SKU11 - Virtual Assistant

28.04.2022

Gautam Srinivasan 9D

Nithish Murugavenkatesh 9D

Overview

Project: SKU11 is a Virtual Assistant with SpeechRecognition Technology and TTS Capabilities. It can derive keywords in your speech and execute commands preprogrammed in its code. The Virtual Assistant ONLY activates when a face is recognized using our State of the Art, HAAR Cascade Facial Landmark Detection.

Goals

1. To Make the Speech Recognition be at the least 65% Accurate(Alexa is 64% accurate)
2. To make the Virtual Assistant have a Natural Interface with a Human on a Verbal Level

Specifications

The Entire Project is Two Modules: The FaceRecognition Module, Which, If it returns True, Will start the Execution of the AI itself for perpetuity. Let us now focus mostly on the Virtual Assistant Module

Virtual Assisstant Module

The Virtual Assistant Has Mono means of Input, Output, and Process. The Input is a Google API client that converts speech to text, The Output is an Offline based WIN32Com.Client TTS(Text To Speech). Basically, the Virtual Assistant takes in speech input and runs it through a hotword detection module

Code Snippet: Speech recognition:

```
r = sr.Recognizer()
with sr.Microphone() as source:
    while True:


        print('listening')
        logging.info("Accessing Microphone.If Command Fails, Check Mic Privacy Settings.")
```

```
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)
        try:
            com = ""
            com = r.recognize_google(audio)

            print(com)
            lis = list(com.split(" "))
```

Code Snippet: Hotword Detection:

```
if 'Youtube' and 'open' in lis:
    speaker.Speak('Opening Youtube')
    wb.open('https://www.youtube.com/')
if "play" in lis:
    res = com.replace("play", "")
    speaker.Speak("Playing " + res)
    kittycat.playonyt(com)
if 'exit' in lis:
    speaker.Speak('Goodbye')

else:
    try:
        cattykit = wikipedia.summary(com,1)
        speaker.Speak(cattykit)
        print(cattykit)

        print('Excecute next function')
    except wikipedia.DisambiguationError or wikipedia.PageError:
        c = "Could Not Find What You Requested"
        speaker.Speak(c)
        print(c)
```

As you can see, If no Hotword is cross-matched successfully, it reverts to the default function, which is to Web-Scrape Wikipedia API on the input given.

WakeUpSequence(FaceRecognition):

This Module is basically the Initializer for the Virtual Assistant. Alexa uses the phrase, 'Hey Alexa'. We use Face Recognition instead, Which is more dependable, if we do say so ourselves :)

Haar Cascade Detection works on the Positive - Negative Image Principle. Basically, It takes in Images that contain the specified object and images that don't contain the object. And transforms it into an XML file, which is a fancy way of saying raw dataset. This is cross-referenced with the webcam input. If a face is detected in the said input, The Program initializes the SKU11 module and termites the face-recognition sequence.

```
import cv2
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades
+'haarcascade_frontalface_default.xml')
import SKU11
import time
import tkinter as tk
window = tk.Tk()
```

```
    # capture frames from a camera
cap = cv2.VideoCapture(0)

    # loop runs if capturing has been initialized.
while 1:

        # reads frames from a camera
        ret, img = cap.read()

        # convert to gray scale of each frames
```

```
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        # Detects faces of different sizes in the input image
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)

        for (x, y, w, h) in faces:

            # To draw a rectangle in a face
            cv2.rectangle(img, (x, y), (x + w, y + h), (0,0,235), 2, 1)
            roi_gray = gray[y:y + h, x:x + w]
            roi_color = img[y:y + h, x:x + w]
            time.sleep(2)
            cv2.destroyAllWindows()

            SKU11.MainProg()




        # Display an image in a window
        cv2.imshow('img', img)

        # Wait for Esc key to stop
        k = cv2.waitKey(30) & 0xff
        if k == 27:
            break


cap.release()
```

```
cv2.destroyAllWindows()
```

MILESTONES!

I. Successfully Created and Tested out Alpha v1.1 of the Project

Links to the Youtube Devlogs of the project will be made available.

Spent 8-9 hours a day making the project

II. Integrated it Successfully in our Daily lives.

We used it in our free time to automate our searching and Youtube Content watching.

https://www.youtube.com/watch?v=nsB4IMxidCQ -- Project Devlog

---