MS Smart Systems & IoT

Projet de Fin d'Études (PFE)

Rapport

Pratiques Liées au Raspberry Pi Basées sur la Technologie de l'IA

Étudiant :
Hao ZHANG

Enseignant :
Besma ZEDDINI
Fakhreddine GHAFFARI

7 juin 2021

1 Présentation du Sujet

Internet est le principal vecteur de diffusion de l'information dans la société de l'information d'aujourd'hui. Alors que l'influence d'Internet sur la société s'approfondit, l'idée de connecter des objets réels à Internet et de les rendre informatisés est naturellement née. C'est l'origine du développement de l'Internet des objets.

Les gens connectent des objets réels au réseau via des étiquettes électroniques, afin qu'ils puissent gérer et contrôler uniformément l'équipement à travers le réseau, et enfin analyser et traiter les données volumineuses collectées, ce qui peut permettre de nombreux changements majeurs, tels que des prévisions météorologiques locales plus précises, prévention du crime et surveillance des maladies, etc. Il existe de nombreux appareils bon marché et faciles à obtenir sur le marché qui peuvent répondre aux besoins de l'Internet des objets, et ils sont très simples à mettre en uvre, comme Arduino, Raspberry Pi, etc.

Ce Projet de Fin dÉtudes (PFE) a sélectionné le Raspberry Pi comme matériau de base du projet, basé sur le Raspberry Pi pour mettre en uvre certaines pratiques basées sur la technologie de l'intelligence artificielle liées à la voix et à la vision, telles que la détection de visage, la synthèse vocale et la reconnaissance vocale. Et j'espère pouvoir réaliser des technologies d'IA plus complexes telles que la reconnaissance faciale dans les études et recherches ultérieures.

2 Avantages d'application de Raspberry Pi

Raspberry Pi (Figure 1) est un petit ordinateur développé par la Fondation Raspberry Pi au Royaume-Uni. Son objectif initial était de fournir un équipement bon marché pour l'enseignement informatique à l'école, mais en raison de son prix bon marché, de sa taille semblable à une carte de crédit et de ses performances puissantes, il est rapidement devenu populaire. parmi les geeks. Dans ce que je vois, diverses plates-formes ont été construites dessus pour réaliser leur créativité. Pour les applications IoT, les avantages du Raspberry Pi peuvent être divisés en deux aspects majeurs : le matériel et le logiciel.

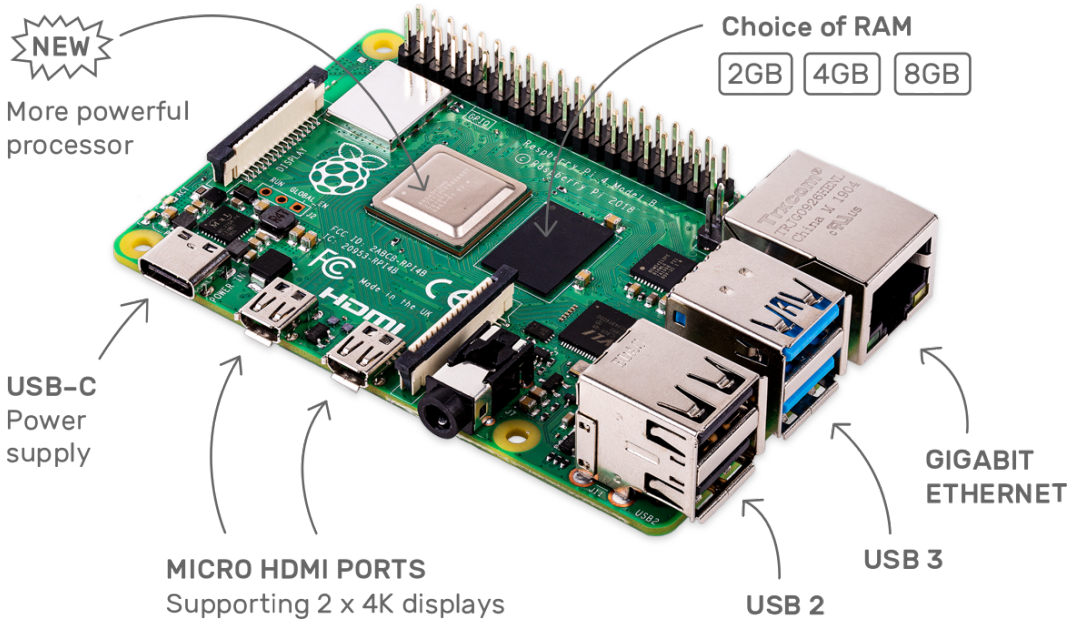

FIGURE 1 – Raspberry Pi 4

2.1 Avantage matériel

Comme nous le savons tous, l'équipement matériel est la base de tout, et toutes les idées ne peuvent être réalisées sans matériel. Le Raspberry Pi a été conçu à l'origine comme un ordinateur personnel, il a donc des performances assez puissantes. Les puissantes performances du processeur permettent au Raspberry Pi d'avoir des capacités de traitement de données plus puissantes et d'exécuter en douceur des tâches multithread, satisfaisant ainsi les divers besoins des utilisateurs. En tant qu'appareil IoT, Raspberry Pi a des performances très élevées et une variété d'interfaces, et son prix n'est pas cher, il est donc pleinement capable de la tâche de la plate-forme matérielle IoT.

2.2 Avantage logiciel

La victoire de n'importe quelle plate-forme est indissociable du support du logiciel, et le plus grand avantage du Raspberry Pi est le logiciel. Jusqu'à présent, il y a eu des dizaines de versions de système d'exploitation publiées dans la communauté Raspberry Pi, y compris Fedora, Ubuntu Mate, Windows IoT, etc., et le système d'exploitation officiel par défaut recommandé Raspbian est une branche de la version open source du système d'exploitation Linux Debian. Le nombre de progiciels dans sa bibliothèque de logiciels a dépassé de 350000, et un écosystème s'est formé. . Cela inclut naturellement la prise en charge de langages largement utilisés tels que Python, Java et C, ce qui offre sans aucun doute une commodité pour le développement ultérieur de logiciels IoT.

3 Se familiariser avec le Raspberry Pi à partir des petits programmes

3.1 Raspberry Pi et GPIO

Le premier module pour apprendre Raspberry Pi est RPi.GPIO. GPIO (General Purpose I/O Ports) signifie Ports dEntrée/Sortie à usage général, c'est-à-dire que certaines broches peuvent produire un potentiel élevé ou faible à travers elles, qui peuvent être utilisées librement par l'utilisateur par le contrôle du programme, communiquer avec l'appareil et atteindre le contrôle de l'appareil.

L'en-tête Raspberry Pi (Figure 2) est la clé de sa capacité à s'interfacer avec le monde réel. Le Raspberry Pi utilise soit un 40 broches ou 26 broches selon le modèle et il est important de comprendre comment ces broches sont disposées et étiquetées.

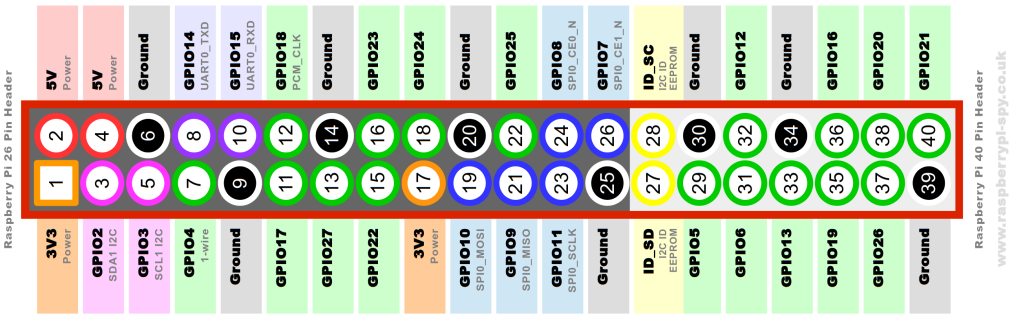

FIGURE 2 – En-tête Raspberry Pi

L'en-tête GPIO fournit les options d'alimentation et d'interface suivantes :

- 3.3V (sur 2 broches)
- 5V (sur 2 broches)
- Masse (sur 8 broches)
- Entrée et sortie à usage général
- PWM (modulation de largeur d'impulsion)
- I2C
- I2S
- SPI
- En série

Ceux-ci permettent à une vaste gamme de capteurs, de moteurs, de LED et d'accessoires d'être connectés au Pi.

3.2 Des petits programmes avec GPIO

3.2.1 Programme basique avec LED

```
import RPi.GPIO as GPIO
LED_PIN = 17

GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)
state = int(input("Enter 0 to power off the LED, 1 to power on the LED: "))

if state == 0:
  GPIO.output(LED_PIN, GPIO.LOW)
elif state == 1:
  GPIO.output(LED_PIN, GPIO.HIGH)
else:
  print("Wrong state value : " + str(state))
  GPIO.cleanup()
  exit

GPIO.cleanup()
```

3.2.2 Programme basique avec bouton

```
import RPi.GPIO as GPIO
BUTTON_PIN = 26

GPIO.setmode(GPIO.BCM)
GPIO.setup(BUTTON_PIN, GPIO.IN)

while True:
  print(GPIO.input(BUTTON_PIN))

GPIO.cleanup()
```

3.2.3 Programme basique avec capteur PIR

```
import RPi.GPIO as GPIO
PIR_PIN = 4

GPIO.setmode(GPIO.BCM)

GPIO.setup(PIR_PIN, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

while True:
  time.sleep(0.1)
  print(GPIO.input(PIR_PIN))

GPIO.cleanup()
```

3.2.4 Programme combiné (Figure 3)

```
import RPi.GPIO as GPIO
LED_PIN = 17
LED_PIN_LIST = [17, 27, 22]
PIR_PIN = 4

def power_on_selected_led_only(selected_led_pin):
  if selected_led_pin not in LED_PIN_LIST:
    return
  for pin in LED_PIN_LIST:
    if pin == selected_led_pin:
      GPIO.output(pin, GPIO.HIGH)
    else:
      GPIO.output(pin, GPIO.LOW)

GPIO.setmode(GPIO.BCM)
GPIO.setup(BUTTON_PIN, GPIO.IN)
GPIO.setup(PIR_PIN, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
for pin in LED_PIN_LIST:
  GPIO.setup(pin, GPIO.OUT)
for pin in LED_PIN_LIST:
  GPIO.output(pin, GPIO.LOW)

previous_button_state = GPIO.input(BUTTON_PIN)
previous_pir_state = GPIO.input(PIR_PIN)
led_index = 0

while True:
  button_state = GPIO.input(BUTTON_PIN)
  pir_state = GPIO.input(PIR_PIN)
  if button_state != previous_button_state:
    previous_button_state = button_state
    if button_state == GPIO.HIGH:
      power_on_selected_led_only(LED_PIN_LIST[led_index])
      led_index += 1
      if led_index >= len(LED_PIN_LIST):
        led_index = 0
  if pir_state != previous_pir_state:
    previous_pir_state = pir_state
```

```
    if button_state == GPIO.HIGH:
      power_on_selected_led_only(LED_PIN_LIST[led_index])
      led_index += 1
      if led_index >= len(LED_PIN_LIST):
        led_index = 0

GPIO.cleanup()
```

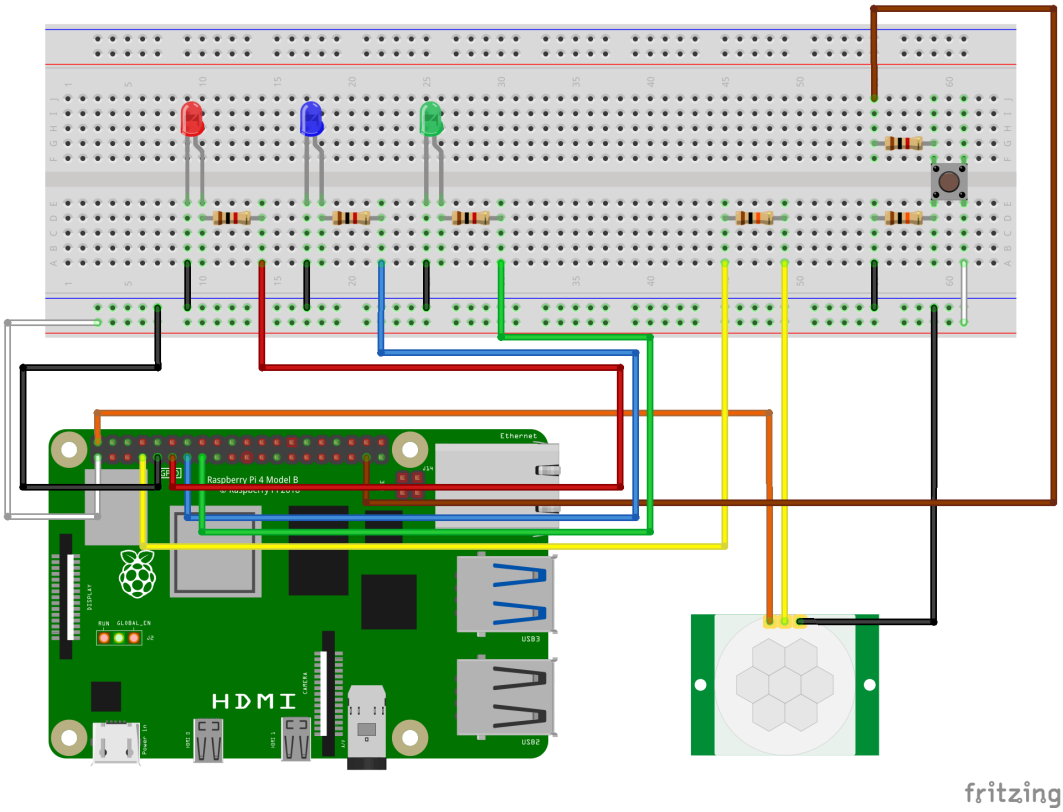

FIGURE 3 – Schéma en fritzing

3.3 Des petits programmes avec camera module v2

Le module de caméra Raspberry Pi v2 (Figure 4) a remplacé le module de caméra d'origine en avril 2016. Le module de caméra v2 dispose d'un capteur Sony IMX219 de 8 mégapixels (par rapport au capteur OmniVision OV5647 de 5 mégapixels de la caméra d'origine).

3.3.1 Programme en prise de photo

```
from picamera import PiCamera

camera = PiCamera()
camera.resolution = (1280, 720)
camera.rotation = 180

file_name = "/home/pi/Camera/image.jpg"
```

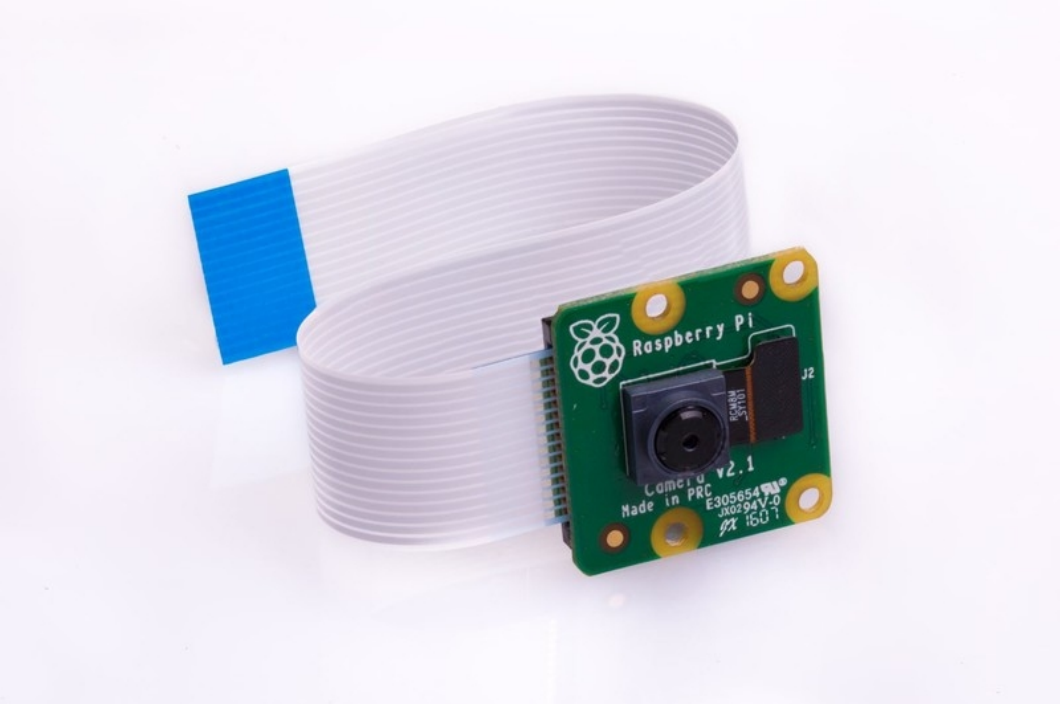

FIGURE 4 – Module de caméra Raspberry Pi v2 g

```
camera.capture(file_name)
print("Done.")
```

3.3.2 Programme en prise de vidéo

```
from picamera import PiCamera

camera = PiCamera()
camera.resolution = (1280, 720)
camera.rotation = 180

file_name = "/home/pi/Camera/video.h264"
camera.start_recording(file_name)
camera.wait_recording(10)
camera.stop_recording()
print("Done.")
```

4 Technologie de lIA liée à la vision

4.1 Détection faciale

La détection de visage est un domaine de la vision par ordinateur consistant à détecter un visage humain dans une image numérique. C'est un cas spécifique de détection d'objet, où l'on cherche à détecter la présence et la localisation précise d'un ou plusieurs visages

dans une image. C'est l'un des domaines de la vision par ordinateur parmi les plus étudiés, avec de très nombreuses publications, brevets, et de conférences spécialisées. La forte activité de recherche en détection de visage a également permis de faire émerger des méthodes génériques de détection d'objet.

La détection de visage a de très nombreuses applications directes en vidéo-surveillance, biométrie, robotique, commande d'interface homme-machine, photographie, indexation d'images et de vidéos, recherche d'images par le contenu, etc. Elle permet également de faciliter l'automatisation complète d'autres processus comme la reconnaissance de visage ou la reconnaissance d'expressions faciales.

4.2 Programme en détection faciale

```
import cv2
import argparse
import os
import numpy

DEFAULT_CASCADE_INPUT_PATH = 'haarcascade_frontalface_alt.xml'
DEFAULT_OUTPUT_PATH = 'FaceCaptureImages/'

class VideoCapture:
def __init__(self):
  self.count = 0
  self.argsObj = Parse()
  self.faceCascade = cv2.CascadeClassifier(self.argsObj.input_path)
  self.videoSource = cv2.VideoCapture(0)

def CaptureFrames(self):
  while True:
  frameNumber = '%08d' % (self.count)
  ret, frame = self.videoSource.read()
  screenColor = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
  faces = self.faceCascade.detectMultiScale(
    screenColor,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(30, 30),
  )
  cv2.imshow('Video', screenColor)
  if len(faces) == 0:
    pass
  if len(faces) > 0:
    print('Face Detected')
  for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    cv2.imwrite(DEFAULT_OUTPUT_PATH + frameNumber + '.png', frame)
  self.count += 1
  if cv2.waitKey(1) == 27:
    break

self.videoSource.release()
cv2.waitKey(500)
cv2.destroyAllWindows()
cv2.waitKey(500)

def Parse():
```

```
45    parser = argparse.ArgumentParser(
46    description='Cascade Path for face detection')
47    parser.add_argument('-i', '--input_path', type=str,
48    default=DEFAULT_CASCADE_INPUT_PATH, help='Cascade input path')
49    parser.add_argument('-o', '--output_path', type=str,
50    default=DEFAULT_OUTPUT_PATH, help='Output path for pictures taken')
51    args = parser.parse_args()
52    return args
53
54  def ClearImageFolder():
55    if not (os.path.exists(DEFAULT_OUTPUT_PATH)):
56      os.makedirs(DEFAULT_OUTPUT_PATH)
57    else:
58      for files in os.listdir(DEFAULT_OUTPUT_PATH):
59        filePath = os.path.join(DEFAULT_OUTPUT_PATH, files)
60        if os.path.isfile(filePath):
61          os.unlink(filePath)
62        else:
63          continue
64
65  def main():
66    ClearImageFolder()
67    faceDetectImplementation = VideoCapture()
68    faceDetectImplementation.CaptureFrames()
69
70  if __name__ == '__main__':
71    main()
```

4.3 Reconnaissance faciale

La reconnaissance faciale est une technique qui permet à partir des traits de visage :

- Dauthentifier une personne : cest-à-dire, vérifier quune personne est bien celle quelle prétend être (dans le cadre dun contrôle daccès).
- Didentifier une personne : cest-à-dire, de retrouver une personne au sein dun groupe dindividus, dans un lieu, une image ou une base de données.

En pratique, la reconnaissance peut être réalisée à partir dimages fixes (photos) ou animées (enregistrements vidéo) et se déroule en deux phases :

- A partir de limage, un modèle ou « gabarit »qui représente, dun point de vue informatique, les caractéristiques de ce visage est réalisé. Les données extraites pour constituer ce gabarit sont des données biométriques au sens du RGPD (article 4-14).
- La phase de reconnaissance est ensuite réalisée par la comparaison de ces modèles préalablement réalisés avec les modèles calculés en direct sur des visages présents sur limage candidate.

4.4 Détection des visages vs. Reconnaissance faciale

- Détection faciale
  - Détecte, ne mémorise pas le visage
  - Simple à mettre en oeuvre
- Reconnaissance de visage

- Détecte et reconnaît le visage
- Stocker les informations faciales
- Difficile et coûteux à mettre en uvre

5 Technologie de lIA liée à la voix

5.1 Synthèse vocale

La synthèse vocale est une technique informatique de synthèse sonore qui permet de créer de la parole artificielle à partir de n'importe quel texte. Pour obtenir ce résultat, elle s'appuie à la fois sur des techniques de traitement linguistique, notamment pour transformer le texte orthographique en une version phonétique prononçable sans ambiguïté, et sur des techniques de traitement du signal pour transformer cette version phonétique en son numérisé écoutable sur un haut parleur. Il s'agit, comme la reconnaissance vocale, d'une technologie permettant de construire des interfaces vocales. Parmi les applications, on peut citer la vocalisation d'écrans informatiques pour les personnes aveugles ou fortement malvoyantes (lecteur d'écran), ainsi que de nombreuses applications de serveurs vocaux téléphoniques, comme les annuaires vocaux de grande taille, où la synthèse vocale est la seule technique viable pour permettre la restitution sonore des noms et des adresses des abonnés.

5.2 Reconnaissance automatique de la parole

La reconnaissance automatique de la parole (souvent improprement appelée reconnaissance vocale) est une technique informatique qui permet d'analyser la voix humaine captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine.

La reconnaissance de la parole, ainsi que la synthèse de la parole, l'identification du locuteur ou la vérification du locuteur, font partie des techniques de traitement de la parole. Ces techniques permettent notamment de réaliser des interfaces homme-machine (IHM) où une partie de l'interaction se fait à la voix : « interfaces vocales ».

Parmi les nombreuses applications, on peut citer les applications de dictée vocale sur ordinateur où la difficulté tient à la taille du vocabulaire et à la longueur des phrases, mais aussi les applications téléphoniques de type serveur vocal interactif, où la difficulté tient plutôt à la nécessité de reconnaître n'importe quelle voix dans des conditions acoustiques variables et souvent bruyantes (téléphones mobiles dans des lieux publics).

5.3 Traitement du Langage Naturel (TAL)

Le traitement du langage naturel (TAL) est un sous-domaine de la linguistique, de l'informatique et de l'intelligence artificielle qui s'intéresse aux interactions entre les ordinateurs et le langage humain, en particulier la façon de programmer des ordinateurs pour traiter et analyser de grandes quantités de données en langage naturel. Le résultat est un ordinateur capable de « comprendre »le contenu des documents, y compris les nuances contextuelles de la langue qu'ils contiennent. La technologie peut ensuite extraire avec précision les informations et les informations contenues dans les documents, ainsi que catégoriser et organiser les documents eux-mêmes.

Les défis du traitement du langage naturel impliquent souvent la reconnaissance vocale, la compréhension du langage naturel et la génération du langage naturel.

5.3.1 Programme lié à la voix

```
import time
from datetime import datetime
from pygame import mixer
import speech_recognition as sr
from gtts import gTTS
from nltk.tokenize import word_tokenize

def SpeechSynthesis():
  mixer.init()
  tts = gTTS(text="Hello, what can I do for you? Recording starts", lang='
    en')
  tts.save("question.mp3")
  mixer.music.load('question.mp3')
  mixer.music.play()

def SpeechRecognition():
  mixer.init()
  response = None
  while (response == None):
    r = sr.Recognizer()
    with sr.Microphone() as source:
      r.adjust_for_ambient_noise(source, duration=1)
      print("Hello, what can I do for you? Recording starts.")
      print('3')
      time.sleep(1)
      print('2')
      time.sleep(1)
      print('1')
      time.sleep(1)
      audio = r.listen(source, phrase_time_limit=5)
    try:
      response = r.recognize_google(audio)
      print("I think you said '" + response + "'")
      tts = gTTS(text="I think you said " + str(response), lang='en')
      print("I think you said " + str(response))
      tts.save("response.mp3")
      mixer.music.load('response.mp3')
      mixer.music.play()
      time.sleep(5)
    except sr.UnknownValueError:
      print("Sphinx could not understand audio")
    except sr.RequestError as e:
      print("Sphinx error; {0}".format(e))
    return word_tokenize(str(response))

SpeechSynthesis()
response_list = SpeechRecognition()
if 'date' in response_list:
  mixer.init()
  tts = gTTS(text="Today is " + datetime.now().strftime("%d %B, %Y") + ".",
    lang='en')
  tts.save("date.mp3")
```

```
51    mixer.music.load('date.mp3')
52    mixer.music.play()
53    time.sleep(5)
54    break
```

6 Étapes pour réaliser le projet final

6.1 Détection faciale → Synthèse de discours

Démarrez le programme et ouvrez l'appareil photo via le programme. Afin de mieux capturer le visage, la vidéo obtenue par la caméra est ajustée au mode vidéo en niveaux de gris. Lorsqu'un visage humain apparaît dans la vidéo, le programme capture le visage humain, prend une photo en même temps et marque la photo dans un cadre de visage. L'image à ce moment est en mode normal.

Lorsque le programme capture le visage, il active immédiatement la fonction vocale pour interroger le sujet.

6.2 Synthèse de discours → Enregistrement vocal

Lorsque la fonction vocale est activée, une fois la question prédéfinie posée par voix synthétique, la réponse du sujet est enregistrée et sauvegardée sous forme de fichier MP3.

6.3 Enregistrement vocal → Reconnaissance de la parole

Ouvrez le fichier MP3 enregistré, analysez l'audio via la fonction de reconnaissance vocale et traduisez-le en texte.

6.4 Reconnaissance de la parole → Traitement du langage naturel

Le texte traduit est traité par traitement du langage naturel et des mots-clés sont proposés pour une analyse ultérieure.

7 Conclusion

Ce projet est encore un tout nouveau domaine pour moi, il y a beaucoup de fonctions préconçues, et malheureusement il y a beaucoup de choses qui n'ont pas été réalisées en raison de mes capacités limitées.

Tant la fonction visuelle que la fonction vocale sont impliquées qu'au début. Cependant, je crois qu'avec ce genre de fondation, il sera possible de faire des percées sélectives dans l'un de ces domaines.