

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE INGENIERÍA MECÁNICA AUTOMOTRIZ

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO MECÁNICO AUTOMOTRIZ**

PROYECTO TÉCNICO

TEMA:

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD
VEHICULAR MEDIANTE RECONOCIMIENTO FACIAL A TRAVÉS DE
VISIÓN ARTIFICIAL”.**

AUTORES:

**MARCO VINICIO CAJAS IDROVO
PABLO ANDRÉS VIRI ÁVILA**

TUTOR:

ING. JUAN DIEGO VALLADOLID, MSc.

**CUENCA – ECUADOR
2017**

CESIÓN DE DERECHOS DE AUTOR

Nosotros Marco Vinicio Cajas Idrovo con C.I.: 0105391239 y Pablo Andrés Viri Avila con C.I.: 0302362348, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD VEHICULAR MEDIANTE RECONOCIMIENTO FACIAL A TRAVÉS DE VISIÓN ARTIFICIAL”, mismo que ha sido desarrollado para optar por el título de: Ingeniero Mecánico Automotriz, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

.....
Marco Vinicio Cajas Idrovo
0105391239

.....
Pablo Andrés Viri Avila
0302362348

Cuenca, Marzo 2017

CERTIFICACIÓN

Yo declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD VEHICULAR MEDIANTE RECONOCIMIENTO FACIAL A TRAVÉS DE VISIÓN ARTIFICIAL“, realizado por, Marco Vinicio Cajas Idrovo y Pablo Andrés Viri Avila, obteniendo el Proyecto Técnico, que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, Marzo del 2017



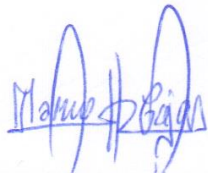
Ing. Juan Diego Valladolid Quitoisaca, Msc.

0104821210

DECLARATORIA DE RESPONSABILIDAD

Nosotros Marco Vinicio Cajas Idrovo con C.I.: 0105391239 y Pablo Andrés Viri Avila con C.I.: 0302362348 autores del “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD VEHICULAR MEDIANTE RECONOCIMIENTO FACIAL A TRAVÉS DE VISIÓN ARTIFICIAL” certificamos que el total contenido del Proyecto Técnico, son de nuestra exclusiva responsabilidad y autoría

Cuenca, Marzo del 2017



.....
Marco Vinicio Cajas Idrovo
0105391239



.....
Pablo Andrés Viri Avila
0302362348

DEDICATORIA

El presente trabajo de titulación lo dedico a Dios por haberme dado el privilegio de vivir, a mis padres por todo el esfuerzo y la confianza que han puesto en mí, a mis hermanos que siempre me han están apoyándome incondicionalmente y que gracias a toda mi familia que son lo más valioso de mi vida he sabido aprovechar esta gran oportunidad.

Marco Vinicio Cajas Idrovo

DEDICATORIA

Dedico este proyecto técnico primero a Dios por guiarme siempre por el buen camino, acompañándome día a día. A mi abuelito Modesto Viri que con su alegría, anécdotas y consejos nos dejó un gran legado de vida y desde el Cielo nos cuida. A mis padres Luis Viri y Beatriz Ávila que son el pilar fundamental de mi familia, que con su trabajo y esfuerzo han sabido guiarme toda la vida para ser una persona de bien y haber depositado su total confianza en mi persona para lograr grandes éxitos como haber terminado mis estudios superiores. A mi hermana Susana Viri, por su incondicional ayuda y ánimos en todo el trayecto de mi vida personal y educativa. Y a una persona muy especial en mi vida Maribel León por su cariño y comprensión y ser la inspiración para cumplir mis metas.

Pablo Andrés Viri Ávila

AGRADECIMIENTOS

Agradezco infinitamente a Dios y a la Virgen Santísima por haber iluminado mi mente, por haberme dado la sabiduría y la inteligencia suficiente para salir en adelante, a mis padres por todo el afán que han puesto en mí ya que con la ayuda de ellos estoy alcanzando un logro tan importante en mi vida, a mis hermanos que siempre me hicieron de ver las cosas primordiales e importantes de la vida.

Agradezco enormemente a mi gran amigo Pablo Viri por colaborar con la realización de este proyecto técnico, por ser una persona paciente y siempre dar lo mejor de él, espero que Dios le bendiga por el camino del bien y siga consiguiendo éxitos en su vida; Al Ing. Juan Diego Valladolid por darme la acogida y por brindar sus conocimientos para llevar a cabo este proyecto técnico, deseo que siga cosechando triunfos porque de verdad se lo merece por ser una gran persona.

Marco Vinicio Cajas Idrovo

AGRADECIMIENTOS.

Agradezco primero a Dios por cuidarme toda la vida con su manto protector, a mis padres, amigos, familiares y a todos los profesores de la universidad que de una u otra manera han logrado inculcarme valores éticos y conocimientos técnicos dentro y fuera de las aulas. A mis compañeros de la Cía. De Teatro, en especial al Director Jorge Adrián Méndez por sus sabios consejos y hacerme conocer el maravilloso mundo de la actuación.

Agradezco también a mi compañero Marco Cajas que con su apoyo incondicional y trabajo conjunto logramos terminar con éxito nuestro proyecto de titulación, dejándonos una grata experiencia de compartir conocimientos y plasmarlos en un excelente proyecto.

Agradezco enormemente también al Ing. Juan Diego Valladolid por apoyarnos desde un principio con la realización de este proyecto y brindarnos su valiosísimo tiempo para la revisión e incondicional ayuda.

Pablo Andrés Viri Ávila

ÍNDICE DE CONTENIDOS

RESUMEN	14
ABSTRACT	15
INTRODUCCIÓN.....	16
CAPÍTULO 1	17
1. MARCO REFERENCIAL.....	17
1.1. Antecedentes.	17
1.2. Justificación del Trabajo de Titulación.	17
1.3. Objetivos.	18
1.3.1. Objetivo general.	18
1.3.2. Objetivos específicos.	18
1.4. Seguridad vehicular convencional.	18
1.4.1. Sistema de rastreo del vehículo.....	19
1.4.2. Traba volante / traba palanca.	19
1.4.3. Goodlock.	19
1.4.4. Corta corriente.	19
1.4.5. Llaves codificadas.	19
1.5. Estadísticas de robo de vehículos.	21
1.6. Efectividad de los sistemas enfocados a la visión artificial.	22
CAPÍTULO 2	24
2. MARCO TEÓRICO.....	24
2.1. Visión Artificial.....	24
2.2. Métodos de reconocimiento de rostros.	24
2.3. Reconocimiento facial	27
2.3.1. Eigenfaces	28
2.3.2. Fisherfaces.	29
2.3.3. Algoritmo LBPH.	31
2.4. Cascada de clasificadores Haar.....	34
2.5. Como trabajan los algoritmos.	36
2.5.1. Los sistemas de reconocimiento facial automático.....	36
2.5.2. Tipos de errores y su medición.....	36
2.6. Reconocimiento de Patrones.....	37
2.6.1. Patrones.	39
2.6.2. Similitud.....	39
2.6.3. Diseño de un sistema de reconocimiento de patrones.	39
2.7. Incidencia de la luz, colores, posición, distancia y accesorios.....	41
2.8 OpenCV	41
2.8.1. Instalación de OpenCV en Raspbian	42
2.8.2. Manipulación de imágenes.....	42
2.8.3. Acceso a la cámara y obtención de imágenes.....	43
2.8.3.1. Cámaras Web USB	44
2.8.4. Detectar rostro.....	44
2.8.5. Inconvenientes de las OpenCV.....	45
2.8.6. Estructura modular OpenCv.	45
2.9. Sistema del automóvil a considerar.....	46
CAPÍTULO 3	47
3. MARCO APLICATIVO.....	47
3.1. Diseño del sistema.....	47
3.2. Selección de elementos para el sistema.....	48
3.2.1. Raspberry Pi 3 Model B	48
3.2.1.1. Pines GPIO de la Raspberry Pi.....	49

3.2.1.2. Tarjeta de memoria Micro SD de 32GB	50
3.2.2. Elementos electrónicos.....	51
3.2.2.1. Arduino Mega 2560.....	51
3.2.2.2. Mini-webcam con conexión USB.....	52
3.2.2.3. La compuerta lógica OR.....	52
3.2.2.4. Compuerta AND o compuerta Y.....	53
3.2.2.5. Teclado de membrana 4x4.....	53
3.2.2.6. Relé 4 patas de 12V – 30 Amp.....	54
3.2.2.7. Transistor bipolar NPN 2N3055.....	54
3.2.2.9. Diodo 1N4007.....	55
3.2.2.10. Display LCD alfanumérico de 16x2.....	55
3.3. Selección de Técnicas de Reconocimiento Facial.....	56
3.4. Circuito de alimentación de la Raspberry.....	57
3.5. Recolección de base de datos de rostros.....	58
3.6. Esquema General del proyecto.....	59
3.7. Programación en Python para Reconocimiento Facial.....	61
3.7.1. Fase de entrenamiento.....	61
3.7.2. Fase de reconocimiento.....	62
3.7.3. Activación de alarmas y actuadores.....	63
3.8. Programación para clave de acceso en Arduino.....	64
3.9. Activación de la bomba de gasolina.....	67
CAPÍTULO 4	69
4. COSTOS, PRUEBAS Y RESULTADOS.....	69
4.1. Costo de elementos.....	69
4.2 Pruebas de laboratorio.....	70
4.3. Pruebas realizadas.....	71
4.3.1. Tiempos que tardan en iniciarse los algoritmos.....	71
4.3.2. Efectividad del reconocimiento.....	74
4.3.2.1. Efectividad de reconocimiento para usuarios autorizados.....	74
4.3.2.2. Efectividad de reconocimiento para usuarios no autorizados.....	78
4.3.2.3. Efectividad de acceso con la clave de usuario.....	80
4.3.3. Verificación de encendido de la bomba de gasolina.....	81
4.3.4. Pruebas de funcionamiento.....	81
CONCLUSIONES.....	83
RECOMENDACIONES.....	85
BIBLIOGRAFÍA.....	86

ÍNDICE DE FIGURAS

Figura 1: Robo de vehículos en el 2014 y 2015.....	21
Figura 2: Primeras siete Eigenfaces (a) y Fisherfaces (b), obtenidas con un conjunto de imágenes de la base Extended Yale Face Database B.	30
Figura 3: Comparación de PCA y FLD para un problema de dos clases.....	31
Figura 4: Operadores LBP.....	32
Figura 5: Cascada de detectores propuesta por Viola y Jones.....	35
Figura 6: Distribución de probabilidad de un usuario impostor y un usuario genuino.....	37
Figura 7: Esquema general de un sistema de reconocimiento de patrones.	40
Figura 8: Sistema operativo instalado.	42
Figura 9: Función GasussianBlur.	43
Figura 10: Función Erode.....	43
Figura 11: Función Dilate	43
Figura 12: Sistema de alimentación de combustible del Chevrolet Aveo Activo.	46
Figura 13: Diagrama de flujo del sistema.	47
Figura 14: Raspberry Pi 3 Model B.	48
Figura 15: GPIO de la Raspberry pi 3 model B.	50
Figura 16: Tarjeta Micro SD SanDisk Extreme de 32GB.....	50
Figura 17: Arduino Mega 2560.	51
Figura 18: Cámara mini-web con conexión USB.	52
Figura 19: Compuerta lógica OR.	53
Figura 20: Compuerta AND o compuerta Y.....	53
Figura 21: Teclado matricial 4x4.	54
Figura 22: Relé 4 patas.....	54
Figura 23. Transistor NPN 2N3055	55
Figura 24. Diodo 1N4007	55
Figura 25: Display LCD alfanumérico de 16x2.....	55
Figura 26: regulador de voltaje de 12V a 5V.	57
Figura 27: Alimentación 12V del regulador de voltaje.....	58
Figura 28: Nueva base de datos de rostros.....	58
Figura 29: Recolección de fotos.....	59
Figura 30: Esquema general.....	60
Figura 31: Fase de entrenamiento.	61
Figura 32: Fase de reconocimiento facial.	62
Figura 33: Predicción del rostro.....	62
Figura 34: Reconocimiento de usuario.	64
Figura 35: Código para usuario desconocido.....	64
Figura 36: Librerías Password y Keypad.....	65
Figura 37: Definición de la Matriz y señales de salida.	65
Figura 38: Start y reinicio del teclado.	66
Figura 39: Conexión entre Arduino y compuertas lógicas.....	66
Figura 40: Esquema de conexión para la bomba de gasolina.	67
Figura 41: Pines del relé de 4 patas.....	68
Figura 42: Pruebas de laboratorio.	70
Figura 43: Línea de comando para la técnica de reconocimiento facial.....	71
Figura 44: Tiempos en seg., que tarda en entrenarse el sistema.	72
Figura 45: Tiempo que tarda en iniciarse el algoritmo en fase de reconocimiento....	74

Figura 46: Efectividad de reconocimiento con Eigenface.	75
Figura 47: Efectividad de reconocimiento con Fisherface.	76
Figura 48: Efectividad de reconocimiento con LBPH.	77
Figura 49: Reconocimiento a Usuario 1.	78
Figura 50: Reconocimiento a Usuario 2.	78
Figura 51: Usuarios no permitidos con Eigenface.	79
Figura 52: Usuarios no permitidos con Fisherface.	79
Figura 53: Usuarios no permitidos con LBPH.	80
Figura 54: Rechazo a usuarios no permitidos.	80
Figura 55: Voltaje de salida del pin 87 del relé.	81

ÍNDICE DE TABLAS

Tabla 1: Costo de instalación de cada sistema antirrobo.....	20
Tabla 2: Variación porcentual de robo de vehículos.	22
Tabla 3: Lista de precios. Presupuesto.	69
Tabla 4: Tiempo en seg., que tarda en entrenarse el sistema.....	72
Tabla 5: Tiempo que tarda en iniciarse el algoritmo en fase de reconocimiento.	73
Tabla 6: Pruebas de reconocimiento de usuarios autorizados con Eigenface.	75
Tabla 7: Pruebas de reconocimiento con Fisherface.	76
Tabla 8: Pruebas de reconocimiento con LBPH.	77
Tabla 9: Negación de acceso a usuarios no permitidos.....	79

RESUMEN

Debido a que en la actualidad los vehículos son fácilmente hurtados, los avances tecnológicos buscan modernizar los sistemas de seguridad para lograr una mayor eficiencia y complicar a los delincuentes. El presente proyecto trata sobre el diseño y ensamblaje de un sistema de seguridad por reconocimiento facial para poner en marcha un vehículo Chevrolet Aveo Activo y permitir el manejo solamente a personas autorizadas. Bajo el parasol izquierdo dentro del habitáculo se monta una mini WebCam, apuntando al apoyacabezas del asiento del chofer; además consta de una base de datos de fotografías de los rostros de los usuarios autorizados en múltiples posturas, gestos y en condiciones de iluminación normales.

El diseño del sistema por reconocimiento facial usa el lenguaje de programación Python y las librerías del software libre OpenCV para procesamiento de imágenes y visión computarizada a través de un computador de placa reducida Raspberry Pi 3 que dispone del sistema operativo Raspbian. El sistema tiene la capacidad de mantener inmovilizado al automotor si el usuario no es reconocido, además tiene una clave de acceso rápido para casos de emergencia o cuando se preste el vehículo.

Se finiquita que la cámara de tipo mini-webcam USB dio buenos resultados en ambientes iluminados, no así en lugares oscuros y de noche; así mismo con el algoritmo de Histogramas de Patrones Binarios Locales (LBPH) se obtuvo mejor rendimiento con 93% de efectividad en detección debido a los ajustes en variaciones de iluminación. El sistema embebido más adecuado fue la Raspberry Pi 3 por sus mejores especificaciones técnicas y porque evita el sobredimensionamiento innecesario del sistema.

ABSTRACT

Because today's vehicles are easily stolen, technological advances seek to modernize security systems to achieve greater efficiency and complicate criminals. The present project is about the design and assembly of a facial recognition safety system to start a Chevrolet Aveo Active vehicle and allow the handling to only authorized persons. Under the left parasol inside the passenger compartment a mini WebCam is mounted, pointing to the headrest of the driver's seat; also consists of a database of photographs of the faces of authorized users in multiple postures, gestures and in normal lighting conditions.

The design of the system for facial recognition uses the computer language Python and the bookstores of the free software OpenCV for prosecution of images and Computer vision across a computer of limited badge Raspberry Pi 3 who has the operating system Raspbian. The system has the aptitude to keep immobilized to the railcar if the user is not recognized, also it has a key of rapid access for emergency cases or when the vehicle lends.

It is concluded that the type camera mini - webcam USB gave good results in illuminated ambiences, not like that in dark and night places; likewise with the algorithm of Histograms of Local Binary Patterns (LBPH) better yield was obtained with 93 % of effectiveness in detection due to the adjustments in lighting changes. The most suitable absorbed system was Raspberry Pi 3 for its best technical specifications and because he avoids on oversize unnecessarily of the system.

INTRODUCCIÓN

Los sistemas de visión artificial son herramientas relativas a los más recientes avances de autenticación que existe hoy en día alrededor del mundo, siendo el reconocimiento facial uno de los más confiables y a los que más se recurre por su fácil utilización.

Con el progreso de la tecnología y el desarrollo de ordenadores más pequeños, eficientes y portátiles, se logra que la aplicación de algoritmos de visión artificial vaya incrementando en inventivas que ayudan a solucionar inconvenientes de mejor modo. El automotor que cuenta con reconocimiento facial a conductores instauro un sistema antirrobo eficaz, de fácil manejo y de bajo costo.

Para el desarrollo de éste proyecto se ha dividido el documento como se detalla a continuación:

El capítulo 1 contiene el marco referencial donde se puntúa los antecedentes, el planteamiento del problema, define los objetivos, la justificación el proyecto y estadísticas del robo de vehículos; El capítulo 2 presenta los principios, conceptos y fundamentos teóricos de los sistemas de visión artificial ligados al reconocimiento facial, a más de ello la seguridad en los vehículos; El capítulo 3 detalla el marco aplicativo que consta de los parámetros de diseño, ensamble e implementación de las partes eléctricas y electrónicas; El capítulo 4 indica los costos, las pruebas y los resultados del diseño; El capítulo 5 presenta las conclusiones y recomendaciones, finalmente se muestra la bibliografía de libros consultados.

CAPÍTULO 1

1. MARCO REFERENCIAL.

1.1. Antecedentes.

Debido a que cualquier persona puede ser víctima del robo de su vehículo, surge la necesidad de invertir en un buen sistema de seguridad que proteja al propietario y al vehículo. Por supuesto que siempre se podrá contar con las tradicionales alarmas, los candados o barras para el volante y pedales y naturalmente con el GPS; sin embargo, dado que los ladrones varían cada vez más sus métodos de atraco, se ha pensado con la presente propuesta desarrollar un sistema de seguridad que aumente la robustez, utilizando tecnología actual (Aguilar, 2016).

De acuerdo con las estadísticas de seguridad ciudadana, proporcionadas por la Coordinación de Investigación y Estadísticas del Ministerio de Turismo, la Zona 6 integrada por las provincias de Azuay, Cañar y Morona Santiago, registro el -44%, que refleja una disminución en el robo de autos en el período enero – abril del 2016 (Turismo, 2016).

Por lo tanto, al implementar este sistema de seguridad vehicular, los indicadores de robos de vehículos pueden disminuir más y los propietarios evitaban pérdidas económicas.

1.2. Justificación del Trabajo de Titulación.

Este proyecto tiene una importancia fundamental debido a que con el diseño y la implementación de un sistema de seguridad vehicular mediante reconocimiento facial a través de visión artificial, se logrará evitar el robo de vehículos lo cual brindará tranquilidad a los propietarios al saber que solo personas autorizadas podrán hacer uso de su vehículo; además podrá estar al alcance de la gran mayoría de propietarios de vehículos, por el bajo costo que tendrá la implementación del sistema.

Tomando en cuenta el problema que tiene actualmente el país en cuanto al robo de vehículos, es necesario realizar este sistema de protección antirrobo para minimizar las pérdidas vehiculares y crear un ambiente de seguridad en la ciudadanía, debido a

que si se decide desde ya implementar sistemas para mejorar la seguridad vehicular los indicadores de robos de vehículos pueden disminuir más.

1.3. Objetivos.

1.3.1. Objetivo general.

- Diseñar e implementar un sistema de seguridad vehicular mediante reconocimiento facial a través de visión artificial.

1.3.2. Objetivos específicos.

- Generar un marco conceptual acerca de los estudios relacionados al presente proyecto.
- Determinar las alternativas técnicas dentro del vehículo para el diseño del sistema de seguridad.
- Seleccionar la técnica para el reconocimiento facial inteligente.
- Seleccionar los componentes eléctricos, electrónicos que permitan realizar el bloqueo y monitoreo del vehículo.
- Implementar el sistema de seguridad mediante reconocimiento facial en el vehículo Chevrolet Aveo Activo.
- Desarrollar las pruebas y realizar las correcciones del sistema de seguridad.

1.4. Seguridad vehicular convencional.

Según estudios en (Llaguno, 2016) a un delincuente promedio, le toma alrededor de 12 segundos en abrir, prender y llevarse el auto mientras está parqueado. Pensar que en los tres minutos que se demora en caminar del trabajo al vehículo ya pudieron robar 15 carros de la ciudad, es de verdad frustrante.

Existen varias maneras de evitar el robo del auto. Parquear en lugares seguros, no dejar objetos de valor a la vista, entre otros. Uno de los métodos más usados es invertir en dispositivos antirrobo para tu auto, pero no todos son los mejores, existen alarmas ineficientes, bloqueos que entorpecen tu manejo y trucos que poco salvarán a tu auto. No obstante, existen también excelentes dispositivos de seguridad, aquí se presenta una lista de 5 sistemas antirrobo para autos (Llaguno, 2016).

1.4.1. Sistema de rastreo del vehículo.

Uno de los dispositivos más eficientes para recuperar el auto. Si, en el peor escenario, el auto es robado, con la localización GPS que este emite, la operadora contratada siempre lo tendrá en su vista. Con un sistema de rastreo GPS, se puede encontrar el vehículo robado una hora después del delito. Los modelos más tecnológicos de este sistema permiten inclusive inmovilizar el vehículo, desde cualquier parte donde esté.

1.4.2. Traba volante / traba palanca.

Ambos son bloqueos mecánicos, manuales y no muy tecnológicos, pero sí eficientes. Traba volante es una de las opciones más económicas y confiables. Por otro lado, el traba palanca es un sistema un poco más sofisticado y sobrio, el cual no permite que la palanca de cambios se pueda usar, por consiguiente, el carro no puede ser manejado.

1.4.3. Goodlock.

Este es un sistema nuevo y curioso creado por un Ingeniero Chileno Alejandro Mackay. Consta de una clave de acciones que se necesita hacer para poder encender el auto. Por ejemplo: pisar el acelerador, pitar y encender el aire. Se puede elegir entre 13.000 combinaciones diferentes y cambiarla a gusto del propietario.?

1.4.4. Corta corriente.

Este dispositivo corta parte del sistema eléctrico del motor para que el auto se detenga. El sistema es activado con un accesorio inalámbrico que puede encenderse con la alarma propia del auto, con un botón escondido o inclusive con una acción, como pisar el freno con el auto apagado. Cortando la recepción de combustible a tu motor, por más que roben tu auto, no podrán llegar muy lejos antes de que este se detenga por completo.






1.4.5. Llaves codificadas.

Esta es la evolución del sistema de llave electrónica o *keyless*, donde no necesitas llaves para encender tu auto, pero un ladrón puede amedrentarte y robar tu tarjeta de encendido. Las nuevas llaves codificadas tienen integrado un código digital. Si alguien intenta encender el auto con cualquier otra llave o método que no sea esta, el sistema lo bloqueará automáticamente.

A continuación se muestra un cuadro comparativo acerca de los costos de instalación de varios sistemas antirrobo para vehículos.

Tabla 1: Costo de instalación de cada sistema antirrobo.

Fuente: Los Autores.

Sistema antirrobo	Costo de instalación (USD \$)	Imagen del sistema
Rastreo satelital y monitoreo	420	
Candado Traba Palanca Pro-t-lock	85	
Goodlock	165	
Switch Botón De Encendido Inmovilizador Con	129	
Llave codificada Chevrolet Aveo	850	

El vehículo puede contar con estas opciones de seguro para cuidarlo o no perderlo en caso de ser víctima de robo.

1.5. Estadísticas de robo de vehículos.

De los nueve delitos de mayor connotación monitoreados, cinco se han reducido drásticamente (homicidios, asesinatos, robo de vehículos, violaciones y muertes por accidentes de tránsito); mientras que los cuatro delitos restantes denotan un crecimiento, (robos a personas, a domicilios a unidades económicas y motos).

A nivel nacional, el robo de vehículos marca una reducción considerable, de 6.461 en el año 2014 se pasó a 5.760 en el 2015 (INEC, 2016). Este hecho en su mayoría se produce en la vía pública y es uno de los indicadores más cercanos a la realidad de inseguridad del país Ecuatoriano, pues el nivel de la no denuncia es mínimo ante el valor que representa el vehículo como también por ser parte del requisito para hacer uso del seguro.

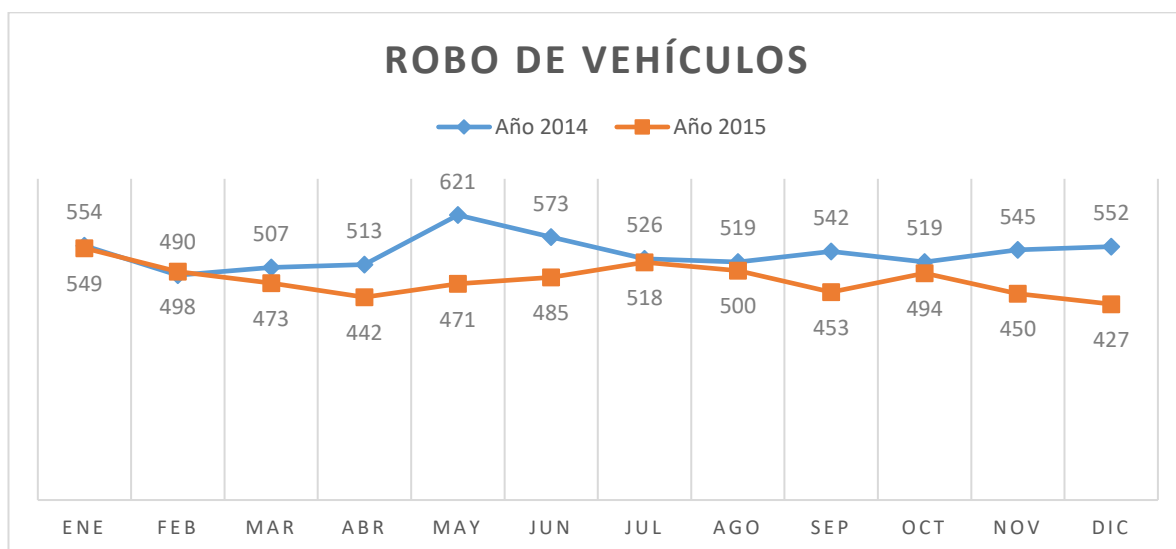


Figura 1: Robo de vehículos en el 2014 y 2015.

Fuente: Los Autores.

Así mismo, a nivel nacional se tiene una tasa de variación acumulada del -15,1%, entre los años 2014 y 2015.

Dentro de la provincia del Azuay la cantidad de vehículos robados es de 248 en el año 2014 y de 136 en el año 2015, por lo tanto, se obtiene una variación porcentual anual del -42,5% (INEC, 2016).

Tabla 2: Variación porcentual de robo de vehículos.

Fuente: (INEC, 2016).

VARIACIÓN PORCENTUAL ANUAL 2015-2014		
Delito	Mes	Variación Porcentual [%]
Robo de vehículos	Enero	-0,9
	Febrero	1,6
	Marzo	-6,7
	Abril	-13,8
	Mayo	-24,2
	Junio	-15,4
	Julio	-1,5
	Agosto	-3,7
	Septiembre	-16,4
	Octubre	-4,8
	Noviembre	-17,4
	Diciembre	-22,6

En Cuenca, los robos a vehículos se dan en mayor número en los sectores Yanuncay, Totoracocha, Ricaurte, Sayausí y El Valle en la zona rural. La mayoría de los robos se dan en la vía pública. Los vehículos pequeños son los más robados. Los ladrones prefieren los colores rojo, blanco, gris y negro, por el contrario los menos robados son los vehículos grandes de color café, tomate y lila. Las noches del sábado y domingo se registra mayor número de robos a vehículos (El Tiempo, 2016).

1.6. Efectividad de los sistemas enfocados a la visión artificial.

Los continuos avances en el sector de la visión artificial permiten desarrollar aplicaciones de seguridad y vigilancia experimentando un importante cambio, que ha hecho necesario la incorporación del control de los sistemas por parte de equipos informáticos mediante software de identificación por visión.

Gracias a las posibilidades que ofrece de obtener la máxima información, la visión artificial se está afianzando como una tecnología imprescindible en los sistemas de vigilancia

Un sistema digital permite controlar todo lo que ocurre frente a la cámara, mediante sistemas de detección de movimiento por visión o por sensores externos, almacenar secuencias de imágenes de distintas cámaras en el disco duro del ordenador, transferir las secuencias a cualquier otro lugar del mundo y posteriormente almacenar las

imágenes en bases de datos, de forma que los sistemas de visión digital son la única alternativa válida en la actualidad, como solución integral de seguridad y vigilancia (INFAIMON, 2014).

Dentro de los sistemas de visión artificial se incluye el software de captura y gestión de imágenes para aplicaciones de seguridad y vigilancia, así como librerías de programación para aplicaciones especiales como lectura de matrículas de automóviles, software de identificación facial, o librerías que permiten realizar seguimientos de individuos o identificación y seguimiento vehículos por visión (INFAIMON, 2014).

CAPÍTULO 2

2. MARCO TEÓRICO.

2.1. Visión Artificial.

La visión artificial es una técnica basada en la adquisición de imágenes en dos dimensiones, para luego ser procesadas por algún tipo de CPU (computadora, microcontrolador, placa, etc.), con la finalidad de extraer y medir determinadas propiedades de las imágenes adquiridas (Herrero, 2005). Se trata, por tanto, de una tecnología que combina las computadoras con las cámaras de video para adquirir, analizar e interpretar imágenes de una forma equivalente a la inspección visual humana.

Podríamos decir que la Visión Artificial (VA) describe la deducción automática de las propiedades de un mundo tridimensional, bien a partir de una o varias imágenes bidimensionales del mundo (Sobrado Malpartida, 2003). Las imágenes pueden ser monocromáticas (de niveles de gris) o colores, pueden provenir de una o varias cámaras e incluso cada cámara puede estar estacionaria o móvil.

La Visión Artificial aplicada a la industria, abarca la informática, la óptica, la ingeniería mecánica y la automatización industrial. A diferencia de la Visión Artificial académica, que se centra principalmente en máquinas basadas en el procesamiento de imágenes. Las aplicaciones de Visión Artificial industrial integran sistemas de captura de imágenes digitales, dispositivos de entrada/salida y redes de ordenador para el control de equipos destinados a la fabricación, tales como brazos robóticos (G.E., 2012). Los sistemas de Visión Artificial se destinan a realizar inspecciones visuales que requieren alta velocidad, gran aumento, funcionamiento las 24 horas del día para aumentar la efectividad de los procesos de producción en el área industrial.

2.2. Métodos de reconocimiento de rostros.

Uno de los grandes problemas en la identificación de rostros es la detección de los mismos por medio de imágenes. Para que un algoritmo de identificación funcione perfectamente se debe hacer una detección precisa de la imagen captando o

reconociendo el rostro. El algoritmo no solo debe detectar el rostro para la identificación o verificación de personas, sino que tiene que tomar en cuenta otros aspectos los cuales podrían dificultar el proceso de detección del rostro como (López Pérez & Toro Agudelo, 2012):

- Pose y orientación del rostro.
- Tamaño del rostro.
- Presencia de lentes, barba, gorros, etc.
- Expresión de la cara.
- Problemas de iluminación.
- Condiciones de la imagen.
- Cantidad desconocida de caras en la imagen.

Métodos Basados en Rasgos Faciales: buscan encontrar aquellas características presentes en cualquier rostro: ojos, cejas, labios, boca, mentón, líneas de contorno (Serratos, 2014).

Métodos Basados en la Imagen: aplican herramientas generales de reconocimiento de patrones para sintetizar un modelo a partir de un conjunto de imágenes de entrenamiento. Trabajan con la imagen completa o una región de esta sin buscar rasgos faciales de forma localizada (Arguello Fuentes, 2011).

❖ **Métodos Basados en Rasgos Faciales**

Estos métodos explotan propiedades aparentes de la cara tal como el color de la piel y la geometría facial. La detección de la cara se resuelve manipulando medidas de distancias, ángulos y áreas de los rangos visuales. Se pueden definir tres ramas dentro del conjunto de métodos basados en rasgos faciales (Serratos, 2014):

- Análisis de bajo nivel: son técnicas que trabajan directamente con los píxeles, principalmente hay dos:
 - ✓ Basados en bordes: buscan bordes, los afinan, etiquetan y finalmente buscan estructuras similares a las de una cara.

- ✓ Basados en regiones: aprovechan el hecho de que hay zonas más oscuras que el resto de la cara (cejas, pupilas, etc.). Separan la imagen en regiones. Localizan la cara comparando la distribución de las regiones presentes con la distribución de regiones tipo de una cara.
- Análisis de rasgos faciales: Dado que el análisis a bajo nivel puede brindar información parcialmente correcta o incompleta, esta familia de métodos busca encontrar implícitamente los rasgos faciales. Se basan fuertemente en las relaciones geométricas que cumplen los diferentes rasgos presentes en una cara. Existen dos grandes aproximaciones al respecto:
 - ✓ Búsqueda de rasgos: intentan realizar una búsqueda ordenada de los rasgos característicos de una cara. Por ejemplo, primero buscan la frente, luego los ojos, continúan con la nariz, etc. Se basan en hipótesis sobre la pose y orientación de la cara y utilizan heurística.
 - ✓ Análisis de Constelaciones: buscan levantar algunas de las hipótesis de los métodos anteriores sobre la pose y orientación de la cara. Se basan en un análisis probabilístico de la ubicación de ciertos puntos característicos (constelación) de una cara.

❖ **Métodos Basados en la Imagen**

En estas técnicas, por el contrario, el objeto de estudio es la imagen misma. Se trabaja directamente con una representación de la imagen a la que se le aplica algoritmos de entrenamiento y análisis.

Los métodos basados en rasgos faciales son muy débiles a cambios que se puedan presentar en las imágenes, por ejemplo, más apariciones de rostros o cambios en el ambiente (iluminación, fondo). Para resolver este problema surgieron las siguientes técnicas (Arguello Fuentes, 2011):

- Sub-espacios lineales. Esta técnica se fundamenta en representar las imágenes de los rostros en espacios lineales buscando a que espacio lineal pertenece mediante un análisis estadístico, entre los cuales se destacan: PCA (análisis de componentes principales), LDA (análisis de discriminante lineal), ICA (análisis de componentes independientes).
- Redes neuronales. Es una técnica de mayor uso para el reconocimiento de patrones ya que se puede verificar si una imagen contiene un rostro. Esto se logra entrenando las redes neuronales con imágenes que contienen rostros y otras imágenes que no. Además, dan solución al problema de saber que si un objeto interfiere con la imagen del rostro.

El proceso de reconocimiento de caras consiste en tomar una imagen de dos dimensiones, a filas y b columnas, a la que se transforma en un vector unitario contenido en un espacio de imágenes n-dimensional ($n = a \times b$). Luego se le subtrae la imagen promedio y se proyecta el vector resultante en un sub espacio de menor dimensión utilizando uno de los métodos de reducción de dimensión (extracción de características) (Ottado, 2010). Esta proyección es comparada con la proyección de un conjunto de imágenes de una base de datos. La clase del vector más similar, utilizando algún criterio de similitud, es el resultado del proceso de reconocimiento, identificando así a una persona con visión artificial.

2.3. Reconocimiento facial

Los sistemas de reconocimiento de rostros son un problema que aún es tema de investigación (Kanade & Hebert, 2015), ya que algunos factores pueden afectar la efectividad del reconocimiento facial, tales como gestos, elementos que cubran la cara, iluminación, distancia hacia la cámara, etc. (Guerrero, 2012).

El avance más enfático en este campo, es la aplicación de algoritmos matemáticos, requiriendo menos de cien valores para cifrar correctamente una imagen facial, y el estudio de patrones del iris como método de identificación (Guerrero, 2012).

Para resolver el problema se instaló el sistema operativo Raspbian, las librerías de OpenCV (Open source Computer Vision library), el lenguaje Linux para la tarjeta

Raspberry Pi 3 que es la encargada de realizar el trabajo de visión artificial.

Para ejecutar correctamente los algoritmos de Eigenfaces y Fisherfaces se requiere implementar una técnica de reducción de la dimensión de los datos generados por las imágenes, análisis de componentes principales PCA (Principal Component Analysis) para el caso de eigenfaces y fisherfaces y análisis de discriminantes lineales LDA (Linear Discriminat Analysis) para el caso de Fisherfaces (Ortiz, 2014).

2.3.1. Eigenfaces

Este método realiza una proyección lineal del espacio de imágenes a un espacio de características de menor dimensión (Chichizola, De Giusti , & Naiouf, 2014). Esta reducción se realiza utilizando la técnica PCA la cual toma aquella proyección lineal que maximiza la dispersión de todas las imágenes proyectadas (Ottado, 2010).

En primer lugar se considera un conjunto de N imágenes con valores en el espacio de imágenes n -dimensional

$$\{x_i\} \quad i = 1, 2, \dots, N \quad (1)$$

Se asume además que cada una de las imágenes pertenece a una de las c clases $\{X_1, X_2, \dots, X_c\}$. Asimismo se considera una transformación lineal que lleva el espacio de imágenes original de n dimensiones al espacio de características de dimensión m , donde $m < n$. Los nuevos vectores de características $y_k \in \mathfrak{R}^m$ son definidos por la siguiente transformación lineal

$$y_k = W^T x_k \quad k = 1, 2, \dots, N \quad (2)$$

Donde $W \in \mathfrak{R}^{n \times m}$ es una matriz con columnas ortonormales. Se define además la matriz de la distribución S_T como

$$S_T = \sum_{k=1}^N (x_k - \mu)(x_k - \mu)^T \quad (3)$$

Donde $\mu \in \mathfrak{R}^n$ es la medida de todas las imágenes de (1). Luego de aplicar la transformación lineal W^T , la distribución de los vectores de las características $\{y_1, y_2, \dots, y_N\}$ es $W^T S_T W$. Se toma aquella proyección W_{opt} que maximiza el determinante de la distribución total de la matriz de las imágenes proyectadas, esto es

$$\begin{aligned} W_{opt} &= \arg \max_W |W^T S_T W| \\ &= [w_1, w_2, \dots, w_m] \end{aligned} \quad (4)$$

Donde $\{w_i | i = 1, 2, \dots, m\}$ es el conjunto de vectores propios n -dimensionales de S_T correspondiente a los mayores m vectores propios. Dichos vectores propios tienen la misma dimensión que las imágenes originales y se les denomina eigenfaces.

2.3.2. Fisherfaces.

El método Fisherfaces (Yuan & Qi, 2009) utiliza el Discriminante Lineal de Fisher (FLD) para la reducción de dimensión. Este método selecciona el W de la ecuación (1) de manera que el cociente entre la distribución entre clases y la distribución intra-clases sea máxima (Belhumeur & Hespanha, 2010). Para esto se define la matriz S_B de distribución entre clases como

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu) (\mu_i - \mu)^T \quad (5)$$

Y la matriz S_W de distribución intra-clases

$$S_B = \sum_{i=1}^c \sum_{x_k \in X_i} N_i (\mu_i - \mu) (\mu_i - \mu)^T \quad (6)$$

Donde μ_i es la imagen media de la clase X_i , y N_i es el número de imágenes en la clase X_i . Si la matriz S_W es no singular, la proyección W_{opt} se elige como la matriz con columnas orto normal que maximiza el cociente del determinante de la matriz de distribución entre clases de las imágenes proyectadas y el determinante de la matriz de la distribución intra-clases de las imágenes proyectadas, esto es

$$\begin{aligned} W_{opt} &= \arg \max_w \left| \frac{W^T S_B W}{W^T S_W W} \right| \\ &= [w_1, w_2, \dots, w_m] \end{aligned} \quad (7)$$

Donde $\{w_i | i = 1, 2, \dots, m\}$ es el conjunto de valores propios de S_B y S_W correspondiente a los m mayores valores propios $\{\lambda_i | i = 1, 2, \dots, m\}$, esto es

$$S_B w_i = \lambda_i S_W w_i \quad i = 1, 2, \dots, m \quad (8)$$

Se observa entonces, que a lo sumo se tienen $c - 1$ valores propios distintos de cero, y por lo tanto el límite superior de m es $c - 1$, donde c es el número de clases. Para el problema de reconocimiento de caras, se tiene que la matriz $S_W \in \mathfrak{R}^{n \times n}$ es siempre singular, dado que el rango de S_W es a lo sumo $N - c$, y en general, el número de imágenes de entrenamiento: N , es mucho más chico que el número de píxeles de cada imagen: n . Por lo tanto puede ser posible elegir una matriz W tal que la distribución

intra-clases de las imágenes proyectadas pueda ser exactamente cero.

Como alternativa entonces, al criterio establecido en la ecuación (7), se proyecta el conjunto de imágenes a un espacio de menor dimensión, de manera que la matriz resultante de la distribución intra-clases S_W es no singular. Utilizando PCA se realiza la reducción de dimensiones del espacio de características a $N - c$ y luego, aplicar FLD definido en (7) para reducir la dimensión a $c - 1$. De esta manera W_{opt} es dado por

$$W_{pca}^T = W_{fld}^T W_{pca}^T \quad (9)$$

Donde

$$W_{pca} = \arg \max_w |W^T S_T W| \quad (10)$$

$$W_{fld} = \arg \max_w \left| \frac{W^T W_{pca}^T S_B W_{pca} W}{W^T W_{pca}^T S_W W_{pca} W} \right| \quad (11)$$



(a)



(b)

Figura 2: Primeras siete Eigenfaces (a) y Fisherfaces (b), obtenidas con un conjunto de imágenes de la base Extended Yale Face Database B.

Fuente: (Ottado, 2010).

En la Figura 2 se ilustra la ventaja de FLD sobre PCA para el caso de un problema de clasificación en el que las muestras de cada clase yacen próximas a un sub-espacio lineal.

El problema consiste de dos clases en el cual las muestras de cada clase han sido ligeramente desplazadas en una dirección perpendicular al sub-espacio lineal, por lo

que cada clase yace próxima a una línea que pasa por el origen en el espacio de características de dos dimensiones (Ottado, 2010). Ambos métodos, PCA y FLD, han sido utilizados para proyectar las muestras en un espacio de una dimensión. Se observa que PCA proyecta las muestras de manera que las clases no puedan ser linealmente separables. Por lo tanto FLD consigue una mayor dispersión entre clases.

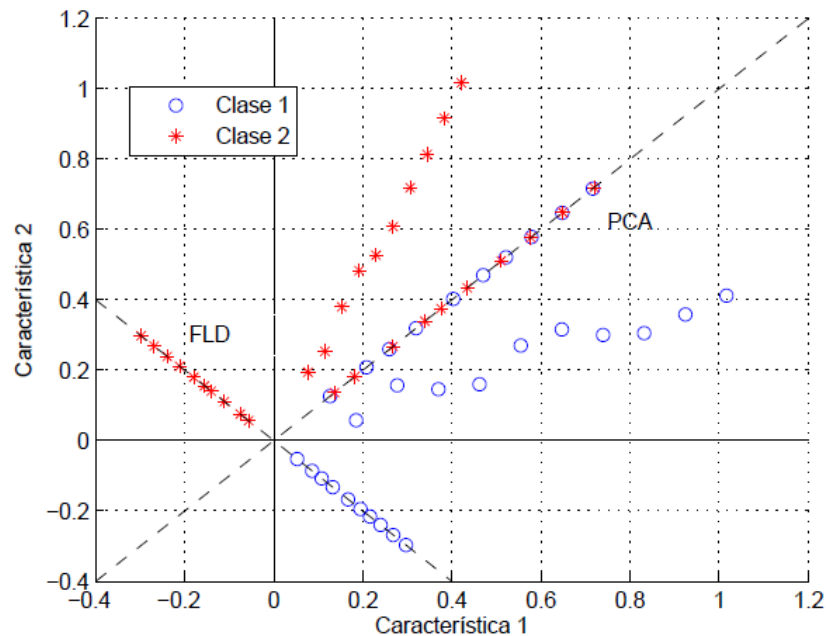


Figura 3: Comparación de PCA y FLD para un problema de dos clases.

Fuente: (Belhumeur & Hespanha, 2010).

2.3.3. Algoritmo LBPH.

Para este algoritmo se puede destacar que el operador LBP Básico es un operador eficiente computacional (Esparza Franco, & Tarazona Ospina, 2015). Tomando cada píxel como un umbral, el operador transfiere su vecindad 3×3 en un código binario de 8 bits, como se muestra en la Figura 4 (a). Más tarde, en, el operador LBP que se extiende un número arbitrario de píxeles interpolados bilineales en un círculo con tamaño arbitrario se usan como píxeles vecinos, en lugar de su vecindad 3×3 , como se muestra en Figura 4 (b).

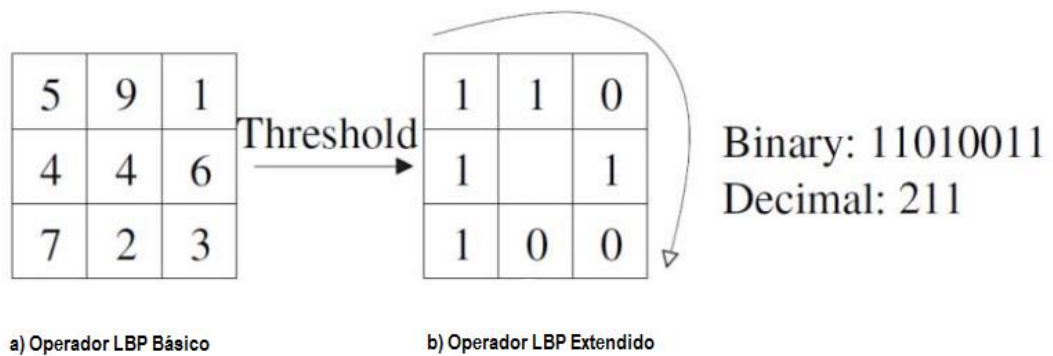


Figura 4: Operadores LBP.

Fuente: http://media.cs.tsinghua.edu.cn/~ahz/papers/ICB07_demographic.pdf

El método de patrones binarios locales fue diseñado para la descripción de texturas (Esparza Franco, & Tarazona Ospina, 2015). El uso de descripciones locales en algunas regiones del rostro aportan más información que otras, por lo que los descriptores de texturas tienden a promediar la información que describen, lo cual no es conveniente al describir rostros puesto que mantener la información de las relaciones espaciales es importante (Salazar Espinoza, 2016).

Para formar la descripción global, la imagen del rostro es dividida en diferentes regiones, a las que se les aplica un histograma con el que se obtiene el operador LBPH que describe información independiente por región. Estas descripciones locales son entonces concatenadas para construir una descripción global del rostro (Álvarez Corrales, 2013).

El método de LBPH asigna etiquetas a cada uno de los píxeles de la imagen tomando en cuenta la distribución de los vecinos. Estos son los pasos que el LBPH realiza para su respectivo reconocimiento de imágenes.

Para el operador básico LBP es la llamada LBP uniforme, que se encontró que la propiedad fundamental de la textura de la imagen local. A LBP se llama uniforme si no hay más de dos 0/1 o 1/0 transiciones bit a bit en su código binario, siendo considerado como un código circular (Salazar Espinoza, 2016).

El operador LBP extendido es donado como LBP, donde P es el número de píxeles interpolados bilineales, R es el radio del círculo vecino y u2 significa criterio uniforme.

La contribución de patrón uniforme a LBP y la LBP es de aproximadamente 87,2% y 70,7%, respectivamente. Es decir, los patrones uniformes toman un porcentaje mayoritario de todos los patrones. Como resultado, cada patrón uniforme se da una etiqueta única y todas las demás minorías se les dan una etiqueta mutua en el cálculo del histograma (Zhiguang & Haizhou , 2007).

Si suponemos que S y M son dos diferentes histogramas, entonces la distancia de Chi cuadrado se la puede definir como:

$$S^2(S, M) = \sum_{i=1}^n \frac{(S_i - M_i)^2}{S_i + M_i} \quad (12)$$

Donde n es el número de elementos en el histograma. Chi distancia cuadrado es una efectiva medida de similitud entre un par de histogramas, por lo tanto, es adecuado para el vecino más cercano. Sin embargo, encontramos un par de muestras similares no tiene sentido en la mayoría de problemas de clasificación binario. En esta sección, se quiere hallar una óptima plantilla de histograma M como la plantilla de referencia para todas las muestras positivas o negativas para calcular una distancia de Chi cuadrado para el pedido.

$$FDR = \frac{(\mu_+ - \mu_-)^2}{\sigma_+^2 + \sigma_-^2} \quad (13)$$

Donde μ y σ es el valor de la media y la varianza de Chi cuadrado. Con el fin de hallar el histograma óptima, primero se inicializa M como el valor medio de histogramas positivos, y luego utilizamos el método decente empinada para hallar una solución óptima. Dada conjunto de muestras $S = (h, y), \dots, (H, y)$, donde h es la característica LBPH, $ey = \pm 1$ es la etiqueta de clase μ y σ de muestras positivas y negativas ilustrados por $y = 1$ e $y = -1$, respectivamente, se pueden escribir como

$$\mu_y = \frac{1}{N_y} \sum_{k=1, y_k=y}^m x^2(h_k, M) \quad (14)$$

$$\sigma_y^2 = \frac{1}{N_y} \sum_{k=1, y_k=y}^m (x^2(h_k, M) - \mu_y)^2 \quad (15)$$

Para el histograma plantilla de n -elementos, cuyos artículos resumir en 1, la anterior $n - 1$ elementos se supone que son independientes, y su derivada parcial de μ y σ se puede escribir como:

$$\frac{\partial \mu_y}{\partial M_i} = \frac{1}{N_y} \sum_{k=1, y_k=y}^m \frac{-4h_k^2(k)}{(h_k(i) + M_i)^2} + 1 \quad (16)$$

$$\frac{\partial \mu_y}{\partial M_i} = \frac{2}{N_y} \sum_{k=1, y_k=y}^m (x^2(h_k, M) - \mu_y) \left(\frac{-4h_k^2(k)}{(h_k(i) + M_i)^2} + 1 - \frac{\partial \mu_y}{\partial M_i} \right) \quad (17)$$

Como resultado, el gradiente de Fisher discrimina relación se puede calcular como sigue

$$\frac{\partial FDR}{\partial M_i} = \frac{2(\mu_+ - \mu_-)}{\sigma_+^2 + \sigma_-^2} \left(\frac{\partial \mu_+}{\partial M_i} - \frac{\partial \mu_-}{\partial M_i} \right) - \frac{(\mu_+ - \mu_-)^2}{(\mu_+ + \mu_-)^2} \left(\frac{\partial \sigma_+^2}{\partial M_i} + \frac{\partial \sigma_-^2}{\partial M_i} \right) \quad (18)$$

Por este medio, un M óptima podría ser encontrado por búsqueda iterativa, y se utiliza como la plantilla de referencia para el parche textura local dada. Dividimos el dominio principal característica de Chi cuadrado de la distancia a partir de muestras de formación a la plantilla de referencia en 32 contenedores.

Con respecto a la distribución de muestras de entrenamiento 'de la distancia Chi cuadrado, se usa un débil clasificador basado en LUT. Su salida en cada dominio puede ser definida como:

$$\forall h \in H_i, f(h) = \frac{1}{2} \left(\frac{W_+^i + \varepsilon}{W_-^i + \varepsilon} \right), i = 1, 2, \dots, 32. \quad (19)$$

Donde W es el peso suma de muestras en el dominio H i -ésimo y ε es una pequeña constante positiva (Zhiguang & Haizhou, 2007).

2.4. Cascada de clasificadores Haar.

El Método de Viola & Jones, es un método de detección de objetos en una imagen, desarrollado por los investigadores Paul Viola y Michael Jones en el año 2001.

El clasificador Haar es un método desarrollado por Viola & Jones y es una versión del algoritmo "Adaboost". Viola & Jones propusieron un esquema basado en una cascada de clasificadores fuertes para cada parte del rostro, cada etapa corresponde a un clasificador fuerte y está entrenada con todos los ejemplos que la etapa anterior no ha clasificado correctamente más algunos nuevos (Guevara & Echeverry, 2008).

Por tanto, en la etapa de entrenamiento, cada etapa se entrena con un conjunto óptimo

de características capaces de detectar cada vez ejemplos más complicados; es decir, las primeras etapas se encargan de descartar sub-imágenes que son muy diferentes de una cara, mientras que las últimas etapas pueden rechazar ejemplos mucho más complejos como pueden ser globos, pelotas, dibujos, etc. (Urtiaga Abad, 2014).

El clasificador Haar está basado en árboles de decisión con entrenamiento supervisado estrictamente. El entrenamiento se realiza determinando una serie de características faciales basadas en sumas y restas de los niveles de intensidad de la imagen como luz y contraste. Basándose en estas características locales se puede obtener un detector de objetos robusto (López Pérez & Toro Agudelo, 2012).

También se denominan estos clasificadores mediante el nombre de cascada, ya que el resultado del clasificador es el fruto de varios clasificadores más simples o etapas.

El candidato a objeto dentro de la imagen a procesar debe superar todas las etapas para ser aceptado. Después de que el clasificador ha sido entrenado, puede ser aplicado a una región de interés de una imagen de entrada. El clasificador devuelve un “1” si la región contiene el objeto, y “0” en otro caso (Navas, 2013).

El clasificador está diseñado para que pueda ser fácilmente redimensionado ya sea automáticamente o manual mediante programación, para que sea capaz de encontrar los objetos de interés en diferentes tamaños, lo cual es mucho más eficiente que redimensionar la propia imagen.

La palabra “cascada” en el nombre del clasificador significa que el clasificador resultante consiste en varios clasificadores más simples que son aplicados uno tras otro a una región de interés hasta que en alguno de los pasos el candidato es rechazado o todos los pasos son satisfactorios (Network, 2014).

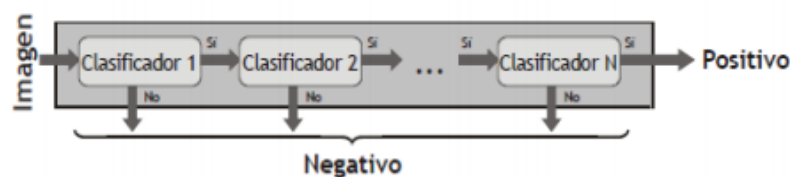


Figura 5: Cascada de detectores propuesta por Viola y Jones.

Fuente: <http://dSPACE.unl.edu.ec/jspui/bitstream/123456789/14237/1/>

2.5. Como trabajan los algoritmos.

De acuerdo a la configuración de cada algoritmo, este trabaja de diferente manera dependiendo a de qué manera fue programado su trabajo, se encuentran los siguientes:

2.5.1. Los sistemas de reconocimiento facial automático.

Un sistema de reconocimiento facial automático (Moreno Díaz, 2004) realiza la siguiente función: dada una o varias imágenes (estáticas o en movimiento) de una cara desconocida, selecciona entre las caras registradas en su base de datos (de personas conocidas), aquella (o aquellas) con un mayor grado de similitud o parecido, devolviéndose la identidad de ésta.

El sistema producirá un fallo en el reconocimiento cuando al presentar una entrada correspondiente a un individuo que se desea reconocer, da como resultado una identidad falsa. Si se desea otorgar al sistema capacidad para que si se le presenta una cara desconocida por él (no registrada), sea capaz de indicar que esa cara no es ninguna de las registradas, es posible establecer un umbral de decisión (Hernández, 2010), de tal manera que si el grado de similitud es muy pequeño (si la distancia entre la cara a reconocer y la cara más parecida de la base de datos de caras supera dicho umbral), indicará que no es posible identificar a ese individuo, pues ese individuo no es conocido por el sistema.

2.5.2. Tipos de errores y su medición

De acuerdo a la variabilidad de los rasgos biométricos, un sistema de reconocimiento automático puede generar dos tipos de errores (Vázquez López, 2014):

- a) Error falso positivo (EFP): Cuando una muestra desconocida es declarada erróneamente como conocida.

- b) Error falso negativo (EFN): cuando una muestra conocida es declarada falsamente como desconocida.

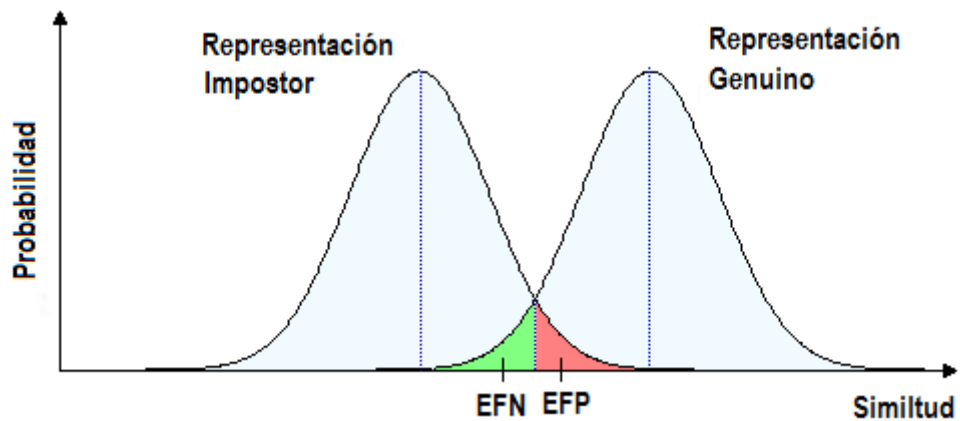


Figura 6: Distribución de probabilidad de un usuario impostor y un usuario genuino.

Fuente: (Vázquez López, 2014).

La Figura 6, representa la distribución de probabilidad de un usuario impostor y uno genuino. Las dos distribuciones de probabilidad se traslapan en algún punto de la gráfica creando un área común que define la tolerancia o sensibilidad del sistema (Vázquez López, 2014). Eligiendo un valor umbral T situado en el área de tolerancia se definen los EFN y EFP, con un valor bajo el sistema tendrá una mayor tolerancia EFP; si se le asigna un valor alto, el sistema tendrá una menor tolerancia al error de tipo falso Positivo haciéndolo más robusto frente intentos de suplantación de identidad. En aplicaciones como los sistemas de seguridad, se da mayor importancia al error del tipo falso positivo, es decir deben ser robustos frente a estos errores y no dar acceso a personas extrañas o a posibles impostores. Dejando con menor importancia los errores del tipo falso negativo.

2.6. Reconocimiento de Patrones.

Con el paso del tiempo los seres humanos, como demás organismos, han desarrollado importantes mecanismos y habilidades para obtener la información del medio y a partir de ello tomar decisiones. La capacidad de reconocer diferentes situaciones y tomar decisión pertinentes es una capacidad inherente de los seres vivos, aparentemente esta acción se aprecia como una simple tarea de sentido común.

Los seres vivos obtienen la información del medio a través de los sentidos como la vista, el tacto, el gusto, el olfato y la audición; en tanto los sistemas inteligentes la obtienen a través de sensores y presentan los datos en forma conveniente para su procesamiento e interpretación en ordenadores. Los datos registrados son llamados

patrones y éstos pueden ser representados como señales, imágenes o tablas de datos. El reconocimiento de patrones (RP) se define como el proceso de la clasificación automática de un objeto, físico o abstracto, en una o más clases con una intervención mínima del ser humano. La clasificación se basa en el conocimiento a priori o en la información extraída de los patrones (Delbracio & Mateu, 2006).

Reconocimiento de Patrones es una área de la tecnología conocido como Aprendizaje de Maquinas (Machine Learning) o Aprendizaje Automático. El único propósito de este método es el clasificar un grupo de patrones conocido como conjunto de pruebas en dos o más clases de categorías. Esto es logrado al calcular las categorías del conjunto en prueba comparándolo con un conjunto de entrenamiento (previo) o training set. Un clasificador dado mide la distancia entre varios puntos dados (compara), para saber cuáles puntos son más cercanos a la meta en un modelo parametrizado.

Con el paso del tiempo los seres humanos, como demás organismos, han desarrollado importantes mecanismos y habilidades para obtener la información del medio y a partir de ello tomar decisiones. La capacidad de reconocer diferentes situaciones y tomar decisión pertinentes es una capacidad inherente de los seres vivos, aparentemente esta acción se aprecia como una simple tarea de sentido común.

Los seres vivos obtienen la información del medio a través de los sentidos como la vista, el tacto, el gusto, el olfato y la audición; en tanto los sistemas inteligentes la obtienen a través de sensores y presentan los datos en forma conveniente para su procesamiento e interpretación en ordenadores. Los datos registrados son llamados patrones y éstos pueden ser representados como señales, imágenes o tablas de datos. El reconocimiento de patrones (RP) se define como el proceso de la clasificación automática de un objeto, físico o abstracto, en una o más clases con una intervención mínima del ser humano. La clasificación se basa en el conocimiento a priori o en la información extraída de los patrones (Delbracio & Mateu, 2006).

Reconocimiento de Patrones es una área de la tecnología conocido como Aprendizaje de Maquinas (Machine Learning) o Aprendizaje Automático. El único propósito de este método es el clasificar un grupo de patrones conocido como conjunto de pruebas

en dos o más clases de categorías. Esto es logrado al calcular las categorías del conjunto en prueba comparándolo con un conjunto de entrenamiento (previo) o training set. Un clasificador dado mide la distancia entre varios puntos dados (compara), para saber cuáles puntos son más cercanos a la meta en un modelo parametrizado.

2.6.1. Patrones.

Un patrón es la entidad que puede ser identificable de acuerdo a sus características o atributos, éste puede ser abstracto o concreto (Carrasco Ochoa, 2013). Los patrones abstractos son ideas conceptuales mientras que un patrón concreto es la representación física de un objeto, por ejemplo los símbolos (letras, caligrafías), imágenes digitales, imágenes tridimensionales, firmas, señales de voz, electrocardiogramas, ondas sísmicas, entre otros.

2.6.2. Similitud

La noción de similitud, es la pieza angular en el proceso del RP, se refiere a los valores parecidos de un atributo en dos o más objetos. En el proceso de reconocimiento, se evalúa la similitud entre un objeto y un modelo (prototipo) que idealiza a la clase a la que pertenece.

Básicamente existente tres enfoques para evaluar la similitud de los patrones (Carrasco Ochoa, 2013):

1. La clasificación
2. El análisis de regresión
3. La descripción

2.6.3. Diseño de un sistema de reconocimiento de patrones.

Un sistema de reconocimiento de patrones, independientemente del paradigma que implementa, está conformado por distintos módulos que operan de manera sistemática sobre los patrones. La Figura 4 corresponde a la estructura de un sistema típico de un sistema de reconocimiento de patrones (Vázquez López, 2014).

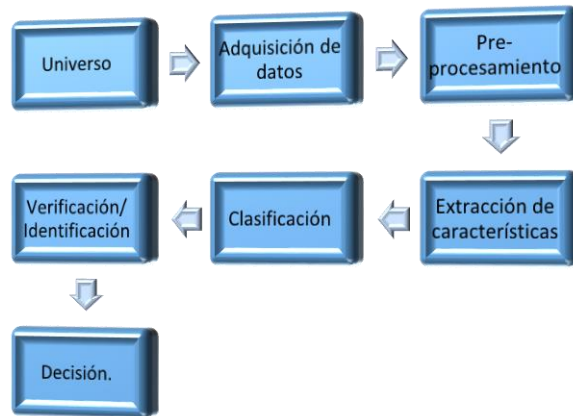


Figura 7: Esquema general de un sistema de reconocimiento de patrones.

Fuente: Autores

Módulo de adquisición de datos: El proceso inicia con la adquisición de datos registrando las variables físicas y las representa de forma conveniente para su procesamiento y análisis en ordenadores. La calidad de los datos adquiridos depende de las características del equipo tales como la resolución y el diseño; también influyen factores los ambientales o las variaciones de la interacción del usuario.

Módulo de pre-procesamiento: Esta tarea se realiza con la finalidad de mejorar la calidad de los datos adquiridos. Realiza tareas tales como la normalización y eliminación de los datos irrelevantes en las muestras.

Módulo de extracción de características: El objetivo de este módulo es generar información compacta pero significativa de los datos. La información relevante es almacenada en los vectores de características.

Módulo de clasificación: En el proceso de clasificación, los vectores de características son analizados bajo un enfoque para definir las clases y posteriormente asignar un objeto desconocido a una de ellas.

Módulo de post-procesamiento: En esta etapa se busca evaluar el resultado de la clasificación y determinar si un patrón fue asignado a la categoría correcta. Una manera de mejorar los resultados de la clasificación es la utilización de múltiples clasificadores.

2.7. Incidencia de la luz, colores, posición, distancia y accesorios.

Para el desarrollo de sistemas de reconocimiento facial automático es necesario contar con Bases de Datos de imágenes de manera que sea posible su evaluación. Éstas han de ser suficientemente amplias, del mismo tamaño para todas las imágenes y deben plasmar las posibles variaciones (en cuanto a pose, cambios de iluminación, expresiones faciales, etc.) entre las diferentes imágenes de un mismo individuo respecto de las que se desee comprobar la robustez de los sistemas experimentados con ellas.

Las posibles variaciones entre imágenes de un mismo individuo pueden deberse a: cambios en la pose (posición y orientación de la cara en la imagen), cambios en la iluminación (variaciones en la posición de la fuente de luz, variaciones en la intensidad de la luz, etc.), variaciones en la expresión facial, oclusiones (gafas, pañuelo, barba, bigote, peinado, etc.), maquillaje, edad, etc.

Para solucionar esto, se crea la base de datos de cada individuo con la misma cantidad de luz, especialmente la que está directamente hacia el rostro. El sistema será capaz de tomar las fotos del mismo tamaño cada una sin importar la distancia la que se encuentra el usuario de la cámara.

2.8 OpenCV

OpenCV significa Open Source Computer Vision Library; por lo tanto, es una librería de tratamiento de imágenes, destinada principalmente a aplicaciones de visión por computador en tiempo real. Una de las ventajas principales es que puede funcionar en muchas plataformas, existen versiones para Windows, Linux y MacOs.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones expresadas.

OpenCV es multiplataforma, Existiendo versiones para GNU/Linux, Mac OSX y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica (Álvarez Corrales, 2013).

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se logra, realizando su programación en código *Python* optimizados, aprovechando además las capacidades que proveen los procesadores multi-núcleo.

2.8.1. Instalación de OpenCV en Raspbian

Actualmente se tiene disponible una versión de OpenCv un tanto genérica para instalarlo sobre Raspbian lo cual implica que no se va a poder tener muchas opciones que normalmente ofrecería una versión completa por lo que Raspbian al estar basado sobre debían y estar con el soporte de este, es posible instalar una versión completa disponible en la nube.

Los pasos que se deberían hacer son:

- Actualizar las cabeceras del sistema operativo.
- Instalar las dependencias necesarias que usara OpenCV.
- Descargar y descomprimir OpenCV.
- Crear una carpeta built dentro de la descompresión y ejecutar cmake y make.

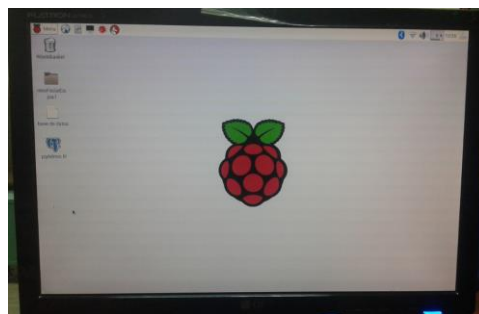


Figura 8: Sistema operativo instalado.

Fuente: Autores

2.8.2. Manipulación de imágenes

OpenCV cuenta con muchas funciones predefinidas las cuales podemos hacer referencia, estas funciones pueden ocuparse para manipular y editar imágenes, ya sea como reducir, filtrar, incluso algoritmos complicados.

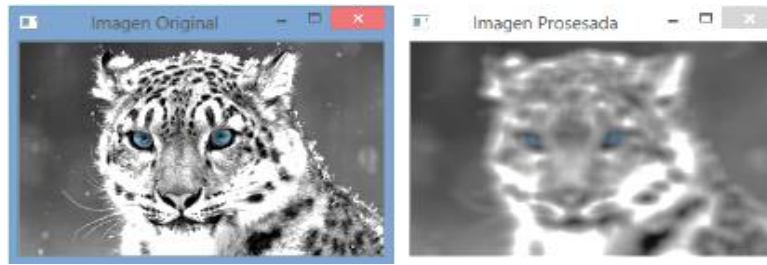


Figura 9: Función GasussianBlur.

Fuente: <http://4.bp.blogspot.com/-zFk5fzn7qZc/>

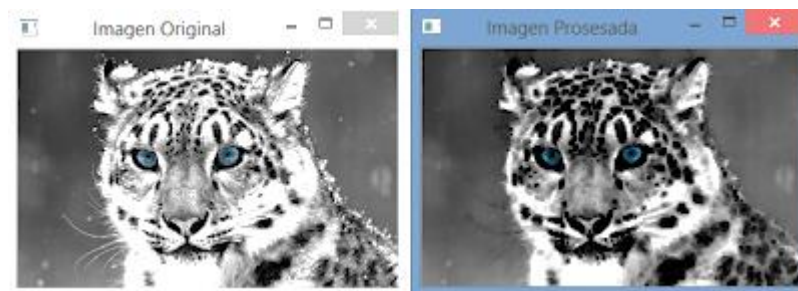


Figura 10: Función Erode

Fuente: <http://1.bp.blogspot.com/-K0tBysflT4M/UZ959mPxlxI/AAAAAAAAAPw/1WiMcA5IOdM/s400/erode.png>

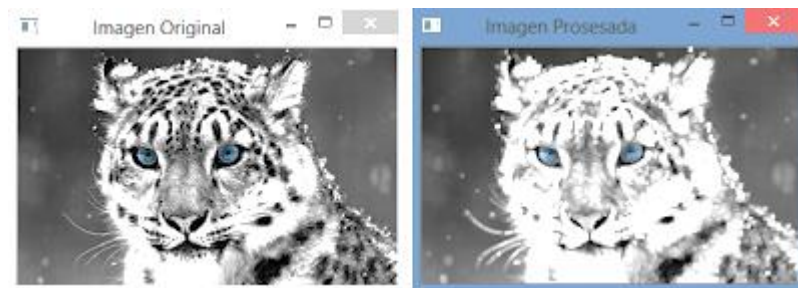


Figura 11: Función Dilate

Fuente: <http://3.bp.blogspot.com/-yJU7vxbyE7E/UZ95-Kr1FtI/AAAAAAAAAP4/OnZQrfhnVsM/s1600/dilate.png>

En muchas ocasiones es necesario guardar las imágenes que se han procesado, para lo cual podemos usar las líneas que aclaran el tipo de compresión que se usara para guardar la imagen, formato, matriz que la representa, nombre de la imagen a guardar, etc.

2.8.3. Acceso a la cámara y obtención de imágenes

El acceso a la cámara es muy importante en el procesamiento de imágenes ya que muchas de las aplicaciones así lo requieren.

Habría tres tipos de cámaras normalmente disponibles de acuerdo a su conexión y forma de introducir la información dentro del programa y que de estas puede haber variantes.

2.8.3.1. Cámaras Web USB

Este tipo de cámara se conectan físicamente al puerto o conector USB, transmitiendo la información de forma paralela, dependiendo de la resolución de la cámara dependerá la cantidad de información a transmitir y por defecto de la latencia de la comunicación, para lo cual se puede contar con puertos USB mucho más rápidos como los de ella versión 3.0

Existen cámaras estéreo que se conectan por USB la desventaja podrían ser que se divide el ancho del puerto reduciendo a la mitad la velocidad de transmisión por cada cámara.

2.8.4. Detectar rostro.

El algoritmo de detección de Viola & Jones en sí tiene dos etapas, primero una etapa de entrenamiento, en la que a una serie de filtros en cascada se les pasan unos patrones positivos (que coinciden con el objeto buscado) y negativos (no tienen el objeto) de forma que el sistema aprenda y se forme un modelo de las características de Haar del objeto a detectar (Parra Barrero, 2015).

La clase que ofrece OpenCV para detección de caras, es la clase `cv: CascadeClassifier`. La clase `CascadeClassifier` tiene dos funciones principales que son las que permiten cargar las características de Haar del objeto a detectar, y otra para la detección propiamente dicha (Network, 2014). Concretamente, la función que permite cargar las características de Haar es la función `cvLoad`, que necesita como parámetro de entrada el fichero xml que contiene la información generada en un entrenamiento previo.

OpenCV viene ya con unos ficheros entrenados con miles de patrones para detección de caras, manos, ojos, personas, etc., por lo que no hace falta realizar un entrenamiento de patrones, sino utilizar los ficheros que trae OpenCV.

OpenCV tiene incluido en sus ficheros clasificadores para varias detecciones, ya sea para rostro frontal, rostro de perfil, ojos, boca, nariz, etc (Haar Carcades, 2016). Es así

que para la detección frontal de rostros, se utilizó el Clasificador Haar Cascade que vienen con OpenCV: haarcascade_frontalface_alt.xml.

2.8.5. Inconvenientes de las OpenCV.

Dadas las grandes posibilidades que ofrece OpenCV para el tratamiento de imágenes, calibración de cámaras, y otras muchas aplicaciones más como por ejemplo, para simular una prótesis ocular basada en un implante cortical y estudiar el funcionamiento de las retinas artificiales, etc.

Quizá de los pocos inconvenientes que se pueden encontrar en ella sea en el caso del seguimiento de objetos, en el cual, el principal inconveniente es que no ofrece un producto completo, tan sólo algunas piezas que sirven como base para montar sobre ellas un producto final. Sin embargo, la presencia de funciones muy interesantes, y las posibilidades ya comentadas que ofrece la librería hacen que estos inconvenientes no sean realmente significantes.

2.8.6. Estructura modular OpenCv.

OpenCV tiene una estructura modular. Cada módulo incluye varias librerías enfocadas a un objetivo concreto (OpenCv, 2016). Los módulos principales son los siguientes:

- **core:** el módulo que define las estructuras básicas y funciones que serán usadas por otros módulos.
- **Imgproc:** dedicado al procesado de imagen, contiene funciones como transformadas lineares o no lineales, transformaciones geométricas, conversiones entre espacios de color, histogramas, etc.
- **Video:** módulo enfocado a funciones varias de vídeo, como seguimiento de movimiento, extracción del fondo y algoritmos de seguimiento de objetos.
- **Objdetect:** funciones de detección de objetos, incluyendo clases predefinidas (por ejemplo ojos, boca, coches, etc).
- **Highgui:** este módulo sirve para añadir un interfaz sencillo a las aplicaciones de imagen y vídeo (botones, barras de desplazamiento, etc).

Entre las múltiples ventajas que ofrece trabajar con una librería como OpenCV está la de tener clases propias y funciones preparadas para el procesado de imágenes y vídeo. Las utilizadas en el presente proyecto se detallaran en el apartado implementación.

2.9. Sistema del automóvil a considerar.

Este vehículo cuenta con sistema de inyección multipunto que tiene un inyector por cada cilindro. La inyección del combustible es de manera indirecta donde los inyectores van ubicados en los colectores de admisión en una zona próxima, en la que se mezcla con el aire antes de entrar en la zona interior donde están los cilindros.

El sistema de alimentación de combustible consta de los siguientes elementos:

- Batería
- Relay de bomba de combustible
- Bomba de combustible
- Filtro de combustible
- Inyectores
- Regulador de combustible
- Deposito

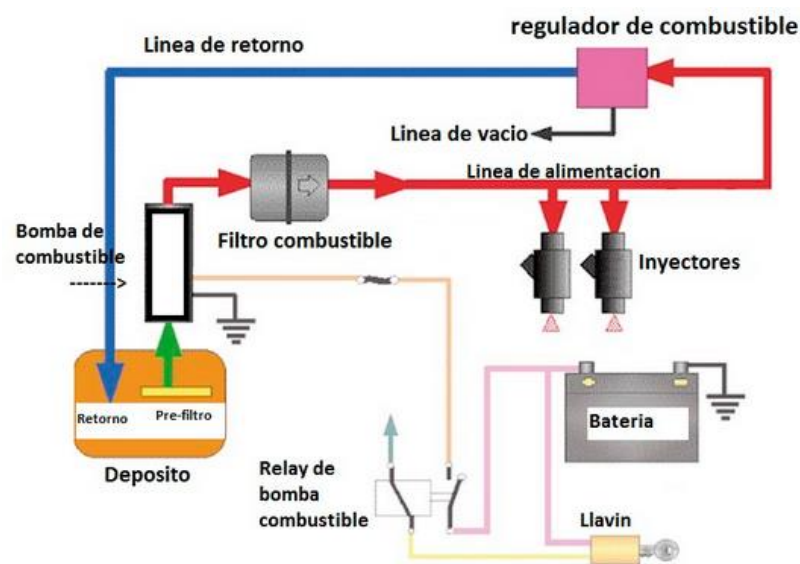


Figura 12: Sistema de alimentación de combustible del Chevrolet Aveo Activo.

Fuente: <http://mecanicabasicacr.com/inyeccion/medir-presion-de-combustible.html>

El componente sobre el cual va a intervenir el sistema de seguridad propuesto es la bomba de combustible, que estará interrumpida en su parte eléctrica por un relé que será activado por la señal de la tarjeta Raspberry Pi 3 una vez que el rostro del usuario sea reconocido o por la señal del Arduino Mega 2560 al digitar la clave de acceso rápido. De esta manera se podrá encender y hacer uso del automóvil.

CAPÍTULO 3

3. MARCO APLICATIVO

3.1. Diseño del sistema

En este punto se conocerá el diagrama de flujo del sistema el cual consiste en reconocer al usuario del vehículo que está registrado en la programación o digitar una clave de acceso para encender al vehículo.

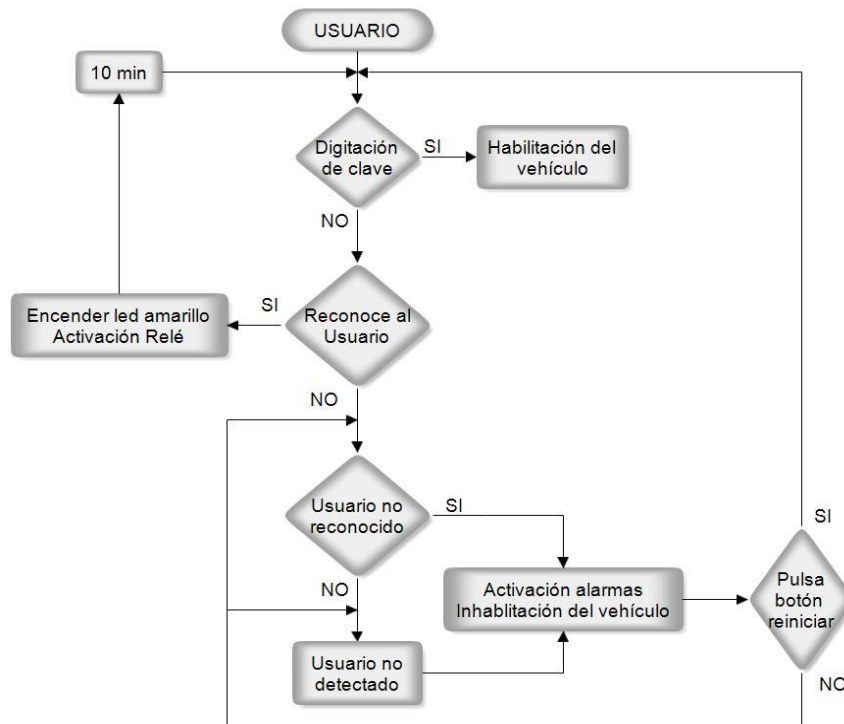


Figura 13: Diagrama de flujo del sistema.

Fuente: Autores.

Como se logra observar, el usuario o persona que esté en el asiento del piloto del vehículo estará sometido al reconocimiento facial cada 10 minutos (se podrá modificar en la programación), el cual al ser reconocido como usuario registrado se activará el relé que dará paso de voltaje para el funcionamiento de la bomba de combustible.

Al detectar a una persona que no esté registrada, el sistema dará la oportunidad de reconocerlo 2 veces más para evitar confusiones, después de estos dos intentos, si el sistema reconoce a una persona no registrada se activará un led rojo y una alarma a la vez por lo que el sistema desactiva el relé cortando el paso de voltaje hacia la bomba de combustible por lo que el vehículo se apagará hasta que se aplaste el botón de reinicio.

Por otra parte, al no estar ninguna persona enfocada a la cámara, se activara una alarma de “usuario no detectado” con un led rojo parpadeante o titileo de sonidos para avisar al conductor que debe enfocarse a la cámara, pues al no hacerlo, después de un cierto tiempo, el sistema inhabilitara el vehículo hasta reconocer un rostro.

Cabe recalcar que existe una clave de acceso para habilitar a la bomba de gasolina, esto es necesario pues para que un usuario que no esté registrado en la programación pueda hacer uso del vehículo sin problema al digitar la clave.

3.2. Selección de elementos para el sistema.

Una vez planteado perfectamente el proyecto, se procede a seleccionar los elementos esenciales para posteriormente armar los circuitos y evaluar el proyecto.

3.2.1. Raspberry Pi 3 Model B

La Raspberry Pi 3 es una placa base con unas dimensiones interesantes como sus especificaciones y la larga lista de aplicaciones que este pequeño aparato puede tener.



Figura 14: Raspberry Pi 3 Model B.

Fuente: <http://es.rs-online.com/web/p/kits-de-desarrollo-de-procesador-y-microcontrolador/8968660/>

La Raspberry Pi es una placa de ordenador simple; se ha desarrollado para fomentar y ayudar en la enseñanza de la programación y la informática. También es un excelente punto de partida para el desarrollo de proyectos para IoT (Internet de las cosas). El bajo coste y la naturaleza "plug-and-play" de Pi ofrece como resultado una placa accesible para todos y con numerosas opciones de conectividad. Pi es la herramienta experimental perfecta, tanto si desea utilizarla como ordenador de sobremesa como centro multimedia, servidor o dispositivo de supervisión/seguridad (Raspberry Pi,

2015). Los sistemas operativos basados en Linux funcionan en la Pi, lo que permite acceso ilimitado a software libre y descargas gratuitas.

Características y ventajas de Pi 3

- Chipset Broadcom BCM2837 a 1,2 GHz
- ARM Cortex-A53 de 64 bits y cuatro núcleos
- LAN inalámbrica 802.11 b/g/n
- Bluetooth 4.1 (Classic y Low Energy)
- Coprocesador multimedia de doble núcleo Videocore IV®
- Memoria LPDDR2 de 1 GB
- Compatible con todas las últimas distribuciones de ARM GNU/Linux y Windows 10 IoT
- Conector micro USB para fuente de alimentación de 2,5 A
- 1 puerto Ethernet 10/100
- 1 conector de vídeo/audio HDMI
- 1 conector de vídeo/audio RCA
- 1 conector de cámara CSI
- 4 x puertos USB 2.0
- 40 pines GPIO
- Antena de chip
- Conector de pantalla DSI
- Ranura de tarjeta microSD
- Dimensiones: 85 x 56 x 17 mm

3.2.1.1. Pines GPIO de la Raspberry Pi

El puerto GPIO (General Purpose Input/Output) es un sistema de E/S (Entrada/Salida) de propósito general para usos múltiples. Los GPIO representan la interfaz entre la Raspberry Pi y el mundo exterior y con ellos se puede hacer multitud de proyectos. Pero para eso se debe saber sus características y como se programan (P.E, 2015).

Todos los pines son de tipo “unbuffered”, es decir, no disponen de buffers de protección, así que se debe tener cuidado con las magnitudes (voltajes, intensidad,...) cuando se conecte componentes a ellos para no dañar la placa. Como se puede apreciar

en la Figura 15, no todos los pines tienen la misma función:

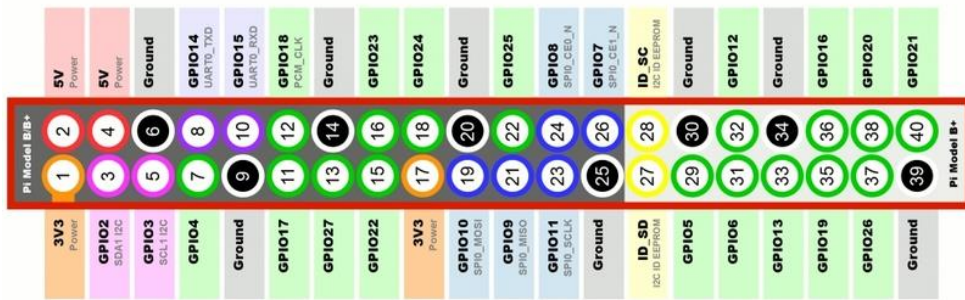


Figura 15: GPIO de la Raspberry pi 3 model B.

Fuente: <http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>

Pines de alimentación: pines de 5v, 3.3v (limitados a 50mA) y tierra (GND o Ground), que aportan alimentación a estos voltajes para tus circuitos. Te pueden servir como una fuente de alimentación, aunque también puedes utilizar otras fuentes (pilas, fuentes de alimentación externas, etc).

GPIO normales: son conexiones configurables que se pueden programar para los proyectos.

GPIO especiales: dentro de éstos se encuentran algunos pines destinados a una interfaz UART, con conexiones TXD y RXD que sirven para comunicaciones en serie.

3.2.1.2. Tarjeta de memoria Micro SD de 32GB

Tarjeta de memoria SanDisk Extreme MicroSDHC UHS-I de 32GB con Adaptador.



Figura 16: Tarjeta Micro SD SanDisk Extreme de 32GB.

Fuente: <https://www.amazon.com/SanDisk-Extreme-microSDHC-Adapter-SDSQXNE-032G-GN6MA/dp/B013CP5HCK/>

Especificaciones:

- Diseñado para los últimos smartphones basados en Android, tablets, cámaras de acción y MIL.
- Recomendado por GoPro para su uso con Hero, Hero3 +, Hero4, & HERO + LCD.

- Dispara a modo de ráfaga continua, Full HD y vídeo Ultra HD de 4K, velocidad de transferencia de hasta 90 MB/s.
- Construido para condiciones duras; Prueba de la temperatura, prueba del agua, prueba del choque y prueba de la radiografía.
- La aplicación SanDisk Memory Zone facilita el manejo de la memoria del smartphone con sus archivos multimedia.
- Software de recuperación de datos de lujo para descargar.

3.2.2. Elementos electrónicos.

En este punto, se presentan los elementos electrónicos que se usan en el proyecto, desde elementos pequeños como diodos hasta más grandes como el arduino., Estos elementos son:

3.2.2.1. Arduino Mega 2560.

Arduino es una plataforma física computacional open-hardware basada en una sencilla placa con entradas y salidas (E/S), analógicas y digitales, y en un entorno de desarrollo que implementa el lenguaje Processing/Wiring. Arduino puede utilizarse en el desarrollo de objetos interactivos autónomos o puede conectarse a un PC a través del puerto serie utilizando lenguajes como Flash, Processing, MaxMSP, etc (Arduino, Sin año).

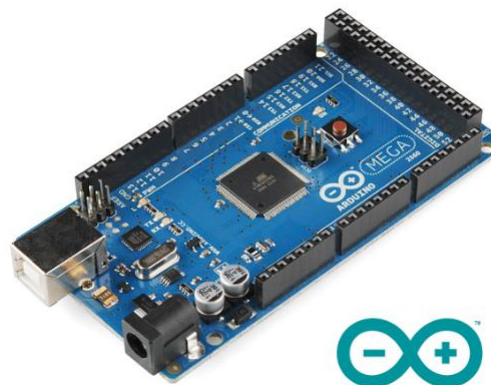


Figura 17: Arduino Mega 2560.

Fuente: <http://arduino.cl/arduino-mega-2560/>

Características:

- Microcontrolador ATmega2560.
- Voltaje de entrada de – 7-12V.
- 54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM).

- 16 entradas análogas.
- 256k de memoria flash.
- Velocidad del reloj de 16Mhz

3.2.2.2. Mini-webcam con conexión USB

ELP megapíxeles Super Mini 720p Módulo de cámara USB con lente de 100 grados (Amazon, 2016).



Figura 18: Cámara mini-web con conexión USB.

Fuente: <https://www.amazon.com/ELP-megapixel-Camera-Module-100degree/dp/B01DRJXAWA/ref>

Especificaciones:

- Mini-webcam USB con lente amplio angular M7 de 100grados para imagen de video.
- Mini lente estándar M7 con amplio ángulo FOV (D) 138 (H) 100 grados.
- Max 1280X720 @ 30fps HD 720P alta resolución.
- USB 2.0 de alta velocidad de tamaño mini 32x32 / 26x26mm para insertar en espacio pequeño
- Cable del USB de 1 metro, sirve de apoyo para la Raspberry Pi, OpenCV.

3.2.2.3. La compuerta lógica OR.

La compuerta O lógica o compuerta OR es una de las compuertas más simples utilizadas en la Electrónica Digital.

La salida X de la compuerta OR será “1” cuando la entrada “A” o la entrada “B” estén en “1” (Electrónica Unicrom, 2016).

La compuerta OR se representa con la siguiente función booleana: $X = A+B$ ó $X = B+A$

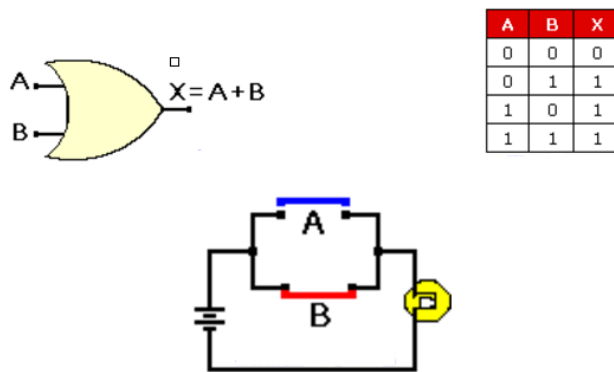


Figura 19: Compuerta lógica OR.

Fuente: http://unicrom.com/Tut_compuertaor.asp

3.2.2.4. Compuerta AND o compuerta Y.

La compuerta AND de 2 entradas tiene la siguiente tabla de verdad. Se puede ver claramente que la salida X solamente es “1” (1 lógico, nivel alto) cuando la entrada A como la entrada B están en “1”. En otras palabras: La salida X es igual a 1 cuando la entrada A y la entrada B son 1 (Electrónica Unicrom, 2016).

Esta situación se representa en álgebra booleana como: $X = A * B$ ó $X = AB$.

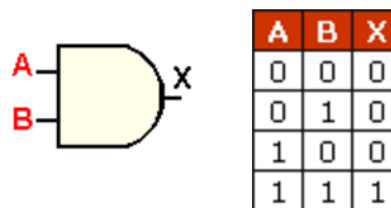


Figura 20: Compuerta AND o compuerta Y.

Fuente: <http://unicrom.com/compuerta-and-compuerta-y/>

3.2.2.5. Teclado de membrana 4x4.

Un teclado matricial es un simple arreglo de botones conectados en filas y columnas, de modo que se pueden leer varios botones con el mínimo número de pines requeridos. Un teclado matricial 4x4 solamente ocupa 4 líneas de un puerto para las filas y otras 4 líneas para las columnas, de este modo se pueden leer 16 teclas utilizando solamente 8 líneas de un microcontrolador. Se asume que todas las columnas y filas inicialmente están en alto (1 lógico), la pulsación de un botón se puede detectar al poner cada fila a en bajo (0 lógico) y comprobar cada columna en busca de un cero, si ninguna columna está en bajo entonces el 0 de las filas se recorre hacia la siguiente y así secuencialmente (Electrónica Unicrom, 2016).



Figura 21: Teclado matricial 4x4.

Fuente: http://electropro.pe/index.php?route=product/product&product_id=555

3.2.2.6. Relé 4 patas de 12V – 30 Amp.

El relé a utilizar es de 4 patas de 12V – 30 Amp., el voltaje y amperaje fue seleccionado en base a los parámetros de la batería y de funcionamiento del vehículo.

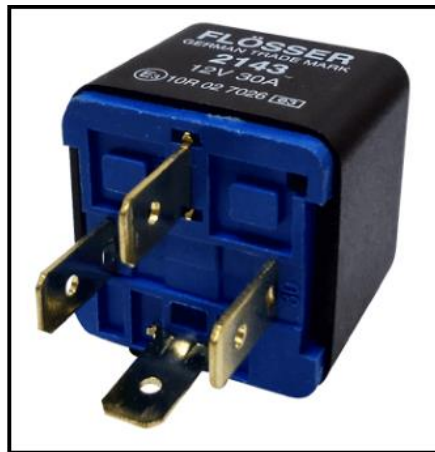


Figura 22: Relé 4 patas

Fuente: http://www.importadorablanco.cl/index.php?id_product=4298&controller=product

3.2.2.7. Transistor bipolar NPN 2N3055.

- Polaridad (N-P-N)
- Transistor [Amplificador[[amplificador]] de potencia de audio
- Corriente máxima de colector (I_c) 15 Ampere
- De colector a base (CBO) 100 Voltios
- De colector a emisor (CEO) 60 Voltios
- De emisor a base (EBO) 7 Voltios
- Ganancia típica de la corriente directa (hfe) 45
- Máxima disipación de potencia en colector (P_d) 115 (Watts)

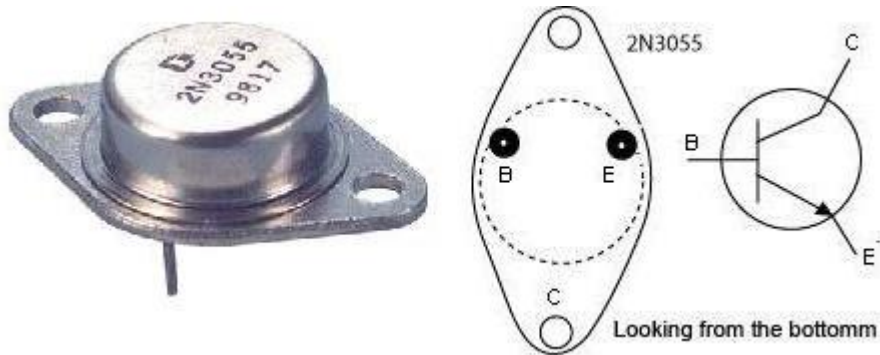


Figura 23. Transistor NPN 2N3055

Fuente: <http://www.geekbotelectronics.com/producto/2n3055-transistor-npn/>

3.2.2.9. Diodo 1N4007.

El diodo necesario para éste proyecto es el 1N4007, es muy usado en electrónica como rectificador en fuentes de alimentación y supresor de picos en bobinas y relés. El 1N4007 soporta un mayor voltaje.

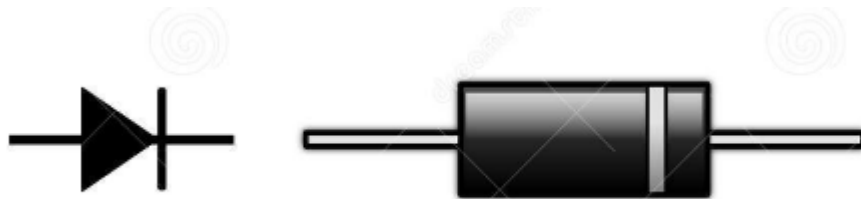


Figura 24. Diodo 1N4007

Fuente: <http://es.slideshare.net/tarik64/diodo-34695767>

3.2.2.10. Display LCD alfanumérico de 16x2.



Figura 25: Display LCD alfanumérico de 16x2.

Fuente: electronicoscaldas.com/displays-lcd-alfanuméricos-y-gráficos

Características:

- 16 caracteres x 2 líneas
- Caracteres de 5x8 puntos
- Tamaño de caracter: 5.23 x 3 mm
- Puede mostrar letras, números, caracteres especiales, y hasta 8 caracteres creados por el usuario
- Backlight de LED color azul
- Caracteres color blanco
- Interface paralela. Puede operar en modo de 8 bits, o de 4 bits para ahorrar pines del microcontrolador
- Posee controlador KS0066U o equivalente on-board (compatible Hitachi HD44780)
- Voltaje de alimentación: 5 V.

3.3. Selección de Técnicas de Reconocimiento Facial.

El sistema de reconocimiento facial está basado en el algoritmo LBPH porque fue el de mejor rendimiento ya que la mayoría de las pruebas fueron correctas y no es necesario tener una gran base de datos como lo requieren otros algoritmos como: EigenFaces, FisherFaces que de igual forma son propios de OpenCv.

Se eligió el algoritmo LBPH porque es el más robusto a cambios de iluminación en las fotografías, es decir, al considerar alta y baja luminosidad queda demostrado que esto no es un factor que influye en el reconocimiento facial. Este algoritmo es ideal para detectar diferentes emociones en un rostro. Además, el entrenamiento se actualiza en cada reconocimiento realizado y tiene una buena eficiencia computacional, por lo que es muy usada en sistemas embebidos.

Se ha seleccionado el algoritmo LBPH para ser el utilizado en el sistema, debido a su bajo tiempo de resolución del reconocimiento facial, dando un tiempo de respuesta al usuario muy corto, ya que trabaja con imágenes en escala de grises, permitiendo una autenticación mucho más rápida en el sistema de seguridad.

Su tiempo de reconocimiento es muy bueno en todas las pruebas, siendo el tiempo máximo de 3.38 segundos, muy inferior al obtenido en los otros algoritmos.

3.4. Circuito de alimentación de la Raspberry.

Es necesario estudiar la alimentación de la Raspberry para evitar que esta se apague cuando se de arranque al vehículo al existir una caída de voltaje, para que esté encendido en todo momento.

Para esto se ha instalado un regulador de voltaje de 12V a 5V ya que este último valor es el voltaje para la alimentación de la Raspberry.



Figura 26: regulador de voltaje de 12V a 5V.

Fuente: Autores.

Para la conexión de este regulador, de la caja de fusibles se conecta un cable a masa o tierra y otro a la salida del fusible de la alimentación del radio del vehículo que es 12V constantes ya este apagado o encendido el vehículo sin importar el arranque de este. El extremo de estos cables va conectados a la entrada del regulador, a la salida de este, es el macho USB con 5V constantes y este a la entrada de la Raspberry para alimentarlo.



Figura 27: Alimentación 12V del regulador de voltaje.

Fuente: Autores

3.5. Recolección de base de datos de rostros.

Aunque la librería OpenCV tenga una propia base de datos, se realiza una nueva base de datos de rostros de varias personas, con gestos distintos, con un número total de 50 fotografías de cada persona de un total de 25 personas.



Figura 28: Nueva base de datos de rostros.

Fuente: Autores.

En la Figura 28 se muestra una pequeña colección de los rostros de la nueva base de datos, estas fueron tomadas con la ejecución del código “Registro de Usuarios” previamente hecha en Python.

Todas las fotografías están con el mismo tamaño de 112x92 pues mandar a ejecutar el código de reconocimiento facial se colocara el tamaño de las imágenes y el nombre de la carpeta de la base de datos, si estos no coinciden, el programa enviara un error.

Las fotografías muestran tan solo los rostros de las personas pues al sistema no le interesa nada más y están en escala de grises con diferentes gestos y tipos de luz haciendo que el sistema sea mucho más rápido al momento del reconocimiento.

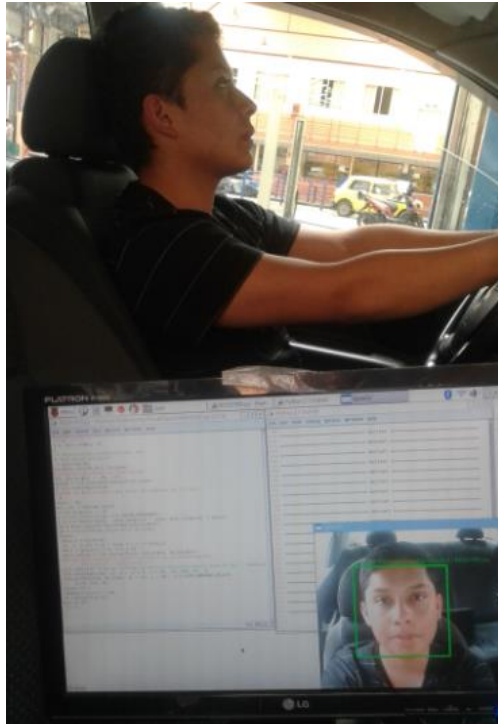


Figura 29: Recolección de fotos.

Fuente: Autores.

Las diferentes fotos fueron tomadas con la persona dentro del vehículo, en el asiento del conductor para simular una persona conduciendo y siendo captada por la cámara como se muestra en la Figura 29 a distintas posturas y a diferentes horas del día para variar la luz por la mañana y la tarde.

3.6. Esquema General del proyecto.

En la figura 30, se muestra un esquema generalizado de las conexiones que tiene el sistema, tanto para la Raspberry, Arduino, pantalla lcd, teclado matricial y las compuertas lógicas que se seleccionaron anteriormente.

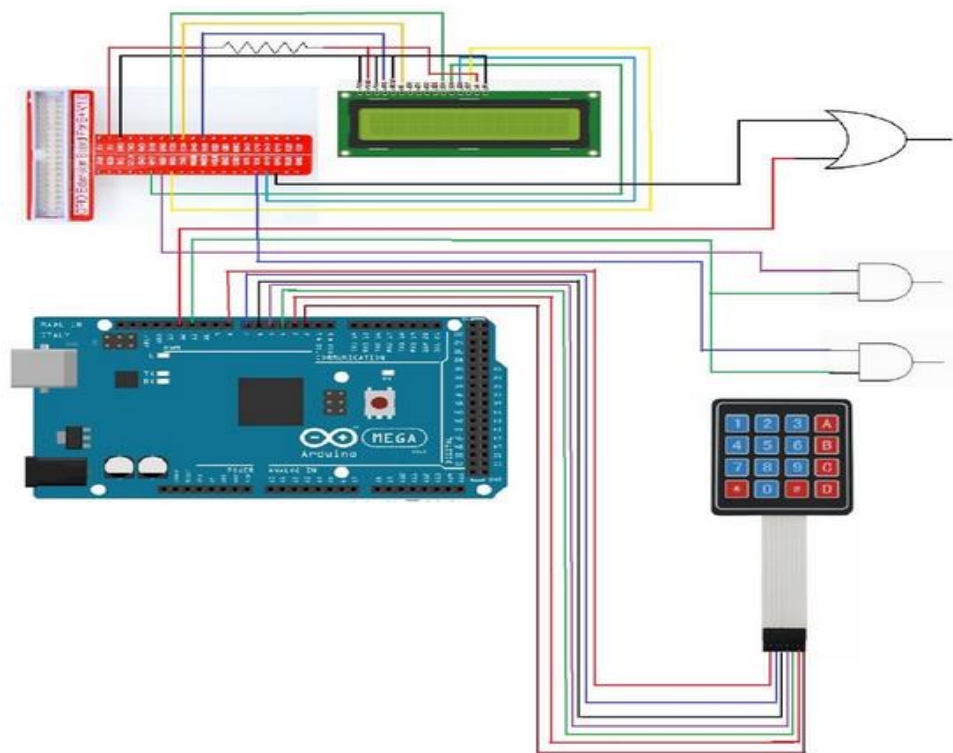


Figura 30: Esquema general

Fuente: Autores.

Entre la Matriz Keypad 4x4 y el Arduino Mega 2560 se hace las siguiente conexiones: el row 1 (fila 1) de la Matriz va conectado al pin 8 de Arduino, el row 2 (fila 2) va conectado al pin 7, el row 3 (fila 3) va conectado al pin 6 y el row 4 (fila 4) va conectado al pin 5; el pin 5 (columna 1) de la Matriz va conectado al pin 4 de Arduino, el pin 6 (columna 2) va conectado al pin 3 y el pin 7 (columna 3) va conectado al pin 2 de la placa Arduino.

El pin 11 de Arduino que va conectado a las compuertas lógicas AND dará un error de la clave y el pin 12 que conecta a la compuerta lógica OR dará una entrada de clave correcta.

Las conexiones entre el Tablero de extensión GPIO y la pantalla LCD son: el pin de 5V va conectado a las entradas VDD y mediante una resistencia; el pin GND va conectado a las entradas VSS, VO, R/W y K de la pantalla LCD; el pin IO23 hace contacto a la entrada D4; el pin IO24 va conectado a la entrada E; el pin IO25 va hacia a la entrada RS; el pin IO17 va conectado a la entrada D5; el pin IO22 va conectado a la entrada D7; el pin IO13 va hacia a la entrada D6 de la pantalla LCD.

De esta manera es como se muestra la situación o estado del sistema mediante mensajes de textos cortos en la pantalla LCD.

Entre el tablero de extensión de GPIO y las compuertas se realizan las siguientes conexiones: el pin IO17 se conecta a una entrada de la compuerta AND; el pin IO6 se conecta a una entrada de la otra compuerta AND; y por último el pin IO19 va conectado a la entrada de la compuerta lógica OR. Estas conexiones transmiten la información para activar o desactivar el relé mediante las salidas de las compuertas lógicas.

3.7. Programación en Python para Reconocimiento Facial

El código para reconocimiento facial se realiza en Python, al ser software libre no se necesita licencia, es una plataforma de fácil acceso y sus líneas de código se encuentran a disposición en la red y se las puede manejar a gusto según los requerimientos.

Para la programación se ha separado en tres fases y cada una de estas cumplirá un rol específico, los cuales se detallan a continuación.

3.7.1. Fase de entrenamiento.

Al tener ya una base de datos de rostros, es necesario entrenar al sistema para que este sea más eficiente y el error sea mínimo y se lo realiza cada vez que exista un nuevo usuario en la base de datos o cualquier cambio que se realizó en la programación, para actualizar al sistema.

La fase de entrenamiento se lo guarda como un archivo aparte con extensión yml, en el mismo directorio donde se encuentran todas las programaciones del sistema.

Este nuevo archivo tiene guardado toda la información sobre la fase de entrenamiento con un peso considerable, se realiza esto ya que en la fase de reconocimiento no se demore mucho en hacer su trabajo.

```
model = cv2.face.createLBPFFaceRecognizer()
model.train(images, labels)
model.load("~/home/pi/Desktop/recoFacialCopial/LBPH.yml")
```

Figura 31: Fase de entrenamiento.

Fuente: Autores.

En la Figura 31 se aprecia las tres líneas de código que comandan la fase de entrenamiento, en la primera se elige la técnica de reconocimiento facial que se va a utilizar, el “model.train” es el entrenamiento del sistema que actualiza la base de datos de rostros y líneas de código nuevas, y esta es la que será guardada con la extensión “yml” como se muestra en la tercera línea de código.

Al final del entrenamiento muestra el resultado de la fase de entrenamiento, se aprecia los nombres de las carpetas que contienen las fotografías de las personas de la base de datos, esto ocurre cada vez que se mande a entrenar el sistema.

3.7.2. Fase de reconocimiento.

Esta fase se ocupa exclusivamente en buscar y comparar el rostro que se muestra en la cámara con una de las carpetas de la base de datos. Al encontrar una semejanza entre los dos rostros, pondrá el nombre al mismo al cual pertenece.

```
names={0: 'Eduardo', 1: 'David', 2: 'Lata', 3: 'Cajas', 4: }  
model1 = cv2.face.createLBPFFaceRecognizer()  
model1.load("/home/pi/Desktop/recoFacialCopia1/LBPH.yml")
```

Figura 32: Fase de reconocimiento facial.

Fuente: Autores.

Los nombres de las carpetas que se muestra al final de la fase de entrenamiento se coloca como una matriz de nombre “names”. Anteriormente se había guardado un archivo de la fase de entrenamiento, en esta nueva fase, se llama a este archivo a un nuevo código con la función “model1.load (ubicación del archivo)” para así evitar que dentro de la fase de reconocimiento se entrene el programa y por lo tanto tener una mejor eficacia y menor tiempo de respuesta en esta fase.

```
prediction = model1.predict(face_resize)  
cara = "%s" % (names[prediction])
```

Figura 33: Predicción del rostro.

Fuente: Autores

El enfoque que da la cámara USB hacia el rostro de la persona, debe de ser del mismo tamaño de las que fueron tomadas las fotografías (en este caso 112x92), caso contrario en el programa aparecerá un error.

La cámara al detectar un rostro, instantáneamente buscara un nombre de las carpetas que se encuentran en la base de datos y de esta predecirá un nombre al encontrar similitud en los rostros, esto se logra con la función “model.predict (tamaño de las fotos)”.

En la programación, se ha propuesto que solo a los autores de este proyecto sean los usuarios autorizados para poner en marcha el vehículo al detectar su rostro, por tanto el sistema al reconocer a estos usuarios realizará la activación del paso de corriente hacia la bomba de gasolina.

3.7.3. Activación de alarmas y actuadores.

Como se había mencionado, el sistema al reconocer el rostro de cualquiera de los dos autores, hará funcionar a la bomba de gasolina, al ser un reconocimiento exitoso, se encenderá un led amarillo como prueba de que la bomba ha sido activada.

Cabe recalcar que el usuario será reconocido 5 veces seguidas para poder activar el paso de corriente hacia la bomba, si no se cumplen estos 5 reconocimientos, no se dará paso de corriente a la bomba para encender al vehículo.

En la Figura 34 se muestra la programación cuando el sistema ha reconocido a uno de los usuarios propuestos para poder conducir el vehículo, “GPIO.output (19, GPIO, HIGH)” indica que el GPIO 19 de la Raspberry es la salida de la señal para activar la bomba y en la pantalla lcd aparecerá “Bienvenido Sr. Usuario” como se muestra en la Figura 35, dando paso para encender al vehículo. Esto ocurre al reconocer a cualquiera de los dos usuarios propuestos.

```

if cara == "Pablo":
    contadorP=contadorP+1
elif cara == "Marco":
    contadorM=contadorM+1
else:
    print intento
    desconocido=desconocido+1
    print desconocido

if contadorM == Nframes:
    GPIO.output(19,GPIO.HIGH)
    GPIO.output(27,GPIO.LOW)

    if reconocido == False:
        LCD lcd_string("Bienvenido" LCD.LINE_1)
        LCD lcd_string("Sr. USUARIO" LCD.LINE_2)

```

Figura 34: Reconocimiento de usuario.

Fuente: Autores.

En caso de no ser un usuario registrado, se apaga el led amarillo y la bomba dejara de funcionar y el vehículo se detendrá. De igual manera el sistema busca 6 veces el rostro de la cámara, si en estas 6 oportunidades el usuario no ha sido reconocido y en la pantalla lcd saldrá un mensaje "ACCESO DENEGADO" como se muestra en la Figura 35, se activaran las alarmas comandadas por el GPIO 27 de a Raspberry y el sistema inhabilitara hasta que el usuario pulse el botón de reinicio.

Pasa lo mismo cuando no existe un rostro en la cámara, en el lcd se mostrara "USUARIO NO DETECTADO" y al cabo de 10 segundos se activaran las alarmas, pero no se inhabilita el vehículo, se paga las alarmas en canto la cámara detecte un rostro y el proceso de reconocimiento empieza nuevamente.

```

if desconocido == denegado:
    LCD lcd_string("ACCSESO" LCD.LINE_1)
    LCD lcd_string("DENEGADO" LCD.LINE_2)

```

Figura 35: Código para usuario desconocido.

Fuente: Autores.

3.8. Programación para clave de acceso en Arduino.

Como se dijo anteriormente, el sistema tiene una clave de acceso para omitir el trabajo del sistema de Visión Artificial, esto dará paso a la habilitación del vehículo para

conducir.

Se ha programado en Arduino y las señales de salida de clave correcta o incorrecta serán enviadas a compuertas lógicas para hacer el trabajo de habilitación o inhabilitación del sistema.

```
#include <Password.h> //Incluimos la libreria Password
#include <Keypad.h> //Incluimos la libreria Keypad

Password password = Password( "1234" ); //Definimos nuestro Password
```

Figura 36: Librerías Password y Keypad

Fuente: Autores.

En la figura 36 se aprecia que para realizar la programación de una clave de acceso en Arduino se requiere incluir unas librerías como *Password* que hará la función de reconocer una clave previamente introducida y la salida de una señal de “clave correcta” o “clave incorrecta” y la librería *Keypad* que reconocerá al teclado de matriz 4X4 para digitar la clave.

En este caso a clave viene dado con la combinación de 4 dígitos que son: “1234”, la misma que se puede cambiar a conveniencia del usuario en la programación y cargarla en el microcontrolador del Arduino.

```
int ledRed = 11;
int ledYellow = 12;

const byte ROWS = 4; // Cuatro Filas
const byte COLS = 3; // Cuatro Columnas

// Definimos el Keypad
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
```

Figura 37: Definición de la Matriz y señales de salida.

Fuente: Autores.

Se utiliza un teclado de membrana matricial 4x4 de la Figura 21, cabe recalcar que para este caso se van a utilizar solamente las teclas numéricas por lo que la matriz será 4x3, 4 filas y 3 columnas y eso es lo que se programa en código de Arduino como se muestra en la Figura 37.

```

switch (eKey) {
  case '*': checkPassword(); break;
  case '#': password.reset(); break;
  default: password.append(eKey);
}

```

Figura 38: Start y reinicio del teclado.

Fuente: Autores.

Para aceptar el código que se digitó se presiona la tecla “*” o “start”, y para reiniciar el mismo se presiona “#” o “stop” en el caso que se digitó mal con código y se requiera ingresar nuevamente.

Las señales de salida de “clave correcta” y “clave incorrecta” serán dadas por los pines 12 y 11 del Arduino respectivamente, es decir, cuando la clave digitada por el usuario es correcta se encenderá un led amarillo de confirmación que el vehículo está habilitado, caso contrario se encenderá un led rojo que indica que la clave es incorrecta y que el vehículo aún sigue inhabilitado.

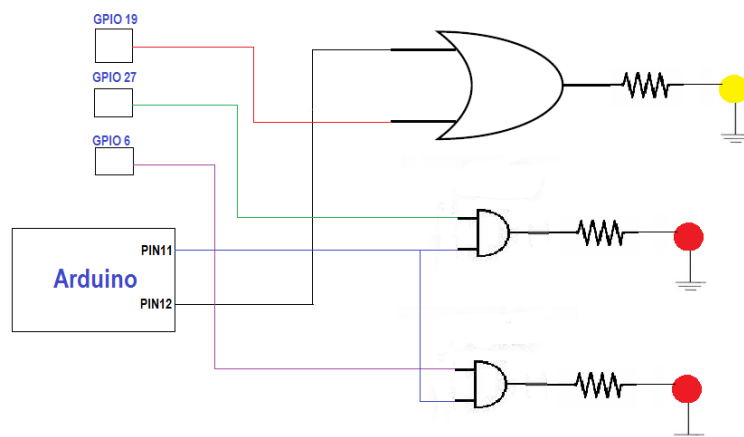


Figura 39: Conexión entre Arduino y compuertas lógicas.

Fuente: Autores.

En la Figura 39 muestra que el GPIO 19 de la Raspberry, la cual es la señal cuando el usuario ha sido reconocido exitosamente, va conectada a la compuerta OR conjuntamente con el PIN12 del Arduino la cual es la señal de que se a digitado la clave correcta en el teclado, a la salida de la compuerta va a ser cualquiera de las dos señales ya sea del Arduino o del sistema de Visión Artificial pues este es el trabajo de la compuerta OR, es decir, cuando se digita en el teclado la clave correcta, la señal sale del Arduino y esta será la salida de la compuerta, sin importar que el sistema de Visión

Artificial mande o no mande señal por el GPIO 19.

Por otra parte, cuando se digite una clave incorrecta, habrá señal del PIN11 del Arduino, el cual está conectada a las dos compuertas AND como se muestra en la Figura 39, las otras entradas de las compuertas son las señales de los GPIO's 27 y 6, las cuales son alertas de usuario no reconocido y usuario no detectado respectivamente.

El trabajo de las compuertas AND es q al tener señal 1 en sus dos entradas, la salida será 1, es decir, al haber digitado la clave incorrecta el GPIO19 estará en 0, la cual dará paso al sistema de Visión Artificial y se tendrá también la activación de las dos alarmas que son salidas de las compuertas AND.

3.9. Activación de la bomba de gasolina.

La siguiente figura muestra el esquema de la conexión de la salida de la compuerta OR con la bomba de gasolina.

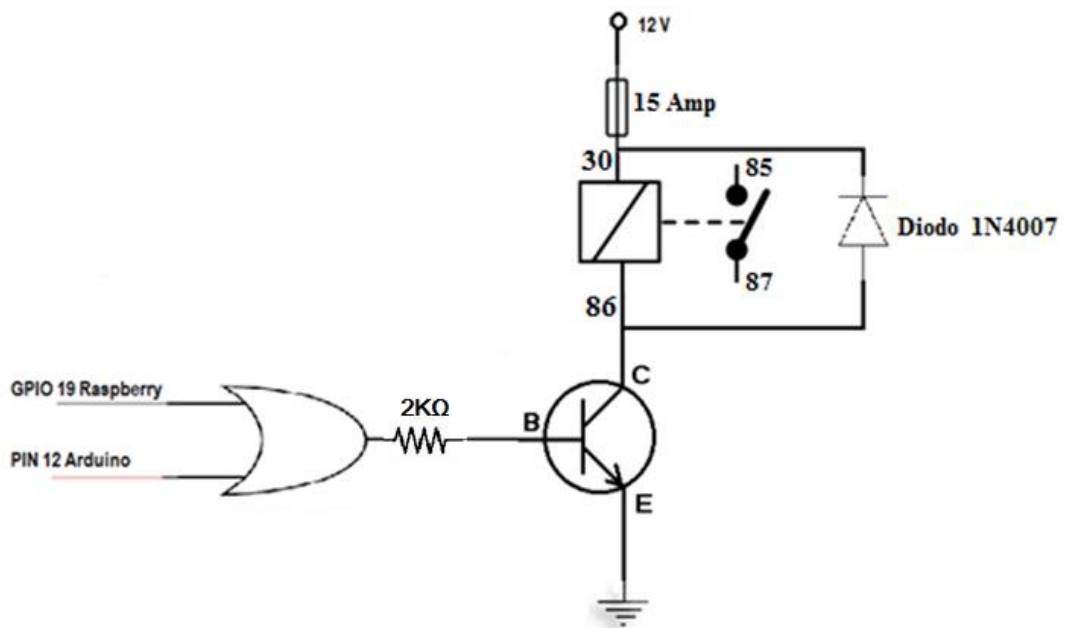


Figura 40: Esquema de conexión para la bomba de gasolina.

Fuente: Autores.

Las entradas de la compuerta OR como se había mencionado son el GPIO 19 de la Raspberry y el PIN 12 del Arduino, la salida directamente va conectado hacia la Base del transistor, el Emisor va conectado a masa, el Colector va hacia el 86 del relé. Los 12V de la batería va conectada al 30 del relé con un fusible de 30Amp entre estos y el

diodo rectificador en paralelo al relé, el 85 va a los 12V y el 87 ira conectada hacia el positivo de la bomba de combustible.

Con esto se logra que al tener señal de salida de la compuerta OR, el transistor dará la potencia suficiente para cerrar el swich del relé y lograr prender la bomba de combustible y consigo poder encender el vehículo.

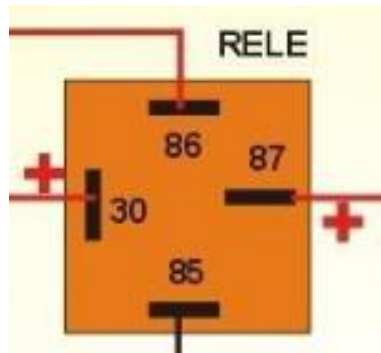


Figura 41: Pines del relé de 4 patas.

Fuente: <https://electroaut.blogspot.com/>

CAPÍTULO 4

4. COSTOS, PRUEBAS Y RESULTADOS.

En este capítulo, se entregan costos y los resultados de las pruebas realizadas para evaluar el rendimiento de los métodos de reconocimiento facial que se han implementado en este proyecto.

4.1. Costo de elementos.

Tabla 3: Lista de precios. Presupuesto.

Fuente: Autores.

CANTIDAD	ELEMENTOS	DESCRIPCIÓN	VALOR UNITARIO(USD)	VALOR TOTAL(USD)
1	Relé	12v, 30A	3,50	3,50
1	mini webcam	Mini USB 720p	28,00	28,00
1	Tarjeta MicroSDHC	SanDisk Extreme de 32GB	17,00	17,00
1	Raspberry pi 3	Paquete Kid started	90,00	90,00
1	Arduino	Mega 2560 R3	25,00	25,00
1	Módulo LCD	2 líneas	15,00	15,00
-	Componentes electrónicos varios	Resistores, transistores, , etc.	16,00	16,00
2	Placa PCB perforada	5x7cm para prototipos	1,50	3,00
1	Cinta adhesiva	Doble faz	3,50	3,50
2	Cinta aislante	Negra	1,00	2,00
-	Cable multipar	5 m.	2,50	2,50
1	Regulador de fuente de alimentación	De 12V a 5V	5,00	5,00
1	Extensión de cable USB	2 m.	3,00	3,00
1	Cable de datos	Bus para LCD	4,00	4,00
-	Cable de audio	5 m.	2,50	2,50
1	Teclado matricial	4x4	5,00	5,00
TOTAL				225,00

Se detallan todos los elementos necesarios con sus precios, para la construcción del sistema de seguridad por reconocimiento facial para la puesta en marcha de un vehículo.

4.2 Pruebas de laboratorio.

Después de haber identificado todos los elementos que serán necesarios para el sistema se procedió a armar el circuito en un tablero de pruebas protoboard para comprobar el funcionamiento en condiciones reales.

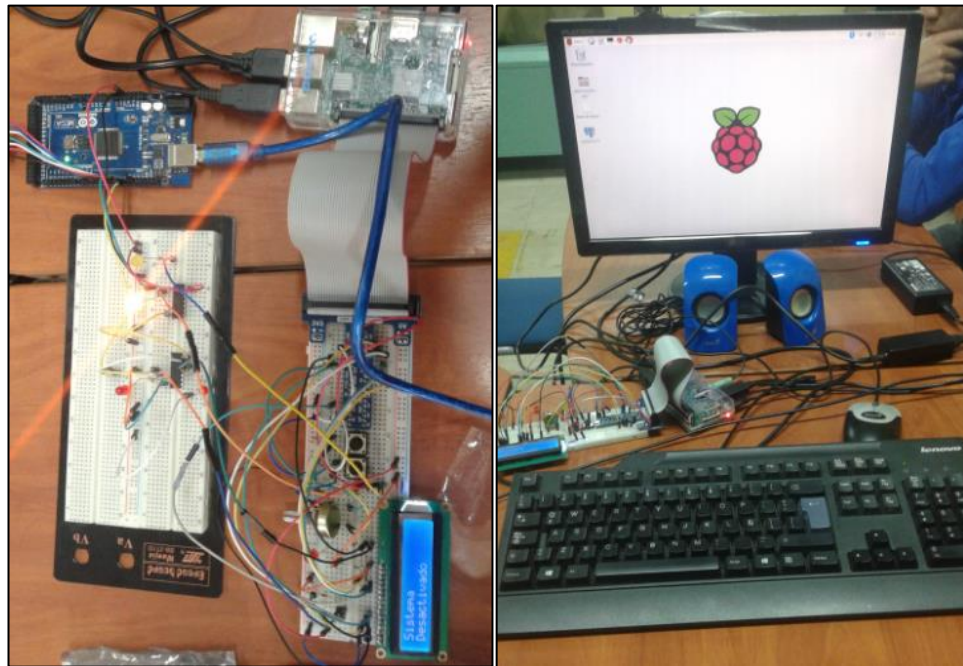


Figura 42: Pruebas de laboratorio.

Fuente: Autores

Habiendo montado el circuito en el tablero de pruebas se puede comprobar que el sistema funciona de la manera esperada; el relé se activa al identificar el rostro del usuario y se cierra el circuito de alimentación de combustible, encendiendo la luz amarilla que indica que el automotor puede ser encendido.

Una vez que se ha asegurado el correcto funcionamiento del circuito con resultados favorables a cada una de las pruebas realizadas, se procede a plasmar el circuito electrónico del sistema en tres placas perforadas. Se debe tener precaución al momento de colocar los componentes en las placas, ya que se los debe colocar en su respectiva posición y con la cantidad suficiente de estaño para que no quede mal sujetado, pero

tampoco invadir partes del circuito que puedan impedir el buen funcionamiento del mismo.

4.3. Pruebas realizadas.

El algoritmo para el reconocimiento facial se programa en Python, usando los métodos de reconocimiento tales como Eigenfaces, Fisherfaces y LBPH, con esto se aprecia que la Raspberry cambia el tiempo para iniciar el algoritmo, lo cual se mostrara posteriormente.

Ya instalado todo el sistema en el vehículo se procede a verificar que funcione de la manera esperada, para esto se realizan diferentes pruebas a bordo del vehículo.

Las pruebas se realizan a diferentes horas del día, con diferentes cantidades de luz con la base de datos ya realizada anteriormente, con los usuarios permitidos y otras personas que no constan en la base de datos, de esta manera se tomara el tiempo de reconocimiento y la efectividad de este último.

4.3.1. Tiempos que tardan en iniciarse los algoritmos.

En la tabla 4, se muestran los tiempos que tardan en iniciarse los tres algoritmos o técnicas usadas para el entrenamiento y el reconocimiento. Para esta prueba, la Raspberry debe de estar conectada hacia un monitor externo para visualizar los comandos y en este cambiar la técnica de reconocimiento como se muestra en la Figura 43.

```
model = cv2.face.createLBPFFaceRecognizer()
model.train(images, lables)
model.load("/home/pi/Desktop/recoFacialCopial/LBPH.yml")
```

Figura 43: Línea de comando para la técnica de reconocimiento facial.

Fuente: Autores

En la tabla 4, se muestra los tiempos en segundos en las que tarda en iniciarse el programa, en este caso para el entrenamiento del sistema que como se mencionó anteriormente, este entrenamiento se usa para actualizar los datos o líneas de comando que fueron cambiadas dentro de la programación o se modificó la base de datos de rostros. De igual manera en la Tabla 5, se muestran los tiempos en segundos de esta misma prueba pero para la fase de reconocimiento.

Tabla 4: Tiempo en s que tarda en entrenarse el sistema.

Fuente: Autores

ENTRENAMIENTO						Promedio
Algoritmo	Prueba1	Prueba2	Prueba3	Prueba4	Prueba5	
Fisherface	362	359	360	360	361	360,4
Eigenface	295	298	295	301	300	297,8
LBPH	75	73	74	75	75	74,4

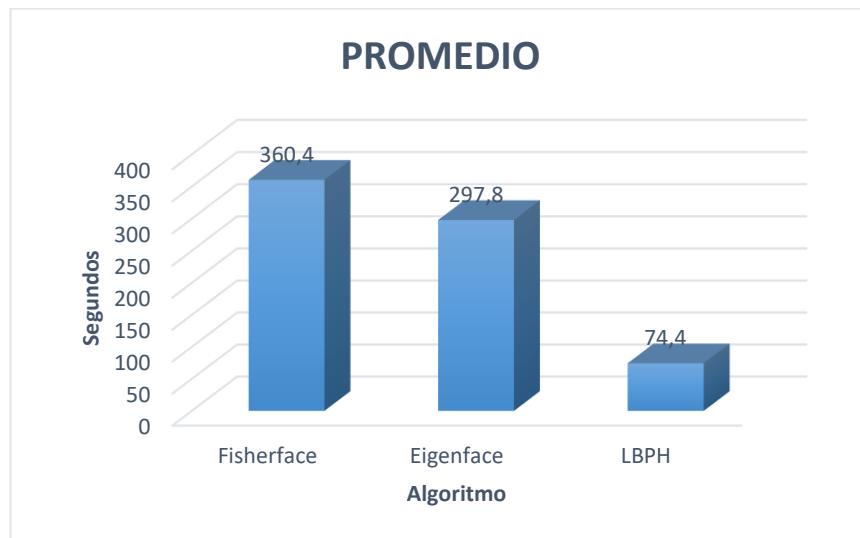
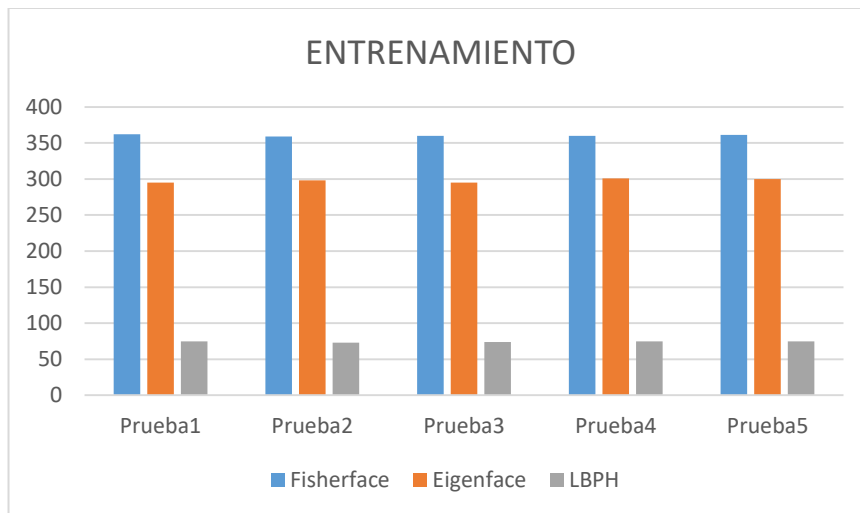


Figura 44: Tiempos en s que tarda en entrenarse el sistema.

Fuente: Autores

Se aprecia claramente que el algoritmo LBPH tarda mucho menos en entrenarse que los demás, esto demuestra que al entrenar tanto el algoritmo Eigenface y Fisherface genera un archivo yml mucho más pesado que el del LBPH, como se indicó anteriormente.

Cabe recalcar que no porque un algoritmo de reconocimiento facial tarde mucho en

entrenarse es más eficiente, esto no es así, pues en la fase de reconocimiento, el sistema tardara mucho más en iniciarse y cuando este corriendo el programa puede tender a detenerse repentinamente o colgarse.

Estas diferencias de tiempo, más que a la hora en la que fueron realizadas, son el resultado del análisis que cada algoritmo realiza a la base de datos de rostros que fue realizada anteriormente, pues esta es muy grande con 25 sujetos de prueba con 40 fotos de cada quien, esto hace demorar el entrenamiento.

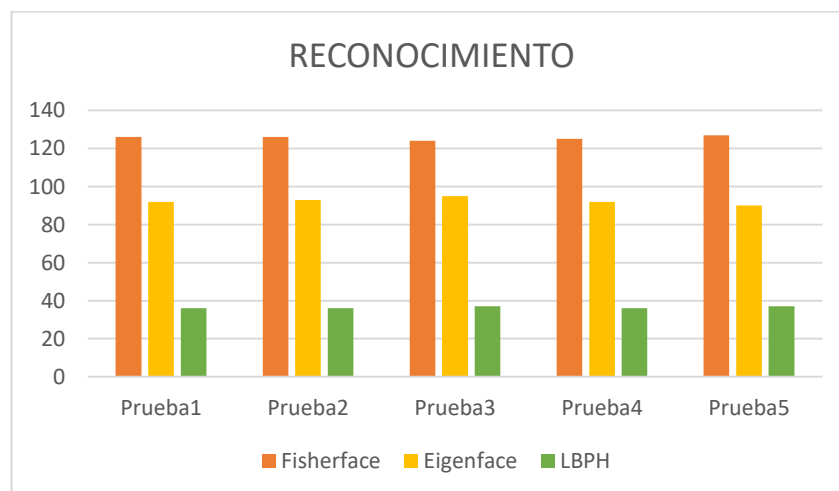
La prueba se realizó analizando varias bases de datos con diferentes números de sujetos y un determinado números de fotos, y aun así el algoritmo Fisherface es el que más se tarda en entrenar, siguiéndole el Eigenface.

Los resultados mostrados son los gráficos del análisis de la última base de datos que fue realizada dentro del vehículo, a diferentes horas del día, que anteriormente fue seleccionada como la mejor base de datos para trabajar el sistema de reconocimiento facial por tanto el algoritmo LBPH es más eficiente en este caso.

Tabla 5: Tiempo en s que tarda en iniciarse el algoritmo en fase de reconocimiento.

Fuente: Autores.

RECONOCIMIENTO						Promedio
Algoritmo	Prueba1	Prueba2	Prueba3	Prueba4	Prueba5	
Fisherface	126	126	124	125	127	125,6
Eigenface	92	93	95	92	90	92,4
LBPH	36	36	37	36	37	36,4



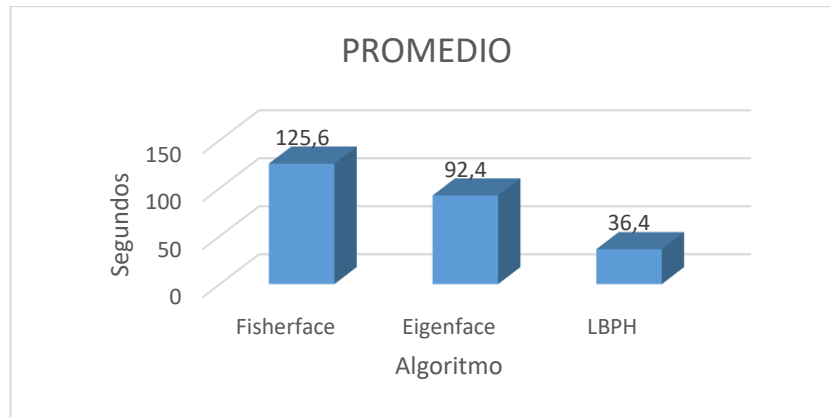


Figura 45: Tiempo en s que tarda en iniciarse el algoritmo en fase de reconocimiento.

Fuente: Autores.

Al entrenar con las tres técnicas de reconocimiento facial, se crean archivos resultados de estos entrenamientos para Fisherface, Eigenface y LBPH con extensión yml para cada uno. En el código para reconocimiento facial, para cada prueba se colocó el nombre de las técnicas usadas y en la siguiente línea llamar al archivo creado en la fase de entrenamiento como se muestra en la Figura 33 para empezar a correr el programa.

Los resultados de la Tabla 5 y Figura 45 muestran que en esta fase también es menor el tiempo que tarda en iniciar el algoritmo LBPH de reconocimiento facial. La explicación es que de igual manera que en la fase de entrenamiento, la técnica Fisherface y Eigenface tarda en abrir el archivo generado anteriormente.

4.3.2. Efectividad del reconocimiento.

En este punto se va a realizar pruebas estrictamente para la fase de reconocimiento, es decir, probar si el sistema logra reconocer con éxito el rostro de las personas o usuarios que pueden encender y conducir el vehículo y rechazar a aquellos que no lo son. De esta manera se logrará observar la efectividad que tiene este sistema de seguridad para la implementación a futuro en otros vehículos.

4.3.2.1. Efectividad de reconocimiento para usuarios autorizados.

Para esto, los sujetos de prueba serán los usuarios autorizados para poner en marcha y conducir al vehículo, se hacen 15 pruebas por cada una de las técnicas de reconocimiento facial, anotar sí reconoce o no y el tiempo en segundos que tarda.

Tabla 6: Pruebas de reconocimiento de usuarios autorizados con Eigenface.

Fuente: Autores.

Pruebas	USUARIO 1		TIEMPO	USUARIO 2		TIEMPO
	RECONOCE			RECONOCE		
	SI (1)	NO (1)		SI (1)	NO (1)	
1	1	0	6	1	0	6
2	0	1	6	1	0	6
3	1	0	5	1	0	6
4	1	0	6	1	0	6
5	1	0	5	1	0	6
6	0	1	6	1	0	6
7	1	0	6	0	1	5
8	0	1	6	1	0	6
9	0	1	6	0	1	6
10	1	0	5	1	0	7
11	1	0	6	1	0	6
12	1	0	6	0	1	5
13	1	0	7	1	0	6
14	1	0	6	1	0	6
15	1	0	6	0	1	6
TOTAL	11	4		11	4	

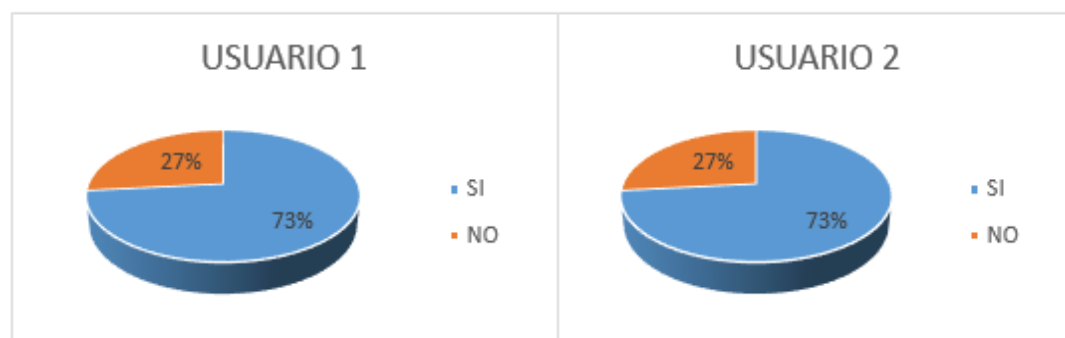


Figura 46: Efectividad de reconocimiento con Eigenface.

Fuente: Autores

Con el algoritmo Eigenface, se aprecia que existen 11 aciertos a un contrario de 4 errores de reconocimiento para los usuarios permitidos, por lo que tendría una efectividad del 73% en pruebas realizadas dentro del vehículo a diferentes horas y cantidades de luz distintas.

Tabla 7: Pruebas de reconocimiento con Fisherface.

Fuente: Autores.

Pruebas	USUARIO 1		TIEMPO	USUARIO 2		TIEMPO
	RECONOCE			RECONOCE		
	SI (1)	NO (1)		SI (1)	NO (1)	
1	1	0	6	1	0	6
2	0	1	6	0	1	6
3	1	0	5	1	0	6
4	0	1	6	0	1	6
5	1	0	5	1	0	6
6	0	1	6	1	0	6
7	1	0	6	0	1	5
8	0	1	6	1	0	6
9	0	1	6	0	1	6
10	1	0	5	1	0	7
11	1	0	6	0	1	6
12	0	1	6	0	1	5
13	0	1	7	1	0	6
14	1	0	6	1	0	6
15	1	0	6	0	1	6
TOTAL	8	7		8	7	

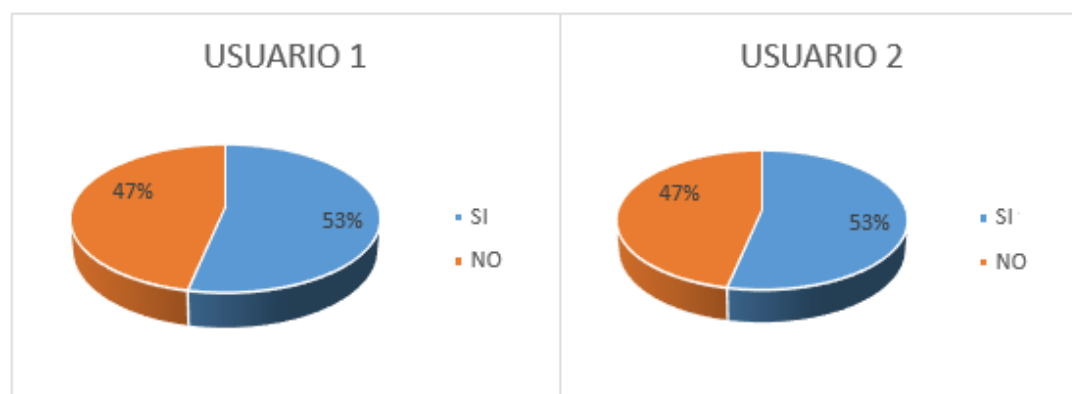


Figura 47: Efectividad de reconocimiento con Fisherface.

Fuente: Autores

Al hacer pruebas con el algoritmo Fisherface, el reconocimiento es menos efectivo al de la prueba anterior al tener una efectividad de 53% con 8 aciertos, por lo que queda totalmente descartado el uso de esta técnica para el proyecto y es comprobado por estos resultados.

Tabla 8: Pruebas de reconocimiento con LBPH.

Fuente: Autores.

Pruebas	USUARIO 1		TIEMPO	USUARIO 2		TIEMPO
	RECONOCE			RECONOCE		
	SI (1)	NO (1)		SI (1)	NO (1)	
1	1	0	6	1	0	6
2	1	0	6	1	0	6
3	1	0	5	1	0	6
4	1	0	6	1	0	6
5	1	0	5	1	0	6
6	1	0	6	1	0	6
7	1	0	6	1	0	5
8	1	0	6	1	0	6
9	1	0	6	1	0	6
10	1	0	5	1	0	7
11	1	0	6	1	0	6
12	1	0	6	0	1	5
13	1	0	7	1	0	6
14	1	0	6	1	0	6
15	0	1	6	1	0	6
TOTAL	14	1		14	1	

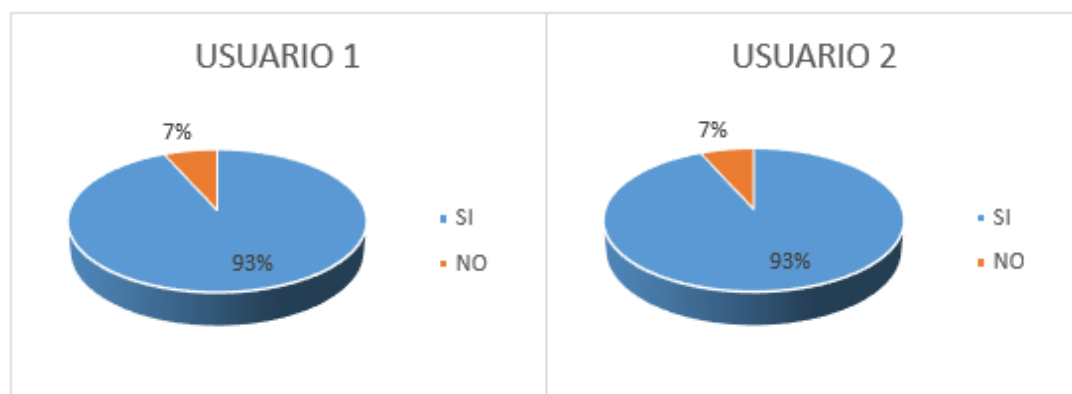


Figura 48: Efectividad de reconocimiento con LBPH.

Fuente: Autores



Figura 49: Reconocimiento a Usuario 1.

Fuente: Autores.



Figura 50: Reconocimiento a Usuario 2.

Fuente: Autores.

Con el algoritmo LBPH se aprecia que la efectividad es bien alta, el 93% con 14 aciertos frente a un solo error, por lo que esta es la técnica más efectiva es esta última. Cabe recalcar que para hacer cada prueba, previamente fue entrenado con la última base de datos que se mencionó anteriormente.

4.3.2.2. Efectividad de reconocimiento para usuarios no autorizados.

Para esta etapa, los sujetos de prueba serán personas que no están permitidas para acceder a poner en marcha y conducir el vehículo, estas pueden ser sujetos que estén o no estén en la base de datos de rostros. Para comprobar la efectividad de estas pruebas, el sistema al detectar los rostros de estas personas, no permitirá activar el relé que alimenta a la bomba de combustible y activará las alarmas que se encuentran dentro del vehículo. De igual manera estas pruebas se realizaran utilizando las tres técnicas de reconocimiento facial.

La prueba se realiza a 30 personas, a cada una de ellas con las tres técnicas de reconocimiento facial y de esto se observara el porcentaje de error al momento de negar el acceso para encender al vehículo.

Tabla 9: Negación de acceso a usuarios no permitidos.

Fuente: Autores

Resultado	Algoritmo	
	Niega el acceso	Permite el acceso
Eigenface	21	9
Fisherface	20	10
LBPH	26	4

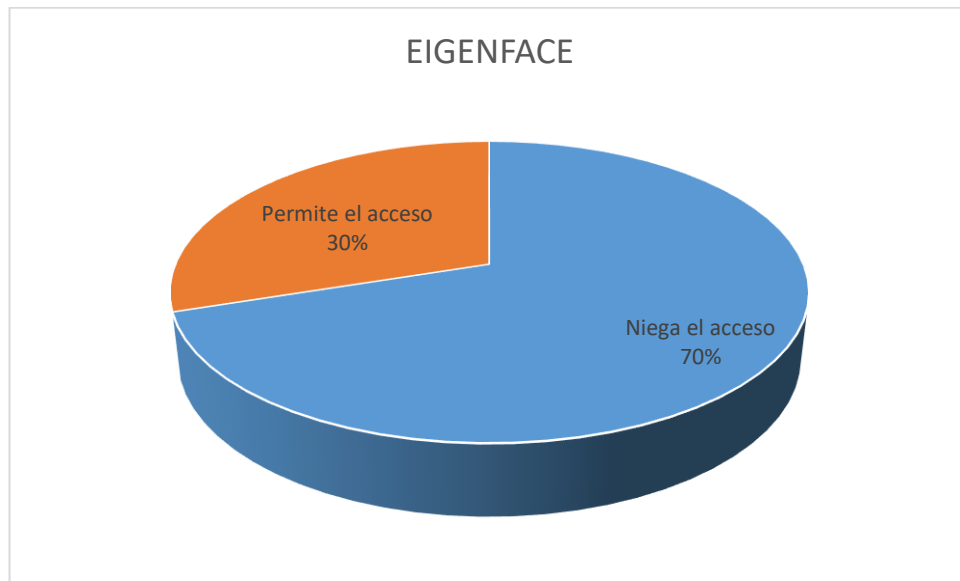


Figura 51: Usuarios no permitidos con Eigenface.

Fuente: Autores.

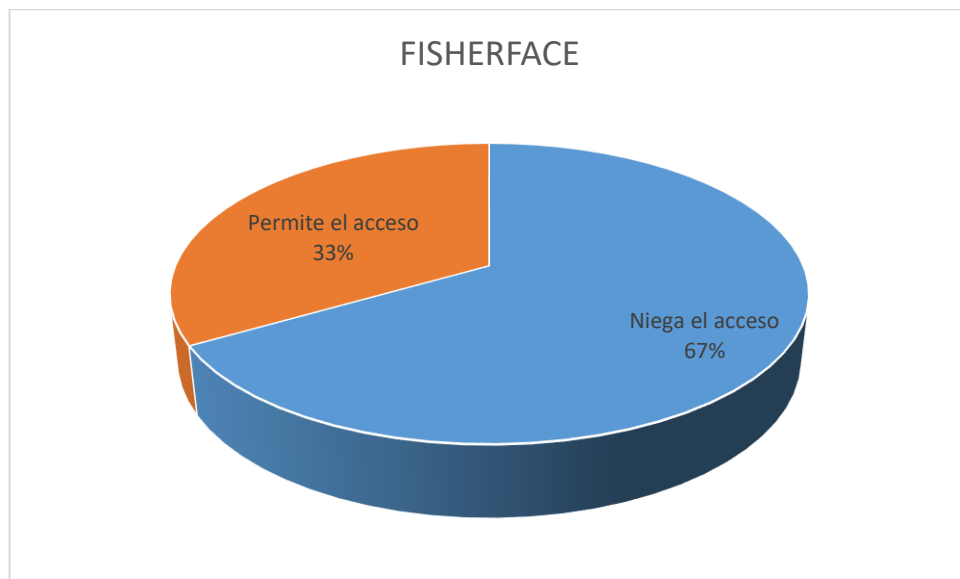


Figura 52: Usuarios no permitidos con Fisherface.

Fuente: Autores.

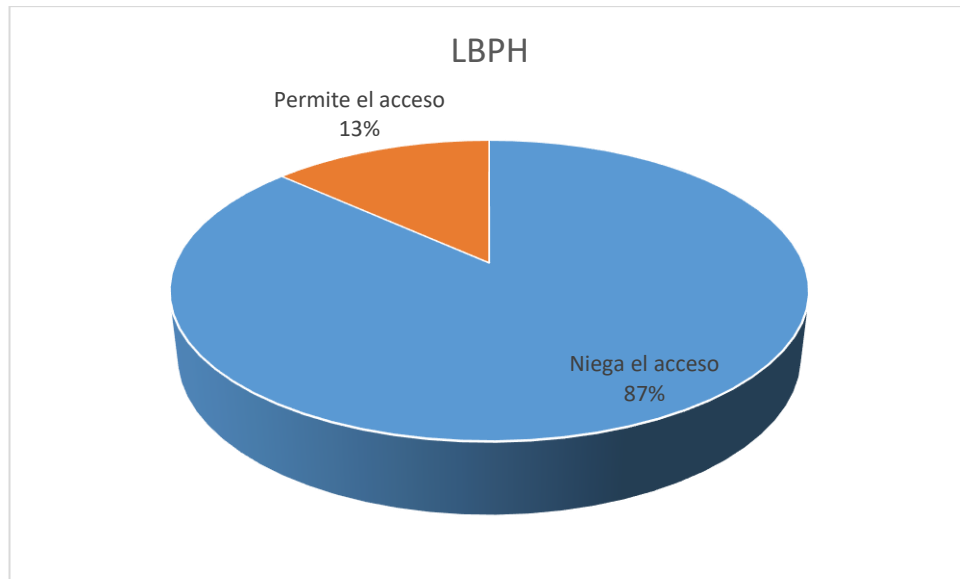


Figura 53: Usuarios no permitidos con LBPH.

Fuente: Autores.

Los resultados anteriores muestran claramente que la mayor efectividad para rechazar a personas no autorizadas se logra con el algoritmo LBPH, pues de los sujetos a prueba a cuatro permitió en acceso, pero después de los 10 segundos programados para una nueva revisión o nuevo reconocimiento, a fueron rechazados por el sistema y no les permitió encender el vehículo y a los otros 26 el sistema los rechazo sin ningún problema.



Figura 54: Rechazo a usuarios no permitidos.

Fuente: Autores

4.3.2.3. Efectividad de acceso con la clave de usuario.

El sistema cuenta con un teclado matricial 4x4, anteriormente se explica la programación de este. En este caso la clave de acceso es “1234”, al digitar esta combinación, seguido por pulsar “start”, de inmediato se enciende el led amarillo y simultáneamente se escucha claramente que el relé cierra, energizando la bomba de

combustible permitiendo encender al vehículo sin importar el reconocimiento facial. Al digitar una clave errónea, se apaga el led amarillo y se desenergiza el relé, cortando el paso de combustible hacia el motor y el vehículo queda en ese instante con seguridad antirrobo por reconocimiento facial.

Por tanto, el uso de una clave de acceso de emergencia o al prestar el vehículo a otro usuario, es totalmente efectivo.

4.3.3. Verificación de encendido de la bomba de gasolina.

Al reconocer al usuario permitido o al haber digitado la clave correctamente, el relé se cierra y energiza a la bomba. Prueba de esto es comprobar el voltaje se salida del 87 del relé que va conectado hacia el positivo de la bomba de gasolina,

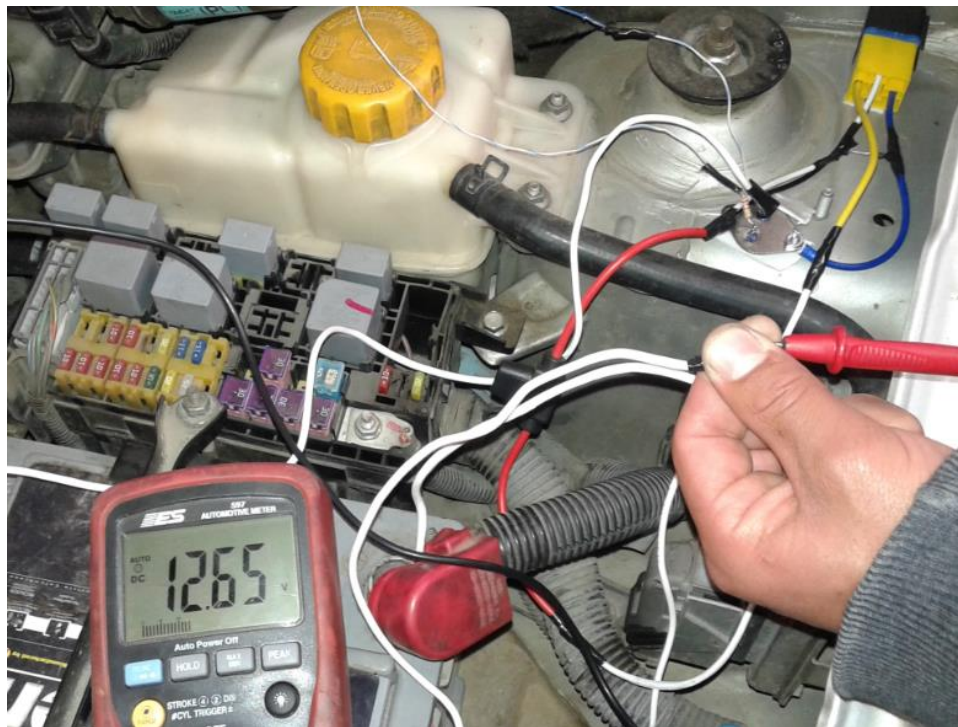


Figura 55: Voltaje de salida del pin 87 del relé.

Fuente: Autores

4.3.4. Pruebas de funcionamiento.

Para esta prueba, el usuario permitido se somete al reconocimiento facial, este lo reconoce y permite encender el vehículo. De inmediato se sube un sujeto desconocido quitándole el lugar al usuario reconocido, simulando un robo con el auto encendido, después de los 10 segundos programados, el sistema vuelve a realizar la fase de reconocimiento y es aquí donde el sistema de seguridad rechaza a este nuevo usuario,

des energiza la bomba de gasolina y al cabo de 15 segundos se apaga el vehículo, este tiempo es el que tarda el motor en consumir el combustible almacenado en las cañerías después de des energizarse la bomba de gasolina, sin poder encenderlo nuevamente, hasta presionar el botón de reinicio.

Lo cual hace que el sistema de seguridad antirrobo vehicular por reconocimiento facial sea efectivo.

CONCLUSIONES.

Para el proyecto, se logra realizar una justificación clara y concisa del porque se va a realizar este trabajo, analizando la situación actual sobre el robo de vehículos ya sea dentro o fuera de la ciudad de Cuenca, para esto se acudió a los datos del Instituto Nacional de Estadísticas y Censos (INEC) que muestran una baja considerable en los robos de vehículo en los últimos 10 años.

Se logró realizar una recolección de información sobre los sistemas de seguridad convencionales que existen actualmente para los vehículo, los precios varían según el tipo o efectividad que ofrecen el sistema, de estos existen varios tipos, las marcas de vehículos construyen sus propias sistemas de seguridad para sus unidades, pero se les puede modificar o aumentar la efectividad incluyendo en su sistema algún tipo de seguridad adicional.

El reconocimiento facial es uno de los métodos bastante utilizados para acceder a ordenadores, lo cual es bastante confiable por su bajo índice de errores, esto se logra a través de varios métodos o algoritmos de Visión artificial como es el OpenCV, que es una librería específicamente para reconocimiento facial pues es bastante útil también para reconocimiento de patrones, nariz, boca, ojos y aspectos relevantes o característicos del rostro de una imagen con las técnicas de reconocimiento que son: Eigenface, Fisherface y LBPH.

El nivel de confiabilidad del sistema de reconocimiento facial es mayor durante el día ya que se puede tener una mejor iluminación del rostro, mientras que en la noche la luz presente en el ambiente es escasa dificultando el reconocimiento.

La implementación del sistema de seguridad por reconocimiento facial en el vehículo Chevrolet Aveo ayuda a tener un mejor control de utilización del automotor ya que es eficiente, alcanza todas las expectativas del propietario y puede ser operado por cualquier persona previa a una ligera instrucción de su funcionamiento.

La cámara web se colocó a la altura del parasol izquierdo sin interferir su despliegue, a 45 cm aproximadamente del rostro del conductor

Según las pruebas realizadas, el algoritmo LBPH es el que menor tiempo tarda en entrenar el sistema (74,4 segundos), y en la fase de reconocimiento (36,4 segundos); en comparación con los algoritmos Eigenface y Fisherface.

En cuanto al reconocimiento para usuarios autorizados, se tiene un porcentaje de efectividad del 73% con el algoritmo Eigenface, del 53% con el Fisherface y del 93% con el LBPH; concluyendo así que el algoritmo con mayor porcentaje de efectividad en el reconocimiento facial es el LBPH.

RECOMENDACIONES.

Es indispensable recolectar información de fuentes confiables para tener en cuenta el grado de gravedad que se encuentra el delito de robo de vehículos en la ciudad o fuera de ella.

Es importante elegir los materiales y métodos realmente eficaces para resaltar un trabajo de investigación, que las placas como Raspberry, Arduino y demás compuestos electrónicos estén en perfecto estado para evitar el desarmado de estos y dañarlos permanentemente y garantizar el trabajo que se está realizando.

Utilizar siempre ropa adecuada para trabajar con sistemas electrónicos y utilizar herramientas certificadas para tales trabajos y así evitar accidentes.

Se recomienda actualizar la base de datos según sea necesario, sobre todo cuando existan cambios en la apariencia de las personas registradas en la base de datos para así garantizar el reconocimiento del rostro, con el mismo número de imágenes o fotografías para no hacer lento al programa o evitar errores.

Se debe procurar que la fuente de luz refleje directamente en el rostro cuando el reconocimiento facial sea efectuado por la noche, ya que de otra manera, las imágenes serán oscuras presentado resultados erróneos.

Se recomienda implementar el sistema primero en un laboratorio, de esta manera si se comete algún error se lo rectifica de inmediato y realizar diagramas didácticos para guiarse durante la implementación en el vehículo. Mas no así cuando ya está implementado en el vehículo, para resolver algún problema de conexión se tendría que desarmar varios accesorios para encontrar el problema.

Cuando ya esté el sistema implementado en el vehículo, hacer pruebas periódicamente durante horas específicas del día, cambiando las posturas, gestos y distancias, para así tener una mayor efectividad durante el reconocimiento.

BIBLIOGRAFÍA.

- Carrasco Ochoa, J. A. (2013). *Reconocimiento de Patrones*. México: Instituto Nacional de Astrofísica Óptica y Electrónica. Recuperado el 25 de octubre de 2016, de <https://ccc.inaoep.mx/~ariel/recpat.pdf>
- Delbracio, M., & Mateu, M. (2006). *Trabajo Final de Reconocimiento de Patrones: Identificación utilizando PCA, ICA y LDA*. Recuperado el 21 de octubre de 2016, de http://iie.fing.edu.uy/investigacion/grupos/biometria/proyectos/patrones/RecPat_MM.pdf
- Esparza Franco, C., & Tarazona Ospina, C. (2015). *FACIAL RECOGNITION BASED ON EIGENFACES, LBPH AND FISHERFACES IN THE BEAGLEBOARD-xM*. Unidades Tecnológicas de Santander - UTS, Facultad de Ciencias Naturales e Ingenierías, Grupo de Investigación en Control Avanzado GICAV. Recuperado el 25 de octubre de 2016, de http://www.unipamplona.edu.co/unipamplona/portaIG/home_40/recursos/05_v25_30/revista_26/01052016/21.pdf
- Hernández, R. G. (2010). *ESTUDIO DE TÉCNICAS DE RECONOCIMIENTO FACIAL*. Barcelona: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Departamento de Procesado de Señal y Comunicaciones. Recuperado el 22 de octubre de 2016, de http://upcommons.upc.edu/bitstream/handle/2099.1/9782/PFC_RogerGimeno.pdf
- Vázquez López, M. Á. (2014). *Sistema de Reconocimiento Facial Mediante Técnicas de Visión Tridimensional*. Guanajuato: Centro de Investigaciones en Óptica, A.C. Recuperado el 21 de octubre de 2016, de <http://biblioteca.cio.mx/tesis/15950.pdf>
- Aguilar, C. A. (17 de Julio de 2016). *MOTORBIT*. Recuperado el 20 de Octubre de 2016, de <http://motorbit.com/mejores-y-novedosos-sistemas-de-seguridad-para-tu-auto/?pais>
- Álvarez Corrales. (2013). *Prototipo de sistema piloto para control de acceso basado en reconocimiento de rostros*.
- Arguello Fuentes, H. (2011). *Recognition systems based on the facial image*. Facultad de Ingenierías Físico-Mecánicas, Universidad Industrial de Santander.
- Belhumeur, P., & Hespanha, J. (2010). *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection*. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. Obtenido de <http://www.cs.columbia.edu/~belhumeur/journal/fisherface-pami97.pdf>
- Chichizola, F., De Giusti, A., & Naiouf, M. (2014). *'Eigenfaces de Imagen Reducida' para el Reconocimiento Automático de Rostros*. Facultad de Informática. Universidad Nacional de La Plata. Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/22881/Documento_completo.pdf?sequence=1
- El Tiempo. (11 de Mayo de 2016). *El Tiempo Cuenca*. Recuperado el 25 de Octubre de 2016, de <http://www.eltiempo.com.ec/noticias/sucesos/9/352512/cifras-reflejan-indices-de-delincuencia-en-la-ciudad>

- G.E. (2012). *Aplicación práctica de la visión artificial en el control de procesos industriales*. Barcelona: Gobierno de España. Ministerio de Educación.
- Guerrero, D. (11 de marzo de 2012). *Reconocimiento Facial. Pasado, presente y futuro*. Recuperado el 11 de noviembre de 2016, de <http://www.diegoguerrero.info/tag/reconocimiento-facial/>
- Guevara, M. L., & Echeverry, J. D. (2008). *Faces Detection in Digital Images Using Cascade Classifiers*. Universidad Tecnológica de Pereira.
- Herrero, I. (2005). *Aspectos de un Sistema de Visión Artificial*. Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial.
- INEC. (14 de Junio de 2016). *Instituto Nacional de Estadística y Censos – Ecuador*. Recuperado el 25 de Octubre de 2016, de <http://www.ecuadorencifras.gob.ec/justicia-y-crimen/>
- INFAIMON. (26 de Mayo de 2014). *INFAIMON VISIÓN ARTIFICIAL*. Obtenido de <http://www.infaimon.com/es/software-de-imagen-seguridad>
- Kanade, T., & Hebert, M. (21 de diciembre de 2015). *People Image Analysis Consortium*. Recuperado el 10 de noviembre de 2016, de <http://www.consortium.ri.cmu.edu/index.php>
- Llaguno, C. (10 de Mayo de 2016). *Seguros 123*. Recuperado el 25 de Octubre de 2016, de <http://ecuador.seguros123.com/5-mejores-dispositivos-antirrobo/>
- López Pérez, N., & Toro Agudelo, J. J. (2012). *TECNICAS DE BIOMETRIA BASADAS EN PATRONES FACIALES DEL SER HUMANO*. UNIVERSIDAD TECNOLOGICA DE PEREIRA.
- Moreno Díaz, A. B. (2004). *Reconocimiento Facial Automático mediante Técnicas de Visión Tridimensional*. Madrid: UNIVERSIDAD POLITÉCNICA DE MADRID. FACULTAD DE INFORMÁTICA. Tesis Doctoral. Recuperado el 21 de octubre de 2016, de <http://oa.upm.es/625/1/10200408.pdf>
- Navas, E. (abril de 2013). *Navas Design*. Recuperado el 11 de noviembre de 2016, de <http://www.eduardonavas.info/2013/04/16/clasificador-haar/>
- Network, D. (2014). *msdn.microsoft*. Recuperado el 09 de noviembre de 2016, de <https://msdn.microsoft.com/es-es/library/dn913079.aspx>
- Ortiz, T. M. (2014). *SISTEMA DE RECONOCIMIENTO FACIAL AUTOMATIZADO PARA EL CONTROL DE ASISTENCIA DE RECURSOS HUMANOS*. Loja.
- Ottado, G. (2010). *Reconocimiento de caras: Eigenfaces y Fisherfaces*. Obtenido de https://eva.fing.edu.uy/file.php/514/ARCHIVO/2010/TrabajosFinales2010/informe_final_ottado.pdf
- Salazar Espinoza, C. F. (2016). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD PARA UN AUTOMÓVIL CON AUTENTICACIÓN POR RECONOCIMIENTO FACIAL UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL*. ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO., Riobamba.
- Serratos, F. (2014). *Reconocimiento de las personas por rasgos de la cara*. Universitat Oberta de Catalunya.
- Sobrado Malpartida, E. A. (2003). *SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO Y MANIPULACIÓN DE OBJETOS UTILIZANDO UN BRAZO ROBOT*. Lima: PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ. Obtenido de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/68/SOBRADO_EDDIE_VISION_ARTIFICIAL_BRAZO_ROBOT.pdf;jsessionid=6329D67F0BCBAEB5608FABF20355A9A4?sequence=2
- Turismo, M. d. (24 de Junio de 2016). *Ministerio de Turismo. Noticias*. Recuperado

el 20 de Octubre de 2016, de <http://www.turismo.gob.ec/ecuador-mejorar-sus-indices-de-seguridad-ciudadana/>

Urriaga Abad, J. A. (2014). *RECONOCIMIENTO FACIAL*. Madrid.

Yuan, C., & Qi, D. (2009). *Face Recognition Using L-Fisherfaces*. Institute of Information Science Beijing Jiaotong University. Obtenido de http://www.iis.sinica.edu.tw/page/jise/2010/201007_23.pdf

Zhiguang, Y., & Haizhou, A. (2007). *Demographic Classification with Local Binary Patterns*. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, Beijing. Recuperado el 25 de octubre de 2016, de http://media.cs.tsinghua.edu.cn/~ahz/papers/ICB07_demographic.pdf