

Développement logiciel

Table des matières

I.	système d'exploitation pour Raspberry Pi	1
1.	Raspbian	1
2.	Ubuntu MATE for Raspberry Pi	2
3.	Raspbian Stretch Lite :	2
II.	Acquisition d'image	2
1.	Picamera :	2
2.	OpenCV	2
III.	Traitement d'image	3
1.	Tesseract OCR :	3
2.	Pillow (Python Imaging Library - PIL fork)	4
IV.	Reconnaissance optique de caractères (OCR)(extraire le texte des images)	4
1.	Tesseract OCR :	4
V.	traduction	4
1.	googletrans	4
VI.	Synthèse vocale	5
1.	eSpeak	5
2.	Flite	5
3.	PicoTTS	5

I. système d'exploitation pour Raspberry Pi

<https://www.raspberrypi.com/software/operating-systems/>

1. Raspbian

Raspbian est le système d'exploitation de référence pour Raspberry Pi. Il est basé sur Linux Debian ([raspberrypi](#) + debian = raspbian) et il est très régulièrement mis à jour. En 2021, Raspbian a changé de nom, et devient [Raspberry Pi OS](#). L'ancien nom était plus facile à retenir, mais la nouvelle version est beaucoup plus facile à installer et utiliser.

<https://www.raspberrypi-france.fr/guide/systeme-exploitation-raspberry-pi/>

2. Ubuntu MATE for Raspberry Pi

Ubuntu MATE est une version de Ubuntu dotée de l'environnement de bureau MATE, spécialement conçue pour les Raspberry Pi. Cette distribution offre une expérience de bureau complète et conviviale, adaptée aux utilisateurs qui souhaitent utiliser leur Raspberry Pi comme un ordinateur de bureau.

<https://ubuntu-mate.org/raspberry-pi/>

3. Raspbian Stretch Lite :

Raspbian Stretch Lite était une version légère de Raspbian, le système d'exploitation basé sur Debian conçu spécifiquement pour les ordinateurs Raspberry Pi. Raspbian est optimisé pour fonctionner sur les processeurs ARM des Raspberry Pi,

II. Acquisition d'image

Développer de programme pour capturer des images de la caméra.

<https://projects.raspberrypi.org/en/projects/getting-started-with-picamera>

- **Choix de la bibliothèque de gestion de la caméra**

1. Picamera :

Picamera est une bibliothèque Python qui permet de contrôler la caméra officielle Raspberry Pi. Elle offre un contrôle complet sur les fonctionnalités de la caméra, y compris la capture d'images, l'enregistrement de vidéos, et bien plus encore. Picamera est optimisée pour fonctionner efficacement sur Raspberry Pi.

- **Exemple d'exécution de la caméra via picamera:**

```
: from picamera import PiCamera
  from time import sleep

  camera = PiCamera()

  camera.start_preview()
  sleep(5)
  camera.stop_preview()
```

2. OpenCV :

OpenCV (Open Source Computer Vision Library) est l'une des bibliothèques les plus utilisées pour les applications de Computer Vision. OpenCV-Python est l'API python pour OpenCV. OpenCV-Python est non seulement rapide car l'arrière-plan est constitué de code écrit en C/C++, mais il est également facile à coder et à déployer (en raison du wrapper Python au

premier plan). Cela en fait un excellent choix pour exécuter des programmes de Computer Vision intensifs en calcul.

- **Exemple d'exécution de la caméra via opencv :**

```
[ ]: import cv2

def capture_images():
    capture = cv2.VideoCapture(0)
    if not capture.isOpened():
        print("Impossible d'ouvrir la caméra.")
        return

    while True:
        ret, frame = capture.read()
        if not ret:
            print("Échec de la lecture de l'image.")
            break

        cv2.imshow('Camera', frame)

        if cv2.waitKey(0) & 0xFF == ord('q'):
            break

    capture.release()
    cv2.destroyAllWindows()

capture_images()
```

III. Traitement d'image

Implémentez des algorithmes de traitement d'image pour prétraiter les images, améliorer la qualité et extraire le texte.

- Conversion en niveaux de gris
- Filtrage pour éliminer le bruit
- Seuillage
- Amélioration du contraste
- Dilatation et érosion

1. Tesseract OCR :

Exemple de code Python qui applique ces techniques de prétraitement d'image à une image avant d'extraire le texte à l'aide de Tesseract OCR

```

• [2]: import cv2
import pytesseract

def preprocess_image(image_path):
    # Conversion en niveaux de gris
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    # Filtrage pour éliminer le bruit
    blurred = cv2.GaussianBlur(image, (5, 5), 0)
    # Seuillage
    thresh = cv2.adaptiveThreshold(blurred, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 11, 4)
    # Amélioration du contraste
    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
    # Dilatation
    dilated = cv2.dilate(thresh, kernel, iterations=1)

    return dilated

image_path = 'image.png'
preprocessed_image = preprocess_image(image_path)
text = pytesseract.image_to_string(preprocessed_image)

print("Texte extrait :", text)

```

2. Pillow (Python Imaging Library - PIL fork)

IV. Reconnaissance optique de caractères (OCR)(extraire le texte des images)

1. Tesseract OCR :

Tesseract OCR est parfaitement adapté pour extraire du texte à partir d'images sur Raspberry Pi. Voici comment vous pouvez l'utiliser pour effectuer la reconnaissance optique de caractères (OCR) et extraire du texte à partir d'images.

<https://circuitdigest.com/microcontroller-projects/optical-character-recognition-ocr-using-tesseract-on-raspberry-pi>

V. traduction

1. googletrans

Googletrans est une bibliothèque Python gratuite qui utilise l'API de traduction de Google Translate. Elle offre une interface simple pour traduire du texte entre différentes langues.

<https://pypi.org/project/googletrans/>

VI. Synthèse vocale



1. eSpeak :

<https://www.dexterindustries.com/howto/make-your-raspberry-pi-speak/>

2. Flite :

Flite est un autre moteur de synthèse vocale open-source qui peut être utilisé sur Raspberry Pi.

<https://learn.adafruit.com/speech-synthesis-on-the-raspberry-pi/speak-easier-flite>

3. PicoTTS :

PicoTTS est un moteur de synthèse vocale open-source développé par SVOX et distribué par Google. Il est léger et peut être utilisé sur Raspberry Pi pour générer de la parole à partir de texte.

<https://rpihome.blogspot.com/2015/02/installing-pico-tts.html>