

An abstract 3D composition featuring several geometric shapes: a blue cube, an orange cube, a yellow cube, and a dark grey cube, all arranged in a dynamic, overlapping manner. Two bright yellow spheres are also present, one near the top and one near the bottom left. The background is a dark blue gradient with a subtle, lighter blue circular pattern. The overall mood is modern and artistic.

감정 진단용 AI를 활용한 라이트테라피 조명 개발

EmotiLight 팀

선예빈
조영
최승희
최유현
정효원
박창선

목차

- 1 Datasets
- 2 SW (소프트웨어)
- 3 HW (하드웨어)
- 4 프로젝트 이슈
- 5 기업 자문
- 6 기대 효과 및 활용방안

실내조명의 색과 조도에 따른 감정 상태에 관한 연구

이 논문을 석사학위 논문으로 제출함

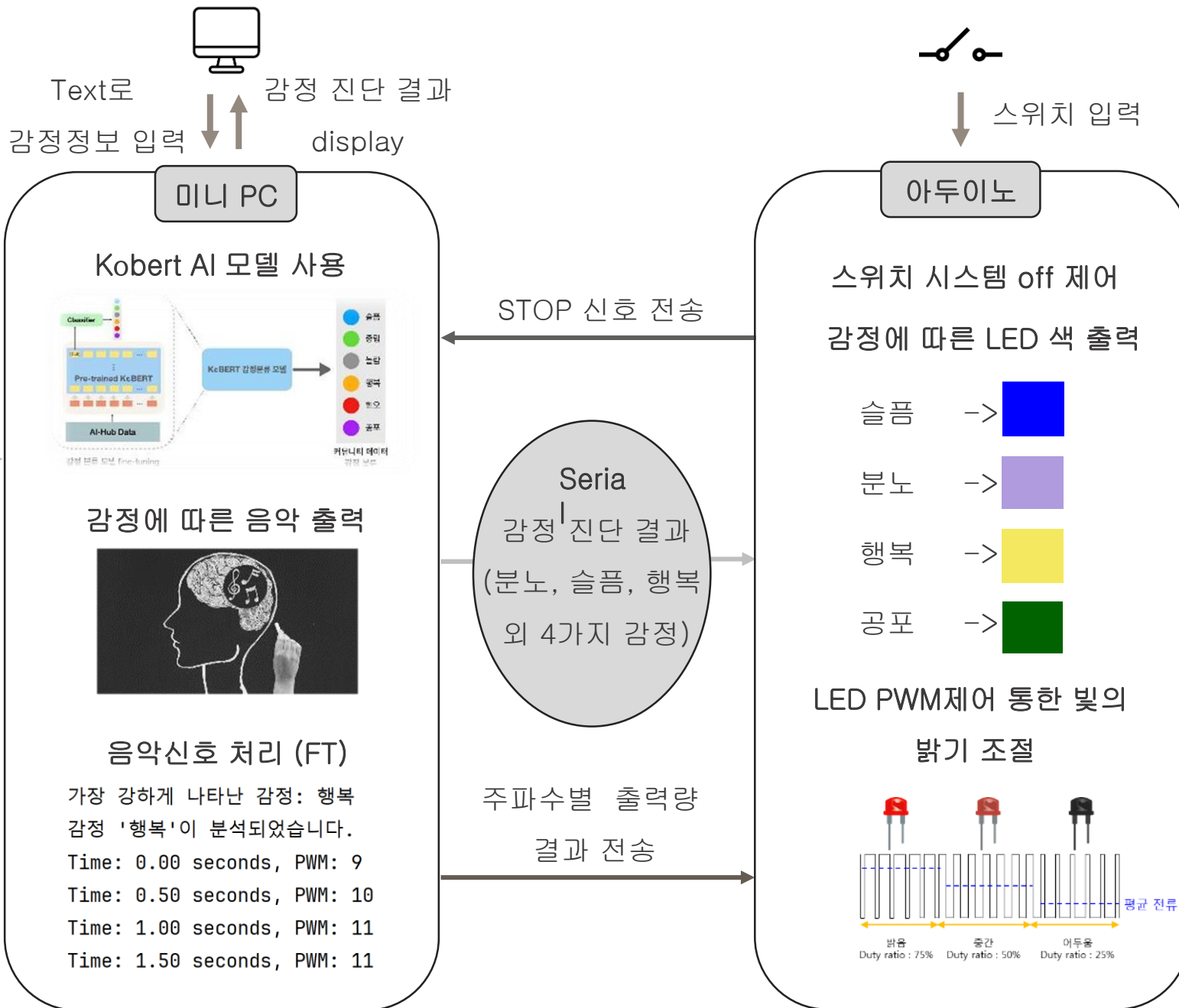
2013년 12월

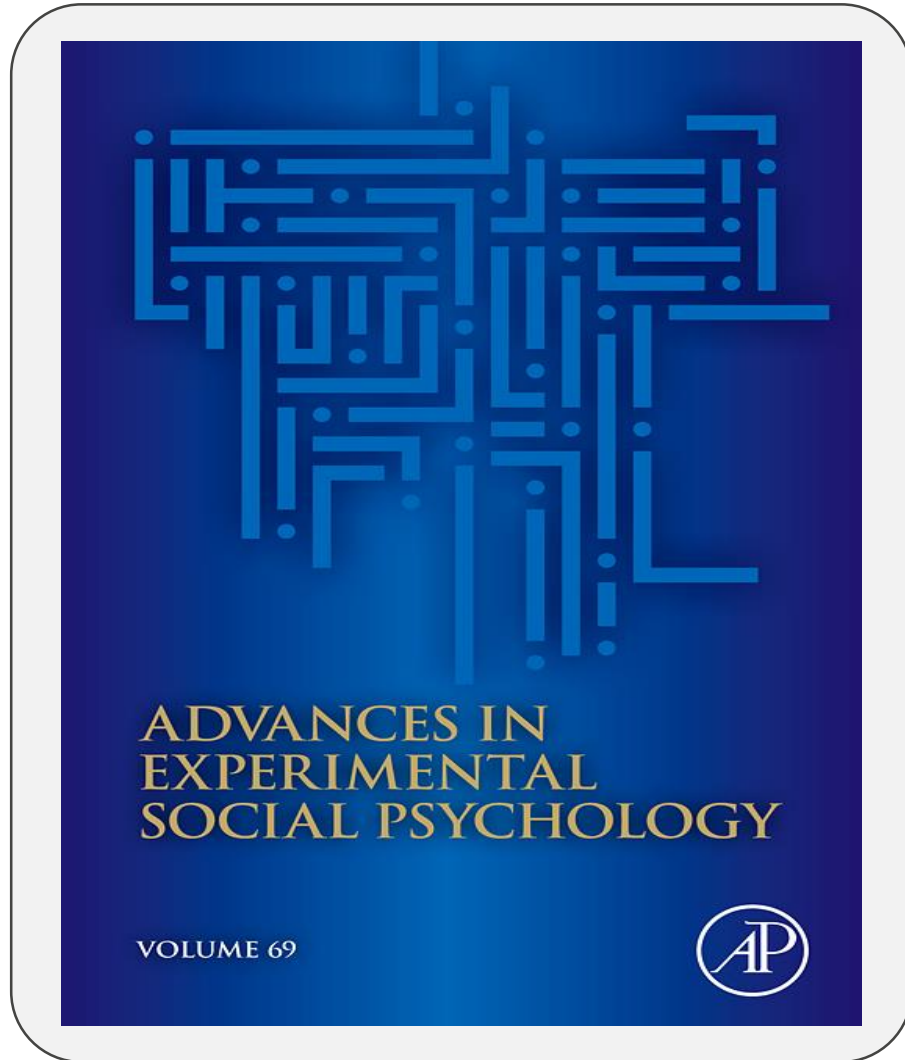
이화여자대학교 대학원

디자인학부 박진희

- 우리의 신체는 햇빛을 충분히 보지 못하면 계절성 정서 장애라는 병이 들 수 있음
- 긍정적인 감정을 느끼게 해주는 색상으로는 초록, 노랑, 주황 계열의 색이 상위권을 차지

대표 그림





<Elliot, A. J., & Maier, M. A. (2012). Color-in-context theory. *Advances in Experimental Social Psychology*, 45>

- 색상이 다양한 맥락에서 심리적 기능에 미치는 영향을 탐구함
- 저자들은 색상이 상황에 따라 다르게 해석되고 이는 개인의 행동과 태도에 영향을 줄 수 있다고 주장함
- 이 연구는 색상의 심리적 영향에 대한 이해를 넓히는 데 기여함

감정	색상	이유
슬픔	파랑	파랑 - 진정 효과가 있어 심박수와 혈압을 낮추는 효과가 있음, 평온함과 안정감을 줌
분노	보라	보라 - 긴장을 풀어주고 진정 효과가 있음
행복	노랑	노랑 - 햇빛을 연상시키며 따뜻함과 행복감을 주는 색, 긍정적인 감정을 자극하는 데 효과적임
공포	초록	초록 - 자연을 연상시키므로 마음을 안정시키고 스트레스를 줄이는 데 효과적임, 연구에 따르면 자연 속에서 시간을 보내는 것과 유사한 진정 효과를 제공함

Part 1 Datasets



한국어 감정 정보가 포함된 단발성 대화 데이터셋

#텍스트 감정 인식 #다분류 감정 #텍스트 데이터 레이블링

분야 미분류 유형 텍스트

조회수: 3,957 다운로드: 656 용량: 1.57 MB

한국어 감정 정보가 포함된 연속적 대화 데이터셋

#일상 대화문 #감정 레이블링 #챗봇

분야 미분류 유형 텍스트

조회수: 2,723 다운로드: 401 용량: 2.44 MB

- 단발성 대화 데이터 셋은 train, 연속적 대화 데이터 셋은 test로 나눔
- 문장 단위 작업을 수행할 수 있도록 문장 분리 작업을 거침
- 7개 감정 (공포, 놀람, 분노, 슬픔, 중립, 행복, 혐오) 레이블링 수행

Part 1

데이터 셋 구체화

id	document	label		
1	언니 동생으로 부르는게 맞는 일인가요...??	공포	놀람	5898
2	그냥 내 느낌일뿐겠지?	공포	공포	5468
3	아직너무초기라서 그런거죠?	공포	분노	5665
4	유치원버스 사고 났다던데	공포	슬픔	5267
5	근데 원래이런거맞나요	공포	중립	4830
6	남자친구가 떠날까봐요	공포	행복	6037
7	이거 했는데 허리가 아플수도 있나요? ;	공포	혐오	5429
8	내가불안해서꾸는걸까..	공포		
9	일주일도 안 남았당...ㅠㅠ	공포		
10	약은 최대한 안먹으려고 하는데좋은 음식있나요?0	공포		
11	구직활동 하면서 남는시간은 뭘로 활용해야 되지..	공포		
12	괜찮은분 같아서 괜히 조금해지네요 ㅏ	공포		
13	제가 스타일을 바꾸면 더 어색하게만 변할것같아서 ㅠ0	공포		
14	아내 있으면 여자 있는 술집가면 법 적으로 문제있나요?	공포		

단발성 대화 데이터셋

id	document	label		
1	아 진짜! 사무실에서 피지 말라니깐! 간접흡연이 얼마나 안좋은데!	분노	놀람	3577
2	그럼 직접흡연하는 난 얼마나 안좋겠니? 안그래? 보면 꼭... 지 생각만 하고.	혐오	공포	91
3	손님 왔어요.	중립	분노	2919
4	손님? 누구?	중립	슬픔	1550
5	몰라요. 팀장님 친구래요.	중립	중립	29586
6	내 친구? 친구 누구?	중립	행복	657
7	그걸 내가 어떻게 알아요!	분노	혐오	214
8	그래서... 무슨 일 해?	중립		
9	그냥 방송일 조금.	중립		
10	방송? 방송 뭐?	중립		
11	케이블 같은데서 아침에 배도 타고 산도 오르고. 있어 그런 거.	중립		
12	사는 덴 어디야?	중립		
13	개포동.	중립		
14	남편은 뭐하는데?	중립		
15	동사무소에서 인구조사 나왔니? 지금.	분노		
16	언제부터 시작할 수 있어?	중립		
17	냐? ... 나보고 하라고?	분노		

연속성 대화 데이터셋

Part 2 SW (소프트웨어)

소프트웨어

데이터셋

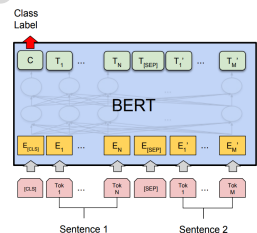
AI Hub

“한국어 감정정보가 포함된
단발성 대화 데이터셋”

“한국어 감정정보가 포함된
연속성 대화 데이터셋”

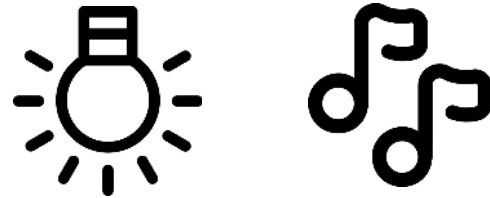
AI 모델 학습

AI



Kobert 모델 사용

“라이트 테라피 , 뮤직 테라피”



- 각각의 감정의 완화를 도와주는
빛의 색과 음악 출력
- 음악의 크기에 따른 빛의 세기 제
어(PWM 제어)
- 스위치로 조명과 음악을 off

감정
진단

감정에 따른 빛의 색 출력



네오픽셀 라이브러리
<Adafruit_NeoPixel.h> 설치

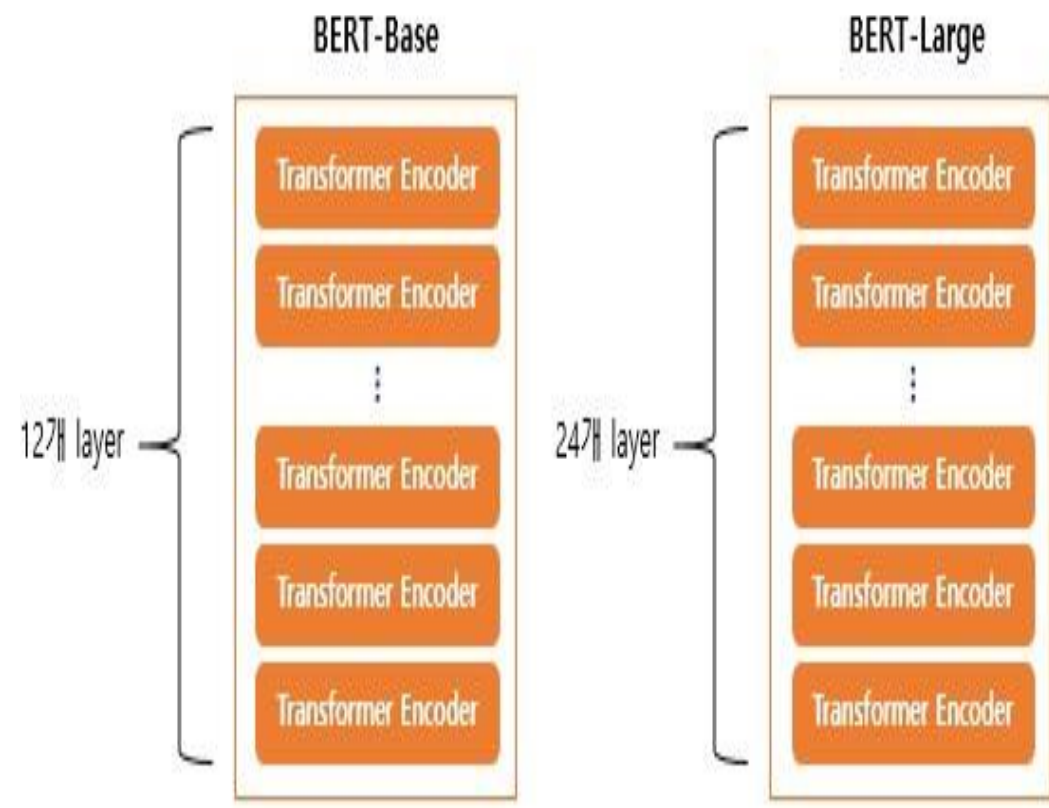
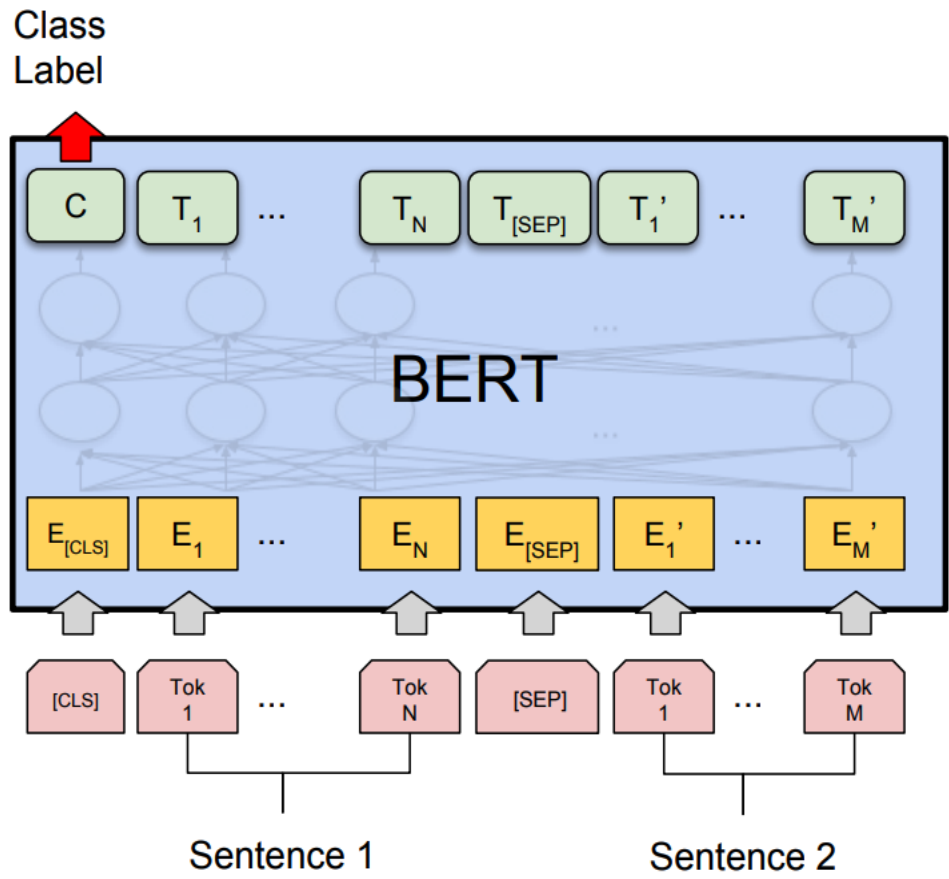
음성 신호처리 (FT)

가장 강하게 나타난 감정: 행복
감정 '행복'이 분석되었습니다.
Time: 0.00 seconds, PWM: 9
Time: 0.50 seconds, PWM: 10
Time: 1.00 seconds, PWM: 11
Time: 1.50 seconds, PWM: 11
Time: 2.00 seconds, PWM: 11
Time: 2.50 seconds, PWM: 11
Time: 3.00 seconds, PWM: 11
Time: 3.50 seconds, PWM: 10
Time: 4.00 seconds, PWM: 11

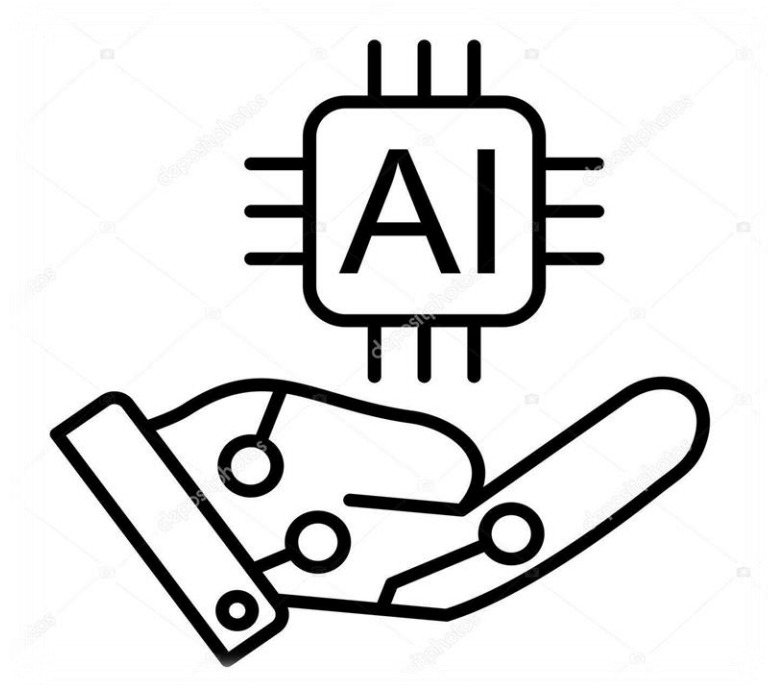
- 1초마다 주파수의 진폭 값의 평균 구함
- 이 평균을 4 구간으로 슬라이싱

PWM
제어

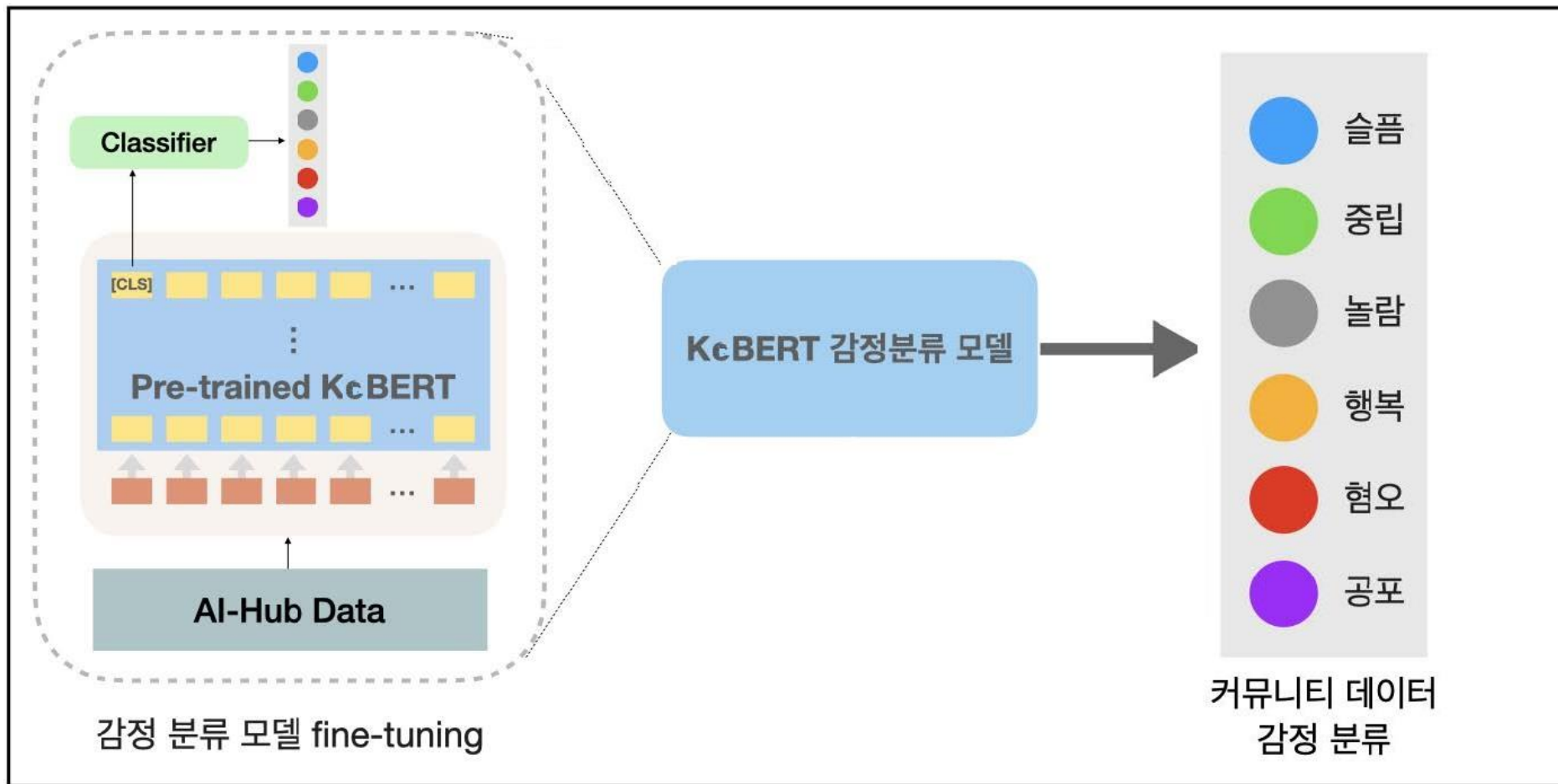
Kcbert 모델



- BERT는 2018년 구글이 공개한 사전 훈련된 모델

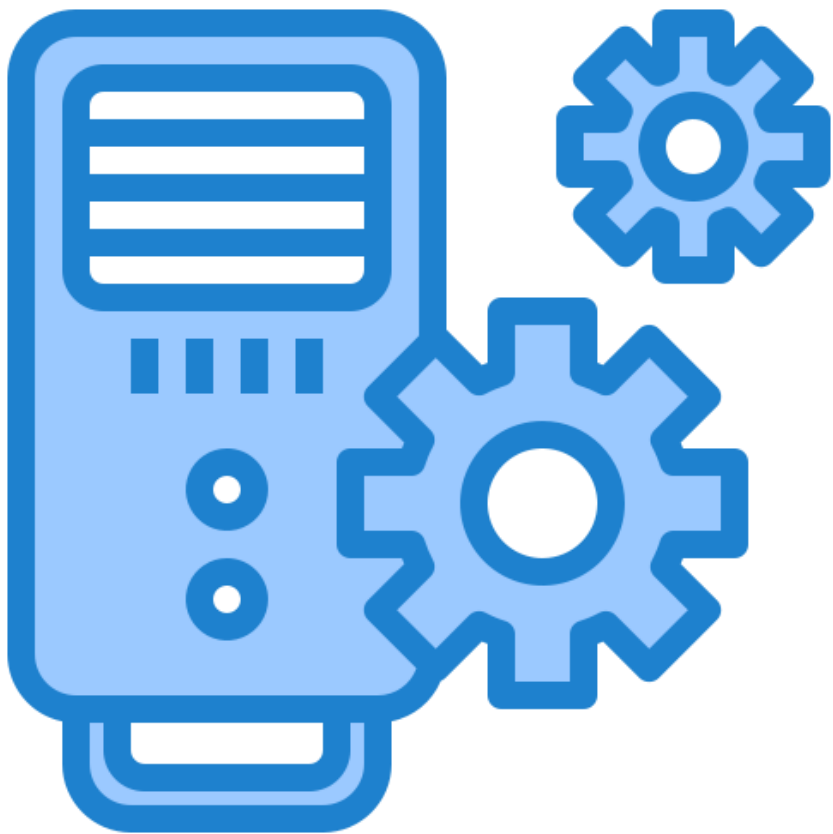


- 한국어에 특화된 BERT 변형 모델
- 한국어의 문법, 어휘, 문화적 특성을 반영하여 훈련
- 문맥 이해와 언어 의미 파악에 뛰어나며 문장 분류, 개체명 인식, 질문 응답, 문장 생성 등의 다양한 한국어 자연어 처리 작업에서 높은 성능을 발휘



Part 2

AI 모델을 구축을 위한 Python 환경 설정



Github 파일을 Python으로 가져오기



해당 라이브러리 import



필요한 라이브러리 설치

Step 1

```
class FormalClassifier:
    def __init__(self):
        self.model_name = "beomi/kcbert-base"
        self.tokenizer = AutoTokenizer.from_pretrained(self.model_name)
        self.model = BertForSequenceClassification.from_pretrained(self.model_name, num_labels=7).to(device)
        self.batch_size = 32
        self.max_len = 64
        self.dataLoader()

    def tokenize_function(self, examples):
        return self.tokenizer(examples["sentence"],
                               padding="max_length", truncation=True, max_length=self.max_len)
```

- 이 클래스는 BERT 모델을 초기화하고 데이터 로딩 함수를 호출하고 num_labels=7은 감정 분류 클래스의 수를 나타냄
- 함수는 입력 문장을 토큰화하고, 최대 길이로 패딩하고, 필요한 경우 잘라냄

Step 2

```
def dataLoader(self):
    train = pd.read_excel(r'C:\Users\cscs0\test.xlsx')
    dev = pd.read_excel(r'C:\Users\cscs0\train.xlsx')

    train = train.dropna()
    dev = dev.dropna()

    # 감정 레이블을 숫자로 변환
    emotions = ['공포', '놀람', '분노', '슬픔', '중립', '행복', '혐오']
    emotion_dict = {emotion: i for i, emotion in enumerate(emotions)}
    train['label'] = train['label'].map(emotion_dict)
    dev['label'] = dev['label'].map(emotion_dict)

    dataset = DatasetDict({
        'train': Dataset.from_dict({'sentence': train['document'].tolist(), 'label': train['label'].tolist()}),
        'dev': Dataset.from_dict({'sentence': dev['document'].tolist(), 'label': dev['label'].tolist()}),
    })

    tokenized_datasets = dataset.map(self.tokenize_function, batched=True)

    self.train_dataset = tokenized_datasets["train"]
    self.dev_dataset = tokenized_datasets["dev"]
```

- 엑셀 파일로부터 데이터를 로드하고, 결측치를 제거하며, 감정 레이블을 숫자로 변환함
- 데이터 셋을 DatasetDict 형식으로 생성합니다. 이후 각 문장을 토큰화하여 학습 및 검증 데이터를 준비함

Step 3

```
def train(self):
    training_args = TrainingArguments(
        output_dir="./saved_model",
        per_device_train_batch_size=self.batch_size,
        num_train_epochs=2,
        learning_rate=3e-05,
        save_strategy="epoch",
        evaluation_strategy="epoch",
        fp16=False,
    )

    trainer = Trainer(
        model=self.model,
        args=training_args,
        train_dataset=self.train_dataset,
        eval_dataset=self.dev_dataset,
        compute_metrics=self.compute_metrics,
    )

    trainer.train()
    trainer.evaluate()
```

- Trainer 객체를 사용하여 모델을 학습시키고 평가하고 학습 및 평가 과정의 파라미터를 매개 변수로 설정함

```
# 추론 함수 정의
def inference(input_doc):
    inputs = tokenizer(input_doc, return_tensors='pt')
    outputs = model(**inputs)
    probs = torch.softmax(outputs.logits, dim=1).squeeze().tolist()
    class_idx = ['공포', '놀람', '분노', '슬픔', '중립', '행복', '혐오']
    results = {class_name: prob for class_name, prob in zip(class_idx, probs)}
    # 가장 높은 확률의 클래스를 찾음
    max_prob_class = max(results, key=results.get)
    max_prob = results[max_prob_class]
    # 결과 표시
    return [results, max_prob_class]

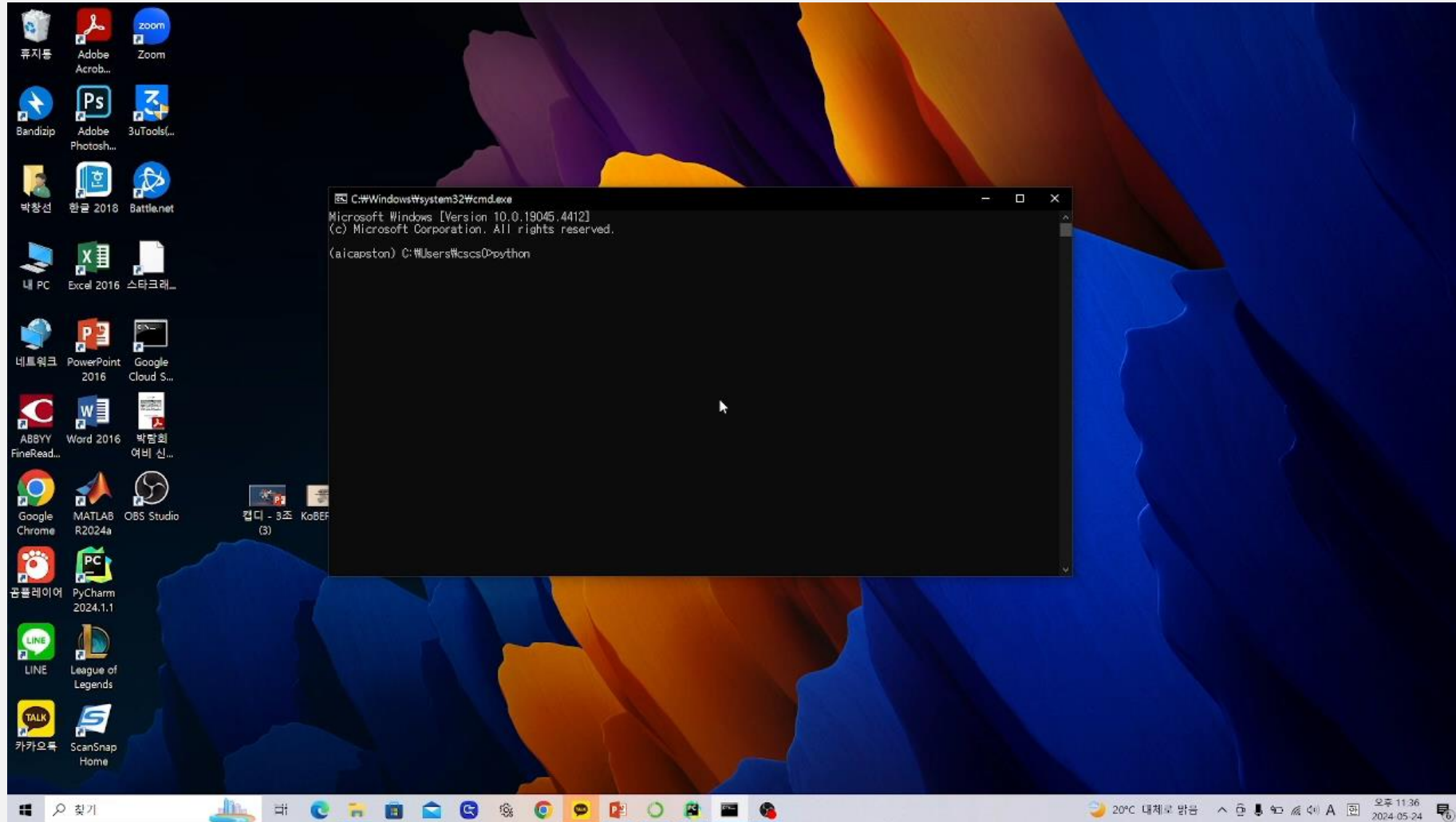
# Streamlit 인터페이스 설정
st.title('감정분석(Sentiment Analysis)')
st.markdown('<small style="color:grey;">글에 나타난 공포, 놀람, 분노, 슬픔, 중립, 행복, 혐오의 정도를 비율로 알려드립니다.</small>',
            unsafe_allow_html=True)
user_input = st.text_area("이 곳에 글 입력(100자 이하 권장):")
```

```
{ 'train_acc': 0.75,  
  'train_f1': 0.75,  
  'train_loss': 0.5262309908866882,  
  'train_precision': 0.8571428571428571,  
  'train_recall': 0.6666666666666666}  
  
{ 'train_acc': 0.8125,  
  'train_f1': 0.7857142857142856,  
  'train_loss': 0.3342723548412323,  
  'train_precision': 0.7333333333333333,  
  'train_recall': 0.8461538461538461}  
  
{ 'train_acc': 0.875,  
  'train_f1': 0.8823529411764706,  
  'train_loss': 0.2689765691757202,  
  'train_precision': 0.8823529411764706,  
  'train_recall': 0.8823529411764706}  
  
{ 'train_acc': 0.875,  
  'train_f1': 0.8461538461538461,  
  'train_loss': 0.3174639940261841,  
  'train_precision': 0.7857142857142857,  
  'train_recall': 0.9166666666666666}  
  
{ 'train_acc': 0.75,  
  'train_f1': 0.5555555555555556,  
  'train_loss': 0.38240528106689453,  
  'train_precision': 0.8333333333333334,  
  'train_recall': 0.4166666666666667}
```

- 5번의 학습을 한 결과 모델의 예측 정확도가 평균 85%가 나오는 것으로 확인

Part 2

모델 테스트



<div> <div>감정(색상)</div> <div>진폭값</div> </div>	분노, 혐오 (보라색)	슬픔 (파란색)	행복 (노랑색)	공포, 놀람 (초록색)
60	22 (0,,30,60)	21 (0,0,60)	20 (60,60,0)	19 (60,0,0)
120	2 (0,60,120)	6 (0,0,120)	10 (120,120,0)	14 (120,0,0)
180	3 (0,90,180)	7 (0,0,180)	11(180,180,0)	15 (180,0,0)
250	4 (0,127,255)	8 (0,0,155)	12 (255,255,0)	16 (255,0,0)

```
def calculate_average_amplitude_in_frequency_range(waveform, sample_rate, freq_start, freq_end):  
    spectrum = fft(waveform)  
    freqs = np.fft.fftfreq(len(waveform)) * sample_rate  
    idx_start = np.where(freqs >= freq_start)[0][0]  
    idx_end = np.where(freqs <= freq_end)[0][-1]  
    avg_amplitude = np.mean(np.abs(spectrum[idx_start:idx_end]) ** 2)  
    if avg_amplitude <= set1:  
        pwm_value = 1  
    elif set1 + 1 <= avg_amplitude <= set2:  
        pwm_value = 2  
    elif set2 + 1 <= avg_amplitude <= set3:  
        pwm_value = 3  
    else:  
        pwm_value = 4  
    return pwm_value
```

fft 함수를 사용하여 각 푸리에 변환 성분에 해당하는 주파수를 계산



Fourier 변환 수행



주파수 범위 계산



시작 및 종료 인덱스 찾기



주파수 범위 내의 평균 진폭 계산



평균 진폭에 따른 PWM 값 결정

Part 2

음악 신호 처리 (PWM 제어)

```
if result[1] == '행복':  
    emotion = 'happy'  
    set1 = 10000000  
    set2 = 30000000  
    set3 = 90000000  
    ti_set = 0.5  
elif result[1] == '슬픔':  
    emotion = 'sad'  
    set1 = 30000000  
    set2 = 70000000  
    set3 = 100000000  
elif result[1] == '공포' or result[1] == '놀람':  
    emotion = 'wonder'  
    set1 = 10000000  
    set2 = 30000000  
    set3 = 100000000  
elif result[1] == '분노' or result[1] == '혐오':  
    emotion = 'angry'  
    set1 = 10000000  
    set2 = 30000000  
    set3 = 80000000
```

4개의 진폭값의 범위 생성



진폭값의 평균에 대한 주기 설정



음원 데이터에 대해 푸리에 변환을 수행



사용할 진폭 범위를 설정하여 반환



주파수에 대한 진폭값 값 계산

가장 강하게 나타난 감정: 행복
감정 '행복'이 분석되었습니다.

Time: 0.00 seconds, PWM: 9
Time: 0.50 seconds, PWM: 10
Time: 1.00 seconds, PWM: 11
Time: 1.50 seconds, PWM: 11
Time: 2.00 seconds, PWM: 11
Time: 2.50 seconds, PWM: 11
Time: 3.00 seconds, PWM: 11
Time: 3.50 seconds, PWM: 10
Time: 4.00 seconds, PWM: 11

Time: 4.50 seconds, PWM: 11
Time: 5.00 seconds, PWM: 10
Time: 5.50 seconds, PWM: 11
Time: 6.00 seconds, PWM: 10
Time: 6.50 seconds, PWM: 11
Time: 7.00 seconds, PWM: 11
Time: 7.50 seconds, PWM: 11
Time: 8.00 seconds, PWM: 11
Time: 8.50 seconds, PWM: 11
Time: 9.00 seconds, PWM: 11
Time: 9.50 seconds, PWM: 12

- 1초마다 주파수의 진폭값을 평균을 구하고 구간별로 설정한 값 출력

Part 2

스위치 제어 (LED, 음악 off 기능)

```
def monitor_switch(stop_event, arduino_port='COM5'):
    try:
        ser = serial.Serial(arduino_port, baudrate=9600)
        time.sleep(2) # Wait for the serial connection to initialize
        while not stop_event.is_set():
            if ser.in_waiting > 0:
                message = ser.readline().decode().strip()
                if message == "STOP":
                    stop_event.set()
                    break
            ser.close()
    except serial.SerialException as e:
        print(f"Error opening serial port: {e}")

# 메인 함수에서 스위치 모니터링 스레드 시작
def main():
    stop_event = Event()
    switch_thread = Thread(target=monitor_switch, args=(stop_event,))
    switch_thread.start()
```

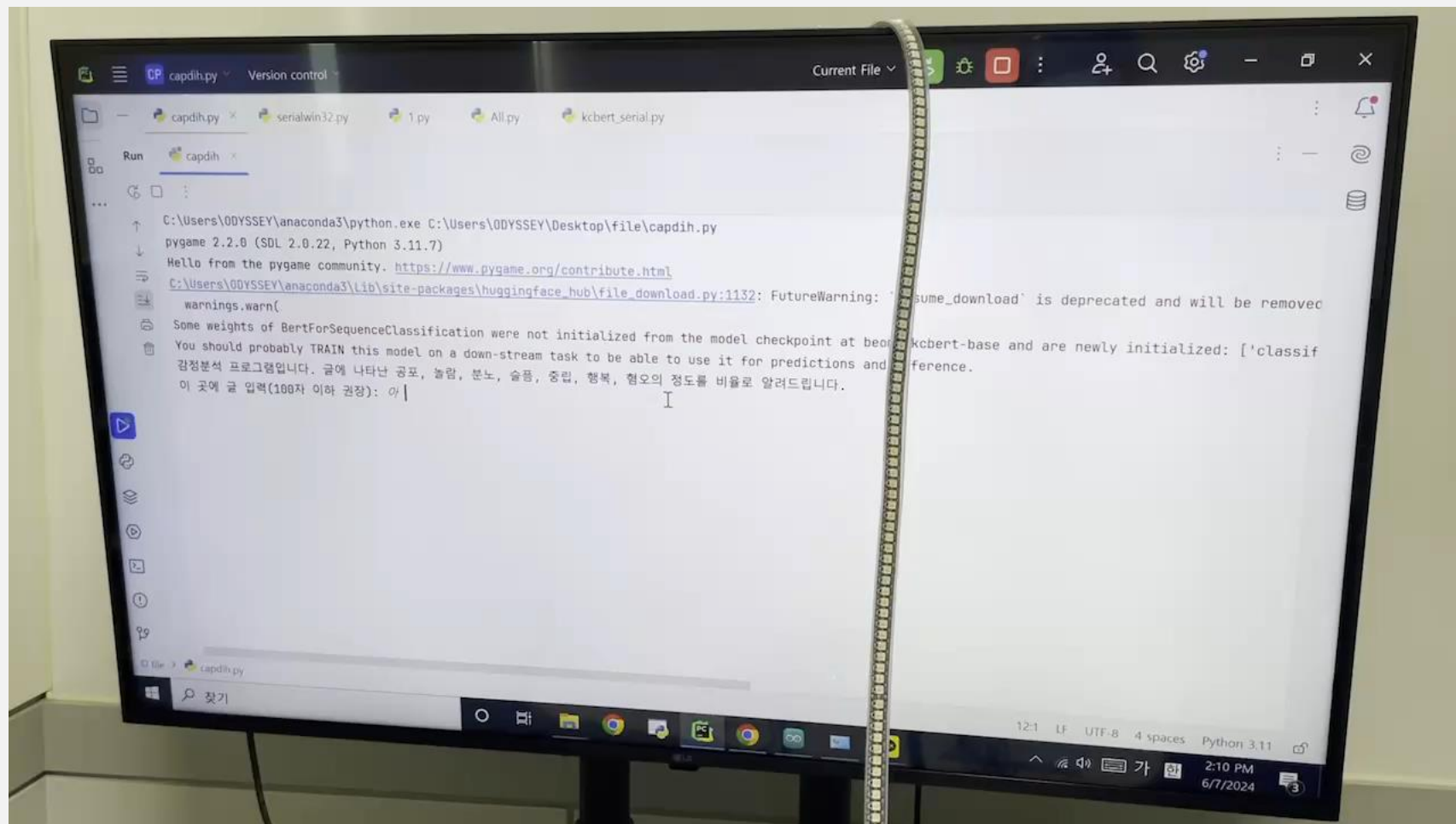
‘monitor_switch’ 함수에서 시리얼 포트를 사용하여
스위치의 상태를 모니터링 함



스위치 “STOP” 메시지를 보내면 프로그램이 중지됨

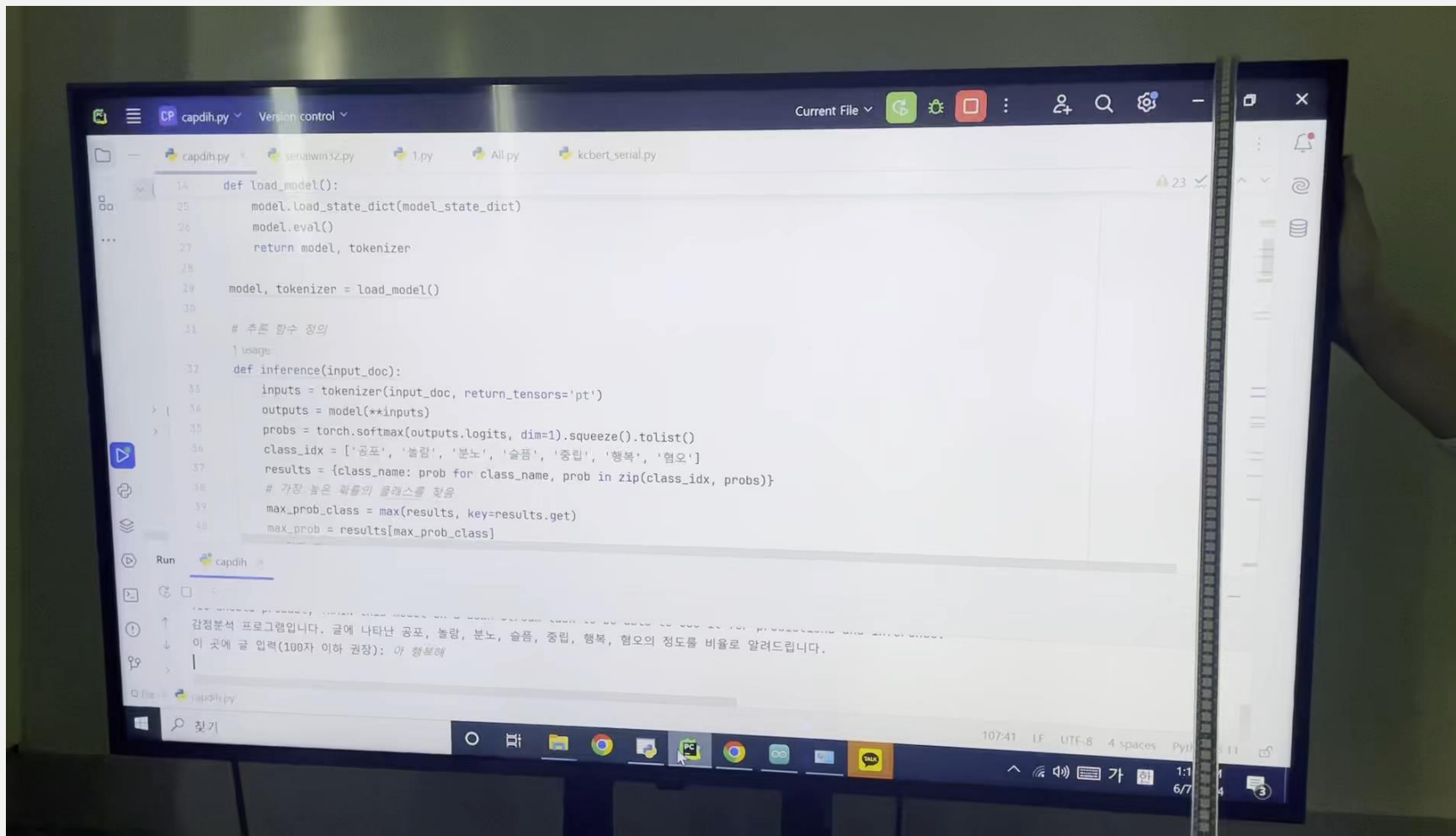
Part 2

스위치 제어 출력



Part 2

감정에 따른 LED, 음악 출력

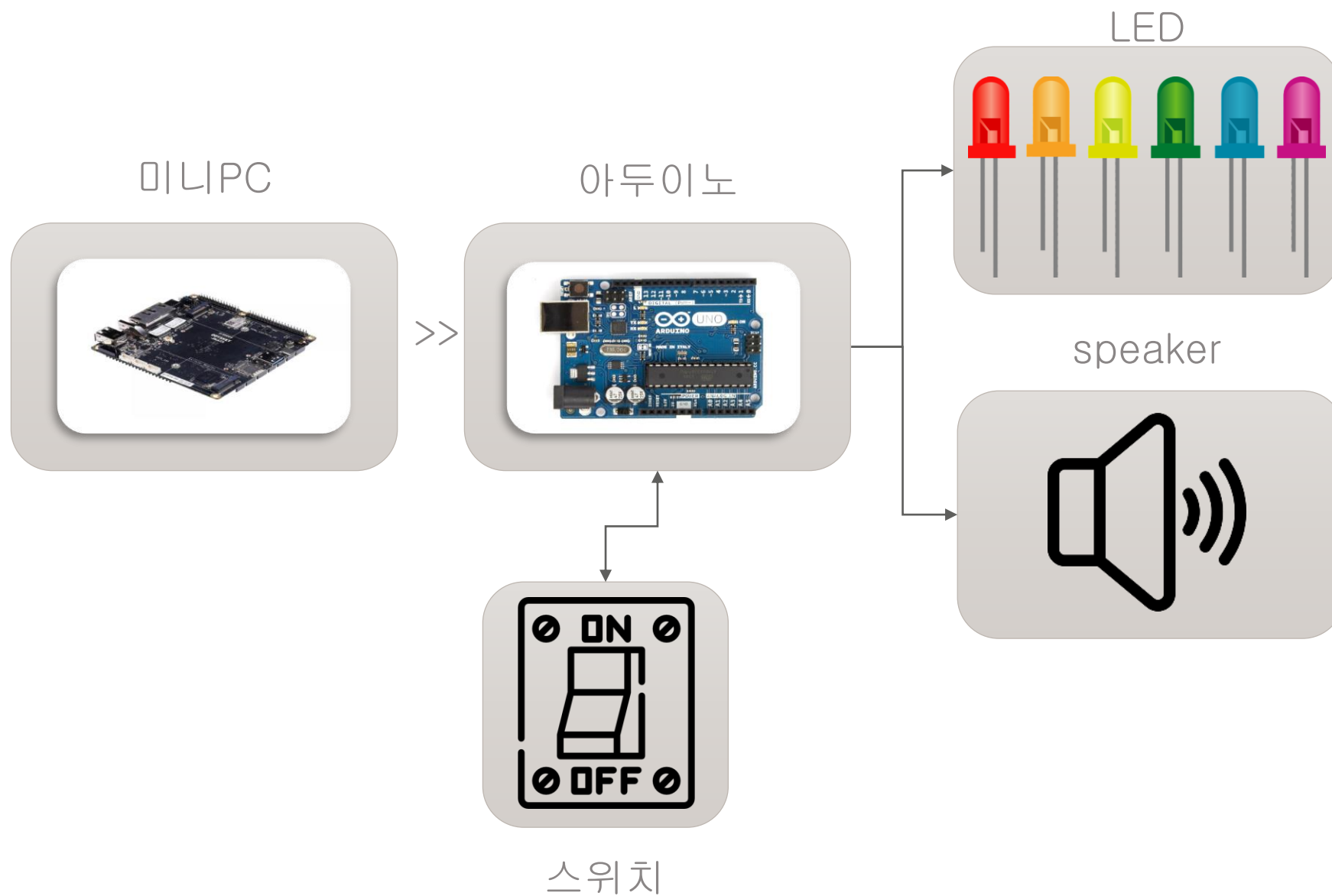


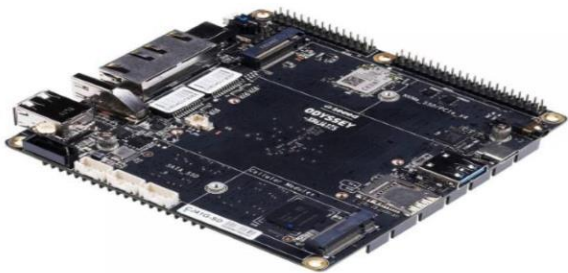


Part 3 HW (하드웨어)

Part 3

HW(하드웨어)

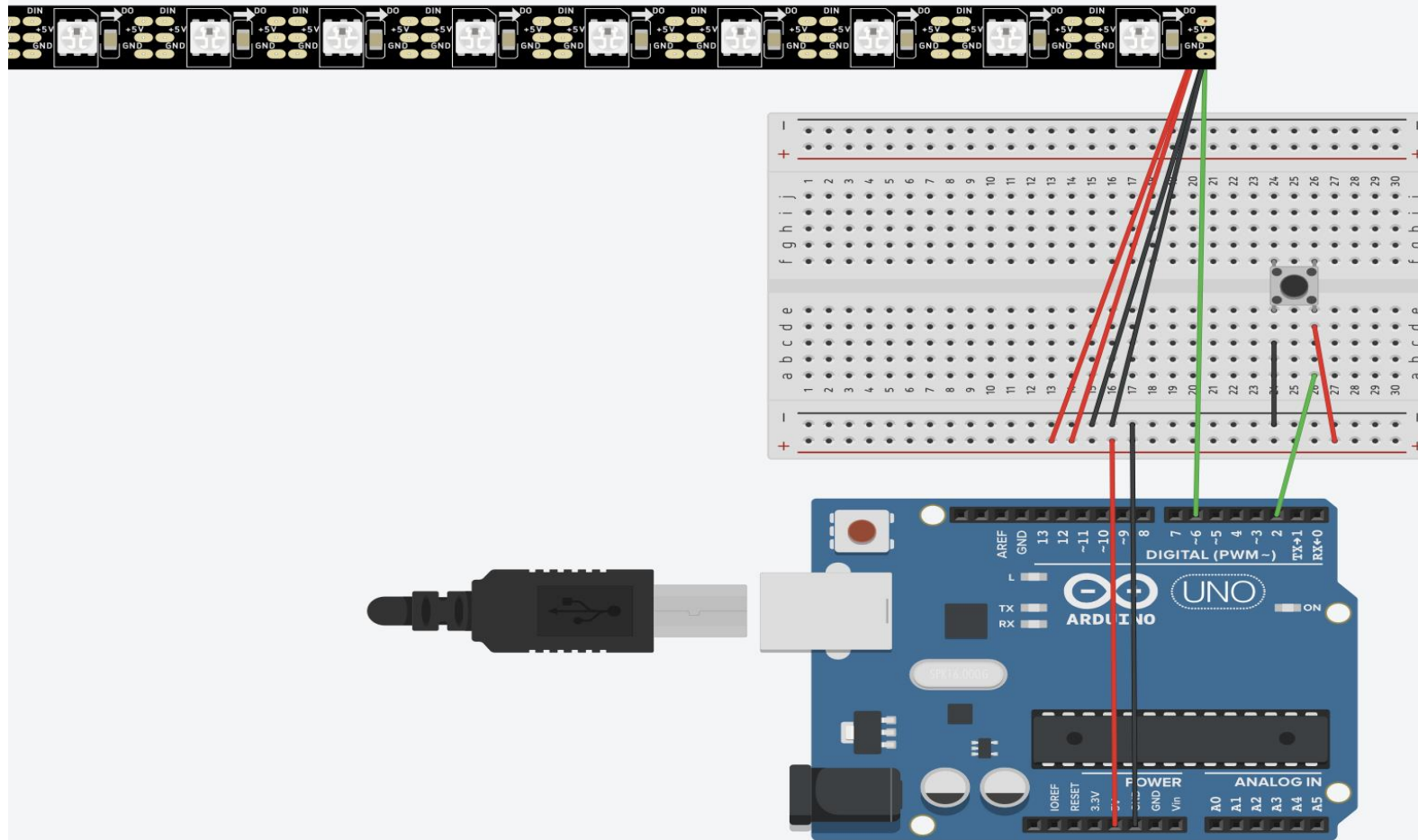




- 아나콘다 설치
- 가상 환경 생성
- 필요한 라이브러리 설치
- 자격증명(json파일) 생성
- Gcloud 설치(서비스 계정 키 등록 및 활성화)

Part 3

회로도 구성





Part 4 프로젝트 이슈

렌즈타입 LED RGB



- 전용 컨트롤러로 제어 가능
- 아두이노로 제어 안됨

네오픽셀 스트립 LED RGB

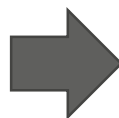


- 아두이노로 제어 가능
- pwm 제어(빛의 밝기 조절) 가능

```
import torch
from torch import nn
import torch.nn.functional as F
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader
import gluonnp as nlp
import numpy as np
from tqdm import tqdm, tqdm_notebook
import pandas as pd

/usr/local/lib/python3.10/dist-packages/mxnet/numpy/utils.py:37: FutureWarning:
  bool = onp.bool

-----
AttributeError                                Traceback (most recent call last)
<ipython-input-2-bdc4d9afcega> in <cell line: 6>()
      4 import torch.optim as optim
      5 from torch.utils.data import Dataset, DataLoader
----> 6 import gluonnp as nlp
      7 import numpy as np
      8 from tqdm import tqdm, tqdm_notebook
```



```
✓ 6초 !pip install numpy==1.23.5
✓ 5초 ▶ import torch
      from torch import nn
      import torch.nn.functional as F
      import torch.optim as optim
      from torch.utils.data import Dataset, DataLoader
      import gluonnp as nlp
      import numpy as np
      from tqdm import tqdm, tqdm_notebook
      import pandas as pd
```

2023년 8월 colab업데이트 이후
gluonnp가 오류 나는 것을 확인

Colab에서 Numpy가 업데이트 됐다는 사실 확인 후
1.23.5버전 설치 후 해결 완료

```
serial_port = 'COM3' # 윈도우
arduino = serial.Serial(serial_port, 9600, timeout=1)
def predict(predict_sentence):
    data = [predict_sentence, '0']
    dataset_another = [data]
    another_test = BERTDataset(dataset_another, 0, 1, tok, vocab, max_len, True, False)
    test_dataloader = torch.utils.data.DataLoader(another_test, batch_size=batch_size, num_workers=5)
    model.eval()
    for batch_id, (token_ids, valid_length, segment_ids, label) in enumerate(test_dataloader):
        token_ids = token_ids.long().to(device)
        segment_ids = segment_ids.long().to(device)
        valid_length = valid_length
        label = label.long().to(device)
        out = model(token_ids, valid_length, segment_ids)
        test_eval = []
        for i in out:
            logits = i
            logits = logits.detach().cpu().numpy()
            if np.argmax(logits) == 0:
                test_eval.append("분노가")
            elif np.argmax(logits) == 1:
                test_eval.append("슬픔이")
            elif np.argmax(logits) == 2:
                test_eval.append("중립이")
            elif np.argmax(logits) == 3:
                test_eval.append("긍정이")
        emotion = test_eval[0]
        # 아두이노로 감정 전송
        arduino.write(emotion.encode())

    print(">> 입력하신 내용에서 " + emotion + " 느껴집니다.")
```

Colab에서 시리얼 통신을 통해 아두이노와 연결하려고 했으나 클라우드 기반 Colab에서는 실행 불가능

```
(base) C:\Users\scscs0>pip install gluonnlp
Collecting gluonnlp
  error: subprocess-exited-with-error
  × python setup.py bdist_wheel did not run successfully.
    exit code: 1
  ↳ [143 lines of output]
    C:\Users\scscs0\anaconda3\lib\site-packages\setuptools\__init__.py:80: _DeprecatedInstaller: setuptools.installer a
    nd fetch_build_egg are deprecated.
      !!

    *****
    Requirements should be satisfied by a PEP 517 installer.
    If you are using pip, you can try `pip install --use-pep517`.
    *****
```

Gluonnlp가 윈도우 환경에서 실행되지않고 colab에서만 실행되는 이슈 발견

BERT 모델은 각 토큰의 인덱스와 타입, 유효 길이를 필요로 함

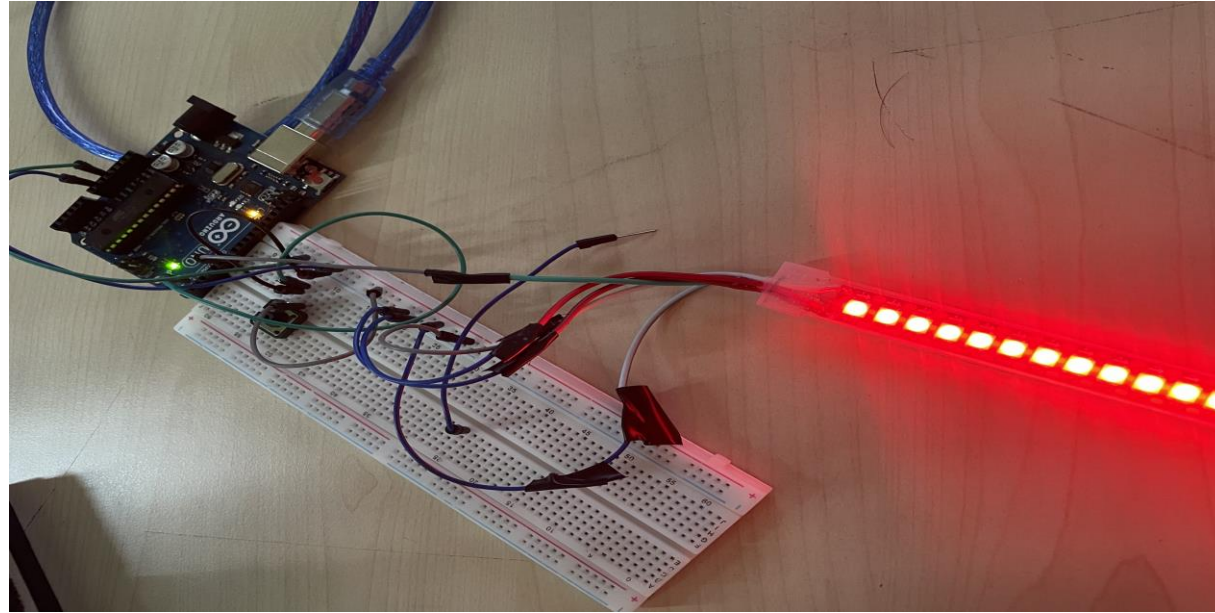
Gluonnlp 라이브러리는 이를 매우 편하게해주는 함수를 제공하므로 반드시 필요

해결한 방법

- Kcbert라는 AI 모델을 찾아 로컬환경에서 실행가능하도록 변경함

Part 4

프로젝트 이슈 4

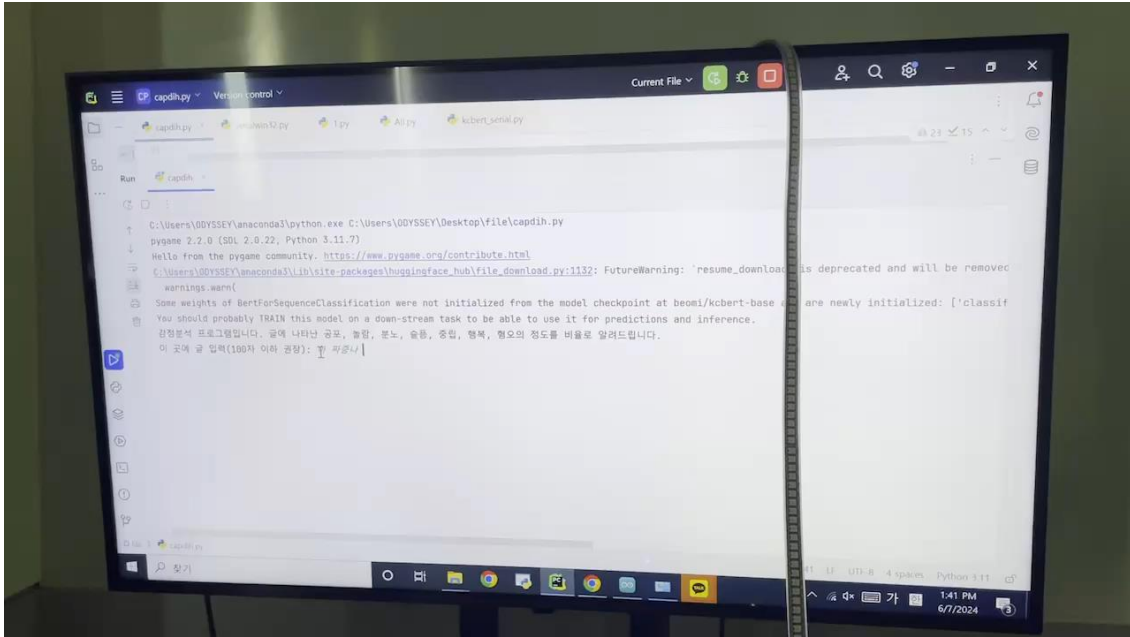


LED 불량으로 인해 빨간색만 나오는 걸로 확인



해결 방안

LED 재구매



분노	슬픔	행복	놀람
PWM 3,4 오류	오류X	PWM8 오류	PWM 13~16 오류

RGB 값을 모두 사용하였을 때 원하는 색상이 나오지 않는 것으로 확인

해결 방안

RGB 값을 1개 또는 2개로 사용하여 오류를 해결함

```
import numpy as np
from scipy.io import wavfile
from scipy.fft import fft
import serial
import time
from playsound import playsound
from threading import Thread
import torch
```

Traceback (most recent call last):

File "C:\Users\ODYSSEY\Desktop\file\capdih.py", line 181, in <module>
 main()

File "C:\Users\ODYSSEY\Desktop\file\capdih.py", line 174, in main
 send_to_arduino(result_list)

File "C:\Users\ODYSSEY\Desktop\file\capdih.py", line 139, in send_to_arduino
 ser.write(f"{pwm}\n".encode())

File "C:\Users\ODYSSEY\anaconda3\Lib\site-packages\serial\serialwin32.py", line 311, in write
 raise SerialException("WriteFile failed ({!r}).format(ctypes.WinError()))

serial.serialutil.SerialException: WriteFile failed (PermissionError(13, '장치가 명령을 인식하지 않습니다.', None, 22))

스위치 기능 오류

빛은 off 가능하지만

Playsound 모듈로는 스위치로 음악 제어가 되지 않음


```
import serial
import time
import numpy as np
from scipy.io import wavfile
from scipy.fft import fft
import torch
from transformers import BertConfig, BertForSequenceClassification, BertTokenizer
import pygame
from threading import Thread, Event

stop_event = Event()
```

해결 방안

Pygame 모듈을 설치하여 해결함

A person wearing a white lab coat is holding a white smartphone in their right hand. Their left hand is positioned near the top of the phone, possibly interacting with the screen. In the background, a white cup of coffee is visible. The scene is brightly lit, suggesting an indoor setting like a laboratory or office.

Part 6 기대 효과 및 활용 방안

감정 관리 및 스트레스 완화

- 감정 변화에 대응하고 스트레스 줄임
- 긍정적인 감정을 촉진함
- 사용자의 정신적 안정을 증진시킴

자가 치유

- 사용자들이 자가 치유를 할 수 있는 도구로 활용
- 사용자가 스스로 감정 상태 식별
- 감정 진단 AI와 라이트테라피 병합하여 사용할 수 있음



치료 및 상담 보조 도구로 활용

- 전문가가 환자를 진단하고 치료하는 데 도움이 되는 도구
- 감정 진단 AI - 전문가에게 빠르고 정확한 정보를 제공
- 라이트테라피 - 환자의 감정은 안정시키는데 도움을 줌

심리적 측면에서의 질병 예방

- 심리적 질병이나 장애의 조기 진단 및 예방에 도움을 줄 수 있음
- 심리적 질병의 발병을 예방할 수 있음



감사합니다