

Face detection for efficient video-surveillance IoT based embedded system

Vela-Medina J.C., Guerrero-Sánchez A.E., Rivas-Araiza J.E. and Rivas-Araiza E.A.

Abstract— Video Surveillance systems are widely used in indoor and outdoor environments for prevention and security monitoring. Most of conventional video surveillance systems are designed to store huge amount of data which difficult efficient access to the data from remote locations due to bandwidth requirements. A smart surveillance system allows efficient data storage and flexible data access. In this document the design and development of an embedded system for intelligent video surveillance with IoT capabilities is presented. For this project, an OMRON biometric sensor with specific features for face, body and hand detection was used. Face detection provides a criterion for event detection and efficient data capture of the data. The information of interest can be retrieved from a smartphone through Telegram X app. The system was tested under different face conditions including variations of pose, partial occlusion and expression. The system was developed with specific and smart devices providing new and different designs, easily to connect and control for users, without forgetting the importance of security.

keywords—Internet of Things, Video Surveillance, Embedded System, Face Detection, Intelligent Building.

I. INTRODUCCIÓN

INTERNET de las cosas o mejor conocido por “IoT” por sus siglas en inglés, es un concepto empleado para denotar la habilidad de conectar, comunicar y monitorear una red de dispositivos electrónicos, que mediante el uso de sensores y una interfaz de programación de aplicaciones (API) pueden intercambiar datos vía internet [1]. Entre las aplicaciones que han aprovechado el uso de IoT se encuentran los sistemas de video vigilancia, los cuales llevan a cabo un papel de suma importancia tanto para la seguridad pública como personal. Sin embargo, el monitoreo continuo en los sistemas de video vigilancia tradicionales es complicado debido a la gran cantidad de información de video que se almacena, además que el proceso es muy repetitivo, cansado y tedioso ya que siempre se obtendrá una imagen similar de la escena a menos que suceda alguna situación específica, esto conlleva a que la búsqueda de un evento particular sea una tarea laboriosa [2].

Este proyecto se ha realizado gracias al apoyo del Consejo Nacional de Ciencia y Tecnología (CONACyT) de México, FOMIX-QRO-2016-2 279773”.

J.C. Vela-Medina, División de Mecatrónica y Tecnologías de la Información y Comunicación, UTSJR (e-mail: juancvelam@hotmail.com).

A.E. Guerrero-Sánchez, Facultad de Ingeniería, Universidad Autónoma de Querétaro (e-mail: eliza_xeww@hotmail.com).

J.E. Rivas-Araiza, Facultad de Ingeniería, Universidad Autónoma de Querétaro (e-mail: jerivas@uaq.edu.mx).

E.A. Rivas-Araiza, Facultad de Ingeniería, Universidad Autónoma de Querétaro (e-mail: erivas@uaq.mx).

Actualmente el desarrollo tecnológico de video vigilancia ha aumentado a gran escala donde empresas líderes de desarrollo se han dado a la tarea de verla no solo como un método de seguridad o prevención al delito, sino como un servicio, facilitando el acceso al usuario, mejorando la administración y análisis del video, permitiendo no solo un reconocimiento facial sino hasta su aplicación como parte de un Sistema de Administración en Edificios (BMS – Building Management System) llevando a cabo tareas de administración, optimización y monitoreo en Edificios Inteligentes.

Inclusive IBM menciona que el análisis de video y la video vigilancia son grandes áreas de investigación activas, de igual manera señala la clave de esta tecnología se encuentra en la detección, identificación y rastreo de personal a partir de secuencias de video obtenidas desde una cámara, así como sistemas de vigilancia a gran escala [3].

En este trabajo se presenta el desarrollo de un sistema embebido que utiliza detección facial para mejorar el desempeño de un sistema de video vigilancia. Para ello se utiliza una cámara inteligente que incorpora algoritmos para detección y reconocimiento de rostros, de esta manera se captura y almacena una imagen en una base de datos en respuesta a un evento ocurrido en una secuencia de video. La habilitación IoT se obtiene mediante la conectividad entre el sistema embebido y una aplicación de mensajería instantánea (Telegram), de tal forma que el usuario recibe avisos de eventos y permite la consulta de las imágenes capturadas.

II. REVISIÓN DEL ESTADO DEL ARTE

El presente trabajo se enfoca hacia dos componentes principales: video vigilancia inteligente y el concepto de IoT aplicado a dichos sistemas. El análisis del estado del arte se enfoca en trabajos dentro de las áreas de interés arriba mencionadas, de tal forma que se contribuya con la búsqueda de mejora continua, lo que nos permite identificar nuevas técnicas de desarrollo, ampliando así el conocimiento y llevando a cabo un nuevo aporte tecnológico.

En el trabajo de Quadri y Sathish[4] se presenta un sistema de control de acceso mediante video vigilancia. Para demostrar su metodología emplean una maqueta a escala que utiliza una tarjeta Raspberry Pi, una webcam conectada vía USB, un motor DC para indicar la apertura o cierre de una puerta. El sistema está conectado a un router ethernet que permite el acceso remoto desde una tablet o teléfono inteligente. Un script realizado en python implementa un formulario de ingreso a la aplicación que permite al usuario monitorear la secuencia de video continuo y realizar acciones

de apertura/cierre de la puerta, así como de captura de imagen. Este trabajo aborda el tema de video vigilancia desde una perspectiva didáctica y presenta limitaciones para su aplicación práctica en un entorno real de operación.

En el trabajo realizado por Ansari et al. [5] desarrollan un sistema que realiza capturas de imagen o video desde una cámara USB conectada a una computadora embebida Raspberry Pi cuando se detecte un movimiento, además los datos obtenidos son transferidos a un servidor externo en la nube para su acceso remoto. La metodología adoptada permite simplificar la detección de movimiento y hacer una interfaz más amigable para el usuario el cual recibe notificaciones cuando se realiza una detección de movimiento. El envío de datos se realiza por tres vías que incluyen transferencia directa por FTP (Protocolo de Transferencia de Ficheros) o SFTP (Protocolo de Transferencia de Ficheros que utiliza SSH [Secure Shell]) o desde un servicio gestionado en la nube para la imagen guardada. Los autores identifican como su principal aplicación las casas, edificios y la posible alternativa de uso en sistemas de seguridad, tomando en cuenta que la aplicación captura fotografías cada que se detectan diferencias entre dos imágenes consecutivas, lo cual no es un método confiable en vista de las posibilidades de variaciones de la escena debida a aspectos propios del ambiente bajo prueba.

Por otra parte en el artículo desarrollado por Kumar et al. [6] presentan un modelo unificado para monitoreo y síntesis de datos correspondientes a una secuencia de video. Además, incluyen el pre-procesamiento de la imagen para mejorar su funcionamiento ante condiciones climáticas adversas a las que podrían enfrentarse los sistemas de video vigilancia. Este sistema permite simplificar partes de video continuo para reducir el espacio de la memoria de almacenamiento logrando un sistema de monitoreo automático, que decide que parte es importante del video y cual es repetitivo. El sistema resuelve uno de los mayores problemas de video vigilancia el cual es la repetitividad; al hacer un resumen de video automáticamente debido a la capacidad humana de decidir si es importante o no podría llegar a ocasionar un corte del video que sea relativamente importante, haciendo un sistema poco eficiente, pero con grandes posibilidades de mejora.

En los trabajos anteriores se encuentran grandes ideas y objetivos similares que cumplen con la función establecida de realizar video vigilancia inteligente, contando con ventajas y desventajas para su uso, así como algunos puntos a favor para perfeccionarlos y trabajarlos, tomando de estos las áreas de oportunidad de cada uno para realizar un sistema que resuelva cada uno de estos logrando no solo un sistema embebido de video vigilancia sino una herramienta para la seguridad del ser humano.

Esta rama de investigación actualmente es de gran importancia para la sociedad y sus avances tecnológicos así mismo el campo de trabajo de video vigilancia conectada a la red nos permitirá tener el control en los hogares, oficinas u otros lugares desde la comodidad de la palma de nuestra mano dando no solo la posibilidad de conocer los horarios de acceso, sino también el control por mencionar algunas de las posibilidades y opciones para el desarrollo de esta aplicación

proporcionando un instrumento de uso diario ya sea en situaciones públicas o privadas.

Gracias a la revisión del estado del arte es posible identificar las áreas de trabajo que se encuentran para esta tecnología hoy en día necesaria para la seguridad, permitiéndonos aportar mejoras en estos estudios que logran la incorporación de IoT con la video vigilancia inteligente ayudando a los sistemas de seguridad dando una automatización y haciendo de ella una mejor herramienta para el usuario y su mayor eficacia.

III. ARQUITECTURA DEL SISTEMA PROPUESTO

Para desarrollar un sistema embebido de video vigilancia inteligente se requiere grabación de video continua a partir de la cual se puedan detectar eventos de interés que permitan almacenar información que pueda ser rápidamente visualizada e interpretada por el usuario. En este trabajo se propone un sistema de video vigilancia que emplea una condición para detección para recabar información de interés, reduciendo la cantidad de datos redundantes e innecesarios. El criterio principal que determina la ocurrencia de un evento de interés queda determinado por la detección automática de rostros en secuencia de video, de esta forma la información sobre los rostros detectados se almacena en forma de una base de datos MySQL. La imagen se puede registrar junto con la fecha y hora del momento que se capturó en la base de datos, esta información puede ser recuperada de manera local o remota.

En la Fig. 1 se muestra el diagrama del sistema propuesto comprende una cámara inteligente desarrollada por OMRON Electronic Components modelo “B5T-007001 Human Vision Components (HVC-P2)”, la cual se aprecia más a detalle en Fig. 2. Ésta se conecta vía USB a una computadora embebida tipo Raspberry Pi 3 B+ la cual cuenta con sistema operativo Raspbian y un servidor de base de datos MySQL configurado. El usuario puede acceder a la información contenida en la base de datos directamente de la tarjeta Raspberry Pi o desde la aplicación de mensajería instantánea Telegram, ya sea desde un dispositivo móvil o un equipo de escritorio.

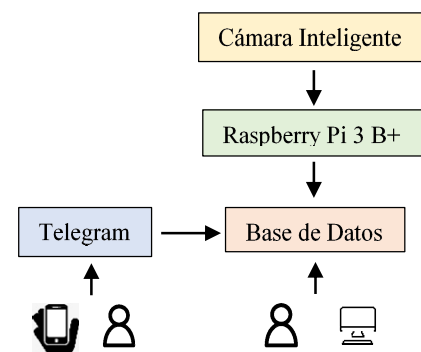


Fig. 1. Diagrama de bloques del sistema propuesto.

La cámara inteligente B5T-007001 emplea una alimentación de 5 VCD y tiene un consumo debajo de 400 mA, el elemento sensor tiene una resolución de imagen de 1600x1200 pixeles con ángulos de detección horizontal y vertical de 54° y 41° respectivamente. La interface USB emula

un puerto serie en el dispositivo host al que se conecta, de esta manera se transfieren mensajes UART entre la cámara y el anfitrión a una velocidad máxima de 921,600 baudios. A pesar de la elevada resolución de la cámara, la interface usada limita la resolución de la imagen transferida entre la cámara y el host a 320x240 pixeles.

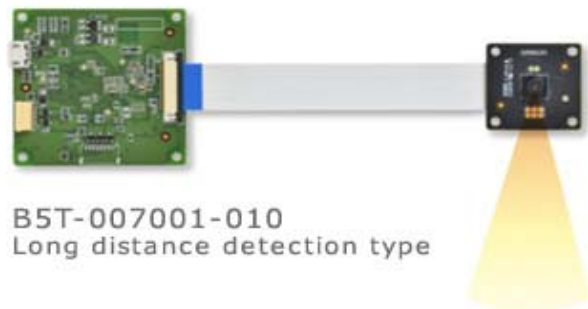


Fig. 2. Cámara inteligente Omron B5T-00701.

En la Fig. 3 se observa el diagrama de los componentes de hardware,

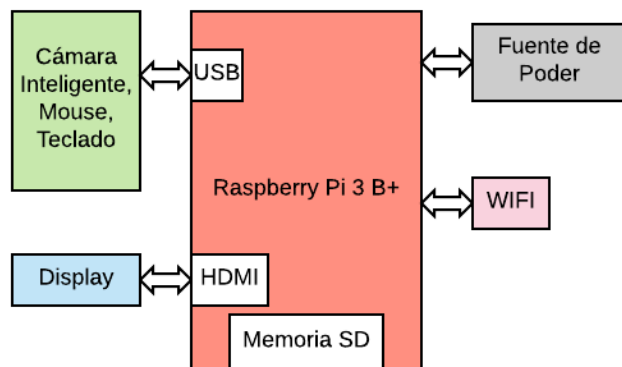


Fig. 3. Componentes de hardware de la plataforma propuesta.

El sistema funciona de la siguiente manera, al iniciar un comando en la terminal de la Raspberry comienza a procesar video desde la cámara inteligente, la cual está configurada para emitir un mensaje vía UART en el momento que detecte un rostro en la secuencia de video. Al momento de detectar algún rostro hace una captura y almacena la fotografía con la fecha y hora del momento que se detectó en la base de datos y la actualiza, por otra parte, la guarda en una carpeta dentro de la memoria de la Raspberry. De esta manera se podrán visualizar las imágenes que se han tomado en dos formas: a través de consultas a la base de datos por internet o desde la Raspberry de forma local.

A la vez que se realiza la ejecución del programa de Detección de Rostros de forma paralela se mandará iniciar en otra terminal el programa de Respuesta de Telegram, el cual estará esperando algún mensaje específico que el usuario le mande desde su teléfono inteligente para que un bot conteste de forma automática a esta petición, mostrando la información deseada ya sea las fechas y horas o la última imagen guardada en la base de datos.

IV. DISEÑO DE HARDWARE

Actualmente grandes empresas de desarrollo tecnológico han desarrollado y creado nuevas tecnologías en sensores de imagen, una de ellas es OMRON Electronic Components quienes por más de 20 años han trabajado su tecnología “OKAOTM Vision” para reconocimiento humano, logrando así uno de sus nuevos sensores de imagen humana B5T-007001 Human Vision Components (HVC-P2) [3] el cual se presenta en dos versiones la primera B5T-007001-010 que permite el reconocimiento humano a larga distancia detectando rostros hasta 10m y la versión B5T-007001-020 de ángulo amplio. En nuestro caso se utiliza la primera versión ya que cumple satisfactoriamente nuestras necesidades.

Las capacidades que tiene esta cámara son muy específicas para reconocimiento, debido a su ciencia aplicada permitiendo las siguientes funciones del sensor de imagen

- Detección del Cuerpo Humano
- Detección de Rostro
- Detección de Mano
- Estimación de Dirección del Rostro
- Estimación de Mirada
- Estimación de Pestaño
- Estimación de Edad
- Estimación de Género
- Estimación de Expresión
- Reconocimiento Facial

Por otra parte, tenemos la tarjeta de computación embebida de bajo consumo Raspberry Pi 3 B+ con el Sistema Operativo Raspbian a la cual se le deben cargar las siguientes librerías para Python, estas son necesarias para el correcto funcionamiento del script principal encargado de hacer el reconocimiento visual, guarda y carga de imágenes a la base de datos, para llevar a cabo el sistema propuesto, estas librerías son de código abierto el cual permite su modificación.

Aparte de ello las librerías mencionadas a continuación son para computación visual, generación de gráficos, análisis de datos, acceso a base de datos y construcción de aplicaciones para el Bot de Telegram.

- OpenCV
- Matplot lib
- Pandas
- MySQLdb
- Telepot

En la Fig. 4 se encuentra el sistema propuesto de forma física mostrando el correcto cumplimiento del Diagrama de componentes.



Fig. 4. Sistema Propuesto implementado físicamente

V. DISEÑO DE SOFTWARE

El funcionamiento de la aplicación se desarrolla en un script de Python de acuerdo con el diagrama de flujo mostrado en la Fig. 5, donde se describe el proceso para lograr el reconocimiento facial y proceder a la toma de fotografía y su actualización de la base de datos.

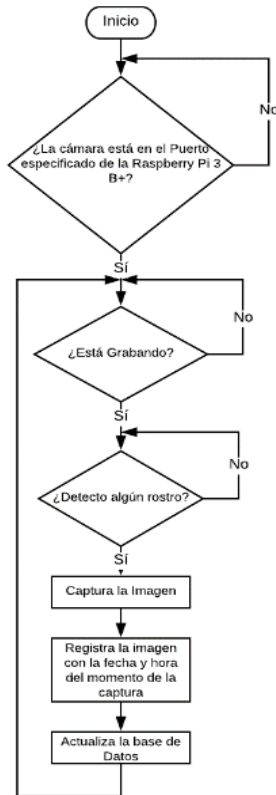


Fig. 5. Diagrama de flujo del sistema.

La comunicación entre Telegram y la base de datos se realiza gracias al *framework* para Telegram Bot API para el lenguaje Python “Telepot”, la cual ayuda a crear aplicaciones móviles para la Interfaz de Programación de Aplicaciones (API) de Telegram Bot, este es un programa informático autónomo que realiza tareas específicas basado en una interfaz HTTP (Protocolo de Transferencia de Hipertexto), por lo tanto

son aplicaciones de terceros ejecutándose dentro de la aplicación de mensajería Telegram, donde el usuario interactúa con el bot debido a su programación establecida previamente pareciera que se está enviando mensajes con otro usuario.

Para crear un bot es necesario tener la aplicación móvil de Telegram ya instalada en el celular, procediendo a enviar un mensaje al usuario de nombre “BotFather”, abriendo el chat de forma normal enviando el siguiente mensaje “/newbot” este le responderá haciendo dos preguntas, el nombre y el usuario del bot, en este caso el bot se llama ‘Pepe’, con un usuario de ‘Pepe_56bot’. Por ultimo recibirá un *Token*, este da el acceso para la API HTTP cuenta del bot, que se deberá agregar al programa de Telegram para poder enviar y recibir mensajes, haciendo la conexión entre la Raspberry y Telegram, de igual manera puede utilizarse en Telegram Web.

De la misma manera, en la Fig. 6 encontramos el diagrama de flujo de la conexión que tiene Telegram con la base de datos y el usuario.

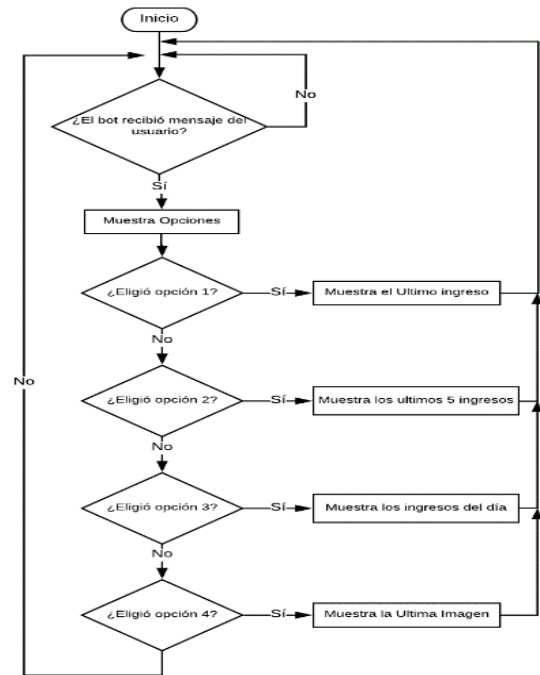


Fig. 6. Diagrama de flujo, respuesta de Telegram

VI. RESULTADOS EXPERIMENTALES

Se llevó a cabo una validación experimental para estudiar, comprender y analizar el sistema propuesto en el cual observamos las siguientes características.

Para corroborar el correcto funcionamiento de la cámara inteligente se realizaron pruebas de desempeño ante diferentes condiciones de orientación, expresión y oclusión del rostro. En las Fig. 7 y Fig. 8 se muestran ejemplos de resultados obtenidos para cambios en la orientación del rostro, oclusión parcial y cambio de expresión facial. El recuadro verde indica el área de la imagen donde se ha detectado un rostro, el cual también incluye etiquetas que indican edad, género y emoción, siendo el principal objeto de interés la función de detección de rostro. En la Fig. 9 se observa la respuesta de la cámara

cuando se presentan distintos rostros en escena, con personas de diferentes edades, género y expresiones .



Fig. 7. Diferentes posiciones del rostro

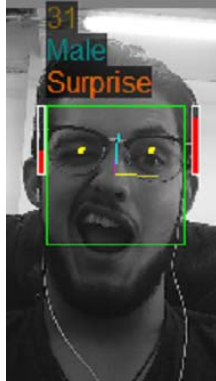


Fig. 8. Posición de un ojo más cerrado que el otro con una expresión de sorpresa

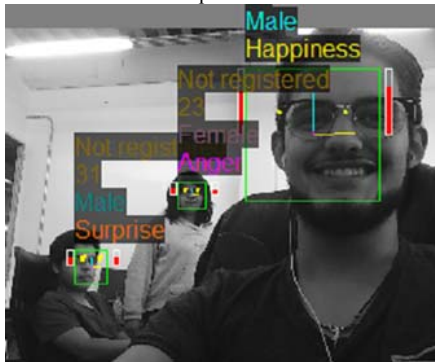


Fig. 9. Prueba con distintas personas

Primero al abrir una nueva terminal se ejecuta el siguiente comando “python guarda.py /dev/ttyACM0 921600” que hace referencia al script principal desarrollado en Python, continuando se especifican el puerto USB “ttyACM0” con una comunicación UART a 921,600 baudios, como se muestra en la Fig. 10 ya se encuentra el programa en ejecución, pero aún no detecta ningún rostro y muestra las tres opciones que la cámara permite gracias a sus características mencionadas anteriormente, así como muestra el tiempo concurrido entre un búsqueda y la otra.

```

pi@raspberrypi:~/Desktop/AHORA_SI $ python guarda.py /dev/ttyACM0 921600
==== Elapsed time:1358.8[msec] ====
Face Count = 0
Body Count = 0
Hand Count = 0

Press Ctrl+C Key to end:

==== Elapsed time:1359.6[msec] ====
Face Count = 0
Body Count = 0
Hand Count = 0

Press Ctrl+C Key to end:

==== Elapsed time:1359.7[msec] ====
Face Count = 0
Body Count = 0
Hand Count = 0

Press Ctrl+C Key to end:

```

Fig. 10. Terminal sin rostros detectados

En la Fig. 11 se observan los resultados que muestra la cámara al detectar un rostro, donde se ven los datos numéricos específicamente dirección del rostro, mirada, pestañeo, edad, así como datos de género y expresión gracias a la tecnología de “OKAO™ Vision” de OMRON.

Cada uno de estos cuenta con datos específicos bajo los siguientes parámetros que pueden consultarse en la hoja de instrucciones oficial [7], así como en el script principal de nombre ‘guarda.py’ puede cambiarse y modificarse los parámetros para hacer ajustes en los sensores de imagen

```

pi@raspberrypi: ~/Desktop/AHORA_SI
Archivo Editar Pestañas Ayuda
Press Ctrl+C Key to end:
==== Elapsed time:1802.4[msec] ====
Face Count = 1
[0] TrackingID:-1 X:915 Y:750, Size:414 Conf:555
Direction LR:-1 UD:11 Roll:-1 Conf:210
Age Age:29 Conf:666
Gender Gender:Male Conf:944
Gaze LR:14 UD:20
Blink R:183 L:188
Expression Exp:Sadness Score:56 (Neutral:18 Happiness:0 Surprise:17 Anger:19 Sadness:56 NegPos:-92)
Recognition Uid:-127 Score:-127
Body Count = 0
Hand Count = 0

!e!
commit
Press Ctrl+C Key to end:
==== Elapsed time:1593.2[msec] ====
Face Count = 1
[0] TrackingID:-1 X:908 Y:561, Size:432 Conf:585
Direction LR:4 UD:10 Roll:1 Conf:258
Age Age:27 Conf:1000
Gender Gender:Male Conf:814
Gaze LR:21 UD:12
Blink R:186 L:163
Expression Exp:Sadness Score:50 (Neutral:1 Happiness:0 Surprise:31 Anger:18 Sadness:50 NegPos:-99)
Recognition Uid:-127 Score:-127
Body Count = 1
[0] TrackingID:-1 X:840 Y:640, Size:960 Conf:737
Hand Count = 0

!e!
commit
Press Ctrl+C Key to end:

```

Fig. 11. Terminal mostrando información de un rostro detectado

En la Fig. 12 se muestra la terminal con los valores respectivos al momento de detectar un cuerpo o una mano, así como los valores detectados que como anteriormente se mencionaba se explican ampliamente en la hoja de datos de la cámara inteligente.

```

==== Elapsed time:1470.1[msec] ====
Face Count = 0
Body Count = 1
[0] TrackingID:-1 X:1440 Y:480, Size:120 Conf:681
Hand Count = 1
[0] TrackingID:-1 X:817 Y:432, Size:522 Conf:1000

Press Ctrl+C Key to end:

```

Fig. 12. Terminal mostrando información de una mano o un cuerpo detectado

Procedemos a la base de datos mostrada en la Fig. 13 que muestra los datos obtenidos guardados del más antiguo hasta el más reciente.

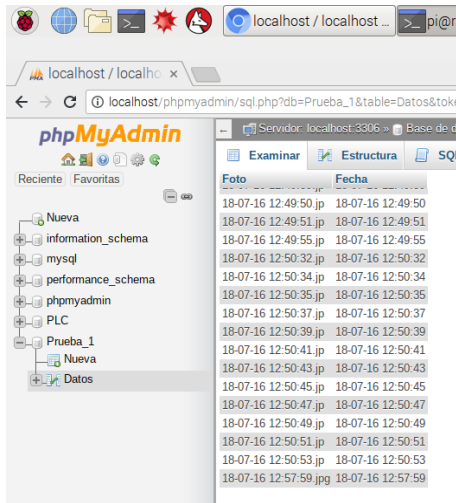


Fig. 13. Base de Datos

Como se observa en la Fig. 14 podemos ver una captura de pantalla donde se está interactuando con el bot de Telegram para obtener la información deseada, enviando los comandos clave para recibir la respuesta solicitada.

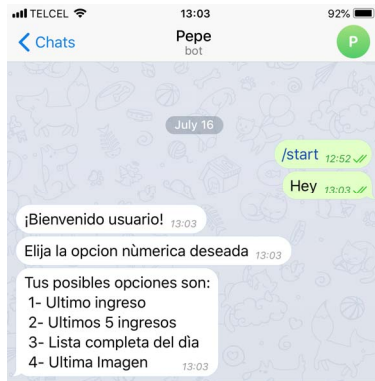


Fig. 14. Inicio de Conversación con Telegram

Por ultimo en la Fig. 15 observamos las posibles respuestas que podría obtener el usuario al enviar los números con la opción deseada

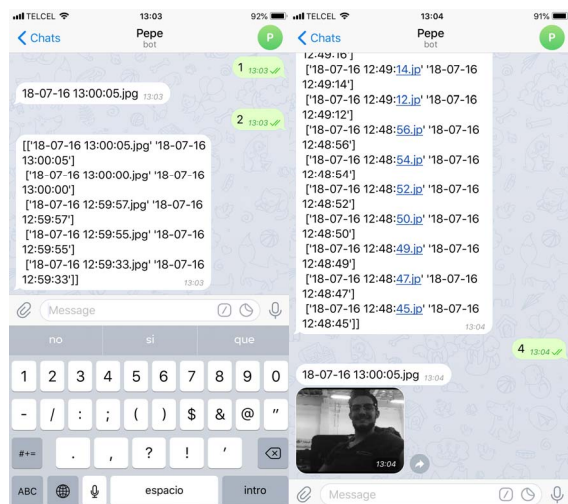


Fig. 15. Respuesta de Telegram

VII. CONCLUSIÓN Y TRABAJOS FUTUROS

Una de las habilidades del Internet de las Cosas es que día con día va evolucionando permitiendo su mejora continua y es por eso que nos otorga una gran cantidad de opciones para continuar trabajando y aportando mayor conocimiento y contribuciones a nuevas metodologías de integración de dispositivos, en este caso la Raspberry Pi junto con la cámara inteligente, así mismo el sistema propuesto es una herramienta sumamente importante debido a la automatización del proceso de reconocimiento facial, control de datos, facilidad y seguridad de manejo de datos gracias a que solo un usuario podrá conocer estos datos haciendo una búsqueda más rápida y eficaz dando reduciendo significativamente costos.

Asimismo, el sistema provee seguridad en tiempo real, debido a que el usuario autorizado podrá conocer los rostros detectados y actuar conforme la situación lo prevé. Se demuestra el correcto funcionamiento del sistema propuesto, mostrando sus capacidades y ventajas de uso, con diferentes pruebas a las que se ha sometido, obteniendo resultados satisfactorios creando un sistema de video vigilancia inteligente y eficaz con dispositivos simples y actualmente funcionales.

De la misma manera como se mencionaba anteriormente en este proyecto se planea continuar trabajando teniendo como meta la conexión de más cámaras inteligentes bajo el mismo concepto y proceso manejando una guía de seguimiento, ubicando un objetivo en una cámara y reubicarla con el resto de las cámaras haciendo un seguimiento de su camino.

VIII. ACKNOWLEDGMENT

Los autores agradecen el apoyo recibido a través del fondo mixto CONACyT-Gobierno del Estado de Querétaro (FOMIX-QRO-2016-02-279773) para la realización del presente trabajo.

El autor Vela-Medina agradece sus padres y hermana, al equipo de Investigación y Desarrollo de Software de la UAQ, por el apoyo incondicional, a la Universidad Autónoma de Querétaro y la Universidad Tecnológica de San Juan del Río por la oportunidad de ser parte de este trabajo de desarrollo.

IX. REFERENCES

- [1] D. Navani, S. Jain, and M. S. Nehra, "The Internet of Things (IoT): A Study of Architectural Elements," *2017 13th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, pp. 473–478, 2017.
- [2] A. U. Case, F. Manufacturing, K. Prince, H. Morán, D. Mcfarlane, and D. Mcfarlane, "White Paper White Paper," no. August, pp. 1–3, 2004.
- [3] L. Brown *et al.*, "IBM Smart Surveillance System (S3): An open and extensible architecture for smart video surveillance," *October*, S. Ali, I. Quadri, and P. Sathish, "IoT Based Home Automation and Surveillance System," pp. 861–866, 2017.
- [4] A. N. Ansari, M. Sedky, N. Sharma, and A. Tyagil, "An Internet of Things Approach for Motion Detection using Raspberry Pi," *Intell. Comput. Internet Things (ICIT), 2014 Int. Conf. on. 17-18 Jan. 2015. Harbin, China*, pp. 131–134, 2015.
- [5] H. Kumar, S. Bhattacharya, S. S. Thomas, S. Gupta, and K. S. Venkatesh, "Design of smart video surveillance system for indoor and outdoor scenes," *Int. Conf. Digit. Signal Process. DSP*, vol. 2017–August, pp. 1–5, 2017.
- [6] "B5T-007001_InstructionSheet_160809.pdf."