

DU-도전학기 결과보고서

성 명	[국문] [영문]	학 번	
단과대학	정보통신대학	학과(전공)	전자제어공학 (복수전공:AI응용전공으로학점신청)
도전학기 과제명	[국문] Open API를 이용한 실시간 음성 영어 번역기 [영문] Real-time English Translator Using Open API		
지도교수 의견	<p>상기 과제를 충실히 수행하였습니다.</p> <p>(성명)</p> <p>(소속) 전자제어공학과 (서명)</p>		

1. 도전 과제의 목표

작품제작에 있어서는 NAVER OPEN API를 통해 저의 목소리를 텍스트로 출력하고 출력된 텍스트를 영어로 번역하는 음성인식 시스템 구축이 저의 1차 목표이며, Google OPEN API의 TTS를 이용하여 번역된 텍스트를 소리로 출력하는 것이 2차 목표입니다. 최종적으로 OPEN API를 이용하여 실시간 음성 인식 및 번역을 하여 실시간 음성 영어 번역기를 구축하고 지정한 캐릭터가 영어와 한글을 읽어주는 학습용 유아장난감을 제작하는 것이 최종 목표입니다.

완성된 작품을 통해 소속된 동아리 회원들에게 세미나를 통해 아이템을 설명하고, 구현 영상을 제작하여 youtube에 업로드 할 것입니다. 또한 작품 관련 기술을 이용하여 특허를 출원하는 것이 최종적 목표입니다.

2. 도전 과제 내용



- 라즈베리파이를 기반으로 음성인식을 활용한 실시간 영어 번역기
- Open API 기술을 기반으로 마이크를 통해 입력한 언어를 음성인식
- 인식한 음성을 문자로 변환 및 번역하여 디스플레이에 출력
- 시·청각을 통해 유아들이 쉽게 영어를 습득하게 하는 기술 개발
- 지정한 캐릭터가 한글과 영어를 읽어주는 학습용 유아 장난감 제작
- 소속 동아리 회원들에게 작품 세미나 발표
- 교내외공모전 참가 및 입상

3. 도전 과제의 성과

□ 도전 과제의 진행 완료 상황

주차	활동 목표	활동 내용	진행 상황
1주차	작품 설계	실시간 음성 번역기 관련 OPEN API 학습 및 관련 소프트웨어 설계	100 %
2주차	작품 설계	라즈베리파이 관련 센서 및 하드웨어 설계	100 %
3주차	작품 설계	최종적인 실시간 음성번역기 관련 OPEN API 소스코드 설계 및 하드웨어 설계	100 %
4주차	작품 제작	라즈베리파이를 이용하여 NAVER OPEN API인 STT 소스코드 구현	100 %
5주차	작품 제작	라즈베리파이를 이용하여 NAVER OPEN API인 NMT 소스코드 구현	100 %
6주차	작품 제작	라즈베리파이를 이용하여 Google OPEN API인 TTS 소스코드 구현	100 %
7주차	작품 제작	라즈베리파이 LCD를 이용한 작품 관련 이미지 및 한글 파일 출력 구현	100 %
8주차	작품 제작	라즈베리파이 스피커를 이용한 실시간 TTS 음성 출력	100 %
9주차	작품 제작	라즈베리파이 마이크를 이용한 실시간 STT 한글 파일 출력	100 %
10주차	작품 제작	라즈베리파이 마이크와 스피커를 연결하여 STT와 TTS 동시 출력 구현	100 %
11주차	작품 제작	라즈베리파이를 통한 STT한 영어를 NMT를 이용하여 한글 번역	100 %
12주차	작품 완성	최종적인 OPEN API 소스코드와 라즈베리파이 마이크, 스피커와 같은 하드웨어 구축	100 %
13주차	작품 완성	최종적으로 영어를 한글로 번역해주고 읽어주는 실시간 음성번역기 구축	100 %
14주차	작품 테스트	소스코드 및 하드웨어 오류 분석	100 %
15주차	작품 테스트	소스코드 및 하드웨어 오류 분석	100 %
16주차	결과보고서 및 영상 제작	완성된 결과물을 통한 결과보고서 작성 및 구현 영상 제작	100 %

□ 특허 출원 고객번호 발급

특허고객번호 통지서

접수 번호 : 4-1-2019-5059423-16
 접수담당자 : nu
 세팅제출자 :
 등록일자 : 2019.03.26
 특허고객번호 : 4-2019-019331-6

기초정보	생년월일		
	성명		
	주민등록번호	<input type="checkbox"/> 출생인구분	<input type="checkbox"/> 국내자연인
	사업자등록번호	없음	
주소정보	전화번호	<input type="text" value="핸드폰 번호"/>	
	Email주소		
	우편번호	<input type="text" value="시도국명"/>	
	소재지(한글)		
출발	소재지(영문)		
	출발주소유관번호	없음	
	소재지(한글)	없음	
	소재지(한글)	없음	

한

갑

NO IMAGE

서

면

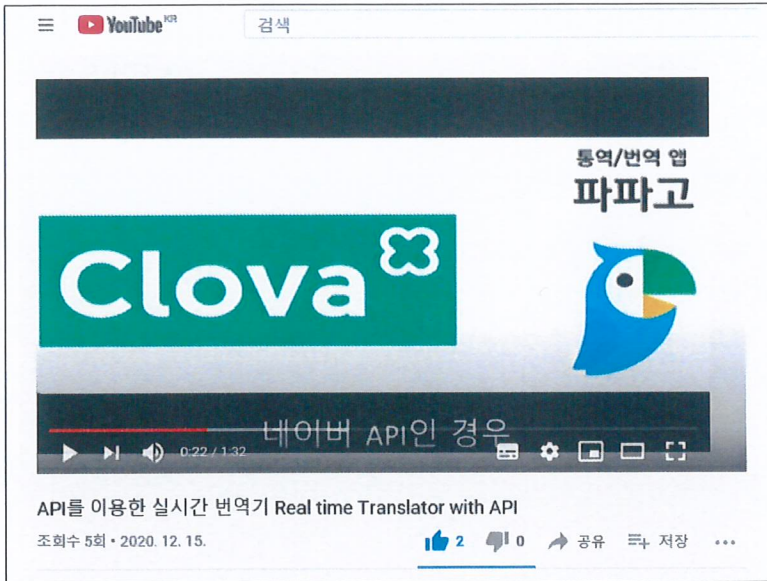
특허법 제29조의 2 및 통신안법 제3조, 디지털정보법 제29, 30조 제2항 규정에 따라 위와 같이 출생인구분을 통지합니다.

특 허 청 장

관 인 생 략

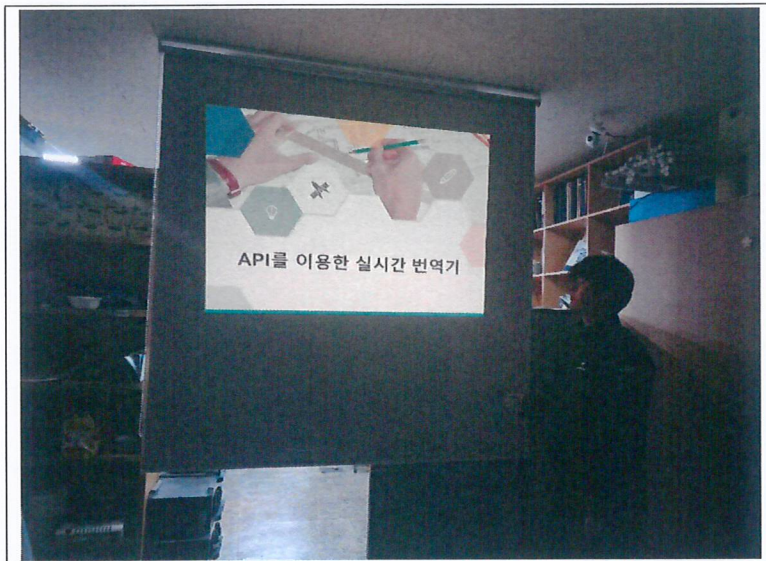
- 번역기 알고리즘을 이용한 특허 기술을 출원하기 위한 특허고객번호를 발급받았다.

□ 작품소개 유튜브 영상 업로드



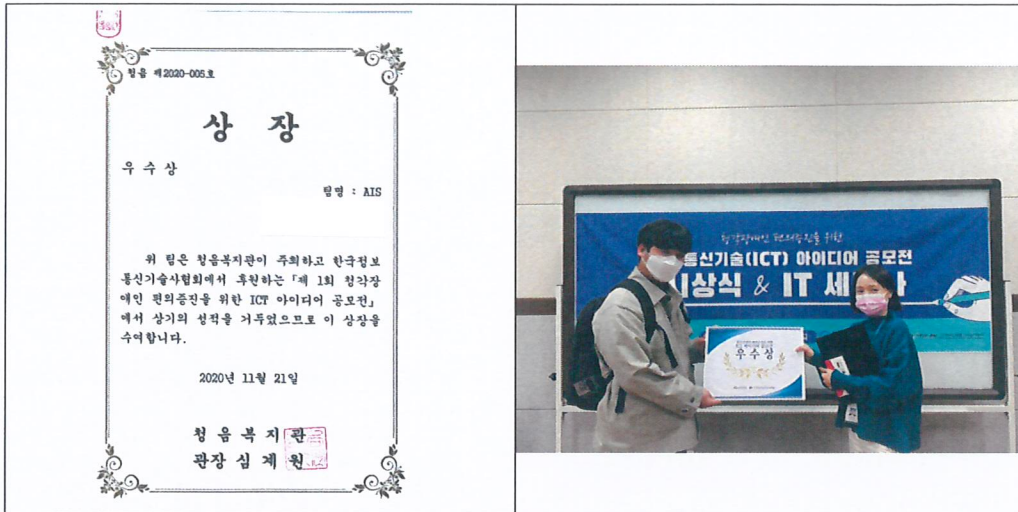
- 작품소개 영상을 직접 제작하여 소속 동아리 유튜브 계정에 업로드 하였다.
- <https://www.youtube.com/watch?v=oGlgGDWiuxs>

□ 작품관련 세미나 발표



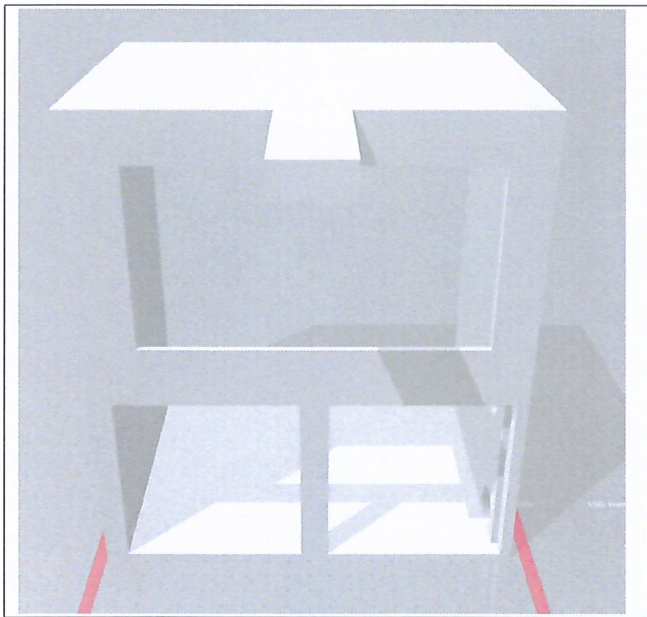
- 작품 관련 세미나발표를 학과동아리에서 하였다.

□ 교외 대회 입상(우수상)



- 제 1회 청각장애인 편의증진을 위한 ICT 아이디어 공모전에서 우수상을 입상하였다.
- 동아리 세미나를 통해 번역기 아이디어를 발전시켜 참가하였다.
- 기존의 실시간 번역기를 청각장애인들의 발음교정 번역기 아이디어를 발전시켜 참가하였다.

□ 3D 프린터를 이용하여 프레임 설계



- 3D inventor를 이용하여 프레임을 설계 하였다.

4. 자기 평가

○ 예상 투입 시간 변경 (5시간 → 6시간)

- 초기 계획으로 주차마다 2시간 30분씩 2회 진행 하여 5시간의 계획이 프로그래밍 언어 및 실무 능력이 부족하여 언어 정보 및 학습을 위한 시간 투입을 증가 시켰다. 최종적으로 주차마다 3시간 2회 진행하여 6시간 투입하여 프로젝트를 완료 하였다.

○ 코로나 바이러스로 인한 계획된 공모전 무산

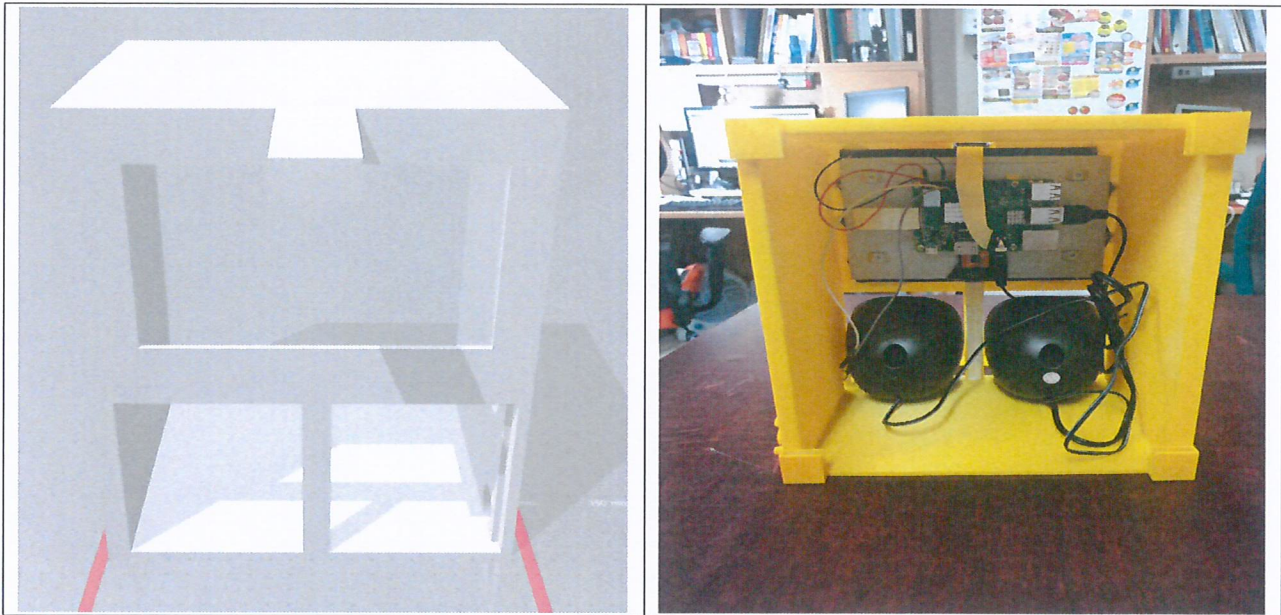
- 초기 계획하였던 창원 전국 로봇 지능 대회가 코로나로 인해 무기한 연기되어 참가하지 못하게 되었다.
- 작품을 통해 나갈 수 있는 교외 공모전을 찾아 참가하게 되었다.

○ 작품을 이용한 기술 및 지식 특허 출원 계획

- 평소 특허를 잘 몰라 단순히 특허 청구서만 작성하여 제출하면 되는 줄 알았지만, 특허 등록을 위해 다양한 정보를 수집하여 제출해야 된다는 것을 깨달았다. 음성 딥러닝 알고리즘을 이용하여 특허기술을 출원 할 것이다.

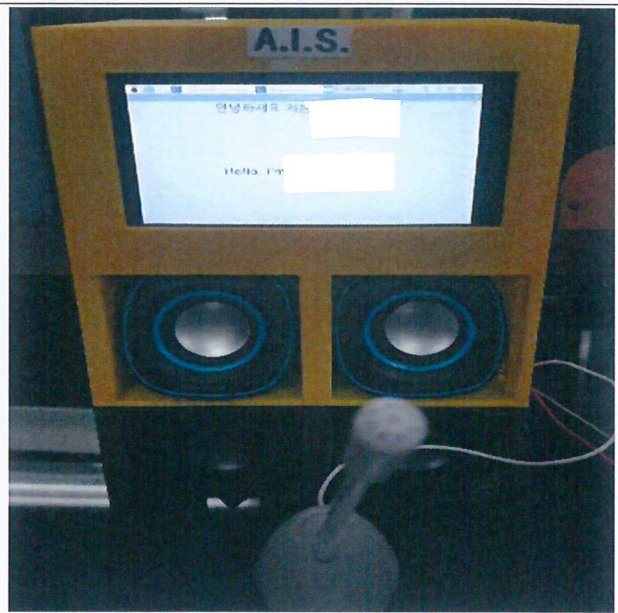
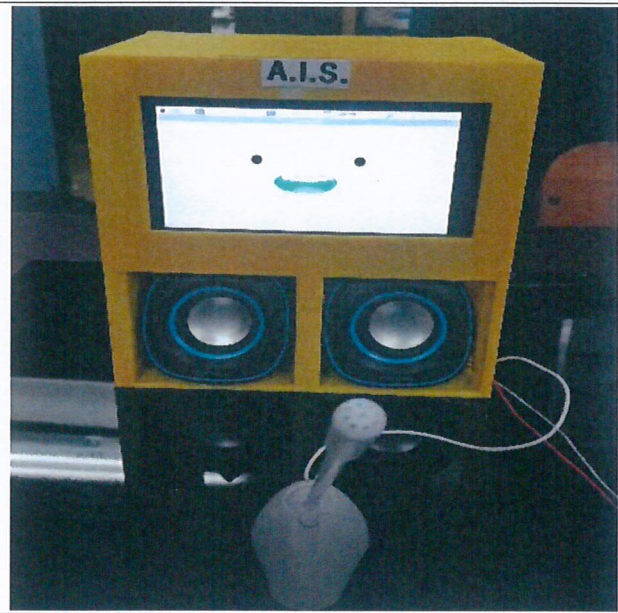
5 최종 결과물

□ 하드웨어





제작 사진



완성사진

□ 소스코드

```
import numpy as np
import tensorflow as tf
import argparse
import re
import sys

modelFullPath = '/home/rastech/Downloads/dataset/tmp/output_graph.pb'
labelsFullPath = '/home/rastech/Downloads/dataset/tmp/output_labels.txt'

# 읽어들이 graph 파일 경로
# 읽어들이 labels 파일 경로

from google.cloud import speech
from google.cloud.speech import enums
from google.cloud.speech import types
import pyaudio
from six.moves import queue

import naverTTS

# [END import_libraries]

#네이버 TTS 클래스

# speaker
#
# 0 : 'mijin', #한국어 여성
# 1 : 'jinho', #한국어 남성
# speed
# 0 = 일반 속도
#tts = naverTTS.NaverTTS(0, 0)
#or
tts = naverTTS.NaverTTS()

target = ""

def create_graph():
    """저장된(saved) GraphDef 파일로부터 graph를 생성하고 saver를 반환한다."""
    # 저장된(saved) graph_def.pb로부터 graph를 생성한다.
    with tf.gfile.GFile(modelFullPath, 'rb') as f:
        graph_def = tf.GraphDef()
        graph_def.ParseFromString(f.read())
        _ = tf.import_graph_def(graph_def, name="")

def run_inference_on_image(imagePath):
    answer = None
```

```

def run_inference_on_image(imagePath):
    answer = None

    if not tf.gfile.Exists(imagePath):
        tf.logging.fatal('File does not exist %s', imagePath)
        return answer

    image_data = tf.gfile.GFile(imagePath, 'rb').read()

    # 저장된(saved) GraphDef 파일로부터 graph를 생성한다.
    create_graph()

    with tf.Session() as sess:

        softmax_tensor = sess.graph.get_tensor_by_name('final_result:0')
        predictions = sess.run(softmax_tensor,
                               {'DecodeJpeg/contents:0': image_data})
        predictions = np.squeeze(predictions)

        top_k = predictions.argsort0[-5:][::-1] # 가장 높은 확률을 가진 5개(top 5)의 예측값(predictions)을 얻는다.
        f = open(labelsFullPath, 'rb')
        lines = f.readlines()
        labels = [str(w).replace("\n", "") for w in lines]
        maxnum = 0
        for node_id in top_k:
            human_string = labels[node_id]
            score = predictions[node_id]
            print('%s (score = %.5f)' % (human_string, score))
            if maxnum < score:
                maxnum = score
                global target
                target = human_string

        target = target[2:-3]

        print('%s' % target)
        answer = labels[top_k[0]]
        return answer

```



```

import os
import sys
import urllib.request

client_id1 = "mgm6dmbkt3"
client_secret1 = "DWqg8wRc2bzLaurfil81YONQecpaLZchPGvQ0lz8"

url1 = "https://naveropenapi.apigw.ntruss.com/voice/v1/tts"

speakers = [
    'mijin', #한국어 여성
    'jinho', #한국어 남성
    'clara', #영어 여성
    'matt', #영어 남성
    'yuri', #일본어 여성
    'shinji', #일본어 남성
    'meimei', #중국어 여성
    'liangliang', #중국어 남성
    'jose', #스페인어 남성
    'carmen' #스페인어 여성
]

tmpPlayPath = './tmp.mp3'

class NaverTTS():
    def __init__(self, speaker=0, speed=0):
        self.speaker = speakers[speaker]
        self.speed = str(speed)
    def play(self, txt):
        encText = urllib.parse.quote(txt)
        data = "speaker=" + self.speaker + "&speed=" + self.speed + "&text=" + encText;

        request = urllib.request.Request(url1)
        request.add_header("X-NCP-APIGW-API-KEY-ID",client_id1)
        request.add_header("X-NCP-APIGW-API-KEY",client_secret1)
        response = urllib.request.urlopen(request, data=data.encode('utf-8'))
        rescode = response.getcode()
        if(rescode==200):
            response_body = response.read()
            with open(tmpPlayPath, 'wb') as f:
                f.write(response_body)

```

naver text to speech

```

import os
import sys
import urllib.request
client_id = "76RLMiToxyprvTo_G4fM"
client_secret = "N16Ey_WSPf"
encText = urllib.parse.quote("번역할 문장을 입력하세요")
data = "source=en&target=ko&text=" + encText
url = "https://openapi.naver.com/v1/papago/n2mt"
request = urllib.request.Request(url)
request.add_header("X-Naver-Client-Id",client_id)
request.add_header("X-Naver-Client-Secret",client_secret)
response = urllib.request.urlopen(request, data=data.encode("utf-8"))
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)

```

naver papago API