PROTOTIPO DE SISTEMA DE SEGURIDAD DE UN EDIFICIO CON FINES DOCENTES

E. Fúnez Fernández, I. Ruano Ruano
Escuela Politécnica Superior de Linares

E. Estévez Estévez, J. Gámez García, J. Gómez Ortega
Escuela Politécnica Superior de Jaén
Universidad de Jaén
Email: edu.ffdez@gmail.com, {alonso, eestevez, jggarcia}@ujaen.es

Resumen

En este trabajo se muestra un prototipo a escala de un sistema de seguridad en el hogar de bajo coste que ha sido diseñado y realizado con fines docentes. El prototipo incorpora diferentes ejemplos de aplicación de integración de dispositivos, Internet de las Cosas (IoT), Inteligencia Artificial (IA) y control remoto. Se ha implementado como una maqueta que incorpora como dispositivo de control principal un ordenador tipo Raspberry Pi y una gran variedad de sensores y actuadores. El lenguaje de programación utilizado en el sistema es Python. Este lenguaje dispone de múltiples librerías, se han usado un gran número de estas para permitir integrar todos los elementos de interacción con el sistema y obtener un sistema controlable remotamente. Partiendo de este prototipo y de los trabajos necesarios para su obtención, se ha realizado una adaptación con el objetivo de diseñar y realizar prácticas docentes cuyas descripciones también forman parte de este trabajo.

Palabras clave: Internet of Things, Inteligencia Artificial, Integración de dispositivos, seguridad, Python.

1 INTRODUCCIÓN

En este artículo, se describe el diseño y desarrollo de un prototipo de un sistema de seguridad de bajo coste cuyos componentes se han adaptado con el fin de elaborar una serie de trabajos prácticos. Estos trabajos prácticos se pueden usar como elementos formativos y de evaluación en planes de diferentes temáticas y grados de ingeniería. Por este motivo se puede considerar como un sistema multidisciplinar que puede ser tratado por el alumnado de asignaturas en grados de Ingeniería Industrial, Ingeniería de Telecomunicaciones/Telemática e Ingeniería Informática.

El objeto inicial del sistema era el de servir como un Trabajo de Fin de Grado que incluyera la implementación práctica de tecnologías IoT (Internet of Things) en un prototipo de sistema de seguridad basado en Python con dispositivos domóticos. Dicho prototipo fue diseñado de manera que pudiese ser controlado por los usuarios del mismo mediante un bot de Telegram. Para la demostración práctica del prototipo se realizó una maqueta simulando su instalación en una vivienda [2]. En la Figura 1 se muestran 2 fotografías del prototipo del sistema obtenido.

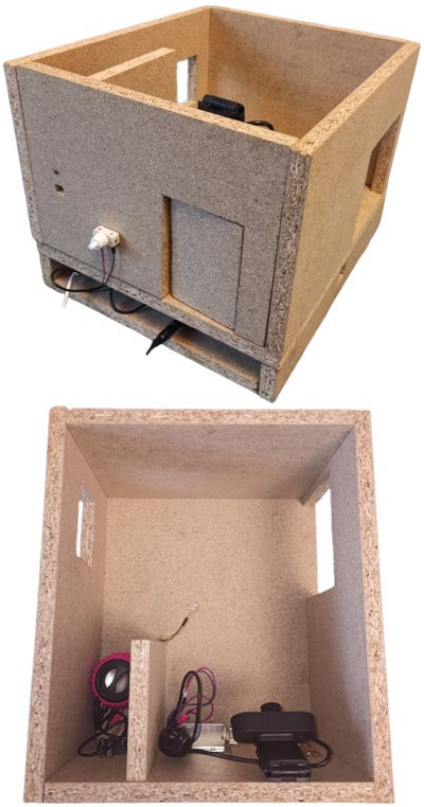

Figura 1: Prototipo de sistema de seguridad

Una de las partes a destacar del prototipo, además de la implementación de tecnologías IoT y de

 https://doi.org/10.17979/spudc.9788497498418.0270

comunicaciones (incluyendo un servidor de *streaming*), es la implementación de una IA (Inteligencia Artificial) capaz de realizar un reconocimiento facial mediante la captación de fotografías realizadas por una de las cámaras de las que dispone el sistema.

El contenido del resto del artículo se ha estructurado de la siguiente forma: en el apartado 2 se hace una descripción general del prototipo que incluye los componentes del sistema y las funcionalidades conseguidas, en el apartado 3 se describe brevemente la estructura del software desarrollado que se ejecuta en el sistema haciendo uso de múltiples librerías Python de libre distribución, el apartado 4 describe brevemente aplicaciones docentes que se pueden conseguir mediante el estudio y replicación de parte de los trabajos realizados para la obtención del sistema, el apartado 5 describe una práctica docente consistente en el uso de una IA para realizar reconocimiento facial y finalmente, en el apartado 6, se incluyen las conclusiones principales que se derivan de este trabajo.

2 DESCRIPCIÓN DEL PROTOTIPO

El elemento central y principal del prototipo es un ordenador de bajo coste Raspberry Pi modelo 4B (RPi4B a partir de ahora). Este dispositivo constituye el servidor donde se ejecuta el software principal que permite la integración y control de todos los elementos y subsistemas. Además, se encarga de actuar como pasarela para la información que irá, tanto a los diferentes elementos del ecosistema IoT del prototipo, como al usuario. El medio de interacción del usuario con el sistema está constituido por un bot de Telegram.

La Figura 2 muestra un esquema del sistema desarrollado que incluye los componentes HW principales y como puede interactuar un usuario con el mismo.

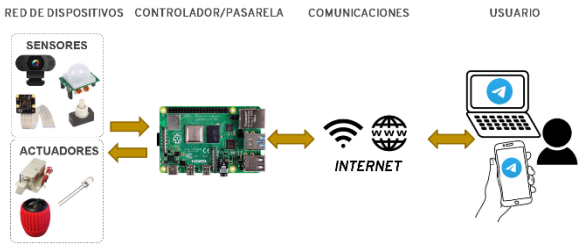

Figura 2: Esquema del Sistema

En un ecosistema IoT se pueden distinguir dos tipos de elementos que interactúan con el entorno: sensores y actuadores. Los dispositivos de este tipo incluidos en el prototipo de sistema de seguridad se enumeran a continuación.

Sensores:

- 3 detectores de presencia infrarrojos.
- 2 cámaras (exterior e interior).
- Un pulsador.

Actuadores:

- Una cerradura electrónica.
- 2 diodos LED (exterior e interior).
- Un altavoz.

Todos los componentes del prototipo se han incluido en una maqueta de madera de 5 mm de grosor que ha sido diseñada para simular un recinto con una única estancia que dispone de una puerta 2 ventanas y un falso suelo que sirve para ocultar parte del sistema (RPi4B, placa de conexionado y mayor parte del cableado del sistema). Todo esto puede apreciarse en la Figura 1.

2.1 RASPBERRY PI 4B

Constituye el elemento central, integrador de todos los dispositivos que se mostrarán a continuación y, sobre todo, el cerebro del sistema. Es un computador de placa única muy potente y asequible, razones por las cuales se ha usado en muchos proyectos relacionados con la robótica y la automatización. La Figura 3 muestra una imagen de la misma con indicación de sus componentes principales.

Figura 3: Raspberry Pi 4B

1. Procesador Broadcom BCM2711 (4 núcleos ARM Cortex-A72, 1,5 GHz y GPU3D VideoCore VI 500MHz).
2. 4GB de Memoria RAM.
3. WLAN 802.11b/g/n/ac a 2,4/5 GHz y Bluetooth 5.0.
4. Puerto Gigabit Ethernet
5. 4 Puertos USB dos 2.0 y otros dos 3.0.
6. Cabecera de 40 pines GPIO (General-Purpose Input/Output).
7. 2 puertos micro HDMI (4K a 60fps).
8. Puerto MIPI DSI para monitor.
9. Puerto MIPI CSI para módulo de cámara.
10. A/V: Puerto Jack hembra de 3.5mm para salida de audio y entrada de voz.
11. Slot de tarjetas micro SD para S.O. y almacenamiento.
12. Puerto USB-C para alimentación a 5V DC.

https://doi.org/10.17979/spudc.9788497498418.0270

2.1 CERRADURA DE SOLENOIDE

Para el prototipo, se ha usado una cerradura de solenoide como la que se puede ver en la Figura 4, controlable mediante el bot de Telegram por el usuario. Se conecta a la RPi4B por medio de pines GPIO (elemento 6 en la Figura 3).

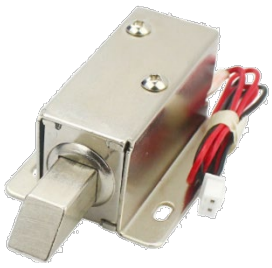

Figura 4: Cerradura de solenoide

La cerradura de solenoide funciona gracias a una bobina de alambre de cobre con un lingote de metal ferromagnético en medio. Este lingote metálico está conectado al pestillo y cuando pasa corriente por la bobina, se mueve el lingote de metal, retirando el pestillo y por lo tanto abriendo la cerradura [6], lo que permite la apertura de la puerta del recinto.

2.2 SENSORES DE PRESENCIA INFRARROJOS (PIR)

Los sensores PIR son un tipo de sensores que reaccionan ante determinadas fuentes de energía tales como el calor humano. El sensor recibe la variación de las radiaciones infrarrojas del área que cubre mediante su componente principal: los sensores piroeléctricos [16].

En el prototipo, los sensores PIR se usan para detectar movimiento y notificar al usuario mediante mensajes de aviso. Se pueden activar o desactivar mediante comandos en el bot de Telegram.

En el prototipo se usan sensores PIR modelo HC-SR501, como el que se ve en la Figura 5.

Figura 5: Sensor PIR HC-SR501

Los 3 sensores PIR utilizados en el sistema están situados en el falso suelo y hacen su función gracias a unos orificios situados en los dos laterales del prototipo y en la pared posterior del mismo. Se conectan a la RPi4B por medio de pines GPIO (elemento 6 en la Figura 3).

2.3 ALTAVOZ Y LUZ

Para la simulación de presencia se usa un altavoz (conectado a la RPi4B por medio del puerto A/V, elemento 10 en la Figura 3) y una luz (conectada a la RPi4B por medio de pines GPIO, elemento 6 en la Figura 3). La simulación de presencia consiste en hacer que desde el exterior parezca que la vivienda está habitada cuando no se está en el recinto. Esto se consigue mediante la automatización y control remoto de elementos de una casa domótica, como son las persianas, las luces o la música. Así, se disuade de intentos de ocupación y robo [1].

En el caso del diseño del prototipo de sistema de seguridad, se han usado un altavoz y una luz LED que, a demanda del usuario, se pueden encender o apagar en intervalos aleatorios de tiempo, en función a la detección de movimiento o manualmente.

2.4 CÁMARAS

Para el prototipo, se han usado dos cámaras: un módulo de cámara de Raspberry Pi conectada al puerto MIPI CSI de la RPi4B (elemento 9 en la Figura 3) y una cámara web USB conectada a la RPi4B por medio de uno de los puertos USB (elemento 5 en la Figura 3). Ambas cubren papeles diferentes dentro del funcionamiento del prototipo.

La cámara web USB se coloca dentro de la vivienda y actúa de cámara de seguridad. Si detecta algún movimiento mientras está activa, compila los fotogramas en los que se ha detectado movimiento en un GIF que se le envía al usuario por Telegram. Esta cámara puede ser activada o desactivada por el usuario mediante Telegram. En la Figura 6 se muestra su funcionamiento.

Figura 6: Detección de movimiento (cámara web)

https://doi.org/10.17979/spudc.9788497498418.0270

Por otro lado, el módulo de cámara, se coloca a la entrada de la vivienda actuando cumpliendo la función de timbre inteligente. Si alguien acciona el pulsador a la entrada de la vivienda, captura una fotografía sobre la que se efectúa un reconocimiento facial con una IA previamente entrenada por el usuario. Una vez efectuado el reconocimiento facial, se le notifica al usuario del resultado por Telegram y, en caso de ser positivo (reconocimiento de un usuario autorizado por el sistema), se desbloquea la cerradura de solenoide permitiendo la apertura de la puerta. Además, a demanda del usuario, se puede activar mediante Telegram un servidor web *Ad-Hoc* desde el que se puede acceder a la transmisión en directo del módulo de cámara, como se muestra en la Figura 7.

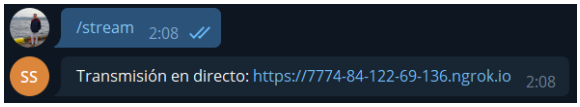

Figura 7: Uso del comando para activar la transmisión en directo

2.5 RESUMEN DE FUNCIONALIDADES

Las funcionalidades finales obtenidas en el sistema han sido las siguientes:

- Detección de movimiento en el exterior del prototipo, aviso a usuarios (Telegram) y posibilidad de activar una simulación de presencia en el sistema.
- Detección de movimiento en el interior del prototipo y aviso a usuarios (Telegram).
- Grabación de imágenes, creación de fichero GIF animado con imágenes del movimiento detectado y envío del mismo a usuarios (Telegram).
- Transmisión en directo (*streaming*) de imágenes del exterior del prototipo.
- Apertura/Cierre de la cerradura electrónica de forma manual remota.
- Apertura de la cerradura electrónica tras usar el pulsador localmente, obtener un reconocimiento positivo de la imagen del usuario captada por la cámara exterior basada en IA. Envío del resultado e imágenes a usuarios (Telegram).
- Simulación de presencia en interior del prototipo para disuadir posibles intentos de robo consistente en el encendido/apagado de luces y sonido que es configurable de forma manual y programada.
- Gestión de usuarios del sistema que incluye la incorporación de nuevos usuarios al sistema con posibilidad de actualizar el modelo de aprendizaje automático que realiza el reconocimiento de usuarios.
- Control remoto de la mayor parte de las funcionalidades del sistema descritas en los puntos anteriores a través de un bot de Telegram que también permite configurar el modo de funcionamiento.
- Registro de eventos sucedidos en el sistema.

3 ESTRUCTURA SOFTWARE

A continuación, se explicará el funcionamiento del software desarrollado y los elementos que lo componen. Lo primero que conviene tener en cuenta es que el servidor principal está programado para ejecutarse de forma automática al iniciarse el sistema, esto permite poder recuperar el funcionamiento normal ante un corte de energía no deseada. Si se deseara mantenerlo en funcionamiento aún en estas circunstancias debería tenerse en cuenta un sistema SAI que no ha sido contemplado inicialmente.
Una vez arranca el sistema, lo primero que se hace es iniciar el bot de Telegram para permitir el control del mismo y, posteriormente, queda a la espera de órdenes de usuario en un bucle continuo. Una vez recibida una orden se ejecuta la funcionalidad requerida en un hilo de ejecución paralela que termina una vez se ha conseguido el objetivo deseado. A continuación, se muestran las funcionalidades programadas de forma estructurada:

- Apertura y cierre de la puerta
- Detección de movimiento con sensores PIR
- Simulación de presencia
- Detección de movimiento con cámara USB
- Reconocimiento facial
- Transmisión en directo de la cámara a la entrada de la vivienda
- Registro de los eventos que suceden mientras el sistema de seguridad está activo

Cada una de estas funcionalidades se ha programado en segmentos de código independientes, lo que ayuda a su mantenimiento y gestión de fallos.
Debido a la importancia que tiene en el sistema para el funcionamiento del mismo en el siguiente subapartado se va a detallar el funcionamiento del bot de Telegram.

3.1 BOT DE TELEGRAM

Telegram es básicamente, y a primera vista, una aplicación centrada principalmente en mensajería instantánea, envío de archivos y comunicación en masa. Sin embargo, sus capacidades van mucho más allá, también es una plataforma de mensajería y VOIP que permite automatización de tareas masivas mediante bots. El bot de Telegram es una herramienta útil y accesible que Telegram proporciona. Se dice que es accesible porque para hacer uso de un bot de Telegram, el único requisito previo es descargar Telegram en un dispositivo compatible, crear una cuenta y abrir la conversación con el bot. El uso de este interfaz de control se puede realizar desde un

https://doi.org/10.17979/spudc.9788497498418.0270

dispositivo de escritorio usando un simple navegador web o desde cualquier dispositivo móvil en el que se use la aplicación de Telegram.

El bot de Telegram que se ha desarrollado tiene un papel doble en el sistema: es un centro de control y de notificaciones para el usuario final. Desde el bot de Telegram, mediante comandos y menús, el usuario puede activar las diferentes funciones del prototipo desde cualquier ubicación, siempre y cuando cuente con conexión a internet [14]. A la vez puede observar el estado del sistema al recibir notificaciones de todos los eventos que se producen en el prototipo.

3.2 LIBRERÍAS PYTHON UTILIZADAS

El catálogo de librerías Python utilizadas para la consecución de este trabajo ha sido muy extenso, como puede comprobarse al observar la siguiente lista:

- OpenCV – *Detección de movimiento con cámara USB y reconocimiento facial*
- Flask – *Transmisión en directo de la cámara a la entrada de la vivienda*
- PyNgrok – *Transmisión en directo de la cámara a la entrada de la vivienda*
- Imutils – *Detección de movimiento con cámara USB y transmisión en directo de la cámara a la entrada de la vivienda*
- Threading – *Ejecución simultánea de las funciones del prototipo*
- Multiprocessing – *Ejecución simultánea de las funciones del prototipo*
- RPi.GPIO – *Apertura/cierre de la puerta y simulación de presencia*
- Numpy – *Detección de movimiento con cámara USB y reconocimiento facial*
- Telegram.ext – *Gestión de las funciones del prototipo y notificaciones a los usuarios*
- Pygame – *Simulación de presencia*
- PyMongo – *Gestión de los usuarios registrados en el sistema con MongoDB*
- Random – *Simulación de presencia*
- SciKit-Learn – *Reconocimiento facial*
- Pickle – *Reconocimiento facial*
- Subprocess – *Registro de los eventos que suceden mientras el sistema de seguridad está activo*
- Signal – *Administración de los hilos de ejecución activos*

4. APLICACIÓN A LA DOCENCIA

En el apartado anterior, se ha explicado como el diseño del software del sistema se ha realizado de forma estructurada. Esto facilita la extracción de módulos que, una vez modificados y adaptados, pueden servir para diseñar prácticas docentes para la formación del alumnado en diferentes grados de Ingeniería. En los siguientes subapartados, se perfilarán algunas posibles aplicaciones docentes de los elementos anteriormente mencionados y, en el siguiente apartado, se incluye una descripción en profundidad de un trabajo práctico basado en la IA que se usa para efectuar reconocimiento facial en el prototipo.

4.1 CONEXIÓN Y CONTROL DE SENSORES Y ACTUADORES

Una primera práctica muy sencilla que se puede derivar del presente trabajo es la conexión a la RPi4B de todos los sensores y actuadores utilizados para, posteriormente, realizar una comprobación de funcionamiento correcto y control de los mismos. Si bien se trata de dispositivos que no ofrecen posibilidades de control muy complejas (la mayoría sólo permiten controles de tipo ON/OFF) puede ser un buen punto de inicio para incluir después dispositivos que ofrezcan más posibilidades, como puede ser un motor para la subida y bajada de persianas o apertura de otras puertas.

4.2 BOT SIMPLE EN TELEGRAM

Se puede crear una práctica que consista en la elaboración de un bot simple en Telegram. Se podría dejar a elección del alumno la utilidad de este bot, fomentando así la creatividad del alumno a la hora de diseñar este trabajo.

Una opción que el alumno podría escoger es la elaboración de un bot que, a petición del usuario, dé un pronóstico del clima. Para la creación de este bot haría falta estar registrado en Telegram y en *OpenWeather* o cualquier otra API abierta de clima. Una vez el alumno esté registrado en Telegram, podrá crear un bot en Telegram mediante el bot *@BotFather* que le proporcionará la clave API (Figura 6), necesaria para poder configurar y programar el bot. Con la clave API del bot y la que obtendrá de *OpenWeather*, podrá elaborar el bot [8], [12].

4.3. SERVIDOR WEB AD-HOC

Como se ha mencionado en la sección anterior, el prototipo de sistema de seguridad incluye la posibilidad de crear un servidor web *Ad-Hoc* que permite visualizar en directo el exterior del recinto gracias al módulo de cámara de RPi4B (servidor de *streaming*). A demanda del usuario este servidor puede iniciarse o cerrarse.

Un posible trabajo práctico que se podría realizar es la implementación de un servidor *Ad-Hoc* en Python para alojar un servicio web. Para ello sólo haría falta instalar las dependencias *pyngrok* y *flask* mediante *pip* [4], [13].

 https://doi.org/10.17979/spudc.9788497498418.0270

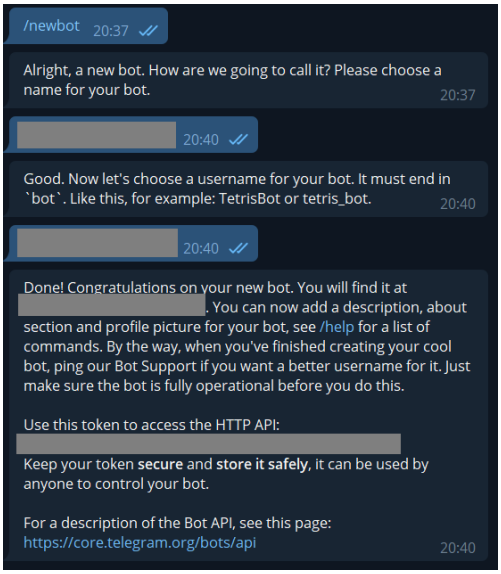

Figura 8: Creación de un bot de Telegram

El servidor web se crea localmente mediante *flask* y se hace público mediante *pyngrok*. *Flask* es un framework para Python que permite crear aplicaciones web y *pyngrok* es una librería que permite la implementación de túneles seguros desde URLs públicas a servidores locales basada en *Ngrok* [10].

Para usar túneles *Ngrok*, primero hace falta hacer algunas configuraciones:

1. En primer lugar, hace falta registrarse en la página web de *Ngrok* y obtener el *Authtoken*.
2. Una vez se tiene el *Authtoken*, se tiene que ejecutar el comando "*./ngrok authtoken [código Authtoken]*" dentro del directorio de instalación de *Ngrok* que se crea al instalar *pyngrok*.

Con esas configuraciones, *Ngrok* quedará listo para poder ser implementado en el código.

4.4 DETECCIÓN DE MOVIMIENTO CON WEBCAM

Otro trabajo práctico derivado del sistema es la implementación de detección de movimiento mediante una cámara web. En el prototipo, esta función usa la librería *OpenCV* y sigue la siguiente lógica [11]:

1. La cámara web empieza a captar imágenes cuando el usuario la activa mediante Telegram.
2. Se almacenan pares de fotogramas en variables (fotograma actual y fotograma anterior) y se comparan, buscando algún cambio entre ellos.
3. Si se detecta algún cambio entre ellos, se almacenan los fotogramas en los que se ha detectado movimiento en el sistema.
4. Cuando se llega al límite de diez fotogramas, se compilan en un GIF y se envían al usuario.

Esta lógica de funcionamiento se puede aplicar para realizar un trabajo práctico. En vez de comparar fotograma a fotograma, también se puede hacer una media entre N fotogramas y comparar nuevos fotogramas con esa media para reducir la probabilidad de falsos positivos.

5 RECONOCIMIENTO FACIAL CON IA

Este trabajo práctico se describe en profundidad en este apartado. Para este propósito, se ha adaptado parte del código del prototipo de sistema de seguridad y se ha realizado el trabajo práctico mencionado.

Se ha escogido el entorno de programación *Google Colab* para este trabajo práctico [5]. Las razones por las que se ha elegido este entorno de programación son su accesibilidad, ya que se puede acceder a él mediante un navegador, y porque tiene preinstaladas muchas de las librerías que se usan en el trabajo práctico. Sólo es necesario que el estudiante tenga una cuenta Google para poder tener acceso a este servicio y realizar la práctica (además del HW necesario y conexión a Internet).

Al alumno se le suministra un código en forma de notebook que debe de manipular para realizar las acciones pedidas. El objetivo del mismo es la realización de una detección biométrica basada en el reconocimiento facial. Como ya se ha comentado, debe utilizar un equipo con conexión a Internet, cámara Web y su cuenta Google. Este notebook contiene un código que sirve para comprobar el rostro captado por la cámara web del dispositivo y detectar si se trata de un usuario desconocido (unknown) o no, indicando el porcentaje de incertidumbre del resultado. El funcionamiento de este código se describe a continuación.

En primer lugar, se importan las librerías necesarias para el funcionamiento del código. Dos de ellas son *OpenCV*, para el tratamiento de las imágenes y detección de caras dentro de las mismas, y *sklearn*, para el entrenamiento y uso del modelo que se usa para efectuar reconocimiento facial [15].

Después, se descargan los componentes del detector facial y el conjunto de imágenes, que se usará para que el modelo pueda reconocer caras como desconocidas (se compone de fotografías de caras de personas aleatorias). Esta descarga se realiza desde un

https://doi.org/10.17979/spudc.9788497498418.0270

repositorio de GitHub que ha sido preparado previamente. Una vez descargados, la carpeta de contenidos ha de quedar como se muestra en la Figura 9.

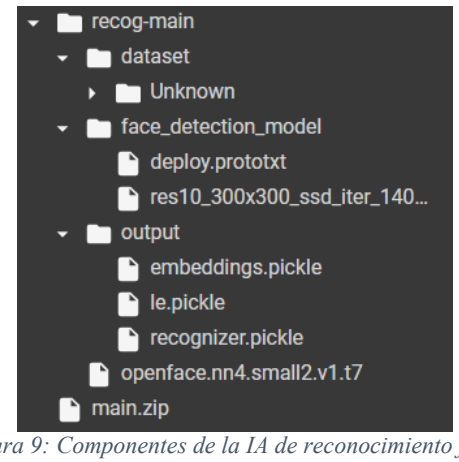

Figura 9: Componentes de la IA de reconocimiento facial

Una vez hechas las preparaciones anteriores, el alumno debe realizar la captura de una serie de fotografías de su cara mediante la cámara web que debe incorporar el dispositivo desde el que trabaje. Las fotografías se almacenan dentro de una carpeta dedicada, como se muestra en la Figura 10, y se usarán para el posterior entrenamiento del modelo, el número de capturas puede determinar la calidad del modelo obtenido.

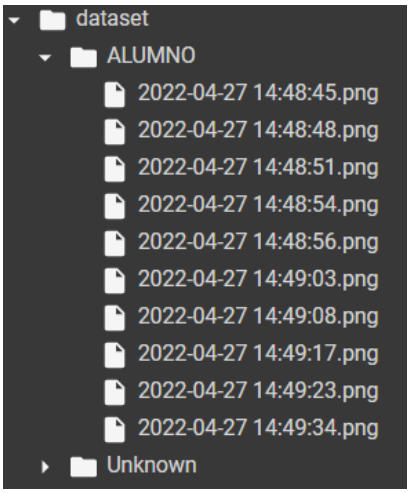

Figura 10: Imágenes capturadas por el alumno

El entrenamiento del modelo es la parte del trabajo práctico en la que el alumno tiene que completar el código. En primer lugar, se carga el detector facial que se ha descargado del repositorio de GitHub y se cuantifican las imágenes que componen el *dataset* de entrenamiento. Cuando las imágenes se han cuantificado, se procesan una a una, detectando la cara dentro de la imagen y obteniendo un *blob* (Binary Large OBject) a partir de la región de interés que contiene a la cara [7]. Este *blob* se pasa como entrada al modelo de incrustación que se ha descargado anteriormente y se genera un vector 128-D que describe las características de la cara [3]. Una vez hecho el procesado, se serializan los datos en un fichero *pickle*, que convierte la jerarquía del objeto que contiene el vector 128-D y los nombres conocidos en una cadena de bytes para poder almacenarlo fácilmente en memoria. Después, el fichero *pickle* se carga y se codifican las etiquetas con los nombres de las incrustaciones faciales. Hecho esto, sólo queda entrenar el modelo con un clasificador de la librería *sklearn*. Para el prototipo de sistema de seguridad se usó una clasificación por vectores de soporte (SVC) con kernel lineal. Sin embargo, como trabajo del alumno se puede proponer el uso de otros algoritmos y realizar un estudio posterior de la calidad ofrecida por cada uno de ellos.

Habiendo entrenado el modelo, el alumno puede capturar una fotografía con la webcam para efectuar un reconocimiento facial sobre ella. Para realizar el reconocimiento facial, primero se carga el detector facial y el archivo *pickle* que se ha guardado al final del entrenamiento del modelo. A continuación, se crea un *blob* de la imagen sobre la que se realiza el reconocimiento facial y se pasa como entrada al detector facial, que saca a su vez una región de interés de las caras que detecte. Luego, se recorren todas las regiones de interés y se extrae el vector 128-D de cada una de ellas. Con este vector, el modelo de reconocimiento facial podrá realizar una predicción de la etiqueta que corresponde a la cara detectada. El resultado del reconocimiento facial se muestra por pantalla.

6 CONCLUSIONES

Se puede concluir, a la luz de los resultados, que la adaptación de partes del Trabajo analizado en este artículo permite proponer trabajos prácticos que pueden resultar enriquecedores desde el punto de vista docente.

El hecho de que los alumnos puedan realizar prácticas de problemas reales y obtener resultados tangibles de su trabajo les resulta satisfactorio y aumenta su motivación por la asignatura. Además, se obtienen conocimientos y habilidades útiles para familiarizarse más con el lenguaje de programación y los elementos usados.

Se podría proponer también un enfoque de aprendizaje basado en proyectos (ABP) a partir del prototipo obtenido. Este enfoque es interesante por su carácter motivador y creativo, además de por el aprendizaje autónomo que afianza los conocimientos adquiridos en el proceso [9].

https://doi.org/10.17979/spudc.9788497498418.0270

Agradecimientos

Los autores quieren agradecer la subvención parcial de este trabajo a través de los proyectos PID2019-110291RB-I00 del Ministerio de España, FEDER Andalucía A1123060E00010 con referencia 1380776 y PIMED01_201921 de la Universidad de Jaén.

English summary

PROTOTYPE OF SECURITY SYSTEM OF A BUILDING FOR TEACHING PURPOSES

Abstract

This paper shows a scale prototype of a low-cost home security system that has been designed and built for educational purposes. The prototype incorporates different application examples of device integration, Internet of Things (IoT), Artificial Intelligence (AI) and remote control. It has been implemented as a model that incorporates a Raspberry Pi-type computer, as the main control device, and a wide variety of sensors and actuators. The programming language used in the system is Python. This language has multiple libraries, a large number of these have been used to allow integrating all the elements of interaction with the system and obtain a remotely controllable system. Starting from this prototype and the necessary work to obtain it, an adaptation has been made with the aim of designing and carrying out teaching practices whose descriptions are also part of this work.

Keywords: Internet of Things, Artificial Intelligence, security, Python.

Referencias

[1] "Aumenta la seguridad de tu hogar gracias a la simulación de presencia," *Simon Electric*, Feb. 2021. https://www.simonelectric.com/blog/aumenta-la-seguridad-de-tu-hogar-gracias-la-simulacion-de-presencia (accessed Jan. 22, 2022).

[2] E. Fúnez-Fernandez and I. Ruano-Ruano, "Diseño de un sistema de seguridad en el hogar basado en IoT y creación de prototipo," Universidad de Jaén, Linares, 2022. Accessed: Apr. 27, 2022. [Online]. Available: https://hdl.handle.net/10953.1/16437

[3] L. Dulčić, "Face Recognition with FaceNet and MTCNN," *arsfutura*, 2020. https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/ (accessed Apr. 27, 2022).

[4] "Flask Documentation," *Pallets Projects*. https://flask.palletsprojects.com/en/2.0.x/ (accessed Feb. 04, 2022).

[5] "Google Colab," *Google Colab*. https://colab.research.google.com/ (accessed Apr. 27, 2022).

[6] "How Solenoids Make Automotive Doors More Secure," *Johnson Electric*. https://us.johnsonelectric.com/solenoids-automotive-doors-locks-secure/ (accessed Apr. 27, 2022).

[7] R. Caubalejo, "Image Processing — Blob Detection," *Towards Data Science*, Jan. 29, 2021. https://towardsdatascience.com/image-processing-blob-detection-204dc6428dd (accessed Apr. 27, 2022).

[8] D. Widya Putra, "Learn to build your first bot in Telegram with Python," Dec. 13, 2018. https://www.freecodecamp.org/news/learn-to-build-your-first-bot-in-telegram-with-python-4c99526765e4/ (accessed Nov. 23, 2021).

[9] M. Panasan and P. Nuangchalerm, "Learning Outcomes of Project-Based and Inquiry-Based Learning Activities," *Journal of Social Sciences*, vol. 6, no. 2, pp. 252–255, 2010.

[10] J. A. Ponce, "Ngrok: una herramienta con la que hacer público tu localhost de forma fácil y rápida," *SDOS*, Sep. 01, 2020. https://www.sdos.es/blog/ngrok-una-herramienta-con-la-que-hacer-publico-tu-localhost-de-forma-facil-y-rapida (accessed Feb. 04, 2022).

[11] "OpenCV," *OpenCV*. https://opencv.org/ (accessed Apr. 27, 2022).

[12] "OpenWeather," *OpenWeather*. https://openweathermap.org/ (accessed Apr. 27, 2022).

[13] A. Laird, "Pyngrok (a Python wrapper for ngrok) Documentation," *Read the Docs*, 2020.

https://pyngrok.readthedocs.io/en/latest/index.html (accessed Feb. 04, 2022).

[14] L. Toledo, "Python Telegram Bot's documentation." https://python-telegram-bot.readthedocs.io/en/stable/ (accessed Apr. 27, 2022).

[15] "scikit-learn," *scikit-learn*. https://scikit-learn.org/stable/index.html (accessed Apr. 27, 2022).

[16] "SENSOR POR MICROONDAS (MWS) EN COMPARACIÓN CON EL SENSOR PIR (INFRARROJO PASIVO)," *Ansell Lighting*. https://ansell-lighting.es/news/Sensor-Por-Microondas (accessed Jan. 21, 2022).

https://doi.org/10.17979/spudc.9788497498418.0270