레이와 4년도　　졸업논문

유치원 아동의 운동과 집중력의 관계를 분석하기 위한 시스템의 개발

지도교원 熊澤典良 준교수

上谷俊平 교수

가고시마대학 기계공학과

박태원

목 차

제 1 장

서론

1.1 연구의 배경과 목적

일반적으로 운동과 집중력 사이에는 관계성이 있는 것으로 알려져 있다. 특히 운동 후에는 집중력이 향상되기 때문에, 보육과 교육에서 운동은 적극적으로 채택되고 있다. 유치원 아동의 운동량과 집중력을 계측하고, 개별 아동에게 적합한 운동 내용과 운동 강도를 제안함으로써 보다 효율적인 집중력 향상 효과를 얻을 수 있다.

선행 연구에서는 Fig.1.1에 나타낸 것과 같은 NFC 통신 활동량계 MTN240을 이용하여 운동량 측정을 수행하였다. 활동량계는 소형 기기이며, 착용함으로써 METs, 보수(걸음 수), 소비 칼로리 등을 측정할 수 있다. 집중력 측정은 책 읽어주기 시간에 실시하였으며, Fig.1.2에 나타낸 것과 같은 세 개의 AR 마커를 부착한 모자를 이용하여 머리 흔들림 상황을 측정함으로써 집중도를 평가하였다.

운동량과 집중력의 측정은 유치원에서 실행되기 때문에, 측정한 데이터의 저장·표시·분석을 자동화할 수 있는 시스템이 필요하다. 본 연구에서는 유치원 아동의 운동과 집중력 데이터의 자동 저장과 표시를 수행함으로써 데이터의 가시화와 운동과 집중력의 관계성을 판단하는 것을 목적으로 한다.

Fig.1.1: NFC 통신 활동량계 MTN240

Fig.1.2: AR 마커를 부착한 모자

1.2 본 연구의 개요

본 논문은 총 4장으로 구성되어 있으며, 제2장에서는 본 연구의 실험 환경에 대해 그림과 함께 제시한다.

제3장에서는 활동량계의 데이터를 MySQL에 자동 저장하는 시스템의 개발과 책 읽어주기 시간대의 판정 및 저장에 대해 기술한다.

제4장에서는 자동 저장한 데이터를 바탕으로, 운동과 집중력의 관계성을 분석하기 위해 필요한 데이터의 선정과 선정한 데이터를 Web 상에 표시한 결과를 제시한다.

제 2 장

실험 환경

2.1 활동량계

본 연구에서는 吉田南 유치원의 연장 아동(만 5, 6세) 29명을 대상으로 실험을 수행하였다. 운동량의 측정은 Fig.1.1에 나타낸 것과 같이 아코스(ACOS)사의 NFC 통신 활동량계 MTN240을 이용하였다.

활동량계는 최대 10일분의 1분마다의 METs 값을 보관한다. METs는 metabolic equivalents(대사당량)의 약자이며, 운동 강도를 나타내는 지표이다. 안정 시를 1로 하여 그때와 비교해 몇 배의 에너지를 소비하는지를 나타내는 신체 활동량이다.

또한 최대 1개월분의 1시간마다의 걸음 수[보], 액티브 걸음 수[보], 총소비 칼로리[kcal], 활동 소비 칼로리[kcal], 액티브 활동 시간[초], 활동 시간[초]을 보관한다.

마지막으로 최대 90일분의 1일마다의 걸음 수[보], 액티브 걸음 수[보], 총소비 칼로리[kcal], 활동 소비 칼로리[kcal], 액티브 활동 시간[분], 활동 시간[분], 거리[km], Ex[METs×시간]을 보관한다. 액티브 활동의 기준은 사전에 임계값을 설정할 수 있다. 본 연구에서는 4METs를 임계값으로 설정하였다.

활동량계에 보관되어 있는 데이터를 csv 파일로 출력하려면, Fig.2.1에 나타낸 것과 같은 소니사의 비접촉 IC 카드 리더/라이터 RC-S380을 PC에 연

결하고, 기기 위에 활동량계를 올려놓는다. 그 후 아코스사가 개발한 애플리케이션을 실행함으로써 csv 파일로 저장할 수 있다.

Fig.2.1: 비접촉 IC 카드 리더/라이터 RC-S380

2.2 책 읽어주기 시의 계측 카메라

책 읽어주기 시의 계측은 Fig.2.2에 나타낸 것과 같은 H.VIEW사의 POE IP 카메라(HV-PTZ501)를 이용하여 계측한다. 카메라는 책 읽어주기가 실시되는 방의 모서리에 한 대씩 설치되어 있다. 각각의 카메라에는 Fig.2.3, Fig.2.4, Fig.2.5, Fig.2.6에 나타낸 것과 같이 1부터 4까지의 번호를 부여하였다.

Fig.2.2: POE IP 카메라 (HV-PTZ501)

Fig.2.3: 카메라 1

Fig.2.4: 카메라 2

Fig.2.5: 카메라 3

Fig.2.6: 카메라 2

제 3 장

데이터의 자동 저장

3.1 운동량 데이터의 자동 저장

데이터의 저장에는 Fig.3.1에 나타낸 것과 같이, 유치원에 설치되어 있는 Skynew사의 소형 PC W4를 사용하였다. PC에 설치되어 있는 MySQL이라는 데이터베이스에 취득한 데이터를 저장한다.

Fig.3.1: 소형 PC

활동량계 데이터의 계측과 저장은 유치원에서 선생님들에 의해 실행되기 때문에, 유치원에서 사용할 수 있는 자동 저장 시스템의 개발을 수행하였다. 시스템의 개발에는 Python의 Tkinter라는 GUI 라이브러리를 사용하였다. 운동량 데이터는 이름별 폴더로 나뉘어 있기 때문에, Fig.3.2에 나타낸 것과 같이 여러 폴더의 드래그 앤 드롭 기능

을 작성하였다. 다음으로, 중복 파일이 저장되지 않도록 Fig.3.3에 나타낸 것과 같이 로그 파일을 작성함으로써 갱신 파일의 검출을 수행하였다. 마지막으로 scp라는 파일 전송 프로토콜을 이용하여 소형 PC에 원본 csv 파일을 전송하였다.

데이터베이스에 저장하는 내용은 시간, 활동량계 ID, 1분마다의 데이터인 METs, 1시간마다의 데이터인 걸음 수[보], 액티브 걸음 수[보], 총소비 칼로리[kcal], 액티브 소비 칼로리[kcal], 액티브 활동 시간[초], 활동 시간[초]을 저장하였다.

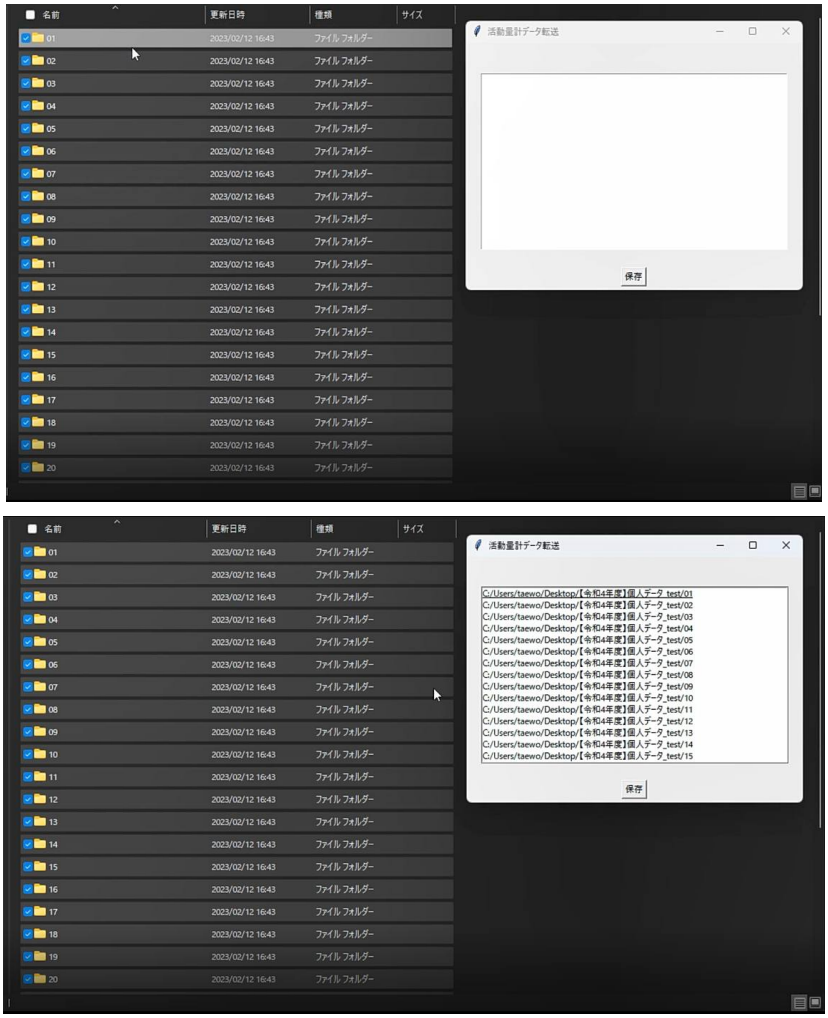

Fig.3.2: ドラッグ アンド ドロップ 機能

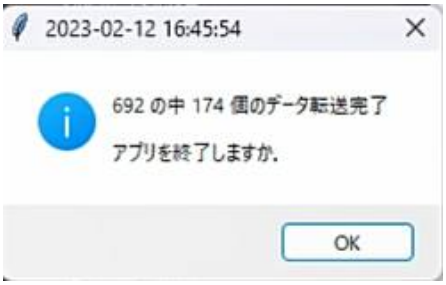

Fig.3.3: 갱신 파일의 검출

3.2 책 읽어주기 영상의 저장

吉田南 유치원에서는 일주일에 2회, 10시부터 11시 사이에 약 10분간 책 읽어주기를 실시하고 있다. 무작위 시간대에 책 읽어주기가 실시되기 때문에, cron이라 불리는 사용자가 설정한 스케줄에 따라 지정된 프로그램을 정기적으로 실행해 주는 프로그램을 이용하여 월요일부터 금요일까지 10시부터 11시까지 카메라 4대의 영상을 Fig.3.1에 나타낸 것과 같은 소형 PC에 저장하였다.

각 카메라의 RTSP(Realtime Time Streaming Protocol) 서버라는 IP 네트워크를 이용하여, 동영상을 실시간으로 전송하는 서버에 접속하여 영상을 저장한다.

영상의 저장에는 GStreamer라 불리는 멀티미디어를 다루기 위한 라이브러리를 사용하였다. 4개의 RTSP 서버에 동시에 접속하기 위해서는 최대 30초가 필요하다. 정확히 10시부터 11시 사이의 영상을 저장하기 위해, 9시 59분부터 11시 1분까지 4대의 카메라 영상을 저장한다. 저장이 끝난 후 ffmpeg이라 불리는 동영상을 기록·변환·재생하기 위한 소프트웨어를 이용하여 처음 1분과 마지막 1분을 잘라냄으로써 정확히 10시부터 11시까지의 동영상을 저장할 수 있었다. 영상 크기는 2592×1944[px]이며, 프레임률은 5[fps]이다.

책 읽어주기는 무작위 시간대에 실시되기 때문에, 저장한 1시간의 영상을 바탕으로 책 읽어주기의 판정을 수행할 필요가 있다. 판정에는 Fig.2.6에 나타낸 것과 같은 카메라 4를 사용하여 프레임 단위로 분석하였다.

첫 번째 판정 기준은 Python의 이미지나 동영상 처리에 사용되는 OpenCV라 불리는 라이브러리를 이용하여 Fig.3.4에 나타낸 것과 같이, AR 마커가 10개 이상 검출되었을 때를 책 읽어주기 시간으로 판정하였다. 그러나 책 읽어주기 전에 준비 시간이 존재하기 때문에 책 읽어주기의 시작 시간에 오차가 발생하였다. 그래서 동일한 선생님들이 단상에 앉아 책 읽어주기를 실시한다는 정보를 이용하여 또 하나의 판정 기준을 마련하였다.

Fig.3.4: 첫 번째 판정 기준

두 번째 판정 기준은 Fig.3.5에 나타낸 것과 같이, 단상 부분을 추출하여 물체 인식으로 사람이 1명만 검출되었을 때, 사람의 높이 h[px]를 구한다. 또한 사전에 측정한 사람이 앉아 있는 상태의 높이를 H[px]라고 한다. 측정한 결과, H = 390[px]이다. 한 사람만 단상에 앉아 있는 상태를 나타내는 식 (3.1)에 나타낸 것과 같은 상태가 10초간 20회 이상 검출되었을 때를 책 읽어주기의 시작 시간으로 판정하였다.

$$H \quad 20 \leq h \leq H + 20 \tag{3.1}$$

Fig.3.5: 두 번째 판정 기준

제 4 장

데이터의 선정과 표시

4.1 운동량 데이터

본 연구에서는 Fig.3.2에 나타낸 것과 같이 소형 PC에 설치되어 있는 Apache2라는 Web 서버를 이용하여 Web 상에 데이터의 표시를 수행하였다. Fig.4.1에 나타낸 것과 같이, 날짜를 선택하고 확인 버튼을 클릭함으로써 전원의 정보를 확인할 수 있다. 전원의 정보는 Fig.4.2에 나타낸 것과 같이 출석번호순, 운동 강도가 높은 순, 운동 강도가 낮은 순을 선택함으로써 정렬할 수 있다.

Fig.4.1: 날짜 선택

Fig.4.2: 정렬 기준 선택란

전원의 운동량 데이터는 Fig.4.3에 나타내었다. 3METs 이상이 최초로 계측된 시점부터 운동 시작으로 판단하고, 출석번호, 책 읽어주기가 시작되기 전까지의 총 METs, 평균 METs, 최대 METs, 운동 시간[분], 소비 칼로리[kcal], 운동 강도, 상세 정보 선택란을 표시하였다. 운동 강도는 일본건강운동연구소의 신체 활동·운동의 강도와 양의 지표를 참고하여 Table.4.1에 나타낸 것과 같이, METs에 의한 운동 강도를 7단계로 나누어 평가하였다.

상세 정보를 선택함으로써 Fig.4.4에 나타낸 것과 같이, 운동 시작 시간부터 책 읽어주기 종료 시간까지의 METS 값을 나타내는 그래프를 표시한다. 파란 부분이 운동 시간이고 빨간 부분이 책 읽어주기 시간이다.

출석번호	총METs	평균METs	최대METs	운동시간 [분]	활동소비칼로리 [kcal]	운동강도	상세정보
1	146.2	3.11	6.5	47	111.0	편함	⦿
2	153.9	3.14	10.5	49	107.6	편함	○
3	107.3	3.46	8.5	31	102.2	편함	○
4	156.0	3.18	6.2	49	122.8	편함	○
5	68.4	2.07	4.8	33	82.4	매우 편함	○
6	72.0	2.06	3.5	35	84.0	매우 편함	○
7	0	0	0	0	0	x	
8	308.3	3.08	10.1	100	161.8	편함	○
9	233.0	3.33	8.9	70	149.0	편함	○
10	0	0	0	0	0	x	
11	67.8	2.12	5.0	32	78.0	매우 편함	○
12	107.6	2.83	7.4	38	83.2	매우 편함	○
13	105.2	3.29	9.2	32	110.2	편함	○
14	271.6	2.72	7.6	100	123.2	매우 편함	○
15	0	0	0	0	0	x	

출석번호	총METs	평균METs	최대METs	운동시간 [분]	활동소비칼로리 [kcal]	운동강도	상세정보
16	109.9	3.33	6.4	33	87.8	편함	○
17	212.3	2.91	7.9	73	121.0	매우 편함	○
18	97.1	2.94	6.2	33	83.8	매우 편함	○
19	310.9	3.05	7.4	102	138.6	편함	○
20	0	0	0	0	0	x	
21	0	0	0	0	0	x	
22	292.4	2.92	8.3	100	141.8	매우 편함	○
23	0	0	0	0	0	x	
24	228.3	2.59	9.6	88	111.0	매우 편함	○
25	178.3	3.07	7.5	58	100.2	편함	○
26	119.7	3.52	8.2	34	89.0	편함	○
27	0	0	0	0	0	x	
28	197.5	2.82	8.4	70	104.2	매우 편함	○
29	139.5	2.97	7.2	47	91.2	매우 편함	○

Fig.4.3: 전원의 운동량 데이터

Table.4.1: 운동 강도의 기준

METs	운동 강도
2 이상 3 미만	매우 편함
3 이상 4 미만	편함
4 이상 5 미만	약간 편함
5 이상 6 미만	다소 힘든 편
6 이상 7 미만	약간 힘듦
7 이상 8 미만	힘듦
8 이상	매우 힘듦

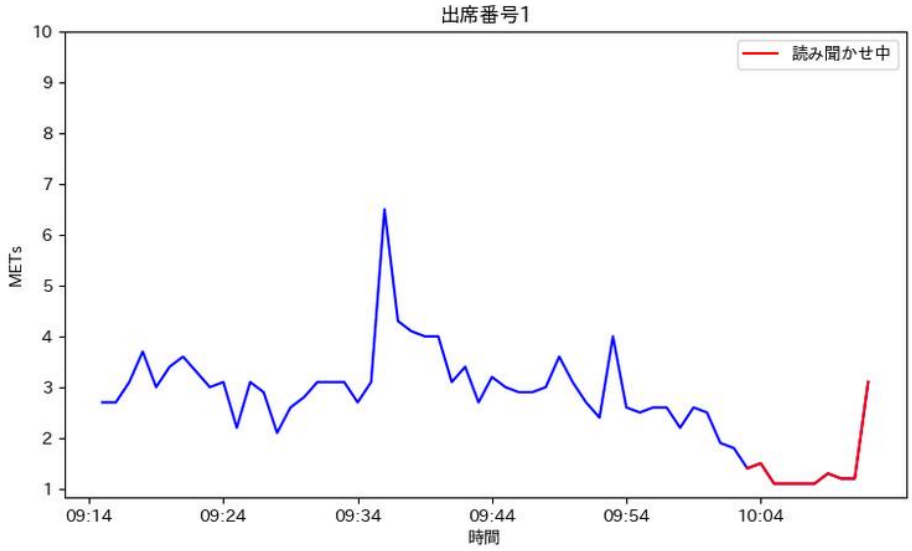

Fig.4.4: 개인의 METs 정보

4.2 책 읽어주기 데이터

날짜를 선택하면 Fig.4.5에 나타낸 것과 같이, 책 읽어주기의 시간과 영상이 표시된다. Fig.4.6에 나타낸 것과 같이 전체 카메라, 카메라 1, 카메라 2, 카메라 3, 카메라 4를 선택할 수 있다. Fig.4.3에 나타낸 것과 같이 상세 정보를 선택한 경우에는 선택한 사람의 위치를 Fig.4.7에 나타낸 것과 같이 화살표로 표시한다. 또한 Fig.4.8에 나타낸 것과 같이, 가로축은 프레임, 세로축은 x, y인 선택한 개인의 머리 흔들림을 확인할 수 있는 두 개의 그래프 동영상을 표시하였다.

Fig.4.5: 책 읽어주기 데이터의 표시

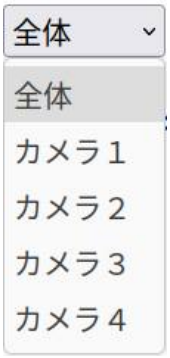

Fig.4.6: 책 읽어주기 카메라 선택란

Fig.4.7: 개인의 책 읽어주기 동영상

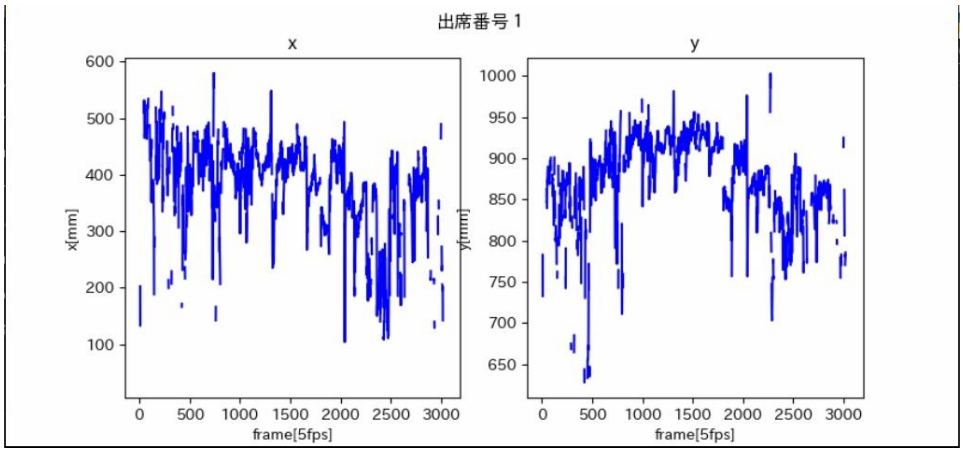

Fig.4.8: 개인의 책 읽어주기 시의 그래프

제 5 장

결론

본 연구에서는 유치원 아동의 운동과 집중력의 관계를 분석하기 위한 시스템 개발을 수행하였다. 운동량과 집중력의 측정은 유치원에서 실시되기 때문에, 시스템의 자동화가 필요하다.

데이터의 자동 저장 시스템으로서는 활동량계 데이터를 자동 저장하는 시스템을 개발하였다. 시스템은 여러 폴더의 드래그 앤 드롭, 로그 파일에 의한 갱신 파일의 검출, scp 전송에 의한 원본 데이터의 저장 기능을 갖는다. 활동량계의 자동 저장 시스템을 이용함으로써 데이터베이스에 손쉽게 데이터를 저장할 수 있다. 책 읽어주기 동영상은 10시부터 11시까지의 카메라 영상을 저장하고, 저장한 영상을 바탕으로 책 읽어주기 시간대를 판정할 수 있었다.

저장한 데이터를 바탕으로 Fig.5.1에 나타낸 것과 같이, Web 상에 운동과 집중력의 관계를 분석하기 위해 데이터의 선정과 표시를 수행하였다.

정확한 운동 강도와 집중력의 판단 기준이 없다는 문제점이 발견되었다. 기준의 설정 등의 대책이 필요하다고 생각된다.

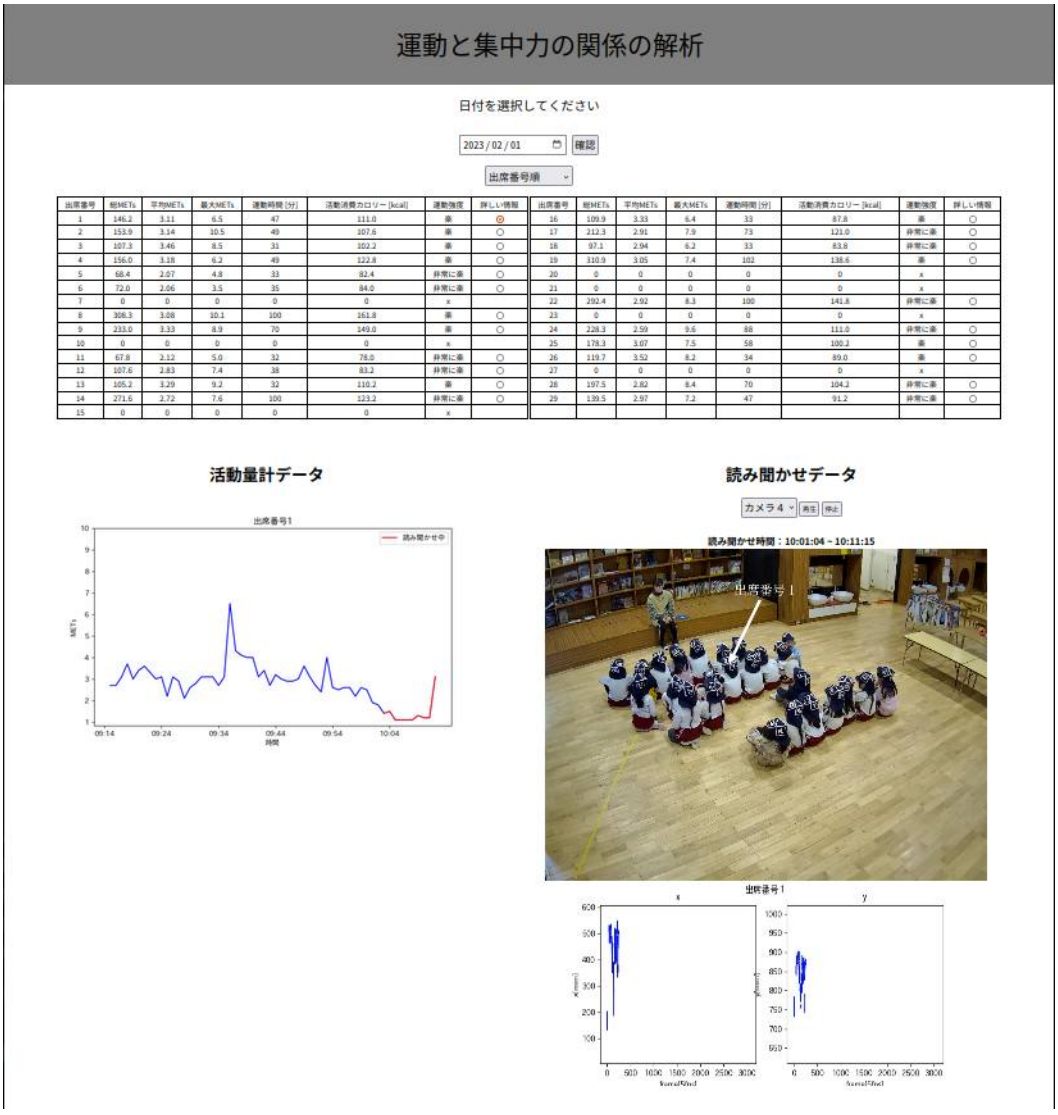
運動と集中力の関係の解析
日付を選択してください
2023 / 02 / 01
確認
出席番号順
活動量計データ
読み聞かせデータ
カメラ４
再生
停止
読み聞かせ時間：10:01:04 ~ 10:11:15
出席番号１

Fig.5.1: 전체 Web

감사의 글

본 연구를 진행하고 논문을 작성함에 있어 지도해 주신 지도교원 熊澤典良 준교수, 上谷俊平 교수께 진심으로 감사의 뜻을 표합니다. 특히 담당 교원이신 熊澤典良 준교수께는 많은 조언을 받았습니다. 깊이 감사드립니다. 본 연구의 측정에 협조해 주신 吉田南 유치원의 橋口 원장 선생님을 비롯한 선생님들과 유치원 아동 여러분께도 깊이 감사드립니다. 또한 본 연구를 진행함에 있어 신세를 진 모든 분들께 진심으로 감사의 말씀을 드리며 감사의 글을 대신하고자 합니다.

Appendix A 장

부록 A장의 제목

2023/02/23 ParkTaewon title：데이터의 저장·표시 by k2519210977@kadai.jp

A.1 활동량계 데이터의 지도 저장 시스템 프로그램

```
import os
import datetime as dt
import paramiko
import scp
from TkinterDnD2 import *
from tkinter import filedialog
from tkinter import messagebox
import csv
try:
    from Tkinter import *
    from ScrolledText import ScrolledText
except ImportError:
    from tkinter import *
    from tkinter.scrolledtext import ScrolledText

import mysql.connector
import pandas as pd

ip'address = '192.168.1.23'

root = TkinterDnD.Tk()
root.withdraw()
root.title('活動量計データ転送')
root.grid'rowconfigure(1, weight=1, minsize=250)
root.grid'columnconfigure(0, weight=1, minsize=250)
#root.grid'columnconfigure(1, weight=1, minsize=250)

def print'event'info(event):
    print('"nAction:', event.action)
    print('Supported actions:', event.actions)
    print('Mouse button:', event.button)
    print('Type codes:', event.codes)
    print('Current type code:', event.code)
    print('Common source types:', event.commonsourcetypes)
    print('Common target types:', event.commontargettypes)
    print('Data:', event.data)
    print('Event name:', event.name)
    print('Supported types:', event.types)
    print('Modifier keys:', event.modifiers)
    print('Supported source types:', event.supportedsourcetypes)
    print('Operation type:', event.type)
    print('Source types:', event.sourcetypes)
    print('Supported target types:', event.supportedtargettypes)
    print('Widget:', event.widget, '(type: %s)' % type(event.widget))
    print('X:', event.x'root)
    print('Y:', event.y'root, '"n')
```

```
Label(root).grid()

list_frame = Frame(root)
list_frame.grid()

listbox = Listbox(list_frame, selectmode="extended", width=70, height=15)
listbox.grid(row=1, column=0, columnspan=2, padx=20, pady=20, sticky='news')

text = Text()

def btnclick():
    list_file = []
    files = filedialog.askopenfilenames(initialdir="/",title = "select file",filetypes = (("*.csv","*csv"),('all files', '*.*')))
    for f in files:
        if os.path.exists(f):
                listbox.insert('end', f)
    if files == '':
        messagebox.showwarning("Warning", "ファイルを選択してください")

def read_log():
    result = []
    try:
        f = open('log.csv','r', encoding='utf-8')
        rdr = csv.reader(f)

        for line in rdr:
            result +=line
        f.close()
        return result
    except:
        return result

def write_log(list_name):
    f = open('log.csv','a', newline='', encoding="utf-8")
    wr = csv.writer(f)
    for i in range(0,len(list_name)):
        wr.writerow([list_name[i]])
    f.close()

def drop_enter(event):
    event.widget.focus_force()
    return event.action

def drop_position(event):
    return event.action

def drop_leave(event):
    return event.action

def drop(event):
    if event.data:
        if event.widget == listbox:
            files = listbox.tk.splitlist(event.data)
            global path_name
            path_name = sorted(listbox.tk.splitlist(event.data))
            path_name_folder = []
            for i in range(0,len(path_name)):
                path = path_name[i].split('/')
                file_path = path[-1]
                if file_path[-3:] == "csv":
                    break
                else:
                    file_list=os.listdir(path_name[i])
                    file_list_csv=[file for file in file_list if file.endswith(".csv")]
                    file_list_csv_all= [path_name[i]+ '/' + file_list_csv[j] for j in range(0,len(file_list_csv))]
                    path_name_folder += file_list_csv_all
            path_name = path_name_folder
            global all_file
            all_file = len(path_name)
            already_saved = read_log()
            if already_saved:
                path_name = list(set(path_name) - set(already_saved))
            for f in files:
                if os.path.exists(f):
                    listbox.insert('end', f)
        elif event.widget == text:
            bd = text['bd'] + text['highlightthickness']
            x = event.x_root - text.winfo_rootx() - bd
            y = event.y_root - text.winfo_rooty() - bd
            index = text.index('@%d,%d' % (x,y))
            text.insert(index, event.data)
    return event.action
```

```
listbox.drop_target_register(DND_FILES, DND_TEXT)
text.drop_target_register(DND_TEXT)

for widget in (listbox, text):
    widget.dnd_bind('<<DropEnter>>', drop_enter)
    widget.dnd_bind('<<DropPosition>>', drop_position)
    widget.dnd_bind('<<DropLeave>>', drop_leave)
    widget.dnd_bind('<<Drop>>', drop)

def drag_init_listbox(event):
    print_event_info(event)
    data = ()
    if listbox.curselection():
        data = tuple([listbox.get(i) for i in listbox.curselection()])
    return ((ASK, COPY), (DND_FILES, DND_TEXT), data)

def drag_init_text(event):
    print_event_info(event)
    data = ''
    sel = text.tag_nextrange(SEL, '1.0')
    if sel:
        data = text.get(*sel)
    return (COPY, DND_TEXT, data)

def drag_end(event):
    print('Drag ended for widget:', event.widget)

class DatabaseConnector:
    def __init__(self):
      self.conn = self.cur = None
      self.connect2db()


    def __del__(self):
      self.conn.close()


    def connect2db(self):
        self.conn = mysql.connector.connect(
            host=ip_address,
            port='3306',
            user='kindergarten',
            password='Chi-kuma3',
            database='kindergarten',
            auth_plugin="mysql_native_password"
        )

        self.cur = self.conn.cursor(buffered=True)

    def insert_test_data(self):
        for i in range(0,len(path_name)):
            path = path_name[i].split('/')
            file_path = path[-1]

            if file_path[11:17] == "Hourly" and file_path[11:16] != "Daily":
                file = pd.read_csv(path_name[i], encoding="shift-jis", skiprows = 6, "
                      names=['date','steps','active_steps','total_calories','active_calories','active_time','time','Nan'])

                date1 = file['date'].tolist()
                id1 = [file_path[0:10] for i in range(0,len(date1))]
                step1 = file['steps'].tolist()
                act_step1 = file['active_steps'].tolist()
                tot_cal1 = file['total_calories'].tolist()
                act_cal1 = file['active_calories'].tolist()
                act_time1 = file['active_time'].tolist()
                time1 = file['time'].tolist()
                data = []
                for dates, ids, steps, act_steps, tot_cals, act_cals, act_times, times in "
                zip(date1, id1, step1, act_step1, tot_cal1, act_cal1, act_time1, time1):
                    data.append((dates, ids, steps, act_steps, tot_cals, act_cals, act_times, times))
            sql = f"INSERT INTO `activity_tracker` (`date`, `id`,`steps`, `active_steps`, `total_calories`, `active_calories`, `active_time`, `time`) "
                      VALUES (%s, %s, %s, %s, %s, %s, %s, %s) ON DUPLICATE KEY UPDATE steps = VALUES(steps),  ac-
tive_steps = VALUES(active_steps), "
                        total_calories  = VALUES(total_calories), active_calories = VALUES(active_calories) , active_time = VAL-
UES(active_time), time = VALUES(time);"

                file2 = pd.read_csv(path_name[i], encoding="shift-jis", nrows=4, names=['get','data','nan'])
                datas = file2['data'].tolist()
                height = int(datas[0])
                weight = int(float(datas[1]))
                age = int(datas[3])
                if datas[2] == "女性":
                    sex = 'f'
                else:
```

```
                sex = 'm'
            sql2 = f"INSERT INTO `student`information` (`activity`id`, `sex`, `age`, `height`, `weight`) VALUES (%s, %s, %s, %s, %s) "
                    ON DUPLICATE KEY UPDATE age = VALUES(age), height = VALUES(height), weight = VALUES(weight);"
                self.cur.executemany(sql, data)
                self.cur.execute(sql2,[file`path[0:10], sex, age, height, weight])
                self.conn.commit()

            elif file`path[11:17] != "Hourly" and file`path[11:16] != "Daily" and file`path[0:2] != ".":
                file = pd.read`csv(path`name[i], encoding="shift-jis", skiprows=1, names=['date','METs'])
                date2 = file['date'].tolist()
                met2 = file['METs'].tolist()
                id2 = [file`path[0:10] for i in range(0,len(date2))]
                data = []
                for dates, ids, mets in zip(date2, id2, met2):
                    if mets != 0:
                        data.append((dates, ids, mets))
                  sql = f"INSERT INTO `activity`tracker` (`date`, `id`, `mets`) VALUES (%s, %s, %s) ON DUPLICATE KEY UP-
DATE mets = VALUES(mets);"
                self.cur.executemany(sql, data)
                self.conn.commit()


def scp`transmit`file():
    with paramiko.SSHClient() as ssh:
        ssh.set`missing`host`key`policy(paramiko.AutoAddPolicy())

        print('[ssh 接続]')
        ssh.connect(ip`address, port=22, username='ubuntu', password='chi-kuma')

        print('[SCP 転送開始]')
        for i in range(0,len(path`name)):
            with scp.SCPClient(ssh.get`transport()) as scpc:
                path = path`name[i].split('/')
                file`path = path[-1]
                file = fr"-path`name[i]"".replace('"u3000', '  ')
                scpc.put(file, f'/home/ubuntu/activity`data/-file`path[0:10]"')

def main():
    database`connector = DatabaseConnector()
    database`connector.insert`test`data()
    scp`transmit`file()
    write`log(path`name)
    now = dt.datetime.now()
    MsgBox = messagebox.showinfo(f"-now.strftime("%Y-%m-%d %H:%M:%S")"',f'-all`file" の中 -len(path`name)" "
            個のデータ転送完了"n"nアプリを終了しますか．')
    if MsgBox == True:
        root.destroy()

buttonbox = Frame(root)
buttonbox.grid(row=2, column=0, columnspan=2, pady=5)
Button(buttonbox, text=' 保存', command=main).pack(side=LEFT, padx=5)

listbox.drag`source`register(1, DND`TEXT, DND`FILES)
text.drag`source`register(3, DND`TEXT)

listbox.dnd`bind('¡¡DragInitCmd¿¿', drag`init`listbox)
listbox.dnd`bind('¡¡DragEndCmd¿¿', drag`end)
text.dnd`bind('¡¡DragInitCmd¿¿', drag`init`text)
root.update`idletasks()
root.deiconify()
root.mainloop()
```

A.2 10시부터 11시까지의 카메라 영상을 저장하는 프로그램

```
#!/bin/sh
DATE=`date +"%Y%m%d`%H%M%S"`

DAY=`date +"%Y%m%d`%H"`

LOG`DATE=`date +"%Y%m%d"`

cam111="`cam111"
cam112="`cam112"
cam113="`cam113"
cam114="`cam114"
cam`all="`4in1"

FNAME111=$DATE$cam111
FNAME112=$DATE$cam112
FNAME113=$DATE$cam113
```

```
FNAME114=$DATE$cam114
FNAME'ALL=$DAY$cam'all
FNAME'LOG=$LOGDATE

date'now=‘date +"%Y"/"%m"/"%d"/"%H"‘
date'now'log=‘date +"%Y"/"%m"/"%d"‘

path111="/hdd/kindergarten/$date'now/camera111"
path112="/hdd/kindergarten/$date'now/camera112"
path113="/hdd/kindergarten/$date'now/camera113"
path114="/hdd/kindergarten/$date'now/camera114"
path'storytelling="/hdd/kindergarten/$date'now/storytelling"

if [ ! -e $path111 ]; then
  mkdir -p $path111
fi

if [ ! -e $path112 ]; then
  mkdir -p $path112
fi

if [ ! -e $path113 ]; then
  mkdir -p $path113
fi

if [ ! -e $path114 ]; then
  mkdir -p $path114
fi

gst-launch-1.0 rtspsrc location=rtsp://admin:chi-kuma@172.16.0.111:554/live/main latency = 10000 ! “
 rtph264depay ! h264parse ! matroskamux ! filesink location=$path111/$FNAME111.mkv & “
gst-launch-1.0 rtspsrc location=rtsp://admin:chi-kuma@172.16.0.112:554/live/main latency = 10000 ! “
rtph264depay ! h264parse ! matroskamux ! filesink location=$path112/$FNAME112.mkv & “
gst-launch-1.0 rtspsrc location=rtsp://admin:chi-kuma@172.16.0.113:554/live/main latency = 10000 ! “
rtph264depay ! h264parse ! matroskamux ! filesink location=$path113/$FNAME113.mkv & “
gst-launch-1.0 rtspsrc location=rtsp://admin:chi-kuma@172.16.0.114:554/live/main latency = 10000 ! “
rtph264depay ! h264parse ! matroskamux ! filesink location=$path114/$FNAME114.mkv & “

sleep 1870
killall -2 gst-launch-1.0 && “

ffmpeg “
-i $path111/$FNAME111.mkv “
-ss 00:01:00 -t 1800 -vcodec copy -acodec copy -an -f mp4 $path111/$FNAME111.mp4 && “

ffmpeg “
-i $path112/$FNAME112.mkv “
-ss 00:01:00 -t 1800 -vcodec copy -acodec copy -an -f mp4 $path112/$FNAME112.mp4 && “

ffmpeg “
-i $path113/$FNAME113.mkv “
-ss 00:01:00 -t 1800 -vcodec copy -acodec copy -an -f mp4 $path113/$FNAME113.mp4 && “

ffmpeg “
-i $path114/$FNAME114.mkv “
-ss 00:01:00 -t 1800 -vcodec copy -acodec copy -an -f mp4 $path114/$FNAME114.mp4 && “

echo "start: "$DATE "END: "‘date +"%Y%m%d'%H%M%S"‘ && “
echo ""

exit 0;
```

A.3 책 읽어주기 시간대를 판단하는 프로그램

```
import time
import cv2
import csv
import numpy as np
import glob
from datetime import datetime, timedelta
import cvlib as cv
from cvlib.object'detection import draw'bbox
import sys
import os
import math
from PIL import Image
import diagonal'crop

# ————————————————————————————
#                          初期設定部分
# ————————————————————————————
```

```
year = sys.argv[1]
month = sys.argv[2]
date = sys.argv[3]

video_directory = f"/hdd/kindergarten/{year}/{month}/{date}/10"
intrinsic_parameters = "/home/ubuntu/camera/calibration_param/camera4/mtx.npy" #カメラ行列
distortion_coeffcients = "/home/ubuntu/camera/calibration_param/camera4/dist.npy" #歪みパラメータ

people_num = 29 # 園児の数
ARmarkers_per_person = 3 # 園児一人当たりにつける AR マーカーの数
ARmarker_num_per_pix = 50 # ピクセルごとの AR マーカー生成数
marker_length = 0.08 # AR マーカーのサイズ

col_names = ["date", "id", "cam_number" ,
    "c1_x" ,"frame"]

col_dic = {col_name : num for num ,col_name in enumerate(col_names)}

# ------------------------------------------
#                 関数設定
# ------------------------------------------
class MakePathArray:
    def __init__(self, directory_name, extension):
        self.file_paths = glob.glob(directory_name + '/**/*' + extension)
        self.file_paths.sort()

class ArmarkerDetecter:
    def __init__(self):
        self.dictionary_4pix = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_4X4_50)
        self.dictionary_5pix = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_5X5_50)

    def setparam(self, video_path, marker_length,
        intrinsic_parameters, distortion_coeffcients
    ):
        self.marker_length = marker_length
        self.video_path = video_path
        self.cam_number = self._get_cam_number(video_path)
        self.intrinsic_parameters = np.load(intrinsic_parameters)
        self.distortion_coeffcients = np.load(distortion_coeffcients)
        self.book_data = []
        self.person_data = []

    def _get_cam_number(self, path):
        self.file_name = (path.split('/'))[-1]
        cam_number = self.file_name[-5]
        return cam_number

    def get_4corner_datas(self):
        video = cv2.VideoCapture(self.video_path)
        frame_count = 0
        datetime_string = year+month+date+"_100000"
        datetime_format = "%Y%m%d_%H%M%S"
        date_time = datetime.strptime(datetime_string, datetime_format)
        frame = round(1/video.get(cv2.CAP_PROP_FPS),1)
        angle = (math.pi * 13) / 180
        base = (0, 210)
        height = 500
        width = 900

        while True:
            people = 0
            info = []
            y_1=0
            y_2=0

            ret, img = video.read()
            if not ret: break

            fourcorner_param = self._detect_markers(img)
            data_frame = self._make_data_frame(frame_count,date_time)
            data = self._put_param(data_frame, fourcorner_param)
            check_start = [i[3] for i in data]

            if check_start.count(None) != 77:
                im = Image.fromarray(img)
                cropped_im = diagonal_crop.crop(im, base, angle, height, width)
                numpy_image = np.array(cropped_im)
                bbox, label, conf = cv.detect_common_objects(numpy_image)

                for _ in range(0,len(label)):
                    info.append([bbox[_],label[_],conf[_]])

                for _ in range(0,len(label)):
                    if "person" in info[_]:
```

```
                    people += 1

                if people == 1:
                    y`1=info[0][0][1]
                    y`2=info[0][0][3]
                    height` = y`2 - y`1
                if height` ¿= 370 and height` ¡= 510:
                    self.person`data.append(date`time)
            date`time += timedelta(seconds=frame)
            frame`count += 1
        video.release()

        return self.person`data

    def `detect`markers(self, img):
        corners`4pix, ids`4pix, rejectedImgPoints`4pix = cv2.aruco.detectMarkers(img, self.dictionary`4pix)
        corners`5pix, ids`5pix, rejectedImgPoints`5pix = cv2.aruco.detectMarkers(img, self.dictionary`5pix)

        return  -
            'corners' : [corners`4pix, corners`5pix],
            'ids' : [ids`4pix, ids`5pix],
            'rejectedImgPoints' : [rejectedImgPoints`4pix,
                rejectedImgPoints`5pix]
        "

    def `make`data`frame(self, frame`count ,date`time):
        empty`data = [
            [date`time, id, self.cam`number,
            None, frame`count]
            for id in range(people`num * ARmarkers`per`person)]
        return empty`data

    def `put`param(self, data`frame, fourcorner`param):
        for pixel`num, (ids, corners) in enumerate(zip(fourcorner`param['ids'], fourcorner`param['corners'])):
            if corners:
                ids = np.squeeze(ids)
                if np.ndim(ids) == 0:
                    ids = np.array([ids])

                corners = np.squeeze(corners)
                if np.ndim(corners) == 2:
                    corners = np.array([corners])
                for id, corner in zip(ids, corners):
                    if pixel`num == 1:
                        id += ARmarker`num`per`pix
                    if id ¡ people`num * ARmarkers`per`person:
                        for values in enumerate(corner):
                            data`frame[id][col`dic["c1`x"]] = values[0]

        return data`frame

def main():
    start`time = 0
    start`time`title = 0
    end`time = 0
    end`time`title = 0
    index = 0
    index`start = 0

    video`paths = MakePathArray(video`directory, 'mp4')
    armarker`detecter = ArmarkerDetecter()

    armarker`detecter.setparam(video`paths.file`paths[3], marker`length,intrinsic`parameters, distortion`coeffcients)
    data`person = armarker`detecter.get`4corner`datas()

    if len(data`person) != 0:
        successive`time = []
        for i in range(1,len(data`person)-50):
            index = 0
            for ` in range(1,50):
             if float(data`person[i + `].time().strftime('%M%S.%f')) - float(data`person[i + ` - 1].time().strftime('%M%S.%f')) ¡= 1.0:
                    index +=1
            if index == 49:
                index`start= i-1
                break

        start`time = data`person[index`start].time().strftime('00:%M:%S.%f')
        start`time`title = data`person[index`start].time().strftime('%H%M%S.%f')
        end`time = (data`person[-1] + timedelta(seconds=2)).time().strftime('00:%M:%S.%f')
        end`time`title = (data`person[-1] + timedelta(seconds=2)).time().strftime('%H%M%S.%f')
        print(start`time,end`time)

        os.system(
            f" "
```

```
            ffmpeg -i -video'paths.file'paths[0]" -ss -start'time" -to -end'time" "
            -vcodec copy -acodec copy -video'directory"/storytelling/-year"-month" "
            -date"'-start'time'title"'-end'time'title"'cam11-video'paths.file'paths[0][-5]".mp4 && "
            ffmpeg -i -video'paths.file'paths[1]" -ss -start'time" -to -end'time" "
            -vcodec copy -acodec copy -video'directory"/storytelling/-year"-month" "
            -date"'-start'time'title"'-end'time'title"'cam11-video'paths.file'paths[1][-5]".mp4 && "
            ffmpeg -i -video'paths.file'paths[2]" -ss -start'time" -to -end'time" "
            -vcodec copy -acodec copy -video'directory"/storytelling/-year"-month" "
            -date"'-start'time'title"'-end'time'title"'cam11-video'paths.file'paths[2][-5]".mp4 && "
            ffmpeg -i -video'paths.file'paths[3]" -ss -start'time" -to -end'time" "
            -vcodec copy -acodec copy -video'directory"/storytelling/-year"-month" "
            -date"'-start'time'title"'-end'time'title"'cam11-video'paths.file'paths[3][-5]".mp4")

        os.system(
            f' "
            ffmpeg  "
            -i -video'directory"/storytelling/-year"-month"-date"'-start'time'title"' "
            -end'time'title"'cam11-video'paths.file'paths[0][-5]".mp4 "
            -i -video'directory"/storytelling/-year"-month"-date"'-start'time'title"' "
            -end'time'title"'cam11-video'paths.file'paths[1][-5]".mp4 "
            -i -video'directory"/storytelling/-year"-month"-date"'-start'time'title"' "
            -end'time'title"'cam11-video'paths.file'paths[2][-5]".mp4 "
            -i -video'directory"/storytelling/-year"-month"-date"'-start'time'title"' "
            -end'time'title"'cam11-video'paths.file'paths[3][-5]".mp4 "
            -filter'complex " "
            [0:v]scale=648*486[v0]; "
            [1:v]scale=648*486[v1]; "
            [2:v]scale=648*486[v2]; "
            [3:v]scale=648*486[v3]; "
            [v0][v1][v2][v3]xstack=inputs=4:layout=0'0—w0'0—0'h0—w0'h0[v]" "
            -map "[v]" -c:v libx264 -preset veryfast -acodec copy -an -f mp4 "
            -video'directory"/-year"-month"-date"'-start'time'title"'-end'time'title"'4in1.mp4')

# ————————————————————————————
#                    実行
# ————————————————————————————
if ''name'' == "'main'":
    main()
```

A.4 データを Web 상에 표시하는 첫 페이지의 프로그램 ラム

```
#!/usr/bin/python3
import mysql.connector
import matplotlib.pyplot as plt
import sys

# ————————————————————————————
#                    初期設定部分
# ————————————————————————————
print("Content-Type: text/html")
print()

def main():
  html1 = """
  ¡!DOCTYPE html¿
  ¡html¿
    ¡head¿
      ¡meta http-equiv="Content-Type" content="text/html; charset=utf-8"¿
      ¡link rel="stylesheet" href="./css/style.css?after"¿
      ¡script src= "./js/myfunction.js?after"¿¡/script¿
      ¡title¿test¡/title¿
    ¡/head¿

    ¡body¿
      ¡header¿
        ¡a href="./index2.py"¿
          ¡p class="title"¿運動と集中力の関係の解析¡/p¿
        ¡/a¿
      ¡/header¿
      ¡div id="date"¿
        ¡label for="date"¿日付を選択してください¡/label¿
      ¡/div¿
      ¡div id="input"¿
        ¡form action="./cgi-bin/create2.py" method="POST"¿
          ¡input type="date" class="input'date" name="date" style="font-size:20px;"¿
          ¡input type="submit" class="input'submit" onclick="input()" value="確認" style="font-size:20px;"¿
```

```
        ¡/form¿
      ¡/div¿
    ¡/body¿
  ¡/html¿
  """
  print(html1)

# ————————————————————————————————
#                実行
# ————————————————————————————————
if ``name`` == ```main```:
    main()
```

A.5 データを Web 上に表示するページの プログラム

```
#!/usr/bin/python3

import cgi
import cgitb
import mysql.connector
import matplotlib.pyplot as plt
from matplotlib import dates
import pandas as pd
import numpy as np
import datetime as dt
import japanize˙matplotlib
import os
import glob

# ————————————————————————————————
#                初期設定部分
# ————————————————————————————————
cgitb.enable()
global date
global sort
global student˙num
sort = "0"
cam˙num = "all"
student˙num = "0"
print("Content-Type: text/html")
print()

form = cgi.FieldStorage()
for data in form:
  if data == 'date':
    date = form[data].value
  elif data == 'sort':
    sort = form[data].value
  elif data == 'cam':
    cam˙num = form[data].value
  elif data == 'choice':
    student˙num = form[data].value

# ————————————————————————————————
#                関数設定
# ————————————————————————————————
class DatabaseConnector:
  def ``init``(self):
    self.conn = self.cur = None
    self.connect2db()

  def ``del``(self):
    self.cur.close()
    self.conn.close()

  def connect2db(self):
    self.conn = mysql.connector.connect(
      host='localhost',
      port='3306',
      user='kindergarten',
      password='Chi-kuma3',
      database='kindergarten',
      auth˙plugin="mysql˙native˙password"
    )

    self.cur = self.conn.cursor(buffered=True)

  def select˙student˙information˙data(self):
    sql = f"SELECT * FROM ‘student˙information‘ ORDER BY number;"
    self.cur.execute(sql)
    self.conn.commit()
```

```
    informations = self.cur.fetchall()
    inforamtion = []

    for x in informations:
      inforamtion.append(x)

    return inforamtion

  def select'activity'tracker'data(self, activity'id, slect'date):
    sql = f"SELECT * FROM 'activity'tracker' WHERE id=-activity'id" "
        AND 'date' BETWEEN '-slect'date" 07:00:00.00' AND '-slect'date" 11:00:00.00';"
    self.cur.execute(sql)
    self.conn.commit()

    informations = self.cur.fetchall()
    inforamtion = []

    for x in informations:
      inforamtion.append(x)

    return inforamtion

class createGraph:
  def createMinGraph(self,info,start,end,stud'num):
    exercise'start=0
    exercise'start'index=0
    exercise'time = 0
    if start == 0 and end == 0:
      date = pd.date'range("08:00","11:00",freq="1min").time
      del info[0:60]
      mets = list(zip(*info))[2]

      fig = plt.figure(figsize=(9,6))
      pd.plotting.register'matplotlib'converters()
      ax = fig.add'subplot(1,1,1)
      ax.plot(date, mets, color='blue')

      ax.set'xticks(pd.date'range("08:00","11:00",freq="5min").time)
      ax.set'yticks(['+1 for ' in range(10) ])

      plt.xticks(rotation=90)
      plt.xlabel("時間")
      plt.ylabel("METs")
      plt.title(f'出席番号-stud'num"')
      plt.savefig('./graph1.png', format='png')

    else:
      date = pd.date'range("08:00","11:00",freq="1min").time
      start'min = ((int(start[0:2]) - 8) * 3600 + int(start[2:4]) * 60 + int(start[4:6])) // 60
      end'min = ((int(end[0:2]) - 8) * 3600 + int(end[2:4]) * 60 + int(end[4:6])) // 60
      del info[0:60]
      mets = list(zip(*info))[2]
      for ' in range(0,len(mets)):
        if float(mets[']) ¿= 3.0:
          exercise'start = date['].strftime('%H:%M:%S')
          exercise'start'index = '
          date = np.delete(date,slice(0,'+1))
          mets = np.delete(mets,slice(0,'+1))
          break
      exercise'time =  dt.datetime.strptime(start[0:4],"%H%M") - dt.datetime.strptime(exercise'start[0:5],"%H:%M")

      fig = plt.figure(figsize=(9,5))
      pd.plotting.register'matplotlib'converters()
      ax = fig.add'subplot(1,1,1)
      ax.plot(date[:end'min-exercise'start'index+1], mets[:end'min-exercise'start'index+1], color='blue')
      ax.plot(date[start'min-exercise'start'index+1:end'min-exercise'start'index+1], "
            mets[start'min-exercise'start'index+1:end'min-exercise'start'index+1], color='red',label='読み聞かせ中')
      ax.legend()
      ax.set'xticks(pd.date'range(f"-exercise'start[0:5]"",f'-end[0:2]":-end[2:4]"",freq="10min").time)
      ax.set'yticks(['+1 for ' in range(10) ])

      # plt.xticks(rotation=90)
      plt.xlabel("時間")
      plt.ylabel("METs")
      plt.title(f'出席番号-stud'num"')
      plt.savefig('./graph1.png', format='png')


class createHtmlCode:
  def createSelectCode(self, info):
    html'code = ""
    if info == "0":
      html'code = f"""
        ¡select name="sort" onchange="this.form.submit();" style="font-size:20px;"¿
```

```
        ¡option value="0" selected¿出席番号順¡/option¿
        ¡option value="1"¿運動強度高い順¡/option¿
        ¡option value="2"¿運動強度低い順¡/option¿
      ¡/select¿
    """
  elif info == "1":
    html`code = f"""
      ¡select name="sort" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="0"¿出席番号順¡/option¿
        ¡option value="1" selected¿運動強度高い順¡/option¿
        ¡option value="2"¿運動強度低い順¡/option¿
      ¡/select¿
    """
  elif info == "2":
    html`code = f"""
      ¡select name="sort" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="0"¿出席番号順¡/option¿
        ¡option value="1"¿運動強度高い順¡/option¿
        ¡option value="2" selected¿運動強度低い順¡/option¿
      ¡/select¿
    """
  return html`code

def createSelectMovieCode(self, info):
  html`code = ""
  if info == "all":
    html`code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all" selected="selected"¿全体¡/option¿
        ¡option value="cam1"¿カメラ１¡/option¿
        ¡option value="cam2"¿カメラ２¡/option¿
        ¡option value="cam3"¿カメラ３¡/option¿
        ¡option value="cam4"¿カメラ４¡/option¿
      ¡/select¿
    """
  elif info == "cam1":
    html`code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all"¿全体¡/option¿
        ¡option value="cam1" selected="selected"¿カメラ１¡/option¿
        ¡option value="cam2"¿カメラ２¡/option¿
        ¡option value="cam3"¿カメラ３¡/option¿
        ¡option value="cam4"¿カメラ４¡/option¿
      ¡/select¿
    """
  elif info == "cam2":
    html`code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all"¿全体¡/option¿
        ¡option value="cam1"¿カメラ１¡/option¿
        ¡option value="cam2" selected="selected"¿カメラ２¡/option¿
        ¡option value="cam3"¿カメラ３¡/option¿
        ¡option value="cam4"¿カメラ４¡/option¿
      ¡/select¿
    """
  elif info == "cam3":
    html`code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all"¿全体¡/option¿
        ¡option value="cam1"¿カメラ１¡/option¿
        ¡option value="cam2"¿カメラ２¡/option¿
        ¡option value="cam3" selected="selected"¿カメラ３¡/option¿
        ¡option value="cam4"¿カメラ４¡/option¿
      ¡/select¿
    """
  elif info == "cam4":
    html`code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all"¿全体¡/option¿
        ¡option value="cam1"¿カメラ１¡/option¿
        ¡option value="cam2"¿カメラ２¡/option¿
        ¡option value="cam3"¿カメラ３¡/option¿
        ¡option value="cam4" selected="selected"¿カメラ４¡/option¿
      ¡/select¿
    """
  return html`code

def createResultTableCode(self, student`info, activity`info, select, radio, start):
  #date, id, mets, step, activestep, totalcalorie, calories, activetime, time#
  html`code = f"""
    ¡div id= 'tb' style="margin: 1% 5% 3% 5%;" ¿
      ¡table border="2" align = "center" style="font-size:15px; text-align:center;
       border-collapse: collapse; border:solid;border-width:3px;margin:0;width:100%;"¿
        ¡tr¿
```

```
          ¡td¿出席番号¡/td¿
          ¡td¿総 METs¡/td¿
          ¡td¿平均 METs¡/td¿
          ¡td¿最大 METs¡/td¿
          ¡td¿運動時間 [分]¡/td¿
          ¡td¿活動消費カロリー [kcal]¡/td¿
          ¡td¿運動強度¡/td¿
          ¡td style="border-right:double;border-right-width:9px;"¿詳しい情報¡/td¿
          ¡td¿出席番号¡/td¿
          ¡td¿総 METs¡/td¿
          ¡td¿平均 METs¡/td¿
          ¡td¿最大 METs¡/td¿
          ¡td¿運動時間 [分]¡/td¿
          ¡td¿活動消費カロリー [kcal]¡/td¿
          ¡td¿運動強度¡/td¿
          ¡td¿詳しい情報¡/td¿
        ¡/tr¿
        """
    all`data = []
    mets = [` for ` in range(0,29)]
    max`mets = [` for ` in range(0,29)]
    index` = 0
    end`index = int(start.split(".")[0][0:2])*60 - 420 + int(start.split(".")[0][2:4])

    for i in range(0,29):
      weight =  float(student`info[i][9])
      step = int(activity`info[i][60][3]) + int(activity`info[i][120][3])
      activestep = int(activity`info[i][60][4]) + int(activity`info[i][120][4])
      totalcalories = float(activity`info[i][60][5]) + float(activity`info[i][120][5])
      calories = float(activity`info[i][60][6]) + float(activity`info[i][120][6])
      activetime = int(activity`info[i][60][7]) + int(activity`info[i][120][7])
      total`time = int(activity`info[i][60][8]) + int(activity`info[i][120][8])


      for ` in range(0,end`index):
        if float(activity`info[i][`][2]) ¿= 3.0:
          index= `
          break
      mets[i] = [r[2] for r in activity`info[i][index:end`index]]
      average`mets = round(sum(mets[i])/len(mets[i]),2)
      all`mets = round(sum(mets[i]),2)
      max`mets[i] = max(mets[i])

      time = (float(activity`info[i][60][8]) + float(activity`info[i][120][8])) / 3600
      hour = total`time // 3600
      total`time = total`time - hour * 3600
      min = total`time // 60
      sec = total`time - min * 60
      # average`mets = round(calories / (eeight * time * 1.05),1)
      if activetime == 0:
        all`data.append([i+1,0,0,0,0,0,"x",""])
      else:
        if average`mets ¿= 2 and average`mets ¡ 3:
          if i+1 == int(radio):
            all`data.append([i+1,all`mets,average`mets,max`mets[i],len(mets[i]), round(totalcalories,1),"非常に楽" "
                                    ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
          else:
            all`data.append([i+1,all`mets,average`mets,max`mets[i],len(mets[i]),round(totalcalories,1),"非常に楽" "
                    ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();'onchange = 'outputVideo();'¿"])

        elif average`mets ¿= 3 and average`mets ¡ 4:
          if i+1 == int(radio):
            all`data.append([i+1,all`mets,average`mets,max`mets[i],len(mets[i]),round(totalcalories,1),"楽" "
                                    ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
          else:
            all`data.append([i+1,all`mets,average`mets,max`mets[i],len(mets[i]),round(totalcalories,1),"楽" "
                    ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();'onchange = 'outputVideo();'¿"])

        elif average`mets ¿= 4 and average`mets ¡ 5:
          if i+1 == int(radio):
            all`data.append([i+1,all`mets,average`mets,max`mets[i],len(mets[i]),round(totalcalories,1),"やや楽" "
                                    ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
          else:
            all`data.append([i+1,all`mets,average`mets,max`mets[i],len(mets[i]),round(totalcalories,1),"やや楽"
                    ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();'onchange = 'outputVideo();'¿"])

        elif average`mets ¿= 5 and average`mets ¡ 6:
          if i+1 == int(radio):
            all`data.append([i+1,all`mets,average`mets,max`mets[i],len(mets[i]),round(totalcalories,1),"ややきつめ" "
                                    ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
```

```
        else:
          all'data.append([i+1,all'mets,average'mets,max'mets[i],len(mets[i]),round(totalcalories,1)“
                          ,”ややきつめ”,f”¡input type='radio' name='choice' value='–i+1″' onclick='this.form.submit();' on-
change = 'outputVideo();'¿”])

      elif average'mets ¿= 6 and average'mets ¡ 7:
        if i+1 == int(radio):
          all'data.append([i+1,all'mets,average'mets,max'mets[i],len(mets[i]),round(totalcalories,1),”ややきつい” “
                          ,f”¡input type='radio' name='choice' value='–i+1″' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿”])
        else:
          all'data.append([i+1,all'mets,average'mets,max'mets[i],len(mets[i]),round(totalcalories,1),”ややきつい” “
                ,f”¡input type='radio' name='choice' value='–i+1″' onclick='this.form.submit();' onchange = 'outputVideo();'¿”])

      elif average'mets ¿= 7 and average'mets ¡ 8:
        if i+1 == int(radio):
          all'data.append([i+1,all'mets,average'mets,max'mets[i],len(mets[i]),round(totalcalories,1),”きつい” “
                          ,f”¡input type='radio' name='choice' value='–i+1″' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿”])
        else:
          all'data.append([i+1,all'mets,average'mets,max'mets[i],len(mets[i]),round(totalcalories,1),”きつい” “
                ,f”¡input type='radio' name='choice' value='–i+1″' onclick='this.form.submit();' onchange = 'outputVideo();'¿”])

      elif average'mets ¿= 8:
        if i+1 == int(radio):
          all'data.append([i+1,all'mets,average'mets,max'mets[i],len(mets[i]),round(totalcalories,1),”非常にきつい” “
                          ,f”¡input type='radio' name='choice' value='–i+1″' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿”])
        else:
          all'data.append([i+1,all'mets,average'mets,max'mets[i],len(mets[i]),round(totalcalories,1),”非常にきつい” “
                ,f”¡input type='radio' name='choice' value='–i+1″' onclick='this.form.submit();' onchange = 'outputVideo();'¿”])

    if select == ”0”:
      all'data.sort(key=lambda x:x[0])
    elif select == ”1”:
      all'data.sort(key=lambda x:-x[1])
    elif select == ”2”:
      all'data.sort(key=lambda x:x[1])

    for i in range(0,len(all'data)-15):
      html'code += f”””
        ¡tr¿
          ¡td¿–all'data[i][0]″¡/td¿
          ¡td¿–all'data[i][1]″¡/td¿
          ¡td¿–all'data[i][2]″¡/td¿
          ¡td¿–all'data[i][3]″¡/td¿
          ¡td¿–all'data[i][4]″¡/td¿
          ¡td¿–all'data[i][5]″¡/td¿
          ¡td¿–all'data[i][6]″¡/td¿
          ¡td style=”border-right:double;border-right-width:9px;”¿–all'data[i][7]″¡/td¿
          ¡td¿–all'data[i+15][0]″¡/td¿
          ¡td¿–all'data[i+15][1]″¡/td¿
          ¡td¿–all'data[i+15][2]″¡/td¿
          ¡td¿–all'data[i+15][3]″¡/td¿
          ¡td¿–all'data[i+15][4]″¡/td¿
          ¡td¿–all'data[i+15][5]″¡/td¿
          ¡td¿–all'data[i+15][6]″¡/td¿
          ¡td¿–all'data[i+15][7]″¡/td¿
        ¡/tr¿
      „„„
    html'code += f”””
        ¡tr¿
          ¡td¿–all'data[14][0]″¡/td¿
          ¡td¿–all'data[14][1]″¡/td¿
          ¡td¿–all'data[14][2]″¡/td¿
          ¡td¿–all'data[14][3]″¡/td¿
          ¡td¿–all'data[14][4]″¡/td¿
          ¡td¿–all'data[14][5]″¡/td¿
          ¡td¿–all'data[14][6]″¡/td¿
          ¡td style=”border-right:double;border-right-width:9px;”¿–all'data[14][7]″¡/td¿
          ¡td¿¡/td¿
          ¡td¿¡/td¿
          ¡td¿¡/td¿
          ¡td¿¡/td¿
          ¡td¿¡/td¿
          ¡td¿¡/td¿
          ¡td¿¡/td¿
          ¡td¿¡/td¿
        ¡/tr¿
      „„„
    html'code += ”¡/table¿¡/div¿”

    return html'code
```

```
  def createVideoCode(self,date,select):
    year = date[0:4]
    month = date[5:7]
    day = date[8:10]
    path = f"../video/kindergarten/{year}/{month}/{day}/10/"
    path_each = f"../video/kindergarten/{year}/{month}/{day}/10/storytelling/"
    files_path = glob.glob(f"{path}*.mp4")
    files_each_path = glob.glob(f"{path_each}*.mp4")
    files_each_path = sorted(files_each_path)

    html_code = ""

    if None not in files_path and len(files_path) != 0:
      file_name = files_path[0].split('/')[-1]
      start_time = file_name.split('_')[1]
      end_time = file_name.split('_')[2]
      html_code = f"""
        <h3 style = "margin:3% 3% 0% 3%;">読み聞かせ時間：
        {start_time[0:2]}:{start_time[2:4]}:{start_time[4:6]} ~ {end_time[0:2]}:{end_time[2:4]}:{end_time[4:6]}</h3>
        """
      if select == "all":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_path[0]}" type="video/mp4"></video>
        """
      elif select == "cam1":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_each_path[0]}" type="video/mp4"></video>
        """
      elif select == "cam2":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_each_path[1]}" type="video/mp4"></video>
        """
      elif select == "cam3":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_each_path[2]}" type="video/mp4"></video>
        """
      elif select == "cam4":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_each_path[3]}" type="video/mp4"></video>
        """
    else:
      html_code = "<h3>動画データが存在しません</h3>"
      start_time = 0
      end_time = 0

    return html_code, start_time, end_time

def main():
  database_connector = DatabaseConnector()
  create_html_code = createHtmlCode()
  create_graph = createGraph()
  activity_tracker_information = []
  student_information = database_connector.select_student_information_data()
  student_id =list(zip(*student_information))[2]

  for index in student_id:
    activity_tracker_information.append(database_connector.select_activity_tracker_data(index,date))

  select_code = create_html_code.createSelectCode(sort)
  select_movie_code = create_html_code.createSelectMovieCode(cam_num)
  video_code, start_time, end_time = create_html_code.createVideoCode(date,cam_num)

 html_table_code = create_html_code.createResultTableCode(student_information, activity_tracker_information, sort, student_num,start_time)
 if student_num != "0":
   create_graph.createMinGraph(activity_tracker_information[int(student_num)-1], start_time, end_time,student_num)
   img_html = '<img id="active" src="image1.py" style="max-width:95%">'
 else:
   img_html = ""

 html1 = """
 <!DOCTYPE html>
 <html>
   <head>
     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
     <link rel="stylesheet" href="../css/style.css?afsddd1231">
     <script src= "../js/myfunction.js"></script>
     <script src="../js/canvas-arrow.js"></script>
     <title>test</title>
   </head>

   <body>
     <header>
       <a href="../index2.py">
         <p class="title">運動と集中力の関係の解析</p>
```

```
      ¡/a¿
    ¡/header¿
    ¡div id="date"¿
      ¡label for="date"¿日付を選択してください¡/label¿
    ¡/div¿
    ¡form action="./create2.py" method="POST"¿
    ¡div id="input"¿
       ¡input type="date" class="input'date" name="date" value=%s  style="font-size:20px;"¿
       ¡input type="submit" class="input'submit" onclick="input()" value="確認"  style="font-size:20px;"¿
    ¡/div¿
    ¡div id="content"¿
      ¡div class = "all"¿%s %s¡/div¿
      ¡div class = "left"¿
 ¡h1¿活動量計データ¡/h1¿
         ¡div class = "up"¿¡/div¿
         ¡div class = "down"¿%s¡/div¿
      ¡/div¿
      ¡div class = "right"¿
       ¡div class = "title"¿
        ¡h1¿読み聞かせデータ¡/h1¿
          %s
          ¡button type="button" onClick="outputVideo()" id="btnResume"¿再生¡/button¿
          ¡button type="button" id="btnPause" onClick="Pause()"¿停止¡/button¿

       ¡/div¿
          %s
          ¡div id="cv"¿
            ¡canvas id="parent" width="864px" height="648px"¿¡/canvas¿
            ¡canvas id="children" width="864px" height="648px"¿¡/canvas¿
            ¡canvas id="children2" width="864px" height="648px"¿¡/canvas¿
            ¡canvas id="children3" width="864px" height="1048px"¿¡/canvas¿
          ¡/div¿
     ¡/div¿

   ¡/form¿
   ¡/body¿
 ¡/html¿
 """ % (date, select'code, html'table'code,img'html,select'movie'code,video'code)

 print(html1)

# ————————————————————————————
#                実行
# ————————————————————————————
if ''name'' == '''main''':
    main()
```

A.6 개인의 METs 그래프를 Web 상에 표시하는 프로그램

```
#!/usr/bin/python3

import sys
import os

src =  "//var/www/html/cgi-bin/graph1.png"
length = os.stat(src).st'size

sys.stdout.write("Content-Type: image/png“n")
sys.stdout.write("Content-Length: " + str(length) + "“n")
sys.stdout.write("“n")
sys.stdout.flush()
sys.stdout.buffer.write(open(src, "rb").read())
```

A.7 Web 디자인을 설정하는 프로그램

```
header−
    width: 100%;
    background-color: gray;
„

body −
    margin: 0%;
„
```

```
header a-
    text-decoration: none;
"

header a:visited, a:hover-
    color: black;
"

header-
    padding:2%;
"
#home-
    margin:auto;
"
header a .title-
    margin: 0%;
    font-size: 50px;
    text-align: center;
    color: black;
"

#date -
    margin-top: 1%;
    font-size: 25px;
    text-align: center;
"

#input-
    margin: 1%;
    font-size: 40px;
    text-align: center;
"
#input .input'date -
    width: 10%;
"

#content -
    width: 100%;

"
#content .all -
    text-align: center;
"

#content .left -
    width: 50%;
    float: left;
    border-right: 5pt;
    text-align: center;
"
/* #content .left .west-
    width: 50%;
    float: left;
    border-right: 5pt;
"

#content .left .east-
    width: 50%;
    float: right;
    border-right: 5pt;
" */

#tb -
    text-align: center;
    font-size: 20px;
"

#content .right -
    width: 50%;
    float:right;
    text-align: center;
"
/* #content .title,h3-
  text-align: center;
" */

#cv-
    text-align: center;
    position: relative;
    margin: 0 5% 0 5%;
"

canvas-
    position: absolute;
```

```
    display: block;
    margin: 0 auto;
"
```

A.8 개인의 책 읽어주기 동영상을 표시하는 프로그램

```
let pause = false

function csvToArray()-
    let csv = new XMLHttpRequest();

    // CSV ファイルへのパス
    csv.open("GET", "../video/kindergarten/2023/02/01/10/ex/0.csv", false);

    // csv ファイル読み込み失敗時のエラー対応
    try -
    csv.send(null);
    " catch (err) -
    console.log(err);
    "

    // 配列を定義
    let csvArray = [];

    // 改行ごとに配列化
    let lines = csv.responseText.split(/"r"n—"n/);

    // 1 行ごとに処理
    for (let i = 0; i ¡ lines.length; ++i) -
    let cells = lines[i].split(",");
    if (cells.length != 1) -
        csvArray.push(cells);
    "
    "

    return csvArray;
"

function outputVideo()-
    var student = document.querySelector("input[name='choice']:checked")
    var cam = document.getElementById("cam");
    if (student != null && cam != null) -
        var cam'num = cam.options[cam.selectedIndex].value;
        var student'num = student.value;
    "
    console.log(cam'num,student'num)

    var a = csvToArray();

    var video = document.getElementById("video");
    var video2 = document.getElementById("video2");


    if(video.paused)-
        video.play();
        video2.play();
    "else-
        video.pause();
        video2.pause();
    "
    pause = false;

    var parent = document.getElementById("parent");
    var child3 = document.getElementById("children3");
    var child = document.getElementById("children");
    var context = child.getContext("2d");
    // graph'active
    // var active = document.getElementById("active");
    // var child2 = document.getElementById("children2");
    // child2.getContext("2d").drawImage(active, 0, 448, 300, 200);
    var x = 0;
    var y = 0;
    var i = 0;

    for(c=0;c¡a.length;c++)-
        if(a[c][3]!=0)-
            x=a[c][3]/3;
            y=a[c][4]/3;
            break;
        "
    "
```

```
    setInterval(function()-
        if(pause)-
            return
        ″
        parent.getContext(”2d”).drawImage(video, 0, 0, 864, 648);
        child3.getContext(”2d”).drawImage(video2, 0, 648, 864, 400);


        if (student'num == ”1” && cam'num == ”cam4”)-
            context.clearRect(0, 0, 864, 648);
            context.font = '25px serif';
            context.fillStyle = 'white';
            context.textAlign = 'center';
            context.fillText(”出席番号1”, 432, 90);
            context.beginPath();
            context.arrow(420, 100, x, y, [0, 3, -20, 3, -20, 9]);
            context.fillStyle = 'white';
            context.fill();

            if(a[i][3] != 0)-
                x= (a[i][3])/3;
                y= (a[i][4])/3;
            ″
        ″
        i++;
    ″, 1000/5);
″

function Pause()-
  var video = document.getElementById(”video”);
  var video2 = document.getElementById(”video”);
  video.pause();
  video2.pause();
  pause = true;
″

window.onload = function() -
    var student = document.querySelector(”input[name='choice']:checked”)
    if (student != null) -
        target = document.getElementById(”video”);
        target2 = document.getElementById(”video2”);
    target.style.display =”none”;
    target2.style.display =”none”;
    ″
    else -
        target = document.getElementById(”video”);
        target2 = document.getElementById(”video2”);
        target.style.display =””;
        target2.style.display =””;
    ″

″
```

A.9 개인의 책 읽어주기 동영상의 그래프를 작성하는 프로그램

```
import numpy as np
import csv
import glob
import os
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import math
import japanize'matplotlib
import cv2

# ————————————————————————————————-
#                                初期設定部分
# ————————————————————————————————-
csv'read'file'1 = ”/hdd/kindergarten/code/123/20230201'100104.200000'101115.000000'cam114'4corner.csv”

col'names = [”date”, ”id”, ”cam'number” ,
    ”c1'x” ,”c1'y” ,
    ”c2'x” ,”c2'y” ,
    ”c3'x” ,”c3'y” ,
    ”c4'x” ,”c4'y” ,
    ”frame”
    ,”is'x”,”is'y”
```

```
    ,"tf`x","tf`y"]

CD = {col`name : num for num ,col`name in enumerate(col`names)}
id`names = ["id0","id1","id2","id3","id4","id5","id6","id7","id8"]

# ————————————————————————————————
#                関数設定
# ————————————————————————————————
class MakePathArray:
    def ``init``(self, directory`name, extension):
        self.file`paths = glob.glob(directory`name + '/*.' + extension)

class CSVOperator:
    def ``init``(self):
        pass

    def read`csv`as`2Darray(self, csv`path):
        with open(csv`path, "r") as f:
            reader = csv.reader(f)
            array = [row for row in reader]
        return array

    def save`2Darray`in`csv(self, csv`path, twoDarray):
        with open(csv`path, "w") as f:
            writer = csv.writer(f, lineterminator = '¥n')
            writer.writerows(twoDarray)
        pass

def intersection`point(line1,line2):
    a1,b1 = line1
    a2,b2 = line2
    x = (b2-b1)/(a1-a2)
    y = a1*x+b1
    return x,y

def make`line(x1,y1,x2,y2):
    #y=ax+b
    a = (y2-y1)/(x2-x1)
    b = y1-a*x1
    return a,b


def main():
    csv`operator = CSVOperator()
    csv`path1 = csv`read`file`1

    array`1 = csv`operator.read`csv`as`2Darray(csv`path1)

    arrays`1`casted = []
    for row in array`1:
        row`frame = {None,None,None,None,None,None,None,None,None,None,None,None,None,None,None,None,None}
        for col,value in enumerate(row):
            if (not value == '') and (not col == CD["date"]):
                row`frame[col] = float(value)
            elif col == CD["date"]:
                row`frame[col] = value
        arrays`1`casted.append(row`frame)

    arrays`2`casted = []

    targetid = 0
    #グラフ作成
    #targetid の時
    #arrays`1`casted の x-frame グラフ y-frame グラフ x-y グラフを作成
    #arrays`2`casted の x-frame グラフ y-frame グラフ x-y グラフを作成
    #6 つのグラフを一画面に表示

    #arrays`1`casted の x-frame グラフ y-frame グラフ x-y グラフを作成
    x1`frame = []
    y1`frame = []
    x1`y1 = []
    for row in arrays`1`casted:
        if row[CD["id"]] == targetid:
            x1`frame.append(row[CD["tf`x"]])
            y1`frame.append(row[CD["tf`y"]])
            x1`y1.append([row[CD["tf`x"]],row[CD["tf`y"]]])

    #arrays`2`casted の x-frame グラフ y-frame グラフ x-y グラフを作成
    x2`frame = []
    y2`frame = []
    x2`y2 = []
    for row in arrays`2`casted:
        if row[CD["id"]] == targetid:
            x2`frame.append(row[CD["tf`x"]])
```

```
            y2'frame.append(row[CD["tf'y"]])
            x2'y2.append([row[CD["tf'x"]],row[CD["tf'y"]]])

    fig = plt.figure(figsize=(8.64,4))
    ax1 = fig.add'subplot(1,2,1)
    ax2 = fig.add'subplot(1,2,2)
    ims = []
    for a in range(len(x1'frame)):
        line1 = ax1.plot(x1'frame[0:a],color='blue')
        ax1.set'title("x")
        ax1.set'xlabel("frame[5fps]")
        ax1.set'ylabel("x[mm]")

        line2 = ax2.plot(y1'frame[0:a],color='blue')
        ax2.set'title("y")
        ax2.set'xlabel("frame[5fps]")
        ax2.set'ylabel("y[mm]")
        ims.extend([line1+line2])
        fig.suptitle("出席番号1")
    ani = animation.ArtistAnimation(fig,ims)
    ani.save("anim.mp4",writer="ffmpeg")

    plt.show()

# ————————————————————————————————————
#                   実行
# ————————————————————————————————————
if ''name'' == '''main''':
    main()
```