UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

COLEGIO CIENCIAS E INGENIERÍAS

Sistema de seguridad para el hogar basado en análisis de video inteligente

Tadashy Peter Curichumbi Guacho

Ingeniería en ciencias de la computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en ciencias de la computación

Quito, 21 de mayo de 2021

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

COLEGIO CIENCIAS E INGENIERÍAS

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Sistema de seguridad para el hogar basado en análisis de video inteligente

Tadashy Peter Curichumbi Guacho

Nombre del profesor, Título académico:	Ricardo Flores, Doctor in Telematic Systems Engineering Daniel Riofrío, Doctor of Philosophy in Computer Science

Quito, 21 de mayo de 2021

© DERECHOS DE AUTOR

Nombres y apellidos: Tadashy Peter Curichumbi Guacho

Código: 00137011

Cédula de identidad: 0605375237

Lugar y fecha: Quito, 21 de mayo de 2021

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

Con el desarrollo económico y mejora en la calidad de vida que nuestra realidad presenta, se evidencia una clara tendencia a la digitalización y el uso de sistemas inteligentes dentro de diversos campos. Uno de ellos es el campo de la seguridad. La seguridad del hogar se ha convertido en una de las principales preocupaciones para las familias e individuos. Es por esto que el uso de sistemas de seguridad para el hogar se ha diversificado en gran medida. Ejemplos de esto es el uso de sistemas pasivos de monitoreo y vigilancia, sistemas de seguridad biométricos o sistema de alarma de corriente eléctrica. Dichos sistemas a pesar de cumplir con su cometido presentan una serie de limitantes que hasta hoy en día han sido levemente exploradas por la mayoría de las compañías dedicadas a la seguridad del hogar. Entre estas tenemos la poca capacidad de atención continua de las personas que monitorean dichos sistemas, la falta de inteligencia de los sistemas y la falta de una respuesta activa ante cierto evento, entre otros. En este contexto surge la necesidad de generar sistemas de seguridad capaces de interpretar y actuar ante eventos que atenten contra la seguridad familiar. Por esto, el presente trabajo explora el diseño e implementación de un prototipo de sistema de seguridad para el hogar capaz de superar las limitantes previamente descritas. Este sistema será construido con la ayuda de un lenguaje de alto nivel (Python) y una serie de librerías relacionadas con visión por computador, aprendizaje automático e interfaz de usuario, entre otros. El uso de este conjunto de herramientas busca construir un prototipo funcional del sistema deseado, en el cual se realizará una prueba de concepto que permita conocer los fundamentos necesarios de este tipo de sistemas en aras de una estandarización global de esta tecnología.

Palabras clave: *Sistema de seguridad, análisis de video inteligente, reconocimiento facial, visión por computador, aprendizaje automático, interfaz de usuario, Python.*

ABSTRACT

With the economic development and improvement in the quality of life that our reality presents, there is a clear trend towards digitization and the use of intelligent systems within various fields. One of them is the field of security. Home security has become one of the top concerns for families and individuals. Therefore, the use of home security systems has greatly diversified. As examples of this, we have the use of passive monitoring and surveillance systems, biometric security systems or electric current alarm systems. These systems, despite fulfilling their objectives, they present a series of limitations that until today have been slightly explored by most companies dedicated to home security. Among these, we have the limited capacity for continuous attention of the people who monitor these systems, the lack of intelligence of the systems and the lack of an active response to a certain event, among others. In this context, the need to generate security systems capable of interpreting and acting because of certain events that threaten family security increases. Thus, the present work explores the design and implementation of a prototype of a home security system capable of overcoming the previously described limitations. This system will be built with the help of a high-level language (Python) and a series of libraries related to computer vision, machine learning and GUI. The use of this set of tools seeks to build a functional prototype of the desired system, in which a proof of concept will be carried out, this will allow to get the necessary knowledge of the foundations of this type of system to achieve a global standardization of this technology.

Keywords: *Security system, intelligent video analysis, facial recognition, computer vision, machine learning, GUI, Python.*

TABLA DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

1 INTRODUCCIÓN

1.1 Descripción del problema

En el contexto actual el campo de la seguridad familiar ha tenido un incremento en su importancia debido a múltiples acontecimientos que han afectado nuestra realidad. Ecuador en el año 2021 ocupa el lugar número 18 dentro de los países con mayor índice de criminalidad de la región, el cual es de 54,41 puntos. Este índice nacional se encuentra a tan solo unas cuantas decimas de nuestro país vecino Venezuela quien tiene un índice de 84,25 y lo coloca entre los países más inseguros a nivel mundial.

Con cada día que pasa las familias ecuatorianas se sienten más inseguras en su entorno. Esto aumenta la probabilidad de optar por el uso de un sistema digital de algún tipo, para así aumentar el nivel de seguridad de sus hogares. Ejemplos de esto son los múltiples sistemas de seguridad disponibles actualmente en el mercado. Sistemas de vigilancia y monitoreo, sistemas de seguridad biométricos o sistemas de alarma de corriente eléctrica son algunos de los ejemplos más comunes.

Todos estos sistemas a pesar de tener un evidente auge a lo largo de los años y cumplir moderadamente con su objetivo, no han optado por solventar problemáticas que pueden limitar su eficiencia. Entre estas esta la falta de concentración continua que presentan las personas que monitorean estos sistemas, la falta de inteligencia de los sistemas digitales, y la falta de una respuesta activa ante cierto tipo de eventos.

Adicionalmente, las compañías encargadas de la producción y venta de estos productos no han optado por mejorar las funcionalidades de sus sistemas con tecnologías emergentes aplicables a la seguridad o en caso de hacerlo, no han intentado abaratar los costos de producción generados en gran manera. Como ejemplo de esto tenemos la gran

variedad de precios actuales de cualquier tipo de cámara de seguridad de algunas de las empresas más populares del mercado de video vigilancia (Bosch – HikVision - Dahua Technology). Pues estos productos poseen un precio que ronda desde los $ 200 hasta los $ 1000 – $ 2000 en adelante, incluyendo o no su sistema complementario.

Es aquí en donde se evidencia una necesidad por solventar esta problemática, a la vez que se exploran los fundamentos necesarios para generar un sistema de seguridad que integre los principios del análisis de video inteligente. El cual ayude a definir las bases de un estándar que beneficie al usuario tanto al nivel económico como de funcionalidad.

1.2 Descripción del proyecto

El proyecto detallado a continuación, consiste en el diseño e implementación de un prototipo funcional de un sistema de seguridad para el hogar basado en los principios del análisis de video inteligente. El prototipo permitirá realizar una prueba de concepto dentro de un ambiente controlado para determinar el grado de precisión y eficiencia obtenido. Por otra parte, es importante mencionar que el proyecto se compone de tres módulos principales.

El primer módulo estará encargado de recolectar, almacenar y proporcionar los datos necesarios al módulo de procesamiento. Con el fin de mantener una base de datos de los rostros conocidos, almacenar los datos de los nuevos rostros definidos por el usuario y la información de los usuarios registrados en el sistema.

El segundo módulo estará encargado de procesar los datos entregados por el módulo de almacenamiento. Estos datos serán comparados con las imágenes provenientes de las cámaras instaladas con el fin de autentificar los rostros conocidos dentro de la base de datos. Esto se logrará gracias al uso de las imágenes provenientes de las cámaras instaladas. Ya que estas serán procesadas por un algoritmo de detección facial y posteriormente por un algoritmo de reconocimiento facial. Con el objetivo de autentificar al usuario y etiquetar rostros familiares o desconocidos. Una vez realizado este proceso, dependiendo del caso de uso, el sistema será capaz de emitir mensajes personalizados o alarmas de distintos tipos sobre algún posible evento.

El tercer módulo consiste en una interfaz de usuario amigable para el uso del sistema. Esta interfaz será construida en un aplicativo de escritorio implementada gracias al uso de una librería integrada. Este módulo permitirá al usuario tener una interacción más amigable con el sistema. Además de la visualización de las imágenes capturadas por las cámaras, el registro de nuevos rostros familiares y la eliminación de la información facial de un usuario.

El sistema deberá implementar sus módulos de tal forma que los mismos sean escalables para futuras actualizaciones o correcciones. El prototipo base consistirá en dos cámaras. No obstante, se considera la opción de integrar más cámaras con el objetivo de mejorar la escalabilidad. El sistema deberá ser capaz de reconocer individuos dentro del rango visual otorgado por las cámaras, identificar rostros y emitir alarmas de acuerdo con los objetivos detallados a continuación.

1.3 Objetivo general

Diseñar e implementar un prototipo funcional de un sistema de seguridad para el hogar basado en los principios del análisis de video inteligente.

1.4 Objetivos específicos

El prototipo deberá ser capaz de realizar lo siguiente:

- Detección de rostros.
- Implementar reconocimiento facial para identificar y diferenciar rostros conocidos y desconocidos.
- Ser capaz de entrenarse con nuevos rostros que se etiqueten como conocidos.
- Generar respuestas o alarmas inteligentes dependiendo del tipo de rostro detectado (Principal: Voz - Secundario: Email, WhatsApp).

1.5 Organización del documento

Este documento se organizará en distintas secciones que se acoplen a la propuesta planteada. En cada sección se abordará la temática indicada. En términos generales se realizará un análisis del estado del arte de los sistemas de seguridad que existen actualmente y las tecnologías que se emplearan dentro del sistema. Posteriormente, se definirá el diseño del prototipo y las funciones principales de los módulos que integran el sistema. Dentro de la fase de implementación, se incluye el desarrollo de los módulos de software respectivos y una breve explicación de su funcionamiento. En esta sección, se propondrá la prueba de concepto a realizar, la cual permitirá medir la eficacia del sistema desarrollado. Finalmente, el trabajo se complementará con una serie de conclusiones y recomendaciones tomando en cuenta el desarrollo del proyecto y sus implicaciones a futuro.

2 ESTADO DEL ARTE

2.1 Sistemas de seguridad

Se puede considerar a un sistema de seguridad como el conjunto de dispositivos digitales o electrónicos que trabajan de manera simultánea para evitar o reducir robos, delitos u otro tipo de elemento que atente contra la seguridad familiar o del individuo.

Los sistemas de seguridad para el hogar se pueden clasificar de muchas maneras, ya sea por su contenido o por su funcionalidad. Dentro de esta clasificación podemos encontrar los siguientes tipos de sistemas de seguridad.

Sistema de seguridad de monitoreo y video vigilancia: Este tipo de sistemas utilizan un circuito cerrado de televisión (CCTV, por sus siglas en inglés) para monitorear el área objetivo con ayuda de cámaras de seguridad y monitores de televisión. Estos sistemas son principalmente pasivos y solo se recomienda su uso de forma preventiva y en áreas con una alta concurrencia de personas.

Sistemas de seguridad biométricos: Este tipo de sistemas de seguridad utilizan el reconocimiento y autentificación de alguna característica física única y no transferible del individuo, como huellas dactilares, iris, rostro, entre otros. Este tipo de sistemas optimizan el proceso de autentificación del usuario al usar características intransferibles, al mismo tiempo que aumentan los niveles de seguridad en gran manera. Lo que los convierten en uno de los tipos de sistemas más populares actualmente en el mercado.

Sistema de alarma de corriente eléctrica: Este tipo de sistemas son los más básicos y baratos que se pueden encontrar. Estos se encargan de monitorear puntos de entrada como puertas o ventanas. Este tipo de sistemas funcionan gracias al monitoreo de un flujo de corriente continua, en donde en caso de existir una brecha de seguridad dicho flujo se

interrumpe. Esto genera la emisión de una alarma sonora o de otro tipo. Por lo general este tipo de sistemas se instalan al momento de construir una casa o edificación.

Sistema de alarma con cable: Este tipo de seguridad se basan en el uso de una conexión de teléfono fijo para transmitir señales a un centro de monitoreo de corto alcance. Este tipo de sistemas presentan una vulnerabilidad importante debido a que depende de la integridad física del sistema cableado, por lo que en la actualidad no son utilizados en gran medida.

Sistema de alarma inalámbrica: Los sistemas de seguridad inalámbricos a comparación de sus antecesores, tal como su nombre lo indica no dependen de una red cableada para funcionar. Estos en su mayoría trabajan en conjunto con un panel de control central y una serie de sensores que entregan señales para su interpretación. Las cuales desencadenan alarmas en caso de ser necesario. Entre sus principales características se encuentra la posibilidad de administración remota del sistema, así como la integración de este a otros sistemas inteligentes del hogar.

Componentes de un sistema de seguridad: Entre los componentes más comunes que forman parte de un sistema de seguridad tenemos los siguientes.

- Panel de control
- Sensores de movimiento
- Cámaras de seguridad
- Alarmas o sirenas
- Señaléticas

Es importante resaltar que, independientemente del tipo de sistema de seguridad que se opte por usar, estos presentan ciertas limitantes comunes y específicas. Entre las más comunes podemos encontrar:

- **Capacidad continua de monitoreo:** Esta limitante hace referencia a la capacidad de atención continua que una persona encargada del monitoreo del sistema posee. Esta capacidad de monitoreo disminuye a medida que pasa el tiempo, ya que se conoce que por lo general una persona que ve una pantalla o monitor de forma pasiva alrededor de 20 minutos o más presenta una caída del 30% en su capacidad de atención, la cual puede disminuir drásticamente con respecto al paso del tiempo, hasta llegar a obtener pérdidas del 60 o 70 % del total de su capacidad (Landa et al., 2017).
- **Falta de inteligencia de los sistemas y falta de una respuesta activa:** Esta limitante se relaciona a los sistemas comunes de vigilancia y seguridad existentes que no cubren la necesidad creciente de una sociedad en desarrollo. Esta se presenta debido a que normalmente dichos sistemas tienen un rol pasivo es decir están basados en el uso manual de los mismos y solo están encargados de vigilar y monitorear el entorno que los rodea. Por consecuencia, no pueden actuar o notificar ante un posible riesgo en tiempo real para así tomar un rol activo en pro de aumentar la seguridad del hogar.

2.2 Visión por computador

Se conoce como visión por computador al campo de la inteligencia artificial que se encarga de entrenar computadoras para interpretar y comprender el mundo visual. En donde por medio del uso de imágenes digitales provenientes de cámaras, videos, modelos de aprendizaje se logra que una computadora pueda identificar, clasificar y reaccionar frente a objetos o situaciones del mundo visual (Szeliski, 2010).

2.2.1 Detección facial

La detección facial es la tecnología existente capaz de identificar la presencia de rostros de individuos dentro de imágenes o videos digitales. Por medio del aprendizaje automático se logra determinar si existe uno o más rostros dentro de la imagen o video analizado sin importar a quien pertenezca el mismo. Pues lo más importante dentro de este proceso es determinar la cantidad de rostros presentes, los cuales son almacenados en una base de datos de búsqueda o almacenamiento (Cabezón, 2008)

Por lo general, un algoritmo de detección facial trabaja en la búsqueda de patrones faciales, es decir el algoritmo permanece en constante búsqueda de estructuras comunes que conformen un rostro. Entre las más usadas tenemos los ojos, cejas, boca, nariz e iris. De igual manera el algoritmo analiza posiciones relativas de dichos elementos, como la distancia entre los ojos y la nariz, el tamaño y forma de la cara o la temperatura de color de la piel.

2.2.2 Clasificador en cascada

La detección facial dentro del prototipo se realizará mediante el uso de un clasificador en cascada, el cual esta implementado dentro de la librería OpenCV. Este método de clasificación fue propuesto originalmente por Paul Viola y Michael Jones en su artículo "*Rapid Object Detection using a Boosted Cascade of Simple Features*" en el año 2001.

Este método está basado en el aprendizaje automático. El clasificador se entrena con una gran cantidad de imágenes positivas y negativas para posteriormente utilizarlo en la detección de objetos. Al utilizar este clasificador en la detección de rostros este necesita una gran cantidad de imágenes positivas (imágenes de rostros) e imágenes negativas (imágenes sin rostros) para su entrenamiento y luego extraer características del clasificador.

Figura 1 Ejemplos imágenes positivas

(Stephens, J. (2005). Face Detection with a Sliding Window)

Figura 2 Ejemplos imágenes negativas

(Stephens, J. (2005). Face Detection with a Sliding Window)

El proceso de detección de rostros se da por el uso de funciones de Haar como las que se muestran en la figura 3. Estas funciones también denominadas características actúan como un núcleo convolucional. Estas se dividen en horizontales, verticales y diagonales. Cada una de estas características es un valor único resultado de la resta de la sumatoria de píxeles debajo del rectángulo blanco con la suma de píxeles debajo del rectángulo negro.

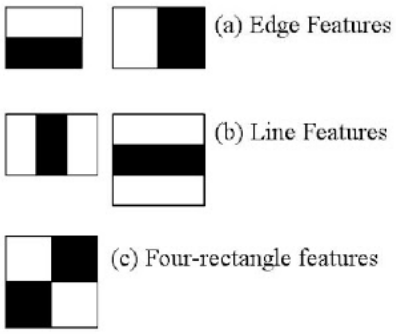

Figura 3 Funciones Haar

Doxygen. (2021). Cascade Classifier

El uso de estas características involucra una gran necesidad de tiempo de cálculo para cada uno de los píxeles presentes en una imagen. Al analizar el ejemplo propuesto por la autora Viola Jones se presenta lo siguiente. Dentro de una imagen con un tamaño de 24 x 24 píxeles, obtenemos un total de mas de 16 millones de características disponibles para analizar. Para cada una de ellas es necesario encontrar la sumatoria de píxeles debajo de cada rectángulo, lo que convierte a este proceso en uno que requiere de una gran cantidad de tiempo de procesamiento. Para reducir este tiempo, la autora introduce el concepto de imagen integral. La cual especifica que no importa que tan grande sea la imagen por analizar, los cálculos por píxeles se reducen considerablemente al utilizar solo cuatro píxeles dentro del cálculo de las sumatorias.

Otro punto a tomar en cuenta en el análisis, es que gran parte de las características calculadas no tienen una relevancia significante. Considerando la siguiente figura como ejemplo, se puede observar que solo se pueden apreciar dos buenas características. La primera se centra en la propiedad de que la región de los ojos tiende a ser mas oscura que la región de la nariz y las mejillas. Mientas que las segunda se centra en que la propiedad de los ojos es mas oscura que el puente de la nariz.

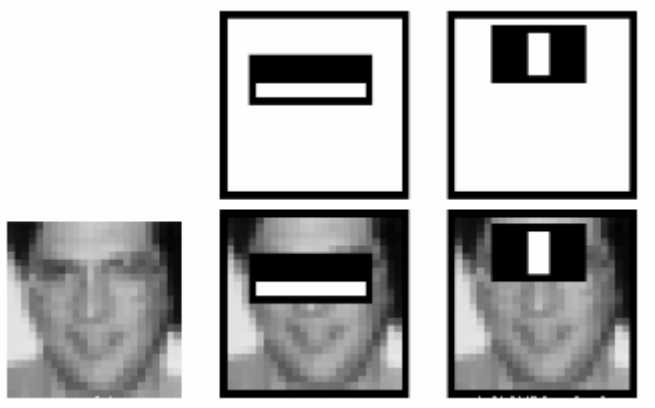

Figura 4 Selección de características

Doxygen. (2021). Cascade Classifier

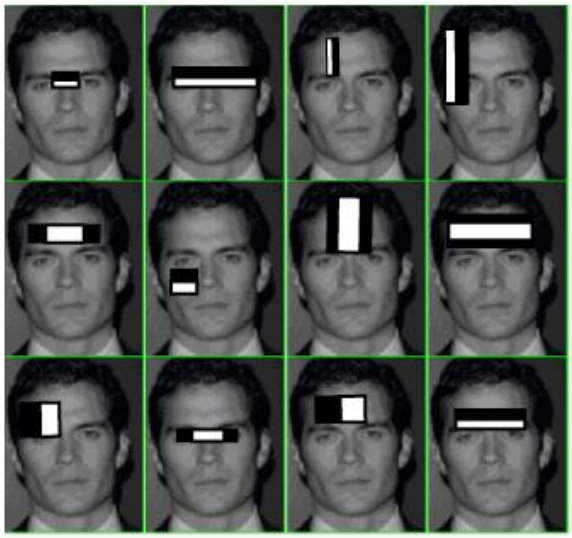

Figura 5 Selección de características Ex. 2

Javatpoint, (s.f). Face recognition and Face detection using the OpenCV

Para reducir el número de características no relevantes, la autora detalla que se deben aplicar todas y cada una de las funciones de Haar calculadas en todas las imágenes del set de entrenamiento. Gracias a este proceso, se encuentra el mejor umbral en el que una característica clasificará los rostros en muestras positivas y negativas, tomando en cuenta el porcentaje de error que esta clasificación involucra por medio del uso de una tasa de error mínimo. Es decir, dentro de este proceso a cada imagen se le asigna un peso común al principio de la clasificación, con cada iteración estos pesos se incrementan para las imágenes mal clasificadas. Posteriormente se recalculan nuevas tasas de error para cada una de las características y se le asigna el nuevo peso correspondiente. Este proceso culmina cuando se llegue al umbral de la tasa de error definida. Lo que significa que las características seleccionadas después de este proceso son aquellas que mejor clasifican imágenes que contengan patrones faciales y no faciales, mientras que las que no son seleccionadas son aquellas que no lo hacen.

Con esto, a pesar de obtener una reducción considerable del número de características calculadas, la autora determina que el proceso de detección aún presenta cálculos innecesarios. Con el ejemplo previo, tras realizar la técnica de reducción descrita pasamos un total de más de 16 millones de características a un total de 6000. Aun así, la autora considera que procesar 6000 características en cada una de las imágenes de entrada no es completamente óptimo. Para solucionar esto, se introduce el concepto de clasificadores en cascada.

Este tipo de clasificador permitirá verificar si se está analizando una región facial o no dentro de una imagen por medio del uso de etapas, con el fin de desechar automáticamente las regiones no faciales dentro de las imágenes de entrada y así optimizar el tiempo de procesamiento para evaluar nuevas posibles regiones faciales.

Para este proceso, se agrupa el total de características en grupos más pequeños o etapas, en donde cada grupo posee un número inferior al total de características disponibles. Estos grupos o etapas se aplicarán secuencialmente uno por uno en cada imagen. En caso de que una etapa falle desde el inicio, se procede a descartar la imagen procesada pues esta posiblemente no posea regiones faciales. En caso de que una etapa concluya exitosamente, la siguiente etapa realiza el mismo proceso con dicha imagen y así sucesivamente hasta concluir con todas las imágenes disponibles. Gracias a esto, el clasificador es capaz de determinar que aquellas muestras que hayan concluido con todas las etapas de manera exitosa son muestras que poseen muchas regiones faciales, por ende, dichas muestras contienen uno o más rostros.

2.2.3 Reconocimiento facial

El reconocimiento facial es un método para identificar o verificar la identidad de un individuo usando su rostro e información facial. Los sistemas de reconocimiento facial pueden ser usados para identificar personas en videos, fotos o en sistemas de tiempo real (THALES, 2021).

En términos generales, los sistemas de reconocimiento facial y su proceso de reconocimiento se dividen en cuatro etapas principales. La primera, denominada detección facial, es aquella donde la cámara o el medio utilizado para la captura de imagen o video detecta y localiza la imagen, contorno u forma de un rostro, que puede ser único o múltiple. La segunda, denominado análisis facial, es la etapa que se encarga de analizar la imagen previamente capturada utilizando tecnología 2D o 3D. El algoritmo empleado analiza la geometría del rostro, así como rasgos característicos presentes tales como distancia entre ojos, tamaño de labios, orejas, nariz, contorno facial, entre otros. Los cuales servirán

posteriormente para autentificar al rostro y compararlo con los presentes en algún medio de almacenamiento disponible. La tercera etapa es aquella que se encarga de transformar la información analógica obtenida en información digital basándose en los rasgos faciales de la persona objetivo. Este set de información es único debido a que cada persona posee rasgos únicos que lo identifican. La cuarta y última etapa se encarga de encontrar una coincidencia de la información facial de un individuo con toda la información disponible en algún medio de almacenamiento o transmisión tales como base de datos, transmisión de video, imágenes entre otros.

2.2.4 Patrones binarios locales

El proceso de reconocimiento facial dentro del sistema se realizará haciendo uso de un clasificador de patrones binarios locales implementado en la librería OpenCV. Los patrones binarios locales (LBP, por sus siglas en inglés) son un descriptor de texturas invariante simple popularizados por *Ojala et al.* en su trabajo "*Multiresolution Grayscale and Rotation Invariant Texture Classification with Local Binary Patterns*" en el año 2002. En términos generales este algoritmo calcula una representación local de una textura. Esta representación se construye comparando cada píxel con sus píxeles vecinos más cercanos y asignándoles un valor binario con respecto al valor del píxel central. Esta construcción se rige bajo una serie de pasos descritos por el autor, los cuales se describen a continuación.

El primer paso para construir el descriptor LBP es convertir la imagen a escala de grises. Posterior a esto, se escoge un píxel de la imagen como píxel central. Luego, se procede a seleccionar una vecindad de tamaño *r* que rodee al píxel central. El valor LBP es calculado para el píxel central escogido y este se almacena en un matriz 2D que tenga la misma altura y ancho que la imagen de entrada.

Por ejemplo, al analizar el descriptor LBP originalmente propuesto por *Ojala et al.*, el cual opera con un valor de vecindad *r* igual a 8, podemos observar que obtenemos una matriz de píxeles de dimensión $3x3$ como la que se muestra a continuación.

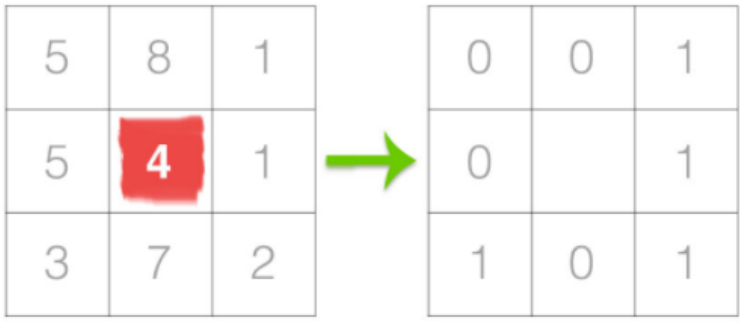

Figura 6 Matriz de píxeles otorgada por el descriptor LBP

(Rosebrock, A. (2015). Local Binary Patterns with Python & OpenCV)

Dentro de este descriptor, escogemos un píxel al que denotamos como píxel central (resaltado en rojo) y lo comparamos con su vecindad de píxeles *r* para así realizar el proceso de creación del umbral necesario. Es decir, si el valor del píxel central es menor a su píxel vecino se procede a otorgarle un valor de 0, mientras que si el valor del píxel central es mayor o igual al valor de su píxel vecino se le da un valor de 1.

En este descriptor, el cual tiene un valor de *r* igual a 8, obtenemos un total de 2^8 combinaciones posibles de valores LBP (*256*). Posterior a esto, se calcula el valor LBP para el píxel central escogido. Para esto, seleccionamos cualquiera de los píxeles vecinos disponibles y el sentido horario deseado para trabajar, el cual puede ser horario o antihorario. Es importante recalcar, que el sentido escogido para trabajar debe ser constante en todos los píxeles de la imagen que es parte del proceso, así como en todas las imágenes disponibles de la muestra de datos.

En el descriptor previo, se elige el sentido horario para trabajar e inmediatamente se aplica una prueba binaria en cada uno de los 8 píxeles vecinos del píxel central. Cada uno de estos resultados se almacena en un arreglo de 8 bits. Posteriormente, se convierte este arreglo en un valor decimal, este resultado corresponde al valor LBP calculado para el píxel central previamente escogido. Dicho valor LBP se almacena dentro de una matriz con la misma dimensionalidad que la imagen de entrada, tal como se detalla en las siguientes figuras.

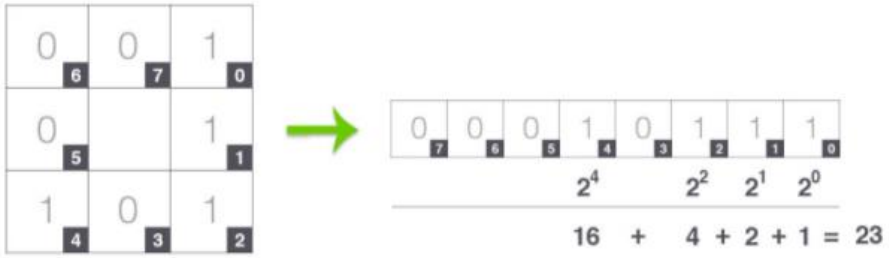

Figura 7 Cálculo del valor LBP

(Rosebrock, A. (2015). Local Binary Patterns with Python & OpenCV

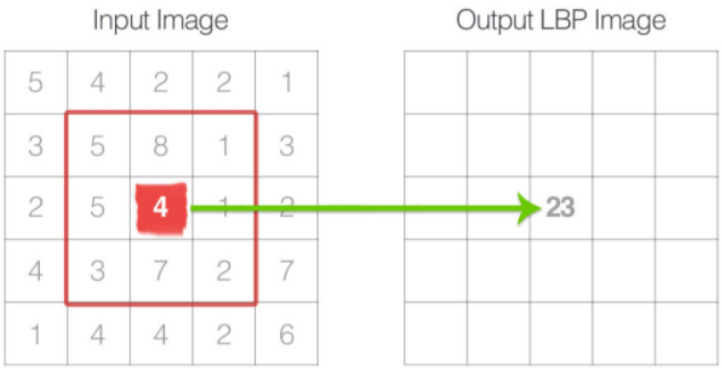

Figura 8 Almacenamiento del valor LBP del píxel central

(Rosebrock, A. (2015). Local Binary Patterns with Python & OpenCV

El proceso previo se realiza con cada uno de los píxeles presentes en la imagen de entrada, hasta concluir con las iteraciones y obtener una matriz de valores LBP calculados. Estos valores LBP son aquellos que describen de mejor manera las características de la imagen de entrada, tal como se puede visualizar en la siguiente figura.

Figura 9 Comparación imagen entrada e imagen LBP

(Rosebrock, A. (2015). Local Binary Patterns with Python & OpenCV

Finalmente, haciendo uso de la imagen LBP creada a partir de la matriz previamente calculada, se procede a dividir la imagen LBP en cuadriculas horizontales y verticales, de las cuales se extraerá un histograma de cada una de las regiones o cuadriculas disponibles. En el caso de nuestro descriptor de ejemplo, al tener 256 combinaciones posibles, su matriz posee un rango de valores que va desde [0-255], esto nos permite construir un histograma que contenga 256 códigos LBP. Este proceso se realiza sucesivamente con cada histograma disponible, los cuales se concatenarán para formar un nuevo histograma final, el cual podremos seleccionar como nuestro vector de características final, culminando así con el proceso de entrenamiento.

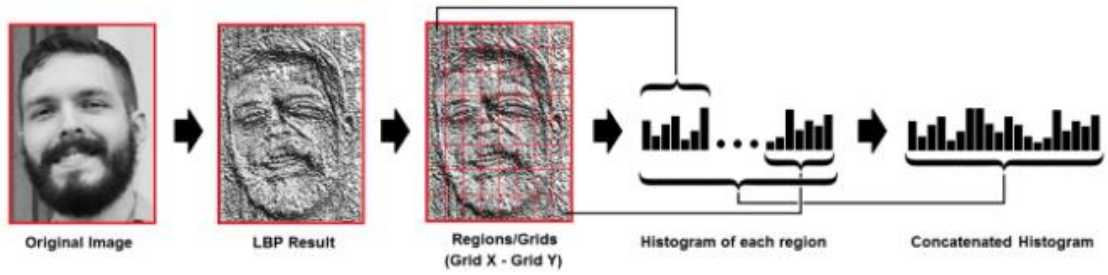

Figura 10 Formación del histograma final

Salton do Prado, K. (2017). Face Recognition: Understanding LBPH Algorithm

Luego del entrenamiento del clasificador, se procede a utilizar este para el proceso de reconocimiento facial. El cual se rige bajo el siguiente proceso. Al ingresar una nueva imagen de entrada en el sistema, repetimos el proceso del patrón binario local para esta. Lo que resulta en la creación de un histograma que representa a esta imagen. Y dado que existe un histograma para representar cada imagen del set de datos de entrenamiento, se procede a evaluar una posible coincidencia de rostros. Para este proceso, se comparan los dos histogramas disponibles, el algoritmo tras todas las iteraciones disponibles devuelve el histograma más cercano al histograma que representa la imagen de prueba ingresada. Las comparaciones de los histogramas se pueden realizar por diferentes cálculos de distancias, entre los más comunes tenemos: distancia euclidiana, chi cuadrado, valor absoluto entre otros.

Tras este cálculo, el clasificador devuelve el ID del histograma más cercano, Este a su vez devuelve la distancia calculada entre los histogramas, la cual puede ser usada como medida de confianza de la predicción hecha por el clasificador. Es importante resaltar que a pesar del término “confianza” que se utiliza, mientras este valor sea menor esto representa

que la predicción realizada por el clasificador es mejor, ya que este valor representa que el ID devuelto pertenece al histograma más cercano al histograma de la imagen de prueba.

2.2.5 OpenCV

OpenCV es una biblioteca multiplataforma de visión artificial desarrollada por Intel. La cual se centra en el procesamiento de imágenes, captura y análisis de video. Actualmente, forma parte de las librerías más conocidas y completas dentro de su campo de aplicación. Entre las funciones y aplicaciones más importantes de esta librería se encuentra el reconocimiento de gestos, reconocimiento facial, comprensión de movimientos, reconocimiento y seguimiento de objetos, integración de librerías IPP de Intel las cuales contienen funciones optimizadas para la comprensión de datos sin perdidas, entre muchas opciones más (Tutorialspoint.com, 2021).

2.3 Aprendizaje automático

El aprendizaje automático se define como la rama de la inteligencia artificial que se encarga del estudio y creación de software que tenga la capacidad de aprender y mejorar con base en la experiencia y tiempo sin estar programados para hacerlo (Expert.ai Team, 2020).

En términos generales, el proceso de aprendizaje automático se divide en cuatro etapas principales. La primera es la encargada de seleccionar y preparar la información necesaria para el aprendizaje la cual por lo general se divide en set de entrenamiento y set de prueba. Esta información está compuesta por una serie de data sets, los cuales son representativos de la información que se busca que el modelo aprenda. Estos data sets pueden ser etiquetados o no dependiendo del resultado que se busca obtener.

En la segunda etapa se escoge el algoritmo que el modelo o clasificador utilizara en el proceso de aprendizaje. El tipo de algoritmo que se va a usar dependerá del tipo de data set que se tiene disponible y de la eficacia y precisión que se busca en los resultados. Entre los algoritmos más comunes de aprendizaje automático tenemos: Algoritmos de regresión, arboles de decisión, algoritmos basados en instancias, algoritmos de agrupamiento, algoritmos de asociación, redes neuronales, entre otros.

La tercera etapa es la encargada de la creación y entrenamiento del modelo o clasificador seleccionado. El modelo se encarga de procesar el set de entrenamiento con el algoritmo seleccionado y comparar los resultados obtenidos con los datos presentes en el set de prueba con el fin de ajustar hiper parámetros del modelo o clasificador. Los cuales permitan mejorar los resultados y eficacia de estos con respecto al tiempo.

Finalmente, en la cuarta etapa, se busca poner a prueba el modelo construido con nuevos data sets. Esta etapa permite corregir errores existentes en el modelo. El cual mejora constantemente en base a su experiencia, tiempo de aprendizaje y cantidad de información procesada.

3 ANÁLISIS Y DISEÑO

3.1 Diagrama referencial de la arquitectura del sistema

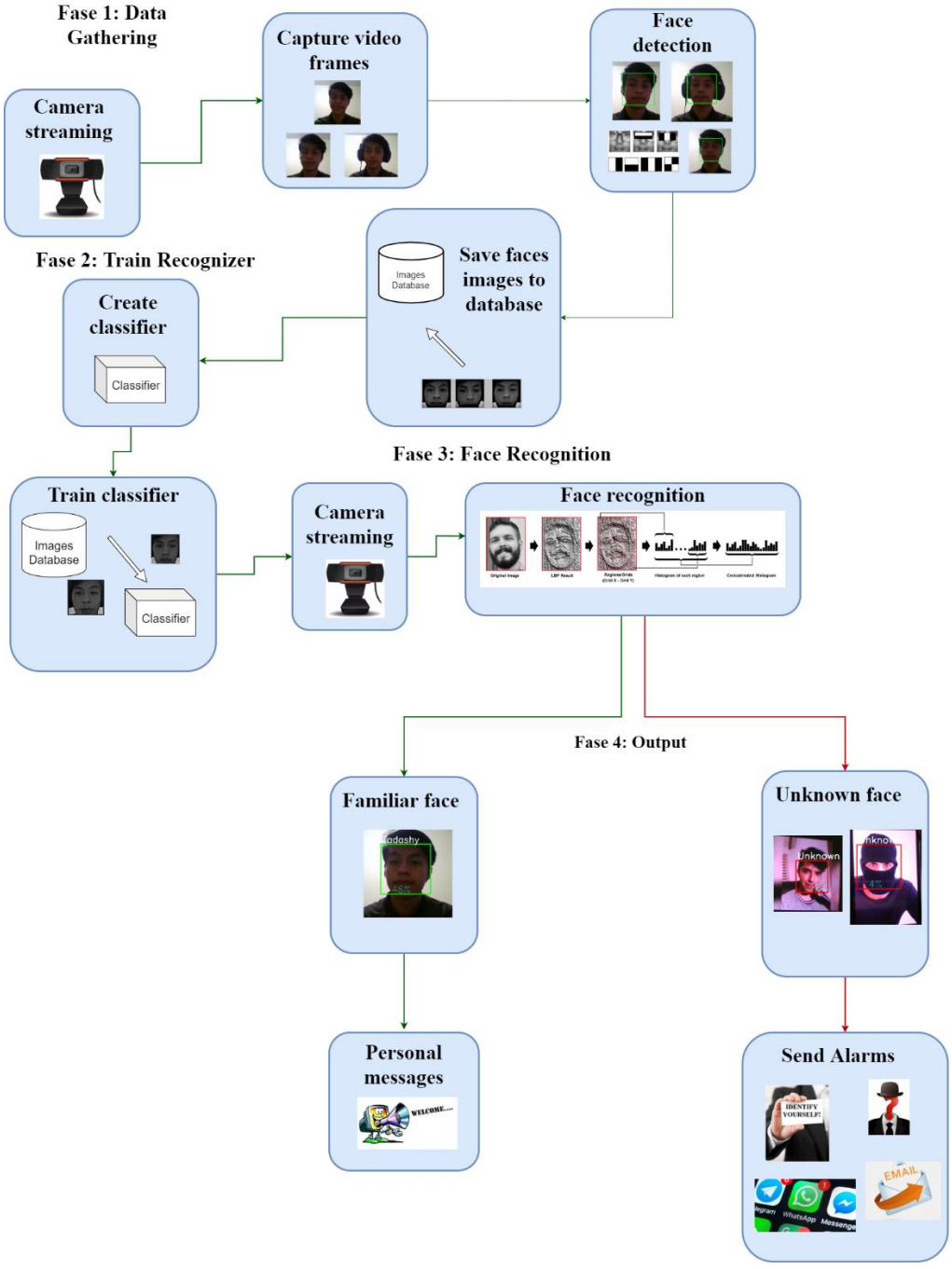

Figura 11 Arquitectura del prototipo

3.2 Módulo de almacenamiento

El módulo de almacenamiento está encargado de recolectar, almacenar y proporcionar los datos necesarios al módulo de procesamiento. Es decir, mantener una base de datos de los rostros conocidos, almacenar los datos de los nuevos rostros definidos por el usuario y la información de los usuarios registrados en el sistema.

3.2.1 Funciones Principales

- Almacenar la base de datos de los rostros conocidos.
- Almacenar nuevos rostros definidos por el usuario.
- Almacenar información adicional de los usuarios.

3.3 Módulo de procesamiento

El módulo de procesamiento es el encargado de procesar los datos entregados por el módulo de almacenamiento. Estos datos serán comparados con las imágenes provenientes de las cámaras instaladas con el fin de autentificar los rostros conocidos dentro de la base de datos como conocido o desconocido. A su vez, dependiendo del caso de uso, el sistema será capaz de emitir mensajes personalizados o alarmas de distintos tipos sobre algún posible delito.

3.3.1 Funciones principales

- Procesar mediante reconocimiento facial las imágenes provenientes de las cámaras de seguridad.
- Autentificar los rostros como conocido o desconocido, comparándolos con los rostros conocidos definidos en la base.

- Emitir mensajes de voz personalizados, mensajes de WhatsApp y correo electrónico al propietario del sistema de seguridad.

3.4 Interfaz de usuario

El módulo de la interfaz de usuario será construido en un aplicativo de escritorio. Y estará encargado de permitir al usuario tener una interacción más amigable con el sistema. Visualizar imágenes capturadas por las cámaras, registrar nuevos rostros familiares, y eliminar información de facial un usuario, entre otros.

3.4.1 Funciones principales

- Interacción del usuario con el sistema.
- Visualización de imágenes.
- Registro de nuevos rostros familiares.
- Eliminación de información de un usuario.

3.4.2 Casos de uso del sistema

Número # 1:

Title: Menú de inicio

Basic Course:

El sistema le presenta el menú de inicio. El usuario puede escoger una de las dos opciones disponibles, iniciar sesión o crear un nuevo usuario. El sistema procede a desplegarle la página seleccionada al usuario para continuar con el flujo del programa principal.

Alternate Course:

El usuario no escoge una de las opciones disponibles para iniciar el sistema, por ende, este no continua el flujo del programa e imposibilita su uso.

Diagram:

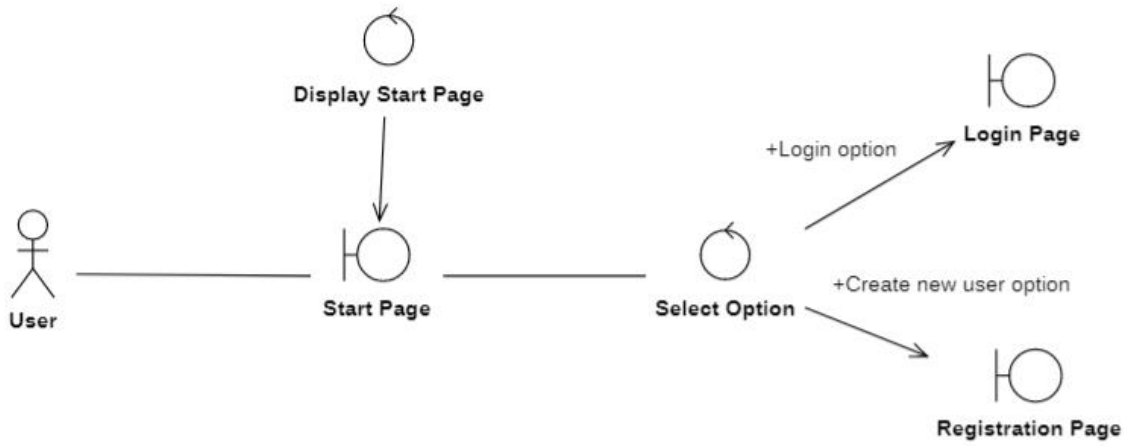

Figura 12 Diagrama menú de inicio

Número # 2:

Title: Creación de usuario

Basic Course:

El sistema le presenta al usuario la página de registro. El usuario procede a escribir sus credenciales en los campos requeridos, luego presiona el botón de agregar usuario. El sistema guarda y almacena al nuevo usuario en la base de datos y procede a redirigir al usuario al menú principal.

Alternate Course:

El sistema no puede validar la información de los campos requeridos ingresados por el usuario o estos se encuentran en blanco, en consecuencia, el sistema procede a desplegar el dialogo de error correspondiente.

Diagram:

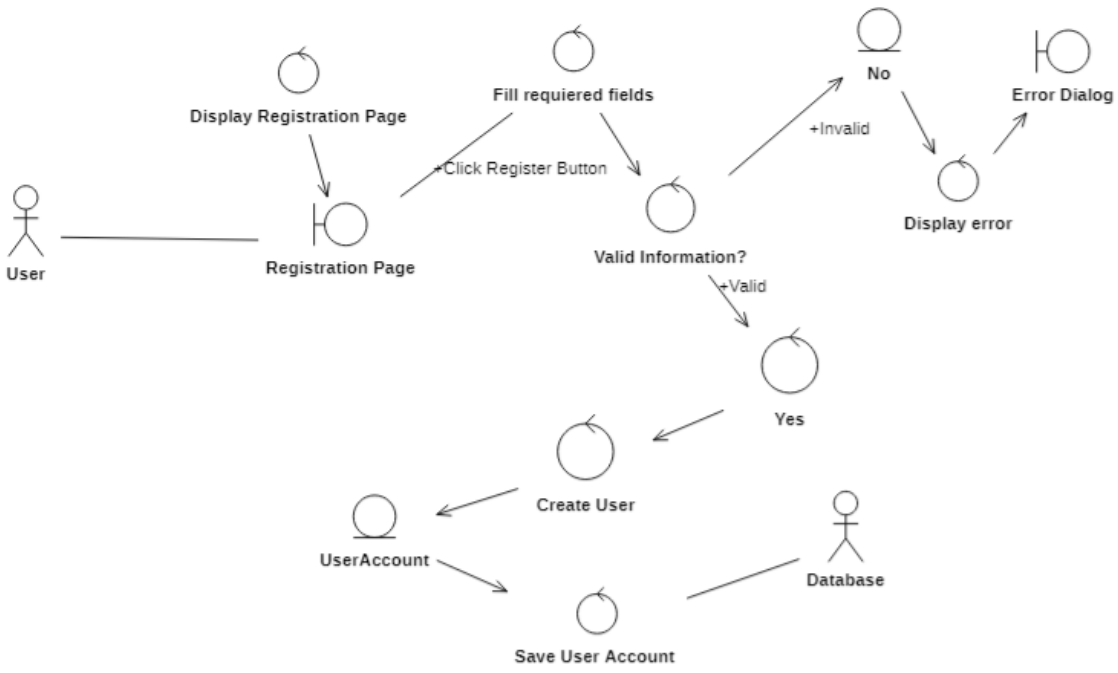

Figura 13 Diagrama creación de usuario

Número # 3:

Title: Iniciar sesión en el sistema

Basic Course:

El usuario accede a la página de inicio de sesión del sistema, procede a escribir sus credenciales en los campos desplegados y presiona el botón de Log In. El sistema verifica la información de la cuenta con la información existente de la base de datos. Una vez validado los datos del usuario y si estos son correctos el sistema redirigirá al usuario al menú principal.

Alternate Course:

El sistema desplegara el dialogo de error correspondiente en caso de que este no pueda verificar la información del usuario con la base de datos, o alguno de los campos se encuentre vacíos.

Diagram:

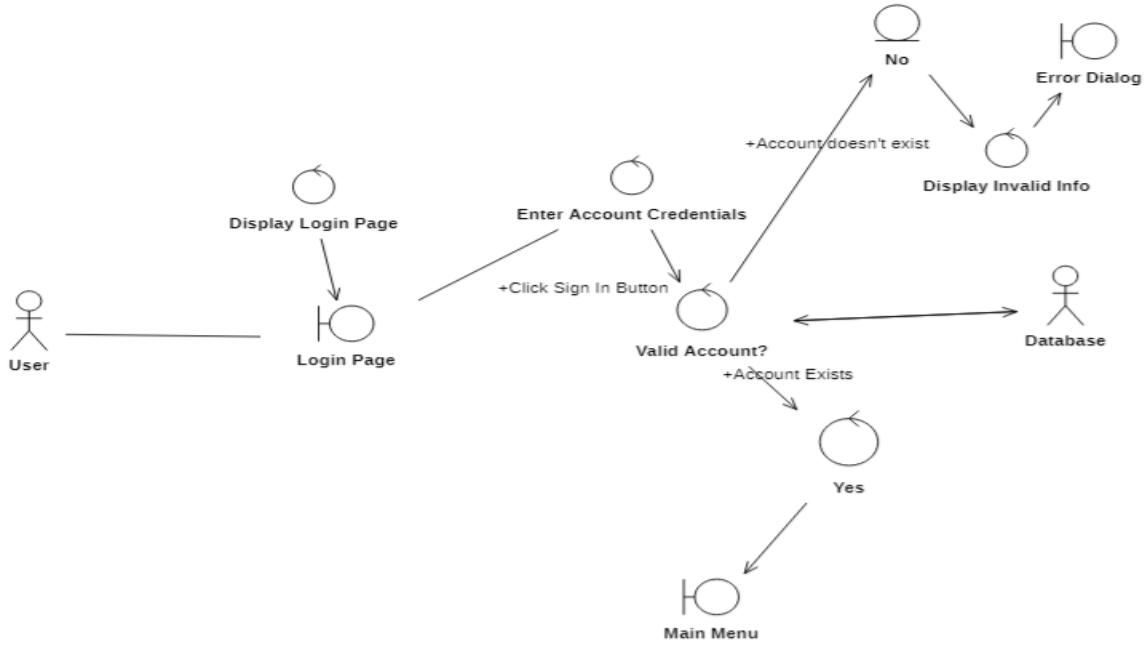

Figura 14 Diagrama inicio de sesión

Número # 4:

Title: Live stream de cámaras

Basic Course:

El sistema presenta al usuario las imágenes provenientes de las cámaras de seguridad en tiempo real y procede a detectar los rostros presentes en el video de vigilancia para predecir si se trata de un rostro familiar o desconocido y emitir la alerta o saludo correspondiente.

Alternate Course:

El sistema no puede desplegar las imágenes provenientes de las cámaras de seguridad por consecuente muestra al usuario el mensaje de error correspondiente, y lo redirige al menú principal.

Diagram:

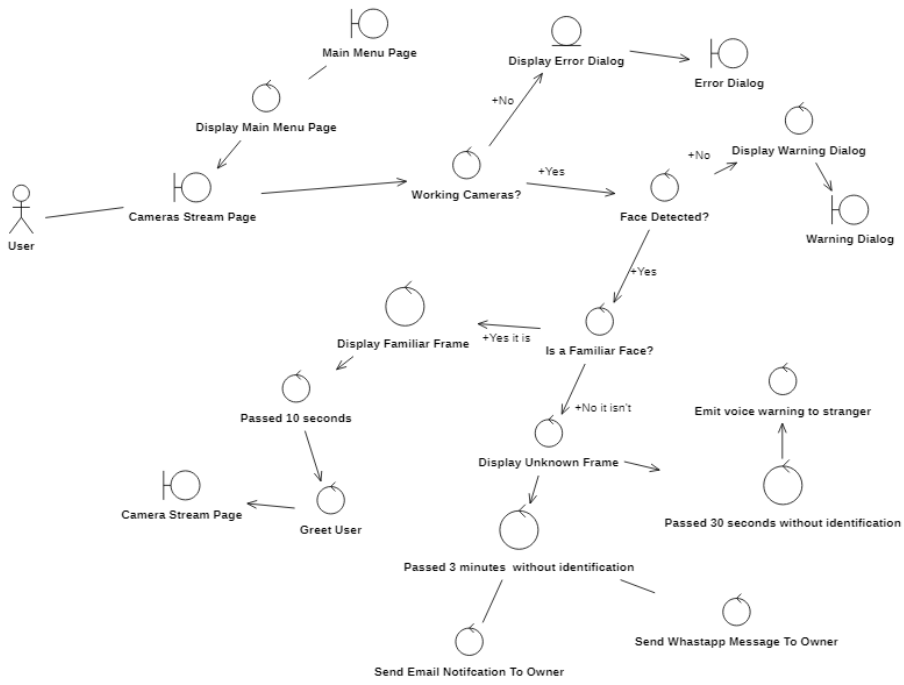

Figura 15 Diagrama cameras stream

Número # 5:

Title: Entrenar clasificador

Basic Course:

El usuario procede a presionar el botón para entrenar el clasificador. El sistema verifica si existe información de un usuario como mínimo. El sistema procede a entrenar al clasificador con la información disponible. El sistema despliega diálogos de confirmación para notificar al usuario.

Alternate Course:

El sistema no logra detectar información registrada de ningún usuario por lo que despliega el mensaje de error correspondiente y cancela el proceso de entrenamiento del clasificador.

Diagram:

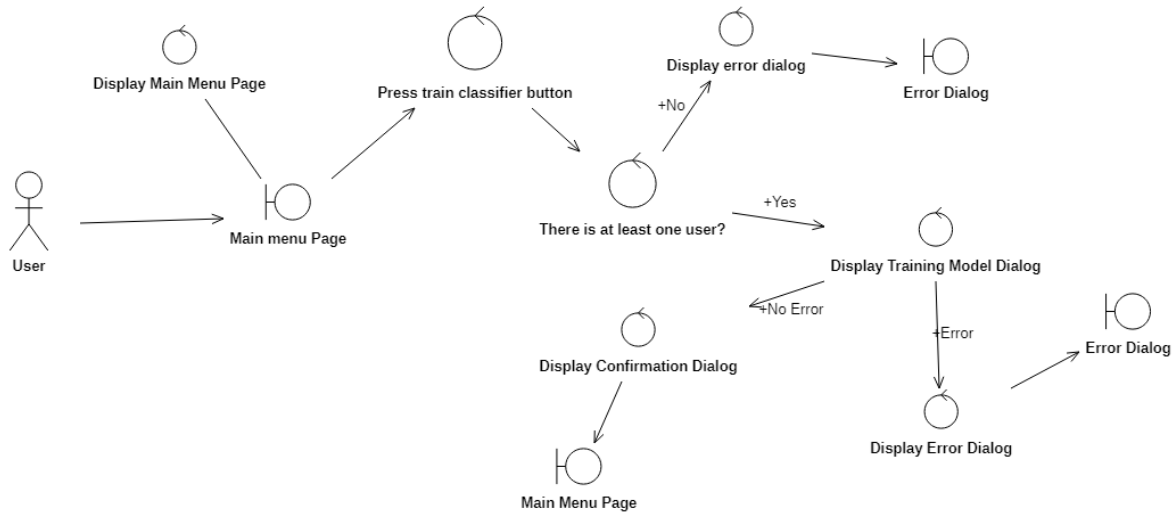

Figura 16 Diagrama entrenamiento

Número # 6:

Title: Agregar nuevo rostro conocido

Basic Course:

El sistema le indica al usuario la imagen proveniente de una de las cámaras de seguridad. El sistema procede a detectar el rostro del usuario que tiene al frente de él y enmarcarlo dentro de un recuadro visible. El sistema le pide al usuario interactuar con el fin de capturar una serie de imágenes del rostro del individuo, las cuales serán enviados al módulo de almacenamiento para su procesamiento.

Alternate Course:

El sistema no logra detectar el rostro del individuo que desea ser añadido, por lo que muestra el mensaje de error correspondiente para notificar al usuario y darle recomendaciones con el fin de lograr el reconocimiento del rostro.

Diagram:

Figura 17 Diagrama agregar nuevo rostro conocido

Número # 7:

Title: Eliminar rostro conocido

Basic Course:

El sistema despliega para el usuario todos los nombres de los rostros conocidos existentes. El usuario selecciona al individuo del cual desea eliminar la información almacenada dentro del sistema. Tras una selección correcta, el sistema desplegará un mensaje de confirmación al usuario. Posteriormente, tras recibir una confirmación por parte del usuario, el sistema procederá a eliminar la información del individuo seleccionado.

Alternate Course:

El sistema presenta un error al desplegar los nombres de los rostros conocidos, por lo que despliega el mensaje de error correspondiente al usuario. En caso de que el usuario no desee eliminar la información de un rostro almacenado, el sistema redirigirá al usuario hacia el menú principal.

Diagram:

Figura 18 Diagrama eliminación rostro conocido

4 IMPLEMENTACIÓN Y FUNCIONAMIENTO

4.1 Implementación

La implementación del prototipo del sistema se basó en los objetivos planteados en el trabajo y en el esquema de la arquitectura definido. Dentro de esta sección, se detalla los apartados y funciones más importantes de los módulos que integran el sistema, así como una descripción de las herramientas y librerías utilizadas.

4.1.1 Módulo de almacenamiento

Para la implementación del módulo de almacenamiento, se utilizó la herramienta MySQL Workbench. Con este software se implementó la base de datos necesaria para el prototipo. Esta presenta dos tablas principales, la primera encargada de almacenar la información de los usuarios registrados dentro del sistema y la segunda encargada de almacenar la información de las imágenes capturadas por la trasmisión de video. La base de datos utiliza una conexión con el módulo de procesamiento por medio del controlador de conexión *mysql.connector*. Este permite tener acceso a la información almacenada dentro de la base de datos para realizar consultas y operaciones sobre los datos y continuar con el flujo del prototipo.

4.1.2 Módulo de procesamiento

Para este módulo se utilizó Python como lenguaje de desarrollo principal en su versión 3.6.5. Ya que este se adapta de manera satisfactoria a las necesidades del prototipo y a su requerimiento de usar múltiples librerías de código abierto. Así mismo, se escogió el entorno de desarrollo integrado (IDE, por sus siglas en inglés) Pycharm de la compañía

JetBrains con el fin de gestionar el proyecto en un ambiente que facilite el proceso de desarrollo, compilación y ejecución del código.

Las funciones desarrolladas permiten una comunicación con el módulo de almacenamiento para obtener y realizar operaciones sobre la base de datos. Así mismo, estas permiten una comunicación activa con la interfaz de usuario para una visualización correcta de los datos procesados. Cada una de las funciones principales están vinculadas con una opción disponible del menú principal del prototipo las cuales son: *(Camera Live Stream, Train Classifier, Add Familiar Face, Delete Familiar Face*). Estas opciones fueron desarrolladas siguiendo los casos de uso previamente descritos.

Es importante recalcar que, para el proceso de detección de rostros y reconocimiento facial, el módulo utiliza la librería de visión artificial OpenCV. La cual implementa un clasificador en cascada y un clasificador de patrones binarios locales necesarios para la arquitectura planteada dentro del proyecto. Estos algoritmos presentan una seria de hiper parámetros disponibles para la mejora y personalización de sus modelos. Los cuales se detallan a continuación:

Clasificador en cascada:

- Image: Matriz que contiene el fotograma de imagen donde se detectaran los objetos.
- scaleFactor: Parámetro que especifica cuanto se reduce el tamaño de la imagen de entrada con respecto a cada escala. Es decir, al re-escalar la imagen de entrada a una escala diferente a la original, los rostros dentro de esa imagen pueden cambiar de tamaño lo que los convierte en coincidencias detectables por el algoritmo. Valores cercanos a 1 requieren más tiempo de procesamiento y generan mejores resultados. Valores altos requieren menos tiempo de procesamiento y aceleran el proceso de detección del algoritmo.

- minNeighbors: Parámetro que especifica cuantos vecinos debe tener cada rectángulo que se considera candidato para retenerlo. Valores altos dan como resultado menos detecciones de rostros, pero estas poseen mayor calidad. Valores bajos dan como resultado, más detecciones de rostros con mayor probabilidad de falsos positivos.
- minSize: Parámetro que especifica el tamaño mínimo de un objeto detectable, cualquier objeto detectado más pequeño que el parámetro definido se ignora automáticamente.
- maxSize: Parámetro que especifica el tamaño máximo de un objeto detectable, cualquier objeto más grande que el parámetro definido se ignora automáticamente.

Patrones binarios locales:

- Radius: Parámetro necesario para la construcción del patrón binario local circular, representa el radio alrededor del píxel central, el cual es necesario para construir el patrón binario local.
- Neighbours: Parámetro que especifica el número de puntos de muestra necesarios para construir el patrón binario local circular. Valores altos requieren un mayor costo computacional. El valor más común es 8.
- Grid X: Parámetro que especifica el número de celdas en dirección horizontal para la división de los histogramas. A mayor número de celdas mayor será la dimensionalidad del vector de características resultantes. Su valor común es 8.
- Grid Y: Parámetro que especifica el número de celdas en dirección vertical para la división de los histogramas. A mayor número de celdas mayor será la dimensionalidad del vector de características resultantes. Su valor común es 8.

- Threshold: Parámetro que especifica el umbral aplicado en la predicción. Este especifica la distancia entre los histogramas que se comparan en la búsqueda de coincidencias. Si la distancia del vecino más cercano es mayor al valor del umbral este devuelve un -1, es decir no existe una coincidencia entre los histogramas comparados.

Los hiper parámetros escogidos para el prototipo están basados en la experimentación y pruebas realizadas, así como en recomendaciones de los autores de la librería y la experiencia de otros usuarios con estos clasificadores. Los parámetros escogidos se detallan en las tablas descritas a continuación. Adicionalmente, dentro de la sección de discusión y resultados se presenta un análisis sobre las ventajas y desventajas de estos clasificadores con respecto a otros algoritmos conocidos.

Clasificador en cascada	
Parámetro	Valor
Scalefactor	1.01
minNeighbors	5
minSize	[150,150]
maxSize	[150,150]

Tabla 1 Parámetros escogidos para el clasificador en cascada

Clasificador patrones binarios locales	
Parámetro	Valor
Radius	1
Neighbours:	8
Grid X	8
Grid Y	8
Threshold	100

Tabla 2 Parámetros escogido para el clasificador de LBP

4.1.3 Interfaz de usuario

Para la interfaz de usuario se utilizó la librería Tkinter, la cual permite realizar una interfaz sencilla pero eficaz de manera ágil. Una de las ventajas de esta librería es que se encuentra integrada en el paquete de instalación de Python de manera predeterminada, lo que permite una comunicación sencilla con el procesamiento del sistema. Las ventanas de la interfaz de usuario se dividieron en clases individuales, donde cada una se conforma por los widgets integrados dentro de la librería, cada una de estas ventanas se diseñaron de tal forma que su interacción sea clara e intuitiva para el usuario. Consiguiendo que el uso del prototipo y sus funciones sea simple y sencillo.

4.2 Herramientas, librerías y módulos complementarios

En las siguientes tablas se muestran las principales herramientas, librerías, y complementos utilizados dentro del desarrollo del prototipo junto a una breve descripción de sus funcionalidades principales.

Herramienta	Descripción
Python	Lenguaje de programación interpretado multiparadigma cuya filosofía radica en la legibilidad y facilidad de leer y escribir código.
JetBrains Pycharm	Ambiente de desarrollo integrado para el lenguaje de programación Python.
MySQL Workbench	Herramienta visual de diseño, administración, gestión y mantenimiento para sistemas de bases de datos.

Tabla 3 Herramientas principales

Librería / Módulo	Descripción
OpenCV	Librería de visión artificial desarrollada para Python, se encuentra posiciona como parte de las mejores y más veloces dentro de su campo aplicación.
Tkinter	Tkinter es un binding de la biblioteca grafica Tcl/Tk para Python la cual proporciona gran facilidad para el manejo de herramientas de GUI dentro de los proyectos.
Pyttsx3	Pyttsx3 es una librería TTS (text to speech) de conversión de texto a voz desarrollada para Python, que funciona sin la necesidad de una conexión de red.
Pywhatkit	Pywhatkit es una librería multifuncional desarrollada para Python la cual permite consultas y redireccionamiento a sitios en internet.
Scikit-learn	Biblioteca de aprendizaje automático de uso libre escrito para Python, la cual contiene gran variedad de funcionalidades para el preprocesamiento, clasificación o aprendizaje.

Smtplib	Smtplib es un módulo que define un objeto de sesión de cliente SMTP que se puede utilizar para enviar correos electrónicos a maquinas dentro del internet.
Ssl	SSL es un módulo diseñado para crear una conexión segura entre un cliente y un servidor para evitar ataques de escucha dentro de la comunicación.
NumPy	NumPy es una librería para Python que permite la creación y manipulación de vectores y matrices multidimensionales.
Datetime	Date time es un módulo que proporciona métodos y clases para la manipulación de fechas y horas,
Mysql.connector	Conector necesario para realizar operaciones dentro de una base de datos desde un programa escrito en Python.

Tabla 4 Librerías y Módulos Complementarios

4.3 Prueba de concepto

Tras el desarrollo de los módulos que integran el sistema, se definió la prueba de concepto a realizar. Esta prueba busca evaluar la eficacia del sistema desarrollado y el cumplimiento de los objetivos planteados dentro de un ambiente controlado. Para esto se definió un flujo de uso que el sistema debe cumplir y las condiciones previas a esta. Las cuales se detallan a continuación.

Condiciones previas

- El sistema tiene un usuario registrado previamente quien es considerado propietario del sistema.
- El propietario del sistema realiza un inicio de sesión exitoso.
- El sistema posee información facial sobre el propietario.

Flujo de uso:

- Un usuario diferente al propietario del sistema añadirá su información facial.
- El usuario y el dueño del sistema serán capaces de ver la transmisión de video y ser reconocidos por el sistema.
- Se recibirá una notificación por parte del sistema.
- Se eliminará la información facial del usuario adicional.
- El usuario deberá recibir las notificaciones respectivas por parte del sistema al no ser reconocido.

4.3.1 Realización de prueba de concepto

Después de la definición de la prueba a realizar se procedió a demostrar la funcionalidad del prototipo al ejecutar este proceso. A continuación, se detalla lo obtenido. Primero, se procede a cumplir con las condiciones descritas en la prueba. Para esto se inicia el prototipo, el cual despliega el menú de inicio.

Figura 19 Menú de inicio

Luego, se registra al usuario “Tadashy Curichumbi” haciendo uso de la ventana de creación de usuario, la cual coincide con el caso de uso definido. Este usuario al ser el primero en registrarse, es considerado el propietario del sistema para la simplificación del proceso.

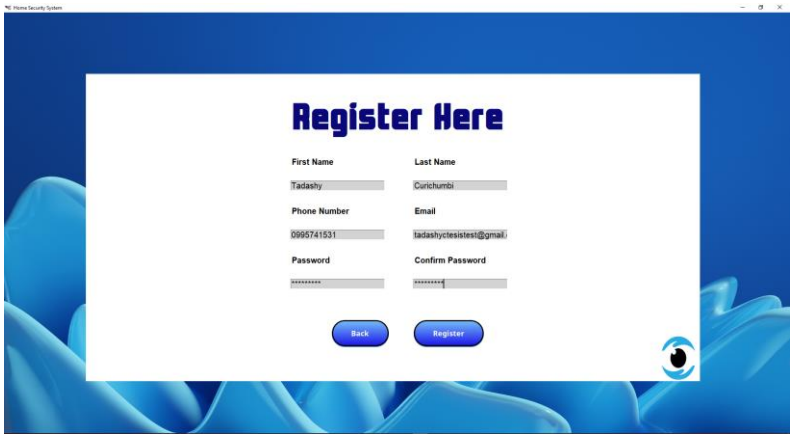

Figura 20 Ejemplo de registro de un usuario

Posteriormente, el usuario es redireccionado a la ventana de inicio de sesión con el fin de ingresar al sistema exitosamente. A continuación, este es redirigido al menú principal del prototipo y procede a agregar su información facial dentro del sistema por medio de la opción *Add Familiar Face* y sus indicaciones.

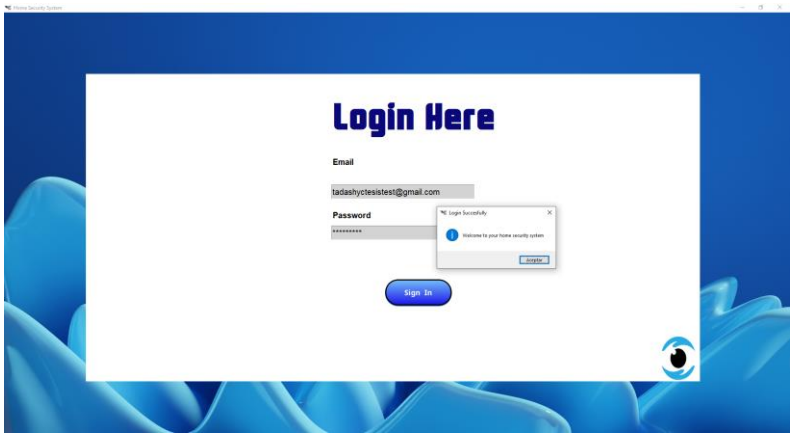

Figura 21 Ejemplo de inicio de sesión del usuario

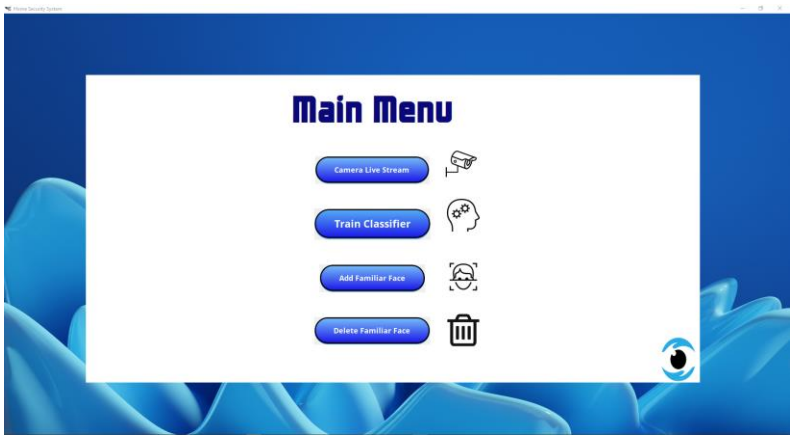

Figura 22 Menú principal

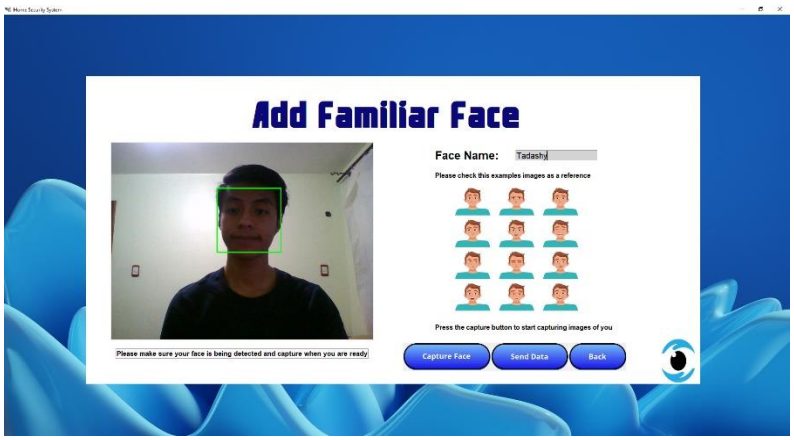

Figura 23 Ejemplo registro de información facial

Tras cumplir con las condiciones previas descritas en la prueba de concepto se procede a cumplir con el flujo de uso definido. Por lo que se agrega la información facial de un nuevo usuario dentro del sistema.

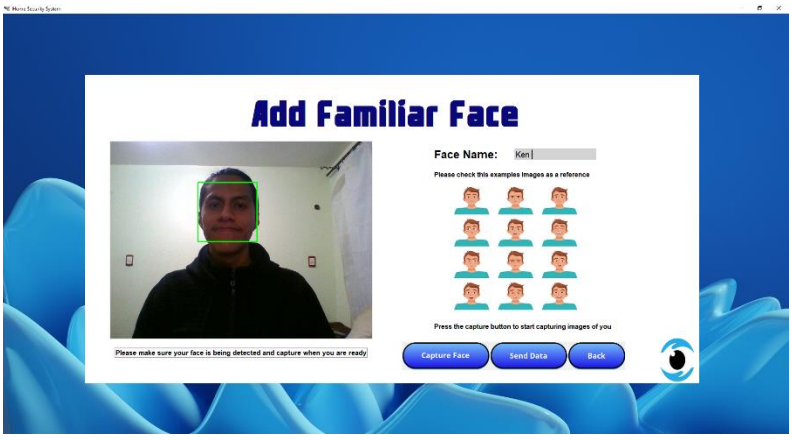

Figura 24 Registro de información facial del usuario adicional

El usuario procede a entrenar el clasificador con la información almacenada dentro de la base de datos por medio de la opción *Train Classifier*. La cual se encuentra disponible en el menú principal. Posterior a recibir el mensaje de confirmación de parte del sistema, el usuario procede a ver la transmisión de la cámara y verificar su detección y autentificación dentro

del sistema. Una vez autentificado, el sistema procede a saludar al usuario que se encuentra en pantalla. Este proceso ocurre de igual manera para el usuario quien es considerado el dueño del sistema.

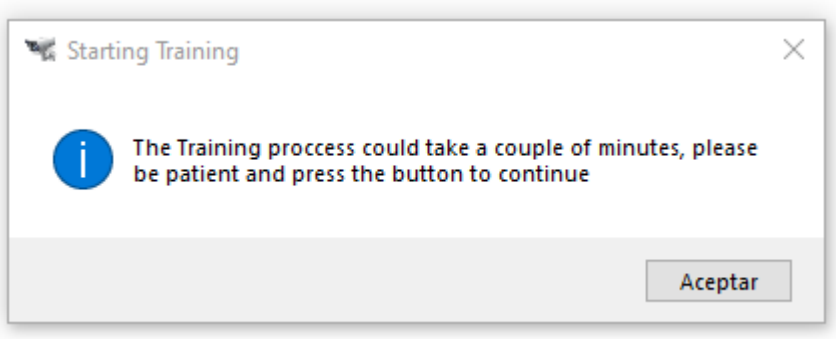

Figura 25 Mensaje de confirmación de inicio del entrenamiento

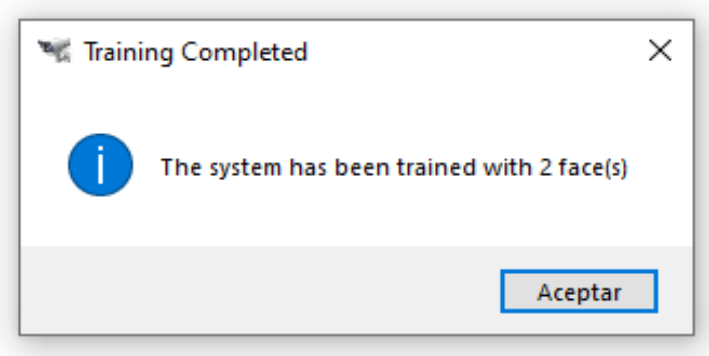

Figura 26 Mensaje de confirmación de entrenamiento completado

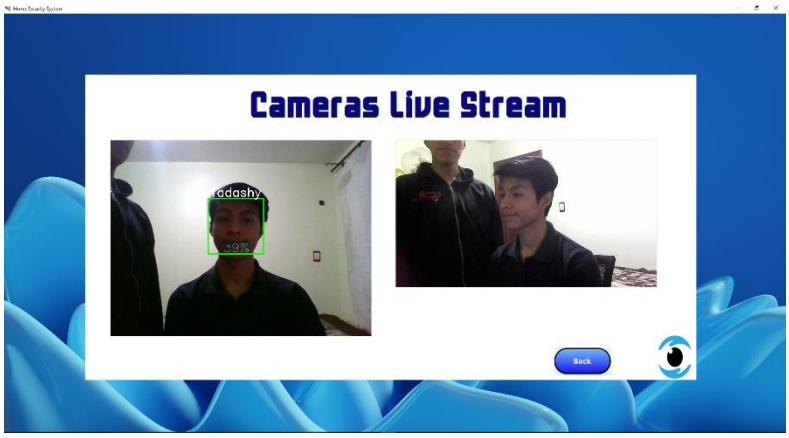

Figura 27 Reconocimiento de rostro familiar 1

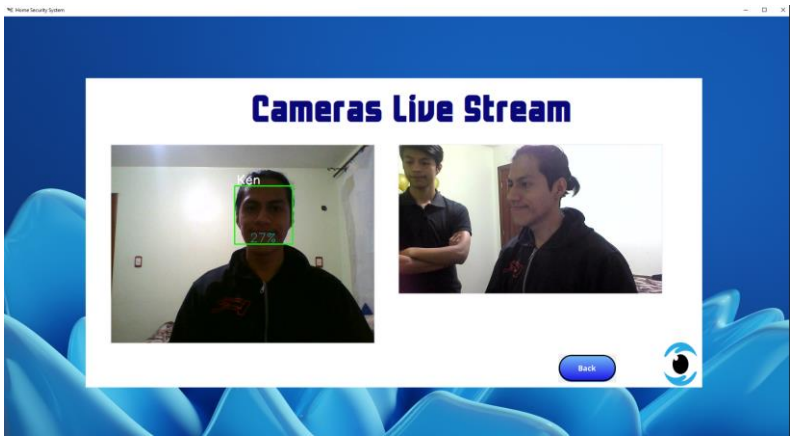

Figura 28 Reconocimiento rostro familiar 2

A continuación, se procede a eliminar la información facial del usuario adicional del sistema mediante la opción *Delete Familiar Face* disponible. El usuario procede a entrenar nuevamente el clasificador con el fin de que este permanezca actualizado con la última información disponible.

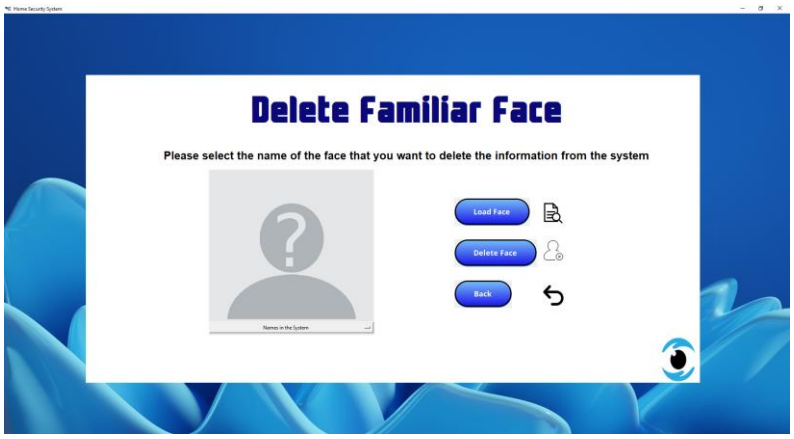

Figura 29 Ejemplo eliminación de información facial

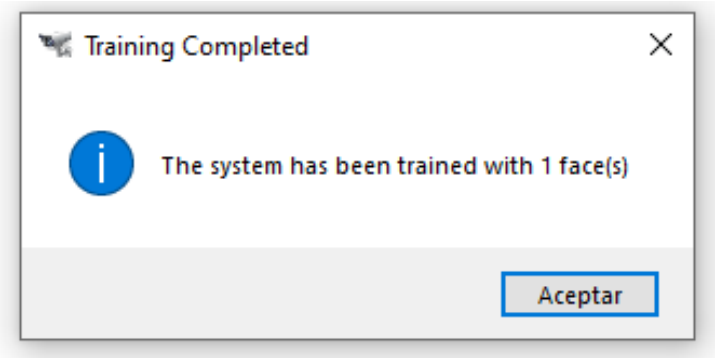

Figura 30 Mensaje de confirmación de entrenamiento

Finalmente, el usuario procede a revisar la transmisión de video dentro del sistema y verificar su detección y autentificación. El sistema al no poseer información sobre este rostro procede a autentificarlo como desconocido y a emitir las alarmas definidas para el prototipo. Concluyendo así con el flujo propuesto dentro de la prueba de concepto.

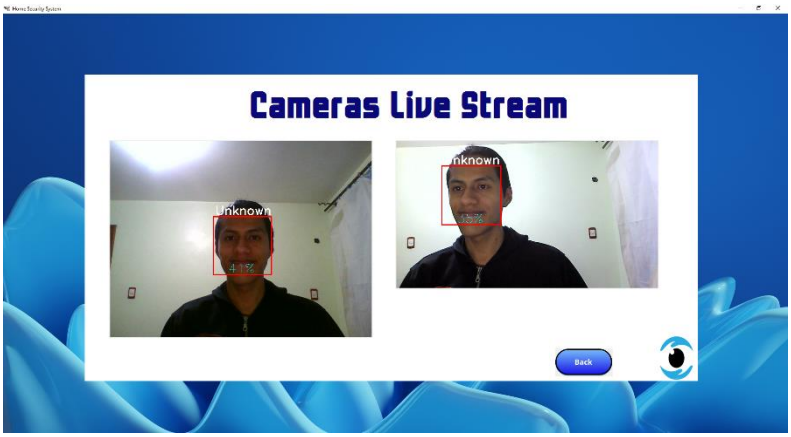

Figura 31 Ejemplo de rostro etiquetado como desconocido

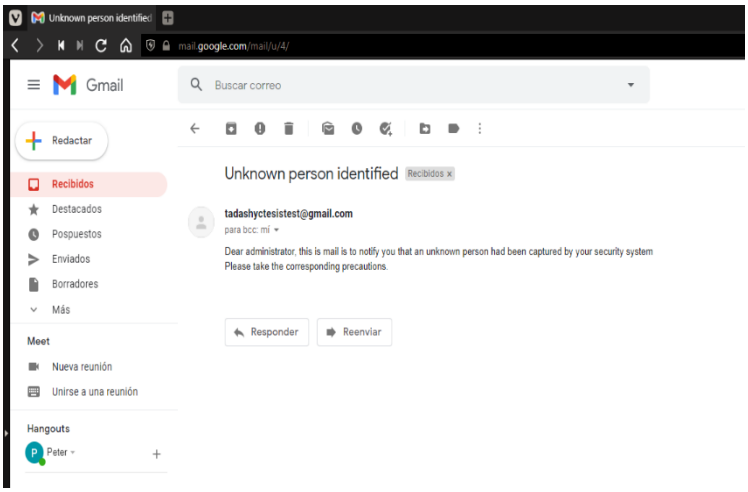

Figura 32 Alarma enviada por correo electrónico

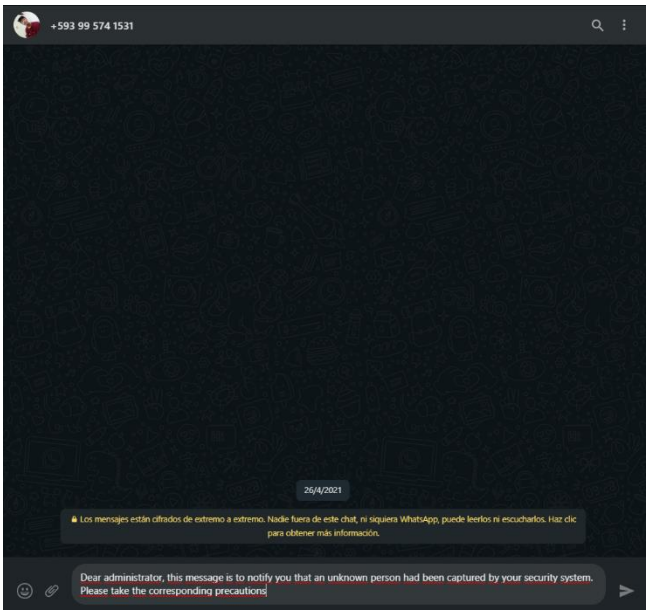

Figura 33 Alarma enviada por mensaje de Whatsapp

Nota: Esta prueba de concepto fue limitada a un ambiente controlado y seguro debido a las condiciones sanitarias presentes al momento del desarrollo del prototipo, con el fin de precautelar la salud y bienestar de los integrantes y participantes del proyecto.

5 RESULTADOS Y DISCUSIÓN

Con la prueba de concepto realizada, se comprobó que cada una de las funciones y módulos desarrollados operan dentro de un rango funcional óptimo, cumpliendo satisfactoriamente con los objetivos planteados para el prototipo.

A su vez, se evidenciaron varios aspectos interesantes sobre los clasificadores utilizados dentro del sistema para el proceso de detección y reconocimiento facial. Se observo que ambos clasificadores poseen ventajas y desventajas con respecto a otras opciones disponibles dentro de la librería de OpenCV.

Con respecto al clasificador en cascada encargado del proceso de detección facial, se observó que la librería nos otorga la facilidad de crear nuevos detectores de rostros. Los cuales estén entrenados con muestras de datos propias, es decir es posible generar y utilizar un clasificador en cascada que este entrenado con imágenes positivas y negativas que el desarrollador proporcione. Sin embargo, a pesar de tener esta opción disponible dentro de la librería es más recomendado utilizar los clasificadores que existen dentro de la librería de OpenCV. Ya que estos poseen un buen nivel de preentrenamiento a partir de conjuntos de datos aleatorios.

A su vez, otra de las razones fundamentales para seleccionar y trabajar con los clasificadores disponibles en la librería. Es que, tal como se mencionó dentro del estado del arte previo, dentro del proceso de detección facial es más importante hacer una correcta detección de rostros, que identificar a quien pertenecen estos. Por lo que, seleccionar un clasificador que este entrenado sin un sesgo de entrenamiento previo con respecto a la localidad de donde se realizan las detecciones es de suma importancia.

Dado que dentro de la librería de OpenCV existen una gran cantidad de clasificadores para detección facial. Para hacer uso de uno de estos, se debe analizar correctamente las situaciones donde el sistema va a operar, ya que cada uno de los clasificadores disponibles presentan un preentrenamiento para diferentes situaciones relacionadas con la detección facial. Como ejemplo de esto tenemos la detección de rostros frontales, detección de rostros y ojos, detección de perfiles, detección de sonrisas, detección de cuerpos completos, entre muchos más con sus respectivas variantes.

Tomando en cuenta que el objetivo principal del prototipo con respecto al proceso de detección facial es la detección de rostros familiares en la mayoría de casos posibles, se optó por el uso y prueba de aquellos clasificadores integrados que no sean tan restrictivos con respecto a su capacidad de detección facial.

Todos los clasificadores se encuentran integrados dentro de la librería en un formato de archivo XML. Entre los principales tenemos los siguientes: *haarcascade_frontalface_default.xml*, *haarcascade_frontalface_alt.xml*, *haarcascade_frontalface_alt2.xml*, *haarcascade_frontalface_alt_tree.xml*. Estos se encuentran listados de mayor a menor con respecto a su capacidad y limitaciones para sus detecciones faciales.

En la fase de desarrollo del prototipo se probaron los tres primeros clasificadores listados previamente y se optó por uno de ellos. Ya que este presenta detecciones agiles y en la mayoría de los casos estas son correctas. A su vez, el clasificador es capaz de otorgar detecciones que no ralentizan en gran medida la transmisión del video mostrado por las cámaras. Además, este permite disminuir la probabilidad de falsos positivos con el uso y modificación de los hiper parámetros de este.

Con respecto al clasificador de patrones binarios locales encargado del proceso de reconocimiento facial, se evidencio que este posee ventajas sobre los otros dos clasificadores implementados en la librería de OpenCV (Fisherfaces, Eigenfaces).

Este clasificador a comparación de los otros presentes en la librería no depende de la reducción de la dimensionalidad de sus componentes. Esto representa una gran ventaja al nivel local, ya que al depender de una alta dimensionalidad de los datos ambas opciones restantes pierden información importante sobre las características del individuo al momento de reducir la dimensionalidad de los datos para su análisis.

A su vez, los clasificadores de Fisherfaces y EigenFaces dependen en gran medida de la cantidad de datos de entrada existentes para la creación de sus vectores de características. Los cuales son necesarios para el proceso de reconocimiento facial de un individuo. Esta limitante puede conllevar a peores resultados dentro un sistema de tiempo real. Por su parte, el clasificador de patrones binarios locales al no tener este nivel de dependencia con respecto al número de imágenes de entrada logra cierto grado de resistencia frente a esta problemática. Ya que este puede identificar individuos a pesar de que exista una pequeña muestra de datos.

De igual manera, al analizar ambos clasificadores y el entorno en donde se desarrollaron las pruebas, se pudo observar que ambos sufren de alteraciones en sus resultados frente a cambios de iluminación, escala, y rotación. Por otra parte, el clasificador de patrones binarios locales, al hacer uso de una descripción local, posee cierto grado de robustez ante estas condiciones. Lo que lo convierte en una opción más viable para el prototipo.

El clasificador de patrones binarios locales fue entrenado con la información proveniente de la base de datos, ya que este al ser encargado del reconocimiento facial debe ser entrenado con los rostros e información facial del lugar donde va a ser usado. Para esto, la creación y entrenamiento de este clasificador se basó en los ya existentes dentro de OpenCV, con el fin de evitar problemas de compatibilidad con el resto de los métodos integrados en la librería y el prototipo.

Al nivel general, se comprobó que ambos clasificadores utilizados dentro del prototipo presentan una reducción en su eficacia debido a factores externos del ambiente tales como: distancia, iluminación, rotación e inclinación del rostro a detectar. Así como de factores relacionados al hardware como la calidad de imagen o la capacidad de procesamiento del equipo donde se ejecuta el sistema.

Se considera que dichas limitaciones se pueden superar implementando mejoras al nivel de hardware como de software. Al hacer uso de mejores equipos para la preparación del sistema, así como una mayor experimentación de todas las posibles combinaciones de los clasificadores con el fin de crear un sistema mucho más robusto que el presente.

6 CONCLUSIONES

- El prototipo diseñado se encuentra en un estado óptimo y es capaz de realizar los procesos descritos en los objetivos y la prueba de concepto dentro de un ambiente controlado.
- El uso del análisis de video inteligente y sus principios dentro del campo de la seguridad constituye una oportunidad de generar un estándar de investigación y desarrollo nacional e internacional.
- Se evidencia una necesidad de impulsar el uso y el mejoramiento continuo de los sistemas de seguridad basados en análisis de video inteligente con la intervención de entidades públicas y privadas con el fin de disminuir los índices de delitos dentro del país.

7 RECOMENDACIONES E IMPLICACIÓN A FUTURO

Con el desarrollo de este trabajo se evidencia la enorme capacidad de uso de tecnologías emergentes como la visión por computador o el análisis de video dentro de sistemas de seguridad para el hogar. Este tipo de tecnologías otorgan la facilidad de introducir nuevos conceptos y formas de mejorar sistemas ya existentes del mercado en muchas maneras y sin muchas complicaciones. Es importante resaltar que, este prototipo a pesar de cumplir con su objetivo funcional requiere de trabajo extra para ser perfeccionado y llegar al nivel de una aplicación en etapa de producción. Este sistema presenta muchas oportunidades de mejora, las mismas que permitirán otorgar un mayor nivel de confianza a los usuarios dentro de una mayor variedad de ambientes de prueba. Algunas de estas recomendaciones a futuro para la mejora del prototipo se detallan a continuación.

- Hacer uso de una interfaz gráfica más moderna relacionada con la normativa de diseño Material Design.
- Complementar el sistema con la implementación de análisis de expresiones faciales, análisis de emociones o la integración de un asistente virtual.
- Integrar el sistema desarrollado a sistemas embebidos de bajo presupuesto.
- Escalar el sistema con mejoras de hardware y otro tipo de distribución como aplicaciones web o aplicaciones para teléfonos inteligentes.

8 REFERENCIAS BIBLIOGRÁFICAS

Cabezón, D. (2008, September). *¿Cómo funciona la detección de caras?* Recuperado el 20 de mayo de 2021 de https://www.xatakafoto.com/camaras/como-funciona-la-deteccion-de-caras

Doxygen. (2021, May). *Face Recognition with OpenCV*. Recuperado el 14 de abril de 2021 de https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html

Doxygen. (2021, May). *Cascade Classifier*. Recuperado el 14 de abril de 2021 de https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

EFF. (2017, October). *Face Recognition.* Recuperado el 20 de febrero de 2021 de https://www.eff.org/es/pages/face-recognition

Expert.ai Team (2020, May). *What is Machine Learning? A Definition.* Recuperado el 16 de marzo de https://www.expert.ai/blog/machine-learning-definition/

Hispanofil. (s.f). *Catalogo Bosch Security*. Recuperado el 14 de marzo de 2021 de https://hispanofil.es/wp-content/uploads/sites/10/2016/10/Cat%C3%A1logo-Bosch-Security-videovigilancia.pdf

Harmouch, M. (2020, July). *Face Recognition Based On LBPH Algorithm.* Recuperado el 14 de abril de 2021 de https://blog.devgenius.io/face-recognition-based-on-lbph-algorithm-17acd65ca5f7

IBM Cloud Education. (2020, July). *Machine Learning.* Recuperado el 16 de marzo de https://www.ibm.com/cloud/learn/machine-learning

JavaPoint. (s.f). *Face recognition and Face detection using the OpenCV*. Recuperado el 14 de abril de 2021 de https://www.javatpoint.com/face-recognition-and-face-detection-using-opencv

Khan, T. (2019, December). *Computer Vision — Detecting objects using Haar Cascade Classifier*. Recuperado el 15 de abril de 2021 de https://towardsdatascience.com/computer-vision-detecting-objects-using-haar-cascade-classifier-4585472829a9

Landa, J., Jun, C., & Jun, M. (2017, January). *Implementation of a Remote Real-Time Surveillance Security System for Intruder Detection*. In 2017 9th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (pp. 102-105). IEEE

Mihajlovic, I. (2019, April). *Everything You Ever Wanted To Know About Computer Vision*. Recuperado el 16 de marzo de https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e

Muñoz, R. (2018, November). *Qué son los Sistemas Biométricos y cómo ayudan a mejorar la seguridad.* Recuperado el 20 de febrero de 2021 de https://www.inacorpsa.com/sistemas-biometricos/

Numbeo. (2021). *América: Índice de Criminalidad por País 2021*. Recuperado el 14 de marzo de 2021 de https://es.numbeo.com/criminalidad/clasificaciones-por-pa%C3%ADs?región=019&title=2021

Pawar, S., Kithani, V., Ahuja, S., & Sahu, S. (2018, August). *Smart Home Security Using IoT and Face Recognition. In 2018 Fourth International Conference on Computing Communication Control and Automation* (ICCUBEA) (pp. 1-6). IEEE.

Postec Technology. (s.f). *SISTEMA DE MONITOREO Y VIGILANCIA ELECTRONICA.* Recuperado el 14 de marzo de 2021 de http://www.postec.com.br/esp/sistema-monitoreo-vigilancia-

electronica#:~:text=El%20sistema%20electr%C3%B3nico%20de%20monitoreo,protecci%C3%B3n%20a%20todo%20el%20sitio.&text=Este%20tipo%20de%20monitoreo%20es,un%20alto%20flujo%20de%20personas.

Rosebrock, A. (2015, December). *Local Binary Patterns with Python & OpenCV*. Recuperado el 14 de abril de 2021 de https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/#:~:text=Local%20Binary%20Patterns%20with%20Python%20and%20OpenCV,for%20raw%20LBP%20histogram%20computation.

Salton do Prado, K. (2017, November). *Face Recognition: Understanding LBPH Algorithm.* Recuperado el 14 de abril de 2021 de https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b

Stephens, J. (2005). *Face Detection Project*. Gatech.edu. Recuperado el 14 de abril de 2021 de https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj5/html/stephens3/index.html

Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media.

Statista Research Department (2016, November). *Ranking mundial por ingresos de los fabricantes de cámaras de video vigilancia en 2015*. Recuperado el 14 de marzo de 2021 de https://es.statista.com/estadisticas/638281/ingresos-por-ventas-de-los-fabricantes-de-seguridad-mas-importantes-en-el-mundo/

SafeWise Team (2020, October). *What Is a Security System and How Does it Work?* Recuperado el 14 de marzo de 2021 de https://www.safewise.com/home-security-faq/how-do-security-systems-work/#:~:text=The%20most%20basic%20definition%20of,of%20interworking%20compon

ents%20and%20devices.&text=Wired%20or%20wireless%20security%20cameras,high-decibel%20siren%20or%20alarm

Tizkowski, D. (2018, November). *What Are the Different Types of Home Alarm Systems?* Recuperado el 14 de marzo de 2021 de https://www.vectorsecurity.com/blog/what-are-the-different-types-of-home-alarm-systems

THALES. (2021, April). *Facial recognition: top 7 trends (tech, vendors, markets, use cases & latest news).* Recuperado el 20 de febrero de 2021 de https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition

Tutorialspoint.com. (2021) *OpenCV Tutorial.* Recuperado el 04 de abril de 2021 de https://www.tutorialspoint.com/opencv/index.htm

Verma, J. P., Agrawal, S., Patel, B., & Patel, A. (2016). *Big data analytics: Challenges and applications for text, audio, video, and social media data*. International Journal on Soft 1Computing, Artificial Intelligence and Applications (IJSCAI), 5(1), 41-51.