

Rapport Projet TER : Conception d'une surveillance vidéo médicale

SALVAN Corentin - MEHONG-SHIT-LI Matthieu - MARTINEZ Damien
BAIRY ONAPA Mikaël - LE LIDEC Tristan

17 avril 2021

Table des matières

1	Avant propos	2
2	Introduction	3
3	Description	4
3.1	Définition du projet	4
3.2	Cahier des charges	4
3.3	Support Logiciels/ Langage de Programmation/ Librairie	4
4	Modélisation	5
4.1	Machine virtuelle OS Linux(Debian)	5
4.2	Machine virtuelle OS Raspberry Pi 2	5
4.3	Schéma de montage Hardware et description des composants	5
5	Réalisation Software-Hardware	6
5.1	Acquisition vidéo	6
5.2	Détection de chute	6
5.2.1	Extraction du sujet	6
5.2.2	Critère de détection de chute	6
5.3	Reconnaissance faciale	6
5.4	Interface Graphique	6
5.4.1	Intégration de l'acquisition vidéo	7
5.5	Commande du matériel	7
6	Bilan du projet	8
6.1	Difficulté du projet	8
6.2	Analyse du résultat confrontée au cahier des charges	8
6.3	Piste d'amélioration	8
6.3.1	Qualité du matériel	8
6.3.2	Résolution du système	8
6.3.3	Performance du code et du temps de calcul	8
6.4	Timeline (Diagramme de Grantt)	8
	Bibliographie	9
.1	Code principal du système	11
.2	Code librairie : RFID.py	11

Chapitre 1

Avant propos

L'objectif de cette unité d'enseignement est de nous préparer aux projets d'études de Master. Non content de consolider notre bagage scientifique, nous collaborons parmi un groupe de projet afin de sortir un produit fini, contexte qui pourrait se retrouver dans le milieu professionnelle également. Nous tenons avant tout à remercier Mr LAN SUN LUK Jean-Daniel pour l'aide précieuse apportée lors des points de blocages. Ce projet n'aurait pas aboutis dans le temps si nous n'avions consulter ouvrages et documents en ligne, sur lequel nous avons pu tirer informations, codes et librairies. Nous remercions donc ces écrivains parfois anonymes.

Chapitre 2

Introduction

Chapitre 3

Description

3.1 Définition du projet

Le but de ce projet est de réaliser une surveillance vidéo avec deux composantes, la détection de chute et la reconnaissance faciale. Pour la détection de chute, lorsqu'une chute sera détectée une alarme sera activée afin d'alerter le personnel soignant. Depuis une interface graphique l'utilisateur pourra visualiser en temps réel le flux vidéo, choisir la caméra, mettre en pause le flux vidéo et désactiver l'alarme. Concernant la reconnaissance faciale, l'utilisateur connaîtra le nom et le prénom de la personne si il est enregistré dans la base de données de l'établissement, ce qui permettra de détecter les éventuels intrus.

3.2 Cahier des charges

3.3 Support Logiciels/ Langage de Programmation/ Librairie

Plusieurs langages de programmation ont été discutés avant la réalisation du projet, dans un premier temps, nous avons opté pour le langage JAVA, notamment pour la programmation orientée objet qui facilite le travail en équipe lors du développement. Finalement nous nous sommes décidés pour le langage Python, car il bénéficie d'une grande communauté, possède de nombreuses bibliothèques, fonctionnalités que nous avons besoin. D'autre part c'est un langage que tous les membres de l'équipe savaient déjà programmer en Python. On utilise Virtual Box comme logiciel de simulation de différents systèmes d'exploitations. A partir des ressources d'une machine hôte ayant des performances correctes, dans notre cas un Windows 10, nous créons deux machines virtuelles de types Linux (Debian et Raspberry OS). La démarche vise avant tout à régler les configurations, installer logiciels et bibliothèques utiles et tester le code. Nous en parlerons dans la prochaine section.

Nous établissons une connexion SSH entre le Raspberry Pi (serveur) et notre PC Windows (client) dû au fait que nous avons pas de moniteur de contrôle. Le protocole SSH est un moyen de communication sécurisé et chiffré sur un réseau. (Une équivalence plus simple existe, le protocole Telnet, mais il transfère les données de façon trop visible sur le réseau et est donc sensible au piratage). (Bibliothèque 3, p9)

Nous réalisons le schéma électrique de la section 4.3 avec le logiciel KiCad, qui est un logiciel multi-plateforme pouvant dessiner un schéma électronique, créer une empreinte PCB pour la réalisation de circuits imprimés ou faire une modélisation 3D du circuit.

Chapitre 4

Modélisation

4.1 Machine virtuelle OS Linux(Debian)

4.2 Machine virtuelle OS Raspberry Pi 2

4.3 Schéma de montage Hardware et description des composants

Sur la partie matériel nous avons choisi d'utiliser un Raspberry Pi, du fait de son aspect compact et de sa grande capacité de calcul. Le modèle utilisé est le Raspberry 3b+, on énumère ici quelques caractéristiques :

- Mémoire vive : 1 GB, 1.2GHz Quad-Core ARM Cortex-A53
- Nombre de coeur : 4
- Alimentation : 5V, 2A
- Ports USB : 4
- Module Bluetooth 4.1 et Wifi
- Interface Carte Graphique : PCI-E
- Processeur : 4 x ARM(v7) 7100
- GPU : Dual Core VideoCore IV

On liste ici les ports GPIO de raspberry utile pour la connexion avec le module RFID :

RFID	Vcc +3.3V	GND	MISO	MOSI	SCK	SDA	RST
Raspberry	01-3.3V	06-Ground	21-GPIO09	19-GPIO10	23-GPIO11	24-GPIO08	22-GPIO25

Chapitre 5

Réalisation Software-Hardware

5.1 Acquisition vidéo

Grâce à la librairie openCV, nous pouvons faire l'acquisition de tout type de caméra, en effet il suffit que la caméra soit branché à l'ordinateur pour qu'on l'utiliser. La librairie prend aussi en charge les caméras IP et le fichier vidéo.

A chaque appel de la fonction `capture.read()`, la librairie capture l'image du flux vidéo en entré, cette image va ensuite être utilisé pour faire les différents traitements et notamment la détection de chute.

5.2 Détection de chute

La détection se décompose en deux parties, la première consiste à extraire le sujet de la scène et la deuxième étudier son déplacement afin de savoir si oui ou non il y a bien eu une chute.

5.2.1 Extraction du sujet

Pour extraire le sujet deux méthodes ont été étudiés, la première methode consiste à utiliser une technique de classification. c'est une technique d'apprentissage supervisé. Il faut d'abord créer une base de données afin d'entraîner le modèle pour pouvoir reconnaître le pattern souhaité. Cette méthode est coûteuse en ressource, mais peut être très efficace si la base de données est grande. Nous avons

5.2.2 Critère de détection de chute

5.3 Reconnaissance faciale

5.4 Interface Graphique

Nous avons cherché à créer une interface graphique pour notre porjet qui permette de modifier certains paramètres, de changer la caméra.

Pour réaliser cette interface nous avons utilisé la librairie intégrée de Python, Tkinter. Cette librairie permet de créer des interfaces graphiques assez simplement. Nous pouvons y ajouter des boutons des canvas (rectangle dans lequel nous pouvons placer du contenu).

La première chose que nous avons cherché à réaliser est l'intégration de l'acquisition vidéo à notre interface.

5.4.1 Intégration de l'acquisition vidéo

Pour intégrer l'acquisition vidéo à notre interface, nous devons faire en sorte qu'elle puisse être rafraichie pour modifier l'image affichée. La première idée que nous avons eu consistait à créer deux objets différents, un objet interface qui contiendrait les éléments de l'interface et permettrait d'en créer une. Puis un second objet acquisition, qui devait être l'acquisition vidéo. Nous n'aurions eu alors qu'à rafraichir l'image affichée.

Nous n'avons pas réussi à mettre cette solution en œuvre car l'acquisition vidéo ne possède pas de constructeur qui permette de créer un objet.

La seconde option que nous avons exploré consistait à réaliser un unique code qui contiendrait et l'acquisition vidéo et l'interface.

C'est cette dernière option qui a été retenue. Elle était plus simple de mise en œuvre.

5.5 Commande du matériel

Chapitre 6

Bilan du projet

6.1 Difficulté du projet

6.2 Analyse du résultat confrontée au cahier des charges

6.3 Piste d'amélioration

6.3.1 Qualité du matériel

6.3.2 Résolution du système

6.3.3 Performance du code et du temps de calcul

6.4 Timeline (Diagramme de Grantt)

Bibliographie

- [1] Ben Nuttall Revision, Edit on GitHub, *Librairie : GPIO zero*.
Consulté le 03/04/2021

`url = https://gpiozero.readthedocs.io/en/stable/index.html`

- [2] Raspberry Pi FR, *Utiliser un lecteur RFID avec Raspberry*.
Consulté le 03/04/2021

`url = https://raspberrypi.fr/rfid-raspberry-pi/`

- [3] Raspberry Lab fr, *Etablir une connexion SSH*.
Consulté le 17/03/2021

`url = https://raspberrypi-lab.fr/Debuter-sur-Raspberry-Francais/Connexion-Bureau-a-distance-Rasp`

Annexes

- .1 Code principal du système
- .2 Code librairie : RFID.py