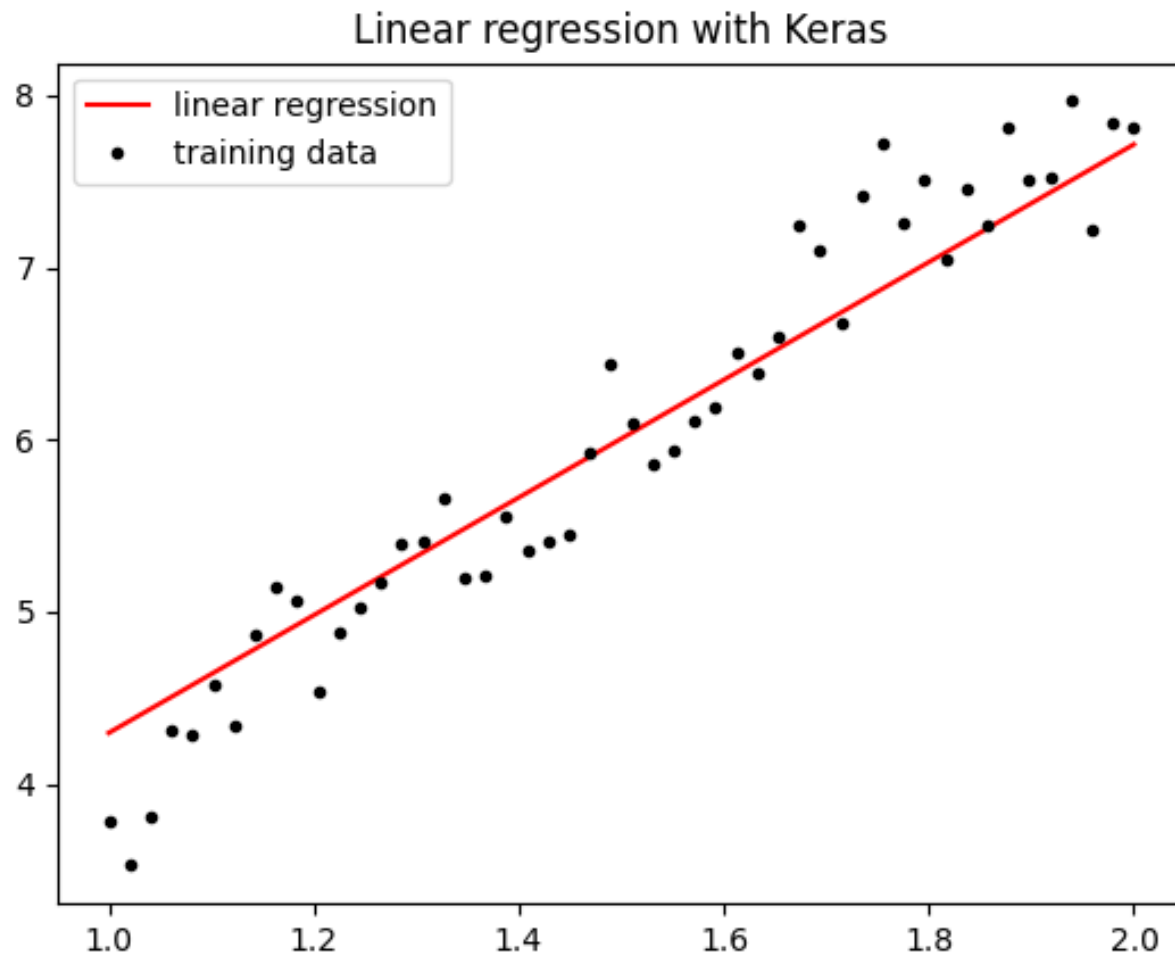
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Dense(1, input_dim=1, activation='linear'))

model.compile(optimizer='sgd', loss='mse', metrics=['mse'])
model.fit(X, y, batch_size=1, epochs=20, verbose=2)

predict = model.predict(data)
plt.plot(data, predict, 'r', label="linear regression")
plt.plot(data, y, 'k.', label="training data")
# blue predict line, black dots of random data
plt.title("Linear regression with Keras")
plt.legend(loc="best")
plt.show()
```



Keras 기반 선형회귀예제 - 결과



Keras 기반 필기체 인식

◆ MNIST 데이터 셋 (<http://yann.lecun.com/exdb/mnist/>)

- 70000개의 데이터 셋 (28x28 크기)
- 모델 학습을 위한 학습용 데이터 55000개 : mnist.train
- 학습된 모델을 시험하기 위한 test data set 10000개: mnist.test
- 모델을 검증하기 위한 검증용 데이터 5000개 : mnist.validation
- training:validation:test = 55,000 : 5,000 : 10,000

파일	목적
train-images-idx3-ubyte.gz	학습 셋 이미지 - 55000개의 트레이닝 이미지, 5000개의 검증 이미지
train-labels-idx1-ubyte.gz	이미지와 매칭되는 학습 셋 레이블
t10k-images-idx3-ubyte.gz	테스트 셋 이미지 - 10000개의 이미지
t10k-labels-idx1-ubyte.gz	이미지와 매칭되는 테스트 셋 레이블



필기체 숫자 인식

◆ CNN 구조의 필기체 숫자 인식 모델 구성

- Import the libraries and load the MNIST dataset
- Data Preprocess and Normalization
- Create the model
- Train the model
- Evaluate the model
- Save the model in file (CNN_model_digits)

◆ CNN 구조의 필기체 숫자 인식 GUI App 실행

- Load the model from file (CNN_model_digits)
- Create GUI to predict digits



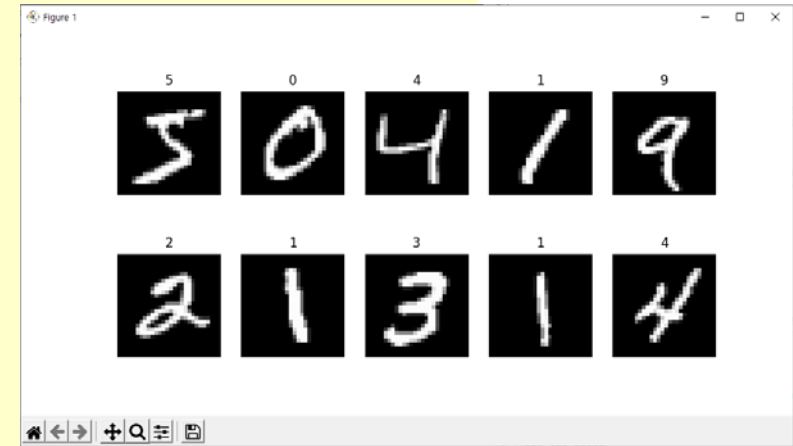
Handwritten Digits Recognition (1)

```
import tensorflow as tf
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
#from keras.utils import to_categorical
kr_utils = tf.keras.utils

from keras import backend as k
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#load dataset directly from keras library
print("Loading MNIST data . . . .")
mnist_npz_path = 'C://MyPyPackage//MNIST//mnist.npz'
(X_train, y_train), (X_test, y_test) = mnist.load_data(path = mnist_npz_path)

digit_names = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]
plt.figure(figsize=(10,5))
for i in range(10):
    plt.subplot(2, 5, i+1)
    plt.imshow(X_train[i], cmap="gray")
    plt.title(digit_names[y_train[i]])
    plt.axis('off')
plt.show()
```



```

# Handwritten Digits Recognition (2)

# reshape format [samples][width][height][channels]
print("Reshaping format . . . .")
X_train = X_train.reshape(X_train.shape[0], 28, 28, 1).astype('float32')
X_test = X_test.reshape(X_test.shape[0], 28, 28, 1).astype('float32')

# Converts a class vector (integer) to binary class matrix
print("Converting class vector . . . .")
# Converts a class vector (integers) to binary class matrix.
y_train = tf.keras.utils.to_categorical(y_train)
y_test = tf.keras.utils.to_categorical(y_test)

# normalize inputs
X_train = X_train / 255
X_test = X_test / 255

print("Preparing a CNN model . . . .")
# define a CNN model
num_classes = 10
model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    Flatten(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(num_classes, activation='softmax')])
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

```



Handwritten Digits Recognition (3)

```
print("Fitting the model . . . .")
# fit the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=20,
batch_size=200, verbose=2)
print("The model has successfully trained")
```

```
# Save the model
model.save("CNN_model_Digits")
print("The model has successfully saved !!")
model.summary() # print model
```

```
# Evaluate the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("CNN error: %.2f%%"%(100 - scores[1]*100))
```

```
Loading MNIST data . . . .
Reshaping format . . . .
Converting class vector . . . .
Preparing a CNN model . . . .
Compiling the model . . . .
Fitting the model . . . .
Epoch 1/20
WARNING:tensorflow:AutoGraph could not transform <bound method Dense.call of <keras.layers.co
2ES0>> and will run it as-is.
Please report this to the TensorFlow team. When filing the bug, set the verbosity to 10 (on l
=10') and attach the full output.
Cause: invalid syntax (tmpaddsv49f.py, line 49)
To silence this warning, decorate the function with @tf.autograph.experimental.do_not_convert
300/300 - 41s - loss: 0.2205 - accuracy: 0.8333 - val_loss: 0.0488 - val_accuracy: 0.9840
Epoch 2/20
300/300 - 30s - loss: 0.0664 - accuracy: 0.9800 - val_loss: 0.0370 - val_accuracy: 0.9866
Epoch 3/20
300/300 - 31s - loss: 0.0475 - accuracy: 0.9855 - val_loss: 0.0298 - val_accuracy: 0.9896
Epoch 4/20
300/300 - 30s - loss: 0.0366 - accuracy: 0.9886 - val_loss: 0.0297 - val_accuracy: 0.9895
Epoch 5/20
300/300 - 30s - loss: 0.0306 - accuracy: 0.9902 - val_loss: 0.0268 - val_accuracy: 0.9913
Epoch 6/20
300/300 - 30s - loss: 0.0251 - accuracy: 0.9916 - val_loss: 0.0270 - val_accuracy: 0.9908
Epoch 7/20
300/300 - 31s - loss: 0.0210 - accuracy: 0.9931 - val_loss: 0.0265 - val_accuracy: 0.9911
Epoch 8/20
300/300 - 31s - loss: 0.0184 - accuracy: 0.9940 - val_loss: 0.0256 - val_accuracy: 0.9924
Epoch 9/20
300/300 - 29s - loss: 0.0150 - accuracy: 0.9951 - val_loss: 0.0275 - val_accuracy: 0.9917
Epoch 10/20
300/300 - 30s - loss: 0.0149 - accuracy: 0.9951 - val_loss: 0.0266 - val_accuracy: 0.9916
Epoch 11/20
300/300 - 30s - loss: 0.0135 - accuracy: 0.9953 - val_loss: 0.0320 - val_accuracy: 0.9896
Epoch 12/20
300/300 - 30s - loss: 0.0130 - accuracy: 0.9957 - val_loss: 0.0258 - val_accuracy: 0.9928
Epoch 13/20
300/300 - 30s - loss: 0.0101 - accuracy: 0.9964 - val_loss: 0.0276 - val_accuracy: 0.9922
Epoch 14/20
300/300 - 31s - loss: 0.0126 - accuracy: 0.9955 - val_loss: 0.0249 - val_accuracy: 0.9926
Epoch 15/20
300/300 - 31s - loss: 0.0082 - accuracy: 0.9972 - val_loss: 0.0275 - val_accuracy: 0.9931
Epoch 16/20
300/300 - 30s - loss: 0.0092 - accuracy: 0.9967 - val_loss: 0.0292 - val_accuracy: 0.9924
Epoch 17/20
300/300 - 30s - loss: 0.0077 - accuracy: 0.9974 - val_loss: 0.0282 - val_accuracy: 0.9926
Epoch 18/20
300/300 - 29s - loss: 0.0073 - accuracy: 0.9975 - val_loss: 0.0410 - val_accuracy: 0.9900
Epoch 19/20
300/300 - 29s - loss: 0.0075 - accuracy: 0.9974 - val_loss: 0.0290 - val_accuracy: 0.9920
Epoch 20/20
300/300 - 30s - loss: 0.0051 - accuracy: 0.9980 - val_loss: 0.0323 - val_accuracy: 0.9921
The model has successfully trained
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
flatten (Flatten)	(None, 7744)	0
dense (Dense)	(None, 256)	1982720
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 10)	2570
Total params: 2,004,106		
Trainable params: 2,004,106		
Non-trainable params: 0		

The model has successfully saved !!

CNN error: 0.80%




```
# GUI for handwritten digits recognition (2)

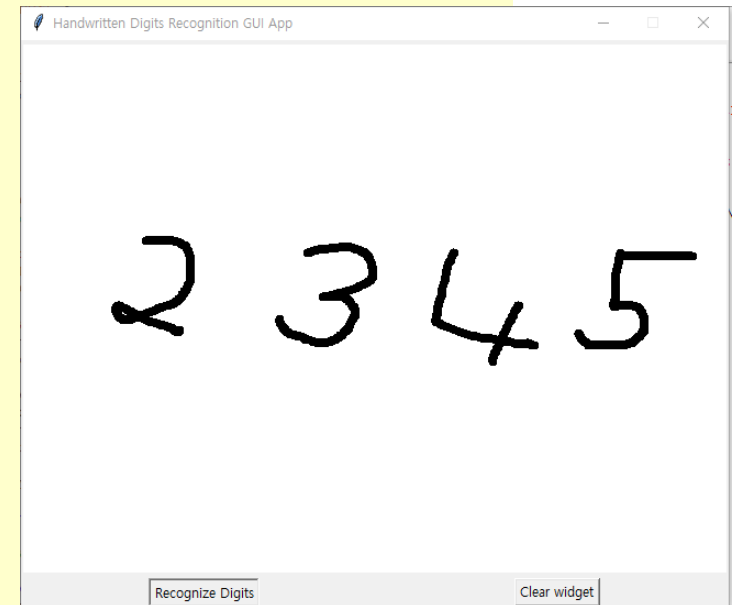
def draw_lines(event):
    global lastx, lasty
    x, y = event.x, event.y
    cv.create_line((lastx, lasty, x, y), width=8, fill='black',
                  capstyle=ROUND, smooth=TRUE, splinesteps=12)
    lastx, lasty = x, y

def clear_widget():
    global cv
    cv.delete("all")

def activate_event(event):
    global lastx, lasty
    cv.bind('<B1-Motion>', draw_lines)
    lastx, lasty = event.x, event.y

def recognize_digit():
    global image_number
    predictions = []
    percentage = []
    #image_number = 0
    filename = f'image_{image_number}.png'
    widget = cv

    #get the widget coordinates
    x = root.winfo_rootx() + widget.winfo_x()
    y = root.winfo_rooty() + widget.winfo_y()
    x1 = x + widget.winfo_width()
    y1 = y + widget.winfo_height()
```

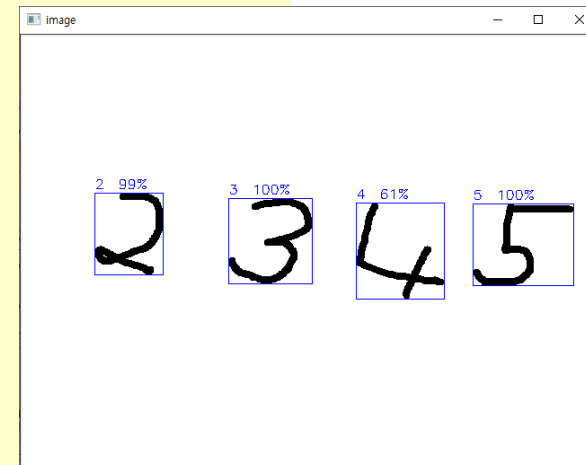
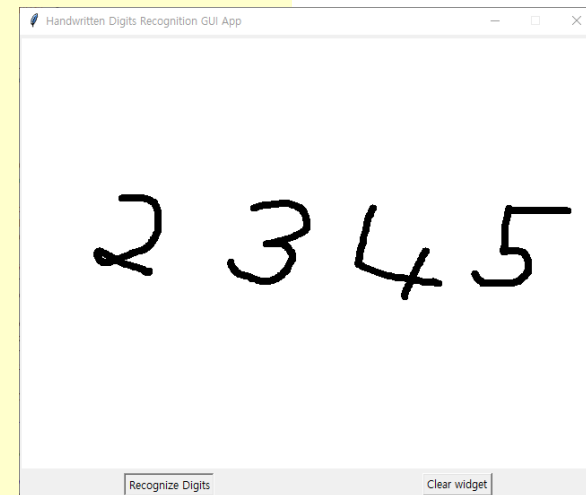


GUI for handwritten digits recognition (3)

```
# grab the image, crop it
ImageGrab.grab().crop((x, y, x1, y1)).save(filename)

# read the image in color format
image = cv2.imread(filename, cv2.IMREAD_COLOR)
# convert the image in grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# applying thresholding (Bobuyuki Otsu's method: grayscale to monochrome)
ret, th = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
# findContour() function helps in extracting the contours from the image
contours = cv2.findContours(th, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[0]

for cnt in contours:
    # get bounding box and extract ROI
    x, y, w, h = cv2.boundingRect(cnt)
    # create rectangle
    cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 1)
    top = int(0.05 * th.shape[0])
    bottom = top
    left = int(0.05 * th.shape[1])
    right = left
    th_up = cv2.copyMakeBorder(th, top, bottom, left, right,
                               cv2.BORDER_REPLICATE)
    # extract the image ROI
    roi = th[y-top:y+h+bottom, x-left:x+w+right]
    # resize roi image to 28x28 pixels
    img = cv2.resize(roi, (28, 28), interpolation=cv2.INTER_AREA)
    # reshaping the image to support our model input
    img = img.reshape(1, 28, 28, 1)
```



```

# GUI for handwritten digits recognition (4)

    # normalizing the image
    img = img / 255.0
    pred = model.predict([img])[0]
    final_pred = np.argmax(pred)
    data = str(final_pred) + ' ' + str(int(max(pred)*100)) + '%'
    font = cv2.FONT_HERSHEY_SIMPLEX
    fontScale = 0.5
    color = (255, 0, 0)
    thickness = 1
    cv2.putText(image, data, (x, y-5), font, fontScale, color, thickness)

cv2.imshow("image", image)
cv2.waitKey(0)

# Tkinter
cv.bind('<Button-1>', activate_event)

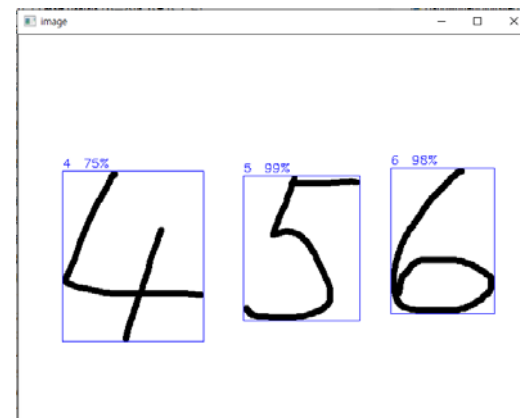
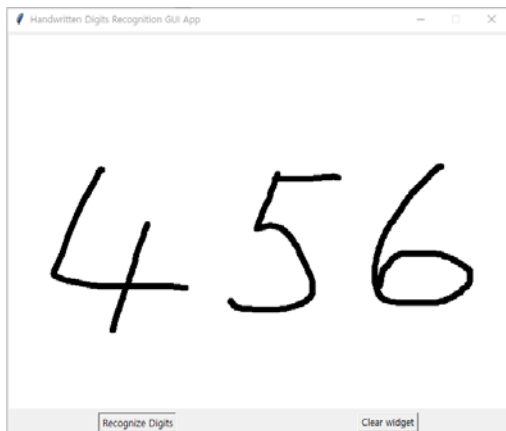
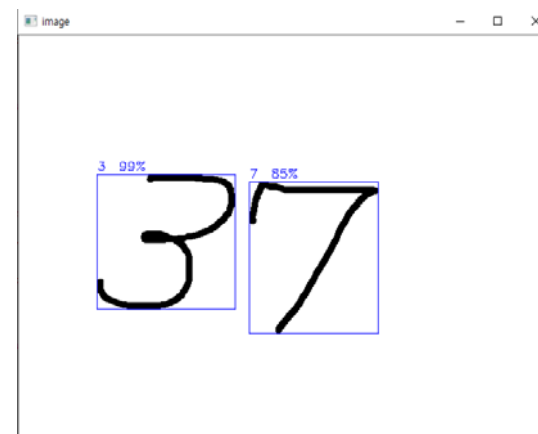
# Add buttons and labels
btn_save = Button(text="Recognize Digits", command = recognize_digit)
btn_save.grid(row=2, column=0, padx=1, pady=1)
btn_clear = Button(text="Clear widget", command = clear_widget)
btn_clear.grid(row=2, column=1, padx=1, pady=1)

# mainloop()
root.mainloop()

```



Results of Recognitions



텐서플로우 CUDA 드라이버 관련 발생 가능 문제

◆ 텐서플로우 CUDA 드라이버 관련 문제

```
5번 필기체 인식 과제 중 tensorflow 프로그램을 실행시키면 이런 에러가 뜨는데 무엇이 문제인가요??  
2021-06-07 16:02:05.836240: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could  
not load dynamic library 'cudart64_110.dll'; dlderror: cudart64_110.dll not found  
2021-06-07 16:02:13.331979: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cud  
art dlderror if you do not have a GPU set up on your machine.
```

◆ 문제 원인

- NVIDIA 그래픽카드 사용시 CUDA 드라이버를 추가로 설치
- 예제 코드를 돌리는 것에는 문제는 없음, 다만 크기가 큰 네트워크를 사용하려고 하는 경
우에는 그래픽 카드를 사용한 연산이 빠르기 때문에 CUDA 드라이버를 설치 해야함

◆ 해결 방법

- 아래 링크를 통해 CUDA 드라이버를 설치
 - [https://developer.nvidia.com/cuda-11.0-download-
archive?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exe_network](https://developer.nvidia.com/cuda-11.0-download-archive?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exe_network)



노트북에서 필기체 인식에서 발생 가능 문제 을 위한 영역 검출 문제

◆ 노트북에서 필기체 인식 기능을 수행하는 경우, 발생 가능 영역 검출 문제

```
Model: "sequential"
=====
Layer (type)                 Output Shape              Param #
=====
conv2d (Conv2D)              (None, 26, 26, 32)        320
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)        0
conv2d_1 (Conv2D)            (None, 11, 11, 64)       18496
flatten (Flatten)            (None, 7744)              0
dense (Dense)                (None, 256)              1982720
dropout (Dropout)            (None, 256)              0
dense_1 (Dense)              (None, 10)               2570
=====
Total params: 2,004,106
Trainable params: 2,004,106
Non-trainable params: 0
=====
Model is loaded successfully ...
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\Wzzxz\AppData\Local\Programs\Python\Python39\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:/MyPyLib/8-5 실행.py", line 83, in recognize_digit
    img = cv2.resize(roi, (28, 28), interpolation = cv2.INTER_AREA)
cv2.error: OpenCV(4.5.2) C:\Users\runneradmin\AppData\Local\Temp\pip-req-build-m8us58q4\opencv\modules\imgproc\src\resize.cpp:4051: error: (-215:Assertion failed) !ssize.empty() in function 'cv::resize'
```



필기체 인식을 위한 영역 검출 문제

◆ 문제 원인

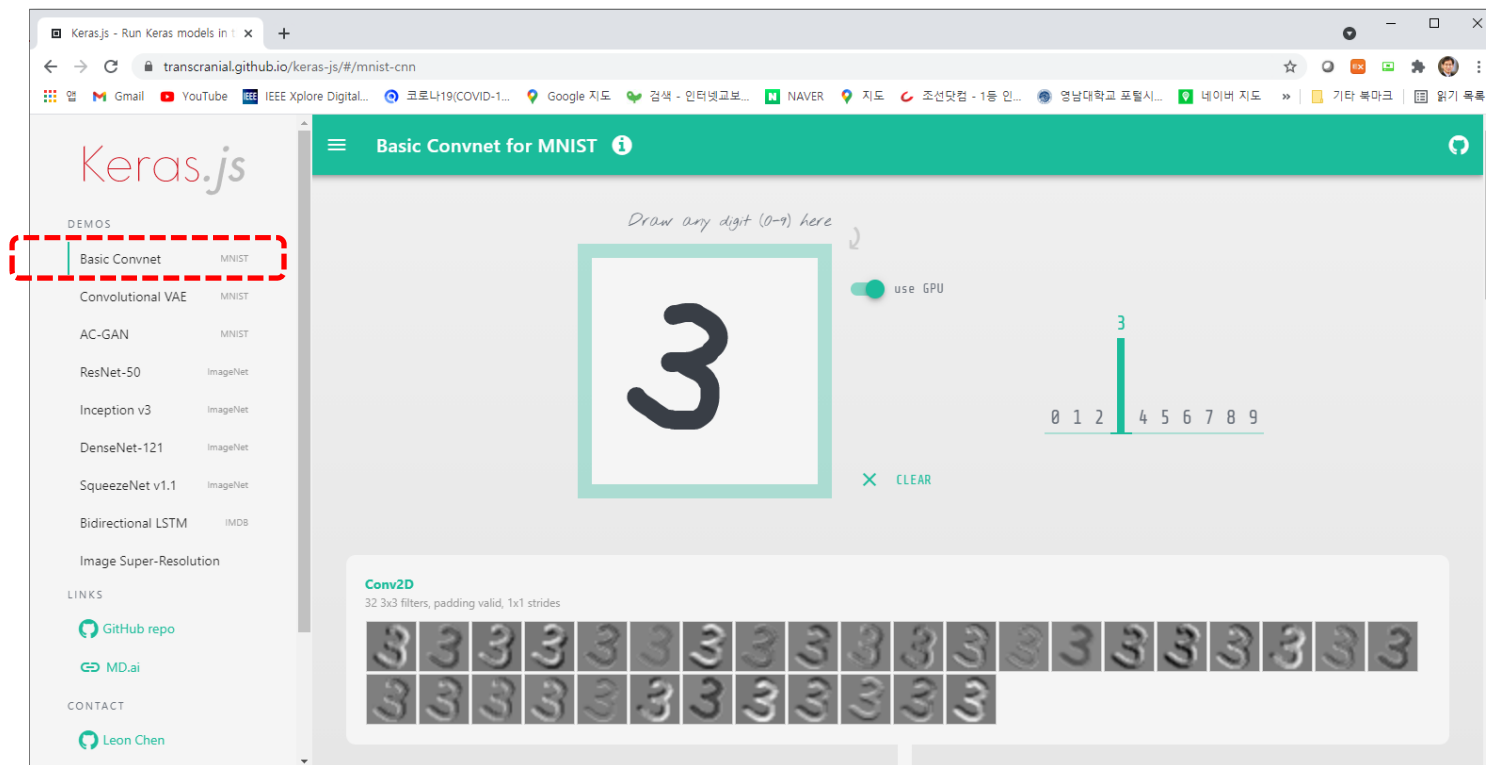
- Recognize_digit() 함수에서 입력 박스의 영역 검출이 제대로 이루어지지 않음
- 해당 문제는 데스크탑에서는 발생하지는 않지만 노트북에서 간혹 발견되는 것을 확인; 해결 방안을 찾고 있는 중

```
def recognize_digit():  
    global image_number  
    predictions = []  
    percentage = []  
    #image_number = 0  
    filename = f'image_{image_number}.png'  
    widget = cv  
    #get the widget coordinates  
    x = root.winfo_rootx() + widget.winfo_x()  
    y = root.winfo_rooty() + widget.winfo_y()  
    x1 = x + widget.winfo_width()  
    y1 = y + widget.winfo_height()
```

Keras 실습

◆ Keras 실습 - 필기체 인식

- <https://transcranial.github.io/keras-js/#/mnist-cnn>



References

<Web server>

- [1] Setting up an Apache Web Server on a Raspberry Pi, <https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md>.
- [2] How to Use Raspberry Pi as a Web Server, <https://www.digikey.com/en/maker/blogs/2020/how-to-use-raspberry-pi-as-a-web-server>.
- [3] Python을 사용하여 간단한 웹서버 구축 – Simple Web Server, <https://webisfree.com/2019-11-19/python-%EC%82%AC%EC%9A%A9%ED%95%98%EC%97%AC-%EA%B0%84%EB%8B%A8%ED%95%9C-%EC%9B%B9%EC%84%9C%EB%B2%84-%EA%B5%AC%EC%B6%95%ED%95%98%EA%B8%B0-simple-web-server>.
- [4] 로컬 테스트 서버 설치하기, [https://developer.mozilla.org/ko/docs/Learn/Common questions/set up a local testing server](https://developer.mozilla.org/ko/docs/Learn/Common%20questions/set%20up%20a%20local%20testing%20server)
- [5] Python 예제: Python 서버 코드(server.py), https://docs.aws.amazon.com/ko_kr/polly/latest/dg/example-Python-server-code.html.
- [6] bottle web server, <http://zetcode.com/python/bottle/>.
- [7] bottle: Python Web Framework, <https://bottlepy.org/docs/dev/>.
- [8] <http://jun.hansung.ac.kr/CWP/htmls/HTML%20Input%20Types.html>.
- [9] <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/range>
- [10] Plate with Self-balancing, https://www.youtube.com/watch?v=j4OmVLc_oDw.
- [11] Delta Robot Do-It-Yourself (DIY) Projects, <https://www.youtube.com/watch?v=g7xa9a76zoU>.

<PyGame>

- [1] <https://devdocs.io/pygame/>.
- [2] 무료 게임 이미지 다운로드 사이트, <https://opengameart.org/>

<Deep learning, 필기체 인식>

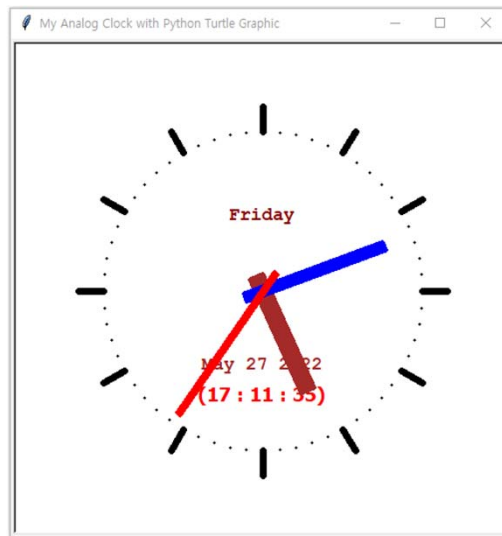
- [1] https://www.tensorflow.org/guide/keras/sequential_model?hl=ko
- [2] <https://qkqhxl1.tistory.com/987>.
- [3] What is neural network, <https://www.youtube.com/watch?v=aircAruvnKk>.



Homework 13.1

13.1 터틀그래픽 기반의 아날로그 시계 구현

- 강의자료를 참고하여 터틀 그래픽 기반의 아날로그 시계를 구현하라.
- 아날로그 시계에는 시, 분, 초침이 회전하며 시간을 나타내도록 하고, 영문 요일 이름, 연/월/일 (예: May 27 2022) , 24시간 체계의 시분초 (예: 17:11:35) 단위 시간을 나타낼 것
- 아날로그 시계 예시



Homework 13.2

13.2 필기체 숫자 인식 기능 구현 및 실험

- 강의 자료를 참고하여 파이썬 프로그램을 구현할 것
- 필기체 숫자 인식 기능을 위한 CNN (convolutional neural network) 구조의 모델(CNN_model_Digits)을 생성하여 저장하는 파이썬 프로그램을 구현하라.
- 필기체 숫자 인식 기능 시험을 위하여 CNN_model_Digits 모델을 load하며, tkinter 기반의 GUI canvas에서 입력된 필기체 숫자에 대한 인식 기능을 수행하는 파이썬 프로그램을 구현하라.
- 0 ~ 9의 숫자를 5개 본인의 필기체로 입력한 후, 어떻게 인식되는지 결과를 capture하라.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
flatten (Flatten)	(None, 7744)	0
dense (Dense)	(None, 256)	1982720
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 10)	2570
Total params: 2,004,106		
Trainable params: 2,004,106		
Non-trainable params: 0		

The model has successfully saved !!

CNN error: 0.80%

